

MS WINCE 기반 VGA급 Camera Module의 영상 캡처와 저장을 위한 Device Driver 개발

Development of Device Driver for saving and capturing an image by using VGA Camera Module based on WinCE

*김승환, **함운철, ***이정환, ****이주연

*Kim Seung-Hwan, **Ham Woon-chul, ***Lee Jung-Hwan, ****Lee Ju-yun

Abstract - 모바일 장치에서 사용하는 카메라 모듈을 가지고 MBA2440 보드에서 WINDOWS CE 운영체제의 디바이스 드라이버를 통하여 카메라 장치로부터 동영상 데이터를 얻어 화면에 표시하고, 여기서 획득된 화면정보를 Nand Flash 메모리에 저장하는 내용을 다룬다. 테스트에 사용된 카메라 모듈은 PIXELPLUS사의 모바일용 초소형 VGA급 30만 화소 카메라로 카메라모듈과 MBA2440보드 사이에 하드웨어적인 부분이 정상 작동하는지 확인을 위하여 펌웨어 상에서 카메라의 동영상 화면을 캡처하는 프로그램을 만들고, 이를 FAT File System을 이용하여 Nand Flash에 Image 파일 형태로 저장할 수 있도록 한다.

Key Words : IIC, Device Driver, Codec view, FAT

1. 서론

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. 자연히 사람들도 카메라에 관심을 많이 보이고 있으며, 카메라가 포함된 제품은 더욱 늘어날 것 보인다. 또한 카메라의 용도도 다양해질 것이다. 그래서 카메라를 제어할 수 있는 기술은 여러 분야에 필요할 것이다.

본 논문은 펌웨어 상에서의 카메라모듈을 작동시키고 이것을 기반으로 Window CE에서 디바이스 드라이버를 통한 카메라를 작동 시키고 Nand Flash에 저장하는 방법에 대해 전반적으로 다룰 것이다. 본 논문에 사용된 보드는 AIJI사의 MBA-2440보드로 삼성의 ARM계열 CPU인 S3C2440A를 사용하며, CPU에 카메라 컨트롤러를 내장하고 있다. 카메라 모듈은 PIXELPLUS사의 PO2030N으로 30만 화소에 CMOS타입으로 VGA급 (640*480)의 카메라이다.

순서는 우선 전반적인 카메라 동작과 각 세부 동작을 살펴보고, 펌웨어 상에서 동작하는 부분을 살펴보겠다. 다음으로 펌웨어 상에서 만든 프로그램을 Windows CE의 디바이스 드라이버로 구현하는 방법에 대해서 다루겠다.

2. 본론

2.1 카메라의 전반적인 작동

우선 카메라의 전체적인 작동은 다음과 같다. 보드에서 클럭을 카메라 모듈에 제공하면 카메라는 이 클럭을 가지고 컨트롤러를 작동 시켜 CMOS 센서로부터 화상 정보를 읽어 보드로 데이터를 보내게 된다. 보드의 카메라 컨트롤러는 카메라로부터 오는 데이터를 카메라의 Pixel Clock, Vertical Sync, Horizontal Sync 신호와 동기를 맞추어서 데이터를 읽어 이를 내부 메모리에 저장하게 된다. 그러면 CPU는 저장된 데이터를 그래픽 디스플레이 장치에 표시하거나, 저장 장치에 저장한다.

2.1.1 카메라 모듈의 세부적인 동작

카메라 내부는 <그림 1> 블록 다이어그램과 같이 구성되어 있으며 648*488 CMOS센서가 있고 여기서 얻은 영상 데이터를 타임 컨트롤러가 Image Signal Processing블록으로 보낸다. Image Signal Process블록은 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.

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.

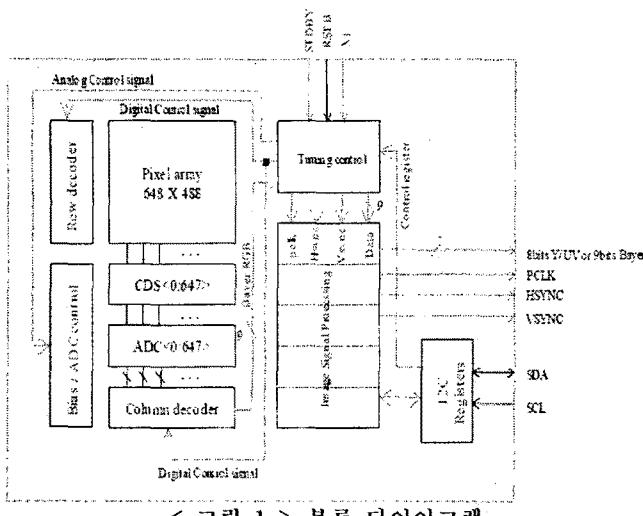
저자 소개

* 김승환: 全北大學 電子學科 碩士課程

** 함운철: 全北大學 電子學科 教授

*** 이정환: 全北大學 電子學科 博士課程

**** 이주연: 全北大學 電子學科 碩士課程



< 그림 1 > 블록 다이어그램

2.1.2 IIC통신

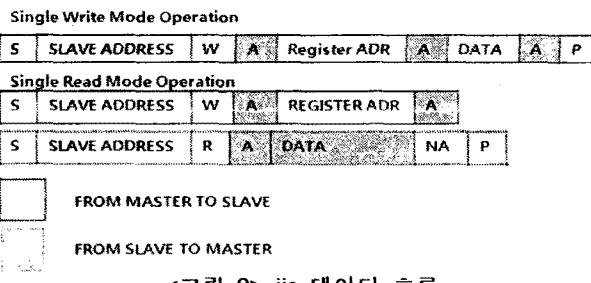
IIC통신은 최초 필립스가 제안한 통신 방식으로 기본 구성은 Master와 Slave로 구성되어 있으며 여러 Slave가 Master와 연결하여 통신을 할 수 있다. IIC통신은 클럭을 제공하는 SCL과 데이터 통신을 하는 SDA 두 선만 필요하므로 회로가 간단하다는 장점이 있으며, 클럭을 Master에서 공급하므로 동기를 맞추지 않아도 되는 장점이 있다. IIC통신으로 데이터를 주고받을 때는 마스터에서 컨트롤하고 Slave는 이를 따르는 방식이다. 통신 속도는 3가지 모드가 있으며 다음과 같다.

Standard Mode : 100kbps

Fast speed Mode : 400kbps

High Speed Mode : 3.4Mbps

우리는 IIC 통신을 통하여 카메라 모듈 내부의 레지스터를 변경함으로써 카메라의 기능을 변경할 수 있다. MBA-2440 보드에서 PO2030N 카메라 모듈과 통신하기 위해서는 < 그림 2 > 과 같은 순서로 데이터를 보내고 받아야 한다.



<그림 2> iic 데이터 흐름

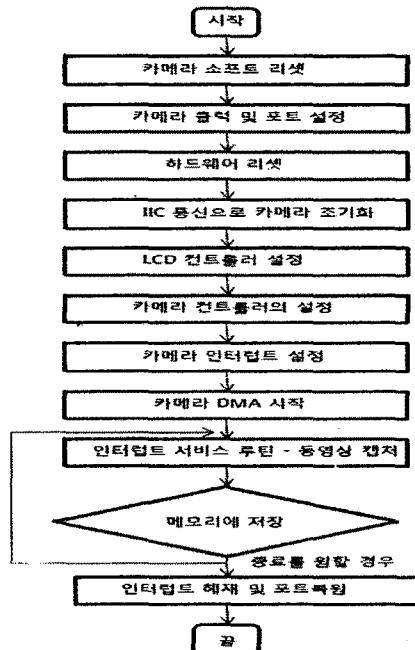
2.1.3 카메라 컨트롤러

S3C2440 CPU에는 카메라 컨트롤러가 내장되어 있으며 7개의 부분 - Pattern Mux, Capturing Unit, Preview Scaler, Codec Scaler, Preview DMA, Codec DMA, SFR - 로 나누어져 있다. 입력은 최대 4096*4096 픽셀을 받을 수 있으며 2048*2048 픽셀 스케일링을 지원한다. 데이터의 흐름을 보면 T_patternMux는 카메라 모듈로부터 오는 픽셀 클럭과 동기하여 영상 데이터를 받아서 CatchCam으로 넘기고, CatchCam은 받은 데이터를 Preview Scaler나 Codec Scaler

로 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.

2.2 펌웨어 상에서 동영상 캡처

펌웨어 상에서 동영상을 캡처하는 것은 레지스터를 설정하고 인터럽트를 설정해서 인터럽트 서비스 루틴만 수행하면 영상 데이터를 받을 수 있다. <그림 3>의 순서도는 카메라로 부터 동영상을 출력하는 전체적인 순서를 나타낸다. 이 순서도는 후에 카메라 디바이스 드라이버를 만들 때 전체적으로 비슷하게 적용된다. 순서도에서 IIC통신을 통해서 카메라를 초기화 하는 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 평평 메모리를 설정하는 부분이다. 카메라 인터럽트는 Preview 인터럽트와 Codec 인터럽트로 나누어져 있으며, 동시에 수행될 수 없다. 캡쳐된 이미지는 Fat file System을 이용하여 nand flash 메모리에 이미지 형태로 저장이 되게 된다.



<그림 3> 카메라 영상 캡처 순서도

2.2.1 Fat(File Allocation Table) File System

파일시스템은 데이터의 최소 단위인 파일이 하드디스크, 플래쉬 메모리등의 저장매체에 저장되는 방식을 말한다. 운영체제는 디스크 상에 파일들을 저장하는데 파일시스템은 여기에 필요한 규칙을 세우고 정한 규칙에 의해 저장하게 한다.

이 파일 시스템 중에서 FAT방식의 파일 시스템이란, 메모리의 한 부분에 FAT이라고 하는 하드디스크 안에 저장되어 있는 파일의 실제위치와 크기, 이름 등의 정보를 기록한 테이

블을 이용해서 파일을 호출하고 저장하는 방식이다. FAT16 방식과 이를 개선한 FAT32방식, FAT32방식과 함께 사용되며 긴 파일이름을 지원하는 VFAT (VIRTUAL FILE ALLOCATION TABLE)이란 파일시스템도 있다. 다음은 FAT 방식의 파일시스템의 구조이다.

< FAT 방식 파일시스템의 구조 >

- MBR (MASTER BOOT RECORD)
- BOOT RECORD
- FAT1 (실제파일의 위치정보)
- FAT2 (FAT1의 복사본)
- 루트 디렉토리
- 데이터 저장영역

단, FAT12방식을 사용하는 플로피디스크은 저장용량이 작기 때문에 하드디스크와는 달리 공간을 나눌 필요가 없으므로 MBR영역이 필요하지 않다.

2.3 카메라 디바이스 드라이버 개발

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

2.3.1 파일 구성

카메라 디바이스 드라이버는 카메라와 IIC 소스 파일과 원도우의 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 사용할 수 있도록 하기위한 것이고, Makefile파일은 소스파일을 컴파일 할 때 설정을 정의한 파일로 상위에서 정의한 파일을 링크해서 사용한다.

2.3.2 디바이스 드라이버 구현

카메라 디바이스 드라이버를 플랫폼에 추가하기 위해서는 WINCE 플랫폼 빌더의 카탈로그 부분에 카메라 BSP를 추가하여야 하며, 추가된 카메라 BSP를 플랫폼에 추가해야 한다. 또한 레지스트 파일에 등록하기 위해 platform.reg 파일에 다음과 같이 카메라를 등록해야 한다.

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

이렇게 설정한 후 플랫폼을 빌드하게 되면 최종적인 카메라 디바이스 드라이버가 포함된 WINDOWS CE 운영체제가 만들어지게 된다.

2.3.3 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 운영체제에게 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 Vertical Sync 신호에 의해 발생하며, Vertical Sync신호는 영상 한 프레임이 얻어질 때 발생한다.

카메라 인터럽트가 발생 됐을 때 인터럽트 서비스 루틴에서 하는 일은 Preview의 경우 DMA가 평통 메모리에 저장한 한 프레임의 영상데이터를 그래픽 프레임 버퍼로 보내는 것이다. Codecview의 경우는 평통 메모리의 YCbCr 데이터를 RGB로 변환한 후 그래픽 프레임 버퍼로 보내는 것이다.

3. 결론

지금까지 Windows CE용 PIXELPLUS 카메라 PO2030N의 디바이스 드라이버를 구현하고 메모리에 저장하는 방법에 대하여 살펴보았다. 카메라와 통신을 하기 위해서 IIC통신을 사용 했으며 카메라 컨트롤러의 레지스터를 설정하고 인터럽트 처리를 하여 동영상 데이터를 화면에 표시하고 메모리에 저장을 하였다. 모든 카메라의 전반적인 작동은 유사하기 때문에 다른 플랫폼 환경에서도 카메라를 제어하는 것은 그리 어렵지 않을 것이다. 현재 테스트용으로 만든 디바이스 드라이버는 운영체제와의 상호관계가 부족하여 이를 수정하면 기본적인 카메라 디바이스 드라이버로 사용하기에 충분할 것으로 보인다.

참 고 문 헌

- [1] Microsoft Windows Embedded, "Developing a Board Support Package for Microsoft Windows CE.NET", 2002.
- [2] 안병찬, "Porting Windows CE Operating System to Arm based board device" ICCAS, 2003.
- [3] 최현석, "MS WINCE기반 VGA급 CameraModule Capture/Display Device Driver 개발", 2006.
- [4] SAMSUNG Electronics, "S3C2440A 32-bit RISC Microprocessor User's Manual", 2004.
- [5] PIXELPLUS, "PO2030N 1/4.5 Inch VGA Single Chip CMOS IMAGE SENSOR", 2005.
- [6] 김상형, "Windows API 정복", 2001.