요 약

SDR(Software Defined Radio)은 여러 세대의 이동통신시스템, 무선방송 및 다양한 통신시스템 혼합하는 환경에서 소프트웨어 엔진그레이드로 무선장치 및 서비스를 제공할 수 있는 효율적인 해결방법으로 대두되고 있다. SDR이라는 기술은 하드웨어 수정 없이 모듈화된 소프트웨어의 변화만으로 단일의 송신수신시스템을 통해 다수의 무선통신규격을 통합 수행하는 폭넓은 인터페이스 기술이다[1]. 이러한 기술을 성립하기 위해서는 소프트웨어 다운로드 기술이 매우 중요하다. 따라서, 시스템에서 프로그램을 다운로드시 하드웨어가 Reconfiguration 되어 사용자가 원하는 대로 시스템이 선회될 수 있어야 하기 때문이다.

소프트웨어 다운로드 방안에는 크게 공중파를 이용하는 방법, 컴퓨터내 이동성의 기기로부터 다운로드 받는 방법, 전화와 케이블 서비스 등을 통해 근거리/근거리 호스트 서버로부터 소프트웨어를 다운로드 받는 방법, 네트워크를 통해 소프트웨어를 다운로드 하는 방법 등이 있다[2]. 이 중에서 본 논문에서는 SDR을 위한 방법으로 하드웨어를 유연성 있게 재구성하는 신개념의 4세대 이동통신시스템이다. 그리고 미래의 무선통신은 다양한 사용자의 요구로 다증가, 다중모드를 지원하는 단말기가 등장할 수밖에 없다[3].

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

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
그림 1. 시스템 구성도

1) Embedded System 설계

임베디드 시스템은 프로세서들이 돌아가면서 동작하는 제어 시스템을 말하며, PC들 일반적이고 다양한 기능을 수행하는 복잡한 시스템과 대비되는 개념으로 특수 목적으로 이용하기 위해 고성능을 요구한다는 특성의 하드웨어 운영체계를 갖춘 독립된 시스템이다[4].

소프트웨어 다운로드는 soft handset을 위한 기술로서 SDR이 완전한 Reconfigurability를 이루기 위해서는 실시간 소프트웨어 다운로드는 불가피한 기술이다. 이를 위해서는 실시간으로 다운로드 할 수 있는 시스템이 필요하다. 일반적인 운영체계는 기기의 ms 단위의 정확성을 요구하지만, 통신기기나 엔진 제어의 경우 수십 마이크로초 단위로 정확하게 시간을 측정해야 하는 경우가 많다. 또한 인터럽트가 발생했을 때 해당 프로세스나 테스크가 수행할 수 있도록 하기 위해서는 일반적인 병렬 운영체계보다 이러한 조건을 만족할 수 있다. 그래서 RTOS(RealTime Operating System)을 이용하여 임베디드 시스템을 동작하려면 다운로드 할 수 있는 실시간 시계의 시스템을 갖춘 임베디드 시스템을 SDR로 하여 동작해야 한다. 이를 그림 2는 본 논문에 적재한 임베디드 시스템 보드이며 보드 안면에 보드를 보여 주며 블록은 각 형태의 기능을 나타낸 것이다.

a. CPU - 최대 235MIPS으로 동작하는 Intel의 고성능, 저전력, 32-bit RISC 프로세서인 StrongARM SA1110을 사용했다. 테이터와 프로그램이 동시에 흐르는 메모리 버스로 구성되는 Harvard 구조로 되어 있으며 5 state pipeline으로 빠르게 처리가 가능하다[5,6].

b. FLASH ROM - Linux Kernel 단순화를 위해서도 필요하며, 제작한 StrongARM 보드에는 최대 32MB 용량이 Intel의 Stratix Flash Memory 두 개를 사용하여 최대 64MB까지 활용이 가능하다.

c. SDRAM - Flash ROM에 탑재된 Kernel Image가 복사되어 프로그램이 실행되는 영역으로, PC-133 SDRAM이 참가되어 고속작성이 가능하다. 또한 32비트 메모리 버스를 가지고 있으므로 SDRAM의 고속성은 충분히 활용하고 있다. 제작한 StrongARM 보드에는 병렬 32MB 용량의 SDRAM을 두 개를 사용하여 최대 64MB까지 활용이 가능하다.


e. JTAG - 프로세서(CPU)의 상태와는 상관없이 미바이어스의 모든 외부 장치 구동 시기나 갭을 없이 동일할 수 있는 기능을 제공한다. 미바이어스 내에서는 외부로 나가는 각각의 패널과 일대일로 연결하여 각 외부 장치의 연결길이 인터럽트한다. 각각의 채널 셋은 시리얼 프로토콜 테스터(바이러시 스캔 테스처)는 형성하기 위해서 서로 연결되어 있다. 전체적인 인터페이스는 5개의 편에 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 205 mV를 사용하는 SA1110BC에서 DC 1.75V가 필요하다. 따라서 전원부는 두 가지 전압을 공급할 수 있도록 구성해야 했다.

그림 2. 임베디드 보드
Linux Porting

리눅스와 유닉스의 인터페이스는 내부적으로 일치하지만, 유닉스의 장점으로는 새로운 여러 기능들이 적용되어 있으며, 커널이 경향을 보이지 않아도 대체하고 빌드하여 운영체제가 된다. 리눅스는 POSIX 운영체제 규격에 따라 SYSV 및 BSD 헤지름을 맞추어 원도우 새롭게 만든 독립 경향(independent implementation)이다. 소유권 문제와 있는 코드는 전혀 들어 있지 않다. 리눅스는 GNU 공개 라이센스(GLP GNU Public License)에 따라 자유롭게 배포할 수 있다.[7]

嵌入式 시스템에 리눅스를 OS로 선택한 이유는 누구나 어렵지 않게 자유롭게 사용할 수 있는 운영체제 이며 항상 오류가 없기 때문에 누구나 소스를 복제, 개발, 재배포 할 수 있는 특성을 가지고 있다. 그리고 반대의 시스템에서 여러 사용자가 사용할 수 있는 대중 사용자 환경 시스템이며, 여러 가지 작업 공간을 사용자에게 제공하므로 한 대의 컴퓨터 내에서 여러 개의 화면을 통하여 여러 작업들을 동시에 할 수 있는 대중 작업 및 가정 휴대용 화면을 제공한다. 마치마이로 윈도우에서는 하드웨어를 설정할 때 반드시 하드웨어를 제조한 업체에서 제공한 드라이버를 사용해야 하지만, 리눅스에서는 하드웨어에서 제공한 드라이버가 아닌라도 하드웨어를 사용하고 있는 원도우 화면 동일하다면 하나의 드라이버만 제공조심의 하드웨어를 사용할 수 있는 드라이버 드립스를 타기 위한 하드웨어를 지원한다[8]. 이러한 리눅스를 이용하여 target 시스템의 StrongARM 보드에 포팅 하였으며, 포팅 순서는 그림 3과 같다.

그림 3. 엑세드 시스템에 리눅스 포팅 순서도

a. Embedded system 정보 분석
CPU와 메모리 그리고 주변 장치에 대한 정보를 분석한다. 메모리 네트워크에 대한 정보는 부트 로드과 커널 이미지 저장 할 수 있는 플래시 메모리와 커널 이미지를 체계에 설치하게 동작할 수 있는 공간이기 때문에 사용자 이외의 시스템을 위한 메모리 영역의 그림 그리고 주변 장치인 시리얼 포트, 기타 네트워크 사용 여부를 I/O 요소로 정리한 정보를 분석한다.
b. Cross compiler 제작
Host에서 연산하는 소프트웨어(컴파일)의 테스트 시스템에서 실행 할 수 있는 바이너리 파일을 만들어 줄 수 있는 Cross compile을 제작한다. 데나하면 컴퓨터들 라이브러리 저장 소프트웨어 필요로 하고 많은 양의 소스 코드의 저장이 불가능 하므로 컴파일 하는데 많은 시간이 소요되기 때문이다.
gcc(GNU Compiler Collection) 소스를 구하여 ARM용 Target Binary File을 생성 할 수는 크로스 컴파일을 하였다.
c. Boot Loader
부트 로더 기능으로는 새로운 커널 이미지를 플래시 메모리에 저장하고 거기에 커널 이미지를 이용하여 시스템을 런시시키고 램 디스크 이미지를 플래시에 저장시킨다. 부트 로더를 제작하는 설계하기 위해서 C 언어로 작성하여 크로스 컴파일을 이용하여 부트 로더 소스 코드를 컴파일 하여 동 라이브러리를 이용하여 플래시 메모리에 저장한다.
d. Kernel Image
커널이란 운영체제에서 가장 핵심적인 역할을 하는 측에서 메모리 프로세서 등에 불과하되 시스템이 완벽히 돌아갈 수 있도록 제어해 준다. 현재 우리가 리눅스 하고 하는 것도 실제로는 리눅스 운영체제 커널의 이들을 담당하는 것이어서 확장된 역할을 살아서 현재의 리눅스의 OS를 저장한다.
 Cash CPU, 램 디스크 사용 여부, 네트워크 사용 여부, 그리고 주변 장치들 위드 드라이버 구동 시기 위치에 따라 디스크 시스템에 얼마나 커널 옵션은 설정한다. 그리고 크로스 컴파일을 이용하여 커널 컴파일하고 부트 로더를 이용하여 커널 이미지를 플래시 메모리에 저장함으로써 커널 이미지로 설정한다[9].
e. RAM Disk 이미지 제작
다바이스 드라이버, 시스템 설정파일, 기본적인 시스템 파일들은 저장할 공간이 필요하고 하드 디스크는 고가이며 이를 사용하기 위해서는 커널의 IDE Device와 File System을 지원해야 하므로 커널 이미자가 되어 얻어진 시스템은 비용을 하루하나 받을 수 있는 크로스 컴파일이 필요하다.
아래 그림 4는 라이브러리가 엑세드 시스템에 포팅 되는 모습을 보여 주고 있다.

그림 4. 다운로딩 되어 부팅되는 화면  

- 827 -
3) Software 다운로딩

Smart Card는 자체 프로그램될 메모리를 가진 특별한 목적을 갖는 micro-controller이다. 기존의 Chip Card (Magnetic Card)는 미리 프로그램된 명령어를 수행할만한 간단한 회로를 가지고 있으며, 다시 프로그램 시키기 어렵다. 이에 비해 스마트카드는 EEPROM의 내용을 변경할 수 있어 유연성을 가진다.[10]

본 논문에서 선보이는 스마트 카드를 이용한 임베디드 시스템에 대한 다운로딩 프로그램을 다루는 것 중 하나인 메모리 공급 기계로, 카드를 가지고 다니야 하는 불편함이 있었지만, 여러 것이 다운로드 할 수 있고, 공중 과제에 비해 속도가 빠르고, 네트워크에 대한 영향이 없으며, 다운로드 과정이 손쉽고, 인터럽트 가능성이 적다는 이점이 있다. 이런 장점들을 이용하여 임베디드 시스템에서 스마트카드를 이용하여 소프트웨어 다운로딩을 실현해 보았다.

스마트 카드의 종류에는 크게 접촉형과 비접촉형으로 나눌 수 있으며 접촉형 카드는 마이크로프로세서가 있는 카드와 없는 카드로 구분된다. 비접촉형은 단순히 시스템 빌드 번호를 임의로 RF-1D 카드와 데이터를 읽고/쓰고 기본적인 연산기능을 갖고 있는 RF-1C 카드가 있으며, 이들의 장점들은 카드에 Combi 카드가 있다.


그래프 5. 스마트 카드 구조

그리고 앞으로, 다이버스에 다운로딩 하기 위해서는 여러 단계의 절차가 필요할 것으로 생각된다.

첫째, 초기화다. 다운로드 요청은 임베디드 시스템과 다운로드 소스 다이버스에서의 협의를 필요로 한다. 둘째, 다운로드 소스와 사용자 별단말기(임베디드 시스템)에 의해 상호 입증이 이루어져야 한다. 셋째는 단말기가 소프트웨어 다운로드한 다음 설치하고 성공적으로 작동하도록 배포되어 있는지 확인하기 위해서 단말기를 Configuration, 유형 숫자 데이터, API 변경, 하드웨어 및 소프트웨어, OS 등에 설치된 전송이 약속되어 있어야 한다. 넷째, 다운로드 소스 다이버스와 사용자 다이버스 사이에 코드 유형 숫자에 대한 정보, 다운로드 과정, 설치된 모든 과정을 제공하는 다운로드 승인 및 교환이 되어야 한다. 다섯째, 코드가 소스 다이버스로부터 단말기 내에 비로 승인된 다운로드 스케줄링에 따라 다운로드 되며, 소프트웨어에 대한 원활한 검사를 통해 다운로드 받은 데이터에 문제가 발생하면 재전송이 실행되어야 한다. 여섯째, 코드가 단말기로 다운로드가 이뤄지면, 코드가 단말기 내에 인식된다.

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

3. 결론

SDR 시스템을 구현하기 위해서는 다운로딩이 필요하며, 그에 따른 문제로서는 다운로드 과정의 불균형, 시간과 전력 소모, 보안 문제나 소프트웨어 다운로드 데이터 문제, 모델 지원 문제, 다운로딩 및 소프트웨어에 대한 신뢰성 문제 등을 겪는 수가 점점 있을 것으로 보이다.

본 논문에서 임베디드 시스템에 스마트 카드를 이용하여 다운로드를 했지만 스마트 카드 메모리 용량이 적어서 다운로드의 프로그램을 저장하는 시스템에 다운로딩하기 위해서는 다운로드 메모리가 여러 해제되어야 소프트웨어 다운로딩에 보안적으로 사용될 수 있을 것이다.

실제적으로는 SDR 다운로딩으로서의 기능을 수행하기 위해서 보다 많은 다운로딩 방법들이 연구되어, 시스템에 안전적으로 운용 될 수 있어야 할 것이며 시스템에 여러 가지 수단으로도 다운로딩 할 수 있도록 인터페이스가 구현되어야 할 것이다.

참고문헌

[8] 서송호, “디스크 7.0 무작정 바라보기” pp 58-60도 사용중 복제권