

Bir Uzaktan Program Yükleme ve Yönetim Sistemi

A Remote Program Installation and Management System

Nilgün İncereis¹, Bekir Tefvik Akgün²



¹Bilgisayar Mühendisliği
İstanbul Okan Üniversitesi
niincereis@stu.okan.edu.tr
ORCID:0000-0001-5508-8159

²Bilgisayar Mühendisliği
İstanbul Okan Üniversitesi
tevfik.akgun@okan.edu.tr
ORCID: 0000-0002-9726-1340

Özet

Uzaktan program yükleme (OTA) hizmeti sistemi, belirli bir ortamdaki akıllı uçbirimlere, nesnelerin interneti (IoT) aygıtlarına yeni program yükleme, sistemi izleme ve yönetme işlemini firma adına güvenli ve korumalı bir şekilde yerine getiren bir sistemdir. Önerilen sistem; hizmet yönetimi, IoT yönetimi, kullanıcılar, yöneticiler ve uçbirimlerden oluşmaktadır. Bu çalışma, önerilen uzaktan yönetim sisteminin uygulamasını ve geliştirilmesini amaçlamaktadır. Uzaktan yönetim sisteminin kesintiye uğramaması için OTA işlemleri uçbirim uygulama programının izin verdiği güvenli bir bölgede yürütülmektedir. Bu sistemde kullanıcılar, bulut üzerinde bulunan istediği dosyayı cihazına uzaktan yükleyebilir ve kullanıcılar sistemi uzaktan takip edebilir veya bulut ortamına istediği yazılım dosyasını gönderebilir ve isterse sisteme yüklediği bu dosyayı cihazına uzaktan yükleyebilir. Önerilen sistem gerçekleştirilmiş ve tasarlanan hizmetlerin başarıyla yerine getirildiği belirlenmiştir.

Anahtar kelimeler: Nesnelerin İnterneti, OTA Güncelleme, Kablosuz Sensör Ağı, İzleme Sistemi, Gömülü Sistem, Uzaktan Yazılım Güncelleme

Abstract

The remote program installation (OTA) service system is a system that performs the process of installing new programs to smart terminal units, IoT devices in a specific environment, monitoring and managing the system on behalf of the company in a secure and protected manner. Recommended system; service management consists of IoT management, users, administrators and terminal units. This study aims to implement and develop of the proposed remote management system. In order not to interrupt the remote management system, OTA operations are carried out in a secure area allowed by the

terminal application program. In this system, users can remotely upload any file on the cloud to their device, and users can remotely monitor the system or send any software files to the cloud and upload this file to their device remotely. The proposed system has been implemented and it has been determined that the designed services have been carried out successfully.

Keywords: Internet of Things (IoT), Over The Air (OTA) Update, Wireless Sensor Network, Monitoring System, Embedded System, Remote Firmware Update.

1. Giriş

Günümüzde uzaktan program yükleme işleminin yaygın kullanılması nedeniyle çok önemli bir rol üstlenir. Nesnelerin İnterneti (IoT) sistemleri, internete bağlı veri ya da bilgi alışverişi yapabilen nesnelere oluşan sistemlerde yazılım güncellemesi bir zorunluluktur. Böyle teknolojilerin kullanıldığı bir sistemi uygulamak ve geliştirmek avantajlar sağlar.

Kısıtlı bir güvenliğe sahip olunması ve güncellenen yalnızca belirli kaynaklardan izin verilmesi firmaların uzaktan program yükleme (OTA) işlemlerini kendisinin gerçekleştirmesini gerektirir. Bu çalışma şekli; yeni sürümle çalıştırma gibi aygıt işlemlerini habersiz yarıda kesmenin firma için önemli bir zarara yol açmadığı durumlarda uygun olacaktır. Ancak sahada çalışan ve kendisine atanmış görevleri kesintisiz, veri kayıpsız ya da kontrol edilen sistemlerin beklenmedik davranışlar olmaması gereken durumlarda ise; aygıtın işlemlerini güvenli bir şekilde ara vermesini sağlamak gerekir. Aygıt üzerinde çalışan bir uygulamanın korumalı bir şekilde ara verilmesi ve yeniden başlatılması ayrıca tasarlanmalıdır.

Bu çalışma; bir uzaktan program yükleme yönetim sisteminin tasarlanmasını ve geliştirilmesini kapsar. İlk olarak, önceki yapılan çalışmalar özet halinde verilmiştir. Sonraki bölümlerde, yerel yönetim/uzaktan erişim sistemi ile önerilen uzaktan yönetim sistemi üzerine uygulamalar sunulmuştur. Önerilen uzaktan yönetim sisteminin geliştirilmesi için yönetimlerin birer bulut hizmeti yazılımına dönüştürülerek tamamen bulut üzerinde dağıtık bir sistem olarak oluşturulduğunda sistemde oluşabilecek kesintilerin nedenleri de belirlenmiştir. Sistemin deneysel sonuçları verilerek, sonuçlar bölümünde güvenlik kapsamındaki değerlendirilmesinin yanında bu çalışmanın mevcut çalışmalara olan üstün yanları ve zayıf yanları üzerinde durulmuştur.

2. Önceki çalışmalar

Nesnelerin İnterneti (IoT) teknolojisi, yerel bir kontrol sisteminin kısıtlamalarına maruz kalan uygulamalar için yeni, iyileştirilmiş izleme ve kontrol yöntemlerinin geliştirilmesini gerekli kılmıştır. IoT ile bağlantılı gömülü uygulamaları izlemek ve kontrol etmek için tahmine dayalı sistem öneren bir çalışmaya bakıldığında, kontrol sisteminin tasarımı üzerindeki sonuçların ortaya çıkardığı etkilere göre bir ilk örnek model sunulmuştur [1]. Bir kontrol sistemi, kontrol döngülerini kullanarak diğer cihazların ve sistemlerin davranışını yönetir, komut verir, yönlendirir ya da düzenler. Bu sistemler, bir ev tipi kazanı kontrol edebilen bir termostat kullanan ev ısıtma kontrolünden, işleri ya da makineleri kontrol etmek için kullanan büyük endüstriyel kontrol sistemlerine kadar değişebilir [2]. Kontrol sistemi, cihazı uygun değer ayarlarında sabitleştirilmiş bir dizi kısıtlamaya neden olur. Bununla birlikte, kontrol sisteminin kendisi, sisteme bağlı olmayan bir dizi kısıtlamaya maruz bırakır ve bu da uzun yerleşme sürelerine ya da aşımına neden olur. Bu durumlarda gecikme ve ölü zamanlar oluşabilir [3].

Kang, K. ve arkadaşları tarafından [4], farklı hizmet tedarikçileri, farklı türde uygulamalar kullanarak farklı hizmetler için uygulama türlerini listelemiştir. Hizmet ya da uygulama kalitesinde, güvenilirlik kullanıcılarla ilgili önemli parçalardan biridir. IoT uygulamaları geliştirmek için, endüstri tarafından kullanılan çeşitli modellerin analizi yapılarak, modellerin kullanıldığı tekniklerin karşılaştırılması, özgünlüğü ve eksiklikleri durumlarına bakılmıştır [5]. IoT uygulamaları geliştirmek için tasarlanan modellerde, nesnelere, aygıtlara, kablosuz iletişim ortamları, ağ geçidi, bulut, uygulama sunucusu, yazılımlar, uygulamalar, kullanıcılar, yöneticiler kullanılır. Nesnelere, çevredeki verileri toplar ve bunları kablosuz iletişim ortamı ve ağ geçitleri aracılığıyla bir sunucuya gönderir. Buluttaki veriler, uygulamalar ve hizmetler aracılığıyla depolanıp, işlenip ya da analiz edildikten sonra son kullanıcılara sunulur. Bir sistemdeki uçbirimlerin çok fazla olmasıyla, kolay bağlantı, kontrol sistemi, iletişim ve aynı zamanda sağlam ve güvenli bağlantı zorlaşmaktadır. Bu nedenle uygun sistem tasarım mimarisinin oluşturulması gerekir. Çoğu IoT mimarisindeki zorluk ve sorun, standardizasyonun olmamasından kaynaklanmaktadır. Standardizasyon eksikliği ve IoT uygulama geliştirmedeki zorluklar [6] ve [7]'de açıklanmaktadır.

IoT platformu, makine ve aygıtların bağlanmasını, ardından makine ve sensör verilerinin toplanmasını, işlenmesini, dönüştürülmesini, düzenlenmesini ve saklanmasını sağlayan bir yazılımdır ve genellikle IoT ara katman yazılımı olarak da adlandırılır. Zdravkovic ve diğerleri [8], 16 farklı bulut tabanlı IoT platformunu sunmuşlardır. Buluttaki tüm IoT aygıtlarını kontrol eden esnek bir platform sağladığı söylenen bir çalışmada, çözüm önerisi olarak bir olaya cevap verme sürecini değiştirme esnekliği üzerinde durulmuştur. Yapılan bu çalışmada, bu sistem yerel makine üzerinde çalıştırılmış ve IBM Bluemix platformu yardımıyla da buluta getirilebileceğinden bahsedilmiştir [9]. Uzaktan kontrol edilebilen Wi-Fi, 3G, 4G ya da 5G işlevselliğine sahip akıllı aygıtların sayısı artmaktadır. IoT, bu tür aygıtların uzaktan bağlanmasını ve kontrol edilmesini sağlamaktadır [10]. Buluttaki aygıtların sayısı arttıkça, bunların yazılımının daha kontrollü güncellenmesi gerekir. Uzaktan güncellenmenin uygulanmasıyla ilgili [11]'de, sistem programlarının seçici olarak yüklenmesi yapılmıştır.

Yeni teknolojilerin gelmesi ve hızlı bir şekilde ilerlemesiyle, IoT'nin ortaya çıkışı günlük yaşamın her alanında önemli bir rol oynamaktadır. Böylece, ev otomasyonu, güvenlik alanındaki IoT çalışmaları ya da bazı özellikler canlıların yaşam tarzını da kolaylaştırır. Wi-Fi teknolojisi kullanılarak, ev otomasyonu kontrolünün uygulanması ve tasarlanmasında IoT, tümleşik, korumalı ve güvenli sisteme sahip aygıtları yönetir. Örneğin IoT ve mikro denetleyici tabanlı bir NodeMcu (Esp8266) ve bir Android mobil uygulaması kullanılarak güvenlik ve otomasyonda nasıl uygulanabileceği hakkında [12]'de bilgi verilmektedir.

Bulut hizmetleri ile IoT hizmetlerine küresel erişimin mümkün olduğu ve böyle hizmetler için sunucu bakımının zorluğunun ortadan kaldırılabildiği söylenmektedir. Özellikle, böyle sistemlerde genel *ip*'yi ayrı ayrı sağlamadan küresel erişim mümkündür, bu da onu bireysel ya da küçük işletme IoT hizmeti kuruluşu için çok uygun hale getirir [13]. Tasarım aşamasında bulut üzerinde uygulama çalıştırmanın maliyeti, oluşturulan sisteme göre [14]'te tahmin edilmektedir. Dell [15] tarafından kullanılan mimaride ise, ağın kenarındaki verileri güvence altına almak amacıyla uygulamalar için özel ağ geçidi kullanılır.

IoT yönetim sistemlerinin esnekliği, hata toleransı, yüksek algılamaya doğruluğu, düşük maliyetli ve hızlı dağıtım özellikleri, uzaktan yönetim için birçok yeni ve hayatımızı kolaylaştıran uygulama alanı yaratır. Ancak IoT yönetim sisteminin gerçekleştirilmesi, hata toleransı, ölçeklenebilirlik, maliyet, donanım, çevre ya da güç tüketimi gibi faktörlerin getirdiği kısıtlamaları da karşılamalıdır. Bu kısıtlamalar IoT için yeni kablosuz geçici ağ oluşturma teknikleri oluşturulmasını gerektirir [16].

Uzaktan program yükleme ile ilgili yapılan bir çalışmada [17], sistem tasarımı için, genel IoT sistemleri, yerel sistem, yerel yönetim/uzaktan erişim sistemi, uzaktan yönetim sistemi ve bulut üzerinden dağıtılmış yönetim sistemi olarak incelenmiştir. Burada, sahadaki IoT uçbirimlere, yeni program yükleme işlemlerini firma adına güvenli bir şekilde yerine getiren bir sistem için "kullanıcılar (program kodu) ve OTA hizmeti" ya da "OTA hizmeti ve IoT uç birimleri (donanım)"

arasındaki güvenli haberleşmenin üzerinde durulur. Güvenlik problemleri, uçbirimin kötü amaçla saldırıya uğrayabilmesi ya da istenmeyen bir kod yükleme sorunlarının ortaya çıkması şeklinde düşünülmektedir. Son yıllarda da OTA (Over the Air) ile ilgili bazı çalışmalar [18, 19, 20, 21] incelendiğinde veri iletişimi, iletişim modeli, güvenlik ve protokoller kapsamında da giderek artan bir ilgi bulunmaktadır.

3. Kullanılan teknolojiler

Bu bölümde, nesnelerin interneti teknolojisi, bulut teknolojisi, uzaktan program yükleme, ESP8266, OTA kütüphaneleri, fonksiyonları ve işlevleri konuları üzerinde durulmuştur.

3.1. Nesnelerin interneti teknolojisi

Nesnelerin interneti teknolojisi, kesintisiz bir ortam sağlamak için, veri alışverişi yapmada birbirleriyle etkileşim içinde olan elektronik cihazları ya da nesneleri güvenli bir şekilde yönetmeyi amaçlar. Bunlar kablosuz ya da kablolu ağlar kullanılarak uzaktan algılanabilir ve kontrol edilebilir [22]. İnternete bağlandıktan sonra, çeşitli hedeflere, kaynaklara veri yüklenebilir ya da indirilebilir. Bu tür sistemler [1], bir merkezi aygıtın bir sunucu olarak hareket ettiği ve aygıtlar arasında istekleri ilettiği bir sunucu istemci mimarisi kullanır.

3.2. Bulut teknolojisi

Bulut bilişim fikrinin temelleri 1950'li yıllarda ortaya atılarak veri merkezlerinin modernize edilmesiyle ilk gerçek bulut bilişim hizmeti oluşturulmuştur. Bulut bilişim teknolojisinde büyük veriler, internette depolanabilir ve istenirse bu verilere erişilebilir. Bulut teknolojisinin çeşitleri, aşağıdaki şekilde verilmektedir [23].

Public Cloud (Genel Bulut): İnternet üzerindeki sunucular ile kurulan bir bulut teknolojisidir.

Private Cloud (Özel Bulut): Bilgileri önemli olan büyük şirketlerin tercih ettiği bir bulut teknolojisidir. Tüm bilgilere erişim güvenliği ve gizliliği yüksektir.

Hybrid Cloud (Melez Bulut): Genel ve Özel Bulut birleşiminden ortaya çıkan bulut teknolojisidir.

Community Cloud (Topluluk Bulut): Birkaç şirket ile ortak kullanılan hizmetleri barındıran bulut teknolojisidir. Topluluk üyeleri, uygulama ve verilere erişebilmektedir.

3.3. Uzaktan program yükleme

Wi-Fi ile internete çıkabilen *ESP8266* serisi tüm devre kartları, hem boyut olarak küçük hem de fiyat olarak ucuz olduğundan yaygın olarak kullanılmaktadır. *ESP8266*'nın üretiliş amacı, öncelikle Arduino'yu *Wi-Fi* üzerinden internete bağlamaktır. Ancak şimdi, *Xbee*, *Ethernet kartı* gibi pahalı parçalar kullanmak yerine daha ucuz olan bu birimi kullanmak sistem maliyetini azaltmaktadır. Arduino IDE üzerinden *ESP8266* kütüphanelerini, yükleyerek, *NodeMCU* kartlar programlanır. Bir kez programlandıktan sonra, internet

üzerinden OTA ile tekrar programlanabilir duruma gelir. Dosya sistemi; yapılandırma dosyalarını, Web sunucusu içeriğini ve diğer dosya verilerini depolayarak yönetir. Arduino ortamında kullanılan dosya yerleşimi Şekil 1'de [24] göstermektedir.



Şekil 1: Arduino Ortamında Kullanılan Dosya Yerleşimi

3.4. ESP8266

ESP8266, seri haberleşme ile kablosuz olarak internet ağına bağlanabilen bir birimdir. Ucuz ve kolay kullanımından dolayı IoT projelerinde çok yaygın olarak kullanılmaktadır. Arduino'yu internete bağlamanın en kolay ve en ucuz yolu *ESP8266 Wi-Fi* birimini kullanmaktır. Ortamda bulunan kablosuz ağlara bağlanabileceği gibi, kendi internet ağını yayarak diğer aygıtların bu ağa bağlanabilmesine de olanak sağlamaktadır [17].

3.5. OTA kütüphaneleri, fonksiyonları ve işlevleri

ESP8266httpUpdate kütüphanesinde, *ESPhttpUpdate* sınıfı bulunmaktadır. Bu sınıf güncellemeleri kontrol edebilir ve *http web* sunucusundan bir ikili dosya indirebilir. Güncellemeler, ağdaki ya da internetteki herhangi bir *ip* ya da alan adından indirilebilmektedir. Bu kütüphanede basit ve güvenli güncelleme işlemlerini yapan fonksiyonlar bulunmaktadır. Basit ve güvenli güncelleme işlemlerini yapan fonksiyonlar aşağıda verilmektedir.

a) Basit Güncelleme İşlemi

• `ESP8266HTTPUpdate::update(const char * url, const char * current_version, const char * httpsFingerprint)`

b) Güvenlikli Güncelleme İşlemi

• `ESP8266HTTPUpdate::update(const char * host, uint16_t port, const char * url, const char * current_version, bool https, const char * httpsFingerprint)`

• `ESP8266HTTPUpdate::update(String host, uint16_t port, String url, String current_version, bool https, String httpsFingerprint)`

Basit güncelleyici, işlev her çağrıldığında dosyayı indirir. Güvenlikli güncelleyici ise, güncelleme işlevi için sunucudaki bir komut dosyasını göstermesi mümkündür. Sunucu tarafı komut dosyası, güncelleme yapılıp yapılmayacağını kontrol etmektedir. Sunucu isteği işlemeye, basit güncelleyici için sunucunun sadece güncelleme için ikili dosyayı vermesi gerekir. Güvenlikli güncelleme işlemini yapan fonksiyonlardaki *httpsFingerprint* parametresi, sertifikanın *SHA1* parmak izidir. Güvenlikli güncelleme yönetimi için bir *betiğin* (örneğin bir PHP betiği) sunucu tarafında çalıştırılması gerekir. Her güncelleme talebinde *ESP*, *HTTP* başlıklarındaki bazı bilgileri sunucuya gönderir [17]. Örnek başlık verileri aşağıdaki şekilde verilmektedir.

```
[HTTP_USER_AGENT] => ESP8266-http-Update
[HTTP_X_ESP8266_STA_MAC] => 18:FE:AA:AA:AA:AA
[HTTP_X_ESP8266_AP_MAC] => 1A:FE:AA:AA:AA:AA
```

```
[HTTP_X_ESP8266_FREE_SPACE] => 671744
[HTTP_X_ESP8266_SKETCH_SIZE] => 373940
[HTTP_X_ESP8266_CHIP_SIZE] => 524288
[HTTP_X_ESP8266_SDK_VERSION] => 1.3.0
[HTTP_X_ESP8266_VERSION] => DOOR-7-g14f53a19
```

Bu bilgilerle *PHP betiği*, bir güncelleme gerekip gerekmediğini kontrol edebilir. *MAC* adresine göre farklı ikili dosyalar teslim etmek de mümkündür.

3.6. OTA işlemlerinde diğer güvenlik yöntemleri

OTA işlemlerinde güvenlik için bazı çözüm önerileri bulunmaktadır. Bunlar, güvenli önyükleme *V1*, güvenli önyükleme *V2* ve OTA aracılığıyla güncelleme şeklindedir.

1- *Güvenli Önyükleme V1 (Secure Boot V1)*: Güvenli Önyükleme, entegre devre üzerinde sadece kodunuzun çalışabilmesini sağlayan bir özelliktir ve *AES* tabanlı güvenli önyükleme şemasını kullanır. Flash'tan yüklenen veriler her sıfırlamada doğrulanır. Güvenli Önyükleme, Flash Şifreleme özelliğinden ayrıdır ve Flash içeriği şifrelenmeden güvenli önyüklemeyi kullanabilirsiniz. Fakat güvenli bir ortam için her ikisinin de aynı anda kullanılması gerekmektedir. [25]

2- *Güvenli Önyükleme V2 (Secure Boot V2)*: Güvenli Önyükleme *V2*, *RSA* tabanlı uygulama ve önyükleyici doğrulamasını kullanır. Ayrıca, önyükleyiciyi imzalamadan *RSA* şemasını kullanarak uygulamaları imzalamak için bir referans olarak da kullanılabilir. Güvenli Önyükleme, önyüklenmekte olan her bir yazılım parçasının imzalanmış olup olmadığını kontrol ederek bir aygıtı herhangi bir yetkisiz (yani imzasız) kod çalıştırmaya karşı korur. Bir *ESP32*'de, bu yazılım parçaları, ikinci aşama önyükleyiciyi ve her bir uygulama ikili dosyasını içerir. İlk aşama önyükleyicinin *ROM* kodu olduğu için imza gerektirmediğini ve dolayısıyla değiştirilemeyeceğine dikkat edilmesi gerekmektedir. Güvenli Önyükleme *V2*'nin avantajları aşağıdaki gibidir. [26]

- *RSA* ortak anahtarı cihazda depolanır. İlgili *RSA* özel anahtarı gizli bir yerde tutulur ve cihaz tarafından asla erişilmez.
- Üretim sırasında entegre devre üzerinde yalnızca bir ortak anahtar oluşturulabilir ve saklanabilir.
- Uygulamalar ve yazılım önyükleyici için aynı görüntü formatı ve imza doğrulama yöntemi uygulanır.
- Cihazda hiçbir sır saklanmaz. Bu nedenle, zamanlama veya güç analizi gibi pasif yan kanal saldırılarına karşı duyarsızdır.

3- *OTA Aracılığıyla Güncelleme*: IoT cihazları dağıtıldıktan sonra, yeniden programlamak ya da güncellemek için fiziksel erişim olmayabilir. Önceden plan yapmak ve gömülü sistemi ya da IoT cihazlarını güncellemek için güvenli bir mekanizmaya sahip olmak çok önemlidir. OTA aracılığıyla güncelleme, donanım yazılımını (*Firmware*) şifreler. [27]

4. Yerel yönetim/uzaktan erişim sistemi üzerine bir uygulama

Tümleşik yönetim başlığı altında, genel IoT sistemlerinde tanıtılmış olan yerel yönetim/uzaktan erişim sisteminin [17] önerilen IoT hizmeti şeklinde uygulanması gerçekleştirilmiştir.

4.1. Tümleşik yönetim ortamı

Tümleşik yönetim; yönetimin yürütüldüğü bilgisayarlar ve yazılımlar, kullanıcılar, internet erişim ortamı ve kurulu sahada bulunan uçbirimlerden oluşan bir yapı üzerinde çalışır. Örnek kurulu sahada, uç birimler ve bir adet *PC*, veri tabanı, hizmet yönetimi ve *IoT* yönetimi bulunmaktadır. Burada *PC*'ye uzaktan bağlı bir yönetici vardır.

Tümleşik yönetim sisteminde *Xampp* kullanılmıştır. *Xampp*, yerel web sunucusu oluşturmayı sağlayan, *Apache* web sunucusu ve diğer uygulamaları kapsayan bir dağıtımdır. Bu isterlerle gerçekleştirilen uygulamada, sunucu yazılımı *PHP* dili ile yazılmıştır. Verileri tutmak için hem açık kaynak kodlu olan *PHP* ile hem de *PHP* ile çok iyi bir şekilde çalışan *MySQL* veri tabanı seçilmiştir. Uçbirimler için, *NodeMCU* cihazları kullanılmıştır. Uçbirimlerin yazılım ve donanım olarak program yüklemeye hazır olması gerekir.

4.2. Tümleşik yönetimde yapılan işlemler

Tümleşik yönetimde yapılan temel işlemler, kullanıcı işlemleri, hizmet yönetim işlemleri, IoT yönetim işlemleri ve uçbirim işlemleridir. Tablo 1'de, yapılan temel işlemler ve alt işlemleri verilmektedir.

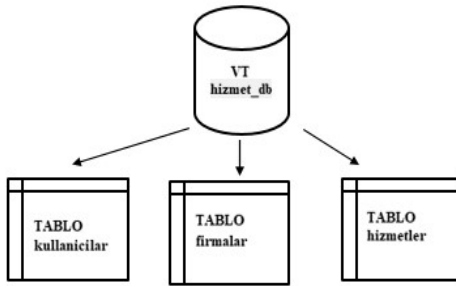
Tablo 1: Tümleşik Yönetimde Yapılan İşlemler

Yapılan Temel İşlemler	Alt İşlemler
Kullanıcı İşlemleri	<ul style="list-style-type: none"> • Kullanıcı kaydı • Firma kaydı/kendine ait olanları listeleme • Hizmet alma/kendine ait olanları listeleme • IoT uçbirim listesi • Yeni sürüm program gönderme • Uçbirim yeni sürüm program sorgulama/durum
Hizmet Yönetim İşlemleri	<ul style="list-style-type: none"> • Tüm kullanıcıların listesi • Tüm firmaların listesi • Tüm hizmetlerin listesi • Kullanıcıdan gelen hizmet talebini onaylama • IoT uçbirim listesi • Uçbirim kayıtları/durumları listesi • Kullanıcıdan gelen istekleri IoT yönetime bildirme

IoT Yönetim İşlemleri	<ul style="list-style-type: none">• Yeni sürüm program sorgulama/alma• Yeni sürüm program gönderimi (uçbirime)• Uçbirim kaydı alma• Uç birim kayıtlarını/durumlarını hizmet yönetimine bildirme
Uçbirim İşlemleri	<ul style="list-style-type: none">• Kendini kayıt etme• Yeni sürüm program sorgulama• Yeni sürüm program alma (OTA)

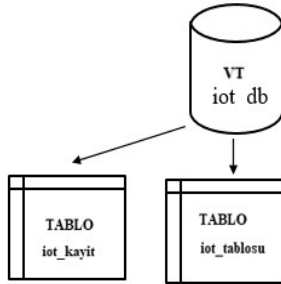
4.3. Tümleşik yönetim veri tabanı tasarımı

Tümleşik yönetimde, hizmet yönetimi için *hizmet_db* adlı veri tabanı oluşturulmuştur ve Şekil 2’de verilmektedir.



Şekil 2: Tümleşik Yönetimdeki *hizmet_db* Adlı Veri tabanı

Tümleşik yönetim sisteminin IoT yönetimi için *iot_db* adlı veri tabanı oluşturulmuştur ve Şekil 3’te verilmektedir.



Şekil 3: Tümleşik Yönetimdeki *iot_db* Adlı Veritabanı

4.4. Tümleşik yönetimde senaryolar

Tümleşik yönetimde senaryolar aşağıdaki şekilde oluşturulmuştur:

- a) Kullanıcı kaydı, firma kaydı ve hizmet alımı olmayan kullanıcının sistemde işlem yapamaması.
- b) Firma hizmet alım talebini gönderdikten sonra firmanın hizmeti kullanabilmek için yöneticinin onay vermesini beklemesi.
- c) Uçbirimlerin bağlantısının kesilmesi durumunda bağlantı tekrar geldiğinde işlemine kaldığı yerden devam edebilmesi.
- d) Uçbirimlerin sisteme kendini kaydettirmesi.
- e) Kullanıcıların uçbirimlere doğrudan müdahale edememesi ve kullanıcılar isteklerini hizmet yönetimine bildirerek IoT yönetimi üzerinden uçbirimlere iletmesi.

4.5. Kullanıcının sisteme kayıt olma durumu

Geliştirilen sistem; kayıtlı firmalar üzerinden kullanıcılara hizmet sunmaktadır. Hizmet satın alan kullanıcılar, kendi sistemine dosya gönderebilir, uçbirimlere yeni sürüm yükleyebilir, uçbirimleri listeleyebilir ve uçbirimlerin durumlarını görebilir. Bu sistemde, kullanıcı ya da yönetici girişi yapılabilmektedir. Eğer kayıtlı bir kullanıcı değilse yeni kullanıcı kaydı yapılarak giriş yapılabilmektedir. Kullanıcı firma kaydı yapmışsa kayıtlı olan firmalarını listeleme yapabilmektedir. Kullanıcı firma seçerek o firma için hizmet alımı yapabilmektedir. Sistemdeki hizmet işlemleri panelinde, kullanıcı eğer hizmet alımı yaptıysa hizmetleri listeleyebilmektedir.

Sistemdeki firmaların hizmet listesinde hizmet talebi yapılmış fakat hizmeti kullanabilmek için ödeme durumunda yöneticinin onay vermesi beklenmektedir. Yönetici ödeme durumunu ödendi olarak değiştirirse ödemesi yapılan hizmet için kullanıcı, dosya gönderme, yeni sürüm yükleme, cihaz durumu ve cihaz listesi hizmetlerinden var olanları kullanabilmektedir.

4.6. Kullanıcı işlemleri

Kullanıcılar, dosya gönderme, yeni sürüm yükleme, cihaz durumu ve cihaz listesi hizmetlerini aldığında ve yönetici ile ilgili gerekli anlaşmalar tamamlandığında firma adına bu hizmetleri kullanabilmektedir. Dosya gönderme hizmeti, kullanıcıların sisteme dosya yükleme yapabildikleri hizmettir. Kullanıcının sisteme gönderdiği dosyanın boyutunda kısıtlama yapılmıştır. Kullanıcı, 4 MB’ dan büyük dosyaları sisteme yükleyemez. Bu boyut yönetici tarafından değiştirilebilir.

Yeni sürüm yükleme hizmeti, kullanıcının sistemdeki dosyalardan istediğini istediği uçbirimine yükleme isteğini hizmet yönetimine bildirilmesidir. Hizmet yönetimi de IoT yönetimine bu isteği ileterek IoT yönetimi onay verirse uçbirime yükleme yapılmaktadır. Cihaz durum hizmeti, kullanıcı tarafından herhangi uçbirimin verilerinin gözlenebilmesidir. Burada programa göre kurgulanmış sensörden gelen veriler eş zamanlı gözükmektedir. Cihaz listesi hizmetinde, kullanıcının kendine ait uçbirimlerinin listesini görebilmesidir. Bu listede cihazın aktif ya da pasif olması durumları eş zamanlı gözükmektedir.

Kullanıcı hizmet satın aldığı IoT yönetim paneline erişebilmektedir. Kullanıcı, sisteme dosya göndermek istediğinde hizmet satın alınmış ve yönetici tarafından onay verilmişse, dosya gönderme işlemi kullanıcı tarafından yapılabilmektedir.

4.7. Yönetici işlemleri

Yöneticinin yapabildiği işlemler, tüm kullanıcıları listeleme, tüm firmaları listeleme, hizmetleri listeleme, hizmet yönetimi, firma kaydı şeklindedir.

Tüm kullanıcıların listesine bakıldığında, yönetici ve kullanıcı şeklinde yetki alanı oluşturulmuştur. Yönetici isterse kullanıcıların bilgilerini güncelleyebilmektedir. Yönetici isterse istediği kullanıcıyı sistemden silebilir. Yönetici isterse hizmetlerin bilgilerini güncelleyebilmektedir. Yönetici isterse

istediği hizmeti sistemden silebilir. Kullanıcıların istediği hizmeti kullanabilmesi için hizmet talebi “Var” şeklinde ve ödeme durumu yönetici tarafından “Ödendi” olarak onaylanmış olması gerekmektedir. Örneğin *kullanıcı_id* değeri 3 *hizmet_id* değeri 30 olan kullanıcı ödeme durumu “Ödendi” yazsa da cihaz durum hizmetinde “Yok” yazdığı için cihaz durum hizmeti kullanılamaz.

4.8. Tümleşik Yönetimde Kullanılan Cihazlar ve Teknolojiler

Tümleşik yönetimde, kurulu saha içinde, veri tabanı, hizmet ve IoT yönetimleri barındıran PC ve uç birim olarak NodeMCU cihazları kullanılmıştır. NodeMCU üzerine yüklenen uygulamalar Arduino’da yazılmıştır. Bu sistemde, web sunucu yazılımı olan Xampp ile yerel bir şekilde bilgisayarda çalışılmıştır. Xampp’da, phpmyadmin kurulu bir şekilde gelmektedir. Phpmyadmin, Php ile yazılmış açık kaynak kodlu bir araçtır. MySQL veri tabanının yönetimi bu araç üzerinden sağlanmaktadır.

4.9. Tümleşik yönetimde kullanılan fonksiyonlar ve kütüphaneler

Hizmet Fiyatını Hesaplayan Fonksiyon: Kullanıcı hizmet seçimini yaptıktan sonra fiyatını hesaplayan basit bir fonksiyon yazılmıştır. Bu fonksiyonda 4 adet hizmet durumu bulunur.

Tümleşik yönetim sistemi uygulaması için kullanılan fonksiyonlar Tablo 2’de açıklamalarıyla verilir.

Tablo 2: Kullanılan Fonksiyonlar

Kullanılan Fonksiyonlar	Açıklama
void wifi_baglan()	WiFi bağlantısı yapar
void iot_guncelle(String ver)	IoT güncelleme işleri yapar
void iot_islemler()	IoT okuma ve IoT yazma işlemi yapar
void updateUI()	Arayüz güncelleme
void handlePage()	El sıkışma

Bu sistemde kullanılan kütüphaneler ise Tablo 3’te verilmiştir.

Tablo 3: Kullanılan Kütüphaneler

Kullanılan Kütüphaneler	Açıklama
#include <ESP8266WiFi.h>	WiFi ile ilgili işlevleri yapmak için gereklidir.
#include <WiFiClient.h>	Web tarayıcısına istek göndermek için gereklidir.
#include <ESP8266WebServer.h>	ESP8266WebServer, web sunucusu
#include <ESP8266HTTPClient.h>	ESP8266HTTP, http İstemcisi
#include <ESP8266HTTPUpdateServer.h>	ESP8266HTTPUpdate Server, Güncelleme Sunucusu
#include <ESP8266httpUpdate.h>	ESP8266httpUpdate, http güncelleme
#include <EmberAJAX.h>	Ajax kütüphanesi

Bu çalışmada, IoT aygıtlarının yeni program güncellemesi kablosuz olarak yapılması gerekir. Bu da, aygıtların kötü amaçla saldırıya uğrayabileceği ya da başka bir program yüklenme ihtimallerini ortaya çıkarmaktadır. Saldırıya uğramanın olasılığını azaltmak için, bu yüklemeleri bir şifreyle, belirli bir OTA bağlantı noktası vb. seçerek korunmalıdır. Güvenliği artırmak için, Arduino OTA kütüphanesi ile sağlanan işlevsellik kontrol edilebilir [24].

```
void setPort(uint16_t port);
void setHostname(const char*
hostname);
void setPassword(const char* password);
```

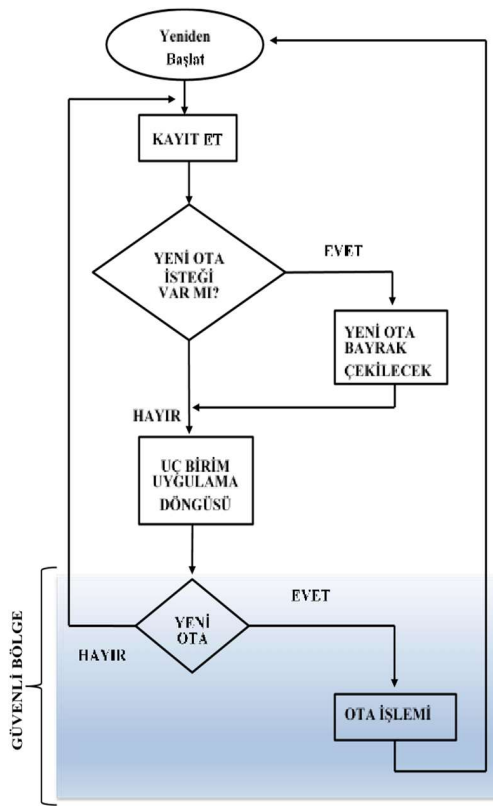
Bunun gibi, belli koruma işlevleri önceden oluşturulmuştur ve geliştirici tarafından herhangi bir ek kodlama gerektirmez. *Arduino OTA* ve *espot.py*, karşıdan yüklemeyi doğrulamak için *Digest-MD5* kullanılır. Aktarılan verilerin bütünlüğü, *ESP* tarafında *MD5* kullanılarak doğrulanır [24].

4.10. Tümleşik yönetimin çalışma prensibi

Bu çalışmanın birinci aşamasında, hizmet yönetimi ve IoT yönetimi aynı PC üzerinde bulunmaktadır. Her iki yönetimin de ayrı ayrı veri tabanı oluşturulmuştur. Aslında tek bir veri tabanı da yeterli olur. Çünkü bütün sistem yereldedir. Sonraki aşama uygulamalarına yönelik veri tabanları ve yönetim sistemleri ayrı ayrı oluşturulur. Bu sistemde, kullanıcılar uzaktan sisteme erişebilmektedir. Kullanıcıların doğrudan IoT yönetime müdahale edebilmesi güvenlik nedeniyle istenmez. Sisteme çok fazla cihaz bağlanması durumunda, bu cihazlara tek tek elle program yüklemek iş yükünü artırır ve zaman kaybına neden olur. Belki de bu cihazlara yüklenmesi gereken programların eş zamanlı yüklenmesi gerekebilir. Parça parça programlar büyük bir sistemi oluşturabilir.

4.11. Uç birimde OTA sağlanması

Şekil 4'te verilen, uçbirim uygulama programı bir sonsuz döngü içinde çalışır. Uygulamanın kesintiye uğramaması için OTA işlemleri uçbirim uygulama programının izin verdiği güvenli bir bölgede yürütülür. Eğer bir OTA işlemi bir başka deęişle yeni bir program yüklenmesi gerçekleşecek ise uçbirim uygulama programının gereken önlemlerini örneğin aktif çıkışlarının pasife alınması ve verilerini almasına, saklamasına izin verilir. Uygulama programının izin verdiği aşamada OTA gerçekleşir ve sistem yeniden başlat noktasından (Reset) çalışmaya başlar.

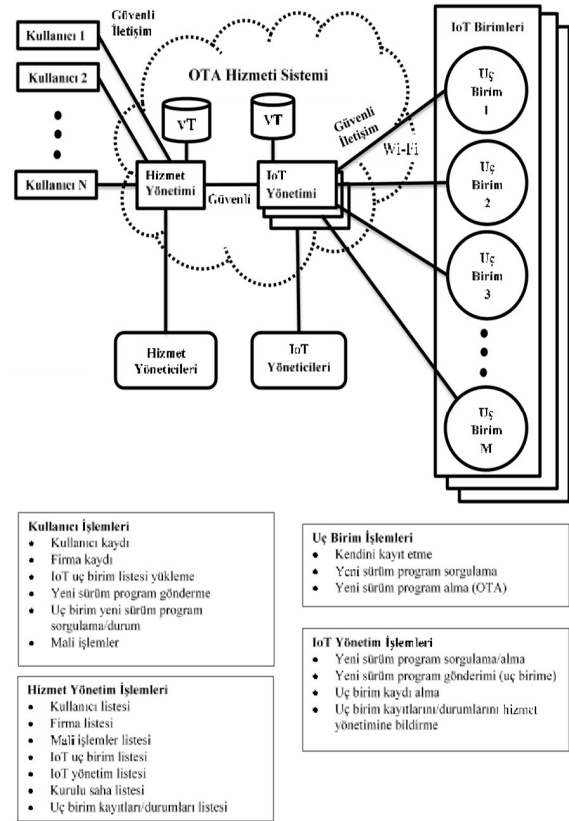


Şekil 4: Uçbirimde OTA Gerçekleştirilmesi

5. Önerilen uzaktan yönetim sistemi üzerine bir uygulama

Uzaktan program yükleme (OTA) hizmeti sistemi, sahadaki IoT uçbirimlere yeni program yükleme işlemi için firma adına güvenilir bir şekilde yerine getiren bir sistemdir. Önerilen uzaktan program yükleme hizmetinin sistem tasarımı Şekil 5'te verilmektedir. OTA hizmeti sistemi; hizmet yönetimi, IoT yönetimi, kullanıcılar, yöneticiler ve uçbirimlerden oluşmaktadır. Hizmet ve IoT yönetimlerinin, ayrı ayrı hizmet ve IoT yöneticileri vardır. OTA hizmeti sistemi olarak, uçbirimlerle kullanıcı arasında, doğrudan veri alışverişi güvenlik nedeniyle kısıtlanmış ve işlemlerin hizmet yazılımı aracılığı ile kullanıcılar, kendi uygulamalarına ait işlemlerinin, haberleşmesinin yapılması sağlamıştır. OTA hizmeti haberleşmesini ayrı bir ortam üzerinden yürütür. OTA

uygulanmasında, kullanıcı A'nın istediği uçbirime program yüklemesi için, kullanıcı A, "Sisteme kayıt yaptırdı mı?", "Kayıtlı ise sözleşmeleri imzaladı mı?" veya "Almak istediği hizmet için ücretini ödedi mi?" gibi mali aşamaları da geçmesi, sonra da isteğini hizmet yönetimine bildirmesi istenir. Eğer gelen uygun bir istek ise hizmet yönetimi isteği ve program dosyasını IoT yönetimine aktarır. IoT yönetimi de gelen programı uçbirimlere gönderir. Sonuç olarak, uçbirimde gerçek program yüklemesi yapılır.



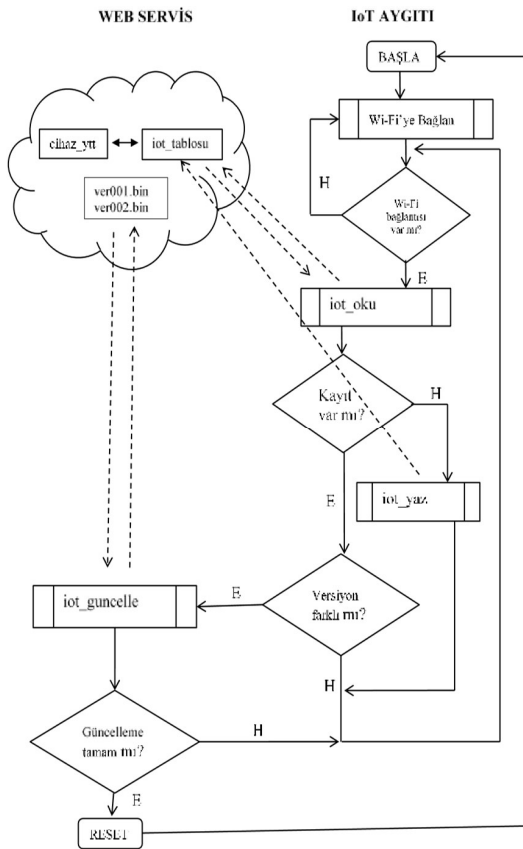
Şekil 5: Önerilen Uzaktan Program Yükleme Hizmeti

Burada Tümlşik sisteme benzer olarak *PHP* ve *MySQL* tabanında uygulanması gerçekleştirilmektedir. Bu seçim; Uzaktan Yönetim Sisteminin bir PC üzerinde gerçekleştirilmesi sağlandığı gibi kolaylıkla Bulut üzerine alınarak Web Hizmeti Sunucuları üzerinde de gerçekleştirilmesine olanak sağlamaktadır. Çalışma öncelikle *PC* üzerinde uzaktan yönetim sisteminin gerçekleştirilmesi üzerinde tamamlanmıştır. Daha sonra bir Web Sunucu üzerine yazılımlar taşınarak yönetim sisteminin çalışması gözlenmiştir.

5.1. Önerilen uzaktan yönetim sisteminin çalışma prensibi

Önerilen uzaktan yönetim sisteminin çalışma prensibi Şekil 6'da gösterilmektedir. Bu sistem çalışmaya başladığında önce, IoT tarafında *Wi-Fi* bağlantısı fonksiyonu çalışır. *Wi-Fi* bağlantısı yoksa tekrar başa döner. Eğer *Wi-Fi* bağlantısı var ise *iot_oku* fonksiyonu çalışır. *iot_oku* fonksiyonu bulut üzerindeki *iot_tablosu* ile bağlantılıdır. *iot_oku* fonksiyonu,

buluta istek gönderir ve cevabını alır. Daha sonra *iot_oku* fonksiyonunun işleri bitince kayıt varsa ve sürüm farklıysa *iot_guncelle* fonksiyonu çalışır. Eğer güncelleme tamamsa cihaz *reset*'e gider ve sonra tekrar sistemin çalışması başa döner. Sistemin çalışmasındaki diğer bir yol ise, sistem çalışmaya başladığında *Wi-Fi* bağlantısı varsa, *iot_oku* fonksiyonu işini bitirince eğer kayıt yoksa *iot_yaz* fonksiyonu çalışır. *iot_yaz* fonksiyonu işlerini bitirince *Wi-Fi* bağlantısı var mı noktasına geri döner. Sistemin çalışmasındaki diğer tüm yollar şekil üzerinde gösterilmektedir. Bu sistemde, *iot_oku.php*, *iot_yaz.php* ve *iot_guncelle.php* dosyaları bulut üzerinde bulunmaktadır ve PHP dilinde yazılmıştır. Aynı zamanda *ver001.bin* ve *ver002.bin* uzantılı dosyaları da bulut üzerinde bulunmaktadır. *iot_guncelleme* fonksiyonu ile bulut üzerinde bulunan sürüm cihaz üzerindeki farklı ise güncelleme başarılı şekilde yapılabilmektedir.



Şekil 6: Önerilen Uzaktan Yönetim Sisteminin Çalışma Prensipleri

Aşağıdaki Tablo 4'te sistemde kullanılan tetikleyiciler ve SQL sorguları gösterilmektedir.

Tablo 4: Tetikleyiciler

Tetikleyiciler	Sql Kod
istekEkle	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (NEW.ip, NEW.iot_adi, NEW.ver, NEW.yeniver, NEW.Date, NEW.Time, NEW.durum, NEW.TimeStamp, 'EKLE')
istekGuncelle	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (NEW.ip, NEW.iot_adi, NEW.ver, NEW.yeniver, NEW.Date, NEW.Time, NEW.durum, NEW.TimeStamp, 'GÜNCELLE')
istekSil	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (OLD.ip, OLD.iot_adi, OLD.ver, OLD.yeniver, OLD.Date, OLD.Time, OLD.durum, OLD.TimeStamp, 'SİL')

5.2. Önerilen uzaktan yönetim sistemindeki kullanıcı ekranı

Bu sistemde kullanıcılar, uzaktan bulut üzerinde bulunan istediği dosyayı cihazına uzaktan yükleyebilir ve kullanıcılar sistemini uzaktan takip edebilir veya bulut ortamına istediği yazılım dosyasını (OTA hizmeti bulunan ".bin" dosyası) gönderebilir ve isterse sisteme yüklediği bu dosyayı cihazına uzaktan yükleyebilir. Bu sistem bu tür hizmet almak isteyenler için hazırlanmış bir uzaktan yönetim sisteminin hizmet yazılımıdır. Sisteme kaydı olmayan sistemi hiç kullanamaz ya da sistemin sınırlı özelliklerini kullanabilir.

5.3. IoT cihaz yönetim paneli

Kullanıcılar veya yöneticiler, IoT cihaz yönetim panelinden sisteme bağlı cihazları görebilir.

5.4. IoT cihazların durum takibi

Bu sistemde IoT cihazların takibi yapılmaktadır. Cihazlar sisteme hazır duruma getirildikten sonra bu sisteme bağlandığında cihazların bilgileri veri tabanına otomatik olarak kaydedilir. Cihazın *ip* adresi, *MAC* adresi, üzerindeki sürüm adı, tarih, zaman ve durum bilgileri tutulur. Cihaz için yeni sürüm talebi yapıldığında belli bir süre sonra sistem yeni sürümü cihaza yükleyinceye kadar tekrar tekrar çalışacaktır. Eğer yeni sürüm talebi verildikten sonra internet bağlantısını keserek ve sonra cihazı tekrar sisteme bağlarsak cihaz kaldığı yerden yeni sürüm talebini yükleyinceye kadar tekrar tekrar çalışacaktır.

5.5. Kullanıcıların sisteme dosya yükleyebilme ekranı

Kullanıcıların sisteme dosya yükleyebilecekleri ekranlar oluşturulmuştur ve nasıl yükleme yapılacağı bu ekranlardan görülmektedir.

5.6. Yönetici ekranı

Yönetici sisteme bağlı olan tüm cihazları yönetici ekranından görebilir. Sisteme 3 adet *NodeMCU* cihaz bağlandıktan sonra sistemde bu cihazların aktif olduğu gözlenmiştir. Bu sistemde cihazların sürümlerini değiştirebileceğimiz ekran da bulunmaktadır.

Yerel yönetim/uzaktan erişim sisteminde, IoT yönetimi PC üzerinde bulunmaktadır. Kullanıcılar PC'ye bağlı uygulamalar ya da programlar vasıtasıyla isteklerini IoT yönetimine iletirler. PC yerine *RaspberryPi* gibi cihazlar kullanılarak IoT yönetimi işlemleri daha küçük ve maliyeti daha az olan cihazlar da kullanılabilir. Aynı zamanda birden çok PC kullanmak yerine *RaspberryPi* kullanılması hizmet almak isteyen kişilerin ya da şirketlerin maliyetini düşürür.

5.7. Uzaktan yönetim sisteminin bir web sunucusuna taşınması

Bu çalışmada *nilguninceis.com* alan adı, web sunucu hizmeti "www.ekonomikhost.net" [28] adresinden sağlanmıştır.

Öncelikle alınan hizmet karşılığında verilen <http://nilgunincereis.com:8880> adresinden, *Plesk* web sunucu paneline [29] girilerek, kullanıcı adı ve şifre ile giriş yapılmaktadır.

6. Önerilen uzaktan yönetim sisteminin geliştirilmesi

Önerilen uzaktan yönetim sisteminin geliştirilmesi için yönetimlerin birer bulut hizmeti yazılımına dönüştürülerek tamamen bulut üzerinde dağıtık bir sistem olarak oluşturulduğunda sistemde oluşabilecek kesintilerin nedenleri belirlenmiştir. Bunlar:

- IoT yönetimi ve uçbirimler arasında güvenli iletişimin kopması ya da iletişimin bilinmeyen güvensiz bir iletişime dönüşmesi,
- OTA hizmeti sistemi içindeki hizmet yönetimi ile IoT yönetimi arasında güvenlik protokolünün kopması ya da bilinmeyen bir protokole dönüşmesi,
- Kullanıcılar ve hizmet yönetimi arasında kullanıcıdan kaynaklı güvenli iletişimin bozulması,
- Kullanıcılar ve hizmet yönetimi arasında güvenli iletişimin kopması ya da iletişimin bilinmeyen güvensiz bir iletişime dönüşmesi,
- İletişimin gecikmesi,

olarak belirlenmiştir.

Daha güvenli bir sistem oluşturmak için güvenlik seviyeleri oluşturulmalı ve bu seviyelerde uygulanması gereken tedbirler belirlenerek hangi seviyede ise o seviyedeki tedbirler uygulanmalıdır.

7. Deneysel sonuçlar

Sistemin deneysel sonuçlarını görebilmek için oluşturulan sistem 21 gün boyunca incelenmiştir. Bu süre boyunca sisteme 4 adet *NodeMCU* cihazı bağlanmış ve sistemde bu cihazların aktif ya da pasif olduğu durumlar görülmektedir. Bu cihazların 455 işlemi aktif durumdayken 45 işlemi pasif durumdadır. Bunun yanında sistemde ekleme, silme ve güncelleme işlem türlerini ve 15 farklı *ip* bilgisini içeren toplam 500 işlem yapıldığı tespit edilmiştir.

Tablo 5'te görüldüğü gibi, sistem üzerindeki programların *Versiyon 1* ve *Versiyon 2* türlerinin işlem durumları incelenmiştir. *Versiyon 1*'deki işlemlerin ekleme sayısı 16, silme sayısı 13, güncelleme sayısı 299 iken, *Versiyon 2*'deki işlemlerin ekleme sayısı 13, silme sayısı 13, güncelleme sayısı 146 olmuştur. *Versiyon 1*, *Versiyon 2*'ye göre %91 daha fazla işleme sahiptir.

Tablo 5: Versiyonların İşlem Durumlarının Sayıları

	Versiyon 1	Versiyon 2
Ekleme	16	13
Silme	13	13
Güncelleme	299	146
Toplam İşlem	328	172

OTA kapsamındaki uygulamalarda; güvenlik, güvenilirlik, doğrulama, içerik dağıtım ağı (CDN) ve bulut yükü dengesi gibi güncel problemler için çözüm önerilerinin deneysel çalışmaları da yapılması gerekmektedir. Bu çalışmada, OTA işlemlerindeki güvenlik yöntemlerinin önemi vurgulanmaktadır.

8. Sonuçlar

Günümüzde kullanımı yaygınlaşmış IoT sistemlerine ve uçbirimlerine yönelik doğrudan program yükleme işlemleri firma içinde kapalı olarak yapılmaktadır. Firma içinde yapılmasının en önemli nedeni güvenli bir ortamda program yüklemenin basitçe gerçekleşmesidir. Ancak uygulamada çeşitli sahalara yerleştirilmiş IOT cihazlarına yeni bir program sürümünün doğrudan yüklenmesi maliyetli ve kolay olmayan bir işlem olacaktır. Bu çalışmada uzaktan program yüklemeyi karşılayan OTA işlemlerinin bir hizmet olarak sunulması önerilmiştir. Güvenlikli uzaktan program yükleme işlemlerinin firma içinde tasarlama, geliştirme ve bakım maliyetlerinin karşılanması gerekmektedir. Firma içinde çözülmesi gereken OTA işlemlerinin, her türlü çalışma türlerine göre kesintisiz ve hatasız yerine getirilmesini ve güvenlik koşullarını da sağlayan genel kullanıma açık, maliyeti düşük hizmetler olarak sunan bir bilişim sistemi tasarlanıp gerçekleştirilmiştir. Bu çalışma ile ilk defa bir açık kullanımlı uzaktan program yükleme hizmeti sistemi geliştirilmiştir. Geliştirilen bilişim sistemi benzer başka problemlere de uyarlanabilir. Sistemin faydalanıcıları olarak; özellikle gelişmiş OTA sistemi gerçekleştirme ve sürdürme maliyetlerine girmesi mümkün olmayan mikro düzeyde veya küçük düzeyde firmaların projelerinde kullanımına bir hizmet sunulmuş olmaktadır. Bu ortamda küçük adımlarla başlayan bir firma projeleri geliştikçe artan OTA hizmeti gereksinimi

karşılanmakta ve sadece kendisine yönelik bir alt sistem kurulmasına imkân verilecek şekilde büyümesi ölçeklenebilmektedir. Çalışmada tasarlanan sistem, *MYSQL* veri tabanı kullanan *PHP* dilinde yazılmış bir uzaktan yönetim sistemi yazılımdır. Bu sistemin geliştirilmesi için yönetim yazılımları birer bulut hizmeti yazılımına dönüştürülebilir.

Güvenlik kapsamı bakımından program yükleme işlemi bir şifreyle ve belirli bir OTA bağlantı noktası seçimi ile korunmaktadır. Güvenliği daha da artırmak için, *Arduino OTA* kütüphanesi ile sağlanan işlevsellikler kontrol edilmektedir. Bu çalışmanın mevcut çalışmalara olan üstün yanları, belli koruma işlevleri önceden oluşturularak, geliştirici tarafından her hangi bir ek kodlamaya gerek duyulmamasıdır. Ayrıca, Önerilen uzaktan yönetim sisteminin kesintiye uğramaması için OTA işlemleri uçbirim uygulama programının izin verdiği güvenli bir bölgede yürütülmüştür. Gerçeklenen sistemde tüm bileşenlerin sayısı arttıkça zaman bakımından kayıp vermeden daha güvenli bir sistemi oluşturma ve yönetmenin zorluğu vardır. Çalışmada bunları karşılayacak ileri uygulama önerileri de verilmiştir.

9. Kaynaklar

- [1] S., Wagle, T., Sathe, G., Vamburkar, A., Gaikaiwari, "Regression Based Prediction Algorithm for Remote Controlling of IoT Based Applications", 2015 International Conference on Computing and Network Communications (CoCoNet), IEEE, 2015.
- [2] Wikipedia – Control Systems, 2021.
- [3] M., Bak, "Control systems with constraints", PhD Thesis, Technical University of Denmark, 2000.
- [4] K., Kang, Z., Pang, L. D., Xu, L., Ma, C., Wang, "An Interactive Trust Model for Application Market of the Internet of Things", IEEE Transactions On Industrial Informatics, VOL. 10, NO. 2, 2014.
- [5] A., Behura, A., Narayan, A. K., Ray S. K., Pani, "A Complete Model for IOT Application", Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017), ISBN:978-1-5386-1959- 9, IEEE Xplore Compliant - Part Number:CFP17M19-ART, 2017.
- [6] S. A., Al-Qaseemi, H.A., Almulhim, M.F., Almulhim, S.R., Chaudhry, "IoT Architecture Challenges and Issues: Lack of Standardization", Future Technologies Conference, San Francisco, United States, IEEE, 2016.
- [7] C., Wang, M., Daneshmand, M., Dohler, X., Mao, R. Q., Hu, H., Wang, "Guest editorial - special issue on Internet of things (IoT): Architecture protocols and services", IEEE Sensors Journal, vol. 13, no. 10, pp. 35053510, 2013.
- [8] M., Zdravkovic, M., Trajanovic, J., Sarraipa, R., Lezoche M., Jardim-Gonçalves, "Survey of Internet-of-Things platforms", 6th International Conference on Information Society and Technology, ICIST 2016, Kopaonik, Serbia, pp. 216-220, 2016.
- [9] A., Rajalakshmi, H., Shahnasser "Internet of Things using Node-Red and Alexa", 2017 17th International Symposium on Communications and Information Technologies (ISCIT), 2017.
- [10] S., Dey, A., Roy, S., Das, "Home automation using Internet of Things", Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual, 2016.
- [11] G., Cuijuan, M., Changyun, W., Zhigang, X., Lina, "Research and Implementing of Remote Update for Network Telephone System Program Based on Ethernet", 2010 2nd International Conference on Signal Processing Systems (ICSPS), 2010.
- [12] W., Tarneberg, V., Chandrasekaran, M., Humphrey, "Experiences creating a framework for smart traffic control using AWS IOT", IEEE/ACM 9th International Conference on Utility and Cloud Computing, 2016.
- [13] D. H., Kang, M. S., Park, H. S., Kim, D.Y., Kim, S.H., Kim, "Room temperature control and fire alarm/suppression IoT service using MQTT on AWS", International Conference on Platform Technology and Service, 2017.
- [14] H., He, Z. H., Ma, X., Li, H., Chen, W., Shao, "An approach to estimating cost of running cloud applications based on AWS", 19th Asia-Pacific Software Engineering Conference, 2012.
- [15] <https://www.businesswire.com/>, "Dell Enters Embedded PC Market with New Embedded Box PCs, Helping Smart Systems Connect to the Internet of Things", 06.03. 2021,
- [16] I.F., Akyildiz, W., Su, Y., Sankarasubramaniam, E., Cayirci, "Wireless sensor networks: a survey", Elsevier, Computer Networks, 38 (2002) 393–422, 2002.
- [17] N. İncereis, B. T., Akgün, "A Remote Program Loading Service Design and Implementation", Turkish National Software Engineering Symposium (UYMS), IEEE, 2020.
- [18] A., Kanda, T., Kurafuji, K., Takeda, T., Ogawa, Y., Taito, K., Yoshihara, M., Nakano, T., Ito, H., Kondo, T., Kono, "A 24-MB Embedded Flash System Based on 28-nm SG MONOS Featuring 240-MHz Read Operations and Robust Over-the-Air Software Update for Automotive Applications", IEEE Solid-State Circuits Letters, Vol. 2, NO. 12, 2019.
- [19] X., Cao, J., Xu, K., Huang, "Cooperative Interference Management for Over-the-Air Computation Networks", IEEE Transactions On Wireless Communications, Vol. 20, NO. 4, 2021.
- [20] A., Ghosal, S., Halder, M., Conti, "STRIDE: Scalable and Secure Over-The-Air Software Update Scheme for Autonomous Vehicles", ICC 2020 - 2020 IEEE International Conference on Communications (ICC), IEEE, 2020.
- [21] M. L., Manna, P., Perazzo, L., Treccozi, G., Dini, "Assessing the Cost of Quantum Security for Automotive Over -The-Air Updates", 2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021.
- [22] J. A., Stankovic, "Research directions for the internet of things", Internet of Things Journal, IEEE, 2014.
- [23] <https://www.endustri40.com/bulut-bilisim-cloudcomputing-nedir/>, 04.03.2021.
- [24] <https://media.readthedocs.org/pdf/arduinoesp8266/latest/arduinoesp8266.pdf>, Ivan Grokhotkov, ESP8266 Arduino Core Documentation Release 2.5.038-g95cf925, Feb 26, 2019.

- [25] <https://docs.espressif.com/projects/espressif/en/latest/esp32/security/secure-boot-v1.html>, 27.01.2022.
- [26] <https://docs.espressif.com/projects/espressif/en/latest/esp32/security/secure-boot-v2.html>, 27.01.2022.
- [27] <https://www.lab4iot.com/2021/02/21/esp32-secure-firmware-update-over-the-air-ota/>, 27.01.2022.
- [28] www.ekonomikhost.net, Ekonomikhost İnternet ve Bilişim Hizmetleri, 2021.
- [29] <https://www.plesk.com>, Plesk International GmbH, 2021.

Özgeçmişler



Bekir Tevfik Akgün, İstanbul D.M.M.A. Elektrik Mühendisliği Lisans eğitimini 1981 yılında, İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar Mühendisliği Yüksek Lisans ve Doktora eğitimini 1984 ve 1991 yıllarında tamamlamıştır. 1992 yılında Yrd. Doç., 1996 yılında Doçent ve 2002 yılında Profesör unvanını almıştır. 1982-1986 yılları arasında Yıldız Teknik Üniversitesi'nde Araş. Gör., 1986-2000 yılları arasında İstanbul Teknik Üniversitesi'nde Araş. Gör. ve Öğretim Üyesi olarak çalışmıştır. 2000-2008 yılları arasında Yıldız Teknik Üniversitesinde Öğretim Üyesi olarak akademik ve yönetici görevlerini yürütmüş ve Dekanlık görevini tamamlayarak emekli olmuştur. 2009 yılından günümüze İstanbul Okan Üniversitesi Bilgisayar Mühendisliği Bölümünde Öğretim Üyesi olarak çalışmaktadır. Ohio State Üniversitesinde kısa süreli konuk Öğretim Üyesi olarak araştırma etkinliklerinde bulunmuştur. Uzmanlık alanları arasında Gömülü Sistemler, Akıllı Sistemler, Bilgisayarda Grafik ve Bilgisayar Oyunları yer almaktadır.



Nilgün İncereis, Ondokuz Mayıs Üniversitesi Fizik Öğretmenliği, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği lisans eğitimlerini ve Fizik Yüksek Lisans eğitimini 2009, 2012 ve 2012 yıllarında, İstanbul Okan Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans eğitimini 2019 yılında, Abant İzzet Baysal Bilgisayar Mühendisliği lisans eğitimini 2020 yılında tamamlamıştır. 2020 yılından beri İstanbul Okan Üniversitesi Bilgisayar Mühendisliği'nde doktora yapmaktadır. 2013-2020 yılları arasında İstanbul Okan Üniversitesi'nde, 2020-2021 yılları arasında Zonguldak Bülent Ecevit Üniversitesi'nde öğretim görevlisi olarak çalışmıştır. 2021 yılından günümüze Bartın Üniversitesi Uzaktan Eğitim ve Uygulama ve Araştırma Merkezinde öğretim görevlisi olarak çalışmaktadır. Çalışma alanları bulut bilişim, nesnelerin interneti, makine öğrenmesi ve derin öğrenme üzerinedir.
