

Tiny-DB와 MySQL을 이용한 유비쿼터스 센서 네트워크 기반의 실시간 정보 서비스 설계 및 구현

강경옥¹, 김용우¹, 권훈¹, 김부림¹, 김도현^{1*}

Design and Implementation of Realtime Information Service based on Ubiquitous Sensor Network Using MySQL and Tiny-DB

Kyoung-Ok Kang¹, Yong-Woo Kim¹, Hoon Kwon¹, Bu-Rim Kim¹,
Do-Hyeun Kim^{1*}

요약 무선 센서 네트워크는 컴퓨팅 기술 능력과 무선 통신 능력을 지닌 센서 노드들을 이용하여 자율적인 네트워크를 형성하고, 서로 간에 정보를 전달한다. 최근 센서 네트워크의 노드의 초저전력, 초소형, 저비용에 대한 연구와 더불어 수집된 센싱 데이터의 활용 방법에 대한 연구가 진행되고 있다. 이에 본 논문에서는 Tiny-DB와 MySQL를 연동하여 센서 네트워크에 수집된 실시간 대용량 데이터를 효과적으로 저장하고, 산불이나 화재 등의 재난을 방지하기 위해 IPv4/IPv6 기반의 인터넷을 이용하는 실시간 정보 서비스의 프로토타입을 설계하고 구현한다. 이를 위해 TinyOS를 이용하여 센싱 쿼리 프로세싱을 통해 정보를 추출하는 모듈, MySQL 데이터베이스 관리 시스템, JDBC를 이용하여 정보를 서버의 데이터베이스의 연동 모듈, IPv4/IPv6 기반에 인터넷에서 ODBC를 이용하여 실시간으로 센싱 데이터를 수신하여 사용자에게 제공하는 클라이언트 모듈 등을 개발한다.

Abstract Wireless sensor network forms the self-organization network, and transfers the information among sensor nodes that have computing technology ability and wireless communication ability. The recent sensor network is study for low-power, micro, low cost of node is proceeded. Recently, the research of application services in wireless sensor networks is proceeded. Therefore, in this paper, we design the prototype of the real-time information service that support a user the information of temperature, illumination etc. And, we implement the alarm application service for the disaster prevention on Internet base on IPv4/IPv6. We develop the module of the extract information using the query processing based on TinyOS, and the module of the server's database using MySQL data base management system and JDBC. Additionally, we develop the client module that receive the real-time sensing data using ODBC in Internet based on IPv4/IPv6.

Key Words : 센서 네트워크, TinyOS, 온도, 정보 서비스, MySQL

1. 서론

무선 센서 네트워크는 센서들이 무선 방식으로 네트워크에 연결되어 데이터를 송수신 할 수 있다. 이러한 센서 네트워크는 네트워크를 구성하는 센서 노드 상호간에 자

율적으로 정보를 수집, 관리 및 제어하는 협력 네트워크를 자동 구성하며, 다양한 산업 응용 분야에서 물리 현상 계와 컴퓨터, 나아가서는 인간을 연결하는 인터페이스 역할을 담당하게 된다. 그리고 언제라도 원하는 정보를 실시간으로 사용자 요구에 맞춘 형태로 제공하게 된다. 이러한 센서 네트워크를 기반으로 하는 USN(Ubiquitous Sensor Network)에서 센서 노드는 대부분 사물에 내장되며 원활한 전원공급을 위해 배터리를 이용한다. 따라서 센서 노드의 배터리는 크기가 작고 전력 소모량이 적어야 하고, 또한 외부의 환경 변화에도 가능한 영향을 받지

이 논문은 2003년 천안대학교 교내연구비의 지원에 의하여 연구되었음

¹제주대학교 통신컴퓨터공학부 컴퓨터공학전공

*교신저자: 김도현(kimdh@cheju.ac.kr)

않아야 한다. 이에 미국 버클리 대학에서는 센서 노드들의 환경에 적합한 TinyOS[1]를 개발하였으며, 노드들의 동작에 따른 소스 프로그램들을 공개하였다.

센서 네트워크에서의 센서 노드에서 센싱된 데이터는 싱크 노드라 불리는 중앙 집중 노드로 보내고, 싱크 노드의 Tiny-DB[2]에 저장된다. Tiny-DB에 저장된 데이터는 사용자의 요청에 의해 원하는 정보로 처리하고 가공하여 제공된다. 이런 Tiny-DB는 저용량 데이터를 저장하고, 저 전력 기반의 내장형 TinyOS를 이용하고 있어 인터넷을 통한 전송과 서비스에는 적합하지 않다. 따라서 실시간 대용량 데이터를 저장하여 센싱 데이터들을 원활하게 보관하기 위한 별도의 데이터베이스 서버가 요구된다. 그리고 센싱 데이터를 가공하여 IPv4와 IPv6 기반의 인터넷 환경에서 실시간 응용 서비스를 제공할 수 있는 응용 서비스 플랫폼과 프로토타입 개발이 요구된다.

이에 본 논문에서는 MySQL 데이터 관리 시스템과 Tiny DB를 연동하여 대용량 실시간 데이터를 처리하고, ODBC(Open DataBase Connectivity)를 이용한 IPv4와 IPv6 기반의 실시간 정보 서비스의 프로토타입을 구현한다. 이를 위해 센서 네트워크에서 센싱 쿼리 프로세싱을 거쳐 필요한 데이터를 추출하는 모듈을 구현하고, 수집된 실시간 대용량 데이터를 서버에 효과적으로 저장하기 위하여 JDBC(Java DataBase Connectivity)를 이용하여 Tiny-DB와 MySQL 데이터 관리 시스템을 연동하고, 서버의 데이터베이스에 저장하는 모듈을 구현한다. 그리고 산불이나 화재 등의 재난을 방지하기 위해 IPv4/IPv6 인터넷 기반에서 실시간 온도 정보를 추적하여 화면에 보여주고, 임계치를 기준으로 온도의 변화에 따라 경보 서비스를 사용자에게 제공하는 실시간 클라이언트 모듈을 개발한다.

서론에 이어 2장에서는 센서 네트워크에 이용된 TinyOS, Tiny-DB 그리고 MySQL을 설명하고, 3장에서는 실제 응용 프로그램 설계 및 구현을 통한 검증하고, 4장에서는 본 논문에 대한 결론을 맺는다.

2. 관련 연구

국외에서 센서 네트워크의 하드웨어 플랫폼으로 미국 버클리 대학의 Mote 시리즈, 인텔의 iMote, ETH의 btNode, Moteiv의 Telos가 개발되었다. 특히 Telos와 MICAz는 센서 네트워크의 표준 Radio 인터페이스로 유력한 IEEE 802.15.4 LR-WPAN(Low Rate-Wireless Personal Area Network) 표준을 지원하는 Chipcon사의 ZigBee 칩을 이용하고 있어 많은 주목을 받고 있다. 국내

에서 개발된 센서 노드 중 주목할 만한 것은 옥타컴사에서 개발한 Nano-24 센서 하드웨어 플랫폼으로 한국전자통신연구원(ETRI)에서 개발한 센서 네트워크 운영체제인 Nano-Qplus가 설치된 유일한 플랫폼이며, TinyOS도 가능하다.

대표적인 이벤트 기반 센서 네트워크 운영체제인 UC Berkeley의 TinyOS와 한국전자통신연구원에서 개발된 Nano-Qplus 운영체제가 있다. TinyOS는 이벤트 발생에 의한 상태 천이 방식을 채택한 상태 머신 기반의 프로그래밍 개념을 사용한 운영체제로서, 제한된 메모리 공간의 효율적인 이용과 프로세싱의 동시성 등을 지원해 주는 운영체제이다. TinyOS에서는 시스템 자원의 제약 때문에 기존의 IP 프로토콜, 소켓, 쓰레드 개념들을 사용하지 않는다.

TinyOS의 구조는 그림 1과 같이 크게 스케줄러와 컴포넌트로 구분되며, 컴포넌트는 세부적으로 명령어, 이벤트 핸들러, 프레임, 그리고 태스크로 구성된다. 그리고 미국 버클리 대학에서는 Tiny-DB를 개발하였으며, Tiny-DB는 센서 노드에서 정보를 얻어내기 위한 쿼리 처리 시스템이다. TinyOS에서 데이터 처리과정을 위한 다른 프로그램과 달리 Tiny-DB는 센서를 위한 코드 작성이 필요 없다. Tiny-DB에서는 SQL과 같은 간단한 인터페이스를 제공함으로써 원하는 데이터를 얻을 수 있다. Tiny-DB는 쿼리가 주어지면, 센서 네트워크 환경에서 노드로부터 데이터를 모으고, 걸러내어 싱크노드로 보내는 작업을 수행한다[1].

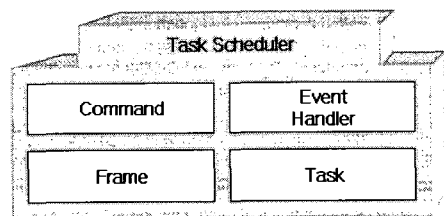


그림 1. TinyOS의 스택 구조

Nano-Qplus 운영체제는 한국전자통신연구원에서 개발 중이며, 제한된 메모리의 사용을 최소화하기 위해 멀티 쓰레드간의 스택을 공유한다. 그리고 멀티 쓰레드 스케줄러(FIFO 방식, 시간 기반 라운드 로빈 방식, 우선순위에 따른 쓰레드 선점 방식)로 다양한 무선 통신 방식(433RF, ZigBee)을 지원한다. 표준형 및 마이크로 임베디드 OS와 동일한 API 세트를 지원하여 기존 시스템 프로그램 개발자에게 친밀감을 제공한다.

MySQL은 SQL(Structured Query Language)을 이용하는 개방 소스 기반의 관계형 데이터베이스 관리 시스템

이다. MySQL은 크기가 작고 간결하여 중소 규모의 데이터베이스 서버로 적합하며, 플랫폼 독립적이다. 또한, MySQL은 다른 상용 데이터베이스보다 단순한 쿼리를 지원하여 빠른 질의 처리 결과와 관리하기가 쉽다는 장점을 갖는다[3].

3. 설계 및 구현

3.1 전체 구성도

센서 네트워크에 얻어진 실시간 정보를 인터넷을 이용한 응용 서비스를 제공하기 위한 기본 플랫폼 구성은 그림 2와 같다. 센서 네트워크는 센싱 데이터를 수집하는 센서노드와 외부 네트워크와 연결되는 싱크노드로 구성되고, 서버는 데이터베이스를 갖고 대용량 실시간 데이터를 효과적으로 저장하고, 실시간 환경에 따라 다양한 응용 서비스를 사용자에게 제공한다. 클라이언트의 응용 서비스는 사용자에게 그래픽하게 제공한다. 본 논문에서 미국 버클리 대학의 MPR2400 제품과 TinyOS를 이용하여 센서 네트워크를 구성하고, TinyDBApp를 이용하여 센서 네트워크로부터 센싱 데이터를 수집한다. 이때 센서 네트워크의 그룹 아이디로 하나의 네트워크를 구성한다. 여기서 게이트웨이와 같은 역할을 하는 싱크노드를 통하여 외부로 데이터를 전송한다. 전송된 데이터는 MySQL을 이용한 대용량 데이터베이스 서버에 저장되며 사용자는 IPv4/IPv6 기반의 인터넷에서 응용프로그램을 이용하여 실시간으로 데이터를 확인할 수 있다. 이때 IPv6 기반의 인터넷을 통해 클라이언트에 전송하기 위하여 릴레이 라우터를 이용한다.

3.2 센서노드와 싱크노드 구성

센서 네트워크에서 센서노드의 하드웨어는 그림 3과 같이 MTS300(Sensor Board) 센서보드와 통신 모듈인 MPR2400(Micaz Processor/Radio Board)로 구성되며, 센싱된 정보를 수집하여 데이터베이스 서버로 전송하는 싱크노드는 그림 4와 같이 MPR2400과 MIB510(Programming and Serial Interface Board)을 노트북에 연결하여 구성한다. MPR2400은 센서 네트워크의 표준 무선 인터페이스로 유력한 IEEE802.15.4 LR-WPAN(Low Rate-Wireless Personal Area Network) 표준을 지원하는 ZigBee칩을 이용하고 있다. MIB510는 여러 가지 MPR군에서 사용할 수 있는 인터페이스 보드이다. MIB510은 RS-232 시리얼 포트를 통해 ISP(in-system processor)에 코드를 다운로드 할 시에 사용된다. 이때 ISP와 Mote는 같은 시리얼 포트를 할당하게 된다. MTS300은 상태를 감지하는 융통성 있는 센서보드이다. MTS300은 마이크로폰, 사운드, 조도, 온도를 센싱 할 수 있다.[4]

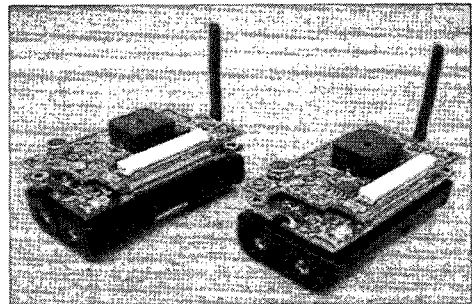


그림 3. 센서노드

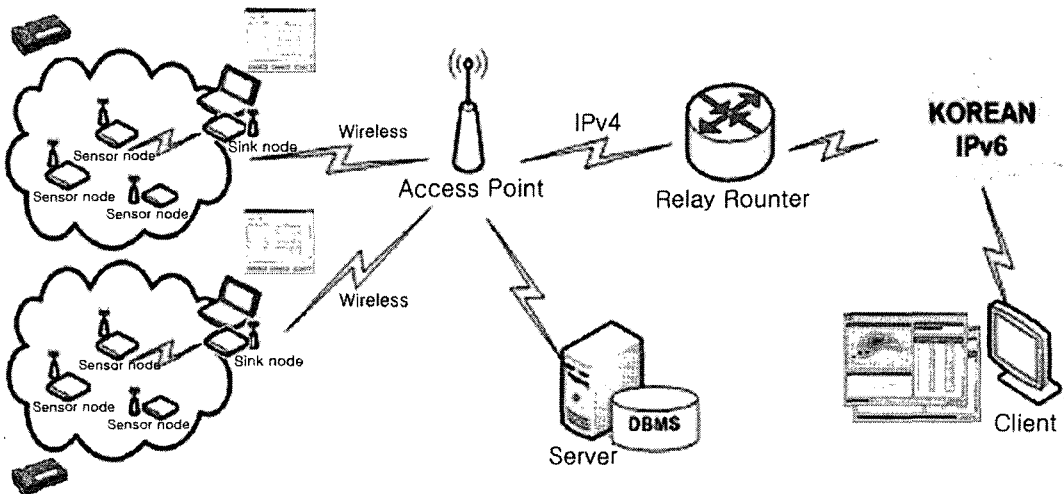


그림 2. IPv4/IPv6 기반의 실시간 정보 서비스 구성

센서 네트워크에서 TinyDBApp 프로그램을 센서노드 및 싱크노드에 설치하여 실시간 데이터를 전달한다. TinyDBApp가 설치된 센서노드는 센서 네트워크에서 쿼리문을 받고 쿼리 프로세서가 내부에서 데이터를 필터링하게 된다. 즉, 여러 수집된 데이터 중에 쿼리문의 조건에 맞는 데이터를 뽑아내며 결과를 싱크노드로 전송한다[5]. 여기서 센서노드의 그룹 아이디는 그 그룹이 하나의 센서 네트워크를 식별한다. 이때 각 센싱된 데이터를 싱크노드로 보내는 역할을 하는 인터페이스 보드의 싱크노드는 id 번호를 0으로 설정한다.

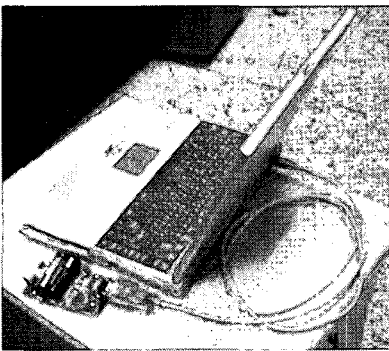


그림 4. 싱크노드

본 논문에서는 많은 센서노드로부터 데이터들을 얻을 수 있으나 온도만을 사용한다. 센싱된 온도를 사용자가 사용하기 위해서는 변환 과정이 필요로 한다. 온도와 빛 센서는 동일한 채널을 공유한다. 이 센서를 사용하기 위해서는 디지털 제어 신호인 PW1은 켜져 있어야 하고 그 출력은 ADC1(analog/digital converter channel 1)에 연결된다. 파워는 세팅신호가 INT1과 INT2이며 각각 빛과 온도를 제어한다. MTS300 센서보드를 장착한 모트의 ADC 출력은 0℃에서 50℃ 정도의 값의 범위에서 아래의 공식에 따라 Kelvin 온도로 변환할 수 있다[4].

3.3 싱크노드와 데이터베이스 서버와의 연동 및 온도 설정

서버와의 연동 및 온도 설정을 위한 응용프로그램에서는 센서 네트워크 관리자가 온도의 표준 값을 설정하여 싱크노드에게 쿼리를 보내며 싱크노드는 각각의 센서노드에게 쿼리를 보낸다. 센서노드는 그 쿼리를 중심으로 데이터를 수집하며, 센싱된 데이터들은 싱크노드를 통해 데이터베이스 서버로 보내어지게 된다. 이 응용프로그램에서는 이렇게 실시간으로 받는 데이터 값들을 사용자에게 유용한 정보로 변환하여 MySQL 데이터베이스에 저장하게 된다.

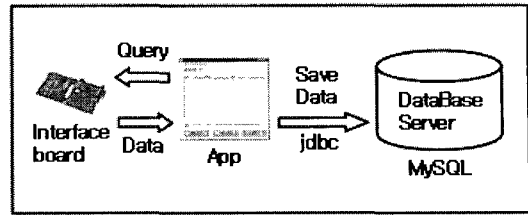


그림 5. 싱크노드와 데이터베이스 서버와의 연동

위의 그림 5와 같이 싱크 노드에서 응용프로그램을 거쳐 데이터베이스 서버까지 데이터 흐름을 보여 주고 있다. 이 응용프로그램으로부터 얻은 센싱된 데이터들은 JDBC를 통해 데이터베이스 서버에 연결하여 저장한다[6].

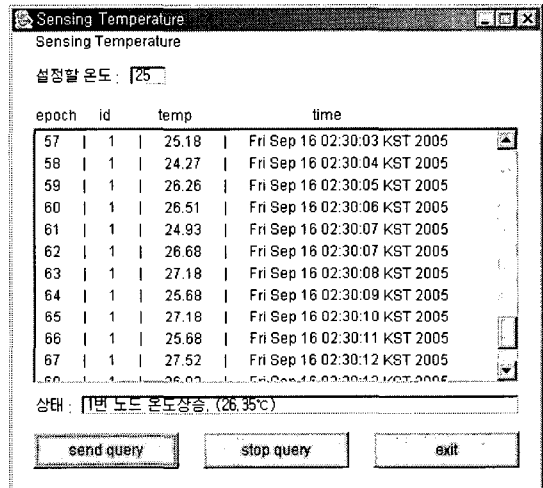


그림 6. 온도 정보의 상태 변화

센서 네트워크 관리자가 응용프로그램에 적당한 온도 값을 설정하고 쿼리 문을 보낸다. 이 응용프로그램은 싱크노드로부터 받는 데이터를 보여준다. 응용프로그램은 화면상에 데이터를 출력과 동시에 데이터베이스 서버에 저장을 한다. 이때 설정한 온도의 값을 초과한 데이터들은 별도의 상태 값을 갖게 된다. 화면에는 그림 6과 같이 간단한 텍스트 형식으로 알려준다.

3.4 데이터 스키마 정의

DBMS로 MySQL을 이용하고 실시간으로 센싱된 온도 정보를 저장한다. 여기에는 각각의 센서노드에서 수집한 데이터를 저장하기 위한 테이블을 갖는다. 표 1처럼 필드 값으로는 기본 키, 노드, 센서의 속성, 현재 시간, 센서 네트워크 관리자가 설정한 온도의 값을 초과한 데이

터들을 구분하기 위한 상태 값이 있다. 표 2에서는 센서 노드의 데이터를 저장하기 위한 테이블의 필드들이다. 여기에는 노드 ID, 온도, 시간, 상태 필드가 있다.

표 1. 센서노드 데이터 스키마

기본 키	노드	센서의 속성			시간	상태
		조도	온도	...		

표 2. 테이블의 필드 속성

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI	NULL	auto_increment
node	int(11) unsigned	YES		NULL	
temp	double(5,2)	YES		0.00	
time	time	YES		NULL	
state	int(2)	YES		NULL	

3.5 사용자 인터페이스

클라이언트 응용 프로그램은 ODBC를 이용하여 MySQL 데이터베이스에 실시간으로 저장되는 센싱된 데이터를 읽어 온 다음에 사용자에게 보여준다. 이 프로그램은 센서 노드들에 의해 데이터가 저장된 데이터베이스 서버로부터 현재 온도, 간략한 위치 정보, 온도 상태 값을 얻어오게 되며, 가져오는 데이터는 가장 최근의 값을 가져오게 된다. 이때, 온도 상태 값을 가져오는 과정은 멀티스레드로 동작하며 사용자는 어떠한 노드든 현재 상태의 변화가 있다는 것을 실시간으로 감시할 수 있다.

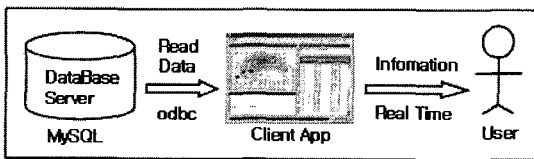


그림 7. 데이터베이스 서버와 클라이언트 관계

위의 그림 7과 같이 데이터베이스 서버와 클라이언트 응용프로그램의 데이터 흐름을 알 수 있다. 이때 클라이언트 응용프로그램은 데이터베이스 서버에 ODBC로 연결하고 가장 최신의 데이터를 읽어온다.

클라이언트 응용 프로그램의 StateCheck()함수는 센서 노드의 현재 온도 상태를 얻어 오는 함수이며, 가장 최근의 온도 상태 값을 가져온다. 기본 값은 0이며 온도가 상승하면 응용프로그램에서 1로 저장하게 되고 그 값에 의해 클라이언트가 상태의 변화를 확인할 수 있다. StateCheck()함수는 StateThread1() 쓰레드 함수를 이용하여 여러 개의 센서 노드의 상태를 확인한다.

```
public string[] NodeInfoCheck(string node)
{
    string[] tmp = new string[3];
    Connection con = new Connection();
    con.MySql("select node, place from info_node where node = " + node + ";");
    data = con.GetNodeInfoData();
    tmp = data.GetData();
    return tmp;
}
```

그림 8. NodeInfoCheck() 함수

클라이언트 응용 프로그램의 NodeInfoCheck()함수는 그림 8과 같이 각 센서 노드에 관련하여 간략한 위치 정보를 제공하는 함수이다. 예를 들면 센서의 위치가 「관 리소로부터 북동쪽으로 50m떨어져 있다」 라는 정보를 얻을 수 있다. 실제 응용 프로그램의 화면에서 마우스 커서가 어떤 센서 노드를 가리키는지에 따라 정보가 변하게 된다.

그림 9에서 보는 바와 같이 클라이언트 응용 프로그램의 TempCheck()함수는 센서 노드가 센싱한 온도 값과 노드 ID, 현재 시간을 얻어 오는 함수이다. 사용자가 마우스 커서로 노드를 클릭하였을 때 응용 프로그램의 오른쪽 화면에 리스트 형태로 보여 지게 된다. 그리고, StateThread1()함수으로써는 온도 상태를 실시간으로 체크하는 스레드이다. 상태를 실시간으로 확인하는 동시에 사용자에게 보여주는 이미지를 상태에 알맞게 설정하는 역할을 한다. 응용프로그램이 시작과 동시에 상태를 확인하는 이 스레드는 가장 먼저 실행되는 부분이라 할 수 있다.

```
public string[] TempCheck(string node)
{
    string[] tmp = new string[3];
    Connection con = new Connection();
    con.MySql("select node, temp, time from node" + node + " order by id desc limit 1;");
    data = con.GetTempData();
    tmp = data.GetData();
    return tmp;
}
```

그림 9. TempCheck() 함수

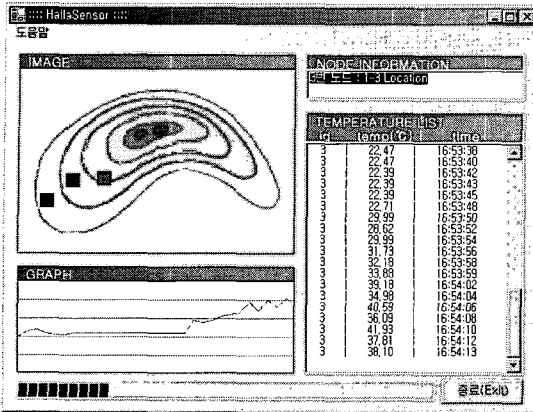


그림 10. 온도 정보를 나타내는 클라이언트 화면

클라이언트 프로그램은 화제를 미리 예방하자는 취지의 시뮬레이션을 실시한다. 이는 사용자가 네트워크가 가능한 PC에서 사용 가능하며 프로그램은 시작과 동시에 센서 네트워크 내의 모든 센서노드들을 감시할 수 있게 된다. 센서 노드가 그림 10과 같이 있다면 응용프로그램 화면 우측에는 실시간으로 센싱된 온도 값이 리스트 형태로 나타나고, 화면 왼쪽 하단에는 선택한 노드의 온도에 따른 그래프가 그려진다. 센서 네트워크 관리자가 설정한 온도 값을 초과한 센서노드에서는 별도의 표시로 사용자에게 알려주게 된다. 그림 10에서는 센싱된 온도가 설정 온도를 초과한 노드의 색이 빨간색으로 변하는 것을 알 수 있다. 사용자는 이러한 시뮬레이션의 알림체계로 인하여 센서노드가 감지할 수 있을 정도의 위험은 미리 알 수 있게 된다.

기존의 Tiny-DB를 이용하는 경우에는 단순히 저용량 데이터베이스를 이용하여 주기적으로 쿼리 요청을 통하여 원하는 센싱 데이터를 리스트 형식과 그래프 형식으로 표현할 수 있다. 그에 반하여, 개발한 시스템은 대용량의 MySQL 데이터베이스를 이용하여 실시간 및 축적된 데이터를 제공하여 IPv4/IPv6 인터넷 기반으로 사용자에게 실시간으로 제공한다. 더불어 설정된 온도 값의 변화에 따라 사용자에게 알림 형태로 제공하여 재난 방지용 경보 서비스를 제공한다.

4. 결론

최근 센서 네트워크가 유비쿼터스 사회를 이끌어 나갈 핵심 기술로 부각되면서 사회 전반에 걸쳐 많은 연구가 이루어지고 있다. 센서 네트워크는 무선 통신 능력을 지닌 센서 노드들을 이용하여 서로 간의 데이터를 획득할

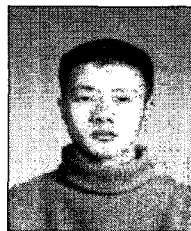
수 있다. 이 센싱된 데이터들은 데이터베이스를 이용하여 수집하고 분석할 수 있으며 이렇게 분석된 데이터들은 사회 전반에 걸쳐 유용함으로 다양한 응용 프로그램 개발이 필요하다. 본 논문에서는 센서 네트워크에 수집된 실시간 대용량 데이터를 효과적으로 저장하기 위하여 JDBC를 이용하여 Tiny-DB와 MySQL를 연동하고, 재난을 방지하기 위해 IPv4/IPv6 인터넷 기반에서 실시간 온도 정보를 추적하여 화면에 보여주고, 온도의 변화에 따라 정보를 사용자에게 제공하는 실시간 경보 서비스의 프로토타입을 설계하고 구현한다. 향후 연구과제로는 Tiny-DB 안에서 데이터 처리 과정을 연구 분석하여 데이터베이스 서버를 거치지 않고 바로 사용자가 이용할 수 있도록 하는 방안을 모색할 예정이다. 또한 본 논문에서 구현한 클라이언트 응용프로그램을 실생활에 적용하는 방안을 검토해 나갈 것이다.

참고문헌

- [1] TinyOS Site, <http://www.tinyos.or.kr>
- [2] Tiny-DB Site, <http://telegraph.cs.berkeley.edu/Tiny-DB>
- [3] MySQL, <http://www.mysql.com>
- [4] Crossbow Technology INC. <http://www.xbow.com>
- [5] Telegraph (Berkeley Database Research), <http://telegraph.cs.berkeley.edu>
- [6] GRAHAM HAMILTON 외, “자바로 처리하는 JDBC 데이터베이스”, 대림, 1997

강 경 옥(Kyoung-Ok Kang)

[준회원]



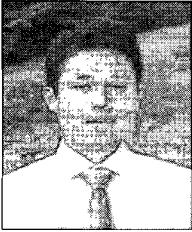
- 2000년 3월 ~ 현재 : 제주대학교 통신컴퓨터공학부 컴퓨터공학 전공 재학

<관심분야>

유비쿼터스 센서 네트워크, 모바일 컴퓨팅

김 용 우(Yong-Woo Kim)

[준회원]



- 2000년 3월 ~ 현재 : 제주대학교 통신컴퓨터공학부 컴퓨터공학 전공 재학

<관심분야>

유비쿼터스 센서 네트워크, 게임프로그래밍

김 부 림(Bu-Rim Kim)

[준회원]



- 2006년 2월 : 제주대학교 통신컴퓨터공학부 컴퓨터공학 전공 (공학사)
- 2006년 3월 ~ 현재 : 제주대학교 대학원 컴퓨터공학과 석사과정

<관심분야>

유비쿼터스 센서 네트워크, 모바일 컴퓨팅

권 훈(Hoon Kwon)

[정회원]



- 2003년 8월 : 제주대학교 해양생물공학과 (이학사)
- 2005년 8월 : 제주대학교 대학원 컴퓨터공학과 (공학석사)
- 2006년 3월 ~ 현재 : 제주대학교 대학원 컴퓨터공학과 (박사과정)
- 2004년 9월 ~ 2005년 12월 : 제주한라대학교 강사

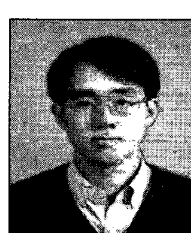
- 2006년 3월 ~ 현재 : 제주한라대학교 겸임교수

<관심분야>

XML, 유비쿼터스 센서 네트워크

김 도 현(Do-Hyeun Kim)

[정회원]



- 1990년 2월 : 경북대학교 전자공학과 (공학석사)
- 2000년 8월 : 경북대학교 전자공학과 (공학박사)
- 1990년 3월 ~ 1995년 3월 : 국방과학연구소 (연구원)
- 1999년 3월 ~ 2004년 2월 : 천안대학교 조교수
- 2004년 8월 ~ 현재 : 제주대학교 부교수

<관심분야>

유비쿼터스 센서 네트워크, 모바일 컴퓨팅