

센서 네트워크를 위한 임베디드 시스템

이종찬* · 박상준*

요 약

본 연구에서는 센서 네트워크 기반의 임베디드 시스템을 제안한다. 지그비 RF 기술과 임베디드 하드웨어 기술의 조합을 통하여, 주택 및 아파트 등에서 각 방별로 온도 및 습도를 취합하여 난방을 관리할 수 있다. 이를 위하여 리눅스 기반의 임베디드 시스템 디바이스를 제어하기 위한 JAVA API를 설계하였다. 임베디드 시스템 디바이스의 JAVA API를 통한 제어에 중점을 두어 구현하였다.

An Embedded System for Sensor Networks

Jong-Chan Lee* · Sang-Joon Park*

ABSTRACT

The object of this paper is to design a embedded system based on sensor networks. The proposed system can manage the heating by sensing and analyzing the temperature and humidity in apartment house and others. For implementing this system, we devide JAVA API into the platform-independent JAVA part and platform-dependent native part for the implementation of JAVA API to manage the linux based embedded system devices, and focus on the control of the embedded system devices using JAVA API.

Key words : JAVA API, Sensor Networks, Embedded System

접수일 : 2010년 5월 14일; 채택일 : 2010년 6월 22일

* 군산대학교 컴퓨터정보과학과

1. 서 론

본 논문에서 사용하는 센서 네트워크의 무선 통신 프로토콜은 IEEE 802.15.4 ZigBee이다. IEEE 802.15.4 ZigBee는 세 개의 무선 공용 주파수 대역에서 동작하는데 2.4GHz 대역에서 250kbps, 915MHz 대역에서 40kbps, 868MHz 대역에서 20kbps의 전송 대역폭을 갖는다. 채널수는 2.4GHz 대역에서 16채널, 915MHz 대역에서 10채널, 868MHz 대역의 1채널을 포함하여 모두 27채널을 할당할 수 있다[1~3].

임베디드 시스템은 범용 컴퓨팅 시스템보다 요구 사항이 까다롭다. 예를 들면, 휴대폰이 무겁고 수시로 충전해야 한다면 고유의 기능을 수행할 수 없고, 특정 계절에 자동차의 엔진제어가 불가능하다면 상품성이 전혀 없다. 임베디드 시스템의 주요한 특징을 정리하면 다음과 같다[4~6].

- 특정기능 수행 : 다양한 작업보다 특별한 작업을 수행하기 위해 설계된다.
- 강한 제약성 : 모바일 기기나 대형 시스템의 일부 기능을 수행하기 위해 소형이며, 전력 소모량이 작고, 실시간 반응을 요구한다.
- 제한된 시스템 자원 : 가격 및 휴대성과 연관되다보니 최적화된 상품을 위해 하드웨어와 소프트웨어를 병행하여 개발한다.
- 호환성 : 무수한 종류의 임베디드시스템에 최적화된 코드, 특히 어셈블리 언어의 사용으로 인하여 이식성이 매우 적다.
- 깔끔한 오류처리 : 임베디드 시스템은 대부분 비전문가가 사용하므로 가능한 사용자의 개입을 최소화해야 한다. 이를 위해 오류가 발생할 경우 깔끔한 처리가 요구된다.

현재 급부상 중인 임베디드 시스템 기반의 홈 오토메이션은 인터넷이나 휴대폰을 통해 난방을 제어하고, 유선 기반으로 난방을 관리하므로 기존의 단독 주택이나 아파트에서는 설치 및 관리의 문제점을 갖는다 [7, 8]. 따라서 임베디드 시스템을 지원하는 JAVA

API를 개발하면 기존 JAVA 개발자와 현재 임베디드 시스템을 개발하는 개발자들도 JAVA를 사용함으로써 코드의 재사용, 객체지향 개념의 시스템 개발들을 가능하게 한다.

2. 관련 연구

기존의 프로그래밍 언어들은 해당 플랫폼에 의존적인 바이너리 파일들로 컴파일 해야만, 해당 시스템에서 실행할 수 있다. 그러나 자바는 플랫폼에 관계없이 단 한번의 컴파일을 통해서 얻은 바이트 코드로 모든 환경에서 동일하게 실행 될 수 있다. 이러한 특징은 서비스 개발자와 제공자, 그리고 사용자에게 편리함과 효율성을 제공해준다. 서비스 개발자는 모든 플랫폼에 맞추어 프로그램을 작성하지 않아도 되며, 서비스 제공자는 동일한 프로그램을 서로 다른 플랫폼들에 제공하기 위해 여러 종류의 바이너리 파일들을 제공해야 할 필요가 없다. 또한 사용자는 임의의 프로그램을 설치하고자 할 때, 자신의 플랫폼에 알맞도록 프로그램을 찾아야 할 필요가 없으며, 자바로 작성된 프로그램은 무엇이든 필요할 때 네트워크를 통해 다운로드 받아서 사용할 수 있다[9, 10].

현재 급부상 중인 임베디드 시스템을 객체 기반 자바로 구현하는 것은 플랫폼 독립성, 네트워크 연결성, 안전성 등의 자바의 장점을 제공할 수 있다. 기존 임베디드 시스템 개발은 임베디드 리눅스 상에서 GTK나 C/C++을 이용한다. 그러므로 기술 구현의 난이도가 높을 뿐만 아니라, 코드의 재사용성 또한 떨어진다.

본 논문의 실행환경은 임베디드 리눅스 상의 자바 플랫폼 환경이다. JVM은 JamVM을 이용하였고, 자바 API 개발은 J2SDK 1.4버전을 이용하였다. 자바 API는 자바 플랫폼의 구성요소이므로, 다른 요소인 JVM과 밀접한 연관이 있다. 이들 사이의 연결자 역할은 JNI를 사용함으로써 가능하다. JNI를

통하여, API들은 JVM의 특성을 이용하고 원하는 값을 얻으며, C 라이브러리와 리눅스 상에서 제공하는 시스템 호출을 이용 할 수 있다.

3. 시스템 설계 및 구현

본 시스템은 컨트롤러와 센서, 보일러 밸브 개폐 장치로 구성된다. 각 방의 센서는 온도와 습도를 측정하여 컨트롤러에 지그비 통신으로 전송한다. 컨트롤러는 전송된 정보를 CLCD(Character LCD)에 디스플레이하고, 채감 온도로 환산하여 보일러 밸브 개폐여부를 판단하고, 지그비 통신을 통해 액츄에이터를 제어하여 보일러 밸브를 개폐한다. (그림 1)은 통신 시스템의 구조를 보인다.



(그림 1) 통신 시스템 구조

본 시스템을 기능상 세부분으로 구성된다. 우선 중앙의 제어장치에서 사용자 정의 온도와 각 방의 현재 실내 온도를 모니터링하는 기능과, 둘째로 각 방 온도 센서와 제어기 사이의 지그비 기반 통신 기능, 그리고 실내 온도와 사용자 정의 온도가 다를 경우 각방 제어기로 데이터 전송 후 각 방 온도 설정을 수행하는 자동 밸브 개폐 기능이다. 컨트롤러는 메인 컨트롤러와 서브 컨트롤러로 구성된다. 서브 컨트롤러는 각방의 상태만을 모니터링하고 제어하며, 메인 컨트롤러는 각방의 모니터링과 제어뿐만 아니라 다른 방의 상태 모니터링과

제어하는 기능을 담당한다.

4. 시스템 실험

(그림 2)에서 CLCDTest는 테스트용 클래스로 main 메소드만을 가진다. ①은 HAL 관련 자바 API인 AbstractCharacterLCDDevice 클래스를 상속받아 어플리케이션에 사용할 수 있는 네이티브 LCD_II 객체를 생성한다. LCD_II 클래스는 CLCD의 On/Off 및 초기화, 출력 관련 메소드를 제공한다. ②는 네이티브 LCD 객체를 인수로 하여 어플리케이션 영역에서 사용할 수 있는 CLCD 객체를 생성한다. ③은 CLCD 화면에 출력할 2줄의 콘텐츠를 생성한다. ④⑤⑥에서는 멤버그룹을 등록하고, 생성된 콘텐츠를 멤버그룹에 등록한다. ⑦은 CLCD에 출력하는 메소드이다.

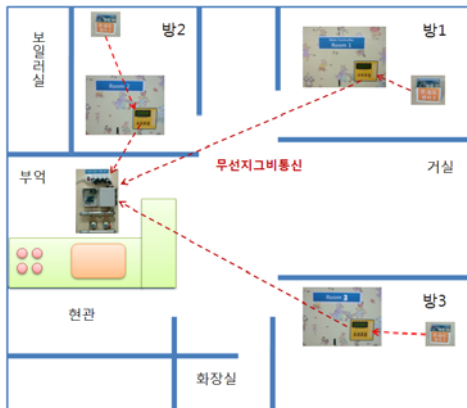
```
public class CLCDTest {
    public static void main(String[] args){
        AbstractCharacterLCDDevice nativeLCD = new
        LCD_II(); --- ①
        CharacterLCD clcd = new CharacterLCD(2,
        nativeLCD); --- ②
        IMemberContent line1Content = new
        LineContent("Line1 : Hello World"); --- ③
        IMemberContent line2Content = new
        LineContent("Line2 : USNLAB");
        CharacterLCDMemberGroup Line1 = new
        CharacterLCDMemberGroup(0, 16, ""); --- ④
        CharacterLCDMemberGroup Line2 = new
        CharacterLCDMemberGroup(1, 16, "");
        Line1.addMember(new CharacterLCDMember(0,
        line1Content)); --- ⑤
        Line2.addMember(new CharacterLCDMember(0,
        line2Content));
        clcd.addMemberGroup(Line1); --- ⑥
        clcd.addMemberGroup(Line2);
        clcd.displayAll(); --- ⑦
        try {Thread.sleep(1000);}
        catch (InterruptedException e) {
            e.printStackTrace();
            clcd.cursorOnOff(true);
        }
    }
}
```

(그림 2) CLCDTest.java

메인 컨트롤러의 개발환경은 호스트 컴퓨터와 타겟 컴퓨터로 나눌 수 있다. 호스트 컴퓨터는 RedHat FC4 운영체제에서 gcc와 J2SDK 1.4버전의 개발툴을 이용하였고 타겟 컴퓨터는 하이버스의 XHyper270-TKU를 이용하여 임베디드 시스템용 JAVA API를 개발하였다. 임베디드 시스템의 JVM은 JamVM을 이용하였고 JamVM은 GNU Classpath 자바 클래스 라이브러리를 이용하게 설계하였다[8]. 메인컨트롤러의 임베디드 시스템 구현을 위해 아래와 같은 환경을 이용하였다.

- 타겟 시스템 : XHyper270-TKU
- 커널 : Linux 2.6
- JVM : JamVM 1.4.5
- Classpath : GNU Classpath 0.9.3
- 호스트 컴퓨터 : IBM 호환 PC
- OS : Redhat FC4
- 자바 컴파일러 : J2SDK 1.4
- IDE : Eclipse 3.3

실험에 사용된 센서는 (주)하이버스의 H-mote와 센싱보드를 사용하였다. H-mote는 TinyOS를 사용하며, 개발 언어는 NesC이다. 보일러 밸브 개폐장치는 H-mote와 액츄에이터를 이용하여 개폐장치를 제어한다.



(그림 3) 시스템 실험 환경

개별 난방 제어 시스템을 실험하기 위해 컨트롤러 3개, 센서 3개, 액츄에이터 2개, 개폐장치 2개로 테스트베드를 구성하였다. (그림 3)은 시스템 실험 환경을 나타낸다.

실험 방법은 각 서버 컨트롤러에서 보일러 밸브 제어 장치를 구동 시키는 것과 센서로부터 센싱된 정보가 모니터링 되고 있는가, 메인 컨트롤러에서 서버 컨트롤러를 제어하고, 제어 신호가 보일러 밸브 제어 장치까지 정확하게 전달되는가를 확인하였다.

5. 결론 및 향후 과제

본 논문에서 구현한 자바 API 들은 네트워크 연결성, 플랫폼 독립성, 클래스 동적로딩, 안전성 등의 자바 장점을 임베디드 시스템에 제공함으로써 홈 네트워크, U-씨티, 헬스케어 구축에 큰 기여를 할 것으로 예측된다. 임베디드 시스템 기반으로 구현하였으므로 유사 센서 네트워크 기반의 시스템을 효과적으로 개발할 수 있는 기술을 축적할 수 있었다. 또한 개발된 지그비 통신 모듈을 센서 네트워크를 필요로 하는 모든 시스템에 쉽게 적용시킴으로서 센서 네트워크 구축의 활성화에 일익을 달성할 수 있다.

참고 문헌

[1] IEEE Std 802.15. 4-2003, IEEE Standard for Information technology-Telecommunication and Information exchange between systems-Local and metropolitan area networks-Specific requirements, Part 15.4 : Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks(LR-WPANs).

[2] David Culler, Deborah Estrin, Mani Srivastava, "Overview of Sensor Networks", IEEE Computer, pp. 41-49, Aug, 2004.

[3] Simon Bichler, 'Ubiquitous and Context Aware Computing : Overview and Systems,' 2nd Joint Advanced Summer School 2004 Course 3 : May, 2004.

[4] 김병호, "IEEE 802.15.4 MAC 프로토콜의 성능 평가 및 실험", 해양정보통신학회논문지, 제11권, 제1호, 2007.

[5] Atmel, AVR 8-bit RISC, <http://www.atmel.com/products/AVR/>.

[6] 남상엽, PXA270을 이용한 임베디드 시스템 구조 및 응용, 상학당, 2007.

[7] Bill Venners, Inside the Java Virtual Machine 2nd, McGraw-Hill, 1997.

[8] <http://jamvm.sourceforge.net/>.

[9] Java(TM) Native Interface : Programmer's Guide and Specification by Sheng Liang, Sun Microsystems.

[10] http://en.wikipedia.org/wiki/Heat_index.



이 종 찬

1994년 군산대학교
컴퓨터학과(이학사)
1996년 숭실대학교
컴퓨터학과(이학석사)
2000년 숭실대학교
컴퓨터학과(공학박사)

2000년~2005년 한국 전자통신연구원 선임연구원
2005년~현재 군산대학교 컴퓨터학과 부교수



박 상 준

1996년 동국대학교
전산학과(학사)
1998년 숭실대학교
컴퓨터학과(석사)
2002년 숭실대학교
컴퓨터학과(박사)

2004년~2007년 숭실대 정보미디어기술연구소
전임연구 교수
2007년~현재 군산대학교 컴퓨터정보공학 조교수