

**T.C.**  
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**  
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

**GÜÇ HATLARI ÜZERİNDEN İLETİŞİM İLE AKILLI PRİZ**  
**KONTROLÜ VE MEDYA AKTARIMI SAĞLAYABİLECEK**  
**SİSTEM TASARIMI**

**YETİŞKAN ELİAÇIK**  
**YÜKSEK LİSANS TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GEBZE**  
**2014**

**T.C.**  
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**  
**MÜHENDİSLİK ve FEN BİLİMLERİ ENSTİTÜSÜ**

**GÜÇ HATLARI ÜZERİNDEN İLETİŞİM İLE**  
**AKILLI PRİZ KONTROLÜ VE MEDYA**  
**AKTARIMI SAĞLAYABİLECEK SİSTEM**  
**TASARIMI**

**YETİŞKAN ELİAÇIK**  
**YÜKSEK LİSANS TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMANI**  
**DOÇ. DR. ERKAN ZERGEROĞLU**

**GEBZE**

**2014**



**GEBZE YÜKSEK  
TEKNOLOJİ ENSTİTÜSÜ**

**YÜKSEK LİSANS JÜRİ ONAY FORMU**

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 27/01/2014 tarih ve 2014/06 sayılı kararıyla oluşturulan jüri tarafından 12/02/2014 tarihinde tez savunma sınavı yapılan Yetişkin ELİAÇIK 'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

**JÜRİ**

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. ERKAN ZERGEROĞLU

ÜYE

: Doç. Dr. MEHMET GÖKTÜRK

ÜYE

: Doç. Dr. ABDÜLKADİR BALIKÇI

**ONAY**

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
..... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

## ÖZET

Günümüz akıllı evlerindeki arařtırmalar, güvenlik, konfor ve enerji yönetimi alanlarına odaklanmıřtır. Ayrıca, ev içindeki birçok cihazın birbiriyle haberleřmesi, kontrol etmesi nedeniyle akıllı ev sistemleri, multimedya sistemleri ile de haberleřen sistemler haline gelmiřtir. Bu tez çalıřması ile Beaglebone adı verilen gömülü sistem ile mevcut evlerin ev içi elektrik hattı alt yapısında herhangi bir deęiřiklik gerektirmeden, güç hatları üzerinden haberleřme ile akıllı evlere dönüřtürölmesini ve bu sistem ile baęlı cihazların uzaktan kontrolünü saęlayarak enerji tüketimini azaltan, iPLUG isimli bir sistem tasarımı üzerinde çalıřılmıřtır. Tez çalıřması, iPLUG sistemi ile elektrik hattı üzerinden medya transferi, baęlı cihazların kontrolü, enerji tüketimin ölçölmesi ve bu sistemin performansının incelenmesi sonucundaki çıktıların paylaşılması řeklindedir.

**Anahtar Kelimeler: Güç Hattı Haberleřmesi; Beaglebone Gömülü Sistem; Akıllı Ev; iPLUG.**

## SUMMARY

Present research in smart home are focused on security, comfort and energy management. In addition, many devices in the home are control and communicate with each other, as well smart home systems should communicate with other multimedia systems in the home. In this thesis, using the Beagleboned named embedded system, current homes transform into smart homes without any modification in-house powerline infrastructure with power line communication and, supplying with the remote control of the connected devices ensuring that reduces energy consumption system has been working which is called iPLUG. Thesis work includes design, performance analysis and sharing the obtained data of iPLUG system, which has capability of media transfer and power control and consumption of connected devices via power line.

**Keywords: Power Line Communication; Beaglebone Embedded System; Smart Home; iPLUG.**

## TEŐEKKÜR

Tez alıőmam sũresince alıőmama yœn veren ve alıőmalarımnda desteęini esirgemeyen tez danıőmanım Sn. Do. Dr. Erkan Zergeroęlu'na,

Tezimin donanımsal kısımlarının hazırlanmasında ve ortak fikirle yola ıktıęım lisans arkadaőım œner Hatipoęlu'na,

Ayrıca manevi olarak beni her daim destekleyen ve yanımda olan eőim Yũk. Mũh. Asya Eliaık'a ve aileme teőekkũr ederim.

# İÇİNDEKİLER

	<b><u>Sayfa</u></b>
ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
SİMGELER ve KISALTMALAR DİZİNİ	iv
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	xi
1. GİRİŞ	1
2. GÜÇ HATLARI ÜZERİNDEN HABERLEŞME İLE AKILLI EV ALT YAPISININ OLUŞTURULMASI	3
2.1. iPLUG Nedir?	3
2.2. iPLUG ile Yapılanlar Ve Yapılabilecekler	5
3. IPLUG SİSTEM TASARIMI VE ÖZELLİKLERİ	7
3.1. Beaglebone'un Teknik Özellikleri ve İşletim Sistemi	7
3.1.1. Beaglebone Angstrom Dağıtımının Güncellenmesi ve Kernel Derleme İşlemleri	8
3.2. TFT Ekran, USB Klavye Fare Teknik Özellikleri ve İşletim Sistemine Adapte Edilmesi	9
3.3. JAVA Geliştirme Ortamının Kurulması	12
3.4. JMF Kurulumu ve Testi	13
3.5. PLC ile Medya Gönderimi Test Ortamı	14
4. iPLUG KONTROL PANEL İLE IPLUG PRİZ KONTROL SİSTEMİ	15
4.1. iPLUG Priz Kontrolü İçin Arayüz ve Kontrol Sistemi Tasarımı	15
4.1.1. iPLUG Sistemi Arayüz Görünümleri ve Özellikleri	15
4.1.1.1. Ana Ekran Görünüm ve Özellikleri	15
4.1.1.2. Kontrol Ekranı Görünüm ve Özellikleri	16
4.1.1.3. Ayarlar Ekranı Görünüm ve Özellikleri	17

4.1.1.4. Test Ekranı Görünüm ve Özellikleri	18
4.2. iPLUG Sistemi Arayüzü İçin Kod Tasarımı	19
4.2.1. Arayüzün Tasarlanması İçin Projenin Oluşturulması	19
4.2.2. iPLUG Priz Bilgilerinin Tutulması İçin Oluşturulan Veri Yapısı ve Dosya Yapısı	21
4.2.3. Ana Ekran Tasarımı ve Ekranlar Arasındaki Geçiş	23
4.2.4. iPLUG Priz ile MCU Arasındaki Seri Haberleşme ve Oluşturulan Protokol	24
5. ANALİZ	26
5.1. PLC Adaptör ile Medya İletim Testi	26
5.2. IPLUG MCU ile IPLUG Priz Kontrol Sistemi	27
6. SONUÇLAR	30
KAYNAKLAR	32
ÖZGEÇMİŞ	34
EKLER	35

# SİMGELER ve KISALTMALAR DİZİNİ

## Simgeler ve Açıklamalar

### Kısaltmalar

GYTE	:	Gebze Yüksek Teknoloji Enstitüsü
IEEE	:	Institute of Electrical and Electronics Engineers
MCU	:	Ana Kontrol Birimi
OC	:	Over Current
OV	:	Over Voltage
PLC	:	Power Line Communication
PM	:	Protective Mode
TFT	:	ThinFilm Transistor

# ŞEKİLLER DİZİNİ

<b><u>Sekil No:</u></b>	<b><u>Sayfa</u></b>
2.1: iPLUG sisteminin genel blok şeması.	4
3.1: Beaglebone sistem görünümü.	7
3.2: da8xx_lcdc_platform_data bbtoys35_pdata veri yapısı.	10
3.3: Bilinen lcd konfigürasyonu (known_lcd_panels) özelliği.	10
3.4: İşletim sistemi çekirdek yapılandırması.	11
4.1: iPLUG kontrol sistemi ana ekran arayüzü.	16
4.2: iPLUG kontrol ekran arayüzü.	17
4.3: iPLUG ayarlar ekran arayüzü.	18
4.4: iPLUG test ekranı arayüzü.	19
4.5: iPLUG test ekranı arayüzü.	20
4.6: iPLUG tasarım ekranı görüntüsü.	20
4.7: iPLUG çift katmanlı yapının tasarım görünümü.	23
5.1: Gürültü bazında paket sıklığı karşılaştırması.	29

# TABLolar DİZİNİ

<b><u>Tablo No:</u></b>	<b><u>Sayfa</u></b>
3.1: Powerline ethernet adaptör teknik özellikleri.	14
4.1: Priz durumuna ait açıklama tablosu.	22
4.2: iPLUG prizlere ait XML'in yapısı.	22
4.3: UART4 portu yapılandırma tablosu.	24
5.1: Farklı test ortamlarındaki paket gönderimi sonuçları.	26
5.2: Yüksüz durumda hesaplanan performans değerleri.	28
5.3: Gürültü bazında performans tablosu.	29

# 1. GİRİŞ

Otomasyon sistemleri için karar alma ve uygulama on temel seviyeden oluşmaktadır. İlk olarak, bilgisayar herhangi bir yönlendirmede bulunmaz; insan tüm karar alma ve uygulama işlemlerini yapar. İkinci olarak, bilgisayar bir dizi karar/aksiyon önerisinde bulunur. Üçüncü olarak, karar sayısını azaltarak daha az öneride bulunabilir. Dördüncü olarak, tek bir öneride bulunabilir. Beşinci olarak eğer kullanıcı onaylarsa öneride bulunduğu aksiyonu uygular. Altıncı olarak, otomatik olarak belirlediği aksiyonu almadan önce kullanıcıya iptal etmek için zaman verir. Yedinci olarak, otomatik olarak aldığı aksiyonu uygular ve kullanıcıyı uygun şekilde bilgilendirir. Sekizinci olarak, tüm kararı ve aksiyonu bilgisayar alır ve sadece kullanıcı istediğinde bilgilendirir. Dokuzuncu olarak, kullanıcıyı yalnızca bilgisayar bilgilendirmesi gerektiğini düşündüğü zaman bilgilendirir. Son olarak, bilgisayar tüm karar alma ve uygulama işlemlerini otomatik olarak yapar ve kullanıcıyı gözardı eder. Buradaki otomasyon seviyesi birden ona doğru artmaktadır. Bu adımlar için temel gruplama 4 temel seviyede yapılmaktadır; bilginin toplanması, bilginin analiz edilmesi, kararın seçilmesi, gerekli aksiyonun uygulanması şeklindedir. Otomasyon sistemi için seviyelendirme bu dört temel seviyedeki otomasyona göre sınıflandırılmaktadır. Her bir adımda farklı düzeyde otomasyon olabilir [18].

Ev otomasyon sistemleri 5 farklı kategoriye ayrılmaktadır. Birinci seviye sistemler, basit olarak lambaların veya diğer uygulamaların bir alıcı verici yardımıyla uzaktan kontrol sisteminden oluşmaktadır. Buna örnek olarak bir radyo verici ile açılıp kapatılabilen veya bir zamanlayıcı ile açıp kapatılan sistemler gösterilebilir. Bu sistemler ucuzdur ancak aynı anda sadece bir cihazı kontrol edebilen sistemlerdir. İkinci seviye kontrolde aynı anda birden fazla cihaz kontrol edilebilen sistemlerden oluşmaktadır. Örneğin, tek bir dokunuş ile ortamdaki ışık kısılır, kısık bir müzik açılır ve su ısıtılmaya başlanabilir. Bu tür sistemlere ait cihazlar özelleşmiş sistemlerdir ve genelde programla için farklı bir cihaz veya bir bilgisayar gerekmektedir. İkinci seviye otomasyon sistemi iyi bir verici sistemidir ancak sistemle ilgili herhangi bir bilgi almaz. Üçüncü seviye otomasyon sisteminde ise koşullu programlama yapılabilir. Örneğin, belirli bir zaman aralığında garaj kapısı onbeş dakikadan uzun süre açık ise alarm ses sistemini devreye sok şeklindedir. Bu seviye sistemler genellikle bilgisayar kontrollüdür veya özelleşmiş işlemciler ile

kontrol edilmektedir. Dördüncü seviye otomasyon sisteminde ise web sitesi, ses tanımlama sistemi, anlık mesajlaşma, e-posta, telefon veya diğer sistemler üzerinden etkileşimli olarak kontrol yapılabilmektedir. Beşinci seviyede ise öğrenebilen bir sistem olması gerekmektedir. Bu sistem için ev içindeki yaşam biçimine göre kendi kendini programlayan bir sistem olması gerekmektedir. Birinci seviye sistemler ucuz iken dört ve üstü seviyedeki sistemler pahalı ve komplike bir hal almaktadır [17].

Bir Avrupa Birliği projesi olan Buttler FP7 projesi temelde otomasyon sistemlerini bir araya getirip bağlantılı hale getirmeyi hedefleyen bir projedir. Bu kapsamda “Internet of Things” kavramının bir parçası olan akıllı ev/ofis sistemlerinin beşinci seviyede sistemler hale getirilmesi ve diğer akıllı sistemler ile entegrasyonunu içermektedir. Buttler projesi kapsamında ev otomasyon sistemleri için çalışma alanı üç başlık altında toplanmıştır. Güvenlik: kişisel güvenlik ile birlikte, gaz kaçağı, su sızdırması yangın gibi durumlarda sistemin bilgilendirmesi. Konfor: evdeki cihazların (lamba, anahtarlı sistemler), kontrolü, ortam parametrelerinin (oda sıcaklığı, su sıcaklığı, hava akışı, vb.) kontrolü, eğlence sistemlerinin (multimedya, oyun, vb.) sistemlerinin kontrolü. Enerji yönetimi: bu kapsamda enerji tüketiminin ve üretiminin anlık olarak izlenmesini ve aksiyon alınmasını içermektedir. Bu proje mevcut sistemlerde de sıkça kullanılan “Zigbee” kablosuz haberleşme protokolünü kullanmaktadır [19].

Kablosuz sistemler ev içi gürültüyü arttırdığından, mevcut kablolu sistemler ise altyapıda ek kablolama değişikliği gerektirdiğinden esnek bir sistem tasarlanmak istenmiştir. Elektrik hattı üzerinden haberleşme akıllı evler için yeni bir teknolojidir ve ek kurulum maliyeti getirmeden ve kablosuz haberleşmenin getirdiği gürültü maliyetini ortadan kaldıran bir teknolojidir.

Bu tez kapsamında, kredi kartı büyüklüğünde işletim sistemi gömülü olarak kullanılabilen ve birçok özelleşmiş giriş-çıkış birimli bir mini bilgisayar olan Beaglebone ile elektrik hattı üzerinden geniş bant ve kısa bant haberleşme ve kontrol sisteminin gerçekleştirilmesi ve mevcut ev ortamındaki performansı üzerinde çalışılacaktır.

## 2. GÜÇ HATLARI ÜZERİNDEN HABERLEŞME İLE AKILLI EV ALT YAPISININ OLUŞTURULMASI

Akıllı evler günümüzde, evdeki lamba kontrolü ve parlaklığının ayarlanması, cihazların priz üzerinden açılıp kapatılması, panjur motorunun çalıştırılması gibi uygulamaları içermektedir. Bunun için özelleştirilmiş sistemler bir arada kullanılmakta ve kısıtlı olarak sadece ev içindeki bir cihaz ile kontrol edilmektedir. Kullanılan çözümler ya esnek değildir ya da çok pahalıdır. Ayrıca, bazı elektronik cihazlar üzerinde, mobil cihazlar ile uzaktan kontrol edilip ayarlama yapılabilmektedir ancak bunlar genelde cihaz tabanlı olmaktadır. Ayrıca bu çözümlerin tamamı kolay kullanıma yönelik tasarlanmıştır. Ancak enerji tasarrufuna odaklanılmamıştır. Bu nedenle kullanımı dar bir alana sıkışmış ve sadece anlık değişimler ile kontrol sağlanmıştır. Bu tez kapsamında evde kullanılan enerji harcayan basit cihazların, uzaktan kontrolünün ve enerji tasarrufunun sağlanarak cihazların farklı zaman aralıklarında çalışabilir hale getirilmesi öngörülmüştür.

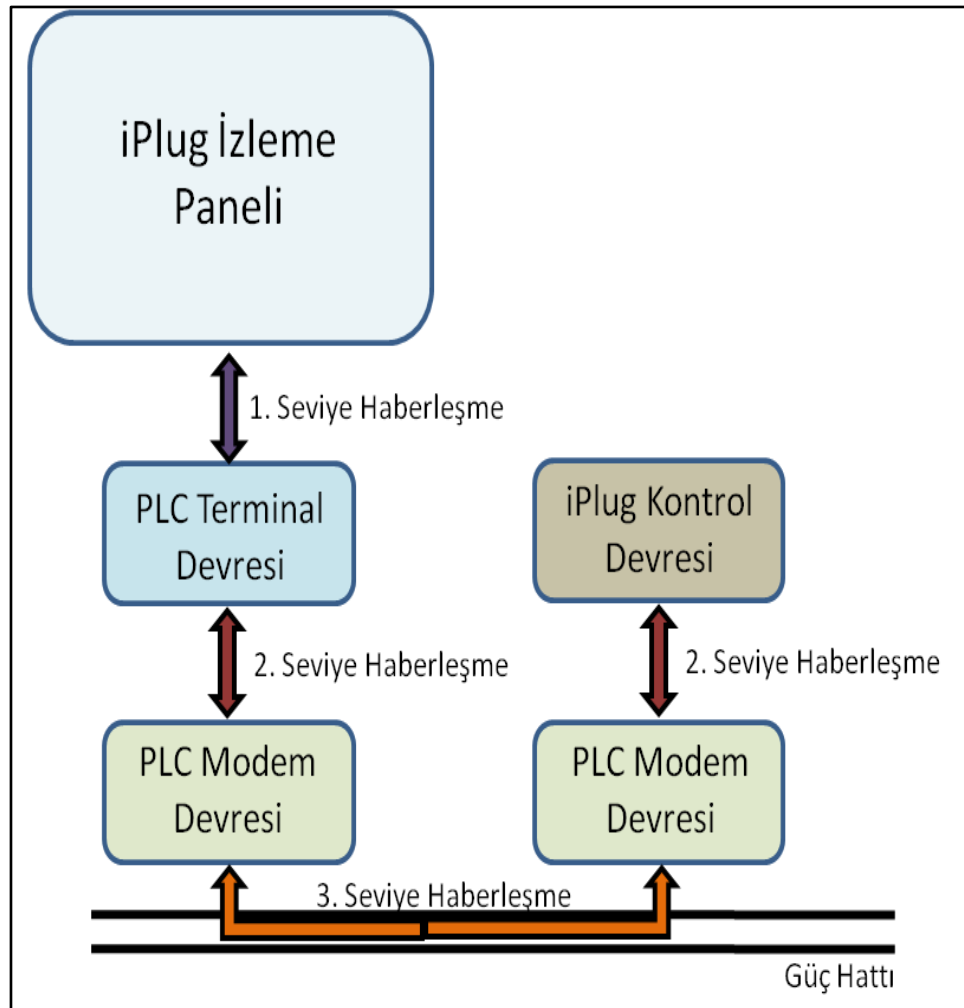
### 2.1. iPLUG Nedir?

iPLUG, akıllı evler için güç hatları üzerinden haberleşme(ing: PLC) çözümü, ev içi elektrik hattı üzerinden haberleşmeyi ve cihazların kontrolünü içermektedir. iPLUG temelde iki birimden oluşmaktadır. Sistemin genel bir blok şeması Şekil 2.1 'de görülmektedir.

Birincisi iPLUG priz, PLC üzerinden haberleşen ve bağlı olan cihazın enerji tüketimini anlık olarak ölçebilen ve uygun protokol ile ölçtüğü değerleri aktarabilen; bağlı cihazları kapatıp açarak kontrolünü sağlayan birimdir. iPLUG priz kendi içerisinde iki alt bölümden oluşmaktadır iPLUG kontrol devresi ve PLC modem devresi. PLC modem devresi enerji hattı üzerinden haberleşmeyi sağlayan birimdir. iPLUG kontrol devresi için gelen bilgiyi bitlere çevirirken iPLUG kontrol devresinden gelen veriyi de PLC ile enerji hattına basar.

İkincisi ise iPLUG ana kontrol birimi, daha sonraki bölümlerde “iPLUG MCU” olarak adlandırılacaktır, tüm iPLUG prizlerinin uzaktan kontrolünü (açma-kapama), ölçüm değerlerinin toplanmasını, prizlerin zamana bağlı açma kapama

işlemlerinin yapılmasını, programlanmasını ve mevcut enerji tüketimini ve bağlı cihazların durumunun izlenmesini sağlayan birimdir. iPLUG MCU kendi içerisinde üç alt bölümden oluşmaktadır. iPLUG izleme paneli kullanıcı arayüzünün ekranda gösterilmesini, prizlerin kontrol edilmesini ve prizlerin kontrolü için gerekli protokol üzerinden haberleşme verisini oluşturan birimdir. PLC terminal devresi iPLUG izleme panelinden aldığı veriyi PLC modeme aktaran birimdir. Böylece iletilecek veri uygun hızda PLC modem devresine aktarılmış olur. PLC modem devresi ise gelen veriyi güç hattına basan ve güç hattından gelen veriyi terminal devresine ileten birimdir. iPLUG sistemine ait detaylı teknik bilgi Bölüm 3 'te anlatılmaktadır.



Şekil 2.1: iPLUG sisteminin genel blok şeması.

## 2.2. iPLUG ile Yapılanlar ve Yapılabilecekler

iPLUG sistemi ile iPLUG prizleri için bağı cihazların, enerji ölçümlerinin anlık olarak iletilebilmesini, enerjisinin uzaktan kesilmesi ve açılabilmesini, aşırı akım çekmesi durumunda kullanıcının uyarılmasını, aşırı voltaj gelmesi durumunda kullanıcının uyarılmasını sağlamaktadır.

iPLUG MCU ile iPLUG prizlerine, özelleştirilmiş isimler verilerek daha rahat erişim ve anlaşılır bir yönetme ortamı oluşturulması sağlanmıştır. iPLUG prizlerinin MCU üzerinden durumunun görüntülenmesi, iPLUG prizlerine bağı olan cihazların açma kapama komutunun gönderilmesi sağlanmıştır. Ayrıca, iPLUG prizlerine bağı cihazlarının anlık, haftalık, aylık ve yıllık olarak enerji tüketiminin incelenmesi için gerekli altyapı oluşturulmuştur.

iPLUG MCU ile enerji tasarrufu amacıyla belirli modlar için özelleşmiş olarak iPLUG prizlerine bağı olan cihazların zaman ayarlı olarak açılıp kapatılması için gerekli altyapı oluşturulmuştur. iPLUG MCU ile Ethernet üzerinden veya geniş bant PLC sistemi üzerinden görüntü aktarılması ve bunun ev içinde uygun bir geniş bant PLC sistemine aktarılması için gerekli altyapı oluşturulmuştur.

Üzerindeki Ethernet ile ev içi ağı üzerinden veya geniş bant ağı üzerinden uzaktan ev içindeki cihazların uzaktan kontrolü sağlayan altyapı oluşturulmuştur. Böylece mobil cihazlar üzerinden veya internet üzerinden evdeki cihazların kontrolü için bir altyapı oluşturulmuştur.

Bu özellikler ile iPLUG prize bağı cihazların bekleme konumunda çektiğı enerji sıfıra indirilerek enerji tasarrufu sağlanmıştır. iPLUG prize bağı cihazların kapalı konumunda iken enerji değışiminden korunması sağlanmıştır. Bunun yanında akıllı evlerde temel olarak sağlanan anlık olarak cihazların kontrolü sağlanmıştır.

Ayrıca, evde bulunan korunmaya muhtaç bireylerin elektrik kaynaklı kazalardan korunması ile ev içinde iş kazalarının önlenmektedir.

iPLUG prizlerine bağı klima, kombi gibi ısıtma soğutma sistemlerinin özellikle çalışan aileler için efektif olarak kullanılması için arayüz oluşturulmuştur.

iPLUG sistemi ile çamaşır makinesi ve bulaşık makinesi gibi yüksek enerji kullanımına neden olan cihazların 22:00 – 06:00 gibi zaman aralığında çalıştırılmasını zorunlu kılmak için programlanması ve enerji maliyetinin düşürülmesi sağlanabilmektedir.

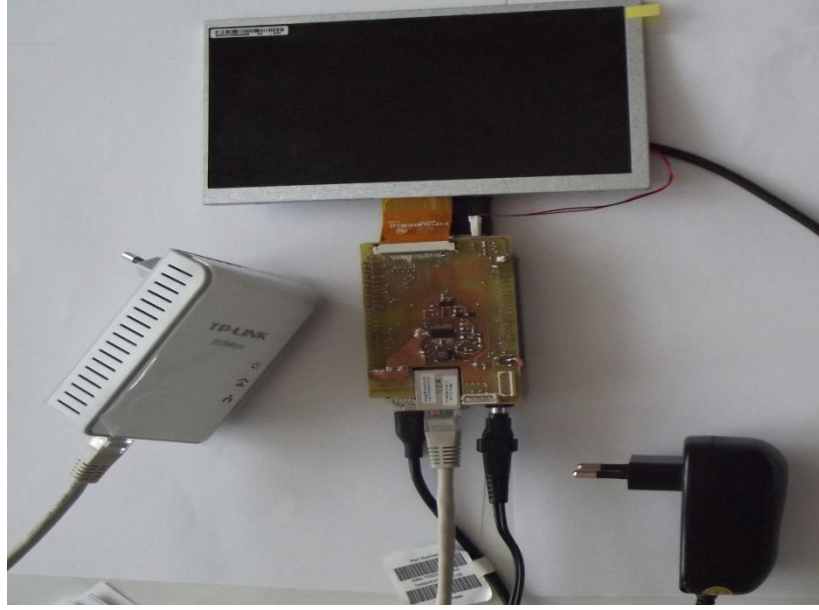
iPLUG sistemi entegre edilen iş yerleri için uzaktan enerji hatlarının açma-kapaması yapılarak enerji tasarrufu sağlanması ve toplam enerji tüketim maliyetini ölçerek maliyetler önceden hesaplanabilmesi için ortam oluşturulmuştur.

Yukarıdaki özelliklerin dışında mevcut iPLUG prizlere ek sensörler eklenerek ve mevcut tüketimler analiz edilerek mevcut özellikleri geliştirilebilir. Yaşlı bireyler için enerji kullanımını analiz ederek eğer normalden daha farklı bir kullanım şekli görüldüğünde erken uyarı sistemi oluşturulabilir. Evdeki cihazların kullanımında oluşan farklılığa göre uyarı sistemi oluşturularak evdeki güvenlik ihlalleri için uyarı sistemi oluşturulabilir. Kamera gibi görüntü aktarım cihazları ile alınan veri akıllı telefonlara ve herhangi bir web arayüz ile kullanıcıya sunulabilir böylece evdeki güvenlik ve çocukların canlı izlenmesi için ortak bir platform sağlanabilir. Kapı merceği yerleştirilecek bir kamera sistemi ile kapı çalındığında kimin geldiğini görebilecek ve bu görüntüyü kaydedebilecek bir sistem oluşturulabilir. Elektrik, su ve doğalgaz dağıtıcıları için uzaktan sayaç okuma sistemleri geliştirilebilir.

iPLUG sistemi kullanımı kolay, esnek, açık kaynak kodlu ve açık mimarili, mevcut enerji altyapı ile kolaylıkla kullanılabilen bir sistemdir.

### 3. IPLUG SİSTEM TASARIMI VE ÖZELLİKLERİ

Deneme ortamı; deneme kiti olarak kullanılan Beaglebone, TFT ekran, USB klavye fare, PLC Ethernet adaptöründen oluşmaktadır. Şekil 3.1 'de oluşturulan medya aktarımı için deneme ortamının bir görüntüsü bulunmaktadır.



Şekil 3.1: Beaglebone sistem görünümü.

#### 3.1. Beaglebone'un Teknik Özellikleri ve İşletim Sistemi

Beaglebone, kredi kartı büyüklüğünde bir Linux bilgisayardır; Ethernet üzerinden İnternete bağlanabilir, Android 4.0 ve Ubuntu işletim sistemi uyumlu yazılımlar kullanılabilir [1], [2]. Birçok giriş çıkış birimi olduğundan ve gerçek zamanlı analiz yapmak için elverişli olan işlemcisi olduğu için tercih edilmiştir. Ayrıca Beaglebone tak-çalıştır genişletilme birimleri ile de satılmaktadır böylece tüm girişleri uygun şekilde kullanan bir mini bilgisayar olarak sunulmaktadır.

Ekran ve PLC birim sürücü kartları için tez çalışmasına özel bir giriş çıkış genişletme birimi tasarlanmış ve işletim sistemi birim özelliklerine bağlı olarak yeniden programlanmıştır.

Beaglebone gömülü sistemi birçok işletim sistemi ile uyumlu çalışabilmektedir. Üzerinde 4GB microSD kart Angstrom işletim sistemi dağıtımı ile

gelmektedir. Bunun yanında Cloud9 IDE Node.JS w/ BoneScript kütüphanesi ile grafik arayüz üzerinden javascript ile geliştirme yapılabilmektedir. Bu deneme kiti ayrıca Android, Ubuntu ve diğer Linux işletim sistemleri ile de çalıştırılabilmektedir. Bu projede, mevcut esnekliği, internet üzerinden kolayca paket yüklenebilir olması ve Linux tabanlı olması nedeniyle Angstrom işletim sisteminin kullanılması olmuştur. MicroSD kart üzerinde gelen Angstrom dağıtımında GNOME masaüstü de gelmektedir, aynı zamanda sisteme USB'den Ethernet'e dönüştürülebilir yerel ağ üzerinden bilgisayar ve Beaglebone arasında haberleşme sağlanabilmektedir. Ayrıca Beaglebone bir kart okuyucu gibi kullanılarak microSD kart içindeki bilgiler okunabilmekte ve değiştirilebilmektedir. microSD kart ile gelen işletim sistemi olan Angstrom kullanıldığı için dağıtımla birlikte gelen tüm özellikler de kullanılabilir durumda olacaktır.

### **3.1.1. Beaglebone Angstrom Dağıtımının Güncellenmesi ve Kernel Derleme İşlemleri**

Beaglebone Angstrom dağıtımını üzerindeki Cloud9 IDE, micro-USB Ethernet adaptör konfigürasyonu, GNOME desktop arayüzü ile birlikte geldiği için bu dağıtım imajı kullanılacaktır. Ancak TFT ekran, USB klavye fare için işletim sistemi çekirdeğinin (ing: kernel) yeniden derlenmesi gerekmektedir.

Angstrom Linux OpenEmbedded derleme sistemini kullanmaktadır, derleme işlemi "Bitbake" ile yapılmaktadır. Bu derleme işlemi farklı çekirdekler ve farklı gömülü sistemler için kök (ing: root) dosya sistemini oluşturmak için kolaylık sağlamıştır. Basitçe, makine tipini ve oluşturulacak imaj tipi verildiğinde gerekli tüm çapraz derleme (ing:cross-compilation) araçları oluşturulmaktadır. Ancak, çalışma kapsamındaki gibi işletim sistemi çekirdeği üzerinde bazı yapılandırma gerektiğinde süreç biraz zorlu hale gelmektedir. Bu nedenle derleme süreci aşağıda detaylı olarak anlatılmaktadır.

Derleme işlemi Ubuntu işletim sistemi kurulu olan bir bilgisayar üzerinde yapılmıştır. Derleme işlemi için takip edilmesi gereken temel adımlar 1) Gerekli kaynak kodların indirilmesi ve paketlerin Ubuntu işletim sistemi üzerine kurulması, 2) Beaglebone için yapılandırmanın yapılması, 3) Çekirdek kaynak kodunun (ing: kernel) ve önyükleme (ing: boot) dosyasının oluşturulması, 4) Oluşturulan dosyaların

Beaglebone SD kartına kopyalanması ile sistemin yeni çekirdek ve sürücüler ile başlatılması şeklindedir.

Çekirdek ve aygıt sürücülerini yeniden derlenerek Beaglebone imajında gelen mevcut özellikleri kullanabileceğimiz bir ortam oluşturmuş olmaktadır. Bununla birlikte, Angstrom işletim sistemine ait kaynak kodu da derleyerek çekirdek üzerinde istediğimiz değişik yapılabilmektedir.

### **3.2. TFT Ekran, USB Klavye Fare Teknik Özellikleri ve İşletim Sistemine Adapte Edilmesi**

Bu bölümde kaynak kodu elde edilmiş olan çekirdek (ing: kernel) dosyaları üzerinde TFT Ekran ve USB klavyenin tanıtılması için gerekli olan işlemler anlatılmaktadır. Beaglebone özelleştirilmiş bir çekirdek imajıyla geldiği için sadece belirli ekran (TFT) ve USB klavye/fareler ile çalışacak şekilde derlenmiştir. Bu nedenle kullanılan USB klavye/fare ve TFT ekranı mevcut işletim sistemine adapte etmek için çekirdeğin bu aygıtların sürücülerini tanımlayıp kullanılabilir hale getirilmesi gerekmektedir.

TFT ekran sürücülerini çekirdek dosyaları üzerinde “frame buffer” içerisine gömülmüştür. Bu sürücü “da8xx-fb” olarak tanımlanmıştır.

TFT ekran için gerekli olan “frame buffer” sürücünün kullandığı değişkenler “board-am335xevm.c” dosyası içerisinde belirlenmektedir. Bu değişkenler LCD panel ile ilgili özellikleri olduğu ve “data sheet” üzerinde tanımlı değerlere göre belirlenen değerlerdir. Bu dosya içindeki “lcdc\_init” rutini “da8xx-fb.c” dosyası içerisindeki “am33xx\_register\_lcd()” rutinini çağırarak bilgileri açılışta sürücüye gönderiyor. Beaglebone için özelleştirilen ve genişletilme birimiyle (ing: extension module) uyumlu olarak çalışması için koyulan LCD panel özellikleri üzerinde değişiklik yapılarak ilk açılışta tanımlama ne olursa olsun bu sürücü ve ekran özellikleriyle sistemi ayağa kaldırması sağlanmaktadır. Bu değişiklik “da8xx\_lcd\_platform\_data bbtoys35\_pdata” üzerinde yapılmıştır ve son hali Şekil 3.2 ‘deki gibidir.

```

struct da8xx_lcdc_platform_data bbttoys35_pdata = {
    .manu_name      = "BBToys",
    .controller_data = &bbttoys35_cfg,
    .type           = "CDTech_S035Q01",
};

```

Şekil 3.2: da8xx\_lcdc\_platform\_data bbttoys35\_pdata veri yapısı.

LCD panel ile ilgili özellikler “da8xx-fb.c” dosyası içerisinde belirtilmektedir. "CDTech\_S035Q01" için olan özellikler kullandığımız TFT panele uygun olarak değiştirilmiştir. Böylece açılışta zorunlu olarak bu özellikler ile ekranı sürmeye başlayacaktır. Burada temel olarak yapılan değişiklik ekran genişliği, yüksekliği, saniyedeki çerçeve sayısı gibi değerlerin TFT ekranın özelliğine uygun olacak şekilde güncellenmesinden oluşmaktadır. LCD yapılandırılmasına ait veri yapısının tüm değerleri Şekil 3.3’deki gibidir.

```

o [4] = {
    /* CDTech S035Q01 */
    .name = "CDTech_S035Q01",
    .width = 320,
    .height = 240,
    .hfp = 58,
    .hbp = 21,
    .hsw = 47,
    .vfp = 23,
    .vbp = 11,
    .vsw = 2,
    .pxl_clk = 8000000,
    .invert_pxl_clk = 0,
}

```

Şekil 3.3: Bilinen LCD konfigürasyonu (known\_lcd\_panels) özelliği.

Aygıt sürücüleri kaynak kodunda yapılan değişiklikten sonra bu değişikliklerin derlenmesi ve işletim sistemi çekirdeği tarafından kullanılması için çekirdek yapılandırılması (ing: kernel configuration) gerçekleştirilmelidir. İşletim sistemi çekirdeği bu özellikler aktifleştirilmeden derlenirse ekran sürücüsü yüklenmemektedir. Çünkü ekranın kullanılıp kullanılmayacağına karar mekanizması

işletim sistemi çekirdek kodunun içine gömülmüş haldedir. Bunun için gereçli değişiklikler yapıldıktan sonra yeniden derleme işlemi yapılmalıdır.

İşletim sistemi çekirdek kaynak kodu değişmediği için eski bilgileri kullanabiliriz, bunun için “./home/user/.oe/environment-angstromv2012.05” dosyası çalıştırılmalıdır. Daha sonra işletim sistemi çekirdeği tarafından kullanılmasını belirteceğimiz aygıt sürücülerini “• MACHINE=Beaglebone ./oebb.sh bitbake virtual/kernel -c menuconfig” komutunun çıktısı sonucunda Şekil 3.4 ‘deki gibi yapılandırılmalıdır [3].

```
Device Drivers --->
  Graphics support --->
    <*> Support for framebuffer devices --->
      <*> DA8xx/OMAP-L1xx Framebuffer support
        (4) Consistent DMA memory size (MB)

Console display driver support --->
  <*> Framebuffer Console support
    [*] Bootup logo --->
      [*] Standard black and white Linux logo
      [*] Standard 16-color Linux logo
      [*] Standard 224-color Linux logo

Device Drivers --->
  Graphics support --->
    [*] Backlight & LCD device support --->
      <*> Lowlevel Backlight controls
      <*> TLC59108 LCD Backlight Driver
```

Şekil 3.4: İşletim sistemi çekirdek yapılandırması.

USB Klavye fare için tüm markaların seçilmesi farklı üreticiler için destek sağlaması açısından önemlidir.

İşletim sistemi çekirdek kaynak kodunda yaptığımız değişikliklerin derlenmesi ve hatasız şekilde derlendiğinden emin olmak için “MACHINE=Beaglebone ./oebb.sh bitbake virtual/kernel -c compile -f” komutu çalıştırılır.

İşletim sistemi çekirdeği derlendikten sonra gerekli aygıt sürücülerin paketlenmesi için “MACHINE=Beaglebone ./oebb.sh bitbake virtual/kernel” komutu çalıştırılmalıdır.

Oluşturulan dosyalar Beaglebone’a transfer edilerek TFT ekran ve USB Klavye Fare ‘nin sürülmesi için gerekli sürücüler sisteme atılarak bu aygıtların çalışması sağlanmıştır.

### **3.3. JAVA Geliştirme Ortamının Kurulması**

Beaglebone üzerinde koşturulan işletim sistemi başarılı bir şekilde çalıştırıldıktan sonra kod geliştirme ortamının kurulması çalışması başlatılmıştır. Yapılan bütün geliştirmeleri Eclipse derleyici olan bir bilgisayar üzerinde yapıp Beaglebone üzerinde derleme yapmaya gerek kalmadan çalışılabilir hale getirilmiştir. Böylece derleme zamanından kazanılıp çapraz-platform(cross-platform) bir derleme ve çalışma ortamı sağlamış olunur.

JAVA kodunun çalıştırılması için Beaglebone Angstorm üzerine aşağıdaki paketlerin yüklenmesi gerekmektedir [4].

- opkg install openjdk-6-common
- opkg install openjdk-6-java
- opkg install openjdk-6-jre
- opkg install openjdk-6-jdk
- opkg install openjdk-6-vm-zero

Beaglebone üzerindeki kurulum tamamlandıktan sonra bilgisayarımız üzerindeki Eclipse’de bir proje yaratıp derleme işlemlerini gerçekleştirebiliriz. Burada dikkat edilmesi gereken önemli yapılandırma “Eclipse Remote System Explorer” ile ilgili işlemlerin yapılmasıdır [4], [5]. Böylece USB-Ethernet arayüzü üzerinden doğrudan Beaglebone’a bağlanarak yazdığımız JAVA kodlarını Beaglebone üzerinde tutarak proje oluşturmuş olunur. Böylece derlediğimiz kodlar sonucunda oluşan obje dosyaları Beaglebone üzerinde çalıştırılabilir ve hata ayıklama yapılabilir. Bu işlem ile JAVA ile yapabildiğimiz tüm grafik arayüz

tasarımları, JMF gibi video transfer kütüphanelerini istediğimiz şekilde derleyebilir ve Beaglebone deneme kiti üzerinde çalıştırabiliriz.

### 3.4. JMF Kurulumu ve Testi

Java Media Framework API (JMF) ile ses, video ve diğer zaman-tabanlı medya bileşenlerini uygulamalara eklenmesi sağlanmaktadır. JMF paketi birçok görüntü ve ses formatındaki dağıtımı yakalayabilir, oynatabilir, dağıtımını ve biçim dönüştürme işlemlerini yapar. Bu API multimedya geliştiriciler için ölçeklenebilir ve çok platformlu ortamlarda geliştirme yapmasını sağlar [6], [7], [11], [12].

JMF kütüphanesinin kurulumu için sadece gerekli “.jar” dosyalarını Beaglebone üzerinde herhangi bir klasöre atılması gerekmektedir. Bilgisayar üzerinde kurulumu için ise [6] sitesi üzerinden uygun platforma yönelik kurulum dosyaları indirilmelidir.

JMF testi için kullanılan kodlar [7], [8] kaynaktan olup; sadece görüntü transferi için olan “VideoTransmit.java” dosyası içindeki kod görüntüyü gömülü sisteme aktarmak için kullanılmıştır. Bu kod hem bilgisayar üzerindeki diskten hem de internet üzerindeki bir kaynaktan görüntüyü yakalayıp argümanlarında belirtilen IP adresi ve port için RTP ile video dağıtımını yapılmaktadır.

Gömülü sistem için ise yine aynı kaynaktan belirtilen [8] “AVReceive2.java” kodu bilgisayar üzerinde derlenmiştir. Bu kod argüman olarak IP/Port bilgisini almaktadır. Tez yazımı sırasında tasarlanan gömülü sistem sadece JPEG/RTP türündeki kodlayıcı/kodçözücüleri (ing:codec) içeren görüntüleri başarılı bir şekilde alabilmektedir. JMF kütüphanesi eski olduğu için MPEG/RTP türü codecler için hata alınmaktadır. Ancak bu testler için herhangi bir engel oluşturmamaktadır. Beaglebone üzerinde kodu çalıştırırken yapılması gereken önemli şey kodu çalıştırmak için JMF kütüphanesine ait “.jar” uzantılı dosyaların kütüphane dosyası olarak belirtilmesi gerekmektedir.

Bu aşamaya kadar gerçekleştirilen tüm testler Ethernet kablosu bağlı iken PLC birimi olmadan yapılabilmektedir.

### 3.5. PLC ile Medya G6nderimi Test Ortamı

PLC ile test iin piyasada kolaylıkla bulunabilecek TP-Link marka TL-PA210 model ‘‘Powerline Ethernet Adaptor’’ kiti kullanılmıřtır. Bu adapt6r6n 6zellikleri Tablo 2 deki gibidir [10].

Tablo 3.1: Powerline ethernet adapt6r teknik 6zellikleri.

Donanım 6zellikleri	
Standartlar ve Protokoller	HomePlug AV, IEEE802.3, IEEE802.3u
Aray6z	10/100Mbps Ethernet baęlantı noktası
Fiř Tipi	EU, US, UK, AU
D6ęmeler	Eřleřtirme D6ęmesi
Aęırlık	103 Gram
LED G6stergeler	PWR,PLC,ETH
Boyutları ( G x D x Y )	3.6 x 2.2 x 1.2 in.(93×56×30 mm)
G6 T6knetimi	< 3W
Aralık	300M ev ii

## **4. iPLUG KONTROL PANEL İLE iPLUG PRİZ KONTROL SİSTEMİ**

iPLUG kontrol paneli ile iPLUG prizlere ait enerji tüketimi, temel kontrol işlemi olan açıp kapama, prizler için isim verilmesi gibi işlemleri içeren ekranlardan oluşmaktadır. Ayrıca test ortamı için kolaylık sağlaması açısından protokole ait komutların olduğu test ekranı da tasarlanmıştır.

### **4.1. iPLUG Priz Kontrolü İçin Arayüz ve Kontrol Sistemi Tasarımı**

Bu bölümde iPLUG priz kontrolü için tasarlanan arayüz ve kontrol sistemlerine ait tasarım detaylarına yer verilmektedir.

#### **4.1.1. iPLUG Sistemi Arayüz Görünümleri ve Özellikleri**

Bu bölümde iPLUG sistemine ait arayüz görünümleri ve sağlanan özellikler belirtilmektedir. Bu bölüm içerisinde ana ekran, kontrol ekranı, ayarlar ekranı ve test ekranlarına ait arayüz görünümleri, bu ekranlardaki fonksiyonel özellikler ve tasarım ile ilgili detaylar bulunmaktadır.

##### **4.1.1.1. Ana Ekran Görünüm ve Özellikleri**

iPLUG sisteminin ana ekran temel görünümü Şekil 4.1 de sunuluyor. iPLUG ana ekranda görüldüğü gibi, mevcut tanımlı iPLUG prizlerinin statüsü ve isimleri tablo şeklinde verilmektedir. Herhangi bir priz seçildiğinde priz üzerinden alınan en son voltaj ve akım bilgisi ekranda yazılmaktadır.

İzleme butonu tez kapsamında bir test menüsü ile oluşturulan protokolün test edilmesi için desteklenen komutlara ait, butonları ve argümanları almaktadır. Kontrol ekranı ile mevcut prizlerin kontrol birimi üzerinden elektrik hattı üzerinden açılıp kapatılması sağlanmaktadır. Ayrıca işlem sonrası son durum da tablo üzerinde gösterilmektedir.

Ayarlar ekranı ile mevcut tanımlı prizler için prizlerin tanımlanması için Arama (ing:Search) ve prizlerin isimlendirilmesi gerçekleştirilebilmektedir. Ayrıca,

iPLUG prize verilen ismin kaydedilmesi için Kayıt (ing:Save) ve kayıt işleminden vazgeçilebilmesi için İptal (ing:Cancel) butonlarını içermektedir.

Modlar ekranı, tez kapsamında gerçekleştirilmemiş olup belirli modlarda prizlerin zaman ayarlı olarak açılıp kapanmasını sağlayacak modların ve zaman aralıklarının girilmesi için tasarlanmış bir ekrandır.

Kontrol ve ayarlar ekranından ana ekrana dönmek için sol alt köşedeki iPLUG butonuna basılması gerekmektedir. Böylece herhangi bir ekrandan ana ekrana geçiş sağlanmıştır.



Şekil 4.1: iPLUG kontrol sistemi ana ekran arayüzü.

#### 4.1.1.2. Kontrol Ekranı Görünüm ve Özellikleri

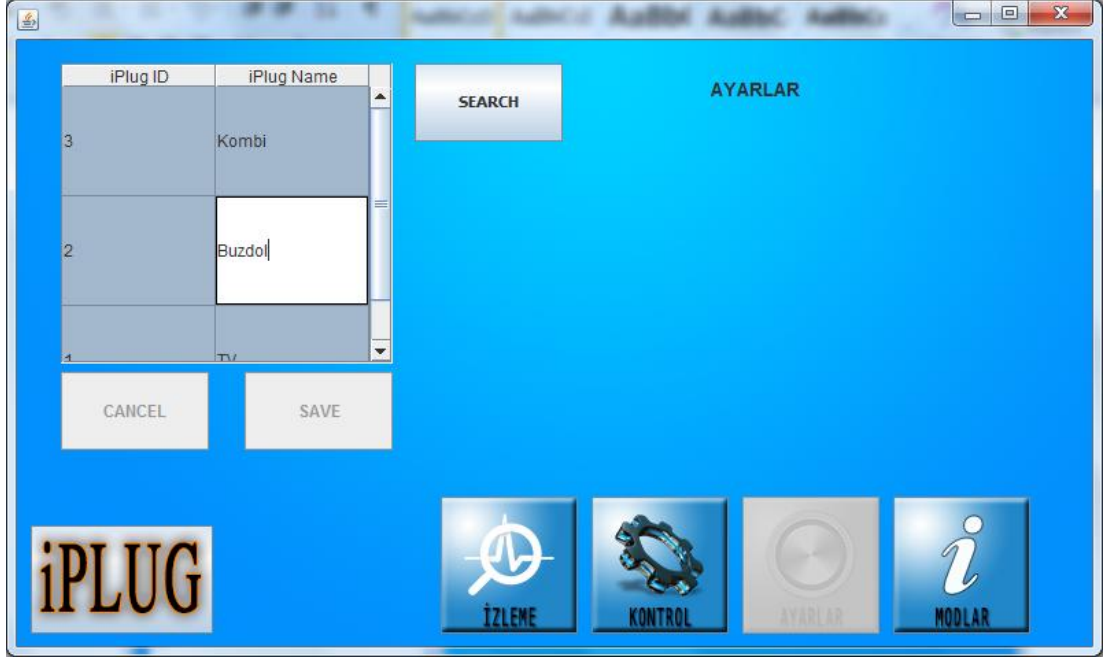
iPLUG sisteminin kontrol ekranı temel görünümü Şekil 4.2 deki gibidir. Kontrol ekranı tanımlı prizlerin uzaktan açılıp kapatılmasını sağlayan ekrandır. Herhangi bir priz tablo üzerinden seçildiğinde; kontrol sistemi prize belirlenen protokole uygun şekilde açma veya kapama sinyali gönderir ve priz son durumu ekranda gösterilir.



Şekil 4.2: iPLUG kontrol ekran arayüzü.

#### 4.1.1.3. Ayarlar Ekranı Görünüm ve Özellikleri

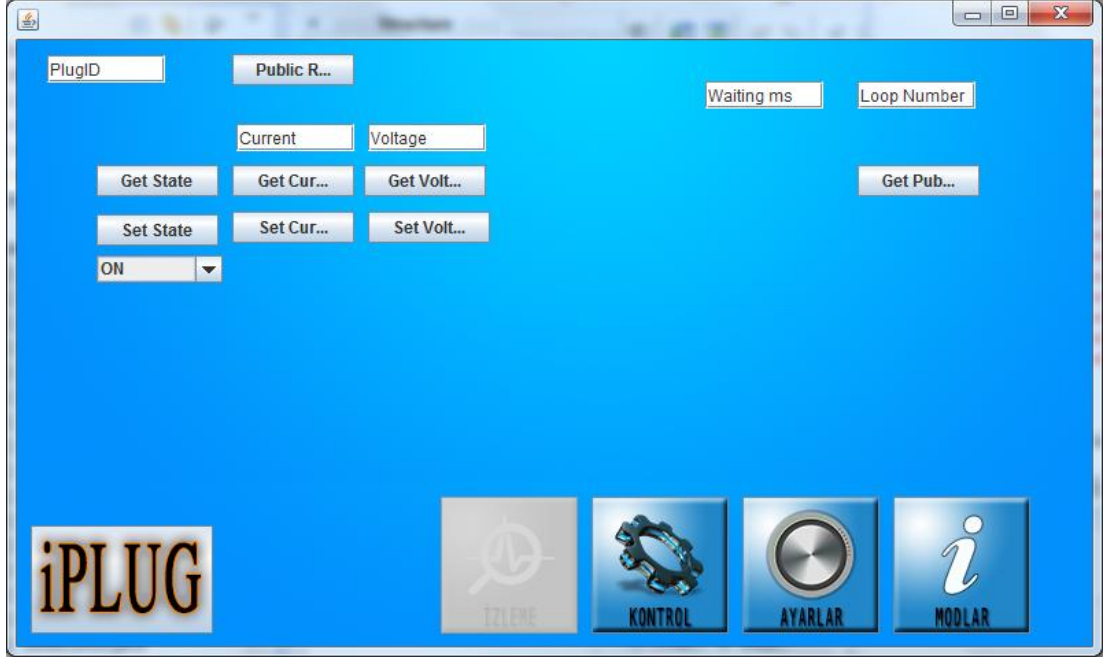
iPLUG sisteminin ayarlar ekranı temel görünümü Şekil 4.3 de sunuluyor. Ayarlar ekranı, tanımlı prizlere ait kimlik (ing:ID) bilgisini gösterir ve bu prize ait fonksiyonel, tanımlayıcı ismin verilmesi için bir tablo sunar. Değişiklik yapıldığında bu değişikliğin sisteme kaydedilmesi için Kayıt (ing:SAVE) veya vazgeçilmesi için İPTAL (ing:CANCEL) butonları koyulmuştur. Ayrıca tanımlı prizlerin bulunabilmesi için ve topolojiye eklenmesini sağlamak için Arama (ing:SEARCH) butonu ekrana yerleştirilmiştir. Ancak arama butonuna ait işlem tez kapsamında gerçekleştirilmemiştir.



Şekil 4.3: iPLUG ayarlar ekran arayüzü.

#### 4.1.1.4. Test Ekranı Görünüm ve Özellikleri

iPLUG sisteminin test ekranı temel görünümü Şekil 4.4 de sunulmaktadır. Test ekranı mevcut protokolün desteklediği komutları test etmek ve art arda komut göndermek için tasarlanmıştır. Bu ekranda iPLUG prizlere ait kimlik (ing:ID), akım ve voltaj değerleri alınmaktadır. Ayrıca seri şekilde statü değerlerinin, akım ve voltaj değerlerinin okunması için bekleme süresi ve döngü sayısı alınarak test ortamı için arayüz oluşturulmuştur.



Şekil 4.4: iPLUG test ekranı arayüzü.

## 4.2. iPLUG Sistemi Arayüzü İçin Kod Tasarımı

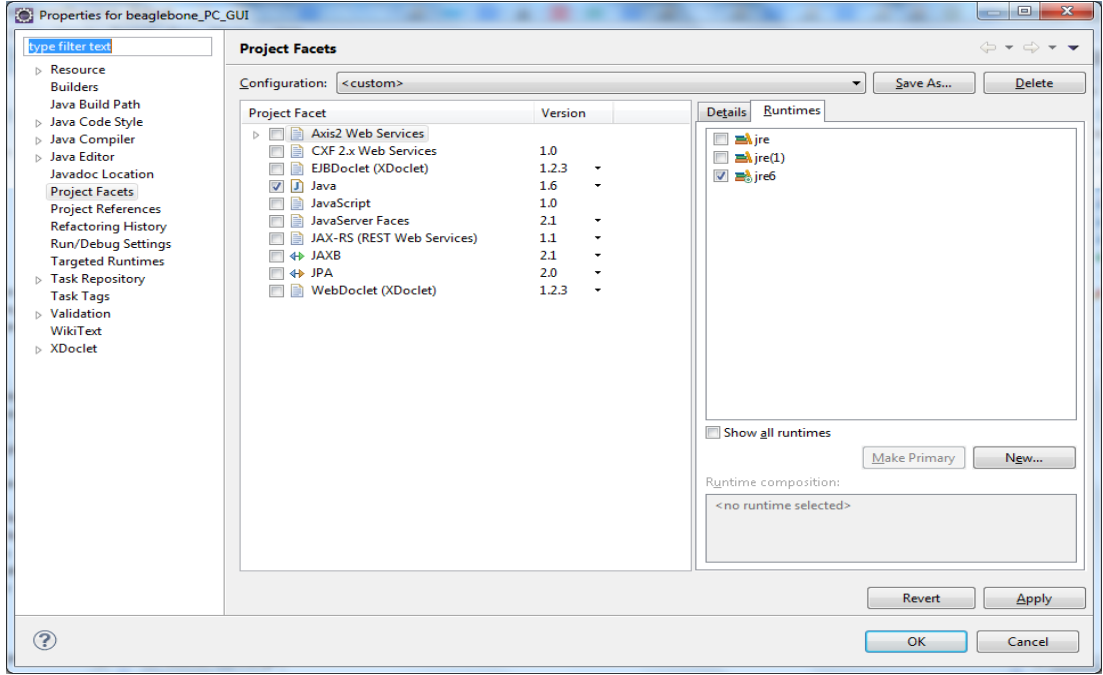
Bu bölüm, iPLUG Sistemi için arayüzün geliştirilmesi, ekran üzerinde gösterilen verilerin toplanması, ekranlar arasında aktarılması, kaydedilmesi ve priz ile haberleşme için oluşturulan veri yapıları ve kod detaylarını içermektedir.

### 4.2.1. Arayüzün Tasarlanması İçin Projenin Oluşturulması

Arayüz tasarımı için Eclipse üzerinde daha önce kullanılan “Remote System Explorer” perspektifi kullanılarak Beaglebone üzerinde oluşturulmuş bir klasör üzerinde proje oluşturulmuştur. Böylece yine kişisel bilgisayar üzerinde derleme işlemi yapıp gömülü sistem üzerinde kod çalıştırılabilmektedir. Ayrıca hedef sistem olmadan bilgisayar üzerinde kod tasarlanıp daha sonra gömülü sistem üzerindeki projeye aynı dosyalar taşınarak derlenip çalıştırılabildiği için tasarımda hız sağlanmıştır. Kişisel bilgisayar üzerinde Eclipse derleyici ve JAVA ile yapılabilen tüm kod tasarımı Beaglebone aktarılabilmektedir. Bu da bize çapraz platform geliştirme ortamı sağlamaktadır.

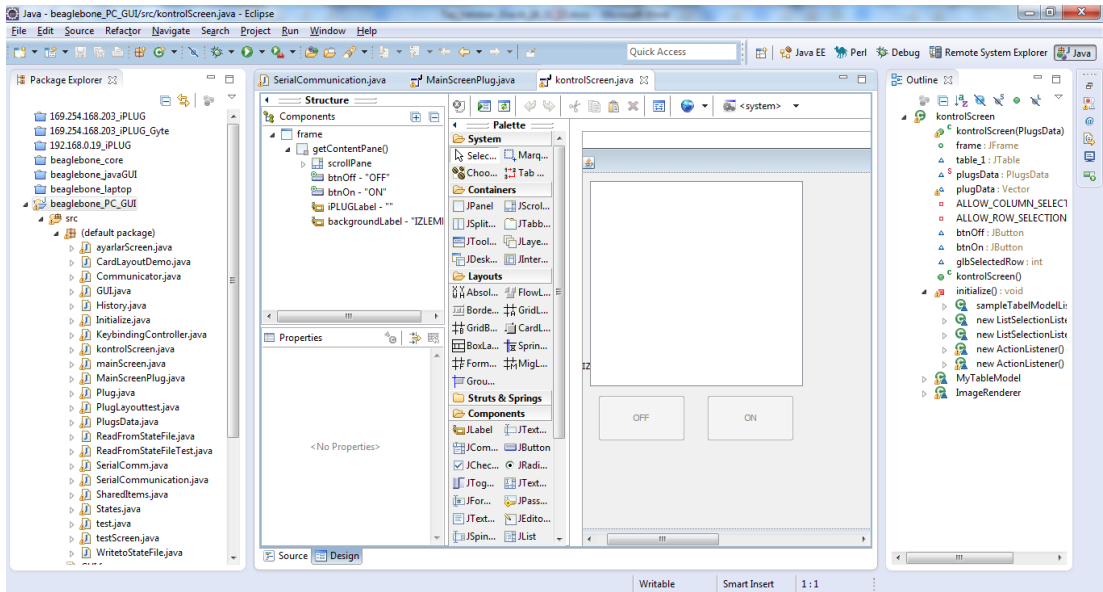
Gömülü sistem üzerindeki klasör üzerinde “Remote System Project” oluşturulduktan sonra JAVA perspektif üzerinden kod görsel tasarımın kolay şekilde

yapılabilmesi için “Java Facelet Project” e dönüştürülür. Şekil 4.5 de yapılan ayarlar görülmektedir.



Şekil 4.5: iPLUG test ekranı arayüzü.

Böylece sürükleyip bırakarak java form oluşturularak arayüz tasarlanabilmektedir. Ayrıca bu arayüz Beaglebone üzerinde doğrudan test edilebilmektedir. Tasarım ekranı Şekil 4.6 deki gibidir.



Şekil 4.6: iPLUG tasarım ekranı görüntüsü.

#### 4.2.2. iPLUG Priz Bilgilerinin Tutulması İçin Oluşturulan Veri Yapısı ve Dosya Yapısı

Bu bölümde iPLUG prizler için yapılan tanımları ve son durumuna ait bilgiyi tutmak için kullanılan XML ve veri yapıları anlatılmaktadır.

iPLUG prizlere erişim ve indeksleme priz kimliği(ID) üzerinden olması ve mevcut durumunu ve verilen ismin tutulması için JAVA 'da "Hash Table" kullanılmıştır. Bu hash tablosunu dosyada tutmak için ise XML kullanılmıştır. Böylece MCU üzerinde çalışan program kapatılıp açılrsa bile prizlerin isimleri ve prizlerle ilgili durum bilgisi dosyadan okunarak ekrana getirilebilmektedir. Hash Table ve XML kullanılmasının nedeni deneme ortamı üzerindeki işlemci maliyetini ortadan kaldırmak içindir. Beaglebone üzerinde herhangi bir grafik kartı bulunmadığı için ve deneme ortamında "Gnome Desktop" arayüzü ve JAVA kullanıldığı için sistem sürekli %100 işlemci kullanım oranında çalışmaktadır. Bu nedenle priz verileri priz kimliğine(ID) sine göre hash'lenerek hafızada tutulmaktadır ve arama maliyeti olmadan priz ile ilgili veri eklenip değiştirilebilmektedir. Priz ile ilgili bilgiler "Plug.java" sınıfında tutulmaktadır. Bu bilgiler priz ID, priz ismi ve priz statüsüdür. Priz ID önceden tanımlı olarak priz üzerinde yazılmaktadır. Bu nedenle bu grafik arayüz üzerinden değiştirilememektedir ve hash ID olarak kullanılmıştır. Priz ismi ise arayüz üzerinden değiştirilebilmektedir. Ayrıca prizlerin açık veya kapalı olduğu durumda bilgilerini tutmak için ve tutarlı olması açısından bu bilgi "States.java" sınıfında tutulmaktadır. iPLUG prizi için kullanılan ve prizden gelecek durumlar ve açıklamaları Tablo 4.1'de belirtildiği gibidir.

iPLUG hash tablosunda ise veri iPLUG priz ID dizgi(string) olarak hash'lenir ve bu hash ID'ye sahip prize ait Plug sınıfındaki objesini tutar. Arayüz üzerinde oluşturulan her bir tabloda ekranda gösterilmese de gizli bir kolon olarak priz ID 'si tutulmaktadır. Böylece herhangi bir prizin verisi getirilmek istendiğinde veya değiştirilmek istendiğinde  $O(1)$  zamanda prize ait verilere ulaşılabilir. Oluşturulan hash tablosu "PlugsData.java" sınıfında oluşturulur ve her bir ekran oluşturulurken bu sınıfa ait obje gönderilir böylece bu bilgiler üzerinde yapılan tüm değişiklik tüm ekranlara taşınmış olur.

Tablo 4.1: Priz durumuna ait açıklama tablosu.

DURUM	AÇIKLAMA
ON	Priz açık ve ölçüm yapılmaktadır
OFF	Priz kapalı ve ölçüm yapılmaktadır
DUMMY	Ölçüm yapmayan ve prize takılı cihaza enerji veren durum
IPLUG	Ölçüm yapmaktadır
IDLE	iPLUG priz ilk açıldığında bu moda girer ve MCU dan bir durum bilgisi bekler
OCF	Aşırı akım durumu
OVF	Aşırı voltaj durumu
NA	Prize erişim yoktur
UNKNOWN	Prizden tanımlanamayan bir durum bilgisi gelmiştir

Tablo 4.2: iPLUG prizlere ait XML'in yapısı.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Plugs>
  <Plug id="1">
    <name>TV</name>
    <state>ON</state>
  </Plug>
  <Plug id="2">
    <name>Buzdolabı</name>
    <state>ON</state>
  </Plug>
  <Plug id="3">
    <name>Kombi</name>
    <state>ON</state>
  </Plug>
</Plugs>
```

PlugsData.java sınıfında oluşturulan hash tablosuna ait veriler saklanması için oluşturulan XML'e yazılması ve sınıfa ait obje yaratılırken XML'den veri okunması için gerekli metodlar bu sınıfın içinde tanımlanmıştır. Böylece sadece gerekli görüldüğünde bilgiler XML'e yazılmaktadır ve işlemci üzerindeki giriş-çıkış(IO) bit/sn(bps) maliyeti azaltılmaktadır.

Oluşturulan XML'in yapısı Tablo 4.2 de gösterilmiştir. XML kullanılmasının nedeni tutulan bilgi iPLUG ID ile indekslendiği için erişimi ve veriyi alması daha hızlı

olmaktadır. Ayrıca JAVA dilinde bu işlemi kolaylaştırmak için geliştirilmiş DOM Kütüphanesi bulunmaktadır.

### 4.2.3. Ana Ekran Tasarımı ve Ekranlar Arasındaki Geçiş

Ana ekran ile ayarlar, kontrol gibi ekranlar arasında geçişi sağlamak için JAVA görsel tasarımındaki “content pane” yapısı kullanılmıştır. Şekil 4.7 de görülen bu yapı ile ekrandaki sabit ve değişken bölümler ayrılarak aynı ekran üzerinde farklı ekranların değiştirilerek görüntülenmesi sağlanmıştır ve ekran üzerindeki objeler değiştirilmeden birden fazla ekran birbirinden bağımsız şekilde tasarlanarak butonlarla bu ekranlar çağırılmıştır. Prizler bilgileri için oluşturulan “plugsData” objesi tüm ekranlar için bir constructor argümanı olarak gönderilmiştir. Böylece singleton yapısında tek bir obje diğer tüm ekranlar tarafından erişilebilmekte ve kullanılabilir.



Şekil 4.7: iPLUG çift katmanlı yapının tasarım görünümü.

### 4.2.4. iPLUG Priz İle MCU Arasındaki Seri Haberleşme ve Oluşturulan Protokol

iPLUG MCU ile priz arasındaki haberleşme seri port üzerinden sağlanmaktadır. Beaglebone üzerindeki /dev/ttyO4 portu kullanılarak seri haberleşme sağlanmaktadır. Ancak bu portun yazma ve okuma işlemi için işletim sistemi

üzerinde UART'ın çekirdek dosyası çalışma zamanında yapılandırılması gerekmektedir. Tablo 4.2 de belirtilen dosyaya güncelleme değeri yazılmalıdır böylece işletim sistemi çalışma zamanı esnasında UART4 seri haberleşme portu olarak yazma ve okuma işlemleri için yapılandırılmış olacaktır.

Tablo 4.3: UART4 portu yapılandırma tablosu.

UART4 Portu Yapılandırması	Kernelde Güncellenecek Dosya	Güncelleme Değeri	Açıklama
Alım(RX)	/sys/kernel/debug/omap_mux/gpmc_wait0	26	Mod 6 - Giriş
Gönderim(TX)	/sys/kernel/debug/omap_mux/gpmc_wpn	6	Mod 6 - Çıkış

Seri haberleşme için JAVA tabanlı olarak yazılmış olan RXTX Kütüphanesi kullanılmıştır. Buradaki haberleşme deneme ortamı için olay tabanlı iki yönlü haberleşme olarak seçilmiştir. Böylece istenildiği zaman paket gönderilecek ve seri porta herhangi bir paket yazıldığında olay tetiklenecektir. Paket yapısının protokole uygunluğu burada işlenebilecektir.

Oluşturulan protokolün yapısı 5 byte olarak tasarlanmıştır[16]. Bu protokol ile gönderilebilen ve alınabilen komutların genel özellikleri iPLUG priz durum bilgisinin okunması ve yazılması, iPLUG priz voltaj bilgisinin okunması ve yazılması, iPLUG priz akım bilgisinin okunması ve yazılması şeklindedir[16].

Gerekli veriler tek bir paket gönderilerek tüm veriler okunabilmektedir[16]. Tüm bu bilgiler iPLUG priz ID üzerinden okunmaktadır ve yazılmaktadır. Bu nedenle bu protokolü desteklemek için "SerialCommunication.java" sınıfı yazılmıştır. Bu sınıf üzerindeki metodlar protokoldeki paketlerin oluşturulmasından ve gelen paketlerin protokole uygunluğuna göre plug hash tablosunu güncellemekten sorumludur.

Bu bölümdeki temel olarak dikkat edilmesi gereken şey RXTX kütüphanesinin işaretli sekizli dizi (ing:signed byte array) almasından ve iPLUG haberleşme modülünün işaretli karakter dizisi (ing:unsigned char array) göndermesinden kaynaklanan dönüşüm ve hesaplama zorluğudur. Bu zorluk işaretli sekizli verinin 0xFF ile VE'lenmesi ile çözülmüştür. JAVA üzerindeki sekizli (ing:binary) tipi işaretli(signed) olarak yazıldığı için gelen değerler STRING tipine çevrildiğinde

negatif deęer d6nmektedir. Bu nedenle CRC hesaplanırken veya CRC kontrol yapılırken bu deęer işaretsiz (ing:unsigned) olarak d6nüşürölüp işlenmektedir.

## 5. ANALİZ

Bu bölümde medya iletim sistem ve priz kontrol sistemine ait yapılan testlere ait analizlere yer verilmektedir. Temel olarak iki bölüme ayrılmıştır, ilk bölümde PLC adaptör ile medya iletim testine ait detaylar bulunmaktadır. İkinci bölümde ise iPLUG sisteminin priz kontrolü sistemine ait detayları bulunmaktadır.

### 5.1. PLC Adaptör ile Medya İletim Testi

Normal ev kullanımında adaptörün kullanım şekli bir bileşeni modeme bağlanacak, diğer bileşeni ise dağıtımın yapılacağı bilgisayara bağlanacak şekildedir. Ancak tez kapsamında izole bir ortam üzerinde test etmek istenildiğinden; bir bileşeni Beaglebone'a diğer bileşeni ise bilgisayara bağlayacak şekilde bir test ortamı hazırlanmıştır.

Testin amacı, ev ortamında kullanılan cihazlara karşı olan tutarlılığı olduğu için test evin içerisinde iki farklı odada video gönderimi ile test edilmiştir. Testler temel olarak iki ana bölümde oluşmaktadır. Birincisi, TV, fırın, buzdolabı, mikrodalga gibi birçok ev elektroniği cihazının bağlı olduğu bir evde aralarında ~30m elektrik kablosu olan iki ayrı odada video transferi denenmiştir. İkincisi ise bu ortama elektrik süpürgesi gibi aşırı gürültü yayan cihazlar çalıştırılırken test edilmiştir. Test 1dk süre boyunca aynı görüntünün gönderilmesi ile yapılmıştır. Test sonuçları Tablo 3 'deki gibidir.

Tablo 5.1: Farklı test ortamlarındaki paket gönderimi sonuçları.

No	Test	PC	BB	KD
1	Sabit Gürültülü	5168	4984	185
2	Sabit Gürültülü	5175	4993	182
3	+ Elektrik Süp.(Sabit)	5168	5079	89
4	+ Elektrik Süp. (Değişken) +Klima +Micro Dalga Fırını	5168	4769	402
5	+ Klima	5168	4959	209
6	+ Elektrik Süpürgesi(Sabit) + Klima + Mikrodalga Fırını	5168	4826	344
7	+ Elektrik Süpürgesi(Sabit) + Klima	5168	5043	127

Buradaki testler kişisel bilgisayardan (PC)'den gönderilen medya paketlerinin Beaglebone(BB) 'dan alınıp gösterilmesini içermektedir. Sabit gürültü ile belirtilen gürültü, evde çalışan TV, aydınlatama ve beyaz eşyayı içermektedir. Buna ek olarak eklenen gürültü kaynakları tabloda elektrik süpürgesi, klima ve mikrodalga fırını olarak belirtilmiştir. Bu tabloda KD (ing: kernel drop) ile belirtilen sayı ise Beaglebone işletim sistemi tarafından yok sayılan paketlerin sayısını içermektedir. Tablo 3 'de belirtilen PC ve BB paket sayıları, IP ve UDP portunun filtrelenmesiyle alınmış sonuçlardır.

## 5.2. IPLUG MCU ile IPLUG Priz Kontrol Sistemi

IPLUG MCU ile iPLUG priz kontrol sistemi testi için 1 adet MCU ve 2 adet priz sistemi olacak şekilde tasarlanmıştır. Testler dört bölümden oluşmaktadır. Ana ekran testi, kontrol ekranı testi, ayarlar ekranı testi ve test ekranı üzerinden çoklu paket gönderim testi şeklindedir.

Ana ekran testinde ana ekran üzerinde herhangi bir priz seçildiğinde bu prize ait voltaj ve akım değerlerinin gösterilip gösterilemediği kontrol edilmektedir. Herhangi bir priz seçildiğinde her 5sn de bir “publicRead” komutu gönderilerek mevcut statü, voltaj ve akım değerleri okunmaktadır ve bu değerler ekranda gösterilmektedir. Burada ekranda voltaj-akım değerlerinin şebeke hattı üzerinde başarılı şekilde okunabildiği doğrulanmıştır.

Kontrol ekranı testinde; ekran üzerinde seçilen bir prizın MCU üzerinden açılıp kapatılabilirliği test edilmiştir. Seçilen bir prize “sendWriteStatus” komutu ile statüsü değiştirilerek prize bağlı lambanın yanıp sönmeye sağlanmıştır.

Ayarlar ekranı testinde; ekran üzerinde seçilen prizın ismi değiştirilip kaydedildiğinde bunun durum XML'ine başarılı şekilde yazıldığı ve ekranlar arasında değişiklik yapıldığında bu verinin korunup korunmadığı kontrol edilmiştir. Ayrıca yapılan değişiklik iptal edildiğinde bu veri XML'e yazılmadan ve “plugsData” veri yapısı güncellenmeden değişiklik öncesi haline gelip gelmediği kontrol edilmiştir.

Test ekranında ise protokol için tasarlanan tüm komutlar tek tek test edilmiştir ve başarılı şekilde konsol üzerinde verinin geldiği doğrulanmıştır. Aynı zamanda çoklu “publicRead” komutu gönderilerek ve gelen paketlerin sayısı ile gönderilen

paketlerin sayısı konsol üzerinde yazdırılarak haberleşme için uygun değer aralığının bulunması için ortam sağlanmıştır. Bu ekran üzerinde girilen plug ID ile komutlar gönderilerek gönderilen ve alınan bilgiler konsol üzerinden kontrol edilmiştir.

Daha sonra sistemin çalışma kapasitesini ölçmek için mevcut sistem, gürültüsüz ortamda, saç kurutma makinesi, elektrikli ısıtıcı ve bulaşık makinesi iPLUG prize takıldığında üçer kez ölçüm alınarak test edilmiştir. Test ortamı ve araçları ile ilgili detay [16] tezinde bulunmaktadır.

Yüksüz ortamda yapılan ölçümlerde Tablo 5.2 deki sonuçlar ortaya çıkmıştır. Bu sonuçlara göre sistem 200ms sıklıkla 200bit/s hızda kullanılabilir bir performansta haberleşebilmektedir.

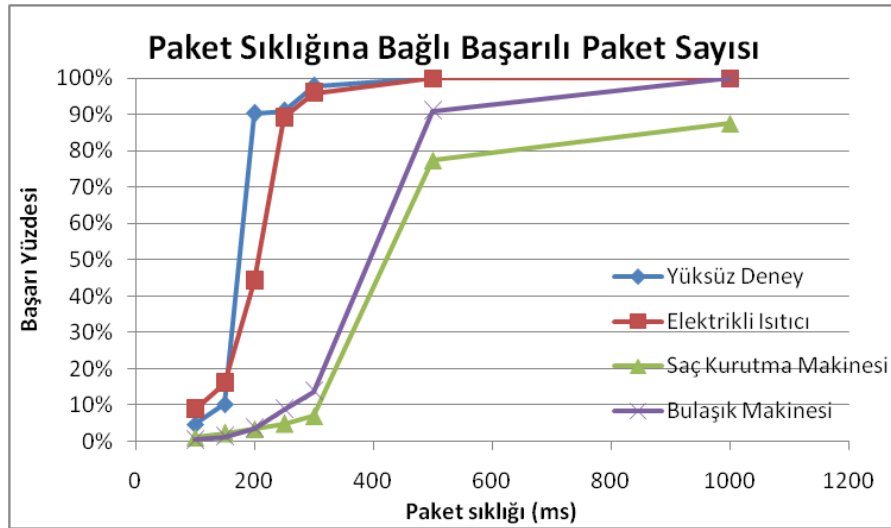
Tablo 5.2: Yüksüz durumda hesaplanan performans değerleri.

Paket Sıklık Süresi		Performans Sonuçları			
ms	bit/s	Deney 1	Deney 2	Deney 3	TOPLAM
100	400	9%	1%	4%	5%
150	267	21%	8%	2%	10%
200	200	85%	91%	94%	90%
250	160	91%	90%	92%	91%
300	133	97%	99%	98%	98%

Ancak ölçümler, bulaşık makinesi, elektrikli ısıtıcı ve saç kurutma makinesi gibi gürültü bileşenleri eklenerek alındığında Tablo 5.3 deki sonuçlar ortaya çıkmıştır. Sisteme ait gürültü bazında paket sıklığına bağlı başarılı paket sayısı, gürültüsüz ve bulaşık makinesi, elektrikli ısıtıcı ve saç kurutma makinesi gürültüleri ayrı ayrı eklendiğinde Şekil 5.1 deki grafik ortaya çıkmaktadır.

Tablo 5.3: Gürültü bazında performans tablosu.

Paket Sıklık Süresi		Performans Sonuçları		
Paket sıklık süresi	bit/s	Elektrikli Isıtıcı	Bulaşık Makinesi	Saç Kurutma
100	400	5%	9%	1%
150	267	10%	16%	2%
200	200	90%	44%	3%
250	160	91%	89%	5%
300	133	98%	96%	7%
500	80	100%	100%	77%
1000	40	100%	100%	88%



Şekil 5.1: Gürültü bazında paket sıklığı karşılaştırması.

Şekil 5.1 de görüldüğü üzere sistem yüksüz ortamda 200ms sıklıkta paket gönderiminde kabul edilebilir düzey olan %90 paket alımını desteklerken bu değer saç kurutma makinesi için ancak 1000ms civarında yakalanabilmiştir.

Saç kurutma makinesi, bulaşık makinesi gibi cihazlar günlük hayatta ev ortamında kullanılan en gürültülü cihazla olduğu için sistemi en kötü durumda çalıştırmak yerine gelen paket oranına göre değiştirilebilir hızlarda haberleşme sağlanabilir. Bu nedenle haberleşme yapısının değiştirilmesi ve bu yapıya uygun hale getirilmesi ile sistem gürültüye daha dirençli hale getirilebilir.

## 6. SONUÇLAR

Tez kapsamında oluşturulan gömülü sistem ile medya aktarımı ve akıllı priz sistemi kontrolü yapılmış ve bunların performansı farklı gürültü şartları altında incelenmiştir.

Gömülü sistem üzerinde medya aktarımı testi yapılan sistem ile PLC Ethernet adaptörlerin performansı, ev ortamındaki gürültü kaynaklarının medya aktarımına etkisi incelenmiştir. Alınan farklı verilerde görüldüğü üzere sistem medya aktarımı için yüzde on eşik değerini aşmamıştır ve görüntü aktarımında insan gözünün algıladığı herhangi bir bozulma görülmemiştir. Böylece oluşturulan sistemin ev ortamında güç hatları üzerinden veri transferine elverişli olduğu görülmüştür.

iPLUG sistemi ile iPLUG prizleri için oluşturulan sistemin performansı incelenirken ev ortamında sistemi en çok zorlayacağını düşündüğümüz farklı gürültü kaynakları, saç kurutma makinesi, elektrikli ısıtıcı ve bulaşık makinesi, gibi cihazlar seçilmiştir. Elde edilen sonuçlar incelendiğinde bulaşık makinesi ve saç kurutma makinesi gürültülerinde sistem üzerinde yüksek hızda paket gönderimi mümkün olmamaktadır. Paket gönderim sıklığı azaltıldığında paket kaybı azalmaktadır.

Sonuç olarak Beaglebone gömülü sistemi ile farklı sistemlere uyarlanabilir olarak çalışabilen ev ortamında kullanılacak PLC üzerinden çalışan bir sistem gerçekleştirilmiştir. Tasarlanan iPLUG sistemi haberleşme problemi giderildiğinde ve sistemin gürültü direnci artırıldığında, ev ortamında kullanılacak ve ev içerisindeki cihazlar üzerinde kontrol ve enerji sarfiyatı ölçümü yapabilecek yeni bir sistem olarak kullanılabilen bir sistem olacaktır.

Güç hatları üzerinden haberleşebilen akıllı ev sistemi iPLUG ile mevcut evlere kolayca entegre edilebilen, esnek, genişletilebilir ve mevcut akıllı ev sistemlerine entegre edilebilir bir sistem tasarlanmıştır.

iPLUG sistemi geliştirilerek mevcutta ikinci seviye ev otomasyon sistemi olarak çalışan sistem zaman bazlı yapılandırma yapılarak üçüncü seviye bir sistem haline getirilebilir. Ayrıca mobil veya internet üzerinden kontrol edilme altyapısı geliştirilerek dördüncü seviye akıllı ev sistemi haline getirilebilir. Son olarak kullanıcının alışkanlıkları göz önünde bulundurularak ve sisteme ek sensörler eklenerek evdeki yaşam zamanı bilgisi alınıp otomatik olarak belirlenen modlarda çalışma sağlanabilir. Böylece sistem beşinci seviye akıllı ev sistemi haline gelecektir.

Bundan sonra yapılacak çalışmalarda sistemin PLC üzerinden haberleşme altyapısının güçlendirilmesi, beşinci seviye akıllı ev sistemine dönüştürmek için yapılması gereken; priz kontrolünün zamanlanması, ev dışından sistemin kontrolü ve sisteme ek sensörler ekleyerek yaşam alanına ait farklı durumlarda analiz yapıp karar alabilecek sistem haline getirilmesi için çalışılması planlanmaktadır.

## KAYNAKLAR

- [1] Web 1, (2014), <http://www.mcu-turkey.com/beaglebone-ilk-incelemeler>, (Eriřim Tarihi: 11/03/2014).
- [2] Web 2, (2014), <http://beagleboard.org/Products/BeagleBone>, (Eriřim Tarihi: 11/03/2014).
- [3] Web 3, (2014), [http://processors.wiki.ti.com/index.php/AM335x\\_LCD\\_Controller\\_Driver%27s\\_Guide](http://processors.wiki.ti.com/index.php/AM335x_LCD_Controller_Driver%27s_Guide), (Eriřim Tarihi: 11/03/2014).
- [4] Web 4, (2014), <http://beaglebone.cameon.net/home/doing-java-development>, (Eriřim Tarihi: 11/03/2014).
- [5] Web 5, (2014), <http://www.mcu-turkey.com/beaglebone-eclipse-rse-eklentisi-ile-java-proje-yonetimi>, (Eriřim Tarihi: 11/03/2014).
- [6] Web 6, (2014), <http://www.oracle.com/technetwork/java/javase/download-142937.html>, (Eriřim Tarihi: 11/03/2014).
- [7] Web 7, (2014), <http://bcook.cs.georgiasouthern.edu/cs523/2vandana/index.html>, (Eriřim Tarihi: 11/03/2014).
- [8] Web 8, (2014), [http://www.cs.odu.edu/~cs778/spring04/lectures/jmfsolutions/exam\\_plesindex.html#receiveboth](http://www.cs.odu.edu/~cs778/spring04/lectures/jmfsolutions/exam_plesindex.html#receiveboth), (Eriřim Tarihi: 11/03/2014).
- [9] Web 9, (2014), <http://www.trullysemi.com/en/product.asp?mid=13>, (Eriřim Tarihi: 11/03/2014).
- [10] Web 10, (2014), <http://www.tp-link.com/en/products/details/?model=TL-PA210kit#spec>, (Eriřim Tarihi: 11/03/2014).
- [11] Tolba O., Briceño H., McMillan L., (1999), “Java-based Streaming MPEG Player”, Multimedia Systems and Applications, 216-224, Boston, MA, USA, 22-23 January.
- [12] Web 11, (2014), <http://www.cimms.ou.edu/~zhang/courses/CS5143Project.pdf>, (Eriřim Tarihi: 11/03/2014).
- [13] Lin C., Lu M., Yeh S., Chen H. H., (2009), “Video Streaming Over In-Home Power Line Networks”, IEEE Transactions Multimedia, 11 (3), 523-534.
- [14] Markarian G., Xufeng H., (2005), “Distribution of digital TV signals over home power line networks”, IEEE International Symposium on Power Line Communications and Its Applications, 409-413, Vancouver, BC, Canada, 6-8 April.

- [15] Carmona C., Pelaez E., (2012), “Network PLC Indoor: Application with VoIP and Video Streaming in HD”, IEEE Latin America Transactions, 10 (1), 1268-1273.
- [16] Hatipođlu Ö., (2014), “Güç Hatları Üzerinden İletişim İle Akıllı Priz Kontrolü Sağlayabilecek Gömülü Sistem Donanım Tasarımı”, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü.
- [17] Web 12, (2014), [http://www.opensmarthome.com/library/technologies/levels\\_of\\_home\\_automation.pdf](http://www.opensmarthome.com/library/technologies/levels_of_home_automation.pdf), (Erişim Tarihi: 11/03/2014).
- [18] Parasuraman R., Sheridan T. B., Wickens C. D., (2002), “A Model for Types and Levels of Human Interaction with Automation”, IEEE Trans Syst Man Cybern A Syst Hum., 30 (3), 286-297.
- [19] Web 13, (2014), <http://www.iot-butler.eu>, (Erişim Tarihi: 11/03/2014).

## ÖZGEÇMİŞ

Yetişkan Eliaçık, 1984 yılında Malatya Yazıhan'a bağlı Karaca köyünde doğmuştur. Lisans eğitimini 2008 Eylül ayında GYTE Bilgisayar Mühendisliği bölümünde tamamlanmıştır. 2008 ile 2012 yılları arasında NETAŞ şirketinde sırasıyla global ürün tasarım destek bölümünde ve araştırma geliştirme bölümünde platform tasarımcı olarak daha sonra da güvenlik mimarı olarak çalışmıştır. 2012 Kasım ayı itibariyle Turkcell ICT departmanında servis monitör uzmanı olarak çalışmaktadır.

## EKLER

### Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Eliaçık Y., Hatipoğlu Ö., Zergeroğlu E., (2013), “Plc İle Veri Transferi İçin Beaglebone Deneme Kitinin Hazırlanması Ve Performansı”. Otomatik Kontrol Ulusal Toplantısı TOK2013, 1247-1251, Malatya/TÜRKİYE, 26-28 Eylül.

### EK B: Beaglebone Çekirdek Derleme İşlemi İçin Gerekli Adımlar

Derleme işlemi için, Ubuntu üzerinde gerekli paketlerin "root" kullanıcısı olarak internet üzerinden kurulması gerekmektedir; kurulum işlemi aşağıdaki şekilde yapılmalıdır:

- apt-get install sed wget cvs subversion git-core coreutils unzip texi2html texinfo docbook-utils gawk python-pysqlite2 diffstat help2man make gcc build-essential g++ desktop-file-utils chrpath dosfstools kpartx ncurses-dev

Derleme işlemi için Ubuntu üzerinde yerel bir kullanıcı yaratın, bu kullanıcı bu bölüm içerisinde "user" olarak bahsedilecektir. Aşağıda "user" kullanıcısı için Angstrom işletim sistemi çekirdek kaynak dosyalarının indirilmesi için gerekli klasörün oluşturulması ve internet üzerinden Angstrom Linux dağıtımına ait derleme kodlarının indirilmesi için gerekli adımlar gösterilmiştir:

- mkdir /home/user
- git clone git://github.com/Angstrom-distribution/setup-scripts.git

Derleme işleminin yapılması için Ubuntu bilgisayarınız üzerinde, derleme işleminin Beaglebone için yapılandırılması ve varsa çekirdek kaynak kodlarının güncel hallerinin alınması gerekmektedir.

- MACHINE=Beaglebone ./oebb.sh config Beaglebone
- MACHINE=Beaglebone ./oebb.sh update

Şu ana kadar çekirdek kaynak kodunu Ubuntu işletim sistemi çalışan kişisel bilgisayar üzerine indirilmesi işlemleri yapılmıştır. Bundan sonra Angstrom Linux işletim sistemi çekirdeğinin derlenmesi ve Beaglebone'un bu çekirdek sürücü dosyaları ile açılması için önyükleme(boot) dosyasının oluşturulması amaçlanmıştır.

- MACHINE=Beaglebone ./oebb.sh bitbake virtual/kernel
- MACHINE=Beaglebone ./oebb.sh bitbake u-boot

Derleme işlemi sonunda oluşan ve sürücüleri içeren “.tar” uzantılı çekirdek birim(kernel module) dosya ve “uImage” dosyası Beaglebone SD kartına kopyalanır. Daha sonra “uImage” dosyası “boot”(önyükleme) klasörü altına kopyalanır ve işletim sisteminin oluşturulan bu çekirdek modül dosyalarını kullanması için “boot” dosyasında çekirdek sürümüne ait gerekli değişiklikler yapılır. Tar dosyası da uygun kernel klasörü olarak Beaglebone dosya sistemi üzerine kopyalanır. Böylece Beaglebone'un yeni çekirdek üzerinden başlatılması sağlanmaktadır. Gerekli dosyalar derlenen klasörün içinde aşağıdaki dizinde yer almaktadır. Bundan sonra yapılan tüm derleme işlemlerinin çıktısı için “./build/tmp-angstrom\_vYYYY\_MM-eglibc/Beaglebone-angstrom-linux-gnueabi/deploy/images /Beaglebone/” klasörü üzerinde işlem yapılıacaktır.

Beaglebone, dosya sisteminde yapılan değişiklikler ile açıldıktan sonra işletim sisteminin yeni çekirdek(kernel) ile ayağa kalkıp kalmadığı “uname -a” komutu ile kontrol edilir; bu komut çıktısında oluşturulan çekirdek sürümünün görülmesi beklenmektedir. Bununla birlikte gerekli modüllerin kullanıldığından emin olmak için “lsmod” komutu ile mevcut modüllerin yüklendiğinden emin olunur.

Böylece Beaglebone imajında gelen mevcut özellikleri kullanabileceğimiz bir ortam oluşturmuş olunmaktadır. Ayrıca bunun yanında Angstrom işletim sistemine ait kaynak kodu da derleyerek çekirdek üzerinde istediğimiz değişiklik yapılabilir.

## **EK C: Diğer Ekler (Kompakt Disk)**

iPLUG sistemine ait yazılmış olan programların kaynak kodları ve proje dosyası iPLUG.rar olarak eklenmiştir.