

Mobil Cihazlar ile Kontrol Edilebilen Akıllı Ev Yönetim Sistemleri için Haberleşme Protokolü Tasarımı ve Gerçeklenmesi

Design and Implementation of a Communication Protocol for Mobile Device Controlled Smart Home Management System

Mustafa Menteşoğlu^{*}, Adnan Kavak^{*,+}, Mehmet Yakut^{**,+}, Ali Tangel^{**,+}, Suhap Şahin^{*}, Hikmetcan Özcan^{*}

^{*}Kocaeli University, Department of Computer Engineering, Kocaeli, Turkey

^{**}Kocaeli University, Department of Electronics and Communication Engineering, Kocaeli, Turkey

⁺Milets Information, Electronics, and Telecom. Ltd. KOU Teknopark, Kocaeli, Turkey

mmentesoglu@gmail.com, {akavak, myakut, atangel, suhapsahin, hikmetcan.ozcan}@kocaeli.edu.tr

Özetçe—Son yıllarda mobil cihaz ve internet erişilebilirliğinin artması ile akıllı ev sistemlerinin mobil cihazlar ile uzaktan yönetilmesi ve durumlarının takip edilmesi yaygın hale gelmeye başlamıştır. Bu çalışmada akıllı ev sistemine yönelik mobil cihaz-gömülü bilgisayar tabanlı kontrol ünitesi- sensörler/eyleyiciler arasında haberleşme protokolü tasarlanıp yazılım ve donanım gerçekleştirilmesi sağlanmıştır. Mobil cihaz ile kontrol birimi arasındaki el sıkışma ve kontrol birimi ile sensörler/eyleyiciler arasındaki mesajlaşma paket yapısı açıklanmaktadır. Bu sistem ile android tabanlı mobil cihazlarda çalışabilen bir GUI yazılım ile sistemin kablosuz yerel ağ üzerinden yetkilendirilmiş kullanıcılar tarafından yönetilmesine ve takip edilmesine imkan sağlanmıştır.

Anahtar Kelimeler — Akıllı ev, haberleşme, mobil, android

Abstract—Recently, with the increase in mobile and internet accessibility, it has become widespread that smart home systems can be managed remotely and monitored by mobile devices. In this study, a communication protocol between a mobile devices-embedded PC based control unit-sensors/actuators is designed and implemented. A handshaking mechanism between a mobile device and control unit, and packet structure for messaging between control unit and sensors/actuators are explained. With this system, the smart home system can be controlled over a wireless LAN by authorized users using an Android based mobile device on which the implemented GUI software is running.

Keywords — Smart Home, communication, mobil, android

I. GİRİŞ

Akıllı ev otomasyon sistemleri, bilişim teknolojilerini kullanarak telesağlık, multimedya, eğlence ve enerji tasarrufu sağlayarak ev sakinlerinin yaşam kalitesini arttırmak için kullanılır [1]. Geleneksel ev otomasyon sistemleri, ısıtma, aydınlatma ve gölgeleme gibi işlevleri sağlayan dijital cihazların kontrolünü içerir. Ama son yıllarda bilişim teknolojileri ve modern eğlence sistemlerindeki hızlı gelişme, bu temel işlevlerin yanında ek hizmetler ile zenginleştirilmiş olmasını bekleniyor (örneğin telefon çaldığında müziğin sesini azaltmak gibi) [2].

Akıllı ev, en anlaşılır tanımıyla içinde yaşayan insanlarla sürekli etkileşim içinde olan ve verilen komutlar doğrultusunda görevlerini yerine getiren evdir. Akıllı ev kavramındaki en önemli nokta, uygulanan otomasyon sisteminin kullanıcı isteklerine bağlı şekilde oluşturulmasıdır. Bir kumanda ile veya dokunmatik bir panel üzerinden evin aydınlatma sisteminden, güvenlik kameralarına ya da perde panjur kontrolünden ev sinemasının yönetimine kadar bütün istekler gerçekleştirilebilir. Hatta bütün bu fonksiyonlar telefon veya internet üzerinden kullanıcıların kontrolünde olabilmektedir [3].

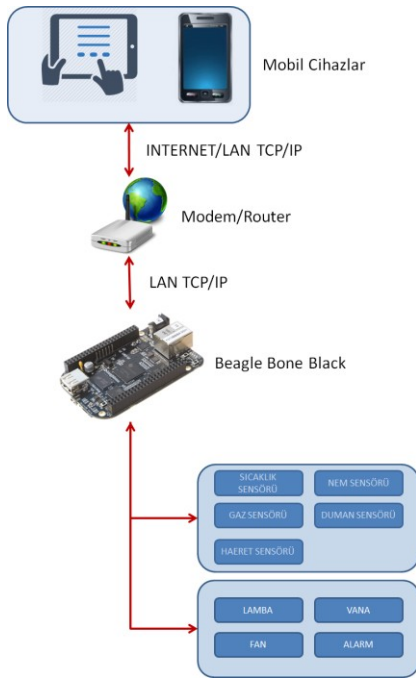
II. MOBİL TABANLI AKILLI EV YÖNETİM SİSTEMİ HABERLEŞME PROTOKOLÜ

Bu çalışmada gerçekleştirilen akıllı ev otomasyon sisteminde kullanılan haberleşme protokolü Şekil 1’de görüldüğü gibi iki kısımdan oluşmaktadır.

Birinci kısım sensör ve eyleyicilerin sunucu ile haberleşmesi, ikinci kısım ise insan bilgisayar etkileşimini sağlayan mobil cihazlar ile sunucunun haberleşmesi üzerine kuruludur.

Bu makalenin odaklandığı kısım kontrolör ünit ile mobil cihazlar ile aktif ve pasif sensör ve cihazların yönetilmesi ve gelen bilgilerin kullanıcıya aktarılması için gerekli haberleşme protokolüdür.

Bu çalışmada sensör ve cihazların yönetilmesini sağlamak amacıyla düşük maliyet ve çok sayıda GPIO(69 adet) ve ADC içermesi nedeniyle üzerinde Linux işletim koşan ARM tabanlı BeagleBone Black gömülü bilgisayar sistemi tercih edilmiştir[4]. Buradan itibaren kullanılan gömülü sistem “sunucu” olarak anılacaktır.



Şekil-1

Çalışma başlangıcında mobil cihazlar ile yönetici panel arasındaki haberleşme protokolü için UDP seçilmiş olup testlerde kablosuz ağlarda yaşanan paket kayıpları nedeniyle TCP/IP protokolünün kullanılması tercih edilmiştir.

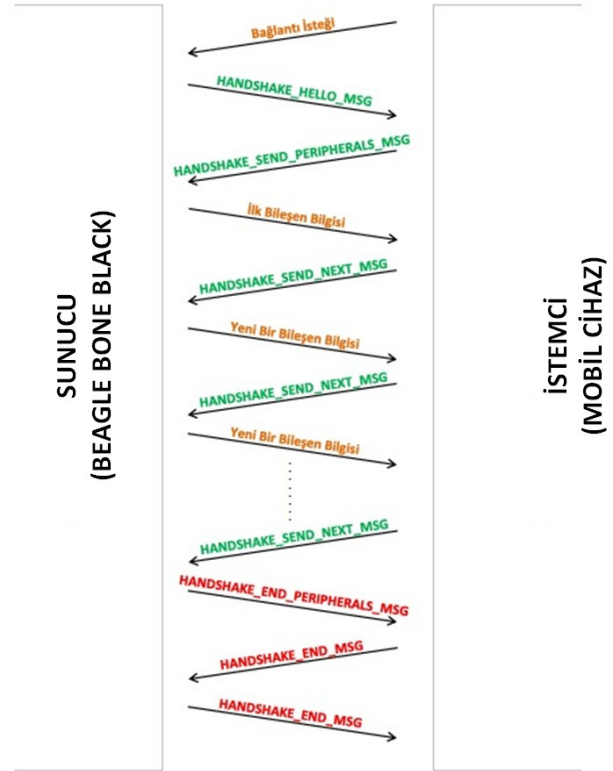
Mobil cihaz olarak Android işletim sistemine sahip farklı firmalara ait tablet ve telefon ürünleri kullanılmıştır. Uygulama geliştirme amacıyla Google firması tarafından sağlanan Android Yazılım Geliştirme Kiti[5], Eclipse yazılım geliştirme ortamı kullanılmıştır[6].

Takip eden bölümlerde sunucu ve mobil cihazlar arasındaki haberleşme protokolü yapısı anlatılmaktadır.

A. El sıkışma yöntemi

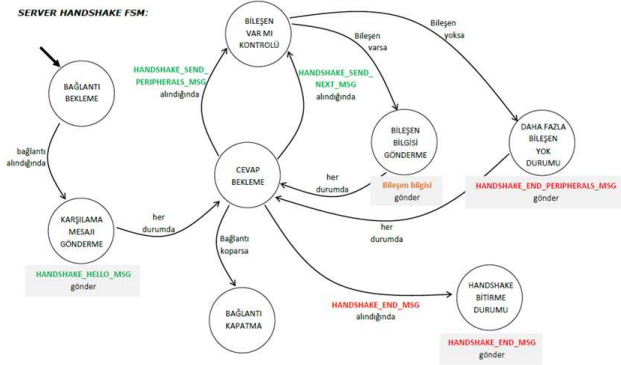
Mobil cihazın sunucuya ilk bağlanmak istediği zaman yürütülen protokoldür. Amacı sisteme bağlanmak isteyen kişinin sistem tarafından yetkilendirilmiş bir kişi olup

olmadığının kontrolü ve yetkilendirilmiş kişi ise akıllı cihaz ile sunucu arasındaki konfigürasyonlarının senkronizasyonunun sağlanmasıdır. Şekil-2’ de el sıkışma yöntemine ait akış diyagramı Şekil 3’ te HandShake metoduna ait sonlu durum diyagramı verilmiştir. Sisteme bağlanmak isteyen kişinin kullanıcı adının ve şifresi “HELLO_MSG” başlığında alınır ve sunucu tarafından kullanıcı adı ve şifre kontrol edilerek, doğru ise sunucu ve mobil cihaz arasındaki senkronizasyonu başlatır, eğer yanlış ise oturumu sonlandırır. Senkronizasyon işleminin başarılı bir şekilde bitirilmesini müteakip “HAND_SHAKE_END_MSG” paketi ile oturum sonlandırılır. Oturum açma işleminin başarılı bir şekilde oluşturulmasından sonra istemci sunucu ile bağlantı halinde kalır ve istemci tarafından oturum sonlandırılana kadar bağlantıyı korur durum değişimleri hakkında istemciyi bilgilendirir.



Şekil-2

SERVER HANDSHAKE FSM:



Şekil-3

B. Haberleşme Paket Yapısı

Sunucunun akıllı ev yönetim sisteminde mevcut bulunan sensör ve cihazları yönetmesi ve topladığı verileri istemciye aktarması amacıyla iki tip mesaj paket yapısı oluşturulmuştur. Birinci mesaj paket yapısı “Konfigürasyon Mesaj Paketi” olup veri yapısı tablo-1’ de verilmiştir. Bu mesaj paketinin amacı sunucu tarafından sensör ve cihazların ayarlarını yapılandırmak amacıyla hazırlanmıştır. İkinci mesaj paket yapısı ise “Yönetim Mesaj Paketi” olup veri yapısı tablo-2’ de verilmiştir. Bu mesaj paketinin amacı sensör ve cihazların durum bilgisini almak ve güncellemek için kullanılır.

Alan Adı	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
IO_ID	1 Byte	Sensör 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 / Kapalı
TIME_STAMP	8 Byte	Mesaj Gönderim Zamanı

Tablo-1

Alan Adı	Boyut	Açıklama
MESAJ_ID	1 Byte	0x01000000b : Gelen Yönetim Paketi 0x01000001b : Giden Yönetim 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
DATA	2 Byte	Algılayıcıdan gelen bilgileri göndermek için ayrılmış alan
FLAG	1 Byte	0. Bit : Takılı/Takılı Değil 1. Bit : Aktif/Pasif 2. Bit : Çalışma Seviyesi 3. Bit : Geçici Devre Dışı 4. Bit : Açık/ Kapalı
START_TIME	4 Byte	Başlangıç Zamanı
STOP_TIME	4 Byte	Bitiş Zamanı
DAY	1 Byte	İlgili bitin “1” olduğu durumlara denk gelen n. bit değerine göre hangi günler açık ve kapalı zamanlarını uygulayacağı bilgisini göndermek için. 0x00000001 Pazartesi 0x00000010 Salı 0x00000100 Çarşamba 0x00001000 Perşembe 0x00010000 Cuma 0x00100000 Cumartesi 0x01000000 Pazar 0x10000000 Tüm Günler
TIME_STAMP	8 Byte	Mesaj Gönderim Zamanı

Tablo-2

İstemci tarafından gönderilen mesajlar eğer HAND_SHAKE metodu başarılı bir şekilde tamamlandıysa sunucu tarafından gelen mesajın ilk byte’ ında bildirilen MESAJ_ID veri alanına göre değerlendirilerek mesaj çözümleme işlemi gerçekleştirilir.

MESAJ_ID veri alanında en düşük değerli(LSB) bit mesaj paketinin sunucu tarafından mı yoksa istemci tarafından mı gönderildiğini belirtmek için kullanılır. LSB bitinin 0 (sıfır) olması durumunda bu paketin istemci tarafından sunucu tarafına, 1 (bir) olması durumunda ise sunucu tarafından istemci tarafına gönderilen bir mesaj paketi olduğunu ifade eder. MESAJ_ID veri alanının en yüksek değerli (MSB) bitinin 1 olması mesaj paketinin konfigürasyon mesaj paketi olduğu, MSB bitinin sağdaki bitin 1 olması ise yönetim paketi olduğunu ifade eder.

C. Sisteme Yeni Sensör/Cihaz Eklenmesi

Akıllı ev yönetim sisteminde cihaz ve sensör ekleme işleminin dinamik bir şekilde yapılabilmesini sağlamak amacıyla mobil cihaz ile sunucu arasında yapılandırma mesajlarını paylaşmak amacıyla oluşturulan “Konfigürasyon Mesaj Paketi” kullanılır. Sisteme eklenebilen sensör ve cihaz tipleri Tablo 1’ de DEVICE_TYPE alan adı başlığı altında listelenmiştir. Kullanıcı tarafından mobil cihaz üzerinde yapılması gereken 3 temel yapılandırma ayarı vardır. Bunlar sırası ile

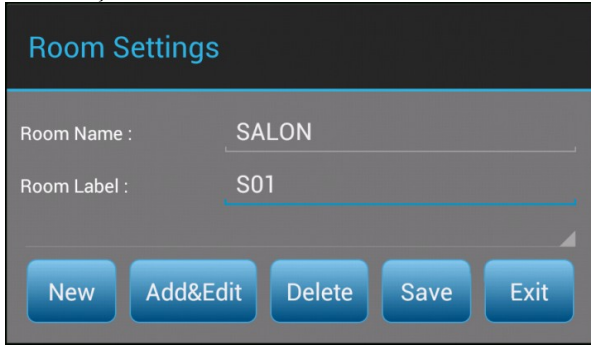
- Kullanıcı tanımlama
- Oda/Mekan tanımlama
- Sensör/Cihaz tanımlama/ekleme’ dir.



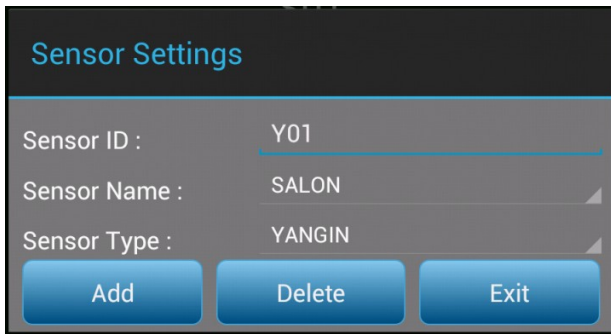
Şekil-4

Kullanıcı tarafından oluşturulabilecek üç farklı seviyede kullanıcı tipi vardır. Şekil 4’ te yeni kullanıcı tanımlama ekranı gösterilmiştir. Yeni oda/mezan tanımlama ekranı Şekil 5’ te ve yeni sensör/cihaz ekleme ekranı Şekil 6’ da verilmiştir.

İstemcinin doğru kullanıcı adı ve şifreyi girmesine müteakiben sunucu tarafından yapılandırma dosyası JSON formatında istemciye aktarılmaktadır. Mobil cihaz tarafında gelen JSON paketinin çözülmesi amacıyla Google tarafından sağlanan GSON kütüphanesi kullanılmıştır .



Şekil-5



Şekil-6

Yapılandırma dosyasının transferi tamamlandıktan sonra istemci tarafında kullanıcının yönetim işlemlerini yapabilmesi ve sensör verilerini okuyabileceği ana ekran (Şekil-7) oluşturulmuştur.

İstemci tarafında kullanıcıya sağlanan arayüz (Şekil-7) vasıtası ile kullanıcı tüm sensör ve cihazların mevcut durumlarını takip edebilmekte, cihaz/sensörlerin aktif/pasif olma durumlarını yetkilendirme seviyesine göre değiştirebilmektedir.



Şekil-7

III. SONUÇ

Bu çalışmada Android işletim sistemli mobil cihazlarla akıllı ev sisteminin gözlenmesi ve kontrol edilmesine imkan sağlayan, mobil cihaz-kontrol birimi-sensörler/eyleyiciler arasındaki uygulamaya özgün bir haberleşme protokol tasarımı ve gerçekleştirilmesi üzerinde durulmuştur. TCP/IP haberleşme protokolünü kullanarak Akıllı ev yönetim sistemi ile mobil cihazların haberleşme protokolü tanımlanmıştır. Oluşturulan bu protokolü kullanarak istemci/sunucu mimarinde Android işletim sistemine uygun mobil uygulama gerçekleştirilmiştir. İstemci-sunucu haberleşmesi kablosuz yerel ağ kullanarak gerçekleştirilmiştir. Bu yapı kullanıcının sadece ev içinde değil uzak konumda da mobil iletişim şebekesini (GPRS/3G) kullanarak sisteme bağlanabilmesine imkan sağlamaktadır. Mobil uygulama üzerinde akıllı ev yönetim sistemine yeni bir cihaz/sensörün eklendikten sonra yapılandırılması işlemi kullanıcı tarafından gerçekleştirilebilmektedir.

TEŞEKKÜR: Bu çalışmada 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.

KAYNAKÇA

- [1] K. Bromley, M. Perry, and G.Webb “Trends in Smart Home Systems, Connectivity and Services”, www.nextwave.org.uk, 2003.
- [2] M. Kovatsch, M. Weiss, and D. Guinard, “Embedding internet technology for home automation”, Proc. Of ETFA, 2010, pp. 1-8.
- [3] Eratek web site, www.erak.com.tr/akill.htm (accessed 10.02.2015)
- [4] BeagleBoard web site, http://www.beagleboard.org/black (accessed 17.01.2015)
- [5] Android SDK Tools, http://developer.android.com/sdk/ (accessed 14.02.2015)
- [6] Eclipse IDE for Java Developers, http://eclipse.org/downloads (accessed 14.02.2015)
- [7] GSON, http://code.google.com/p/google-gson (accessed 14.02.2015)