

내장형 리눅스를 이용한 블루투스 개인 휴대용 단말기에 관한 연구

손재기*, 민수영*, 박창원*
*전자부품연구원 정보시스템 연구센터
e-mail : {jgson, minsy, parkcw}@keti.re.kr

A Study on Bluetooth PDA for Embedded Linux

Jae Gi Son*, Soo Young Min*, Chang Won Park*
*IT System Research Center, Korea Electronics Technology Institute

요 약

최근 내장형 리눅스에 대한 연구가 활발히 진행되고 있다. 또한 개인 휴대용 단말기에 블루투스와 같은 네트워크 장치의 탑재에 관한 연구들이 진행 중에 있다. 본 연구는 내장형 리눅스 기술을 바탕으로 한 블루투스 개인 휴대용 단말기에 관한 것이다. 내장형 리눅스가 개인 휴대용 단말기와 같은 작은 장치들에 탑재할 수 있는 방법과 포켓 리눅스를 통한 어플리케이션 개발에 대해 기술한다. 또한 블루투스를 이용한 인터넷 액세스에 관하여 기술한다.

1. 서론

최근 내장형 리눅스에 대한 연구가 정보기기 및 정보 산업 분야에서 활발히 진행되고 있으며, 많은 분야에서 리눅스의 활용성에 대한 연구가 진행 중이다. 또한 내장형 리눅스를 이용한 개인용 휴대 장치에 대한 연구가 활발하게 진행 중이다. 블루투스는 1~10m 내의 통신망을 구축할 수 있는 무선 인터페이스 기술이다. 이러한 블루투스와 개인용 휴대용 단말기의 결합에 관한 연구가 진행 중이다.

본 논문은 이러한 내장형 리눅스 기술을 바탕으로 한 블루투스(Bluetooth)를 탑재한 개인 휴대용 단말기(PDA: Personal Digital Assistant)에 관한 연구이다.

2 장에서는 개인 휴대용 단말기와 관련된 연구에 대해 기술하고 3 장에서는 개발 환경에 대해, 4 장에서는 개발 전반에 대해, 5 장에서는 개발 결과에 대해, 마지막으로 결론 및 향후 연구과제에 대해 기술한다.

2. 관련 연구

내장형 리눅스(Embedded Linux)

내장형 운영체제는 전자 제품에 들어가는 운영체제를 말하는 데 윈도우 CE 나 팜(Palm) OS 등이 대표적이다. 내장형 운영체제의 가장 중요한 요건은 가격과 안정성이다. 리눅스는 기존의 상용 운영체제에 비해

비교도 안 될 장점을 가지고 있다. 가격면에서는 거의 무료라고 할 수 있으며 기술만 있으면 누구나 사용할 수 있다. 안정성면에서는 적용사례가 많지 않지만, 리눅스의 안정성은 이미 많은 부분에서 확인된 바 있다. 또한 많은 개발자들의 참여로 인해 상용 운영체제와 견줄만한 안정성을 지니고 있다. 이러한 이유에서 최근 리눅스를 이용한 내장형 기술이 국내외에서 많이 연구되고 있으며 최근 내장형에 적합한 커널 버전을 새로 발표해 새로운 활력을 불어 넣고 있다.

리눅스가 내장형 운영체제 분야에서 각광 받는 이유는 여러 가지가 있지만 오픈 소스 정책이 가장 큰 이유이다. 하드웨어 개발자의 시각에서 본다면 자기가 개발한 하드웨어에 적합하게 소스를 수정할 수 있다는 것은 개발이 수월할 뿐 아니라 개발 기간 단축이라는 측면에서 엄청난 장점을 가진다. 또한 기술력만 있으면 공개된 리눅스 소스를 이용하여 다양한 분야에 활용할 수 있다. 또 다른 이유는 리눅스가 이미 다양한 하드웨어에 이식되어 있는 다는 것이다. CPU 의 경우 인텔 계열, 알파, 파워 PC, 밍스, ARM 계열, SH 계열을 지원하는 등 많은 CPU 를 지원하고 있으며 각종 장치들도 지원을 하고 있다. 장치 드라이버의 개발도 비교적 용이하다.

포켓 리눅스

포켓 리눅스[4]는 자바를 바탕으로 한 어플리케이션 프레임 워크(FrameWork)이다. 자바를 이용하기 때문에 개발자는 쉽게 어플리케이션을 작성하고 배포할 수 있다. 또한 개발자는 XML의 사용으로 어플리케이션을 쉽고 유연성 있게 작성할 수 있다. 수행 성능과 관련된 하드웨어 제어 부분은 자바의 네이티브(native) 코드를 통해 액세스한다.

리눅스의 장점인 이식성을 바탕으로 새로운 장치나 플랫폼에 쉽게 이식할 수 있다. 포켓 리눅스는 PDA나 셀룰러 폰, 페이지, 인터넷 TV와 같은 정보기에 효과적으로 이식될 수 있다.

3. 개발 환경

개인 휴대용 단말기 개발을 위한 환경은 다음과 같다.

개발 시스템 보드는 팜팜테크[5]에서 제공하는 타이눅스 박스(TynuxBox)를 이용하여 개발하였다. 타이눅스 박스는 인텔 스트롱암 프로세서(Intel StrongARM SA-1110)를 탑재하고 있으며 디스플레이 장치는 5.7인치 LCD(320x240) 터치 스크린을 탑재하고 있다. 또한 네트워크 인터페이스, USB 인터페이스, 사운드, 3개의 시리얼 인터페이스(RS-232C)를 제공한다.

블루투스 접속을 위해 CSR의 CASIRA 블루투스 개발자 키트를 사용하였다. 블루투스 모듈과의 인터페이스는 UART를 사용하였다.

임베디드 리눅스를 포팅하기 위한 호스트 시스템으로 미지 리눅스 2.0을 사용하였으며 ARM 플랫폼을 위한 크로스 컴파일러(Cross Compiler)로 GCC[6]를 설치하였다.

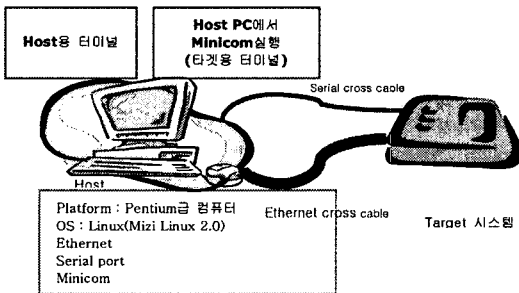


그림 1 개발 환경

4. 개발

개인 휴대용 단말기의 운영체제는 리눅스를 이용하였다. 개발 커널은 2.4.0 버전을 이용하여 개발하였다. 커널 버전 2.4.0은 기존의 버전에 비해 내장형 시스템을 위해 많은 부분을 최적화 시켰으며 기본 버전에서 모듈화 되지 않은 부분을 모듈화 시켰다. 커널의 크기는 640K를 넘지 않도록 하였으며 각 모듈을 압축하여 필요 시에 로딩할 수 있도록 하였다.

커널 구성

개발 커널은 버전 2.4.0을 사용하였다. 커널 설정을

통해 불필요한 부분을 제외시켰다. 본 연구에서 사용한 타겟 보드는 인텔의 스트롱암 프로세서(SA-1110)로 개인 휴대용 단말기와 같은 정보기기를 개발하는데 적합하다. 커널 설정시 SCSI 디바이스와 HDD 디바이스와 같은 불필요한 부분을 제외시켰다. 실제 개인 휴대용 단말기를 구성할 수 있는 부분만을 선택하여 구성하였다. 파일 시스템으로 저널링 플래시 파일 시스템(JFFS)을 선택하여 플래시 메모리를 사용하였다.

또한 LCD에 이미지 데이터를 빠르게 쓰기 위해 프레임 버퍼(FrameBuffer)를 사용하였으며 터치 스크린을 제어하기 위해 터치 스크린 모듈을 설정하였다.

루트 파일 시스템(root filesystem) 구성

루트 파일 시스템은 MTD(Memory Technology Device)[7]를 이용하여 구성하였다. MTD 서브시스템은 리눅스 시스템에서 플래시 메모리 디바이스를 사용하기 위해 개발되었다. 각 제조사에 따라 다른 플래시 메모리 디바이스에 대해 일관되고 공통된 소프트웨어 인터페이스를 제공한다. MTD는 사용자 모듈과 드라이버 모듈 두 가지로 구성되어 있으며 JFFS(Journaling Flash File System)[8]를 사용하여 데이터 손실이 없는 신뢰성 있는 임베디드 시스템을 구성할 수 있다.

MTD를 이용한 파일 시스템의 구조는 다음과 같다.

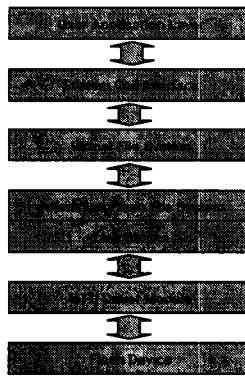


그림 2 플래시 메모리 블록 디바이스 구조

JFFS는 일반적인 파일 시스템 대신 플래시 메모리의 특성을 고려한 파일 시스템으로 기존의 파일 시스템이 메타 정보를 특정 블록에 고정하고 파일 시스템이 수정될 때마다 해당 블록을 갱신하던 것을 JFFS는 플래시 메모리의 특성에 맞게 특정 블록을 고정해 사용하지 않고 플래시 메모리 전 영역을 고르게 사용할 수 있게 한다. 또한 비동기적 전원 실패에 대비해 데이터 손실이 없도록 하기 위해 일정 시간마다 전체 파일 시스템의 메타 정보를 로깅(logging) 한다.

MTD와 JFFS를 이용하여 플래시 메모리 디바이스에 접근하여 루트 파일 시스템을 구성하여 타겟 시스템의 루트 디바이스를 /dev/mtdblock3으로 설정하였다.

표 1은 타겟 시스템의 플래시 디바이스 파티션 테이블 정보를 표시한 것이다.

표 1 플래시 디바이스 파티션 테이블

파티션	디바이스 파일	크기	이름
1	mtdblock0	0x00200000	loader+scripts
2	mtdblock1	0x00200000	kernel
3	mtdblock2	0x00400000	configuration
4	mtdblock3	0x00800000	root filesystem

블루투스 스택

리눅스에서 블루투스 모듈을 제어하기 위해 UART를 이용하였다. 현재 리눅스 커널에는 블루투스 장치를 위한 모듈이 있으며 본 연구에서는 BlueZ 1.2[9]를 사용하여 블루투스 모듈과 연결 및 제어를 하였다.

BlueZ 1.2는 다음과 같이 구성되어 있다.

- HCI Core
- HCI UART, USB and Virtual HCI device drivers
- L2CAP protocol module
- Configuration and testing Utilities

BlueZ 1.2에서는 CSR의 CASIRA[10] 모듈에 대한 인터페이스를 제공하지 않고 있다. CASIRA 모듈에 접근하기 위해 CASIRA 모듈의 설정 값을 변경하여 UART로 인터페이스 하도록 하였다. 또한 BlueZ 내부 소스에서 CASIRA 모듈에 맞도록 패치를 적용하였다.

블루투스 서비스를 제공하기 위해 몇 가지 유틸리티를 적용하였다. 블루투스와 TCP/IP 네트워크를 연결하기 위해 점대점(Point-to-point) 프로토콜을 사용하였다. 블루투스 액세스 포인트 역할을 하는 서버와 클라이언트 역할을 하는 개인 휴대용 단말기에 점대점 프로토콜을 설치하고 블루투스 rfcomm 프로토콜로 연결하였다. 그리고 NAT(Network Address Translation) 서비스를 통해 블루투스 액세스 포인트와 인터넷을 연결하였다.

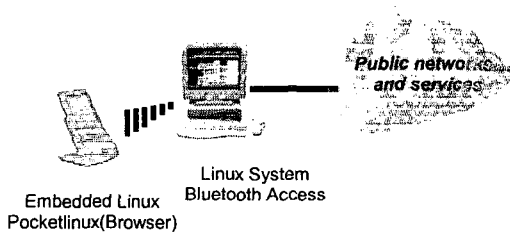


그림 3 블루투스를 이용한 인터넷 접속

포켓 리눅스 포팅

포켓 리눅스는 자바 가상머신 상에서 동작하는 프레임 워크이다. 포켓 리눅스가 동작하기 위해서는 타겟 시스템에 자바 가상머신(JVM)을 포팅하여야 한다. 자바 가상머신의 포팅을 위해 Kaffe OpenVM[11]을 사용하였다. 인텔 스트롱암에서 동작하기 위해 크로스 컴파일러로 Kaffe를 컴파일하였다. 일반적으로 자바 가상머신은 실행 파일에 비해 실행 속도가 느리다. 또한 사용자 인터페이스 관리자에 의해 그래픽을 출력하기 때문에 사용자 인터페이스 관리자에 의존적이다.

윈도우나 X-윈도우와 같은 사용자 인터페이스 관리

자는 내장형 시스템에 적용하기에는 상당히 덩치가 크다. 자바 가상머신을 내장형 시스템에 적용하기 위해 자바 가상머신의 그래픽 출력 부분을 수정하여 내장형 시스템으로 직접 출력할 수 있다. 최근의 그래픽 인터페이스 카드는 프레임 버퍼를 사용할 수 있으므로 자바 가상머신의 그래픽 출력 부분을 프레임 버퍼를 사용하여 빠르게 출력하도록 하였다.

또한 자바 가상머신은 기본적으로 사용자 인터페이스 관리자 상에서 동작하기 때문에 키보드나 마우스 같은 입력을 제공하지 않기에 자바 가상머신에 입력 부분을 추가해야 한다.

포켓 리눅스를 위한 자바 가상머신에서 몇 가지 에러가 있기 때문에 디버깅하여 본 연구에서 사용한 목적 보드에 탑재하였다.

시스템 최적화

리눅스 시스템의 라이브러리를 최적화하거나 최소화 시킴으로써 시스템을 최적화할 수 있다. 최근의 리눅스 시스템의 라이브러리 크기는 이전의 라이브러리에 비해 많은 기능을 가지고 있기 때문에 방대해져 있으므로 이전의 라이브러리를 사용하였다. 또한 원래의 라이브러리는 디버깅 정보와 같은 것을 포함하고 있기 때문에 이러한 불필요한 정보들을 제거함으로써 최적화하였다.

시스템 유틸리티 최적화

리눅스 시스템의 유틸리티는 내장형 시스템에 적용하기에는 부담스런 크기를 가진다. 내장형 시스템을 제어하기 위한 유틸리티만 선택하고 불필요한 유틸리티를 제거함으로써 최적화할 수 있다. 각 내장형 시스템의 목적에 따라 시스템 유틸리티를 선택하여 구성하면 된다. 본 연구에서는 개인 휴대용 단말기를 위해 최소한의 유틸리티를 선택하였다. 예를 들어 네트워크 관련된 유틸리티, 시스템 관리용 유틸리티 등으로 구성하였다.

또한 시스템 유틸리티는 최적화된 소형으로 구성하였다. BusyBox, TinyLogin, Ash 유틸리티를 사용하여 시스템 유틸리티를 최적화하였다.

BusyBox[12]는 기존 리눅스의 시스템 유틸리티에 비해 최적화된 환경을 제공한다. 다중 호출(multi-call) 바이너리 형태의 소형 유틸리티이다. 입출력 부분과 공통된 부분을 한꺼번에 묶어 다중 호출에 의해 각 유틸리티를 호출하고 있다.

TinyLogin[13]은 시스템 로그인에 관련된 유틸리티를 하나로 합친 것이다. TinyLogin은 시스템 보안을 향상시키기 위해 Shadow Password 기능을 제공한다. 기존 리눅스의 로그인 관련 유틸리티에 비해 크기를 최적화할 수 있다.

Ash[14]은 필수적인 셸(shell)의 기능만을 제공한다. 기존 Bash에 비해 크기가 작으며, POSIX 인터페이스 규격을 지원한다. 본 연구에서 시스템 셸로 사용하여 최적화하였다. Ash을 사용하기 위해 기존의 스크립트(script) 파일을 수정하여 최적화하였다. 리눅스 커널의 부팅시 init 프로세스를 실행시키고 init 프로세스에 의

해 시스템 콘솔 및 루트 파일 시스템 마운트 등을 수행한다. 또한 /etc/inittab 파일을 참조하여 하위의 새로운 프로세스들을 생성한다. 하위의 새로운 프로세스들은 각 목적 시스템에 맞게 수정하여 사용할 수 있다. 기본적인 리눅스 시스템에서는 많은 프로세스들이 동작하므로 이 부분을 최적화하여 실행 프로세스의 수를 줄였다.

5. 개발 결과

내장형 리눅스와 포켓 리눅스 프레임 워크를 이용한 개인 휴대용 단말기는 기존의 윈도우 CE 에 비해 손색 없는 사용자 인터페이스를 제공한다. 그림 4는 타겟 보드에 내장형 리눅스와 포켓 리눅스의 구동화면이다.

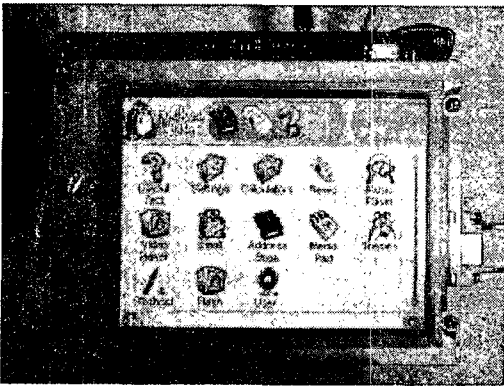


그림 4 Pocketlinux 구동 화면

6. 결론 및 향후 연구과제

현재 많은 분야에서 내장형 리눅스를 이용한 시스템이 개발 중이다. 본 연구는 개인 휴대용 단말기의 운영체제로 리눅스를 사용하였으며 블루투스를 이용하여 인터넷에 접속할 수 있도록 하였다. 또한 포켓 리눅스와 자바 가상머신을 이용하여 그래픽 사용자 인터페이스(GUI)를 구성하였기 때문에 손쉽게 응용 프로그램을 개발, 탑재 할 수 있다.

향후 연구과제로 리눅스의 경우 개인 휴대용 단말기와 같은 정보기에 필수적인 저전력 기능이 부족하다. 이러한 저전력 기능은 내장형 리눅스가 휴대용 정보기기에서 우위를 차지하기 위한 필수 조건이다. 또한 윈도우 CE 와 같은 운영체제에 비해 응용 프로그램의 수가 적다. 다수의 응용 프로그램 개발은 내장형 리눅스가 성공할 수 있는 조건이다.

참고문헌

[1] Brian Ward, "The Linux Kernel HOWTO", <http://www.linux.org>.
 [2] Remy Card, Eric Durnas, and Frank Mevel, The Linux Kernel book, 2ed., John Wiley & Sons, 1998.
 [3] Daniel P. Bovet and Marco Cesati, Understanding the Linux Kernel, O'Reilly & Associates Pub., 2001.

[4] PocketLinux, <http://www.pocketlinux.com>.
 [5] TynuxBox, <http://www.palmpalm.com>.
 [6] GCC, <http://gnu.org>.
 [7] MTD, <http://www.linux-mtd.infradead.org>.
 [8] JFFS, <http://developer.axis.com/software/jffs>.
 [9] BlueZ, <http://bluez.sourceforge.net>.
 [10] CSR, <http://www.csr.com>.
 [11] Kaffe, <http://www.kaffe.org>.
 [12] BusyBox, <http://busybox.lineo.com>.
 [13] TinyLogin, <http://tinylogin.lineo.com>.
 [14] Ash, <http://www.debian.org/Package/unstable/shells/ash.html>.