International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2018 | Issue 36 | Volume 10 | Number 3 | Pages 6-13

AN IOT APPLICATION WITH PARTICLE CARD OVER CLOUD

I. Tarimer [1] S. Eren [2]

1. *Information Systems Engineering, Mugla Sitki Kocman University, Mugla, Turkey, itarimer@mu.edu.tr*
2. *Information Systems, Bodrum Vocational and Technical High School, Mugla, Turkey, ereneren55@gmail.com*

Abstract- This paper deals with the technical processes to operate and to control far electrical device(s) over cloud computing. In order to do this, a hardware unit that can be connected to itself four electrical appliances/devices, and both IOS and Android applications which can control this equipment remotely via Wifi or GSM have been developed. For these purposes, the Photon and Electron development cards were used for prototyping the hardware. The iOS app to provide remotely control was designed and coded in "X Code" platform, and the Android app was designed and coded in "Android Studio" platform. The coordination between the developed hardware and the mobile device using in this work is provided for controlling appliances/things by a mobile program over cloud. Thus, the electrical devices at far point have been managed to control as real time. Moreover, the developed application (a sample of IoT) has detected both the current flow and the current failure. The developed application system informs the current circumstance to a central monitoring unit in the network by wireless. The developed system is expected to be one of the good samples that will adapt to Industry 4.0 concept practically.

Keywords: Cloud Computing, IoT, Mobile Programming, iOS, Android, Industry 4.0.

I. INTRODUCTION

Nowadays, we are witnessing that the era of Internet of Things (IoT) emerges by new applications. A British technology pioneer, Kevin Ashton, coined the phrase "the Internet of Things" first time in the world [1]. His aim was to describe the physical objects' internet that connect to the internet. Internet of Things connects every objects which are alive or not, has a noticeable economic value, and communicates/interacts with others. In the last decade, IoT has taken much attention from IT scientists and engineers. Such a revolution in internet of things or smartness have been progressed rapidly. It is now being moved towards to a fully interconnected and smart world anymore. Moreover, not only mobile platforms but also industrial fields are affected from this revolution. As the best knowledge of the authors, daily life in future would be easier and more efficient by means of IoT. In terms of Industry 4.0 concept, the usable home or office appliances needs to have several IoT abilities.

Thanks to emerging IT technologies, is starting enormousoccasionfor a lot ofnovel implementations that promise to evolve the life quality. In this context, an IoT platform should include everything which is needed to connect an electrical device to the network over internet: a device/cloud platform, a hardware,a firewall to prevent unauthorized access, and SIMs for cellular devices. The Particle development card(s) help the things to connect to internet and the cloud. Program uploads and whole data communication and firmware updates are made over the cloud of Device Cloud. Hence, remote management gives opportunity for updating, and as well as data transmission. Provided the Photon device to be managed is located in a Wi-Fi medium,and/or provided the Electron device is located in a GSM medium, the remote devices can be controlled from anywhere in the world.

In the literature, we have seen several studies related to IoT applications. In recent years, a group of scientists put forwards the design of home automation and security system using Android accessory development kit and Android mobile device [2]. Kovatsch et al. have proposed internet technology to become the future solution for home automation [3]. Das et al. have dealt with design and the implementation a home automation and security system for mobile devices [4]. Recently 3 of scientists have purposed to describe the integration of a gesture control system for industrial collaborative robot [5].

Another scientist group controlled a toy car standing in a wireless network remotely by a BlackWidow card as an iPhone application [6]. Tarımer and Dağıstanlı made a remote machine automation for a heating center by using an Android smart device within Wi-Fi or 3G medium [7]. Modh et al. controlled remotely a toy car standing in a Wi-Fi network over a web site via spark core development card [8]. Pradeep et al. introduced an advanced technique with GSM module for pumps [9]. Gutierrez et al. automated an irrigation system to optimize water utilization for agricultural plants by wireless network [10].

Tarımer and Eren controlled two submersible pumps at far distance by iOS software adopted to the hardware [11]. They also presented their second work regarding to an IoT apps in the IOTPE 2017 Conference [12]. Guller discussed how to write a data processing application using Spark [13], and also presents the current trends in 'big data' technology in his book. Arslan and Kirbas performed a microcontroller-based wireless measurement and control system as an IoT application [14].

This paper deals with an apps developed for switching all types of electrically operated device(s) remotely with a hardware unit. One ofthe software is executed by iOS based devices, the other is executed by Android based devices and controls things over cloud in Wi-Fi or GSM. Neither it needs to write SMS messages accordance to any rule, nor need the phone keys to press accordance to any rule, since the interface program is designed as user friend. Primarily the developed apps was successfully tested and operated in laboratory conditions then, was applied to connect the physical things remotely. Unauthorized access is one of the most vital problems at remote management apps. So, the access token needs to prevent automatically. The Particle development cards present firewall as default, prevent all unauthorized accesses without any additional planning required. In the following sections, the developed cloud apps and its IoT communication are discussed.

II. THE PARTICLE AND THE PARTS

To build an IoT product, it is started with getting the device. The Particle which is an IoT development prototype is built from the ground up us a fully-integrated IoT platform, all rolled up into one. The Particle has developed Photon, Electron, and Particle Mesh. In the study, Photon and Electron were used. The Photon family is a product that works on Wi-Fi network, and it is the next generation of the Spark Core [15].

Both boards look very similar to each as seen in Figure 1. The main differences are that the Photon uses a different Wi-Fi chip, and a faster processor with more RAM [16]. Figures 2 and 3 show a Photon and an Electron with some of its main features label-led [16].

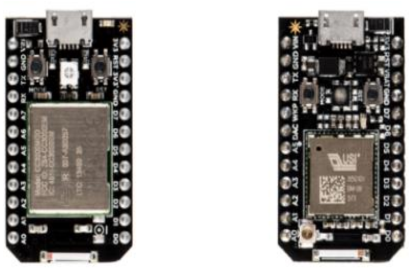


Figure 1. The Spark Core (left) and Photon (right) [15]

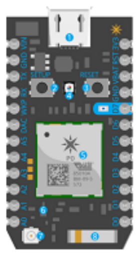


Figure 2. The Photon

Figure 3. The Electron

The Electron is a product that works on cellular networks. The Electron family pairs industry leading u-blox GSM and LTE modules with a powerful STM32 ARM Cortex M3 microcontroller.Particle Mesh creates local networks that collect data, share messages, and connect to the internet.

The Particle's software services are Device OS, Device Cloud, and Developer Tools. Device OS is low-level firmware code that supports a Particle device's basic. The Particle firmware is split into modules. Each module can be updated independently. Device OS features an easy-to-use programming framework to help you write applications that run on your devices [17]. Figure 4 shows Device OS pieces.

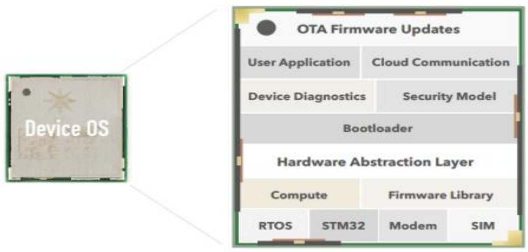


Figure 4. Device OS [18]

Device Cloud is built to work with Particle Devices and Particle Devices are pre-configured to communicate with Device Cloud. Device Cloud gives you intuitive and powerful device management features that just work. Developer tools that Particle's offers us "Device Console, IDEs, CLI, SDKs, Libraries, and RESTful API".

A. IoT Connectivity

As known, the Photon is developed to serve in Wi-Fi networks, and the Electron is developed to serve in cellular networks. When to connect the Electrondevice to Device Cloud over a cellular network, the Particle services provide connection without any problem.

By the end of 2020, there will be 21B IoT device worldwide [19]. The Particle devices are encrypted in communication as default. TheParticle uses RSA in TCP services, and uses DTLS in UDP services. Keeping a secure connection with the Device Cloud ensures that all the authority on the device is always with user.

Radio connections are encrypted by WPA2, or 3G/2G radio protocols. A secure device protocols end-to-end encryption to the cloud.Additionally the network authentication are never passed to the cloud, and are only stored locally or on a SIM card.Device Cloud uses hosting with ISO 27001, 27017 and 27018 physical security and risk management. Particle reduces its attack surface area through a variety of techniques. API-based attacks are filtered out with a scalable traffic load balancer. The API utilizes a 2048-bit TLS certificate. It also supports the OAuth 2.0 standard for secure login for integrations. It keeps the information stored at the cloud as minimum.

Particle uses a sophisticated dev-ops system of automated deployments, containers, and service checks to automatically ensure servers are up-to-date, and running only the expected applications. Particle Platform can scale machines in the face of an attack or large customer demands. Particle regularly hires professional penetration testing for any emerging threat models [19]. Figure 5 shows essential elements of Device Cloud security.

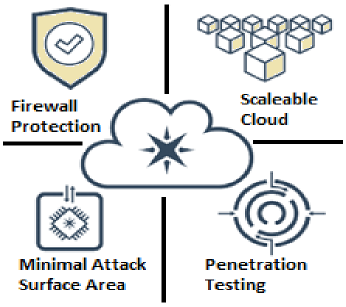


Figure 5. Essential elements of Device Cloud security [19]

Particle ensures the physical security of their servers by using hosting via established and trusted hosting companies. AWS is used when possible. Trusted cloud hosting vendors protect against physical threats and more [19].

B. Power Management

When developing an IoT product, power consumption, storage and production must be considered. Often, IoT devices are not connected to the power grid. The main components of IoT device when consuming power are radio, microprocessor, sensor and actuators. The energy budget model for most IoT devices is as following [20]:

$$\begin{aligned}&(\text{Battery capacity})+(\text{power generated})=\\&=(\text{send data})+(\text{receive data})+(\text{collect data})+\\&+(\text{control actuators})\end{aligned} \tag{1}$$

The battery is the primary power supply. Its capacity is measured in miliamp-hours (mAh). Sending or receiving data need to consume energy much. In all type of networks (Wi-Fi, cellular, bluetooth), when sending data (Tx), the energy is consumed more than receiving data (Rx). Selection of sensor and actuator affect power consumption directly as well. Therefore, one of the most important solutions is sleeping mode for reducing power consumption. The Particle firmware API has 3 sleeping modu commands: *WiFi.off()*, *Cellular.off()*, and *System.sleep()*. Another method to optimize power consumption is to cathegorize messages as in groupes.

When desiging an IoT product, an attention is paid to energy budget. Energy consumption, battery capacity, and power generation are keys components of the energy budget. It is possible to extend the life of a product by reducing energy consumption. Once the amount of energy use for an optimized product has been determined, it is then possible to select a battery for product and a power replenishment strategy.

III. MATERIAL AND METHOD

The material of this study are the electrical devices at remote, the Photon or Electron cards operating over cloud, and necessary software for this operation. The method of the research comes about need analysis, software/hardware designing, their integration and application. The research has been carried on the irrigation motors at farmlands in Muğla province over the internet.

A. Operating of the Particle's Cards Over Cloud and Idea of Internet of Things

The devices interacting each other's as wire or wireless provide to new applications within cooperation to use advanced services. Gupta and Raspaileo says that connecting each and every small device to the cloud and to access it from anywhere and anytime is called as Internet of Things (IoT) [21, 22]. The concept of Industry 4.0 was introduced first time in 2011 in Germany. According to this idea, integration of virtual and physical systems to each of is foreseen in the future and each of the things connected to internet is taken part more. This foresight forces to occur fundamental changes at both industrial field and daily life, and to initialize a new era, as the fourth industry revolution. It is expected that IoT and similar concepts would integrate to serial production machines. That means that machines would talk to each not only in production but also every part of life. Figure 6 shows the general schematics of an IoT structure.

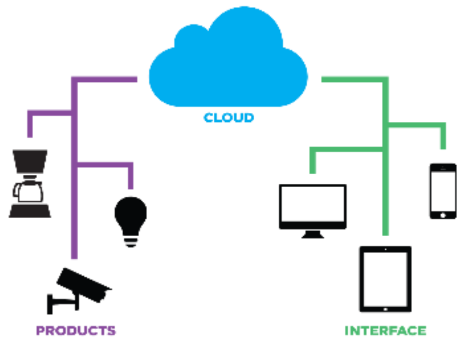


Figure 6. The general schema of IoT [23]

Gubbi, et al. expressed a vision that IoT vision is expanding on the need for convergence of wireless sensor networks (WSN), and Internet [24]. Zhu et al. presented a prototyping practice of IOT passage [25]. Tebje, et al. reported an implementation for IoT for monitoring domestic conditions [26]. Lazarescu showed functional design of a WSN to monitor IoT applications [27].

While number of the connected devices to internet were 8.7 billion in 2012, it is expected that the number reaches to 50 billion up to 2020. The scantiness of number of IP addresses has been fulfilled by IPv6. As the best estimates of the Cisco Company, knowledge traffic to be produced by 20 indoor devices in 2020 would be more than the entire internet traffics produced in 2008 [28].

The IoT should be understand not only the devices connecting to internet, but also some sensors produce knowledge with mobile phones, radio frequency identification (RFID) readers, and base stations. It is possible to gather all historical information of a product in a database from its manufacturing stage to transport, delivering and using at home thanks to its RFID label by using together with other sensors. If the objects are equipped with sensors and electronic circuitries, they begin to obtain "thinking, feeling and speaking" features. Herewith, they reach to an ability of that they will update their current information by communicating with human. The more mobile networks and internet evolve, the more these objects/things will communicate with people/firms. Naturally, persons will observe, control them in any place, in every time.

Hence, dreams of smart world would be probable [1]. Lehmann et al. showed an architecture by IoT devices in their research [29]. The Photon and the Electron cards provides good solution(s) to computing; photonhas got an internal Wi-Fi module; electron has got an internal GSM module; all of them suitable to use with Arduino; all of them an open source coded with a special cloud for developers. When to begin to use the Particle's cards (Photon, Electron), a user account was created by taking e-mail address and password, since the device needs to sign up to communicate over cloud.

Inputs of the Photon are between 3.6-5.5 VDC. If any power saving method is not selected, it takes 80 mA at 5 VDC input. Inputs of the Electron are between 3.9-12 VDC. If any power saving method is not selected, it takes 2 A at 5 VDC input. The Photon has 14 digital inputs/outputs, 6 analog input pins. The Electron has 26 digital input/output, 12 analog inputpins [16]. In this study, no power saving method is selected. In order to write the application software with Particle's cards (photon, electron), the microprocessor's program was written in the platform in which it is found in the link of https://build.particle.io and afterwards, it was uploaded to the device over cloud.Since both cards are based on Ardunio, same codes work in both same cards.

B. Transmitting and Reading Data

Huang et al. have presented a new mobile cloud framework to provide traditional computation services [30]. Communicating between Particle's cards (photon, electron) and objects needs to activate itself by supply voltages. Then, it is connected to the cloud in wi-fi medium by signing in (writing e-mail address and password). In this study, 4 devices are controlled, but the system almost controls up to 8 small devices.

Low profile wide area (LPWA) technologies are called as the systems that should operate at hard-to-reach areas and consume low power energy. In LPWA solutions, narrow-band width is preferred in terms of reducing power consumption. However, it hasn't been needed to take measures for reducing power consumption by considering to reach to the devices to be controlled is difficult (they're at far) and all these devices will operate by mains voltages. The Photon development card is available at markets with two options: internal antenna and uFL connector.

On the one hand, both user and device are not needed to become in the same network in order to connect the electrically operated device to the cloudin the system developed by using Photon. On the other hand, none network restriction is available for the system developed by using Electron.

Unauthorized remote access problem is prevented by security operations with over access token. The outputs are switched on and off by transmitting data over cloud in the study. What outputs are used actively is determined at Settings section by user. The outputs at both two cards are connected physically to the relay card.

After [7, 12], we have pursuited new works. We provided progress not only in the hardware, but also in the software. We developed both iOS and Android versions for this work. So, independency of network and platform are provided for users.

To send data to the Particle cards (photon, electron), the file in which it is in the library of *SparkSDK.h* called as *callFunction* is used in iOS app. The sequence of this functions is given as following:

MyPhotoncallFunction: FunctionName with Arguments: args completion: error{}];

The *FunctionName* parameter determines the output in which the operation would be done on it. The args parameter determines type of the operation as on/off. The statement "off" is used for energizing outputs, the statement "on" is used for de-energizing outputs regarding to the relay's usage. The necessary statement for energizing Exp1 output is as following:

*MyPhotoncallFunction: @ "Digital Write" with arguments: @ [@"D0 LOW'] completion:^(NSNumber * result code, NSError *error { }];*

The Particle cards have a lot of pins to read information digitally. Existing the electricity at the device is understand by reading data over cloud. To do this, existing of electrical current is sensed by connecting 220V relays as parallel to mains line. Thus, the particle's cards (photon, electron) produces 3.3V signals, and switches the inputs on. While the electrical current is passing, '1' value is read; otherwise, the value is '0' digitally. The existing and absent of electricity are controlled by using 2 timers to trigger another at iOS application. The *callFunction* function is used to read the information of 'current exists'. In order to learn the information of 'electricity on' at the device connected to system, the statement below must be instructed below. Here the value '1 or 0' shall be read over *resultCode* variable:

*MyphotoncallFunction: @ "Digital Read" with arguments: @[@"A0"] completion: ^(NSNumber *resultCode, NSError *error { }];*

In the Android app, operations of reading / writing from particle cards are performed with the *AsyncTask* command as a background processing. For these operations, ParticleCloud, ParticleCloudException and ParticleDevice libraries are used; the operations are done with *callFunction* command:

Device.callFunction(FunctionName,list)

The *FunctionName* parameter determines the output in which the operation would be done on it. The list parameter determines type of the operation as on/off and pine to be processed. The statement "off" is used for energizing outputs, the statement "on" is used for de-energizing outputs regarding to the relay's usage. The necessary statement for energizing Exp1 output is as:

List<String> list = new ArrayList<>();
list.add("D0");
list.add("LOW");
Device.callFunction("digitalwrite", list);

There are a lot of pins to read digital information on the Particle cards (Photon, electron). Existing the electricity at the device is understand by reading data over cloud. To do this, existing of electrical current is sensed by connecting 220 V relays as parallel to mains line. Thus, the particle's cards (photon, electron) produces 3.3 Vsignals, and switches the inputs on. While the electrical current is passing, '1' value is read; otherwise, the value is '0' digitally. The current flow and deficiency of electricity are controlled by 2 timers at Android app. The *callFunction* function is used to read information of 'current exists'. In order to learn the information of 'electricity on' at the device, the statement below must be instructed below. The value '1 or 0' shall be read over *result* variable:

List<String> list = new ArrayList<>();
list.add("A0");
List.add("LOW");
Device.callFunction("digitalwrite", list);

IV. DEVELOPMENT OF THE ELECTRONIC CIRCUIT AND THE SOFTWARE

In the study, a Wi-Fi based or GSM based electronic card (Photon or Electron) and a software which is adopted to the card to control remotely are developed for both iOS and Android operation systems. The hardwares developed are 8 optional outputs, but they are designed as 4 outputs. None of power consumption methods are selected in both two of cards. Because, the system operates depending on the mains voltage. In case of power failure, system operation can last for a certain time with Li-Po battery. In apps, the access is provided via Wi-Fi with photon and via GSM with Electron.

The integrated solution developed in the study consists of three parts: the hardware units connected to the Photon or Electron development cards, the program for Photon or Electron microprocessor, and the application softwares for an iOS or Android based mobile devices. The outputs of the circuit are activated via the data came over the cloud. The developed software for the iOS based device was written in Objective-C language, for the Android based device was written in Java language. When the softwaresis coded, it is uploaded to the computer. Then, entire circuit is prepared to mount with the hardware parts. The mounted circuit is provided in Figure 7.

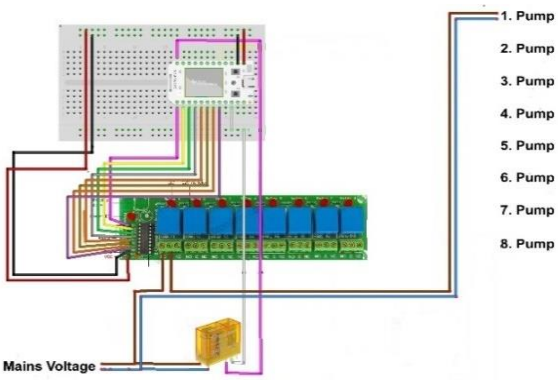


Figure 7. Physically layout of the circuit

The circuit shown in Figure 7 has a Photon or Electron, 8×5 V relays card, and 1×220 V relays. The outputs of Photon or Electron from D_0-D_7 are connected to the relay driver, and the system operated with 220 V are switched. When any of the outputs is activated, 3.3 V Photon or Electron signal is switched on by 220 V relay, and then is connected to the inputs of Photon or Electron from A_0. It is checked-up whether electricity is on or off. An USB connection, a lipo cell or an external charge adapters are provided for supplying Photon or Electron by suitable voltages. It is not suggested to supply Electron with USB connection.

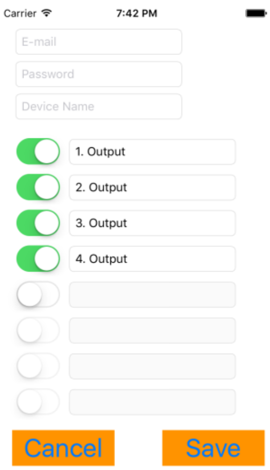
The microprocessor used in the cards was programmed in Arduino programming language by using Arduino IDE. It is depended upon processing programming language. To upload program or to revise the codes is not needed to have physical connection, since the microprocessor operates over cloud. Two parts of the developed software for inputs/outputs are givenas pseudo codes below routines:

```
inttinkerDigitalRead String pin);
inttinkerDigitalWrite String command);
void setup(){
pinMode(0,OUTPUT);
pinMode(1,OUTPUT);
digitalWrite(0,HIGH);
Spark.function("digital read", tinkerDigital Read);
Spark.function("digitalwrite", tinkerDigitalWrite) ;}
void loop() {  }
inttinkerDigital Read (String pin){
int  pin Number = pin.charAt(1)-'0';
ifpinNumber< 0 || pinNumber>7) return -1;
if(pin.startsWith ("D"))
{  returndigitalReadpinNumber) ;    }
else ifpin.startsWith "A"))
{pinMode(pinNumber+10, INPUT_PULLDOWN);
returndigitalRead pinNumber+10)  }
return -2;}
inttinkerDigitalWrite (String command){
bool value=0;
int  pin Number = command.charAt(1)-'0';
ifpinNumber< 0 || pinNumber>7) return -1;
if (command.substring(3,7)== "HIGH") value=1
else if (command.substring(3,7)== "LOW") value=0)
else return -2;
```

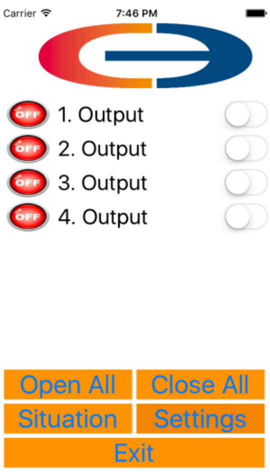
if(command.startsWith ("D"))
{ pinMode(pinNumber, OUTPUT) ;
digitalWrite(pinNumber, value);
return 1;}
elsereturn -3;}

The written codes above are uploaded to the microprocessor. The opening position is set by *setup()* and *loop()* functions. The user should identify outputs, inputs, and the values; to identify *setup()* and *loop()* functions are optional. The *setup()* function primarily defines input pins by *pinMode* command; afterwards, the *digitalWrite* command sends '1' signal to the outputs, and the relay outputs are switched off. The *setup()* function makes iOS or Android application to start sending instructions. Defining the codes in Loop function is not needed, since the application reads value(s), and repeats this by a timer in iOS or Android device side.

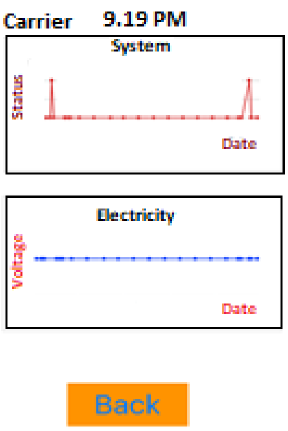
In iOS app, to implement Particle's functions, the SDK library must be configured. Then, the View *Controller.h* file is informed by *#import<Spark-SDK.h>* statement. Therefore, the program could use functions of *SparkSDK* library. All the codes were written in Objective-C by using Xcode IDE. The setup, interface and situation screens are given in Figure 8.



(a) Setup screen



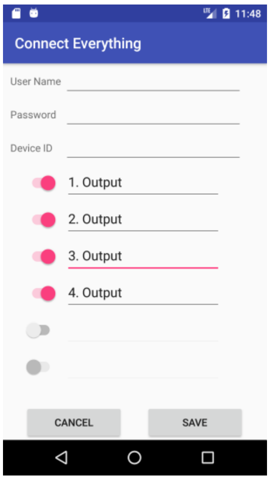
(b) Application screen



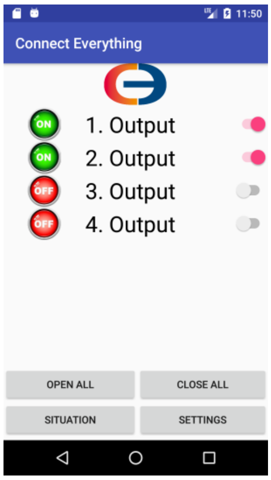
(c) Situation screen

Figure 8. Screen views of the iOS app

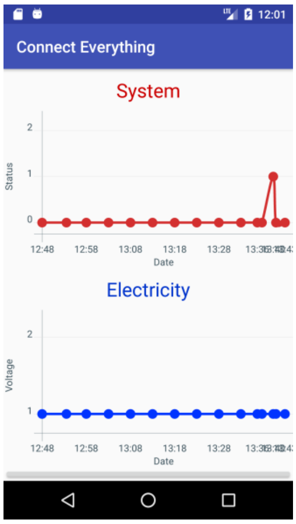
In Android app, to implement the Particle's functions, the SDK library must be configured. All the codes were written in Java by using Android Studio IDE. The setup, interface and situation screens are given in Figure 9.



(a) Setup screen



(b) Application screen



(c) Situation screen

Figure 9. Screen views of the Android app

In order to switch on/off each of devices in both two apps, the switch must be slipped into the proper position. All the outputs can be switched on or off in one time by using the buttons 'switch on all and switch off all'. To connect user to updated by the *Settings* button. This button also declares the number of outputs and their names to see at the screen. When the application is operated, firstly internet access for the device is controlled, then it is connected to Device Cloud by signing in procedure.

As shown from Figure 8c and 9c , electricity flow in the remotely controlled devices can be pursued in several time intervals. This process is done via connection of the webhook establishing with ThingSpeak.com Moreover, presence and interruption of the electric current drew by the remote controlled devices are being demonstrated on the interface software.

The developed hardware system has been mounted as 4-outputs, and located in a closed box. The whole physical arrangement (wire diagram) is seen in Figure 10. Figure 11 shows wiring diagram of the remote controlled 0.75 kW 1 phase motor without submersible pumps for testing.

Figure 10. The mounted hardware structure

Figure 11. View of the remotely controlled motor

By means of this study, the motor and submersible pumps were settled to make agricultural irrigation in a rural field.The developed mobile application was usedat an agricultural field to irrigate cropsin Milascityof Muğla province on June 2018. Farmers and peasants of the selected region informed that they were pleased to make easy irrigation and to harvest productive crops by using this apps.

V. CONCLUSIONS AND SUGGESTIONS

The paper dealt with the procedures needed when to design an IoT device and operating, controlling of electrical devices at distance point by iOS or Androidbased tools.The most important feature of the developed system than the others is to have user friendly interface. The user can control the devices at distance point without writing an SMS according to the rule or clicking on button(s). On the other hand, all the limitations and environmental restrictions are exceeded by the internal GSM connection which presenting by the Electron.

The two parts of the system, the developed hardware, and the mobile device could access to internet in Wi-Fi or GSM. The developed system makes life easier by controlling any electrical device and objects remotely. Because of that a low-cost working possibility has been presented at daily life applications. It has seen that Particle's development cards are one of the most practical solutions to create IoT application prototypes. The graphical interface of all these processes are presented to the user(s) on the site of https://thingspeak.com/.

In further step, we consider to add logging records to the system, and to increase the user/system ability. The developed software and hardware in the study can control every remote device(s) that integrated to the development card, and ready to take on/off instruction(s).

REFERENCES

[1] K. Ashton, "That 'Internet of Things' Thing", RFiD Journal, June 2009.

[2] D. Javale, M. Mohsin, S. Nandanwar, M. Shingate, "Home Automation and Security System Using Android", Int. Journal of Electronics Communication Computer Tech., Vol. 3, No. 2, pp. 382-385, March 2013.

[3] M. Kovatsch, M. Weiss, D. Guinard, "Embedding Internet Technology for Home Automation", IEEE Emerging Technologies and Factory Automation, pp. 1-8, September 2010.

[4] S.R. Das, S. Chita, N. Peterson, B.A. Shirazi, M. Bhadkamkar, "Home Automation and Security for Mobile Devices", 1st IEEE PerCom Workshop on Pervasive Communities and Service Clouds (PerCoSC 2011), pp. 141-148, Seattle, USA, 21 March 2011.

[5] G. Tang, S. Asif, P. Webb, "The Integration of Contactless Static Pose Recognition and Dynamic Hand Motion Tracking Control System for Industrial Human and Robot Collaboration", Industrial Robot Journal, 2015.

[6] C. Fernandes, K.Y. Ng, B.H. Khoo, "Development of A Convenient Wireless Control of an Autonomous Vehicle Using Apple iOS SDK", IEEE Region 10th Conf., pp. 141-148, Bali, Indonesia, 21-24 November 2011.

[7] I. Tarımer, E. Dagistanli, "An Application of a Remote Machine Automation via an Android Smart Device", Journal of Sigma, Vol. 32, pp. 1-9, 2014.

[8] A. Modh, M. Dabhi, L.N. Mishra, V.N. Mishra, "Wireless Network Controlled Robot Using a Website, Android Application or Simple Hand Gestures", Journal of Computer Networks, Vol. 3, pp. 1-5, 2015.

[9] E. Pradeep, R. Ganeshmurthy, K. Sekar, E. Arun, "Automation of PV Farmers Pump", 2nd Int. Conf. on Sustainable Energy and Intelligent System (SEISCON 2011), pp.163-166, Chennai, India, 20-21 July 2011.

[10] J. Gutierrez, J.F. Medina, A. Garibay, M. Gandara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module", IEEE Trans. on Instrumentation and Measurement, Vol. 63, No. 1, pp. 166-176, 2014.

[11] I. Tarimer, S. Eren, "Developing a Mobile Software Based iOS for Remote Control of Submersible Pump", 1st International Conference on Engineering Technology and Applied Sciences (ICETAS 2016), Afyonkarahisar, Turkey, 20-21 April 2016.

[12] I. Tarimer, S. Eren, "Operating of a Wireless Mobile Programming Application Over Cloud", The 13th International Conference on Technical and Physical Problems of Electrical Engineering (ICTPE 2017), Van, Turkey, 21-23 September 2017.
[13] M. Guller, "Big Data Analytics with Spark: Writing a Spark Application and Big Data Technology Landscape", Chapter: 1-2, pp. 1-15 and 71-78.
[14] K. Arslan, I. Kirbas, "Developing Wireless Sensor/Actuator Node Prototype for Internet of Things Applications", The Journal of Graduate School of Natural and Applied Sciences, Mehmet Akif Ersoy University, Special Issue 1, pp. 35-43, Burdur, Turkey, 2016.
[15] S. Monk, "Getting Started with the Photon", Maker Media, San Francisco, USA, 2015.
[16] https://docs.particle.io/datasheets, 27.08.2018.
[17] https://www.particle.io, 27.08.2018.
[18] Z. Crockett, "How Device OS Powers Your IoT Product", January 2017.
[19] Particle Editors, "Security Checklist for the Internet of Things", January 2017.
[20] Particle Editors, "Power Management for IoT Devices".
[21] V.S. Gupta, P. Raspaile, "Low Cost Standard Internet of Things", International Journal of Engineering Science & Advanced Technology, Vol. 5, Iss. 2, pp. 78-80, 2015.
[22] J.M. Ferris, "Red Hat, Inc. Self-management of Virtual Machines in Cloud-Based Networks", Patent No: US 8935692 B2, Jan.13, 2015.
[23] http://www.gnexlab.com/2014/10/02/internet-of-things-iot/, 27.08.2018.
[24] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", Future Generation Computer Systems, Vol. 29, pp. 1645-1660, 2013.
[25] Q. Zhu, R. Wang, Q. Chen, Y. Liu, W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things", IEEE/IFIP 8th International Conference on Embedded and Ubiquitous, pp. 347-352, Hong Kong, China, 11-13 December 2010.
[26] S.D. Tebje Kelly, N.K. Suryadevara, S.C. Mukhopadhyay, "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes", IEEE Sensors Journal, Vol. 13, No. 10, pp. 3846-3853, October 2013.
[27] M.T. Lazarescu, "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Vol. 3, No. 1, pp. 45-54. March 2013.
[28] D. Evans, "The Internet of Things How the Next Evolution of the Internet Is Changing Everything", Cisco Internet Business Solutions Group (IBSG), April 2011.
[29] M. Lehmann, A. BiornHansen, G. Ghinea, T.M. Gronli, M. Younas, "Data Analysis as a Service: An Infrastructure for Storing and Analyzing the Internet of Things", Vol. 9228, Lecture Notes in Computer Science, pp. 161-169, 08 August 2015.
[30] D. Huang, M.Kang, J. Luo, "MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication", 5th IEEE International Symposium on Service Oriented System Engineering (SOSE 2010), Nanjing, China, 4-5 June 2010.

BIOGRAPHIES

Ilhan Tarimer was born in Manisa, Turkey, 1963. He received the B.Sc., M.Sc. and Ph.D. degrees from Gazi University, Ankara, Turkey, in 1984, 1991 and 2003, respectively. Currently, he is a Full Professor at Information Systems Engineering in Mugla Sitki Kocman University, Mugla, Turkey. He teaches microcomputer systems designing, software engineering, and mobile programming. His research interests are in the area of information and communication, internet of things, remote control of the machines and industry 4.0.

Sinan Eren was born in Samsun, Turkey, 1979. He received the B.Sc. degree from Suleyman Demirel University, Isparta, Turkey, and the M.Sc. degree from Mugla University, Turkey, in 2002 and 2016, respectively. He works as an Information Technologies Teacher in Bodrum Vocational and Technical High School, Mugla, Turkey. He is interested in the areas of information, remote control of the things, software engineering, and industry 4.0.