

PC-based DAB 수신기용 USB Interface Bridge 설계 및 구현*

정회원 박노경*, 진현준*, 박상봉**, 김상복*, 한성호*, 이상철*

Design and Implementation of USB Interface Bridge for PC-based DAB Receiver

Nho-Kyung Park*, Hyun-Joon Jin*, Sang-Pong Park**, Sang-Pok Kim*,
Sung-Ho Han*, Sang-Chul Lee*

요약

일반적으로 DAB 시스템은 크게 stand-alone 방식과 PC/PDA 기반 방식으로 구분할 수 있다. PC/PDA 기반 방식은 stand-alone 방식에 비하여 휴대성이 떨어지나 연결된 PC/PDA의 메모리와 오디오/비디오 디코더 등 다양한 리소스를 이용할 수 있는 장점이 있다. 본 논문에서 구현한 DAB 수신기는 USB 인터페이스를 이용한 PC 기반의 수신기 시스템으로 USB 인터페이스 브리지는 FPGA와 EZ-USB 개발 키트를 이용하여 설계하였으며 실험을 위하여 기존의 stand-alone 방식의 DRK-026 DAB 수신기를 이용하여 PC 기반의 수신기를 구현하였다. USB 인터페이스 브리지는 시리얼 데이터를 USB 패킷 형태로 변환시켜주며 모든 신호는 하드웨어적으로 제어한다. 구현한 PC 기반 DAB 수신기의 동작 여부는 DAB 수신기로부터 오디오 데이터를 수신하여 USB 인터페이스를 통하여 PC에 전송한 후 디코딩 기능을 수행하는 것으로 확인하였다.

Key Words : DAB, EZ-USB, USB Interface, Stand-alone, PC/PDA-based

ABSTRACT

Generally, DAB systems are divided into two categories, a stand-alone type and a PC/PDA-based type. The PC/PDA-based type has less mobility comparing to the stand-alone type, nevertheless, it has the advantage of using memory, audio/video decoder, or other resources of PC/PDA. The DAB receiver implemented in this paper is a PC-based receiver system employing USB interface. The USB interface bridge is designed using FPGA and EZ-USB development kit and the implemented DAB receiver adopts the bridge and makes use of the stand-alone typed DRK-026 receiver for experiments. The USB interface bridge transforms serial data into USB packets and all of related signals are controlled by hardware logics. The operation of the implemented DAB receiver is verified by sending audio data into the PC for decoding through USB interface bridge.

I. 서론

21세기 정보화 사회로 접어들면서 고품질의 다양

한 디지털 콘텐츠를 즐기고자 하는 소비자의 욕구가 늘어나고 있다. 이동 통신, 위성 통신, 초고속 인터넷 망 등의 다양한 유·무선 통신 시스템들은 디지털화

* 정회원, 호서대학교 정보통신공학과(Dept. of Information & Communication Eng. Hoseo Uni.)

** 정회원, 세명대학교 정보통신학과

논문번호: KICS2004-10-213, 접수일자: 2004년 10월 5일

* 본 연구는 2004년 호서대학교 학술연구조성비에 의해 연구되었음.

에 성공하여 광대역 멀티미디어 서비스를 제공하고 있으나, 텔레비전과 라디오 등의 방송 통신 분야는 디지털 시스템으로의 전환이 상대적으로 늦어진 상태이다^[1]. 이러한 시점에서 CD 수준의 고음질 오디오 서비스, 교통, 일기예보 등 다양한 부가 데이터 서비스는 물론, 7" 미만 화면에서는 고속 이동 중에도 선명한 화질의 영상정보 수신이 가능한 차세대 디지털 오디오 방송인 DAB (Digital Audio Broadcasting)가 부각되고 있다^{[2][3]}.

DAB 수신기는 자동차용 수신기와 Hi-Fi용, Portable 용 형태의 stand-alone 방식과 PC나 PDA 등에 장착할 수 있는 PC/PDA-based 방식으로 구분할 수 있다. 이중 PC-based 형태는 PC의 메모리와 오디오/비디오 디코더 등 다양한 리소스를 이용할 수 있는 장점을 가지고 있다^[4].

II장에서는 Cypress 사의 EZ-USB 개발 키트와 FPGA를 이용하여 USB 인터페이스 브릿지를 설계하였으며, III장에서는 II장에서 설계한 USB 인터페이스를 Atmel 사의 DRK-026 개발 키트에 접목하여 DAB 수신기를 구현하였다. IV 장에서는 구현한 시스템의 동작 여부를 측정하였으며, DRK-026 개발키트를 통하여 수신된 오디오 데이터를 USB 인터페이스를 통하여 PC에 전송한 후 오디오 디코딩 기능을 수행하는 것으로 확인 하였다.

II. PC-based DAB 수신기용 USB 인터페이스 브릿지 설계

2.1 Eureka-147

본 논문에서 구현할 DAB 시스템은 Eureka-147 방식으로 ITU-R의 권고안 가운데 Digital System A로 분류되어 있다. Eureka-147은 약 2MHz의 대역폭을 사용하며, CD 수준의 고음질 오디오 서비스가 가능하도록 MPEG Audio Layer II에 기반한 오디오 압축 기술을 사용한다^[3]. Eureka-147 시스템의 전송모드는 4가지로 다음 표 1과 같다.

현재 우리나라에서의 지상파 DAB 시스템은 전송 모드 I 방식을 위성 DAB 시스템은 전송모드 IV 방식으로 결정하였으며, 본 논문에서 구현한 DAB 시스템은 전송모드 I 방식의 지상파 DAB 시스템을 모델로 잡았다. Eureka-147 모드 I의 전송 프레임 구조는 그림 1과 같다.

2.2 DAB 수신기 구조

DAB 수신기는 Independent Portable Receiver 로서

표 1. Eureka-147 전송 모드

Table 1. Eureka-147 Transmission Mode

Parameters	Transmission Mode			
	I	II	III	IV
슬롯길이(Ts)	1.246ms	312us	156us	624us
보호길이(Tg)	246us	62us	31us	124us
심볼길이(Tu)	1ms	250us	125us </td <td>500us</td>	500us
Null 심볼길이	1.297ms	324us	168us	848us
프레임길이(TF)	96ms	24ms	24ms	48ms
프레임당 심볼수(L)	76	76	153	76
심볼당 방송파수	1536	384	192	768
방송과간격(f _d)	1 KHz	4 KHz	8 KHz	2 KHz
변조 방식	p/4 -DQPSK			
채널부호	Convolutional 1/4 Coding Rate, K=7, Variable Puncturing			
인터리빙	Timing/Frequency Interleaving			
대역폭(BB)	1.536MHz			
최대주파수	≤375MHz	≤1.5GHz	≤3GHz	≤1.5GHz

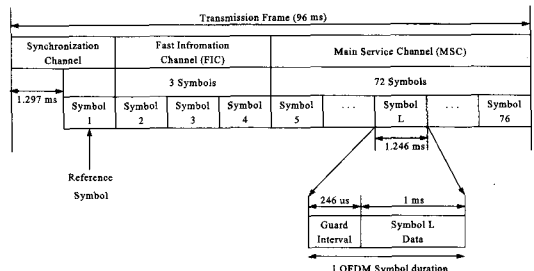


그림 1. Eureka-147 모드 I의 전송 프레임 구조

Fig 1. Frame structure of Transmission Mode I Eureka-147

Stand-alone 방식과 다른 기기와의 통합을 통한 PC/PDA-based 방식을 이용해서 구현할 수 있다.

2.2.1 Stand-alone 방식의 DAB 수신기

Stand-alone 방식의 DAB 수신기는 크게 RF Tuner, Baseband DAB Chip, D/A Converter로 나눌 수 있다. Stand-alone 방식의 전체 블록도는 아래의 그림 2와 같다^[4].

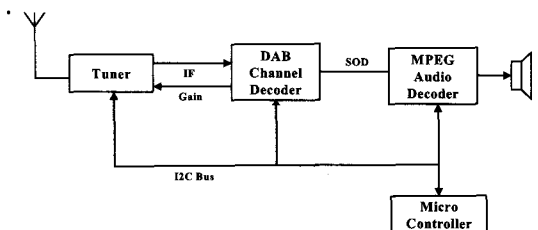


그림 2. Stand-alone 방식의 DAB 수신기

Fig 2. Stand-alone DAB Receiver

1) RF Tuner

Tuner의 목적은 선택된 DAB 신호를 IF(Intermediate Frequency)로 Down-Converting 하여 채널을 선택하는데 있다. 전송모드 I의 주파수 대역은 기존의 TV 채널과 겹치기 때문에 영상 잡음을 제거하기 위하여 식(1)에 의하여 IF를 38.912MHz로 결정하게 되었다.

$$f_{imag} = 174 + 2 \times f_{IF} = 250MHz \quad (1)$$

Gain Control은 ADC 출력에서 높은 신호대 잡음비를 얻는데 사용된다.

2) Baseband DAB Chipset

DAB Chip은 크게 Channel Decoder, Source Decoder의 기능을 수행하게 된다. Channel Decoder는 전송되어진 FIC와 MSC로 이루어진 DAB 신호를 OFDM Demodulation, Viterbi Decoding 하는데 목적이 있으며, Source Decoder는 Audio Decoder와 Data Decoder로 구성되어 음원과 데이터를 복원하는 기능을 수행한다. Audio Decoder는 ISO MPEG-1과 MPEG-2를 지원하며, Data Decoder는 두개의 독립적인 Packet Mode Decoder를 지원하게 된다.

3) DAC(Digital Analog Converter)

DAC는 Source Decoder에서 나오는 Digital Signal을 Analog Signal로 변환하여 스피커로 출력을 하며, 이는 Stand-alone 방식의 DAB 수신기에서만 사용이 된다.

2.2.2 PC-based 방식의 DAB 수신기 설계

PC에 기반을 둔 DAB 수신기는 PCI와 USB 인터페이스를 이용 할 수 있는데 사용자의 편의성을 고려하여 그림 3과 같이 USB 인터페이스를 이용한 PC-based DAB 수신기를 구현하였다.

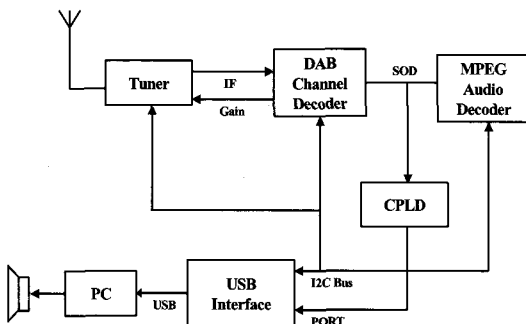


그림 3. PC-based 방식의 DAB 수신기
Fig 3. PC-based DAB Receiver.

PC-based DAB 수신기는 Stand-alone 방식과 비교

할 때 PC의 메모리와 오디오/비디오 디코더 등 다양한 리소스를 이용 할 수 있는 장점이 있다⁶⁾. 즉, Stand-alone 방식의 DAB 수신기에 비하여 보다 저렴하고, 소형화 할 수 있다. 또한, 수신된 Digital 신호를 PC의 고성능 CPU를 통한 데이터 처리와 대용량 하드 디스크에 방송중인 Audio 파일을 저장 할 수 있으며 인터넷과 연결되어 풍부한 데이터 서비스의 개발이 가능하다⁸⁾. 즉, 기존에 PCI 버스를 이용해서 브릿지를 구현하였으나, 본 논문에서는 USB 브릿지를 이용하여, 구현한 USB 브릿지는 서비스될 DMB 서비스에서도 충분한 데이터 전송대역폭을 가지고 있어, 어느 PC에나 손쉽게 연결 할 수 있는 장점을 가지고 있다. 따라서 앞으로 서비스될 DMB 대용량 정보량도 처리 할 수 있다.

2.3 USB 인터페이스 브릿지 설계

PC-based DAB 수신기는 PC와 DAB 수신기 사이의 통신을 위한 USB 인터페이스가 필요하며, USB 인터페이스는 신뢰성을 가지고 고속으로 데이터를 전송하는 4가지 방식으로 구별되어 진다. 우선순위는 낮지만 오류발생시 재전송을 할 수 있는 BULK 방식, 일정 주기에 일정량의 데이터를 전송할 때 사용되는 Isochronous 방식, 주기적으로 소량의 데이터를 처리할 때 사용하는 Interrupt 방식, 그리고 디바이스가 설정 정보를 호스트에 전송하는 Control 방식이 있다¹⁵⁾⁶⁾.

이 중에서 DAB 수신기를 통하여 수신된 실시간 데이터를 PC에 전송하기 위해서 USB 인터페이스 전송 방식 중에서 Isochronous 방식을 선택하였다. 아래 그림 4는 설계한 USB 인터페이스 브릿지를 나타낸 것이다.

DAB 수신기로 사용될 Atmel사의 DRK-026 개발 키트는 Stand-alone 방식만 지원 하기 때문에 Channel

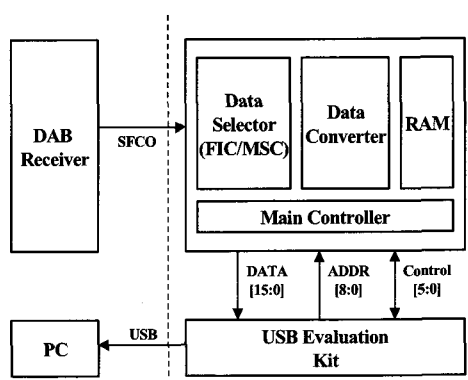


그림 4. 설계한 USB 인터페이스 브릿지 구조
Fig. 4. Structure of Designed USB Interface Bridge.

Decoder의 Output 인터페이스중 SFCO(Simple Full Capacity Output)를 이용하여 DAB 수신기에서 Channel Decoding된 데이터를 얻었다.

SFCO는 3.072MHz의 Burst 모드 형태의 인터페이스로 아래 그림 5는 SFCO에서 출력되는 5bit로 구성된 데이터 형태도이다^[7]. 이 데이터는 FPGA 모듈을 통하여 FIC와 MSC로 나누어 저서 EZ-USB 개발 키트를 통하여 Isochronous 방식으로 PC에 전송을 하게 된다.

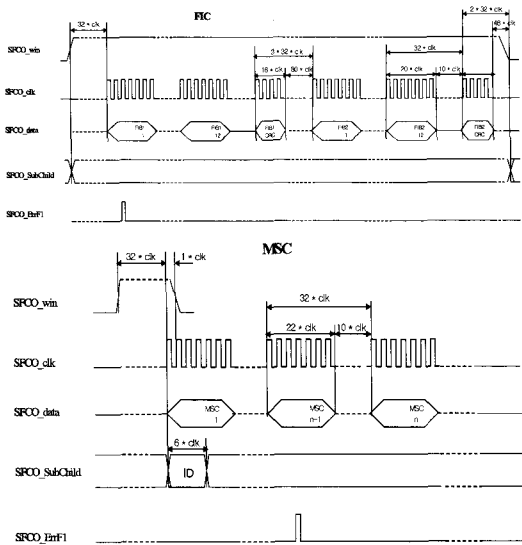


그림 5. SFCO 인터페이스의 타이밍도
Fig. 5. SFCO Interface Timing Diagram

그림 6는 EZ-USB 개발 키트에서 발생하는 이벤트의 상태를 나타낸 것이다. IDLE 상태에서 FPGA 모듈에 데이터가 수신되면 State1에서 FPGA 모듈의 RAM Address 정보를 읽어 들이고, State2는 FPGA 모듈의 Data 존재 여부를 검사한다. State3 에서는 FPGA 모듈의 데이터를 PC에 전송하며, State4에서는 데이터 전송 후 다음 상태의 동작 여부를 결정한다.

III. PC-based DAB 수신기용 USB 인터페이스 브릿지 구현

3.1 USB 인터페이스 브릿지 구현

USB 인터페이스 브릿지 구현은 EZ-USB 개발 키트와 FPGA 모듈로 구성되어 있다. DRK-026 개발 키트의 SFCO에서 출력되는 데이터는 Serial 형태의 데이터이므로, EZ-USB에서 PC로 데이터를 전송하기 위해 USB Packet 형태의 데이터로 변환해야 한다. 이러한 데이터 변환 작업은 구현된 FPGA 모듈을 통하

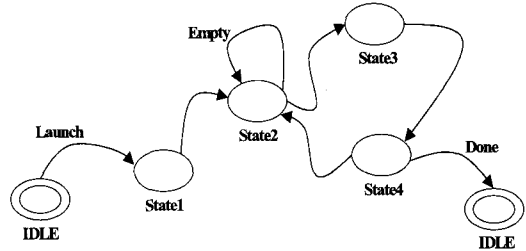


그림 6. EZ-USB 개발 키트의 상태도
Fig. 6. State Machine of EZ-USB.

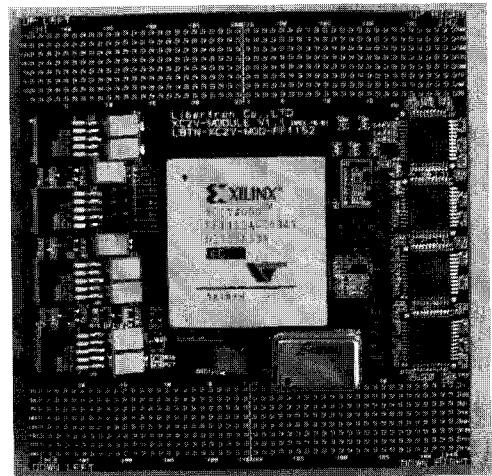


그림 7. FPGA로 구현된 타겟 보드
Fig. 7. FPGA Target Board.

여 이루어지고 변환된 데이터는 EZ-USB 개발 키트로 전송되어진다.

그림 7은 FPGA로 구현된 타겟보드로서 VHDL을 이용하여 하향식(Top-Down) 방식으로 설계하였다. 합성 및 P&R(Plase and Rounting) 과정은 Synplify 툴과 Xilinx ISE 툴을 이용하였고 검증과정은 Modelsim 툴을 이용하였다. FPGA 모듈은 EZ-USB 개발 키트와 GPIF(General Programmable Interface)방식으로 31bit의 I/O 핀으로 연결하였다.

그림 8은 EZ-USB 개발 키트와 FPGA 모듈의 GPIF를 나타내고 있다. GPIF는 EZ-USB에서 제공되는 외부의 ASIC, DSP, 다른 기타 로직을 직접 연결하는 인터페이스 방법이다. IFCLK는 인터페이스 동기화 클럭으로 40MHz가 사용이 되었으며, Data bus(FD [15:0]), Address Line(GPIFADR[8:0]), Control Line(Port I/O, EN[1:0], GSTATE(1:0))을 각각 연결하였다. 내부의 레지스터 설정과 Firmware를 통하여 FPGA 모듈과 GPIF 방식으로 연결한 EZ-USB 개발키트를 Single-Read 트랜잭션 방식으로 제어 하였다.

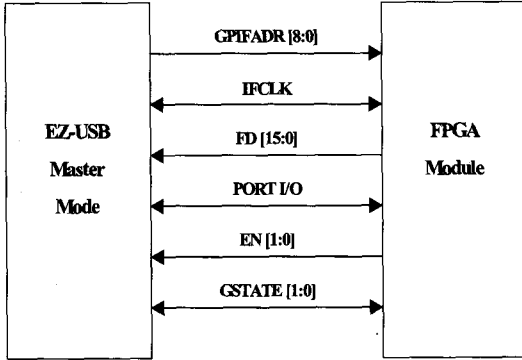


그림 8. EZ-USB 개발 키트와 FPGA 모듈의 인터페이스
Fig 8. EZ-USB Interface to FPGA Module.

그림 9는 Single-Read 트랜잭션의 처리 순서를 나타내고 있다.

1. GPIF Configuration 레지스터와 웨이브폼 기술자 초기화
2. Single-Read 트랜잭션을 시작하기 위해 GPIFS GLDATLX 레지스터의 더미 읽기 수행
3. GPIF가 트랜잭션이 완료될때까지 대기. 트랜잭션이 종료되면, DONE 비트(GPIFIDL-ECS.7 or GPIFTRIG.7)을 1로 설정. 만약 활성화되었다면, GPIFDONE 인터럽트 생성
4. 버스의 전송폭과 다음 트랜잭션이 대기하고 있다면, XGP-IFSGLDATH, XGPIFSGLD-ATLX 그리고 XGPIFSGLD-DATLNOX 레지스터로부터 읽은 데이터 추출

그림 9. Single-Read 트랜잭션 순서
Fig 9. State of Single-Read Transaction

그림 10은 Single-Read 트랜잭션 초기화 함수로 GPIF 프레임워크 기반의 8051 Firmware이다.

3.2 PC-based DAB 수신기 구현

PC에서 USB 인터페이스를 통하여 입력되는 데이터는 6개의 S/W 모듈을 통하여 재생된다.

```
#define PERIPHCS 0x00AB
#define AOKAY 0x80
#define BURSTMODE 0x0000
#define TRISTATE 0xFFFF
#define EVER ;;

//GPIF 초기화 함수 원형 선언
void Gpiflnit( void );
```

```
//GPIFADR[8:0]의 주소값을 DAB 수신기에 설정
void Peripheral_SetAddress(WORD gaddr)
{
    if( gaddr < 512 )
    { // GPIF 주소 버스 설정
        GPIFADRH = gaddr >> 8;
        SYNCDELAY;
        GPIFADRL = (BYTE)gaddr; //GPIF 주소
    } else
    { // 3상 GPIFADR[8:0] 핀 설정
        PORTCCFG = 0x00; // 포트 입출력을 위해
        [7:0] 설정
        OEC = 0x00; // 입력을 위해 설정
        PORTECFG &= 0x7F; // 포트 입출력을위해
        [8] 설정
        OEC &= 0x7F; // 입력으로 설정
    } //End of if
} //End of Peripheral_SetAddress

//GPIF를 이용하여 DAB 수신기로 부터 한개 바이트 읽기
void Peripheral_SingleByteRead( BYTE xdata *gdata )
{
    static BYTE g_data = 0x00;
    while( !( GPIFTRIG & 0x80 ) ) {} // GPIFTRIG.7
    Done 비트 poll
    //XDATA 공간의 레지스터를 이용하여 더미 리드
    g_data = XGPIFSGLDATLX; // GPIF 한개
    바이트 트랜잭션 트리거
    while( !( GPIFTRIG & 0x80 ) ) {} // GPIFTRIG.7
    Done 비트 poll
    // using register(s) in XDATA space, GPIF read
    byte from PERIPHERAL here
    *gdata = XGPIFSGLDATLNOX;
} //End of Peripheral_SingleByteRead
```

그림 10. Single-Read 트랜잭션 초기화 함수
Fig 10. Initial fuctions of Single-Read transactions.

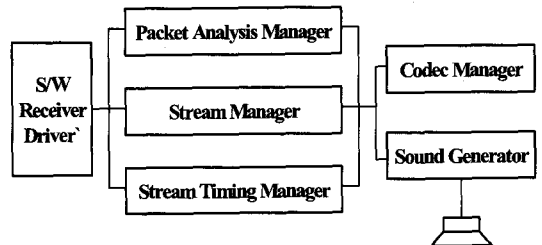


그림 11. PC 소프트웨어 구성도
Fig 11. Block diagram of PC softwares.

그림 11은 PC 소프트웨어 구성도를 나타낸 것인데 S/W Receiver Driver는 EZ-USB 개발 키트로부터 스트리밍 데이터를 ISO 모드로 수신한다. Packet Analysis Manager는 수신된 패킷의 포맷과 형식을 구분하기 위한 객체이다. Stream Manager는 ISO Stream

연결을 관리 및 지능형 버퍼링 기능을 제공하며, Stream Timing Manager는 제한된 시간 이내에 도착하지 않은 데이터를 폐기한다. Codec Manager는 Packet Analysis Manager에서 분석 및 해석된 데이터를 적합한 코덱으로 링크하기 위해 사용된다. Sound Generator는 수신된 데이터와 Codec Manager에서 링크된 Codec을 이용하여 사운드 재생 및 각종 사용자 인터페이스 기능을 제공하게 된다.

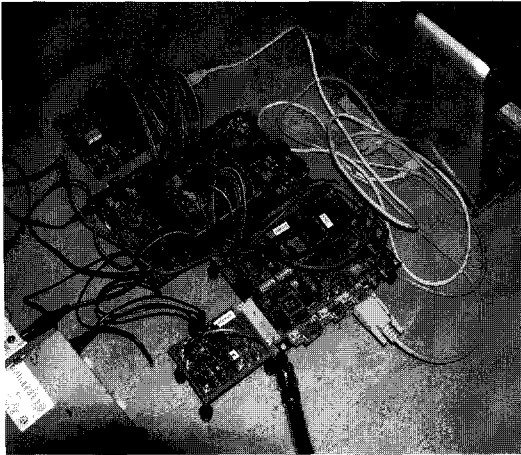


그림 12. 구현된 PC-based DAB 수신기
Fig 12. Implemented PC-based DAB Receiver.

그림 12은 구현된 PC-based DAB 수신기로 DRK-026 개발 키트로 구현을 하였으며, 수신기의 컨트롤은 DRK-026 개발 키트의 특성상 I2C 인터페이스를 이용하여 PC에서 직접 제어 한다.

DAB 수신기의 USB 인터페이스는 FPAG 모듈과 USB 개발 키트로 구성하였다. PC에서는 DAB 수신 신호를 디코딩 및 확인하기 위하여 아래의 그림 13과 같이 USB 인터페이스로부터 ISO 패킷 Stream을 읽는다.

IV. 실험 결과

설계된 PC-based DAB 수신기의 동작여부 확인은 DRK-026 개발 키트에서 SFCO를 통하여 전송된 데이터가 FPGA 모듈에서 Packet 형태로 변환이 되어 EZ-USB의 USB 인터페이스를 통하여 PC에 전송이 되고, 전송된 데이터는 PC에서 디코딩 기능을 수행하는 것으로 확인 하였다. 구현된 FPGA 모듈의 동작 파형은 아래 그림 14, 15와 같다.

```

while (!StopTest) //스레드 정지 플래그 활성화될때까지 반복
{
    //ISO 패킷 수신
    bResult = DeviceIoControl (hDevice, IOCTL_EZUSB_READ_ISO_BUFFER, &IsoControl, sizeof(ISO_TRANSFER_CONTROL), buffer, bytesToRead, &nBytes, NULL);

    :
    packetsRead = nBytes / (IsoControl.PacketSize + sizeof(USBD_ISO_PACKET_DESCRIPTOR));
    ptr = buffer;

    // ISO Packet Descriptor 획득
    isoDesc = (PUSBD_ISO_PACKET_DESCRIPTOR), (buffer + (packetsRead * IsoControl.PacketSize));

    //수신된 패킷수만큼 Read
    for (i=0;i<packetsRead;i++)
    {
        framesRead++;
        :
        if (count != *ptr) //카운터의 순번이 틀리면 프레임 Skip
        {
            framesMissed++;
            SetDlgItemInt(hDlg, IDC_FRAMES_MISSED, framesMissed, FALSE);
            sprintf(tempbuff, "Missed frame = 0x%x", count);
            SendMessage (hOutputBox, LB_ADDSTRING, 0, (LPARAM) tempbuff);

            DumpBuffer(ptr, isoDesc[i].Length, hOutputBox);
            count = *ptr;
        }
        :
        count++;
        if (isoDesc[i].Length != IsoControl.PacketSize) //패킷 길이 검사
        {
            framesMissed++;
            SetDlgItemInt(hDlg, IDC_FRAMES_MISSED, framesMissed, FALSE);
            :
        }
        ptr += IsoControl.PacketSize;
    }
}

CloseHandle (hDevice); //드라이버 Close
free(buffer); //버퍼 메모리 해제
:

```

그림 13. ISO 패킷 Stream 소스
Fig 13. Source of ISO packet stream.

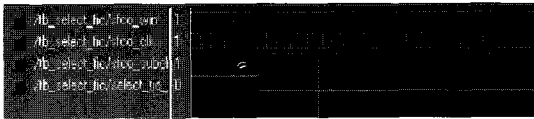


그림 14. FIC와 MSC 신호를 분류
Fig 14. Selection of FIC and MSC.

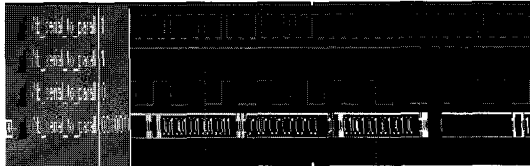


그림 15. 데이터 Packet화 된 출력신호
Fig 15. Output Signal of Packeted datas.

그림 14의 파형에서는 DAB 수신기에서 SFCO를 통해 FPGA 모듈에 입력이 되는 데이터를 FIC와 MSC로 구분해 주고 있는 것을 확인 할 수 있다. 그림 15의 파형에서는 EZ-USB로 전송 되어질 FIC와 MSC 신호가 16bit 데이터로 Packet화 되어 있는 것을 나타낸다.

그림 16은 PC에서 구현된 시스템의 제어 프로그램으로 구현된 USB 인터페이스를 통하여 PC에 전송되는 신호를 Stream 형태로 읽고 분석하여 디코딩 기능을 수행하는 것을 확인 할 수 있다. PC에서 수신된 데이터는 자체 제작한 ISO Test 프로그램을 통하여 확인 하였고, 구현한 DAB Player는 재생, 멈춤, 녹음의 세 가지 동작이 수행되는 것을 확인 하였다. ISO 테스트 프로그램은 ISO 패킷 송수신 상태를 테스트 한다. 드라이버를 통해 USB 인터페이스로부터 ISO 패킷 스트림을 읽고 수신된 누적 패킷의 수와 버퍼에 누적된 수신량을 비교하여 테스트 결과를 그림 16과 같이 화면에 모니터링을 해준다.

V. 결론

본 논문에서는 PC-based DAB 수신기용 USB 인터페이스 브릿지를 설계하고 구현하였다. 또한 구현된 브릿지를 적용한 PC-based DAB 수신기를 구현 하였고, DAB 수신기로부터 수신된 데이터를 USB 인터페이스를 통하여 PC에 전송, 디코딩 기능이 수행되는 것을 확인 하였다. 구현된 PC-Based DAB 수신기는 Stand-alone 방식의 DAB 수신기와 비교하여 PC의 메모리와 오디오비디오 디코더 등 다양한 리소스를 이용할 수 있으며, 인터넷서비스와 연동되어 풍부한

데이터 서비스 및 부가 서비스의 개발이 가능하다는 장점이 있다. 향후 연구 과제로 현재 국내의 DAB 시스템은 영상 서비스가 강화된 DMB (Digital Media Broadcasting) 시스템으로 상용화될 예정이기 때문에 PC-based DMB 수신기용 USB 인터페이스를 설계 및 구현 하는 것이다.

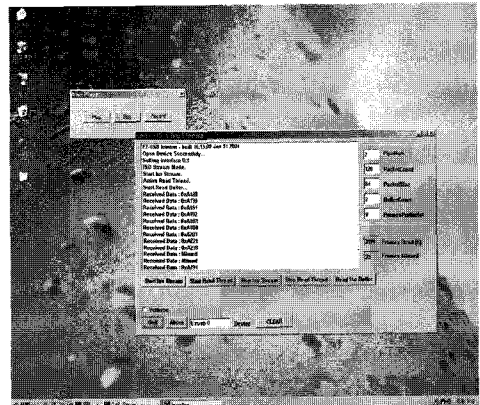


그림 16. 구현된 소프트웨어 (ISO Test and DAB Player)
Fig 16. Implemented Software (ISO Test and DAB Player).

참고 문헌

- [1] Walter Tuttlebee, "Enabling technologies of Software Defined Radio," John Wiley&Sons, LTD, 2002.
- [2] Walter Tuttlebee, "Origins, Drivers and International Perspectives of Software Defined Radio," John Wiley&Sons, LTD, 2002.
- [3] Wolfgang Hoeg and Thomas Lauterbach, "Digital Audio Broadcasting," John Wiley & Sons, LTD, 2001.
- [4] Darran Nathan, Bernhard Spath, Oliver Faust and Chua Beng Koon, "Design and features of an intelligent PC-Based DAB receiver," *IEEE Transactions on Consumer Electronics*, vol.48, No 2. p.322-328, 2002
- [5] 김영훈 편저, "USB Guide," Ohm사, p.339-446, 2002.
- [6] "EZ-USB FX2 Technical Reference Manual," Cypress, 2001.
- [7] "DAB Baseband Demoboard Manual," Atmel, 2001.
- [8] Jaehee Cho, Namshin Cho, Keukjoon Bnag,

Myunghee Park, Heeyoung Jun, Hyuncheol Park, "PC-Based Receiver for Eureka-147 Digital Audio Broadcasting," *IEEE Transactions on Broadcasting*, vol,47, No 2. JUNE 2004.

박 노 경(Nho-Kyung Park)

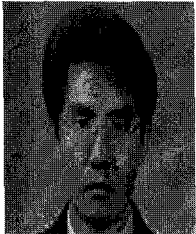
정회원



He received the B.S, M.S and Ph.D degree in Electronics from Korea University, Korea, in 1984, 1986 and 1990, respectively. From 1999.3 to 2002.2, he worked as a visiting professor in Information & Communication Engineering, Hoseo university, Asan, Korea. His current research interests include CAD, SoC Design, DAB/DMB etc.

진 현 준(Hyun-Joon Jin)

정회원



1984년 고려대학교 전자공학과 졸업
1986년 고려대학교 전자공학과 석사 졸업
1998년 미국 리하이대학교 전산학 박사 졸업.
1998년~현재 호서대학교 전기 정보통신공학부 부교수

<관심분야> 시스템 프로그래밍, 멀티미디어 정보처리 등

박 상 봉(Sang-Pong Park)

정회원

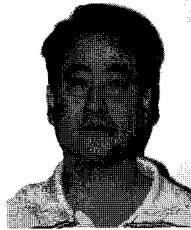


1985년 광운대 전자재료공학과 졸업
1987년 고려대 전자공학 석사 졸업
1992년 고려대 전자공학과 공학 박사
1992년 3월~1999년 2월 삼성전자 선임 연구원

1999년 3월~현재 세명대학교 정보통신학과 조교수
2000년 7월~현재 @lab(주) Digital 설계팀 기술자문
<관심분야> Sigma-Delta ADC, Serial ATA

김 상 복(Sang-Pok Kim)

정회원



1991년 호서대학교 정보통신공학과 석사. 졸업
2004년 호서대학교 정보통신공학과 공학 박사
1991년~현재 한국기술교육대학교 기술연구원
<관심분야> 반도체 설계 제어,

신호처리

한 성 호(Sung-Ho Han)

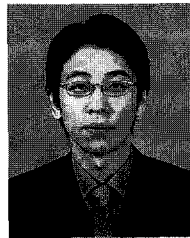
정회원



1991년 호서대학교 정보통신공학과 석사졸업
현재 호서대학교 박사과정 수료
<관심분야> 멀티미디어 정보처리, DAB

이 상 철(Sang-Chul Lee)

정회원



2003년 호서대학교 정보통신공학과 졸업
2005년 호서대학교 정보통신공학과 공학석사
현재 (주)실리콘이미지웍스 연구원
<관심분야> DSP 칩설계 펌웨어 등

어 등