

## 임베디드 리눅스를 이용한 홈오토메이션 시스템

지준근, 김구남, 이강준  
순천향대학교 정보기술공학부

### Home Automation System Using Embedded Linux

Jun-Keun Ji, Koo-Nam Kim, Kang-Jun Lee  
Division of Information Technology Engineering, Soonchunhyang University

**Abstract** - 본 논문에서는 embedded Linux 환경에서 device driver와 application의 개발에 목적을 두었고, 그 대상 시스템으로 선정된 Home Automation System의 각각의 플랫폼에 대한 하드웨어 설명과 제어용 프로그램을 다룬다.

#### 1. 서 론

임베디드(embedded) 시스템이란 어떠한 장치가 다른 시스템에 의존하지 않고 독립적으로 기능을 수행하는 것으로써 우리 생활에서 쓰이는 각종 전자기기, 가전제품, 제어장치가 단순히 회로로만 구성된 것이 아니라 마이크로 프로세서가 내장되어 있고, 그 마이크로 프로세서가 시스템을 구동하여 특정한 기능을 수행하도록 프로그램이 내장되어 있는 시스템을 가리키는 것이다.

이러한 임베디드 시스템은 시간이 흐름수록 기능이 다양해지고 시스템의 크기가 날로 커져가서 임베디드 시스템을 운용하기 위해서는 이를 적절히 통제할 만한 운영체제가 필요하게 되었다.

21C의 가장 전망이 밝은 분야는 Mobile Networking(이동형 정보시스템)이다. 국내외 전 세계적으로 PC의 한계를 극복하고 이동전화기의 장점을 합치면서 다른 기능을 수용할 수 있는 것을 찾고 있다. 소비자의 최근 요구사항은 가벼우면서 쉽게 사용할 수 있는 것이며, 네트워크(인터넷)의 기능은 필수가 되었다. 즉 Mobile Networking으로 대표가 된다.

이러한 요구에 대해 가장 강력하게 발전 가능성이 풍부한 것은 바로 Linux이다. 외국의 경우에는 많은 곳에 적용이 완료가 되어 있고 계속적으로 새로운 분야를 개척중이다. 또한, 최근에는 CPU를 만드는 업체에서도 상용 OS와 함께 Linux를 같이 적용할 수 있게 해주는 경우가 많이 생겨났다.

Linux는 embedded system 분야에서 많은 성장을 해왔고, 앞으로도 폭발적인 수요에 힘입어서 상당한 분야에 적용될 것으로 보인다. Linux가 가지고 있는 많은 장점이 이제 국내에도 알려지기 시작하였다. 앞으로 보다 많은 개발자와 투자자들의 연구와 관심이 필요한 때라고 생각이 되었기 때문에 논문주제를 선정하였다.

#### 2. 홈오토메이션 시스템의 구성

본 논문에서는 embedded Linux 환경에서 device driver와 application의 개발에 목적을 두었고, 그 대상 시스템으로 선정된 Home Automation System의 각각의 플랫폼에 대한 하드웨어 설명과 제어용 프로그램을 다룬다.

Home Automation System에 쓰인 하드웨어 부분에서의 메인부분은 SrtongARM SA1110을 사용한 EZboard를 사용하였고, 각각의 플랫폼의 동작을 위한 외부회로는 따로 설계하여 사용하였다. embedded Linux 환경에서의 핵심이라 할 수 있는 Linux kernel은 2.4.10의 버전을 사용하였으며 모든 프로그램은 Linux 환경에서 작성되었다.

본 논문에서 가장 중요하고 많은 시간을 소비한 부분은 device driver의 작성이었으며, Srtong ARM SA1110에

올라가 있는 Linux kernel(ver2.4.10)을 이용하여 device driver를 작성한 이후 이 device driver를 바탕으로 application을 작성할 수 있었다.

본 논문에서는 두 개의 device driver와 세 개의 application이 있으며, 각각의 application에는 그에 해당하는 구동부가 있어 사용자가 Linux 환경의 minicom을 이용한 target system과의 통신상에서 원하는 application을 입력하게 되면 그에 상응하는 동작을 하게 되어 있다.

아래의 구성도는 Linux kernel이 올라가 있는 상태의 StrongARM SA1110과 각각의 플랫폼들과의 상호 관계를 나타낸다. 각각의 플랫폼들은 그에 해당하는 외부회로를 구성하며 동작은 GPIO 포트를 이용하여 구동하게 된다. 서보모터를 이용하여 창문 및 도어락을 구동하고 적외선 센서의 입력에 의해 외부 상태를 점검하고 부저 및 LED가 동작하며, 더미스터의 출력을 AD 변환을 해서 사용자가 입력한 설정온도와 비교하여 발열기능 또는 냉각기능을 수행함으로써 홈오토메이션 시스템을 구현한다.

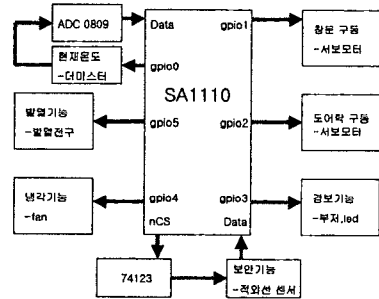


그림 1 시스템 전체 구성도

#### 3. Embedded Linux 개요

##### 3.1 커널(kernel)

OS(Operating System)란 시스템 자원을 효율적이고 쉽게 쓰도록 하기 위한 소프트웨어를 말하는데, 이러한 OS의 핵심적인 요소를 커널이라 한다.

Kernel의 주요 역할로는 프로세서 관리, 메모리 관리, H/W 관리, interrupt 관리 등이 있다.

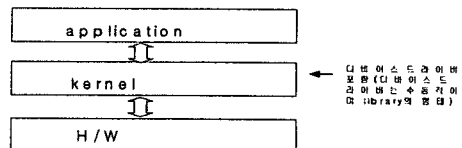


그림 2 Application 과 Kernel의 H/W 상관관계

## 3.2 부트로더(Boot Loader)

일반적으로 부트로더라 하면 일반 x86리눅스에선 LILO를 많이 사용할 것이다. LILO란 Linux Loader로써 도스나 NT, 리눅스 등 다른 OS를 선택적으로 부팅할 수 있도록 하는 기능을 제공한다. LILO는 하드디스크의 MBR에서 동작이 되는 프로그램으로 OS가 실행할 수 있도록 점프하는 기능을 수행한다.

EZboard에서 사용하는 BLOB이란 부트로더는 플래쉬 블록에서 실행되고 여러가지 다양한 기능들을 수행한다.

먼저 kernel이나 RAMdisk등의 데이터를 Host로부터 SDRAM영역으로 다운로드할 수 있는 기능이 있고 SDRAM에 있는 데이터를 플래쉬 영역으로 writing할 수도 있다. 그리고 kernel이 이미 올라가 있다면 부팅할 수 있는 기능도 같이 제공한다.

## 3.3 디바이스 드라이버

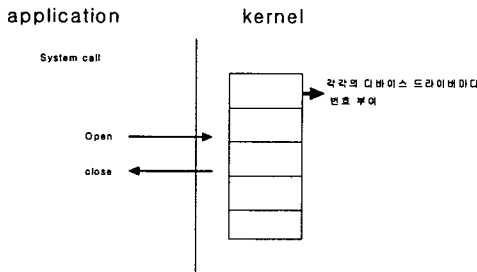


그림 3 디바이스 드라이버와 application의 상관관계

디바이스 드라이버는 커널의 한 부분이다. 커널의 4/5 정도가 디바이스 드라이버이다. 디바이스 드라이버는 수동적이고, H/W를 제어하기 위한 라이브러리라고 볼 수 있다.

디바이스 드라이버의 기본 종류
• Character device driver-연속적으로 데이터 교환
• Block device driver-블럭단위로 데이터 교환
• Network device driver

## 4. Kernel 포팅 및 구동

### 4.1 크로스 컴파일 환경 구축

일반적으로 컴파일러는 자신의 시스템에 맞는 바이너리 코드를 만든다. 예를 들어 x86의 시스템에서 gcc를 사용하여 컴파일을 하게 되면, x86의 바이너리가 생긴다. 그러나 타겟 보드에서 직접 응용프로그램이나 kernel 컴파일을 할 수가 없다. 저장할 수 있는 디스크 공간이 매우 부족하기 때문이다. 그래서, 타겟용 kernel 및 응용프로그램을 개발하기 위하여 호스트 시스템에 타겟용 크로스 컴파일 환경을 구축한다. x86머신에서 ARM용 바이너리 코드를 만들기 위해서 크로스컴파일러가 필요하다.

### 4.2 Kernel 설치 및 패치

C 프로그램 작성을 위해서 ARM에 관련된 헤더 파일을 작성해야 하고 부트로더 BLOB가 Linux kernel의 헤더 파일을 참조하기 때문에 필요하다. 마지막으로 최종적으로 Linux kernel을 올려야 함으로 EZboard 용 Linux kernel이 필요하다. EZboard는 Linux kernel ver 2.4.10으로 설치되어 있다.

## 4.3 RAMdisk 이미지 제작

RAMdisk는 별다른 물리적 장치를 지칭하는 것이 아니라, 메모리의 일부분을 디스크드라이브로 인식시켜 이를 하드디스크처럼 사용하기 위한 것이다. EZboard에는 하드디스크가 없으므로 이 RAMdisk에 의해 부팅이 된다. RAMdisk에는 커널이 완전히 로드된 후, 시스템의 구동에 필요한 init, 유틸리티, 바이너리 파일들이 있어야 부팅이 된다. 여기서 RAMdisk 위에 구성되어 있는 유틸리티들과 라이브러리들을 모아 압축한 것이 RAMdisk 이미지이다.

## 4. 각각의 플랜트 동작

### 4.1 보안기능

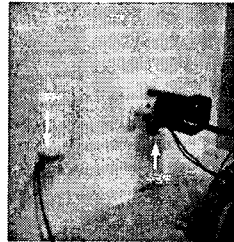


그림 4 적외선 센서 구성(1)

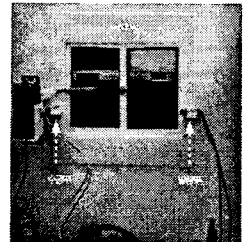


그림 5 적외선 센서 구성(2)

그림 4와 그림 5는 각각 door와 창문 양옆에 침입 감지용 적외선 센서를 부착한 모습이다. 도어나 창문에 있는 수광용 센서의 입력이 끊기게 되면 수광부 출력이 0이 되어 그에 상응하는 동작으로 LED와 부저가 동작을 하게 됨으로서 보안기능을 하게 된다.

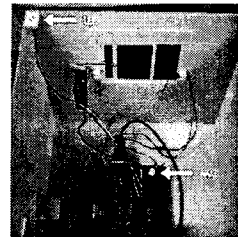


그림 6 경보기능 동작 모습

프로그램을 동작시키기 위해서는 device driver인 driver1.o와 driver2.o, 그리고 application인 temp guard2 auto를 다운로드한 후 guard2를 실행시켜 수광센서와 발광센서 사이에 아무것도 존재하지 않으면 security OK!! 라는 메시지를 출력시키고 만약 그 사이에 무엇인가가 존재하게 되면 emergency라는 메시지를 출력시키게 된다.

```

root@ezboard /root#
root@ezboard /root# ./guard2
SHELL: GP10: 00-07, 017, 024, red led All Off.
EZBOARD DEVICE OPEN

security OK!!
security OK!!
security OK!!
security OK!!
security OK!!
security OK!!
security OK!!
emergency
emergency
emergency
emergency
emergency
emergency
emergency
emergency
emergency
emergency [redbackboard /root# ]
[업데이트] [문 닫기] [무시하기]

```

그림 7 guard2 실행화면

#### 4.2 자동 창문 및 자동 문 잠금 기능

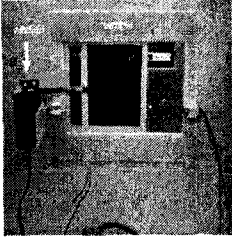


그림 8 자동 창문 동작

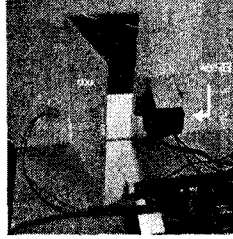


그림 9 자동 도어락 동작

창문과 도어쪽에 서보모터를 연결하였고 창문의 자동 열고 닫음과 문의 자동 닫고 풀림을 서보모터를 이용하여 사용자의 키입력에 의해 그에 상응하는 동작을 함으로서 자동 기능을 구현하였다.

#### 4.3 자동 온도 조절 기능



그림 10 발열 기능 구성

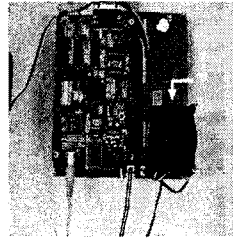


그림 11 냉각 기능 구성

외부 회로에 발열기능의 발열전구와 냉각기능의 팬을 부착하여 구성하였고 온도센서(더미스터)로 현재의 온도 정보를 얻어 이를 사용자가 입력한 온도와 비교하여 그에 상응하는 발열동작 및 냉각동작을 함으로서 사용자가 원하는 온도를 유지할 수 있도록 구현하였다.

프로그램을 실행시키면 가장 먼저 사용자가 원하는 온도를 입력시키고 그 설정온도를 기준으로 냉각기능이나 발열기능을 수행하면서 현재의 온도를 화면상에 출력한다.

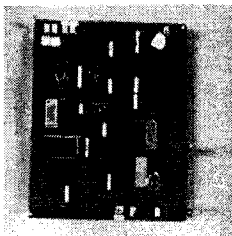


그림 12 홈오토메이션 시스템의 외부회로부

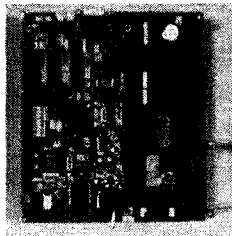


그림 13 메인회로부와 외부회로부

### 5. 결 론

Embedded Linux의 설치후 application을 구동시킬 수 있는 device driver를 작성하고 최종 목표인 application을 작성함으로써 기존 OS에 두 개의 device driver와 세 개의 application을 추가로 탑재하여 Home Automation System의 동작을 수행할 수 있도록 제작하였다.

Embedded Linux를 사용함으로써 Home Automation System

과 같은 복합적인 시스템에서 우수한 성능을 보임을 확인할 수 있었고 더 나아가 이더넷(ethernet)을 이용하여 Home Automation System 기능의 인터넷 제어 및 GUI 환경에서의 기능 제어가 앞으로의 연구과제라고 생각된다.

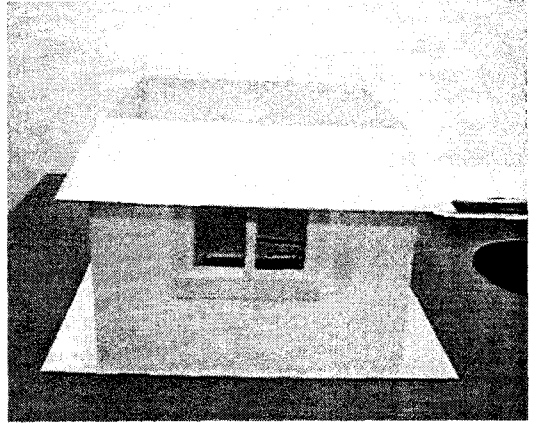


그림 14 완성된 모형(1)

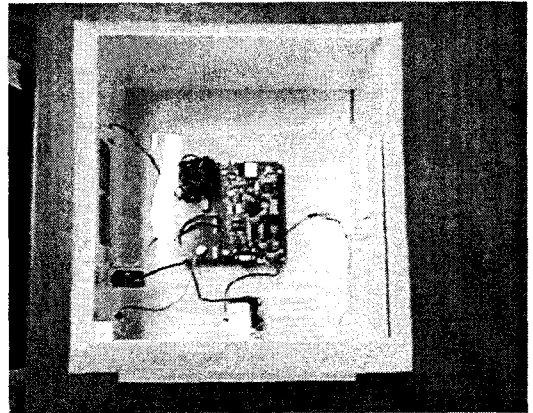


그림 15 완성된 모형(2)

#### [참 고 문 헌]

- [1] (주)제이닷디엔티 편저, "EZboard M01 HowTo 문서".
- [2] 캐머런 뉴햄, 빌 로젠블랫, "베시 셸 시작하기", 한빛미디어, 2001.
- [3] Alessandro Rubini "리눅스 디바이스 드라이버", 한빛미디어, 2000.
- [4] Matt Welsh, Lar Kaufman, "러닝 리눅스", 한빛미디어, 2000.
- [5] Mike Loukides, Andy Oram, "GNU 소프트웨어로 프로그래밍하기", 한빛미디어, 2000.
- [6] Silberschatz, Galvin, Gagne, "응용 운영 체제 개념", 홍릉과학출판사, 2001.
- [7] 이광수, "직렬통신과 Visual Basic을 이용한 Home Automation System 구현", 순천향대학교 정보기술공학부 학사학위논문, 2001.
- [8] 아경산업 자동화연구소 편저, "서보 모터 제어이론과 실습", 성안당, 1998.
- [9] 김구남, 이경준 "임베디드 리눅스를 이용한 홈오토메이션 시스템", 순천향대학교 정보기술공학부 학사학위논문, 2002.