

논문 2007-44SC-4-5

WINDOWS CE 기반 VGA 카메라 모듈의 영상 획득과 저장을 위한 디바이스 드라이버 개발

(Development of Device Driver for Image Capture and Storage by
Using VGA Camera Module Based on Windows CE)

김 승 환*, 함 운 철**, 이 정 환*, 이 주 연*

(Seung-Hwan Kim, Woon-Chul Ham, Jung-Hwan Lee, and Ju-Yun Lee)

요 약

본 논문에서는 마이크로소프트사의 Windows CE 운영체제를 기반으로 한 소형의 모바일 시스템의 카메라 영상 획득을 위한 디바이스 드라이버에 관하여 살펴본다. 또한 NAND 플래시 메모리에 획득된 이미지를 저장하기 위하여 FAT 파일 시스템을 사용하였으며 NAND 메모리 특성을 반영한 FAT 파일 시스템에 대하여 다루어 본다. 영상획득을 위하여서 픽셀플러스사의 CMOS 카메라 모듈과 아지시스템에서 개발한 MBA2440 PDA 개발보드를 이용하여 하드웨어를 구현하였다. 이 카메라 모듈은 VGA 640 x 480 픽셀의 해상도를 지니고 있으며, 영상획득 속도 및 영상의 화질을 테스트하기 위한 디바이스 드라이버의 성능을 테스트 할 수 있는 응용 프로그램도 함께 제작하였다. 실험을 통하여 영상 획득을 위한 디바이스 드라이버와 FAT 파일 시스템을 이용하여 NAND 플래시 메모리에 획득한 영상을 저장하는 응용프로그램이 상호 잘 작동을 함을 확인하였다.

Abstract

In this paper, device driver for camera capture in hand held mobile system is implemented based on microsoft windows CE operating system. We also study the storage device driver based on the FAT file system by using NAND flash memory as a storage device. We use the MBA2440 PDA board for implementing the hardware for image capture by using CMOS camera module produced by PixelPlus company. This camera module has VGA 640x480 pixel resolution. We also make application program which can be cooperated with the device driver for testing its performance, for example image capture speed and quality of captured image. We check that the application can be cooperated well not only with the device driver for camera capture but also with the device driver for FAT file system designed especially for the NAND flash memory.

Keywords : Camera Capture, Device Driver, FAT, Mobile System, NAND flash memory.

I. 서 론

최근 인터넷의 폭발적인 성장에 따른 소비자들의 욕구 변화, 기술 발전 및 산업간 융합의 진전, 기존 PC 및 가전 시장의 성장세 둔화 등의 요인들이 맞물리면서, PC 및 가전 산업의 돌파구로 포스트 PC(Post PC) 제

품에 대한 관심이 높아지고 있다. 시장의 성장 한계, 경쟁 심화, 낮은 수익성 등의 문제로 고심하고 있는 PC, 가전, 정보통신 산업의 기업들에게 새롭게 부상하고 있는 포스트 PC 시장은 매력적인 사업 기회로 인식되고 있는 것이다.

포스트 PC의 개념과 범위에 대해서는 다양한 접근이 가능하겠지만, 단순히 설명하자면 포스트 PC 제품들은 특정 기능에 특화된 Non-PC형 정보기기로서 인터넷 접속과 간단한 컴퓨팅이 가능하며, 저렴하고 사용하기 쉬운 특징을 갖고 있다.

정보통신 기기를 중심으로 한 포스트 PC 시장이 활성화 되면서 이들에 내장되는 소형 운영체제인 임베디

* 학생회원, ** 정회원, 전북대학교 전자정보공학부
(Division of Electronics and Information
Engineering, Chonbuk National University)

※ 이 연구는 2단계 BK21사업의 지원을 통하여 이루어 졌음(This work was supported by the second stage of Brain Korea 21 Project)

접수일자: 2007년4월27일, 수정완료일: 2007년6월4일

드 운영체제에 많은 기업들이 주목하고 있다. 채택되는 임베디드 운영체제의 성능에 따라 포스트 PC형 정보통신 기기들의 신뢰성과 호환성이 큰 영향을 받기 때문이다.

임베디드 시스템에서 사용되는 OS는 여러 종류가 존재 하지만 크게 상용 RTOS (Real Time OS), 임베디드 리눅스, Windows CE등이 있다. RTOS로는 VxWorks, NucleOS, pSOS 등이 있다. 이런 상용 RTOS들은 실시간 처리가 가능하다는 것이 강력한 장점이지만 아울러 많은 비용을 소요하는 등의 단점이 있다. 임베디드 리눅스는 소스 코드가 공개되어 있어서 기술만 있으면 누구든지 독자적으로 임베디드 시스템을 개발할 수 있다는 장점이 있지만 실시간 처리가 상용 RTOS들보다 미흡하고, Windows CE에 비해 개발환경이 좋지 않다는 단점이 있다.^[1-2] Windows CE는 마이크로소프트사의 OS인 Windows 중에 임베디드 시스템을 겨냥한 OS로 Win32 API를 공유하고 소규모 롬 (ROM)기반에서 작동한다. 현재 6.0버전까지 출시되었으며, 네트워킹 가상메모리, 전력관리, 전원관리, 블루투스, 1394, 802.11기능 등이 추가되었고, 강력한 OEM지원이 가능하다.^[2]

Windows CE의 장점은 다음과 같은 6가지를 들 수 있다. 첫째, 다양한 종류의 기기에 사용가능한 개방형 구조로 디자인 되었다. 둘째, 32비트(bit), 멀티 테스킹, 멀티 쓰레드를 지원한다. 셋째, 적외선 포트, 시리얼 포트, USB등의 다양한 종류의 통신장치를 지원한다. 넷째, 4개의 코어 CPU(x86, ARM, MIPS, SH4) 프로세서를 지원하고 데스크탑 Windows OS 환경의 Win32 API(Application Programming Interface), MFC, .NET Application 등을 지원한다. 다섯째, 플랫폼 빌더라는 통합형 IDE환경 툴로 쉽게 개발 및 디버깅을 할 수 있어서 응용프로그램의 개발이 타 운영체제에 비하여 매우 쉽다. 이것은 제품 개발시간에 쫓기는 개발자들에게도 커다란 이점으로 다가오고 있다. 이러한 장점들로 인하여 타 임베디드 운영체제보다 Windows CE가 각광을 받고 있다.^[3]

그리고 요즘 출시되는 제품들을 보면 카메라를 내부에 포함한 제품이 많은 것을 알 수 있다. 이는 카메라 수요의 증가와 기술의 발전으로 좋은 성능의 카메라 모듈이 개발되고 또 그것의 가격이 하락 하여, 휴대폰 및 컴퓨터 등의 제품들에 많이 보급되었기 때문이다. 자연스럽게 사람들도 카메라에 관심을 많이 보이고 있으며, 카메라가 포함된 제품은 더욱 늘어날 것 보인다. 또한 카

메라의 용도도 다양해질 것이다. 그래서 카메라를 제어할 수 있는 기술은 여러 분야에 필요할 것이다.

본 논문은 많은 장점을 가지고 있는 운영체제 Windows CE를 이용하여 수요가 늘고 있는 카메라의 디바이스 드라이버를 제작하는 것을 서술한다. 아지시스템사의 MBA2440보드를 이용하여 여기에 카메라모듈을 연결시켜 우선 펌웨어를 제작하여 구동시키고, 이것을 기반으로 Window CE에서 사용가능한 디바이스 드라이버를 제작하여 카메라를 작동 시키고, 여기서 획득한 영상을 NAND 플래시 메모리에 저장하는 방법에 대해 전반적으로 다룰 것이다. 본 논문에 사용된 보드인 아지 시스템사의 MBA2440보드는 삼성의 ARM계열 CPU인 S3C2440A를 사용하며, CPU에 카메라 컨트롤러를 내장하고 있다. 카메라 모듈은 픽셀플러스사의 PO2030N으로 30만 화소에 CMOS타입으로 VGA급 (640*480)의 카메라 기능을 갖고 있다.

본 논문의 전개는 다음과 같다. II장에서는 전반적인 카메라 동작과 각 모듈, 통신, 파일시스템의 내용을 서술한다. III장에서는 펌웨어가 동작하는 부분을 서술한다. 다음으로 IV장에서는 펌웨어 상에서 만든 프로그램을 Windows CE의 디바이스 드라이버로 구현하는 방법과 이것을 적용하는 것을 다룬다. 마지막으로 V장에서 결론과 향후 연구 방향에 대해 서술한다.

II. 카메라 디바이스 드라이버의 전반적인 작동

우선 카메라의 전체적인 작동은 다음과 같다.

그림 1에 따르면, 보드에서 클럭을 카메라 모듈에 제공하면 카메라는 이 클럭을 가지고 컨트롤러를 작동 시

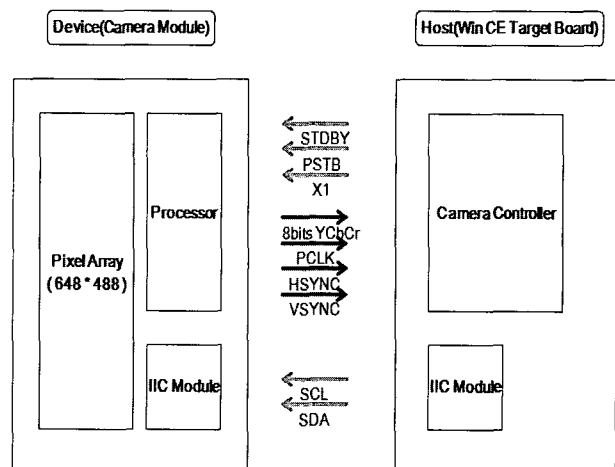


그림 1. 카메라의 전반적인 작동 구조
Fig. 1. General operation structure of camera.

켜 CMOS 센서로부터 화상 정보를 읽어 보드로 데이터를 보내게 된다. 보드의 카메라 컨트롤러는 카메라로부터 오는 데이터를 카메라의 픽셀 클럭, 수직동기 (Vertical Sync), 수평동기(Horizontal Sync) 신호와 동기를 맞추어서 데이터를 읽어 이를 내부 메모리에 저장하게 된다. 그러면 CPU는 저장된 데이터를 그래픽 디스플레이 장치에 표시하거나, 저장 장치에 저장한다.

1. 카메라 모듈의 세부적인 동작

카메라 내부는 그림 2의 블록 다이어그램과 같이 구성되어 있으며 648*488 CMOS센서가 있고 여기서 얻은 영상 데이터를 타임 컨트롤러가 영상신호처리(Image Signal Processing) 블록으로 보낸다. 영상처리 블록은 받은 영상 데이터를 변환 후 동기 신호와 함께 외부로 내보내게 된다. 영상 출력 방식에 있어서 카메라는 ITU-R BT.601/656 방식을 지원하며 본 논문에서는 ITU-R BT.656 방식을 사용하였다^[4].

카메라 모듈 내부는 외부 장치와 IIC로 통신을 해서 카메라 모듈 내부의 레지스터를 변경하여 카메라의 설정을 변경하게 된다. 외부로 나와 있는 선을 보면 크게 3부류로 나누어지는데, 카메라 모듈에 클럭을 공급하고 리셋하는 제어선이 있으며, 영상 데이터를 외부로 보내고 영상 데이터를 동기화 하기위한 클럭 신호선이 있다. 그리고 마지막으로 위에서 설명한 IIC 통신선이 있다.

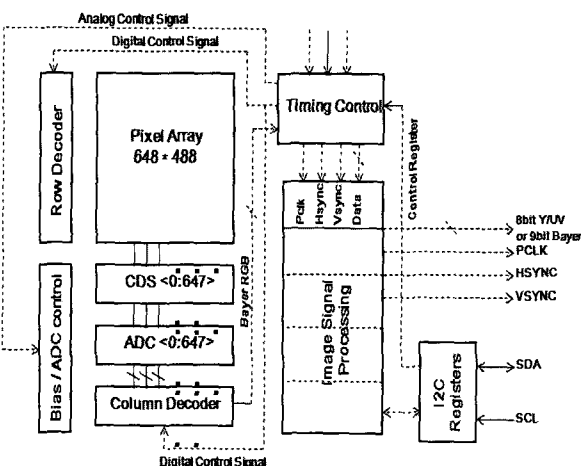


그림 2. 카메라 모듈의 블록 다이어그램
Fig. 2. Block diagram of camera module.

2. IIC 통신

최초 필립스가 제안한 통신 방식인 IIC 통신은 기본 구성은 2개의 bus를 사용한다. 이 두 가지의 선은 각각

Single Write Mode Operation

S	Slave Address	W	A	Register ADR	A	DATA	A	P
---	---------------	---	---	--------------	---	------	---	---

Single Read Mode Operation

S	Slave Address	W	A	Register ADR	A	
S	Slave Address	W	A	DATA	NP	P

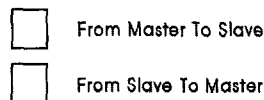


그림 3. IIC 데이터 흐름
Fig. 3. IIC Data Flow.

통신에 필요한 클럭을 위한 라인(Serial Clock Line)과 데이터 전송용 라인(Serial Data Line)으로 구성되어 있다. 그리고 이 라인은 데이터 전송과 기준 클럭을 발생 및 중지 시키는 마스터(Master)와 그 마스터의 신호를 받는 슬레이브(Slave)로 나눌 수 있으며 1:1통신만이 아닌 여러 슬레이브가 하나의 마스터와 연결하여 통신을 할 수 있다.

IIC통신은 클럭을 제공하는 SCL과 데이터 통신을 하는 SDA 두 선만 필요하므로 회로가 간단하다는 장점이 있으며, 클럭을 마스터에서 공급하므로 동기를 맞추지 않아도 되는 장점이 있다. IIC통신으로 데이터를 주고받을 때는 마스터에서 컨트롤하고 슬레이브는 이를 따르는 방식이다.

IIC 통신은 7비트 지원, 시리얼 통신, 양방향 데이터 전송이 가능하다. 통신 속도는 3가지 모드가 있으며 다음과 같다.^[5]

- Standard Mode : 100kbps
- Fast speed Mode : 400kbps
- High Speed Mode : 3.4Mbps

IIC 통신을 통하여 카메라 모듈 내부의 레지스터를 변경하여 카메라의 기능을 변경할 수 있다. MBA2440 보드에서 PO2030N 카메라 모듈과 통신하기 위해서는 그림 3과 같은 순서로 데이터를 보내고 받아야 한다.

3. 카메라 컨트롤러

S3C2440 CPU에는 카메라 컨트롤러가 내장되어 있으며 7개의 부분[T_pattern Mux, Capturing Unit, Preview Scaler, Codec Scaler, Preview DMA, Codec DMA, SFR]로 나누어져 있다. 입력은 최대 4096*4096

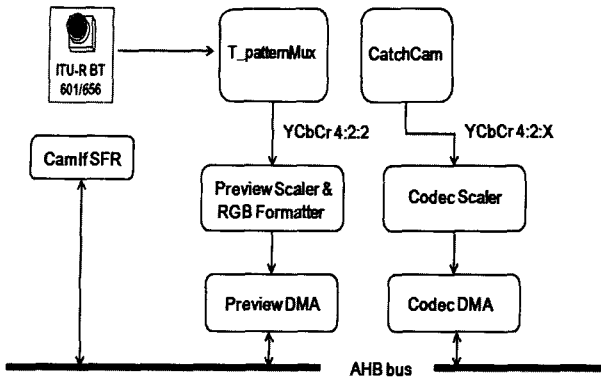


그림 4. 카메라 컨트롤러 블럭도
Fig. 4 Block Diagram of camera controller.

픽셀을 받을 수 있으며 2048*2048 픽셀 스케일링을 지원한다^[6]. 그림 4는 카메라 컨트롤러의 구성도이다. 데이터의 흐름을 보면 T_patternMux는 카메라 모듈로부터 오는 픽셀 클럭과 동기하여 영상 데이터를 받아서 CatchCam으로 넘기고, CatchCam은 받은 데이터를 Preview Scaler나 Codec Scaler로 보낸다. 각 Scaler는 이미지의 크기를 변환하고 각 DMA로 데이터를 보내게 된다. DMA는 받은 데이터를 시스템 메모리에 저장하고, CPU는 메모리에 저장된 데이터를 그래픽 장치나 저장장치로 보내게 된다.

카메라 컨트롤러는 프리뷰(Preview) 모드와 코덱 뷰(Coдек View) 모드를 지원하는데 프리뷰 모드는 특별한 처리 없이 바로 화면에 출력하기 위해서 사용하고, 코덱 뷰모드는 YCbCr로 받은 데이터를 중간에 CPU가 RGB로 변환하여 좀 더 나은 화질로 화면에 출력하기 위해 사용한다. 프리뷰 모드의 경우 오직 YCbCr 4:2:2로 데이터를 받을 수 있으며 하드웨어가 YCbCr을 RGB로 변환한다. 이와는 반대로 코덱 뷰는 YCbCr 4:2:x로 표본화 주파수 비율이 다른 방식을 받을 수 있다. 여기서 4:2:2는 아날로그 영상 정보를 디지털화 할 때 표본화 주파수 비율을 말하는 것으로 4:2:2의 경우 Y가 4번 샘플링 될 때 R-Y와 B-Y가 각각 2번씩 샘플링 됨을 의미한다. 코덱 뷰는 프리뷰와는 다르게 중간에 YCbCr을 RGB로 변환해 주어야 하는데, PO2030N 카메라 모듈의 경우 (1) 식으로부터 (2) 식의 RGB를 구한다.^[7]

$$\begin{aligned}
 Y &= 0.299R + 0.587G + 0.114B \\
 Cb &= 0.564(B - Y) + 128 \\
 Cr &= 0.713(R - Y) + 128
 \end{aligned}
 \tag{1}$$

$$\begin{aligned}
 R &= Y + (Cr - 128) / 0.713 \\
 B &= Y + (Cb - 128) / 0.564 \\
 G &= (Y - 0.299R - 0.114B) / 0.587
 \end{aligned}
 \tag{2}$$

4. 파일 시스템

저장 장치의 특징은 그 저장 장치를 관리하는 파일 시스템의 설계와 성능에 큰 영향을 준다. 플래시 메모리는 하드디스크와는 다른 특징을 가지고 있기 때문에 기존의 하드디스크의 기법을 그대로 사용하는 것은 비효율적이다.

플래시 메모리의 파일시스템을 생각할 때 두 가지 문제점을 고려해야 한다. 첫째로, 한번 쓴 데이터를 쓴 영역에 바로 다시 쓸 수 없다는 점이다. 새로운 데이터를 쓰기 전에 플래시 메모리를 지워야 하는데 플래시 메모리는 1 EU (64 KBytes~256KBytes) 가 삭제되는데 1초가 소요된다. 그래서 그 자리에 갱신을 하려고 하면 속도저하가 심하게 일어난다. 둘째로, 플래시 메모리는 지우는 횟수가 한정되어 있다는 점이다. 즉 영구사용이 가능하지 않기 때문에 지우는 작업을 최대한 억제하여 수명을 늘려야 한다. 그래서 한정된 지역을 자주 사용하는 하드 디스크의 파일 시스템과는 달리 플래시 메모리에 사용하는 파일 시스템은 달라야 한다.

이러한 플래시 메모리의 특성 때문에 기존에 개발된 플래시 메모리를 이용한 파일 시스템에서는 LFS(Log-Structured 파일 시스템)을 사용한다. LFS는 데이터가 저장된 블럭에 수정된 데이터를 기록하는 것이 아니라, 다른 빈 블럭에 수정된 데이터를 기록하는 방법이다.^[8-10]

본 논문에서는 이 LFS를 이용한 FAT 파일 시스템 기반의 플래시 파일 시스템을 구현하였다. 기존의 방식은 Inode를 사용하였는데, 소용량 저장 매체에서 FAT 파일 시스템은 Inode를 사용한 유닉스 파일 시스템보다 더 좋은 성능을 보인다. Inode 방식은 각 Inode 정보를 읽어야 파일 접근 가능하지만 링크드리스트(linked-list) 형태의 FAT 파일 시스템은 FAT만 버퍼 캐쉬에 저장되어 있으면 쉽게 접근을 할 수 있다. 소용량 저장

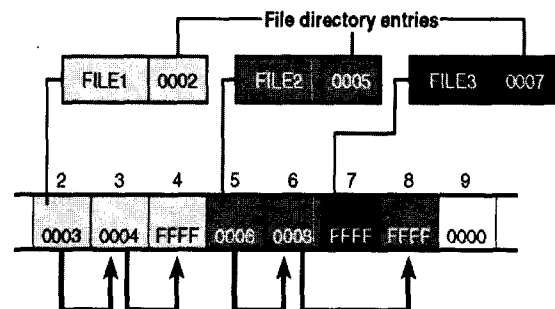


그림 5. FAT 파일 시스템의 구조
Fig. 5 Structure of FAT File System.

매체의 경우 FAT의 크기가 작아서 모든 FAT정보가 버퍼 캐쉬에 올라올 수 있다.^[11~12]

FAT 파일 시스템은 몇 개의 섹터로 구성되는 클러스터 단위로 저장 매체를 관리하는데, 이와 같은 정보가 부트 섹터에 저장된다. 부트섹터는 저장 매체의 맨 첫 번째 섹터에 위치한다. 부트섹터는 섹터당 바이트 수, 클러스터 당 섹터 수, FAT의 개수, 루트 디렉토리 내 엔트리 개수, 전체 섹터 수, FAT 당 섹터 수 등의 정보가 기록된다.^[13]

FAT는 부트 섹터 다음에 위치하게 된다. 엔트리의 크기가 12비트, 16비트, 32비트인지에 따라서 FAT12, FAT16, FAT32로 나뉜다.

그림 5에 FAT의 구조가 나타나 있다. 시작 클러스터가 0x002 라면, FILE1에 할당된 클러스터는 0x002, 0x004 가 된다. 이러한 구조는 소용량의 NAND 플래시 메모리에 적용하여 실험하는데 적당하다고 판단된다.

III. 펌웨어 상에서 동영상 획득

펌웨어 상에서 동영상을 획득하는 것은 레지스터를 설정하고 인터럽트를 설정해서 인터럽트 서비스 루틴만 수행하면 영상 데이터를 받을 수 있다.

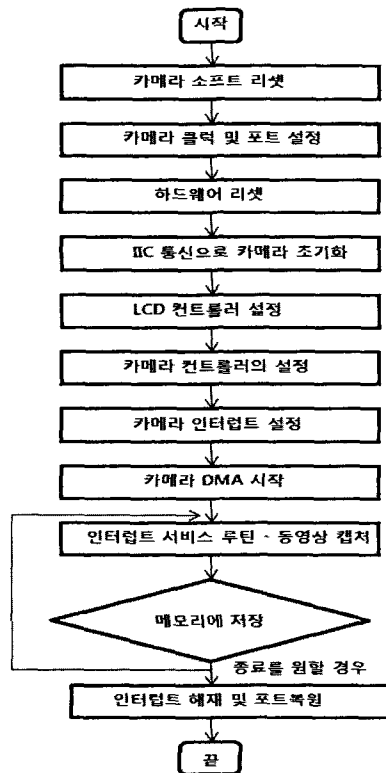


그림 6. 영상 획득 순서도
Fig. 6. Flow chart of image capture.

그림 6의 순서도는 카메라로부터 동영상을 출력하는 전체적인 순서를 나타낸다. 이 순서도는 후에 카메라 디바이스 드라이버를 만들 때 전체적으로 비슷하게 적용된다. 순서도에서 IIC통신을 통해서 카메라를 초기화하는 부분은 카메라의 영상데이터를 YCbCr로 보내도록 설정해야 한다. 주의할 점은 하드웨어 리셋신호를 만들 때 하강 엣지 신호를 만들어서 리셋을 해주어야 카메라와 IIC가 잘 동작하게 된다. 카메라 컨트롤러를 설정하는 부분은 전반적으로 영상의 스케일을 설정하고, DMA가 사용할 핑퐁 메모리를 설정하는 부분이다. 카메라 인터럽트는 프리뷰 인터럽트와 코덱 인터럽트로 나누어져 있으며, 동시에 수행 될 수 없다. 획득된 이미지는 Fat 파일 시스템을 이용하여 NAND 플래시 메모리에 이미지 형태로 저장이 되게 된다.

IV. 카메라 디바이스 드라이버 개발

카메라 디바이스 드라이버는 펌웨어와는 약간 다른 처리를 해주어야 하는데 이는 운영체제와 협력하여 작동해야 하기 때문이다.

1. Windows CE 디바이스 드라이버 구조

Windows CE 디바이스 드라이버는 드라이버의 로딩 시기에 따라 내장 드라이버(Built-in Driver)와 설치 가능 드라이버(Installable Driver)로 나눌 수 있고 계층 구조상으로 계층구조 드라이버(Layered Driver)와 단일 구조 드라이버(Monolithic Driver)로 구분 할 수 있다.

가. 내장 드라이버와 설치 가능 드라이버

내장 드라이버는 커널이 로딩될 때 메모리상에 올라가서 실행 할 수 있는 상태가 되어야 할 필요가 있는 경우에 해당하는 디바이스의 드라이버로, DLL처럼 독립적으로 존재하지 않고, GWES (Graphic Windowing Event Sub System)에 정적으로 링크되어 있다. 예를 들어 배터리 드라이버나 LED 드라이버를 들 수 있다.

반면 설치 가능 드라이버는 커널이 로딩된 후 필요에 의해서 메모리상에 올라가는 드라이버를 말한다. 이 드라이버는 플랫폼에 독립적으로 존재하며, 핫-플러그(hot-plug)가 가능한 장치가 추가될 때 디바이스 매니저에 의해 동적으로 로드된다. 설치 가능 드라이버는 PCMCIA.dll, Serial.dll, ATA.dll 등이 있다.

카메라 디바이스 드라이버는 내장 드라이버의 경우로 커널이 로딩될 때 메모리상에 올라가서 초기화를 한

번 수행하고 대기하게 된다.

나. 단일구조 드라이버와 계층구조 드라이버

디바이스 드라이버를 구조상으로 구분한다면, 계층구조 드라이버와 단일구조 드라이버로 나눌 수 있다. 계층구조 드라이버는 계층적 구조로 이루어져 있어 일반적인 기능을 하는 상위계층인 Model Device Driver (MDD)와 특별한 기능을 하는 하위 계층인 Playform Dependent Driver(PDD)로 나눌 수 있다. 단일구조 드라이버는 단독으로 실행하는 형태로 모든 드라이버에게 공통적으로 있는 코드를 포함하고 있다. PDD는 특정 하드웨어를 위한 코드로 구성되어 있다. 그래서 MDD는 PDD를 통하여 하드웨어에 접근한다. 즉 디바이스 드라이버를 제작할 때는 마이크로 소프트웨어에서 제공한 공통적인 MDD 코드를 사용하고, 시스템의 하드웨어에 따른 PDD를 새롭게 구성하여 사용하면 된다.

카메라 디바이스 드라이버의 경우는 단일구조 드라이버로 계층구조 드라이버의 경우 여러 단계를 거쳐야 하기 때문에 오버헤드가 있으나, 단일구조 드라이버는 상위 단계를 거치지 않고 직접 사용되어 빠른 처리를 할 수 있다.^[14]

2. 파일 구성

카메라 디바이스 드라이버는 카메라와 IIC 소스 파일과 윈도우의 라이브러리인 coredll.lib와 drvlib.lib파일을 참조해서 최종 camera.dll이라는 드라이버 파일을 만들게 된다. 이를 설정하기 위해서는 sources 파일을 만들어서 라이브러리 파일과 소스파일을 지정해 주어야 한다. 또 디바이스 드라이버 파일을 구성하기 위해서는 *.def, Makefile 파일이 필요한데, *.def파일은 외부에서 디바이스 드라이버 내부의 함수를 사용 할 수 있도록 하기위한 것이고, Makefile파일은 소스파일을 컴파일

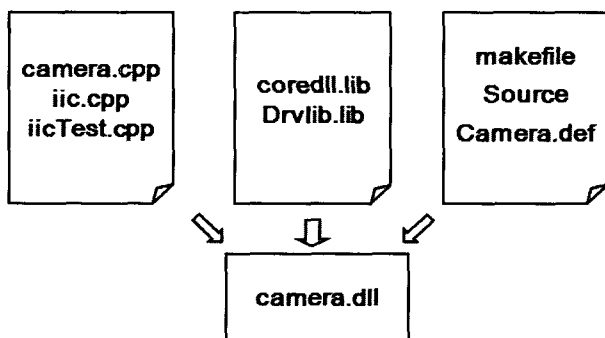


그림 7. Camera.dll 의 구성
Fig. 7. Construction of camera.dll.

할 때 설정을 정의한 파일로 상위에서 정의한 파일을 링크해서 사용한다.

3. 디바이스 드라이버 구현

카메라 디바이스 드라이버를 플랫폼에 추가하기 위해서는 WINCE 플랫폼 빌더의 카달로그 부분에 카메라 BSP를 추가하여야 하며, 추가된 카메라 BSP를 플랫폼에 추가해야 한다. 또한 레지스트 파일에 등록하기 위해 platform.reg, source, camera.def 파일에 관한 정보를 등록해야 한다.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\
Camera]
"Prefix"="CIS"
"Dll"="camera.dll"
"Index"=dword:1
"Order"=dword:1

LIBRARY CAMERA
EXPORTS
    CIS_Close
    CIS_Deinit
    CIS_Init
    CIS_IOControl
    CIS_Open
    CIS_PowerDown
    CIS_PowerUp
    CIS_Read
    CIS_Seek
    CIS_Write

TARGETNAME=camera
RELEASETYPE=PLATFORM
TARGETTYPE=DYNLINK
TARGETLIBS=\ $(COMMONSDKROOT)\lib\
b\$_CPUINDPATH)\coredll.lib
\ $_TARGETPLATROOT)\lib\$_CPUINDPAT
H)\drvlib.lib
DEFFILE=camera.def
DLENTY=DllEntry
INCLUDES=..\inc
INCLUDES=$(WINCEROOT)\public\common\
sdk\inc;$(INCLUDES)
INCLUDES=$(WINCEROOT)\public\common\
oak\inc;$(INCLUDES)
INCLUDES=$(WINCEROOT)\public\common\
ddk\inc;$(INCLUDES)
SOURCES= \
    llc.cpp \
    camera.cpp \
    iicTest.cpp \
```

그림 8. Camera.lib를 위한 레지스트 파일 작성
(a)Sources, (b) Platform.reg, Camera.def
Fig. 8. Coding regist file for camera.d;
(a)Sources, (b) Platform.reg, Camera.def

그림 8과 같이 설정한 후 플랫폼을 빌드하게 되면 최종적인 카메라 디바이스 드라이버가 포함된 WINDOWS CE 운영체제가 만들어지게 된다.

4. 디바이스 드라이버의 내부

처음 운영체제가 시작되면서 카메라 디바이스 드라이버는 디바이스 매니저에 의해 메모리로 올려지며, 초기화가 수행된다. 초기화 할 때 대부분은 펌웨어와 같다. 하지만 펌웨어와는 약간 다른 처리가 있는데, 펌웨어 상에서는 카메라의 레지스터 주소를 알면 직접 레지스터 주소를 사용했지만 드라이버의 경우 운영체제에 접근하는 주소를 할당 받아서 사용해야 한다. 또한 디스플레이 관련해서는 디스플레이 디바이스 드라이버가 초기화 설정을 함으로 특별히 디스플레이에 관련한 처리를 해줄 필요가 없다. 인터럽트의 경우 펌웨어와는 다르게 운영체제에서 만든 시스템 인터럽트를 사용해야 하며, 운영체제에 인터럽트를 요청하고 이벤트를 받아서 처리해야 한다. 이때 인터럽트는 카메라의 수직동기 (Vertical Sync) 신호에 의해 발생하며, 수직동기 신호는 영상 한 프레임이 언어질 때 발생한다.

카메라 인터럽트가 발생했을 때 인터럽트 서비스 루틴에서 하는 일은 프리뷰의 경우 DMA가 평풍 메모리에 저장한 한 프레임의 영상데이터를 그래픽 프레임 버퍼로 보내는 것이다. 코덱 뷰의 경우는 평풍 메모리의 YCbCr 데이터를 RGB로 변환한 후 그래픽 프레임 버퍼로 보내는 것이다.

그림 9은 이러한 과정을 거친 후 나온 결과이다.



그림 9. 응용프로그램의 실행
Fig. 9. execution of application program.

V. 결 론

지금까지 Windows CE용 픽셀플러스 사의 카메라 모듈인 PO2030N의 디바이스 드라이버를 구현하고 응용프로그램을 실행하여 영상의 획득 및 메모리에 저장하는 방법에 대하여 살펴보았다. 카메라 모듈과 통신을 하기 위해서 IIC통신을 사용 했으며 카메라 컨트롤러의 레지스터를 설정하고 인터럽트 처리를 하여 동영상 데이터를 프리뷰 방식과 코덱 뷰방식을 이용하여 화면에 표시하고 메모리에 저장을 하였다. 본 연구를 통하여 Windows CE의 실시간 운영체제로서의 성능을 일부 파악하여 볼 수 있었으며, 아울러 카메라 모듈의 구조 및 디바이스 드라이버의 구조를 확인 할 수 있었다. 모든 카메라 모듈의 전반적인 작동은 표준화가 되어 있는 관계로 다른 플랫폼 환경에서도 카메라 모듈을 구현하고 제어하는 것도 그리 어렵지 않을 것이다. 본 연구에서 개발된 화면 획득 및 저장 디바이스 드라이버는 범용 모바일 시스템의 기본적인 카메라 디바이스 드라이버로 사용하기에 충분할 것으로 사료된다. 향후 연구로는 현재 제작한 디바이스 드라이버에 사용된 NAND 플래시의 파일 시스템의 단점을 보완하기 위한 연구를 계속하여 메모리 관리 성능을 향상시킬 예정이다. 또한 MBA2440 보드에 이식한 Windows CE 코드들의 최적화를 통해 안정성 및 실시간 성능의 향상을 위한 연구도 진행할 예정이다.

참 고 문 헌

- [1] 김성훈, 임삼수, "Windows CE에 기반한 키패드 디바이스드라이버 개발", ICASE, 제 7권 제 1호, pp.227-231, 2004.
- [2] 안병찬, "Porting Windows CE Operating System to Arm based board device", ICCAS 2003, pp.2159-2163, 2003.
- [3] Douglas Boling, "Programming Windows CE Third Edition", Microsoft Press
- [4] PIXELPLUS, "PO2030N 1/4.5 Inch VGA Single Chip CMOS IMAGE SENSOR", 2005.
- [5] PHILIPS SEMICONDUCTORS, "The I2C-Bus Specification", 2000.
- [6] SAMSUNG Electronics, "S3C2440A 32-bit RISC Microprocessor User's Manual", 2004.
- [7] 최현석, "MS WINCE기반 VGA급 Camera Module Capture/Display Device Driver 개발", 대한임베디드공학회, pp 18-19, 2006.
- [8] Atsuo Kawaguchi, Shingo Nishioka and Hiroshi

Motoda, "A Flash-Memory Based File System", Proceedings of 1995 USENIX Technical Conference, pp.155-164, 1995.

[9] Chiang, M. -L, Lee, P. C. H and Chang, R. -C, "Managing Flash Memory in Personal Communication Devices", Proceedings of the IEEE International Symposium on Consumer Electronics, pp.177-182, 1997.

[10] M.Wu and W. Zwaenepoel, "eNVy : A Non Volatile, Main Memory Storage System", Proceeding of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems, pp.86-97, 1994.

[11] Michael Tischer and Bruno Jennrich, "PC INTERN : the Encyclopedia of System Programming", pp377-387, Abacus, 1996.

[12] Steven Holzner and The Peter Norton Computing Group, "Advanced Assembly Language", pp214-255, 영진출판사, 1993.

[13] 백승재, "플래시 메모리 파일 시스템을 위한 순수도 기반 페이지 할당 기법에 대한 연구", 정보처리학회 논문지 A 제 13-A권 제 5호, 2006.

[14] 임삼수, "Windows CE에 기반한 USB 호스트 관련 드라이버 구현", 전북대학교 석사 학위논문, pp 8-9, 2006.

저 자 소 개



김 승 환(학생회원)
 2007년 전북대학교 전자정보공학부 학사 졸업.
 2007년~현재 전북대학교 전자공학과 석사과정.
 <주관심분야: 마이크로프로세서, 시스템 프로그래밍, 디바이스 드라이버, Windows CE>



이 주 연(학생회원)
 2000년 원광대학교 전기공학부 학사 졸업.
 2006년~현재 전북대학교 전자공학과 석사과정
 <주관심분야: Arm Processor, 이산신호처리, 이미지 프로세서>



이 정 환(학생회원)
 2003년 충남대학교 전기공학과 학사졸업.
 2005년 충남대학교 전기공학과 제어 및 시스템 석사졸업.
 2006년~현재 전북대학교 전자공학과 박사과정.

<주관심분야: 무인항법제어, Embedded System>



함 운 철(정회원)
 1979년 서울대학교 전자공학과 학사 졸업
 1982년 서울대학교 전자공학과 석사 졸업
 1988년 서울대학교 전자공학과 박사 졸업

1982년~현재 전북대학교 전자정보공학부 교수
 <주관심분야: 적응제어, 로봇제어 >