

# 임베디드 시스템에서의 데이터관리방안

최신형\*, 이봉섭\*, 진광윤\*\*, 정명수\*\*, 이현창\*\*\*, 박우철\*\*\*, 한군희\*\*\*\*

\*강원대학교 전기제어공학부, \*\*강원대학교 컴퓨터공학과

\*\*\*강원대학교 기계자동차공학부, \*\*\*\*백석대학교 정보통신학부

e-mail:cshinh@kangwon.ac.kr

## Data Management Method in Embedded System

Sin-Hyeong Choi\*, Bong-Sub Lee\*, Kwang-Youn Jin\*\*, Myung-Soo Jung\*\*

Hyun-Chang Lee\*\*\*, Woo-Cheul Park\*\*\*, Kun-Hee Han\*\*\*\*

\*Division of Electrical & Control Engineering, Kangwon National University

\*\*Dept. of Computer Engineering, Kangwon National University

\*\*\*Division of Mechanical & Vehicle Engineering, Kangwon National University

\*\*\*\*Division of Computer and Information, Baekseok University

### 요 약

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 이와 같이 무선센서네트워크를 통해 수집되는 센서정보를 호스트가 아닌 임베디드 시스템에서 데이터베이스화하여 관리하는 방안에 대해 연구하였다. 이를 활용하면 각 영역별로 분포되어있는 임베디드 시스템에서 데이터를 관리·저장함으로써 호스트의 부하를 줄여줄 수 있다.

### 1. 서론

무선 통신 기술의 발달과 센서 네트워크 기술의 발전으로 각종 센서를 사용하여 수집되는 다양한 정보를 이용하는 것이 쉬워졌으며, 이런 기술은 군사, 환경 및 홈 네트워크 등 다양한 분야에 적용되고 있다. 또한, 센싱 기능과 통신 기능을 가진 다수의 센서 노드들로 구성된 센서 노드 그룹을 통해 온도, 조도 및 습도 등의 센싱 데이터를 실시간적으로 수집이 가능하다[1, 2, 3, 4, 5].

이런 센싱 데이터에 대한 수집 및 관리는 전적으로 게이트웨이를 통해 호스트에서 관리하는 것이 일반적이다. 이와 같은 상황은 데이터베이스와 같은 데이터 관리시스템이 호스트와 같은 고성능, 대용량 저장장치가 있는 곳에만 설치되어 온 것에도 기인한다. 하지만 SDRAM이나 Flash 메모리와 같은 저용량 저장장치를 가지는 임베디드 시스템을 위한 데이

터베이스가 개발되고 이를 활용하여 데이터를 관리하려는 움직임이 활발하게 진행되고 있다.

따라서 본 논문에서는 센서 노드로부터 수집되는 센싱 데이터를 일차적으로 임베디드 시스템에서 데이터베이스화하여 관리하는 방안을 제시한다. 이를 위해 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.

본 논문은 2장에서 관련연구로 현재까지 개발되고 많이 사용되고 있는 임베디드용 데이터베이스에 대해 살펴보고, 3장에서는 제안한 시스템 구조에 대해 설명하며, 4장에서 결론을 맺는다.

## 2. 관련연구

### 2.1 임베디드 데이터베이스

임베디드 데이터베이스란 특정 애플리케이션의 한 부분으로 동작하는 데이터베이스 소프트웨어 컴포넌트라고 정의할 수 있다. 즉, 임베디드 데이터베이스란 독립적으로 운용되는 DBMS가 아니라, 소스에 링크시켜서 사용되는 데이터베이스라고 할 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[6, 7, 8].

### 2.2 임베디드 데이터베이스 종류

#### (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 데이터베이스이면서도 일반 DBMS에서 지원하는 대부분의 기능을 지원하고 있다. 즉, 트랜잭션, 로깅, 분산처리 등의 기능을 기본으로 한다.

Berkeley DB는 어플리케이션에 직접적으로 연결되기 때문에 임베디드라고 부르며, Berkeley DB는 어플리케이션과 동일한 주소공간에서 운영된다. 또한, Berkeley DB는 C, C++, Java, Perl, Tcl, Python, 그리고 PHP 등의 프로그래밍 언어를 위한 간단한 function-call API만을 제공하고, 모든 데이터베이스 작업은 라이브러리 안에서 이뤄진다.

Berkeley DB 라이브러리는 이식성이 좋아서, 모든 Unix 및 Linux 변종, 윈도우즈, 그리고 수많은 실시간 임베디드 운영체제에서 작동한다. 그러나 Berkeley DB는 SQL 같은 쿼리 언어와 테이블 스키마 등의 개념은 지원하지 않는다.

#### (2) GNU dbm

GNU dbm은 사용하기 쉽고 크기가 작으며 검색 속도가 빨라 임베디드용으로 적합하고, 보통 gdbm으로 부른다. 배포판에 따라서 gdbm이 리눅스에 설

치되어 있는 경우도 있고 아닌 경우도 있다. gdbm은 표준 UNIX dbm의 함수와 유사하게 작동하는 데이터베이스 함수모음이다. 이 함수들을 사용하여 데이터베이스 파일들을 만들거나 처리할 수 있다. 그리고 gdbm은 기본적으로 key/data를 짝으로 데이터베이스 파일에 저장하여 처리한다. 여기서 key와 data는 하나의 쌍으로 gdbm의 디스크 파일에 저장된다. 이것을 gdbm 데이터베이스라고 부른다. 또한, gdbm에서는 하나의 애플리케이션이 동시에 여러 데이터베이스를 여는 것이 가능하다.

#### (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 전자 장비에 MySQL 데이터베이스 서버의 기능을 삽입할 수 있다. 내장 MySQL 데이터베이스는 인터넷 장비, 공공 키오스크, Turn-key 하드웨어-소프트웨어 조합 장비, 고성능 인터넷 서버, CD-ROM에 담긴 독립 데이터베이스에 이용된다.

## 3. 시스템 구조

일반적으로 임베디드 시스템은 호스트 컴퓨터에 비해 상대적으로 저사양, 저성능, 저용량이라는 인식으로 인해 각종 센서로부터 전송되는 센싱 데이터를 수집하여 호스트로 보내어주는 중계기 역할을 수행한다.

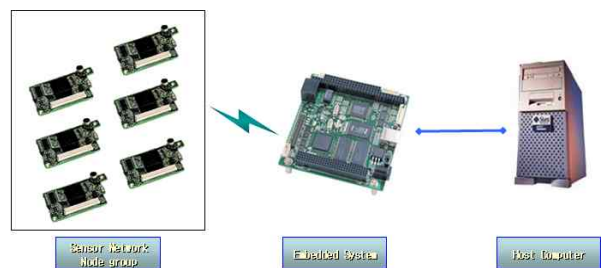


그림 1. 기존의 센싱 데이터 수집방안

이런 구조는 제한된 공간에서 적은 수의 센서로부터 센싱 데이터를 수집하는 환경에서는 별다른 문제없이 작동할 수 있다. 하지만 센서 노드 그룹의 수가 방대하고 특정 영역별로 분산되어 있는 환경 하에서는 매번 수집되는 센싱 데이터를 처리하기 위

해 게이트웨이 역할을 하는 임베디드 시스템과 호스트가 통신함으로써 호스트의 부하는 높아질 것이다. 또한, 무선전송의 특성상 부정확한 데이터나 신뢰하지 못하는 데이터 또한 걸러지지 않고 호스트에 직접 전송됨으로써 보안 혹은 데이터 신뢰성 면에서 문제를 야기할 수도 있다.

본 논문에서는 데이터 중계기 역할만 하는 임베디드 시스템의 역할 확대 및 잦은 데이터 처리로 인한 호스트 컴퓨터의 부하 경감을 위한 방법으로 임베디드 데이터베이스를 이용한 센싱 데이터 처리방안을 제시한다.

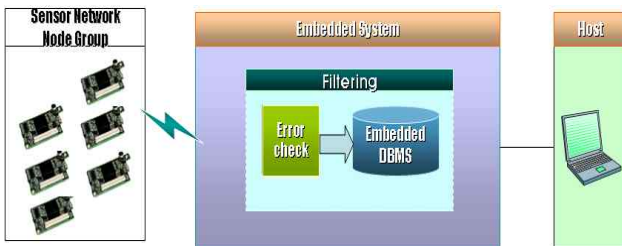


그림 2. 제안한 센싱 데이터 관리방안

그림 2에서 나타난 것처럼 센서노드로부터 전송되는 센싱 데이터는 우선 임베디드 시스템의 오류검사 기능을 통해 유효한 데이터를 임베디드 시스템에 설치된 임베디드 데이터베이스에 저장되고 일정조건이 되거나 호스트 컴퓨터에서 요청이 있으면 부분 혹은 일괄적으로 전송할 수 있는 구조이다.

#### 4. 시스템 설계

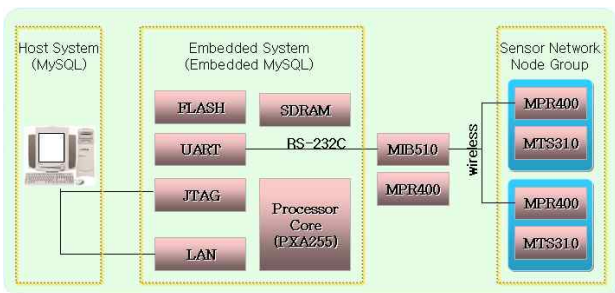


그림 3. 시스템 개발환경 구성도

그림 3은 시스템의 개발환경을 나타낸다. 호스트 시스템, 임베디드 시스템 그리고 센서 네트워크 노드 그룹으로 구성된다. 호스트 시스템에서는 리눅스 운영체제가 설치된 환경으로 각종 응용 프로그램 개발을 위한 Qt/Embedded와 데이터 저장을 위한 MySQL DBMS가 있으며, 임베디드 시스템에는 호

스트 시스템을 통해 개발된 시리얼 통신 프로그램이 포함된 센싱 데이터 수집부분과 임베디드 데이터베이스인 embedded MySQL이 설치되어 있다. 센서 네트워크 노드 그룹은 센서노드인 MTS310과 무선송신기능을 가진 MPR400로 구성되어 수집된 센싱 데이터를 무선으로 임베디드 시스템으로 전송하는 역할을 한다.

#### 5. 결론

본 논문에서는 무선센서네트워크를 통해 수집되는 센서 데이터를 호스트 시스템이 아닌 임베디드 시스템에서 데이터베이스화하여 관리하는 방안에 대해 연구하였다. 이를 활용하면 각 영역별로 분포되어있는 임베디드 시스템에서 데이터를 관리·저장함으로써 기존의 게이트웨이를 통해 전송되는 센싱 데이터를 직접 호스트가 처리하는데 소요되는 호스트 측면의 부하를 줄여줄 수 있다.

#### 참고문헌

- [1] 김대영 외 3, “센서 네트워크 운영체제/미들웨어 기술동향”, 2005
- [2] 정보통신부, “u-센서 네트워크 구축 기본계획”, 2004
- [3] 남상엽, 송병훈 공저, “무선 센서 네트워크 활용”, 상학당.
- [4] Shneidman, J. et al., Hourglass: An Infrastructure for Connecting Sensor Networks and Applications, Harvard Technical Report TR-21-04,2004.
- [5] I. F. Akyildiz et al., “Wireless Sensor Networks: a survey,” Computer Networks, Vol. 38, pp. 393-422, March 2002.
- [6] <http://www.embeddedworld.co.kr/>
- [7] <http://www.openwith.net>
- [8] <http://www.mysql.com>