

# 화상·음성 레코더를 위한 플래쉬 메모리 설계

신필순<sup>\*</sup> 김동현<sup>\*\*</sup> kwak윤식<sup>\*\*\*</sup> 김백기<sup>\*\*\*\*</sup> 신재룡<sup>\*\*\*\*</sup>

<sup>\*</sup>경희대학교 <sup>\*\*</sup>충주대학교 <sup>\*\*\*</sup>원주대학 <sup>\*\*\*\*</sup>충북대학교

## Design of the Flash Memory for Image/Voice Recorder

Shin Phil-Soon<sup>\*</sup> Kim Dong-Hyun<sup>\*\*</sup> Kwak Yoon-Sik<sup>\*\*\*</sup> Kim Baek-Ki<sup>\*\*\*\*</sup> Shin Jae-Ryong<sup>\*\*\*\*</sup>

<sup>\*</sup>Kyunghee University <sup>\*\*</sup>Chungju National University <sup>\*\*\*</sup>Wonju National College <sup>\*\*\*\*</sup>Chungbuk

University

E-mail : yskwak@gukwon.chungju.ac.kr

### 요 약

본 논문에서는 표준 화상·음성 코덱 알고리즘을 기반으로 화상 및 음성 레코더를 위한 메모리 설계방법을 제안하였다. 이를 구현하기 위하여 GDS30C6001 USB 컨트롤러를 이용하여 플래쉬 메모리 및 카드 응용시스템인 화상·음성 브라우저를 설계하였다. 화상 및 음성 데이터를 처리하기 위하여 음성 파일 루트 디렉터리와 화상 파일 루트 디렉터를 분리, 설계하였으며 각각의 데이터에 대한 응용을 확대하기 위해서 다양한 정보를 삽입시켰다.

### ABSTRACT

In this paper, we proposed flash memory design method for image and voice recoder based on the standard image·voice codec algorithm. For implementation of this method we designed image·voice browser which is application system of flash memory and card using GDS30C6001 USB controller. To process image and voice data we designed root directory of image and voice files repectively. To extend application of image and voice data we added various information to the system.

### 키워드

화상, 음성, 레코더, 플래쉬 메모리

### 1. 서 론

컴퓨터 기술의 발전과 더불어 컴퓨터 기능 및 디바이스에 대한 수요자의 욕구가 증대됨에 따라 컴퓨터 응용 기술은 획기적으로 발전되고 있다.

화상·음성브라우저는 다양한 디바이스의 활용도 증진을 목적으로 다양하게 개발되고 있으며 그 응용분야는 예측하기 어려운 것이다. 화상·음성에 대한 연구는 지금까지 다양하게 이루어져 왔으며 그 응용분야 또한 다양하다. 이렇게 다양하고 폭넓게 연구되고 있는 화상·음성분야는 함축적으로 화상·음성자료의 압축 및 해제를 위한 코덱(Coдек) 설계로 집중되고 있다[1][2]. 그 결과로 화상·음성자료의 코딩 및 디코딩을 위한 표

준화 알고리즘이 제시되어 있으며 이를 활용한 다양한 응용이 이루어지고 있는 상황이다[3][4]. 본 논문에서는 표준화로 제공되고 있는 알고리즘을 기반으로 하는 화상·음성 브라우저를 개발한 것으로 GDS30C6001이라는 USB 컨트롤러를 기반으로 하는 플래쉬메모리 및 카드 응용 시스템에 관한 것이다. 화상 및 음성 레코더를 위한 메모리 설계방법을 제안하였으며 이를 기초로 화상·음성 브라우저를 설계하였다[5][6][7]. 플래쉬 메모리는 화상 및 음성 데이터를 처리하기 위하여 음성 파일 루트디렉터리와 화상 루트 디렉터를 분리, 설계하였으며 각각의 데이터에 대한 응용을 확대하기 위해서 다양한 정보를 삽입시켰다.

본 논문의 구성은 1. USB를 이용한 사용자 인

터페이스 2. FAT 구조의 설계, 3. 메모리 액세스 알고리즘, 4. 시스템구현, 그리고 결론으로 구성되어 있다.

## II. 본 론

### 1. USB를 이용한 사용자 인터페이스

응용시스템의 인터페이스는 기존에 직렬(Serial) 또는 병렬(Parallel)통신 방식에 의존하였으나 응용시스템의 다양화 및 처리속도, 디바이스의 지원 문제 등 많은 장점을 갖고 있는 USB의 활용이 확대되고 있다. 구현된 시스템에서는 플래쉬메모리 내 데이터를 PC상으로 Upload·Download가 가능하고 기존 USB 테스트 프로그램을 upgrade 시키는 방향으로 설계되었으며, 그 파일구조는 그림 1과 같다. 파일 시스템 구조는 3레벨로 구성되며, 저 레벨의 H/W부분에서는 GDS3DC6001등과 관련된 인터페이스 부분이며, 커널 레벨의 Usbfmc.sys파일과 유저 레벨의 Cwusbmc.dll, P2MMC.dll파일로 구성되어 있다[3][4].

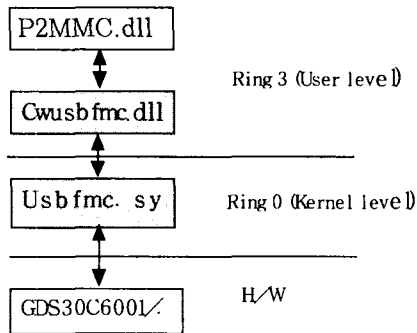


그림 1. 파일 구조

### 2. FAT 구조의 설계

순번	모델명	페이지크기 (바이트)		블럭크기	페이지 수	블럭수
		기본 페이지	여분			
1	K9F6408U0A	512	16	1블럭/16P	16	1024
2	K9F2808U0M	512	16	1블럭/32P	32	1024
3	K9D5608V0M	512	16	1블럭/32P	32	2048

그림 2. 모델별 메모리 구성 요소

본 시스템에서 사용한 메모리 및 카드는 8, 16M의 K9F6408U0A, K9F2808U0M, 32M SMC의 K9D5608V0M으로 그림 2와 같은 메모리 구조를

가지고 있다[8][9][10].

그림 2에서 제시된 것과 같이 기본적으로 블록의 크기는 256바이트 2개인 512 바이트 크기를 갖고 있으며 여기에 기본적인 16바이트의 여분 공간을 갖고 있다. 또한 8M 바이트인 경우에 있어서는 블록의 수가 1024이며 페이지 수는 16개이다. 이를 정리하면 식(1)과 같다.

$$\begin{aligned}
 1\text{Page} &= 528\text{Byte} \\
 1\text{Block} &= 528\text{Byte} \times 16\text{Page} \\
 &= (8K+256)\text{Bytes} \\
 1\text{Devices} &= 528\text{Bytes} \times 16\text{Pages} \times 1024\text{Blocks} \\
 &= 66\text{Mbits} \text{ -----(1)}
 \end{aligned}$$

이와 같은 메모리를 기초로 본 시스템의 설계를 위한 여러 가지의 메모리사양의 설계는 다음과 같다. 먼저 블록의 구조이다. 블록 0~3까지는 시스템에서 사용하게 될 다양한 정보가 저장될 것이며 블록4부터는 화상·음성 데이터가 저장되는 영역으로 설계되었다. 블록 0은 파일 루트 디렉토리(File Root Directory)이다. 본 영역에는 화상·음성비트, 일반 데이터비트, 주파수비트, 압축률 등의 11종류의 정보가 저장되며 각 비트에 대한 사양은 그림 3과 같다.

블록 1번은 FAT(File Allocation Table) 저장 영역이며, 블록 2번은 시스템정보 영역, 블록 3번은 MCU Modify 영역으로 사용된다.

내용	음성 데이터 비트	화상 데이터 비트	일반 데이터 비트	CMOS 센서	압축률
데이터 크기	1	1	1	1	1
비트 순서	0 Bit	1Bit	2Bit	3Bit	4Bit

내용	Group 정보	최초 녹음 상태	데이터 크기	데이터 시작 블록	Insert Voice File No	보존
데이터 크기	1	1	4Byte	2Byte	2Byte	7Byte
비트 순서	5Bit	6Bit	7Bit			

(a) 화상파일 루트 정보

그림 3. 정보 구조

### 3. 메모리 액세스 알고리즘

본 시스템구현을 위해 FAT 액세스 및 플래쉬 메모리에 대한 알고리즘은 다음과 같다. 먼저 FAT 영역 정보를 획득하기 위한 과정이다. 본 영역에는 "BYTE Buff [512+16]"으로 기본 데이터를 액세스시키고 여기에는 시작점정보, 크기, 블록정보가 저장되고 있다. 이를 위한 것이 "DWORD

stAddress, Size, Block;" 이다.

<FAT Access 알고리즘>

```

BYTE* FAT::ssfdc_READ_1st()
{
    BYTE Buff[512+16];
    DWORD stAddress, Size, Block;

    stAddress = 0x00;
    Size = 512;
    Block = 1;

    if(!SSFDCReadMultiBlock(Buff,
        stAddress, Size+32, Block)) {
        AfxMessageBox("Read error!.",
            MB_OK | MB_ICONHAND);
        return NULL;
    }

    return Buff;
}

BOOL FAT::OpenSSFDC(void)
{
    char str[100];

    if (!IOOpenPort(NULL, NULL)) {
        AfxMessageBox("Cannot open port.",
            MB_OK | MB_ICONHAND);
        return false;
    }

    // Samsung Flash Memory Select
    DEVICEWriteGPIO(3, 0);
    if( !SSFDCCheckCard() ) {
        DEVICEWriteGPIO(4, 0);
        if( !SSFDCCheckCard() ) {

AfxMessageBox("Not Found.(SSFDC or Flash
Memory).", MB_OK | MB_ICONHAND);
        return false;
        }
    }

    BYTE Device =
SSFDCGetDeviceCode();

    switch (Device) {
        case SSFDCDEVICE_4MB:
            wsprintf(str, "Flash Memory
            = %02x -> 4MB ", Device);
            break;
        .
        .
    }
}

```

```

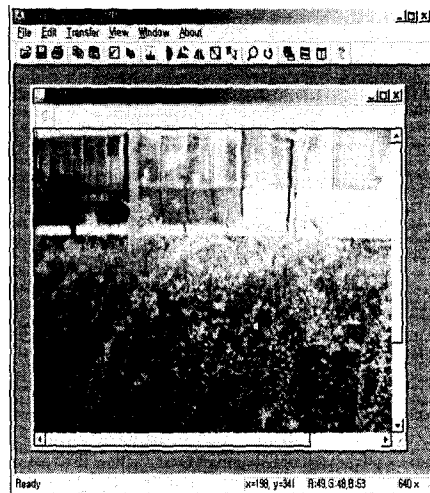
default:
    AfxMessageBox("Card
verification error, at OnSsfdcRead #2",
    MB_OK | MB_ICONHAND);
    IOClose();
    return false;
}

return true;
}

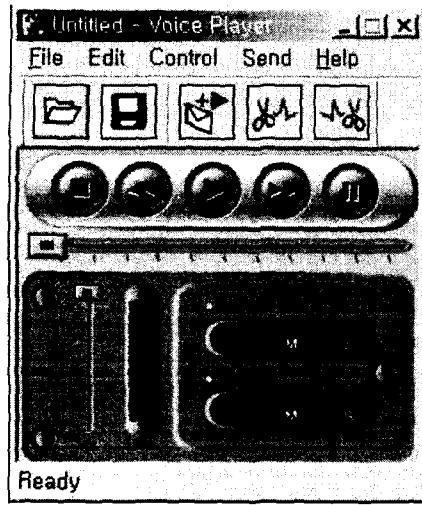
```

#### 4. 시스템 구현

본 시스템은 윈도우 환경에서 사용이 가능하도록 시스템을 설계하기 위해서 OS는 윈도우 98, 그리고 펜티엄 II PC를 사용하였으며 구현 언어는 C++을 사용하였다. 사용된 USB 컨트롤러는 현대전자에서 생산되고 있는 GDS30C6001 USB 컨트롤러를 기반으로 하였으며, 플래쉬 메모리는 삼성전자에서 생산되는 K9F6408U0A, K9F2808U0M 8, 16Mbyte와 32Mbyte SMC인 K9D5608V0M로 실험을 수행하였다. 구현된 시스템을 나타낸 것이 그림 4이다.



(a) 화상브라우저



(b)음성브라우저

그림 4. 구현 시스템

### III. 결 론

본 논문에서는 GDS30C6001이라는 USB콘트롤러를 기반으로 하는 플래쉬메모리 및 카드 응용 시스템에 관한 것이다.

화상 및 음성 레코더를 위한 메모리 설계방법을 제안하였으며 이를 기초로 화상/음성 브라우저를 설계하였다. 플래쉬 메모리는 화상 및 음성 데이터를 처리하기 위하여 음성 파일 루트디렉터리와 화상 루트 디렉토리를 분리, 설계하였으며 각각의 데이터에 대한 응용을 확대하기 위해서 다양한 정보를 삽입시켰다.

또한 기존의 시스템의 경우는 메모리에 대한 기능이 부족하고 부가적으로 제공되는 기능이 하드웨어 시스템의 설계 및 디버깅 과정에 비효율적인 관계로 이를 해결할 수 있었다. 시스템 구현에 사용하는 GDS30C6001은 MMC, SSFDC카드 인터페이스 콘트롤러로 최고속도 12Mbit/s, USB Ver 1.1 사양/윈도우 95, 98 그리고 MacOS 8.1에서 구동되며 플래쉬 메모리로는 8M, 16M 그리고 32M SMC에 대해서 자동 읽기 기능, 메모리 블록 읽기/쓰기 기능, 메모리 덤프 기능, 파일 property 기능을 구현하였다.

### 참고문헌

- [1] CHRIS CANT, Writing Windows WDM Device Drivers, R&D Books,1999
- [2] Dekker, Newcomer,Developing Windows NT

- Device Drivers, Addison Wesley,1999
- [3] RANDY CRANE, "A simplified approach to Image Processing", Prentice-Hall,1997
- [4] Calstleman, "Digital Image Processing", Prentice-Hall,1979
- [5] Universal Serial Bus Specification Revision 1.1
- [6] Universal Serial Bus SSFDC/MMC I/F Controller Data Sheet(GDS30C6001), HYUNDAI Electronics industries.co.LTD, 2000
- [7] MAS3504D G.729 Annex A Voice Codec MICRONAS, 2000
- [8] K9F6408U0A-TCB0,K9F6408U0A-TIB0 Flash Memory Data Sheet, Samsung Semi.co.LTD,1999
- [9] K9F2808U0M Flash Memory Data Sheet,Samsung Semi.co.LTD, 1999
- [10] K9D5608V0M Flash Memory Data Sheet,Samsung Semi.co.LTD,1999