International Journal of Future Generation Communication and Networking
Vol.10, No.7 (2017), pp.1-12
http://dx.doi.org/10.14257/ijfgcn.2017.10.7.01

Bluetooth-enabled Software Defined Radio Platform

Hyoseok Yoon*[1], Ji-Eun Lee[2], Saet-Byeol Yu[3] and Se-Ho Park[4]

[1,2,3,4]*Korea Electronics Technology Institute*
[1]*hyoon@keti.re.kr*, [2]*jieun7583@keti.re.kr*, [3]*sbyu@keti.re.kr*, [4]*sehopark@keti.re.kr*

Abstract

There are already several digital radio standards available such as DAB, DRM and HD-Radio. In addition, multiple recent proposals are being standardized and deployed that would require different configurations to be used in different regions, devices and applications. To efficiently switch among multiple standards in a convergent platform such as an in-car radio platform, we adopt the concept of software defined radio. By using software defined radio, we support a range of standards and various Bluetooth-enabled applications through relatively convenient software updates. In this paper, we propose a Bluetooth-enabled software defined radio platform to communicate through Bluetooth for controlling and changing configurations of software defined radio and realizing connected audio applications.

Keywords*: Bluetooth, Software Defined Radio, Bluetooth Profile, Digital Radio*

1. Introduction

Currently, various digital radio standards including HD-Radio[1], DAB[2] (Digital Audio Broadcasting), DAB+, DRM[3] (Digital Radio Mondiale), DRM+, CDR[4] (China Digital Radio) and ISDB-Tsb[5] are available and adopted in different countries for providing localized and tailored broadcasting services. Furthermore, specifications of recently proposed standards are constantly being updated. Both devices and application need to conform to the latest specification. For this reason, we consider a flexible update strategy used in Software Defined Radio (SDR) to provide "different levels of reconfiguration within a transceiver" [1] or "reconfigurable architecture" [2]. In this paper, we propose a Bluetooth-enabled SDR Platform (BeSDRP) for updating and controlling multiple digital radio standards. We also present implementation details of Bluetooth profile tester and smartphone based controller application. Lastly, we discuss application scenarios to demonstrate benefits of the proposed BeSDRP.

2. Design of Bluetooth-enabled SDR Platform

We designed our BeSDRP based on two sub-modular platforms: a multi-standard SDR platform for receiving various digital radio standards and a Bluetooth connected application platform for Bluetooth connectivity to smart devices. By supporting multi-standard digital radio, developers and service providers can select standards to use as audio sources and use for connected audio applications via Bluetooth communication. An established Bluetooth connection is used to control software updates via an SDR control interface and to verify the current status of the multi-standard SDR platform. Fig. 1 shows an overall architecture of the BeSDRP.

---

* Corresponding Author
[1] http://www.hdradio.com
[2] http://www.worlddab.org/
[3] http://www.drm.org/
[4] http://www.chinesestandard.net/PDF-English-Translation/GYT268.1-2013.html
[5] http://www.dibeg.org/

ISSN: 2233-7857 IJFGCN

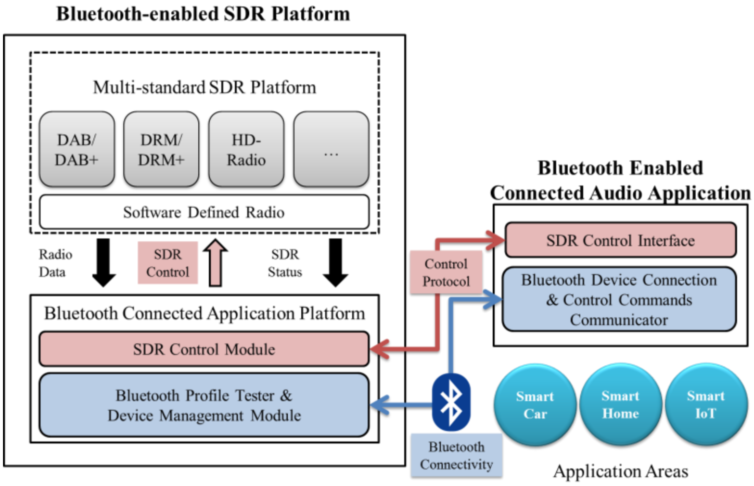

Figure 1. Overall Architecture of the BeSDRP

2.1. Multi-Standard Digital Radio for Connected Audio Applications

To implement connected data/audio applications based on Bluetooth communication, we need to access and control Bluetooth software stack. For our purposes, the proposed BeSDRP needs to provide API (Application Programming Interface) to inquire, discover, connect, pair, and authenticate a variety of Bluetooth devices such as headset, video game controller, tracker, keyboard and speaker. Providing these APIs allows developers to implement various connected data/audio applications that use multi-standard digital radio data. Additionally, we need a mechanism to check and manage a list of connected Bluetooth devices for multi-user and multi-device use cases as well. Furthermore, to increase interoperability with previously developed Bluetooth devices, we integrate development and testing supports for classic Bluetooth profiles. When a Bluetooth connection is established through one of the supported Bluetooth profile such as Serial Port Profile (SPP), then the BeSDRP is able to transmit multi-standard radio stream to the connected Bluetooth devices. Details of Bluetooth profile tester are presented in Section 3.

2.2. SDR Control Interface

In the BeSDRP, an SDR control module receives SDR control commands from a Bluetooth-connected device such as a smartphone using a pre-defined protocol of data packet and control flow as shown in Figure 2. Details of a smartphone-based controller application are presented in Section 4.

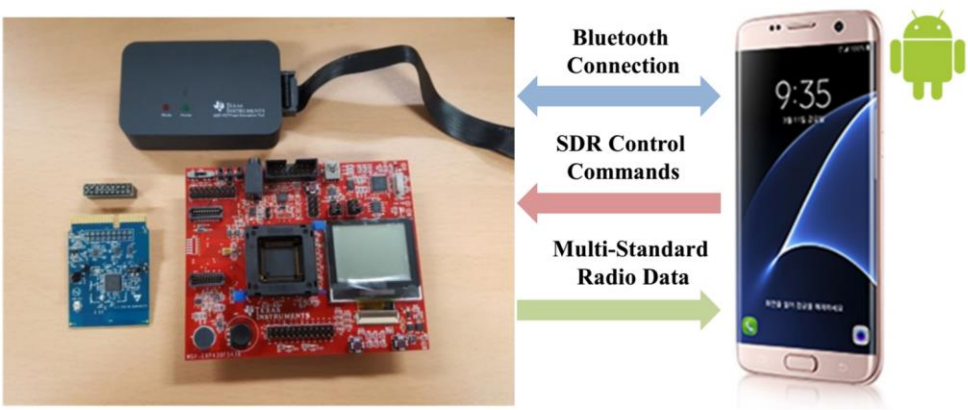

Figure 2. Communication Between the BeSDRP and a Bluetooth Device

In accordance with this protocol, the BeSDRP parses and interprets received control commands to identify which standard is in use and to configure internal settings accordingly. To send digital radio data and responses to control commands, the BeSDRP uses SPP as a base profile. Table 1 summarizes different profile usages for connected audio applications and the SDR control interface in the BeSDRP.

Table 1. Supported Bluetooth Profiles in the BeSDRP

Bluetooth Profile	Connected Audio Applications	SDR Control Interface
Serial Port Profile (SPP)	O	O
Human Interface Device (HID)	O	
Hands-Free Profile (HFP)	O	
Customized (SPP & A2DP)	O	O

3. Bluetooth Profile Tester

3.1. Concept of Bluetooth Profile Tester

We implemented the BeSDRP on TI CC256X Bluetooth module[6] and TI Bluetooth stack[7]. To assist developers for rapid development, we developed a PC version of Bluetooth profile tester using the TI Bluetooth stack for several Bluetooth profiles including SPP (Server/Client), Human Interface Device (HID) and Hands-Free Profile (HFP) to simulate and test different configurations of data communication, control commands and Bluetooth devices. The BeSDRP is designed to send and receive data and control commands to and from an SPP server while connected Bluetooth devices (ex., Android smartphones) act as SPP clients. Fig. 3 illustrates the concept of testing various Bluetooth devices, Bluetooth profiles and processes through APIs.

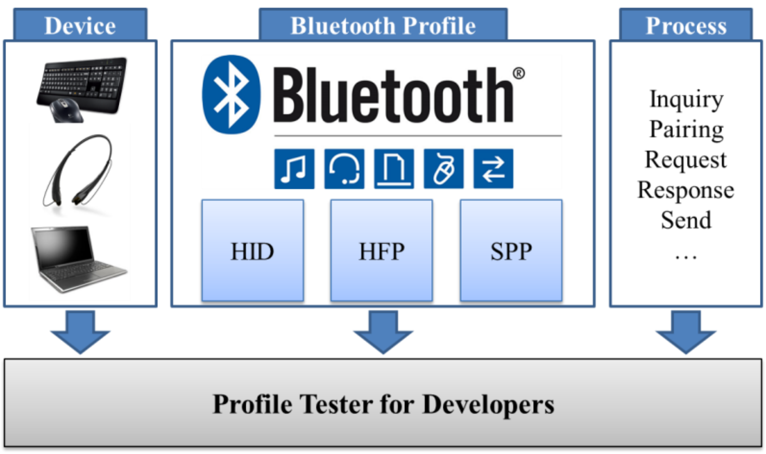

Figure 3. Profile Tester for Various BT Devices, Profiles and Processes

3.2. Implementation

The goal of our Bluetooth profile tester was to assist developers easily simulate and test many Bluetooth classic profiles[8] such as SPP, HID and HFP. In general, SPP (Serial Port Profile V12) is used for data transmission, HID (Human Interface Device V1.1) is used for user interface and HFP (Hands-Free Profile V1.7) is used for answering calls.

---

[6] http://www.ti.com/tool/cc256xqfnem
[7] http://www.ti.com/tool/tibluetoothstack-sdk
[8] https://www.bluetooth.com/specifications/adopted-specifications

We implemented an integrated software tester that developers can use on a PC using GUI. Developers can use the developed tester to select and activate a Bluetooth profile of their interest through GUI. Since there is a common set of commands, we separated commands into Common and profile-specific commands. Common commands include Inquiry, Display Inquiry List, Pair, PIN Code Response, Pass Key Response, User Confirmation Response, Change Simple Pairing Parameters, Set Discoverability Mode, Set Connectability Mode, Set Pairability Mode, Set Local Name, Set Class of Device, Get Local Address, Get Local Name, Get Class of Device, and Get Remote Name. Table 2 shows lists of commands for various Bluetooth profiles.

Table 2. Supported Commands for Bluetooth Profiles

Common	SPP Server & Client	HFP
Inquiry	Sniff Mode	Open HF Server
Display Inquiry List	Exit Sniff Mode	Close HF Server
Pair	Open	Manage Audio
PIN Code Response	Close	Answer Call
Pass Key Response	Read	Close
User Confirmation Response	Write	**HID Host/Device**
Set Discoverability Mode	Get Config Params	Control Request/Response
Set Connectability Mode	Set Config Params	Get Report Request/Response
Set Pairability Mode	Get Queue Params	Set Report Request/Response
Change Simple Pairing Parameters	Set Queue Params	Get Protocol Request/Response
Get Local Address	Loopback	Set Protocol Request/Response
Get Local Name	Display Raw Mode Data	Get Idle Request/Response
Set Local Name	Automatic Read Mode	Set Idle Request/Response
Get Class of Device	Set Baud Rate	Data Write
Set Class of Device	Send	Enable Debug
Get Remote Name	-	

SPP profile has two roles of server (Device A) and client (Device B), respectively. An SPP server is defined as a device, usually a PC, which in our case is the BeSDRP that initiates a connection. An SPP client is defined as a device that waits for the initiated connection from the server to accept the connection. In our case, we implemented an SPP client on an Android smartphone application. SPP server and client supports various commands (Sniff Mode, Exit Sniff Mode, Open, Close, Read, Write, Get Config Params, Set Config Params, Get Queue Params, Set Queue Params, Loopback, Display Raw Mode Data, Automatic Read Mode, Set Baud Rate, Send). HFP profile is used for a device that manages audio I/O of audio gateway (ex., mobile phone). HFP profile also provides remote control functions. HFP profile commands are shown in the third column of Table 2. HID profile has two roles of Host and Device, respectively. A HID host is defined as a Bluetooth device that uses or requests service of a Bluetooth HID device such as PC, mobile computer, game console, industrial machines, and data recording devices. A HID device is defined as a Bluetooth device that provides transmitting and receiving services to a Bluetooth HID host. Examples of HID device include keyboard, mouse, joystick and gamepads for user input mechanisms. Figure 4 shows GUI of the implemented Bluetooth profile tester for SPP server, SPP client, HFP, HID host and HID device where various Bluetooth processes are presented for testing on a PC.

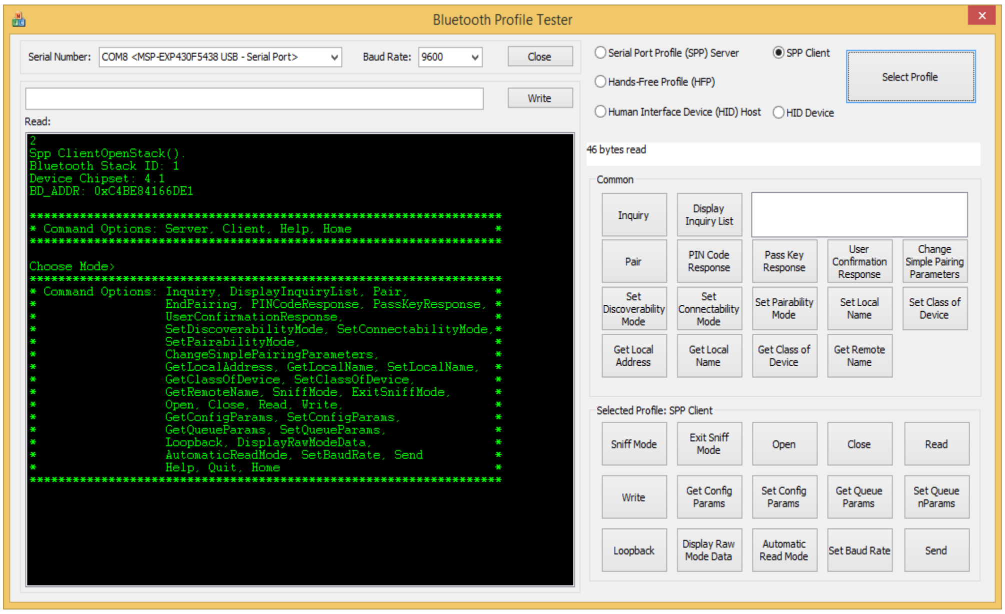

Figure 4. Bluetooth Profile Tester GUI

4. Smartphone-based Controller Application

4.1. Design of Smartphone-based Controller

We also developed an Android-based smartphone application to remotely control and access multi-standard radio data received from the BeSDRP as shown in Fig. 5.

Figure 5. GUI of the Implemented Android Application

Figure 5 shows GUI of the implemented Android application that displays available channel information from the BeSDRP. To communicate between the smartphone application and the BeSDRP, we used SPP to send and receive formatted control commands. When the smartphone controller application is launched, it enters into Bluetooth Pairing mode. Then, the smartphone application broadcasts Bluetooth advertising packets and attempts to establish a connection by responding to the request from the BeSDRP. After the connection is successfully established with the BeSDRP, the application becomes Service Connection state and starts Channel Scan. In this stage, the smartphone application requests Channel Scan to scan channel information on the BeSDRP. Once Channel Scan request is successfully acknowledged by the BeSDRP, the BeSDRP sends Channel Information of available and supported radio channels to the connected smartphone application. In the application side, received Channel Information packets are parsed and data/audio meta-information such as channel type (DAB/DAB+, DRM/DRM+, HD Radio, CDR), frequency, and audio type, is extracted. To receive compatible audio data from the BeSDRP, appropriate Channel Information is used to first configure the application setting. Then the BeSDRP sends audio data that corresponds to the configured setting, and the application can control (play, pause, and stop) radio channels from this point. Fig. 6 depicts the flow of channel scan and playback control process for connecting the BeSDRP to a smart device.

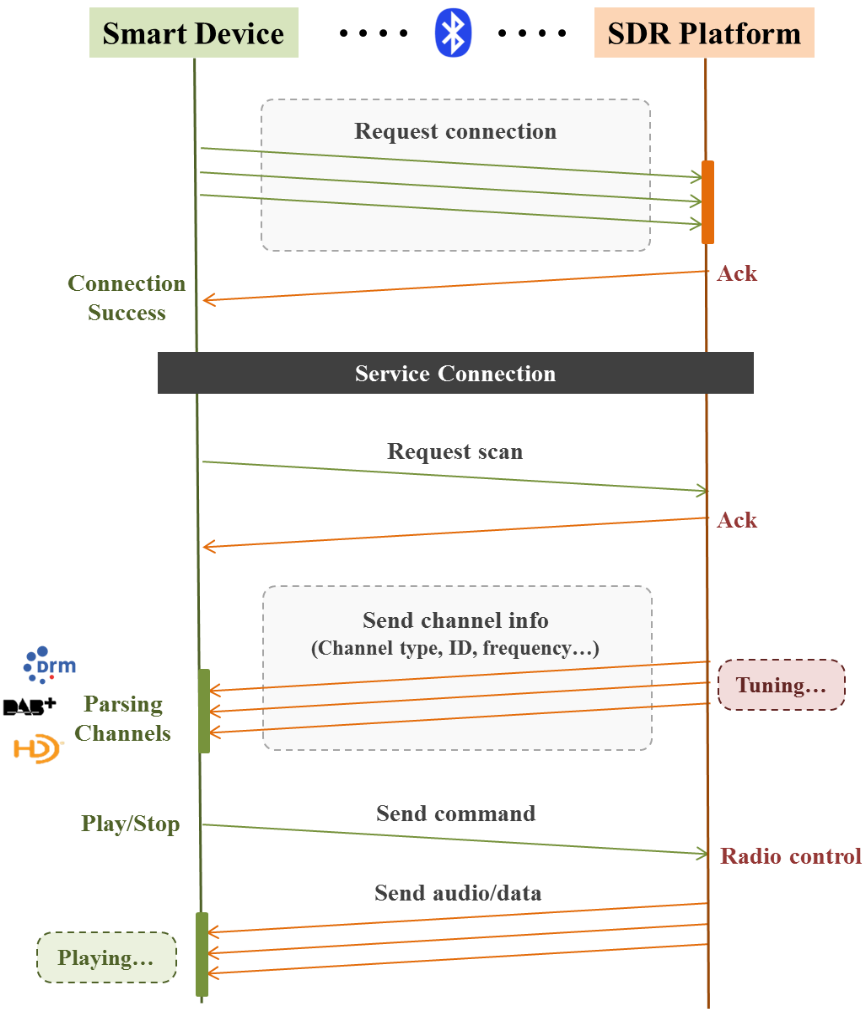

Figure 6. An Example of SDR Control Flow

4.2. Implementation of Control Functions

To support Bluetooth connection between the BeSDRP and smart devices, we implemented several control functions of digital radio in the implemented Android application. The developed Android application has 1) Bluetooth device discovery, 2) Bluetooth device connection, 3) SDR channel scanning, and 4) SDR control functions.

1) Bluetooth device discovery. The first-triggered function of the controller application discovers nearby Bluetooth devices. More specifically, the smartphone controller application implemented as an SPP client searches for an SPP server. We implemented Bluetooth device discovery to find nearby Bluetooth devices, so a user can easily select a compatible device to pair and select from a list of already paired/registered device. In the controller application, discovered Bluetooth devices are displayed in a list as shown in Figure 7. Previously paired and registered devices are shown in the upper area while unregistered devices are displayed on the bottom. We used Android Bluetooth Adapter[9] to retrieve a list of previously registered devices and scan for unregistered devices to retrieve its Media Access Control (MAC) addresses and device names.

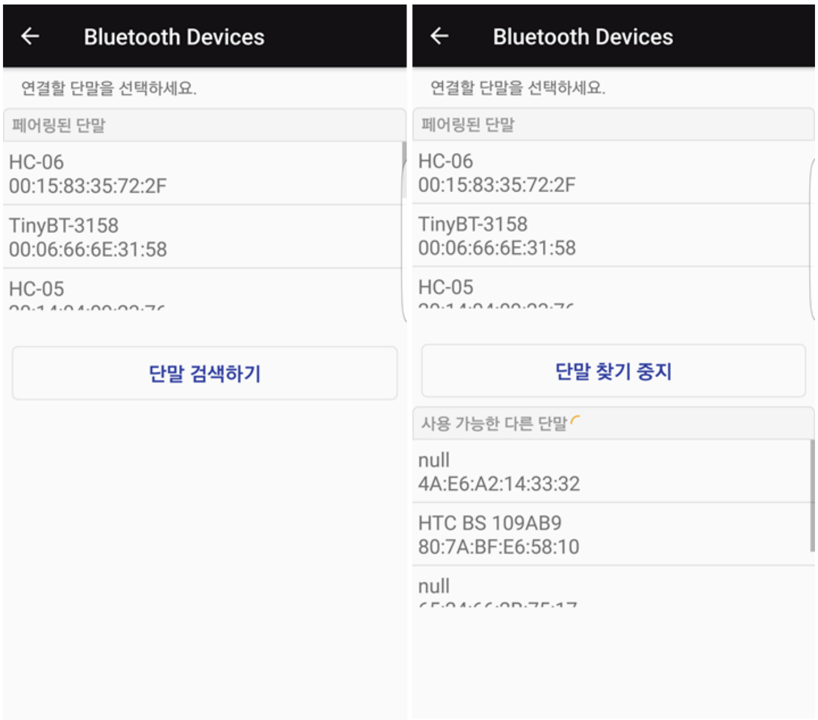

Figure 7. Bluetooth Device Discovery

2) Bluetooth device connection. When a user selects a device to connect from the list of discovered Bluetooth devices, connection process is started and the connection messages are progressively displayed at the bottom section of the application. If the discovered device, in our case the BeSDRP, is successfully connected with the smartphone application, then the smartphone application sends Service Connection request to the BeSDRP 5 times in a one second interval and waits for the response. The service connection request is a 10-byte packet composed of 1 byte to specify command flag, 1 byte to specify command type, 6 bytes for MAC address of a Bluetooth device to connect and 1 byte of NULL to mark the end of a command, respectively. If there is no response from the BeSDRP within 5 seconds, a message appears in the controller application informing the user for reconnection. Normally, the BeSDRP sends a response

[9] https://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html

Ack to Service Connection request of the controller application on time. In this case, the smartphone application stops sending request commands and starts to display monitoring messages as shown in Figure 8.

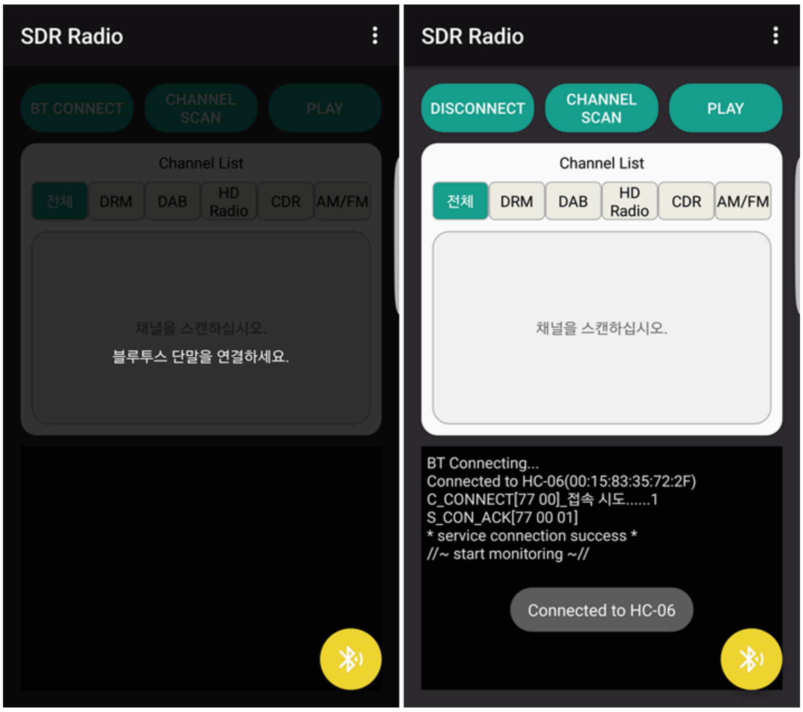

Figure 8. Bluetooth Device Connection

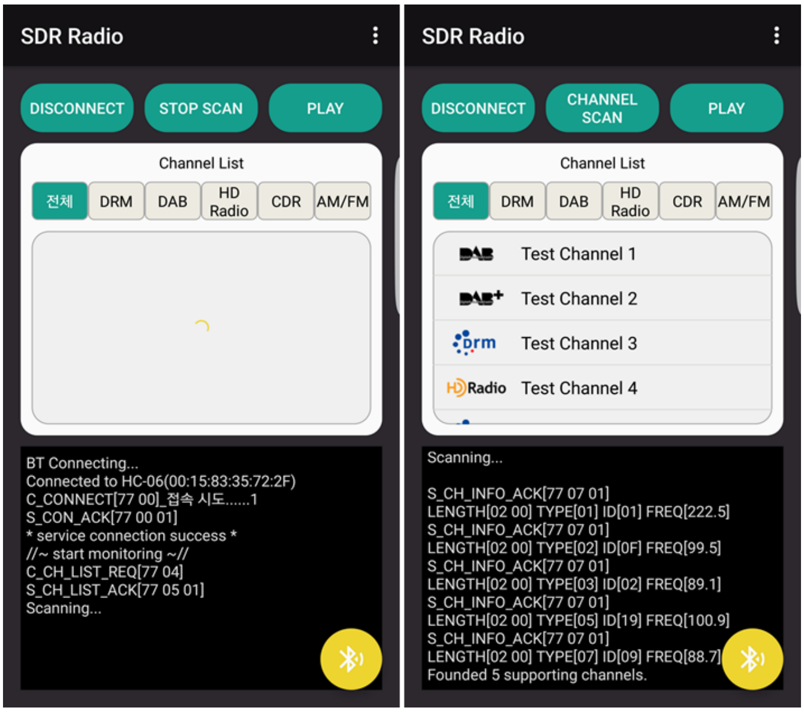

Figure 9. SDR Channel Scanning

3) SDR channel scanning. Once the connection is established, the smartphone application scans standard radio channels supported in the BeSDRP, parses and displays

the retrieved channel information as shown in Figure 9. When the BeSDRP receives channel scan request in Service Connection state, the BeSDRP sends supported and available standard channels information. The BeSDRP sends packets of channel information such as standard types (DAB/DAB+, DRM/DRM+, HD Radio, CDR, and FM/AM), channel name, frequency, service ID, and audio format using Bluetooth SPP. Figure 10 shows how the controller application supports multi-standards (DAB/DAB+, DRM/DRM+, HD Radio, AM/FM, CDR) when a standard in use is identified.

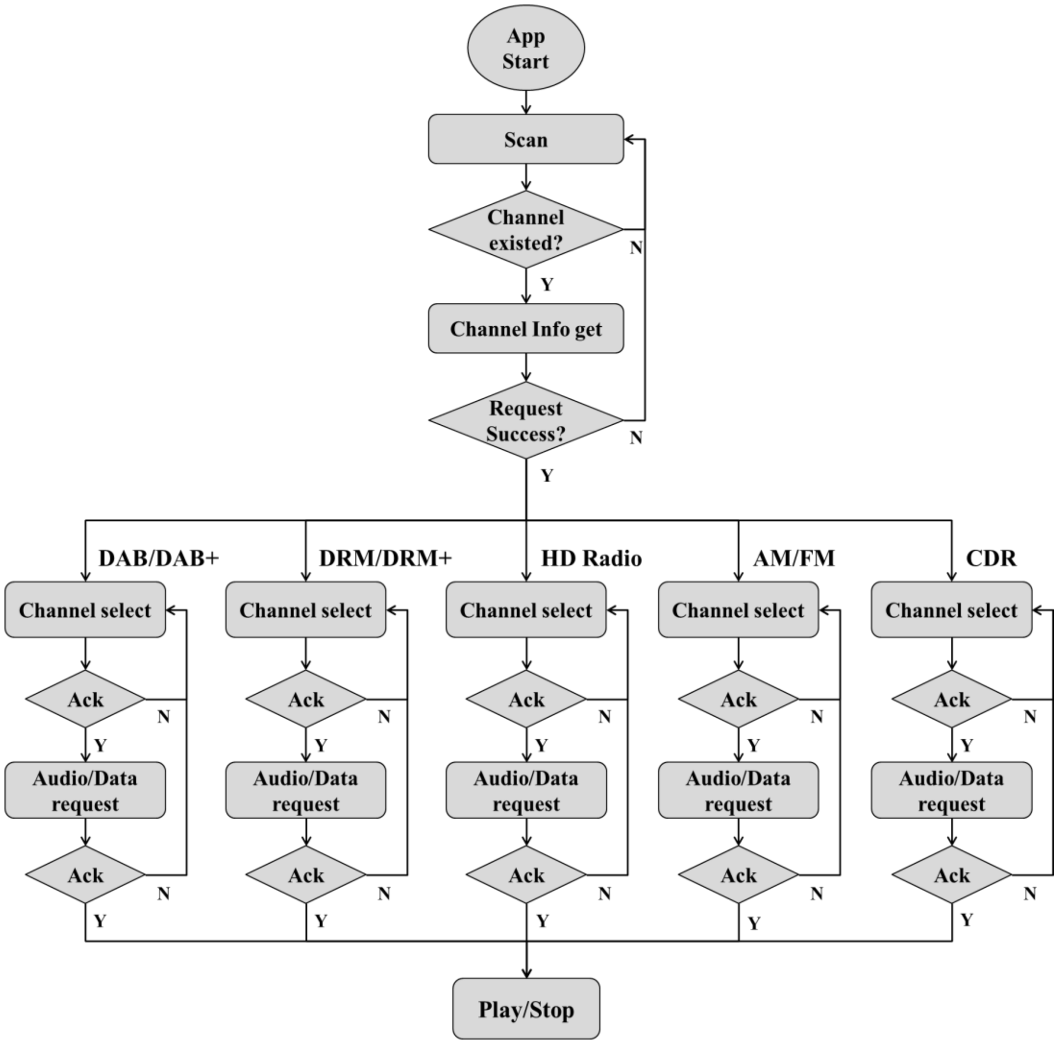

Figure 10. SDR Standard Selection

On the controller application side, after receiving a successful response to the channel scan, it waits for the channels information to arrive. The detailed channel information is included in a variable-sized packet consisting of 1 byte of command flag, 1 byte of command type, 1 byte of data length, 10 bytes of service ID and varying n bytes of channel ID, channel type, channel frequency and channel name. As the packets are arrived, the controller application parses the received packets and updates channel lists in the application view.

4) SDR control: The last function of the controller application is to control (play and stop) available SDR channels of the BeSDRP. Within the controller application, a user can use GUI to select from different standards (DAB/DAB+, DRM/DRM+, HD-Radio, AM/FM, CDR), play and stop the selected channel. A packet for this control command

includes previously played channel and newly selected channel to completely stop the previous channel and reconfigure for the new channel.

5. Application Scenarios

In this section, we discuss possible application scenarios by referencing previous work to demonstrate benefits of the BeSDRP. One classical use of Bluetooth and Bluetooth Low Energy is localization of a user carrying a Bluetooth device [3] or a Bluetooth device's whereabouts. Alletto et al. have developed an indoor location-aware system to locate a user in a beacon-installed museum for reducing burden of complex image processing [4]. Similar localization techniques can be used in a connected car [5][6] or smart car [7] environment to provide intelligent connected audio applications based on locations of users such as driver's seat or passengers' seat. Bluetooth-equipped cars or Bluetooth vehicles can be monitored and tracked for traffic [8][9] while its travel time statistics including travel trajectory are collected [10] using Bluetooth detectors on the highway. Similar concept is discussed in literature concerning vehicle-to-infrastructure [11] and vehicular social network [12]. This concept can be used with the BeSDRP on travel applications. For example, when a user travels different countries, a control command for default setting in that area can be advertised and received by the BeSDRP. This enables seamless transition to a region-specific digital radio service without direct user intervention. Within a car, Bluetooth is also used as an intra-vehicular wireless sensor network [13] and to transmit in-car information such as On-Board Diagnostics [14]. If we apply this concept of intra-vehicular network to the BeSDRP, information about built-in Bluetooth devices (screen, audio/video systems, input devices, and speakers) and Bluetooth devices of users in a car can be collected and managed for intelligent in-car infotainment systems [15]. Figure 11 illustrates an in-car infotainment system consisting of multiple Bluetooth devices using the BeSDRP.

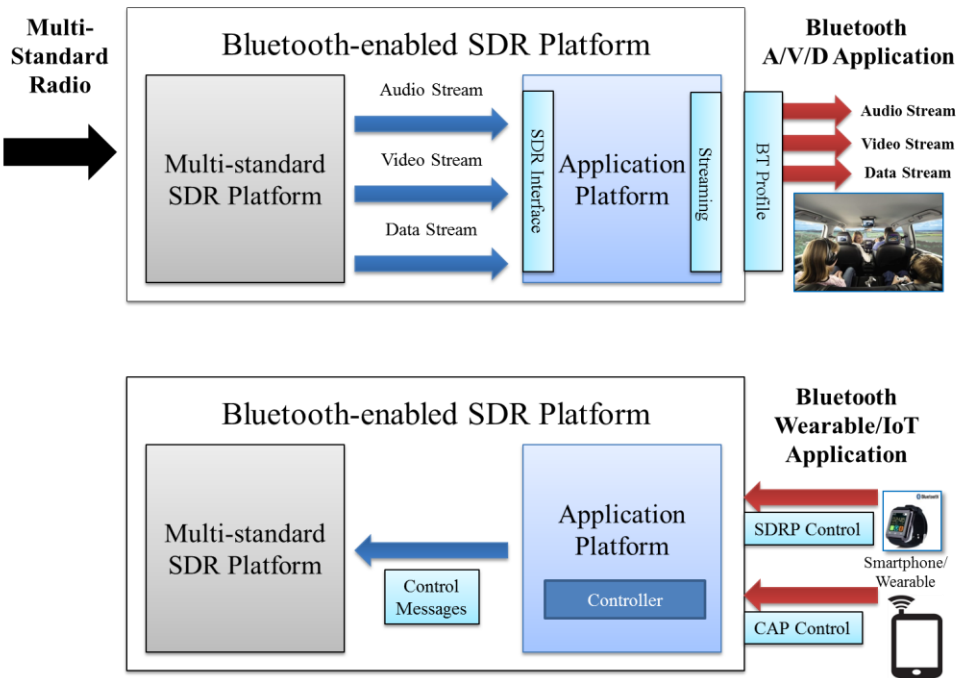

Figure 11. In-car Infotainment System based on the BeSDRP

6. Conclusions

In this paper, we introduced the Bluetooth-enabled SDR Platform (BeSDRP) for using multi-standard digital radio data for connected audio application based on Bluetooth connectivity. We presented design choices and overall architecture of the BeSDRP in Section 2. Two important aspects of the BeSDRP, the Bluetooth profile tester and the smartphone-based controller application are detailed in Section 3 and Section 4, respectively. In Section 5, we discussed previous work in terms of application scenarios and possible adaptation to the BeSDRP and highlighted an exemplar scenario in an in-car infotainment system. We believe that the BeSDRP will be widely useful in relevant domains of smart car, smart home and smart IoT by making multi-standard radio data as an available audio source for various applications. For future work, we are interested in developing and evaluating applications in smart car domain to further polish functionalities and tools provided by the BeSDRP.

Acknowledgments

This paper is a revised and expanded version of a paper entitled “Smart Convergence Platform for Software Defined Radio Application” [16] presented at the 3rd International Mega-Conference on Green and Smart Technology, Jeju Island, Korea, December 21-23, 2016. This research was supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Evaluation Institute of Industrial Technology (KEIT). [10063285, Development of RF SoC and Smart Convergence Platform for SDR (Software Defined Radio)]

References

[1] F. K. Jondral, “Software-Defined Radio: Basics and Evolution to Cognitive Radio”, EURASIP Journal on Wireless Communications and Networking., vol. 2005, no. 1, **(2005)**, pp. 275-283.
[2] U. Ramacher, “Software-Defined Radio Prospects for Multistandard Mobile Phones”, Computer., vol. 40, no. 10, **(2007)**, pp. 62-69.
[3] H.-S. Kim, “Design of Tour Guide System using Bluetooth 4.0 and WiFi Sensor Technology,” Journal of the Korea Academia-Industrial Cooperation Society., vol. 16, no. 10, **(2015)**, pp. 6888-6894.
[4] S. Alletto, R. Cucchiara, G. D. Fiore, Luca Mainetti, V. Mighali, L. Patrono, and G. Serra, “An Indoor Location-Aware System for an IoT-Based Smart Museum,” IEEE Internet of Things Journal., vol. 3, no. 2, **(2016)**, pp. 244-253.
[5] R. Coppola and M. Morisio, “Connected Car: Technologies, Issues, Future Trends”, ACM Computing Surveys, vol. 49, no. 3, **(2016)**, Article No. 46.
[6] M. Swan, “Connected Car: Quantified Self becomes Quantified Car,” Journal of Sensor Actuator Networks., vol. 4, no. 1, **(2015)**, pp. 2-29.
[7] S. C. Huang, B. H. Chen, S. K. Chou, J. N. Hwang and K. H. Lee, “Smart Car [Application Notes],” IEEE Computational Intelligence Magazine., vol. 11, no. 4, **(2016)**, pp. 46-58.
[8] M. R. Friesen and R. D. McLeod, “Bluetooth in Intelligent Transportation Systems: A Survey,” International Journal of Intelligent Transportation Systems Research., vol. 13, no. 3, **(2015)**, pp. 143-153.
[9] J. J. V. Díaz, A. B. R. González and M. R. Wilby, “Bluetooth Traffic Monitoring Systems for Travel Time Estimation on Freeways,” IEEE Transactions on Intelligent Transportation Systems., vol. 17, no. 1, **(2016)**, pp. 123–132.
[10] A. Bhaskar, M. Qu and E. Chung, “Bluetooth Vehicle Trajectory by Fusing Bluetooth and Loops: Motorway Travel Time Statistics,” IEEE Transactions on Intelligent Transportation Systems., vol. 16, no. 1, **(2015)**, pp. 113–122.
[11] E. Ndashimye, S. K. Ray, N. I. Sarkar and J. A. Gutiérrez, “Vehicle-to-Infrastructure Communication over Multi-Tier Heterogeneous Networks: A Survey,” Computer Networks., vol. 112, **(2017)**, pp. 144-166.
[12] A. M. Vegni and V. Loscrí, “A Survey on Vehicular Social Networks,” IEEE Communications Surveys & Tutorials., vol. 17, no. 4, **(2015)**, pp. 2397-2419.
[13] J.-r. Lin, T. Talty and O. K. Tonguz, “On the Potential of Bluetooth Low Energy Technology for Vehicular Applications,” IEEE Communications Magazine., vol. 53, no. 1, **(2015)**, pp. 267–275.
[14] V. C. Magana and M. Munoz-Organero, “Artemisa: A Personal Driving Assistant for Fuel Saving,” IEEE Transactions on Mobile Computing., vol. 15, no. 10, **(2016)**, pp. 2437–2450.

[15] H. Yoon and S.-H. Park, "Toward Bluetooth Connected In-Car Infotainment," Proceedings of the International Symposium on Innovation in Information Technology and Applications, Da Nang, Vietnam, **(2017)**, January 10-13.
[16] H. Yoon, J.-E. Lee, S.-B. Yu and S.-H. Park, "Smart Convergence Platform for Software Defined Radio Application," Proceedings of the 3rd International Mega-Conference on Green and Smart Technology, Jeju, Korea, **(2016)** December 21-23.

Authors

Hyoseok Yoon, he is a senior researcher at Contents Convergence Research Center, Korea Electronics Technology Institute (KETI). He received PhD and MS from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea in 2012 and 2007 respectively. He received his BS in computer science from Soongsil University in 2005. Prior to joining KETI, he was with KAIST Culture Technology Research Institute and GIST Culture Technology Institute. His research interests include human-computer interaction, ubiquitous computing, wearable UI/UX and personal informatics.

Ji-Eun Lee, she is an assistant researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. She received her MS and BS in Computer Science from Korea University of Technology and Education (KOREA TECH) in 2014 and 2016 respectively. Her research interests include wearable device, sensor-based wearable UI/UX, natural and gestural interface, android/iOS programming and Bluetooth applications.

Saet-Byeol Yu, she is a researcher at Contents Convergence Research Center, Korea Electronics Technology Institute (KETI). She received her BS and MS degrees in Multimedia from the Seoul Women's University, Korea, in 2010 and 2012, respectively. Formerly, she was an assistant researcher at Advanced Mobile Technology Research Center, Korea Electronics Technology Institute (KETI). Her research interests include digital/mobile/data broadcasting, Bluetooth-based applications and wearable UI/UX.

Se-Ho Park, he is a managerial researcher at the Contents Convergence Research Center, Korea Electronics Technology Institute (KETI), Seoul, Korea. He received his BS and MS degrees in Electrical Engineering from Kyungpook National University in 1998 and 2000, respectively. Before joining KETI in 2005, he was with I&C Technology and Samsung Electronics, where he worked in projects relevant to SoC (System on Chip) for digital broadcasting and wireless networks. His current research interests are in the areas of digital broadcasting, wireless mobile networks and embedded software developments.

