http://dx.doi.org/10.14400/JDPM.2014.12.1.333

An Implementation of Mobile Gateway Based on Android Smartphone

Lee Donggeon[*], **Lim Jae-Hyun**[**]
GCT Research, Inc[*]
Dept. of Computer Science & Engineering, Kongju National University[**]

안드로이드 스마트폰 기반의 모바일 게이트웨이 구현

이동건[*], 임재현[**]
㈜지씨티리서치[*]
공주대학교 컴퓨터공학부[**]

Abstract Zigbee is a wireless communication technology optimized for WSN (Wireless Sensor Network) environment. A WSN gateway is used for node control and data transmission. However, a fixed-type gateway can restrict the flexibility of the WSN environment. A smartphone-mounted high-performance processor and Android OS can be easily used in a mobile WSN gateway. In this paper, we proposed a mobile WSN gateway based on Android smartphones. In the proposed system, a Zigbee sensor module is connected with a smartphone via USB (Universal Serial Bus) port. We also implemented an Android application for the mobile WSN gateway.

Key Words : Zigbee, Wireless Sensor Network, mobile gateway, Android

요 약 Zigbee는 무선 센서 네트워크 환경에 최적화된 무선 통신 기술 중의 하나이다. 무선 센서 네트워크에서 게이트웨이는 센서 노드들을 제어하고 송수신된 데이터를 전송하는데 사용된다. 하지만 고정된 형태의 게이트웨이는 무선 센서 네트워크의 유연성을 제한한다. 고성능 프로세서가 탑재된 스마트폰과 안드로이드 OS를 이용하면 쉽게 스마트폰을 모바일 게이트웨이로 사용할 수 있다. 이에 본 논문에서는 안드로이드 스마트폰을 이용한 모바일 게이트웨이를 제안하였다. 제안한 시스템에서 Zigbee 통신 모듈은 USB 포트를 이용하여 스마트폰과 연결된다. 또한 모바일 게이트웨이를 위한 안드로이드 어플리케이션도 구현하였다.

주제어 : 지그비, 무선 센서 네트워크, 모바일 게이트웨이, 안드로이드

1. Introduction

WSN is a self-organized wireless network, which consists of sensor nodes and gateways. A sensor node has a processor, several sensors, and wireless network interfaces. A sensor node is in charge of monitoring the environment, such as its temperature, pressure, etc., and transmitting data cooperatively through a network.

* 공주대학교 연구년 사업에 의하여 연구되었음

Received 28 October 2013, Revised 28 November 2013
Accepted 20 January 2014
Corresponding Author: Lim Jae-Hyun(Kongju National University)
Email:defacto@kongju.ac.kr

ISSN: 1738-1916

A gateway usually has a better performance processor and multiple network interfaces to send/receive data and control the sensor nodes. The WSN is used in many applications, such as the military, robot control, home appliance control, health care, environment monitoring, automobiles, etc.

Zigbee is a wireless communication technology based on IEEE 802.15.4 LR-WPAN (Low-Rate Wireless Personal Area Network) standard. Zigbee defines specifications from low-level PHY behavior to high-level protocol. Zigbee is optimized for low-power consumption, low data rate, and short-distance wireless communication environment. Because of those advantages, Zigbee is widely used as a wireless network interface for sensor nodes.

In a WSN environment, a gateway is needed to control sensor nodes and receive data. However, a fixed type gateway can restrict the flexibility of WSN since a fixed device needs infrastructure [1-3]. A mobile type WSN gateway is not a required infrastructure. Also, a mobile WSN gateway is convenient to use, can support portability, and can adapt to various topology changes.

Recently, smartphones which possess mounted high-performance processors, various network interfaces, and USB interfaces have been widely distributed. Android OS is the most widely distributed smartphone OS. Android OS is based on Linux kernel and is optimized for the smartphone environment. Further, the Android kernel and framework are open to the public. Software developers can modify the Android kernel and framework to support external devices, to optimize code, or to extend the features of Android devices. Also software developers can develop an Android application using Java language and can get help from the Android developer's community. Thus, an Android smartphone can be easily used for mobile WSN gateway.

In this paper, we proposed a mobile WSN gateway based on Android smartphones. In the proposed system, a sensor module based on Zigbee is connected with a smartphone via a USB port. We implemented the proposed system in a commercial Android smartphone. Android kernel provides a USB-related driver to support the Zigbee module. The mobile WSN gateway application supports user interface, controls sensor nodes, and sends/receives data. To access the sensor module in the Android application, it supports the USB Serial Converter Driver and Serial Communication Library based on JNI (Java Native Interface).

2. Related Works

Related works designed and proposed various WSN systems and introduced gateways [1,2,3,4]. A previous work [1] proposed robot control system supporting wired and wireless network. A previous work [2] suggested sensor node control system using smartphone. A previous work [3] proposed web service-based monitoring system for WSN. They also used a fixed PC as a WSN gateway to control sensor nodes and receive data [1,2,3]. Some studies used embedded systems with Bluetooth interface for WSN gateway [4]. In the previous work [1,2,3,4], smartphone should connect to WSN gateway via WiFi or 3G network to access sensor node. PC or embedded system-based gateway requires infrastructure to support power and wired network. However, that requirement can restrict the flexibility and scalability of a WSN environment.

A previous work [5] proposed a mobile WSN gateway with smartphones based on Symbian OS. The proposed system used SDIO (Secure Digital Input Output) interface to connect a sensor module. However, almost all commercial smartphones have only one SDIO interface or none at all. Further, SDIO is used for external storage. Symbian OS-based smartphones are not globally distributed yet.

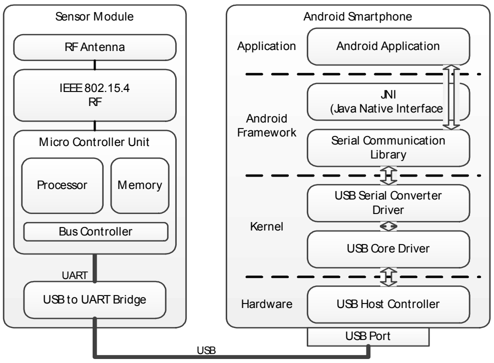
A past research proposed a mobile WSN gateway based on Android smartphones [6]. In that work, a sensor module was directly connected with smartphone CPU via UART (Universal Asynchronous Receiver Transmitter) pin. Such was hard to implement without HW change in the already distributed commercial smartphone. Table 1 presents a comparison of the mobile WSN gateway.

〈Table 1〉 Comparison of Difference of Mobile WSN Gateway

	Mobile WSN Gateway	Related Work [5]	Related Work [6]
Operating System	Android OS	Symbian OS	Android OS
Interface	USB	SDIO	UART
Development Environment	Android Framework 4.0.3	QT Framework	Android Framework 2.3
Language	JAVA/C	C/C++	JAVA/C

3. Mobile WSN Gateway

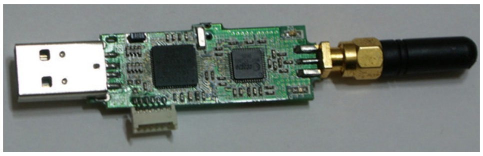
3.1 System Architecture



[Fig. 1] System Architecture of Mobile WSN Gateway

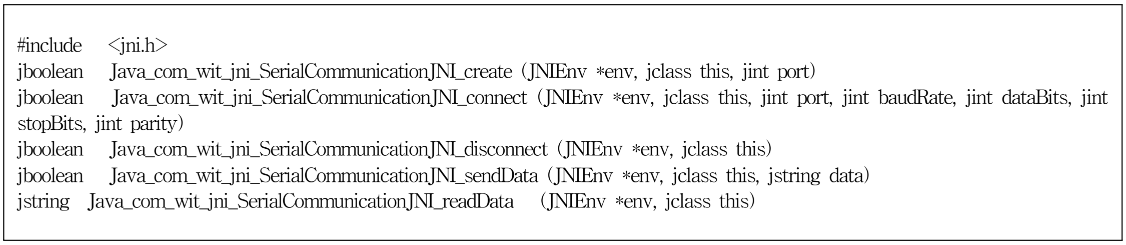
Figure 1 presents the structure of the proposed mobile WSN gateway. The sensor module consists of MCU (Micro Controller Unit), RF module, and USB to UART Bridge. MCU is a processor including internal memory to store data, and bus controller. RF module provides IEEE 802.15.4 MAC (Medium Access Control) and Zigbee protocol stack. The USB to UART Bridge provides UART and USB features to create an interface between sensor modules and Android smartphones. The sensor module connects with a smartphone's USB Host Controller via USB port. USB Core Driver is in charge of initializing and enumerating USB device and transmitting USB data. USB Serial Converter Driver is used for converting USB data to serial data and providing TTY interface. Since Android application is not able to access the device driver directly, Serial Communication Library provides API (Application Programming Interface) to open and close TTY device and read and write data. Android application provides UI (User Interface) to display the received data and status of sensor node. The Android application can read and write data and generate commands via JNI (Java Native Interface) by calling API, which is provided by the Serial Communication Library.

3.2 Hardware Module



[Fig. 2] Sensor Module

Figure 2 presents a picture of the sensor node that is used in the mobile WSN gateway. The sensor node used CC2420 made by Texas Instruments for IEEE 802.15.4 compliant RF transceiver, ATmega128L made by ATMEL for processing sensor data, and CP2102 made by Silicon Laboratories as USB to UART Bridge. The CC2420 is 2.4GHz RF transceiver with MAC support maximum 250kbps data rate. The CC2420 is suitable for both FFD (Full-Function Device) and RFD (Reduced-Function Device). The ATmega128L is 8-bit microcontroller support 128Kbytes of in-system programmable flash memory. The CC2420 is connected with ATmega128L via SPI (Serial Peripheral Interface)



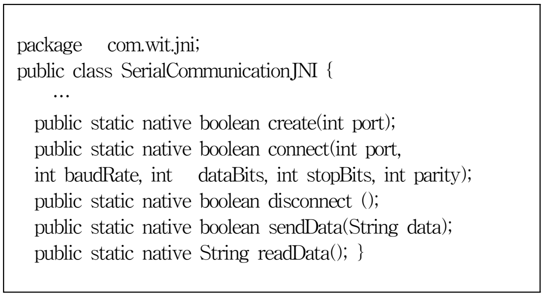
```
#include   <jni.h>
jboolean   Java_com_wit_jni_SerialCommunicationJNI_create (JNIEnv *env, jclass this, jint port)
jboolean   Java_com_wit_jni_SerialCommunicationJNI_connect (JNIEnv *env, jclass this, jint port, jint baudRate, jint dataBits, jint
stopBits, jint parity)
jboolean   Java_com_wit_jni_SerialCommunicationJNI_disconnect (JNIEnv *env, jclass this)
jboolean   Java_com_wit_jni_SerialCommunicationJNI_sendData (JNIEnv *env, jclass this, jstring data)
jstring  Java_com_wit_jni_SerialCommunicationJNI_readData   (JNIEnv *env, jclass this)
```

[Fig. 4] Function Prototype of the Serial communication Library

bus. The CP2102 is USB-to-UART bridge controller support USB 2.0 full-speed and full UART modem control.

3.3 Mobile Gateway Android Application

Figure 3 presents a part of the class definition of Android application. The SerialCommunicationJNI class is in charge of serial communication with sensor node. However, Java class can't access a serial device directly, so the class defines functions with a "native" keyword to access the device via JNI. When the native function is utilized in Android application (Dalvik virtual machine in Android framework), a function at the Serial Communication Library is tapped, which corresponds with the native function.

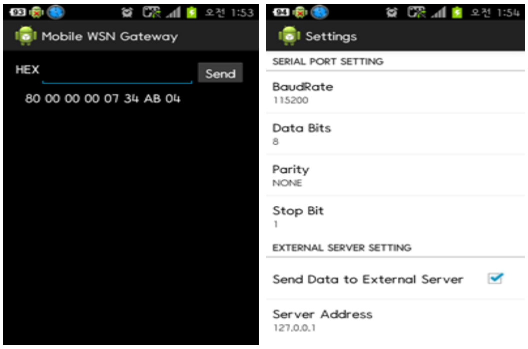


```
package   com.wit.jni;
public class SerialCommunicationJNI {
    …
  public static native boolean create(int port);
  public static native boolean connect(int port,
  int baudRate, int   dataBits, int stopBits, int parity);
  public static native boolean disconnect ();
  public static native boolean sendData(String data);
  public static native String readData(); }
```

[Fig. 3] Java Class for Access Sensor Module

As seen in Figure 3, the create() function opens a serial device from "/dev/ttyUSB0." The connect() function configures the serial port and connects the port. The disconnect() function disconnects and closes the port opened by the create() function. The sendData() and readData() functions send and receive data to/from the serial port.

Figure 4 presents APIs provided by the Serial Communication Library. The Serial Communication Library is written in C language and built with Android NDK (Native Development Kit). To determine the corresponding function by Dalvik virtual machine in Android framework correctly, the function-naming rule is very important. In case of the first function in Figure 4, "com_wit_jni" is the same as the package name seen in Figure 3. Also, "SerialCommunicationJNI" is a class name and "create" is a native function name. The first parameter "JNIEnv *env" is a pointer variable for JNI interface, which provides most of the JNI functions. The second parameter "jclass this" is a Java native type variable to access Java objects. The first and second parameters are mandatory. The other parameters are Java native type variables.



Mobile WSN Gateway

HEX

80 00 00 00 07 34 AB 04

Settings

SERIAL PORT SETTING

BaudRate
115200

Data Bits
8

Parity
NONE

Stop Bit
1

EXTERNAL SERVER SETTING

Send Data to External Server ☑

Server Address
127.0.0.1

(a) (b)

[Fig. 5] Sensor Module and Android Application

Figure 5-(a) and 5-(b) present the main UI and the configuration UI of the mobile WSN gateway application. The mobile WSN gateway is tested at Samsung Galaxy S II (SHW-M250L). The main screen display receives data from the sensor node. Users can select data format hexadecimal (HEX) or ASCII. Also, to send commands directly, users can use the text box. Figure 5-(b) presents the setup screen to configure the serial port and external server. Users can set the baud rate, data bits, parity, and stop bit. They can choose whether the gateway forwards to the external server or not, and set the address of the external server. If the "Send Data to External Server" option is checked, the Android application forwards the received data to the external server's specified server address via the network interface of the Android smartphone.

4. Conclusion

WSN is applied in a wide range of fields such as robot control, home network, automobile sensor, e-health, environment monitoring, etc. In a WSN environment, a gateway is used for data transmission and node control. In many WSN application studies, fixed type WSN gateway is considered. Because a fixed device needs infrastructure, such as a power line or wired network, the flexibility of WSN may be restricted. Android is the most widely distributed smartphone OS and is based on an open source. Software developers can modify the Android kernel and develop Android applications easily. Almost all commercial Android smartphones have a high-performance processor, an external USB port, and various network interfaces.

In this paper, we proposed a mobile WSN gateway based on Android smartphone. In the proposed system, a Zigbee sensor module is connected to a smartphone via USB port. To support the Zigbee module, the Android kernel provides the USB Core Driver and USB Serial Converter Driver. Also, we developed a mobile WSN gateway application. The application supports user interface to display received data and send commands. Because an Android application can't access a device directly, the application provides the Serial Communication Library. The application can access a sensor module through JNI. We implemented the proposed system in a commercial Android smartphone without any hardware change. Also the proposed system provides high scalability and flexibility by using a USB port and supporting mobility. The proposed system can adapt to almost any WSN application, such as home appliance control, robot control, etc. Further research would be required to apply Android-based mobile WSN gateway to various sensor network application environments.

ACKNOWLEDGMENTS

This work was researched by the Kongju National University Research year project.

REFERENCES

[1] Jaechang Shim, Fixed Mobile Convergence System for Robot Communication, Journal of Korean Institute of Information Technology, Vol. 10, No. 4, pp. 77-87, 2012.

[2] Yanjun Yao, PhoneCon: Voice-driven SmartPhone Controllable Wireless Sensor Networks, In Proceeding of IEEE 31st International Performance Computing and Communications Conference (IPCCC), pp. 440-447, 2012.

[3] N. Moreira, @Sensor - Mobile application to monitor a WSN, In Proceeding of 6th Iberian Conference on Information Systems and Technologies, pp. 1-6, 2011.

[4] Chiung-Hsing Chen, Intelligent Home Energy Conservation System Based On WSN, In Proceeding of International Conference on

Electrical, Electronics and Civil Engineering (ICEECE), pp. 166-169, 2011.

[5] Jinfeng Zhang, uSink: Smartphone-based moible sink for wireless sensor networks, In Proceeding of Consumer Communications and Networking Conference (CCNC), pp. 90-95, 2011.

[6] Siquan Hu, Enabling Zigbee Communication in Android Devices, In Proceeding of International Conference on Computer and Information Application (ICCIA), pp. 659-662, 2012.

이 동 건(Lee, Dong geon)

- Feb. 2007 : M.S. in Pusan National University
- Jan, 2010 ~ June, 2012:Worked for Wipro Technology
- July, 2012 ~ present: Working for GCT Research
- Interested Topic : WSN, Mobile Platform, Optimization
- E-Mail : KirkLee@gctsemi.com

임 재 현(Lim, Jae Hyun)

- Feb. 1988 : M.S. in Chung-Ang National University
- Aug. 1998 : Ph.D. in Chung-Ang National University
- Aug, 1998 ~ present: Working for KongJu National University as a professor
- Interested Topic : WSN, Context Prediction, Green Energy
- E-Mail : defacto@kongju.ac.kr