Broadcasting the Status of Plant Growth Chamber using Bluetooth Low Energy

Moh Noor Al-Azam[1*], *Mochamad Mizanul* Achlaq[1], *Aryo* Nugroho[1], *Adri* Gabriel Sooai[2], *Aris* Winaya[3], and *Maftuchah*[3]

[1]Faculty of Computer Science, Narotama University, Jl. Arif Rahman Hakim, Surabaya, 60111, Indonesia
[2]Faculty of Informatics Engineering, Widya Mandira Catholic University, Jl. Jend. Achmad Yani, Kupang, 85226, Indonesia
[3]Faculty of Agriculture and Animal Science, University Muhammadiyah of Malang, Jl. Raya Tlogomas 246, Malang, 65144, Indonesia

Abstract. Mobile users are getting smarter in using their phones. Many tasks are usually completed or monitored via a computer screen, nowadays can be taken anywhere with Android phones or tablets. Likewise with features in the phone is increasingly sophisticated. Currently bluetooth version 4 is almost mandatory in all phones. Even for entry level phones are now equipped with bluetooth version 4. This paper discusses the use of Bluetooth Low Energy—which is part of bluetooth version 4, in providing information about the status of plant growth chamber conditions. By using this concept, all phones or tablets that have bluetooth version 4 and there are applications in it will be able to receive the latest chamber status. The results show that this way of broadcasting is very effective. The data required to be monitored by a laboratory technician can be continuously broadcasted so that anyone on duty will get instant information at his grasp.

Key words: Bluetooth Low Energy (BLE), data transmission, growth chamber, Internet of Things (IoT), temperature and relative humidity

1 Introduction

Bluetooth Low Energy (BLE) is part of bluetooth version 4, with lower power consumption than previous bluetooth versions or classic bluetooth. BLE still utilizes the 2.4 GHz frequency which is also used by Classic Bluetooth. However, BLE does not have communications compatibility with the classic bluetooth. In contrast to the classical

* Corresponding author: noor.azam@narotama.ac.id

bluetooth that has the rule that it can connect up to seven devices, in BLE it has no limit on the number of devices that can be connected simultaneously. Therefore BLE is very suitable for use in broadcasting an information. Especially in the last few years, a device that supports BLE is more and more [1].

What also makes BLE interesting to learn is that this technology is the right technology, and by way of the right compromise and made at the right time [2]. For a relatively new standard (originally designed by Nokia as Wibree before being adopted by Bluetooth Special Interest Group—Bluetooth SIG, in 2010), BLE has been seen to have a very fast adoption rate, and with the number of "BLE enable" products ranked leading edge than any other wireless technology at the same time point in their release cycle [1, 2].

One of the famous BLE classes is iBeacon technology-introduced by Apple Inc. in 2013, which is used to broadcast unique information to nearby BLE devices [3]. Compared to traditional bluetooth technology, iBeacon signals with BLE are meant to have the same coverage area but lower power consumption. Most smartphones, like the latest iPhone, Android and Blackberry, are compatible with BLE technology that indicates they can perform collaborative operations with iBeacon [4].

BLE implementation is widely used in indoor Location Based Services (LBS) applications. The workings of this application is to calculate the Received Signal Strength Indication (RSSI) received by the BLE observer compared to the signal strength during transmission performed by BLE broadcaster [5]. The result of this RSSI measurement shows the relative distance of BLE observer to BLE broadcaster based on the calculation of RSSI in equation 1.

$$\text{RSSI (dBm)} = -10n \log_{10}(d) + A \tag{1}$$

In the implementation of BLE for indoor LBS, BLE broadcaster will continuously send the major, minor and Universal Unique Identifier (UUID) data as the identification of the broadcast location. The same way will be done in this study, only the data packet transmitted is containing data about the number of laboratory, reference chamber and the status of the plant growth chamber that will be translated by BLE observer and displayed in a visual form the report to determine the condition of the micro climate Plant Growth Chamber.

2 BLE and the hardware

BLE is designed to perform small and sparse data transfers at a simple data rate with very little energy. This design is different from the classic bluetooth design in general. Bluetooth SIG has therefore introduced two trademarks: Bluetooth Smart for single-mode devices (devices that only support BLE) and Bluetooth Smart Ready for dual-mode devices (devices that support Bluetooth Classic and BLE) to distinguish them. BLE has begun to enter in various types of accessories that collaborate with mobile devices such as mobile phones, tablets and notebook computers. Fancy devices such as smartwatches and other wearing devices are already equipped with Bluetooth 4.0 or later chipsets also have BLE functionality.

2.1 BLE stack

BLE stack has two layers that are complementary to provide maximum performance (Figure 1). We call it the lower layer and uppler layer. These two layers are separated by a control function that can be ruled from the upper layer to do something on the lower layer. Lower layer on BLE stack consists of Physical Layer (PHY), Link Layer (LL), and Direct Test Mode (DTM) for the test interface. PHY is in charge of transmitting and receiving data packets. LL is responsible for providing media access, connection establishment, error control, and data flow control. DTM is only used if required a test [6].

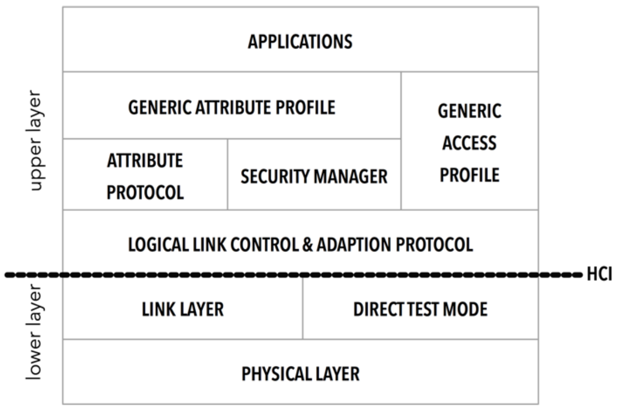


Fig. 1. BLE protocol stack [6].

The upper layer consists of Logical Link Control and Adaptation Protocol (L2CAP), Attribute Protocol (ATT), Security Manager (SM), Generic Attribute Profile (GATT), and Generic Access Profile (GAP). A Host Controller Interface (HCI) separates lower layers-often implemented for BLE controllers, while from the upper layer, often implemented as a host stack [6]. GATT and BLE profiles together make it possible for apps in standard ways without using IP Addess. While L2CAP provides multiplexing capabilities with multiplexed data channels from the above layers. L2CAP also provides fragmentation and reassembly for large data packets. Security Manager defines protocols and mechanisms for installation, distribution of keys, and security equipment boxes for BLE devices [1–4, 6].

2.2 The processing unit and sensors

A key element in a Wireless Sensor Network (WSN) is a sensor node consisting of four basic elements: the sensor unit, the processing unit, the communication unit and the power supply [7]. In this study, the processing unit uses a Raspberry Pi computer, which is reasonably priced, flexible, adaptable and programmed quite easily. One of the elements that determine Raspberry Pi's choice is Raspberry Pi version 3 which has been embedded with Bluetooth 4.0—which became the main objective in this research. In some studies, Raspberry Pi is a cheap computer whose use is very successful in the domain of sensor networks and various research applications.

Raspberry Pi is a single-board computer developed in the UK by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and developing countries [8]. Several generations of Raspberry Pi have been released. The first generation (Raspberry Pi 1) was released in February 2012 with the base model A and with higher specifications on model B. Model A+ and B+ released a year later. Raspberry Pi 2 Model B was released in February 2015 and Raspberry Pi 3 model B in February 2016.

In this study, the processing unit used a Raspberry Pi 3 model B (figure 2) equipped with Broadcom system-on-chip (SoC) system, which includes a CPU—compatible ARM unit and a VideoCore IV graphics processing unit (GPU) chip. The 1.2 GHz CPU speed has 1 GB of RAM. As storage media used SD-Card in MicroSDHC. This type of Raspberry Pi is also equipped with four USB slots, HDMI output, and 3.5 mm jack for audio. As for the purposes of digital communication provided 40 pin General-Purpse Input/Output (GPIO) that supports common protocols such as I2C and 1 Ethernet port, Wi-Fi 802.11n and Bluetooth 4.0 [9].

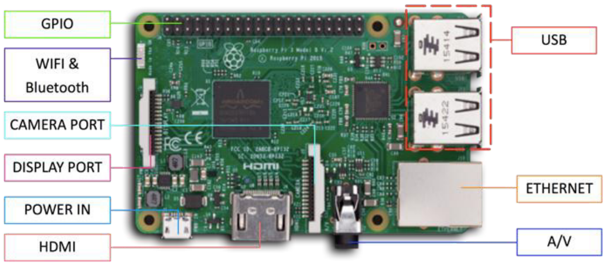


Fig. 1. Raspberry Pi & its's ports [5].

Meanwhile, as the main sensor to obtain the data of temperature and humidity chamber, this study uses production BME280 Bosch Sensortec, which can provide three types of information as well, such as temperature, humidity and air pressure (Figure 3). This sensor already has the ability of Inter-integrated Circuit (I2C) protocol, so in communication with Raspberry just use four wires only.

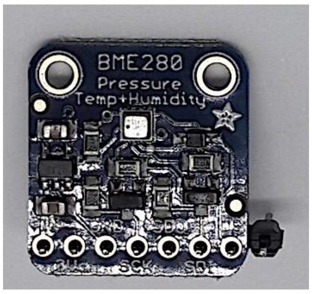


Fig. 2. BME280 presure, temperature & humidity sensor

The BME280 sensor is a temperature sensor that has a very rapid response to temperature changes, especially for environmental temperature measurements that have no physical contact on the sensor. In addition, because the packaging that already supports communication through the I2C protocol, thus making cabling between the sensor and processing unit becomes more simple [10].

3 Programming

The programming in this study is divided into two major sections. The first is the programming on the BLE broadcaster side or on the Raspberry Pi side. In this section the main program task is to read the data from the temperature and relative humidity sensor, then format the data in the 128-bit UUID form, and lastly broadcast the UUID in BLE (including major and minor data as laboratory and chamber numbers). Meanwhile the air pressure data is not included as it is not the part of the Plant Growth Chamber yet.

The second programming is on the mobile device, which in this study used an Android phone. This programming is to receive all BLE broadcast signals around the device, and sort them out specifically for the required data. The data is then interpreted into an information that will be displayed visually on the Android screen.

3.1 Broadcaster side: Operating system & programming language

The broadcaster processing unit using a Raspberry Pi 3 that is already equipped with a bluetooth 4.0. This single-board computer has several options operating system that can be used, namely Noobs, Raspbian, Ubuntu, Windows 10 IoT Core, OSMC, LibreElec and several other Linux variants. This study uses the Raspbian operating system for several reasons. The main reason for choosing Raspbian is because Raspbian is a derivative of Debian GNU/Linux ported for Raspberry, so when dependencies are needed it will be easier to get it. At least through the existing source code and then compiled in the Raspbian system. For the purposes of programming language, there are also several options. Starting from the C++ language to a lot of choice scripting language based can be done in Raspbian. In this study, will be used Python language which is the official programming language in Raspberry [8, 11].

3.2 BlueZ library

In Raspbian distribution standard, bluetooth access must go through low level language. This method of course will be difficult in doing programming and looking for errors—when something goes wrong. Therefore, to access BLE on Raspberry Pi, in this study selected using the BlueZ library [12].

To simplify the application of this research, this BlueZ library, will be executed in python application by using subprocess. This is not the best way to access bluetooth via python, but it can still be done primarily in research. BlueZ Library provides support for the bluetooth core layer and protocol. The library is also flexible, efficient and uses a modular implementation. BlueZ also has many interesting features, such as:

i). Complete modular implementation
ii). Symmetric multi processing safe
iii). Multithreaded data processing
iv). Support for multiple Bluetooth devices
v). Real hardware abstraction
vi). Standard socket interface to all layers
vii). Device and service level security support

BlueZ provides downloadable source code and must be compiled in the desired environment—in this case Raspbian. To do the BlueZ installation can be done easily, simply follow the following commands:

```
i). mkdir bluez
ii). cd bluez
iii). wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.48.tar.gz
iv). tar xvzf bluez-5.48.tar.gz
v). cd bluez-5.48
vi). ./configure --disable-systemd
vii). make all
viii). sudo make install
ix). sudo reboot
```

3.3 Data format

Broadcasting chamber status data via BLE in this study, will be done through UUID-like when iBeacon broadcasts (Figure 4). This UUID contains four data fields, that is:

i). proximityUUID: a data of 32 digits hexadecimal which will be used as a place of broadcasting chamber status data.

ii). major: data with a width of four hexadecimal digits to be used as a laboratory number in which the chamber resides.

iii). minor: data with a width of four hexadecimal digits to be used as a chamber number that broadcasts.

iv). power: a data with four hexadecimal width that we do not need for temporarily, and we will fill it with 0 × 0000 values.

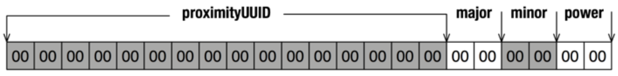


Fig. 3. BLE UUID format.

In the proximityUUID data, the first eight digits of hexadecimal will be used as the value of the measured air temperature and relative humidity. The first two digits are the air temperature and followed by the next two digits as the numbers behind the comma in hexadecimal format. So the data 20.95 ºC will be written as 0 × 14 for the first byte and 0 × 5F for second. Similarly for write the relative humidity, in proximityUUID is also done in the same way on the third up to the forth byte. The full picture format for proximityUUID can be seen in Figure 5.

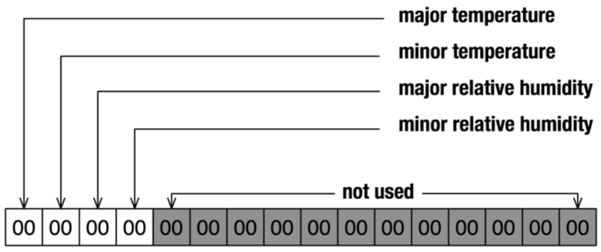


Fig. 4. The data in proximityUUID format.

Meanwhile, for major and minor BLE information, each of which has a width of four hexadecimal digits, we will use the laboratory code number and the chamber number code that does the advertising UUID. With four digits hexadecimal this will be accommodated as many as 65 656 labs and each laboratory has 65 536 pieces of chamber. The amount is very sufficient for now. For example, suppose in the measurement of chamber number 10 in laboratory no. 3 get a temperature of 21.36 °C and 78.95 % relative humidity, then complete the UUID will be like in Figure 6.

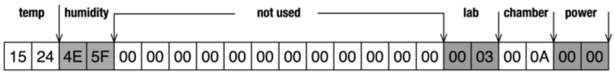

temp
humidity
not used
lab
chamber
power
15 24 4E 5F 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 0A 00 00

Fig. 5. Complete UUID to sent via BLE.

3.4 Flowchart

Like most applications, the Raspberry Pi application in this study will first initialize the necessary devices, the BME280 sensor and the Low Energy Bluetooth module. Flowchart application used in Raspberry Pi is like in Figure 7.

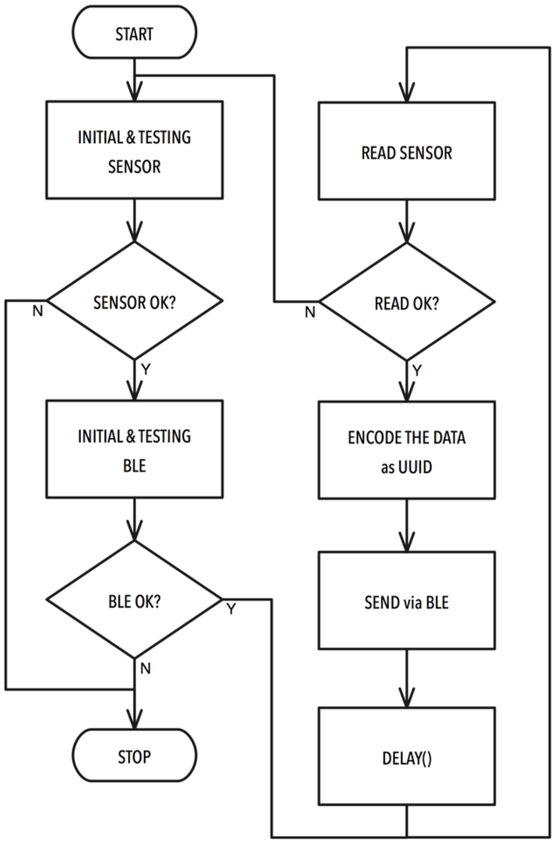


Fig. 6. Flowchart BLE broadcaster.

When executed, the first time the application will initialize and test on the sensor device (BME280 and I2C). If in this check is encountered an error, then the application will stop. If successful then it will proceed with sensor initialization and proceed with check on BLE module. After all initialization and checking are done, the application will read the data present on the sensor. The data obtained from this sensor module is numerical so it needs to be converted into hexadecimal data and placed on the first digit up to the eighth digit of proxymityUUID (the remaining digit proximityUUID will be assigned a value of 0 × 00).

The proximityUUID data is then added with four digits of laboratory number data as major and four digit BLE broadcasting as minor (laboratory code number and chamber code number filled in application configuration), and ending with 0 × 0000 value as calibrated power data. Once all data is ready, the data will be broadcasted via BLE with BlueZ library as a command to HCI with OGF = 0 × 08 and OCF = 0 × 0008.

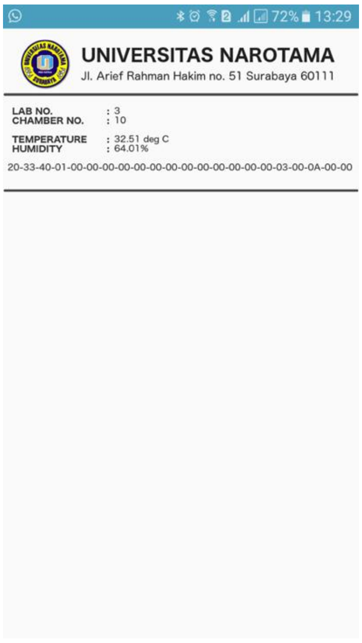
This broadcaster application is an infinite loop application. Therefore the application will stop briefly according to the time specified in the configuration and will start again reading the sensor, formating it as UUID, send it to BLE and so on. The application will only stop when there is a sensor read error or force-down from the system.

3.4 Observer side

Capturing the BLE signal on the mobile device can be done quite easily. All programming applications on mobile operating system environments are equipped with modules or plugins to access BLE and perform the iBeacon data scanning. In this study we use an Android with hybrid programming using Cordova framework [13, 14].

Cordova is a development framework with open source licenses. This framework that developed by Apache Software Foundation enables the use of standard web technologies—using HTML, CSS and JavaScript, for cross-platform application development. Applications will be run in the environments targeted to specific platforms, and rely on standard APIs suitable for accessing the capabilities of individual devices such as bluetooth or BLE modules [15, 16].

By using hybrid programming, this research requires the help of plugin for applications running on the top layer (javascript) in accessing hardware (in this case BLE module) to retrieve its data. The plugin used for this purpose is cordova-plugin-ibeacon [17] which according to information on its website can run in Android and iOS environments. UUID results obtained from the plugins in applications, then we translated in decimal form and display on screen as in Figure 8 and Figure 9.

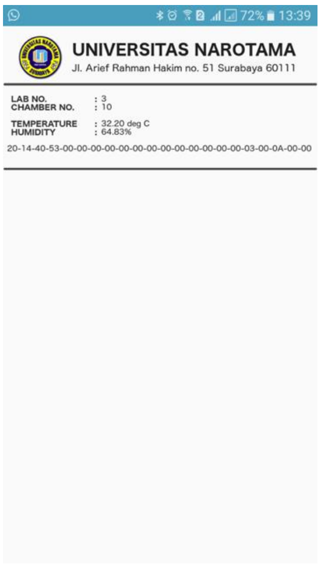


LAB NO.	: 3
CHAMBER NO.	: 10
TEMPERATURE	: 32.51 deg C
HUMIDITY	: 64.01%

```
20-33-40-01-00-00-00-00-00-00-00-00-00-00-00-03-00-0A-00-00
```

Fig. 7. Screenshot of BLE scanner application #1.

In this study, we attach broadcaster BLE delay in the amount of 10 min. So the next 10 min, the data on the sensor will be broadcasted via BLE broadcaster and received BLE observer as in Figure 9.


72% 13:39

UNIVERSITAS NAROTAMA
Jl. Arief Rahman Hakim no. 51 Surabaya 60111

LAB NO.	: 3
CHAMBER NO.	: 10
TEMPERATURE	: 32.20 deg C
HUMIDITY	: 64.83%

```
20-14-40-53-00-00-00-00-00-00-00-00-00-00-00-00-00-03-00-0A-00-00
```

Fig. 8. Screenshot of BLE scanner application #2.

4 Conclusion and future works

By using BLE as a medium of data transmission from chamber (BLE broadcaster) to mobile device (BLE observer), we can make the data available in chamber can be sent to all mobile devices equipped with the appropriate application. Broadcast delivery mode can be read by mobile devices instantly-no waiting time. So that the changing conditions in the chamber will be directly informed to the technician on duty.

In addition, by way of broadcast also reduces power consumption of mobile devices because it does not require power to make the request so it suitable for use in moving indoors. In addition to some of the results observed above, the way we do in this study also has some disadvantages. Especially on the system developed at this time and we plan to have improvements in the mining of the system to come.
The first is the weakness of the effectiveness of UUID data format usage, that is:

i). The hexadecimal format refers directly to the numeric format of the numbers in front and behind the comma. This makes numerical handling ineffective because it can not generate code for negative numbers.
ii). The same format can make the maximum data becomes very large in a case. For example, the maximum relative humidity to 255 %, which in real conditions is not possible.
iii). Major and minor uses as laboratory code numbers and chamber number codes also tend to be overkill. There can be almost no 65 536 labs or chambers in the same location.

The next weakness is the security factor of transmitted data. Currently it is not considered if this data needs encoding, given the data that is sent is general data about the condition of a chamber. But this encoding should be thought of for the future. In addition to the BLE observer side, data received can not be checked data integrity. So that all incoming data is considered as valid data and translated directly into visual data. Integrity data checking is also required especially if it will be developed commercially. Another thing to do is to bring light intensity data, both in Red-Green-Blue and light intensity in white format. For this need will be discussed with the related science field for real needs in the field.

This research is part of Automatic Plant Acclimatization Chamber research funded by DIPA Ministry of Research, Technology and Higher Education, no. SP DIPA-042.06.1.401516/2017. Authors express appreciation and gratitude to all those who have assisted in this research activity.

References

[1] R. Tabata, A. Hayashi, S. Tokunaga, S. Saiki, M. Nakamura, S. Matsumoto. *Implementation and evaluation of BLE proximity detection mechanism for pass-by framework*. International Conference on Computer and Information Science (ICIS), 26–29 June 2016 (Okayama, Japan, 2016). IEEE, pp. 1–6 (2016). http://ieeexplore.ieee.org/document/7550872/

[2] T. Nilsson, C. Hogsden, C. Perera, S. Aghaee, D.J. Scruton, A. Lund, et al. ACM Trans. Multimed. Comput. Commun. Appl., **12**(4):1–23 (2016). https://dl.acm.org/citation.cfm?id=2962720

[3] S.M.H. Sharhan, S. Zickau. *Indoor mapping for location-based policy tooling using bluetooth low energy beacons*. Wireless and Mobile Computing, Networking and Communications (WiMob) Proceedings, 19–21 October 2015 (Abu Dhabi, United Arab Emirates, 2015). IEEE, pp. 28–36 (2015). http://ieeexplore.ieee.org/document/7347937/

[4] Y. Yang, Zhouchi Li, K. Pahlavan. *Using iBeacon for intelligent in-room presence detection*. Cognitive Methods in Situation Awareness and Decision Support (CogSIMA) Proceedings, 21–25 March 2016 (San Diego, CA, USA, 2016). IEEE, pp. 187–191 (2016). http://ieeexplore.ieee.org/document/7497808/

[5] M.N. Al-Azam, B. Anindito. *BLE observer device menggunakan raspberry Pi 3 untuk menentukan lokasi Ble broadcaster*. [BLE observer device using raspberry Pi 3 to determine the location of Ble broadcaster]. Seminar Nasional Teknologi Dan Informatika (SNATIF), Universitas Muria Kudus (Kudus, Jawa Tengah, Indonesia, 2016). pp. 181–188 (2016). [in Bahasa Indonesia]. https://jurnal.umk.ac.id/index.php/SNA/article/view/646/658

[6] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, C. Gomez. *IPv6 over BLUETOOTH(R) low energy*. Internet Engineering Task Force (IETF), RFC7668 (2015). https://tools.ietf.org/html/rfc7668

[7] V. Vujovic, M. Maksimovic. *Raspberry Pi as a wireless sensor node: Performances and constraints*. Information and Communication Technology, Electronics and Microelectronics (MIPRO) Proceedings, 26–30 May, 2014 (Opatija, Croatia, 2014). IEEE, pp. 1013–1018 (2014). http://ieeexplore.ieee.org/document/6859717/

[8] E. Upton, G. Halfacree. *Raspberry Pi user guide*. New Jersey: Willey Publishing (2014). pp. 4. https://dl.acm.org/citation.cfm?id=2721682

[9] D. Rachman, M.N. Al-Azam, B. Anindito. *Sistem pemantau & pengendalian rumah cerdas menggunakan infrastuktur internet messaging*. [Intelligent Home Monitoring and Control System Using Internet Messaging Infrastructure]. Jurnal Ilmiah Link, **26**(1):1–6 (2017). [in Bahasa Indonesia]. http://link.narotama.ac.id/files/1-SISTEM%20PEMANTAU%20&%20PENGENDALIAN%20RUMAH%20CERDAS

%20MENGGUNAKAN%20INFRASTUKTUR%20INTERNET%20MESSAGING.pdf

[10] D. Japhet, K. Ndai. Int. J. Sci. Eng. Res., **7**(10):1043–1052 (2016). https://www.ijser.org/researchpaper/COMPARISONS-OF-THERMAL-RESPONSES-OF-DIFFERENT-TEMPERATURE-SENSORS.pdf

[11] S. Monk. *Programming the Raspberry Pi: getting started with Python*. New York: McGraw Hill Education, (2016). pp. 24. https://www.amazon.com/Programming-Raspberry-Pi-Getting-Started/dp/0071807837

[12] J. Hendberg. *Release of BlueZ 5.48.* BlueZ, Official Linux Bluetooth Protocol Stack [online] from http://www.bluez.org/release-of-bluez-5-48/ (2017). [Accessed on 5 January 2018].

[13] F. E. Karuna, M.N. Al-Azam. *Pengembangan prototipe quick response code (QR Code) sebagai autentikasi keamanan login sistem dengan memanfaatkan teknologi android*. [Development of Prototype Quick Response Code (Qr Code) As Login Authentication System Security Utilizing Android Technology]. Surabaya: Sistem Komputer Universitas Narotama, (2016). [in Bahasa Indonesia]. http://sistemkomputer.narotama.ac.id/wp-content/uploads/2016/04/PENGEMBANGAN-PROTOTIPE-QUICK-RESPONSE-CODE-QR-CODE-SEBAGAI-AUTENTIKASI-KEAMANAN-LOGIN-SISTEM-DENGAN-MEMANFAATKAN-TEKNOLOGI-ANDROID.pdf

[14] C. Darujati, M. Hariadi. *Facial motion capture with 3D active appearance models*. The 3rd International Conference Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 7–8 November 2013 (Bandung, Indonesia, 2013). IEEE, pp. 59–64 (2014). http://ieeexplore.ieee.org/document/6698465/

[15] J. Stark, B. Jepson, B. MacDonald. *Building Android apps with HTML, CSS, and JavaScript: making native apps with standards-based web tools*. Beijing: O'Reilly Media (2010). pp. 110. https://www.amazon.com/Building-Android-Apps-HTML-JavaScript/dp/1449316417

[16] J.M. Wargo. *PhoneGap essentials: Building cross-platform mobile apps*. Boston: Addison-Wesley (2012). pp. 45. https://www.amazon.com/PhoneGap-Essentials-Building-Cross-platform-Version/dp/0321814290

[17] Cordova. *Cordova plugin for ibeacon*. Mobile apps with HTML, CSS & JS [online] from https://www.npmjs.com/package/cordova-plugin-ibeacon. [Accessed on 7 December 2017]