

# MPC860 프로세서에 기반한 내장형 시스템의 구현

김 경 회, 조 병 현, 오 하 령, 성 영 락, 안 현 식  
국민대학교 전자공학과  
e-mail:d995552@hanmail.net

## An Implementation of an MPC860 based Embedded System

Kyeong Hoi Kim, Byeong Heon Cho, Ha Ryoung Oh,  
Yeong Rak Seong, Hyun Sik Ahn  
Dept. of Electronics Engineering, Kookmin University

### 요 약

본 논문에서는 MPC860T 프로세서를 이용하여 통신 제어가 필요한 환경에 적합한 내장형 시스템 보드를 개발한다. 개발된 보드는 1개의 100Mbps 이더넷 포트, 4개의 10Mbps 이더넷 포트, 2개의 직렬 통신 포트를 가진다. 또한 외부의 입출력 보드와의 연결을 위한 회로를 구성하여 통신 제어 용도에 맞도록 시스템의 확장성을 높였다. 운영체제 및 소프트웨어 개발을 위해서 부트 로더의 이식이 필요하다. 부트 로더로는 PowerPC 계열의 프로세서에 적합한 PPCBoot를 사용한다. 시스템의 운영체제로는 리눅스를 이식하였다. 이를 위해서 사용된 메모리 소자의 특성과 크기 등 개발된 보드의 규격에 맞추어 리눅스 커널 소스를 수정하였다.

### 1. 서론

리눅스 운영체제의 장점을 그대로 내장형 시스템에서 이용할 수 있다는 측면에서 내장형 리눅스의 사용이 늘고 있다. 내장형 리눅스는 소스가 공개되어 있어 저가로 시스템을 구성할 수 있으며, 네트워크 운영자 입장에서 운용이 용이하고 유연성이 높다. 또한 실시간 속성을 가진 리눅스의 경우 신속, 정확하고 지속적인 정보처리가 가능하다.

본 논문에서는 모토롤라의 MPC860T를 사용한 내장형 시스템을 개발하고 여기에 내장형 리눅스를 이식한다. 물론 리눅스는 이미 파워 매킨토시를 비롯하여 몇몇 PowerPC 기반 시스템에 이식된 바가 있다[1-4]. 그러나 동일한 PowerPC 코어라 하더라도 프로세서의 종류에 따라 서로 다른 설정이 필요하고 동일한 아키텍처에서도 부팅 방법에 따라 운영 체제를 다르게 구성해야 한다.

PowerPC 계열의 프로세서들은 통신 제어가 필요한 환경에 적합한 프로세서로서 라우터, 무선 AP(access point), 이동 통신 중계기 제어 등의 분야에서 많이 사용되고 있다. 본 논문에서는 통신 제어용 환경에 적합한 보드의 기반을 구현하는 것을 목적으로 하고 있다.

본 논문에서는 하드웨어를 설계하고, 목표 보드에 리눅스

를 이식하기 위한 환경을 구성한다. 하드웨어를 구현한 후, 램 디스크로 부팅하여 디바이스들의 동작 여부를 확인한다. 본 논문은 이러한 과정들에 대한 자세한 소개보다는 전체적인 작업과 관련된 문제에 대한 경험에 초점을 맞추어 작성한다.

### 2. 관련연구

#### 2.1. MPC860T

MPC860T PowerQUICC는 다양한 제어 응용 장치들에 사용될 수 있는 단일 칩 상에 마이크로 프로세서와 주변장치들을 함께 통합한 것이다[5]. 이런 이유로 인해 MPC860T 프로세서는 특히 통신과 네트워크 시스템들에서 널리 사용된다. MPC860T 프로세서는 MMU(Memory Management Unit)를 사용할 수 있으며, 데이터 캐시와 명령 캐시를 통합하고 있는 32비트의 PowerPC이다. 높은 DSP 기능에 적합하도록 CPM(Communications Process or Module)이 탑재되어 있다. 또한 CPM은 시스템에서 데이터 통신에 필요한 일들을 처리하는데 있어 효율적인 수단을 제공하는 여러 장치들을 가지고 있다. 메모리 컨트롤러는 높은 성능의 메모리들과 최근의 DRAM 및 SDRA

M들을 포함하여 거의 모든 형태의 메모리를 지원할 수 있도록 되어 있다.

MPC8xx 계열의 프로세서는 GPCM(General Purpose Chip-select Machine), UPMA(User Programmable Machine A), UPMB(User Programmable Machine B) 등 3 가지의 장치를 이용하여 메모리를 관리할 수 있게 되어 있다. GPCM과 UPMx 중에서 어느 장치를 사용하느냐는 것은 장치와의 인터페이스 타이밍을 얼마나 세밀하게 조절해야 하는 지에 따라 결정된다. 본 논문에서는 플래시메모리는 GPCM을 사용하고, SDRAM에 대해서는 UPMx를 사용하였다.

## 2.2. 리눅스

리눅스는 멀티유저, 멀티태스킹을 지원하는 유닉스와 유사한 운영체제이다. 하지만 리눅스는 유닉스와는 독립적으로 개발된, 유닉스 계열 운영체제가 아닌 유닉스 클론(clone) 운영체제이다. 처음에는 인텔 프로세서를 장착한 PC용으로 개발되었으나 뛰어난 기능으로 인해 요즘은 인텔 프로세서가 아닌 다른 프로세서를 장착한 PC와 워크스테이션에까지 이식되고 있다. 또한 유닉스가 한 회사나 조직 내에서 몇몇 사람에 의해 폐쇄적으로 개발, 유지되는데 비해 리눅스는 인터넷으로 연결된 전 세계의 수많은 자원자들에 의해 개발이 되고 있어 유닉스에 비해 뛰어난 성능을 나타낸다. 리눅스는 멀티 플랫폼, 다른 운영체제에 대한 지원을 하며, 특히 강한 네트워크 기능을 가진다.

## 3. 시스템 개발 환경

본 논문에서는 크로스 개발 환경을 이용하여 시스템을 개발하였다. 보통 개발된 보드에 컴파일러를 올리는 것이 힘들기 때문에 호스트(Host)라 불리는 시스템에서 개발된 코드(본 논문에서는 "목표 코드"라 칭함)에 필요한 바이너리를 컴파일해서 개발된 보드에 올린다. 이때 만약 호스트 시스템이 x86계열이고 일반적인 컴파일러를 사용하면 생성되는 바이너리는 x86용으로 만들어지며, 이는 PowerPC 계열에서는 실행할 수 없다. 그러나 크로스 컴파일러를 사용하게 되면 x86 호스트 시스템에서 PowerPC용 바이너리를 생성할 수 있다. 크로스 개발 환경은 부트 로더(Boot Loader) 구축으로부터 커널, 파일 시스템 구성에 있어서 반드시 필요하므로 사용하고자 하는 구조에 맞는 환경을 구축해야 한다. 다음 그림은 크로스 개발 환경의 전체적인 구성을 나타내고 있다.

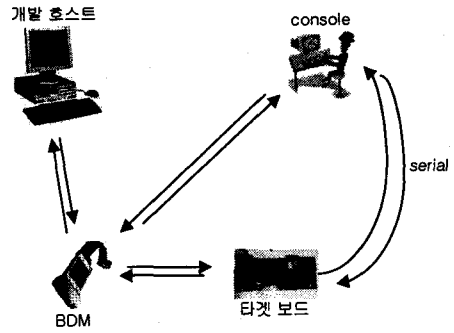


그림 1 크로스 개발 환경

본 논문에서 개발 호스트에 사용한 운영체제는 레드햇 리눅스 7.2 이다. 개발 호스트의 역할은 목표 보드에서 사용될 커널과 각종 데이터들 즉, 일반 프로그램, 공유 라이브러리, 설정 파일 등을 목표 보드에 맞추어 컴파일을 해서 이미지를 만들어 내는 역할을 한다.

디버깅을 위한 환경이나 부팅 후의 과정을 조사하기 위한 목적으로 또 하나의 PC에 윈도우 운영체제를 설치하였다. 이 PC에서 X 윈도우 애플레이터 프로그램을 이용하여 개발 호스트를 제어한다. 또한 디버깅 장치를 통하여 목표 보드를 제어하는 역할도 한다. 내장형 시스템을 위한 디버깅 장치로 주로 사용되는 것이 JTAG, 직렬 모니터, BDM 등이 있다. MPC860T의 경우에는 이들 모두를 사용할 수 있도록 되어 있다. 본 논문에서는 BDM장치를 이용한다. BDM 장치로는 WindRiver사에서 제작한 VisonProbe를 이용하였다. 이를 사용하면 MMU가 동작하기 이전까지 목표 보드를 소스코드 수준에서 디버깅할 수 있다. 또한 프로세서 내부의 레지스터와 메모리의 내용을 참조하고 수정할 수 있는 기능을 제공한다. 또한 목표 보드와 BDM 장비를 제어하기 위한 모니터 프로그램용 PC가 필요하다. 모니터 프로그램은 목표 보드에 리눅스를 이식시키는 과정과 보드의 동작을 제어하기 위해 사용된다. 기본적으로 플래시에 리눅스를 쓰기 하는 과정부터 목표 보드가 제대로 동작하는 지를 시리얼 콘솔 프로그램을 통해서 확인을 하게 된다. 그리고 BDM 장비를 이용 하드웨어 디버깅 및 소프트웨어 디버깅을 할 수 있는 프로그램을 사용해서 보드의 동작을 살펴보게 되는 것이다.

## 4. 하드웨어 설계

그림 2는 개발된 하드웨어의 전체적인 구성이다. 본 논문에서 MPC860T 프로세서를 사용한 것은 앞서 설명한 기능 외에 통신제어가 필요한 환경에 적합한 기능을 가지고 있기 때문이다. MPC860 계열의 프로세서는 MSB와

LSB가 다른 프로세서들과는 달리 반대로 사용되고 있다. 따라서 프로세서와 주변의 칩들을 연결할 때는 이점에 주의를 해서 설계를 해야 한다.

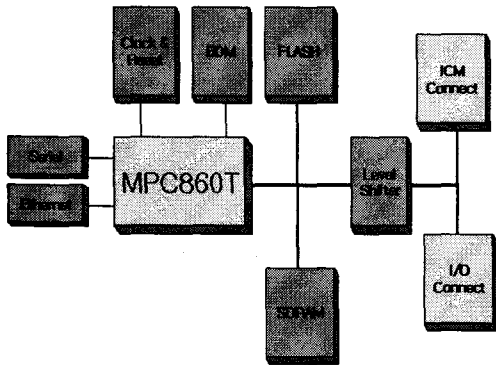


그림 2 시스템의 전체 구성

시스템의 롬 역할을 하는 메모리로서 인텔 계열의 플래시메모리 중에서 스트라타 플래시 메모리를 사용했다[6]. 램으로는 삼성에서 제작한 32M바이트 크기를 가진 두 개의 SDRAM 메모리를 사용하였다[7]. 램을 선택할 때는 램의 종류에 따른 인터페이스 규격에 맞추어 다른 UPM표 값을 사용해야 한다. 본 논문에서는 모토로라사에 제공하고 있는 SDRAM용 UPM표 값을 사용하였다.

이더넷을 연결을 위해서는 LX1970A 칩을 사용했다[8]. 이 칩은 사용자가 원하는 값을 세팅할 수 있도록 MF(Multi Function)값을 제공하고 있는데, 5V, 3.3V, 1.2V, 0V의 전압을 이용해서 값을 세팅할 수 있다.

개발된 보드에는 두 개의 클럭을 사용했다. 하나의 클럭은 수정을 사용했으며 다른 하나는 발진기를 사용하였다. MPC8xx 계열의 프로세서는 두 개의 클럭을 동시에 연결해서 사용할 수 있도록 되어 있다. 이 중에서 하나는 부팅 후에 전체 동작을 위해 사용되며 다른 하나는 시스템이 저전력 상태나 전력이 다운되고 나면 보조 전원을 사용해서 실시간 타이머를 동작시키기 위해서 사용된다.

직렬 통신은 보드의 부팅 과정에서 부팅이 이루어지고 난 후 보드의 동작 상태를 콘솔로부터 확인할 수 있는 인터페이스를 제공하기 위해 사용된다. 사용되는 방식은 RS-232C 통신을 통해서 이루어지며, 윈도우 환경에서 이를 제어하고 확인하도록 하였다.

레벨 변환기의 기능은 MPC860T 프로세서의 경우 기본적으로 3.3V로 동작이 되는데, 외부 시스템과의 인터페이스를 위한 I/O 보드는 5V로 동작할 수 있도록 위해 사용되었다. 이 레벨 변환기는 프로세서에서 나오는 3.3V 신호를 5V로 변환해서 출력을 내보내며, 5V의 입력을 3.3V로 변환해서 프로세서에 전달하는 역할을 하는 것이다[9]. 프

로세서를 비롯해서 주변 장치들은 모두 3.3V로 동작할 수 있는 것을 선택하여 구성하였다.

그리고 두 개의 커넥터가 있는데, 이 중 ICM 커넥터는 MPC860T 프로세서가 지원하는 추가적인 기능(PCMCIA, PCI 버스)을 추가적으로 구성할 수 있는 연결부분을 만들어 놓은 것이고, I/O 커넥터는 확장성을 위한 I/O 보드와의 연결을 위해 사용하는 것이다.

## 5. 리눅스 이식

### 5.1. 리눅스 이식 시 하드웨어 관련 고려사항

리눅스를 이식하기 위해서는 MPC860T 프로세서의 기본 구성과 동작을 파악해야 하고, 이에 맞는 운영체제를 이식하여야 한다. 해당되는 것은 다음과 같다.

- MPC860T내의 MMU를 사용하지 않을 것인지에 대한 것을 결정해야 한다.
- 메모리 장치를 설정할 경우 어떤 방식으로 메모리를 제어 할 것인지를 결정해야 한다. 여기에는 시스템의 메모리 맵을 정하는 것이 포함된다.
- CPM을 위한 RISC 컨트롤러를 초기화를 해야 한다. 여기에는 직렬 포트 콘솔, 이더넷 등에 대한 초기화가 포함된다.
- MPC860T에서는 부동 소수 연산이 되지 않으므로 이를 위한 라이브러리를 지원해야 한다. 이는 컴파일 시에 msoft-float 옵션을 사용해서 해결할 수 있다.
- 바이트 순서로는 Big Endian을 사용하고 이에 맞게 리눅스를 이식해야 한다.

### 5.2. 커널 수정

커널을 설치한 후에 하드웨어에 관련 부분을 수정하였다. 수정된 부분은 커널 내에 있는 arch, include/asm, driver 디렉토리 내에 파일들이다. 목표 보드의 메모리 맵을 지원하도록 커널 소스를 수정하였다.

커널 소스 코드 구성 시에 목표 보드에 맞도록 필요한 기능만을 선별적으로 선택해서 다른 기능들과의 충돌을 피하도록 하였고, 이더넷을 사용하기 위해 파일을 수정하고 IP를 부여하였다.

### 5.3. 이미지 생성

실제 커널 이미지 파일(그림 3)은 리눅스 커널의 모든 기능이 포함된 실행 파일 이미지이며 모든 프로세서, 플랫폼에 무관하게 동일한 과정을 통해 생성된다. 최종 결과 파일은 ELF 형식의 vmlinux 파일이 된다.

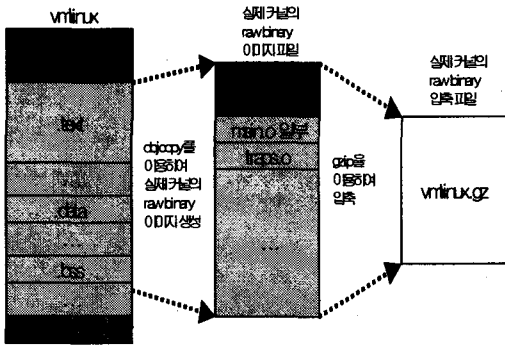


그림 3 실제 커널 이미지

압축 커널 이미지 파일 (그림 4)은 압축된 실제 커널 이미지 파일의 이진형식의 이미지를 포함한다. 위에서 설명한 이미지 파일과는 다르게 프로세서, 플랫폼, 부트 로더에 따라서 다양한 생성 방법이 있다. 일반적으로 간단한 하드웨어 초기화, 메모리 재배치, 실제 커널 압축 해제 및 실제 커널로 분기 루틴을 포함한다.

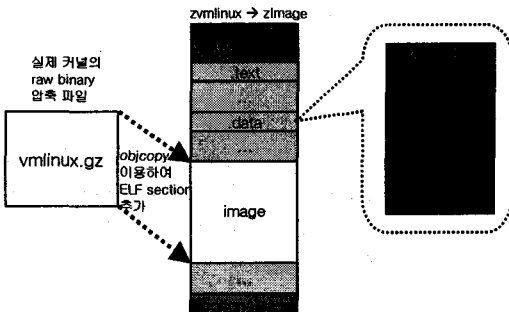


그림 4 압축 이미지 커널

#### 5.4 부트 로더

PPCBoot는 PowerPC계열은 물론 최근의 ARM 시리즈의 프로세서에 기반한 보드를 지원하는 오픈 소스로 제작된 부트 로더이다. 이는 보드의 초기화를 수행하며, 커널을 램에 올리는 부트 로더와 모니터로서의 역할을 수행한다.

부트 로더로서의 기능은 시리얼 포트를 통한 이미지 적재뿐만 아니라 TFTP를 이용하여 설정한 자신의 IP주소와 TFTP 서버의 IP 주소를 이용, 이더넷을 통해 이미지 적재할 수 있다. 또한 이미지 적재 시 다양한 이미지 형태의 커널을 인식할 수 있다. 모니터로서는 메모리에 읽고/쓰기뿐만 아니라, 플래시 메모리를 지우고 쓰는 기능을 가지고 있다.

본 논문에서는 PPCBoot를 그림 2와 같이 제작된

프로세서와 보드에 적합하게 플래시 메모리와 SDRAM, 이더넷 관련 코드를 수정하여, 커널 이식과 응용 프로그램 개발을 용이하게 하였다.

#### 6 결론

내장형 리눅스를 이식하기 위해서는 시스템의 특성과 구성 요소 등을 잘 파악하는 것이 무엇보다 중요하며 이를 위해서는 먼저 프로세서에 대한 이해가 선행되어야 한다. 본 논문에서는 MPC860T 프로세서를 이용하여 통신 제어기가 필요한 환경에 적합한 보드를 제작하고 거기에 리눅스 운영체제를 이식하였다. 이를 위해 PC를 이용해 크로스 개발 환경을 설치하고, 부트 로더를 개발하였으며, 커널 루틴의 일부를 수정하였다. 개발된 보드는 1개의 100Mbps 이더넷 포트, 4개의 10Mbps 이더넷 포트, 2개의 직렬 통신 포트를 가진다. 또한 외부의 입출력 보드와의 연결을 위한 회로를 구성하여 통신 제어 용도에 맞도록 시스템의 확장성을 높였다. 개발된 시스템에 NFS와 JFFS2를 이용하여 네트워크 동작 및 독립적인 환경에서 동작을 테스트해 본 결과 정상적으로 작동함을 확인하였다.

#### 참고 문헌

- [1] 강경태, 김태웅, 박상수, 신현식, 장래혁, "PowerPC에 기반한 내장형 시스템을 위한 리눅스의 이식", 정보과학회 추계학술대회, vol.27, no.2-3, pp.94~96, 2000.
- [2] <http://www.rcotel.co.kr>
- [3] <http://www.adelinux.com>
- [4] <http://www.mvista.com>
- [5] MPC860 PowerQUICC User's Manual, Motorola, 1998.
- [6] 3 Volt Intel StrataFlash Memory 28F640J3A(x8/x16), Intel, 2002.
- [7] K4S561632D 256Mbit SDRAM 4M×16bit×4Banks Synchronous DRAM LVTTTL, Samsung, 2002.
- [8] LXT970A Dual-Speed Fast Ethernet Transceiver, Intel, 2001.
- [9] IDTQS34X245 High-speed CMOS Quickswitch 32-bit Multiwidth Bus Switches, IDT, 2000.