

SDR을 위한 Embedded System의 소프트웨어 다운로딩 방법

* 황승용, ** 강환민, * 양희성, * 김용남, *조성호

* 한양대학교 정보통신공학과
** 한양대학교 전자공학과

A Software Downloading Method of Embedded system for SDR

* Seung Yong Hwang, ** Hwan Min Kang * Hee Sung Yang, * Yong Nam Kim
*Sung Ho Cho

* Department of Information & Telecommunication Engineering, Hanyang University
** Department of Electronic Engineering, Hanyang University

Phone: 031-400-3739/5178
E-mail: shcho@casp.hanyang.ac.kr

요약

SDR(Software Defined Radio)은 여러 세대의 이동통신 시스템, 무선방송 및 다양한 통신 시스템이 존재하는 환경에서 소프트웨어 업그레이드로 무선장치 및 서비스를 제공할 수 있는 효율적인 해결 방안으로 대두되고 있다. SDR이라는 기술은 하드웨어 수정 없이 모듈화 된 소프트웨어의 변경만으로 단일의 송수신 시스템을 통해 다수의 무선 통신 규격을 통합 수용하는 무선 인터페이스 기술이다[1]. 이러한 기술을 실현하기 위해서는 소프트웨어 다운로드 기술이 매우 중요하게 된다. 왜냐하면 시스템에 프로그램을 다운로딩해서 하드웨어가 Reconfiguration 되어 사용자가 원하는 대로 시스템이 셋팅이 될 수 있어야 하기 때문이다.

소프트웨어 다운로드 방법에는 크게 공중파를 이용하는 방법, 컴팩트한 이동식 디바이스로부터 다운로딩 받는 방법, 전화나 케이블 서비스 등을 통해 근거리/원거리 호스트 서버로부터 소프트웨어를 다운로딩 받는 방법, 네트워크를 통해서 소프트웨어를 다운로드 하는 방법 등이 있다[2]. 이 중에서 본 논문에서는 SDR을 위한 임베디드 시스템을 구현 하였고, 리눅스 포팅을 하였으며 스마트 카드를 이용한 소프트웨어 다운로드 방법을 제시함으로써 SDR 의미 구현을 시도해 보았다.

Defined Radio)은 하드웨어 보다 소프트웨어가 훨씬 많은 부분을 차지하는 기존의 무선통신 기술에서 새로운 통신기술로의 천이라고 할 수 있다. SDR은 무선통신 서비스 제공 업체 입장에서는 사용자에게 통합 서비스를 제공 할 수 있고, 새로운 서비스 추가가 용이하며 특별한 하드웨어 수정 없이 다중모드 전환 문제를 해결 할 수 있다. 또한 사용자에게 있어서도 다양한 시스템 사이에 roaming, 장래 규격 변경에 따른 단말기 재구입이나 하드웨어 업그레이드가 불필요하며, 시스템 제조 업체는 하드웨어 플랫폼의 간소화 및 이종 규격 간의 통합형 단말기 판매가 가능하고 사용자에게 추가적인 기능을 매우 손쉽게 제공 할 수 있는 이점이 있다. SDR은 하나의 하드웨어 단말기에 소프트웨어 다운로드에 의한 방법으로 하드웨어를 유연성 있게 재구성 하는 신개념의 4세대 이동통신 시스템이다. 그리고 미래의 무선통신은 다양한 사용자의 요구로 다중모드, 다중표준을 지원하는 단말기의 등장은 불가피 할 것이다[3].

이런 이유로, 본 논문에서는 리눅스가 Porting 된 임베디드 보드에 스마트카드 R/W기를 연결하여 다운로딩 해 보았고, 그에 관련한 임베디드 시스템에 대한 구체적인 설명과 리눅스 포팅 그리고 소프트웨어 다운로딩에 관해서는 본론에서 다루었다.

1. 서 론

4G(Generation) 통신이라고 일컫는 SDR (Software-

2. 본 론

먼저 소프트웨어 다운로딩을 위한 임베디드 보드와

그 위에 포팅 될 리눅스 그리고 보드에 시리얼 포트로 연결 될 스마트 카드 R/W의 각각 구성과 특성에 대해서 살펴보기로 한다.

그림 1에서 알 수 있듯이 컴퓨터에서 작업한 프로그램을 카드에 저장해 스마트 카드 R/W와 RS-232C를 통해 임베디드 보드에 다운로딩 하게 된다.

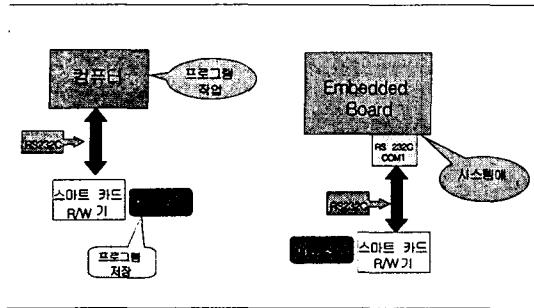


그림 1. 시스템 구성도

1) Embedded System 설계

임베디드 시스템이란 프로세서들이 들어가서 동작하는 제어 시스템을 말하며, PC등 일반적이고 다양한 기능을 수행하는 범용제품의 시스템과 대별되는 개념으로 특수 목적으로 이용하기 위해 그 기능과 구성을 최적화 한 것으로 독자적인 하드웨어와 운영체계를 갖는 독립된 시스템이다[4].

소프트웨어 다운로드는 soft handset을 위한 기술로써 SDR이 완전한 Reconfigurability를 이루기 위해서는 실시간 소프트웨어 다운로드는 불가피한 기술인 것이다. 그리고 이러한 SDR의 소프트웨어 다운로드를 구현하기 위해서는 실시간으로 다운로드 될 시스템이 필요하다. 일반적인 운영체제는 기껏해야 ms 단위의 정확성으로 제어를 하지만, 통신기기나 정밀 제어의 경우 수십 마이크로초 단위로 정확하게 시간을 측정해야 하는 경우가 많다. 또한 인터럽트가 발생했을 때 해당 프로세스나 태스크가 수십 마이크로초 이내로 동작해야 하는 경우에는 일반적인 범용 운영체제로는 이러한 조건을 만족할 수 없다. 그래서 RTOS(RealTime Operating System)을 지원하는 임베디드 시스템을 이용하여 다운로드 할 수 있는 실시간 오퍼레이팅 시스템 갖는 임베디드 시스템을 SDR용 하드웨어로 사용해야 한다. 아래 그림 2는 본 논문을 위해 제작한 임베디드 시스템 보드이며 보드 앞면과 뒷면을 보여 주며 블록은 각 칩의 기능을 나타내 주고 있다.

a. CPU - 최대 235MIPS로 동작하는 Intel사의 고성능, 저전력, 32-bit RISC 프로세서인 StrongARM SA1110를 사용했다. 데이터와 프로그램이 분리된 메모리 버스로 구성되는 Harvard 구조로 되어 있으며 5 State pipeline으로 병렬처리가 가능하다[5,6].

b. FLASH ROM - Linux Kernel 및 사용자 프로그램을 저장하는데 필요하며, 제작한 StrongARM보드에는 현재 32MB 용량인 Intel사의 StrataFlash Memory 두 개를 사용하여 최대 64MB까지 활용이 가능하다.

c. SDRAM - Flash ROM에 탑재 된 Kernel Image가 복사되어 프로그램이 실행되는 영역으로, PC-133

SDRAM이 장착되어 고속처리가 가능하다. 또한 32비트 메모리 버스를 가지고 있으므로 SDRAM의 고속성을 충분히 활용하고 있다. 제작한 StrongARM 보드에는 현재 32 MB 용량의 SDRAM을 두 개를 사용하여 최대 64 MB까지 활용이 가능하다.

d. Serial Port - 보드에는 두 개의 UART Port를 사용할 수 있도록 설계되어 있고, 외부에 Level Converter(MAX232 등)를 연결하여 RS-232C Port를 확보하고 있다. 차후 개발 할 보드는 나머지 Serial Channel를 추가하여 USB, IrDA등의 기능을 사용 할 예정이다.

e. JTAG - 프로세서(CPU)의 상태와는 상관없이 디바이스의 모든 외부 핀을 구동 시키거나 값을 읽어 들일 수 있는 기능을 제공한다. 디바이스 내에서 외부로 나가는 각각의 핀들과 일대일로 연결하여 모든 외부와의 연결점을 인터셉트 한다. 각각의 셀은 시리얼 쉬프트 레지스터(바운더리 스캔 레지스터)를 형성하기 위해서 서로 연결되어 있다. 전체적인 인터페이스는 5개의 핀에 의해 제어(TDI, TMS, TCK, nTRST, TDO)한다.

f. POWER - SA1110 보드에서 필요한 전압은 정격전압으로 DC 3.3V를 사용하고 있으나 프로세서 내부의 CPU Core 전압으로 206 MHz를 사용하는 SA1110BC에서는 DC 1.75V가 필요하다. 따라서 전원부는 이 두 가지 전압을 공급 할 수 있도록 구성해야 했다.

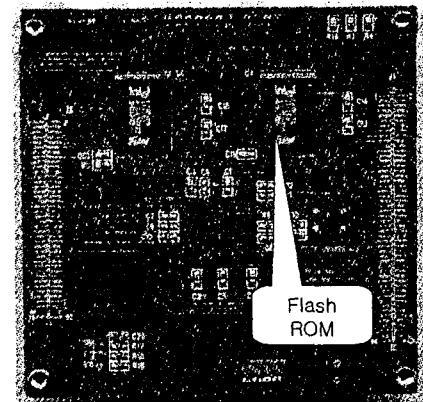
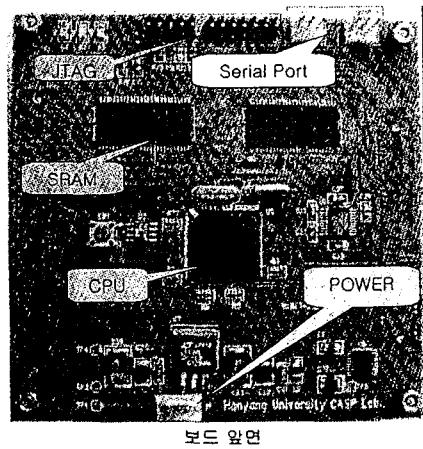


그림 2. 임베디드 보드

2) Linux Porting

리눅스란 유닉스와 거의 같은 운영 체제이지만, 유닉스의 장점을 포함하면서 새로운 여러 기능들이 적용되어 유닉스와 혼존하는 어느 운영 체제 보다도 강력하고 벼어난 운영 체제이다. 리눅스는 POSIX 운영체계 규격에 따라 SYSV와 BSD 확장을 덧붙여 완전히 새로 만든 독립 창작품 (independent implementation)이다. 소유권 문제가 있는 코드는 전혀 들어 있지 않다. 리눅스는 'GNU 공개 라이센스(GPL:GNU Public License)'에 따라 자유롭게 배포할 수 있다 [7].

임베디드 시스템에 리눅스를 O/S로 선정한 이유로는 누구나 제약 없이 자유롭게 사용할 수 있는 운영체제이며 항상 오픈 되어 있기 때문에 누가나 소스를 변형, 개발, 재배포 할 수 있는 특징을 가지고 있다. 그리고 한대의 시스템에서 여러 사용자들이 사용할 수 있는 다중 사용자 환경 시스템이며, 여러 가상 작업 공간을 사용자에게 제공하므로써 한 대의 컴퓨터 내에서 여러 개의 화면을 통하여 여러 작업들을 동시에 할 수 있는 다중 작업 및 가상 터미널 환경을 제공한다. 마지막으로 윈도우에서는 하드웨어를 설정할 때 반드시 하드웨어를 제조한 업체에서 제공한 드라이버를 사용해야 하지만, 리눅스에서는 하드웨어에서 제공한 드라이버가 아니더라도 하드웨어가 사용하고 있는 칩만 동일하다면 하나의 드라이버로 모든 제조업체의 하드웨어를 사용할 수 있는 드라이버 특성을 타지 않는 하드웨어를 지원한다 [8]. 이러한 리눅스를 이용하여 target 시스템인 StrongARM 보드에 포팅 하였으며, 포팅 순서는 그림 3과 같다.

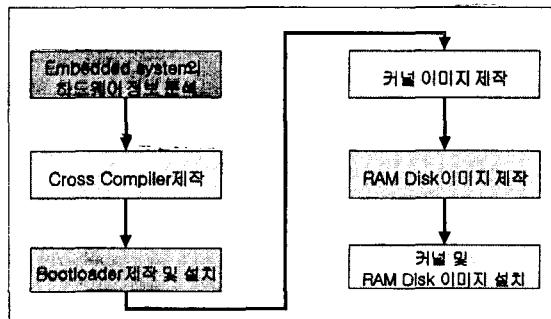


그림 3. 임베디드 시스템에 리눅스 포팅 순서도

a. Embedded system 정보 분석

CPU와 메모리 그리고 주변 장치에 대한 정보를 분석한다. 메모리 맵에 대한 정보로는 부트 로드와 커널 이미지 공간과 램 디스크 이미지를 저장 할 수 있는 플래쉬 메모리와 커널 이미지를 패치 시켜서 실제로 동작할 수 있는 공간이자 사용자 어플리케이션을 위한 메모리 영역인 램 그리고 주변 장치인 시리얼 포트, 이더넷 포트 등의 사용 여부와 I/O 맵에 대한 정보를 분석한다.

b. Cross compiler 제작

Host에서 원하는 소스 코드를 컴파일 하여 타겟 시스템에서 실행 할 수 있는 바이너리 파일을 만들어 줄 수 있는 Cross compile를 제작한다. 왜냐하면 컴파일

러와 라이브러리 저장시 많은 메모리를 필요로 하고 많은 양의 소스 코드의 저장이 불가능 하므로 컴파일하는데 많은 시간이 소요되기 때문이다.

gcc(GNU Compiler Collection) 소스를 구하여 ARM 용 Target Binary File을 생성 할 수는 크로스 컴파일러를 제작한다.

c. Boot Loader

부트 로더 기능으로는 새로운 커널 이미지를 플래쉬 메모리에 저장하고 저장 된 커널 이미지를 이용하여 시스템을 부팅시키고 램 디스크 이미지를 램에 패치시켜 준다. 부트 로더를 제작하고 설치 하기 위해서 C 언어로 작성하여 크로스 컴파일을 이용하여 부트 로더 소스 코드를 컴파일 하여 톰 라이터를 이용하여 플래쉬 메모리에 저장한다.

d. Kernel Image

커널이란 운영체계에서 가장 핵심적인 역할인 자원 즉 메모리 프로세서 등을 관리하며 시스템이 원활히 돌아갈 수 있도록 제어해 준다. 현재 우리가 리눅스라고 하는 것도 실제적으로는 리눅스란 운영체계 커널의 이름을 말하는 것이어서 확장된 의미로 사용되어 현재의 리눅스란 OS를 지칭한다.

타겟 CPU, 램 디스크 사용 여부, 네트워크 사용 여부, 그리고 주변 장치들을 위한 드라이버를 구동 시키기 위해 임베디드 시스템에 알맞은 커널 옵션을 설정한다. 그리고 크로스 컴파일러를 이용하여 커널 컴파일하고 부트 로더를 이용하여 커널 이미지를 플래쉬 메모리에 저장함으로써 커널 이미지를 설치한다 [9].

e. RAM Disk 이미지 제작

디바이스 드라이버, 시스템 설정파일, 기본적인 시스템파일 등을 저장 할 공간이 필요하고 하드 디스크는 고가이며 이를 사용하기 위해서는 커널이 IDE Device 와 File System을 지원해야 하므로 커널 사이즈가 커지고 많은 메모리가 필요하게 되어 임베디드 시스템에는 부적합하게 된다. 따라서 파일 시스템과 램 디스크 이미지를 만들어야 한다.

아래 그림 4는 리눅스가 임베디드 시스템에 포팅 되는 모습을 보여 주고 있다.

```

ttyp1 on 501100 UART3 (ctrl 17)
ttyp2 on 501100 UART2 (ctrl 16) using TIOBA
501100 Real Time Clock Driver v0.02
NetWinder Floating Point Emulator V0.95 (c) 1996-1999 Rebel.com
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 40K.
INIT: version 2.74 booting!
INIT: Entering runlevel: 3
Starting system logger: syslogd couldn't obtain descriptor for socket /dev/log
Address family not supported by protocol
Bringing up interface lo: No usable address families found.
socket: No such file or directory
/proc/net/route: No such file or directory
socket: Address family not supported by protocol
No usable address families found.
socket: No such file or directory
/etc/sysconfig/network-scripts/ifup-post: cd /var/run/reiserfsd; no such file or directory
Starting portmapper: Portmap
Starting rcmcia Starting PCMCIA services: module directory /lib/modules/2.4.3-rc1-k1-MP2/pcmcia not found
Starting INET services: inetc
Linux login: 
  
```

그림 4. 다운로딩 되어 부팅되는 화면

3) Software 다운로딩

Smart Card는 자체 프로그램된 메모리를 가진 특별한 목적을 갖는 micro-controller이다. 기존의 Chip Card (Magnetic Card)는 미리 프로그램 된 명령어를 수용 할 만큼만 간단한 회로를 가지고 있으며, 다시 프로그램 시키질 못한다. 이에 비하여 스마트카드는 EEPROM의 내용을 변경할 수 있어 유연성을 가진다[10].

본 논문에서 선보인 스마트 카드를 이용한 임베디드 시스템에 대한 다운로딩은 프로그램을 저장 할 많은 메모리 공간이 필요하고, 카드를 가지고 다녀야 하는 불편함이 있지만, 어려 없이 다운로드 할 수 있고, 공중파에 비해 속도가 빠르고, 네트워크에 대한 영향이 없으며, 다운로드 과정이 손쉽고, 인터셉션 가능성이 적다는 이점이 있다. 이런 장점을 이용하여 임베디드 시스템에 스마트카드를 이용하여 소프트웨어 다운로딩을 실현해 보았다.

스마트 카드의 종류에는 크게 접촉형과 비접촉형으로 나눌 수 있으며 접촉형 카드는 마이크로프로세서가 있는 카드와 없는 카드로 구분되고, 비접촉형은 단순히 시리얼 번호를 읽어오는 RF-ID 카드와 데이터를 읽고/쓰고 기본적인 연산기능을 갖고 있는 RF-IC 카드가 있으며, 이들의 장점만을 통한 Combi 카드가 있다.

그림 3은 일반적인 접촉식 스마트·카드의 구조를 보여준다. 본 논문에서는 처리비용과 유지비용이 낮아서 가장 많이 사용되고 있는 Contact Type Card[11.12]를 이용했다.

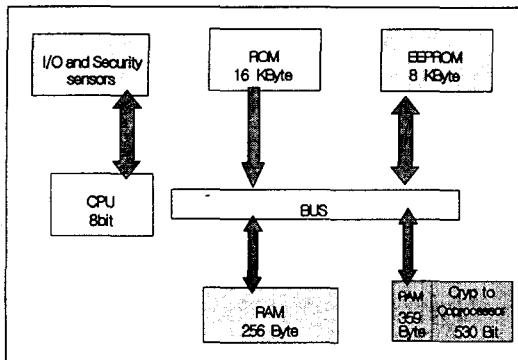


그림 5. 스마트 카드 구조

그리고 끝으로, 디바이스에 다운로딩을 하기 위해서는 여섯 단계의 절차가 필요 할 것으로 생각한다.

첫째, 초기화다. 다운로드 요청은 임베디드 시스템과 다운로드 소스 디바이스/서버에 의한 초기화 되어야 한다. 둘째, 다운로드 소스 디바이스와 사용자 단말기(임베디드 시스템)사이에 상호 인증이 이루어 져야 한다. 셋째는 단말기가 소프트웨어를 다운로드해서 설치하고 성공적으로 작동하도록 배열되어 있는지 확인하기 위해서 단말기의 Configuration, 유형 승인 데이터, API 변경, 하드웨어 및 소프트웨어, O/S 등의 메시지들의 전송이 약속되어 있어야 한다. 넷째, 다운로드 소스 디바이스와 사용자 디바이스 사이에 코드 유형 승인에 대한 정보, 다운로드 과정, 스케줄, 설치옵션 등 모든 과정을

제공하는 다운로드 승인 및 교환이 되어야 한다. 다섯째, 코드가 소스 디바이스로부터 단말기 내에 버퍼로 승인된 다운로드 스케줄링에 따라 다운로드 되면, 소프트웨어에 대한 무결성 검사를 통해 다운로드 받은 데이터에 어러가 발견되면 재전송이 실행되어야 한다. 여섯째, 코드가 단말기로 다운로드가 이뤄지면, 코드는 단말기 내에 인스톨 된다.

이와 같은 절차를 통해 소프트웨어 다운로드가 이뤄지는 것이 바람직 할 것이라고 생각된다.

3. 결 론

SDR 시스템을 구현하기 위해서는 다운로딩이 필요한데, 그에 따른 문제로는 다운로딩 과정의 복잡성, 시간과 전력 소모, 보안 문제나 소프트웨어 다운로딩 트래픽 문제, 로밍 지원 문제, 다운로드 될 소프트웨어에 대한 신뢰성 문제 등을 거론 할 수 있을 것이다.

본 논문에서 임베디드 시스템에 스마트 카드를 이용하여 다운로드를 했지만 스마트 카드 메모리 용량이 적어서 대용량의 프로그램을 저장해서 시스템에 다운로드하기 위해서는 고용량의 메모리가 먼저 해결되어야 소프트웨어 다운로드에 보편적으로 사용 할 수 있을 것이다.

실제적으로는 SDR 다운로드으로써의 기능을 수행하기 위해서 보다 더 많은 다운로딩 방법들이 연구 되어, 시스템에 안정적으로 운용 될 수 있어야 할 것이며 시스템에 여러 가지 수단으로도 다운로드 할 수 있도록 인터페이스가 구현 되어야 할 것이다.

참고문헌

- [1] J. D. Ralston, "The State of SDR Technologies," The 20th SDR Forum, Phoenix, France, Nov. 13-16, 2000.
- [2] SDRF Technical Report Chapter 6 Rev 0.3 - 28 November 1999
- [3] E.Del Re(Ed), "Software RadioTechnologies and Services" : Software Radio is Walking into Implementation Stage-pp82
- [4] 마이클 바 " Programming Embedded Systems", 한빛 미디어, 2000/08/26
- [5] <http://developer.intel.com/design/strong/schemes/27s/27879.htm>,
- [6] <http://download.intel.com/design/stron/datashts/27824105.pdf>
- [7] http://kelp.org/HOWTO/html/META-FAQ/META_FAQ-1.html
- [8] 서자룡, "리눅스 7.0 무작정따라하기" pp 58~60도서출판 혜지원
- [9] http://kldp.org/KoreaDoc/html/Kernel124_Intro-Kldp/kernel124_Intro-KLDP-1.html
- [10] <http://www.hellocec.net/network/smartcard.htm>
- [11] <http://ktwww.kotel.co.kr/smart/>
- [12] <http://www.jaeik.co.kr/scard/>