

ANDROİD THINGS İŞLETİM SİSTEMİ İLE BİR ALARM SİSTEMİ YAZILIMININ TASARLANMASI VE UYGULANMASI

Sadık Arslan^{1*}, Barış Tekin Tezel², Moharram Challenger³, Geylani Kardaş⁴

¹ Kent Kart Ar-Ge Merkezi, Kent Kart Ege Elektronik A.Ş

sadik.arslan@kentkart.com.tr

² Bilgisayar Bilimleri Bölümü, Dokuz Eylül Üniversitesi

baris.tezel@deu.edu.tr

^{3,4} Uluslararası Bilgisayar Enstitüsü, Ege Üniversitesi

moharram.challenger@ege.edu.tr, geylani.kardas@ege.edu.tr

* Sorumlu Yazar

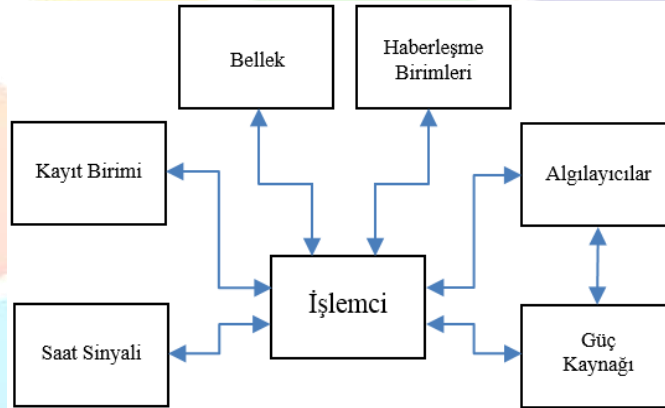
Özet

Gömülü sistemlerde Nesnelerin İnternet'inin (IoT) kullanımı için en yeni özel işletim sistemlerinden biri Google firmasının Android Things'dir. Bu bildiri Android Things işletim sistemine dayalı gömülü sistem yazılımlarının geliştirilmesi için bir yaklaşım sunulmuş; gerçekleştirilen bir alarm sistemi uygulaması üzerinden de söz konusu işletim sisteminin Raspberry Pi platformu üzerinde gömülü sistem yazılımı geliştirmesinde kullanılmasına yönelik elde edilen deneyimler ve niteliksel bir değerlendirme ortaya konmuştur. Ölçüm sonuçları geliştirilen IoT yazılımının Android Things üzerinde çalışmasının klasik Android işletim sistemi üzerinde çalıştırmaya kıyasla yaklaşık %35 daha hızlı olduğunu göstermiştir.

Anahtar Kelimeler: Nesnelerin İnterneti; Android Things işletim sistemi, Raspberry Pi Geliştirme Platformu

1. Giriş

Genel olarak yavaş sayılabilecek bir işlemci, bir bellek ve diğer yardımcı birimler içeren gömülü sistemler, özel amaçlar için tasarlanmış ve önceden tanımlanmış görevleri yerine getiren sistemler olarak tanımlanabilir. Gömülü sistemlerde Gömülü Linux, JavaOS, LynxOS, Moblinux, Windows CE gibi birçok işletim sistemi kullanılmaktadır [1]. Şekil 1'deki gibi bir gömülü sistemin yapısında temel olarak işlemci, tüm birimleri besleyen güç kaynağı, bellek, kayıt birimleri, haberleşme arayüzleri, algılayıcılar gibi bölümler bulunmaktadır.



Şekil 1: Basitleştirilmiş gömülü sistem blok çizimi

Günümüzde oldukça sık duyulmaya başlanan Nesnelerin İnterneti, (ing. Internet of Things) (IoT) benzersiz bir şekilde adreslenebilir nesnelerin kendi aralarında oluşturduğu bir ağıdır ve bu ağdaki nesneler belirli bir protokol ile birbirleriyle iletişim halindedirler. IoT içerisinde cihazlar çeşitli haberleşme protokolleri sayesinde birbirleri ile haberleşen ve bilgi paylaşan akıllı bir ağ oluşturmaktadırlar [2].

IoT kavramının Endüstri, Sağlık, Taşımacılık, Üretim, Tarım, Perakende, Akıllı şehirler gibi birçok sektörde etkisini yakın gelecekte yaygın olarak göstermesi beklenmektedir. IoT'nin birbirine bağlı akıllı telefonlar, tabletler ve bilgisayarlardan oluşan bir ağdan çok daha fazla olduğunun altını çizmek önemlidir. Kısacası IoT, nesnelerin birbirine bağlı olduğu ve aynı zamanda İnternet'e bağlandığı bir ekosistemdir. IoT potansiyel olarak İnternet'e bağlanabilen, veri ve bilgi alışverişi yapabilen her nesneyi içermektedir. Bu nesneler her zaman, her yerde bağlıdır ve veri alışverişinde bulunurlar.

Bağlı nesneler kavramı yeni değildir ve yıllar geçtikçe geliştirilmiştir. Devrelerin küçülmesi ve daha az enerji tüketimiyle işlemcinin artan gücü, birbirleriyle konuşan milyonlarca nesnenin bulunduğu bir geleceği hayal etmemizi mümkün kılmaktadır [2]. IoT'nin temelindeki bir akıllı nesne, İnternet'e bağlanan ve veri alışverişi yapabilen bir cihazdır. Bu cihaz basınç, sıcaklık ve benzeri ölçüm



yapan basit bir algılayıcı olabilir. Bu kavram fırın, kahve makinesi ve çamaşır makinesi gibi Internet'e bağlanabilen akıllı nesnelere ile genişletilebilmektedir. Aynı zamanda otomobiller, binalar, endüstriyel cihazlar da örnek olarak verilebilmektedir.

Gömülü sistemlerde IoT'un yaygın kullanımı için özel işletim sistemleri geliştirilmektedir. Bunlardan biri de Google tarafından sunulan Android Things'dir. Bu işletim sistemi IoT ekosistemi bünyesinde mevcut Android geliştirme araçlarının, uygulama programlama arayüzlerinin (ing. Application Programming Interface) (API) ve kaynaklarının kullanılmasını sağlar. Android Things tüketici, perakende ve endüstriyel uygulamalar için geniş kullanımlı bir işletim sistemidir [3].

Bölüm 2'de IoT, Raspberry Pi ve Android Things gibi konulardaki ilgili çalışmalar özetlenmiştir. Bölüm 3'te Android Things işletim sisteminin yapısı incelenmiştir ve sistemin Raspberry Pi ortamında kullanımı anlatılmıştır. Bölüm 4'te geliştirilen IoT yazılımının tasarımı ve uygulanması yer almaktadır. Son bölümde, Android Things işletim sisteminin kullanımına yönelik değerlendirmeler ve elde edilen sonuçlar verilmiştir.

2. İlgili Çalışmalar

Raspberry Pi gibi geliştirme platformları IoT kavramında önemli bir rol oynar ve bağlı nesnelere geliştirilmesine yardımcı olmaktadır. Geliştirme kartlarını kullanarak, IoT sistemleri üzerinde uygulama geliştirme yapılabilmektedir. Raspberry Pi ile yapılmış IoT uygulamalarına literatürde oldukça sık rastlanmaktadır. Bu çalışmalar çok farklı alanlara yayılmaktadır. Sağlık uygulamalarında [4], akıllı ev çalışmalarında [5] ve diğer endüstriyel alanlarda [6] örneklere rastlamak mümkündür. Bunlarla birlikte Alanda programlanabilir kapı dizileri (ing. Field-programmable gate array) (FPGA) [7], iklim görüntüleme [8], sulama sistemleri [9], veri kaydediciler [10], yangın alarm sistemleri [11], güneş enerjisi sistemleri [12], hız sınırlandırma sistemleri [13] gibi birçok farklı çalışmada da Raspberry Pi ile yapılmış IoT uygulamaları bulunmaktadır.

Yukarıda verilmiş olan tüm bu çalışmalarda işletim sistemi olarak Linux ve türevleri kullanılmaktadır. Ancak bu çalışmalar içerisinde Android Things kullanımı kısıtlıdır [14] ve bu işletim sistemine dayalı gömülü yazılım sistemlerinin tasarımı ve uygulanmasına yönelik yöntem belli değildir. Yukarıda belirtilen bu eksiklikten yola çıkılarak bu bildiride yeni Android Things işletim sistemine dayalı gömülü sistem yazılımlarının geliştirilmesi için bir yaklaşım sunulmuştur. Bir alarm uygulaması üzerinde işletim sisteminin Raspberry Pi platformu ile gömülü sistem yazılımı geliştirilmesinde bir alt katman olarak kullanılmasına yönelik elde edilen deneyimler ve niteliksel bir değerlendirme yapılmıştır.

3. Android Things İşletim Sistemi

Android Things, Google tarafından IoT projeleri oluşturmak üzere geliştirilen bir işletim sistemidir. Bu işletim sistemi güvenilir platformların oluşturulmasına ve Android kullanarak profesyonel uygulamaların geliştirilmesine yardımcı olmaktadır. Android Things, Android'in değiştirilmiş bir sürümüdür. Bu işletim sistemi büyük bir potansiyele sahiptir, çünkü Android geliştiricileri sorunsuzca IoT'ye geçebilir ve kısa bir süre içinde yeni uygulamalar geliştirmeye başlayabilirler [3]. Android Things işletim sistemi Şekil 2'deki diyagramda gösterilen katmanlı bir yapıya sahiptir.



Şekil 2: Android Things mimarisi

Bu mimari Android işletim sisteminden biraz farklıdır, ve Android'e göre daha basit ve birleştirilmiş bir yapısı vardır. Böylece Android Things için uygulamaların altında daha az bileşen bulunmaktadır. Uygulamalar normal Android uygulamalarına göre sürücü ve çevre birimlerine daha yakındırlar. Her ne kadar Android Things, Android'ten türetilmiş olsa da, Android Things'te desteklenmeyen birçok Android API bulunmaktadır.

Bu bildiride tanıtılan çalışma sırasında Android Things'in genel yapısına ait bazı özellikler de sistem geliştirme sırasında tespit edilmiş; sistemin bazı bileşenlerinin de kullanımı deneyimlenmiştir. Android'de yaygın olarak kullanılan ancak Android Things yazılım geliştirme ortamında bulunmayan içerik sağlayıcıların mimariden çıkarıldığı görülmektedir. Ayrıca, normal bir Android uygulaması gibi, Android Things uygulaması, isteğe bağlı olsa bile bir kullanıcı arayüzüne sahip olabilmektedir. Bir kullanıcı, bir Android uygulamasında olduğu gibi olayları tetiklemek için kullanıcı arayüzüyle etkileşimde bulunabilir. Bu bakış açısıyla bir kullanıcı arayüzünün geliştirilmesi süreci Android'de kullanılanlarla aynıdır. Çalışmadan elde edilen deneyimler Android'e ait genel

bilgi sahibi olunması durumunda IoT kullanıcı arayüzlerinin Android Things üzerinde kolayca ve hızlı bir şekilde geliştirilebileceğini göstermiştir.

Android Things Google hizmetlerine oldukça yüksek seviyede uyum sağlamaktadır. Google tarafından uygulanan hemen hemen tüm bulut servisleri Android Things'te mevcuttur. Bunlarla birlikte Android Things, mobil dünyayla sıkı sıkıya bağlı olan, kullanıcı girişi veya kimlik doğrulaması gerektiren Google hizmetlerini desteklemez. Kullanıcı arayüzünün Android Things'te mecburi olmaması bu özelliklerin desteklenmemesini doğurmaktadır.

Android açısından önemli bir konu da izin yönetimidir. Bir Android uygulaması, kaynaklara sınırlı erişimi olan bir sanal alanda çalışır. Bir uygulamanın sanal alanın dışındaki belirli bir kaynağa erişmesi gerektiğinde izin istemesi gerekmektedir. Bir Android uygulamasında, bu *Manifest.xml* dosyasında yapılmaktadır. Android Things'te de aynı şekilde uygulama tarafından istenen tüm izinler yükleme zamanında verilir.

Android Things kullanıcı arayüzü bildirim durum çubuğunu desteklemez ve uygulamalardan bildirimler tetiklenmez. Kullanıcı arayüzü ile ilgili tüm hizmetlerin, görevi gerçekleştirmek için bir kullanıcı arabirimi gerektirir. Bu nedenle Android Things'te kullanıcı arayüzü ile ilgili hiçbir özelliğin çalışması garanti edilmez.

Things destek kütüphanesi [3], çevre birimleri ve sürücülerle olan iletişimi idare etmek için Google tarafından geliştirilen ve Android yazılım geliştirme platformunda bulunmayan tamamen yeni bir kütüphanedir. Duyargalara, çıkış birimlerine, harici aygıtlara bağlantı ve bağlantı değişimi için kullanılan bir dizi Java arabirimine ve sınıflara erişmek için bu kütüphane kullanılır. Ayrıca bu kütüphane, iç iletişim bilgilerini gizler ve GPIO (ing. General-Purpose Input/output), I2C (ing. Inter-Integrated Circuit), PWM (ing. Pulse-Width Modulation), SPI (ing. Serial Peripheral Interface), UART (ing. Universal asynchronous receiver-transmitter) gibi birkaç endüstri standardı olan protokolü desteklemektedir.

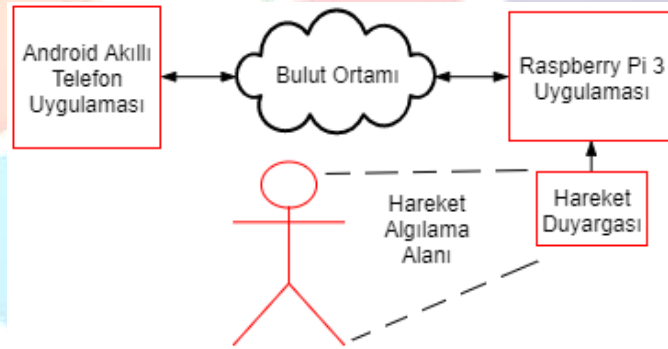
Bu çalışmada kullanılan Raspberry Pi 3 geliştirme platformu [15] Android Things ile uyumludur. Bu platform 1.2Ghz 4 çekirdekli ARMv8 İşlemci, Wi-Fi 802.11n, Bluetooth 4.0 gibi özellikleri içermektedir. Android Things işletim sisteminin Raspberry Pi platformunda kurulumu için bir Windows bilgisayar kullanılmıştır. Resmi Android Things internet sitesindeki yönergeler kullanılarak bir SD (ing. Secure Digital) kart hazırlanmıştır. Bu kartta işletim sisteminin imajı bulunmaktadır ve çalışma yapılabilecek ortam hazır hale getirilmiştir. Raspberry Pi platformu HDMI (ing. High Definition Multimedia Interface) arayüzü ile bir ekrana bağlanmıştır. Ayrıca Ethernet ile lokal ağa da bağlanmıştır. Enerji verildikten sonra, önyükleme aşamasını tamamlayana kadar beklenmiş ve açılış ekranı elde edilmiştir. Sistem, cihaza ağ üzerinden bağlanıp uygulama atılabilecek hale gelmiştir.

4. Alarm Sisteminin Geliştirilmesi

Bu çalışmada Android Things işletim sistemi kullanılarak bir alarm IoT yazılımı geliştirilmiştir. Sistem temelde bir Android Things Raspberry Pi uygulaması, bir bulut platformu ve bir Android akıllı telefon uygulamasından oluşmaktadır. Alarm sisteminin temel yapısı Şekil 3'te görülmektedir.

4.1. Android Things Raspberry Pi Uygulaması

Çalışmada izlenen Android Things işletim sisteminin geliştirme süreci ve uygulama yapısı, yaygın bir Android uygulaması ile aynıdır. Bu nedenle, Android Things için kullanılacak geliştirme aracı Android Studio [16] olarak seçilmiştir.



Şekil 3: Alarm sisteminin yapısı

Bu uygulamada, bir hareket duyargasından GPIO aracılığı ile hareket bilgisi alınmaktadır. Hareket olayı oluştuğunda Android Things uygulaması bulut ortamına ve dolayısı ile kullanıcı akıllı telefonuna bir bildirim gönderecektir.

Bilindiği üzere bir alarm sistemi, evleri ya da işyerlerini güvende tutmak için çeşitli duyargaları kullanan karmaşık bir sistemdir. Bu tür sistemlerin kalbinde, hareketi algılayan duyargalar gibi elemanlar bulunur. Bu duyargalar, bir nesnenin algılama alanında hareket edip etmediğini algılayabilir. Bu çalışmada hazırlanan sistemde de hareket duyargası algılama alanını tarar. Bir hareket algıladığında, olay Android Things uygulamasını içeren karta bildirilir. Android Things uygulaması, bildirim etkinliğini ele alır ve kullanıcının akıllı telefonuna bir mesaj göndermek için Google Firebase [17] bulut sistemi ile iletişim kurar.

Sistemde kullanılan hareket duyargası yaygın olarak kullanılan PIR (ing. Passive Infrared Sensor) duyargasıdır. Kızılötesi algılama işlemi yapmaktadır. İnsan gibi sıcak bir cisim algılama alanından geçtiğinde duyarga algılama işlemi yapmakta ve Raspberry Pi

platformunun GPIO bacağına bir sinyal göndermektedir. Platform bu sinyali algılayınca hareketli bir nesnesinin bulunduğu anlaşılmaktadır. GPIO'da bulunan sinyal düşük seviye veya yüksek seviye olabilmektedir. Hareket algılaması da yüksek seviyeden düşük seviyeye geçişte geliştirilen uygulama ile yapılmıştır. Android Things işletim sistemi, iletişim ayrıntılarını gizleyen GPIO pinleri ile etkileşime girmeye yardımcı olan *PeripheralManagerService* sınıfını geliştiricilere sağlamaktadır. Bu sınıf ile birçok işlem uygulama seviyesinde kolaylıkla yapılabilmektedir.

PeripheralManagerService sınıfı kullanılarak bir GPIO için bu çalışmadaki iklendirme işlemleri tamamlanmıştır. Aşağıda verilen ve *onCreate* methodunda bulunan satırlar ile duyarganın bağlantı GPIO ayarlamaları yapılmıştır. Burada GPIO'nun giriş yönünde çalışacağı, yüksek seviyede aktif olacağı gibi ayarlamalar yapılmaktadır. Ayrıca burada *sensorCallBack* kullanılarak sinyalin düşen kenarı algılandığında hareket işlemi olduğu anlaşılmaktadır. Bu koşulda *sensorCallBack* ile sistemde bir hareket olduğu bulut ortamına iletilir.

```
PeripheralManagerService service = new PeripheralManagerService();  
gpioPin = service.openGpio(Pins.getPirPin());  
gpioPin.setDirection(Gpio.DIRECTION_IN);  
gpioPin.setActiveType(Gpio.ACTIVE_HIGH);  
boolean status = gpioPin.getValue();  
sensorCallBack = new SensorCallBack();  
gpioPin.registerGpioCallback(sensorCallBack);  
gpioPin.setEdgeTriggerType(Gpio.EDGE_RISING);
```

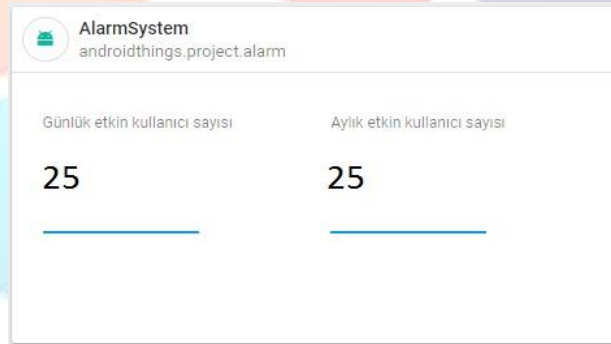
Çalışmada bulut ortamına bilgilendirmenin gönderilmesi için *GpioCallback* sınıfından genişletilen *SensorCallBack* sınıfı kullanılmıştır. Bilgilendirme için *NotificationManager* nesnesi üzerinden *sendNotificaton* metod çağırımı yapılmaktadır. Bu işlemin kod örneği aşağıda verilmiştir. *ALARM_KEY* değişkeni Google Firebase sisteminin üretmiş olduğu bir anahtardır. Böylece sisteme izinsiz girişler önlenmektedir.

```
NotificationManager.getInstance().sendNotificaton("Alarm!", ALARM_KEY);
```

4.2. Bulut Platformu ve Ayarlamaları

Çalışmada hareket algılanması işleminden sonra bildirim işleminin de yapılması gerekmektedir. Mesajlaşma sistemi olarak, Google Firebase [17] kullanılmıştır. Bu sistem, Google tarafından geliştirilen ve ücretsiz hizmet sunan bir bulut platformudur. Sistemin bildirim hizmeti bu çalışmada kullanılmıştır. Android Things uygulamalarından bir akıllı telefona bildirim göndermenin birkaç yolu vardır. Bu çalışmadaki yöntemde haberleşme kanalı gibi düşünülecek bir sistem kullanılmıştır. Akıllı telefon uygulaması sisteme abone olduktan sonra bu kanalda yayınlanan tüm mesajları alır. Çalışmada akıllı telefon uygulaması, kanaldan mesajlar alan bir abone gibi davranmaktadır. Android Things uygulaması ise iletileri yayınlayan bir yayıncı gibi davranmaktadır. Firebase bildirim sisteminin yapılandırılması da yerine getirilmiştir.

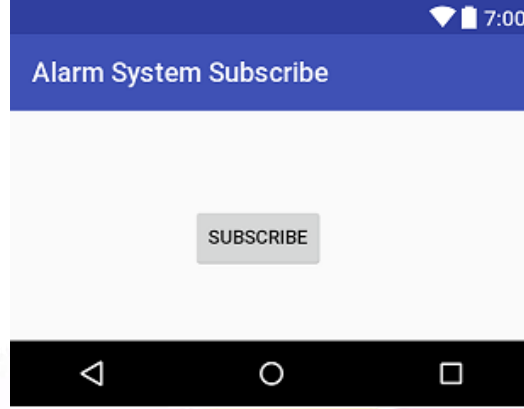
Sistemin web arayüzü üzerinden Firebase projesi yapılandırıldıktan sonra, bildirim özelliği alarm sistemine eklenmiştir. Android Things uygulamasındaki *ALARM_KEY* değişkeni bu platformda oluşturmuş ve geliştirilen sistemin yazılım kütüphanesinin içine kopyalanmıştır. Şekil 4'te Firebase projesindeki uygulamanın kullanımının kaydı görülebilmektedir.



Şekil 4: Bulut bağlantı arayüzü

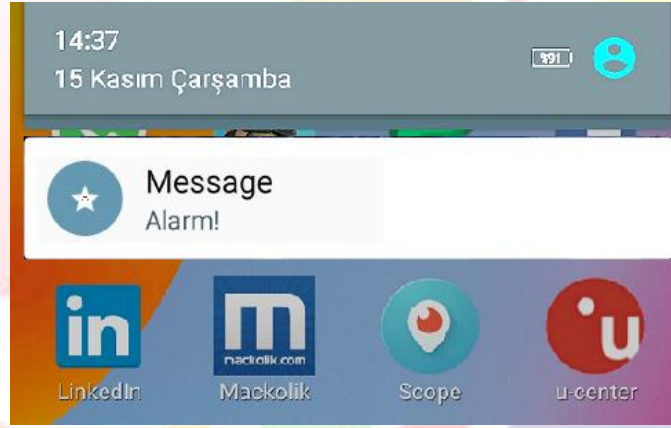
4.3. Akıllı Telefon Bilgilendirme Uygulaması

Bildirimleri almak için çalışmada aynı zamanda akıllı telefonlara bir Android uygulaması tasarlanmıştır. Sistemi basit tutmak için Android uygulaması birkaç işlem ile sınırlandırılmıştır. Bu uygulama, bildirimleri almak için öncelikle Android Things uygulaması tarafından kullanılan kanala abone olur, gelen bildirimleri dinler ve bildirimleri kullanıcıya gösterir. Şekil 5'ten görülebileceği gibi arayüz oldukça basit tutulmuştur. Arayüzden sadece kanala abone olma ve gelen bildirimleri görme işlemleri yapılmaktadır.



Şekil 5: Akıllı telefon bildirim uygulaması

Çalışmada geliştirilen Android Things uygulaması bir nesnenin hareket ettiğini algıladığında, bildirim mesajı göndererek Google Firebase platformuyla iletişime geçmektedir. Bu işlemin ardından, Firebase platformu mesajı kullanıcının akıllı telefonuna gönderir. Böylece kullanıcı sistem tarafından üretilen bildirimini telefonunda görmektedir. Şekil 6'da akıllı telefona gelen bildirim görülebilmektedir.



Şekil 6: Telefon bilgilendirme işlemi

5. Sonuç ve Değerlendirme

Bu çalışma ile Android Things ortamında yazılım geliştirmeye yönelik bir yaklaşım ortaya konmuş; örnek sistem geliştirme çalışması ile bu yeni ortamın IoT açısından yetenekleri deneyimlenmiştir.

IoT sistemleri kullanımına özel olarak geliştirilmiş olan Android Things işletim sistemi bu çalışma kapsamında incelenmiştir. Bu işletim sistemi yaygın olarak kullanılan Raspberry Pi platformuna kurulmuş ve bir IoT uygulaması geliştirilmiştir. Sistemde bir Android Things uygulaması ve Android akıllı telefon uygulaması Google Firebase bulut platformu kullanılarak haberleşmektedir.

IoT uygulamalarının geliştirilebileceği Android Things tabanlı platformların Android kullanıcıları için oldukça bilindik bir geliştirme ortamı sunduğu görülmüştür. Bu ortam kullanarak geliştirme yapılmasının hızının Android geliştirme alışkanlığı olan geliştiriciler için oldukça iyi olduğu söylenebilir. Bununla birlikte Google firmasının sağlamış olduğu geliştirme API'si IoT uygulaması geliştirmeyi hızlı ve etkin bir hale getirmektedir. Öte yandan geniş özelliklere sahip olan Android sisteminin Android Things'te basitleştirilmesinin IoT uygulamaları için yeterli olduğu ve sistemin çalışma hızına olumlu yönden etkilerinin bulunduğu gözlenmiştir. Geliştirilen IoT uygulaması klasik Android işletim sistemi içeren bir platformda da sınıanmıştır. Android Things ve klasik Android işletim sistemi çalışma hızları karşılaştırılmıştır. Bu karşılaştırmada alarm koşulunun oluşmasından itibaren alarm bilgilendirmesinin bulut ortamına yapılmasına kadar geçen zaman göz önüne alınmıştır. Klasik Android işletim sisteminin bulunduğu ortamda işlem ortalama 0.9 sn sürmekteyken, Android Things'te ortalama 0.6 sn sürmektedir. Böylelikle Android Things işletim sisteminin hafifletilmiş yapısının iyileştirmesi de basit bir uygulama ile sınıanmıştır.

Android Things işletim sistemi piyasada bulunabilen Raspberry Pi gibi birçok platform ile uyumludur. Böylece geliştirmeye başlamanın süresi oldukça kısalmaktadır. Bir IoT uygulamasının bulut ortamı kullanılarak geliştirilebilmesi de ayrıca süreci kolaylaştırmakta ve zamanı kısaltmaktadır.



6. Kaynakça

- [1] C. Hallinan, “Embedded Linux Primer: A Practical”, Real-World Approach, Prentice Hall, 2006.
- [2] The Internet of Things, “ITU Internet Reports, Executive Summary”, International Telecommunication Union, 2005.
- [3] Android Things, <https://developer.android.com/things/index.html>, son erişim: Şubat 2018.
- [4] R. Kumar, M.P. Rajasekaran, “An IoT based patient monitoring system using raspberry Pi”, R. Kumar, M.P. Rajasekaran, “An IoT based patient monitoring system using raspberry Pi”, 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), 2016, 1-4.
- [5] V. Patchava, H.B. Kandala, P.R. Babu, “A Smart Home Automation technique with Raspberry Pi using IoT”, 2016 International Conference on Smart Sensors and Systems (IC-SSS), 2015, 1 – 4.
- [6] E. Kadiyala, S. Meda, R. Basani, S. Muthulakshmi, “Global industrial process monitoring through IoT using Raspberry pi”, 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), 2017, 260 – 262.
- [7] A. Rupani, P. Whig, G. Sujediya, P. Vyas, “A robust technique for image processing based on interfacing of Raspberry-Pi and FPGA using IoT”, 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017, 350 – 353.
- [8] R. Shete, S. Agrawal, “IoT based urban climate monitoring using Raspberry Pi”, 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, 2008 – 2012.
- [9] A. Imteaj, T. Rahman, M.K. Hossain, S. Zaman, “IoT based autonomous percipient irrigation system using raspberry Pi”, 2016 19th International Conference on Computer and Information Technology, 2016, 563 – 568.
- [10] T. Tavade, P. Nasikkar, “Raspberry Pi: Data logging IOT device”, 2017 International Conference on Power and Embedded Drive Control (ICPEDC), 2017, 275 – 279.
- [11] A. Imteaj, T. Rahman, A.M.S. Hossain, S.A. Rahat, “An IoT based fire alarming and authentication system for workhouse using Raspberry Pi 3”, 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, 899 – 904.
- [12] B. Inner, “Data monitoring system for solar panels with bluetooth”, 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, 1 - 4.
- [13] S. Divarci, S. Demir, O. Urhan, “Computer vision approach based speed limit assistant”, 2016 24th Signal Processing and Communication Application Conference (SIU), 2016, 1405 – 1408.
- [14] T. Cho, H. Kim, J.H. Yi, “Security Assessment of Code Obfuscation Based on Dynamic Monitoring in Android Things”, IEEE Access, 2017, 5, 6361 – 6371.
- [15] Raspberry Pi 3 Android Things Development, <https://developer.android.com/things/hardware/raspberrypi.html>, son erişim: Mayıs 2018.
- [16] Studio and SDK Tools, <https://developer.android.com/studio/>, son erişim: Mayıs 2018.
- [17] Google Firebase Cloud Platform, <https://firebase.google.com/>, son erişim: Mayıs 2018.

NOT: Bu çalışma Ege Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi (Proje No: 17-UBE-002) tarafından desteklenmiştir.

