

임베디드 리눅스 기반의 스마트폰 에디션 구현

김도형[†] · 윤민홍^{**} · 김선자^{***} · 이철훈^{****}

요약

본 논문에서는 스마트폰 미들웨어 및 기본 응용 프로그램으로 구성된 임베디드 리눅스 기반의 스마트폰 에디션의 구현에 대해 기술한다. 기존의 스마트폰 플랫폼들은 상용화 시 로열티를 지불해야 하고, 소스 코드가 공개되어 있지 않아 기능 확장의 문제점이 있었다. 임베디드 리눅스 기반의 GPE는 상용화 시 로열티 문제점은 해결하지만, PDA 급 장치들을 목표로 개발되어 스마트폰과 같은 작은 LCD 크기 및 키 입력만으로 응용 프로그램을 수행할 수 없는 문제점이 있었다. 구현된 Qplus-P스마트폰 에디션에서는 임베디드 리눅스 기반으로 개발되어 로열티가 없으며, 사용자가 키 입력만으로 폴더 이동 및 특정 응용 프로그램을 수행할 수 있는 키 네비게이션 기능을 제공한다. 그리고, 기존의 셀룰러 폰과 유사한 화면 구조 및 인디케이터를 제공함으로써 사용자가 보다 편리하게 스마트폰 에디션을 사용할 수 있는 환경을 제공한다.

키워드 : Qplus-P, 스마트폰 에디션

The Implementation of SmartPhone Edition based on Embedded Linux

Do-Hyung Kim[†] · Min-Hong Yun^{**} · Sun-Ja Kim^{***} · Cheol-Hoon Lee^{****}

ABSTRACT

The paper describes implementation of smartphone edition based on embedded Linux, which is composed of smartphone middleware and basic applications. Existing smartphone platforms entails loyalty expenses and has difficulties on customization due to the closed policy of their source codes. Although GPE - based on embedded Linux has solved the loyalty issues, other problems are remained unsolved. Since GPE has been designed for relatively large PDA-level devices, applications are generally inoperable under the condition of small LCD size and input methods without touch-screen; the condition is typical limitations of the smartphones. In the implemented smartphone edition of the paper no loyalty is required, because the smartphone edition is based on embedded Linux. It also supports keypad-only operating functionality such as moving between folders or running applications with operations. Besides, it provides more user-friendly environment to use the smartphone editionscreen interfaces and indicators to those of the regular cellular phones.

Key Words : Qplus-P, SmartPhone Edition

1. 서론

21세기 정보통신 분야에서 네트워크 기술을 탑재한 정보 가전들의 정보 서비스가 우리의 일상 생활에 직접적으로 활용될 것이라는 예견은 이미 실현되고 있다. 현재 가정에서 흔히 사용되고 있는 텔레비전, 냉장고, 전화기, 게임기 등이 지능을 가진 정보 단말기 역할을 수행하여 가정 내뿐만 아니라 집 밖에서의 생활 양식까지도 바꾸어 놓고 있다. 디지털 텔레비전, 인터넷 셋탑박스, 인터넷 전화기 등 디지털 및 네트워크 기술을 바탕으로 한 정보가전 제품이 속속 등장하면서 이들 제품의 기능을 제어하는데 필수적인 정보가전용 실시간 운영체제 시장이 크게 성장하고 있다[1].

한국전자통신연구원에서는 소형의 휴대 정보 단말에서부터 홈 서버까지 다양한 종류의 정보가전 기기에 공통으로 사용될 수 있는 임베디드 리눅스 기반의 운영체제 Qplus-P를 개발하였다. Qplus-P는 임베디드 리눅스 기반의 커널, 라이브러리, 통합 개발 도구로 구성되고, PDA, 디지털 TV 셋탑박스, 웹 패드 등에 탑재되었다[1-4]. 본 논문에서는 소형 정보가전 기기인 스마트폰에 탑재되는 Qplus-P 기반의 스마트폰 에디션(edition)의 구현에 대해서 기술한다.

스마트폰은 일종의 전화 중심의 장치로서 셀룰러폰과 유사한 디자인을 갖지만, PDA와 같은 풍부한 응용 프로그램들을 제공한다. 즉, 스마트폰은 전화 송수신 기능과 다양한 응용 프로그램들, 예를 들어 개인정보관리시스템(PIMS), 게임, 미디어 재생기 등을 제공하는 디바이스이다. 일반적으로 스마트폰에는 전화 송수신 기능을 제공하기 위한 모뎀 형태의 콜(CALL) 프로세서와 다양하고 복잡한 응용 프로그램을 위한 어플리케이션 프로세서를 탑재하고 있다.

† 정 회 원 : 한국전자통신연구원 임베디드S/W연구단 선임연구원
 ** 정 회 원 : 한국전자통신연구원 임베디드S/W연구단 연구원
 *** 정 회 원 : 한국전자통신연구원 임베디드S/W연구단 책임연구원
 **** 정 회 원 : 충남대학교 컴퓨터공학과 교수
 논문접수 : 2005년 8월 4일, 심사완료 : 2005년 12월 12일



(그림 1) 임베디드 리눅스 기반의 GPE 수행 화면

현재 다양한 형태의 스마트폰이 출시되고 있는데, 스마트폰용 S/W 플랫폼으로 마이크로소프트의 스마트폰 2002와 스마트폰2003이 많이 사용되고 있다[7]. 스마트폰 2002와 2003은 마이크로소프트의 스마트폰용 플랫폼으로써, 미디어 재생기, 문서 편집기, 웹 브라우저 등과 같이 사용자에게 PC와 유사한 수행 환경을 제공하고 있다. 하지만, 스마트폰 2002와 스마트폰2003은 상용화 시 로열티를 지불해야 하고, 소스 코드가 공개되어있지 않아 기능 확장 및 소스 코드의 수정이 제한되는 단점이 존재한다. 이러한 문제점들은 공개 소스인 임베디드 리눅스를 사용함으로써 해결될 수 있다. 즉, 임베디드 리눅스를 스마트폰에 적용함으로써 라이선스로 인한 경제적인 문제점을 해결할 수 있고, 개발자가 소스 코드를 자유롭게 수정할 수 있다.

현재 PDA를 위한 리눅스 배포 판인 GPE[6]에서는 마이크로소프트의 스마트폰 2002/2003과 같은 다양한 응용 프로그램들이 제공되고 있다. (그림 1)은 임베디드 리눅스 기반의 GPE 팜탑(Palmtop) 환경을 보여준다.

스마트폰에 임베디드 리눅스 배포 판인 GPE를 적용하는데 발생할 수 있는 문제점 중의 하나는 디바이스 크기이다. GPE는 240x320과 같이 큰 LCD 사이즈와 터치 스크린을 갖는 PDA 디바이스를 타겟으로 개발되었다. 하지만, 대부분의 셀룰러폰과 스마트폰들은 PDA LCD 크기의 절반 정도의 작은 LCD 크기를 갖고, 터치 스크린을 제공하지 않는다. Qtopia[12] 폰 에디션은 176x208, 176x220, 240x320과 480x640 픽셀 스크린을 지원하지만, 마이크로소프트의 스마트폰 2002/2003과 같이 라이선스를 받아야 하는 문제점이 있다.

본 논문에서는 작은 화면 크기와 키 입력만으로 동작하는 스마트폰에 적용 가능한 Qplus-P기반의 스마트폰 에디션의 구현에 대해서 기술한다. 본 논문은 다음과 같이 구성되어 있다. 2절에서는 구현된 스마트폰 에디션의 구조 및 기능에 대해 기술하고, 3절에서는 스마트폰 에디션의 구현에 대해서 다룬다. 4절에서는 스마트폰 에디션의 특징과 실행 예를 보여주고, 마지막으로 5절에서는 결론 및 향후 과제에 대해서 기술한다.

2. 스마트폰 에디션의 구조

임베디드 리눅스 기반의 스마트폰 에디션은 크게 Qplus-P 리눅스 커널, 스마트폰 미들웨어, 스마트폰 응용 프로그램으로 구성된다. (그림 2)는 스마트폰 에디션의 소프트웨어 구조를 보여준다.

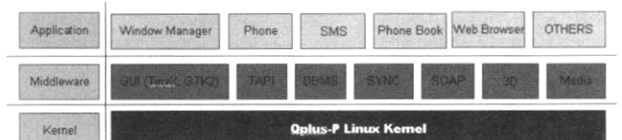
스마트폰 커널은 리눅스 커널 2.4.18에 기반을 둔 Qplus-P를 사용한다. Qplus-P는 PDA와 같은 정보기전을 위한 한국전자통신연구원의 임베디드 리눅스 솔루션이다[2]. 본 논문에서는 스마트폰 미들웨어와 응용 프로그램의 구현에 대해 다루고, Qplus-P 리눅스 커널에 대한 자세한 설명은 생략하도록 한다.

스마트폰 미들웨어는 크게 TinyX/GTK+2 기반의 그래픽 라이브러리, 셀룰러 망을 접근할 수 있도록 지원하는 TAPI(Telephony API), 임베디드 SQL 데이터베이스 엔진인 SQLite[12], SYNC, 3D 그래픽 라이브러리, SOAP 등으로 구성된다.

Tiny-X 그래픽 서버는 한글 처리 및 160x240 화면 크기를 지원할 수 있는 기능을 제공하고, GTK+2 그래픽 킷은 스마트폰 에디션에 적합한 크기를 갖는 위젯들을 제공한다. TAPI는 응용 프로그램이 CDMA 네트워크를 접근할 수 있도록 지원한다. Phone과 SMS(Short Messaging System) 응용 프로그램들이 TAPI를 이용하여 전화 걸기, SMS 송수신, 네트워크 상태 표시 기능을 제공한다. SYNC는 PC와 스마트폰의 데이터 동기화를 위한 라이브러리로써 사용자 하여금 직접 데이터를 백업하기 위한 수고를 덜어준다. 스마트폰 에디션에서는 동기화 라이브러리로 SyncML[14]을 제공한다. SQLite[9]는 임베디드 시스템을 위한 SQL 데이터베이스 엔진으로, 일부 SQL 언어와 C/C++ 인터페이스를 제공한다. 스마트폰 북, Phone, SMS 응용 프로그램들이 SQLite를 이용하여 데이터 저장 및 검색 기능을 제공한다.

3D 그래픽 라이브러리는 응용 프로그램이 벡터 그래픽을 쉽게 사용할 수 있도록 지원하는 것으로 OpenGL/ES 1.0[12]을 구현하였다. 미디어(Media) 라이브러리는 리눅스용 공개소스인 gstreamer-0.6.1[13]을 사용하는데, 사용자가 필요 시 새로운 코덱을 쉽게 추가할 수 있도록 지원한다. SOAP는 웹 서비스를 위해 사용되는 프로토콜로써, 웹 서비스 서버와 클라이언트 사이에 데이터를 전달하는데 사용된다. 스마트폰 에디션에서는 웹 서비스를 지원하기 위해서 gSOAP[8]를 포팅하였다.

스마트폰 에디션에서의 응용 프로그램들은 크게 윈도우 매니저, 주소록을 저장하고 관리하는 스마트폰 북, 전화 콜



(그림 2) 스마트폰 에디션 소프트웨어 구조도

응용인 Phone, 단문 송수신 응용인 SMS, 웹 브라우저, ICQ 클라이언트, 한글 입력기 등이 포함된다.

윈도우 매니저는 공개 소스인 매치박스(matchbox)에 스마트폰 화면 크기에 적합한 계층 구조 및 사용자가 키 입력만으로 응용 프로그램을 수행할 수 있는 기능을 추가하여 구현하였다. 그리고, 스마트폰 윈도우 매니저는 기존의 셀룰러폰과 유사한 인터페이스를 제공하기 위해 배터리 레벨, 안테나 신호세기 레벨, 전화 상태, SMS 상태, 현재 입력 모드 등이 윈도우 매니저에 표시되도록 하였다. 스마트폰 북에서는 주소록 데이터가 SQLite 데이터베이스를 이용하여 저장되는데, 주소록 데이터에는 사진, 사용자 설정 링톤, 개인 메모 등이 포함된다. 이 주소록 데이터는 X/GTK+2 기반의 통신 메커니즘인 Selection과 Property를 사용하여 응용 프로그램들 사이에 전달된다.

SMS응용 프로그램은 CDMA 네트워크를 통해SMS 문자를 전송하거나 수신하기 위한 것으로, 송수신하는 SMS 메시지를 SQLite에 저장하고 나중에 검색할 수 있는 기능을 제공한다. 전화 콜 응용인 Phone은 사용자가 전화를 걸거나 종료할 수 있는 기능을 제공하는데, 통화 기록을 SQLite에 저장하여 검색할 수 있는 기능을 제공한다. SMS와 Phone 응용은 일반적인 셀룰러폰 기능과 동일한 인터페이스를 가지도록 구현하여, 사용자가 쉽게 사용할 수 있도록 하였다. 스마트폰 에디션에서는 공개소스인 dillo[10]를 웹 브라우저로 사용한다. Dillo는 GTK1.2 버전용으로 개발된 것으로 스마트폰 에디션에서는 이를 GTK+2.0으로 확장하고, 한글 처리 기능을 갖도록 구현하였다. ICQ 클라이언트 응용은 ICQ를 사용하는 인스턴트 메신저 응용으로써 사용자는 ICQ를 사용하여 서로 통신할 수 있다. ICQ클라이언트 응용은 사용자와 사용자 그룹을 추가하거나 삭제할 수 있고, ICQ 프로토콜을 이용하여 수신자가 오프라인인 경우에도 메시지를 전송할 수 있는 기능을 제공한다.

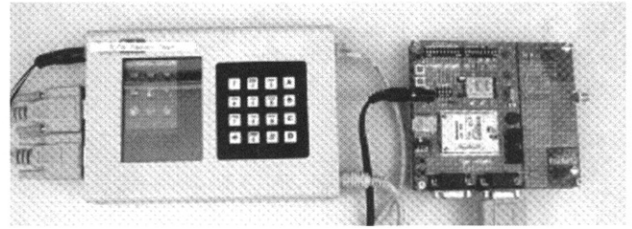
스마트폰 에디션에서는 한글 입력기로 vkey를 제공한다. Vkey는 리눅스용 입력기인 ami를 기반으로 천지인 방식의 한글 입력을 지원하도록 기능을 추가하였다. 즉, 사용자는 일반 셀룰러폰과 동일한 방식으로 키를 선택하여, 한글 및 숫자 등을 입력할 수 있도록 하였다. 마지막으로, 스마트폰 에디션에서는 웹 서비스 응용으로 주식 시세를 검색할 수 있는 주식 검색, 현재 환율을 검색할 수 있는 환율 검색, 스펠 체크를 위한 스펠 체크 등을 제공한다.

3. 스마트폰 에디션 구현

이 절에서는 스마트폰 에디션의 하드웨어 개발 환경에 대해 소개하고, 스마트폰 에디션의 미들웨어 및 응용 프로그램의 구현에 대해 다룬다.

3.1 스마트폰 에디션 개발 하드웨어

스마트폰 에디션은 Tynux Box X(일명, TBX)[4]와 SKT용 CDMA 모뎀을 사용하여 구현되었는데, (그림 3)은 스마트폰 개발 하드웨어 환경을 보여준다.



(그림 3) TynuxBox X와 CDMA 모뎀

<표 1> 스마트폰 에디션에서 정의된 키 및 용도

키 값	용도
0~9, #, *	셀룰러폰에 있는 키와 동일함
SEND/END	전화를 걸거나 끊는 키
Home	홈 스크린으로 복귀하는 키
Back	Back Space
Left/Right Soft Key	메뉴를 활성화시키기 위한 키로, 일반적인 셀룰러폰의 키와 동일함
Up/Down/Left/Right	셀룰러폰의 방향키와 동일
OK	메뉴, 아이콘 등을 선택하기 위한 키

TBX는 인텔 PXA255 프로세서(XScale), 32MB 램, 32MB 롬, 터치스크린이 지원되는240x320 TFT 컬러 LCD를 가지고 있다. TBX에서는 CDMA 모듈과 시리얼 통신을 이용하여 AT 명령어를 전달하게 된다. CDMA 모듈은 벨웨이브(B)의 BCM-850E[7]이고, 이 모듈은 AT 명령어 셋을 이용하여 SKT의 011 망을 사용할 수 있다.

스마트폰 에디션에서는 TBX의 240x320 LCD 영역 중에서, 160x240 영역만 사용할 수 있도록 tinyX 그래픽 서버를 수정하였다. 이는 현재 출시되고 있는 스마트폰들이 240x320보다 작은 LCD 크기를 지원하므로, 스마트폰 에디션에서도 LCD 영역을 160x240으로 정의하여 사용하였다. TBX는 터치 스크린 기능을 제공하지만 스마트폰 에디션은 터치스크린뿐만 아니라, 사용자가 키 값을 입력하여 스마트폰 응용 프로그램들을 동작시킬 수 있도록 하였다.

TBX에서는 총 16개의 키 버튼들을 제공하는데, 스마트폰 에디션에서는 이보다 많은 23개의 키 값을 정의해서 사용한다. 부족한 키는 키 버튼이 눌렸을 때의 윈도우 상태를 체크하여, 키 값의 의미를 다르게 정의하여 사용하였다. <표 1>은 스마트폰 에디션에서 정의된 키 와 그 기능을 간략히 보여준다.

3.2 스마트폰 에디션 미들웨어 및 응용 프로그램 구현

이 절에서는 스마트폰 에디션 중에서 그래픽 라이브러리, TAPI, 윈도우 매니저, 한글 입력기의 구현에 대해서 다룬다.

3.2.1 그래픽 라이브러리

스마트폰 에디션에서는 그래픽 서버로 tinyX, 그래픽 툴킷으로 GTK+2.2를 사용한다. TinyX는 X(버전 4.2.0) 소스를 기반으로 구현되었는데, 일반적으로 tinyX 소스는 X 소스 내 xc/pro-grams/Xserver/hw 에 위치한다. 스마트폰 에디션의 tinyX는 한글 처리 기능, 터치 스크린 입력 기능 등

```

Bool fbdevScreenInitialize (KdScreenInfo *screen,
FbdevScrPriv *scrpriv) {
    FbdevPriv *priv = screen->card->driver;
    ....
    #if 1 // 160x240
        screen->width = 160;
        screen->height = 240;
    #else
        screen->width = priv->var.xres;
        screen->height = priv->var.yres;
    #endif
    ....;
}

```

(그림 4) fbdevScreenInitialize 함수 변경 부분

```

#include <gdk/gdkkeysyms.h>
#include "gtkbindings.h"
...
#ifdef GSE
#define DEFAULT_SCROLLBAR_SPACING 0
#else
#define DEFAULT_SCROLLBAR_SPACING 3
#endif
...

```

(그림 5) gtkscrolledwindow.c에서 변경된 부분

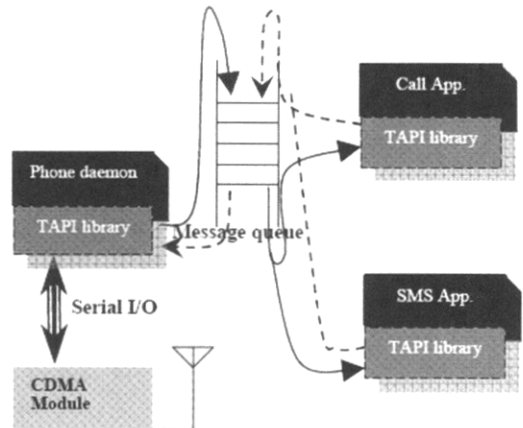
을 제공한다. 그리고, 160x240의 해상도를 지원할 수 있도록 xc/programs/Xserver/hw/kdrive/fbdev/fbdev.c를 (그림 4)와 같이 변경하였다. (그림4)에서 보듯이 tinyX프레임버퍼 디바이스의 스크린 초기화 함수 fbdevScreenInitialize에서 화면의 크기를 160x240으로 설정함으로써, 320x240의 LCD 영역 중에서 160x240 영역만 사용할 수 있도록 하였다.

스마트폰 에디션에서는 160x240의 작은 화면 크기에 적합하도록 기존의 GTK+2 위젯들을 수정하였다. 공개된 GTK+2 위젯들은 작은 화면 영역에 맞도록 설계된 것은 아니기 때문에, 160x240의 화면 영역에서 제대로 표시되지 못하는 문제점이 있었다. 스마트폰 에디션에서는 160x240 화면 영역에서도 GTK+2 위젯이 제대로 표시되도록 하기 위해, 기존의 GTK+2 위젯들 중에서 gtkcombo, gtkdialog, gtklist, gtkrange, gtkscrollbar, gtkscrolledwindow, gtktext-view, gtktreeviewcolumn 위젯들의 크기를 변경하였다. (그림 5)는 gtkscrolledwindow.c 에서 스크롤 윈도우의 바 영역 크기를 변경하기 위한 코드를 간략히 보여준다. (그림 5)에서 보듯이 gtkscrolledwindow에서 DEFAULT_SCROLLBAR_SPACING의 크기를 0으로 설정하여 위젯 크기를 줄이고 있음을 알 수 있다.

3.2.2 TAPI

스마트폰 에디션의 TAPI는 (그림 6)과 같이 하나의 TAPI 라이브러리와 폰 데몬으로 구성된다.

폰 데몬은 CDMA 네트워크 상태를 체크하는 것뿐만 아니라, 전화 걸기, SMS 송수신 기능을 제공한다. (그림 6)에서 보듯이 오직 폰 데몬만이 CDMA 모듈과 시리얼 포트를



(그림 6) 폰 데몬과 TAPI 라이브러리

```

BOOL makeVoiceCall(const char* receiverNumber,
PFNCALLBACK fp, void* arg) {
    ...
    msg_ds.common.mtype =
    MTYPE_TO_DAEMON;
    msg_ds.common.cmd = CMD_MAKE_CALL;
    ...
    if (msgq > -1) {
        if (msgsnd(msgq, (struct msgbuf *)
        &msg_ds, size, MSG_NOERROR) == -1)
            return TAPI_FALSE;
    }
    else return TAPI_FALSE;

    if (fp) { libState = LIBTAPIS_CALL_ASYNC;
    ...
    return TAPI_TRUE; }
    libState = LIBTAPIS_CALL_SYNC;
    ...
    return (cmdResult == CMD_OK ? TAPI_TRUE :
    TAPI_FALSE);
}

```

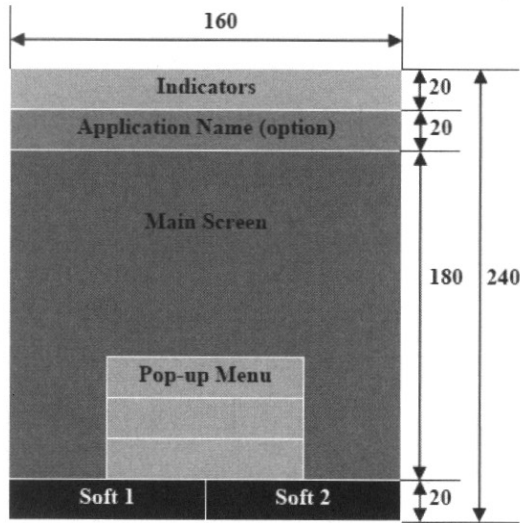
(그림 7) TAPI 라이브러리 중 전화 콜 함수

통해 직접 통신할 수 있고, 폰 응용 프로그램, SMS 응용 프로그램, 신호세기 인디케이터(indicator)는 리눅스 메시지 큐를 통해 폰 데몬과 통신하여 CDMA 네트워크 서비스를 이용하게 된다.

TAPI 라이브러리는 응용 프로그램에 전화 걸기, SMS 메시지 송수신 기능, CDMA 신호 세기와 같은 추가적인 정보를 제공한다. TAPI라이브러리는 폰 데몬과 전화 응용 프로그램 사이에 하나의 메시지 큐를 사용하여 통신하게 되는데, CDMA 모듈을 제어하기 위한 모든 명령어들은 폰 데몬에 의해서 전달된다. (그림 7)은 TAPI중에서 전화 콜(CALL) 함수를 간략히 보여준다.

3.2.3 윈도우 매니저

스마트폰 에디션의 윈도우 매니저는 공개소스인 matchbox-0.7을 기반으로 구현되었으며, 사용자에게 셀룰러폰과 유사한 기능을 제공하기 위해서 사용자가 입력한 키 값을 처리하기



(그림 8) 스마트폰 화면 계층 구조

위한 기능과 스마트폰 화면 계층 구조를 설정하기 위한 기능이 추가되었다. (그림 8)은 스마트폰 에디션의 화면 계층 구조를 보여준다. 이 화면 계층구조는 스마트폰의 홈(Home) 스크린 및 각 응용 프로그램에도 동일하게 적용된다.

(그림 8)에서 보듯이 화면의 계층 구조는 인디케이터 표시 영역, 응용 프로그램 이름 표시 영역, 메인 스크린 표시 영역, 소프트 버튼 표시 영역으로 구성된다. 신호세기 인디케이터, 입력 모드 인디케이터, 전화 인디케이터, SMS 인디케이터들이 인디케이터 영역에 표시된다. 응용 프로그램 이름을 표시하기 위한 160x20 영역은 옵션으로 응용 프로그램의 특성에 따라 지정하여 사용할 수 있다. 즉, 응용 프로그램의 특성에 따라 프로그램 명칭을 표시할 필요가 없을 때는 프로그램 이름 표시 영역을 메인 스크린 표시 영역으로 함께 사용할 수 있다. 소프트 버튼 표시 영역은 두 개의 소프트버튼(각 각은 80x20)으로 구성되고, 사용자가 특정 버튼(소프트버튼 구동 키)을 누르게 되면, 소프트버튼에 등록된 팝업 메뉴가 화면에 표시된다. 마지막으로, 메인 스크린 표시 영역인 160x180 혹은 160x200은 각 응용 프로그램이 사용할 수 있는 영역이 된다. (그림 8)과 같은 화면 계층 구조 및 기능을 지원하기 위해서 matchbox 윈도우 매니저를 수정하였는데, (그림 9)은 matchbox 소스 중에서 mbdesktop_vew.c에서 변경된 부분을 보여준다.

(그림 9)에서 보듯이 GSE로 정의된 부분이 새롭게 추가된 코드들이고, GSE_BOTTOM_SPACING 값을 이용하여 화면 계층을 분할하고 있다. 스마트폰 에디션에서는 GSE_BOTTOM_SPACING 값이 20으로 설정되어 있다.

스마트폰 에디션에는 신호세기 인디케이터, 전화 인디케이터, SMS 인디케이터, 입력 모드 인디케이터, 전원 인디케이터들이 있다. 각 각의 인디케이터는 별도의 프로세스로 동작하며, 스마트폰 에디션 내의 다른 응용 프로그램과는 (그림 10)과 같이 Property를 설정함으로써 인디케이터 값을 전달하게 된다. 예를 들어, 신호세기 레벨을 측정하는 폰(Phone) 데몬이 주기적으로 CDMA 신호세기 값을 측정한다

```
void mbdesktop_view_paint(MBDesktop *mb, Bool use_cache) {
...
img_dest = mb_pixbuf_img_clone(mb->pixbuf, mb->bg_img);
#ifdef GSE
    mb->workarea_height -= GSE_BOTTOM_SPACING;
#endif
...
#ifdef GSE {
    char buf[128];
    XFillRectangle(mb->dpy, mb->backing_cache, mb->invert_gc, mb->workarea_x, mb->workarea_y + mb->workarea_height, mb->workarea_width, GSE_BOTTOM_SPACING);
    ...
}
    mb->workarea_height += GSE_BOTTOM_SPACING;
#endif // GSE
...
}
```

(그림 9) mbdesktop_vew.c에서 화면 계층 구조를 위해 변경된 부분

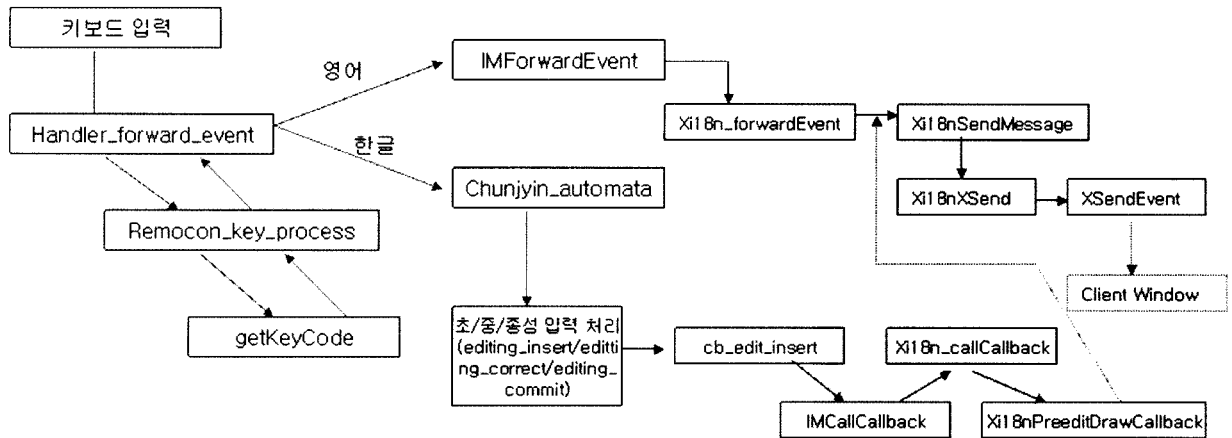
```
Void my_event_callback(MBTrayApp *app, XEvent *event) { //안테나 레벨 표시 콜백
...
    Atom antenna_atom = XInternAtom(dpy,"GSE_ANTENNA_LEVEL", False);

    if ( event->type == PropertyNotify && event->xany.window == xroot && event->xproperty.atom == antenna_atom ) {
        if ( event->xproperty.state == PropertyNewValue ) {
            ...
            if ( XGetWindowProperty(dpy, xroot, antenna_atom, 0, 4, False, XA_INTEGER, &ret_type, &ret_format, &nbytes, &rbytes, &data) != Success )
                fprintf(stderr, "Error in XGetWindowPropertyWn");
            if ( ret_format == 0 ) fprintf(stderr, "No such propertyWn");
            else { /* Actual return values */
                int tmpval = (*data);
                if ( tmpval >= 0 && tmpval <= 5 ) {
                    g_level = tmpval; mb_tray_app_repaint(app);
                }
            }
            ...
        }
    }
}
```

(그림 10) 신호세기 인디케이터 동작 함수

후 신호세기 인디케이터가 등록된 Property 값을 변경하고, 신호세기 인디케이터는 변경된 신호세기 값을 인디케이터 영역에 표시하게 된다.

스마트폰 에디션에서 각 응용 프로그램은 각 소프트 버튼에 팝업 메뉴를 등록하고, 사용자의 키 값을 받아 팝업 메뉴를 효과적으로 표시하기 위해서 새롭게 gsewidget를 구현하였다. Gsewidget은 크게 사용자의 입력 키 값이 소프트 키 값인지를 체크하는 gse_window_keyboard_event_handler, 소



(그림 12) 한글입력기 vkey의 흐름도

```
static gint
gse_window_keyboard_event_handler(GtkWidget
*widget,GdkEventKey *event,gpointer data) {
    GtkWidget *soft;
    ...
    toplevel = gtk_widget_get_toplevel(widget);
    if ( event->keyval == GDK_F5 ) { /* soft button1 */
        soft = g_object_get_data(G_OBJECT(toplevel),
"soft1");
        if ( soft ) { g_signal_stop_emission_by_name(
GTK_OBJECT(widget), "key_press_event");

g_signal_emit_by_name(GTK_OBJECT(soft), "clicked");

                return TRUE; }
    }
    else if ( event->keyval == GDK_F6 ) { /* soft button2
*/
        soft = g_object_get_data(G_OBJECT(toplevel),
"soft2");
        ...
    }
    return FALSE;
}

```

(그림 11) Gsewidget의 gse_window_keyboard_event_handler 함수

프트 키 값인 경우에 해당 콜백(Callback) 함수를 등록하고 처리하는 do_gse_window_set_softkey_callback, 소프트 키가 선택되었을 때 팝업 메뉴를 생성하는 gse_menu_popup, 소프트 버튼에 각 응용 프로그램의 메뉴를 등록하기 위한 gse_window_set_softkey_menu 등으로 구성된다. (그림11)는 gse_window_keyboard_event_handler 의 일부를 보여준다.

3.2.4 한글입력기

스마트폰 에디션에서는 한글 입력기 vkey를 제공한다. 한글 입력기 vkey는 리눅스용 한글입력기인 ami-1.0.11에 친지인 방식의 한글 입력모드를 추가함으로써 구현되었다. Vkey는 한글/영대/영소/숫자/특수기호 등과 같이 총 5가지의 입력 모드를 가지며, 사용자가 키 값 '*'를 누름으로써 모드

값을 변경하게 된다. 입력모드 인디케이터는 한글 입력 모드인 경우에는 "가", 영어 대문자 입력 모드인 경우는 "ABC", 영어 소문자 입력 모드는 "abc", 숫자 입력 모드는 "123"을 표시한다. 특수문자 입력 모드에서는 특수 문자를 선택할 수 있는 팝업 테이블이 화면에 표시되고, 사용자가 원하는 문자를 선택하여 입력하도록 하였다. (그림 12)는 vkey의 흐름도를 보여주는데, 기존 ami 흐름도 중에서 스마트폰 에디션에 불필요한 부분을 삭제하고 친지인 한글 입력 부분을 추가한 것이다.

리눅스용 한글 입력기 ami는 gtk-1.2용으로 개발되었다. 하지만, 스마트폰 에디션에서는 그래픽 킷으로 GTK+2를 사용하므로, 한글입력기 vkey를 GTK+2로 이식하는 작업도 수행하였다. Vkey 소스 중에서 amikey.c, dispwin.c, gtkbd.c, ic.c, util.c 등에 GTK+2를 위한 코드들이 추가되었고, (그림 13)은 ic.c의 ic_destroy 함수에서 추가된 코드를 간략히 보여준다.

```
void ic_destroy(IC *ic) {
    ...
    if (ic->top_win && ic->client_win != ic->top_win &&
!unique_han && ic->client_win !=
GDK_ROOT_PARENT()) {
        #if GTK_MAJOR_VERSION >= 2
            gdk_window_unref(ic->top_win);
        #else
            GdkWindowPrivate *private =
(GdkWindowPrivate *)ic->top_win;
            ... // 기존 코드
        #endif
    }
    ....
}

```

(그림 13) GTK+2로 포팅하기 위해서 ic_destroy 함수에서 변경된 코드

4. 스마트폰 에디션의 특징과 수행 예

이 절에서는 스마트폰 에디션의 특징과 수행 예를 보여준다.

4.1 스마트폰 에디션의 특징

임베디드 리눅스 기반으로 구현된 스마트폰 에디션은 다음과 같은 특징들을 가지고 있다.

- 공개 소스인 임베디드 리눅스 기반으로 개발되어 로열티가 없다.
- 사용자가 키 입력만으로 폴더 이동 및 응용 프로그램을 수행할 수 있는 키 네비게이션(navigation) 기능을 제공한다
- 스마트폰과 같이 작은 화면 크기에 적합한 윈도우 매니저와 스마트폰용 GTK+2 위젯들이 제공된다.
- 스마트폰 응용 프로그램들이 소프트 키 버튼에 팝업 메뉴를 등록하고, 사용자의 키 입력으로 원하는 기능을 수행할 수 있도록 스마트폰용 gsewidget이 제공된다
- 전화 걸기, SMS 송수신을 지원할 수 스마트폰 미들웨어 및 기본 응용 프로그램들이 제공된다.

<표 2>는 스마트폰 에디션과 Qtopia 스마트폰 에디션, 마이크로소프트 스마트폰 2002와의 기능을 비교한 것이다. <표 2>에서 보듯이 구현된 스마트폰 에디션은 기존 스마트폰 플랫폼과 비교해서 라이선스 비용, TAPI, 윈도우 시스템 등에서 장점을 가지고 있음을 알 수 있다.

<표 2> 스마트폰 에디션의 장점

	ETRI SmartPhone	Qtopia Smartphone Edition	Microsoft SmartPhone 2002
OS	Linux(Qplus-P)	Linux	WinCE
화면크기	160x240	174x220	174x220
TAPI	지원	N/A	지원
웹 서비스 스택	지원	N/A	지원
라이선스 비용	없음	있음	있음
한글입력기	있음(천지인)	N/A	있음
윈도우 시스템	X11, GTK+2	QT	Win32
조작 방식	키패드/터치스크린	키패드/터치스크린	키패드

4.2 스마트폰 에디션 수행 예

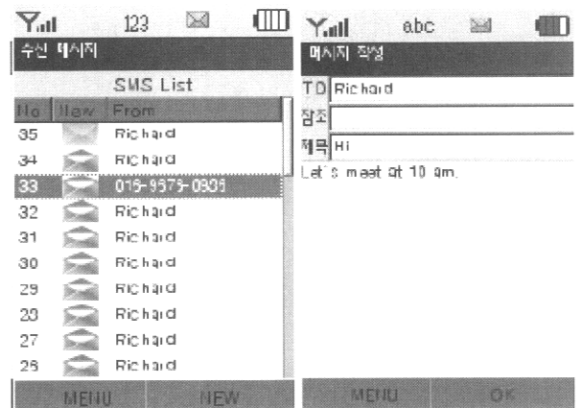
(그림 14)은 스마트폰 에디션의 홈 스크린과 윈도우 매니저를 보여준다.



(그림 14) 홈 스크린 화면



(그림 15) 전화 응용 화면



(그림 16) SMS 응용 화면

(그림 14)에서 보듯이 홈 스크린의 영역은 인디케이터 표시 영역, 메인 스크린 표시 영역, 소프트 버튼 표시 영역으로 구분된다. 인디케이터 표시 영역에는 신호세기 인디케이터, 입력 모드 인디케이터, SMS 인디케이터, 전원 인디케이터가 표시되고 있음을 알 수 있다. 그리고, 스마트폰 기본 응용 프로그램으로 전화번호부, 전화 다이얼러(Dialer), SMS, 인터넷 브라우저, 일정관리 프로그램인 Today 등이 제공됨을 알 수 있다, 사용자는 방향키(Up/Down)를 이용하여 홈 스크린의 메인 영역을 상하로 스크롤 하여 응용 프로그램을 수행할 수 있다.

(그림 15)는 전화 응용을 수행했을 때의 화면과 통화 내역 검색 화면을 보여준다. 사용자는 전화번호를 직접 입력하여 상대방과 통화할 수 있고, 저장된 목록에서 상대방을 선택하여 전화를 걸 수 있다. 통화 내역은 SQLite 데이터베이스에서 목록을 읽어서 화면에 표시하게 된다. (그림 15)에는 숫자 입력을 나타내는 입력모드 인디케이터와 전화 인디케이터가 표시됨을 알 수 있다.

(그림 16)는 SMS 응용을 수행시켜 SMS 문자를 보내는 응용을 보여준다. 사용자는 주소록에 있는 특정 사용자를 지정하여 문자를 보낼 수 있고, 송수신된 SMS 목록을 보고 이를 확인할 수 있는 기능도 제공한다. (그림 16)에서는 SMS 인디케이터도 표시되고 있음을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 임베디드 리눅스 기반의 스마트폰 에디션의 구현에 대해 기술하였다. 스마트폰 에디션은 임베디드 리눅스 기반의 Qplus-P 커널, TAPI 등으로 구성된 스마트폰 미들웨어, 전화, SMS와 같은 스마트폰 기본 응용으로 구성된다.

스마트폰 에디션은 공개소스 기반으로 개발되어 로열티가 없으며, 사용자가 키 입력만으로 풀더 이동 및 특정 응용 프로그램을 수행할 수 있는 키 네비게이션 기능을 제공한다. 그리고, 기존의 셀룰러폰과 유사한 화면 구조 및 인디케이터를 제공함으로써 사용자가 보다 편리하게 스마트폰 에디션을 사용할 수 있도록 하였다.

앞으로 스마트폰 에디션은 작은 화면 사이즈에 적합한 스마트폰용 GTK+2 위젯을 추가로 제공하여야 하고, 스마트폰 환경에 적합한 스마트폰 미들웨어 및 스마트폰 기본 응용 프로그램을 추가로 개발하여야 한다.

참고 문헌

- [1] 김도형, 강우철, 정영준, 김성우, "소형 정보가전 기기용 Qplus-P 윈도우 매니저 개발", 정보처리학회논문지, 제11-D권 제1호, pp.195-202, 2004.
- [2] 강우철, 윤희철, "Qplus 타겟 빌더: 임베디드 리눅스 툴킷", 한국정보처리학회 춘계학술발표논문집, 제9권 1호, pp.663-666, 2002.
- [3] Jerry Epplin, "A developer's review of Qplus, an open source embedded Linux toolkit, <http://www.linuxdevices.com/AT5640843706.html>, Dec., 16, 2002.
- [4] Official Qplus Web site, <http://www.qplus.or.kr>
- [5] PalmPalm Technology Inc, <http://www.palmpalm.co.kr>
- [6] <http://www.handhelds.org/geeklog/index.php>.
- [7] Bruce Brown, "Microsoft's Smartphone Exceeds Expectations," http://www.pcmag.com/print_article/0,3048,a=113391,00.asp, Dec., 30, 2003.
- [8] <http://www.bellwave.com/>
- [9] R.van Engelen and K. Gallivan, "The gSOAP toolkit for web services and peer-to-peer computing networks," 2nd IEEE Int'l Symposium on Cluster Computing and the Grid, 2002
- [10] <http://www.sqlite.org>
- [11] <http://www.dillo.org>
- [12] Trolltech, "Qtopia Phone Whitepaper," <http://www.trolltech.com>
- [13] <http://www.khronos.org/opengles/index.html>
- [14] <http://freedesktop.org/~gstreaemer>
- [15] <http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html>



김도형

e-mail : kimdh@etri.re.kr
 1993년 경북대학교 컴퓨터공학과(공학사)
 1995년 포항공과대학 전자계산학과
 (이학석사)
 1995년~현재 한국전자통신연구원
 임베디드S/W연구단 선임연구원
 관심분야: 결합포용, 실시간시스템, 성능감시, 성능평가



윤민홍

e-mail : mhyun@etri.re.kr
 1998년~2001년 8월 고려대학교
 컴퓨터공학과(학사)
 2001년 9월~2003년 8월 고려대학교
 컴퓨터공학과(석사)
 2003년 7월~현재 한국전자통신연구원
 임베디드S/W연구단 연구원
 관심분야: 임베디드 리눅스, 리눅스 스마트폰, 임베디드 멀티미디어 시스템



김선자

e-mail : sunjakim@etri.re.kr
 1985년 숙명여자대학교 수학과(이학사)
 1995년 충남대학교 컴퓨터공학과
 (공학석사)
 1987년~현재 한국전자통신연구원
 임베디드S/W연구단 책임연구원
 관심분야: 운영체제, DSM, 임베디드 시스템



이철훈

e-mail : chlee@comeng.cnu.ac.kr
 1979년~1983년 서울대학교 전자공학과
 (전자공학학사)
 1986년~1988년 한국과학기술원
 전기전자공학과 컴퓨터공학(석사)
 1988년~1992년 한국과학기술원
 전기전자공학과 컴퓨터공학(박사)
 1983년~1986년 삼성전자 개발실 연구원
 1992년~1994년 삼성전자 개발실 선임연구원
 1994년~1995년 Univ. of Michigan 객원연구원
 1995년~현재 충남대학교 컴퓨터공학과 교수
 2004년~2005년 Univ. of Michigan 교환교수
 관심분야: 운영체제, 병렬처리, 결합포용, 실시간 시스템 등