Information Engineering Express
International Institute of Applied Informatics
2017, Vol.3, No.4, P.33 - 42

Experimental Evaluation of BLE-based Proximity Detection for Pass-by Applications

Ryoma Tabata [*] , Arisa Hayashi [*] , Seiki Tokunaga [*] ,
Sachio Saiki [*] , Shinsuke Matsumoto [†] , Masahide Nakamura [*]

Abstract

A pass-by application which utilizes proximity detection has a problem that it is always a burden on its developers. This is because the proximity detection performance largely depends on the settings of parameters. In order to implement the pass-by system which satisfies the needs of the users, the application developer must set appropriate parameters. If each developer researches appropriate parameters, their burden of development will increase. In this study, we evaluate effects of proximity detection performance caused by differences in methods of implementing pass-by sonar. Therefore, we develop the pass-by application using Bluetooth Low Energy as a first effort. Then, we do evaluation experiments in order to confirm the dependence of proximity detection performance with the parameters.

Keywords: BLE, Android, Pass-by Framework, Proximity Detection Mechanism

1 Introduction

In recent years, mobile devices such as smartphones or tablets have been rapidly becoming popular all over the world. On such devices, people can use applications for various purposes such as improving QoL or enjoying entertainment. Some of the applications provide services which detect the proximity of other devices by using sensors such as Bluetooth, Wi-Fi or GPS. In this paper, an event in which the objects get close within a fixed distance and within a certain period of time is defined as *pass-by*. Pass-by can be detected by proximity detection mechanism. Furthermore, we define a system which utilizes pass-by as *pass-by system*. As an example of the pass-by system, "TohakuNavi"[1] which is provided at Tokyo National Museum can automatically display an explanation about nearby exhibitions. As another example, "Streetpass Communication"[2], which is a function of Nintendo 3DS, can convert facts of pass-by in the actual world into values in a virtual world of games. Further creations of new value by pass-by systems is expected.

However, the pass-by system can be a big burden on the developers. This is because the proximity detection performance depends on a format of pass-by data and management of

[*] Kobe University, Hyogo, Japan
[†] Osaka University, Osaka, Japan

pass-by data. In addition, the pass-by systems using different communication technology are incompatible with each other. Hence, the reusability and the interoperability of the implementation is low.

In order to solve these problems, we have proposed *pass-by framework*[3] which aims to support application developers. The pass-by framework separates the proximity detection which depends on the device, and the management of pass-by data which is used by applications. Then, this framework can decrease the costs of development by abstracting the management of pass-by data. Moreover, this framework can handle pass-by data without dependence on proximity detection technology. Thus, the framework can make pass-by data between the objects, which use different communication technologies.

The previous research on pass-by framework focused on the management of pass-by data mainly. Nevertheless, in order to the realization of the pass-by system, we have to implement proximity detection. The proximity detection performance depends on parameters, which is different depending on technology or execution environment such as "Communication Frequency" or "Communication Range". The modification of parameter settings influences the accuracy of pass-by detection and current consumption of devices. In order to implement the pass-by system which satisfies the needs of the users, the application developer must set appropriate parameters. However, if each developer researches appropriate parameters, the burden of their developments will increase.

Then, we provide the data which is the good reference for determination of setting parameters at various technologies or execution environments for the developers of the pass-by application. Thereby, we can achieve the reduction of the cost in the development of a pass-by application. In this study, we consider the correlation between settings of parameters depending on technology and proximity detection performance. First, we develop an application which utilizes proximity detection. In this paper, we use Bluetooth Low Energy (*BLE*)[4] as a proximity detection technology for the application. Next, we do evaluation experiments in order to research these items: the correlation between parameters and current consumption; the effects of distance between devices on the Received Signal Strength Indication (*RSSI*); and effects of moving speed on accuracy and frequency of detection. Finally, we analyze the experimental data and consider it.

2 Preliminary

2.1 pass-by system

We define an event in which the objects get close within a fixed distance and within a certain period of time is defined as *pass-by*. Moreover, we define a system which detects pass-by data and use this data for arbitrary operations as a *pass-by system*. Pass-by is detected in two methods. The first one is the method which detects pass-by from relative location information between devices. This method utilizes proximity detection by BLE or Ad-hoc mode of Wi-Fi. The second one is the method which detects pass-by from absolute location information such as latitude and longitude. The absolute location information is detected by a positioning system such as GPS.

2.2 Pass-by framework

On pass-by system, the proximity detection performance depends on the proximity detection technology. Furthermore, the format of pass-by data depends on the implementation

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

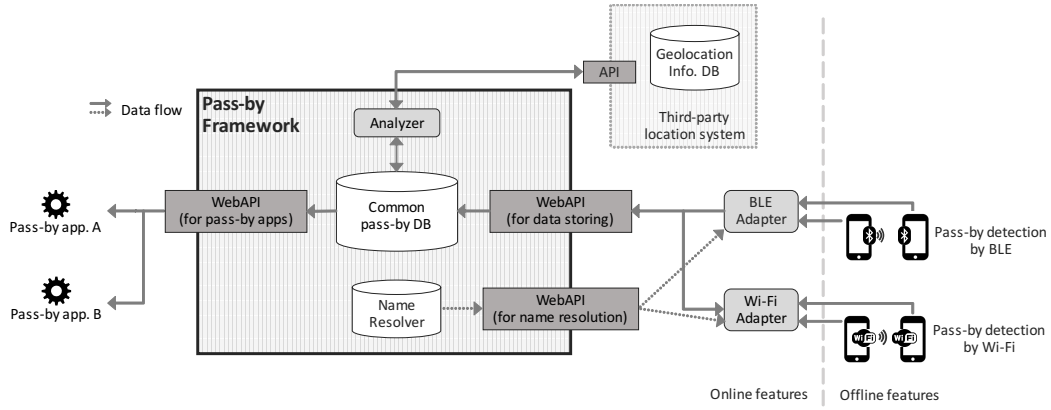

Figure 1: Pass-by framework

way of the pass-by system. Thereby, the burden on application developer for developing is large. In order to solve this problem, we propose pass-by framework[3], which is the mechanism to reduce the burden of the pass-by application development. This framework separate and abstract the pass-by detection mechanism depending on a device and a communication technology; and the pass-by data management which is utilized by applications. Thereby, an application developer can develop without minding types of device or technology.

The architecture of pass-by framework is shown in Figure 1. The Framework consists of the pass-by detection mechanism (*sonar*) and the server for data management. Pass-by is detected by sonar of the device. Then, the pass-by data is transmitted to *Adapter* which is provided for each pass-by detection technology, and the data is converted into the standardized format. This standardized data passes *WebAPI*, and then this is recorded in *Common pass-by DB*. Furthermore, the location informations which is recorded by GPS is transmitted to *Geolocation Info. DB*. *Analyzer* reasons new pass-by by utilizing both data of Common pass-by DB, and location information of Geolocation Info. DB. Then Analyzer transmits the new pass-by data to *Common pass-by DB*. When an application refers pass-by data in Common pass-by DB , the application can get pass-by data by using WebAPI.

2.3 BLE

BLE[4] is a part of the Bluetooth4.0 standard. The correct consumption of BLE is lower than Bluetooth before 3.0 (*Classic Bluetooth*). BLE utilize 2.4 GHz band as with Classic Bluetooth. Nevertheless, BLE does not have backward compatibility with Classic Bluetooth. Classic Bluetooth has a rule that it can connect up to seven devices, however, BLE is not set a maxumum number of connectable devices concurrently. In recent year, the devices supporting Bluetooth4.0 become popular. We expect that many devices utilize BLE as pass-by detection mechanism on pass-by systems in the future.

2.4 The scope of this study

In the past research of pass-by framework, we focused on the data management. Therefore, we have not consider the implementation method of pass-by sonar yet. In order to utilize the framework, we have to develop the pass-by sonar. The proximity detection performance changes with implementation method of the pass-by sonar. For utilizing a proximity

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

detection technology using radio wave such as BLE on devices, the developer must set parameters which affect the accuracy of pass-by detection and the current consumption. The parameters are such as the radio wave strength and detection frequency. If a developer set inappropriate parameters, the pass-by application causes serious problems such as rapid battery consumption or low accurate consumption. The parameters are different by pass-by detection technology.

Then, we provide the data which is a good reference for setting parameter of pass-by applications at various technology or execution environment. Thereby, we can achieve the goal of the pass-by framework, "Reduction of the development cost". In addition, even when the application developer does not use the framework, we present suitable settings of parameters for developers of the pass-by application.

In this study, we research variations of proximity detection performance which depends on the difference in setting the parameters. In this paper, as a first step, we check about parameters of BLE as a pass-by detection mechanism. The reasons for selecting BLE at first is that many devices adapt it. We expect that it is utilized as a typical pass-by detection in the future. On Android OS, the parameters of BLE include "Transmission Frequency", "Transmission Frequency" and "Reception Frequency". We consider the setting of appropriate parameters for pass-by detection from the points of view such as battery consumption, RSSI, and relative speed.

3 Implementation of proximity detection mechanism using BLE

3.1 Requirement of the experimental pass-by application

A pass-by application developer needs to consider the following E1-E3, which are elements depending on differences in the implementation of proximity detection mechanism.

E1 Influence of parameters to current consumption

E2 Correlation between RSSI and distance of targets

E3 Influence of the movement of target to pass-by detection

By researching E1, the developer becomes able to predict the current consumption by the application. By researching E2, the developer gets to know the decrement of radio waves from the device which is transmitting. By researching E3, the developer gets to know the influence of increasing speed on pass-by detection. In order to research the above E1-E3, we develop the pass-by application for experiments. Considering the above E1-E3, we have to implement the following three functions F1-F3 in the experimental application.

F1 Measuring current consumption

F2 Measuring distance

F3 Measuring speed

Furthermore, the following three additional functions F4-F6 are also necessary in order to do the experiments while changing parameter and recording log data of pass-by.

F4 Switching execution state of transmitting or receiving

F5 Modifying settings of parameters

F6 Recording result of pass-by detection

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

3.2 The experimental pass-by application

We develop the pass-by application for the experiments in the following development environments.

IDE : Android Studio 1.4

Target OS : Android5.0 Lollipop

Target device : Nexus9 (HTC, 2014)

In advertiser application (*BLEAdvertiser*), the byte string of advertising data is created according to the data format of iBeacon[5]. On this format, the data has *UUID*, which is identification of the application. Furthermore, the data has *major* and *minor* which are identifications of the device. In BLEAdvertiser, "*AdvertiseMode*" (transmission frequency) and "*TxPowerLevel*" (transmission strength) are changeable.

In scanner application (*BLEScanner*), the byte string which is transmitted from BLEAdvertiser is analyzed, and we can check UUID, major and minor of BLEAdvertiser. In BLEScanner, "*ScanMode*" (reception frequency) is changeable.

For measuring current consumption, we develop another application (*PowerConsumptionMonitor*). PowerConsumptionMonitor has a function that measures average current consumption and records it at regular time intervals[6].

4 Evaluation experiments

4.1 Current consumption

The desired current consumption of pass-by application is different by each purpose of using the application. For example, in the disaster situation, people cannot find an opportunity in order to charge the battery of mobile device easily. In this case, the applications such as the safety confirmation should make battery consumption as small as possible. Therefore, the pass-by application developer should know the relationship between the current consumption and the battery life. The relationship between a performance of the BLE communication and battery consumption is trade-off . Thus, the developer should know the difference between the current consumption depending on a difference of parameters.

4.1.1 Conditions

In Android OS environment, AdvertiseMode has three types ("LOW POWER", "BALANCE" and "LOW LATENCY"), and TxPowerLevel has four types ("ULTRA LOW", "POWER LOW", "POWER MEDIUM" and "POWER HIGH") for BLE advertising. For the evaluation of current consumption of BLE advertising, we measure current consumption about each combination of three AdvertiseMode and four TxPowerLevel. We calculate the result by the average of five trials about each combination. ScanMode has three types ("LOW POWER", "BALANCE" and "LOW LATENCY") for BLE scanning. Furthermore, Android device operates callback function when scanning all BLE signal (not only BLE signals from BLEAdvertiser). Therefore, the amount to receive the BLE signal and the contents of the callback function affect the current consumption. For the evaluation of current consumption of BLE scanning, we measure current consumption about each combination

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

Table 1: Conditions of experiment **4.1** on BLEScanner

	w/ callback	w/o callback
w/ surrounding BLE signals	S1	S2
w/o surrounding BLE signals	S3	-

Table 2: BLEAdvertiser's current consumption on Nexus9

Advertise Mode	Tx Power Level	CC (μA)	Increase (%)
LOW POWER	U.LOW	125.2	0.4
	P.LOW	437.8	1.3
	P.MEDIUM	562.8	1.6
	P.HIGH	2187.6	6.3
BALANCE	U.LOW	1625.2	4.6
	P.LOW	1593.8	4.6
	P.MEDIUM	1594.2	4.6
	P.HIGH	2250.0	6.4
LOW LATENCY	U.LOW	2187.6	6.3
	P.LOW	2594.0	7.4
	P.MEDIUM	3062.6	8.8
	P.HIGH	3187.6	9.1

of three types of ScanMode and three situations S1-S3 which are shown in Table 1. We calculate the result by the average of five trials about each combination.

In order to measure current consumption, we use PowerConsumptionMonitor, which we developed in **3.2**. By using this application in Nexus9, we can calculate the average of current consumption during 11.25 seconds just before. This application calculates and records the average of current consumption every 12 seconds even if the screen of the device is off. When we measure current consumption, we make the device operate nothing tasks without the exception of PowerConsumptionMonitor, BLEAdvertiser, and BLEScanner. Then, we measure the current consumption when the monitor of the device is off and the CPU of the device operates stably.

4.1.2 Result

The result of measuring the current consumption of BLE advertising is shown in Table 2. We call the current consumption when the device operate nothing (34999.4μA) as *natural current consumption*. "CC" in Table 2 is the difference between the current consumption when advertising and the natural current consumption. (In other words, it is the current consumption by BLE advertising or scanning). "Increase" is the rate of increase the current consumption compared to the natural current consumption. According to the result of Table 2, the current consumption of BLE is 9% of the natural current consumption at most.

The result of measuring the current consumption of BLE scanning is shown in Table 3. When BLEScanner is scanning and analyze UUID, the device consumes twice the current consumption as compared with the case when not analyzing UUID. Furthermore, the current consumption changes depending on the signal detection count. At the place where there are many BLE signals, the current consumption of BLE scanning becomes large.

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

Table 3: BLEScanner's current consumption on Nexus9

Situation	Scan Mode	CC (μA)	Increase (%)
S1	LOW POWER	10281.6	29.4
	BALANCE	36437.8	104.1
	LOW LATENCY	70281.4	200.8
S2	LOW POWER	6500.2	18.6
	BALANCE	18875.2	53.9
	LOW LATENCY	50844.2	145.3
S3	LOW POWER	2655.6	7.59
	BALANCE	4125.4	11.8
	LOW LATENCY	17625.2	50.4

4.2 Relationship of distance between devices and RSSI

The RSSI of the received radio wave decreases with increasing the distance from the transmission source. In BLEAdvertiser, communication distance of the signal is controllable by modifying AdvertiseMode. In BLEScanner, the device can check RSSI of scanned BLE signals.

In this experiment, we use two devices and try BLE communication while changing the distance between these devices. Then we check RSSI and check how radio waves intensity decreases. By referring data that obtained in this experiment, the developer can set the maximum distance to be included in the pass-by (the maximum pass-by distance L) by filtering data by RSSI.

4.2.1 Conditions

This experiment is carried out by the persons **A** and **B** who have a device. First, **A** start to advertise of BLEAdvertiser with setting AdvertiseMode to LOW LATENCY, and **A** stands still while keeping the device at the height 1m. **B** stands a certain distance away from **A**, and scan signals by BLEScanner during 20 seconds with setting ScanMode to LOW LATENCY. **A** and **B** do this trial about various distance (1, 5, 10, 15, ..., 100 meters) between devices. Furthermore, **A** and **B** do this trial about each type of TxPowerLevel in BLEAdvertiser.

4.2.2 Result

The experimental result of measuring RSSI about each TxPowerLevels and each distance (1, 5, 10, 15, ..., 100 meters) between the devices are shown in Figure 2. Four curve lines in Figure 2 show logarithmic approximations of results about each TxPowerLevels. It is said that the maximum distance which can be communicated by BLE is about 50 meters. Nevertheless, we can detect pass-by at the distance of more than 50 meters when TxPoweerLevel is POWER HIGH or POWER MEDIUM.

Average RSSI is -93dBm when the distance between devices is 5 meters and TxPowerLevel is ULTRA LOW. If the developer implements the pass-by system that judges BLE signals as pass-by when RSSI is over -93dBm and TxPowerLevel is ULTRA LOW, the developer can develop pass-by detection mechanism that the maximum pass-by distance L is 5 meters.

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

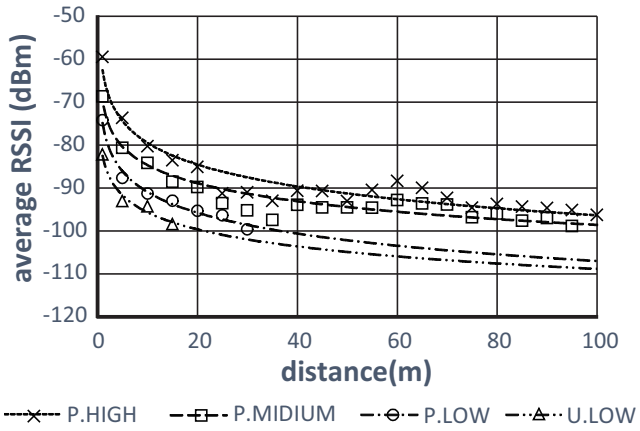

Figure 2: The distance between devices and average RSSI

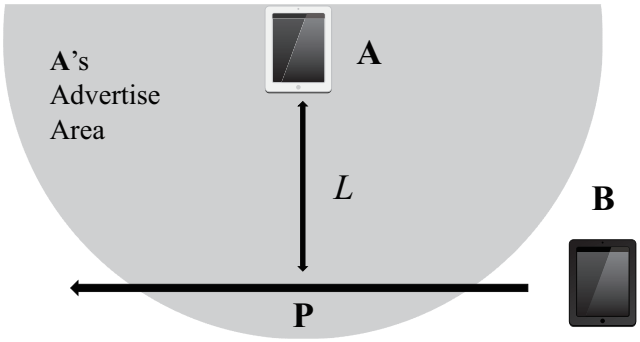

Figure 3: Conditions of experiment **4.3**

4.3 Relationship of speed and detection accuracy/frequency

We expect that device move in common pass-by situations. We consider that there is dependence between moving speed and RSSI, or that between speed and detection accuracy about pass-by. Proximity detection performance may be different depending on moving speed of devices. The developer should get to know how the speed of the device affects proximity detection performance.

4.3.1 Conditions

In this experiment, we set the maximum pass-by distance L to 5 meters. This experiment is carried out by the persons **A** and **B** who has a device each. The device of **A** handles BLEAdvertiser, and the device of **B** handles BLEScanner. In this experiment, both devices are put in backpacks. The illustration which expresses the method in this experiment is shown in Figure 3. The point 5 meters away from **A** is defined as **P**. At First, **A** stands still and **B** stands at out of **A**'s BLE advertising range. **B** moves at a uniform speed on the line, which is perpendicular to **AP** and passes on **P**, to the another side of out of **A**'s BLE advertising range. In this experiment, **B** moves at the speeds of the walk (about 5km/h), run (about 10km/h) or bike (about 20km/h). We do this trial at ten times about each speed. After finishing a total of thirty trials, **A** and **B** exchange device and do the same experiment. In this experiment, TxPowerLevel is set the smallest parameter (POWER LOW) that can

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

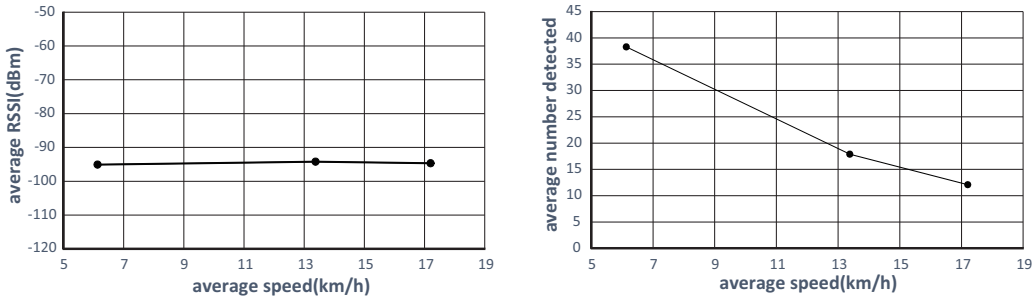

Figure 4: The average of RSSI and the average number of scanned signal about each speed

detect pass-by certainly when **B** stands still at point **P**.

4.3.2 Result

The success rate of pass-by detection is over 90% about each three speeds. The following shows the result when the receiver moves. The result of analyzing RSSI about each speed and the average number of scanned signals are shown in Figure 4. We find that the average of RSSI does not change mostly at all speeds. In conclusion, the influence to RSSI by speed is small enough to be ignored. Furthermore, the average number of scanned signals tends toward a decrease in the form being direct proportion to speeds.

5 Discussion

5.1 Summary of findings

The current consumption of BLE advertising is under 10% of the natural current consumption with all combination. Therefore, when the developer develops power saving application, the setting of TxPowerLevel affects only a few on the power saving. Furthermore, because BLEScanner executes the callback function when a device scan BLE signals, the current consumption change by the content of the callback function and the setting of ScanMode. We consider that the current consumption increases in a situation that many people who use BLE application exist. Developing the pass-by application, the developer should consider a situation of using the application at where there are many people.

We find that RSSI change only a few by changing the speeds of the devices. If we do the experiment with even greater speed, RSSI might not change too large probably. Nevertheless, the number of scanned signals decrease with the speed ups. We consider that the time when the scanning device shortens as speed increases. Hence, the maximum of the speed that the device can detect pass-by is finite.

All devices which are used in these experiments are Nexus9. If we use other devices than Nexus9, the results of these experiments are not same as this time. If we want to utilize the result of this experiment as a useful data for the sonar of pass-by framework, we need to do experiments using other devices and research the difference of results between types of devices.

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.

5.2 Applications

We expect developments of various new applications, which utilize pass-by systems. The pass-by systems can be applied in many applications for various purposes, such as safety confirming of close family members in a disaster, and searching lost articles. BLE is one of the easiest technology for realizing pass-by system. In this study, we showed the performances of pass-by by BLE as the findings. By utilizing these findings, the pass-by application developers can set suitable parameters in accordance to purposes of applications. Consequently, the developers can develop pass-by applications more efficiently by the findings of this study.

6 Conclusion

In this study, in order to implement the sonar of pass-by framework, we aim at taking the reference data about the difference of proximity detection performance between settings of parameter about each proximity detection technologies. In this paper, we showed the reference data of BLE in Android device by evaluation experiment. As the challenges for the future, we will do the same experiment using other Android devices or iOS devices, and we should show the data which is more generic. Then, we will confirm general versatility after implementation by the method such as a simulation.

Acknowledgments

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (16H02908, 15H02701), Grant-in-Aid for Scientific Research (A) (17H00731), Challenging Exploratory Research (15K12020)], and Tateishi Science and Technology Foundation (C) (No.2177004).

References

[1] Tokyo National Museum, ”Tokyo national museum - applications about TohakuNavi,” http://www.tnm.jp/modules/r_free_page/index.php?id=1467&lang=en.

[2] Nintendo Co. Ltd., ”StreetPass - Nintendo 3DS - SpotPass Information, Details,” https://www.nintendo.com/3ds/built-in-software/streetpass.

[3] A. HAYASHI et al., ”Designing feature of generating and storing common pass-by data in pass-by framework,” in IEICE Technical Report, vol. 115, no. 371, December 2015, pp. 19-24.

[4] Bluetooth SIG Inc., ”Low Energy: Point-to-Point — Bluetooth Technology Website,” https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-p2p.

[5] A. Uehara, iBeacon handbook, Tatsujin Press, 2014.

[6] Google Inc.,”Measuring device power - android open source project,” https://source.android.com/devices/tech/power/device.html.

Copyright © by IIAI. Unauthorized reproduction of this article is prohibited.