

무선 센서 노드 데이터를 이용한 홈 네트워크 서비스

나선웅[○] 김동균 최영길* 이상정
 순천향대학교 컴퓨터공학부, 한국전기연구원*
 {nsw97[○], kdk70}@sch.ac.kr, ykchoi@keri.re.kr, sjlee@sch.ac.kr

Home Network Service using Data of Wireless Sensor Nodes

Sun-Wung Na[○], Dong-Kyun Kim, Young-Kil Choi*, Sang-Jeong Lee
 Dept. of Computer Science and Engineering, Soonchunhyang University
 Korea Electrotechnology Research Institute*

요 약

본 논문은 홈 네트워크 환경에서 무선 센서 네트워크를 이용하여 홈 네트워크 서비스를 제공하는 서비스 모델을 제안한다. 제안된 서비스 모델은 맥내에 고정 센서 노드들을 배치하고 사용자에게 부착 가능한 센서 노드로 무선 센서 네트워크를 구성한다. 홈 서버에는 등록된 사용자 선호도 프로파일과 센서 노드들로부터 수집된 데이터를 데이터베이스로 구축한다. 이 데이터를 상황정보로 분석하여 사용자 개인의 선호도에 따라 맥내 가전기기를 자동 설정하고 자동 제어하는 서비스를 제공한다. 제안된 방식은 리눅스 환경에서 MySQL 데이터베이스가 내장된 홈 서버와 TinyOS가 탑재된 센서 노드들을 사용하여 구현하고 서비스를 테스트한다.

1. 서 론

최근 맥내의 가전기기를 유무선 네트워크에 연결하여 지능화된 서비스를 제공하는 지능형 홈 네트워크에 많은 관심과 연구개발이 진행되고 있다[1,2]. 지능형 홈 네트워크 구성을 위해서는 사용자의 맥내 기기에 대한 개입을 최소화하면서 사용자 개인의 특성에 맞추어진 서비스를 제공해야 한다. 이를 위해서는 사용자에 대한 정보와 환경데이터를 수집, 관리, 가공하는 시스템이 필요하다. 따라서 사용자 상황의 실시간 데이터 수집을 위한 무선 센서 기술과 사용자의 가전에 대한 설정값 및 구동 상황에 맞는 선호도를 수집, 관리, 가공할 수 있는 서버가 필요하다.

본 논문은 홈 네트워크 환경에서 맥내에 배치된 고정 센서 노드들과 사용자에게 부착된 이동 센서 노드로 구성된 무선 센서 네트워크를 이용하여 지능화된 홈 네트워크 서비스를 제공하는 서비스 모델을 제안한다. 제안된 서비스 모델은 홈 서버에 데이터베이스 형태로 등록된 사용자 선호도 프로파일과 환경데이터 센서들로부터 수집된 데이터를 데이터베이스로 구축하고 상황정보를 분석하여 사용자 개인의 선호도에 따라 맥내 가전기기를 자동 설정하고 자동 제어하는 서비스를 제공한다. 홈 서버에는 사용자의 가전기에 대한 사용 선호도 등을 가지는 프로파일 데이터를 등록하고 관리하는 시스템을 개발하고, 이를 이용하여 사용자의 상황에 적합하게 기기들을 동작시켜 개인화된 서비스를 제공한다. 제안된 방식은 리눅스

이 논문은 2005년도 중소기업청에서 지원하는 기술연구회 공동 연구개발사업에 의하여 연구되었음(S0505616-J1530003-15000011)

환경에서 MySQL 데이터베이스가 내장된 홈 서버와 TinyOS[3]가 탑재된 센서 노드들을 사용하여 구현하고 서비스들을 테스트한다.

2. 홈 네트워크 서비스 시나리오

사용자 선호에 맞는 홈 네트워크 서비스를 제공하기 위해서는 개인 정보, 사용자 선호도, 상황 판단을 위한 환경 데이터 등이 제공되어야 한다. 맥내에 배치된 센서 네트워크는 온도, 조도, 습도, 압력, 소리 등의 환경데이터를 수집하여 홈 서버에 전송하고, 홈 서버는 수신된 환경 데이터를 분석하고 사용자의 선호도에 맞는 냉난방, 조도 및 방방 등의 서비스를 제공한다.

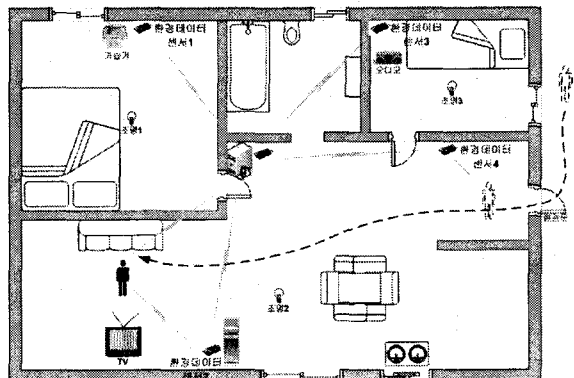


그림 1 단일 사용자에 대한 무선 센서 네트워크 시나리오 예

그림 1은 홈 거주자가 귀가하여 거실의 TV를 시청하는 과정을 무선 센서 네트워크를 이용하여 상황 인식 서비스를 수행하는 시나리오의 예이다. 거주자가 현관문을 열고 댁내로 들어오면 현관문에 부착된 마그네틱 센서는 이를 감지하고 센서노드(환경데이터 센서4)를 경유하여 홈 서버로 전달된다. 사용자는 식별정보가 저장된 사용자 식별노드를 지니고 있어 댁내에 진입하면 환경데이터 센서4의 노드를 경유하여 멀티홈으로 홈서버에 사용자 정보를 전송한다. 전송된 사용자 식별정보와 댁내기기(현관문) 정보는 홈 서버의 데이터베이스에 저장되어 댁내기기에 대한 히스토리, 사용자의 위치 및 프로파일 정보 등을 관리하는데 이용된다. 사용자가 거실에서 TV를 작동시키면 기기에 대한 정보와 TV 주위에 있는 사용자에 대한 정보가 홈 서버 데이터베이스에 저장된다. 홈 서버는 사용자의 선호도를 분석하여 온도, 조도, 습도, 볼륨 등 사용자가 선호하는 환경으로 서비스 구역을 자동 제어한다. 사용자의 위치 정보는 환경데이터 센서2에 수신되는 사용자 식별 노드들의 신호를 감지하여 사용자의 위치를 추론하여 일정한 간격으로 홈 서버의 데이터베이스를 업데이트한다. 사용자 선호도 정보에서 선호 온도값은 에어컨,

보일러, 창문 등과 같은 냉난방과 관련된 기기들과 연동하여 동작하며, 선호 습도값은 가습기, 선호 조도값은 조명기기, 선호 볼륨값은 TV, 오디오 등과 연동 동작한다.

시나리오 상에서 댁내 서비스 구역은 안방, 거실, 작은방, 현관으로 구분되어 있다. 그림 2는 댁내에 다중 사용자가 있는 경우 서비스 시나리오 예이다. 서비스 구역의 사용자가 여러 명이므로 사용자 프로파일도 여러 개가 감지된다. 따라서 여러 사용자의 선호도 충돌에 대한 처리 및 가전 기기를 조작했을 때 어떤 사용자가 조작했는지 여부(선호도 수정에 필요) 등의 문제가 발생한다. 사용자 선호도 충돌은 가전기기에 대한 사용자의 우선순위를 고려하여 처리한다. 그리고 서비스는 가전기기를 조작했을 때 어떤 사용자가 조작했는지 여부와 환경데이터 센서노드에 의해 감지되는 사용자 식별 노드들을 모두 적용한다.

사용자1(아들)이 거실에서 TV 시청중인 상황에서 사용자2(아버지)가 안방에서 거실로 나오는 경우 환경데이터 센서 2가 사용자2를 감지한다. 사용자 선호에 따른 홈 네트워크 서비스를 적용하려 할 때 거실에서 다중 사용자를 감지하고 홈 서버는 데이터베이스를 참조하여 각 사용자에 대해 기기별 우선순위를 적용한다. 에어컨과 조명은 아버지에게 우선순위가 있어 아버지 선호도 값으로 재설정되며 TV는 현재 아들이 사용하고 있으므로 아들이 우선순위가 높다. 따라서 TV의 볼륨은 그대로 변화이 없게 된다. 기기에 대한 사용자별 우선순위는 관리자에 의해 입력 및 수정된다.

모든 사용자가 외출한 경우(댁내에 어떠한 사용자의 신호도 잡히지 않는 경우)에 댁내에서는 방법모드가 실행된다. 방법모드가 동작하면 홈 서버는 모든 댁내 기기들의 상태를 체크하여 방법모드에서 설정되어 있는 값으로 자동 변환된다. 방법모드에서 침입자에 의해 창문 또는 현관문에 설치된 센서로부터 감지신호가 홈 서버에 전송될 경우 사용자 노드 식별을 시도한다. 사용자 식별에 실패할 경우 각 창문, 현관문 근처에 설치되어 있는 웹 카메라

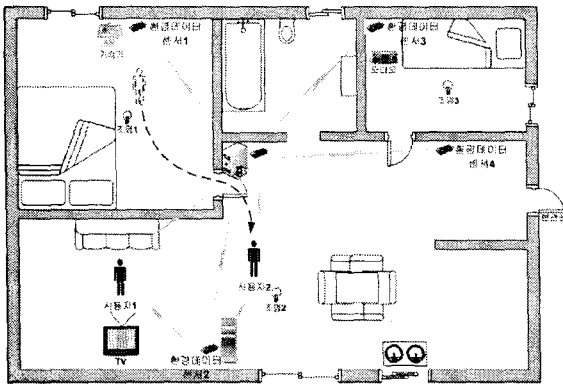


그림 2 다중 사용자에 대한 무선 센서 네트워크 시나리오 예

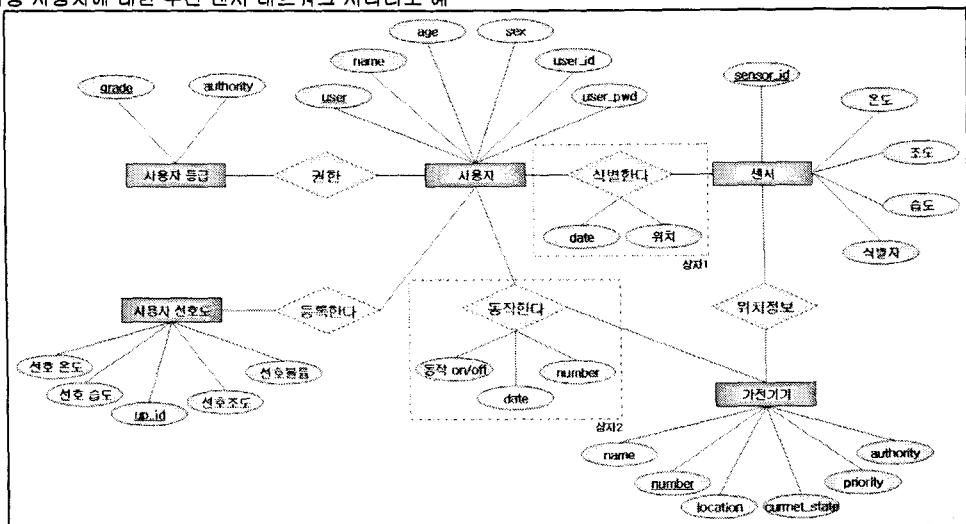


그림 3 무선 센서 노드 데이터베이스 E-R 관계도

가 작동하며, 액내 경고방송이 시작된다. 웹 카메라는 침입자의 동영상과 사진 이미지를 홈 서버에 전달한다. 홈 서버는 데이터베이스에 등록되어 있는 액내 거주자의 핸드폰에 경고메시지와 동영상, 사진이미지 등을 전송한다. 동시에 112에 문자전송 무단주거 침입을 알린다. 홈 네트워크 사용자는 관리자와 일반 사용자로 구분된다. 관리자는 일반 사용자의 프로파일을 생성 및 수정 할 뿐 아니라 서비스(기기동작)에 대한 접근 권한을 부여 할 수 있다. 일반 사용자는 홈 네트워크 환경에서 자신에게 부여된 접근 권한에 따라 서비스를 제공받는다. 사용자 프로파일의 등록 및 관리는 홈 네트워크의 홈 서버 시스템에 의해 관리된다.

3. 무선 센서 노드 데이터베이스 설계

무선 센서노드로부터 많은 실시간 측정 데이터가 지속적으로 발생한다. 발생한 데이터를 수집, 저장 및 가공을 위해 데이터베이스를 구축한다. 그림 3은 본 논문에서 설계한 무선 센서 노드에 대한 데이터베이스 E-R 관계도를 나타낸다. 크게 사용자, 센서, 가전기기 개체로 구성된다. 사용자 개체는 사용자 등급 개체로부터 사용자 권한을 참조하며, 사용자 선호도 개체로부터 사용자의 선호 정도를 참조한다. 사용자 선호도 개체는 사용자의 선호 온도, 습도, 조도, 불륨 등의 속성을 가진다. 사용자 개체는 일반적인 속성과 시스템에 접근하기 위한 사용자 id, 패스워드 속성들을 포함한다. 센서 개체는 온도, 습도, 조도 등의 환경데이터뿐만 아니라 사용자 식별과 사용자의 위치를 추적하는데 참조된다. 사용자 개체와 센서 개체를 통해 추출된 날짜시간, 위치 등의 속성들을 조합하여 사용자 위치가 파악된다. 가전기기 개체는 센서 개체로부터 위치정보를 참조하며 사용자 개체와 함께 가전기기의 동작상태, 사용시간 등의 속성을 참조하여 가전기기의 히스토리 테이블을 구축한다. 데이터베이스 테이블은 관리자가 직접 작성하는 사용자 기본정보, 센서 기본정보, 사용자 등급, 기기 기본정보, 사용자 선호 테이블이 있다. 또한 액내에서 발생한 상황정보(사용자 위치, 기기 히스토리, 환경 데이터 등) 등을 저장관리 테이블들로 구성된다.

표1 사용자 기본정보 테이블

user(FK)	name	age	sex	usr_id	user_pwd	grade(FK)	sensor_id(FK)
사용자 1	나선용	50	남	Now	123	1	센서 5
사용자 2	오홍목	46	미	Ohr	456	2	센서 6
사용자 3	김희성	23	남	Uhs	789	2	센서 7

표 1은 사용자 기본정보 테이블이다. 사용자의 기본정보를 포함하며, 사용자가 지닌 사용자 식별 노드의 센서 ID를 보여준다. 사용자 ID와 사용자 패스워드는 홈 서버 시스템에 접속하기 위해 사용된다.

표 2는 사용자의 등급을 참조하는 테이블이다. 이 테이블을 참조하여 사용자의 가전기기에 대한 우선순위를 결정한다. 표 3은 센서 테이블을 나타낸다. 센서 id 1~4번은 각 서비스 구역에 배치된 고정노드이며, 5~7번은 사용자가 지니고 있는 사용자 식별 노드를 나타낸다.

표 2 사용자 등급 테이블

grade(FK)	authority
1	관리자
2	거주자
3	외부인

표 3 센서 테이블

sensor_id(FK)	location(FK)
센서 1	안방
센서 2	거실
센서 3	작은방
센서 4	현관
센서 5	사용자 1
센서 6	사용자 2
센서 7	사용자 3

표 4 위치별 모니터링 테이블

location	current_temperature	current_humidity	current_조도	static_sensor_id
안방	20	60	밝음	1
거실	19	50	중간	2
작은방	20	60	중간	3
현관	22	50	중간	4

표 4는 서비스 구역별 모니터링 테이블로 각 위치(서비스 구역)의 현재 온도, 현재 습도, 현재 조도 등의 정보를 가진다.

표 5 기기 기본 정보 테이블

name	number(FK)	location	current_state	static_sensor_id(FK)	authority
베어컨	1	거실	off	2	2
보일러	2	거실	off	2	2
조명 1	3	안방	off	1	2
조명 2	4	거실	off	2	2
조명 3	5	작은방	off	3	2
창문 1	6	안방	close.	1	2
창문 2	7	거실	close.	2	2
현관문	8	현관	close.	2	2
가습기	9	안방	off	1	2
TV	10	거실	Off	2	2

표 5는 기기 기본정보 테이블로 액내 기기들의 위치, 현재 상태, 사용권한 등을 포함한다.

표 6 사용자 선호 테이블

up_id(FK)	user(FK)	prefer_temperature	prefer_humidity	prefer_illumination	prefer_volume
1	1	20	60	밝음	6
2	2	19	50	중간	7
3	3	20	60	중간	8

표 6은 사용자 선호 테이블로서 각 서비스 구역별 사용자의 선호정보를 가진다.

그림 4는 그림 1의 시나리오 예에 대해 데이터베이스 테이블의 관계도로 나타낸 것이다. 사용자가 거실에서 TV를 작동시키면 홈 서버의 Device_history 테이블에 기기번호와 사용자 정보, 기기상태가 저장된다. Device_information 테이블로부터 TV의 기기번호를 참조한다. User_information 테이블로부터 사용자에 부착된 식별노드 ID를 참조한다. User_location 테이블에는 작동된 기기의 위치와 감지된 사용자 정보 데이터가 저장된다. 홈 서버는 구축된 테이블을 참조하여 사용자의 선호

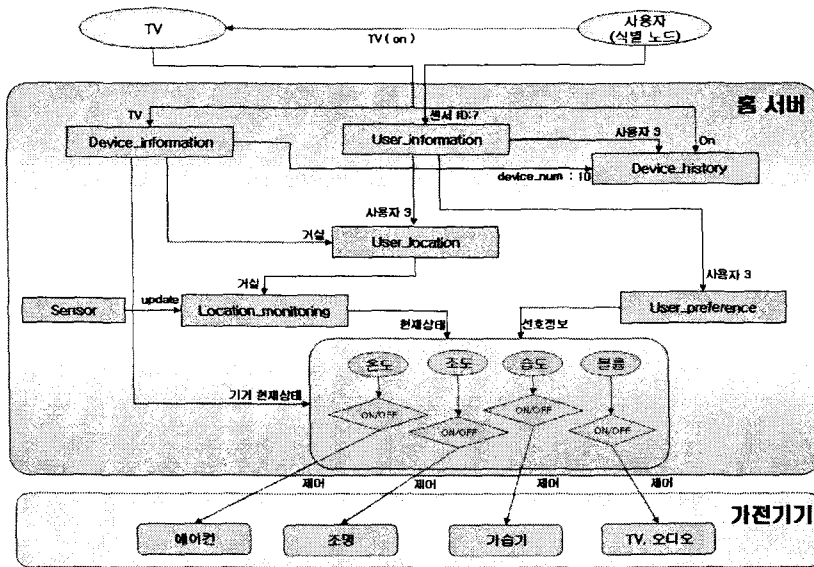


그림 4 센서 노드 데이터베이스 흐름도

정보와 사용자가 위치한 구역의 현재상태 등을 파악한다. Location_monitoring 테이블은 각 고정센서를 통해 지속적으로 현재상태를 업데이트 한다. User_preference 테이블에 저장된 온도, 조도, 습도, 분광 등의 사용자 선호정보와 기기 및 환경의 현재상태를 비교하여 사용자가 선호하는 서비스를 수행한다.

본 논문에서는 제안된 홈 네트워크 서비스를 위해 리눅스의 GUI 툴인 QT와 연동된 MySQL을 사용하여 서비스 시스템을 구현하였다. 각종 기기와 센서노드 등으로부터 구축된 데이터베이스로부터 서비스 정보가 추출된다. 다음은 각 서비스 정보 추출을 위해 SQL 쿼리를 적용한 예를 보여준다.

- ▶ 사용자 정보 테이블로부터 센서 id 에 맞는 사용자 이름과 등급 가져오기

```
QSqlQuery query(
" SELECT user,grade
FROM User_informtion
WHERE sensor_id="+sensed_node_no+"");
```

- ▶ 디바이스 정보 테이블에 동작된 기기의 현재상태 업데이트

```
QSqlQuery query(
"UPDATE Device_informtion
SET current_state = "+TV_state+"
WHERE number="+triggerred_device_no+"");
```

- ▶ 디바이스 history 테이블에 기기의 동작 상태, 사용자, 시간 등을 저장

```
QSqlQuery query("insert into User_Location
values('autoincrement'," +user+"', 'now()',
"+triggerred_device_location+", "+sensed_sensor_no+"");
```

- ▶ 사용자 위치 테이블로부터 사용자 위치 읽기

```
QSqlQuery query("SELECT location
from User_Location ORDER BY ulid DESC LIMIT 1");
```

- ▶ 사용자의 위치를 확인하기 위해 trigger 된 디바이스 정보 테이블로부터 위치정보를 가져오기

```
QSqlQuery query("SELECT location
from Device_informtion
WHERE number="+triggerred_device_number+"");
```

- ▶ 사용자 정보와 기기위치정보를 통해 사용자의 위치를 테이블에 저장

```
QSqlQuery query("insert into User_Location
values('0'," +user+"', now(),
"+triggerred_device_location+", "+sensed_sensor_no+"");
```

- ▶ 사용자 선호 테이블로부터 선호값 읽기

```
QSqlQuery query("select
prefer_temperature,prefer_humidty,prefer_illumination,prefer_volume
from User_Preference
WHERE user="+user_id+"");
```

- ▶ 모니터링 테이블로부터 위치에 대한 현재상태 읽기

```
QSqlQuery query("SELECT * from Location_monitoring
WHERE location="+current_location+"");
```

4. 구현 및 테스트

그림 5는 본 논문에서 테스트를 위해 구성한 시스템 구성도이다. 홈 서버 환경은 리눅스(kernel 2.4.20-8)이며

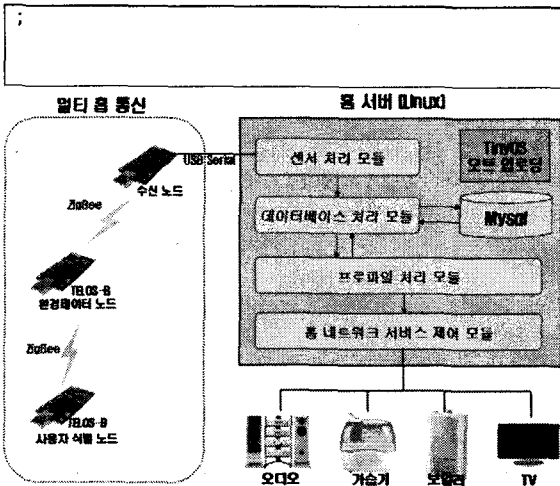


그림 5 시스템 구성도

프로그램을 센서노드에 업로딩한다. 멀티 홉 통신을 통해 사용자 식별 노드 또는 환경데이터 센서로 데이터가 전송되고, 수신노드는 USB serial 형태로 홈 서버의 센서처리 모듈을 통해 데이터를 최종 수집한다. 데이터베이스 처리 모듈은 데이터베이스와 연동하여 데이터를 가공 관리한다. 프로파일 처리 모듈은 사용자의 개인정보 및 선호도를 관리한다. 가공된 데이터는 홈 네트워크 서비스 제어 모듈을 통해 맥내 가전 기기들을 제어한다.

노드 간 통신을 위하여 TinyOS[3]에서 기본적으로 제공하는 오실로스코프 프로그램에 멀티홉 라우팅과 timestamp 정보의 브로드캐스팅 기능을 추가하였다.

그림 6은 멀티 홉 기능이 추가된 오실로스코프 프로그램의 부분을 보여주는 그림이다.

```

#include OscopeMsg;
#include MultiHop;
#include Bcast;

Configuration Oscilloscope {
  Implementation
  {
    Components Main, OscilloscopeM
    , TimerC
    , LedsC
    , DemoSensorC as Sensor
    , GenericCommPromiscuous as Comm
    , Bcast, MultiHopRouter as multihopM, QueuedSend ;

    Main.StdControl -> Sensor;
    Main.StdControl -> Comm;

    Main.StdControl -> OscilloscopeM;
    Main.StdControl -> TimerC;
    Main.StdControl -> Bcast.StdControl;
    Main.StdControl -> multihopM.StdControl;
    Main.StdControl -> QueuedSend.StdControl;

    OscilloscopeM.Timer -> TimerC.Timer[unique("Timer")];
    OscilloscopeM.Leds -> LedsC;
    OscilloscopeM.SensorControl -> Sensor;
    OscilloscopeM.ADC -> Sensor;

    OscilloscopeM.Bcast -> Bcast.Receive[AM_OSCOPERESMSG];
    Bcast.ReceiveMsg[AM_OSCOPERESMSG] -> Comm.ReceiveMsg[AM_OSCOPERESMSG];

    OscilloscopeM.RouteControl -> multihopM;
    OscilloscopeM.Send -> multihopM.Send[AM_OSCOPERESMSG];
    multihopM.ReceiveMsg[AM_OSCOPERESMSG] -> Comm.ReceiveMsg[AM_OSCOPERESMSG];
  }
}
    
```

그림 6 수정된 멀티 홉 프로그램 주요부분

TinyOS(1.1.3)를 포함하였다. TinyOS는 nesC로 작성된 가전기기의 ON/OFF 상태를 모니터링하는 화면 예이다. 즉, 각 위치 별 온도, 습도, 조도, 불꽃에 대한 현재상태를 실시간으로 보여주며 맥내 가전기기의 on/off 여부를 보여준다.

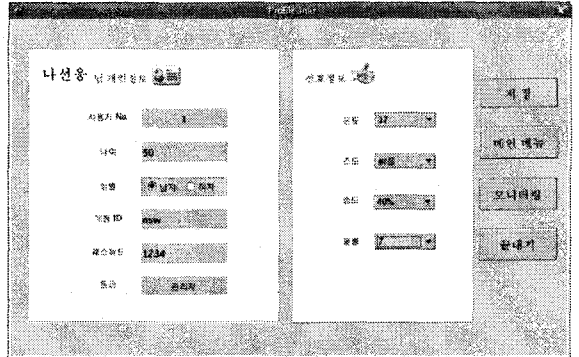


그림 7 개인 프로파일 설정 화면

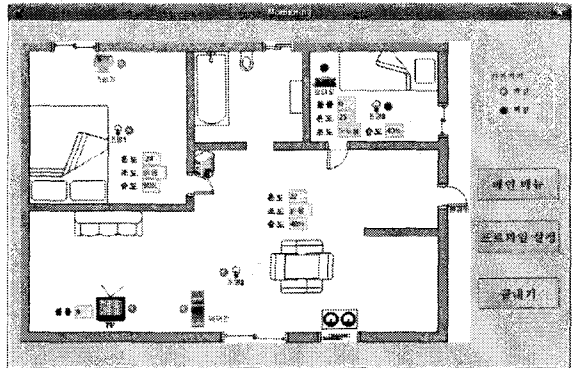


그림 8 맥내 위치별 모니터링 화면

5. 결론

본 논문에서는 홈 네트워크 환경에서 무선 센서 네트워크를 이용하여 지능화된 홈 네트워크 서비스를 제공하는 서비스 모델을 제안하였다. 제안된 서비스 모델은 맥내 환경데이터 센서들을 위한 고정 센서 노드들을 배치하고, 사용자에게는 식별 노드를 부착하였다. 이들 무선 노드들로부터 수집된 상황정보(환경 데이터, 사용자 정보) 등을 이용하여 사용자 선호도에 따라 맥내 가전기기를 자동 제어 및 자동 설정하는 서비스를 제공하였다. 제안된 방식의 타당성 검증 및 효율성을 보여주기 위해 서비스 시나리오를 제시하였다.

또한 맥내와 사용자에게 배치된 무선 센서노드로부터 실시간으로 측정 데이터의 수집, 저장 및 가공을 위해 데이터베이스를 설계하고 구축하였다. 환경데이터 센서로부터 수집되는 데이터들을 관리하는 데이터베이스를 홈 서버에 설계 구현하였고, 사용자 선호도 프로파일을 등록

관리 및 서비스하는 프로파일 처리 모듈을 개발하였다.

그림 7은 사용자에 대한 개인 정보와 선호정보를 입력 및 수정하는 화면 예이다. 또 그림 8은 맥내 현재 상태와 제안된 서비스 모델은 리눅스 환경에서 MySQL 데이터베이스가 내장된 홈 서버와 TinyOS가 탑재된 센서 노드들을 사용하여 구현하고 제시된 서비스 시나리오들을 테스트하였다.

향 후 홈 서버에 센서 노드들로부터 수집된 데이터의 신뢰성 향상을 위해 수집된 데이터를 필터링하고, 손실된 데이터를 추론하는 알고리즘을 개발할 예정이다. 또한 각 센서 노드로부터 수집된 실시간 상황정보를 그래픽으로 표현하고, 이의 히스토리를 분석하여 사용자의 행동패턴을 예측 추론하는 기능을 연구할 예정이다.

6. 참고문헌

- [1] Raúl Jimeno et.al " An architecture for the personalized control of domotic resources " , Proceedings of the 2nd European Symposium on Ambient Intelligence, pp. 51-53, 2004.
- [2] 안동인, 김명희, 주수종, "ON/OFF 스위치와 센서를 이용한 홈 거주자의 위치추적 및 원격 모니터링 시스템", 한국정보과학회 논문지 제12권 6호, pp66-77, 2006.2. ISSN 1229-7712
- [3] TinyOS, <http://www.tinyos.net/>
- [4] Moteiv사, <http://www.moteiv.com>
- [5] nesC, <http://nesc.sourceforge.net/>
- [6] QT, <http://www.trolltech.com>
- [7] MySQL, <http://www.mysql.com/>
- [8] F. Wang, P. Liu, Temporal Management of RFID Data. In Proc. of the 31st International Conference on Very Large Data Bases (VLDB'05). Trondheim, Norway; August 30-September 2, 2005. p1128-1139.
- [9] TinyOS korea community Forum, <http://www.tinyos.or.kr/>
- [10] 안동인, 김명희, 주수종, "ON/OFF 스위치와 센서를 이용한 홈 거주자의 위치추적 및 원격 모니터링 시스템", 한국정보과학회 논문지 제12권 6호, pp66-77, 2006.2. ISSN 1229-7712
- [11] Zhang, T., An Architecture for Building Customizable Context-Aware Applications by End-Users, Pervasive 2004 Doctoral colloquium.
- [12] 신춘성, 우운택, "스마트 홈에서의 사용자간 의도충돌 해결," 한국정보처리학회(KIPS04), vol.11, pp. 937-940, 2004.