

組込システム向けFATファイルシステム

SAVER60 *for FAT*

RAM ディスクドライバ

ユーザーズ・マニュアル

第1.02版(平成 17 年 2 月 10 日)

はじめに

この度は、「SAVER60 for FAT」をご購入いただきありがとうございます。

本マニュアルは、FAT ファイルシステム「SAVER60 for FAT」に付属しているサンプル用 RAM ディスクドライバについて記述したものです。

なお、本マニュアルは、RTOS(Real Time OS)及び C 言語に関する一般的知識をもっておられる事を前提としています。

参考資料: I/O ドライバ作成マニュアル

SAVER60 for FAT ユーザーズ・マニュアル

本製品に付属しています RAM ディスクドライバはサンプルとしてご提供させていただくものです。(以下、サンプルドライバといいます)。

本サンプルドライバのご使用に関しては、以下の点にご注意ください。

- (1)本サンプルドライバは、保守サービスの対象外ですので、あらかじめご了承ください。
- (2)弊社は、本サンプルドライバの品質ならびに性能の保証、商品価値の有無、特定の目的下での利用可能性、第三者の所有する知的所有権、その他の権利の非侵害保証を含む、いっさいの保証をいたしません。
- (3)本サンプルドライバのご使用に起因して、貴社ならびに本サンプルドライバの使用者に生じうるいっさいの損害について、賠償責任を負いません。

ご注意

- (1) 本マニュアルの一部、またはすべてを無断で転載複製する事は、固くお断りします。
- (2) 本マニュアルに記載されている製品、及び製品の仕様につきましては、製品改良その他により予告無しに変更することがあります。
- (3) 本製品の運用結果につきましては、責任を負いかねますのでご了承下さい。
- (4) 本マニュアルの内容に関し、お気づきの点などございましたら、弊社担当者にご連絡いただければ、幸いです。

本マニュアルに記載されている会社名及び製品名は、各社の商標または登録商標です。

改訂履歴

版番号	改定日	改訂内容
01.00	2004/08/02	初版
01.01	2004/10/14	誤記訂正
01.02	2005/02/10	サンプルドライバに関する制限事項 追加 4-2.アプリケーションからの利用方法 削除

目次

1.	概要.....	1
1-1.	特徴.....	1
1-2.	仕様.....	1
2.	構成.....	2
3.	A P I	3
3-1.	呼び出し形式.....	3
3-2.	初期化(I/O ファンクション番号 = 0x00)	4
3-3.	マウント(I/O ファンクション番号 = 0x01)	5
3-4.	アンマウント(I/O ファンクション番号 = 0x02)	6
3-5.	チャンネル情報取得(I/O ファンクション番号 = 0x03)	7
3-6.	デバイス情報取得(I/O ファンクション番号 = 0x04)	8
3-7.	ユニット情報取得(I/O ファンクション番号 = 0x05)	9
3-8.	データ読み込み(I/O ファンクション番号 = 0x06)	10
3-9.	データ書き込み(I/O ファンクション番号 = 0x07)	11
3-10.	データ消去(I/O ファンクション番号 = 0x08)	12
3-11.	メディア挿抜チェック(I/O ファンクション番号 = 0x09)	13
4.	利用方法.....	14
4-1.	FATファイルシステム部への組込方法.....	14
4-2.	ソースファイル構成.....	16
5.	エラーコード.....	17

1. 概要

1-1. 特徴

本 RAM ディスクドライバは、SAVER60 for FAT 上で動作する、チャンネル単位でリエントラントな構造をもつドライバです。

基本機能は以下の通りです。

▶指定された RAM 領域を固定長サイズのセクタに分割して、セクタ単位の入出力を実現します。

1-2. 仕様

(1)ドライバ ID

0x03

(2)セクタサイズ

512バイト固定長。

(3)チャンネル、デバイス、ユニット

チャンネル数 = 1、チャンネル番号 = 0x00固定

デバイス数 = 1、デバイス番号 = 0x00固定

ユニット数 = 1、ユニット番号 = 0x00固定

なお、これらの値は、RAM_Def.h に最大値として定義してあります。

(4)パーティション

パーティションはサポートしていません。従って、本ドライバでは、I/O ファンクション番号 = 0x0a:「パーティション作成」はサポートしていません。

ファイルシステム API のマウント(FMR_Mount)では、MBR 無しを指定してください。

本ドライバは、RAM ディスク先頭から FAT ファイルシステムを展開します。

(5)RAM 領域定義

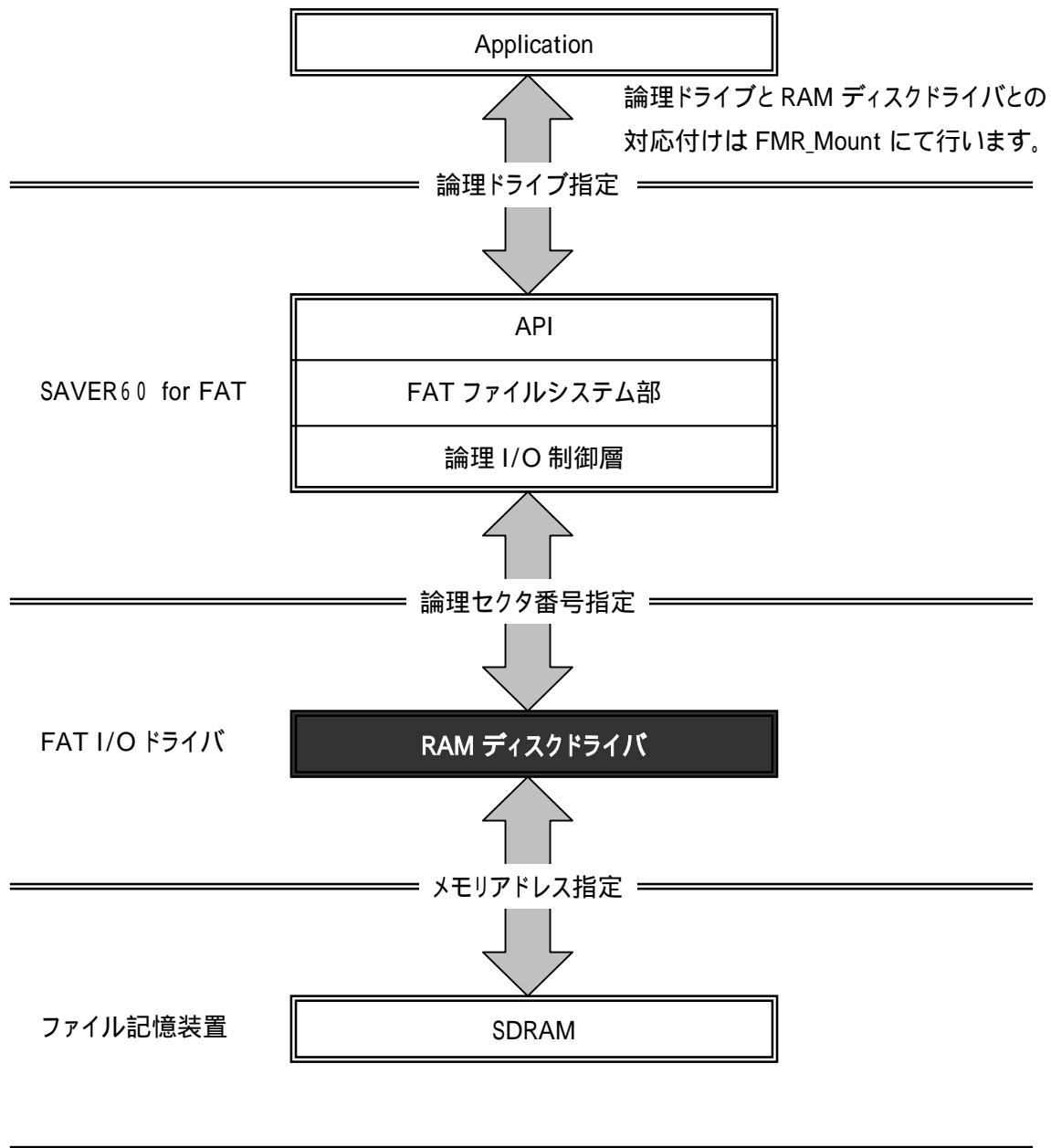
RAM ディスクとして使用する RAM 領域は、スタートアドレスとサイズを RAM_hw.h ファイルに定義してください。

その際、スタートアドレスは CPU のメモリバウンダリにあわせてください。またサイズは512バイトの整数倍を指定してください。

(6)スタック領域に関して

本ドライバで使用するスタック領域は、上位アプリケーション(タスク)のスタック領域を使用するものとします。

2. 構成



3. API

3-1. 呼び出し形式

本ドライバは、ドライバ構成情報のドライバエントリアドレスを元に FAT ファイルシステム部より、下記の呼出し形式で起動されます。

・呼出し形式

```
int   ドライバ構成情報・ドライバエントリ(
      unsigned char チャンネル番号/ユニット番号,
      unsigned char I/O ファンクション番号,
      struct IO パラメータ * I/O パラメータ格納領域)
```

・引数

IN: unsigned char チャンネル番号/ユニット番号

論理ドライブが接続されているチャンネル番号、ユニット番号を
上位 4 ビット=チャンネル番号、下位 4 ビット=ユニット番号で指定します。
本ドライバでは、チャンネル番号、ユニット番号とも、0 固定です。

IN: unsigned char I/O ファンクション番号

ドライバに要求する機能を指定します。

I/O ファンク ション番号	機 能	I/O ファンク ション番号	機 能
0x00	初期化	0x06	データ読込み
0x01	マウント	0x07	データ書込み
0x02	アンマウント	0x08	データ消去
0x03	チャンネル情報取得	0x09	メディア挿抜チェック
0x04	デバイス情報取得		
0x05	ユニット情報取得		

IN: struct IO パラメータ * I/O パラメータ格納領域

I/O ファンクション番号に対応したそれぞれの I/O パラメータを指定します。
詳細は、各ファンクションの説明に記述します。

・戻り値

0=正常終了

負の値=エラーコード

3-2. 初期化(I/O ファンクション番号 = 0x00)

機能

RAM ディスクドライバを初期化します。

I/Oパラメータ

```
struct {  
    unsigned char    予備[4]  
    unsigned short   予備[2]  
    unsigned long     予備  
    unsigned long     予備  
} T_INIT_PARAM;
```

解説

本ドライバ内部の初期化を行う為に、FATファイルシステムの初期化時に呼出されます。

その他

3-3. マウント(I/O ファンクション番号 = 0x01)

機能

RAM ディスクドライバをマウントします。

I/Oパラメータ

```
struct {
    unsigned char    MBRモード
    unsigned char    パーティションタイプ
    unsigned char    セクタサイズ
    unsigned char    予備[1]
    unsigned short   予備[2]
    unsigned long    ユニット全体のセクタ数
    unsigned long    ユニット開始セクタ番号
} T_MOUNT_PARAM;
```

解説

RAMディスクをマウント済みとし、以降、各種I/O要求を受付可能とし、RAMディスクに関する情報をマウントI/Oパラメータにセットします。

その他

*1 T_MOUNT_PARAM

項 目	説 明
MBR モード	MBR(MasterBootRecord)の有無を示します。 本ドライバでは、"01H=MBR 無し"を必ず指定します。
パーティションタイプ	本ドライバでは不定です。
セクタサイズ	セクタサイズを 512 バイト単位で本ドライバが返します。 本ドライバでは、常に"1"を返します。
ユニット全体のセクタ数	RAMディスクの総セクタ数を本ドライバが返します。
ユニット開始セクタ番号	本ドライバでは、常に"0"を返します。

3-4. アンマウント(I/O ファンクション番号 = 0x02)

機能

RAM ディスクドライバをアンマウントします。

I/Oパラメータ

```
struct {  
    unsigned char    予備[4]  
    unsigned short   予備[2]  
    unsigned long     予備  
    unsigned long     予備  
} T_UNMOUNT_PARAM;
```

解説

RAMディスクをアンマウント状態とします。以降、本ドライバでは、RAMディスクに対する、マウント要求以外のI/O要求を受付不可とします。

その他

3-5. チャネル情報取得(I/O ファンクション番号 = 0x03)

機能

チャネル情報を取得します。

I/Oパラメータ

```
struct {  
    unsigned char    予備[4]  
    unsigned short   予備[2]  
    unsigned long     予備  
    unsigned long     予備  
} T_CHANEL_INFO_PARAM;
```

解説

チャネル情報を取得します。現在、FATファイルシステム部では、本ファンクションは使用していません。

その他

3-6. デバイス情報取得 (I/O ファンクション番号 = 0x04)

機能

デバイス情報を取得します。

I/Oパラメータ

```
struct {  
    unsigned short    1トラック当りのセクタ数  
    unsigned short    シリンダ数  
    unsigned short    ヘッド数  
    unsigned short    予備  
    unsigned long     合計セクタ数  
    unsigned long     予備  
} T_DEVICE_INFO_PARAM;
```

解説

デバイス情報を取得します。

その他

*1 T_DEVICE_INFO_PARAM

項 目	説 明
1トラック当りのセクタ数	本ドライバでは、常に“0”を返します。
シリンダ数	本ドライバでは、常に“0”を返します。
ヘッド数	本ドライバでは、常に“0”を返します。
合計セクタ数	本ドライバでは、常に“0”を返します。

3-7. ユニット情報取得(I/O ファンクション番号 = 0x05)

機能

ユニット情報を取得します。

I/Oパラメータ

```
struct {
    unsigned char    パーティションタイプ
    unsigned char    セクタサイズ
    unsigned char    ユニット状態
    unsigned char    予備[1]
    unsigned short   予備[2]
    unsigned long    ユニット全体のセクタ数
    unsigned long    ユニット開始セクタ番号
} T_UNIT_INFO_PARAM;
```

解説

ユニット情報を取得します。

その他

*1 T_UNIT_INFO_PARAM

項 目	説 明
パーティションタイプ	本ドライバでは不定です。
セクタサイズ	セクタサイズを 512 バイト単位で本ドライバが返します。 本ドライバでは、常に“1”を返します。
ユニット状態	ユニット状態を本ドライバが返します。 Bit0: マウント状態 0=未マウント、1=マウント済み
ユニット全体のセクタ数	R A M ディスクの総セクタ数を本ドライバが返します。
ユニット開始セクタ番号	本ドライバでは、常に“0”を返します。

3-8. データ読み込み(I/O ファンクション番号 = 0x06)

機能

データを読み込みます。

I/Oパラメータ

```
struct {  
    unsigned long    読み込み開始セクタ番号  
    unsigned short * 読み込みセクタ数  
    unsigned char *  読み込みデータ格納領域  
    unsigned long    予備  
} T_READ_PARAM;
```

解説

データ読み込みI/Oパラメータに従って、RAMディスクよりデータを読み込みます。

その他

*1 T_READ_PARAM

項 目	説 明
読み込み開始セクタ番号	読み込みを開始する論理セクタ番号を指定します。
読み込みセクタ数	読み込みセクタ数を格納した領域のアドレスを指定します。 正常終了時、実際に読み込んだセクタ数を本ドライバがセットします。
読み込みデータ格納領域	読み込みデータを格納する領域のアドレスを指定します。

3-9. データ書込み(I/O ファンクション番号 = 0x07)

機能

データを書込みます。

I/Oパラメータ

```
struct {  
    unsigned long    書込み開始セクタ番号  
    unsigned short * 書込みセクタ数  
    unsigned char *  書込みデータ格納領域  
    unsigned long    予備  
} T_WRITE_PARAM;
```

解説

データ書込みI/Oパラメータに従って、RAMディスクにデータを書込みます。

その他

*1 T_WRITE_PARAM

項 目	説 明
書込み開始セクタ番号	書込みを開始する論理セクタ番号を指定します。
書込みセクタ数	書込みセクタ数を格納した領域のアドレスを指定します。 正常終了時、実際に書込めたセクタ数を本ドライバがセットします。
書込みデータ格納領域	書込みデータが格納された領域のアドレスを指定します。

3-10. データ消去(I/O ファンクション番号 = 0x08)

機能

データを消去します。

I/Oパラメータ

```
struct {  
    unsigned long    消去開始セクタ番号  
    unsigned short * 消去セクタ数  
    unsigned char    消去データ  
    unsigned char    予備[3]  
    unsigned long    予備  
} T_ERASE_PARAM;
```

解説

データ消去I/Oパラメータに従って、RAMディスクに消去データを書込みます。

その他

*1 T_ERASE_PARAM

項 目	説 明
消去開始セクタ番号	消去を開始する論理セクタ番号を指定します。
消去セクタ数	消去セクタ数を格納した領域のアドレスを指定します。 正常終了時、実際に消去できたセクタ数を本ドライバがセ ットします。
消去データ	消去データを指定します。

3-11. メディア挿抜チェック(I/O ファンクション番号 = 0x09)

機能

メディアの挿抜状態を取得します。

I/Oパラメータ

```
struct {  
    unsigned char    挿抜状態  
    unsigned char    予備[3]  
    unsigned short   予備[2]  
    unsigned long    予備  
    unsigned long    予備  
} T_DETECT_INFO_PARAM;
```

解説

メディアの挿抜状態を取得します。

その他

*1 T_DETECT_INFO_PARAM

項 目	説 明
挿抜状態	メディアの挿抜状態を本ドライバが返します。 本ドライバでは、常に“0x00:同一メディアが挿入されている”を返します。

4. 利用方法

4-1. FATファイルシステム部への組込方法

(1)FAT ファイルシステム部の FMR_DriverConf.h に下記のドライバ構成情報を定義してください。

(2)ドライバ構成情報:TFMR_DRVINFO

ドライバ構成情報は、FAT ファイルシステム部で使用するドライバに関する情報を格納します。

FMR_DriverConf.h は、ファイルシステム共通RAM領域定義ファイル(FMR_ram.c)にてインクルードされます。

データ型	項 目	内 容
unsigned char	driverid	ドライバIDを定義します。 0x03=RAMドライバを指定します。
unsigned char	channelnum	本I/Oドライバで制御するチャンネル最大数を定義します。 0x01を指定します。
unsigned char	mediatype	メディアタイプを定義します。 0xf8=non-removable mediaを指定します。
int	(* drventry) (unsigned char, unsigned char, void *)	本I/Oドライバのドライバエントリアドレスを定義します。 RAM_mainを指定します。
unsigned short	semaphoreid [DFMR_USR_MAXCHANNEL]	セマフォIDを定義します。 ID番号自動割付け機能を使用しない場合、使用するセマフォのIDを定義します。 DFMR_USR_MAXCHANNEL:FMR_User.hで定義される値で、FATファイルシステム部に組み込まれるドライバの中での最大チャンネル数を意味します。この値が1以上の場合は、残りのセマフォIDは、本ドライバでは使用しませんので、任意の値を指定してください。

(3)RAM 領域定義

RAM_hw.h 内に下記 define 値を定義してください

```
#define RAM_START    RAM ディスク開始メモリアドレス  
#define RAM_SIZE     RAM ディスクサイズ
```

*1 RAM ディスク開始メモリアドレスは必ず、CPU メモリバウンダリに合せてください。

4 バイトバウンダリの場合は、4 の倍数値を指定してください。

*2 RAM ディスクサイズは必ず、512 の倍数値を指定してください。

(4)O/S 依存部の修正

RAM_OS_inh.c 内の下記関数は、O/S に依存した内容になっていますので、使用する O/S に応じて、修正してください。

・RAM_osGetTID: 自タスク ID の取得

(5)本ドライバをリコンパイルし、ライブラリの再構築を行います。

RAM ディスクドライブライブラリ: LIBRAM.LIB

リコンパイル及びライブラリ構築は、付属している makefile を参考に行ってください。

4-2. ソースファイル構成

・RAM_main.c	API I/F 処理
・RAM_io.c	I/O 制御部
・RAM_OS_inh.c	OS call ルーチン
・RAM_Sub.c	共通関数
・RAM_Ram.c	グローバル変数の定義
・RAM_hw.h	RAM ディスク領域定義
・RAM_Def.h	定数定義
・RAM_Prot.h	プロタイプ宣言 & 変数領域宣言
・RAM_Struct.h	構造体定義
・Makefile	メイクファイル

5. エラーコード

ラベル名	エラー内容
DFMR_SUC	正常終了
DFMR_DRVERR_PARAM	パラメタエラー
DFMR_DRVERR_NOTINIT	未初期化エラー
DFMR_DRVERR_UNITOVER	該当ユニット無し
DFMR_DRVERR_NOTMOUNT	未マウントエラー