

상용 디지털 TV를 위한 임베디드 리눅스 시스템

문상필*, 김주원**, 이성학**, 서대화*

*경북대학교 전자공학과

**LG전자

e-mail : octor@palgong.knu.ac.kr

Embedded Linux for Commercial Digital TV System

Sang-Pil Moon*, Sung-Hak Lee**, Joo-Won Kim**, Dae-Wha Seo*

*Dept of Electronics, Kyungpook National University

**Display Product Research Lab., LG Electronic Inc.

요 약

기존의 아날로그 TV에 비해 새로운 방식의 디지털 TV는 단순한 영상과 음성의 처리뿐만 아니라, 데이터를 처리 및 저장해야 한다. 그리고 필요에 따라서 데이터를 요구하기 위한 리턴 채널을 관리해야 한다. 이처럼 많은 기능을 동시에 충족시켜줘야 하기 때문에 디지털 TV시스템에서는 운영체제가 필요하다. 현재 출시되는 디지털 TV 시스템은 고비용의 상용 OS를 사용하므로, 가격 경쟁력을 확보하기가 어렵다. 이에 비해 오픈 소스인 임베디드 리눅스를 사용함으로써 가격 경쟁력을 확보할 수 있다. 또한 공개된 장치 드라이버와 응용프로그램을 재사용 가능하며, 오픈 소스 공동체를 통해 문제를 해결할 수 있고, 셸, 파일 시스템을 이용하여 편리한 개발 환경을 제공해 주는 이점이 있다. 본 논문에서는 디지털 TV 시스템을 위해서 임베디드 리눅스를 재설계하고, 기존 디지털 TV 시스템에 이식하였다.

1. 서 론

텔레비전의 다매체 성향에 따라 지상파방송의 경쟁력 강화를 위하여 디지털 방송이 본격적으로 추진되고 있다. 이에 따라 디지털 TV시스템의 수요가 급격히 증가하고 있다.

디지털 TV 시스템을 설계함에 있어 기존의 아날로그 TV 시스템에 사용되는 마이크로 컨트롤러를 디지털 TV 시스템에 적용할 경우 디지털 방송이 요구하는 다양한 수신 능력을 발휘하지 못한다.

다채널·고화질을 추구하던 아날로그 TV시스템과는 달리 디지털 TV시스템은 영상과 음성의 처리뿐만 아니라 데이터 채널로 입력받은 정보를 처리해야 한다. 따라서 고성능 프로세서 및 대량의 데이터를 저장하기 위한 고용량 메모리가 필요하며 이러한 자원을 효율적으로 활용하기 위해 운영체제가 제공되어야 한다.

디지털 TV 시스템에 적용된 운영체제로 pSOS와 같은 상용 운영체제가 있다. 그러나 상용 운영체제를 사용할 경우 TV 시스템을 생산할 때마다 사용료(Royalty)를 지불해야 되므로 생산 비용이 증가한다. 이에 반해 오픈소스

OS인 리눅스를 사용하게 되면 생산 비용을 절감할 수 있다. 뿐만 아니라 소스가 공개된 장치 드라이버와 응용 프로그램들을 재사용 가능하기 때문에 개발 기간을 단축할 수 있으며, 모듈 기능과 셸·파일 시스템을 이용하여 보다 편리한 시스템 개발 환경을 구축할 수 있다. 또한 시스템 개발 중 문제가 발생할 경우 오픈 소스 공동체(Open Source Community)를 통해 해결할 수 있다.

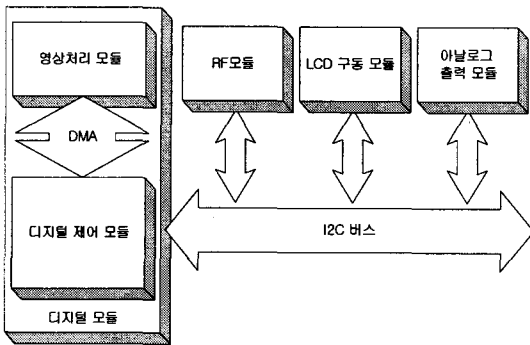
본 논문에서 임베디드 리눅스가 적용된 디지털 TV 시스템은 한국형 디지털 방송 수신기 가능한 LCD 프로젝션 TV이다. 이 시스템에 리눅스를 적용하기 위해 교차 개발 환경과 커널을 빅 엔디언(Endian)으로 변경하였고, 커널 구동에 필수적인 장치들을 재설계하였으며, 커널을 메모리에 적재할 수 있도록 시스템 메모리 맵을 재설정하였다. 또한 시스템 장치제어를 위한 디바이스 드라이버를 작성하였다.

본 논문의 구성은 다음과 같다. 2장에서는 상용 디지털 TV 시스템의 구성을 살펴보고, 3장에서는 리눅스를 적용하기 위한 이식과정을 제시하고, 4장에서는 리눅스가 적용된 시스템의 결과물로서 장치 제어 결과와 최종적으로 출력되는 화면을 보여주고, 마지막으로 결론을 제시한다.

2. 상용 디지털 TV 시스템

일반적으로 상용 디지털 TV 시스템은 4가지 기능 모듈로 이루어져 있다. 디지털 모듈, RF 모듈, LCD 구동 모듈, 아날로그 출력 모듈로 구성되어 있다.

각각의 모듈들은 디지털 제어 모듈에 의해 제어된다. 즉, 디지털 모듈의 관점에서 각각의 모듈들은 버스와 시그널에 의해 처리되는 I/O 장치로 볼 수 있다. 이식작업의 과정에서 중심이 되는 부분이 이러한 디지털 모듈이다. [그림1]은 디지털 TV 시스템의 모듈 구성에 관한 블록 다이어그램이다.



[그림 1] 디지털 TV 시스템 모듈 구성

디지털 모듈은 디지털 제어 모듈과 영상처리 모듈로 구성된다. 디지털 제어 모듈은 전체 시스템을 제어하는 부분으로 시스템 제어 프로세서(MCU)와 SDRAM, 플래시 메모리, I2C(Inter IC)버스 인터페이스 등으로 구성되어 있다. 영상처리 모듈은 MPEG데이터를 인코딩·디코딩하는 모듈로서, 신호 처리 장치(SPU-signal processor unit)와 데이터 프레임을 저장할 SDRAM으로 구성된다. RF모듈은 변조된 디지털 방송 신호를 MPEG 신호로 복조한다. LCD 구동 모듈은 스크린 확대 방식의 TV에서 광원으로 사용되는 LCD패널을 제어한다. 아날로그 출력모듈은 음성 출력, 편향판 제어, 외부 출력 신호등을 제어 한다.

3. 리눅스 이식

리눅스를 디지털 제어 모듈에 이식하는 과정은 교차 개발 환경 구축, 커널 재구성, 장치 드라이버 생성, 응용 프로그램 생성으로 이루어진다[1].

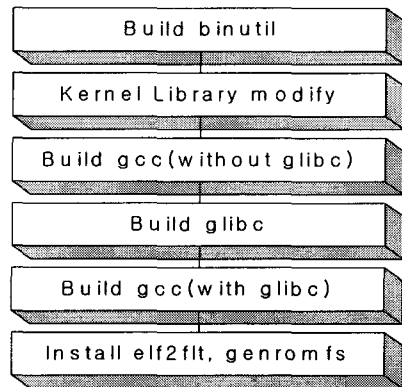
3.1 교차 개발 환경 구축

교차 개발 환경을 구축하기 위해서는 gcc컴파일러와 glibc라이브러리 및 유틸리티가 필요하다. [표1]은 교차 개발 환경에 사용된 도구모음을 보여주고 있다.

[표 1] 교차 개발 환경 구축 도구 모음

Cross compiler	
gcc-2.95.3	C 컴파일러 및 Fortran, C++등의 컴파일러
glibc-2.1.3	C 프로그램 컴파일을 위한 라이브러리
Utility	
binutils-2.12	binary utility로 ar, nm, ld, ranlib, objdump 등 바이너리 파일에 관련된 유틸리티들 모음
elf2flt-20020218	elf 포맷을 BFLT로 변환하는 도구
genromfs-0.5.1	ROM 파일 시스템 이미지 생성 도구

[그림2]는 교차 개발 환경 구축 과정을 나타낸다. 첫 번째, 교차 컴파일러 및 라이브러리, 커널 모듈 생성등 전 과정에서 사용될 링커, 어셈블러와 오브젝트 도구(objcopy, objdump 등)등의 바이너리 도구의 모음을 생성한다. 두 번째, 교차 컴파일러, 라이브러리 등이 커널 라이브러리를 참조하므로 아키텍처 의존적인 부분을 수정한다. 세 번째, 교차 컴파일러와 라이브러리의 의존성으로 인해 교차 라이브러리를 포함하지 않는 교차 컴파일러를 생성하고 이것을 이용하여 교차 라이브러리를 생성한다. 그리고 생성된 교차 라이브러리를 포함하는 교차 컴파일러를 재 생성한다. 네 번째, rpm 파일 시스템을 생성하기 위한 도구와 ELF(Executable and Linkable format)형태의 실행 바이너리를 BFLT(Binary Flat format)로 변환하기 위한 도구를 설치한다. 임베디드 리눅스의 경우 사이즈가 큰 ELF 대신 a.out의 변종인 BFLT를 기본으로 하기 때문에 위의 도구들이 필요하다[2].

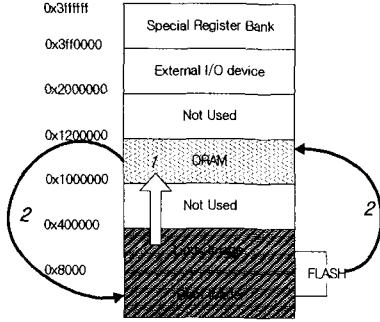


[그림 2] 교차 개발 환경 구축 과정

3.2 커널 초기화 재구성

리눅스를 이식하기 위해서는 하드웨어에 의존적인 커널 초기화 코드를 재구성해야 한다. 재구성할 부분은 리눅스를 적재하기 위한 메모리 맵, 커널 동작의 필수 장치 등이다.

메모리 맵은 MCU(Micro Control Unit)내의 시스템 관리 레지스터를 통해 결정된다. 압축된 이미지 형태의 리눅스가 메모리에 적재되는 과정은 롬에서 램으로의 복사(1), 램과 롬의 재배치(Remapping)(2), 압축 해제, 커널 코드 재배치(Relocation)를 통해 이루어진다. 각 단계에서 메모리 맵이 변경되고 이것을 타겟 시스템에 맞게 재구성한다.



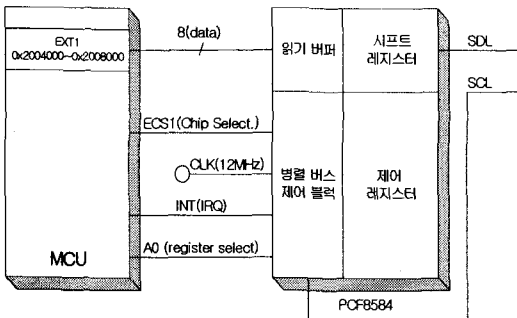
[그림 3] 메모리 맵 변경

커널 동작의 필수 장치는 인터럽트 컨트롤러, 타이머, 콘솔장치 등이 있다. 이러한 장치들이 MCU 내부 레지스터로 매핑되어 있고, 각 장치와 관련된 레지스터의 재구성을 통해 커널 동작에 필요한 기능을 제공한다.

3.3 장치 드라이버 설계

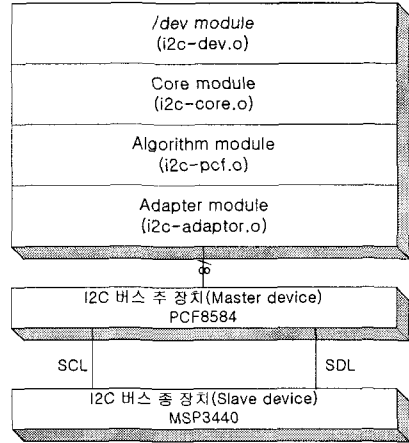
디지털 TV시스템을 구동하기 위한 장치에는 커널 동작의 필수 장치 외에 영상 처리 장치와 아날로그 제어 장치가 있다. 본 논문에서는 아날로그 제어 장치를 구동하는 장치 드라이버의 설계 및 구현에 대해 설명한다.

아날로그 제어 장치는 I2C(Inter IC)버스를 통해 제어 된다. 아날로그 장치들을 제어하기 위해 MCU의 외부 병렬 버스와 I2C 시리얼 버스 인터페이스인 PCF8584[3]의 장치 드라이버를 설계하였다. [그림4]는 PCF8584 블록 다이어그램을 나타낸다.



[그림 4] PCF8584 블록 다이어그램

PCF8584를 구동하기 위한 장치 드라이버를 [그림5]와 같이 설계하였다.



[그림 5] PCF8584장치 드라이버 구조

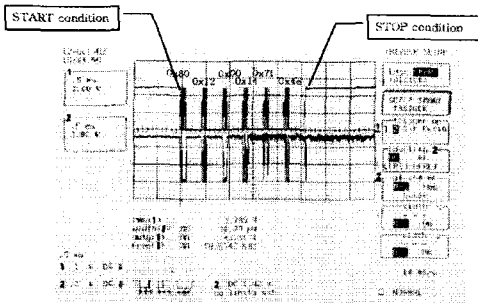
각 모듈의 기능은 아래와 같다.

- /dev 모듈
 - 장치 드라이버와 응용프로그램사이의 인터페이스.
- 코어 모듈
 - 장치 초기화 함수, 어댑터의 추가/제거 함수 정의
 - /proc 상태정보 파일 시스템에 관한 초기화.
 - I2C 버스로의 기능적 인터페이스 정의(transfer, master_send/recv, control).
- 알고리즘 모듈
 - PCF8584버스 인터페이스 장치의 기능 함수 정의.
- 어댑터 모듈
 - 메모리에 매핑된 주소를 이용 실제 데이터 전송.

4. 구현 및 실행결과

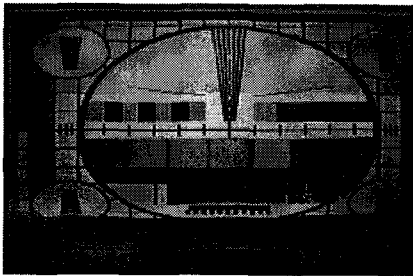
본 논문의 상용 디지털 TV시스템은 ARM7TDMI기반의 s3c3410 MCU, 8M바이트 플래시롬, 16M바이트 SDRAM (디지털 제어 모듈), SPU(MPEG II decoder), 32M바이트 SDRAM(영상 처리 모듈)으로 구성되어 있다.

이식된 리눅스의 동작 확인을 위해 I2C버스 인터페이스를 이용한 아날로그 제어 장치를 구동하였다. 결과 확인을 위해 사용된 아날로그 제어 장치는 Micronas의 MSP3440G 사운드 프로세서[4]이다. [그림6]은 사운드 프로세서 제어를 위해 I2C버스로 전송되는 데이터와 클럭을 디지털 오실로스코프를 통해 나타낸 것이다.



[그림 6] 사운드 프로세서 제어를 위한 데이터

[그림7]은 리눅스가 적용된 상용 디지털 TV 시스템에서의 화면 출력 결과이다.



[그림7] 화면 결과

5. 결론

최근 디지털 TV시스템의 수요 증가에 따라 여러 가지 상용 모델이 출시되고 있다. 이러한 디지털 TV시스템에서는 기본적으로 운영체제를 필요로 한다. 따라서 기존 디지털 TV 시스템에는 고비용의 상용 운영체제를 사용하고 있다.

본 논문에서는 고비용 상용 운영체제의 대처 방안으로 임베디드 리눅스를 재설계 이식하였다. 리눅스를 이식하기 위해 교차개발 환경을 재설계하고 리눅스 커널을 재구성 하였으며 장치 드라이버를 설계 및 구현하였다. 또한 이식된 리눅스의 동작 확인을 위해 I2C버스 인터페이스를 이용한 아날로그 제어 장치를 구동하였다.

6. 참고문헌

- [1] Craig Hollabaugh, "Embedded Linux, Hardware, Software, and Interfacing", Addison Wesley 2002
- [2] Embedded Linux/Microcontroller Project
<http://www.uclinux.org/description>
- [3] Philips. Product Information
http://www.semiconductors.philips.com/acrobat/datasheets/PCF8584_4.pdf

http://www.semiconductors.philips.com/acrobat/various/I2C_BUS_SPECIFICATION_3.pdf

[4] Micronas. Product Information

http://www.micronas.com/products/documentation/consumer/msp34x0g/downloads/msp34x0g_4pd.pdf

[5] Steve furber, "ARM system-on-chip architecture", second edition, Addison Wesley 2000