

실시간 시스템 교육 및 실험 Tool kit 개발

박 성 호** 이상문* 박 동 환* 강 순 주*

*경북대학교 전자공학과

**경북대학교 정보통신학과

{slblue, moony, hwany}@palgong.knu.ac.kr, sjkang@ee.knu.ac.kr

Development of Education-purposed Real-Time System experimental tool kit

Sung Ho Park** Sang Moon Lee* Dong Hwan Park* Soon Ju Kang*

*School of Electronic and Electrical Engineering Kyungpook National University

**School of Information and Communication Kyungpook National University

요 약

본 논문에서는 교육 실험용으로 사용할 수 있는 실시간 시스템 프로토타입 tool kit을 개발하였다. 하드웨어 부분에서는 8개의 센서 입력 채널과 8개의 제어 신호 출력 채널을 가지는 AT(ISA) 버스용 I/O 보드를 제작했다. 소프트웨어 부분에서는 실시간 운영체제인 VxWorks상에서, 이 I/O 보드를 제어할 수 있는 디바이스 드라이버를 구현하고, 이를 바탕으로 멀티 스레딩과 우선 순위 스케줄링을 이용한 세차장 모델 제어 프로그램을 제작했다. 또, 현재 부각되고 있는 인터넷을 통한 원격 제어를 실현하기 위해 TCP/IP 기반의 웹서버와 자바 애플릿으로 구현된 원격 제어 클라이언트도 내장시켰다. 그리고 이 tool kit의 효용성을 입증하기 위해 Lego Dacta를 이용하여 세차장 모형을 제작하고 본 tool kit에 접속하여 인터넷에서 원격 제어가 가능한 실시간 세차장 시스템의 프로토타입을 구현하여 보았다.

1. 서론

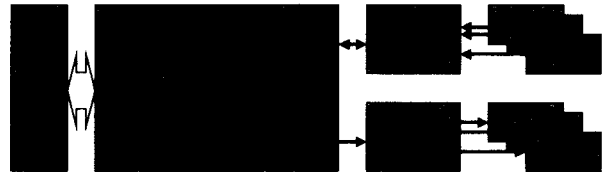
세탁기, 자동차, 항공기, 인공위성 등, 수없이 많은 장치들이 오늘날의 우리 생활과 밀접한 관계를 맺고 있다. 앞에서 나열한 모든 것이 실시간 시스템으로, 현재 우리의 생활에 차지하는 비중을 적도할 수 있는 좋은 본보기다. 앞으로의 사회에서는 이것이 더욱더 큰 영향을 미칠 것임은 명확하다. 그러나, 이러한 실시간 시스템을 실질적으로 개발하기 위한 교육 환경은 미비하며, 단지 이론 수준에만 그쳐온 것이 사실이다. 또한, 실시간 시스템은 하드웨어와 소프트웨어가 밀접한 연관을 맺고 개발되어야 하기 때문에, 둘 중 한 분야에 치우쳐서 진행되는 교육에는 한계가 있다.

본 논문에서는 실시간 시스템을 개발하는데 있어서 가장 하위 단계인 하드웨어의 제작에서부터, 하드웨어를 구동시키기 위한 디바이스 드라이버의 제작을 거쳐, 최종 단계인 실시간 시스템용 애플리케이션을 개발하는 단계까지의 과정을 기술하여, 실시간 시스템을 실질적으로 개발하기 위한 참조 모델을 제시하고자 한다.

2. I/O 보드

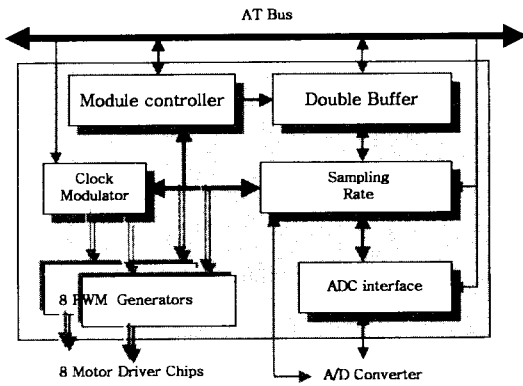
실시간 시스템은 대부분 적은 자원을 가지고 특수화된 기능을 수행한다. 때문에 필요한 기능과 환경에 시스템을 최적화 시켜야 한다. 그러기 위해선 필요한 하드웨어를 직접 제작할 필요가 있다.

본 논문에서는 8개의 센서 신호 입력 채널과 8개의 PWM(Pulse Width Modulate) 제어 신호 출력 채널을 가지는 AT(ISA) bus 용 I/O 보드를 제작하였다.



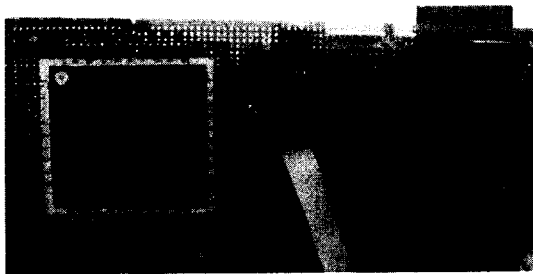
<그림 1> AT bus 용 I/O 보드 구조

제작한 I/O 보드에서 Digital logic 인 AT bus interface logic, A/D converter interface logic, sensing 된 data를 임시로 저장해 놓을 Buffer, PWM controller logic은 VHDL로 설계하고, 시뮬레이션을 통해 검증한 후 Altera사의 FPGA(Flex10k) chip으로 구현하였다(Controller). Analog logic인 A/D converter는 ADC0809 chip을 Motor driver는 L298 chip을 이용해 구현하였다.



<그림 2> Controller chip 구조

Controller chip은 AT(ISA) bus에서 들어오는 8MHz의 신호를 PWM control 신호를 발생시키기에 적당한 주파수로 변환시켜 주는 Clock Modulator와 8개의 PWM Generator, 8개의 센서 입력 채널의 센싱 빈도를 조절해 주는 Sampling Rate, A/D converter와 interface 해주는 ADC interface, I/O port 입출력을 통해 각 모듈을 제어하는 Module controller, 그리고 센싱된 Data를 일정량 모아서 DMA로 전달하는 Double buffer로 이루어져 있다.



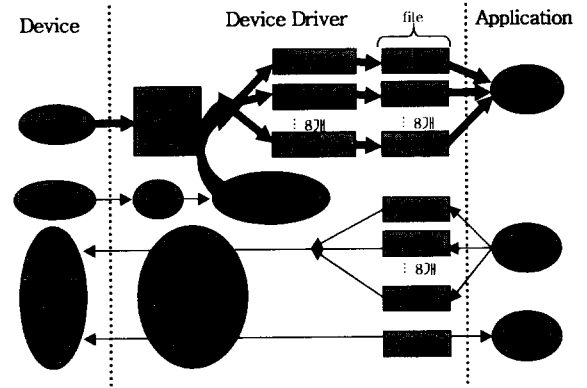
<그림 3> 보드의 실제 사진

3. 디바이스 드라이버

디바이스 드라이버는 하드웨어와 소프트웨어를 이어주는 프로그램이다. 응용 프로그램이 운영체제에 요청을 하거나 또는 운영체제 스스로가 필요에 의해서 하드웨어를 제어하려 할 때 device driver의 함수들이 호출되고, device driver의 함수들이 하드웨어를 직접 제어하게 된다. 때문에 디바이스 드라이버는 하드웨어 그리고 운영체제와 밀접하게 연관되어 있다.

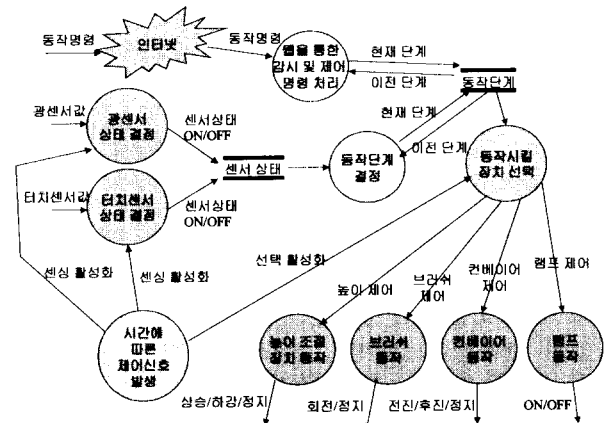
본 연구를 위해 제작한 디바이스 드라이버는 OS에서 제공하는 일반적인 파일 I/O함수인 ioctl, read, write를 이용해 제작한 제어 보드와 interface할 수 있도록 해 준다. 때문에 응용프로그램에서는 하드웨어에 관한 직접적인 고려가 필요 없고, 응용프로그램과 하드웨어는 서로 독립적일 수 있다.

그리고 실시간 시스템 제작에 사용되므로 실시간성이 고려되어야 한다. OS에서 Interrupt Service Routine(ISR)은 가장 높은 우선 순위를 가지기 때문에 ISR에서 많은 일을 해 주게 되면 응급도가 높은 작업이 제대로 수행되지 못할 수도 있다. 때문에 본 디바이스 드라이버에서는 ISR이 Interrupt Service Task를 활성화 시키는 역할만을 수행하고 실제 Interrupt Service는 Interrupt Service Task가 응급도에 맞는 우선순위를 가지고 수행하게 된다.



<그림 4> 디바이스 드라이버 구조

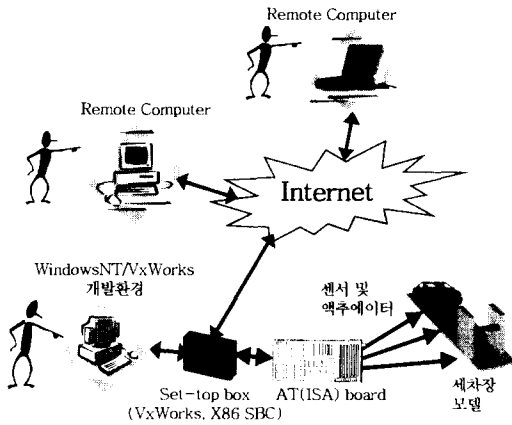
4. 제어 프로그램



<그림 5> 세차장 제어 프로그램의 구조

그림 5와 같이 설계한 소프트웨어를 시스템을 테스트하기 위한 세차장 모델에 적용해서 구현한다. 광센서와 터치센서로부터 센싱한 값으로 동작 단계를 결정해서 실제로 동작해야 하는 장치들, 우리가 개발한 디바이스 드라이버를 이용해서 구동시킨다. 센싱값을 주기적으로 받아들이기 위해서 정확한 시간 간격마다 센싱을 활성화시키는 태스크가 존재하며, 이것은 또한 장치 선택 태스크에 대한 타이밍 작업도 수행한다. 동작단계와 센서상태를 저장 및 참조하는 영역은 상호 배제를 위해서 세마포어를 이용한다.

웹 기반의 시스템 구현을 위해서 TCP/IP를 이용한 소켓으로 인터넷을 통한 동작 명령과 상태 정보를 주고 받을 수 있는 서버를 구축했다. 그리고, 서버와 통신할 수 있는 클라이언트 애플릿과, 이것을 웹 브라우저로 전달할 수 있는 웹 서버를 내장시켰다. 이렇게 함으로써 범용 웹 브라우저만으로 시스템을 제어하고 감시할 수 있는 효과를 얻을 수 있다. 일반적인 웹서버는 내장시기기엔 크기가 문제가 될 뿐만 아니라 필요로 하지 않는 기능들도 존재하기 때문에, 이 시스템을 위한 내장형 웹 서버[4][5]를 개발했다.



<그림 6> 교육용 실시간 시스템 실험 키트의 구성도

<그림 6>은 개발한 시스템의 전체 구조도이다. 세차장 모델에 사용된 각 센서 및 구동 장치를 I/O 보드에서 제어한다. 실시간 운영체제인 VxWorks 상에서 이 보드에 대한 디바이스 드라이버를 set-top box에 구축한다. VxWorks의 개발환경[2]을 이용해서 디바이스 드라이버 및 실험 키트의 제어 프로그램을 제작해서 테스트해 볼 수 있다. 그리고, 제어 프로그램에서 TCP/IP 서버 프로그램을 연동하고, 클라이언트 애플릿, 웹 서버를 내장해서 웹 기반의 제어가 가능하게 된다.



<그림 7> 실험 키트의 실제 사진

5. 결론 및 향후 연구 과제

실시간 시스템은 일반적으로 적은 자원을 가지고 특수화된 기능을 수행한다[1]. 그러므로, 사용되는 하드웨어와 소프트웨어가 일반적이 못한 경우가 대부분이며, 필요한 기능과 환경에 맞게 시스템을 최적화시킬 필요가 있다. 필요한 경우, 하드웨어에서부터 시스템 소프트웨어까지 직접 제작해야만 한다.

그렇기 때문에, 실시간 시스템을 구축하기 위해서는 하드웨어에서 시스템 소프트웨어 그리고 응용 프로그램까지의 개발에 대한 전체적인 지식이 요구된다.

하지만 대개의 경우, 하드웨어와 소프트웨어의 교육은 독립적으로 진행된다. 본 논문에서는 하드웨어 개발에서부터 최종 소프트웨어 제작까지 전체 과정을 다룸으로써 실시간 시스템 개발에 대한 실제적인 교육적 효과를 얻을 수 있는 모델을 제시한다.

향후 연구 방향으로, 실시간 시스템의 확인 및 검증 과정을 교육하고 실험할 수 있는 기능을 추가하고, 제작한 시스템의 안정성과 스케줄 가능성을 검증할 수 있는 기능도 추가할 예정이다.

6. 참고 문헌

- [1] Wolfgang A. Halang, Krzysztof M. Sacha. Real-Time Systems. World Scientific, 1992
- [2] VxWorks 5.3.1 Programmer's guide. WindRiver, 1997
- [3] Steve Heath. Embedded System Design. Newnes, 1997
- [4] Y. Sheinin, M. Emelianov, and M. Ignatiev. "Embedded Real Time Web Servers," Real-Time Magazine, pp84-89, Jan, 1998
- [5] Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen. HTTP/1.0 -- Informational RFC 1945, 1996
- [6] Frank Van Gilluwe. The Undocumented PC : A Programmer's Guide to I/O, Cpus, and Fixed Memory Areas, 1996
- [7] Don Anderson, John Swindle, Tom Eisa System Shanley. Isa System Architecture (PC System Architecture Series), 1995