RT-Linux를 OS로 하는 VME시스템을 이용한 Data Logging System 구현

The Implementation of Data Logging System by Using VME Modules based on Real Time Linux

황 석 균*, 구 경 모**, 주 문 갑+, 이 진 수++
(SeokKyun Hwang, Kyungmo Koo, Moon G. Joo and Jin S. Lee)

*포항공과대학교 전자전기공학과(전화:(054)279-5574, 팩스:(054)279-2903, E-mail : kyun@tau.postech.ac.kr)
** 포항공과대학교 전기전자공학과(전화:(054)279-5574, 팩스:(054)279-2903, E-mail : pumpkins@postech.ac.kr)
+부경대학교 전자컴퓨터정보통신공학부 (전화:(051)620-6479, 팩스:(051)620-6450, E-mail : gabi@pknu.ac.kr)
++포항공과대학교 전자전기공학과(전화:(054)279-2230, 팩스:(054)279-2903, E-mail : jsoo@postech.ac.kr)

Abstract : In this research, we port the RT-Linux to MVME 5100 board which is driven by VxWorks or Vertex until now. And, we developed the data logging modules by using the RT-Linux. This module gathers two different scan timing data from plant and sends this data to the host controller with real time.

Keywords : Data Logging System, RT-Linux, VME

Ⅰ. 서론

이 연구에서는 안정적이고 활용범위가 넓은 Linux 기반의 실시간 운영체제를 MVME 5100 보드에 이식을 하고 개발환경을 구축함으로서 기존의 상용 실시간 운영체제를 대체하는 시스템을 개발하였다. 기존의 임베디드 시스템에 적용되었던 상용 실시간 운영체제는 비싼 로열티를 지불해야 할 뿐만 아니라 특정 벤더에서 제공하는 기능만을 사용할 수밖에 없는 한계를 지니고 있었다. 이에 따라, 오픈 소스 기반의 Linux와 같은 공개되어 있고 안정성도 검증된 운영체제가 그 대안으로 주목을 받고 있다. 특히 최근에 활발히 진행되고 있는 Linux기반 프로젝트를 바탕으로 활용범위가 급속도록 확대됨에 따라 임베디드 환경의 운영체제로 Linux를 사용하는 것이 일반적인 추세로 되어가고 있다. 이에 이번 연구에서는 VME 버스를 이용하는 MVME 5100 보드에 부트로더 및 Real Time(RT)-Linux 커널을 하드웨어에 맞게 변형 이식하였으며, 이를 바탕으로 하여 실제 산업현장에서 쓰이고 있는 데이터 로깅 시스템을 개발하였다. 이번에 개발된 데이터 로깅 시스템은 기존의 VxWorks기반으로 만들어진 것과 비교하여 Real Time성능이 전혀 뒤쳐지지 않았으며, 데이터들의 실시간 처리를 확인할 수 있었다.

이 논문의 구성은 다음과 같다. 2장에서는 실시간으로 처리되는 데이터 로깅 시스템의 기본 구조 및 데이터 처리과정을 서술하였고, 3장에서는 이러한 로깅 시스템을 구성하기 위한 하드웨어에 대한 설명 및 운영체제에 대한 전반적인 설명을 하였으며, 4장에서는 VME 시스템에 RT-Linux를 포팅하는 과정을 설명하였다. 5장은 연구 결과 및 향후의 개발과제를 서술하였다.

Ⅱ. 실시간 데이터 처리를 필요로 하는 플랜트에서 사용되는 데이터 로깅 시스템

이 연구에서 개발된 Data Logging System(DLS)은 플랜트에서 실시간으로 생성되는 제어 데이터 및 현장의 센서 정보를 관리하기 위하여, VME 시스템을 하드웨어로 이용하여 개발되었다.[1][2][3] 이 시스템은 High Speed Data Logging(HSDL) 데이터 및 Low Speed Data Logging(LSDL) 데이터의 선택, 로깅 과정의 시작/재 시작/중지 명령 수행, Unix 머신에서 로깅 데이터 저장 등의 명령을 VME 시스템에 전달하고, VME 시스템의 응답을 표시하는 기능을 갖는다. 그리고 호스트 시스템과 TCP/IP를 이용한 통신을 하여 저장된

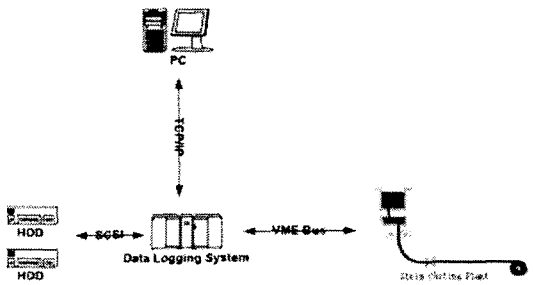

그림 1. Data Logging System 구조

데이터를 전송하게 된다.
DLS가 구현된 VME 시스템은 PLC 및 현장기기로부터의 정보를 모두 로깅한다. 표 1.에서 보는바와 같이 많은 양의 데이터를 고속으로 처리하기 위해서 CPU의 32Mbyte on-board 메모리에 저장하는 방법을 썼다.
DLS의 전체 데이터 이동 구조는 그림 1과 같다. 플랜트에서 생성된 데이터들은 VME 버스를 이용하여 DLS로 전송된다. DLS에서는 호스트의 명령에 따라 데이터들을 처리하고, 저장하기도 하며, TCP/IP를 통해 호스트로 보내지게 된다.

그림 1에서 각각의 버스들이 다루는 데이터들의 형태 및 주기는 표 1과 같다.

HSDL	총 아이템 수 : 48 Scan time : 10 msec
LSDL	총 아이템 수 : 500 Scan time : 1 sec

표1. *Data Logging System*에서 다루는 데이터구조.

플랜트에서 측정되는 데이터는 2가지의 서로 다른 데이터 입력시간을 가지고 있다. HSDL의 데이터는 대략 100 Hz * 50 items * 4 byte/item = 20 KB 정도의 양을 초당 처리되어야 한다. LSDL은 1Hz * 300 items * 4 byte/item = 1.2 KB정도의 양이 초당 만들어진다. 일반적으로 이더넷의 패킷 사이즈가 64 ~ 1508 bytes 이므로, LSDL은 1 프레임씩 한꺼번에 보내지게 프로그래밍 되어있고, HSDL은 6~7 프레임씩을 한번에 보내게 되어있다.

RT-Linux로 구현된 DLS는 기존의 VxWorks로 구현된 것과는 다르게 2중의 구조로 되어있다. 이것은 RT-Linux만의 특징으로서, RT-Linux에서는 RT 태스크와 Non-RT 태스크는 구분하여 프로그래밍 되어야 한다. RT-Linux에서 이렇게 두개의 태스크를 구별하는 이유는 Real Time 기능이 없는 기존의 Linux OS에서 사용되어지는 기능들을 모두 사용할 수 있게 하기 위함이다. 즉, Non-RT 태스크쪽에서는 기존의 Linux에서 사용하는 모든 기능들을 다 사용할 수 있으며, RT 태스크쪽은 프로그래밍 환경이 표준 POSIX 규약을 따르고 있다.
Non-RT 태스크쪽 프로그램은 User Space Program이라고 불리며, 외부와의 통신 프로그램(TCP/IP, UDP, Serial)등이 여기서 구현되어진다. 그림 1.에서 보

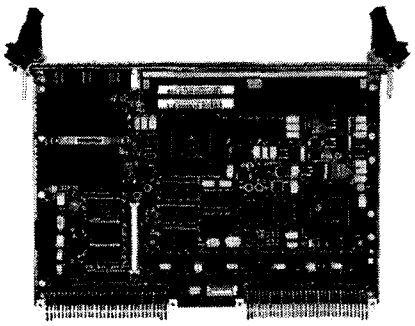

그림 2. MVME5100 Processor 모듈

는 바와 같이 DLS에서는 Host PC와의 TCP/IP를 이용한 통신이 Non-RT 태스크로 구현되어있다.
RT 태스크쪽 프로그램은 실시간으로 구현되는 프로그램들이다. 실시간으로 수행되어야 하는 제어 알고리즘들이 여기에서 구현된다. DLS에서는 제어 알고리즘이외에 플랜트에서 나온 데이터들이 VME 버스를 통해서 MVME 5100보드로 전달되는 부분이 추가로 RT 태스크로 만들어졌다.
데이터 처리과정은 다음과 같다. 플랜트에서 측정된 HSDL과 LSDL의 데이터들은 VME 버스를 이용하여 MVME 5100보드에 들어와서 RT-Linux의 RT 태스크쪽 버퍼에 저장되게 된다. 이후, 호스트 PC의 요구가 있을때 Non-RT 태스크 프로그램인 TCP/IP 통신 프로그램을 이용하여 호스트쪽으로 보내지게 된다. 그 과정에서 RT-Linux 프로그램 내부에서의 데이터 이동 과정은, RT 태스크 버퍼에 받아들여진 데이터를 손실 및 시간지연 없이 Non-RT 태스크로 전달하여야 한다. 따라서 이 두 태스크간의 데이터 이동방식은 DLS에서는 제어 명령어 및 응답 등 비교적 적은 데이터는 Real Time Fifo를 이용하여 구현되어있고, 실제 옮겨야할 많은 양의 플랜트 정보 데이터들은 Real Time Shared Memory 방식으로 구현되어 있다. 호스트 PC에서는 TCP/IP를 이용하여, DLS의 전체 과정을 제어하는 명령어를 보내고 데이터 로깅 시스템으로부터의 데이터를 받는 프로그램이 윈도우 프로그램으로 제작되어 있다.

III. 데이터 로깅 시스템 하드웨어 구성 및 운영체제

DLS 시스템에 사용된 VME Processor 모듈은 Motorola사의 MVME5100 보드를 사용하고 있다.

MVME5100 보드는 PowerPC 계열인 Motorola의 450MHz MPC750 CPU를 사용하고, 512MB ECC SDRAM 및 17MB Flash 메모리가 내장되어 있다. 내부적으로 64bit/33MHz Local PCI 버스를 사용하고

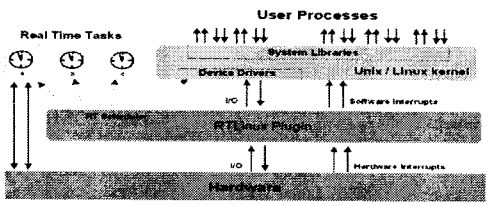

그림 3. RT-Linux의 커널 구조

있고, Tundra사의 Universe를 장착하여 VME 버스와 통신한다.

기존에 설비되어 있는 데이터 로깅 시스템은 MVME5100 Processor 모듈의 운영체제로 Windriver 사의 VxWorks를 사용하고 있다. 그러나 서론에서 밝힌 바와 같이 비싼 로열티 지불 및 한정적으로 제공되는 기능으로 인해 그러한 것을 극복하는 OS를 바탕으로 하는 DLS의 개발이 필요하게 되었다.

Linux는 기본적으로 실시간 운영체제가 아니므로 우리가 적용한 데이터 로깅 시스템과 같이 실시간 응답을 요구하는 시스템에 부적한 면이 있다. 우리는 Linux의 실시간 성능을 보완하기 위하여 FSMLabs사의 RT-Linux Pro를 사용하였다. RT-Linux는 그림 3.과 같이 일반 Linux 커널 아래에 서브 커널을 plugin 형태로 추가하여 하드웨어 및 상위 시스템을 관리하는 구조로 되어 있다. 일반 Linux 커널은 RT-Linux에서는 하나의 태스크로 존재하며, 다른 RT 태스크와 마찬가지로 실시간 스케쥴러에 의하여 최하위 우선순위를 갖고 스케쥴링 된다. 이러한 구조를 통하여 RT 태스크에 대하여 보다 확실한 실시간 보장을 가능하게 하는 동시에 오픈 커뮤니티를 통하여 개발된 Linux의 여러 고급 기능을 일반 Linux를 통하여 사용할 수 있다.

Ⅳ. MVME5100 보드에 RT-Linux 이식

일반적으로 임베디드 시스템에 Linux를 이식하는 과정은 다음과 같은 순서를 따른다.[4]

1. 교차 개발환경 구축
2. 운영체제의 커널 이미지를 보드에 부팅할 수 있는 부트로더 구성
3. 커널 수정 및 부트 이미지 생성
4. 운영체제가 마운트하기 위한 루트 파일시스템 구성

MVME5100 보드도 임베디드 시스템이므로 보드에 RT-Linux를 이식하는 과정도 위와 같은 절차를 따른다. 이식 작업은 x86 CPU 기반의 PC에서 하게 되는데, 생성된 커널 이미지는 PowerPC CPU에서 수행되어야 하므로 x86 PC에서 PowerPC를 위한 이미지를 컴파일 할 수 있도록 PC에 교차 캄파일러를 설치하여야 한다. PowerPC를 위한 교차 개발환경은 RT-Linux Pro 패키지에도 포함되어 있고, 인터넷을 통하여도 쉽게 구할 수 있다. 한편, MVME5100 보드에는 기본적으로 PPCBug라는 firmware가 내장되어 있어 부트로더의 기능을 수행하므로 별도로 부트로더를 구성할 필요는 없다. 그리고 MVME5100 보드는 외부와 VME 버스를 통하여 다른 제어기 모듈과 통신하므로, VME 버스 접근을 위한 별도의 작업이 필요하다.

일반 Linux의 MVME5100 보드로의 이식은 이미 penguinppc.org 및 MontaVista와 같은 커뮤니티를 통하여 개발되어 공개되어 있다. 그러나 이와 같이 PowerPC를 지원하기 위하여 이식된 Linux 커널에는 RT-Linux를 바로 적용하여 사용할 수 없으므로 RT-Linux를 지원할 수 있도록 커널을 수정해야할 필요가 있다. 반면에 일반 Linux의 경우에는 RT-Linux를 바로 적용하여 사용할 수 있는 패치가 제공되고 있으나 MVME5100 보드를 지원하고 있지 않으므로 별도의 커널 수정을 통한 이식 작업이 필요하다. 우리는 후자의 방식으로 MVME5100 보드에 RT-Linux를 이식하였다. 즉, 일반 Linux를 RT-Linux를 지원하도록 패치한 후에 MVME5100 보드에 적재될 수 있도록 보드의 메모리 맵, PCI 버스 설정 및 보드 setup 관련 코드 등을 수정하여 하드웨어에 이식하였다. MVME5100 보드에 적재될 수 있도록 커널을 수정하는 과정에서 penguinppc.org에 공개된 MVME5100 관련 커널 소스를 참고하였다.[5]

커널 이미지가 마운트하는 루트 파일 시스템에는 부팅 및 로그인 과정에서 수행되어야 하는 스크립트들과 쉘 및 기본적인 유틸리티와 라이브러리 등이 포함되어야 한다. 특히 데이터 로깅 시스템이 로깅 데이터를 빠른 시간에 기록해 두어야 하기 때문에 램디스크를 이용하여 루트 파일 시스템을 구성하였다. 램디스크는 램의 일부 영역을 디스크와 같이 접근할 수 있도록 하는 것으로써 메모리에 읽기/쓰기를 하므로 빠른 속도가 보장된다. 반면에 램은 휘발성이므로 전원이 꺼진 상태에서 저장된 데이터가 사라지는 특성이 있으나, 데이터 로깅 시스템의 경우에는 조업종료 시에 로깅 데이터를 하드 디스크에 저장하게 되므로 문제가 되지 않는다. 따라서, tinylogin 패키지, bash 쉘, busybox를 포함하여 루트 파일 시스템을 작성하고, 데이터 로깅 프로그램을 포함하여 램디스크 이미지를 작성하였다. busybox는 임베디드 시스템을 위하여 기본적인 유틸리티를 모아 하나의 이미지로 만들어진 패키지로써 비교적 작은 크기의 루트 파일 시스템을 쉽게 만들 수 있도록 한다.[6] 램디스크 이미지는 MVME5100 보드에 내장되어 있는 플래시 롬에 저장하고, 커널 부팅

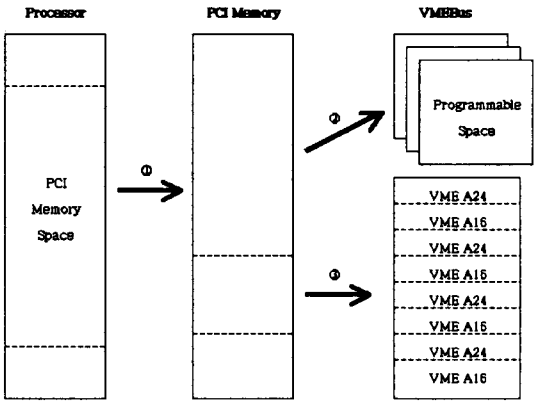

그림 4. VME 버스 메모리 맵핑 과정

시에 램에 복사되어 마운트된다.

한편, MVME5100 Processor 모듈이 VME 버스에 접근하는 과정은 그림 4 와 같다. 즉, 로컬 PCI 버스와 외부의 VME 버스를 맵핑하기 위하여 Tundra사의 Universe ASIC을 이용하는데, Universe ASIC을 통하여 VME 버스에 접근할 수 있도록 드라이버를 만들어야 한다.

V. 결론

이번 연구에서는 RT-Linux를 OS로 하는 VME 시스템을 이용하여 실제 플랜트에 이용되는 DLS시스템을 개발하였다. 일반적으로 VME 시스템은 산업현장에서 제어기로서 많이 사용되어 왔는데, 이번 연구 결과를 바탕으로 하여 앞으로 이용되는 VME 시스템은 VxWorks, Vertex 등과 함께 RT-Linux도 실시간 OS로 사용할 수 있게 되었다. 향후 연구 계획으로는 플랜트의 DLS 시스템뿐만 아니라 핵심 제어기 모듈에도 적용시켜 보는 것이다. 이를 성공시키기 위해서는 실시간 성능뿐만 아니라, 시스템의 안정성 또한 매우 중요한 요소이다.

참고문헌

[1] M.G. Joo., et "Strip Caster 2의 MMI 시스템", RIST Journal of R&D, Vol.14, No.1, pp. 130-135

[2] DaeSung Lee. et,. al. "The development of the MMI system for automating steel-making process", Proc. KIEE 2002, Korea, pp. 2469-2471, 2002

[3] Moon G. Joo. et, al. "An introduction to the MMI system of strip casting process", Proc. KIEE 2000, Korea, 2000.

[4] FSMLabs, Inc.. "Getting Started with RT-Linux", 20, Jan. 2002..

[5] http://www.busybox.net, Embedded Linux page ,Erik Anderson, 2003.

[6] http://penguinppc.org/dev/kernel.shtml, the page of the PowerPC GNU/Linux project. 2003.