An IoT LDR Bulb Application with Android Things Operating System for Smart Cities

Sadik Arslan
R&D Department
Kent Kart Ege Elektronik A.Ş.
İzmir, Turkey
sadik.arslan@kentkart.com.tr

Orhan Dagdeviren
International Computer Institute
Ege University
İzmir, Turkey
orhan.dagdeviren@ege.edu.tr

Geylani Kardas
International Computer Institute
Ege University
İzmir, Turkey
geylani.kardas@ege.edu.tr

***Abstract*—Nowadays, Internet of Things (IoT) is a highly active research area with wide range of industrial applications such as smart city services. These applications generally aim to meet the growing demands of society in terms of power, water, gas, and lighting services. Although there are studies related to this area, technological developments in terms of both hardware and software lead to better solutions. In this study, an IoT application which is a key feature for a smart city service has been developed for street lighting. The developed application works with a light dependent resistor (LDR) and it controls a bulb according to the ambient light intensity. The control information is kept and analyzed in a cloud environment. Raspberry Pi development environment and Android Things operating system are used for realization of the system. Light bulb control and LDR hardware circuits have been developed within the scope of this study. With this study, a platform has been developed which can be realized at an affordable price to add additional sensors such as air pollution and humidity. Additionally, the easy development possibilities of the Android Things operating system have been examined.**

Keywords—Internet of Things, Android Things, Raspberry Pi, Smart City, Smart Lightning

I. INTRODUCTION

The Internet of Things (IoT), frequently heard today, is a worldwide network of uniquely addressable objects (nodes) that are generally embedded in the environment [1]. These systems are generally designed for special purposes to perform predefined tasks. An IoT node is mainly composed of a processor, a memory, communication units, clock, secondary storage units and power supply as given in Fig. 1. The nodes residing in IoT network communicate with each other through specific protocols. IoT concept can simply be explained as a system of devices that communicate with each other through a variety of Internet communication protocols, and create an intelligent network by sharing information [2].

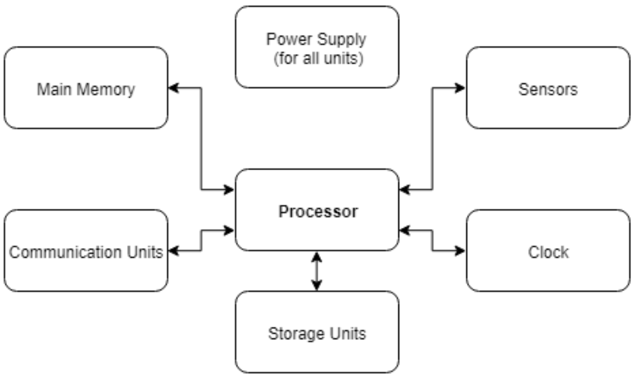

Fig. 1. Simplified IoT node block diagram.

With smart technologies, it is possible to implement different applications to solve the problems of the society in the cities to managing the resources better. In the vision of "smart city", IoT has a very important and strategic position that can be described as a prerequisite. Applications that will improve the city life are realized by analyzing the data coming from the connections and sensors provided by IoT. These applications provide wide range of services such as power, water, gas and lighting. Although there are efficient applications in this area, technological advancements in hardware and software bring new opportunities to enhance the capabilities of current services.

In this paper, we give the design and development of an IoT based smart street lighting application. Our proposed system works with a light dependent resistor (LDR) to control the bulb according to ambient light intensity. We deploy a cloud infrastructure to store and analyze the control information generated by the system. Light bulb control and LDR hardware circuits are designed and developed. We use Raspberry Pi and its development environment with Android Things operating system. This operating system is implemented by Google to use the existing widely used Android development tools, application programming interfaces (APIs) and resources [3].

The rest of the paper is organized as follows. In Section II of the paper, related work is given. In Section III, an overview of IoT concept is discussed. In Section IV, Android Things operating system is examined. In Section V, the designed system is described in detail as hardware, software and cloud platforms. In the last section, conclusions are drawn.

II. RELATED WORK

Smart cities provide a wide range of applications to society through IoT and many technologies. Smart parking, structural health monitoring, traffic congestion management, waste management, fire detection, air pollution detection and smart lighting are significant examples of these applications. With the help of smart lighting, energy efficiency as well as automated control will be provided.

IoT based smart lighting is an attractive research area [4 - 7]. The study given in [4] is a solar supplied lighting system for smart city control cards. In [5], Raspberry Pi development environment is used for control of the IoT system hardware to provide smart lighting. A supply system design for smart cities is given in [6]. Energy efficiency is very important in IoT applications [7] especially for smart home related services [8 - 10]. In study [8], IoT system can be controlled by an Android smart phone application. Green house environment is provided in [9]. In [10], sensors nodes are used for IoT system development. There also exist IoT studies using LDRs in many fields such as farming [11], environmental monitoring [12], solar tracking [13], office [14] and renewable energy [15]. In all of these studies, operating systems are used as Linux and its derivatives and

978-1-7281-2868-9/19/$31.00 ©2019 IEEE

they all use LDR and cloud. In some studies, ready tools such as Raspberry Pi are used. However, none of these studies have used Android Things operating system. Moreover, this study brings smart city, LDR and IoT environments together with a compact and efficient design from hardware to cloud environment.

III. IoT Overview

Today, IoT is one of the most promising trends in technology. The IoT concept is expected to show its impact in many sectors such as Industry, Health, Transport, Production, Agriculture, Retail, Intelligent cities etc. in the future.

In IoT, there are some definitions that refer to different aspects and according to different application areas. It is important to note that IoT is much more than a network of interconnected smartphones, tablets and computers. In short, IoT is an ecosystem where objects are connected to each other and connected to the Internet at the same time. IoT potentially includes any “thing” that can connect to the Internet and exchange data and information. These things are nearly always connected and ready to exchange data. The concept of connected things is not new and has been developed over years. The downsizing of circuits, less energy consumption and the increasing power of the CPU make it possible to imagine a future where there are millions of things talking to each other [16].

There are several elements contributing to the creation of the IoT ecosystem. It is important to understand the role that these elements play to obtain a clear picture of IoT. A smart thing that forms the basis of IoT is a device that connects to the Internet and can exchange data. This device can be a simple sensor that measures pressure, temperature etc. This concept can be expanded with intelligent objects that can be connected to the Internet, such as ovens, coffee machines, washing machines, cars, buildings, and industrial devices.

At a low level, IoT devices exchange data using a network layer. The most important and known protocols in the concept of IoT are Wi-Fi, Bluetooth, Zigbee, Cellular Network, NB-IoT, LoRA. On the application side, there are widely used protocols. Some of them can be listed as HTTP, MQTT, CoAP, AMQP and XMPP [16].

Development platforms play an important role in the IoT concept and help in the development of connected things. In this study, we will focus on how to create an IoT system with Raspberry Pi, a development platform compatible with Android Things. In the next section, the basic features of the Android Things operating system will be discussed.

IV. Android Things Operating System

Android Things is an operating system developed by Google and it enables developers to create professional IoT applications using trusted platforms and Android. Android Things is a modified version of the well-known Android. This operating system has great potential, because Android developers can easily switch to IoT and start developing new projects in a short time. Before going into Android Things, it is necessary to look at the general structure of the system. The Android Things operating system has the layered structure shown in the diagram given in Figure 2.

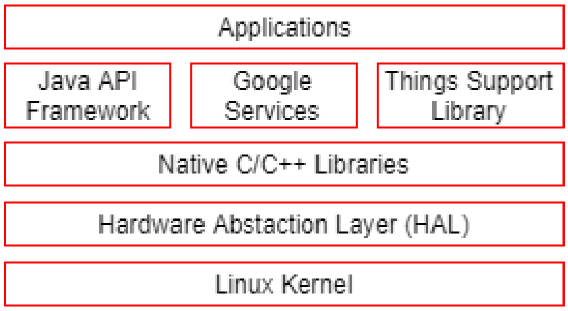

Fig. 2. Android Things architecture.

This structure is slightly different from the Android operating system because it has a simpler and unified structure. So there are fewer layers under the apps for Android Things. Applications are closer to drivers and peripherals than normal Android applications. Even if Android Things is derived from Android, there are many APIs that are not supported in Android Things.

Some content providers that are commonly used on Android have been removed from the software development environment. These are listed on the official website of Android Things [3]. Additionally, like a normal Android application, the Android Things applications can have a user interface even if it is optional. A user can interact with the user interface to trigger events as in an Android application. From this perspective, the process of developing a user interface is the same as Android. The IoT user interface can be developed easily and quickly.

Android Things is highly compatible with Google services. Almost all cloud services implemented by Google are available in Android Things. In addition, Android Things does not support Google services that require user input or authentication, which is closely linked to the mobile world.

An important issue for Android is permission management. An Android application runs in a virtual area with limited access to resources. An application must ask for permission when it needs to access a specific resource outside the virtual area. In an Android application, this is done in the *Manifest.xml* file. This is also true for Android Things. All permissions requested by the application are given at the time of installation. Android 6 (API level 23) also offers a new way to ask for permission. An application may request permission at run time, not just the installation time (using the *Manifest.xml* file). Android Things does not support this new feature, all permissions in the *Manifest.xml* file must be requested [3].

The Things support library is a completely new library developed by Google to handle communication with peripherals and drivers. This library is one of the most important features of Android Things operating system. This library is used to connect to the Java interface and classes that are used to connect sensors, output units, and external devices. In addition, this library hides internal contact information and supports several industry standard protocols such as GPIO (General-Purpose Input/output), I2C (Inter-Integrated Circuit), PWM (Pulse-Width Modulation), SPI (Serial Peripheral Interface) and UART (Universal asynchronous receiver-transmitter). In next section, we will provide the design of the proposed system.

V. LDR Bulb System Development

The Raspberry Pi 3 development platform that is supported by Android Things is used in this study. This platform, which includes features such as 1.2 GHz 4-core ARMv8 Processor, Wi-Fi 802.11n, Bluetooth 4.0, can be seen in Figure 3.

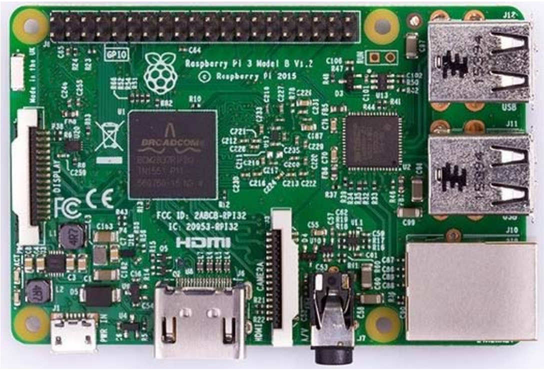

Fig. 3. Raspberry Pi 3 development board.

A Windows computer was used for the installation of the Android Things operating system for Raspberry Pi platform. An SD (Secure Digital) card was prepared using the instructions on the official Android Thing website. This card contains the image of the operating system and the working environment has been made available [17].

In this study, a LDR sensor bulb control system has been developed in order to implement an IoT application using the Android Thing operating system. The system basically consists of LDR sensor input and bulb power control circuit, Android Things Raspberry Pi application and a cloud platform. The basic structure of the system can be seen in Figure 4.

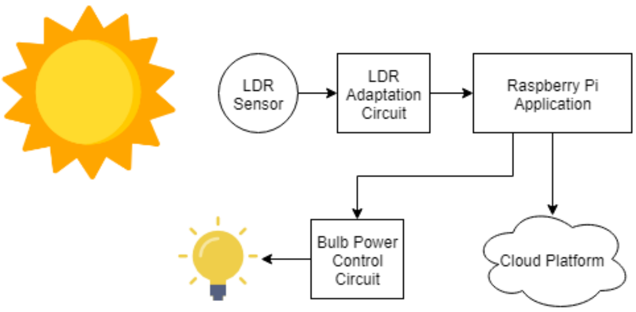

Fig. 4. The basic structure of the system.

A. LDR Sensor Input and Bulb Power Control Circuits

An LDR sensor is used to detect the light intensity in the system. This sensor is a passive element whose resistance changes according to the intensity of the light. It has an operating principle which is inversely proportional to the light intensity. In other words, as the light intensity increases, the resistance value decreases and as the light intensity decreases, the resistance value increases. LDRs act as a switching function by changing their resistance values. In this study, a circuit has been installed for this switching process. The output of this circuit is connected to a GPIO of the Raspberry Pi board.

When the low light level information from the GPIO of the Raspberry Pi is detected, the lamp is energized by the bulb control circuit. Thus, illumination of the environment is provided. If the ambient light level goes above a threshold, the sensor sends reverse information. This time it is switched off by the bulb power control circuit. The power control circuit is established via a relay. Relays are electromagnetic circuit elements that run when current flows through. Relays are used to switch a high-power receiver with a small valuable current. Figure 5 shows an LDR on the left and a relay on the right used in this system and Figure 6 shows final test hardware of the system.

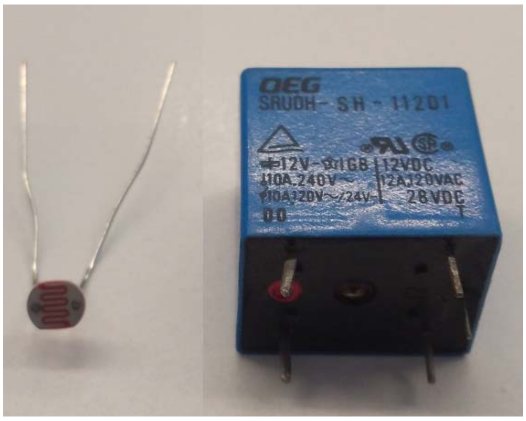

Fig. 5. LDR and relay components.

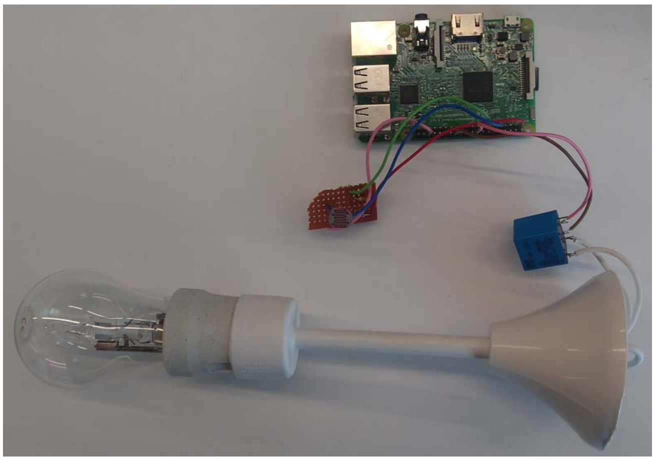

Fig. 6. Final hardware of the system.

B. Android Things Raspberry Pi Application

Android Things development process and application structure is similar to those of an Android application. Because of this reason, the development tool to be used for Android Things has been selected as Android Studio. This development environment has been established and the system has been implementation.

In this application, information is received from the sensor via GPIO port. The Android Things application sends a notification to the cloud environment when light intensity level change occurs. The principles of this application are similar to street and house lighting systems that operate bulbs according to the light intensity. However, in this study, the system was established with Android Things, a completely new operating system. A cloud mechanism has also been established. With this study, an IoT application development was made in the Android Things environment.

A bulb control system is very important in terms of energy efficiency in lighting. At the center of such systems there are sensors that detect light intensity. In this system, the LDR light sensor is a passive element that continuously measures the light level. When the light level exceeds a threshold value, the event is detected by Raspberry Pi, which includes the implementation of the Android Things. The Android Things application controls the lamp via the bulb control circuit. The application also informs the Ubidots Cloud system [18].

The light intensity sensor used in the system is a commonly used LDR sensor. The BCM27 GPIO pin of the Raspberry Pi is an input port for the sensor. The signal in the GPIO can be low or high. Light perception is also made with the application developed in the transition from high to low level. However, the BCM16 pin is used to turn the bulb power on and off. This pin controls the bulb power at high or low level. Android Things operating system provides the *PeripheralManagerService* class to developers. This class hides the contact details and allows interaction with GPIO pins. With this class, many operations can be done easily at application level. Initialization procedures are performed for GPIOs using the *PeripheralManagerService* class. With the code which can be seen below in the *onCreate* method, GPIO adjustments have been made.

```
PeripheralManagerService manager = new PeripheralManagerService();
BulpIO = manager.openGpio("BCM16");
BulpIO.setDirection(Gpio.DIRECTION_OUT_INITIALLY_LOW);
BulpIO.setValue(false);
Button LdrInput = new Button("BCM27", Button.LogicState.PRESSED_
WHEN_LOW);
```

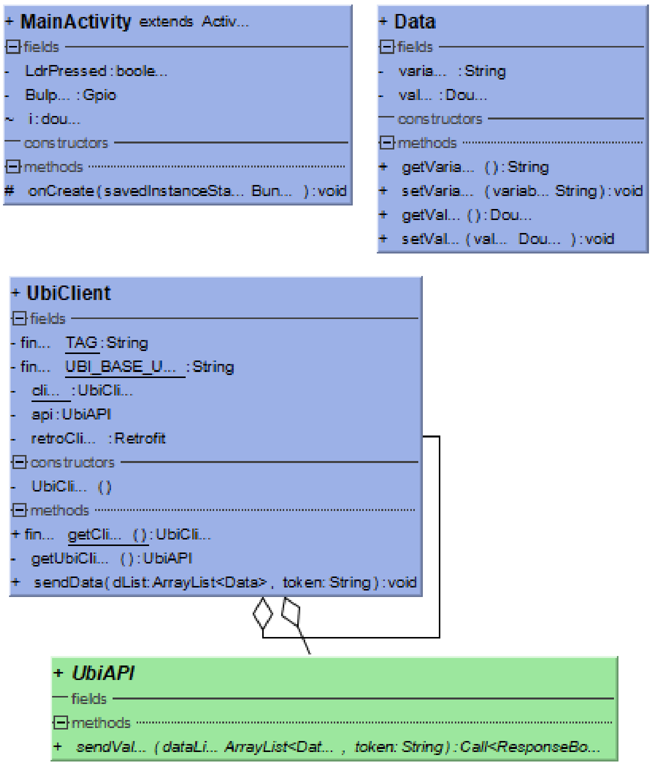

Fig. 7. UML class diagram of the application.

The object created from the *UbiClient* class is used to send messages to the cloud environment. The object generated from the *Data* class is used for the data that sent. The *sendData* method is used for the notification process. The code example of this operation is given below. *VARIABLE_KEY* and *TOKEN_KEY* variables are keys that generated by the Ubidots IoT cloud system. These keys prevent unauthorized access to the system. The UML class diagram of the developed system can be seen in Figure 7. The *MainActivity* class is where basic operations are performed on Android. The *UbiClient* class uses the *UbiAPI* class.

```
final Data dPress = new Data();
dPress.setVariable("VARIABLE_KEY");      // Variable id
dPress.setValue(i++);
UbiClient.getClient().sendData(new ArrayList<Data>() {{add(dPress);}},
"TOKEN_KEY");
```

C. Cloud Platform Setup and Usage

In this study, the cloud notification process must be realized after the detection of the light intensity change and the operation of the lamp. As a messaging system, Ubidots IoT Cloud platform was used. This system is a cloud platform that provides services specific to the IoT domain. The dashboard panel can be customized and has useful tools.

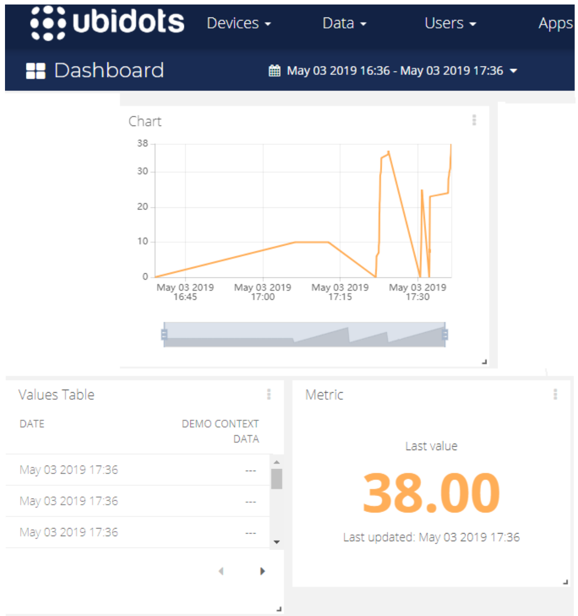

Fig. 8. Cloud dashboard interface.

First, the LDR Bulb system is configured via the web page. Then, the keys to be used in the application were taken from Ubidots. Figure 8 shows an example of the dashboard panel of the Ubidots IoT Cloud environment. On the main screen, received data graph, timeline and number of data received can be seen. The received data graph chart shows incoming data values in a timeline graph. The timeline chart shows data incoming detailed timestamps. The number of data received chart shows numbers of the discrete data arrivals.

VI. Conclusion

In this study, a low-cost system design has been made for light control systems, which are frequently used in Smart City and Smart Home IoT domains. Android Things operating system, which is developed specifically for the use of IoT systems, is used. This operating system is installed on the widely used Raspberry Pi platform. The system uses Ubidots IoT cloud platform, which can be easily implemented and provides specific services to IoT domain. Within the scope of this study, sensor and actuator circuits are established, an Android Things application is developed and a cloud service is created.

The system established in this study can be adapted very quickly to networked environments in the field. In environments where there is no network infrastructure, an additional GSM module can be used to access the Internet easily. A full multi-sensor IoT environment is planned to be designed in the future with additional sensors specific to Smart City and Smart Home domains.

ACKNOWLEDGMENT

We would like to thank Kent Kart Ege Elektronik A.Ş. company for all equipment and laboratory support.

REFERENCES

[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, Official Web Site, https://developer.android.com/things/index.html, last access: June 2019.

[4] M. Muhamad, M. I. Md. Ali, "IoT Based Solar Smart LED Street Lighting System," 2018 IEEE Region 10 Conference, October 2018.

[5] M. A. Pradhan, S. Patankar, A. Shinde, V. Shivarkar, P. Phadatare, "IoT for smart city: Improvising smart environment," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 2003-2006, 2017.

[6] P. B. Sowmiya, B. K. Nagaswetha, D. Priyadharshini, "Design of Automatic Nutrition Supply System Using IoT Technique in Modern Cities," 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC), pp. 109-111, 2017.

[7] S. S. Badgelwar, H. M. Pande, "Survey on energy efficient smart street light system," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 866-869, 2017.

[8] P. S. N. Reddy, K. T. K. Reddy, P. A. K. Reddy, G. N. K. Ramaiah, S. N. Kishor, "An IoT based home automation using android application," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 285-290, 2016.

[9] D. Shinde, N. Siddiqui, "IOT Based Environment change Monitoring & Controlling in Greenhouse using WSN," 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), pp. 1-5, 2018.

[10] H. Singh, V. Pallagani, V. Khandelwal, U. Venkanna, "IoT based smart home automation system using sensor node," 2018 4th International Conference on Recent Advances in Information Technology (RAIT), pp. 1-5, 2018.

[11] S. Rajendrakumar, V. K. Parvati, B. D. Parameshachari, K. M. S. Soyjaudah, R. Banu, "An intelligent report generator for efficient farming," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 1-5, 2017.

[12] S. C. Padwal, M. Kumar, P. Balaramudu, C. K. Jha, "Analysis of environment changes using WSN for IOT applications," 2017 2nd International Conference for Convergence in Technology (I2CT), pp. 27-32, 2017.

[13] A. Sawant, D. Bondre, A. Joshi, P. Tambavekar, A. Deshmukh, "Design and Analysis of Automated Dual Axis Solar Tracker Based on Light Sensors," 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on, pp. 454-459, 2018.

[14] S. Coelho, R. Rozario, R. Sharma, M. Mehra, "An IOT Based Smart Cubicle System for Effective Power Usage and Employee Monitoring in Offices," 2018 International Conference on Smart City and Emerging Technology (ICSCET), pp. 1-6, 2018.

[15] C. Nayanatara, S. Divya, E. K. Mahalakshmi, "Micro-Grid Management Strategy with the Integration of Renewable Energy Using IoT," 2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), pp. 160-165, 2018

[16] A. Rayes, S. Salam, Internet of Things From Hype to Reality: The Road to Digitization, Springer Switzerland, 2019.

[17] Raspberry Pi 3 Android Things Development Platform, https://developer.android.com/things/hardware/raspberrypi.html, last access: June 2019.

[18] Ubidots: IoT platform, https://ubidots.com/, last access: June 2019.