

組込システム向けFATファイルシステム

SAVER60 *for FAT*

I/Oドライバ
作成マニュアル

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」と共に動作する I/O ドライバの作成方法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:PCATA カードドライバ ユーザーズ・マニュアル
RAM ディスクドライバ ユーザーズ・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各種 I/O 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	3-12.パーティション作成 追加
01.02	2005/02/10	サンプルドライバに関する制限事項 追加

目次

1. 概要	1
1-1. 特徴	1
1-2. 動作概要	2
2. 論理構造	3
2-1. ファイル記憶装置構成定義	3
2-2. リエントラント構造	7
2-3. ドライバ構成情報	9
2-4. MBR 構造	10
3. API 関数	11
3-1. 呼出し形式	11
3-2. 初期化 (I/O ファンクション番号 = 0x00)	12
3-3. マウント (I/O ファンクション番号 = 0x01)	13
3-4. アンマウント (I/O ファンクション番号 = 0x02)	14
3-5. チャンネル情報取得 (I/O ファンクション番号 = 0x03)	15
3-6. デバイス情報取得 (I/O ファンクション番号 = 0x04)	16
3-7. ユニット情報取得 (I/O ファンクション番号 = 0x05)	17
3-8. データ読み込み (I/O ファンクション番号 = 0x06)	18
3-9. データ書き込み (I/O ファンクション番号 = 0x07)	19
3-10. データ消去 (I/O ファンクション番号 = 0x08)	20
3-11. メディア挿抜チェック (I/O ファンクション番号 = 0x09)	21
3-12. パーティション作成 (I/O ファンクション番号 = 0x0a)	22
4. エラーコード	23

1. 概要

1-1. 特徴

I/O ドライバは FAT ファイルシステムの下位に位置するもので、FAT ファイルシステムからのセクタ番号指定の I/O 要求を、ファイル記憶装置への物理的な I/O 要求に変換して出力します。

例えば、PCATA カードドライバの場合、LBA(Logical Block Address)形式の物理セクタ単位の ATA コマンドに変換し、PCMCIA コントローラを通じて、PCATA カードに出力します。

I/O ドライバは、大別して、API 処理、H/W I/F 処理、割込み処理、O/S I/F 処理 からなります。

▶API 処理

上位 FAT ファイルシステムからの各種 I/O 要求を、物理的な I/O 要求に変換し H/W I/F 処理を通して、ファイル記憶装置に出力します。

▶H/W I/F 部

ファイル記憶装置への物理的な I/O 要求を行います。

▶割込み処理

ファイル記憶装置からの割り込みを受付、I/O 要求完了等の判断を行います。

▶O/S I/F 処理

割込み処理登録、I/O 処理完了待ち時のタスクデレイ等の機能を O/S へ要求します。

また、I/O ドライバは、リエントラント構造を必須とし、I/O ドライバ1本のみで、同一種類の複数のファイル記憶装置を制御できるものとします。

I/O ドライバに対する排他制御は、FAT ファイルシステム側で行われるものとします。

1-2. 動作概要

I/O ドライバは、FAT ファイルシステムからの I/O 要求により起動されます。

この I/O 要求により、I/O ドライバの管理しているチャンネル番号/デバイス番号、又はユニット番号、I/O ファンクション番号、I/O パラメータが I/O ドライバに引き渡されます。

I/O ドライバでは、これらの情報を基に、物理的な I/O 要求を実行し、実行完了をもって、呼出し元へ戻ります。

FAT ファイルシステムの初期化時、I/O ドライバに対し、初期化要求を FAT ファイルシステムが発行します。

FAT ファイルシステムにて、論理ドライブとファイル記憶装置の対応を取る為に、I/O ドライバに対し、マウント要求を発行します。

I/O ドライバでは、このマウント要求により、当該ファイル記憶装置をアクティブとします。

(PCATA カードドライバでは、この時点で、PC カードの挿入チェックや電源 ON 処理を行います。)

以降、I/O ドライバでは、当該ファイル記憶装置に対する、読み込み要求、書き込み要求等の I/O 要求を有効とします。

FAT ファイルシステムでは、マウント完了後、読み込み要求、書き込み要求、ドライバ状態取得要求等を必要に応じて発行します。

FAT ファイルシステムにて、当該ファイル記憶装置を使用しなくなった場合、I/O ドライバに対しアンマウント要求を発行します。

I/O ドライバでは、このアンマウント要求により、当該ファイル記憶装置を非アクティブにします。(PCATA カードドライバでは、この時点で、電源 OFF 処理を行います。)

以降、I/O ドライバでは、当該ファイル記憶装置に対する、読み込み要求、書き込み要求等の I/O 要求を無効とします。

2. 論理構造

2-1. ファイル記憶装置構成定義

I/O ドライバで管理するファイル記憶装置の構成を下記のように定義します。

- ・メディア: I/O ドライバにて、データを実際に格納するファイル記憶装置の種類をメディアと称します。

メディアとしては、ハードディスク、PC カード、RAM ディスク等があります。

またメディア毎にそれぞれ専用のドライバが必要となります。

但し、I/O ドライバは、同一メディアであれば複数のメディアを 1 本でサポートするものとします。

- ・チャンネル: 各メディア内で同時並行動作可能な単位をチャンネルと称します。

I/O ドライバでは、ファイル記憶装置のコントローラをチャンネルとします。

また I/O 要求の排他制御は、このチャンネル単位で行います。

このチャンネル単位の排他制御は、FAT ファイルシステム側で行うものとし、I/O ドライバでは排他制御は行いません。

但し、I/O ドライバでは、複数コントローラのサポートを必須とし、I/O ドライバはチャンネル単位で、リエントラント可能な構造とします。

例えば、PCMCIA コントローラが 2 つ搭載されている場合、I/O ドライバではチャンネル数 = 2 として動作し、チャンネル単位の同時並行動作可能とします。

チャンネル数はハードウェア構成に依存します。

- ・チャンネル番号: チャンネルを識別する為の番号です。

ドライバ単位の 0 ~ 15 までの連続した正の整数。

チャンネル番号と対応するコントローラのポートアドレス等の対応は、I/O ドライバ内で対応付ける事とします。

またチャンネル番号と実際のファイル記憶装置との対応は、FAT ファイルシステムの使用者側にも明示する必要が有ります。

- ・デバイス: 各チャンネルに接続される記憶装置をデバイスと称します。

コントローラに物理的に接続されているファイル記憶装置そのものを、デバイスとします。

同一チャンネルに接続された各デバイスは、同時並行動作は出来ないものとします。

従って、1チャンネルに複数デバイスが接続されている場合は、その中のいずれか一つのデバイスしか動作しないことになります。

例えば、PCMCIA コントローラ 1 つに対して、PC-ATA カードが 1 つ接続される場合、デバイス数は 1 となります。

デバイス数はハードウェア構成に依存します。

- ・デバイス番号: デバイスを識別する為の番号です。

チャンネル単位の 0 ~ 15 までの連続した正の整数。

デバイス番号と実際のファイル記憶装置との対応は、FAT ファイルシステムの使用者側にも明示する必要が有ります。

- ・ユニット: デバイス内の記憶領域をユニットと称します。

デバイス内がパーティションに分割される場合、パーティション単位をユニットとします。

ユニット数はハードウェア構成に依存します。

- ・ユニット番号: ユニットの識別する為の番号です。

チャンネル単位の 0 ~ 15 までの連続した正の整数。

デバイス単位ではなく、チャンネル単位である点に注意して下さい。

デバイス番号とユニット番号の対応付けは I/O ドライバ内で行う事とします。

ユニット番号と実際のファイル記憶装置との対応は、FAT ファイルシステムの使用者側にも明示する必要が有ります。

例えば、ハードディスクの場合、MBR(Master Boot Record)のパーティションテーブルに定義してある順序をユニット番号とします。

- ・論理セクタ番号: FAT ファイルシステム側で使用するユニット内のセクタを識別する為の番号です。

ユニット先頭セクタを 0 とする正の整数。

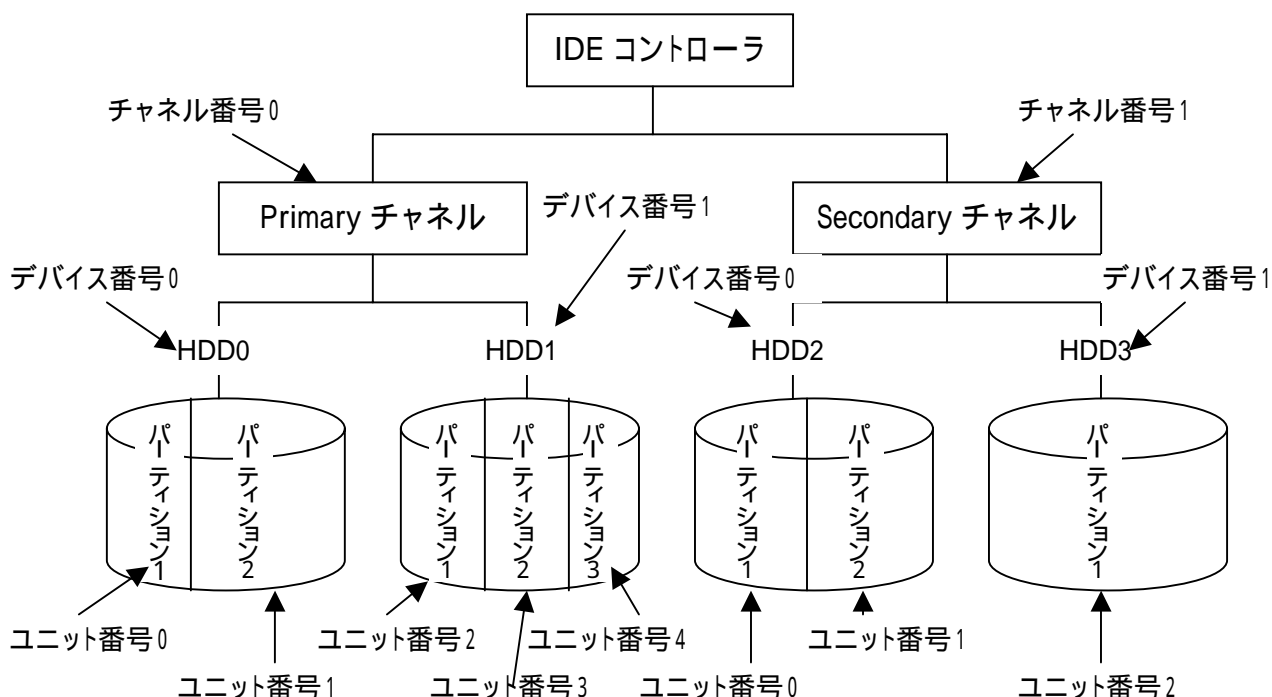
論理セクタ番号 0 は、FAT の Reserved Region の先頭セクタつまり BPB(BIOS Parameter Block)に該当します。

例えば、PCATA カードドライバの場合、与えられた論理セクタ番号を LBA(Logical Block Address)形式の、デバイス先頭セクタを 0 とする物理セクタ番号に変換して、ATA コマンドで使します。

この場合、物理セクタ番号 0 は、MBR(Master Boot Record)に該当します。

FAT ファイルシステムからは、上記定義の、チャンネル番号とデバイス番号、又はチャンネル番号とユニット番号を指定して、I/O 要求が I/O ドライバに対して行われます。

例) IDE コントローラの Primary、Secondary にそれぞれ 2 台ずつの HDD が接続されている場合の、チャンネル番号、デバイス番号、ユニット番号の関係



上記の様な構成の場合、

- ・チャンネル数 = 2、チャンネル番号 0 を Primary チャンネルとします。
チャンネル番号 1 を Secondary チャンネルとします。
- ・チャンネル番号 0 に接続されているデバイス数 = 2
デバイス番号 0 を HDD0、デバイス番号 1 を HDD1 とします。
- ・チャンネル番号 1 に接続されているデバイス数 = 2
デバイス番号 0 を HDD2、デバイス番号 1 を HDD3 とします。
- ・チャンネル番号 0 に接続されているユニット数 = 5
ユニット番号 0 を HDD0-パーティション 1、ユニット番号 1 を HDD0-パーティション 2、
ユニット番号 2 を HDD1-パーティション 1、ユニット番号 3 を HDD1-パーティション 2、
ユニット番号 4 を HDD1-パーティション 3 とします。
- ・チャンネル番号 1 に接続されているユニット数 = 3
ユニット番号 0 を HDD2-パーティション 1、ユニット番号 1 を HDD2-パーティション 2、
ユニット番号 2 を HDD3-パーティション 1 とします。

2-2. リエントラント構造

I/O ドライバは、チャンネル単位でリエントラントな構造とします。

(1)スタック領域

I/O ドライバで使用するスタックは、上位アプリケーション(タスク)のスタック領域を使用します。

タスクから FAT ファイルシステム経由で I/O ドライバを使用する場合、スタック領域は、タスク毎に割当てられているものとします。

複数タスクから I/O ドライバが呼び出されても、タスク固有のスタック領域を使用するのでリエントラントとなります。

(2)グローバル変数

ドライバ内のグローバルデータ領域に、ドライバデータ領域、チャンネルデータ領域、デバイスデータ領域、ユニットデータ領域の4種類のデータ領域を設け、必要なグローバル変数は全てこの4種類の領域のいずれかに定義します。

ドライバデータ領域:ドライバ全体に必要なデータを配置します。

但し、当領域のデータのアクセスには、リエントラント時の考慮が必要となります。

基本的には、当データ領域には定数値等の読み込み専用のデータを配置して下さい。

チャンネルデータ領域:接続されているチャンネル数分、領域を確保し、チャンネル単位に必要なデータを配置します。

チャンネル単位で、FAT ファイルシステム側で排他制御されるので、当データ領域に配置されたデータは、リエントラント時の考慮が不要となります。

殆どのデータが当データ領域に配置されものと考えられます。

FAT ファイルシステムから I/O ドライバへの I/O 要求には、全てチャンネル番号が与えられるのでこのチャンネル番号で、チャンネルデータ領域を決定します。

これにより、複数タスクから別々のチャンネル指定で I/O ドライバが呼び出されても、チャンネル固有のデータ領域を使用するのでリエントラントとなります。

デバイスデータ領域:接続されているデバイス数分、領域を確保し、デバイス単位に必要なデータを配置します。

チャンネル単位で、排他制御されるので、当データ領域に配置されたデータも、リエントラント時の考慮が不要となります。

デバイス単位で管理が必要となるデータを配置します。(デバイスサイズ等)

ユニットデータ領域: 接続されているユニット数分、領域を確保し、ユニット単位に必要なデータを配置します。

チャンネル単位で、排他制御されるので、当データ領域に配置されたデータも、リエントラント時の考慮が不要となります。

ユニット単位で管理が必要となるデータを配置します。(ユニットサイズ等)

2-3. ドライバ構成情報

FAT ファイルシステムでは、機器構成に従って、使用するドライバに関する情報を、ドライバ構成情報として定義します。

当情報は別インクルードファイルとして、FAT ファイルシステムに組み込まれます。

従って、I/O ドライバでは、下記ドライバ構成情報のうち、ドライバ ID、チャンネル最大数、メディアタイプ、ドライバエントリアドレスを明示しておく必要が有ります。

```
struct {
    unsigned char    ドライバ ID;
    unsigned char    チャンネル最大数;
    unsigned char    メディアタイプ;
    unsigned char    予備;
    (*起動番地)()    ドライバエントリアドレス;
    unsigned short   セマフォ ID[チャンネル最大数];
} ドライバ構成情報[ドライバ最大数]
```

項 目	内 容
ドライバ ID	ドライバの種別 0x01=PCATA カードドライバ、0x02=IDE ドライバ、 0x03=RAM ディスクドライバ、 以降ドライバ開発毎に追加。0x40～0x7F は、ユーザー開発ドライバ用に開放してあります。 Bit7 は予約領域の為、使用できません。
チャンネル最大数	ドライバで制御する最大チャンネル数
メディアタイプ	当該ドライバのメディアタイプ 0xF0=removable media、0xF8=fixed media
ドライバエントリアドレス	各ドライバ内で定義されるドライバエントリアドレス ファイルシステムからのドライバ呼出は全てこのエントリアドレス 及び I/O ファンクション番号で行われます。
セマフォ ID	FAT ファイルシステム側で、排他制御時に使用するセマフォの ID を定義します。

2-4. MBR 構造

ハードディスク等のメディアで、デバイスの物理セクタ番号 0 に配置されている MBR(Master Boot Record)の構造を下記に示します。

フィールド名	オフセット (Byte)	サイズ (Byte)	説 明
起動コード	0	446	ブートコード
パーティションテーブル	446	64	パーティションテーブルエントリ × 4 (4 パーティション分)
シグネチャ 1	510	1	0x55
シグネチャ 2	511	1	0xAA

パーティションテーブルエントリ

フィールド名	オフセット (Byte)	サイズ (Byte)	説 明
アクティブフラグ	0	1	0x80: 起動パーティション、 0x00: 起動パーティション以外
CHS 開始セクタ	1	3	CHS 形式によるパーティション開始セクタ <div><div>MSB<div>オフセット 1<div>H(8 ビット)</div></div><div>オフセット 2<div>C 上位 2 ビット</div><div>S(6 ビット)</div></div><div>オフセット 3<div>C 下位 8 ビット</div></div></div>LSB C:シリンダ番号、H:ヘッド番号、S:セクタ番号</div>
パーティションタイプ	4	1	ファイルシステム種類。0x00=未使用 FAT ファイルシステムがサポートする パーティションタイプは下記の通り 0x01: FAT12、0x04: FAT16(32MByte 未満)、 0x06: FAT16(32MByte 以上)、0x0B: FAT32(CHS) 0x0C: FAT32(LBA)、0x0E: FAT16(LBA)
CHS 終了セクタ	5	3	CHS 形式によるパーティション終了セクタ
LBA 開始セクタ	8	4	LBA 形式によるパーティション開始セクタ (リトルエンディアン)
パーティションサイズ	12	4	セクタ単位のサイズ(リトルエンディアン)

パーティションテーブルエントリオフセット = $0x01BE + ((N-1) \times 16)$ N: パーティション番号

3. API 関数

3-1. 呼出し形式

I/O 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されます。

I/O ファンクションが初期化及びドライバ情報取得の場合、ダミー値(任意の値)が指定されます。

I/O ファンクションがチャンネル情報取得の場合、ユニット番号にはダミー値(任意の値)が指定されます。

IN: unsigned char I/O ファンクション番号

I/O ドライバに要求する機能が指定されます。

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	0x0a	パーティション作成
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 パラメータが指定されます。

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(「4.エラーコード」参照願います。)

3-2. 初期化(I/O ファンクション番号 = 0x00)

機能

I/O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

I/O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

I/O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

指定されたユニットをマウント済みとし、以降、各種I/O要求を受付可能とし、デバイスに関する情報をマウントI/Oパラメータに返します。

その他

*1 T_MOUNT_PARAM

項 目	説 明
MBR モード	MBR(MasterBootRecord)の有無が指定されます。 00H=MBR 有り、 01H=MBR 無し
パーティションタイプ	デバイスのパーティションタイプを返します。 0x01: FAT12 0x04: FAT16(32MByte 未満) 0x06: FAT16(32MByte 以上) 0x0B: FAT32(CHS) 0x0C: FAT32(LBA) 0x0E: FAT16(LBA)
セクタサイズ	セクタサイズを 512 バイト単位で返します。
ユニット全体のセクタ数	ユニットの総セクタ数を返します。
ユニット開始セクタ番号	ユニットの開始セクタ番号を返します。

3-4. アンマウント(I/O ファンクション番号 = 0x02)

機能

I/O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

デバイスをアンマウント状態とします。以降、I/Oドライバでは、ユニットに対する、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. チャンネル情報取得 (I/O ファンクション番号 = 0x03)

機能

チャン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

チャン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	デバイスより取得し返します。
シリンダ数	デバイスより取得し返します。
ヘッド数	デバイスより取得し返します。
合計セクタ数	デバイスより取得し返します。。

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	デバイスのパーティションタイプを返します。 マウントの I/O パラメータを参照して下さい。
セクタサイズ	セクタサイズを 512 バイト単位で返します。
ユニット状態	ユニット状態を返します。 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	ユニットの総セクタ数を返します。
ユニット開始セクタ番号	ユニットの開始セクタ番号を返します。

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、ユニット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が指定されます。
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が指定されます。 正常終了時、実際に読み込んだセクタ数を返します。
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が指定されます。

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、ユニット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が指定されます。
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が指定されます。 正常終了時、実際に書込めたセクタ数を返します。
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が指定されます。

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、ユニット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が指定されます。
消去セクタ数	消去セクタ数を格納した領域のアドレスが指定されます。 正常終了時、実際に消去できたセクタ数を返します。
消去データ	消去データが指定されます。

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返します。 0x00=カードが挿入されている 0x01=別カードが挿入されている 0x02=カードが抜かれた

3-12. パーティション作成(I/O ファンクション番号 = 0x0a)

機能

メディア上にパーティションを作成します。

I/Oパラメータ

```
struct {  
    unsigned char    動作モード  
    unsigned char    パーティションタイプ  
    unsigned char    パーティション番号  
    unsigned char    予備[1]  
    unsigned short   予備[2]  
    unsigned long    パーティションサイズ  
    unsigned long    予備  
} T_PART_CREATE_PARAM;
```

解説

パーティション作成I/Oパラメータに従って、基本領域又は拡張領域にパーティションを作成又は削除します。

その他

*1 T_PART_CREATE_PARAM

項 目	説 明
動作モード	パーティション作成時の動作が指定されます。 0x00=基本領域にパーティション作成 0x01=拡張領域にパーティション作成 0x02=拡張領域上のパーティション削除
パーティションタイプ	動作モードが「0x00=基本領域にパーティション作成」、 「0x01=拡張領域にパーティション作成」の場合、 作成するパーティションタイプが指定されます。
パーティション番号	動作モードが「0x02=拡張領域上のパーティション削除」 の場合、削除対象パーティションのパーティション番号が 指定されます。 パーティション番号は基本領域上のパーティションを0と し、以降パーティションテーブル作成順に1ずつインクリメ ントされた値とします。
パーティションサイズ	動作モードが「0x00=基本領域にパーティション作成」、 「0x01=拡張領域にパーティション作成」の場合、 作成するパーティションのサイズがセクタ単位で指定され ます。

4. エラーコード

ラベル名	エラー内容
DFMR_SUC	正常終了
DFMR_DRVERR_PARAM	パラメータエラー
DFMR_DRVERR_NOTINIT	未初期化エラー
DFMR_DRVERR_NOCARD	カード抜きエラー
DFMR_DRVERR_ILGALVS	VS1,2 状態エラー
DFMR_DRVERR_MRSHPC	MR-SHPC-01 異常
DFMR_DRVERR_PCCIS	PC カード CIS 異常
DFMR_DRVERR_PCREADYTMO	PC カード Ready TimeOut
DFMR_DRVERR_PCCARD	PC カード異常
DFMR_DRVERR_TSLP	PCATA_osSleep 異常終了
DFMR_DRVERR_UNITOVER	該当ユニット無し
DFMR_DRVERR_NOTMOUNT	未マウントエラー
DFMR_DRVERR_DEFCARD	別カード挿入
DFMR_DRVERR_ILGCARD	カード挿抜発生
DFMR_DRVERR_MOUNT	マウント済みエラー
DFMR_DRVERR_NOPRIMARY	基本領域無し
DFMR_DRVERR_NOEXTEND	拡張領域無し