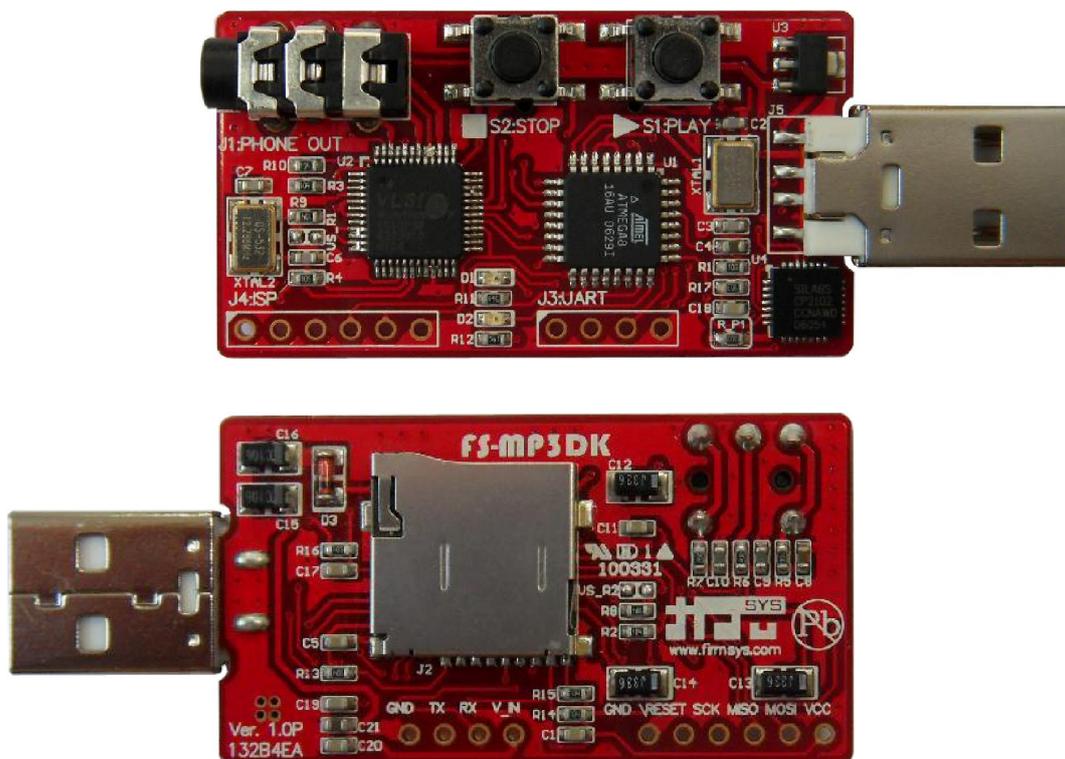


FirmSYS MP3 Developer Kit

User Manual

FS-MP3DK





목 차

| | |
|----------------------|----|
| 1. 개요 | 3 |
| 2. 블럭도 설명 | 4 |
| 3. 회로도 설명 | 5 |
| 부품 목록 | 7 |
| 4. 펌웨어 설명 | 8 |
| 헤더파일 목록 | 9 |
| SD 메모리 | 11 |
| 파일시스템 | 12 |
| 5. 프로토콜 설명 | 13 |
| 프로토콜 목록 | 14 |
| 6. USB 드라이버 설치 | 16 |
| 7. 유틸 설명 | 17 |
| 섹터읽기 | 19 |
| 8. 프로그램 설명 | 20 |
| 9. 기술적 특성 | 21 |
| 프로토콜 | 21 |
| 모듈 포트 | 22 |
| 10. 전기적 특성 | 23 |
| 11. 기구적 특성 | 24 |
| 모듈 치수 | 24 |
| 모듈 무게 | 24 |
| 12. 메뉴얼 정보 | 25 |

1. 개요

펄시스의 FS-MP3DK는 MP3 decoder IC(VS1003)와 SD card 사용하는 개발자를 위해 모든 자료를 제공한다. 회로도, 펌웨어, 소스코드, 어플리케이션, 데이터시트 등 제공되는 자료로 개발기간을 단축시킬 수 있다.

FAT32 파일시스템으로 2GB 이하의 microSD 와 2GB 이상의 SDHC 를 모두 지원한다.

자세한 펌웨어 설명과 라인별 주석으로 누구나 쉽게 이해할 수 있다.

외부 전원과 USART 뿐 아니라 USB타입으로 별도 전원 없이도 PC에 꽂아 바로 사용할 수 있다.

MP3파일의 재생, 정지뿐 아니라 볼륨, 밸런스, 이퀄라이저, 홀드, 파일목록 등등 다양한 기능을 지원하며, 펌웨어 다운로드 포트(ISP 6P)가 있어 직접 소스코딩 하여 테스트할 수 있다.

FS-MP3DK 특징 요약

- USB 타입 소형 모듈로 전원공급 및 통신
- 펌웨어, 회로도, 유틸 등 개발자를 위한 CD 제공
- 다양한 기능 (다음곡, 이전곡, 볼륨, 홀드, 밸런스, 이퀄라이저, 파일목록 등등)
- 저전력 설계 및 파워다운 모드 지원
- FAT32 microSD SDHC 지원
- USART 포트
- AVR ISP 포트로 직접 소스코딩

제공되는 CD의 내용은 아래와 같다.

- Firmware Hex File
- Firmware Full SourceCode
- 회로도
- 윈도우용 어플리케이션 및 소스코드
- 통신 유틸
- USB 드라이버(Window XP, 2K, vista, 7, CE, MacOS, Linux)
- BOM 리스트 및 데이터시트 모음

2. 블록도 설명

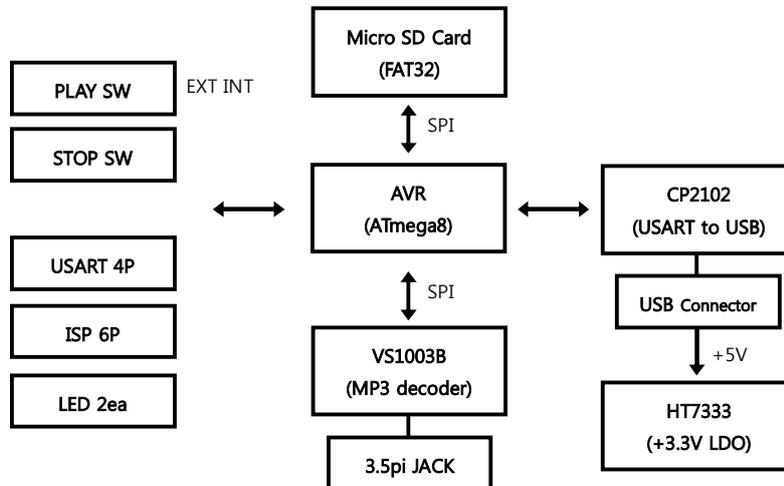


그림1. 블록도

- ATmega8는 16Mhz의 크리스탈을 사용하고 SD 메모리와 VS1003B를 SPI로 제어한다.
- CP2102는 제어가 필요 없는 IC 이다.
- PLAY SW(INT1)와 uSD 슬롯(INT0)은 외부 인터럽트 핀에 연결되어 파워다운 모드에서 wake-up 소스로 사용된다.
- USB에서 공급되는 +5V를 HT7333에서 +3.3V로 변환하여 공급한다.
- 외부포트는 USART 4핀(5V_IN, RX, TX, GND) 와 펌웨어 다운로드 포트 6핀(VCC, MOSI, MISO, SCK, RESET, GND) 이 있다.

3. 회로도 설명

- ① ATmega8 로 외부 16MHz의 크리스탈을 사용한다.
 - SPI통신으로 VS1003와 SDcard 를 제어
 - 2개의 SW 입력 제어
 - USART 입출력
 - 적색 녹색 LED Sink 제어
- ② VLSI의 VS1003b는 MP3 decoder IC로 외부 12.288MHz의 크리스탈을 사용한다.
 - DAC 내장형으로 phone out 출력 포트 지원
 - VS1033과 호환 가능하도록 VS_R1, VS_R2 패드 지원
- ③ SILAB의 CP2102는 USART를 가상 시리얼 USB로 변환해주는 IC이다.
 - SILAB사의 USB 드라이버 지원(Window XP, 2K, vista, 7, CE, MacOS, Linux)
- ④ microSD 슬롯으로 3.3V를 사용하고 SPI모드로 제어한다.
 - 슬롯의 CD pin 은 SDcard의 유무를 확인하는 스위치로 삽입되었을 때 low 이다.
- ⑤ HOLTEK의 HT7333는 최대전류 250mA의 LDO(+3.3V) 이다.
 - 누설전류와 전압강하가 각각 4uA, 80mV로 매우 우수하다.
- ⑥ 1N4148 범용 다이오드로 전압강하 0.7V~1V를 이용하여 VS1003에 필요한 2.4V~2.65V를 얻는다.
- ⑦ 재생과 정지 버튼으로 누르면 low가 되어 ATmega8에 입력된다.
- ⑧ 전원LED(녹)와 신호LED(적)로 ATmega8에서 전류 sink 제어한다.
- ⑨ USB를 사용하지 않을 때 전원과 USART 4pin으로 R_P1(0Ω)을 반드시 제거하고 사용한다.
- ⑩ ATmega8의 펌웨어 다운로드 ISP(6pin)포트로 시중에 판매되는 전원 공급형 AVR USB ISP로 핀배열을 맞추어 사용한다.
- ⑪ 3.5파이 DIP타입 stereo phone jack
- ⑫ USB male DIP타입 커넥터로 전원 공급과 가상 시리얼 통신을 한다.

【 부품 목록 】

| Item | Quantity | Reference | Part |
|------|----------|-----------------------------|---------------|
| 1 | 7 | C1,C5,C11,C17,C18,C19,C20 | 100nF |
| 2 | 4 | C2,C3,C6,C7 | 22pF |
| 3 | 2 | C21,C4 | 1uF |
| 4 | 1 | C8 | 47nF |
| 5 | 2 | C10,C9 | 10nF |
| 6 | 3 | C12,C13,C14 | 33uF Tantalum |
| 7 | 2 | C15,C16 | 10uF Tantalum |
| 8 | 1 | D1 | LED_SIG |
| 9 | 1 | D2 | LED_PW |
| 10 | 1 | D3 | D1N4148 |
| 11 | 1 | J1 | PHONE OUT |
| 12 | 1 | J2 | MicroSD |
| 13 | 1 | J3 | UART |
| 14 | 1 | J4 | ISP |
| 15 | 1 | J5 | CON4 |
| 16 | 1 | R_P1 | 0 |
| 17 | 3 | R1,R16,R17 | 10k |
| 18 | 8 | R2,R3,R8,R9,R10,R13,R14,R15 | 100K |
| 19 | 1 | R4 | 1M |
| 20 | 1 | R5 | 10 |
| 21 | 2 | R7,R6 | 20 |
| 22 | 2 | R11,R12 | 560 |
| 23 | 1 | S1 | PLAY/FF |
| 24 | 1 | S2 | STOP |
| 25 | 1 | U1 | ATmega8 |
| 26 | 1 | U2 | VS1003B |
| 27 | 1 | U3 | HT7333 |
| 28 | 1 | U4 | CP2102 |
| 29 | 2 | VS_R2,VS_R1 | NC |
| 30 | 1 | XTAL1 | 16.00MHz |
| 31 | 1 | XTAL2 | 12.288MHz |

4. 펌웨어 설명

소스코드의 전체 구조를 충분히 이해하고 사용한다. (전역변수의 쓰임은 반드시 이해한다.)
 많은 기능을 넣어 DK보드로써 활용도를 높였다. (제공CD\W\펌웨어\FirmSYS-MP3DK_V1.0P(20100401).zip)
 소스코드 각 라인마다 자세한 주석이 있어 직접 소스코드를 보면 쉽게 이해할 수 있다.
 전역변수를 최소화하고, 함수안에 너무 많은 것을 넣지 않았다.
 기능별로 모듈화 시켜 사용자가 수정 및 확장 할 수 있게 최대한 신경을 썼다.
 MP3파일의 아날로그 출력 때문에 재생, 정지, 곡전환 시 잡음을 최소화하기 위한 코드가 삽입되어 있고,
 또 전류를 최소화하기 위한 코드도 포함되어 다소 복잡해 보일 수 있다.
 대부분의 필요한 기능을 인터럽트 처리하였고, IC특성상(VS1003) 메인문에서 폴링 처리한 부분도 있다.
 예상치 못한 상황에서 무한루프에 빠지는 현상을 고려하여 while(1)문을 최소화 하였다.
 왓치독 타이머 리셋으로 먹통 현상을 막는다.
 무료의 GCC컴파일러를 사용했으나, 약간의 수정을 거쳐 IAR, KEIL로 쉽게 변환할 수 있다.
 (제공CD\WAVR 컴파일러)

- Compiler : WinAVR-20060426
- AVR Memory Usage
- Program: 7756 bytes (.text + .data + .bootloader)
- Data: 876 bytes (.data + .bss + .noinit)

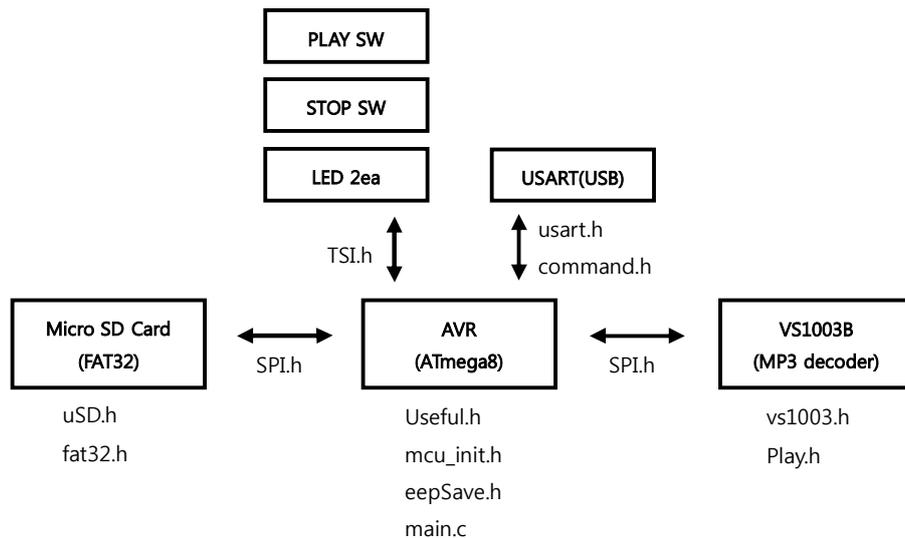


그림4. 펌웨어 구조

【 헤더파일 목록 】

| Header | Function | Description |
|-------------------|---|---|
| Useful.h | NOP(); void SleepMs(u16 D) void SleepUs(u16 D) | 변수타입 정의 딜레이 함수 (ms, us) |
| mcu_init.h | void MCU_init(void) void LEDR_blink(void) void LEDG_blink(void) void SW_scan(void) | MCU 포트 및 레지스터 정의 LED 함수 PLAY, STOP 입력 스캔 함수 |
| eepSave.h | void EEP_setup(void) uint8_t EEP_read_byte(const uint8_t *addr) void EEP_write_byte(uint8_t *addr, uint8_t val) | 볼륨값 저장을 위해 내부 EEPROM 초기화, 읽기, 쓰기 함수 |
| SPI.h | SPI_init() SPI_Mode0() SPI_Mode3() SPI_Mode3_Slow() | VS1003 과 SD메모리 통신을 위해 SPI 초기화 및 모드 함수 |
| TSI.h | void TimerCount2_init(void) SIGNAL(SIG_OVERFLOW2) void TSI_set(u08 TSInum, u16 TSIcnt, u32 TSIprd, void(*Func)(void)) | Timer Sequence Interrupt 로 매우 유용하게 쓰인다. 8-bit Timer Overflow Interrupt2 를 10ms tick으로 걸리게 하여 함수 포인터를 사용한다. 실행함수, 실행횟수, 실행간격을 설정한다. SW_scan(), LEDR_blink(), LEDG_blink(), sleep_down() 이 TSI_set 사용하여 인터럽트 처리한다. |
| usart.h | void uart_init(void) void tx(u08 T) SIGNAL(SIG_UART_RECV) void RX_wait(void) void hex_print(u08 h) void put_char(char *a) void dec_print_int(u16 d) u08 ASCIItoHEX8(u08 pH, u08 pM, u08 pL) u16 ASCIItoHEX16(u08 H1, u08 H2, u08 L1, u08 L2) | usart 초기화 및 관련 함수 RX는 인터럽트 처리한다. |
| command.h | void command_action(void) u08 uart_command(void) u08 comm_check(char *comm, u08 num) void sleep_down(void) void sleep_check(void) | usart 로 받은 명령어를 확인하여 분류하고 처리하는 관련 함수 |
| uSD.h | UINT8 MMC_init(void) void mmcCommand(UINT8 cmd, UINT32 arg) void mmcRead(UINT32sector, UINT8* buffer) void mmcWirte(UINT32 sector, UINT8 *buffer) | SD메모리 초기화 및 읽고 쓰기 함수 SD메모리는 섹터(512-byte)단위로 읽고 쓴다. 2G이하 일반SD 와 4G이상의 SDHC 모두 지원한다. |
| fat32.h | void FAT_init(u16 num) u32 ClusterToSector(u32 tmp) | FAT32 파일시스템을 인식하여 MP3 파일을 검색한다. SD메모리를 FAT32로 포맷할 때 MBR 영역이 0번지 |

| | | |
|-----------------|---|---|
| | <p>u32 NextCluster(u32 tmp)</p> <p>u32 SearchRootDir(u08* name, u08 Ext0, u08 Ext1, u08 Ext2, u08 Attr)</p> <p>u32 SearchSubDir_Name(u32 SubClus, u08* name, u08 Ext0, u08 Ext1, u08 Ext2, u08 Attr, u16* Song_num, u08 name_num)</p> <p>u32 SearchSubDir_Num(u32 SubClus, u16 MP3_num, u08 Ext0, u08 Ext1, u08 Ext2, u08 Attr)</p> <p>u08 FileList_SubDir(u32 SubClus, u08 Ext0, u08 Ext1, u08 Ext2, u08 Attr)</p> | <p>섹터가 아닌 경우가 있어 주의 해야 한다.</p> <p>MP3 파일 검색에 필요한 함수로 구성되어 있다.</p> |
| vs1003.h | <p>void VS1033_io_init(void)</p> <p>void VS1033_Reset(void)</p> <p>void VS1033_init(void)</p> <p>void SCI_Write(u08 addr, u16 tx_data)</p> <p>u16 SCI_Read(u08 addr)</p> <p>void VS1033_SetVolume(u08 Lvol, u08 Rvol)</p> | <p>VS1003 의 초기화 및 관련 함수</p> |
| Play.h | <p>u08 Playing(u08* Name, u08 name_num, u16 MP3_num, u08 Mode)</p> <p>void Playing_sub(void)</p> | <p>MP3파일 데이터를 VS1003 으로 전송하여 재생하는 함수</p> |

【 SD메모리 】

개인 PC에서 SD리더를 사용하여 FAT32로 포맷한다.

 또는  폴더를 만들어 mp3파일을 넣는다.
MP3_FILE mp3_file

MP3DK 보드 뒷면의 마이크로SD 슬롯에 메모리를 꽂아 사용한다.

반드시 FAT32 파일시스템으로 포맷하여 사용한다. (FAT16은 지원하지 않는다)

2GB 이하의 microSD 와 2GB 이상의 SDHC 를 모두 지원한다.

SD메모리의 초기화 방법은 펌웨어 소스코드와 SD 데이터시트를 참고한다.

용량에 따라 FAT32 으로 포맷할 때 MBR(Master Boot Record) 영역이 0x0000 번지가 아닌 경우가 있다.

MBR 영역이 0x0000 번지가 아닐 때 0x2000번지를 검색한다.

~2G, 4G, 8G 의 메모리는 MBR 영역이 0x0000 또는 0x2000 번지로 써지는 것을 확인하였다.

그러나 8G 이상의 메모리는 확인되지 않았다.

만약 8G 이상의 메모리가 예상치 못한 곳에 MBR 영역을 쓴다면 펌웨어 수정을 해야 된다.

SD메모리에 "MP3_FILE" 또는 "mp3_file" (대소문자 구분없음)의 이름으로 폴더를 만들어 mp3 파일을 넣는다.

mp3_file 폴더 안의 mp3파일만 검색하여 재생된다.

폴더명칭과 폴더개수는 펌웨어 수정으로 얼마든지 직접 변경할 수 있다.

【 파일시스템 】

파일시스템(FAT32)에 대한 자세한 설명은 관련 도서를 참조하길 권장한다.

인터넷으로 검색하여 정보를 얻을 수 있지만 정확하고 신뢰할 수 있는 정보는 아니다.

FAT32는 오래 전에 나온 파일시스템으로 생각보다 꽤 간단한 시스템이다.

기본단위인 섹터(512byte)를 묶어 클러스터로 명칭하고 클러스터 주소로 데이터를 읽고 쓴다.

그리고 MBR과 FAT영역에 모든 정보를 담고 있어 이것만 분석되면 FAT32를 모두 다 아는 것이다.

특히 중요한 것이 MBR(Master Boot Record) 인데 이것의 내용은 명확히 정의 되어 있어 쉽다.

문제는 이것의 위치이다. 가장 먼저 읽어서 파악해야 될 부분이 MBR인데,

이것이 SD메모리의 어떤 섹터번지에 있는지 명확한 설명이 없다.

그냥 offset 0번지로 되어있는데, offset의 크기는 어떻게 알 수 있는지 별 얘기가 없다.

실제 코딩에서 이것 때문에 많은 시간이 낭비되었다. MBR영역의 내용을 모르면 SD메모리는 인식불가이다.

offset의 크기를 알기 위해 섹터 0번지부터 검색해 본 결과 MBR영역은 0x0000 또는 0x2000 섹터에 있었다.

8G SDHC 까지 검색한 결과이고 이것이 모두 적용된다고 말할 수 없다.

만약 8G 이상의 메모리가 예상치 못한 곳에 MBR 영역을 쓴다면 펌웨어 수정을 해야 된다.

MBR의 내용이 인식되면 규칙대로 따라가면 그 뿐이다. 어렵지도 않고 복잡하지도 않고 내용이 많지도 않다.

파일시스템의 관련 자료로 아래 도서를 추천한다.

파일시스템의 원리와 실습

정준석 정원용 공저

한빛미디어

5. 프로토콜 설명

프로토콜은 ASCII문자로 쉬운 영어단어로 되어 있다.

14개의 명령어들과 그에 대한 응답으로 구성되어 있다. (Request & Response)

사용자는 각각의 명령어 기능을 충분히 이해하고 사용해야 한다.

USB는 가상 시리얼 통신을 사용한다.

USART 설정은 아래와 같다.

- Baud Rate : 115200
- Data Bit : 8
- Parity : none
- Stop Bit : 1
- Handshakig : none

프로토콜 프레임 기본구조

| Request | Response |
|------------------------------------|--|
| ASCII word | ASCII word EOF |
| 영어단어로 대소문자 구분되어 있고 응답과 달리 EOF가 없다. | EOF(End Of Frame) : 0x0D(CR) 0x0A(LF) 영어단어로 대소문자 구분되어 있고 끝에 EOF가 반드시 붙는다. |

【 프로토콜 목록 】

| Command | Request (ASCII) | Response (ASCII+HEX) | Description |
|---------|-----------------|-------------------------------|--|
| 재생 | "play>>" | "PLAY_00001" EOF | 다음곡 재생 (재생모드는 전체 순차 반복이다.) |
| | "play<<" | "PLAY_00001" EOF | 이전곡 재생 (재생모드는 전체 순차 반복이다.) |
| | "Play_oooooooo" | "PLAY_00001" EOF | 해당 파일명 재생 ① (파일명이 8byte 초과된 Long File Name 인 경우 Short File Name 으로 변환되어 기록된 엔트리에서 검색한다. "list" 명령어의 응답이 Short File Name 이다. 첫자부터 비교하여 같으면 재생되고, 8byte 이하로 검색 가능하다.) |
| 일시정지 | "hold" | "HOLD" EOF "PLAY" EOF | 일시정지 일시정지 해제 (정지/해제가 토글로 동작한다.) |
| 정지 | "stop" | "STOP" EOF | 정지 (재생하던 곡의 처음으로 정지된다.) |
| 파일목록 | "list" | "00001. FileName" EOF | MP3_FILE 폴더의 MP3 File Name 목록 출력 ② (파일명이 8byte 초과된 Long File Name 인 경우 Short File Name 으로 변환되어 기록된 엔트리에서 출력한다. 순번이 파일명 앞에 붙어 출력된다.) |
| 볼륨 | "volume_215" | "ACK" EOF | 볼륨 크기 조절 ③ (3자리수로 "000" ~ "255" -0.5dB 스텝으로 감쇠한다. 실제 "200" 이상부터 의미 있는 증감이 된다.) |
| 이퀄라이저 | "bass_0000" | "ACK" EOF | 이퀄라이저 조절 ④ (고음과 저음이 선택된 주파수에 따라 증감된다.) |
| 밸런스 | "balance_L" | "ACK" EOF | 좌우 채널 선택 ⑤ ("L":좌, "R":우, "M":좌우) |
| 리셋 | "reset" | "ACK" EOF | 시스템 리셋 (ATmega8 이 watchdog 리셋 된다.) |
| 볼륨저장 | "save" | "ACK" EOF | 현재 볼륨 값을 eeprom 에 저장한다. (리셋시 보존된다.) |
| 슬립타이머 | "sleep_060" | "ACK" EOF "SLEEP DOWN" EOF | 설정시간 후에 파워다운 된다. 재생버튼 누르면 깨어난다. (1분 단위로 최대 255분) |
| 파워다운 | "pdown" | "POWER DOWN" EOF | 즉시 파워다운 된다. (재생버튼 누르면 깨어난다.) |
| 섹터읽기 | "sector_2000" | 일정 포맷으로 출력 | SD카드의 섹터(512 bytes)를 읽어 나열한다. (입력값 : 16bit hex 값으로 SD의 섹터 번지수) |

① FAT 파일시스템 LFNs(Long File Names)를 지원하여 최대 255자까지 유니코드(UTF-16)방식으로 인코딩되어 있어 다국어 지원이 가능하며, 기존의 Short File Name과 호환된다.

- ② 8자가 넘는 File Name은 Short File Name 변환 규칙에 따라 별칭을 생성하여 File Name Entry에 저장된다. 문자열에서 앞의 6자리만 얻어 ~를 넣은 후 숫자(1,2,3,...)를 적는다.
- ③ "255"가 최대 볼륨값이고 "000" 최소 볼륨값(mute) 이다.
예를 들어 "215"는 최대 볼륨에서 -20dB 된 값이다.
- ④ 이퀄라이저 조절은 VS1003 데이터시트 34페이지를 참조한다.

8.6.3 SCLBASS (RW)

| Name | Bits | Description |
|--------------|-------|---|
| ST_AMPLITUDE | 15:12 | Treble Control in 1.5 dB steps (-8..7, 0 = off) |
| ST_FREQLIMIT | 11:8 | Lower limit frequency in 1000 Hz steps (0..15) |
| SB_AMPLITUDE | 7:4 | Bass Enhancement in 1 dB steps (0..15, 0 = off) |
| SB_FREQLIMIT | 3:0 | Lower limit frequency in 10 Hz steps (2..15) |

예를 들어 "7A00" 은 10kHz의 고음부에 10.5dB 증가한다.

"00F6" 은 60Hz의 저음부에 15dB 증가한다.

- ⑤ "L", "R", "M" 으로 선택된 채널만 현재 볼륨으로 유지되고 다른 쪽은 음소거 된다.

6. USB 드라이버 설치

MP3DK는 SILAB의 USART를 가상 시리얼 USB로 변환해주는 IC(CP2102)를 사용한다.

CD 디스크에 들어 있는 드라이버를 사용 OS에 맞게 설치 해주면 된다. (제공CD\USB드라이버\Driver)

- SILAB사의 USB 드라이버 지원(Window XP, 2K, vista, 7, CE, MacOS, Linux)

SILAB은 가상 시리얼 드라이버 외에도 USBXpress 드라이버도 지원한다.

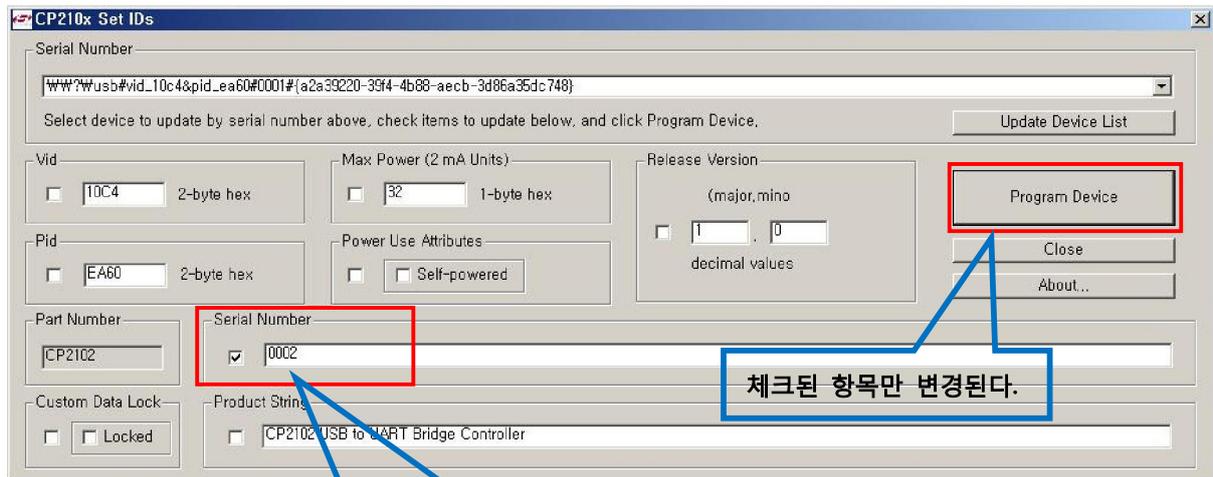
USBXpress는 사용 방법과 정보는 SILAB의 홈페이지에서 얻을 수 있다.

CP2102를 PC에 하나 이상 꽂아 사용할 때 USB Device ID가 겹쳐 인식되지 않으므로 설정 유틸 (CP210xSetIDs.exe) 을 사용하여 변경해 주어야 한다. (하나만 사용할 때는 변경이 필요 없다)

아래 그림에서 설명된 것 이외의 값들은 변경하지 않는다.

잘못 변경되었을 경우 인식되지 않을 수 있으므로 주의한다.

설정 유틸에 관한 자세한 설명은 SILAB 홈페이지와 관련 문서를 참조한다. (<http://www.silabs.com>)



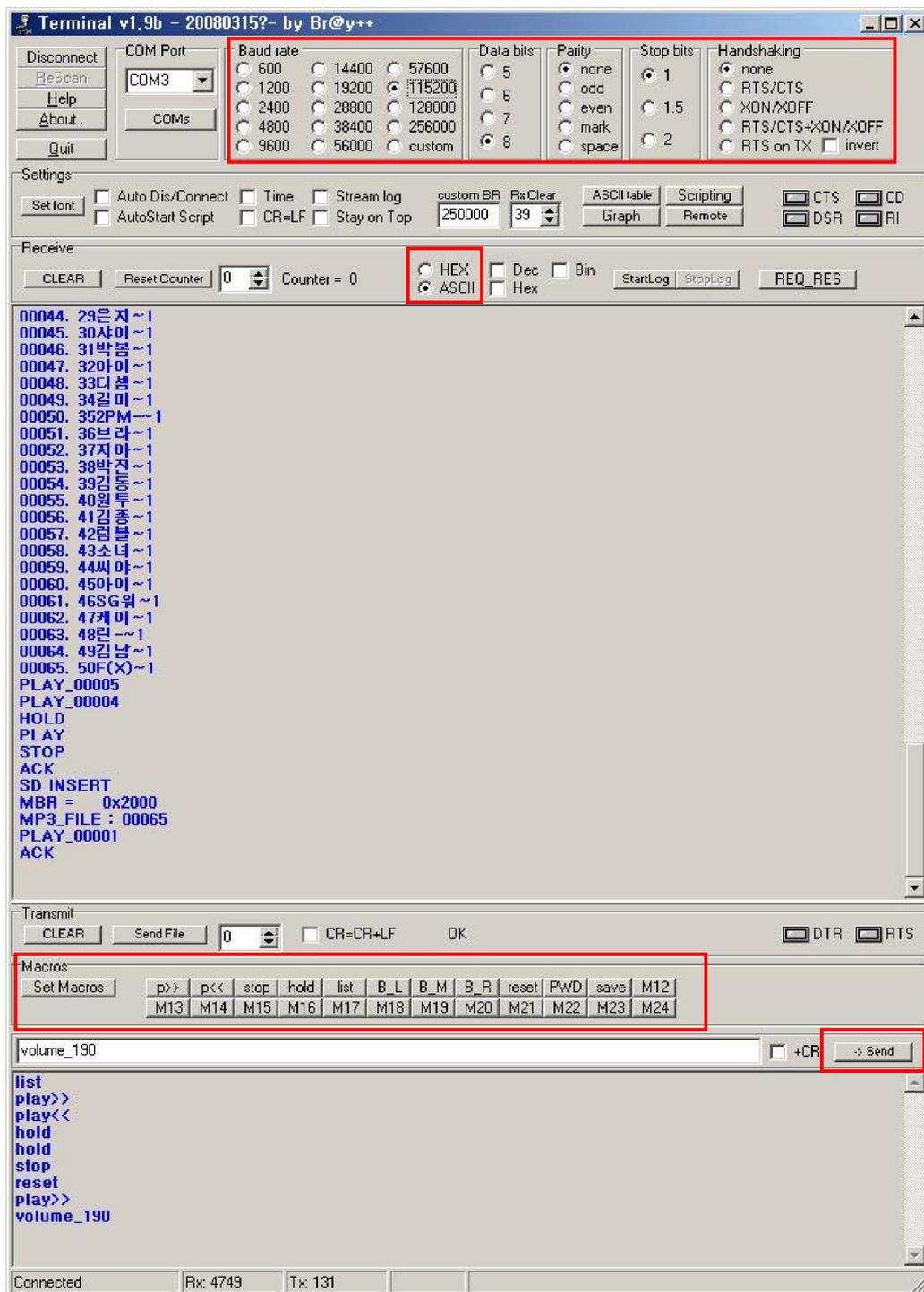
기본값이 '1'로 되어있고 이 값을 USB 별로 다르게 해주고 체크박스에 체크한다.

7. 유틸 설명

범용 통신 유틸로 MP3DK의 모든 기능을 쉽고 간편하게 조작할 수 있다.

Terminal v1.9b 는 아래 주소에서 무료로 다운받을 수 있다. (제공CD\유틸\Terminal20080315.zip)

<http://sites.google.com/site/braypp/terminal>





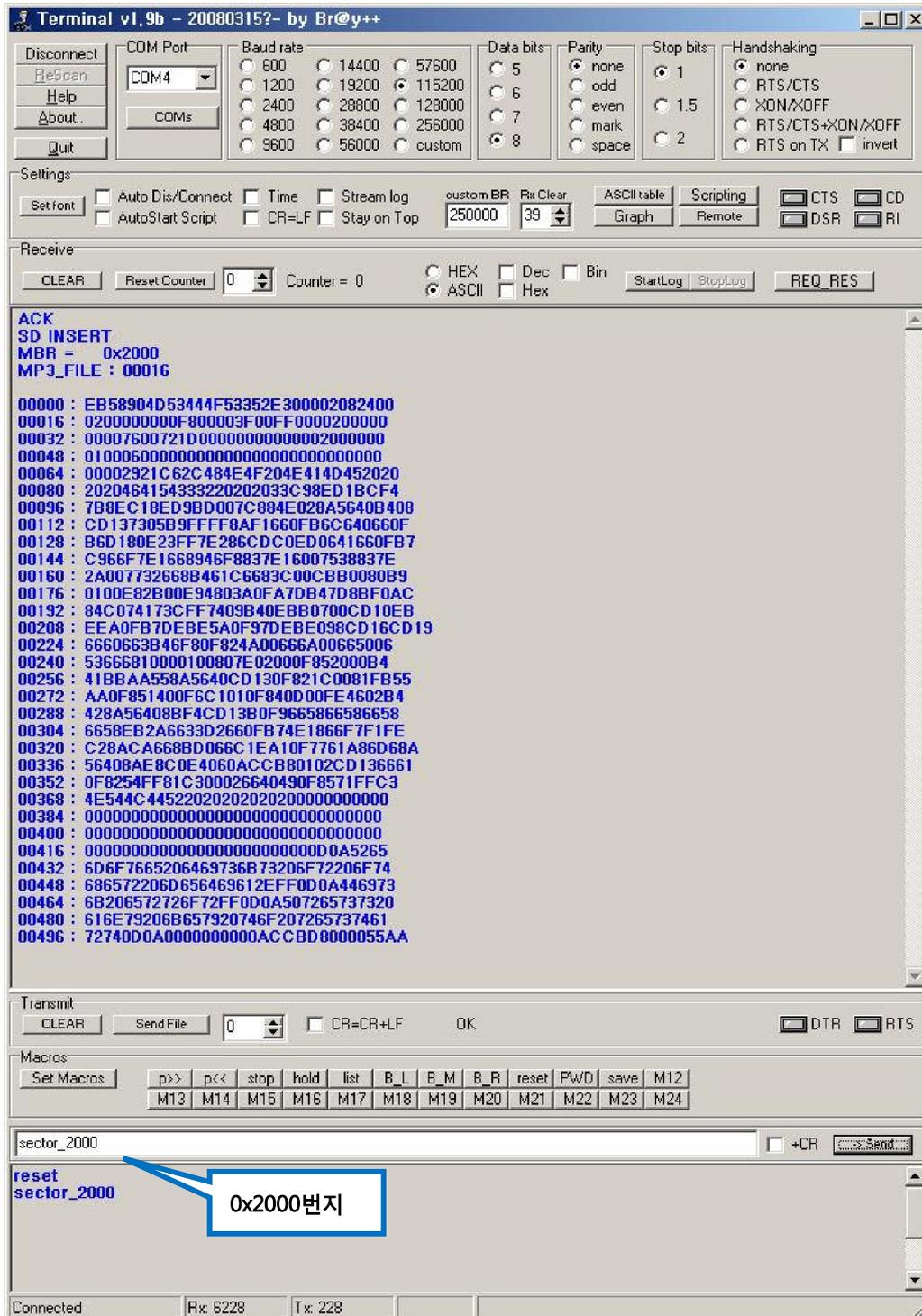
명령어 등록

버튼명 등록

【 섹터읽기 】

섹터읽기 명령으로 SD메모리의 섹터내용을 확인할 수 있다.

SD메모리는 섹터(512bytes)단위로 R/W 가능하며, 개발시 직접 SD메모리의 내용을 확인할 수 있다.

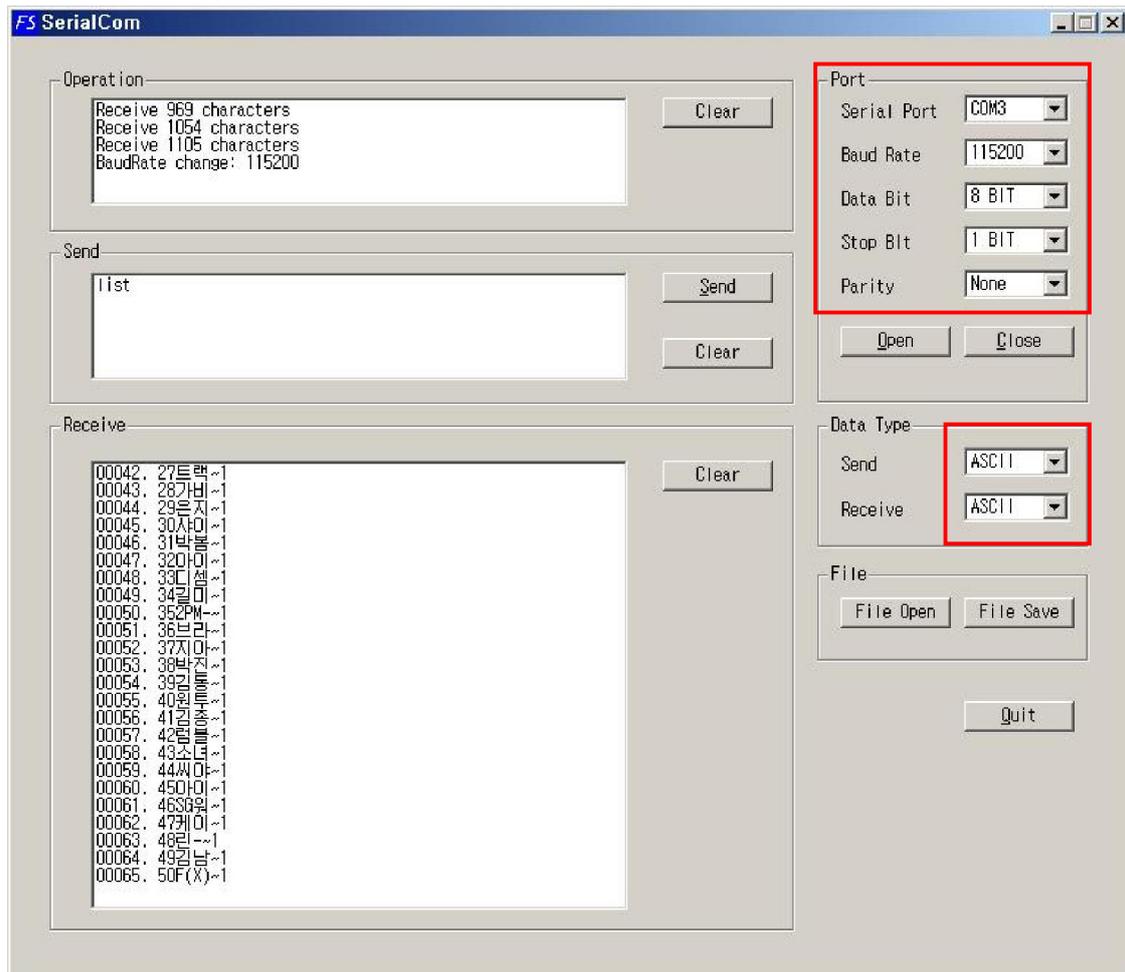


8. 프로그램 설명

어플리케이션 프로그램은 Microsoft Visual Studio 6.0 C++로 코딩 되었고 소스코드와 실행 파일이 CD에 들어 있다. (제공CD\유틸\SerialCom.zip)

간단한 시리얼 통신 프로그램으로 소스코드 설명은 라인별 주석으로 되어 있다.

이 소스코드를 사용하면 PC 어플리케이션 코딩에 도움이 될 것이다.

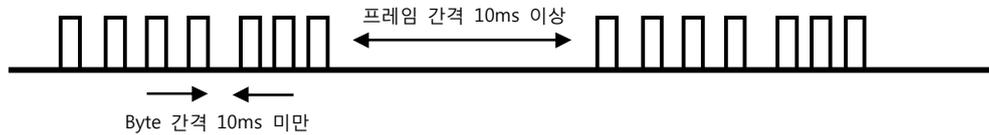


9. 기술적 특성

【 프로토콜 】

보낼 때

명령어 프레임은 끊김 없이 전송되어야 한다. 전송 바이트 간격이 5ms 미만으로 전송되어야 명령어를 인식한다. 그러므로 명령어 프레임당 간격은 5ms 이상으로 전송되어야 명령어 구분을 할 수 있다.

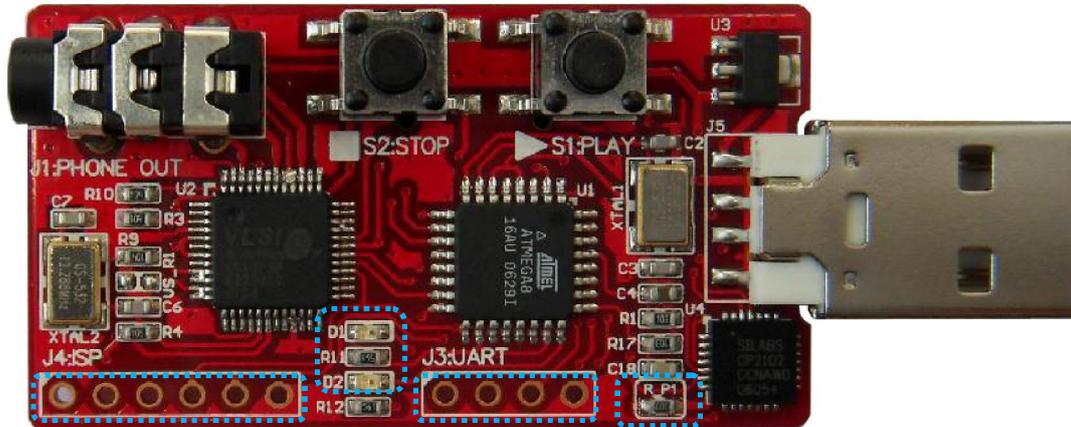


받을 때

응답 역시 프레임 단위로 끊김 없이 전송된다.

보낼 때와 다르게 받는 프레임에는 끝에 EOF(2byte)가 포함 되어 있다.

【 모듈 포트 】

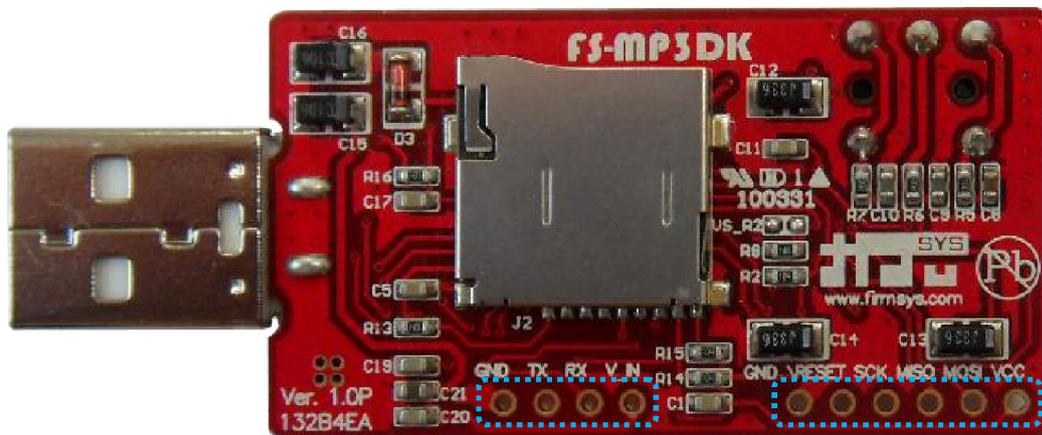


J4 : AVR ISP
 펌웨어 다운로드용으로
 전원(3~5V) 공급형 USB
 타입으로 사용하면 된다.

D1 : 신호LED(R)
 D2 : 전원LED(G)

외부전원과 USART 사용시
 반드시 저항(R_P1)제거

J3 : USART,외부전원(DC 5~9V)
 USB는 자체 전원을 사용하므로 USART
 사용 할 경우만 전원입력(R_P1 제거후)



USART 핀배열을 확인하고 연결한다.

TX와 RX는 MP3DK 모듈기준으로 타보드와
 연결할 때 크로스 연결한다.

V_IN은 전원입력은 DC 3.3V ~ 5V 이다.

ISP 핀배열을 확인하고 연결한다.

+3.3V는 +5V 호환된다.

10. 전기적 특성

본 모듈은 사용된 IC의 데이터시트를 참조하여 작성되었다.

FS-MP3DK

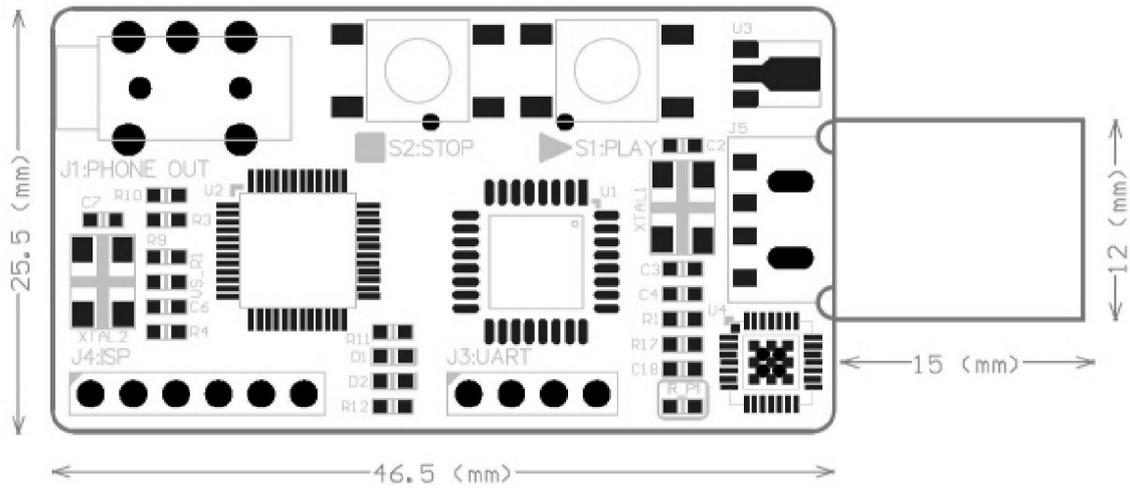
| Parameter | Min. | Typ. | Max. | Unit | Condition / Note |
|---------------------------------|------|------|------|------|----------------------|
| Supply voltage for regulator | 3 | | 12 | V | Input Port J3:Power |
| Operating Temperature | -40 | | 85 | °C | |
| Microphone Load Resistance | 16 | 30 | | Ω | |
| Microphone Load Capacitance | | | 100 | pF | |
| Microphone input amplifier gain | | 26 | | dB | |
| Microphone S/N Ratio | 50 | 68 | | dB | |
| Play Current Consumption | 42 | 45 | 47 | mA | 5V EXT. Power ① |
| Stop Current Consumption | 13 | 14 | 15 | mA | 5V EXT. Power ① |
| Power Down Current | | 100 | 115 | uA | 5V EXT. Power ① |
| USB Current Consumption | 16 | 18 | 19 | mA | Only USB Connected ② |

① J3: USART 포트에 DC 5V 를 공급할 때 (R_P1 제거) 소모 전류이고, USB포트로 연결되면 USB 소모전류 (약 18mA)를 더해야 한다.

② USB포트로 PC와 연결되었을 때 USB IC (CP2102)가 기본적으로 일정 전류를 소모한다.

11. 기구적 특성

【 모듈 치수 】



【 모듈 무게 】

8g (SD메모리 제외)



12. 메뉴얼 정보

주소: 서울 마포구 토정도 6-97 성안빌딩 3층 펌시스

URL: www.firmsys.com

전화: 02 943 2208

팩스: 02 6008 7475

E-mail

구입문의: info@firmsys.com

기술문의: min@firmsys.com

본 사용자 설명서는 별도의 공지 없이 수정 보완될 수 있습니다.