

실시간 운영체제 Qplus-P용 그래픽 윈도우 라이브러리 구현

김도형[†]·김선자[†]·김성우^{††}

요약

디지털 TV, 인터넷 셋탑박스, 인터넷 전화기 등과 같은 정보가전 제품이 속속 등장하면서 이들 제품의 기능을 제어하는데 필수적인 실시간 운영체제 시장이 크게 성장하고 있다. 한국전자통신연구원에서는 소형의 휴대 정보 단말에서부터 디지털 셋탑박스 및 홈 서버까지 다양한 종류의 정보가전 기기에 공통으로 사용될 수 있는 확장 가능한 표준 실시간 운영체제 Qplus-P를 개발하였다. 본 논문에서는 정보가전용 실시간 운영체제 Qplus-P에 탑재되는 그래픽 윈도우 라이브러리의 구현에 대해 기술한다. Qplus-P 그래픽 윈도우 라이브러리는 공개소스인 tiny-X 그래픽 서버와 GTK 그래픽 툴킷을 기반으로 구현되었다. 구현된 그래픽 윈도우 라이브러리는 다양한 정보가전 기기에 탑재하기 위해 tiny-X 그래픽 서버에 한글 처리 기능, 화면 회전 기능, 터치스크린 입력 기능, 그래픽 가속 기능 등을 추가하였다. Qplus-P 그래픽 윈도우 라이브러리는 현재 arm 프로세서를 사용하는 아이팩 3600 PDA, 삼성 S3C2400 보드, 자우루스 PDA와 x86 프로세서를 사용하는 홈서버에 탑재되었으며, Qplus-P 타겟 빌더에 통합된 형태로 제공된다.

The Implementation of Graphic Window Library for RTOS Qplus-P

Do-Hyung Kim[†] · Sun-Ja Kim[†] · Seung-Woo Kim^{††}

ABSTRACT

As the Internet appliances like digital TV, Internet set-top boxes, and Internet phone, are showing up in the market, the economics of real-time operating system (RTOS), which is an essential for controlling those devices, is expanding faster than ever before. ETRI has developed an extensible RTOS called Qplus-P, targeting various platforms ranging from PDA to Internet set-top box and home server. This paper describes the implementation of graphic window library for Qplus-P. The Qplus-P graphic window library was implemented using tiny-X graphic server and gtk graphic toolkit, which are open source softwares. To port this library to various appliances, hangul processing, screen rotation, touch screen, and graphic acceleration functions are added to the tiny-X graphic server of the implemented graphic window library. Currently, Qplus-P graphic window library is running on ARM-based appliances such as iPaq PDA, Samsung S3C2400 board, and Zaurus PDA, and on Home Server that uses x86 processor. Qplus-P graphic window library is provided as a part of Qplus-P target builder.

1. 서론

21세기 정보통신 분야에서 네트워크 기술을 탑재한 정보가전들의 정보 서비스가 우리의 일상 생활에 직접적으로 활용될 것이라는 예견은 이미 실현되고 있다. 현재 가정에서 흔히 사용되고 있는 TV, 냉장고, 전화기, 게임기 등이 지능을 가진 정보 단말기 역할을 수행하여 가정 내 뿐만 아니라 집 밖에서의 생활 양식까지도 바꾸어 놓고 있다. 예를 들어, 인터넷을 통하여 원격의 시스템과 멀티미디어 데이터를 교환하고 각종 제어 정보를 처리할 수 있게 되었으며, 가정 내에 있는 각종 정보 가전기기들 간에도 유무선 통신을 통한 대용량 멀티미디어 전송이 가능해 졌다. 디지털 TV, 인터넷 셋탑박스, 인터넷 전화기 등 디지털 및 네

트워크 기술을 바탕으로 한 정보가전 제품이 속속 등장하면서 이들 제품의 기능을 제어하는데 필수적인 정보가전용 실시간 운영체제 시장이 크게 성장하고 있다. 컴퓨터 운영체제 업체들과 기존의 실시간 운영체제 전문업체들은 그동안 통신, 산업전자, 항공 및 우주산업 분야의 제어 시스템을 중심으로 형성되어온 실시간 운영체제 시장이 멀티미디어의 필요성에 따라 정보가전용으로 확대됨에 따라 이를 선점하고자 노력하고 있다[15-19].

한국전자통신연구원에서는 소형의 휴대 정보 단말에서부터 홈 서버까지 다양한 종류의 정보가전 기기에 공통으로 사용될 수 있는 확장 가능한 표준 실시간 운영체제 Qplus-P를 개발하였다. 실시간 운영체제 Qplus-P는 라이브러리[1], 커널[2], 통합 개발 도구[3]로 구성되고, 2002년 12월까지 홈서버, 아이팩(IPAQ) 3600 PDA(Personal Digital Assistance), 삼성 S3C2400 보드, 자우루스(Jaurus) SL-5500 PDA 등

[†] 정 회 원 : 한국전자통신연구원 컴퓨터소프트웨어기술연구소 선임연구원

^{††} 정 회 원 : 동의대학교 컴퓨터영상공학과 교수

논문접수 : 2003년 7월 3일, 심사완료 : 2003년 10월 7일

에 탑재되었다. Qplus-P 라이브러리는 크게 C/C++ 표준 라이브러리, 그래픽 윈도우 라이브러리, 파일 시스템으로 구성된다. 본 논문에서는 Qplus-P에 탑재되는 그래픽 윈도우 라이브러리의 구현에 대해 다루고, 커널, 통합 개발도구, 다른 라이브러리에 대한 설명은 생략하도록 한다.

현재 임베디드 리눅스용 그래픽 시스템은 크게 그래픽 서버와 툴킷(toolkit)이 통합된 시스템과 분리된 시스템으로 나눌 수 있다. 그래픽 서버와 툴킷이 통합된 시스템으로는 GTK+/FB(Frame Buffer)[6]와 Qt/E(Qt/Embedded)[7]가 대표적이다. 이러한 그래픽 시스템들은 경량의 그래픽 라이브러리(수백 KBytes~수 MBytes)를 구성할 수 있는 장점이 있지만, 단일 프로세스 기반으로 동작하여 동적으로 응용 프로그램들의 설치 및 제거가 불가능하고 응용 프로그램의 오류가 전체 그래픽 시스템을 정지시키는 문제점이 있다. 그래픽 서버와 툴킷이 분리된 경우에는 동적으로 응용 프로그램들의 설치 및 삭제가 자유롭지만, 다소 큰 그래픽 라이브러리(수 MBytes~수십 MBytes)가 생성된다. 그래픽 서버와 툴킷이 분리된 그래픽 시스템을 구성하는 대표적인 그래픽 서버로는 X 윈도우[8], 마이크로윈도우즈(Microwindows)[21] 등이 있고, 그래픽 툴킷에는 gtk+[1], qt[7], fltk(일명 fulltick)[14] 등이 있다.

X 윈도우는 개발된지 20년이 넘는 클라이언트/서버 구조의 그래픽 서버로써, 안정성이 높고 gtk+와 qt 등 다양한 툴킷들이 개발되어져 왔다. 하지만, X 윈도우 시스템은 데스크 탑용으로 개발되어 다소 크기가 크고 속도가 느리기 때문에, 제한된 하드웨어를 가진 정보가전 기기에는 적합하지 않다. 최근에 X 서버의 장점을 지니면서도, 서버 크기를 1 MByte 이하로 줄인 tiny-X가 임베디드 시스템용으로 많이 연구되고 있다. Tiny-X는 일반적인 X와 달리 컴파일 시 서버 구성 요소를 선택하여 그래픽 서버를 구성할 수 있으므로 서버 크기를 상당히 줄일 수 있는 장점이 있다. 현재 tiny-X는 별도의 소스로 공개되지 않고, xfree86 소스 트리에 포함되어 있다.

X 윈도우를 기반으로 하는 그래픽 툴킷으로는 qt+와 gtk+가 있다. Qt+는 TrollTech사에서 개발한 그래픽 툴킷으로 C++ 기반의 OOP(Object Oriented Programming) 구조를 하고 다양한 위젯(widget)들과 qt 디자이너(designer) 등 편리한 통합 개발 환경을 제공하지만, 상용화시 로얄티를 지불해야 하는 문제점이 있다. 반면, gtk+는 C 기반의 OOP 구조를 하고 glade 등 다소 미흡한 개발 환경을 제공하지만, 다양한 위젯들을 지원하고 로얄티를 지불할 필요가 없는 공개 소스이다.

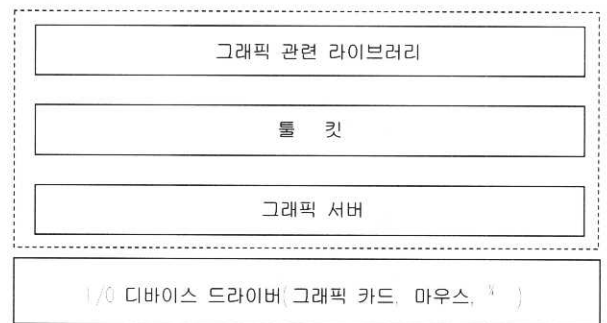
Qplus-P 그래픽 윈도우 라이브러리의 기본 요구사항이 로얄티가 없는 공개 소스를 기반으로 다양한 위젯들을 제공하고, 동적으로 응용 프로그램을 쉽게 설치할 수 있게 하

는 것이다. 따라서, Qplus-P 그래픽 윈도우 라이브러리는 공개 소스인 tiny-X 그래픽 서버와 gtk+ 그래픽 툴킷을 기반으로, Qplus-P 그래픽 응용에 필요한 기능을 추가하여 구현되었다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 Qplus-P 그래픽 윈도우 라이브러리의 구조에 대해 기술하고, 3절에서는 그래픽 윈도우 라이브러리의 구현에 대해 다룬다. 4절에서는 Qplus-P 그래픽 윈도우 라이브러리의 특징과 실험예를 보여주고, 마지막으로 5절에서는 결론 및 향후 과제에 대해서 기술한다.

2. 그래픽 윈도우 라이브러리 구조

Qplus-P 그래픽 윈도우 라이브러리는 (그림 1)과 같이 그래픽 서버와 툴킷, 그리고 그래픽 관련 라이브러리들로 구성된다.



(그림 1) 그래픽 윈도우 라이브러리 구조

그래픽 서버는 공개 소스인 tiny-X를 기반으로 구현되었다. Tiny-X 그래픽 서버는 기본적으로 클라이언트/서버 구조를 가지고, 클라이언트가 요구한 그래픽 기능을 실제로 제공하게 된다. (그림 1)에서 보듯이 그래픽 서버는 클라이언트의 요청을 눈에 보이는 결과물로 전환하기 위해 디바이스에 종속적인 소프트웨어와 통신하는데, 모든 장치 종속적인 소프트웨어는 그래픽 서버에서 관리된다.

툴킷은 상위 레벨의 응용 프로그램 인터페이스를 제공하는 GUI(Graphic User Interface) 요소들을 포함하고, GUI에 사용할 수 있는 다양한 위젯들을 제공한다. 즉, 프로그램 개발자가 직접 제어판, 메뉴 등과 같은 복잡한 객체들을 관리하는 대신 그래픽 시스템에서 쉽게 관리할 수 있도록 한다. Qplus-P 그래픽 윈도우 라이브러리의 툴킷은 공개 소스인 gtk+(버전 1.2.10)를 기반으로 구현되었다.

그래픽 관련 라이브러리는 공개 소스를 기반으로 구현되었으며, gtk 기반의 그래픽 응용을 개발하는데 필수적인 glib, gdk-pixbuf[9], libjpeg[10], zlib[11], libtiff[12], libpng[13] 등으로 구성된다. Glib은 gtk+의 기본이 되는 저수준(low-level)

의 라이브러리로써, C 언어를 처리하기 위한 데이터 구조 및 래퍼(wrapper) 등을 제공한다. Gdk-pixbuf는 이미지를 로드하고 렌더링하기 위한 기능을 제공하고, libjpeg은 JPEG으로 압축된 이미지 파일을 읽고 쓰기 위한 함수들을 제공한다. Libpng는 PNG(Portable Network Graphics) 포맷으로 된 이미지를 인코딩 및 디코딩하는 기능을 제공하고, libtiff는 태그 이미지 파일 포맷(Tag Image File Format)으로 인코딩된 파일을 읽고 쓸 수 있는 함수를 제공한다. 마지막으로, zlib은 데이터를 압축하기 위한 함수들을 제공한다.

3. 그래픽 윈도우 라이브러리 구현

Qplus-P 그래픽 윈도우 라이브러리는 공개 소스를 기반으로 타겟(target) 정보가전기용 그래픽 윈도우 라이브러리와 호스트용 크로스 컴파일 패키지를 제공한다. 타겟용 그래픽 윈도우 라이브러리는 arm 프로세서용 그래픽 윈도우 라이브러리와 x86 프로세서용 그래픽 윈도우 라이브러리로 구성되는데, Qplus-P 타겟빌더[2]에서 사용자가 타겟 프로세서 타입을 선택하게 되면 자동적으로 해당 그래픽 윈도우 라이브러리가 설치되게 된다.

호스트에 탑재되는 크로스 컴파일용 패키지는 호스트에서 타겟용 그래픽 응용을 개발할 수 있도록 지원한다. 현재 Qplus-P 그래픽 윈도우 라이브러리를 위한 크로스 컴파일 패키지는 arm 프로세서와 x86 프로세서를 위한 패키지를 별도로 제공하는데, 사용자는 타겟 정보가전 기기에 적합한 패키지를 선택하여 rpm을 통해 리눅스 호스트에 설치하면 된다. <표 1>는 리눅스 호스트용 크로스 컴파일 패키지 일부를 보여준다.

<표 1> Qplus-P 크로스 컴파일 패키지

ARM 프로세서용	X86 프로세서용
qp-tinyx-armdevel-4.2.0-0.i386.rpm	qp-tinyx-i386devel-4.2.0-0.i386.rpm
qp-glib-armdevel-1.2.10-0.i386.rpm	qp-glib-i386devel-1.2.10-0.i386.rpm
qp-libjpeg-armdevel-6b-0.i386.rpm	qp-libjpeg-i386devel-6b-0.i386.rpm
qp-libpng-armdevel-1.0.7-0.i386.rpm	qp-libpng-i386devel-1.0.7-0.i386.rpm
qp-gtk+-armdevel-1.2.10-0.i386.rpm	qp-gtk+-i386devel-1.2.10-0.i386.rpm

Qplus-P 크로스 컴파일 패키지를 리눅스 호스트에 설치하게 되면, /opt/q+esto/cdt/arm-linux 혹은 /opt/q+esto/cdt/i386-linux 디렉터리에 관련 라이브러리들이 생성되게 된다. 그래픽 윈도우 라이브러리를 위한 크로스 컴파일 패키지는 통합 크로스 컴파일 패키지(크로스 컴파일러, C/C++ 라이브러리 등을 포함)의 일부로써 제공되는데, 본 논문에서는 통합 크로스 컴파일 패키지에 대한 자세한 설명은 생략하도록 한다.

3.1 Tiny-X 그래픽 서버

Tiny-X 그래픽 서버는 X(버전 4.2.0) 소스를 기반으로 구현되었는데, 일반적으로 tiny-X 소스는 X 소스 내 xc/programs/Xserver/hw에 위치한다. Qplus-P tiny-X는 다양한 정보가전 기기에 이식하기 위해 한글 처리 기능, PDA와 같은 소형 정보가전 기기에 적합한 터치 스크린 입력 기능, 전자책 리더에서 요구되는 화면 회전 기능, 그래픽 가속 기능 등 기존의 tiny-X 소스에 Qplus-P 그래픽 응용을 위한 기능들이 추가되었다. Qplus-P 그래픽 윈도우 라이브러리는 크게 x86 계열과 arm 계열의 정보가전 기기에 탑재되었기 때문에, 추가되는 기능들을 패치 파일로 저장하여 Qplus-P 타겟빌더[2]에서 선택된 프로세서 타입에 따라 적절한 패치 파일이 적용되도록 하였다. 정보가전 기기마다 요구되는 기능들이 달라질 수 있으므로, 이 절에서는 프로세서 타입과 상관없이 tiny-X의 주요 기능에 대해 기술한다.

3.1.1 크로스 컴파일 및 서버 구성

(그림 2)는 X 소스에 arm용 크로스 컴파일 환경을 지정하기 위해 수정된 cross.def 파일 내용을 일부 보여준다.

```
#undef i386Architecture
#define Arm32Architecture

#undef OptimizedCDebugFlags
#define OptimizedCDebugFlags -O2
#define ServerCDebugFlags -O2
#undef StandardDefines
#define StandardDefines -Dlinux -D_arm_ -D_POSIX_SOURCE \
-D_BSD_SOURCE -D_GNU_SOURCE

#undef CcCmd
#define StdIncDir /opt/q+esto/cdt/arm-linux/include
#define PrefIncDir
#undef PostIncDir
#define PostIncDir /opt/q+esto/cdt/lib/gcc-lib/arm-linux/2.95.3/include
#define CcCmd /opt/q+esto/cdt/bin/arm-linux-gcc
#undef CplusplusCmd
#define HasCplusplus YES
#define CplusplusCmd /opt/q+esto/cdt/bin/arm-linux-g++
#define DoRanlibCmd YES
#define RanlibCmd /opt/q+esto/cdt/bin/arm-linux-ranlib
```

(그림 2) xc/config/cf/cross.def 파일

(그림 2)에서 보듯이 X 소스에 포함되어 있는 cross.def 파일의 컴파일 환경을 호스트에 설정된 크로스 컴파일 환경(/opt/q+esto/cdt/arm-linux)으로 재 설정하고 있다. x86용 크로스 컴파일 환경도 arm 크로스 환경과 유사하게 /opt/q+esto/cdt/i386-linux로 설정된다.

(그림 3)은 X 소스에서 Qplus-P tiny-X 그래픽 서버를 구성하기 위한 host.def 파일 내용을 간략히 보여준다. Host.def 파일은 본래 X 소스에 포함되어 있지 않고, handhelds 프로젝트[20]를 참조하여 Qplus-P 구현환경에 맞도록 새롭게 구성하였다. (그림 3)은 아이팩 3600 PDA를 위한 tiny-

X 서버 구성을 보여준다.

Host.def 파일에서는 X 컴파일 시 구성할 서버들을 지정하게 되는데, (그림 3)에서 보듯이 tiny-X 서버(TinyXServer), 프레임버퍼 서버(XfbdevServer), 터치스크린(Touch Screen), 화면 회전 기능(BuildRandR) 등이 YES로 설정되어 있다. X 소스 컴파일 시에 host.def에 설정된 값들을 참조하여, 해당 서버 및 기능들을 선택적으로 설치하게 된다.

```
#define KDriveXServer YES
#define XipaqServer YES
#define KdriveServerExtraDefines -DMAXSCREENS = 3
#define TinyXServer YES
#define XfbdevServer YES
#define CrossCompiling YES
#define TouchScreen YES
#define ItsyCompilerBug YES
#define BuildXinerama YES
#define BuildRandR YES
#define BuildFreeType YES
#define BuildXInputLib YES
#define HasLibCrypt YES
#define ProjectRoot /usr/X11R6
```

(그림 3) xc/config/cf/host.def 파일 내용

3.1.2 한글 처리 기능

일반적으로 X에서는 한글처리 기능을 제공하지만, tiny-X에서는 한글 처리기능을 제대로 제공하지 못한다. 즉, tiny-X에서는 한글 출력시에, 한글이 깨지는 문제점이 발생하였다. 실시간 운영체제 Qplus-P는 국내 정보가전 기기에 사용되는 것을 목표로 하였기 때문에 한글처리 기능은 필수적으로 요구되었다. Qplus-P 그래픽 윈도우 라이브러리는 한글 처리를 위해서 다수의 X 소스 파일을 수정하였다. (그림 4)는 수정된omText.c 파일 내용을 일부 보여준다.

```
static int DrawStringWithFontSet(...) {
    ...
    ptr = (unsigned char *)text;
    is_xchar2b = fs -> is_xchar2b;

    while (length > 0) {
        ...
        if (is_xchar2b) {
            // char_len = ptr_len / sizeof(XChar2b);
            char_len = ptr_len / 2; // modified
            XDrawString16(dpy, d, gc, x, y, (XChar2b *)
                ptr, char_len);
            x += XTextWidth16(font, (XChar2b *)ptr,
                char_len);
        } else {
            char_len = ptr_len;
            XDrawString(dpy, d, gc, x, y, (char *)ptr,
                char_len);
            x += XTextWidth(font, (char *)ptr, char_len);
        }
        break;
    }
    ...
}
```

(그림 4) xc/lib/X11/omText.c 파일 내용

(그림 4)에서 보듯이 기존의 omText.c 파일의 한글 처리 부분인 sizeof(XChar2b)는 하드웨어에 따라 2가 아닌 다른 값(예를 들어, 4)을 갖게 된다. 한글 처리를 위해서는 2 바이트 단위로 처리되어야 하기 때문에, Qplus-P 그래픽 라이브러리에서는 sizeof(XChar2b)을 2로 변경하였다. 한글 처리를 위해서 다수의 파일들이 변경되었는데, 여기에는 omTextEsc.c, ULabel.c, XlibInit.h 등이 있다. XlibInit.h 파일에는 MUSTCOPY가 새롭게 정의되었는데, MUSTCOPY는 데이터 복사시 4바이트 단위로 복사하게 된다. (그림 5)는 XlibInit.h 파일 내용을 일부 보여준다.

```
# define MUSTCOPY /* added */
#ifdef _XLIBINT_H
#define _XLIBINT_H_1
...
#ifdef MUSTCOPY

/* for when 32-bit alignment is not good enough */
#define OneDataCard32(dpy, dstaddr, srcvar) \
    { dpy -> bufptr -= 4; Data32(dpy, (char *) &(srcvar), 4); }

#else

/* srcvar must be a variable for large architecture version */
#define OneDataCard32(dpy, dstaddr, srcvar) \
    { *(CARD32 *)dstaddr = (srcvar); }

#endif /* MUSTCOPY */
...
```

(그림 5) XlibInit.h 파일 내용

3.1.3 터치스크린 입력

QPlus-P 그래픽 윈도우 라이브러리는 PDA와 같은 소형 정보 가전기에도 탑재되므로, 터치스크린 방식의 입력도 지원해야 한다. (그림 6)는 터치스크린 입력을 지원하기 위한 kinput.c 파일 내용을 보여준다.

```
static int KdMouseProc (...) {
    ...
    switch (onoff)
    {
        case DEVICE_INIT :
            ...
        case DEVICE_ON :
            pDev -> on = TRUE;
            pKdPointer = pDevice;
#ifdef TOUCHSCREEN
            if (kdTsFuncs)
                (*kdTsFuncs -> Init) ();
#endif
            if (kdMouseFuncs)
                (*kdMouseFuncs -> Init) ();
            break;
        case DEVICE_OFF :
```

(그림 6) xc/programs/Xserver/hw/kdrive/kinput.c 파일 내용

(그림 6)에서 보듯이 터치스크린 입력을 지원하기 위해

기존의 kinput.c 파일에서 마우스 초기화보다 먼저 터치스크린을 초기화하여 사용자가 펜으로 입력할 수 있도록 하였다.

3.1.4 그래픽 가속

Qplus-P 그래픽 윈도우 라이브러리는 한국전자통신연구원 에서 시험용으로 제작한 x86 프로세서를 사용하는 정보가전 기기 홈서버 1에 이식되었는데, 홈서버 1에는 그래픽 카드로 nVidia GeForce II를 탑재하고 있다. PDA와 같은 소형 정보가전 기기와 달리 홈서버 I에 탑재된 nVidia 그래픽 카드는 고 사양의 카드로서, 하드웨어 가속이 가능한 제품이다. 홈서버 1은 중대형 정보가전 기기로서, DVD 재생기 등과 같은 다양한 응용 프로그램들이 수행되어야 한다. 하지만, 기존의 tiny-X 그래픽 서버에서는 nVidia 그래픽 카드를 위한 가속 기능이 제공되지 않기 때문에, 멀티미디어 데이터를 재생하는 데 속도가 떨어지는 문제점이 발생하였다. 따라서, Qplus-P 그래픽 라이브러리 구현에서는 xfree86 소스에 포함된 nVidia 드라이버(xc/programs/Xserver/hw/xfree86/driver/nv)를 참조하여, tiny-X 용 nVidia 그래픽 드라이버(xc/programs/Xserver/hw/kdrive/nv)를 새롭게 구현하였다. (그림 7)는 하드웨어 초기화 부분이 포함되어 있는 nv.c 파일 일부를 보여준다.

```
#include "nv.h"

static void nVRIVASetup(NvCardInfo *nvc)
{
    CARD32 regBase = nvc->IOAddress ;

    nvc->riva.PRAMIN   = KdMapDevice (regBase+0x00710000,
                                     0x00010000) ;
    nvc->riva.PCRTC    = KdMapDevice (regBase+0x00600000,
                                     0x00001000) ;
    nvc->riva.IO       = VGA_JOBASE_COLOR ;
    nvc->riva.PRAMDAC  = KdMapDevice (regBase+0x00680000,
                                     0x00001000) ;
    nvc->riva.PFB      = KdMapDevice (regBase+0x00100000,
                                     0x00001000) ;
    nvc->riva.PFIFO    = KdMapDevice (regBase+0x00002000,
                                     0x00002000) ;
    nvc->riva.PGRAPH   = KdMapDevice (regBase+0x00400000,
                                     0x00002000) ;
    nvc->riva.PEXTDEV  = KdMapDevice (regBase+0x00101000,
                                     0x00001000) ;
    nvc->riva.PTIMER   = KdMapDevice (regBase+0x00009000,
                                     0x00001000) ;
    nvc->riva.PMC      = KdMapDevice (regBase+0x00000000,
                                     0x00009000) ;
    nvc->riva.FIFO     = KdMapDevice (regBase+0x00800000,
                                     0x00010000) ;
    ...
}
```

(그림 7) xc/programs/Xserver/hw/kdrive/nv/nv.c 파일 내용

nVidia 그래픽 카드를 위한 가속 기능을 제공하기 위해 xc/programs/Xserver/hw/kdrive에 nv라는 서브 디렉터리를

새롭게 생성하고, 이 디렉터리에 Imakefile, nv.c, nv.h, nvcurs.c, nvdraw.c, nvdraw.h, nvstub.c, riva_hw.c, riva_hw.h 등과 같은 tiny-X 그래픽 서버를 위한 파일들을 생성하였다. nVidia 그래픽 가속기능은 x86 용 tiny-X 서버에만 제공되고, arm 용 tiny-X에서는 지원하지 않는다.

3.2 Qplus-P 그래픽 툴킷과 그래픽 관련 라이브러리

Qplus-P 그래픽 툴킷은 공개 소스인 gtk-1.2.10을 기준으로 삼성 S3C2400 보드, 아이팩 PDA, 자우루스 PDA, 그리고 홈서버에 탑재하기 위해서, gtk-1.2.10 소스 파일을 크로스 컴파일하여 구현하였다. Gtk+-1.2.10 소스는 크로스 컴파일을 제대로 지원하지 않아 구현 시에는 configure 파일을 수정하여 컴파일 하였다.

QPlus-P 그래픽 관련 라이브러리인 glib, libjpeg, libpng, libtiff, gdkpixbuf 등은 공개소스를 기반으로 구현되었다. Glib의 소스 버전은 glib-1.2.10이고, arm 프로세서와 x86 프로세서를 위한 패치 파일을 적용하여 컴파일하였다. Libjpeg 라이브러리의 버전은 libjpeg-6b이고, libpng는 libpng-1.07을 기준으로 하였다. 그리고, libtiff는 libtiff-v3.5.5 버전을 기준으로 하였고, gdk-pixbuf는 gdk-pixbuf-0.9.0을 사용하였다. 이들 그래픽 관련 라이브러리들은 크로스 컴파일을 위해 적절히 패치한 다음, 삼성 S3C2400 보드, 아이팩 PDA, 그리고, 홈 서버에 탑재되었다.

Qplus-P 그래픽 툴킷과 그래픽 관련 라이브러리들은 크로스 컴파일 환경을 지원하도록 수정되었지만, 기능상으로도 별도로 추가된 내용이 없으므로 본 논문에서는 자세한 설명은 생략하도록 한다.

4. 그래픽 윈도우 라이브러리의 특징 및 실행 예

4.1 그래픽 윈도우 라이브러리의 특징

Qplus-P 그래픽 윈도우 라이브러리는 로얄티가 없고 응용 프로그램에 다양한 인터페이스를 제공하며, 동적으로 응용 프로그램을 설치 및 제거가 가능해야 한다는 요구사항을 충족시키기 위해 공개 소스인 X와 gtk+ 그래픽 툴킷, 그리고 기타 그래픽 관련 공개 소스들을 기반으로 구현되었다. Qplus-P 그래픽 윈도우 라이브러리는 다음과 같은 특징이 있다.

- Qplus-P tiny-X 그래픽 서버는 한글 처리 기능을 제공한다. 이는 기존의 tiny-X 그래픽 서버가 한글 출력 시 한글이 깨지는 문제점을 해결한 것으로, 국내 정보가전 기기에 탑재될 때 한글 입출력 기능을 제공한다. 그리고, tiny-X 그래픽 서버는 화면 회전, 터치 스크린 입력 등 PDA와 같은 소형 정보가전 기기에 필요한 기능들도 제공한다.

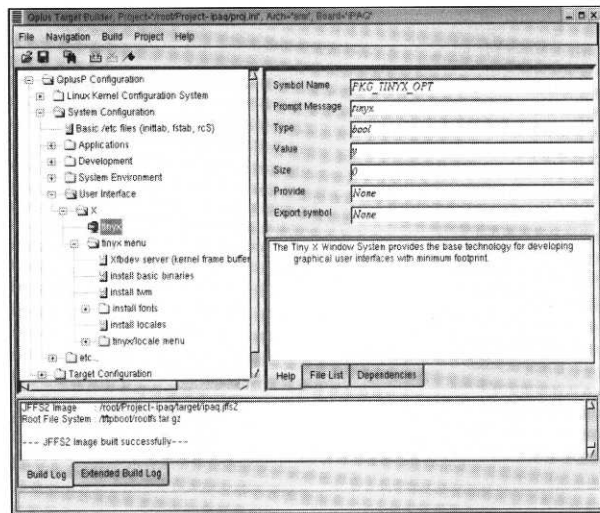
- Qplus-P 그래픽 윈도우 라이브러리의 tiny-X 그래픽 서버는 nVidia GeForce II 그래픽 카드에 대한 가속 기능을 제공한다. 기존의 tiny-X 그래픽 서버에서는 nVidia 그래픽 카드를 위한 가속 기능을 지원하지 않기 때문에, tiny-X 그래픽 서버용으로 새롭게 구현되었다. NVidia 그래픽 카드를 위한 그래픽 가속 기능은 홈서버 1에서 멀티미디어 데이터를 재생하는 데 필요한 가속 기능을 제공하게 된다.
- Qplus-P 그래픽 라이브러리는 기존의 X 라이브러리보다 라이브러리 크기를 상당히 줄이고 있다. 하드웨어 자원이 제한되어 있는 소형의 정보가전 기기에 탑재되기 위해서 그래픽 윈도우 라이브러리를 정보가전 응용에 적합한 기능들로 구성하여, 라이브러리 크기를 줄이는 것이 중요하다. <표 2>은 x86용 Qplus-P 그래픽 윈도우 라이브러리와 표준 X 라이브러리의 크기를 비교한 것이다. <표 2>에 표시되는 라이브러리의 크기는 /usr/X11/R6/lib 디렉터리 중에서 X11/, modules/ 등을 제거한 순수한 라이브러리 크기를 보여준다.

<표 2> 라이브러리 크기 비교

구분	Qplus-P 그래픽 윈도우 라이브러리	표준 X 윈도우 시스템
크기	3.3M	12M

4.2 그래픽 윈도우 라이브러리의 수행

Qplus-P 그래픽 윈도우 라이브러리는 일반적으로 다른 응용 프로그램에 포함되어 수행되므로, 이 절에서는 Qplus-P 응용 프로그램들의 수행 예제를 통해 그래픽 윈도우 라이브러리의 수행을 간접적으로 보여준다.



(그림 8) 타겟빌더에 포함된 그래픽 윈도우 라이브러리

Qplus-P 그래픽 윈도우 라이브러리는 Qplus-P 타겟 빌

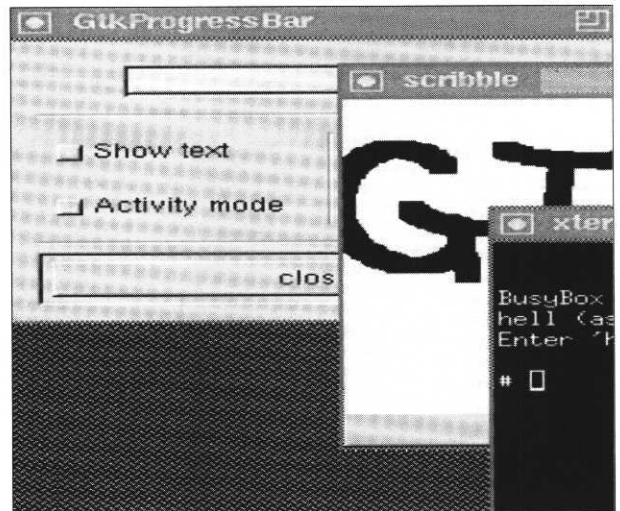
더에 통합되어 제공되는데, (그림 8)은 타겟 빌더에 통합된 그래픽 윈도우 라이브러리를 보여준다.

(그림 9)은 홈서버 1에서 Qplus-P를 이용한 DVD 플레이어의 수행 화면을 보여준다. (그림9)에서 보듯이 Qplus-P는 홈 서버에 탑재되고, DVD 플레이어는 홈 서버의 한 응용으로 nVidia 그래픽 가속 기능을 이용하여 동영상 재생을 한다.



(그림 9) DVD 플레이어 수행 화면

(그림 10)은 arm 프로세서를 사용하는 S3C2400 보드에 탑재된 그래픽 윈도우 라이브러리의 수행 화면을 보여준다. S3C2400 보드에서는 터치 스크린 입력과 USB 마우스 입력을 선택하여 사용할 수 있다.



(그림 10) S3C2400 보드에서의 수행 화면

(그림 11)은 홈서버 1에서 전자책 리더(eBook Reader)의 수행 화면을 보여준다. (그림 11)에서 보듯이 전자책 리더에서 화면 크기 때문에 전자책이 제대로 보이지 않을 경우 화면 회전 기능을 사용하여 전자책을 제대로 출력하고 있다.



(그림 11) 홈서버에서 전자책 리더 수행 화면

5. 결론 및 향후 연구

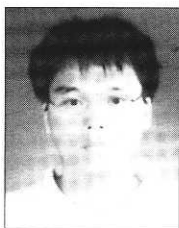
본 논문에서는 정보가전용 실시간 운영체제 Qplus-P에 탑재되는 그래픽 윈도우 라이브러리의 구현에 대해 기술하였다. 실시간 운영체제 Qplus-P는 크게 커널, 라이브러리, 통합 개발 도구로 구성된다. 그래픽 윈도우 라이브러리는 공개 소스인 X(버전 4.2.0)와 gtk+(1.2.10)을 기반으로 하여, 소형 정보가전 기기에 적합한 기능과 크기를 갖는 tiny-X 그래픽 서버와 gtk+ 그래픽 툴킷을 제공한다. 그리고, Qplus-P 그래픽 응용 프로그램 개발에 필요한 glib, gdk-pixbuf, libjpeg, libtiff, zlib과 같은 라이브러리들도 제공한다. 구현된 그래픽 윈도우 라이브러리는 표준 X 라이브러리보다 라이브러리 크기가 상당히 작고, PDA와 같은 소형정보가전 기기에 적합한 터치 스크린 입력 기능, 전자책 리더를 위한 화면 회전과 같은 기능 뿐만 아니라, nVidia 그래픽 카드를 위한 그래픽 가속 기능도 제공한다.

앞으로 Qplus-P 그래픽 윈도우 라이브러리는 임베디드

환경에 적합한 위젯들을 제공하여야 하고, 다양한 하드웨어에 탑재될 수 있도록 기능이 확장되어야 한다. 그리고, 그래픽 툴킷으로 gtk-2.x를 지원하는 연구도 진행해야 한다.

참 고 문 헌

- [1] 김도형, 신창민, 박승민, "EL/IX 단계 3을 적용한 실시간 운영체제 Qplus-P용 C 표준 라이브러리의 설계 및 구현", 정보처리학회논문지A, Vol.9-A, No.4, pp.485-490, Dec., 2002.
- [2] 강우철, 윤희철, "Qplus 타겟 빌더 : 임베디드 리눅스 툴킷", 한국정보처리학회 춘계학술발표논문집, 제9권 제1호, pp. 663-666, 2002.
- [3] 김대희, 남영광, 김홍남, 이광용, "RTOS 용 원격 대화형 웹 설계 및 구현", 정보처리학회논문지D, Vol.9-D, No.4, pp. 677-686, Aug., 2002.
- [4] 한국전자통신연구원, "인터넷정보가전기술개발 워크샵 자료집", 2001.
- [5] <http://www.gtk.org>.
- [6] <http://www.directfb.org/gtk.xml>.
- [7] <http://www.trolltech.com/products/embedded/indepth.html>.
- [8] <http://www.xfree86.org>.
- [9] <http://developer.gnome.org/doc/API/gdk-pixbuf>.
- [10] <http://www.jpeg.org>.
- [11] <http://www.zip.org/zlib>.
- [12] <http://www.libtiff.org>.
- [13] <http://www.libpng.org/pub/png>.
- [14] <http://www.ftk.org>.
- [15] 김도형, 박승민, "실시간운영체제 Q+를 위한 라이브러리 설계 및 구현", 정보처리학회논문지D, Vol.9-D, No.1, pp.153-160, Feb., 2002.
- [16] "VxWorks 5.3.1 Programmer's Guide Edition 1," Wind River Systems, 1997.
- [17] "VxWorks Training Workshop," Wind River Systems, 1996.
- [18] "pSOSystem Programmer's Reference," Integrated Systems, 1997.
- [19] "VTRX Reference Guide," Mentor Graphics Corporation, 1997.
- [20] <http://www.handhelds.org>.
- [21] <http://www.microwindow.org>.



김도형

e-mail : kimdh@etri.re.kr

1993년 경북대학교 컴퓨터공학과(공학사)

1995년 포항공과대학 전자계산학과

(이학석사)

1995년~현재 한국전자통신연구원 컴퓨터

소프트웨어기술연구소 선임연구원

관심분야 : 결함포용, 실시간시스템, 성능감시, 성능평가



김 선 자

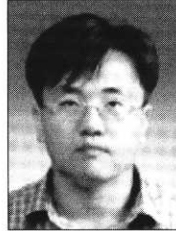
e-mail : sunjakim@etri.re.kr

1985년 숙명여자대학교 수학과(이학사)

1995년 충남대학교 대학원 컴퓨터 공학과
(공학석사)

1987년~현재 한국전자통신연구원 컴퓨터
소프트웨어기술연구소 선임연구원

관심분야 : 운영체제, DSM, 임베디드 시스템



김 성 우

e-mail : libero@dongeui.ac.kr

1991년 한국과학기술원 전기 및 전자공학과
(공학사)

1993년 한국과학기술원 전기 및 전자공학과
(공학석사)

1999년 한국과학기술원 전기 및 전자공학과
(공학박사)

1999년~2001년 한국전자통신연구원 컴퓨터소프트웨어기술연구소
선임연구원

2002년~현재 동의대학교 컴퓨터영상공학부 소프트웨어공학전공
전임강사

관심분야 : 실시간 운영체제, 그래픽 윈도우 시스템, 내고장성
시스템