

임베디드 리눅스를 이용한 위치기반 관광안내 단말기의 설계 및 구현

김정원* · 임병진**

*신라대학교 컴퓨터정보공학부

Design and Implementation of LBS-based Tour Guide System using Embedded Linux

Jeong-Won Kim* · Byung-Jin Lim**

**Department of Computer & Engineering, Silla University

E-mail : jwkim@silla.ac.kr, database@silla.ac.kr

요 약

본 논문에서는 임베디드 리눅스를 이용하여 위치기반 관광정보안내 단말기를 설계 및 구현하였다. 구현된 단말기는 GPS 인터페이스를 통해 단말의 현재 위치를 파악하고 해당 관광지의 정보를 Qt-embedded 윈도우상에서 디스플레이한다. 논문의 단말기는 타겟보드상에서 부트로더와 임베디드 리눅스 커널 및 각종 디바이스 드라이버가 구현되었으며 GUI로는 임베디드 윈도 매니저인 Qtopia를 이용하여 애플리케이션을 개발하였다. 구현 결과 무료 운영체제를 이용한 모바일 관광안내 단말기로서 기능을 충분히 수행할 수 있음을 확인하였고, 향후 연구과제로는 통신 기능을 추가하여 단말기의 관광 정보를 실시간에 업데이트할 수 있는 기능을 구현하는 것이다.

ABSTRACT

This paper has implemented a tour guide terminal using the embedded linux. The implemented terminal can locate its position using GPS interface and display the sightseeing information of the current location. With view of this implementation results, we confirmed the possibility of embedded linux terminal as a tour guide PDA. The realtime update of sightseeing information using a mobile communication functionality will be included in our future works.

키워드

Embedded Linux, LBS, Qt

1. 서 론

최근 PDA, 핸드폰, GPS, 무선통신의 발달로 이동통신과 무선단말기를 결합한 지도 검색과 위치 파악 및 추적서비스 분야가 새로운 이머징마켓으로 등장하고 있고, 이를 실현하기 위한 무선 단말기 개발과 위치기반서비스(LBS: Location-Based Service) 기술 개발이 새로운 emerging technology로 등장하고 있다[1,2,3].

위치기반서비스(LBS)는 GPS 칩을 내장한 휴대

폰이나 PDA 단말기를 휴대한 사용자가 자신의 위치를 기반으로 다양한 응용 서비스를 제공받을 수 있다. 현재 위치기반서비스를 위한 국내 여건은 GPS수신기, 휴대폰이나 PDA, 이동통신사의 무선통신 기술, 상세한 전자지도의 CP(Contents Provider)가 모두 충족되어 있어 LBS를 위한 시장성과 필요한 기술 요소를 모두 충족하고 있다[4, 5].

현재 위치기반서비스를 위한 요소기술들을 모두 충족하고 있지만, 위치기반서비스는 물류 택배

와 같은 큰 규모의 사업장에서만 개발되어 사용되고 있다. 이러한 이유는 PDA의 가격이 비싸고, 위치기반서비스 개발 기술이 보편화되지 않았기 때문이다[6,7,8]. 논문에서는 고가의 외국산 PDA를 사용하지 않고, 무료 운영체제인 리눅스를 탑재한 위치기반서비스를 위한 전용 모바일 단말기를 개발하고, 또한 국내 기술로 PDA를 위한 위치기반서비스 엔진을 개발하였다.

본 논문에서는 임베디드 리눅스를 탑재하여 위치기반 관광 안내서비스를 할 수 있는 단말기를 개발하고 Qt[9] 기반의 애플리케이션을 개발하였다. 논문의 구성은 다음과 같다. 2장에서는 구현된 단말기를 소개하고, 3장에서는 임베디드 리눅스 커널, 부트로더, USB, 키패드등 대표적인 디바이스 드라이버 구현결과를 소개하며, 4장에서는 클라이언트 프로그램의 구현 내용을 기술하며 5장에서 결론을 제시한다.

II. 단말기의 설계 및 구현

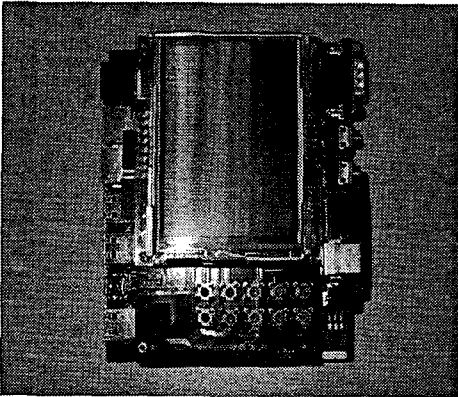


그림 4. 개발된 타겟보드

표 1. 주요 사양

디바이스	주요 사양
CPU	Intel PXA255(400MHz)
메모리	Flash 32MB, RAM 128MB
디스플레이	TFT LCD 3.5"
통신장치	Ethernet 2port, Serial, Bluetooth, PCMCIA, USB 2.0
저장장치	CF
입력장치	10 Keypad

그림 1은 개발된 타겟보드이다. 초기 모델은 하나의 보드에 각종 디바이스를 통합한 시범 모델이었으나 최종 개발된 프로토타입은 스택형 방식으로 각종 디바이스를 포함하고 있다. 다음 표 1은 보드의 주요 사양이다.

III. 임베디드 리눅스 포팅

(1) 부트로더 개발

원래 u-boot는 PPC용으로 개발된 부트로더이지만 ARM 용으로 확대된 범용 부트로더이다. 간단하게 흐름을 실행 흐름을 설명한다.

- ① startup(cpu/arm920t/start.S) - cpu초기화, dram 초기화, 그 후에 부트로더의 ram으로의 relocation, C코드에서의 main인 start_armboot()를 호출함.
- ② c쪽의start코드(lib_arm/board.c) - dram초기화 이후의 flash, uart등의 주변 디바이스 초기화
- ③ 메인루프(command line shell: common/main.c) - 유닉스 shell과 같은 명령어 처리 루틴. autoboot delay동안 enter키 입력을 기다리는데 그동안 키 입력이 안되면 바로 boot command(설정파일을 보면 나옴)를 실행시킴.
- ④ app의 실행 혹은 리눅스등의 OS로 부팅(lib_arm/armlinux.c) - 헤더가 붙은 커널의 헤더를 읽어서 압축을 풀고 리눅스로 제어권을 넘겨주는 코드

(2) 메모리 맵

flash에서 부팅하기 위해서 u-boot, 커널 이미지, 램디스크가 저장될 위치, 즉 주소를 지정해야한다. 아래 그림은 본 과제에서 사용한 플래시 메모리 맵이다. 부트로더는 0x00000000, 커널이미지는 0x00040000부터 저장되고, 램디스크 이미지는 0x00240000에 저장되면 나머지 공간은 사용자 영역으로 할당된다.

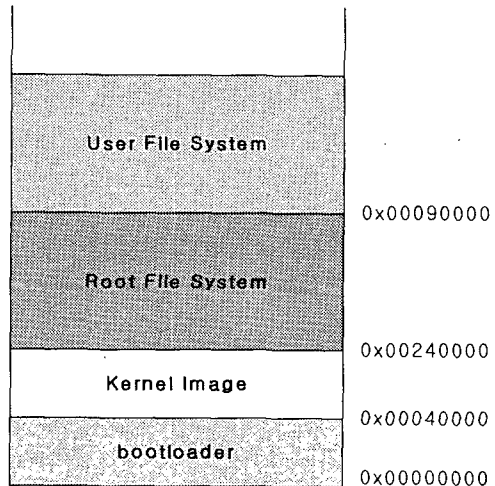


그림 5. 메모리 맵

(3) 루트 파일 시스템 개발

본 논문에서 루트 파일 시스템은 하드디스크가 존재하지 않기 때문에 메모리의 일정 부분을 할

당하여 하드 디스크처럼 사용가능하도록 하고 있는데 이것을 램디스크(ramdisk)라고 한다. 이 램디스크는 램 자체가 휘발성이므로 전원을 차단하면 모든 데이터가 사라지며, 램의 특성상 실행이 빠르며 압축을 이용하므로 용량을 효율적으로 사용할 수 있는 장점이 있다. 본 논문에서는 몬타비스타 하드햇 리눅스의 램디스크 사용하여 램디스크를 구성하였다. 다음은 개발 절차이다.

- ① 16메가바이트 크기의 루프백 디바이스를 생성한다.


```
rm -f bigRamdisk
dd if=/dev/zero of=bigRamdisk bs=1024 count=16384
```
- ② ext2 파일시스템을 생성한다.


```
echo "make bigramdisk file system"
mke2fs ./bigRamdisk
```
- ③ 생성된 파일시스템을 개발용 루트파일시스템에 마운트한다.


```
echo "mounting bigramdisk to newramdisk"
/bin/mount -t ext2 -o loop bigRamdisk /mnt/newRamdisk
```
- ④ 개발용 루트파일시스템의 소스를 새롭게 생성된 파일시스템에 복사한다.


```
echo "copy original ramdisk to new ramdisk"
cp -rf /mnt/ramdisk/* /mnt/newRamdisk/
```
- ⑤ 생성된 파일시스템을 언마운트한다.


```
echo "unmounting new ramdisk"
umount /mnt/newRamdisk
```
- ⑥ 생성된 파일시스템을 gzip으로 압축한다.


```
echo "gzip new ramdisk"
gzip bigRamdisk
```
- ⑦ mkimage 툴로 u-boot 용 램디스크로 만든다.


```
echo "making new uimage new ramdisk"
./mkimage -A arm -O linux -T ramdisk -C gzip \
-n 'Ext2 Ramdisk Filesystem' \
-a 0x12000000 -e 0x12000000 \
-d ./bigRamdisk.gz ./uramdisk
```
- ⑧ /tftpboot 에 복사하여 타겟보드의 플래시메모리에 복사한다.


```
cp ./uramdisk /tftpboot
```

(4) USB 디바이스 드라이버

그림 3에서 보듯이 GPIO 8번이 High가 되면 USB 케이블이 연결되었으므로 GPIO 3을 High로 설정하여 UMS를 enable 시킨다. 이때부터 하드디스크는 연결된 PC에서 제어하게 된다. USB 케이블이 비연결되면 GPIO 8 이 Low 상태가 되는데 이때 GPIO 3을 Low 상태로 만들어 UMS를 disable 시킨다. UMS 가 enable 상태인 동안 PMP는 idle 상태가 된다. 참고로 s3c24a0에서 idle 상태는 cpu로의 클럭만 정지되고 나머지 장치로의 클럭은 정상 입력되는 모드이다.

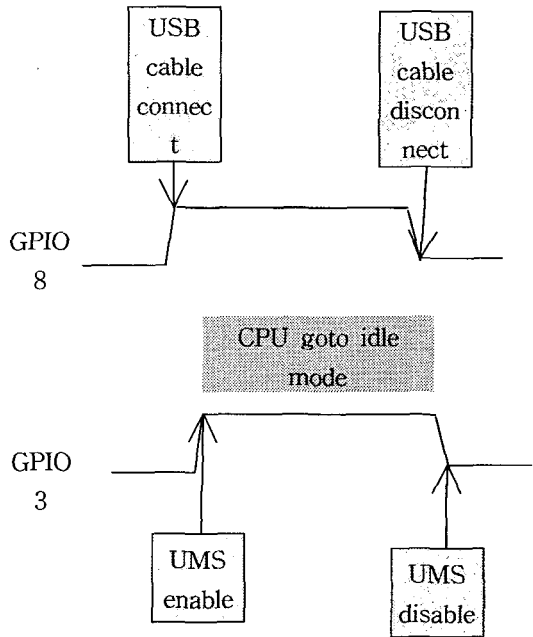


그림 3. UMS 타이밍

(5) 키패드 디바이스 드라이버

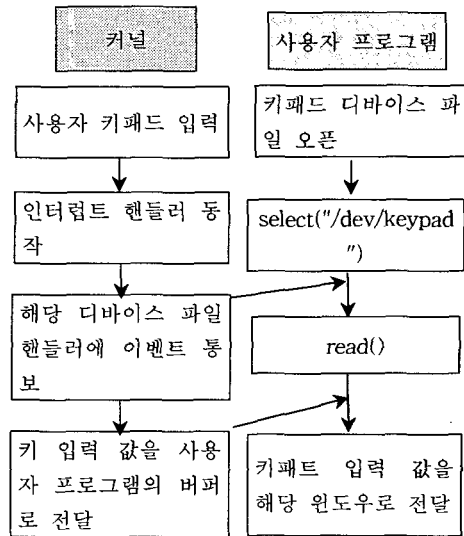


그림 7. 키 패드 동작 원리

그림 4는 키 패드 디바이스 드라이버의 동작 원리를 나타내고 있다. 커널에서는 사용자의 키패드 입력이 발생하면 인터럽트 핸들러가 동작을 하고 키패드 디바이스 파일 핸들러에 이벤트를 통보한다. 그리고 키 입력 값을 사용자 프로그램의 버퍼로 전달하게 된다. 사용자 프로그램에서는 키패드 디바이스 파일을 오픈하면 select() 시스템 콜로

키패드 입력의 발생 여부를 폴링하여 발생되면 read() 시스템 콜에 의해 입력값을 커널로부터 획득하게 된다. 주요 동작 절차는 다음과 같다.

Linux # mknod keypad c 233 0
 ;/dev 밑에 keypad 란 디바이스 파일이 생성됨.

IV. Qt/Embedded 기반의 관광안내 프로그램

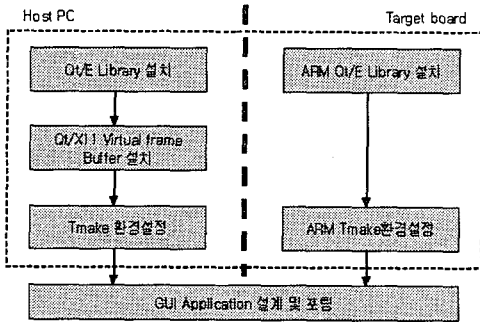


그림 8. Qt/E 개발 환경

그림 5는 Qtopia 용 GUI 프로그램 개발 과정이다. 일반적으로 Linux용 Qt 의 경우 Qt/X11, Qt/E, 그리고 Qtopia가 필요하다. 우선 Host PC 에서 위 세 개의 패키지를 설치하고 가상 프레임 버퍼를 이용하여 Qtopia를 에뮬레이션한다. 이 가상 버퍼상에서 제측용 애플리케이션을 개발하게 된다. 애플리케이션 개발이 완료된 후 타겟 보드용 Qt/E 라이브러리를 컴파일하여 실제 타겟보드에 포팅하게 된다. 본 논문에서 사용한 Qt 버전은 다음과 같다.

- Qt/X11 2.3.2
- Qt/Embedded 2.3.7
- Qtopia 1.7.0
- tmake 1.13

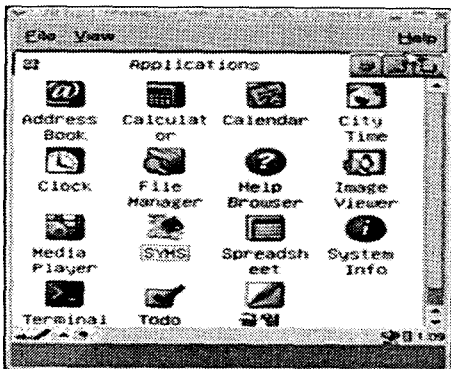


그림 6. 단말기 초기화면(Qtopia)

그림 6은 Qtopia가 실행된 화면이고 그림 7은 관광안내 애플리케이션이다. 구현된 인터페이스에는 관광지의 이미지, 텍스트 설명, 그리고 비디오 안내 등의 기능을 가지고 있다.

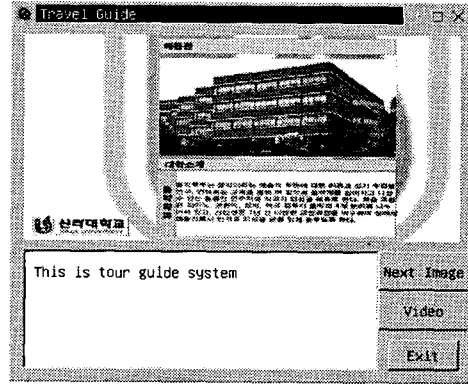


그림 7. 관광안내 애플리케이션

V. 결론

본 논문에서는 임베디드 리눅스를 이용한 관광안내 단말기를 설계 및 구현하였다. 구현된 단말기는 리눅스 운영체제를 탑재하고 있는 PDA와 비슷한 인터페이스를 가지고 있으며 Qtopia가 구동되고 있어 사용자에게 WinCE 와 필적할 만한 편리한 인터페이스를 제공할 수 있다. 향후 연구 과제로는 무선 통신 기능을 추가하여 관광정보를 실시간으로 업데이트하며 RFID와 연계한 관광안내 단말기를 제작하는 것이다.

참고문헌

- [1] Embedded Linux tops developers' 2002 wish list, LinuxDevices.com, 2001. 17.
- [2] Stephen Balaco, Whitepaper: Linux's Future in the Embedded Market, LinuxDevices.com, 2001.6.
- [3] Rick Lehrbaum, "Using Linux in Embedded and Real-time Systems," LinuxDevices.com, 2000.2.
- [4] 박영준, 리눅스 플랫폼의 임베디드 프로그래밍, 프로그램 세계, 2000.7.
- [5] John Lombardo, Embedded Linux, New Riders, 2002.
- [6] <http://linux-embedded.com/software.php3>.
- [7] BlueCat Linux Users Guide - Release 4.0, LinuxWorks, 2001.
- [8] <http://www.peeweelinux.org/Documentation/PeeWeeLinux.html>.
- [9] <http://www.trolltech.com/t>