

상황인식 기술을 적용한 홈오토메이션 시스템

테스트베드 구현

이승훈 조용곤 신동규 신동일
세종대학교 컴퓨터 공학과

shlee@gce.sejong.ac.kr, ygon21c@gce.sejong.ac.kr, shindk@sejong.ac.kr dshin@sejong.ac.kr

Implementation of Home-Automation System Test Bed based on Context-Awareness Technology

Seunghun Lee Yonggon Cho DongKyoo Shin Dongil Shin
Department of computer Engineering, Sejong University

요 약

최근 국내에서 홈네트워크에 관한 제품과 정책 및 제도의 개발이 다양하게 진행 중이고 대학에서의 관련분야 연구도 활발하지만 몇 가지 문제점이 있다. 첫째로 실험환경을 구축하는데 어려움이 있다. 홈네트워크 시스템에 사용되는 미들웨어나 게이트웨이 및 센서장비의 성능, 또는 지능형 서비스를 위한 알고리즘을 테스트하기 위해서는 실제 환경과 가까운 실험환경을 구축해야하는데, 비용과 시간의 측면에서 많은 부담이 있다. 둘째로 지능형 서비스를 위한 실험을 하기 위해서는 필수적으로 실 사용자가 실험환경에서 생활을 하여 실험을 하여야하는데 이 또한 많은 어려움이 있다. 그래서 본 논문에서는 이러한 문제점들을 해결하기 위해 상황인식 홈오토메이션 테스트베드를 구축하여 비교적 적은 비용으로 홈 네트워크 시스템을 테스트할 수 있는 방법을 제안한다. 2장에서는 홈게이트웨이에 대한 기본적인 배경지식에 대해 알아보고 본 연구에서 사용된 홈네트워크 미들웨어에 관해서 설명한다. 3장에서는 실험에 사용된 게이트웨이, 테스트베드, 통신방법에 대해서 설명한다. 5장에서는 실제로 홈오토메이션 시스템 테스트베드를 구현하고 5장에서 결론 및 향후연구에 관해 논한다.

1. 서론

최근 국내에서 홈네트워크에 관한 제품과 정책 및 제도의 개발이 활발히 진행 중이다. 제품 동향에 관해서 알아보면 다음과 같다. KT의 아파트단지 솔루션 'U-APT'는 아파트 냉난방, 공조 설비 같은 공용설비를 365일 안정적으로 운영 및 관리하고, 통합센터 구축 등으로 안전한 주거환경과 생활편의 서비스를 제공한다. 삼성전자의 홈네트워크 솔루션인 홈비타는 집 안팎에서택내의 다양한 디지털 기기의 제어 및 침입감시, 단지 커뮤니티 등을 이용할 수 있다[1]. 위드 솔루션은 홈네트워크와 보안 기술을 접목하여 디지털 컨버전스 제품으로서 유비쿼터스 솔루션의 중심축을 담당할 홈 서버 역할까지 수행하도록 관련 발명특허와 실용신안을 3개 이상 출원하였다[2]. 오픈테크놀러지는 유비쿼터스 홈네트워크 플랫폼을 위한 지능 융합형 가입자망 장비개발과 발광 다이오드(LED)시스템 개발에 착수중이다[3]. 에이

블 스토어는 홈서버, 웹서버, 사진서버 등으로 쓰이는 네트워크 스토리지 제품을 출시했으며 서울통신기술은 국내 최초로 '지그비(Zigbee)' 기술을 적용한 네트워크 디탈 Door Lock 시스템을 출시했다[4][5]. MDS네트웍스는 차량용 네트워크, 홈네트워크, 디지털 정보가전에 적용될 핵심 모듈 및 응용 제품을 개발하고 있다[6]. 정책 및 제도에 관해서는 일정기준 이상의 구내 통신 설비를 갖춘 공동 주택이나 업무시설에 대해 정부가 인정해주는 제도로서 기술기준에 따라 특등급 및 1,2,3등급을 부여하는 중앙전파관리소의 초고속 정보 통신건물 인증이 있으며 국토해양부는 2009년 3월 28일 지정한 '유비쿼터스 도시의 건설 등에 관한 법률'의 후속조치로, 시행령 제정안을 마련해 6월17일 입법예고를 했다[7][8]. 지식경제부는 2008년에 지능형 홈 네트워크 산업발전 분야의 연구개발(R&D)에 264억원을 투입하는 지원 전략을 발표했다[9]. 이렇듯 제품과 정책 및 제도의 개발이 다양하게 진행 중이고 대학에서의 관련분야 연구도 활발하지만 몇 가지 문제점이 있다. 첫째로 실험환경을 구축하는데 어려움이 있다. 홈네트워크 시스템에 사용되는 미들웨어나 게이트웨이 및 센서장비의 성능, 또는 지능

본 연구는 서울시 산학연 협력사업(JP091000)의 지원에 의해 수행되었음.

형 서비스를 위한 알고리즘을 테스트하기 위해서는 실제 환경과 가까운 실험환경을 구축해야하는데, 비용과 시간의 측면에서 많은 부담이 있다. 둘째로 지능형 서비스를 위한 실험을 하기 위해서는 필수적으로 실 사용자가 실험환경에서 생활을 하여 실험을 하여야하는데 이 또한 많은 어려움이 있다. 그래서 본 논문에서는 이러한 문제점들을 해결하기 위해 상황인식 홈오토메이션 테스트베드를 구축하여 비교적 적은 비용으로 홈 네트워크 시스템을 테스트할 수 있는 방법을 고안한다.

2. 배경연구

2.1 홈 게이트웨이란?

아래 그림 1과 같이, 홈 네트워크 서비스를 제공하는 홈 게이트웨이는 외부의 인터넷으로 유/무선 접속되어 있고, 가전(세대)의 안전과 유통및 인프라 시설에 관계되는 정보들을 수집하기 위하여 각종 주차관제, 무인택배, CCTV, 무인경비, 원격검침 기기들과 접속되어 있기 때문에 가정의 공간에 설치되어 있는 기기들과 새로운 출현 기기들의 이력을 등록 관리 한다. 홈네트워크 서비스가 외부의 이동통신이나 무선통신과 유/무선 방송은 물론 u-자동차, u-로봇과 같은 u-기기들에 연동되어 인터넷 포털 서비스나 전자상거래/TV상거래/전화상거래가 이루어지도록 지원하는 컴퓨터로서 기기 간의 콘텐츠를 공유할 수 있도록 차세대 디지털 정보가전 기기들의 기능 및 성능과 결합하는 역할도 수행한다[10].

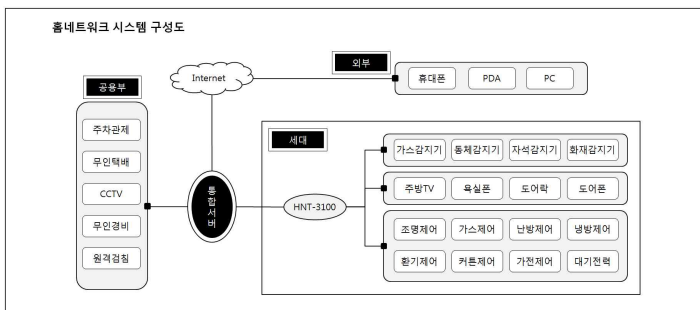


그림 1. 홈서버와 홈네트워크의 구성

이로써 홈네트워크 기술은 가정 내의 모든 정보가전 기기와 전기에너지/가스 에너지/상하수도/공기/정수/보안/보호 전담기기들이 유/무선 홈 네트워크로 연결되어 가족의 구성원이나 가족이 허가한 방문객 누구든지 기기, 시간, 장소에 구애 받지 않고 다양한 서비스를 제공받도록 지원하기 위하여 오늘날 유비쿼터스 IT 기술 개념과 사용자중심의 행복서비스 철학을 수용하여 고객의 가치를 창조해 나아가고 있다. 현존하는 제품들 중에서 홈게이트웨이는 가정 내 홈네트워크 시스템을 가능하게 하는 유비쿼터스 컴퓨터 기기의 규격을 반영하고 있고, 유/무선 통합, 일부는 음성 데이터 통합을 위한 기능도 포함하는 제품이 출시되고 있으며, 가정에 있는 각종 미디어의 정보들을 저장, 통합, 분배, 변환, 삭제하는 시스템으로서 유선가입자 망의 연결, 무선 LAN의 연결, 무선

PAN(Personal Area Network)의 연결까지 수용해 나아가고 있다.

2.2 상황인식 홈오토메이션 미들웨어 아키텍처

현재까지 스마트 홈에 관한 많은 연구가 다양한 영역에서 진행되고 있다. Ranganathan과 Campbell은 유비쿼터스 환경에서의 상황인식 에이전트에 대한 미들웨어를 제시했다[11]. 그들은 유비쿼터스 환경이 상황인식을 위한 미들웨어를 제공하여야 하는 점에 관한 연구를 제시하였고, 그 미들웨어는 상황 정보(context information)에 대한 획득을 에이전트가 담당하여 에이전트에 의하여 변화되는 상황을 감지하는 연구를 진행하였다. 하지만 상황(context)은 일반적인 데이터와는 달리 정의하고 사용하기가 어렵다.

2.3 시스템 아키텍처

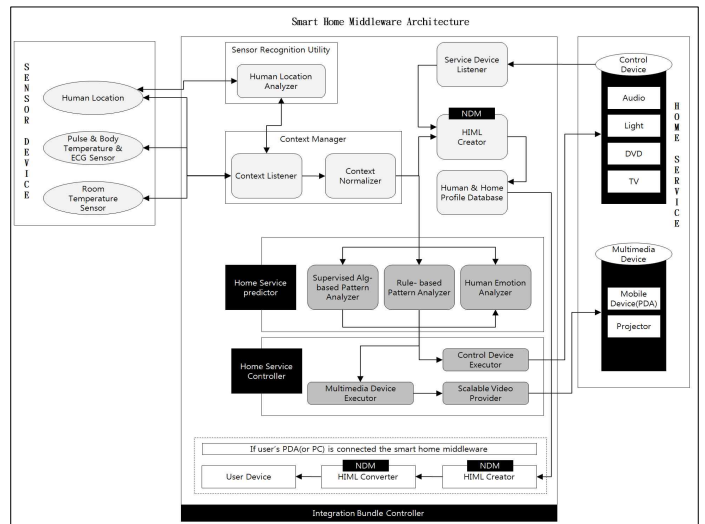


그림 2. Smart Home Middleware Architecture

그림 2는 본 연구팀에서 기 제시한 스마트 홈 서버에서 동작하는 프로세스에 대한 흐름도이며 사용자를 위한 가전 서비스 예측을 위하여 감독형 패턴분석기와 규칙기반 패턴분석기 그리고 사용자감정 분석기가 설계되어 있다[12].

2.4 컨텍스트(Context)란?

유비쿼터스 컴퓨팅 환경에서 많이 인용되는 컨텍스트란 각종 센서장치로부터 획득된 사용자 생체 데이터 및 환경 데이터를 의미한다.

3 상황인식 홈오토메이션 시스템 테스트베드

3.1 상황인식 홈오토메이션 게이트웨이 구조

본 논문에서 설계한 상황인식 홈게이트웨이를 위한 메인보드에 대한 사양은 다음과 같다.

표 1. 홈게이트웨이 스펙

항목	사양
CPU	VIA C7 1.3GHz NanoBGA2
RAM	1G Byte (DDR2 533MHz)
Network	Ethernet 100M (RJ45 Port)
Graphics	Integrated UniChrome Pro AGP graphics with MPEG-2 가속 / VT1625M HDTV Encoder
Audio	VIA VT18 8-channel with AC'97 codec
Back Panel I/O Ports	PS/2 Mouse/Keyboard, COM, VGA out, USB 2.0 4개, RJ45, S-Video out, RCA/SPDIF out, Line-out, Line-in, Mic-in
IDE	UltraDMA 133/100 connector
Power	ATX 20-pin connection
Form Factor	17 cm x 17 cm
OS	Windows XP Embedded, Embedded Linux, Win CE

위의 사양을 구비한 메인보드의 layout은 다음과 같다.

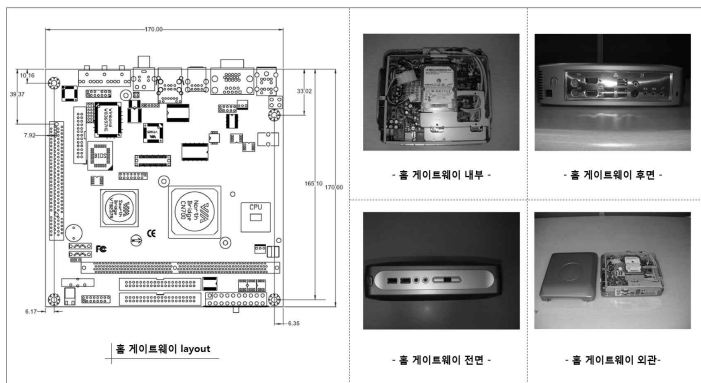


그림 3. 홈 오토메이션 시스템 구현에 사용된 홈게이트웨이

3.2 상황인식 홈오토메이션 시스템 테스트베드 구조



그림 4. 홈오토메이션 미들웨어 테스트베드

그림 4는 본 논문에서 구현한 상황인식 홈오토메이션 프로그램을 테스트하기 위해 주문제작한 테스트베드이다. 본 논문에서는 Ubi-House라 부르도록 한다. Ubi-House의 크기는 960x730x800m(가로x세로x높이)이고 전원은 220VAC를 사용한다. 그리고 테스트베드 내부에는 일반적인 가정기구들인 현관문 전등, 냉장고, 가스차단기, 환풍기, 에어컨, 커튼, TV가 있다[13]. Ubi-House와 상황인식 홈오토메이션 프로그램이 탑재되어 있는 홈 게이트와의 통신은 ZigbeX 모트(Mote)를 사용한다. 홈게이트웨이에 연결되어 Ubi-House에 장착되어 있는 ZigbeX망과 통신하기 위한 모트의 이름은 ZigbeX Base 모트이다. 구성요소는 수신노드 1개와 USB_ISP 프로그램 보드로 이루어져 있다. Ubi-House에 장착되어 있는 ZigbeX망은 움직임검출 모트, 문열림 검출 모트, 가스차단 모트, 전등 제어 모트, 가스검출 모트, 커튼 제어 모트로 이루어져 있다.

3.3 테스트베드에 사용된 통신 방법

Ubi-House와 상황인식 홈오토메이션 프로그램이 탑재된 홈게이트웨이가 통신하는 방법이 홈게이트웨이에 연결된 ZigbeX Base 모트가 Ubi-House에 장착된 ZigbeX망과 통신한다. 본 절에서는 ZigbeX Base 모트와 ZigbeX망과 통신하는 과정에서 좀 더 세부적인 사항으로 패킷 구성에 대해 알아보도록 하겠다. 패킷은 크게 ZigbeX Base 모트에서 ZigbeX 망으로 명령을 보내는 "Command Packet"과 ZigbeX망에서 ZigbeX Base 모트로 데이터를 보내는 "Data Packet"이 있다.

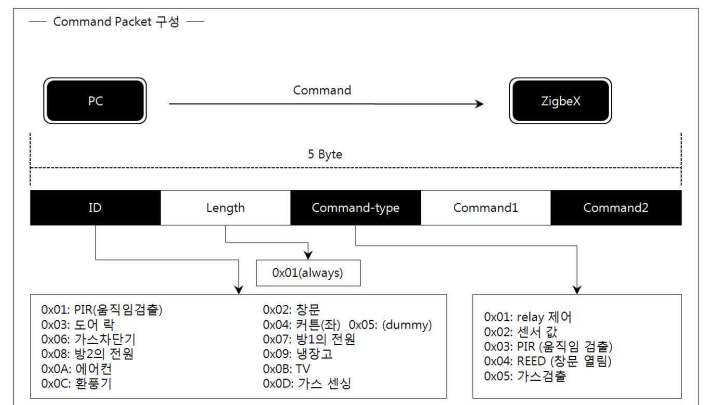


그림 5. Command Packet 구성

그림 5는 ZigbeX Base 모트에서 ZigbeX 망으로 명령을 보내는 "Command Packet"을 설명하는 그림이다. Command Packet은 총 5 Byte의 길이를 가지고 있으며 Packet의 구성은 ID, Length, Command-type, Command1, Command2로 구성되어 있으며 각각 1 Byte의 길이를 차지한다.

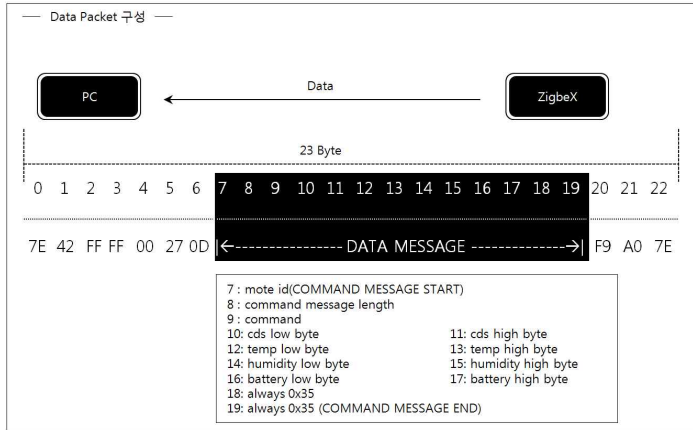


그림 6. Datapacket 구성 - 1

그림 6은 ZigbeX망에서 ZigbeX Base 모드로 데이터를 보내는 "Data Packet"에 해당하는 그림이다. 총 23 Byte의 길이이며 0에서부터6까지 그리고 20에서22까지의 값은 고정된 값이다. 7에서부터 19까지의 데이터를 설명하면 다음과 같다. 7번은 해당 데이터 패킷을 보낸 Ubi-House안에 장착된 모트의 ID값이다. 8번은 홈게이트웨이로부터 받은 커맨드의 패킷의 길이이다. 9번은 홈게이트웨이로부터 받은 커맨드이다. 10과 11번은 조도 값이 온다. 12번과 13번은 온도 값이 오고, 14번과 15번은 습도 값이 온다. 16번과 17번에는 배터리 상태 값이 온다. 18번과 19번 값은 고정된 값이며 19번은 커맨드 메시지가 종료 됐다는 것을 의미한다.

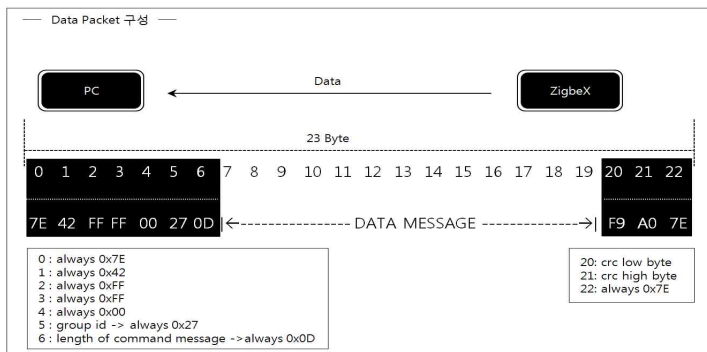


그림 8. DataPacket구성 - 2

그림 8은 그림 3-4와 마찬가지로 ZigbeX망에서 ZigbeX Base 모드로 데이터를 보내는 "Data Packet"에 해당하는 그림이다. 총 23 Byte의 길이이며 0에서부터6까지 그리고 20에서22까지의 값은 고정된 값이 오는데 이 고정된 값으로 무엇이 오는지를 보여주고 있다.

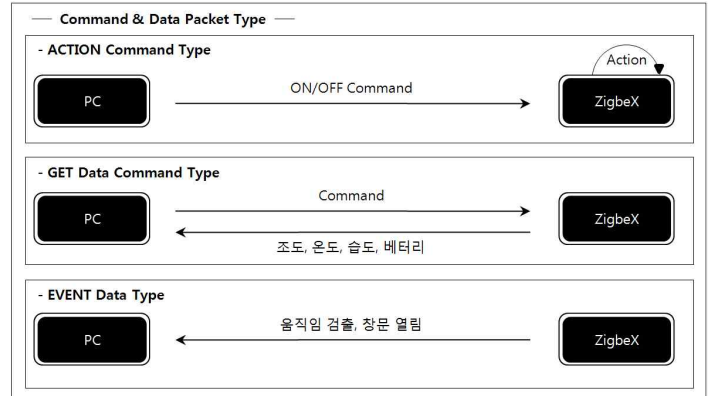


그림 7. Command & Data Packet Type

그림 7은 본 논문에서 구현한 프로그램과 테스트베드 사이에 오고가는 커맨드와 그에 해당하는 데이터의 유형을 보여주고 있다. 첫째로 방 거실등의 전등을 키고 끄거나, 또는 에어컨을 키고 끄는 명령인 가전기기 ON/OFF 명령이 있다. 둘째로 ZigbeX망에 설치된 모트들에서 Ubi-house 내부의 조도, 온도, 습도, 배터리 값을 읽어오는 명령이다. 셋째로 Ubi-house내부에 어떤 움직임이 발생하거나 또는 창문이 열려있을 때 Ubi-house에서 홈게이트웨이로 데이터를 보내는 유형이다.

4. 상황인식 홈 오토메이션 시스템 프로그램 구현

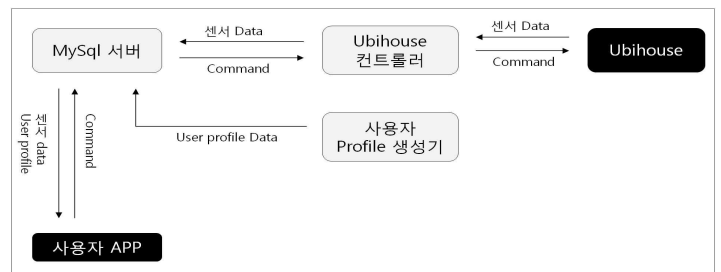


그림 9. 상황인식 홈오토메이션 시스템 테스트베드 메커니즘

제4장에서는 실험환경과 기 설계된 홈네트워크 시스템 아키텍처를 가지고 실제로 상황인식 홈오토메이션 시스템 테스트베드를 구현한다. 그림 9는 구현된 프로그램이 실험환경에서 실행되는 프로그램 메커니즘에 대한 그림이다. 프로그램의 외부적인 요소로 Ubi-house와 MySQL서버가 존재하며 상황인식 홈오토메이션을 위한 세부프로그램으로 Ubi-house컨트롤러, 사용자 Profile 생성기, 사용자 애플리케이션이 있다. 메커니즘의 흐름을 살펴보면 사용자 애플리케이션에서 센서데이터를 확인하겠다는 커맨드를 내리면 그 명령은 MySQL서버로 전송되고 MySQL서버에서는 해당 명령을 로그 기록에 남김과 동시에 Ubi-house컨트롤러로 전송한다. 그리고 Ubi-house컨트롤러는 센서데이터수집 커맨드를

Ubi-house로 보내며 Ubi-house는 해당모트에서 센서데이터를 수집한 후 Ubi-house로 센서데이터를 다시 전송한다. Ubi-house컨트롤러는 Ubi-house로부터 받은 센서데이터를 MySql서버로 전송하고 MySql서버는 전송받은 센서데이터를 데이터베이스에 저장함과 동시에 사용자 애플리케이션으로 센서데이터를 보낸다. 사용자 애플리케이션은 전송받은 센서데이터를 가공하여 사용자에게 보여준다. 상황인식 홈오토메이션 시스템 테스트베드 메커니즘에 나와 있는 세부프로그램을 하나씩 설명해 보면, 사용자 Profile생성기는 본 논문에서 실험환경의 특성상 실제로 사용자로부터 데이터를 수집하고 또는 사용자가 특정 공간 안에 홈네트워크 서비스를 이용하는 과정에서 추출되는 데이터를 수집할 수 없으므로 가상의 데이터인 User profile data를 임의로 생성해주는 프로그램이다. Ubihouse 컨트롤러는 상황인식 홈오토메이션 게이트웨이에 탑재된 프로그램으로써 실질적으로 Ubi-house를 컨트롤 하고 Ubi-house로부터 데이터를 수집해오는 역할과 각종 데이터를 가공해서 사용자 애플리케이션으로 보낸다거나 또는 사용자 애플리케이션에서 사용자가 수동으로 실행한 가전기기 컨트롤 명령을 받아서 Ubi-house를 제어하는 역할을 한다. 사용자 애플리케이션은 Ubi-house에서 수집된 데이터 그리고 사용자 Profile생성기에서 생성된 데이터 등을 가공하여 분석된 결과를 사용자에게 보여주며 사용자의 가전기기 컨트롤 명령을 받을 수 있는 인터페이스가 있다. 그리고 사용자가 수동으로 입력하는 데이터를 처리해주는 역할을 한다. 또한 Ubi-house의 상태를 모니터링 할 수 있는 서비스를 제공한다.

분석되어 가전기기 ID부분과 Command명령 부분으로 나누어지게 된다. ID의 가전기기 Command명령을 분해하고 이전에 기록되어있는 해당 가전기기의 상태정보와 비교하여 수행하여야할 제어명령이라는 판단이 내려진다면 명령패킷을 Ubihouse 컨트롤러로 전송해 사용자가 수동으로 제어한 가전기기 명령을 수행한다. 그림 11은 Ubihouse 컨트롤러의 소스코드 중 일부를 클래스 다이어그램으로 표현한 것이다. 표현된 클래스 다이어그램은 사용자 애플리케이션에서 Ubi-house로 전송되는 명령패킷과 Ubi-house에서 사용자 애플리케이션으로 전송되는 데이터패킷을 분석하고 처리해주는 클래스이다. 그림의 좌측에 있는 TransData클래스부터 StreamServer클래스가 이 역할을 수행하는 핵심 클래스로써 각각의 패킷을 본 논문의 “3.3장 테스트베드에 사용된 통신방법”에서 분석한 내용에 근거하여 처리하고 있다.

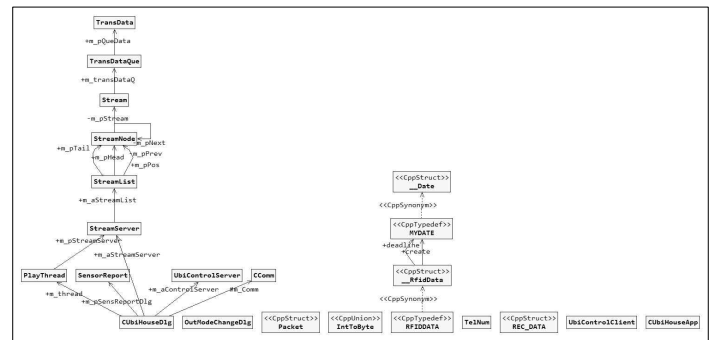


그림 11. Ubihouse 컨트롤러 클래스 다이어그램

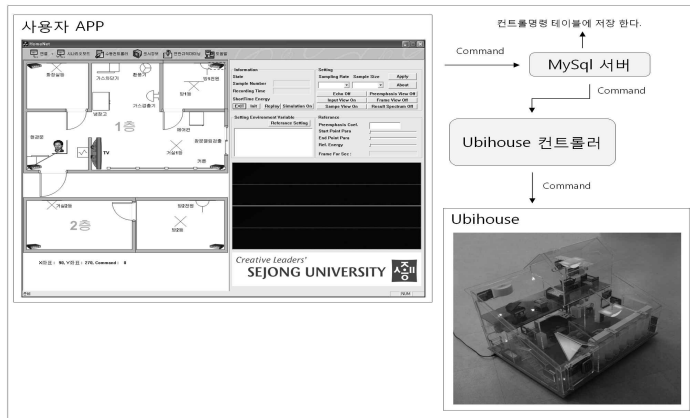


그림 10. 가전기기 컨트롤 메커니즘

그림 10은 사용자 애플리케이션을 사용해 Ubi-house를 컨트롤하고자 할 때 이를 처리해주는 메커니즘에 대한 그림이다. 사용자 애플리케이션에서 상단에 있는 수동컨트롤러 버튼을 클릭하면 Ubi-house를 수동으로 제어할 수 있는 인터페이스가 출력되며 그 인터페이스로 Ubi-house를 수동으로 제어할 수 있다. 사용자가 Ubi-house내부의 한 가전기기를 제어하는 명령을 내리면 해당 패킷은 MySql서버로 전송이 되어 가전기기 컨트롤명령 테이블에 저장된다. 그리고 해당 패킷은

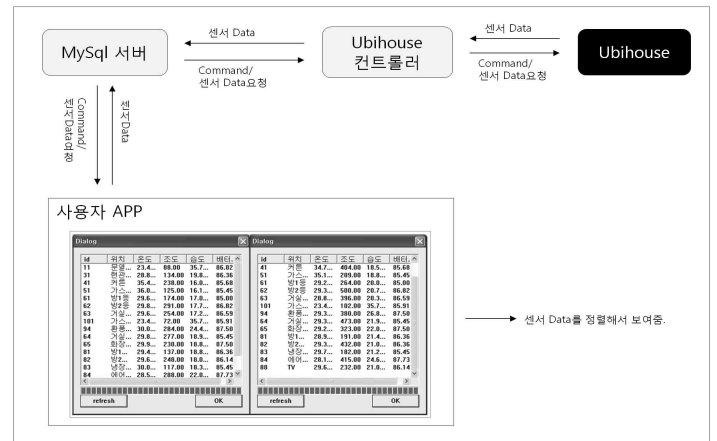


그림 12. 센서 Data 뷰어

그림 12는 Ubi-house에서 센서데이터 정보를 받아와 사용자 애플리케이션에서 사용자에게 보여주는 흐름과 실제 수행 결과를 보여주고 있다. 환경컨텍스트 정보를 요청하는 5byte길이의 명령패킷이 Ubi-house로 전송되고 그에 대응하는 센서정보인 23byte 길이의 센서데이터패킷이 사용자 애플리케이션으로 넘어오면 사용자 애플리케이션은 이를 분석하여 사용자가 인식하기 쉬운 형태로 보여준다. 그림 12하단부분에 있는 그림이 실제로

출력된 홈내의 환경 컨텍스트 정보이며 속성은 id, 위치, 온도, 조도, 습도, 배터리로 나누어져 있고 그에 맞는 데이터들이 정렬되어 출력되고 있다.

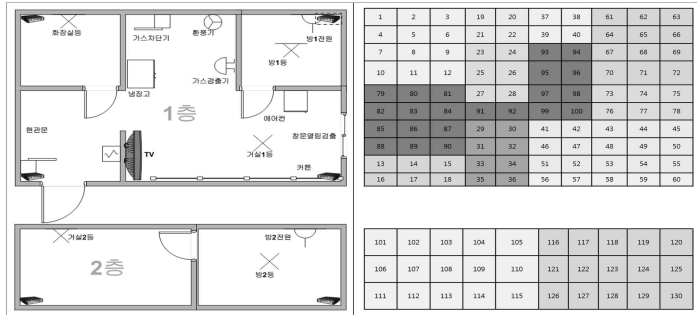


그림 13. Ubi-house 섹터 구분

```

287 void ChromeView::UCLIDistance (user, int userY)
288 {
289     //1번 온도 (10, 382)
290     //2번 온도 (10, 30)
291     //3번 온도 (425, 10)
292     //4번 온도 (425, 382)
293     //5번 온도 (10, 425)
294     //5번 온도 (425, 425)
295
296     int Moto_P_1[] = {10, 10, 425, 10, 425};
297     int Moto_P_2[] = {425, 10, 382, 425, 425};
298     double Moto_Dist[];
299
300     int cha1;
301     int cha2;
302     int num;
303     int Moto_Flag;
304
305     for(int i = 0; i < 5; i++)
306     {
307         cha = Moto_P_1[i] + userX;
308         cha2 = Moto_P_2[i] + userY;
309         Moto_Dist[i] = sqrt((cha - cha2) * (cha - cha2));
310     }
311     Moto_Flag = 1;
312
313     if(Moto_Dist[0] < Moto_Dist[1] && Moto_Dist[0] < Moto_Dist[2] && Moto_Dist[0] < Moto_Dist[3] && Moto_Dist[0] < Moto_Dist[4])
314         Moto_Flag = 1;
315     else if(Moto_Dist[1] < Moto_Dist[0] && Moto_Dist[1] < Moto_Dist[2] && Moto_Dist[1] < Moto_Dist[3] && Moto_Dist[1] < Moto_Dist[4])
316         Moto_Flag = 2;
317     else if(Moto_Dist[2] < Moto_Dist[0] && Moto_Dist[2] < Moto_Dist[1] && Moto_Dist[2] < Moto_Dist[3] && Moto_Dist[2] < Moto_Dist[4])
318         Moto_Flag = 3;
319     else if(Moto_Dist[3] < Moto_Dist[0] && Moto_Dist[3] < Moto_Dist[1] && Moto_Dist[3] < Moto_Dist[2] && Moto_Dist[3] < Moto_Dist[4])
320         Moto_Flag = 4;
321     else if(Moto_Dist[4] < Moto_Dist[0] && Moto_Dist[4] < Moto_Dist[1] && Moto_Dist[4] < Moto_Dist[2] && Moto_Dist[4] < Moto_Dist[3])
322         Moto_Flag = 5;
323
324     switch(Moto_Flag)
325     {
326     }
327
328     void ChromeView::updateRecord()
329 }
    
```

그림 14. 사용자 위치 계산 소스코드 일부

상황인식 홈게이트웨이가 구현되기 위한 필수 요소 중 하나는 사용자의 위치를 인식하는 시스템이다. 일반적으로 사용자의 위치를 인식하기 위해 사용자의 위치를 인식하는 특정한 센서장비들이 사용된다. 본 논문은 테스트베드의 상황을 고려하여 위치인식 센서장비들이 그림 13의 좌측 그림처럼 총6개의 위치인식 장비가 설치되어 있다고 가정한 후 구현하였다. 가정된 위치인식 장비는 점선 사각형 박스 안에 있는 모트가 위치인식 베이스 모트이고 사용자의 위치는 나머지 5개의 장비에 가까운 곳으로 인식된다. 그림 14는 사용자의 위치와 위치인식 모트들 간의 거리를 계산해주는 소스코드이고 유클리디언 거리계산 알고리즘을 기반으로 구현되어 있다. 그림 13의 우측에 있는 Ubi-house 섹터구분은 효율적인 홈서비스를 위하여 가전기기 서비스 중심으로 섹터를 구분하였다. 가령 화장실 전등의 제어가 중심이 되는 구역은 1에서 12까지 또는 커튼의 제어가 중심이 되는 구역은 56부터 60까지로 구분되어있다.

5 결론 및 향후 연구

본 논문은 홈네트워크에 관련된 미들웨어, 게이트웨

이, 센서장비 등을 실험하기 위해서 실제환경에 가까운 실험환경을 구축하기에는 비용과 시간이 많이 든다는 문제점을 해결하기 위해 가상의 테스트베드를 구축하는 방법을 보였다. 홈오메이션 테스트베드에 적용할 미들웨어는 본 연구팀에서 기 설계한 스마트 홈 서버 아키텍처를 사용하였고, 사용된 스마트 홈 서버 아키텍처는 사용자를 위한 가전 서비스 예측을 위하여 감독형 패턴분석기와 규칙기반 패턴분석기 그리고 사용자감정 분석기가 설계되어 있다. 상황인식 홈오메이션 시스템 테스트베드에 구현된 프로그램의 주요기능은 사용자에게 윈도우 기반의 인터페이스를 제공하며 기초데이터의 조회/생성/삭제 수정기능이 있으며 위치인식 기능을 제공한다. 그리고 사용자 애플리케이션에서 테스트베드를 수동으로 제어할 수 있는 기능이 구현되어있다. 하지만 본 논문은 기 설계된 미들웨어와 하나의 홈게이트웨이만을 테스트를 하였다는 단점이 있다. 향후연구에서는 보다 많은 미들웨어와 다양한 스펙의 홈게이트웨이를 테스트해보아야 할 것이다.

참고문헌

- [1] 전자신문, 'KT-삼성전자, U아파트 협력', 2008년 8월 28일
- [2] 위드솔루션 <http://www.withsolution.com>
- [3] 오픈테크놀러지 <http://www.okopen.com>
- [4] 에이블스토어 <http://www.ablestor.com>
- [5] 전자신문, '지그비 적용 디지털 도어록 첫선', 2008년 8월 29일
- [6] MDS네트웍스, <http://www.mdsnet.co.kr>
- [7] 전자신문, 2008년 9월 1일
- [8] 전자신문, '국토부, U시티 자가통신망 구축 허용', 2008년 6월 17일
- [9] 전자신문, '홈네트워크 R&D에 264억원 지원', 2008년 6월 4일
- [10] 현대통신, <http://www.hyundaitel.co.kr>
- [11] Anand Ranganathan, Roy H. Campbell., "A middleware for context-aware agents in ubiquitous computing environments", In ACM/IFIP/USENIX International Middleware Conference, 2003
- [12] JH Choi, DK Shin, DI Shin, Research and "Implementation on the Mobile", IntelligenceControllerforConsumerElectronics - Consumer Electronics, IEEE Transactions on, 2005
- [13] 한백전자, <http://hanback.co.kr>