Smart IT Convergence Framework in IoT Environment

Hyeyoung Kim and Cheong Ghil Kim

Abstract—**In this paper, we have presented a framework design method for content production using mobile terminals and sensors based on IoT environment. Also, this paper includes the prototype framework applying the proposed techniques.**

Keywords—IoT, framework, convergence, smart IT, prototype

I. Introduction

In order to make a service control platform in IoT environment, there must be a network module that enables connection to environment as well as these devices. And a controller is needed to establish connection between server and devices. However, it could be possible that the controllers may not be required, so they may create connections to server directly. In this paper, we introduce a prototype framework with the function of communication between server, sensor module, and mobile devices in IoT environment. The rest of the paper is organized as follows. In Section 2, we review the basic concept of IoT. Section 3 introduce a prototype implementation results. Section 4 concludes this paper.

II. Background

The term Internet of Things (IOT) was created by Kevin Ashton of MIT Auto-ID Sensor in 1999 [1,5], and presented the vision of connecting objects through RFID technology to the Internet. IoT technology is similar but defined differently depending on the organization dealing with the technology and its purpose. The common point is that all objects have global scale connectivity and provide intelligent services [2]. There are various categories of domestic and foreign IoT industry. However, some are accepted as an extension of ubiquitous or machine-to-machine (M2M). Key domestic industry examples include SKT's smart farm service, KT's smart home service, and LG U +'s IoT @ Home. SKT's smart farm is a service that manages farming through Smartphone by installing various sensor and W-CDMA-based terminals in the vinyl house of the farmhouse [3].

Haeyoung Kim

Hongik University
Korea
hykim@hongik.ac.kr

Cheong Ghil Kim (corresponding author)
Namseoul University
Korea
cgkim@nsu.ac.kr

As for the business models of KT and LG U Plus's, they are the services that allow the home terminals with outlet, lamp, gas valve, etc. In a sense those services may be regarded as an extension of the previous USN or M2M not like real IoT for global service [4].
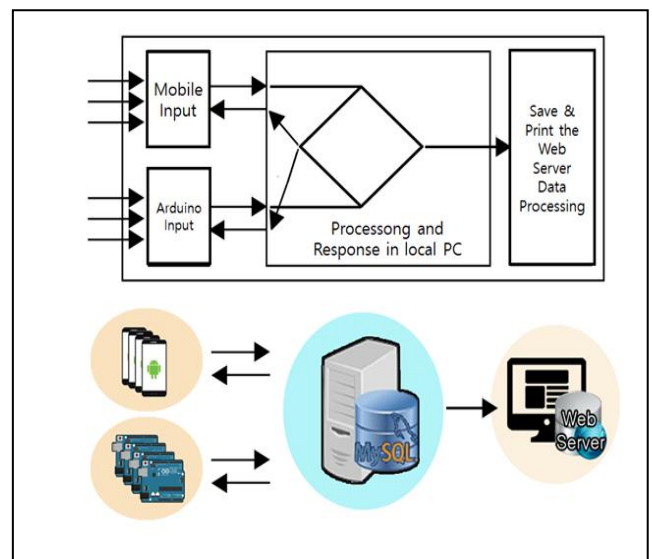
Figure 1.   Implementaton environment

III. Proposed Method

This paper propos a IT convergence framework using Lego type sensor module design techniques and Fig. 1 shows the implementation environment. The network environment applied in this work is shown in Fig. 1. For implementation C++ language and MySQL were used. All mobile sensors and Arduino devices communicate through local PCs. The examples of local PCs functions are as follows: confirm setting of completion of the device, data processing, backs up, and so on;

(1) Initialization step: When all sensors are operating normally, the initialization code is transmitted to the server. The server receives the code and sets the state of each sensor to the initialization phase. If it is determined that all the sensors (mobile and Arduino) have been set to the initialization step, the server sends the initialization completion code as a packet to all connected sensors collectively. (2) Data exchange phase: The sensor receiving initialization completion code assumes that the data is ready for exchange, and transmits the input value to the server. This includes various parameter values including on/off values. The server stores the received data in the web server. In addition, it analyzes the received data and can process them   appropriately   for   mobile   or   Arduino.   Through

Arduino they can communicate with the mobile devices and local server. All data are automatically saved, so that the interface for the user to manage the flow of data can be obtained through web server. (3) Error handling step: Arduino sensor is very vulnerable to change of external environment. Therefore, error handling according to the situation is an essential addition. In this paper, we propose a time dependent processing. When exchanging data with Arduino, update the Socket object's limited time to an arbitrary initial value. The value may decrease over time until the next data exchange occurs. When a delay occurs more than a certain time, the server defines that a problem occurs in the corresponding address. It is recommended to inform the user of the information of the relevant Arduino and reboot. In addition, the user can set whether or not to process data for all sensors operating in conjunction with the device. When the Arduino caused the interrupt is rebooted and reconnected, the server sends the initialization complete code to the corresponding Arduino, and all devices operate normally.

Figure 2. Schematic diagram

The design of server data is designed according to the following rules. They are applied collectively to the data linkage between mobile devices and Arduino. The basic length of the packet is designed with 9 bytes. They are divided into three 3-byte module, and consists of the command type, the corresponding value 1 for the command, and the corresponding value 2 for the command. For simulation, various types of packet are used. There are four types of Lego type modules applied and details are in Table 1. The identifier of each mobile terminal is based on the socket generated upon Accept. The sensor terminals composed of Arduinos are numbered according to the rules as shown in Table 2. They should be configured in a scalable and easy-to-manage manner. The following criteria were proposed and designed for the production of sensor devices to be applied to this platform. For Arduino instrument control, all Arduino modules set up on this platform are controlled by the central server from the moment the power is connected. Therefore, in order to efficiently control each module, all control commands are created by combining the three parts of device name, command, and parameter value. In addition, by defining a specific standard, it is convenient and easy to add even if

additional equipment is added later. Fig 2 shows the circuit design and function for each device.

TABLE I. PACKET FORMAT

	Items		
	3 Bytes	*3 Bytes*	*3 Bytes*
Elements	Instruction Type	Value 1 for Instruction Type	Value 1 for Instruction Type

TABLE II. LEGO TYPE MODULE SENSORS CONFIGURATION AND LABELS

Sensor number	*Sensor category*
xx1	{LED, Pressure, Vault}Sensor
Xx2	Vault Sensor
Xx3	RFID*4 Sensor
xx4	{Pressure, RFID} Sensor

IV. Conclusion

In this paper, we introduce a framework that can be effectively applied to contents production using various sensors of environment. IOT environment can be easily created by registering the same standard command to the server.

References

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami, "Internet of Things(IoT)": A visoin, architectural elements and future directins", Future Generation Computer Systems, Vol. 29, pp.1645-1660, 2013

[2] Jay Avardhana Gubbi, Raikums Buyya, Slaven Marusic, Mrimuthu Palanswami, "Internet of Things(IoT): A vision, architectural elements and future directins", Future generation computer systems, Sep. 2013, vol. 29, Issue. 7

[3] SKTSmart Farm : http://biztworld.co.kr/biz2013/solution/template_00000000000000000 153.jsp

[4] Yun Seung Ko,"Study of Policies of Major Countries on Internet of Things and Market Forecast,", KOrea Association for International Commerce and Information, vol. 16, no. 5, 2014

[5] Cheong Ghil Kim, "Web of Things: Future Internet Technologies and Applications", Journal of Platform Technology, Vol 1 (1) pp. 33-40, 2013