

임베디드 리눅스 시스템을 이용한 디지털 사진 액자 구현

현경석¹, 이명의^{2*}

Implementation of Digital Photo Frame using Embedded Linux System

Kyung-Seok Hyun¹ and Myung-Eui Lee^{2*}

요약 본 논문에서는 디지털 카메라의 사진을 메모리 카드를 통해 입력받고 디스플레이하며 각 사진에 대한 음성 레코딩과 MP3 플레이가 가능한 디지털 사진 액자 시스템 구현에 대하여 기술한다. Intel PXA255 보드의 시스템 제어 위한 부트로더와 리눅스 커널을 포팅하며 외부 장치들을 위한 디바이스 드라이버를 작성한다. 리눅스 시스템 상에서 이미지 출력 및 음성 레코딩, MP3 플레이어 기능을 구현하기 위해 마이크로윈도즈 시스템의 구성 파일을 수정하고 응용 프로그램을 작성한다. 본 논문 연구를 통해서 저 전력, 고성능의 임베디드 프로세서와 리눅스 시스템을 이용한 디지털 사진 액자 개발에 쉽게 접근할 수 있으며 구현된 디바이스 드라이버와 응용 프로그램 개발 절차를 통해 임베디드 시스템 개발과 관련한 분야에 기초 자료로 사용할 수 있을 것이다.

Abstract In this paper, we describe the implementation of the digital photo frame system that displays the images coming through the memory card of a digital camera. Each image can be recorded with voice in this system, and a function of the mp3 player is implemented as well. We use Intel PXA255 to control the system and modify the bootloader and linux kernel. Also we adapt device driver for this system. For the realization of image display, voice recording and mp3 playing in the basis of the linux system, we program some of the Microwindows system configuration files and program applications here. This study will be a good example to access the development of the digital photo frame based on the linux system using less-power and high performed embedded processor.

Key Words : 임베디드 리눅스, 임베디드 시스템, 커널, 디바이스 드라이버

1. 서론

디지털 기술의 발달은 디지털 카메라와 같은 디지털 영상 기기의 급속한 확산을 가능하게 하여 주었고, 전 세계 어느 관광지에서나 디지털 영상기기를 다루고 있는 사람을 쉽게 발견할 수 있다. 하지만 이러한 디지털 영상 기기들은 현대의 편의성을 고려하여 제작 되어진 관계로 매우 작은 뷰파인더를 채택하고 있어 현대인의 특성 중의 하나인 현장감 추구에 크게 못 미치고 있다. 이러한 사회 환경적인 변화에 맞추어 여가선용 중 취득한 영상을 관리, 보존 하여 여가생활에 대한 만족도를 배가시킬

수 있도록 하는 디지털 사진 액자에 대한 필요성이 대두되고 있으며 이러한 디지털 사진 액자의 기능을 구현하기 위해서는 임베디드 시스템이 있어야 한다.

본 논문에서는 Intel의 PXA255 CPU를 사용한 보드에 맞도록 개발환경을 구축하고 외부 사용자 입력, 사운드 출력, 디스플레이 장치, 저장 장치들을 위한 디바이스 드라이버를 작성한 후 커널을 포팅하는 절차를 서술한다 [1]. 그리고 리눅스 시스템 상에서 이미지 출력 및 음성 레코딩, MP3 플레이가 가능한 응용프로그램을 개발하는 절차를 서술한다.

2. 시스템 개발 환경 구축

임베디드 시스템을 개발하기 위해서는 개발 환경을 갖

¹토피드 연구소

²한국기술교육대학교 정보기술공학부

*교신저자: 이명의(melee@kut.ac.kr)

추여야 하는데 일반적으로 호스트 PC에서 개발을 한 후 통신채널을 이용해 임베디드 보드에 적용한다. 호스트 PC에서의 개발을 위해서는 임베디드 시스템에 맞는 교차 컴파일러(Cross-Compiler) 설치, 상호간의 통신을 위한 설정, 커널 설정, 개발 보드용 응용 프로그램 개발, 프로그램의 테스트와 검증 등의 순서로 개발을 진행한다.

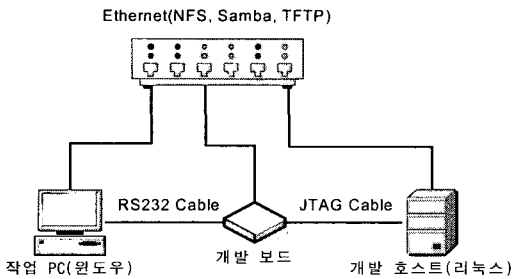


그림 5. 개발 환경 구성도

본 논문에서의 개발환경은 [그림1]과 같이 구성하며 개발 호스트로 리눅스를 설치하고 크로스 컴파일 개발 환경을 구축한다. 주 작업은 윈도우 PC에서 하며 초기 개발 보드에 부트로더 이미지를 써 넣기 위해서 JTAG를 사용한다. 그러나 JTAG의 처리속도는 매우 느리므로 개발 보드에서 부트로더가 동작 한 이후에는 실행 이미지를 다운로드하기 위해서 이더넷을 이용한 네트워크상에서의 다운로드를 수행하는데 이때 사용되는 것이 TFTP이다.

2.1 크로스 컴파일 환경 구축

일반적인 시스템에서의 컴파일러는 자신의 시스템에 맞는 바이너리코드를 만든다. 개발 보드에 맞는 바이너리 파일을 만들려고 한다면 개발보드에서 직접 응용프로그램이나 커널 컴파일을 하여야 하지만 저장할 수 있는 공간도 부족 할 뿐더러 시스템 성능도 따라주질 않는다. 그래서 개발보드용 커널 및 응용프로그램을 개발하기 위해서 호스트 시스템에 개발보드용 크로스 컴파일 환경을 구축한다.

표 1. 크로스 컴파일 환경 구축에 필요한 패키지

패키지	설명	버전
binutils	어셈블러 및 로더, 기타 툴	2.13.90
glibc	크로스 컴파일을 위한 라이브러리 및 일반 라이브러리	2.3.1
gcc	컴파일러	3.2.1

표 2. 리눅스 커널 설정에 필요한 파일

용도	파일 이름	웹 주소
일반 리눅스	linux-2.4.19.tar.gz	ftp.kernel.org
ARM용 패치	patch-2.4.19-rmk7.gz	ftp.arm.uk.linux.org
XScale용 패치	diff-2.4.19-rmk7-pxa1.gz	ftp.arm.uk.linux.org
보드용 패치	diff-2.4.19-rmk7-pxa1-ez-x5.gz	falinux.com

크로스 컴파일 환경을 구축하려면 각종 소스들을 컴파일하고 빌드하여 실행바이너리를 생성하는데 필요한 각종 유틸리티 및 라이브러리가 필요하다. 기본적으로 어셈블러, 링커, C컴파일러, C라이브러리 등으로 구성되며 본 논문에서는 GNU에서 제공하는 [표 1]과 같은 패키지를 사용한다.

2.2 리눅스 커널 설정

ARM 리눅스용 커널은 따로 존재 하는 것이 아니라 일반적인 리눅스 커널에서 ARM용 패치를 적용하여 만든다. 리눅스 커널 자체가 상당히 이식하기에 좋은 구조로 되어 있기 때문에 이를 버리고 다른 커널을 디자인 할 필요는 없다. 본 논문에서 사용 할 리눅스 커널을 [표 2]와 같이 해당 웹 사이트에서 다운로드 받은 후 설정을 한다.

2.3 부트로더 설정

부트로더는 시스템에 전원을 올린 후 처음으로 동작하는 소프트웨어를 말하며 사용자의 반응에 따라 상호 대화식으로 동작하기도 하면서 운영체제 부팅에 필요한 각종 사항을 처리하는 기본 시스템 소프트웨어이다.

부트로더의 가장 큰 기능은 하드웨어의 초기화와 커널 부팅에 필요한 작업을 들 수 있다. 이와 더불어 커널과 루트 파일시스템의 다운로드 기능이 있다. 디지털 사진 액자의 부트로더는 FALINUX에서 제공하는 ezboot를 기본으로 하여 시스템 구성에 필요한 부분들을 수정한다.

우선 디지털 사진 액자에서의 PXA255 CPU는 200Mhz 용으로 사용된다. 이를 위해 CPU Speed를 설정하고 디스플레이 설정에 메모리 클럭이 사용되는데 이 클럭을 맞춰주기 위해 메모리 클럭을 100Mhz로 수정한다.

3. 하드웨어 구성 및 디바이스 드라이버

본 논문의 하드웨어 구성은 Falinux의 EZ-X5 보드를

바탕으로 하여 수정한 후 제작한다. EZ-XS 보드의 하드웨어 구성과 디바이스 드라이버를 참조하여 이용하며 새로이 추가되는 하드웨어에 관해서는 디바이스 드라이버를 새롭게 작성하며 [그림 2]는 디지털 사진 액자의 전체 하드웨어 구조를 나타낸다.

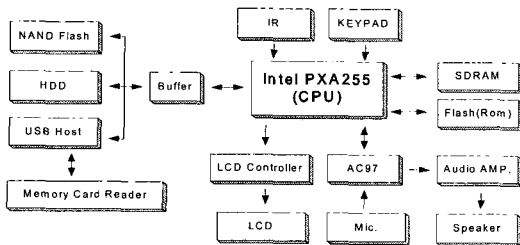


그림 2. 디지털 사진 액자 하드웨어 구조

하드웨어 구성에 있어 프로세서는 Intel Xscale PXA255를 사용하였다. PXA255 프로세서는 ARM5TE와 호환되고 빠른 업그레이드를 위해 Intel SA-1110 프로세서와 호환될 수 있는 애플리케이션 코드를 갖고 있다. 또한 저전력으로 높은 코어 속도를 내도록 첨단 Intel 0.18 μm 처리를 이용한 Intel Superpipelined RISC 기술을 채택하고 있다.

3.1 메모리

SDRAM으로 사용되는 메모리는 커널과 램디스크, 어플리케이션용으로 뱅크 0에 16Bit 폭의 2개의 SDRAM 64Mbytes를 사용한다. 그리고 부트로더와 램디스크 이미지, 커널 이미지를 저장하는 용도로 4Mbytes NOR Flash를 ROM으로 사용한다. [그림 3]은 Flash ROM의 메모리 맵 구조를 나타낸다. 4Mbytes 사이즈의 Flash 롬을 이용하여 커널 이미지와 램디스크 이미지를 모두를 저장하게 되는데 이를 위해서 커널 이미지와 램디스크 이미지의 최소화가 필요하다.

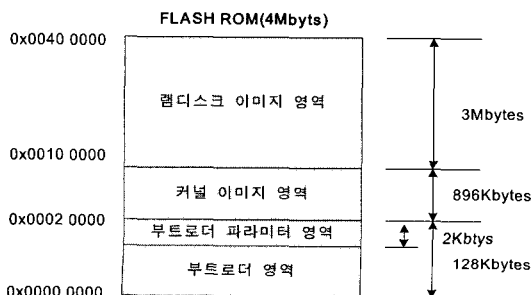


그림 3. Flash ROM 메모리 맵

3.2 입력 장치

디지털 사진 액자의 입력은 키패드와 IR 리모콘을 통해서 받는다. 키패드인 경우 각각의 버튼마다 GPIO 포트를 할당하여 입력을 받도록 하며 디바이스 드라이버는 IR 수신부와 유사하게 작성한다.

리모콘에서 송신되는 제어신호는 디지털 사진 액자의 IR 리시버로 받아들인 후 PXA255의 GPIO 포트로 펄스 형태의 데이터를 보낸다. 리모콘의 송신 신호는 [그림 4]와 같은 펄스가 나타나는데 각 펄스의 Falling Edge 마다 인터럽트가 발생하도록 하여 시간을 체크한 후 데이터를 얻는다.

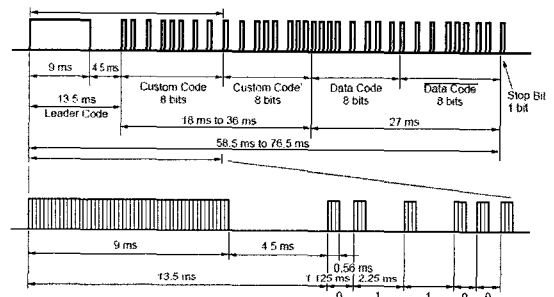


그림 4. NEC Simple Repeat 코드

응용 프로그램에서 데이터 값을 읽어 갈 경우에는 select 함수를 이용하게 되는데 이 함수를 호출하면 디바이스 드라이버의 poll 함수를 호출 한 후 데이터 값의 상태를 체크한다. 유효한 데이터가 있을 경우 응용 프로그램의 read 함수를 이용하여 데이터를 읽는다. 이러한 구조에 적합 하도록 리모콘 수신 데이터는 인터럽트에 의한 처리로 유효한 데이터가 들어오는지 여부를 체크한 후 플래그를 설정하게 하고 응용 프로그램에서 데이터를 읽어 가면 다시 플래그를 해지 하도록 구현한다.

3.3 저장 장치

디지털 사진 액자의 내부에 사용된 저장 장치로는 NAND Flash와 하드디스크가 있다. 하드 디스크는 리눅스에 제공되는 소스가 있으므로 구현된 하드웨어에 맞도록 include/asm-arm/arch-pxa/ide.h의 어드레스 매핑을 맞춘다.

외부 저장 장치로 메모리 카드가 사용되는데 각각의 장치에 대해 하드웨어 인터페이스를 구성하고 디바이스 드라이버를 작성하기에는 한계가 있으므로 USB Host Controller에 카드 리더를 연결하는 방식으로 사용한다. USB Host Controller로는 Cypress의 SL811HS를 사용하

는데 리눅스용 디바이스 드라이버를 제공한다. 디바이스 드라이버를 리눅스 커널 소스에 추가 한 후 커널 설정 메뉴에서 Mass Storage 항목을 추가한다.

3.4 디스플레이 장치

본 논문의 화면 출력장치로 SHARP의 8인치 LCD인 LQ080V3DG01을 사용한다[2]. LQ080V3DG01은 입력으로 TTL 방식을 사용하는데 PXA255의 TTL 출력을 그대로 보내준다. LCD Controller를 패널의 특성에 맞도록 설정하기 위해 커널을 수정해야 하는데 출력 신호는 패널의 특성에 맞도록 설정한다.

출력 설정은 픽셀 클럭, 수평, 수직 주파수, 시작, 끝대기 라인 값들을 설정한다. 픽셀 클럭 값은 메모리 클럭을 이용하여 만들어지며 LCCR3(Lcd Controller Control Register)의 PCD 값과 메모리 클럭 값을 이용하여 다음과 같은 공식에 의해 픽셀클럭 값이 얻어진다.

$$\text{픽셀 클럭} = \text{메모리클럭} / (2 * (\text{PCD} + 1))$$

4. 시스템 설정 및 소프트웨어 구현

본 논문의 어플리케이션은 마이크로윈도우즈 윈도우 환경에서 동작하며 이를 위한 선행 작업이 이루어진다. 모든 어플리케이션이 완성이 되면 라이브러리와 실행 파일에 필요한 모든 파일들을 포함하는 램디스크 이미지를 작성한다.

4.1 GUI 환경 설정

임베디드 시스템에 윈도우 환경이 꼭 필요하지는 않지만 일반 사용자들을 위한 사용자 인터페이스 부분을 고려해야 될 일이 많게 되면서 윈도우 환경도 시스템 구축에 중요한 관건이 되었다. 수행속도와 용량을 고려하여 프레임 버퍼를 통한 직접적인 컨트롤을 이용한 윈도우 환경을 구축 할 수도 있지만 시간단축과 안정성을 위해 기존에 나와 있는 윈도우 시스템을 이용한다. 현재 임베디드 시스템에 사용되고 있는 윈도우 시스템에는 Qt/임베디드, 타이니 X, 마이크로윈도우즈 등이 있는데 본 논문에서는 마이크로윈도우즈로 윈도우 환경을 구축한다[3-4].

그래픽 윈도우 환경에서의 입력은 리눅스 디바이스 드라이버 위에 추가적인 연결 블록을 필요로 한다. 본 논문에서는 키패드와 리모콘의 입력을 사용하는데 이를 위해 open, close, read 함수를 구현 한 후 장치파일에 추가한다.

이미지가 로드된 후 지정 된 사이즈의 이미지로 스케일링을 하게 되는데 마이크로윈도우즈 내부에 포함되어 있는 스케일링 기법은 사이즈 변형이 있을 경우 이전 이미지에서 스케일 된 이미지와 가장 유사한 위치의 픽셀 정보를 그대로 가져오는 형태의 스케일링을 하게 된다. 이와 같은 방법으로 스케일링이 이루어지면 빠른 속도의 장점이 있지만 곡선이나 대각선 같은 경우 너무나도 부자연스러운 이미지가 나타나게 된다. 이와 같은 단점을 보완하기 위하여 마이크로윈도우즈의 스케일 함수에 양선형(Bilinear) 보간 기법에 의한 스케일 함수를 추가한다[5].

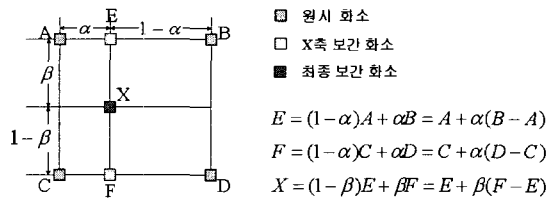


그림 5. 양선형 보간 기법

완벽한 양선형 보간 기법을 적용하려면 floating point 연산을 필요로 하게 되는데 floating point 연산을 PXA255 프로세서에서 지원하지 않기 때문에 고정 소수점 방식으로 연산을 수행하도록 한다[6-7].

4.2 어플리케이션 구현

어플리케이션에서는 입력장치 선택, 환경 설정, 이미지 뷰어, MP3 플레이, 음성 레코딩 기능들을 구현하는데 전체구조는 [그림 6]와 같다. 이미지 뷰어 기능과 타 기능들을 함께 사용하기 위하여 리눅스의 pthread 라이브러리를 이용하여 쓰레드 구성을 한다[8]. 그리고 음성 레코딩과 출력부는 실행파일 형태로 프로세서 호출을 하는 방식으로 구현한다.

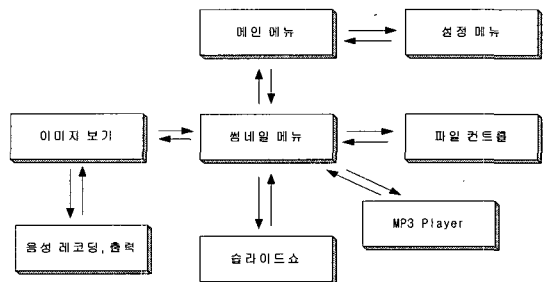


그림 6. 어플리케이션 블록도

어플리케이션의 상태 변화는 사용자의 입력에 의해 발생하는데 각 상태의 변화와 입력처리의 효율성을 위해 다음과 같은 구조로 프로그램을 구성한다.

우선 각 상태별 이벤트 처리를 위하여 다음과 같은 이벤트 핸들러 구조체를 정의 하는데 각 상태에서 처리하고자 하는 이벤트가 있다면 해당 되는 함수를 정의 한 후 그 함수 포인터를 원소로 하는 구조체를 선언한다.

이와 같은 구조체를 정의 한 후 `add_event_handler(handler_list_t *handler)` 함수를 이용하여 등록하며 상태에서 빠져나올 때는 `remove_event_handler(handler_list_t *handler)` 함수를 호출하여 이벤트 처리에서 빠져나온다. 그리고 프로그램의 통일성과 확장성을 위하여 각 상태의 기본 함수를 [표 3]과 같은 구조로 정의한다.

표 3. 어플리케이션 함수 구조

상태	함수명
진입	<code>void open_xxxx(GR_WINDOW_ID parent)</code>
종료	<code>void close_xxxx(void)</code>
화면 그리기	<code>void draw_xxxx(void)</code>
이벤트 처리	<code>void handle_xxx_events_xxxx(GR_EVENT *event)</code>

어플리케이션을 모두 작성한 후 마무리 단계로 필요한 파일만을 추출하여 최소한의 용량으로 램디스크 이미지를 작성한다. 어플리케이션이 간단하여 수행 파일 하나만으로 모든 작업이 가능하다면 간단하게 끝이 나지만 시스템 콜을 이용하여 외부 프로그램을 수행하거나 공유라이브러리를 이용하여 작성을 하게 되면 그에 해당되는 파일들도 포함 시켜야 한다.

본 논문에서는 약 4M 바이트 사이즈로 램디스크를 작성한다. 공유라이브러리와 시스템 유틸리티용으로 2M 바이트를 할당하고 어플리케이션 관련해서 2M 바이트를 할당한다.

5. 결론

본 논문에서는 디지털 카메라의 사진을 메모리 카드를 통해 입력받고 디스플레이 하며 각 사진에 대한 음성 레코딩과 MP3 플레이가 가능한 디지털 사진 액자 시스템 구현에 대하여 기술하였다.

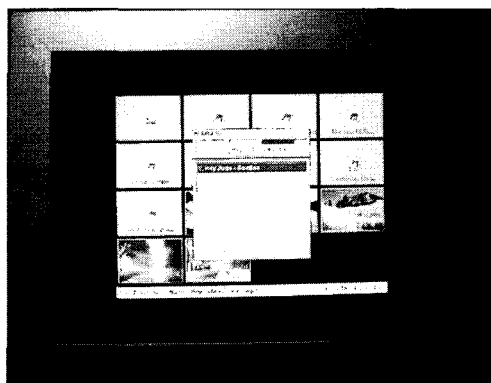


그림 7. 디지털 사진 액자 실행 화면

시스템 제어를 위해 부트로더와 리눅스 커널을 포팅하였으며 외부 사용자 입력, 사운드 출력, 디스플레이 장치, 저장 장치들을 위한 디바이스 드라이버를 수정 또는 작성하였다. 리눅스 시스템 상에서 이미지 출력 및 음성 레코딩, MP3 플레이가 가능한 어플리케이션을 개발하였다. 윈도우 환경을 위해 마이크로윈도즈 시스템의 구성 파일을 수정하여 키보드와 리모콘의 사용자 입력을 가능하도록 하였으며 이미지 처리가 효율적으로 되도록 해당 부분을 포팅하였다. 음성 레코딩과 MP3 플레이어 위해 고정 소수점 연산만을 사용하는 라이브러리를 이용해 구현하였다.

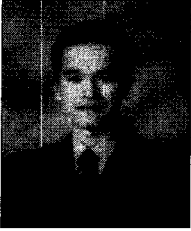
본 시스템 제작 연구를 통해서 저전력, 고성능의 임베디드 프로세서와 리눅스 시스템을 이용한 디지털 사진 액자 개발에 쉽게 접근할 수 있으며 이와 유사한 제품 개발에도 이용될 것으로 기대된다. 그리고 구현된 디바이스 드라이버와 응용프로그램 개발 절차를 통해 임베디드 시스템 개발과 관련한 분야에 응용 자료로 사용할 수 있을 것이다.

참고문헌

- [1] Intel XScale Microarchitecture for the PXA255 Processor User's Manual, 2003. 3
- [2] LQ080V3DG01 Data sheet, Sharp Corporation, 2003. 7
- [3] J. Lombardo. "Embedded Linux", New Riders Publishing 2001.
- [4] FALINUX(주), <http://www.falinux.com>
- [5] Korea Embedded Linux Project, <http://kelp.or.kr>
- [6] ARM Ltd., <http://www.arm.com>
- [7] ARM Linux, <http://www.arm.linux.or.uk>
- [8] Greg Haerr's Microwindows and NanoGUI, <http://microwinodw.org>

현 경 석(Kyung-Seok Hyun)

[정회원]



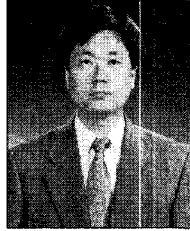
- 2003년 8월 : 한국기술교육대학교 정보기술공학부(공학사)
- 2006년 8월 : 한국기술교육대학교 대학원 전기전자(공학석사)
- 2006년 7월 ~ 현재 : 토필드 연구소(연구원)

<관심분야>

Embedded System, Operating System, 데이터 통신.

이 명 의(Myung-Eui Lee)

[정회원]



- 1985년 2월 : 인하대학교 전기공학과 (공학사)
- 1987년 2월 : 인하대학원 기기및 제어 (공학석사)
- 1991년 8월 : 인하대학원 기기및 제어 (공학박사)
- 1991년 8월 ~ 1995년 8월 : 현대전자 산업전자연구소 (선임연구원)
- 2004년 1월 ~ 2005년 1월 : U.C.Berkeley (객원교수)
- 1995년 9월 ~ 현재 : 한국기술교육대학교 정보기술공학부 교수

<관심분야>

Real-time Control System, Satellite Attitude & Orbit Control System, Embedded system