제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

MPC860T 프로세서에 기반한 내장형시스템에 uC/OS 이식

유상훈, 송의석[0], 오하령, 성영락, 안현식
국민대학교 전자공학과
e-mail:eui@kornet.net[0]

Porting uC/OS on an MPC860T based Embedded System

Sang-Hun Yu, Eui-Seok Song, Ha-Ryoung Oh,
Yeong-Rak Seong, Hyun-Sik Ahn
Dept of Electronics Engineering, Kook-Min University

요 약

본 논문에서는 MPC860T 프로세서에 기반한 내장형 시스템에 실시간 운영체제 uC/OS를 이식한다. 실시간 운영체제는 외부에서 발생한 요구에 대하여 제한된 시간 내에 빨리 처리할 수 있게 하고, 여러 작업이 동시에 수행될 수 있도록 하여, 시스템이 보다 능률적이고 효과적인 성능을 발휘하도록 한다. 또한 부트 로더 PPCBoot를 이식하여 보다 편리한 사용자 환경을 구축한다.

1. 서론

최근 들어 내장형 시스템의 규모가 커짐에 따라, 네트워크나 멀티미디어 기능이 시스템의 기본 사양으로 요구되고 있다. 따라서 내장형 시스템이 처리해야 할 일들도 한층 많아지고 복잡해졌다. 이런 일들을 순차적인 프로그램으로 작성하는 것은 매우 어렵다. 이에 따라 내장형 시스템에서도 운영체제의 필요성이 절실해졌고, 내장형 시스템의 특성상 실시간 운영체제가 도입되었다. 실시간 시스템은 정해진 시간 내에 결과를 출력하는 시스템을 말한다. 즉 주어진 작업을 빨리 처리하는 것이 아니라 정해진 시간을 넘기지 않는다는 것이다. 현재 많은 내장형 시스템에서 그 목적에 맞게 실시간 운영체제가 사용되고 있다.

본 논문에서는 고속 네트워크 데이터 처리와 통신 기능을 가진 MPC860T[1] 프로세서를 탑재한 내장형 시스템에 실시간 운영체제인 uC/OS[2][3]를 이식한다. uC/OS는 소스가 공개된 이식성이 뛰어난 작은 크기의 선점형 멀티태스킹 실시간 운영체제이다. 또한 uC/OS 이미지의 간편하게 시스템에 적재하고, 보다 편리하게 응용프로그램 개발할 수 있도록 PowerPC 계열의 프로세서들을 위한 전용 부트로더인 PPCBoot[4]를 이식한다.

2. 관련연구

2.1 MPC860T

MPC860T는 PowerPC 코어에 여러 통신용 주변장치를 내장하고 있어서 통신용 장비에서 널리 사용되는 프로세서로서 성능 대비 가격이 우수한 장점을 가지고 있다. MPC860은 PowerPC 코어, 시스템 인터페이스부(System Interface Unit: SIU), 통신 프로세서 모듈(Communications Processor Module: CPM)의 세 블록으로 구성된다[5][6]. PowerPC 코어에는 캐시와 MMU가 포함되어 있으며 66MHz 클락에서 88MIPS의 처리 능력을 가지고 있다. SIU는 프로세서 내부의 버스와 외부의 버스 사이의 인터페이스를 제공한다. CPM은 여덟 개의 통신 장치를 이용하여 외부와 데이터를 주고받는다. 이 통신 장치들은 모두 독립적으로 사용될 수 있다.

2.2 PPCBoot

PPCBoot는 PowerPC계열의 프로세서에 기반한 보드들을 지원하는 부트 로더로서 소스가 공개되어 있다. 즉, 보드를 초기화하고, 플래시메모리에 저장된 운영체제를 램에 올리는 역할을 수행한다. 이미지 적재는 직렬 포트를 통해서 뿐만 아니라 이더넷을 통해서도 이루어 질 수 있다. 이더넷을 사용할

경우에는 TFTP 프로토콜을 이용한다. 이미지 적재 시에는 S-레코드, 이진 데이터, ELF 등 다양한 형태의 이미지를 인식할 수 있다. 또한 모니터 프로그램으로서의 기능도 가지고 있어서, 램 혹은 플래시 메모리의 데이터를 읽고 지우고 기록할 수 있다.

3. 시스템 개발 환경

개발 환경의 하드웨어적인 구성은 그림 1과 같이 타겟 보드, BDM 툴, 호스트로 이루어진다. 타겟 보드는 MPC860T 기반의 내장형 시스템 보드이며, BDM 툴은 BDM 포트를 통해 타겟 보드를 디버깅하거나, 타겟 보드에 부트 로더를 설치할 때에 사용된다. 본 논문에서는 BDM 툴로 VisionProbe를 사용한다. 호스트는 BDM 툴 제어 프로그램을 사용하기 위하여 윈도우즈 운영체제 상에 동작한다.

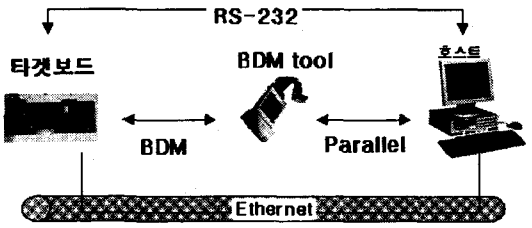

그림 1 개발환경

uC/OS가 비록 이식성이 뛰어난 운영체제이기는 하지만 프로세서에 종속되는 부분들이 있으며, 같은 프로세서일 지라도 컴파일러에 따라 소스의 수정이 필요하다. 본 논문에서는 고가의 상업용 컴파일러 대신에, GNU 툴 내의 GCC 컴파일러를 이용하여 소프트웨어 개발 환경을 구성하였다. 이때 생기는 문제는 GNU 툴을 사용하기 위해서는 유닉스 운영체제와 유사한 환경이 필요한데, BDM 툴을 사용하기 위해서는 윈도우즈 운영체제가 되어야만 한다는 것이다. 이를 해결하기 위해서 두 개의 운영체제를 병용할 수도 있겠으나 크게 불편할 것이다.

이런 이유에서 본 논문에서는 윈도우즈용 유닉스 운영체제 애뮬레이션 환경인 Cygwin[7]을 이용하여 GNU 툴 체인을 구성하였다. 이 결과로 단일한 윈도우 시스템 상에서 BDM 툴과 GNU 툴 체인을 사용할 수 있다. Cygwin은 RedHat 상에서 개발한 윈도우즈용 유닉스 애뮬레이션 환경으로 유닉스 환경에서 사용되는 여러 가지 유닉스용 프로그램들을 윈도우즈 상에서 그대로 사용할 수 있는 환경을 제공한다. 본 논문에서 Cygwin을 이용하여 재구성하여 사용된 GNU 툴 체인으로는 GCC 3.2.1, Binutils 2.13.90, Newlib 1.11.0 등이 있다.

3.1 타겟 보드

그림 2는 본 논문에서 실시간 운영체제를 이식할 타겟 보드의 블록도이다. 타겟 보드는 인텔 스트라타 플래시메모리와 삼성 SDRAM을 사용하여 구현되었으며, MAX232를 이용한 RS-232C 직렬 포트와 LXT970A을 이용한 이더넷 회로가 구성되어 있다.

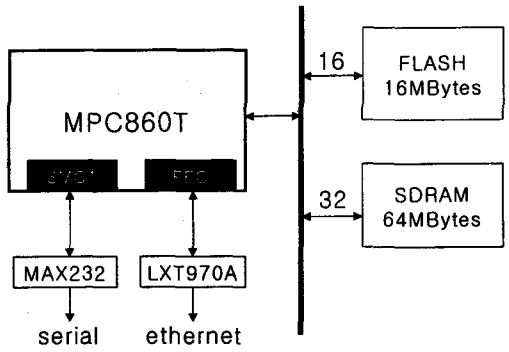

그림 2 개발보드의 블록도

타겟 보드의 메모리맵은 표 1과 같다. 특히 플래시메모리 영역은 8비트의 플래시메모리 2개를 병렬로 연결하여 16비트 접근이 가능하도록 되어 있다. 보드에는 두 개의 플래시 메모리 영역이 있으나 본 논문에서는 1개의 영역만 사용한다. SDRAM 영역은 32비트 버스트 모드를 지원할 수 있다[8][9].

표 1 메모리 맵

주소	장치	크기 (MBytes)	포트크기 (Bytes)
0x00000000 ~ 0x04000000	SDRAM	64	32
0xFE000000 ~ 0xFEFFFFFF	Flash memory	16	16
0xFF000000 ~ 0xFFFFFFFF	Flash memory	16	16

4. uC/OS의 이식

4.1 문맥 교환

uC/OS는 모든 문맥교환에서 인터럽트 처리시의 문맥 교환 방식을 모방하고 있다. 즉 모든 태스크들은 다른 태스크에 의해 선점되는 것이 아니라 오직 인터럽트에 의해 선점된다. 따라서 실제적인 문맥 교환은 인터럽트 처리 루틴 안에서 수행된다. 그러므로, 스택에 저장되는 모든 문맥은 인터럽트가 발생해서 복구를 위해 저장된 시점의 형태를 가져야 한다. 표 2는 MPC860에서 교환되어야할 문맥의 내

용으로서, 문맥은 모두 39개의 레지스터의 값으로 이루어진다.

표 2 MPC860의 문맥

레지스터	기 능
R0 ~ R31	General Purpose Registers
SRR0, SRR1	Used during exception processing
CTR	Count Register
XER	Interger Exception Register
CR	Condition Register
LR	Link Register(Return address)
MSR	Machine State Register

표 3은 문맥교환에 관련된 함수들이다. 문맥의 교환은 주로 OSCtxSw()와 OSIntCtxSw() 함수에 이해 이루어진다. OSCtxSw()는 태스크가 자발적으로 프로세서를 양도하고자 문맥을 저장/복구할 때 사용되며, 소프트웨어 인터럽트에 의해서 호출되는 인터럽트 처리 루틴이다. OSIntCtxSw()는 타이머 인터럽트에 의한 선점형 문맥 교환 시에 호출되어 문맥의 복구만을 수행하는데 사용된다. 이때 이전 문맥의 저장은 하드웨어적인 타이머 인터럽트 처리 루틴에 의해 수행된다. 나머지 문맥 교환에 관련된 함수로는 OSTaskStkInit()와 OSStartHightRdy()가 있다. 이 중에서 OSTaskStkInit()는 태스크 생성시에 호출되며 한번도 실행되지 않은 태스크를 위해 가상적으로 문맥을 만들어 넣어주는 역할을 한다. 이렇게 함으로서 OSCtxSw()와 OSIntCtxSw()의 동작 방식을 그대로 이용할 수 있다. OSStartHighRdy()는 커널 초기화 후에 사용자 태스크들이 동작하기 시작할 때 호출되며, uC/OS가 처음으로 스케쥴링을 하기 전에 OSTaskStkInit()에 의해 만들어진 문맥을 복구하고 인터럽트 리턴을 하여 첫 태스크를 구동시킨다. 또한 이후 정상적인 태스크 스케쥴링이 수행되도록 한다[10]. 본 논문에서는 위의 함수들을 어셈블리 언어로 제작하여 고속으로 문맥의 저장과 복구가 가능하도록 구현하였다.

표 3 문맥교환에 관련된 함수

함수명	기 능
OSCtxSw()	비선점형 문맥 저장과 복구
OSIntCtxSw()	타이머를 이용한 선점형 문맥 복구
OSTaskStkInit()	스케쥴링이 시작되기 전에 최초의 문맥을 만들어 저장
OSStartHighRdy()	최초로 선택된 문맥을 레지스터로 문맥 복구

4.2 클락 틱의 활성화

uC/OS는 타임아웃 기능과 시간지연 기능을 위해서 주기적으로 발생하는 클락 틱이 필요하다. 클락 틱은 초당 10회에서 100회 정도의 발생되도록 설정되어진다. 클락 틱은 멀티태스킹을 시작한 후에 클락 틱 인터럽트가 발생할 수 있도록 활성화된다. 다시 말해서, 처음으로 실행되는 태스크에서 클락 틱 인터럽트 초기화하고 활성화시켜야 한다. 만일 첫 번째 태스크를 실행하기 전에 틱 인터럽트가 발생하게 되면, 예상하지 못한 상황으로 응용프로그램이 다운될 수 있다. 따라서, 본 논문에서는 우선순위가 높은 태스크로 StartTask()를 두고, StartTask()의 시작 부분에서 클락 틱을 활성화시키도록 하였다.

4.3 부트 로더 이식

본 논문에서는 가장 최근의 PPCBoot 버전인 PPCBoot 2.0.0을 이식하였다. PPCBoot는 초기 구동시에 MPC860T 프로세서 내부의 메모리를 스택으로 사용함으로써 C 함수 호출을 가능하게 한다. PPCBoot의 시스템 초기화 코드에서는 MPC960T를 초기화하고, 곧바로 디버그 문자 출력을 위한 직렬 포트 초기화를 수행한다. 이후 SDRAM을 초기화한 후 SDRAM의 상위 주소 영역으로 플래시메모리에 있던 PPCBoot의 자신의 이미지를 복사하여 램 상의 PPCBoot 이미지에 의해 프로그램이 수행되도록 한다.

SDRAM의 코드로 제어권을 넘긴 후에는 플래시메모리를 초기화 과정을 수행한다. 본 개발 환경에서는 하나의 섹터 크기가 128K바이트인 4M바이트 플래시메모리 2개가 병렬로 사용되었다. 이 경우 표 4와 같이 초기화된다.

표 4 플래시메모리의 초기화

영역 시작주소	사용 영역	크기
0xFF000000	FLASH BASE (uC/OS 이미지 적재 가능)	15M
0xFFF00000	PPCBoot의 초기 적재 위치	256K
0xFFF40000	PPCBoot의 환경 값	256K
0xFFF80000	uC/OS 이미지 적재	512K
0xFFFFFFFF	플래시메모리의 마지막 주소	

이후 주로 이식과 관련되는 부분은 플래시메모리 관련 부분이다. 플래시메모리에 쓰기 전에는 지우기 작업이 필요한데 이 지우기 작업은 플래시메모리의 섹터 단위로만 가능하다. 이때 플래시메모리를 하드웨어 적으로 병렬로 구성했을 경우엔 지우기의 기본 섹터의 크기가 병렬 연결한 개수만큼 배가되기 때문

에 메모리가 낭비되는 현상이 나타난다. 즉 본 논문에서 사용된 플래시메모리의 섹터 크기는 128K바이트이므로, 병렬로 2 개를 사용할 경우엔 플래시메모리를 지울 때의 최소 단위는 256K바이트가 되는 것이다. 이런 점의 보완을 위해서는 조금 더 적은 크기의 섹터로 나누어진 플래시메모리를 사용하는 것이 좋다. 본 논문의 경우에도 PPCBoot와 uC/OS를 위해선 사용된 플래시메모리 영역은 1M바이트가 넘지 않았다.

본 논문에서는 보다 편리한 사용자 환경을 위하여 PPCBoot에 사용 중인 플래시메모리를 인식하고, 락을 걸거나 풀며, 데이터를 지우고 쓰고 검증하는 코드를 추가하였다.

4.4 커널 이미지의 적재

개발된 uC/OS 이미지는 2가지 방법으로 시스템에서 수행된다. 한 가지는 BDM 툴을 이용하여 시스템내의 램으로 이미지를 직접 다운로드하는 것이고, 다른 하나는 BDM 툴의 도움 없이 PPCBoot를 이용하여 플래시메모리상의 이미지를 램 영역으로 복사한 후 제어권을 넘김으로서 실행하는 것이다. 그러나 이 경우에도 모든 커널 이미지가 램으로 복사되는 것은 아니다. 즉 플래시메모리 이미지중에서 데이터 영역과 BSS 영역은 램으로 옮겨져서 사용되지만 코드 영역은 그대로 플래시메모리에 남아서 실행된다. 따라서 후자의 방법이 램을 보다 절약해서 사용할 수 있는 장점이 있다.

플래시메모리에 커널 이미지를 적재하는 것 역시 BDM 툴을 이용하는 방법과 PPCBoot를 이용하는 방법이 있다. 이 중에서 BDM 툴을 이용하는 방법은 시스템에 아직 PPCBoot가 적재되지 않았을 경우에 사용하며, 그 이후로는 PPCBoot를 이용한다. PPCBoot를 이용한 다운로드 시에는 두 가지의 방법이 있다. 하나는 직렬 포트를 통하여 Kermit 프로토콜을 이용한 데이터 전송이고, 다른 하나는 이더넷을 통하여 TFTP 프로토콜을 이용한 데이터 전송이다. 속도는 후자의 방법이 훨씬 빠르다. 양 방법 모두에서 S-레코드, 이진 데이터, ELF 등 3가지 형식의 이미지를 인식할 수 있다.

본 논문에서는 이더넷과 직렬 포트를 통한 램 이미지의 적재와 구동뿐만 아니라, 플래시메모리로의 이미지의 적재와 구동이 가능하도록 PPCBoot에 새로운 기능과 명령어를 추가하였다.

5. 결론

본 논문에서는 강력한 통신 기능과 안정적인 성능으로 최근 각광받고 있는 모토롤라사의 MPC시리즈 중에서 MPC860T을 이용한 내장형 시스템 보드에 실시간 운영체제인 uC/OS를 이식하였다. uC/OS는 이식이 용이하고, 커널 사이즈 조절이 가능한, 선점형 실시간 멀티태스킹 커널이다. 이식된 uC/OS는 내장형 시스템의 자원과 작업들을 관리하고 배치하는 역할을 한다. 또한 PPCBoot를 이식하여 호스트로부터 다양한 형식의 uC/OS 이미지를 이더넷 또는 직렬 포트를 통해 다운로드 받아 플래시메모리에 적재하고 구동하도록 하였다. 아울러 PPCBoot에 개발 환경에 맞는 플래시메모리와 SDRAM 구동 코드를 이식하였으며, 새로운 명령어를 추가하여 응용프로그램 개발을 용이하도록 하였다.

참고문헌

[1] Motorola Inc.,*MPC860 PowerQUICC User's Manual*, 1998.

[2] Jean J. Labrosse, MicroC/OS-II The Real-Time Kernel, R&D Publication, 1999.

[3] uC/OS 공식 홈페이지, http://www.ucos-ii.com.

[4] Embedded PowerPC LinuxBootProject, http://ppcboot.sourceforge.net.

[5] Motorola Inc., PowerPC Embedded Application Binary Interface, 32Bit Implementation, 1995.

[6] Thomas E. Besemer, The Motorola MPC8xx Family Designer Handbook, EST Corporation, 1998.

[7] Cygwin 홈페이지, http://www.cygwin.com.

[8] Jean J Labrosse, Embedded Systems Building Blocks, CMP, 1999.

[9] Arnold S. Berger, Embedded System Design, CMP 2002.

[10] Programming Embedded Systems in C and C++, 한빛 미디어, 2000.