International Journal of Grid and Distributed Computing
Vol. 11, No. 4 (2018), pp.49-58
http://dx.doi.org/10.14257/ijgdc.2018.11.4.05

Building Customizable Beacon Services for Pattern-specifiable Beacons

Jae-Hwan Jin[1] and Myung-Joon Lee[2*]

[1,2]*Department of Electrical/Electronic and Computer Engineering, University of Ulsan, 93, Daehak-ro, Nam-gu, Ulsan 44610, Republic of Korea*
[1]*jjhok2000@gmail.com*, [2*]*mjlee@ulsan.ac.kr*

Abstract

The beacon service is a beacon identification service or a location-based service which operates by receiving beacon data transmitted from a Bluetooth beacon device. Beacon data generated from each beacon plays a crucial role in providing a beacon service because it includes the type and identification of the service and the status information of the service user. To provide a sophisticated beacon service, various beacon data should be generated according to the beacon usage patterns of the user. This leads us to develop a service for handling such various beacon data.

In this paper, we propose a user-defined beacon service providing methods for effectively analyzing and processing the data broadcasted from a pattern-specifiable beacon, which can generate different data according to the patterns of pushing the button attached to the beacon device. Also, we present the customized beacon service developed using the proposed technique as an Android application. The application is composed of a service daemon and a service manager. It supports systematic specification and management of function execution related to the beacon usage pattern of a user.

Keywords: *Bluetooth Beacon, Beacon Service, Button Beacon, Pattern-specifiable Beacon, Beacon Data*

1. Introduction

The beacon service receives the beacon data transmitted from the BLE beacon device and provides a service suitable for the situation. Most beacon services are user identification services using beacon IDs or location-based services using beacon's Received Signal Strength Indicator (RSSI) values [1]. The beacon data is used as the main context information for providing a rich service to the user of the beacon service.

Generally, a beacon device broadcasts a fixed beacon data to neighboring beacon receivers. In this traditional beacon data delivery method, there is a limit to provide more diverse and complicated beacon services in that the data that can be delivered is fixed and the intention and context of the user cannot be utilized dynamically. To address this issue in the next generation beacon service, various types of beacon data transmitted by a single beacon device should be diversified, and the data value also should be dynamically changeable according to the user's request. This allows beacon service providers to obtain a variety of additional information from the beacon device and to provide a personalized beacon service in real time. However, current beacon services have no interests in handling various beacon data transmitted from a single beacon device. So, to provide a sy analyzing and processing the data broadcasted from a pattern-specifiable beacon, which can generate different data according to the patterns of pushing the button attached to the beacon device. Based on the broadcasted beacon data according to a user's operation

Received (October 19, 2017), Review Result (January 26, 2018), Accepted (January 30, 2018)
* Corresponding Author

ISSN: 2005-4262 IJGDC

pattern, the proposed method launches the service function predefined for the beacon data received from the beacon device. For this, the proposed method provides systematic specification and management of function execution related to the beacon usage pattern of a user. It also handles the execution parameters used with the executables to enable more complex beacon services for rich functionality. In addition, we describe the development of a user-defined beacon service using the proposed technique. Coupled with a button beacon, the developed user-defined beacon service presents the service daemon and the beacon pattern management tool in an Android application. The service daemon provides various functions such as emergency call, sending SMS, taking photo and video, playing music, bursting alarm, and running installed application according to the specified function.

The rest of this paper is organized as follows. We summarize background knowledge on the Bluetooth beacon and the beacon service in Section 2. In Section 3, the user-defined beacon service is described. In the next section, a development of user-defined beacon service specifier is presented. Finally, the authors conclude the paper in Section 5.

2. Background

2.1. Bluetooth Beacon

A Bluetooth beacon is a short-range wireless communication device that transmits unidirectional information at low power according to the Bluetooth 4.0 protocol [2]. Beacons are now attracting attention as IoT-based solutions because they are easy to measure the indoor location and can transmit data over a longer distance than the NFC technology [3]. Generally, a Bluetooth beacon periodically broadcasts a fixed beacon data to a nearby Bluetooth device. The beacon data is available in 31 bytes, and various beacon data types are defined, such as Apple's iBeacon [4] and Google's EddyStone [5], depending on the purpose. Bluetooth beacon developers use the modifiable area of the beacon data type to convey the user's identity or URL. In recent years, there have been emerged services that utilize sensor data together with beacon data by attaching temperature, humidity, light intensity sensor, *etc.*, to beacon devices so as to effectively utilize the beacon in the IoT environment.

2.2. Beacon Service

Surely, a beacon service is provided based on the beacon data transmitted from the Bluetooth beacon device. Popular beacon services are a beacon identification service that uses the identification ID of beacon data, and a location-based service that calculates the distance between a beacon and a receiver. The beacon identification service is included in applications such as a coupon-providing service in offline markets, and a student automatic attendance confirmation system [6, 7]. The location-based service using the beacon measures the distance between the beacon device and the beacon receiver to provide a service according to the real-time location of the beacon device. Location-based services are used in large stadium seat guidance systems and indoor localization and navigation system [8]. Recently, a beacon service platform has emerged so that a service provider can easily construct a high-level beacon service [9, 10]. Service providers can use the beacon service platform to specify a beacon data type according to the purpose and to specify a beacon service to provide a customized beacon service as well.

3. User-defined Beacon Service

In this section, we describe a new technique to provide user-defined beacon services, which can utilize various kinds of beacon data. The user-defined beacon service is used

with a beacon device such as a button-shaped beacon, which is able to transmit various beacon data according to a user's beacon operation pattern.

3.1. Pattern-specifiable Beacon

A typical beacon device continuously transmits a fixed beacon data to a beacon receiver for beacon service. However, in order to provide a high-level beacon service, it is necessary to transmit different kinds of beacon data according to a user's request and execute the corresponding functions. So, the beacon device used in the proposed service should transmit different beacon data according to a pattern in which the user operates the beacon. As a typical example, a button beacon device can generate various patterns by clicking the attached button. The button beacon supports long or short clicks according to the method of clicking the button, using it to generate a beacon pattern. When the user clicks the button in accordance with the pattern, the beacon data corresponding to the pattern is broadcasted to the neighboring beacon receiver. Such beacon devices may also specify pattern types and the associated beacon data via Bluetooth pairing technology. Figure 1 shows the process of generating the pattern through the button beacon device and transmitting various beacon data.

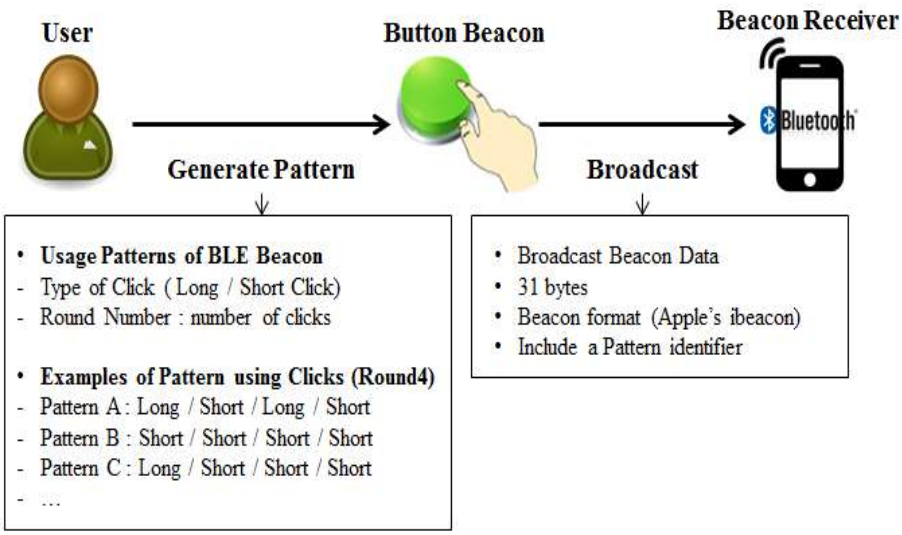

Figure 1. Process of Generating a Beacon Pattern

Unlike normal beacons, the button beacon broadcasts beacon data repeatedly for a limited period of time when the user presses the button. In the proposed beacon service using the button beacon, the beacon data should be recognized only once and the receiver should execute the corresponding action only once in one period while the beacon data is broadcasted. To prevent duplicate recognition of the broadcasted data in the beacon receiver, we assign the same random value to the beacon data in one broadcasting period. The beacon receiver checks the random value and does not take any action if it is equal to the previously received random value. Through this method, a beacon service using a button beacon can provide the functionality demanded by the user in a rich and correct way.

3.2. Structure of the Proposed Service

The proposed custom beacon service is composed of a button beacon, the beacon service daemon, and the pattern management tool. Figure 2 below shows the structure of the proposed service.

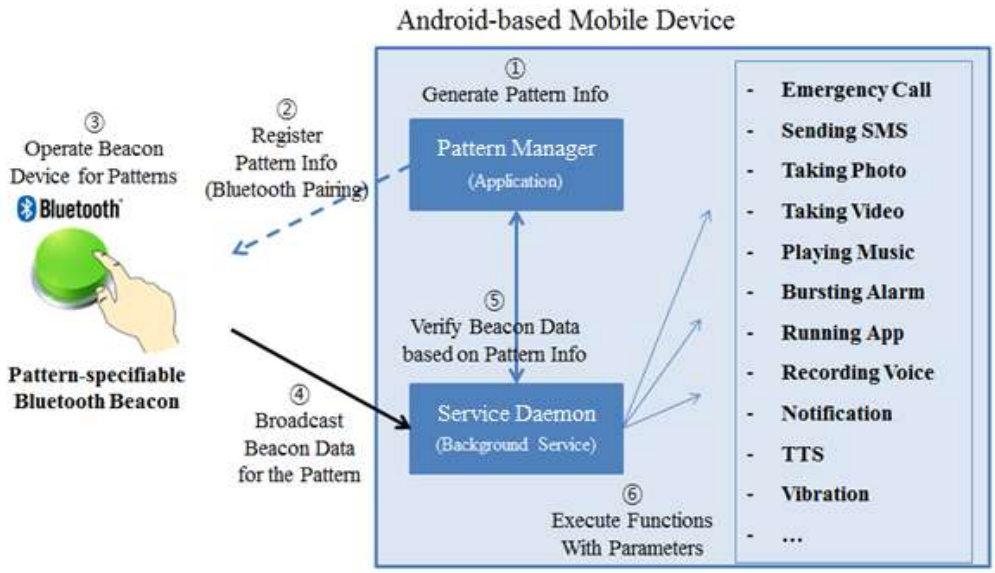

Figure 2. Structure of User-defined Beacon Service

When the button beacon broadcasts its data according to the user's operation, the beacon service handles the data through the pattern management tool and the beacon service daemon. The beacon pattern management tool is used to specify the beacon pattern to be used in the button beacon and to associate the execution function with the beacon pattern. In addition, the beacon pattern management tool systematically manages the information specified by the user and transmits the specification information to the beacon device through Bluetooth Pairing technology. The beacon service daemon is a tool that receives beacon data broadcasted from a beacon device and executes the associated functions according to the types of beacon data. For this, the beacon service daemon checks the beacon data based on the pattern information specified previously in the beacon pattern management tool, and perform the execution function (with the parameters) associated with the pattern.

3.3. Process of the Proposed Service

The proposed beacon service is provided in the following order: a pattern specification step, a beacon data reception and identification step, and a beacon service function execution step. The pattern specification step is a process in which a user specifies patterns to be used in a button beacon through the beacon pattern management tool. Figure 3 shows the activity diagram for the pattern specification step of the proposed service.

The information contained in a beacon pattern includes the type of the pattern, the beacon data to be broadcasted from a beacon device, the function to be executed on the receiver, and the parameters for the execution function. Upon completion of the beacon pattern specification by the user, the beacon pattern management tool transmits the specified pattern and corresponding beacon data information in connection with the button beacon. At this time, to use the beacon device more precisely, the beacon pattern management tool can also transmit the beacon transmission period designated by the user. The beacon device then saves the beacon pattern and beacon data based on the received specification information. With this, the button beacon can broadcast the beacon data specified in the pattern management tool according to the pattern designated by the user.

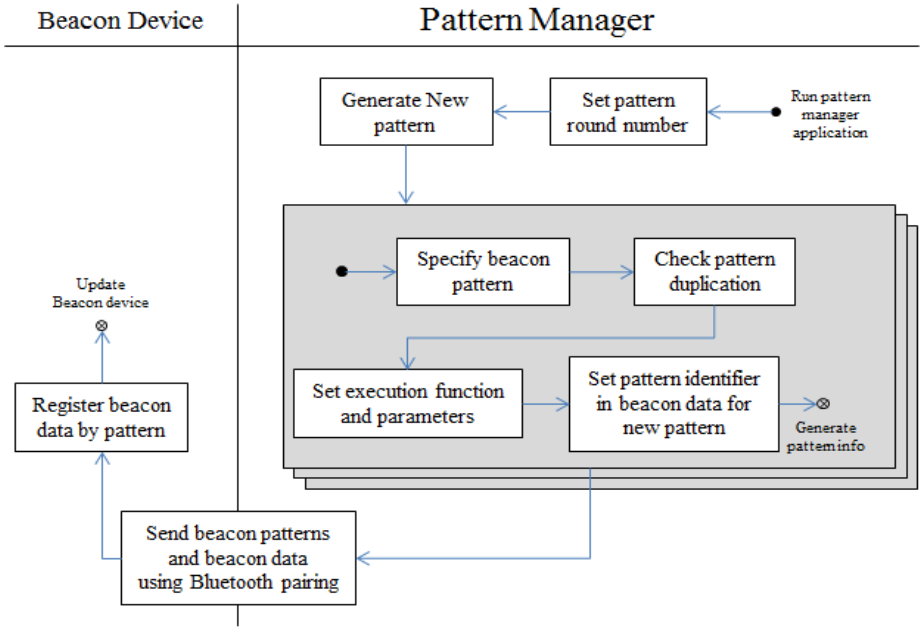

Figure 3. Activity Diagram for Pattern Specification Step

In the pattern reception and identification step, the beacon service daemon receives the beacon data broadcasted from the button beacon and recognizes the pattern. Figure 4 shows the process of the beacon data reception and identification step. Upon receipt of the beacon data, the beacon service daemon checks the pattern information stored in the beacon pattern management tool, finding the pattern which matches with the received beacon data based. If there is a pattern matching with the beacon data, the beacon service daemon recognizes the pattern including the execution function and parameter information associated with it.

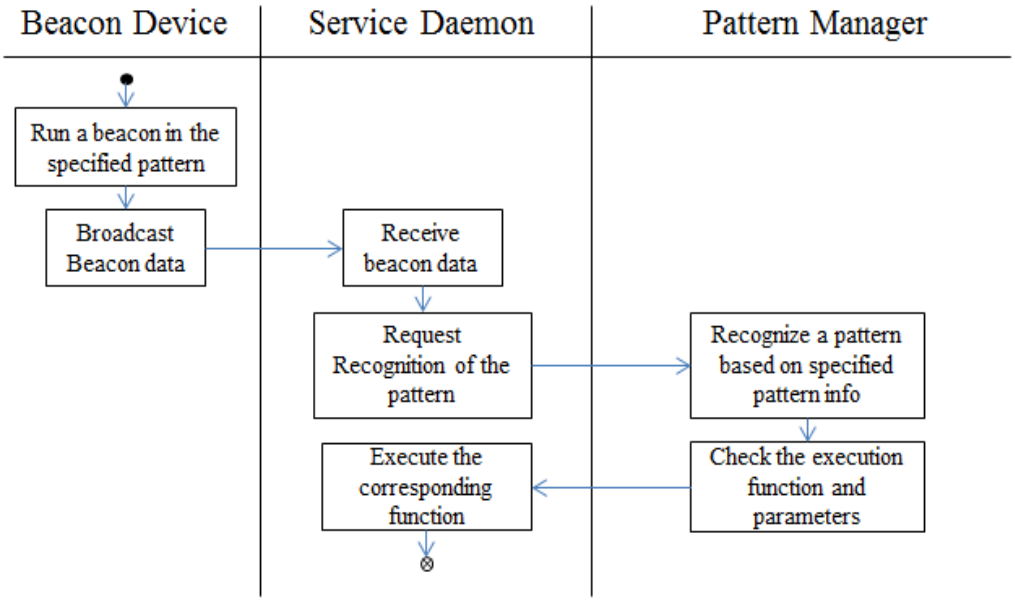

Figure 4. Activity Diagram for Beacon Data Identification Step

In the beacon service function execution step, the execution function associated with the pattern recognized in the pattern identification step is performed. In this step, the beacon service daemon executes the associated function along with the parameters assigned in the pattern specification information. Since the user operates by using the button beacon, the function control should be simple and executed immediately. So, based on the scenarios in the operating environment using the button beacon, we have summarized the appropriate functions as Table 1.

Table 1. Execution Functions of Proposed Service

Function Type	Function Name	Parameters
Functions with Parameters	Emergency Call	Phone Number
	Sending SMS	Phone Number, Text
	Playing Music	Music file path
	Playing Voice	Text
	Notification(Alarm)	Time(s)
	Vibration	Time(s)
Functions without Parameters	Taking Photo & Video	-
	Recording Voice	
	Running Installed Apps	

Execution functions are classified according to the presence or absence of parameters required for execution. The functions with parameters include emergency call, sending SMS, and playing music. On the other hand, the functions that do not require execution parameters include taking photo and video, recording voice, and running installed applications. The functions that require parameters are based on the information designated previously by the user so that functions tailored to the situation can be executed immediately.

4. User-defined Beacon Service Specifier

In this section, we describe the development of a user-defined beacon service specifier using the proposed technique. The specifier is developed as an Android application and a background service to effectively provide user-defined beacon services.

4.1. System Parameters

The user-defined beacon service specifier consists of a beacon device and an Android-based service daemon and pattern management application. We define service by Major and Minor areas using the 2-byte iBeacon data format, respectively, to broadcast beacon data in Apple's iBeacon data format. As shown in Figure 5, the button beacon broadcasts BLE packets including the pattern ID, which is the information specified in advance in the major area. The minor area contains a random value for preventing duplicate recognition. In this way, the button beacon can deliver a predefined value according to the beacon pattern input from the user. Patterns used in button beacons are generated by combining up to four times of long/short clicks.

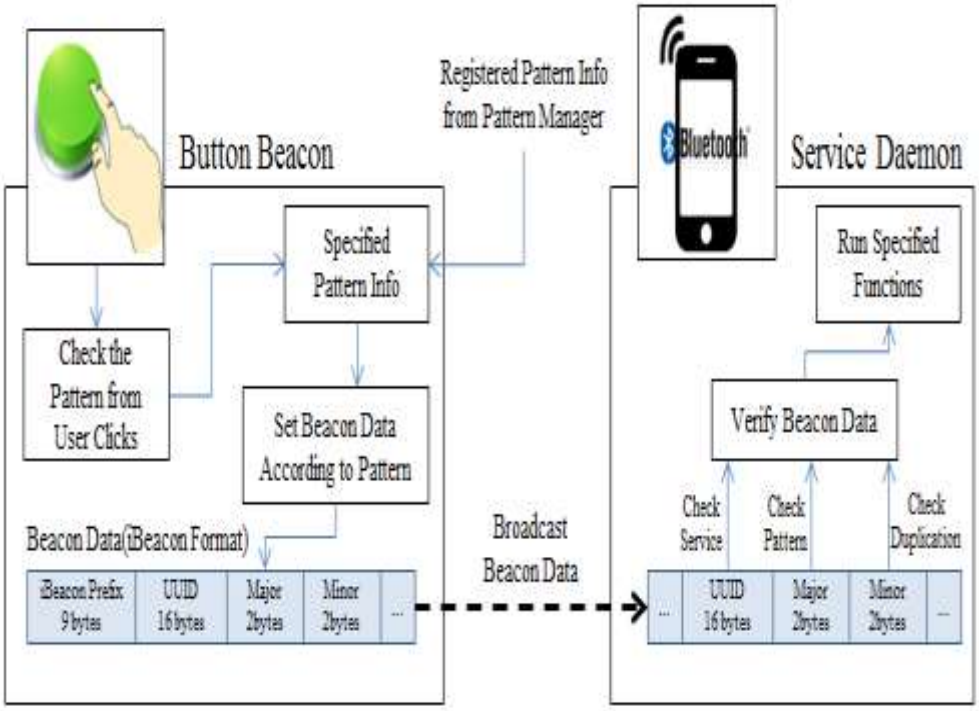

Figure 5. Process of Broadcasting Beacon Data

The beacon pattern management application stores the user-specified pattern information in a SharedPreferences repository of Android so that it can be inquired and utilized by the service daemon. This repository stores the information in the key-value format, which uses the user's pattern information as the key and the execution function and parameters for the pattern as value. In this way, the stored specification information can be retrieved from the service daemon, which is an Android background service. The service daemon is always running as a background service when the Android device starts up. The operation of the service daemon can be controlled by the beacon pattern management application to control the start and end of the service daemon. The service daemon scans the peripheral beacon device every 500ms to receive beacon data. When the service daemon receives the beacon data, it checks the major and minor areas in the iBeacon data format and processes the corresponding operation. First, the service daemon retrieves the pattern through Major and verifies that the function is already executed through Minor. In this way, the service daemon can immediately execute the defined service from the beacon signal.

4.2. User Interface

In the developed system, the user performs the beacon pattern management application to specify the pattern. In order to easily create a new pattern, the beacon pattern management application provides the input method of Long / Short according to the round number, which means the number of combinations of patterns, as shown in Figure 6. The round number for generating the pattern is specified when the user first creates the pattern and cannot be changed while the pattern is being generated.

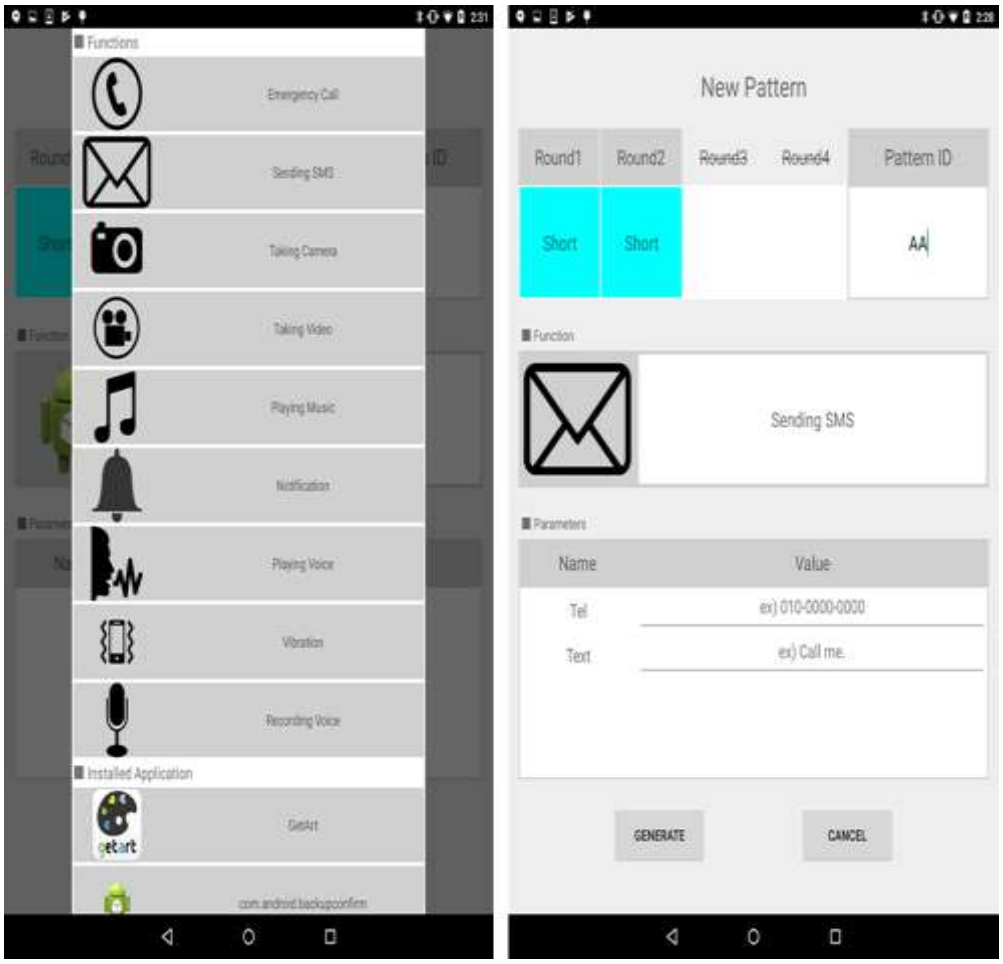

Figure 6. Screens of Generating a Pattern

The user can select the pattern type and specify the pattern ID and execution function. The pattern ID is the data of the major area transmitted from the beacon device when the user inputs the beacon pattern. The user can choose between Android basic functions and installed applications as the execution function. When the user inputs both the execution function and execution parameters, the pattern generation is completed. The generated pattern information can be confirmed by the pattern list on the main screen of the application as shown in Figure 7.

In the main screen, the user can change the round number of the pattern, and the summary of the added pattern is displayed. In addition, the user can test the added execution function with execution parameters by using the test button on the right side of each pattern. When the pattern registration is completed, the specification information is transmitted to the beacon device using the pattern registration button at the bottom of the screen. After registering the specification information on the beacon device, the user can click the service daemon start button at the top of the screen, starting the service daemon.

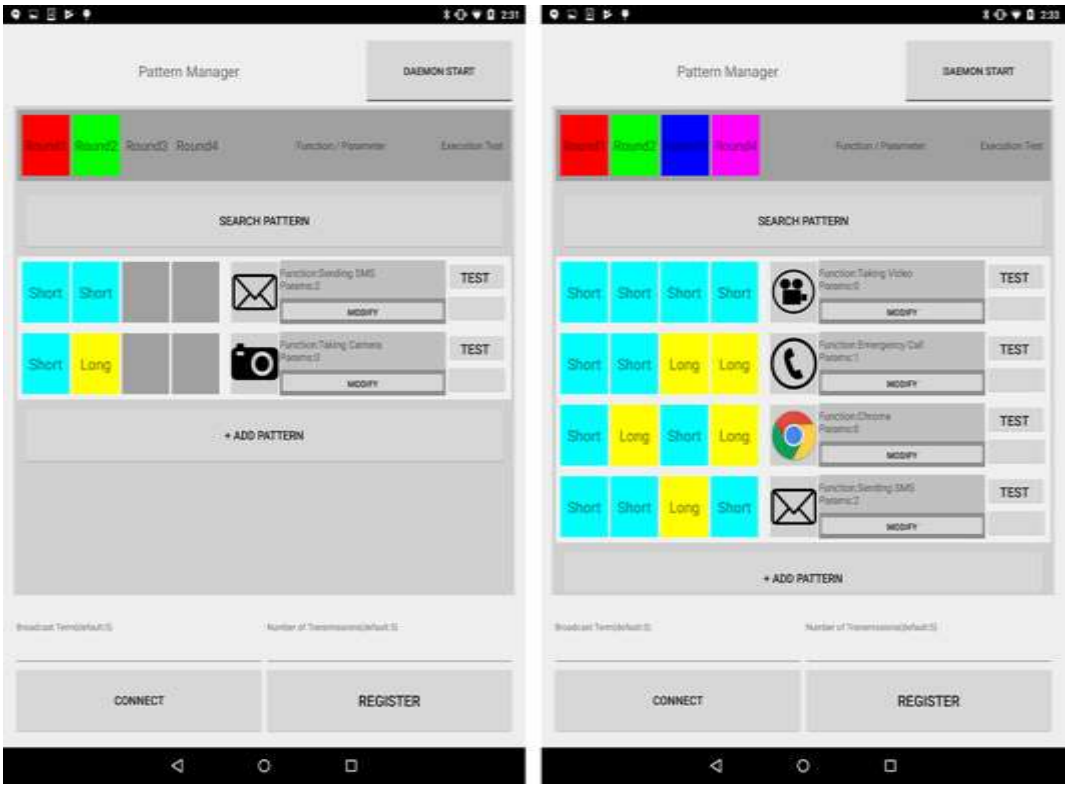

Figure 7. Screens of Managing Patterns

5. Conclusion

We proposed a method to provide a user-defined beacon service using beacons with pattern specification capability. In the method, beacon devices such as button-type beacons are used to transmit different kinds of beacon data according to the user's operation pattern. The service provider specifies a beacon operation pattern and an execution function to define a beacon service to run according to the pattern. The user-defined beacon service systematically manages the specified pattern information and provides various beacon services to the user according to the beacon data. We also described a customized beacon service using Android mobile application based on the proposed technique. In the developed service, we use a button type beacon device which transmits various beacon data according to the patterns of pushing the attached button, and a pattern specification management tool and a beacon service daemon are implemented in an Android application. Through the developed service, service users can execute installed applications and basic functions of Android such as emergency call, sending SMS, bursting alarm as needed. The proposed scheme can be extended to services using various types of beacons with more complex patterns as well as button-type beacons. In addition, the services using the proposed technique can provide various beacon services with a single beacon device, which can greatly reduce the cost and time required to construct a rich beacon service environment.

Acknowledgments

This paper is a revised and expanded version of a paper entitled "User-defined Beacon Services for Pattern-specifiable Beacons" presented at the 14th 2017 International Interdisciplinary Workshop Series, which was held on 21st-23rd December in Daejeon (Korea). This work was supported by the Technological Innovation R&D Program (S2437993) funded by the Small and Medium Business Administration (SMBA, Korea).

References

[1] S. Kajioka, T. Mori, T. Uchiya, I. Takumi and H. Matsuo, "Experiment of indoor position presumption based on RSSI of Bluetooth LE beacon", Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference, **(2014)**, pp. 337-339.
[2] Bluetooth 4.0 Spec, http://www.bluetooth.org/Technical/Specifications/adopted.html.
[3] K. Nair, J. Kulkami, M. Warde, Z. Dave, V. Rawalgaonkar, G. Gore and J. Joshi, "Optimizing power consumption in iot based wireless sensor networks using Bluetooth Low Energy", Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on IEEE, **(2015)**, pp. 589-593.
[4] Apple iBeacon, http://developer.apple.com/ibeacon.
[5] EddyStone, http://developers.google.com/beacons.
[6] K. J. Lee and K. H. Lee, "A Study of Security Threats in Bluetooth v4.1 Beacon based Coupon Convergence Service", Journal of the Korea Convergence Society, vol. 6, no. 2, **(2015)**, pp. 65-70.
[7] M. Y. Bae and D. J. Cho, "Design and Implementation of Automatic Attendance Check System Using BLE Beacon", International Journal of Multimedia and Ubiquitous Engineering, vol. 10, no. 10, **(2015)**, pp. 177-186.
[8] C. H. Yun and J. M. So, "A Bluetooth Beacon-Based Indoor Localization and Navigation System", Advanced Science Letters, vol. 21, no. 3, **(2015)**, pp. 372-375.
[9] J. H. Jin, W. G. Lee, H. S. Koo and M. J. Lee, "BSmart: A Service Platform for Rapid Development of Beacon-based Applications", International Journal of Smart Home, vol. 10, no. 7, **(2016)**, pp. 125-132.
[10] Y. U. Ha, J. H. Jin, H. S. Koo and M. J. Lee, "Automatic Generation of BLE Beacon Applications Using Service Specifications", Contemporary Engineering Sciences, vol. 9, no. 19, **(2016)**, pp. 925-932.

Authors

Jae-Hwan Jin, is a Ph.D. student in the School of IT Convergence at the University of Ulsan in Korea. He received his M.S. from the University of Ulsan in Korea. His research interests include Beacon-based service, cloud storage service and collaborative system.

Myung-Joon Lee, is a professor in the School of IT Convergence at the University of Ulsan in Korea. He received his Ph.D. from the Kaist (Korea Advanced Institute of Science and Technology) in Korea. He has (co-)authored more than 200 research publications including numerous works on collaborative system, distributed system and biological system.

