

데이터베이스를 활용한 센서 데이터 관리

권대곤¹, 최신행^{2*}

¹경남도립남해대학 컴퓨터정보계열, ²강원대학교 제어계측공학과

Sensor Data Management using Database

Dae-Gon Kweon¹ and Sin-Hyeong Choi^{2*}

¹Dept. of Computer Information, Gyeongnam Provincial Namhae College

²Dept. of Control & Instrumentation Engineering, Kangwon National University

요약 임베디드 분야에 대한 발전으로 임베디드 시스템을 이용한 각종 장비가 개발되고, 실제 생활에 많은 부분에서 활용되고 있다. 또한, 무선센서네트워크를 통해 각종 센서로부터 데이터를 수집할 수 있으며, 호스트를 거치지 않더라도 임베디드 시스템만으로도 수집된 데이터를 실시간으로 보여줄 수 있다. 본 논문에서는 센서 노드로부터 무선으로 전송되는 센싱 데이터 관리를 위해 임베디드 시스템에 임베디드 데이터베이스를 설치하여 게이트웨이 역할만을 하는 임베디드 시스템의 기능을 개선한 방안을 제시한다. 즉, 임베디드 시스템에 임베디드 데이터베이스를 설치한 다음 센싱 데이터를 저장 및 관리함으로써 센싱 데이터가 전송될 때마다 게이트웨이를 통한 호스트와 통신하는 전송 횟수를 줄일 수 있으며, 임베디드 시스템에서 필터링 프로그램을 수행하여 보다 정확하고 유효한 데이터만을 호스트에 전송함으로써 수집된 센싱 데이터에 기반한 분석결과와 신뢰성을 높일 수 있다.

Abstract All kinds of equipment which used an embedded system is developed, and these are used as to an actual life in developments regarding an embedded field in a lot of sections. Also, we can collect data from all kinds of sensors through wireless sensor networks, look by real time data collected could be brought if only through embedded system. In this paper we present a plan which improve the capabilities of embedded system only act as a gateway by installing embedded database in an embedded system for the sensing data management that was transmitted by radio from sensor nodes. In other words, by installing an embedded database to store and manage data by sensing data can be reduced the transmission frequency to communicate with a host and by performing the filtering program in embedded system and then by transmitting only valid data to the host can be increase the reliability of the analysis results based on data collected.

Key Words : Embedded System, Database, USN

1. 서론

30년 전만 해도 PC라는 개념이 아닌 터미널과 서버의 개념에서는 컴퓨터와 사용자의 비율이 1:N이었다. 하지만 1980년대부터 PC가 등장을 함으로써 그 비율은 1:1이 되었고, 현재에는 다수개의 컴퓨터가 한 명의 사용자를 위해 동작하는 N:1의 비율로 변화하고 있는 시대에 살고 있다. 하지만 이것은 이제 임베디드 시스템이 등장함으로써, 컴퓨터가 더욱 소형화되고 전문화되면서 사람이 컴퓨

터를 사용하고 있음을 인식하지 않는 유비쿼터스 세상을 만들게 되었다. 이제 컴퓨터와 사람의 비율이 ∞:1이 되는 세상을 맞게 된다[1].

임베디드 시스템이란 한마디로 '특정목적으로 구성된 마이크로프로세서 위에 소프트웨어를 내장하여 최적화시킨 시스템'이라고 할 수 있다[2]. 지금까지는 임베디드 시스템의 특징 상 호스트 PC에 비해 저사양, 저용량의 하드웨어로 구성되어 관리업무 보다는 중계업무의 역할을 수행함으로써 보조역할 및 단순한 응용프로그램 수행의 장

*교신저자 : 최신행(cshinh@kangwon.ac.krr)

접수일 09년 05월 04일

수정일 (1차 09년 06월 16일, 2차 09년 07월 15일)

게재확정일 09년 07월 22일

으로써 활용하는 수준이었다. 하지만 임베디드 시스템을 구성하는 마이크로프로세서의 처리속도 증가와 플래시메모리나 SDRAM의 용량증가로 보다 복잡한 프로그램 수행이 가능해졌고, 많은 데이터를 관리하기 위한 임베디드용 데이터베이스관리시스템도 등장함으로써, 임베디드 시스템은 보조서버로써의 역할뿐 아니라 데이터베이스관리시스템까지 탑재 가능한 수준까지 발전하였다. 이에 본 연구에서는 각종 센서로부터 무선으로 전송되는 센싱 데이터 관리를 위해 임베디드 시스템의 역할을 보다 확장하여 센싱 데이터 수집 및 필터링 등의 업무와 센싱 데이터 저장 등의 관리업무를 수행하도록 시스템을 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 임베디드 시스템과 임베디드용으로 개발된 데이터베이스관리시스템에 대해 살펴보고, 3장에서는 제안한 시스템 구조에 대해 설명하며, 4장에서는 제안한 시스템에 대한 구현을 보여주고, 마지막으로 결론 및 향후 연구 과제를 나타낸다.

2. 관련연구

2.1 임베디드 소프트웨어

임베디드 소프트웨어는 특정한 기능을 수행하기 위해 설계된 임베디드 시스템에 내장되는 소프트웨어를 말한다. 휴대폰, 디지털가전, 자동차, 국방 및 항공, 로봇에 이르기까지 다양한 장비에 내장되어 제품의 기능을 다양화하고 부가가치를 결정하는 핵심기술로 각광받고 있다. 즉, 임베디드 시스템에서 소프트웨어가 차지하는 비중이 점점 커지고 복잡해짐에 따라 임베디드 소프트웨어의 중요성도 이에 비례하여 커지고 있다[3,4,5].

2.2 임베디드 데이터베이스

임베디드 데이터베이스란 특정 애플리케이션의 한 부분으로 동작하는 데이터베이스 소프트웨어 컴포넌트라고 정의할 수 있다. 이와 같이 임베디드 데이터베이스는 애플리케이션이 필요할 때 수시로 이를 불러서 수행되므로 별도의 관리자가 관리활동 또는 인터럽트를 할 수 없는 경우가 많다. 따라서 이처럼 데이터베이스에 대한 관리활동이 지극히 제한되거나 특정한 용도로 정해져 있는 시스템에는 임베디드 데이터베이스를 활용하는 것이 좋다 [2,5,6]. 지금까지 개발된 임베디드 데이터베이스로는 Berkeley DB, GNU dbm, Embedded MySQL 등이 일반적으로 많이 사용된다[6-10].

2.2.1 Berkeley DB

Berkeley DB는 오픈소스기반 임베디드 데이터베이스이다. 기반 엔진은 C, C++로 구현되어 있으며, 자바 인터페이스도 지원한다. 많은 기능을 포함하고 있지만 매우 가벼우며, 속도도 빠른 편이다. 또한 경량 데이터베이스 이면서도 일반 DBMS에서 지원하는 대부분의 기능을 지원하고 있다.

2.2.2 GNU dbm

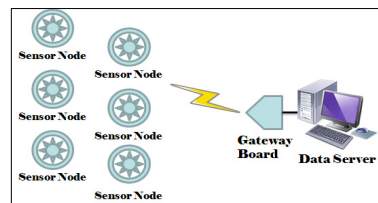
GNU dbm은 사용하기 쉽고 크기가 작으며 검색속도가 빨라 임베디드용으로 적합하고, 보통 gdbm으로 부른다. gdbm은 표준 UNIX dbm의 함수와 유사하게 작동하는 데이터베이스 함수모음이다. 이 함수들을 사용하여 데이터베이스 파일들을 만들거나 처리할 수 있다.

2.2.3 Embedded MySQL

잘 알려진 MySQL에는 임베디드 서버 라이브러리인 libmysqld가 포함되어 있는데, 이와 같은 임베디드 데이터베이스 라이브러리를 이용해, 최종 사용자가 기반 데이터베이스에 대해 알 필요도 없이 어플리케이션과 전자장비에 MySQL 데이터베이스 서버의 기능을 삽입할 수 있다.

2.3 기존의 센싱 데이터 관리

기존에는 센서노드로부터 전송되는 데이터를 수집하기 위해서는 그림 1과 같이 게이트웨이보드가 부착된 데이터 서버에 무선으로 직접 전달하는 형태이다.



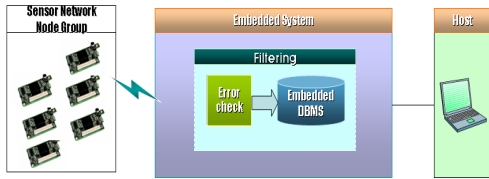
[그림 5] 기존의 센싱 데이터 관리

3. 시스템 설계

본 장에서는 본 논문에서 제안하는 임베디드 시스템 기반의 센싱 데이터 관리방안을 중심으로 시스템 구조를 살펴본다.

3.1 시스템 개요

앞서 살펴보았듯이 임베디드 시스템 특징과 대용량의 하드디스크 대신 적은 용량의 메모리로 인해 센서네트워크[11-14] 응용분야에서 단순한 게이트웨이 역할만을 수행하였다. 하지만 하드웨어 가격하락 및 관련 소프트웨어 기술발달로 인해 임베디드 시스템의 성능향상과 임베디드 시스템이 보다 다양하게 활용될 수 있다.



[그림 6] 임베디드 시스템을 이용한 센싱 데이터 관리

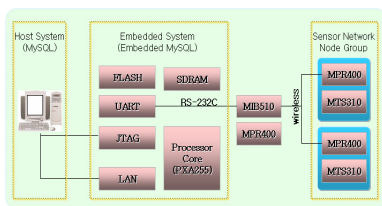
이에 본 연구에서는 그림 2와 같이 센서노드로부터 전송되는 각종 센서 데이터 수집 및 처리를 위해 임베디드 시스템의 역할을 확대하고 보다 정확한 데이터만을 호스트에 전송하고자 한다. 즉, 매순간 센서노드로부터 무선으로 전송되는 센싱 데이터를 임베디드 시스템에서 수집뿐만 아니라 필터링과 저장할 수 있다.

3.2 시스템 구조

전체 시스템 구조는 그림 2와 같이 구성된다.

크게 세부부분으로 구성되는 시스템은 호스트 시스템, 임베디드 시스템 그리고 센서 네트워크 노드 그룹으로 구성된다.

첫째, 호스트 시스템에는 리눅스 운영체제가 설치되어 있고, 각종 응용 프로그램 개발을 위해 필요한 gcc와 같은 C Compiler와 리눅스에서 사용되는 그래픽 라이브러리인 Qt/Embedded를 설치되어있다. 또한, 데이터베이스 서버로서 MySQL DBMS가 설치되어 중앙 데이터 저장소로서의 역할을 수행한다.



[그림 7] 시스템 구성도

둘째, 임베디드 시스템에는 호스트 시스템을 통해 개발된 통신 모듈을 포함한 센싱 데이터 수집 및 필터링 프

로그래밍과 임베디드 데이터베이스인 embedded MySQL이 설치되어 있다.

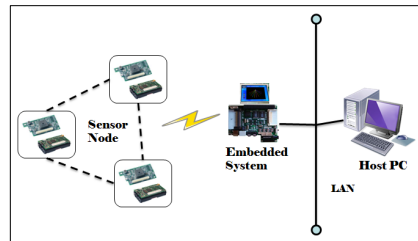
셋째, 센서 네트워크 노드 그룹은 온도, 습도 등의 센서정보를 수집하는 센서노드와 센싱 데이터를 임베디드 시스템에 무선으로 전송하는 노드로 구성된다.

4. 실험 및 고찰

본 장에서는 임베디드 시스템의 기능을 확대한 센싱 데이터 관리 시스템의 구현 환경과 구현과정에 대하여 서술하고자 한다.

4.1 구현환경

임베디드 시스템에서의 센싱 데이터 관리 시스템 구현을 위한 개발 환경은 호스트 시스템, 임베디드 시스템, 센서 노드 그룹으로 나눌 수 있다.



[그림 8] 실험 환경

본 논문에서 제안하고 설계 구현한 시스템을 실험하기 위하여 그림 4와 같은 실험환경으로 구성한다. 이에 대한 자세한 내용은 다음과 같다.

4.1.1 호스트 시스템

호스트 시스템에는 개발 환경 구축을 위해 우선 RedHat Linux 9.0을 운영체제로 설치하고, 센싱 데이터 수집 및 필터링을 위한 프로그램 작성을 위해 GNU gcc를 컴파일러로 이용한다. 또한, 임베디드 시스템에 설치할 응용 프로그램은 Qt-Embedded-free-3.3.3을 이용하여 프로그래밍 한다.

[표 1] 호스트 시스템

System	Pentium4(Memory 512)
OS	RedHat Linux 9.0
Compiler	X86용 GNU gcc
Database	MySQL
GUI Library	Qt-embedded-free-3.3.3

즉, 응용 프로그램이 임베디드 시스템의 TFT/LCD 상에서 실행되기 위해 GUI 부분은 Trolltech사의 임베디드 Qt 라이브러리를 사용하여 구현하였다. 이렇게 C언어로는 통신 및 필터링을 위한 주요 모듈을, Qt로는 GUI 구현을 위해 사용함으로써, 차후 GTK+와 같은 리눅스 혹은 윈도우 기반의 다른 GUI 라이브러리를 사용할 경우 이 부분만 수정하여 이용할 수 있다.

4.1.2 임베디드 시스템

임베디드 시스템은 하이버스의 X-Hyper270 TKU를 사용했다. 이 시스템의 주요사양으로는 FWPXA 270-520Mhz CPU가 장착되어 있으며, 128Mbyte의 램과 32Mbyte의 플래시메모리, 64Mbyte의 낸드 플래시메모리를 제공하고 있으며, 6.4인치 TFT LCD가 장착되어 있다. 임베디드 리눅스 2.4를 운영체제로 사용한다.

[표 2] 임베디드 시스템 사양

CPU		FWPXA270 (520MHz)
Memory	SDRAM	128MB
	Falsh ROM	32MB
	Nand Falsh	64MB
Display		6.4" TFT LCD
Interface		JTAG I/F UART I/F for Debug 1EA Full Function UART 4EA PCMCIA Card I/F USB Client 1 port
Ethernet		10Base-T 1Port 10/100Base-T 2Port

본 연구에서는 임베디드 시스템이 센서 노드 그룹으로부터 전송되는 센싱 데이터 수집과 필터링 및 관리역할을 하므로 임베디드 시스템에 설치되는 프로그램은 크게 세 부분으로 나누어진다.

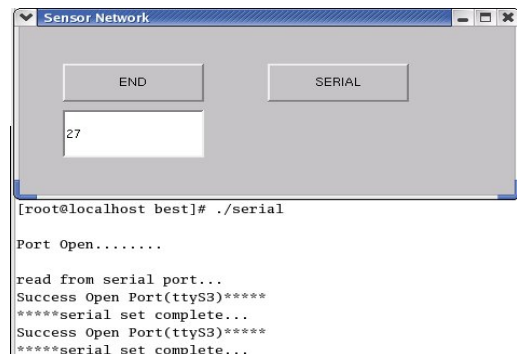
임베디드 시스템의 프로그램이 실행되면, 센서 노드로부터 전송되는 센싱 데이터는 임베디드 시스템과 시리얼 케이블로 연결되어있는 게이트웨이 노드인 MIB510을 통해 전달된다. 이들 데이터를 시리얼 통신 프로그램을 통해 수집한다. 표 3은 센싱 데이터의 패킷구조를 나타낸다.

[표 3] 센싱 데이터 패킷구조

항목	크기(byte)
Destination Addr	2
handlerID	1
groupID	1
msg length	1
source Addr	2
Counter	2
Channel	2
Data	20

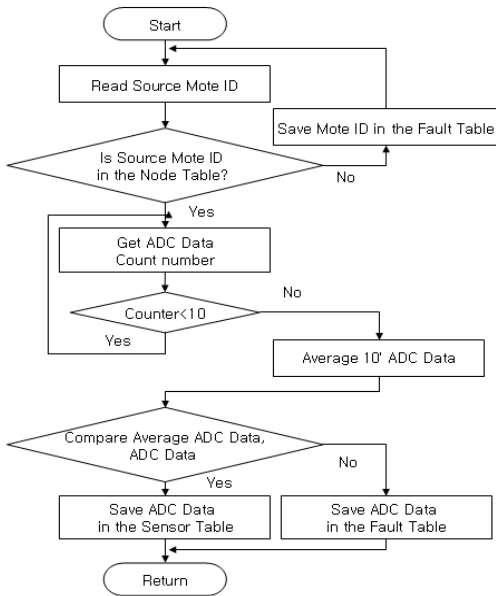
다음으로 필터링 프로그램에서는 수집된 센싱 데이터가 어느 노드로부터 전송되었는지를 파악하고, Data 영역에 있는 정보들의 전후 값을 비교하여 크게 차이가 나는 데이터는 문제가 있던지 신뢰할 수 없는 데이터로 처리하여 별도로 저장한다.

임베디드 시스템에서의 센싱 데이터 관리를 위해 본 연구에서는 임베디드 데이터베이스 중의 하나인 Embedded MySQL을 설치하여 임베디드 시스템에 데이터베이스를 구현하였다. 즉, 임베디드 시스템에 임베디드 솔루션을 위해 제공하는 embedded DB server library인 libmysqld를 설치하고, 센싱 데이터 관리프로그램에서 DB 서버를 시작할 수 있도록 필요한 모듈을 삽입하여 embedded server library에 링크한다.



[그림 9] TFT/LCD 상에서의 센싱 데이터 값

그림 5는 센서노드로부터 수집된 센싱 데이터를 시리얼 포트에 연결된 게이트웨이 보드인 MIB510으로부터 전송받아 임베디드 시스템의 TFT/LCD에 디스플레이해주는 화면이다.



[그림 10] 데이터 처리 알고리즘

이렇게 전송된 센싱 데이터는 그림 6의 처리 알고리즘에 의해 특정시점을 기준으로 10개씩 데이터 값을 비교하여 평균값에 비해 많은 차이가 날 경우에는 별도의 테이블에 저장함으로써 보다 정확한 데이터만을 센싱 데이터 테이블에 저장할 수 있다. 그림 7은 임베디드 시스템 상에서 Qt로 구현한 센싱 데이터 관리 프로그램의 일부 부분이다.

이와 같이 C/C++언어로는 시리얼 통신 프로그램 및 센싱 데이터의 필터링 프로그램 작성에 이용하고, 임베디드 시스템의 TFT/LCD에 보여주기 위한 GUI부분은 Trolltech사의 임베디드 Qt라이브러리를 사용하여 구현하였다.

본 연구에서 제안하고 구현한 임베디드 시스템을 이용한 센싱 데이터 관리방안을 사용함으로써 얻을 수 있는 이점은 다음과 같이 정리할 수 있다.

첫째, 센서노드로부터 전송된 센싱 데이터를 수집하기 위하여 임베디드 시스템을 사용함으로써 1차적인 필터링을 통해 데이터서버에 잘못 혹은 악의적인 데이터가 전송되는 것을 차단할 수 있다.

둘째, 임베디드 시스템에 임베디드 DB를 설치함으로써 일시적 혹은 데이터에 효과적인 관리를 할 수 있다.

셋째, 호스트 PC에 비해 상대적으로 저가인 임베디드 시스템을 중간위치에 설치함으로써 도난과 같은 안전문제 또한 예방할 수 있다.

```

Form1::Form1( QWidget* parent, const char* name, WFlags fl )
: QWidget( parent, name, fl )
{
    if ( !name )
        setName( "Sensor Network" );
    resize( QSize(459, 181).expandedTo( minimumSizeHint() ) );
    connect( send, SIGNAL( clicked() ), this, SLOT( serial_init() ) );
    connect( pushButton1, SIGNAL( clicked() ), QApplication, SLOT( quit() ) );
    fd = ::open( MODEMDEVICE, O_RDWR | O_NOCTTY |
        O_NONBLOCK );

    if( fd < 0 )
    {
        perror( MODEMDEVICE );
    }
    else
    {
        QSocketNotifier* notifier = new
            QSocketNotifier( fd, QSocketNotifier::Read, this );
        QObject::connect( notifier, SIGNAL( activated( int ) ), this,
            SLOT( Read_Data() ) );
    }
}

void Form1::Serial()
{
    int result;
    serial_init();
    char txbuff = {0x31};
    result = ::write( fd, txbuff, sizeof( txbuff ) );
    ::close( fd );
}
    
```

[그림 7] 센싱 데이터 관리 프로그램 일부

5. 결론

기존에는 임베디드 시스템의 하드웨어 사양이 PC에 비해 저사양, 저용량인 관계로 센서 네트워크 분야에 응용될 때는 센싱 데이터의 중계기 역할만을 수행하였다. 본 논문에서는 이런 임베디드 시스템의 역할을 확대하기 위하여 embedded MySQL을 임베디드 시스템에 설치하여 C/C++언어로 개발한 시리얼 통신 프로그램과 연동하게 함으로써 임베디드 시스템이 센서노드로부터 전송된 센싱 데이터의 관리를 가능하다. 또한, 임베디드 시스템의 TFT/LCD상에서 모니터링을 위해 embedded Qt를 사용하여 GUI를 구현하였다. 이와 같은 센싱 데이터 관리 기능 외에 추가로 필터링 프로그램을 설치하여 오류가 있거나 특정구간내의 평균값에 큰 차이를 보이는 데이터는 별도로 관리할 수 있다. 즉, 임베디드 시스템에 임베디드 데이터베이스를 설치한 다음 센싱 데이터를 저장 및 관리함으로써 센싱 데이터가 전송될 때마다 게이트웨이를 통한 호스트와 통신하는 전송횟수를 줄일 수 있으며, 임베디드 시스템에서 필터링 과정을 통해 보다 정확하고 유효한 데이터만을 호스트에 전송함으로써 수집된 센싱 데이터에 기반한 분석결과의 신뢰성을 높일 수 있다. 향후에는 필터링 과정과 더불어 센싱 데이터에 대한 보안성에 대해 추가연구가 필요할 것으로 사료된다.

참고문헌

- [1] 조진성, "임베디드 시스템 프로젝트 2", 홍릉과학출판사, 2008.
- [2] 한국정보통신인력개발센터, "ESDP 표준교재", 사이텍 미디어, 2004.
- [3] 김현철, "임베디드소프트웨어산업 현황 및 전망", 정보산업지 2007권 1호, pp30-33, 2007.
- [4] 최진영, "임베디드 소프트웨어와 전통산업의 융합", 한국정보산업연합회, "FKII Digital 365", pp. 36-39, 2008.
- [5] 윤영준, "모바일 네트워크를 이용한 임베디드 전광판 원격제어 시스템의 구현", 군산대 석사논문, 2006.
- [6] 장동욱, "USN을 이용한 모바일 u-Health Care 시스템의 구현", 호서대 석사논문, 2007.
- [7] 남상엽, 송병훈 공저, "무선 센서 네트워크 활용", 상학당, 2005.
- [8] <http://www.embeddedworld.co.kr/>
- [9] <http://www.openwith.net>
- [10] <http://www.mysql.com>
- [11] 김대영 외 3, "센서 네트워크 운영체제/미들웨어 기술동향", 2005.
- [12] 정보통신부, "u-센서 네트워크 구축 기본계획", 2004.
- [13] I. F. Akyildiz et al., "Wireless Sensor Network: a survey," Computer Networks, Vol. 38, pp. 393-422, March 2002.
- [14] Shneidman, J. et al., Hourglass: An Infrastructure for Connecting Sensor Networks and Applications, Havard Technical Report TR-21-04,2004.
- [15] 김민수 외 1, "유비쿼터스 환경에서의 센서 데이터베이스 기술", IITA주간기술동향 통권 1187호, 2005.
- [16] KESIC사무국, "국내 임베디드 소프트웨어 산업 현황 및 실태분석", 2007.

권 대 곤(Dae-Gon Kweon)

[정회원]



- 2000년 2월 : 경남대학교 컴퓨터공학과 (공학박사)
- 1997년 ~ 현재 : 경남도립남해대학 컴퓨터정보과 교수

<관심분야>

정보시스템, 소프트웨어 테스트, 품질평가 및 신뢰도, 웹프로그래밍

최 신 형(Sin-Hyeong Choi)

[종신회원]



- 1993년 2월 : 울산대학교 전자계산학과 (공학사)
- 1995년 2월 : 경남대학교 전자계산학과 (공학석사)
- 2002년 8월 : 경남대학교 컴퓨터공학과 (공학박사)
- 1995년 7월 ~ 1998년 6월 : 해군사관학교 전산과학과 전임강사
- 2003년 8월 ~ 현재 : 강원대학교 전기제어공학부 조교수

<관심분야>

임베디드 시스템, 무선센서네트워크, 분산시스템 보안, 테스트 및 품질평가