

논문 2007-44SD-5-5

Windows CE 5.0 기반의 DMB 디바이스 드라이버 설계 및 구현

(Design and Implementation of DMB Device Driver based on the
Windows CE 5.0)

박 광 회*, 김 덕 환**, 김 영 훈***, 장 준 혁**

(Kwanghee Park, Deok-Hwan Kim, Young-Hoon Kim, and Joon-Hyuk Chang)

요 약

최근 휴대용 멀티미디어 기기들의 수요가 증가하고 지상파 DMB 방송이 시작됨에 따라 핸드폰, 내비게이션, 멀티미디어 플레이어 등의 휴대 기기 통합에 대한 연구의 필요성이 높아지고 있다. 휴대기기의 통합을 위해서는 빠른 성능의 마이크로프로세서와 멀티미디어 서비스를 지원하는 다양한 장치 지원은 필수이다. 이를 위해 빠른 성능을 내는 ARM11 코어 계열의 임베디드 시스템 보드와 다양한 장치 및 어플리케이션을 지원하는 Windows CE 5.0 플랫폼을 구축했다. 그리고 DMB 디바이스 드라이버를 Busy Waiting과 Interrupt 의 두 가지 방식으로 구현하고 성능을 비교하여 임베디드 시스템의 자원을 효율적으로 사용하는 방법을 제안한다.

Abstract

Recently, as the demand of mobile multimedia devices increases and T-DMB is started in Korea, the need of research for integration of mobile devices such as cellular phone, navigation, and portable multimedia player becomes higher. In order to integrate mobile devices, it is necessary to support microprocessor with fast speed and various devices with multimedia service. In this paper, we construct Windows CE 5.0 platform whose BSP supports the embedded system board with ARM11 core and various devices and applications. We also implement the DMB device driver which supports busy waiting and interrupt driven I/O techniques, compare their performance, and then suggest the method to efficiently use the resources of embedded system.

Keywords : 지상파 DMB, 임베디드 시스템, Windows CE, 디바이스 드라이버

I. 서 론

DMB(Digital Multimedia Broadcasting) 기술은 방송국의 증가로 인한 주파수 자원의 고갈과 아날로그 FM(Frequency Modulation)방송의 혼선 및 이동 시 음

질저하 문제의 해결책으로 처음 생겨나게 되었다. DMB란 기존의 라디오방송에서 음성뿐만 아니라 문자·CD 수준의 음질·양방향성·다양한 데이터 서비스 등 멀티미디어 데이터를 서비스하는 차세대 라디오 방송을 말하는 것으로, DMB는 이동수신이 뛰어나기 때문에 소형 TV·PDA 등 휴대폰 단말기를 통한 전달이 용이하다.^[1] 지상파 DMB의 경우 이동형 단말기 수신을 목적으로 하기 때문에 채널 상에서 발생할 수 있는 오류 처리 및 안정적인 디바이스 드라이버의 필요성이 더욱 클 것으로 예상되며, 현재 세계 최초로 2005년 12월부터 지상파 DMB 본방송을 수도권 지역에 시작했다. DMB 방송이 본 궤도에 오르면, 양방향 서비스를 지원하는 자바 미들웨어 OCAP(Open Cable Application Platform),

* 학생회원, ** 정회원, 인하대학교 전자공학과
(Dept. of Electronic Engineering, Inha University)

*** 정회원, 충북대학교 컴퓨터공학과
(Dept. of Computer Engineering, Chungbuk National University)

※ 본 논문은 정보통신부 출연금으로 ETRI, SoC산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

접수일자: 2007년1월17일 , 수정완료일: 2007년3월19일

MHP(Multimedia Home Platform), ACAP(Advanced Common Application Platform) 등을 지원해야 하며, DMB의 표준 중의 하나인 TPEG(Transport Protocol Expert Group)을 통한 다양한 교통정보를 제공하기 위해서는 현재 내비게이션 시스템을 제공하는 플랫폼과 연동이 가능해야 한다.

또한 많은 종류의 데이터를 처리 및 가공하기 위해 다양한 응용 프로그램이 제공되고 있는 플랫폼을 고려해야 하며, 임베디드 시스템의 성능 역시 향상되어야 한다. 타깃 보드는 2006년 말 현재 존재하는 임베디드 시스템의 마이크로프로세서 중에 멀티미디어 서비스에 가장 적합한 ARM11 기반의 프리스케일사의 i.MX31 ADS(Application Development System) 보드를 사용하였으며, 마이크로소프트사의 Windows CE 5.0을 운영 체제 커널로 사용했다. DMB모듈로는 PNP Network사의 PN3030 모듈을 이용하고 타깃 보드와의 통신을 위해 SPI(Serial Peripheral Interface)를 사용했으며, 이외에도 인터럽트 처리 편을 사용하기 위해 GPIO를 이용했다.

본 논문에서는 임베디드 시스템을 설계 및 구현 시 발생하는 문제를 해결하는 방법과 DMB 디바이스 드라이버를 Busy Waiting 과 Interrupt 두 가지 방식으로 구현하여 임베디드 시스템의 자원을 효율적으로 사용할 수 있는 방법을 제안한다.

본 논문은 다음과 같이 구성된다. II장에서는 i.MX31 보드와 DMB 모듈에 사용할 하드웨어와 소프트웨어의 개괄적인 내용 그리고 문제 및 해결방법을 소개했다. III장에서는 디바이스 드라이버의 개발 방법에 따라 차이점을 나타내는지 검증하기 위해 Busy waiting 방식과 Interrupt 방식을 이용해 실험을 하고 결과를 논의한다. 마지막으로 IV장에서는 결론을 맺는다.

II. 본 론

1. i.MX31

i.MX31은 프리스케일 사에서 만든 멀티미디어 어플리케이션 프로세서로 ARM11 기반의 532MHz의 빠른 성능을 가진다. 또한 3D를 지원하며 뛰어난 전력 관리에 다양한 보안 기능까지 갖추고 있어 차세대 임베디드 마이크로프로세서로 각광 받고 있다. 본 논문에서 사용하고 있는 i.MX31 ADS 보드는 다양한 디바이스들을 지원하고 있어 차세대 모바일 제품을 설계하고 구현하기에 무리가 없다.

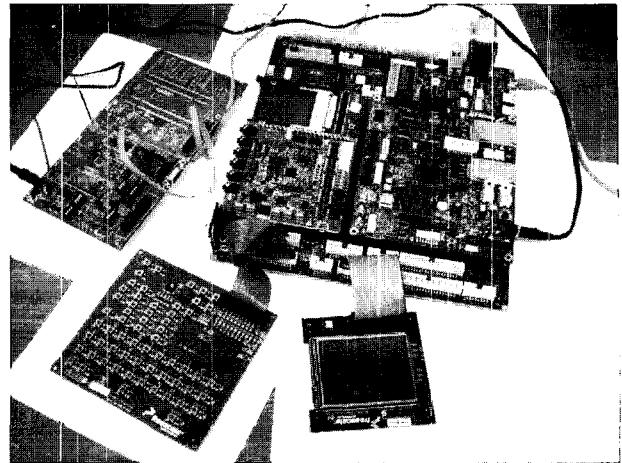


그림 1. i.MX31 ADS 보드와 DMB 모듈
Fig. 1. i.MX31 ADS board and DMB Module.

가. i.MX31 ADS 보드 구조

i.MX31 ADS보드는 CSPI, UART, USB 등과 같이 다양한 인터페이스를 지원하고 다양한 스토리지 미디어를 제공한다. 그림 1은 i.MX31 ADS 보드와 DMB 모듈, 키패드 및 LCD 모듈을 각각 보여준다. ADS 보드의 전체적인 구조는 그림 2와 같다. 이 많은 장치들을 모두 서술 할 수 없기 때문에 본 논문에서는 DMB 모듈과 통신할 인터페이스인 CSPI와 CSPI의 문제점을 해결하기 위해 사용한 GPIO를 위주로 설명한다.^[2]

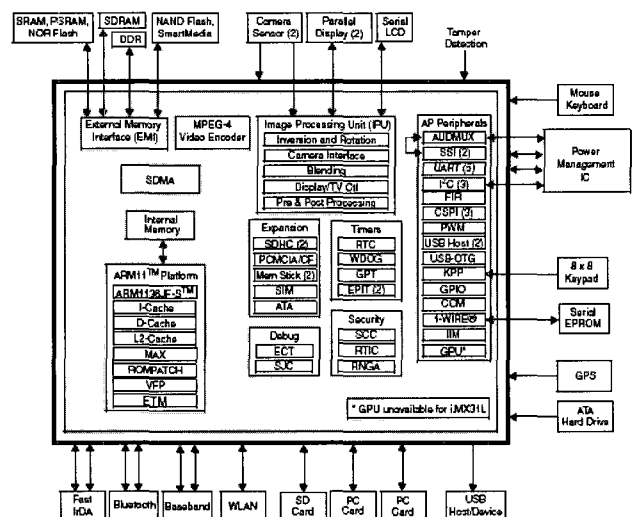


그림 2. i.MX31 인터페이스 블록 다이어그램
Fig. 2. i.MX31 Interface Block Diagram.

(1) CSPI

CSPI(Configurable Serial Peripheral Interface)는 설정가능한 시리얼 보조 장치 인터페이스로 기본적인 SPI

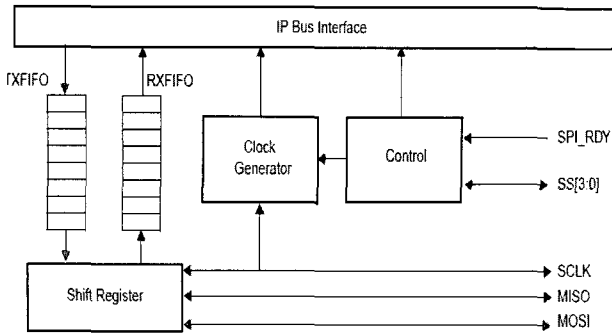


그림 3. CSPI 블록 다이어그램
Fig. 3. CSPI Block Diagram.

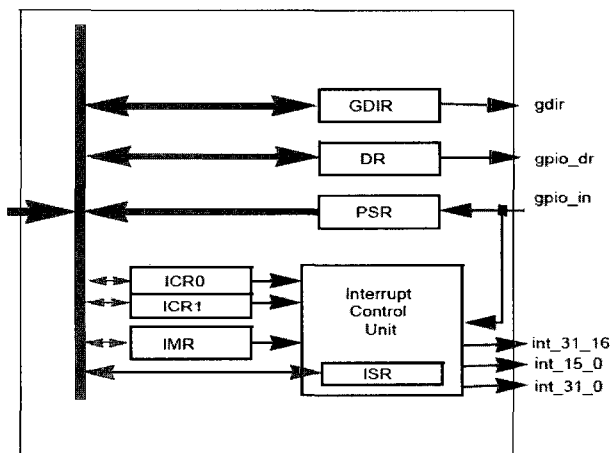


그림 4. GPIO 블록 다이어그램
Fig. 4. GPIO Block Diagram.

에 CS(Chip Select), MISO(Master Input Slave Output), MOSI(Master Output Slave Input), SCLK(Serial Clock) 이렇게 4개의 핀으로 구성 되어 있고 다양한 옵션을 설정 가능하도록 했다. 그림 3과 같이 CSPI의 특징은 타이밍, 비트 단위의 통신 등과 같이 다양한 설정이 가능하다.

본 논문에서는 CSPI 1번을 이용하여 DMB 모듈과 통신하고 TS(Transport Stream) 데이터를 받아 어플리케이션에 전달한다. 그림 3에서와 같이 CSPI 하드웨어 구성은 32bit RXFIFO, TXFIFO 레지스터가 각각 8개씩 16개로 구성되어 있으며 Shift Register를 통해 각 FIFO(First In First Out)에 순서대로 입/출력 된다.

(2) GPIO(General Purpose Input Output)

GPIO는 일반적으로 자주 쓰이는 입출력 핀으로 사용자가 필요로 하는 방식으로 인터페이스를 구성할 수 있어 임베디드 시스템 설계 및 구현 시 많이 사용 된다. 이번 실험에서는 CSPI의 CS(Chip Select)핀이 DMB 모듈과 통신 방식이 틀려서 GPIO3_1이 CS핀 출력 역할

을 대체하고 GPIO3_0이 인터럽트 처리 핀의 역할을 한다. GPIO는 다양한 방식으로 이용되므로, 그림 4와 같이 제어를 위한 다양한 레지스터를 제공한다.

2. 지상파 DMB 시스템 및 DMB 모듈

가. 지상파 DMB 시스템

지상파DMB 시스템에서는 MPEG-4 AVC(Advanced Video Coding), BSAC(Bit-sliced Arithmetic Coding) 및 BIFS(Binary Format for Scenes)을 각각 비디오, 오디오 그리고 데이터의 소스를 코딩하고 MPEG-2 TS 패킷으로 다중화 한다.

지상파DMB 시스템에서는 이렇게 다중화 된 결과의 이동 수신 성능을 증가시키기 위해 추가적인 채널 코딩 방법으로 RS(Reed-Solomon Coding)와 컨볼루션 바이트 인터리빙(Convolutional byte interleaving)기법을 사용하고 채널코딩 결과가 Eureka-147 DAB의 스트림 모드를 통해 전송된다. 그림 5는 DAB Eureka-147의 스트림 구조를 나타낸다.^{[4][7][8]}

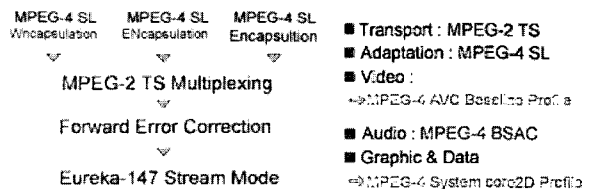


그림 5. DAB Eureka-147 스트림 구조
Fig. 5. DAB Eureka-147 Stream Structure.

나. DMB 모듈

DMB 모듈은 PNP Network사의 PN3030 DMB 칩을 사용하고 튜너는 인티그런트사의 ITD3010을 사용 한다. 하지만 PN3030은 I²C를 통해 튜너를 제어하기 때문에 본 논문에서 만드는 디바이스 드라이버는 PN3030 DMB 칩을 통하여 튜너 즉 송신 단을 포함한 전체 DMB 모듈을 제어한다. PN3030은 튜너를 통해 데이터를 송신하면, 그림 6과 같이 Data Decoder를 통해 디코딩 한 후 데이터 형식 별로 나누어 놓은 메모리 영역에 저장한다. DMB 모듈의 순서는 앞서 내려오는 FIC(Fast Information Channel)를 파싱하여 채널정보를 확인한 후 채널을 선정하여 그 채널에 맞는 데이터만 MSC(Main Service Channel) 메모리에 쌓도록 설정하고 메모리가 임계 치에 도달한 경우 인터럽트와 레지스터 플래그를 이용하여 디바이스 드라이버에 알리고 데

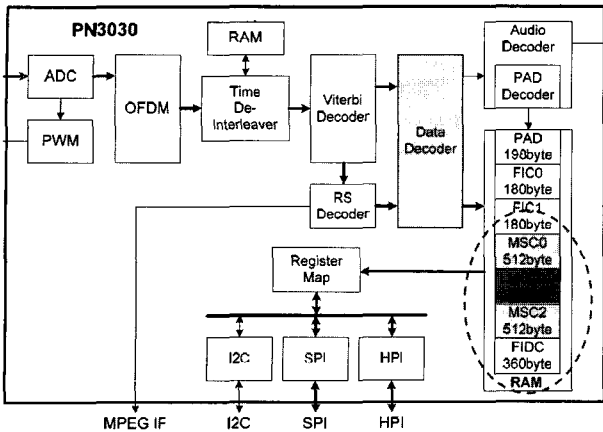


그림 6. PN3030의 블록 다이어그램
Fig. 6. PN3030 Block Diagram.

이터를 메모리로부터 SPI를 사용하여 전송하는 방식으로 처리한다.^[5]

다. DMB 디바이스 드라이버

DMB 디바이스 드라이버의 전체구조는 그림 7과 같다. DMB 모듈의 RF Chip과 DMB Chip은 I²C로 내부 통신하며, DMB 모듈과 i.MX31 ADS보드는 SPI를 이용하여 통신하며 데이터를 주고받는다.

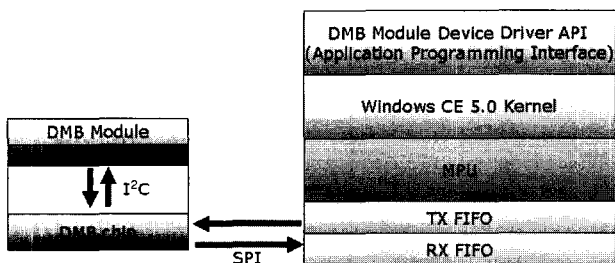


그림 7. DMB Device Driver Overview
Fig. 7. DMB 디바이스 드라이버 전체보기.

3. Windows CE 5.0 BSP
(Board Support Package)

가. CSPI 문제 해결

본 논문에서는 CSPI를 사용하기 위해 두 가지 문제가 발생 했다. 첫째 CSPI가 프리스케일 사에서 제공하는 BSP에서 정상 동작 하지 않으므로 전체 소스 분석을 통해 코드의 오류를 찾아야 한다는 점, 둘째 DMB 모듈의 SPI 인터페이스와 동일하게 작동해야하기 때문에 CSPI를 SPI와 동일하게 최적화를 해야 한다는 점이다.

(1) CSPI 활성화

CSPI는 기본적으로 프리스케일 사에서 제공하는 BSP에 디바이스 드라이버를 포함한다. 하지만 CSPI를 제어하는 레지스터의 값이 정상적으로 변경되지 않음을 발견하고 BSP의 소스를 분석한 결과, 그 원인은 CSPI 레지스터의 클록게이트를 제어하는 함수가 소스 상에서 활성화 되어 있지 않기 때문이며, 그 결과 제어 레지스터 및 상태 레지스터의 데이터 입/출력이 불가능 했다. 이 문제를 해결 한 뒤 CSPI의 4개 핀의 입출력을 테스트한 결과 정상 동작했다.^[3]

(2) CSPI 최적화

프리스케일 사의 CSPI는 하드웨어적으로 버스트 모드를 최대 32byte까지 밖에 지원 하지 않는다. 그림 8을 참고하면 CSPI의 CS핀이 32바이트를 처리하는 동안 레벨 상태는 Low를 유지 하지만 매 32바이트 전송할 때 마다 그림 8 CSPI SPI(上)에서와 같이 CS핀의 레벨 상태가 Low에서 High로 올라가는 것을 알 수 있다. DMB 모듈은 SPI가 요구되는 크기의 버스트 모드 일 시, CS핀의 레벨이 전송이 종료 시까지 떨어지지 않아야 하므로 CSPI의 통신 방식을 수정을 해야 할 필요성이 존재한다. 그러므로 GPIO핀을 CS핀 대신 연결하여 32바이트 이상 처리 시에도 레벨 상태를 Low로 유지하게 함으로써 대용량 데이터 전송의 문제점을 해결 했다. 자세한 해결 방법은 다음 장에서 GPIO를 통해 CSPI의 CS핀을 처리하는 방법을 설명 한다.

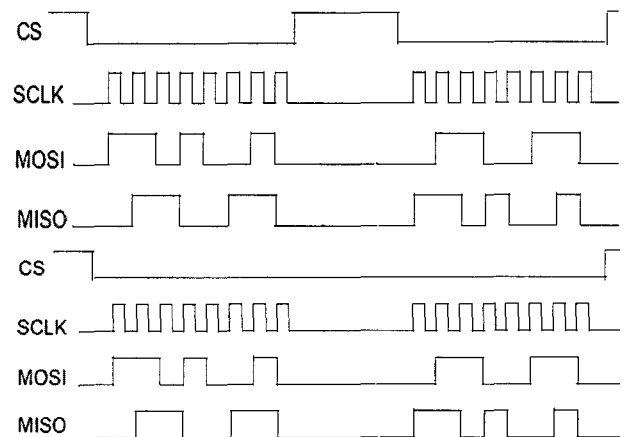


그림 8. CSPI(上)와 DMB 모듈 SPI(下)
Fig. 8. CSPI(Top) and DMB Module SPI(Bottom).

나. GPIO를 이용한 CS와 인터럽트 처리 핀

(1) GPIO를 이용한 CS핀 설정

GPIO는 다양한 입/출력을 인터페이스에 따라 설정 가능하므로 사용자가 다양한 방식으로 사용이 가능하다. 이를 통해 CSPI 소스 중 버스트 모드로 동작하는 메인 루프 안에 GPIO3_1을 제어하는 코드를 삽입했다. DMB 모듈에 연결해야 하는 CS핀을 GPIO3_1과 연결함으로써, DMB모듈에서는 CS핀이 메인 루프 안에서는 레벨 상태가 항상 Low로 떨어지게 설정했다.

(2) GPIO를 이용한 인터럽트 처리 핀 설정

모든 통신이 정상적으로 동작하고 실제 DMB 모듈의 MSC1 메모리의 임계 치에 도달 시, 들어오는 신호를 받기 위해 GPIO3_0을 이용하여 확인하는 IST(Interrupt Service Thread)를 구현했다.

위의 두 가지 문제를 해결 후 디바이스 드라이버와 T-DMB(Terrestrial-DMB) 테스트 MPEG4 플레이어와 통신을 하기 위해 API(Application Programming Interface)를 설계 및 구현하고 연동은 DLL(Dynamic Linked Library)방식으로 제공한다.

III. 실험 및 결과 논의

DMB 모듈과의 통신을 통해 FIC(Fast Information Channel) 데이터를 수집하고 현재 수신 가능한 서브 채널 리스트 중 하나를 선택하여 TS(Transport Stream) MPEG4 Player에 실시간으로 TS 데이터를 넘겨줌으로써, 끊김 없이 화면과 음성이 동작하는지를 실험했다.

이와 더불어 임베디드 시스템에서 적은 자원을 가지고 움직이는 환경을 비교 평가하기 위해 DMB 모듈과의 통신 방법을 Busy Waiting과 Interrupt 두 가지 방식으로 설계 및 구현하여 실제적으로 임베디드 시스템에서 Busy Waiting 방식과 Interrupt의 차이점을 설명한다.

1. 실험 구성 방안

가. 실험 시스템 구성

실험을 위한 하드웨어 체계는 지상파가 수신이 잘 될 수 있는 탁 트인 공간과 수신율이 좋은 고감도 안테나를 설치하고 본 논문에서 설계 및 구현한 구조를 통해 DMB 모듈 칩에서부터 TS를 버퍼링하여, MPEG4 동영상 플레이어에 전달해주며 실제로 LCD 화면에 정상적으로 화상과 음성이 끊임없이 동작하는지를 테스트한다.

나. 실험 방법 및 내용

실험 방법은 원활하고 빠른 실험을 위해 각각의 방식 별로 DLL(Dynamic Linked Library)을 만들고 Busy Waiting 방식으로 구현한 DLL을 DLL A, 그리고 Interrupt 방식으로 구현한 DLL을 DLL B로 지칭하고 이 DLL을 사용해 DMB 테스트 플레이어를 통해 결과를 테스트 한다.

Busy Waiting 방식인 DLL A는 DMB 모듈의 MSC 메모리에 TS 데이터를 쌓으면서 메모리가 임계 치에 도달했는지를 알려주는 DMB 모듈의 레지스터 값을 무한 루프를 돌면서 계속 확인한다. 임계 치에 도달 시에는 DMB 모듈에 명령을 내려 디바이스 드라이버의 수신 버퍼로 데이터를 읽어 온다. 만약 없을 시에는 일정 지연 시간 후에 다시 확인을 하는 방식으로 동작한다.

Interrupt 방식인 DLL B는 IST(Interrupt Service Thread)라는 Windows CE 5.0에서 지원하는 인터럽트 방식을 사용한다. 하드웨어에서 인터럽트 발생 시 운영 체제에서는 인터럽트 이벤트를 발생시키고 대기상태에 있던 높은 우선순위를 가지고 있는 대기 쓰레드가 이벤트 수신하는 즉시 MSC 메모리의 데이터를 디바이스 드라이버의 수신 버퍼에 읽어온다. 그림 9, 10은 i.MX31 보드의 LCD 화면을 통해 MBC 방송이 끊임없이 재생되는 것을 보여준다.

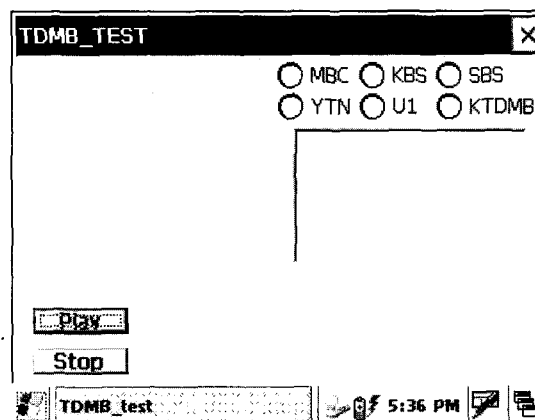


그림 9. 지상파 DMB 플레이어 사용자 인터페이스
Fig. 9. User Interface of T-DMB player.

다. 실험 결과

그림 11, 12는 Busy Waiting과 Interrupt 방식의 CPU의 이용률에 대한 백분율 그래프이다. 이 그래프는 마이크로소프트사에서 제공하는 Platform Builder 5.0의 도구 중 하나인 Remote Performance Monitor 프로그램을 통해 얻은 결과이다. 그리고 표 1은 그래프에서



그림 10. 지상파 DMB 실시간 화면
Fig. 10. Realtime screens of T-DMB.

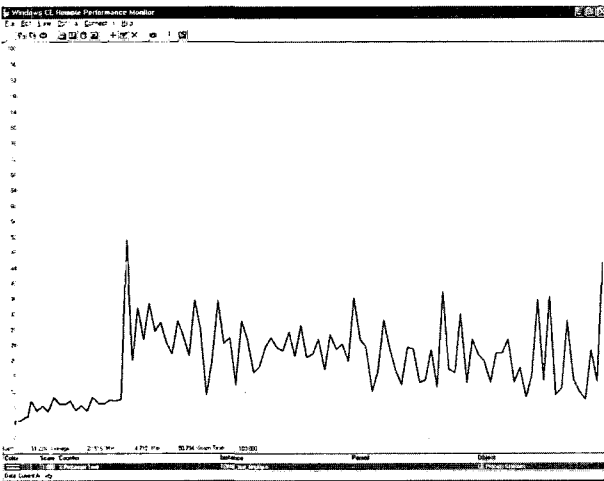


그림 11. Interrupt 방식의 CPU 이용률
Fig. 11. CPU utilization of interrupt driven method.

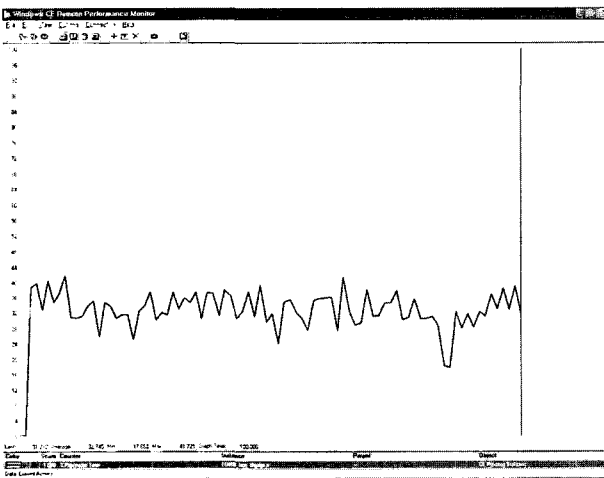


그림 12. Busy Waiting 방식의 CPU 이용률
Fig. 12. CPU utilization of busy waiting.

CPU 이용률 중 높은 점들을 순서적으로 나열 한 것이며 표 2는 제안된 두 가지 방식에 대하여 전체 그래프의 평균, 최소, 최대 이용률과 표1에서 구했던 이용률의

표 1. 시간 순으로 나타낸 8개의 CPU 이용률 값
Table 1. CPU utilization values according to chronological order.

Busy Waiting	40	41	42	39	40	41	42	40
Interrupt	33	34	35	36	36	38	29	40

표 2. CPU 이용률의 통계값
Table 2. Statistical values of CPU utilization.

	평균	최소	최대	표1 평균
Busy Waiting	32.745	17.668	41.721	40.625
Interrupt	20.516	4.710	50.794	35.125

평균을 각각 비교한 것이다.

그림 11, 12, 표 1, 2에서와 같이 Busy Waiting 방식은 커널이 할당한 자신의 쿼텀을 다 소비할 때까지 DMB 모듈의 상태를 확인하기 때문에 CPU 이용률이 높고, Interrupt 방식은 DMB 모듈의 메모리가 임계치에 도달하기 전까지는 CPU를 사용하지 않다가 도달하면 CPU를 소비하므로 CPU 이용률이 낮다. 결과적으로 임베디드 환경을 고려하고 DMB 모듈과 같은 실시간 스트림 디바이스 드라이버 설계 및 구현 시에는 CPU 이용률이 적은 Interrupt 방식으로 접근하는 것이 좋다. 하지만 Interrupt 방식은 인터럽트 벡터의 개수가 제한되어 있어 항상 쓸 수 없는 제약 조건이 있다. 그러므로 중요도가 낮은 루틴에서는 Sleep과 같이 CPU 이용률을 줄이는 기능을 포함한 Busy Waiting 방식을 쓰는 것도 한 가지 방법이 된다.

IV. 결 론

본 논문에서는 임베디드용 ARM 11과 풍부한 어플리케이션을 지원하는 Windows CE 5.0으로 CSPI를 이용한 DMB 디바이스 드라이버를 설계 및 구현했다. 이번에 사용된 SPI는 산업용으로 널리 쓰이지만 표준이 정립되지 않아서 각 하드웨어마다 차이가 존재하므로 SPI의 표준 정립이 시급하다고 여겨진다. 이번 설계 및 구현에서도 CSPI의 CS핀의 재설계를 할 수 밖에 없었던 것도 이와 같은 이유이다. 또한 지상파 DMB 디바이스 드라이버를 설계 및 구현함으로써 DMB의 구조와 지상파 DMB가 어떻게 서비스 되는지를 알 수 있었으며 실용을 통하여 Busy Waiting 방식과 Interrupt 방식의 비교를 통해 장·단점을 파악 할 수 있었다. 지상파 DMB 모듈이 실시간 스트림 장치이고 많은 데이터가 오는 것을

감안하여 CPU의 이용률을 줄일 수 있는 Interrupt 방식을 사용하는 것이 바람직하다는 것을 알 수 있었다.

향후 각각의 휴대용 기기들이 융합 되면 많은 통신 모듈과 다양한 장치들이 통합되면 자원의 공유와 시스템 성능향상이 중요한 연구주제가 될 것이며, 통신시장이 현재 음성과 데이터, 유·무선 통합, 통신·방송 융합의 패러다임으로 바뀌는 추세에서 지속적인 연구가 필요할 것이다.

참 고 문 헌

[1] 김규현, 기명석, “지상파 이동멀티미디어방송 (DMB-T) 기술개발 동향” 전자공학회지, 제31권, 제5호, pp551-557, 2004년 5월

[2] “i.MX31 Applications Processor Reference Manual”, Freescale Semiconductor, 2006.
 [3] “i.MX31ADS BSP Reference Guide”, Freescale Semiconductor, 2006.
 [4] “EN 300 401 Radio Broadcasting System; Digital Audio Broadcasting(DAB) to mobile, portable and fixed receivers,” ETSI, Aug, 2000.
 [5] “PN3030 Application Notes”, PNP Network
 [6] 이봉성, 류명희 “Windows CE 실전 가이드”, 에이콘, 2006.
 [7] 한국전파진흥협회, “지상파 DMB 시스템” <http://www.t-dmb.org>
 [8] 신필섭, 김정우, 박상오, 김재룡, “지상파 DMB 수신 기술” 전자공학회지, 제32권 제5호 pp570-579, 2005년 5월

— 저 자 소 개 —



박 광 희(학생회원)제1 저자
 2005년 용인대학교 컴퓨터정보
 처리학과 학사 졸업.
 2007년 현재 인하대학교
 전자공학과 석사 2년차
 2004년 인프니스 기술연구소
 연구원

2004년~2005년 인테그라 정보통신 부설연구소
 연구원
 <주관심분야 : 임베디드 시스템, 플래시메모리 스토리지 시스템, 방송통신서비스>



김 덕 환(정회원)-교신저자
 1987년 서울대학교 계산통계학과
 학사 졸업
 1995년 한국과학기술원 정보통신
 공학과 석사 졸업
 2003년 한국과학기술원 정보통신
 공학과 박사 졸업

1987년~1997년 LG전자 통신기기연구소
 선임연구원
 1997년~2006년 동양공업전문대학 전산경영기술
 공학부 부교수
 2004년 University of Arizona, Tucson
 박사후연구원
 2006년~현재 인하대학교 전자전기공학부 부교수
 <주관심분야 : 임베디드 시스템, 시스템소프트웨어, 멀티미디어, 패턴인식, 데이터마이닝>



김 영 훈(정회원)
 1994년 충북대학교 컴퓨터공학과
 학사 졸업
 1997년 충북대학교 컴퓨터공학과
 석사 졸업
 2005년 충북대학교 컴퓨터공학과
 박사 수료

1997년~2005년 LG전자 선임연구원
 2005년~2006년 레인콤 차장
 2005년~현재 동양공업전문대학 겸임전임강사
 2007년~현재 LG텔레콤 차장
 <주관심분야 : 한국어처리, 자연언어처리, 자연언어검색>



장 준 혁(정회원)
 1998년 경북대학교 전자공학과
 학사 졸업
 2000년 서울대학교 전기공학부
 석사 졸업
 2004년 서울대학교 전기컴퓨터
 공학부 박사 졸업

2000년~2005년 넷더스 연구소장
 2004년~2005년 University of California, Santa Barbara 박사후연구원
 2005년 한국과학기술연구원 연구원
 2005년~현재 인하대학교 전자전기공학부
 전임강사
 <주관심분야 : 디지털신호처리, 음성신호처리, 스피치 코딩>