

## < SD-FLASH WRITER 사용설명서 >

모델명 : SFW01

### 제품특징.

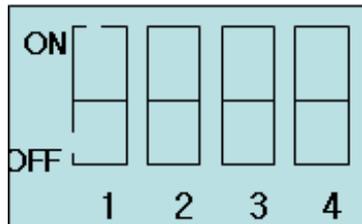
- 단일 5V 전원.
- TTL - SCI 통신을 이용
- FAT16 지원, 최대 SD-FLASH 용량 2G byte 까지 지원
- 1byte로 단위에서 최대 512byte 씩 저장가능
- 응답 메시지에 따른 MCU 모드 및 Terminal 모드 지원
- DIP Switch을 이용하여 Baud 및 MCU 모드 설정  
(Baud : 2400~230K Bps 설정가능)
- 간단한 serial command을 이용하여 SD-FALSH에 데이터 write 및 read 기능
- LED 로 SD-FLASH의 상태 표시 (TXD, RXD, CARD 삽입여부, POWER)
- Compact SIZE



### DIP SWITCH 에 따른 설정

SW1~3번까지는 BPS를 설정할 수 있으며, SW4는 MODE를 설정할 수 있습니다.

(BPS는 전원인가 후 1번만 체크하므로, 파워 리셋 후에 변경된 BPS로 적용됩니다.)



BPS	SW1	SW2	SW3
2400	OFF	OFF	OFF
4800	ON	OFF	OFF
9600	OFF	ON	OFF
19200	ON	ON	OFF
38400	OFF	OFF	ON
57600	ON	OFF	ON
115000	OFF	ON	ON
230000	ON	ON	ON

SW4	MODE 설정
ON	MICOM MODE
OFF	Terminal Mode

## 통신 Command

- 기본 프로토콜은 DOS 명령어를 기본으로 정했습니다.
- 기본적으로 file에 Data를 쓰고 읽을 때, file을 Open 해야 합니다. 또한 open 상태의 옵션에 따라 파일의 상태를 지정할 수가 있는데, /a 인 경우 파일 이어쓰기 (APPEND), /w 인 경우 파일 덮어쓰기 (write), /r 인 경우 파일 읽기 (READ)로 구성되어 있습니다.
- ., SFW01의 모듈의 모드에 따라 응답 메시지가 다릅니다. 즉 Terminal 모드일 때에는 서술적으로 모듈의 상태를 string으로 알려주며, MCU 모드 일 때에는 간략한 메시지로 (2byte 형태) 응답합니다.

### 부호설명

[CR] : Carrige return(0x0d) [LF] : Line Field(0x0a) ^ : space (0x20)

## 기본 명령어

- **init[CR] : SD CARD 초기화**

```
-----
> init
Initialize OK
>
> -
```

- **dir[CR] : 디렉토리 및 파일보기**

```
-----
> dir
File Listing for: ROOT###
  DEMO.TXT      [48] bytes   (2)
  HANGUL.TXT    [177] bytes  (3)
>
```

- **fsize^[파일명][CR] : 파일 size을 표시**

```
-----
File Listing for: ROOT###
  DEMO.TXT      [48] bytes   (2)
  HANGUL.TXT    [177] bytes  (3)
> fsize hangul.txt
File Size:177 byte
```

- **dize[CR] : SD-CARD의 전체 용량을 표시**

```
-----
File Listing for: ROOT###
  DEMO.TXT      [48] bytes   (2)
  HANGUL.TXT    [177] bytes  (3)
> dsize
Total Size:245919 Kbyte
```

- **del[파일명][CR] : 파일 삭제**

```

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    TEST.TXT      [0] bytes      (0)
> del test.txt
File deleted
> dir

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    
```

- **fcreate^[파일명][CR] : 파일 생성**

```

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
> fcreate test.txt
Create file OK
> dir

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    TEST.TXT      [0] bytes      (0)
    
```

- **rename^[현재 파일명]^[변경할 파일명][CR] : 파일 이름 변경**

```

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    TEST.TXT      [0] bytes      (0)
> rename test.txt lim.txt
Rename ok
> dir

File Listing for:  ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    LIM.TXT [0] bytes      (0)
    
```

- **reset[CR] : SD-CARD를 reset 한다. (SD 안전제거)**

```

> reset
Reset OK!
    
```

- **card[CR] : SD-CARD 장착유무 확인.**

```

> card
Card Insert!
    
```

- **format[CR] : SD-CARD를 포맷.**

```

> format
format OK
    
```

- **help[CR] : Command에 대한 도움말을 표시.**

```

CD <directory><CR> : Directory Change
dir<CR> : View Directory&file
fsize <file><CR> : file size
dsize <file><CR> : SD-card Size
ftime <file><CR> : Information File Time
md <directory><CR> : Make Directory
rd <directory><CR> : Remove Directory
del <file><CR> : Delete File
fcreate <file><CR> : Make File
rename <old_file> <new_file><CR> : Change File Name
fopen <file> <option><CR>: File Open
    Option=> /a : File Append, /w : File write, /r : File Read
fclose<CR> : File Close
fputc <file> <option> <1byte_data><CR>: 1byte Data Put
fputs <file> <option> <string_data><CR>: Max 256byte string Data Put
fputs2<CR> : Max 512byte Custimize Data Puts
fwrite </><save_number><CR> : Max 512byte Packet Data put
fgetc </><read_number><CR> : max 512 byte packet Data Read
fread <file><CR> : file all Read
reset<CR> : SD-Card Reset
card<CR> : Card Insert Inform
format<CR> : SD-Card Format
fdemo<CR> : Demo file create
    
```

- **fdemo[CR] : Demo file 생성**

```

> dir
File Listing for: ROOT###
> fdemo
Demo file Created!!
> dir
File Listing for: ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    
```

## 디렉토리 명령어

- . SFW은 기본적으로 디렉토리를 생성, 삭제, 이동 등을 설정할 수 있습니다.

- **md^[디렉토리 명][CR] : 디렉토리 생성**

```

File Listing for: ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
> md folder
Make Directory
> dir
File Listing for: ROOT###
    DEMO.TXT      [48] bytes      (2)
    HANGUL.TXT    [177] bytes     (3)
    [FOLDER]      (4)
    
```

- `cd ^[디렉토리 경로][CR]` : 디렉토리를 경로를 바꿉니다.

ex. folder 디렉토리로 이동

```
File Listing for: ROOT###
  DEMO.TXT      [48] bytes      (2)
  HANGUL.TXT    [177] bytes     (3)
  [FOLDER]     (4)
|> cd folder
|Changing Directory OK
|> dir
|
|File Listing for: ROOT###FOLDER#
|> =
```

- `cd^W[CR]` : 디렉토리를 초기 위치로 이동

```
File Listing for: ROOT###FOLDER#
> cd #
Changing Directory OK
> dir
.
File Listing for: ROOT###
  DEMO.TXT      [48] bytes      (2)
  HANGUL.TXT    [177] bytes     (3)
  [FOLDER]     (4)
>
```

## File open 및 close 명령어

- . fdemo command을 이용하면 sd-flash에 hangul.txt와 demo.txt 파일이 생성됩니다.

```
.File Listing for: ROOT###
> fdemo
Demo file Created!!
> dir
.
File Listing for: ROOT###
  DEMO.TXT      [48] bytes      (2)
  HANGUL.TXT    [177] bytes     (3)
  [FOLDER]     (4)
>
```

기본적으로 생성된 파일에 데이터를 읽고, 쓰기 할 때에는 File이 Open 상태이어야 합니다. 또한, 선택된 파일에 대해 작업이 끝날 때에는 반드시 File Close 해야 합니다.

- `fopen^[파일명]^[/옵션][CR]` : 파일 open

(sp. 파일을 읽고 쓸 때, 특정 command에 대해서는 파일이 오픈 상태이어야 합니다.)

옵션. /a 파일 이어쓰기 준비상태(append)

```
> fopen demo.txt /a
Open OK
> =
```

/w 기존 파일의 데이터를 새로 쓰기 준비상태(write)

```
> fopen demo.txt /w
Open OK
>
```

/r 파일 읽어 오기 (read) (크기가 0 byte인 파일은 열 수 없습니다.)

```
> fopen demo.txt /r
Open OK
```

- `fclose[CR]` : 파일 close

- 열린 파일을 close 해야 다른 파일에 대해 Access가 가능합니다.

```
> fclose
File Close OK
```

## Data 저장 및 읽기 기능

- `fputc^[파일명]^[/옵션]^[1byte 데이터][CR]` : 파일에 1byte Data를 저장.

옵션 : /w 파일 덮어쓰기 (write) test.c 에 'A' 문자를 새로 덮어서 저장

```
> fputc test.c /w A
Put character OK
```

/a 파일 이어쓰기 (append) test.c 에 'A' 문자를 이어서 저장

```
> fputc test.c /a A
Put character OK
```

- `fputs^[파일명]^[/옵션]^[string 데이터][CR]`

: 파일에 [CR]이 들어올 때 까지 최대 256byte을 저장

옵션 : /w 파일 덮어쓰기(write) test.c 에 "12345" 문자열을 새로 덮어서 저장

```
> fputs test.c /w 12345
Put String OK
```

/a 파일 이어쓰기 (append) test.c 에 "abcdefg" 문자열을 이어서 저장

```
> fputs test.c /a abcdefg
Put String OK
```

- `fputs2[CR]` : 0x7E(~)가 들어 올 때 까지 최대 512byte까지 데이터를 저장합니다.

(파일이 open 상태이어야 하며, /a , /w의 옵션 상태이어야 하며 저장이 끝난 상태에서는 fclose을 이용하여 파일을 닫습니다.)

ps. 문자열 저장 시작은 MICOM인 경우 ~로 저장 시작신호가 들어옵니다.

ex. test.c 에 “~”문자가 들어 올 때 까지 저장. 즉, “123456” 저장

```

-----
> fopen test.c /w
Open OK
> fputs2
.
Write to file: Press [~] to end 'Write' mode
String Put Success
> fclose
File Close OK
-----
    
```

- **fwrite^[/저장할 byte 수][CR]** : 데이터 packet 단위 (최대 512byte)로 저장.

*(파일이 open 상태이어야 하며, /a , /w의 옵션 상태이어야 하며 저장이 끝난 상태에서는 fclose을 이용하여 파일을 닫습니다.)*

ex. test.c 에 10byte packet data 저장

```

-----
> fopen test.c /a
Open OK
> fwrite /10
Packet 10 bytes write OK> 1234567890
> fclose
File Close OK
-----
    
```

- **fgetc^[/read할 byte 수][CR]** : 데이터의 개수를 정하여 읽어 옵니다.

*(파일이 open 상태이어야 하며, /r 의 옵션 상태이어야 하며 저장이 끝난 상태에서는 fclose을 이용하여 파일을 닫습니다..)*

ex. test.c 의 내용을 5byte, 7byte 읽기

```

-----
> fopen demo.txt /r
Open OK
> fgetc /5
1.123
> fgetc /7
4567890
> fclose
File Close OK
-----
    
```

- **fgets[CR]** : 저장된 데이터가 [CR]값이 올 때 까지 읽어 옵니다.

*(파일이 open 상태이어야 하며, /r 의 옵션 상태이어야 하며 저장이 끝난 상태에서는 fclose을 이용하여 파일을 닫습니다.)*

ex. test.c 의 내용 중 <CR>이 있을 때 까지 데이터를 읽어옵니다.

```

-----
> fopen demo.txt /r
Open OK
> fgets
1.1234567890
> fclose
File Close OK
-----
    
```

- **fread^[파일명][CR]** : 파일에 저장된 데이터를 모두 읽어옵니다.

```
> fread hangul.txt

안녕하세요!!리얼시스 입니다..
SD Flash writer를 구매해 주셔서 감사합니다.
문의 사항: Tel 031-420-4326
즐거운 하루 되세요...
```

### MICOM 모드의 Error 메시지

MICOM MODE Error Code	
E0	기능에 없는 명령어 입니다.
E1	옵션 처리가 잘못 되었습니다.
E2	파일이 Open 되어 있지 않습니다. fopen 명령어를 실행하세요
E3	파일이 Open 되어 있습니다. fclose 명령어를 실행하세요.
E4	Data의 Size가 잘못 되었습니다.
Command 수행에 따른 message	
NG	명령 수행이 올바르지 않습니다. (No Good)
OK	명령이 수행되었습니다. (Good)

### Size (MM)

