

임베디드 시스템 어셈블리 프로그래밍을 통한 ARM 프로세서 교육

김도연*, 김준원*, 전재욱*
*성균관대학교 전자전기컴퓨터공학과
rainbox91@skku.edu, junwon91@skku.edu, jwjeon@yurim.skku.ac.kr

An ARM Processor Course: Assembly Language Programming of One Embedded System

Do Yeon Kim*, Jun Won Kim*, Jae Wook Jeon*
* Department of Electrical and Computer Engineering

요 약

본 논문은 공학계열 학생들이 어셈블리 프로그래밍을 활용하여 임베디드 시스템(ARM 프로세서)을 학습하는 교육 과정을 소개한다. 이 교육 과정은 어셈블리 프로그래밍을 통해 실제 임베디드 시스템을 실습하여 학생들이 전공 교과 과정에서 학습한 마이크로프로세서 이론을 확인하고 이해하도록 도와준다. 임베디드 시스템을 학습하기 위해 Texas Instruments 의 TM4C123GH6PGE 마이크로컨트롤러가 탑재된 평가보드가 사용되었다. 교육 과정은 선행된 마이크로프로세서 이론 수업 진행에 맞추어 과제가 학생 개인에게 주어지며, 학생들은 결과물을 직접 시연하는 방법으로 과제를 평가 받았다. 본 논문은 이론 수업에 맞물려 진행된 과제의 일정과 과제 내용에 대해 설명한다.

1. 서론

주변의 여러 사물들이 네트워크를 통해 인터넷에 연결되어 사물 간의 새로운 형태의 커뮤니케이션이 가능하다는 패러다임을 제시한 IoT(Internet of Things) 기술이 여러 산업에서 각광받고 있다. [1] IoT 에서 말하는 사물이란 임베디드 시스템(Embedded System)을 의미한다. 임베디드 시스템이란 각종 전자제품을 둘러싸고 있는 정보 처리 시스템이다. [2] 이 임베디드 시스템은 특정 기능을 위해 마이크로프로세서에 내장된 소프트웨어가 하드웨어를 제어한다. 이런 임베디드 시스템의 예로는 자동차, 기차, 비행기, TV, 카메라, 스마트폰 등을 들 수 있다. 임베디드 시스템은 일반적으로 프로세서 또는 코어와 ROM, RAM, 시스템 목적에 따른 주변 장치로 구성된다. [3]

임베디드 시스템에서 가장 핵심적인 구성요소는 코어 즉 마이크로프로세서이다. 임베디드 시스템의 소스 코드는 시스템에 적재된 특정 ISA (Instruction Set Architecture)에서 실행되도록 컴파일 되어야 한다. [4] 마이크로프로세서 산업에는 두가지의 ISA 가 있다. CISC 인 x86 ISA 와 RISC 인 ARM ISA 이다. x86 ISA 는 데스크톱 및 노트북에 주로 적용되며, ARM ISA 는 태블릿 및 스마트폰에 주로 적용된다. [5]

IoT 시장이 커지면서 전문성을 갖춘 미래 엔지니어

를 양성하기 위해 공학계열 학생들에게 임베디드 시스템에 대한 교육을 제공하는 것의 필요성이 높아지고 있다. C 언어 혹은 C++언어와 같은 고급 프로그래밍 언어를 사용하여 임베디드 시스템을 프로그래밍할 경우, 마이크로프로세서 구조에 대한 이해 없이 프로그래밍이 가능하다. 하지만 어셈블리 언어로 프로그래밍할 경우 마이크로프로세서 구조의 이해 없이는 프로그래밍이 불가능하다. 학생들은 어셈블리 프로그래밍을 통해 임베디드 시스템을 제어하면서 마이크로프로세서의 특성과 구조를 익히게 된다. 특히 ARM 프로세서는 저 사양 임베디드 시스템과 강력한 워크 스테이션에 도입되고 있으므로 ARM ISA 를 이해하는 미래 인력 양성이 필요하다. 또한 ARM ISA 는 x86 ISA 에 비해 프로그래밍하기 쉬우므로 마이크로프로세서를 처음 접했을 때 학생들이 쉽게 이해하기에 용이하다. [6] 따라서 본 논문은 학생들에게 이론을 통해 이해한 ARM 프로세서를 실제 임베디드 시스템에서 어셈블리 프로그래밍을 이용하여 마이크로프로세서의 이해 능력을 향상시키는 교육 과정을 제안한다.

본 논문이 제안하는 교육과정의 시작은, 학생들이 고급 프로그래밍 언어인 C 언어와 어셈블리 언어를 혼합하여 프로그램을 작성하는 것으로 어셈블리 언어에 친숙함을 쌓게 한다. 그 뒤로는 순수 어셈블리 언

어를 이용하여 임베디드 시스템이 가지고 있는 여러 모듈들을 활용하여 실습한다. 교육과정 중 학생들은 R0 레지스터에서 R15 레지스터나 PSR(Program Status Register), 인터럽트 등을 활용하는 법을 익히게 된다.

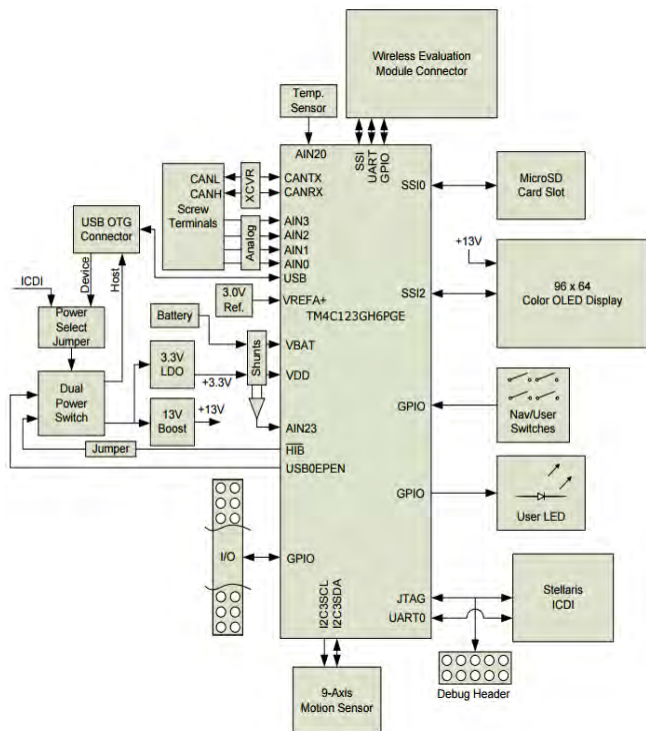
교육을 위해 임베디드 시스템은 Texas Instruments 의 TM4C123GH6PGE 마이크로 컨트롤러가 탑재된 평가 보드 (Evaluation Board)가 사용되었으며, 학생들의 IDE(Integrated Development Environment)으로는 Texas Instruments 가 제공하는 CCS(Code Composer Studio) 5.4.0 버전이 사용되었다. 학생들은 부여된 과제에 맞추어 어셈블리 코드를 CCS 에서 작성하고 이를 임베디드 시스템에 다운로드하여, 자신이 의도한대로 소프트웨어가 작동하는지를 확인한다.

본 논문의 구성은 다음과 같다. 2 장에서는 교육에 사용되는 임베디드 시스템, CCS 와 같은 학생들의 개발환경에 대해 소개한다. 3 장에서는 교육 과정에 대해 자세히 서술한다. 4 장에서는 평가 방법에 대해 설명한다. 마지막으로 5 장에서 논문을 마무리지며 논문을 마친다.

2. 개발 환경

2.1. 평가 보드

학생들의 ARM 프로세서 교육을 위해 사용된 임베디드 시스템은 Texas Instruments 의 TM4C123GH6PGE 마이크로 컨트롤러가 탑재된 Tiva™ TM4C123G 평가 보드이다. 평가 보드에는 학생들이 활용 가능한 여러 모듈들이 (그림 1)와 같이 존재한다.



(그림 1) Tiva™ TM4C123G 블록 다이어그램.

예를 들면, OLED 그래픽 디스플레이, 스위치, LED, JTAG, CAN 통신을 위한 터미널 블록, 타이머 모듈, ADC 모듈, 그리고 SCI 모듈 등이 있다. 교육 과정에서는 OLED 그래픽 디스플레이, 스위치, LED, 타이머 모듈, 그리고 SCI 모듈이 사용된다. 학생들은 해당 모듈들을 사용하기 위해 Texas Instruments 가 제공하는 데이터시트를 참조하여 각종 레지스터들을 어셈블리 프로그래밍을 통해 세팅해야만 한다. 학생들은 여러 종류의 모듈을 직접 레지스터들을 이용하여 활성화시킴으로써 임베디드 시스템을 제어하는 방법을 익히게 된다.

2.2. IDE

어셈블리 프로그래밍을 하기 위한 IDE 는 Texas Instruments 가 평가 보드와 함께 제공하는 CCS 를 사용한다. CCS 에서 개발하고자 하는 마이크로 컨트롤러 모델을 선택하고, 프로젝트를 생성하여 프로그래밍을 한다. CCS 에서 소프트웨어를 작성하고 소프트웨어를 다운로드하여 평가 보드에서 실행시킬 수 있다. CCS 는 Debug 창에서 TM4C123GH6PGE 의 모든 레지스터들의 값과 메모리에 저장된 값들을 디스플레이 하는 기능을 지원한다. 따라서 학생들은 자신의 의도에 맞게 레지스터나 메모리의 값이 제어되는지를 직접 확인할 수 있다.

3. 교육 과정

수업은 6 주로 매주 1 시간씩 교육이 진행된다. 학생들은 어셈블리 프로그래밍을 통해 임베디드 시스템을 제어하는 방법에 대해 교육받는다. 또한 학생들에게 매주 수행해야 할 과제가 주어진다. 이 교육 과정에서 학생들은 매주 3 시간씩 전공 교과 과정에서 학습한 ARM 프로세서 이론을 실습한다. <표 1>은 본 논문이 소개하는 교육 과정을 보여준다.

<표 1> 교육 과정

주차	강의주제
1	강의 소개
2	스위치 및 OLED 그래픽 디스플레이
3	스위치 및 LED
4	SCI, 스위치 및 LED
5	GPIO 인터럽트를 사용한 스위치 및 LED
6	타이머 인터럽트를 사용한 스위치 및 LED

3.1. 강의 소개

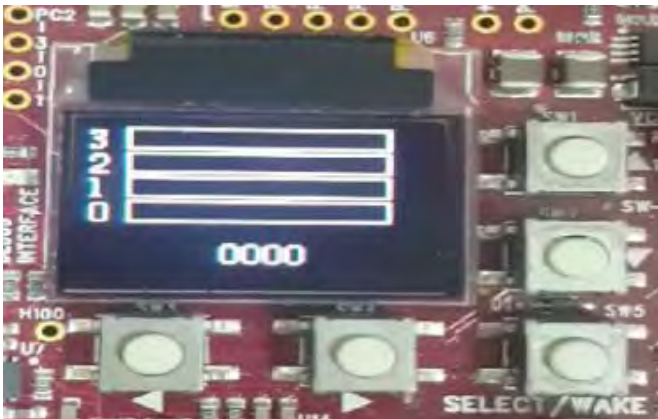
학생들에게 TM4C123GH6PGE 마이크로 컨트롤러와 아키텍처에 대한 설명이 강의 일정과 함께 학생들에게 소개된다. 학생들은 평가 보드를 구성하고 있는 모듈들에 대해 설명을 듣는다. 또한 학생들에게 개발

환경을 소개하고 CCS 를 설치하는 방법에서부터 첫 프로젝트를 생성하는 방법 까지를 설명한다.

3.2. 스위치 및 OLED 그래픽 디스플레이

전체가 어셈블리 언어로 이루어진 프로젝트를 다루기에 앞서, 학생들은 친숙한 C 언어와 친숙하지 않은 어셈블리 언어를 함께 사용하여 프로그래밍 하는 방법을 교육받는다. 이때 학생들은 CCS 에서 제공하는 라이브러리를 이용하여 OLED 그래픽 디스플레이를 C 언어로 제어하는 것을 학습한다. 또한 어셈블리 언어로 스위치를 제어하는 것을 학습한다.

교육 후, 학생들에게 첫번째 과제가 주어진다. 과제의 내용은 스위치와 OLED 그래픽 디스플레이를 제어하여 4 자리의 숫자를 디스플레이 하는 기능을 구현하는 것이다. 숫자를 증가시키는 버튼, 숫자를 감소시키는 버튼, 앞자리로 넘어가는 커서 버튼, 뒷자리로 넘어가는 커서 버튼, OLED 그래픽 디스플레이를 초기화하는 버튼을 구현해야 한다. OLED 그래픽 디스플레이에서 (그림 3)와 같은 화면이 출력된다.



(그림 3) 첫번째 과제 결과물 예시.

3.3. 스위치 및 LED

학생들은 본격적으로 어셈블리 언어만을 이용하여 평가 보드를 제어하는 방법에 대해 학습한다. 이때 스위치와 LED 를 레지스터 세팅을 통해 제어하는 방법에 대해 설명한다. 또한 이론 수업에서 학습한 Branch 관련 명령어들을 사용하여 원하는 코드로 뛰는 프로그래밍 기법을 설명한다.

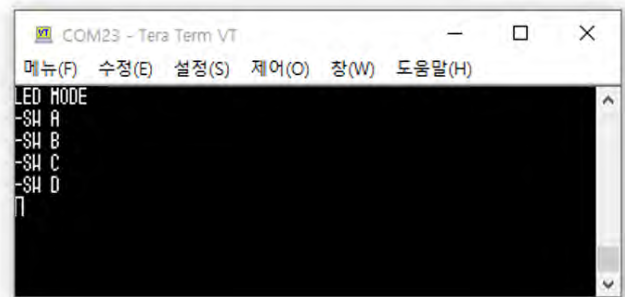
교육 후, 학생들에게 두번째 과제가 주어진다. 과제의 내용은 스위치를 이용하여 LED 를 끄고 키도록 하는 것이다. LED 를 켜는 버튼, LED 를 끄는 버튼, LED 를 5 번 빠르게 점멸하는 버튼, LED 를 5 번 느리게 점멸하는 버튼을 구현해야 한다.

3.4. SCI, 스위치 및 LED

학생들은 평가 보드가 제공하는 SCI 모듈에 대해 학습한다. 학생들에게 SCI 즉 시리얼 통신의 기본 개념과 컴퓨터와 보드 간의 통신을 위해 사용할 Tera

Term 에뮬레이터 프로그램을 설명한다. 그리고 평가 보드에서 SCI 모듈이 하드웨어적으로 어떻게 설계되어 있는지를 함께 분석한다. 학생들의 어셈블리 프로그래밍 실력의 향상을 위해, 시리얼 통신으로 문자를 출력하는 방법을 어셈블리 언어 코드가 아닌 C 언어 코드로 설명한다. 또한 이론 수업에서 학습한 LDM, STM 관련 명령어들을 사용하고, 명령어에 의해 레지스터들의 변화가 어떻게 일어나는 지에 대해 복습한다.

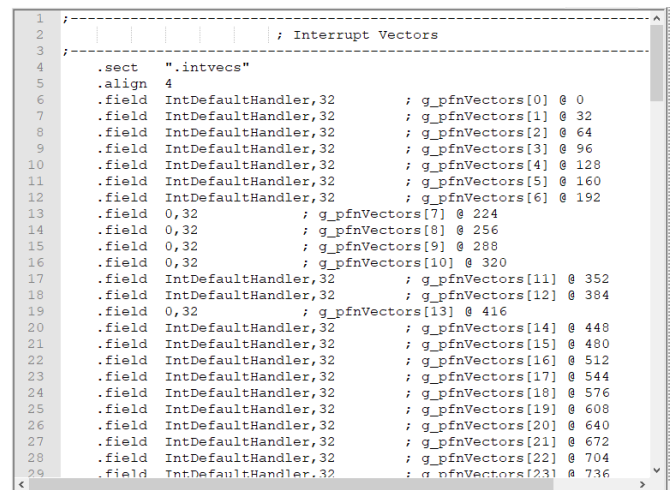
교육 후, 학생들에게 세번째 과제가 주어진다. 과제의 내용은 두번째 과제와 맥을 같이한다. 다만 스위치를 눌렀을 때, 평가 보드는 컴퓨터에서 실행되는 Tera Term 프로그램에 시리얼 통신으로 현재 어떤 버튼이 눌렸는지를 출력해서 사용자에게 (그림 4)과 같이 알려주어야 한다. 각 출력은 차례대로 LED 를 켜는 버튼, LED 를 끄는 버튼, LED 를 5 번 빠르게 점멸하는 버튼, LED 를 느리게 5 번 점멸하는 버튼을 눌렀을 때 해당 스위치가 눌렸다는 것을 알려주고있다.



(그림 4) 세번째 과제 결과물 중 SCI 모듈 사용 예시.

3.5. GPIO 인터럽트를 사용한 스위치 및 LED

학생들은 이론 수업에서 학습한 인터럽트를 평가 보드에서 구현하는 방법에 대해 학습한다. (그림 5)와 같은 벡터 테이블을 제공하고, GPIO 인터럽트를 벡터 테이블을 이용하여 활성화시키는 방법에 대해 설명한다.



(그림 5) 제공된 벡터 테이블.

학생들은 인터럽트를 활성화시키기 위해 어떻게 데이터시트를 활용하는가를 익힌다. 또한 에지 트리거와 레벨 트리거를 설명하여 학생들이 임베디드 시스템이 GPIO를 통해 발생하는 이벤트를 감지하는 방법에 대해 이해할 수 있도록 돕는다.

교육 후, 학생들에게 네번째 과제가 주어진다. 과제의 내용은 두번째 과제를 활용하도록 한다. 다만 스위치가 인터럽트 방식으로 동작하여 LED를 제어하는 차이점이 있다.

3.6. 타이머 인터럽트를 사용한 스위치 및 LED

학생들은 평가 보드가 제공하는 타이머 모듈에 대해 학습한다. 16비트, 32비트 타이머의 차이점과 타이머 모듈과 관련된 여러 레지스터들을 학습한다. 또한 5주차에서 학습한 벡터 테이블 활용 방법을 복습한다.

교육 후, 학생들에게 마지막 과제가 주어진다. LED는 계속해서 점멸해야 한다. 또한 LED를 느리게 점멸하는 버튼, LED를 빠르게 점멸하는 버튼을 구현해야 한다. 다만 LED가 점멸하는 속도는 타이머에 의해 제어되어야 한다.

4. 평가 방법

이 교육 과정의 목표는 어셈블리 프로그래밍을 통해 임베디드 시스템을 제어하여 ARM 프로세서의 이해의 향상을 도모하는 것이다. 또한 다양한 모듈들을 사용하여 학생들이 임베디드 시스템을 제어하는 방법을 습득하게끔 한다.

모든 과제는 1주일의 시간이 주어지며, 매주 강의 시간에 학생들의 결과물 시연을 통해 과제에서 요구하는 기능을 구현했는지에 따라 PASS/FAIL이 평가된다. 만약 요구사항을 하나라도 충족시키지 못했다면 FAIL로 평가된다. 또한 과제 제출 후 각 학생들의 코드를 비교하여 표절 검사도 진행된다. 표절이 적발된 경우 FAIL로 평가받게 된다.

또한 본 논문이 제시하는 교육 과정은 이론 수업으로 이루어진 전공 교과 과정과 함께 병행되어 진행된다. 따라서 학생들은 해당 수업에서 본 논문이 제안하는 교육 과정과 관련된 문제가 포함된 시험을 보게 되어 2차적으로 평가를 받는다. 시험에서 데이터시트와 같은 참고자료가 학생들에게 모두 주어지므로, 교육 과정을 충실히 따르고 성실히 과제를 진행했다면 풀기에 어렵지 않은 문제들이다. 따라서 학생들은 과제에서 평가를 받는 것뿐만 아니라 이론 수업에서도 본 논문이 제시하는 교육 과정 내용을 평가받게 된다.

5. 결론

이 논문에서는 어셈블리 프로그래밍을 통해 임베디드 시스템을 제어하여 ARM 프로세서의 이해 능력을 향상시키는 교육 과정을 소개하였다. 특히 매주 진행되는 과제를 통해서 학생들은 선행된 전공 교과 수업에서 배운 ARM 프로세서 이론을 실제로 임베디드 시스템에 적용하는 경험을 쌓았다. 학생들은 어셈블리 언어로 임베디드 시스템의 다양한 모듈들을 프로그래밍 하였다. 사용된 모듈로는 OLED 그래픽 디스플레이, 스위치, LED, SCI, 타이머가 있다. 매주 새로운 모듈이나 기능을 사용하게 함으로써 학생들의 흥미도가 유지될 수 있도록 하였다.

대다수의 학생들이 충실하게 과제를 해왔으며, 과제를 통해 학생들은 더욱 전공 교과 수업에 더 높은 집중도를 보였다. 더불어 학생들이 본 교육과정으로 하여금 많은 양질의 질문을 하여 전공 교과 과정이 질문을 하지 않은 다른 학생들에게도 더 유익해질 수 있었다.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Science and ICT (MSIP), South Korea, through the G-ITRC Support Program supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2018- 20150-00742.

참고문헌

- [1] Bandyopadhyay, Debasis, and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization." *Wireless personal communications* 58.1 (2011): 49-69.
- [2] Marwedel, Peter. *Embedded system design*. Vol. 1. New York: Springer, 2006.
- [3] Camposano, Raul, and Jörg Wilberg. "Embedded system design." *Design Automation for Embedded Systems* 1.1-2 (1996): 5-50.
- [4] Karaki, Hussein, Haitham Akkary, and Shahrokh Shahidzadeh. "X86-ARM binary hardware interpreter." 2011 18th IEEE International Conference on Electronics, Circuits, and Systems. IEEE, 2011.
- [5] Blem, Emily, Jaikrishnan Menon, and Karthikeyan Sankaralingam. "Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures." 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2013.
- [6] Clements, Alan. "Selecting a processor for teaching computer architecture." *Microprocessors and Microsystems* 23.5 (1999): 281-290.