International Journal of Smart Home
Vol. 9, No. 11, (2015), pp. 133-144
http://dx.doi.org/10.14257/ijsh.2015.9.11.15

Design and Realization of Personal IoT Architecture Based on Mobile Gateway

Soonuk Seol[1], Yejin Shin[2] and Wooseong Kim[3*]

[1,2] *Korea University of Technology and Education, Cheonan, S. Korea.*
[3] *Gachon University, Seongnam, S.Korea*
[1]*suseol@koreatech.ac.kr*, [2]*yepp1252@koreatech.ac.kr*, [3]*wooseong@gachon.ac.kr*

Abstract

In IoT, connectivity for local and/or wide area is fundamental to collect sensed data from IoT field devices or to send control information to the devices. Up to now, most of IoT devices equip still personal area level wireless radio interfaces due to cost of radio modules, energy consumption, and subscription requirement of wireless cellular networks such as 3G, WiMAX and LTE. In order to collect data from such devices possibly moving in the Internet, a gateway or relay node that can transfers the data using wide area communication techniques is necessary. A smartphone which provide a tethering function can play a role of the gateway for personal IoT environment and it does not require additional subscription for personal IoT devices and provides location independent connectivity. In this paper, we design personal IoT architecture based on mobile gateway and realize it with two case studies, remote control of car navigation system and home automation examples[1].

Keywords: *IoT, M2M, smartphone, wireless sensor network, personal IoT*

1. Introduction

New era of Internet imagines connecting everything in the world and provides integrated services for machines or human beings. Various manufacturers from home appliances to mobile devices are participating on developing related technologies and standards. However, IoT world is not realized yet even though many of technologies and solutions are already proposed. Many different IoT standards or proprietary technologies tackle earlier deployment of the IoT in the world.

In order to connect the IoT devices to a remote IoT server that manage devices and provides user access, wired or wireless communication is necessary. Although some newly birthed devices equip wireless cellular transceivers, many of existing things, *e.g.*, thermostat, do not support cellular connection because the cellular module is costly and it needs subscription. Instead, many IoT devices use Bluetooth, Zigbee, Wi-Fi, *etc.* for wireless local access. Therefore, an Internet gateway near the IoT devices is needed to relay traffic from those devices to the Internet.

In addition to the wireless connectivity, IoT protocols are also diversified according to IoT device types. Application protocols between the IoT server and devices are used to query resources or data objects and put them for controlling devices. Since the resources or data are represented mostly in Web, legacy HTTP for the RESTful protocol is popularly considered, but other protocols such as CoAP, MQTT, *etc* are suggested for replacing the HTTP that is bit heavy and complicate for IoT device clients. In addition, there are many proprietary protocols based on transport protocols, TCP or UDP, for legacy IoT devices.

* Corresponding Author
[1] This paper is a revised and expanded version of a paper entitled "Smart phone assisted personal IoT service" presented at ISI 2015, Daejeon, Korea, September 21, 2015.

ISSN: 1975-4094 IJSH

Smartphones have been changing people's life by empowering pervasive computing which enables people to be always connected to the Internet by cellular connectivity and to interact with other machines or people. Thus smartphone can be a key device for opening IoT world. First, the smartphones with broadband wireless cellular communication services like LTE can play a role of the gateway for personal IoT network since IoT device vendors consider an interface to the smartphone because the smartphone is almost available to people and has very limited variants of middleware types; Android and IOS covers more than 80% of all deployed smartphones. Here note that all personal data from IoT environment are centralized in the personal smartphone. From this, interoperability between IoT devices and smartphones can be easily supported in terms of the wireless connectivity and protocols.

Accordingly, the smartphone can be an integrated hub that provides wireless connectivity and transient service from various IoT devices to Web-based IoT platforms. Also, it can operate individually a mobile platform for IoT without a fixed remote server for the IoT platform in the Clould for personal IoT environment, in which the mobile platform can be linked to friends or family' mobile platforms by social networking techniques using phone books of the smartphones like messenger services. From this, socially connected people can access those personal IoT devices, for example, to monitor health condition or to control house temperature via the smartphones.

In following sections, we first introduce background of IoT related technologies and issues. Then, we discuss problems of the current IoT architecture and propose our smartphone based IoT model [6]. As case studies, we introduce two case studies we developed based on our system model. First one is a smart phone assistant navigation system in which family members or friends help configuring a navigation system of elderly people of them in remote. Second one is a thermostat system in a house that is also controlled via the personal IoT platform using a smartphone.

2. Background

IoT integrates various technologies across layers from physical network connectivity to applications or services for specific vertical markets. For example, Figure 1 shows ETSI M2M functional architecture model. M2M devices are linked using local wireless technologies in M2M area networks. Gateway or device service capability (GSCL, DSCL) interoperates with Network Service Capability Layer (NSCL) that is a kind of IoT platform for various M2M applications or services.

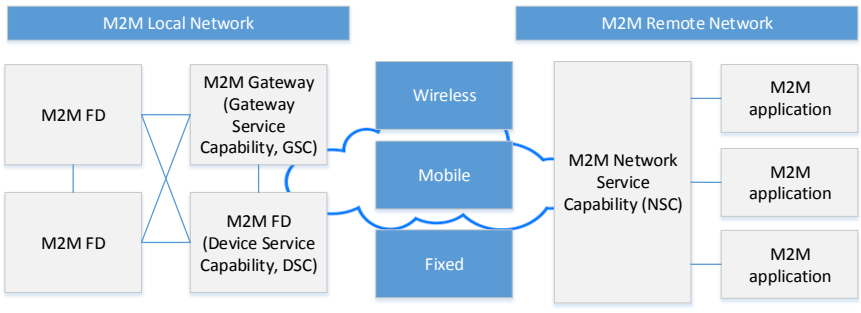

Figure 1. ETSI M2M Functional Architecture Overview

2.1. IoT M2M Area Connectivity

Wireless sensor networks (WSN) have been researched intensively during last decades, in which nodes deployed in field area monitor environment and report sensed information to a remote server using single or multi-hop communication. For this, various wireless connectivity technologies are developed for connecting those nearby field devices. Most

important virtue of the connectivity technique is low power consumption because the field devices equip only a few small batteries for power supply. In consequence, the IEEE 802.15 based wireless techniques such as Zigbee, Wireless HART, *etc* are popularly considered for the WSN. However, many recent gadgets such as smart watch, health monitoring gear, *etc* companying with smartphones have Bluetooth module inside for connection to the smartphones. Bluetooth provides robust connectivity to peer device and easy to use with simple configuration. Recently, the Bluetooth low energy (BLE) is proposed to improve power consumption for IoT field devices. From this, upcoming IoT home devices are becoming to use the Bluetooth. IEEE 802.11 based Wi-Fi is one of key technologies for home IoT solution. Recent home IoT gadgets such as smart bulb, smart power outlet, *etc* including home appliance can be controlled by smartphone users via home Wi-Fi gateway, access point (AP). Most of users have own or provided Wi-Fi AP inside home which allows connectivity to IoT field devices at home from a remote site. In addition, others like Z-wave, infrared, *etc* can be used for specific devices or purposes.

2.2. IoT Communication Protocol

IoT protocol inquires data or resources from IoT field devices or orders specific action to them with predefined syntax. HTTP is popularly considered which is completely well operating for RESTful services with XML or JAVA based Web contents; IoT data or resources can be also dealt by the REST type data without many changes or new features. Thus, basic HTTP methods for data acquisition or insertion can be inherited for the IoT protocol. However, HTTP designed mainly for wired networks can be burden in low data-rate wireless networks. In addition, implementation complexity of HTTP is inappropriate for low-cost IoT devices. Constrained Application Protocol (CoAP) has been proposed for IoT devices as a subset of the HTTP, which is a light-weight protocol for the clients [1]. CoAP is running on top of UDP instead of TCP and only support a subset of the HTTP methods and RESTful services, but interoperable with the HTTP. Also, CoAP reduces the overheads using binary data rather than ASCII and enhances security level with datagram Transport Layer Security (DTLS). However, the DTLS causes redundant traffic and bothers service aware traffic handling in network side. Message Queue Telemetry Transport (MQTT) developed by IBM is another lightweight protocol for IoT communications. MQTT is based on message queues managed by a MQTT server, in which publish and subscribe model meet better the IoT requirements rather than conventional request and response mode because the MQTT can avoid disastrous connection re-establishment procedure in intermittent connectivity environment of IoT and massive connectivity between users and devices. Once IoT devices publish own data to a remote server, users who subscribe the data can obtain the data, which requires delay tolerance so it is appropriate for non-realtime applications such as environment monitoring. MQTT is running on top of the TCP instead of the UDP. UDP based CoAP is responsive than MQTT, but CoAP can cause longer delay at lossy channel because upper-layer retransmission mechanism is not very reactive compared to the TCP [2, 3]. Extensible Messaging and Presence Protocol (XMPP) standardized by the IETF is for messaging services like Internet chatting. XMPP is an old fashion protocol, but scalable and extensible for IoT applications providing near real-time communication and robust delivery using TCP connection. There are several other IoT protocols such as Advanced Message Queuing Protocol (AMQP), Data Distribution Service for Real-Time Systems (DDS), Java Message Service (JMS), *etc*. Each IoT protocol has pros and cons depending on target applications and services.

2.3. IoT Service Platform

IoT platform provides common interfaces to different vertical services or applications from various IoT field devices. Such IoT devices are manufactured by different vendors so

they might not be interoperable. In order to manage the very different types of IoT devices, management layer is necessary. IoT platform first provides network connections to many IoT devices using the IoT protocols for data acquisition or IoT device control, and is able to manage M2M network topology (*e.g.*, chain, star or mesh). Second, IoT platform is capable of searching or discovery resources of the IoT field devices and managing data obtained from the IoT devices. Third, it deals with IoT device management like updating software of the IoT devices over the air and configuring device parameters. Recently, the IoT platform is extended to manage virtual IoT devices or resources that are just data objects without hardware. Fourth, IoT platform handles authentication and authorization from service layer, *i.e.*, users or devices. . Such upper layer security is important in self-organized IoT networks in which many different users or devices can share the IoT platform. Malicious users or devices try to access or modify others' data intentionally. Although IoT protocols provide security mechanism for data delivery with security layer like secure tunneling, upper layer security (*e.g.*, authentication, authorization) needs to be provided in the platform. Fifth, the IoT platform provides open APIs for third party developers or users. Up to now, many IoT platforms have been developed such as Xively [4] (formerly known as Cosm and Pachube), ThingSpeak, SmartThings, ioBridge, and so on. Pachube links users and IoT devices based on Web-platform, which deals with real-time data collected from globally distributed IoT devices. Open APIs of the Pachube enable users to access those stored IoT data anywhere and anytime. Xively is an open IoT platform that can be built by users themselves in order to create own IoT environment. A platform based on Xively can be instantiated as PaaS (Platform as a Service) in cloud environment for own IoT devices, in which the IoT devices can use own proprietary data format for reporting sensed data to the platform.

3. Problem Statement

Even though many IoT solutions have been emerged across layers, there are still many challenges to integrate them for a complete solution which delays earlier deployment of IoT networks. We introduce key issues to solve in this paper.

- **Wide-area Wireless Connection.** Most of IoT services are implemented by cloud and Web technologies, such as device search or directory service, big data analysis, *etc*. For this, Internet connection from IoT field devices to Cloud is necessary. However, direct Internet connection using Wide-area wireless connection such as 3G or LTE is costly because the modems of such access technologies are expensive and wide-area connection requires subscription. Currently, there is no service model of cellular network subscription for device groups. Also, there are field devices that are unclear about ownership. In addition, power consumption of the wide-area wireless communication is higher than local wireless connectivity, such as Wi-Fi, Bluetooth, ZigBee, *etc*.

- **Various Devices From Different Vendors.** IoT is very integrated concept of previous sensor networks that focus on connectivity, energy efficiency and data gathering, and recent IoT based vertical services such as ITS, medical area, logistics with cloud computing and big data technologies. Thus, there are many types of IoT field devices according to the IoT vertical services, *e.g.*, ITS, heath care, *etc*. In the previous sensor networks, many small and low-powered monitoring devices with various sensors were introduced. However, recent IoT devices include those with mid or high computing power and large battery such as smartphones, tablets, signages, vehicular AVN systems, wireless cameras, *etc*. Also, DIY devices using open hardware platforms, *e.g.*, arduino, Raspberry pie, iobridge, *etc* are used for personal IoT environment. Such various devices from different manufacturing delay integration among those devices since they

use different wireless connectivity technologies, network and application protocols even if there are several standards for them.

- **Various IoT Platforms.** Recently, many IoT platforms have been introduced for last several years. Service providers or individuals use one of those platforms according to their environment. Basically, interoperability between the IoT field devices and the platforms is a most important factor to choose the platform since those platforms provide very similar functionalities that were defined in IoT standards like ETSI M2M, OneM2M, *etc*. HTTP is almost supported in most of platforms, but MQTT or CoAP is also provided for light-weight communications between the field devices and the platforms. Even though such communication protocols are converged into limited number, resource representation is very different to the platforms, which implies the field devices should be adapted to each platform to send data correctly.

- **Old IoT Devices.** There are many field devices that have been developed before IoT standards. Also, there can be DIY devices that do not follow the standard. Those devices are difficult to communicate the IoT platforms without modifying them to use defined interfaces for the platforms. To say, suppose that a field device using proprietary protocols and Bluetooth are already installed behind wall, we have to get it out of the wall and update software for a certain IoT platform we want to use. It will be very lucky if there is the software for the field device. Otherwise, we have to develop by ourselves or change the devices to another that uses standard connectivity and protocols.

- **IoT Device Identification (ID).** The field devices have various identifications according to the machine types. If the IDs are unique in the world or country, it can be distinguished in the IoT platform naturally. However, IDs cannot be assigned to guarantee the uniqueness except some of ID types which are governed by international organizations or association for standard IDs, *e.g.*, auto ID, Wi-Fi alliance, *etc*. In addition, small manufacturers or individuals develop their gadgets for personal IoT services without any certificated IDs.

- **Self-organized Network.** Considering most of field devices belong to personal environment, cloud based IoT platform services can allow users to build own IoT environment for privacy or cost saving. Security or privacy guaranteed personal IoT environment should be easily established.

4. Proposed System Architecture

In this section, we propose smartphone based personal IoT concept which allows users to create personal IoT environment that consists of heterogeneous IoT devices and platforms. Also, it can support interoperability between proprietary devices and their communication protocols. Our system consists of a smartphone as a hub to connect with many near IoT devices, a remote server that is typically providing Web-based user interface, and IoT devices as shown in Figure 2. As a gateway to Internet, the smartphones equip local wireless connectivity modules such as Bluetooth, Zigbee, and Wi-Fi and wireless cellular interfaces for 3G, WiMAX or LTE. The remote server for IoT platform is located anywhere in Internet such as data centers, or even inside the smartphone itself. Recently, Web-based platform attracts much attention from IoT areas although challenges in web-platform such as long delay, asynchronous status, protocol overhead, are presence. However, Web can be easily deployed and migrated in user computing environment without much effort.

In our system, local wireless connectivity between a smartphone and devices can be adaptively used for type of IoT field devices (FD), FD A, B, C, and communication

distance as shown in Figure 2. Recent smartphones have only Bluetooth and Wi-Fi typically. Both connectivity technologies are different in communication range and power consumption. Many home appliances are now supporting Wi-Fi since they need to be controlled in any place within home area. In contrast, indoor beacons for localization use low powered Bluetooth Low Energy (BLE) and small gadgets like audio speakers, car AVN system, wearable devices, *etc* also use Bluetooth. Small dongle for Zigbee or other wireless technologies can be attached to the smartphone as an accessory which uses USB or micro SD interface for communication.

Most of FD device manufacturers have own proprietary protocols to link their gadgets with smartphones and they are implemented on top of mobile OSs like Android. Users can install the driver from App-store as an application, which reduces maintenance effort for their softwares of FDs by using well-built eco-system of the smartphones. Otherwise, users have to update the software manually by visiting Web-sites of the FD vendors. In addition, the FD manufactures can enhance competitiveness hiding their technologies for optimizing connectivity from others.

Our middleware as a thin convergence layer for personal IoT platform (PIP) between the application and existing mobile OS provides a common interface which define basic sets of programmable interfaces to exchange information from FDs to a smartphone and vice versa. They can be extended and included in mobile OSs. Those interfaces can be derived by basic attributes of the FDs with several methods for representing the information in smartphones or the remote platforms.

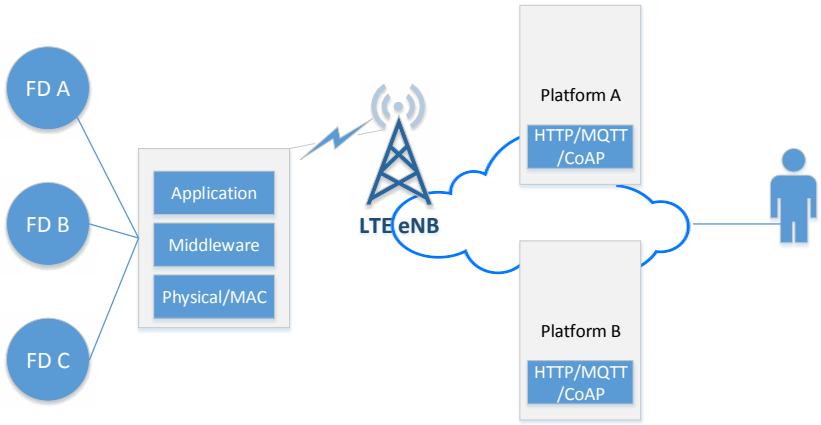

Figure 2. Mobile Gateway Based Personal IoT System Architecture

The remote platforms shown in Figure 2 can be provided by service providers or users themselves, which operates and maintains personal IoT devices registered by the users. Then, the servers manage those registered IoT devices; they check periodically status of IoT devices and collect information from them if need. There can be many different types of platforms for vertical services, *e.g.*, medical and car services, or for different service providers. If users are subscribed to the two different services with different IoT platforms, interfaces to both remote platforms are necessary; they can use different IoT protocols, resource formats, *etc*. Typically, each IoT platform defines own interfaces to attach the FDs to their platforms, which demands FD vendors to develop different types of application protocols for each IoT platform. In our architecture, instead, the platform vendors or open source community can provide client software for their platform. In order to implement light client software, smartphone middleware can provide necessary libraries for base protocols such as CoAP, MQTT, *etc*. as shown in Figure 3. Users who want to use a specific platform 'A' just download the client application from the service provider 'A' or app-store and install the app in their smartphones. For this, we define upper layer programmable interfaces in our convergence layer that are used by the client software of

platform providers. Basically, information handled in the lower layer interfaces for FDs and upper layer interfaces for the platforms are almost the same except some control information for data transmission.

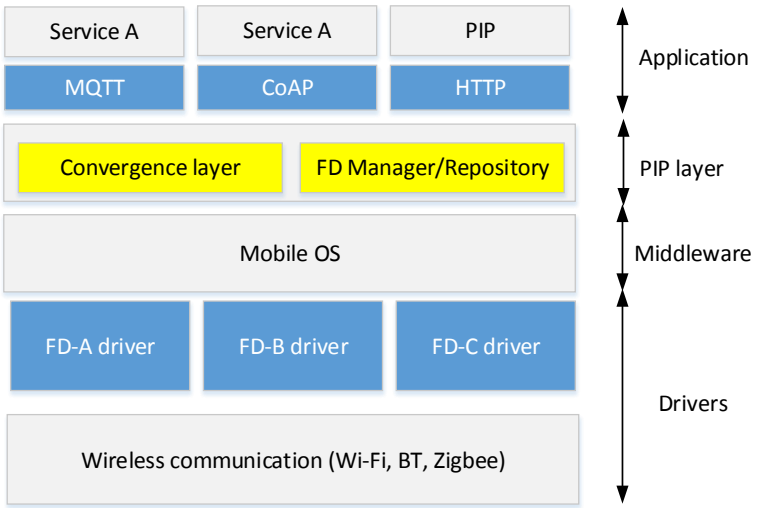

Figure 3. Mobile Platform Architecture for Personal IoT

In Figure 3, personal IoT platform (PIP) provides a mobile platform that is independently providing Web-based IoT service. Using the function, users can control directly FDs nearby without cellular connections and also allow the own smartphones as proxy servers to access users' FDs. For this, the PIP layer also provides IoT device management function like directory service and repository to store gathered data temporarily. Moreover, the PIP layer provides naming resolution service for IoT FDs. Self-made or old IoT devices without global identifiers are named by the PIP layer and resolved also by PIP layer when queries from outside platforms arrive. For example, own bulb that does not have globally assigned identifier can be assigned by locally unique identifier, *e.g.*, myphone.domain.net/bulb1, where prefix of the identifier is a global identifier of the smartphone and suffix is a local identifier. The local identifier can use a link layer address or identifier for a connection to the FD, *i.e.*, bulb1.

5. Case Studies

In this section, we apply our personal IoT architecture to real world applications. First one is on car navigation system where you can utilize driver's smartphone as a mobile gateway for the system and a remote user can monitor and control it through our platform. Another case study is on a thermostat system in a house that is also controlled via the personal IoT platform using a smartphone. Both case studies show that IoT connectivity is established through a smartphone and so the legacy devices become new IoT devices.

5.1. Automotive Navigation System

An automotive navigation system gives directions to the destination for a driver by using a GPS navigation device to acquire position data to locate the car on a road in the map database. Manipulating car navigation system while driving has been one of main reasons of car accidents. However, due to the lack of connectivity to the Internet, the navigation system could only be seen and configured by the driver. With our personal IoT platform, the navigation system is connected to the Internet and becomes an IoT field device.

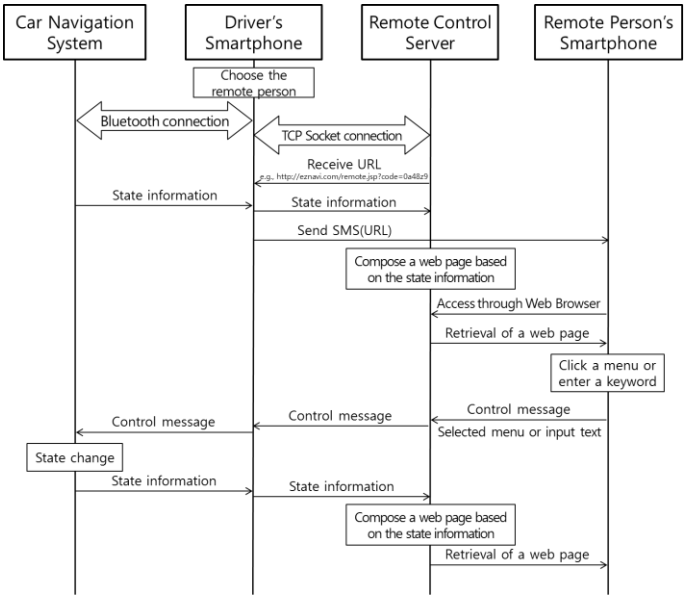

Figure 4. Remote Control of Automotive Navigation System

Figure 4 shows the remote control procedure of car navigation system based on our preliminary implementation [5]. It consists of a car navigation device, driver's smartphone, remote control server, and remote control client. In driver's smartphone, there is an application which maintains a Bluetooth connection with the car navigation system and a wireless cellular connection to the remote control server. After establishing all connections, the application on the driver's smartphone receives an access URL from the server and sends it to a remote person via SMS message. The URL is a one-time link that is valid within this session. Thus, only the designated remote person can control the device at a moment. The application also receives state information from car navigation system and forwards it to the remote control server. Then, the server composes a dynamic web page based on state information. The remote person can then access the web page by just clicking the link given in the SMS message. The selection or text input made in the web page is delivered all the way back to the car navigation system as a control message and it will change the state of the system. This state change will refresh the web page so that the remote person can make the subsequent controls. In this application, we employ text-based representation for remote users because graphical information requires more bandwidth and longer latency. Screenshots of remote clients are shown in Figure 5. One may extends the system to provide route maps as bitmap data.

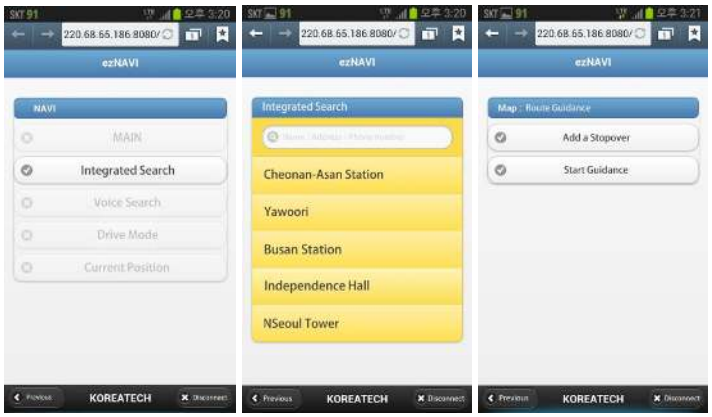

Figure 5. Screenshots of the Remote User

5.2. Home Automation - thermostat System in a House

Smart home is one of key application areas with IoT technology, but adapting all the existing devices with consistent manner is still a challenge [7, 8]. In order to show the feasibility of applying legacy embedded systems to our personal IoT platform, we have implemented a thermostat system by using Arduino board as shown in Figure 6. The current temperature and the current operating mode (Heating room, Hot water, Going out) is displayed. There are three buttons for changing the operating mode and temperature. A Bluetooth module is used for connectivity with a smartphone. Note that a remote user's screen (rightmost in Figure 6) contains all information shown in the actual boiler controller (middle in Figure 6) and also can make all required actions can be made such as temperature change, mode change, power on/off, *etc*.

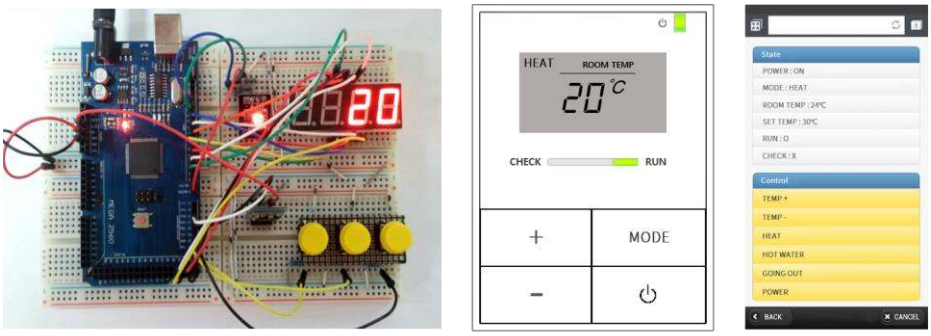

Figure 6. House Thermostat System

Table 1 shows some of basic programmable interfaces provided as a FD driver in mobile platform for exchanging information from the field device, *i.e.*, thermostat system device in this case, to a smartphone and vice versa.

Table 1. Application Programmable Interfaces Given by the Field Device (FD) Driver

method	Description
`int getMode( void )`	returns the current operating mode of the boiler controller. ROOM or WATER
`int getTemperature( void )`	returns the current temperature at the current operating mode; either room or water temperature.
`void modeChange ( int mode )`	changes the current operating mode to the specified mode.
`void tempUp( void )`	increases the current temperature by one degree at the current mode.
`void tempDown( void )`	decreases the current temperature by one degree at the current mode.
`void onChange(JSONObject info)`	receives all state information as a JSON object when any change happens

The first two methods, getMode() and getTemperature(), are used to get values from the IoT device and the following three methods, modeChange(), tempUp(), and tempDown(), are used to control the device. The last method is used to reactively handle events occurring in the device.

6. Conclusion

In this paper, we have designed personal IoT architecture which employs smartphones as mobile gateways for IoT devices to easily have connectivity to the Internet. Those devices do not need to equip with a cellular network interface like LTE because the smartphone support it. Our smartphone's mobile platform enables for different IoT platforms and for different field devices to interwork via convergence layer in our platform.

In order to analyze feasibility and effectiveness, we have conducted two case studies with actual implementations. In the case of car navigation system, it has shown that the navigation device can be remotely controlled through driver's smartphone. Beginner drivers, disabled people, or elders may benefit from such a system. This is a useful application scenario when an IoT field device has no wide-area wireless connection so the smartphone works as a mobile gateway. Another case study is on a thermostat system in a house that is also controlled via the personal IoT platform using smartphone. We have shown that the remote user's screen has a similar GUI design with that of car navigation system. This implies that our architecture can provide consistent user experience even in the coexistence of multi-vender IoT devices and multi-platform with different protocols.

References

[1] S. Raza, S. Hossein, H. Kasun and H. Ren," Thiemo Voigt, Lithe: Lightweight Secure CoAP for the Internet of Things", Sensors Journal, IEEE , vol.13, no.10, **(2013)**, pp. 3711-3720.
[2] S. Lee, H. Kim, D.-K. Hong, H. Ju, "Correlation Analysis of MQTT Loss and Delay According to QoS Level", International Conference on Information Networking (ICOIN), **(2013)**, pp. 714-717.
[3] T. Dinesh, X. Ma, A. Valera and H.-X. Tan, "Colin Keng-Yan Tan, Performance Evaluation of MQTT and CoAP via a Common Middleware", IEEE, Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), **(2014),** pp. 1-6.
[4] Xively, https://xively.com/
[5] Y. Shin and S. Seol, "Smartphone as a Remote Control Proxy in Automotive Navigation System", Contemporary Engineering Sciences, vol. 7, **(2014)**, pp. 683-689.
[6] W. Kim, Y. Shin and S. Seol, "Smart phone assisted personal IoT service", Advanced Science and Technology Letters, vol. 110, **(2015),** pp. 61-66.
[7] H. Liu, "Design and Realization of Smart Home Terminal Applications Based on IOT Technology", International Journal of Smart Home, vol. 9, no. 8, **(2015),** pp. 123-132.
[8] N. K. Lee, H. W. Lee, and R. Won, "Considerations for Web of Object Service Architecture on IoT Environment", International Journal of Smart Home, vol. 9, no. 1, **(2015)**, pp. 195-202.

Authors

Soonuk Seol, he received his B.S. degree from Korea University of Technology and Education (KOREATECH) in 1998, M.S and Ph.D. degrees in Information and Communication Engineering from KAIST in 2000 and 2004, respectively. He worked as a senior researcher at KT from 2004 to 2012. He is currently an assistant professor in School of Electrical, Electronics, and Communication Engineering at KOREATECH. His research interests include mobile Internet, IoT, QoS, and software testing.

Yejin Shin, she received the B.S. degree from Korea University of Technology and Education (KOREATECH) in 2014. She is in the M.S. degree in Information and Communication Engineering from KOREATECH.

Wooseong Kim, he received his Ph.D. degree from computer science, UCLA. Now he is an assistant professor of computer engineering department, Gachon University, South Korea. He used to work as a researcher of Samsung electronics, Hyundai motor, LG electronics, SK Hynix semiconductor, *etc*. He had standardization activity in several SDOs like 3GPP, TTA and ETSI. He is interested in multi-hop ad hoc networks, LTE and 5G wireless telecommunication system, wireless LAN, SDN/NFV, IoT protocols, *etc*.