

실습에 기반한 임베디드 소프트웨어 설계 교육

A Project-Based Embedded Software Design Course

문정호 · 박래정
Jung-Ho Moon and Lae-Jeong Park

강릉원주대학교 전자공학과

요 약

이 논문은 임베디드 소프트웨어 설계 과목을 위한 실습 키트 하드웨어와 이 키트를 사용한 임베디드 소프트웨어 설계 교육 과정에 대해서 소개한다. 임베디드 소프트웨어 설계 과목은 그 특성상 실습과 한 학기에 걸친 프로젝트 위주로 진행되는데 이를 위해서는 소프트웨어를 실행시킬 실습 키트가 꼭 필요하다. 학생들이 하드웨어를 완벽하게 이해하고 소프트웨어 설계 및 개발을 진행할 수 있도록 학생들의 수준에 맞는 맞춤형 실습 키트 하드웨어를 설계하고 제작하였다. 학생들은 제작된 실습 키트를 사용하여 디바이스 구동 소프트웨어에서부터 사용자 인터페이스까지 임베디드 소프트웨어 전 계층에 걸친 프로그램 설계하고 구현해 봄으로써 보다 수월하게 임베디드 시스템에 대한 이해를 넓히고 프로그램 개발 능력을 향상시킬 수 있었다.

키워드 : 임베디드 소프트웨어, 실시간 운영체제.

Abstract

This paper presents a senior-level embedded software design course using a customized training kit. Embedded software design courses commonly entail a lot of practice hours and a semester-long project and thus requires a hardware platform on which the embedded software runs. A training kit has been designed such that both hardware system and operating system are not too complicated or heavy for undergraduate students to fully understand and to develop embedded software on their own. The course using the customized training kit gives the students hands-on experience of embedded software design and programming ranging from device drivers to user interface, thereby enabling them to have in-depth understanding of embedded software and to improve their programming skills more easily and faster than when using commercial training kits.

Key Words : Embedded software design course, Real-time operating system (RTOS) , Training kit

1. 서 론

컴퓨터 기술의 발전과 함께 다양한 임베디드 시스템이 일상생활에 사용되면서 임베디드 시스템 과목은 전자공학 및 컴퓨터공학 교육의 핵심 주제들 중 하나의 위치를 차지하게 되었다. 임베디드 시스템은 특정 용도를 위한 하드웨어와 그 위에서 동작하는 소프트웨어가 밀접하게 결합되어 동작하는 컴퓨터 시스템이므로 임베디드 시스템을 이해하기 위해서는 그 기반이 되는 마이크로프로세서 관련 하드웨어 및 소프트웨어에 대한 이해와 프로그래밍 기술이 필수적으로 요구된다. 이런 이유로 임베디드 시스템 관련 과목은 기본 선수 과목의 수강을 마친 4학년 과정에서 개설되는 것이 일반적이다.

임베디드 시스템의 다양성만큼이나 임베디드 시스템 과목을 개설하는 학교와 학과의 특성에 따라 매우 다양한 방식과 내용으로 수업이 진행되고 있다. [1]과 [2]에서는 임베디드 시스템 설계 과목의 최종 결과물로서 라인 트레

이서를 개발하게 하여 팀별로 경쟁시키는 교육 방법을 소개하였다. 센서 네트워크를 활용하는 예 [3], LEGO사의 Mindstorms를 입문 교육용 자료로 활용하는 예 [4], 그리고 SoC 컴퓨터와 상용 실시간 운영체제 (RTOS)를 활용한 임베디드 시스템 수업 방법 [5] 등도 소개되었다. [6]에서는 하드웨어/소프트웨어 동시 설계 (co-design)에 기반을 둔 임베디드 소프트웨어 개발 교육 과정을 소개하였으며, [7]에서는 무선 임베디드 시스템 교육 방법을 설명하고 이를 위해 개발한 실험 키트를 소개하였다. 한편 [8]에서는 IEEE/ACM 모델 교육과정에 기반을 둔 임베디드 시스템 교육 과정을 소개하였다.

이와 같이 다양한 임베디드 시스템 관련 교육 과정은 그 세부적인 차이에도 불구하고 실습과 프로젝트에 큰 비중을 둔다는 공통점을 가지고 있다. 임베디드 시스템 과목은 본질적으로 이론에 대한 이해뿐만 아니라 직접적인 경험을 통한 기술 습득을 요구하기 때문이다. 실습이나 프로젝트의 진행을 위해서는 프로세서를 포함한 실습 키트가 필요한데 대부분의 경우에 교육용으로 판매되고 있는 상업용 실습 키트를 사용하고 있다. 시중에 판매되고 있는 대부분의 교육용 키트는 32 비트 고성능 프로세서와 많은 주변 장치를 내장하고 임베디드 리눅스 (Embedded

Linux)나 Windows CE와 같은 비교적 큰 규모의 운영체제를 실행시킬 수 있는 제품이 주류를 이룬다.

이러한 고성능의 실습 키트는 임베디드 시스템 개발 경험이 없는 학부 학생들이 한정된 시간 내에 운영체제의 동작을 모두 이해하고 이를 활용하여 소프트웨어 구현 프로젝트를 수행하기에는 너무 크고 복잡하다. 실습 키트의 높은 사양이 사전 지식과 개발 경험이 부족한 학부 학생의 실습 교육에 오히려 부정적인 요소로 작용할 수도 있다. 실제로 복잡한 하드웨어와 규모가 큰 운영체제의 사용법을 익히는 것만도 학생들에게 상당한 시간을 요구하며 이런 복잡한 시스템을 다루어야 한다는 것에 대해 학생들은 큰 부담을 느끼기 때문이다. 더욱이 이는 수강 전에 가졌던 임베디드 시스템에 관한 학생들의 흥미와 자신감을 떨어뜨리는 요인이 될 수도 있다.

이런 점에서 보면 오히려 작은 크기의 운영체제와 이를 실행시키는데 무리가 없을 정도의 간단한 하드웨어를 갖춘 소규모 시스템을 사용하는 것이 임베디드 시스템의 본질을 이해시키고 학생들의 흥미를 이끌어 내기에는 더 유리할 수 있다. 학생들은 실습 시스템의 전체 구조, 주변 장치들의 동작과 인터페이스 방법, 운영 체제의 동작에 이르기까지 전반을 훨씬 용이하게 이해할 수 있기 때문이다. 고성능 시스템은 아니라 할지라도 하드웨어와 직결된 디바이스 구동 프로그램에서부터 상위 계층의 사용자 인터페이스까지를 스스로 설계하고 구현하여 완전하게 동작하는 임베디드 소프트웨어를 개발해 보는 것은 임베디드 소프트웨어의 각 계층과 시스템 전체를 이해하는데 큰 도움이 된다.

이 논문에서는 학부생을 대상으로 한 임베디드 소프트웨어 설계 교육을 위해 저자들이 설계/제작한 실습 키트와 이를 활용한 교육 내용에 관하여 소개한다. 이 논문에서 소개하는 임베디드 소프트웨어 실습 키트는 복잡하지 않으면서도 실시간 운영체제를 실행시키기에 충분한 성능을 가지고 있다. 또한 학생들이 평소에 많은 관심을 가지고 있거나 산업 현장에서 표준으로 사용되고 있는 여러 주변 장치들을 포함하고 있다. 실습 키트는 ATMEL사의 8 비트 마이크로컨트롤러인 ATmega128과 터치스크린 기능을 가진 QVGA급 TFT LCD 패널, MP3 코덱, RTC, 직렬 플래쉬 메모리 등을 포함하고 있는 휴대용 시스템이며 소스가 공개되어 있는 실시간 운영체제인 uC/OS-II를 탑재하여 구동한다.

2. 실습 키트 하드웨어

실습에 사용할 실시간 운영체제와 소프트웨어를 실행시킬 기반 하드웨어 플랫폼의 선택은 상당히 중요하다. 아무리 성능이 좋아도 너무 복잡하거나 사용하기 어렵다면 학생들이 한정된 시간 동안 실시간 운영체제용 디바이스 구동 프로그램부터 사용자 인터페이스까지 전체를 이해한다는 것은 거의 불가능한 일이다.

기존의 상용 제품을 사용하여 수업을 진행해본 경험으로 학생들은 아래와 같은 어려움을 느끼는 것으로 파악되었다.

- 이전에 사용해 본 경험이 별로 없는 고성능 프로세서를 파악하는데 많은 시간이 걸리고 어렵다.

- 하드웨어 구조가 복잡하여 각 디바이스 구동 소프트웨어를 이해하기 어려워서 단순히 업체에서 제공하는 샘플 코드를 가져다 사용하는 경우가 대부분이다.
- 실습 키트와 운영체제의 사용법을 한 번 따라 해보는 것만으로도 너무 많은 시간이 걸린다.
- 실습 키트가 휴대용이 아니어서 실습 시간의 제약을 많이 받는다.

학생들이 느끼는 이러한 어려움을 해결하고 수업에 대한 이해도를 높일 목적으로 구조가 간단하면서도 동시에 임베디드 시스템의 기본 기능을 잘 갖추고 있는 맞춤형 하드웨어 시스템을 설계하였다. 실습 키트를 설계할 때 다음과 같은 점들을 염두에 두었다.

- 학부생 수준에서 키트 하드웨어 전체 구조를 이해하고 디바이스 각각을 제어하는 것이 가능할 것
- 작은 규모의 실시간 운영체제를 실행하는데 충분한 성능을 갖추고 있을 것
- 업계 표준 통신 인터페이스를 가지고 있을 것
- 배터리로 동작하며 휴대가 가능할 것
- 학생들이 흥미를 가지고 재미있게 실습할 수 있을 것

배터리를 주전원으로 사용하는 휴대용 실습 키트를 설계하고자 한 것은 소프트웨어 개발 환경이 갖추어진 컴퓨터만 있으면 학생들이 언제 어디서든지 실습을 진행할 수 있도록 하기 위한 것이다. 그리고 휴대가 가능하다면 자신이 개발한 임베디드 소프트웨어를 주위의 동료들에게 쉽게 시연해볼 수 있으므로 이것이 학생들에게는 좋은 동기 유발 요인이 될 것이라 판단하였다. 저자들은 학생들이 실습 키트를 사용하여 전원을 켜는 순간부터 끄는 순간까지 전체를 제어하는 임베디드 소프트웨어 전체를 스스로 설계하고 구현할 수 있기를 기대하였다.

실습 키트의 핵심이라 할 수 있는 프로세서는 8051과 같은 8 비트 프로세서에서 ARM이나 X86 기반의 32 비트 고성능 프로세서에 이르기까지 선택의 폭이 넓지만 그 중에서 ATMEL사의 AVR ATmega128을 선택하였다. AVR ATmega128은 최고 16 MHz의 클럭을 지원하며 128 KB의 플래쉬 메모리, 4 KB의 데이터 RAM, 4 KB의 EEPROM 외에 타이머, 다양한 직렬 통신 포트, A/D 변환기 등의 장치를 내장하고 있는 8 비트 RISC 마이크로컨트롤러이다 [9]. 시장에 나와 있는 수많은 프로세서 중에서 ATmega128을 선택한 주요 이유는 다음과 같다.

- 8 비트 프로세서라 구조가 간단하고 사용하기 쉬워서 교육용으로 적합함
- 실시간 운영체제인 μ C/OS-II를 구동하기에 성능이 충분함
- RAM, Flash 메모리, EEPROM 등의 메모리와 각종 주변 장치를 모두 포함하고 있어서 사용하기에 편리함
- AVR 계열 마이크로컨트롤러는 관련 문서와 참고 자료가 많아서 정보를 구하기 쉬움

많은 학생들이 그래픽 LCD와 터치스크린에 대해 높은 관심이 있다는 사실을 고려하여 터치스크린 기능을 갖춘 QVGA급 TFT LCD를 채택하였다. 터치스크린은 실습 키트의 주된 사용자 인터페이스 역할을 한다. 이외 실습

키트에 포함된 장치들은 다음과 같다.

- QVGA급 TFT LCD 및 감압식 터치스크린
- 터치스크린 제어기 (TSC2003)
- MP3 코덱 (VS1033) 및 이어폰 잭
- SD/SDHC 메모리 카드 슬롯
- 2 MB 시리얼 플래쉬 메모리 (AT45DB161)
- 리얼 타임 클럭 (DS1307)과 동전 배터리
- 2축 가속도 센서 (ADXL202)
- IR(infrared) 송수신부
- 밝기 센서
- 푸쉬 버튼형 사용자 스위치 3개

SD 메모리 슬롯이 포함되어 있어서 SD/SDHC 메모리 카드를 읽고 쓰는 것이 가능하고 MP3 코덱이 포함되어 있으므로 메모리 카드에 MP3 오디오 파일 저장해둔다면 이를 MP3 디코더를 통해 이어폰 잭으로 출력할 수 있다. 또한 그래픽 이미지를 LCD를 통해 출력할 수 있기 때문에 MP3 플레이어의 기본적인 기능은 실습 키트로 구현할 수 있다.

ATmega128은 내부에 128 KB의 플래쉬 메모리를 가지고 있는데 이는 웬만한 크기의 실행 코드를 저장하기에는 충분히 큰 공간이다. 하지만 실행 코드 외에도 LCD 화면에 출력할 사용자 인터페이스용 그래픽 데이터나 글꼴 데이터도 플래쉬 메모리에 저장해야 한다면 ATmega128의 내부 플래쉬 메모리로는 충분하지 않다. 실행 코드 이외의 다른 데이터를 저장하기 위한 용도로 SPI 인터페이스를 사용하는 2 MB 시리얼 플래쉬 메모리를 외부에 추가하였다. 물론 데이터를 메모리 카드에 저장해두고 필요할 때 읽어서 사용할 수도 있지만 이런 방식으로는 메모리 카드가 슬롯에 꽂혀 있지 않으면 전체 소프트웨어가 제대로 동작하지 않는다.

리얼 타임 클럭은 시간 정보를 관리하기 위한 것이다. 리얼 타임 클럭에는 동전 배터리가 추가로 연결되어 있어서 주전원인 배터리가 없을 때에도 전원이 공급되기 때문에 동전 배터리가 완전히 소모되지 않는다면 한번 설정한 시간 정보가 계속 유지된다. 이 외에 학생들의 아이디어에 따라 다양한 기능을 추가로 구현할 수 있도록 몇 가지 장치들을 추가하였다. IR 송수신 부를 사용하면 실습 키트로 전자 제품을 제어하는 리모컨 기능을 구현할 수도 있으며 실습 키트끼리 간단한 무선 통신을 하도록 만들 수 있다. 2축 가속도 센서를 활용하면 실습 키트를 들고 있는 방향이나 순간적인 움직임을 감지할 수 있다. 또한 밝기 센서는 LCD 백라이트의 밝기 조절에 활용할 수 있다.

다음 그림은 실습 키트의 전체 구조를 설명하는 블럭선도이다.

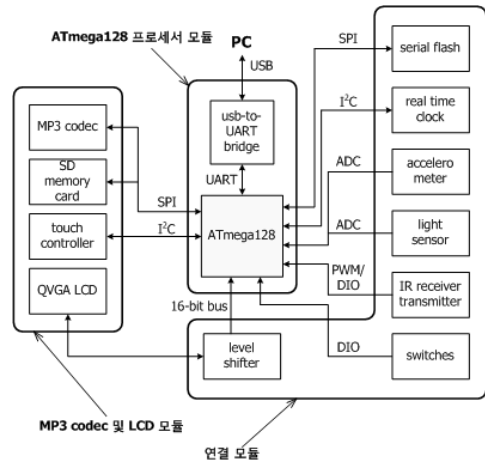


그림 1. 실습 키트의 전체 구조를 설명하는 블럭선도
Fig. 1. Block diagram of the training kit.

전체 실습 키트는 ATmega128 프로세서 모듈, MP3 코덱 및 LCD 모듈, 그리고 이 두 모듈을 연결하는 역할을 하는 연결 모듈 3 개로 구성되어 있다. 전체를 하나의 보드로 만들 수도 있지만 별도의 3개의 모듈로 만든 이유는 활용도와 유연성을 높이기 위한 것이다. 마이크로컨트롤러는 아주 다양한 목적으로 사용 가능하므로 ATmega128 프로세서 모듈은 임베디드 소프트웨어 설계 과목의 실습 외에도 학생들의 캡스틴 디자인 과목 등 다른 용도로 활용할 수 있다. MP3 코덱 및 LCD 보드는 프로세서와는 독립적이므로 ATmega128 이외의 다른 마이크로컨트롤러와 연결해서 사용할 수 있다.

다음 그림은 ATmega128 프로세서 모듈의 사진이다.

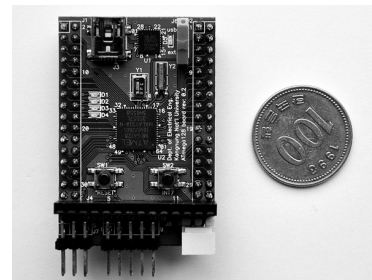


그림 2 ATmega128 프로세서 모듈
Fig. 2. The ATmega128 processor module.

ATmega128 프로세서 모듈에는 ATmega128 프로세서, USB-to-UART bridge, LED 등이 장착되어 있다. ATmega128의 모든 신호는 커넥터를 통해 외부로 제공된다. 또한 텍스트 LCD를 직접 구동할 수 있도록 미리 ATmega128의 핀들과 연결한 별도의 커넥터가 있어서 별도의 연결 없이 표준 텍스트 LCD를 장착할 수 있다. USB-to-UART bridge는 PC의 USB를 가상의 UART로 사용하도록 만들어 주기 때문에 ATmega128이 PC와 직렬 통신을 할 수 있도록 한다. USB 커넥터를 통해 ATmega128에 5 V DC 전원을 공급할 수 있으므로 ATmega128 프로세서 모듈을 PC에 연결하는 것만으로 전원 공급 및 ATmega128 마이크로컨트롤러와 PC간의 통신이 가능하다. 이를 통해 PC에서

ATmega 128의 펌웨어 코드를 다운로드하여 내부의 플래쉬 메모리에 기록할 수 있으며 또한 펌웨어 디버깅 과정에 필요한 정보를 PC로 출력할 수도 있다. 부트로더 (bootloader)를 미리 ATmega 128의 플래쉬 메모리에 기록해 둔다면 별도의 케이블이나 개발 장비가 필요하지 않다.

다음 그림은 MP3 코덱 및 LCD 모듈의 사진이다.

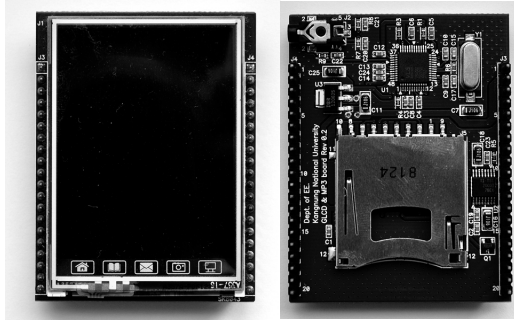


그림 3. MP3 코덱 및 LCD 모듈의 앞뒷면.

Fig. 3. Front and back sides of the MP3 codec/LCD module.

이 보드의 앞면에는 터치스크린과 LCD 컨트롤러를 포함한 QVGA 급 TFT LCD가 있으며 뒷면에는 터치스크린 제어기, MP3 코덱, SD 메모리 슬롯, 이어폰 잭이 실장되어 있다. MP3 코덱의 출력은 이어폰 잭으로 연결된다. 이 보드는 프로세서와 독립적이기 때문에 ATmega128이 아닌 다른 프로세서와 연결해서 사용할 수도 있다.

다음 그림은 연결 모듈에 프로세서 모듈과 LCD 모듈을 장착한 사진이다.

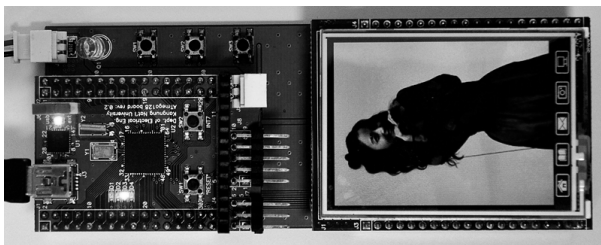


그림 4. ATmega128 모듈과 LCD 모듈을 장착한 모습.

Fig. 4. ATmega128 LCD/MP3 codec modules mounted on the interface module.

연결 모듈에는 ATmega128 모듈과 MP3 코덱 및 LCD 모듈을 위한 커넥터가 있어서 두 보드를 꽂으면 3V/5V 레벨 슈프팅 버퍼를 통해 두 보드가 연결된다. 레벨 슈프팅 회로가 필요한 이유는 ATmega128은 5V 전원을 사용하는데 LCD는 3V 전원을 사용하기 때문이다. 이외에도 이 연결 모듈에는 배터리로부터 3 V와 5 V 전압을 만들기 위한 스위칭 레귤레이터, 시리얼 플래쉬 메모리, 2축 가속도 센서, 리얼 타임 클럭, IR 송수신부, 밝기 센서가 실장되어 있다.

실습 키트에 대한 보다 상세한 사양과 회로도 및 관련 예제 코드는 저자의 블로그 <http://eslectures.blog.me>에 모두 공개해 두고 있다.

3. 실습 임베디드 소프트웨어 및 개발 환경

실습에서는 커널 (kernel)의 소스코드가 공개되어 있는 실시간 운영체제인 $\mu\text{C}/\text{OS-II}$ 이다 [10]. $\mu\text{C}/\text{OS-II}$ 는 비교적 크기가 작고 커널 소스가 공개되어 있기 때문에 실시간 운영체제를 처음 접하는 학생들의 학습용으로 유용하다. 커널의 실행에 많은 메모리를 필요로 하지 않아서 내부 메모리의 크기가 작은 8 비트 마이크로컨트롤러에서도 무리 없이 구동할 수 있다. 또한 매우 다양한 종류의 프로세서에 이식되어 있다는 장점도 있다. 임베디드 소프트웨어 설계 과목의 수강생들은 이미 선수 과목에서 실시간 운영체제의 개념과 $\mu\text{C}/\text{OS-II}$ 에 대해서 이해하고 있으므로 이 과목에서는 $\mu\text{C}/\text{OS-II}$ 를 사용한 소프트웨어 설계와 구현에 집중하며 경험을 쌓는다.

실습 키트용 임베디드 소프트웨어를 개발하기 위해서는 ATmega128 크로스컴파일러를 포함한 소프트웨어 개발 환경이 필요하다. 수업에 사용한 컴파일러는 AVR GCC (WinAVR)이다. 이를 선택한 이유는 GCC가 매우 다양한 종류의 프로세서에 대한 크로스 컴파일러가 개발되어 있으므로 표준 사용법을 익히면 대상 프로세서가 바뀌더라도 쉽게 적용할 수 있기 때문이다. 또한 AVR GCC는 무료 소프트웨어라는 것이 다른 장점이기도 하다. 학생들은 대부분 Visual C++와 같은 통합 개발 환경에 익숙하지만 이런 개발 환경이 사용 가능하지 않을 수도 있기 때문에 make 유틸리티 기반의 개발 환경을 경험하는 것도 유익할 것으로 판단했다.

ATmega128의 실행 바이너리 코드는 프로세서 내부의 플래쉬 메모리에 저장되는데 내부의 플래쉬 메모리에 기록하기 위해서는 ISP (In-System Programmer)나 JTAG 디버거와 같은 장비가 필요하다. 학생들 입장에서 프로그래밍 장비를 들고 다니며 사용하는 것은 불편한 일이다. 그래서 프로그래밍 장비 없이도 ATmega128의 펌웨어 코딩이 가능하도록 플래쉬 메모리 프로그래밍 기능을 갖춘 부트로더 (bootloader)¹⁾를 미리 ATmega128의 플래쉬 메모리에 기록하여 학생들에게 제공하였다.

부트로더가 플래쉬 메모리에 기록되어 있으면 학생들은 별도의 장비가 없어도 PC의 USB 포트를 가상의 직렬 포트로 사용하여 PC에서 간단하게 바이너리 코드를 다운로드 할 수 있다. USB 연결만으로 전원 공급, 바이너리 코드 다운로드, 직렬 통신 세 가지 기능을 실행할 수 있어서 USB 케이블 하나만 가지고 다니면 된다는 장점이 있다.

4. 주요 실습 교육 내용

수강 학생들은 실습 키트에 포함되어 있는 여러 가지 주변 장치들의 사용법에 대해 그다지 익숙하지 않기 때문에 처음부터 $\mu\text{C}/\text{OS-II}$ 용 디바이스 드라이버 개발을 진행하기는 어렵다. 그래서 학기 시작 후 9 주 동안은 강의를 통하여 각종 장치들의 인터페이스 방법, 통신 방법, 사용법을 설명하고 담당 교수가 작성한 예제 C 코드를 배포하여 학생들의 이해를 돕는다. 그리고 의도적으로 예제 C 코드에 포함시키지 않은 기능은 과제로 제출하게 하여 학생들이 데이터시트를 보며 원하는 기능

1) 부트로더는 전원이 가해진 다음 사용자 프로그램이 실행되기 전 맨 처음 실행되는 프로그램이다.

을 구현하도록 한다.

강의에서 다루는 주제는 다음과 같다.

- AVR 마이크로컨트롤러
- 직렬 통신 인터페이스 (UART, SPI, I2C)
- 외부 플래쉬 메모리 프로그래밍
- 그래픽 LCD 제어
- FAT32 파일 시스템
- A/D 변환기
- 부트로더 프로그래밍

담당 교수는 다음과 같은 예제 코드를 학생들에게 제공하여 학생들의 이해를 돕는다.

- WinAVR용 $\mu\text{C}/\text{OS-II}$ port 및 예제 코드
- 직렬 통신 인터페이스 함수들 (UART, SPI, I2C)
- 그래픽 LCD에 글자를 표시하는 예제 코드
- SD/SDHC 메모리 카드의 내용을 읽는 예제 코드
- FAT32 파일 시스템 예제 코드
- MP3 코덱 사용 예제 코드
- 터치스크린 제어기 사용 예제 코드
- A/D 변환기 사용 예제 코드

그래픽 LCD 제어기 이외의 모든 외부 장치들은 직렬 통신 인터페이스를 통하여 ATmega128과 데이터를 주고받는다. I2C 버스는 터치스크린 제어기 및 RTC와의 인터페이스에 사용되고, SPI 버스는 외부 플래쉬 메모리, MP3 코덱, SD/SDHC 메모리 카드와의 인터페이스에 사용된다. UART는 PC와의 통신을 통한 디버깅 용도로 주로 사용된다. 이 세 가지 직렬 통신 프로토콜은 거의 대부분의 임베디드 시스템에서 사용되고 있는 표준 프로토콜이므로 임베디드 소프트웨어 개발자라면 꼭 이해해야 한다. 동작 원리를 이해하면 프로세서가 바뀌더라도 쉽게 적용할 수 있기 때문에 활용도가 매우 높다.

ATmega128과 SD/SDHC 메모리 카드의 통신은 SPI 인터페이스를 사용하기 때문에 SPI 마스터 기능을 구현하면 메모리 카드와의 물리적인 통신은 가능하다. 하지만 메모리 카드에는 데이터가 파일 단위로 저장되므로 메모리 카드에 저장된 mp3 오디오 데이터나 그래픽 데이터를 읽으려면 파일 시스템이 있어야 한다. 파일 시스템을 완전히 이해하고 이를 학생들이 스스로 구현한다면 매우 좋은 공부가 되지만 이를 위해서는 무척 많은 시간이 요구된다. 그래서 FAT32 파일 시스템의 읽기 기능을 구현한 예제 코드를 배포하여 활용하도록 하였다.

9 주가 지나면 학생들은 실습 키트에 포함된 여러 주변 장치의 동작 원리와 사용법을 익힌 상태이며 이들을 사용하기 위한 기본적인 C 코드는 준비가 되어 있다. 하지만 예제 코드와 그 동안 학생들이 작성한 코드는 실시간 운영체제를 염두에 두고 작성한 것이 아니므로 멀티태스킹에 대한 고려가 전혀 없다. 따라서 $\mu\text{C}/\text{OS-II}$ 라는 실시간 운영체제 환경에서 기존의 코드는 바로 실행할 수 없으며 기존의 코드에 세마포어 (semaphore)나 메시지 큐 (message queue) 등을 추가하여 수정해야 한다. 이 시점부터 실시간 운영체제를 활용한 임베디드 소프트웨어 설계가 시작된다.

과제가 주어지면 학생들은 전체적인 소프트웨어의 구

조에 대한 진지한 고려 없이 무턱대고 컴퓨터 앞에 앉아 코딩을 시작하는 경향이 있다. 이 과목의 핵심은 실시간 운영체제 기반의 소프트웨어를 설계하고 이를 구현하는 일이므로 담당 교수는 학생들이 충분한 시간을 투자하여 소프트웨어 구조를 설계한 후 세부적인 코딩을 시작하도록 수업을 이끌어 나간다. 이 목적을 위하여 다음과 같은 부분에 초점을 맞추어 개발 제안서를 작성하도록 지도한다.

- 개발할 임베디드 시스템의 사양과 기능의 정의
- 정의된 기능을 구현하기 위한 태스크의 수와 각 태스크의 기능 정의
- 태스크끼리 주고받을 데이터와 통신 방법 정의
- 필요한 디바이스 드라이버의 기능 정의
- 디바이스 드라이버와 태스크 사이의 통신 데이터와 통신 방법 정의

실습 키트로 구동할 임베디드 소프트웨어의 핵심은 사용자 입력 검출 및 처리 방법, 공유 자원 문제의 해결 방법, 정보 출력 방법 등이라고 할 수 있다. 실습 키트로 구현 가능한 가장 기본적인 기능은 메모리 카드에 저장된 이미지 데이터를 그래픽 LCD에 출력하고, mp3 오디오 파일을 읽어서 이어폰 잭을 통해 출력하는 것이며 가속도 센서, 밝기 센서, 적외선 송수신 모듈 등의 나머지 장치들은 학생들의 아이디어에 따라 다양한 응용이 가능하다. 실습 키트의 성능으로는 구현하기에 무리한 목표를 정하거나 그래픽 인터페이스에 집착하여 눈에 보이는 부분에 많은 시간을 투자하려는 학생들이 있기 마련인데 이런 경우는 제안서 발표를 통해 목표를 낮추거나 핵심 기능에 먼저 집중하고 부차적인 부분은 핵심 부분의 구현이 끝나면 진행하도록 조언한다.

대부분의 임베디드 시스템은 제한된 크기의 메모리를 가지고 있기 때문에 메모리를 효율적으로 사용하는 일은 임베디드 소프트웨어 개발에 있어 상당히 중요한 부분을 차지한다. ATmega128의 내부 데이터 메모리 (RAM)의 크기는 4 KB인데 $\mu\text{C}/\text{OS-II}$ 커널의 실행에도 RAM이 필요하므로 결코 충분한 크기라고 할 수 없다. 이런 부분에서 학생들은 임베디드 시스템 프로그래밍과 데스크탑 프로그래밍의 차이를 극명하게 느낄 수 있다. 태스크의 수가 많으면 많을수록 필요한 메모리의 양도 비례해서 증가하므로 태스크의 기능을 너무 세분화시키면 메모리가 부족할 가능성이 높다. 따라서 태스크 수를 정하고 각 기능을 정의할 때에도 메모리의 크기를 염두에 두어야 한다. 또한 커널 중 불필요한 기능은 제거해서 메모리의 낭비를 없애야 한다. 학생들은 메모리가 제한된 환경의 개발 경험이 별로 없기 때문에 이런 부분에 대해 고민해 보는 것은 좋은 공부가 된다.

소비 전력을 줄이는 것은 휴대용 임베디드 시스템에 있어서 매우 중요한 부분의 하나이다. 실습 키트 역시 배터리로 구동되는 시스템이므로 배터리 사용 시간을 늘리기 위해서는 전력 소비를 줄이는 것이 필수적이다. 소비 전력을 최소화하는 것은 회로 설계자가 책임이지만 회로의 많은 부분이 소프트웨어로 제어되는 임베디드 시스템에서는 소프트웨어 개발자 역시 이에 대한 책임을 나누어 가진다. 당장 사용하지 않는 장치는 파워다운 (power down)이나 스탠바이 (standby)모드로 설정하는 것이나

파워다운 상태에서 풀업 저항에 의한 전력 소비가 발생하지 않도록 풀업 저항을 비활성화시키는 일 등이 이에 해당된다. 실습 키트를 끌 때 물리적으로 전원을 차단하지 않고 ATmega128 CPU가 파워다운 (power down) 모드로 진입하도록 구현한다. 그렇기 때문에 파워다운 모드에서도 불필요한 전력 소비가 발생하지 않도록 만들어야 한다. 소비 전력은 휴대용 임베디드 시스템의 배터리의 사용 시간에 큰 영향을 미치기 때문에 이에 대한 중요성을 일깨우기 위한 목적으로 최종 시연 시간에 파워다운 상태에서의 소비 전력을 측정하여 학생들의 평가에 반영한다.

5. 평가 및 교과목에 관한 학생들의 의견

이 과목의 목표는 임베디드 소프트웨어 개발 실습을 통하여 임베디드 소프트웨어 개발 과정과 소프트웨어 설계 방법에 대한 학생들의 이해를 높이고 소프트웨어 설계 능력과 개발 능력을 향상시키는 것이다. 이런 점에서 이 과목에서는 학생들의 평가를 위한 별도의 시험은 없으며 다음의 기준으로 학생들을 평가한다.

- 제안서 및 개발 완료 보고서
- 제안서 및 완료 보고서 발표와 질문에 대한 응답
- 최종 개발 작품의 완성도
- 파워다운 모드에서 소비전력

제안서와 보고서는 앞에서 설명한 작성 기준에 대한 준수 여부, 시스템 사양 정의의 정확성, 소프트웨어 설계의 타당성과 독창성 등을 위주로 평가한다. 학생들은 두 번의 발표 기회를 가지는데 발표 시간에는 담당 교수의 질문에 대해 답을 해야 한다. 질문의 내용은 소프트웨어 구조 설계, 디바이스 제어 방법, 프로그래밍 기법 등 실습 과정 전반에 걸친 것으로 이 질문을 통해 자신이 수행하고 있는 프로젝트 내용을 얼마나 잘 이해하고 있는지를 평가한다. 특히 소프트웨어 구조, 즉 시스템 구현에 이용한 태스크의 수와 각 태스크의 기능 및 태스크 간의 통신 방법에 대해 명확하게 설명하도록 요구한다.

시연을 통해 학생들은 자신의 개발 작품이 어떻게 동작하는지를 보인다. 최종 개발품의 전원을 켜고 끄는 방법은 모두 동일하게 구현한다. 특정 스위치를 1초간 누르면 켜지고 또 다시 이 스위치를 1초간 누르면 꺼지도록 하는데 전원을 끌 때에 마이크로컨트롤러는 파워다운 모드로 진입한다. 물론 이런 기능을 별도의 회로를 사용하여 구현할 수도 있지만 소프트웨어로 구현하도록 하는 이유는 파워다운 모드에서 어떤 부분이 전력 소비를 일으키는지 또 전력 소비를 줄이기 위해 어떤 일을 해야 하는지에 대한 경험하도록 하기 위한 것이다. 실제로 전력계로 소비 전력을 측정해 본 결과 학생들마다 꽤 큰 차이를 보였다.

한 학기 수업을 진행한 후 수강 학생들에게 이 과목에 대한 전반적인 의견과 평가를 위한 설문 조사를 진행하였다. 표 1은 2 년간 이 과목을 수강한 학생들 30 명의 답변을 간단하게 요약한 것이다. 이 과목은 필요한 선수 과목을 모두 수강하고 기본적인 코딩 능력이 있는 학생들에게만 수강을 권하기 때문에 수강 학생들의 수가 타 과목에 비해 많지 않은 편이다. 점수는 1점에서 10점까지를 부여한다.

표 1. 설문조사 결과.
Table 1. The survey result.

질문	평균
이 과목을 통해 임베디드 소프트웨어 설계 및 구현 기술에 발전이 있었다.	9.22
이 과목에서 사용한 실습 키트가 임베디드 소프트웨어 공부에 도움이 되었다	9.38
사용하는 운영 체제가 바뀌더라도 적용할 자신이 있다	9.06
실습과 프로젝트 중심인 이 과목의 운영 방향이 바람직하다고 생각한다	9.67
후배들에게 이 과목을 수강할 것을 추천할 것이다	9.78

기타 내용으로 다음과 같은 의견들도 있었다.

- 실습 키트가 휴대용이라 편리했으며 내가 개발한 소프트웨어를 주변 친구들에게 쉽게 시연할 수 있다는 것이 큰 동기 유발 요인이 되었다.
- 이 과목의 실습과 프로젝트가 임베디드 시스템을 이해하는데 큰 도움이 되었으며 다른 과목도 실습이 강화되는 것이 바람직하다고 생각한다.
- 다른 과목과 비교해서 이 과목에 투입한 시간이 너무 많은 편이라 힘들었지만 실무 능력을 향상시키는데 큰 도움이 되었다
- 이 과목에서 사용한 실습 키트에 평소에 많은 관심을 가지고 있는 MP3 디코더나 터치스크린 등이 포함되어 있어서 무척 재미있게 프로젝트를 진행했다.

프로젝트를 진행하면서 학생들이 가장 힘들어 했던 부분은 메모리를 효율적으로 사용하는 것이었다. µC/OS-II 커널 구동, FAT32 구현, SD/SDHC 메모리 카드 읽기를 400 KB의 메모리로 구현한다는 것은 어려운 일이라는 하지만 이 부분에서 학생들은 임베디드 시스템의 큰 특징을 명확하게 체험해 볼 수 있었다. 조금 특이한 점은 휴대용 실습 키트에 대한 학생들의 선호도가 매우 높았다는 것이다. 모든 수강생들에게 실습 키트를 무료로 배포하며 수업이 끝난 다음에도 반납하지 않는다. 이 실습 키트를 가지고 싶어서 이 과목을 신청했다는 학생들도 있었다. 평소에 학생들이 관심을 가지고 있는 기술을 직접 체험하고 학생 스스로 프로그램을 개발해서 실행해 볼 수 있다는 것이 학생들에게는 임베디드 소프트웨어 설계 과목에 대한 공부 동기를 부여하는 장점으로 작용하는 것으로 보인다.

6. 결 론

이 논문에서는 임베디드 소프트웨어 설계 과목의 수업 내용과 실습 키트를 사용한 프로젝트에 대해서 소개하였다. 상용 실습 키트를 사용하지 않고 이 과목의 교육 목표에 맞춘 실습 키트를 자체 설계/제작하여 임베디드 소프트웨어 설계와 구현 프로젝트를 진행하였다. 실습 키트에는 학생들이 평소에 기술적으로 흥미를 가지고 있는 여러

소자 및 디스플레이 장치가 포함되어 있어 학생들이 재미 있게 실습하도록 유도하였다. 실습 키트는 많은 주변 장치를 포함하되 학부 학생 수준에서 전체 하드웨어 구조를 명확하게 이해하고 소프트웨어를 개발할 수 있도록 실습 키트의 난이도를 조절하였다. 실시간 운영체제 하에 다양한 주변 장치에 대한 구동 프로그램부터 간단한 사용자 인터페이스에 이르기까지 학생들이 임베디드 소프트웨어 전 계층을 직접 설계하고 구현하도록 함으로써 임베디드 시스템과 임베디드 소프트웨어 설계에 대한 이해와 설계 능력 향상을 도모하였다.

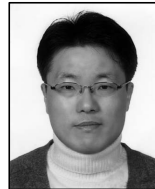
본문에서 제시한 설문조사 결과가 상용 제품을 사용하는 것보다 자체 제작한 실습 키트를 사용하는 것이 교육의 효과가 더 높다는 것을 뒷받침하지는 않는다. 다만 수년간 이 수업을 진행하면서 학생들의 수업에 대한 이해도와 관심의 강도를 관찰해 본 결과, 수강 학생들의 수준을 고려하여 자체 제작한 실습 키트를 사용한 수업에서 학생들의 참여도와 이해도가 훨씬 높은 것을 체감할 수 있었다.

참 고 문 헌

- [1] C.-S. Lee, J.-H. Su, K.-E. Lin, J.-H. Chang, and G.-H. Lin, "A project-based laboratory for learning embedded system design with industry support," *IEEE Trans. Educ.*, vol. 53, no. 2, pp. 173-181, 2010.
- [2] S. Hussmann and D. Jensen, "Crazy car race contest: multicourse design curricula in embedded system design," *IEEE Trans. Educ.*, vol. 50, no. 1, pp. 61-67, 2007.
- [3] E. Taslidere, F. S. Cohen, and F. K. Reisman, "Wireless sensor networks—A hands-on modular experiments platform for enhanced pedagogical learning," *IEEE Trans. Educ.*, vol. 51, no. 3, pp. 400-412, 2008.
- [4] S. H. Kim and J. W. Jeon, "Introduction for freshmen to embedded systems using LEGO mindstorms," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 99-108, 2009.
- [5] J. O. Hamblen, "Using a low-cost SoC computer and a commercial RTOS in an embedded systems design course," *IEEE Trans. Educ.*, vol. 51, no. 3, pp. 356-363, 2009.
- [6] H. Mitsui, H. Kambe, and H. Koizumi, "Use of student experiments for teaching embedded software development including HW/SW co-design," *IEEE Trans. Educ.*, vol. 52, no. 3, pp. 436-443, 2009.
- [7] J.-S. Chenard, Z. Zilic, and M. Prokic, "A laboratory setup and teaching methodology for wireless and mobile embedded systems," *IEEE Trans. Educ.*, vol. 51, no. 3, pp. 378-384, 2008.
- [8] K. G. Ricks, D. J. Jackson, W. A. Stapleton, "An embedded systems curriculum based on the IEEE/ACM model curriculum," *IEEE Trans. Educ.*, vol. 51, no. 2, pp. 262-270, 2008.
- [9] ATMEL coporation, ATmega128 datasheet.

[10] J. J. Labrosse, *MicroC/OS-II: The Real-Time Kernel*, 2nd ed. CMB books, 2002.

저 자 소 개



문 정 호 (Jung-Ho Moon)

1991년 : 서울대학교
제어계측공학과
1993년 : 한국과학기술원
전기및전자공학과 (공학석사)
1998년 : 한국과학기술원

전기및전자공학과 (공학박사)
2003~현재 강릉원주대학교 전자공학과 부교수
관심분야: 지능제어, 디지털 제어 시스템, 임베디드 시스템

Phone : 033-640-2427
Fax : 033-646-0740
E-mail : itsmoon@gwnu.ac.kr



박 래 정 (Lae-Jeong Park)

1991년 : 서울대학교 전기공학과
1993년 : 한국과학기술원
전기및전자공학과 (공학석사)
1997년 : 한국과학기술원
전기및전자공학과 (공학박사)

2000~현재 강릉원주대학교 전자공학과 교수
관심분야: 기계 학습, 센서네트워크 및 응용

Phone : 033-640-2389
Fax : 033-646-0740
E-mail : ljpark@gwnu.ac.kr