

GR-USB/HOST#
マスストレージクラスドライバ
API 仕様書

第 1.01 版

2008 年 10 月

株式会社グレースシステム

[注意事項]

- ・すべての著作権は、株式会社グレープシステムにあります。
- ・本ドキュメントの内容の一部または全部を無断で転載、複写、複製する事を禁じます。
- ・本製品の仕様は予告なく変更される事があります。
- ・本ドキュメントに記載されている会社名、製品名は各社の商標または登録商標です。

Copyright (C) 2007-2008 Grape Systems, Inc. All Rights Reserved

はじめに

本書は、GR-USB/HOST#におけるマストレージクラス（以下、MSC）の API 仕様を記述したものです。

改訂履歴

Rev.	日付	改訂内容
1.00	2008 年 7 月	初版リリース
1.01	2008 年 10 月	全般 ・以下のコールバックに関するメンバ名変更による修正 grp_msc_reg 構造体 pfnMscEvCallback pfnCallback

目次

1 MSC ドライバの概要.....	1
1.1 MSC ドライバとは	1
2 インターフェース	3
2.1 コーディング規約	3
2.1.1 型名	3
2.1.2 変数名.....	3
2.2 エラーコード	5
2.2.1 API 関数エラーコード	5
2.2.2 通信完了エラーコード.....	5
2.3 定義値	6
2.3.1 デバイス状態コード	6
2.3.2 サブクラスコード.....	6
2.3.3 プロトコルコード.....	6
2.3.4 リセットモード	6
2.4 構造体	7
2.4.1 grp_msc_reg 構造体.....	7
2.4.2 grp_msc_cmd 構造体	7
2.5 API 関数一覧	8
2.6 API 関数詳細	8
grp_msc_Init.....	9
grp_msc_Register	10
grp_msc_Open	12
grp_msc_Close	13
grp_msc_ReqCmd.....	14
grp_msc_Cancel.....	16
grp_msc_Abort.....	17
grp_msc_Reset.....	18
grp_msc_GetMaxLun	19
grp_msc_GetMaxLunCancel.....	21
grp_msc_GetDeviceId.....	22
grp_msc_GetSubClass.....	23
grp_msc_ReadSector	24
grp_msc_WriteSector	26
grp_msc_Inquiry.....	28
grp_msc_ReadFormatCapacity.....	30
grp_msc_ReadCapacity	32
grp_msc_ModeSense	34
grp_msc_TestUnitReady	36

grp_msc_RequestSense 38



API 関数索引

(アルファベット順)

grp_msc_Abort	17
grp_msc_Cancel	16
grp_msc_Close.....	13
grp_msc_GetDeviceId	22
grp_msc_GetMaxLun.....	19
grp_msc_GetMaxLunCancel	21
grp_msc_GetSubClass	23
grp_msc_Init.....	9
grp_msc_Inquiry	28
grp_msc_ModeSense	34
grp_msc_Open.....	12
grp_msc_ReadCapacity.....	32
grp_msc_ReadFormatCapacity	30
grp_msc_ReadSector.....	24
grp_msc_Register.....	10
grp_msc_ReqCmd.....	14
grp_msc_RequestSense.....	38
grp_msc_Reset	18
grp_msc_TestUnitReady.....	36
grp_msc_WriteSector	26

1 MSC ドライバの概要

MSC (マストレージクラス) ドライバの概要について説明します。

1.1 MSC ドライバとは

MSC ドライバとは、USB のクラス仕様で定義されたマストレージクラスに準拠したドライバで、主に USB メモリや USB ハードディスクなどのストレージデバイスにアクセスするためのものです。本ドライバでは、接続された機器のプロトコル/サブクラスを意識することなくアクセスすることが可能です。

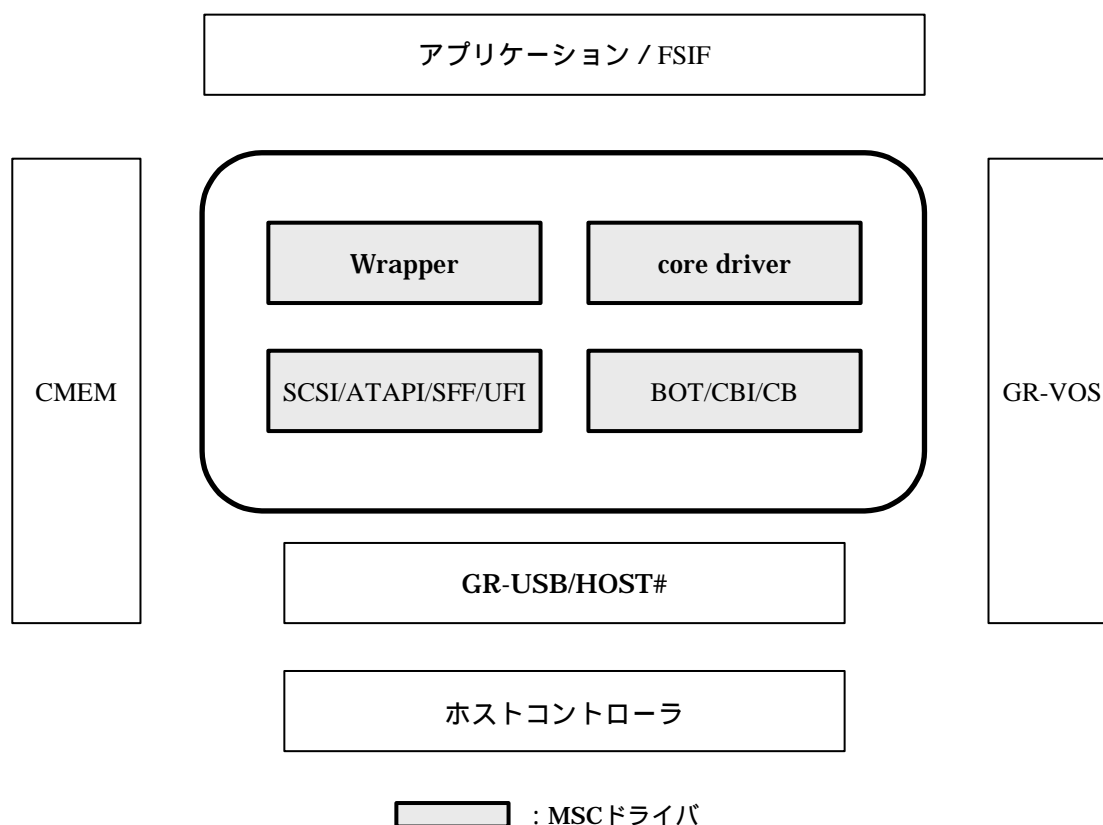


図 1-1 MSC ドライバの概要

(1) BOT/CBI/CB

本ドライバでサポートしている通信プロトコルです。

(2) SCSI/ATAPI/SFF/UFI

本ドライバでサポートしているサブクラスです。

(3) core driver

MSC ドライバのコアドライバです。

初期化処理や登録処理など上位アプリケーションへの API を提供します。

(4) Wrapper

上位アプリケーションからサブクラスを意識することなくアクセスするためのラッパー処理です。以下に示すコマンドに対応しています。

- ・ READ
- ・ WRITE
- ・ INQUIRY
- ・ READ CAPACITY
- ・ MODE SENSE
- ・ TEST UNIT READY
- ・ REQUEST SENSE

2 インターフェース

MSC (マストレージクラス) ドライバが提供するインターフェースについて説明します。

2.1 コーディング規約

Grapeware 製品では、以下の規約に準じてコーディングを行っております。

2.1.1 型名

環境に応じて変更できるよう、Grapeware 製品では一般的に使用する型を `typedef` 宣言を用いて別名に置き換えて使用しています。(`grp_std_types.h`)

Grapeware 製品で使用する型名は以下のとおりです。

表 2-1 Grapeware 製品で使用する型名

#	型名	内容
1	<code>grp_ui</code>	<code>unsigned int</code>
2	<code>grp_si</code>	<code>signed int</code>
3	<code>grp_u32</code>	<code>unsigned long</code>
4	<code>grp_s32</code>	<code>signed long</code>
5	<code>grp_u16</code>	<code>unsigned short</code>
6	<code>grp_s16</code>	<code>signed short</code>
7	<code>grp_u8</code>	<code>unsigned char</code>
8	<code>grp_s8</code>	<code>signed char</code>

2.1.2 変数名

変数や構造体のメンバなどは、変数の名称からその型がわかるようにするため、変数の先頭に次の識別用の文字を付けています。

表 2-2 変数名の先頭に付ける識別用文字

#	識別用文字	内容
1	<code>g_</code>	グローバル変数
2	<code>l_</code>	ローカル (Static) 変数
3	<code>p</code>	ポインタ
4	<code>a</code>	配列
5	<code>v</code>	<code>void</code>
6	<code>i</code>	<code>grp_si</code>
7	<code>ui</code>	<code>grp_ui</code>
8	<code>c</code>	<code>grp_s8</code>
9	<code>uc</code>	<code>grp_u8</code>
10	<code>s</code>	<code>grp_s16</code>
11	<code>us</code>	<code>grp_u16</code>
12	<code>l</code>	<code>grp_s32</code>
13	<code>ul</code>	<code>grp_u32</code>
14	<code>f</code>	<code>float</code>
15	<code>d</code>	<code>double</code>

16	t	typedef (構造体など)
17	fn	関数
18	h	ハンドル

2.2 エラーコード

MSC ドライバで使用するエラーコードについて説明します。

2.2.1 API 関数エラーコード

API 関数の返却値として返るエラーコードは以下のとおりです。

表 2-3 API 関数エラーコード

#	エラーコード	値	内容
1	GRP_MSC_OK	0	正常終了
2	GRP_MSC_ERROR	0x8500FFFF	異常終了
3	GRP_MSC_NOSUPPORT	0x8500FFFE	未サポート
4	GRP_MSC_BUSY	0x8500FFFD	ビジー
5	GRP_MSC_UNINIT_ERROR	0x8500FFFC	未初期化エラー
6	GRP_MSC_REQUEST_ERROR	0x8500FFFB	リクエストエラー
7	GRP_MSC_RESOURCE_ERROR	0x8500FFFA	リソースエラー
8	GRP_MSC_VOS_ERROR	0x8500FFF9	VOS リソース生成エラー
9	GRP_MSC_CMEM_ERROR	0x8500FFF8	CMEM モジュールエラー

2.2.2 通信完了エラーコード

通信完了時に呼び出されるコールバック関数の引数 `grp_msc_cmd` 構造体のメンバ `lStatus` に格納されるエラーコードは以下のとおりです。

表 2-4 通信完了エラーコード

#	エラーコード	値	内容
1	GRP_MSC_NOT_PROCESS	0	通信要求が未処理状態
2	GRP_MSC_NO_FAIL	1	正常終了
3	GRP_MSC_CANCEL	2	キャンセル終了
4	GRP_MSC_ILLEGAL_FAIL	0x8502FFFF	異常終了
5	GRP_MSC_TIMEOUT	0x8502FFFE	タイムアウト
6	GRP_MSC_CHECK_CONDITION	0x8502FFFD	Request Sense にて確認が必要なエラー

2.3 定義値

MSC ドライバで使用する定義値は以下のとおりです。

2.3.1 デバイス状態コード

表 2-5 デバイス状態コード

#	定義値	値	内容
1	GRP_MSC_ATTACHED	GRP_CNFSFT_DEVICE_ATTACHED	接続状態
2	GRP_MSC_DETACHED	GRP_CNFSFT_DEVICE_DETACHED	切断状態

2.3.2 サブクラスコード

表 2-6 サブクラスコード

#	定義値	値	内容
1	GRP_MSC_ATAPI_CODE	0x02	ATAPI
2	GRP_MSC_SFF8070I_CODE	0x05	SFF8070i
3	GRP_MSC_SCSI_CODE	0x06	SCSI
4	GRP_MSC_UFI_CODE	0x04	UFI

2.3.3 プロトコルコード

表 2-7 プロトコルコード

#	定義値	値	内容
1	GRP_MSC_BOT_CODE	0x50	BOT
2	GRP_MSC_CBI_CODE	0x00	CBI
3	GRP_MSC_CB_CODE	0x01	CB

2.3.4 リセットモード

表 2-8 リセットモード

#	定義値	値	内容
1	GRP_MSC_RESET_MASS	1	BOT の場合：マスタストレージリセット CBI の場合：コマンドブロックリセット
2	GRP_MSC_RESET_ENUM	2	USB バスリセットおよびエニュメレーション

2.4 構造体

MSC ドライバで使用する構造体は以下のとおりです。

2.4.1 grp_msc_reg 構造体

表 2-9 grp_msc_reg 構造体

#	メンバ名	型	内容
1	ucSubClass	grp_u8	サブクラス
2	ucProtocol	grp_u8	プロトコル
3	ucInputIfNum	grp_u8	インターフェース番号
4	pfnMscEvCallback	grp_s32 (*)(grp_msc_notify *)	コールバック関数
5	pvUserRef	void*	ユーザポインタ
6	usVendorId	grp_u16	ベンダ ID
7	usProductId	grp_u16	プロダクト ID
8	ulMode	grp_u32	登録モード
9	ulStatus	grp_u32	ステータス

2.4.2 grp_msc_cmd 構造体

表 2-10 grp_msc_cmd 構造体

#	メンバ名	型	内容
1	hMscHdr	grp_msc_hdr	マストレージクラスハンドル
2	ucLun	grp_u8	論理ユニット番号あるいは数
3	ucDir	grp_u8	データ転送方向、またはデータなし
4	lStatus	grp_s32	ステータス
5	pucReqBuffer	grp_u8*	バッファポインタ
6	ulReqLength	grp_u32	要求転送長
7	ulActualLength	grp_u32	実際の転送長
8	pfnCallback	grp_s32 (*)(grp_msc_cmd *)	コールバック関数
9	pvUserRef	void*	ユーザポインタ
10	ulCmdLength	grp_u32	コマンド長
11	aucCmdContent[]	grp_u8	コマンド
12	tStaff	grp_msc_staff	ドライバ使用領域

2.5 API 関数一覧

MSC ドライバは、以下の API 関数を用意しています。

初期化・登録関数

<code>grp_msc_Init</code>	MSC ドライバの初期化
<code>grp_msc_Register</code>	デバイスの登録

基本関数

<code>grp_msc_Open</code>	デバイスのオープン
<code>grp_msc_Close</code>	デバイスのクローズ
<code>grp_msc_ReqCmd</code>	コマンドの要求
<code>grp_msc_Cancel</code>	コマンドのキャンセル
<code>grp_msc_Abort</code>	コマンドの中断
<code>grp_msc_Reset</code>	デバイスのリセット
<code>grp_msc_GetMaxLun</code>	Get Max LUN の要求
<code>grp_msc_GetMaxLunCancel</code>	Get Max LUN のキャンセル
<code>grp_msc_GetDeviceId</code>	デバイス ID の取得
<code>grp_msc_GetSubClass</code>	サブクラスの取得

コマンド要求関数

<code>grp_msc_ReadSector</code>	READ SECTOR コマンド要求
<code>grp_msc_WriteSector</code>	WRITE SECTOR コマンド要求
<code>grp_msc_Inquiry</code>	INQUIRY コマンド要求
<code>grp_msc_ReadFormatCapacity</code>	READ FORMAT CAPACITIES コマンド要求
<code>grp_msc_ReadCapacity</code>	READ CAPACITY コマンド要求
<code>grp_msc_ModeSense</code>	MODE SENSE コマンド要求
<code>grp_msc_TestUnitReady</code>	TEST UNIT READY コマンド要求
<code>grp_msc_RequestSense</code>	REQUEST SENSE コマンド要求

2.6 API 関数詳細

本項では API 関数の詳細なフォーマットについて記載しています。

grp_msc_Init

MSC ドライバの初期化

【構文規則】

grp_s32 grp_msc_Init(void *pvRsv)

【入力パラメータ】

pvRsv 予約引数
GRP_USB_NULL を設定してください

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
GRP_MSC_VOS_ERROR	VOS リソース生成エラー
GRP_MSC_CMEM_ERROR	CMEM モジュールエラー

【機能】

本関数では MSC ドライバの初期化を行います。

本関数を呼び出す前に、GR-VOS の初期化 (grp_vos_Init) と GR-USB/HOST# の初期化 (grp_usbc_HostInit) を実行する必要があります。

また、GR-USB/HOST# の起動処理 (grp_usbc_Enable) は、本関数実行後に呼び出してください。

【備考】

本関数がエラーとなった場合、本ドライバや下位ドライバおよび GR-VOS モジュールのポーティングに問題がある可能性があります。各ドキュメントを参照し、設定値をご確認ください。

grp_msc_Register

デバイスの登録

【構文規則】

```
grp_s32    grp_msc_Register( grp_msc_reg *ptUsrReg, grp_u32 ulParam )
```

【入力パラメータ】

ptUsrReg	デバイス登録情報
- ucSubClass	サブクラスコード
- ucProtocol	プロトコルコード
- pfnMscEvCallback	デバイス接続・切断時に呼び出されたいコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ
- ulMode	予約 GRP_MSC_REG_PROTOCOL を設定してください

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数では、このクラスで取り扱うデバイスのサブクラスとプロトコル組み合わせの登録を行います。

接続したいサブクラスとプロトコルの組み合わせが複数ある場合は、繰り返し登録してください。登録したサブクラスとプロトコルの組み合わせを持つデバイスが接続（または切断）すると、設

定したコールバック関数が呼ばれます。この時に渡されるマスメージハンドルの、以降の転送で使用されますので、保存しておく必要があります。

なお、複数のインターフェースがあり、複数のマスメージクラスがあるデバイスに関しては、その数分のコールバック関数が呼ばれます。

【コールバック関数】

デバイス接続・切断時に呼び出されるコールバック関数の形式は以下の通りです。

`grp_s32 (*pfnMscEvCallback)(grp_msc_notify *ptUsrNotify)`

< 出力パラメータ >

<code>ptUsrNotify</code>	デバイス登録情報
- <code>iEvent</code>	デバイス状態コード
- <code>ucSubClass</code>	デバイスのサブクラスコード
- <code>ucProtocol</code>	デバイスのプロトコルコード
- <code>pvUserRef</code>	登録時に設定したポインタ
- <code>hMscHdr</code>	マスメージハンドル（以後の転送で使用）

< 返却値 >

GRP_MSC_OK を返してください。

grp_msc_Open

デバイスのオープン

【構文規則】

```
grp_s32    grp_msc_Open( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数では接続されたデバイスのオープンを行います。
デバイスの使用前には、必ず本関数を実行する必要があります。
なお、多重オープン処理やデバイス切断後のオープン処理ではエラーとなります。

grp_msc_Close

デバイスのクローズ

【構文規則】

```
grp_s32    grp_msc_Close( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数では切断されたデバイスのクローズを行います。

デバイスの使用完了後には、クローズを行う必要があります。しかし、必ずしも切断後に実行する必要はありません。接続中に使用を完了した場合は、そのタイミングでクローズすることもできます。その場合、切断前であれば、再オープンが可能です。

なお、オープン処理を行っていない状態でクローズ処理を行った場合、エラーとなります。

grp_msc_ReqCmd

コマンドの要求

【構文規則】

```
grp_s32    grp_msc_ReqCmd( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- ucDir	データ転送方向 GRP_USBD_TX_OUT OUT GRP_USBD_TX_IN IN 0 データなし
- pucReqBuffer	転送用バッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ
- ulCmdLength	コマンド長 (単位 : バイト)
- ucCmdContent[]	コマンドブロック

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK 正常終了

GRP_MSC_ERROR

異常終了

othres

下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、コマンドの要求を行います。

基本的なコマンドは、コマンド要求関数として用意していますので、そちらをご利用ください。

本関数は、コマンドブロック用に用意した領域に、サブクラスにあわせたコマンドブロックを作成し、そのコマンドに合わせた情報をパラメータとして設定していただくことで使用することができます（プロトコルについては意識する必要はありません）。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ IStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

```
grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)
```

< 出力パラメータ >

ptUsrCmd	コマンド情報
- IStatus	通信完了ステータス
- pucReqBuffer	転送データ
- ulActualLength	実際の転送データ長（単位：バイト）
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本関数をご利用の際には、コマンド要求関数を参考にしてください。

grp_msc_Cancel

コマンドのキャンセル

【構文規則】

```
grp_s32    grp_msc_Cancel( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	キャンセルを行うコマンド情報 通信要求時にパラメータで渡した通信要求を そのまま渡してください。
-------	--

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数では要求中のコマンドのキャンセルを行います。

grp_msc_Abort

デバイスに対するコマンド処理の中断

【構文規則】

```
grp_s32    grp_msc_Abort(grp_msc_hdr * hMscHdr)
```

【入力パラメータ】

hMscHdr	マストレージハンドル
---------	------------

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数では要求中のコマンドの中断を行います。
マストレージハンドルで示されるデバイスに対する要求すべてを中断します。

【備考】

Get Max LUN コマンド (grp_msc_GetMaxLun) については、別途キャンセル処理が必要です。

grp_msc_Reset

デバイスのリセット

【構文規則】

```
grp_s32    grp_msc_Reset( grp_msc_cmd *ptCmd, grp_u32 ulMode )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マスタストレージハンドル grp_msc_hdr 構造体
- ulMode	リセットモード GRP_MSC_RESET_ENUM USB バスリセットとエニュメレーション GRP_MSC_RESET_MASS BOT の場合：マスタストレージリセット CBI の場合：コマンドブロックリセット

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対してリセットを行います。

リセットモードによっては通信が発生する場合がありますが、全て同期要求として処理されます。

(完了するまで復帰しません。)

grp_msc_GetMaxLun

Get Max LUN の要求

【構文規則】

```
grp_s32    grp_msc_GetMaxLun( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- pucReqBuffer	取得した LUN(論理ユニット数) を格納するバッファの 先頭アドレス (共通メモリ管理モジュール(CMEM) から確保した領 域、または非キャッシュ領域をご利用ください) (1 バイト以上のバッファを用意する必要があります)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
GRP_MSC_NOSUPPORT	未サポート
GRP_MSC_BUSY	ビジー
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、Get Max LUN 要求を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ `lStatus` に格納されます。

本関数では、BOT プロトコルのデバイスに対してのみ `GetMaxLUN` 要求を行います。

その他のプロトコル (CBI など) のデバイスに対して本関数を実行した場合、デバイスに対しては何も処理されず、本関数からは `GRP_MSC_NOSUPPORT` が返ります。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

```
grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)
```

< 出力パラメータ >

<code>ptUsrCmd</code>	コマンド情報
- <code>lStatus</code>	通信完了ステータス
- <code>pucReqBuffer</code>	LUN (論理ユニット数)
- <code>ulActualLength</code>	実際の転送データ長 (単位: バイト)
- <code>pvUserRef</code>	要求時に設定したポインタ

< 返却値 >

`GRP_MSC_OK` を返してください。

【備考】

論理ユニット数は、「出力パラメータで得られる値 + 1」として求められます。

また本リクエストに対して `STALL` を返すデバイスがあった場合、本関数内では正常終了として扱い、論理ユニット数には 0 (1 ユニット) を設定してコールバック関数を実行します。

grp_msc_GetMaxLunCancel

Get Max LUN のキャンセル

【構文規則】

```
grp_s32    grp_msc_GetMaxLunCancel( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	キャンセルを行うコマンド情報 通信要求時にパラメータで渡した通信要求情報を、そのまま渡してください
-------	--

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
GRP_MSC_NOSUPPORT	未サポート
othres	下位モジュールでのエラー

【機能】

本関数では、Get Max LUN 要求のキャンセルを行います。

通信要求をキャンセルした場合、通信要求時に指定したコールバック関数が呼び出され、その通信完了情報の lStatus には GRP_USBD_TR_CANCEL がセットされます。

grp_msc_GetDeviceId

デバイス ID の取得

【構文規則】

grp_s32 grp_msc_GetDeviceId (grp_msc_hdr hMscHdr, grp_u8 * pusDevId)

【入力パラメータ】

hMscHdr	マストレージハンドル
pusDevId	デバイス ID を格納する領域のポインタ

【出力パラメータ】

pusDevId	デバイス ID
----------	---------

【返却値】

GRP_MSC_OK	正常終了
------------	------

【機能】

本関数では、マストレージハンドルで表されるデバイスのデバイス ID を取得します。

grp_msc_GetSubClass

サブクラスの取得

【構文規則】

```
grp_s32    grp_msc_GetSubClass( grp_msc_hdr hMscHdr, grp_u8 *pucSubClass )
```

【入力パラメータ】

hMscHdr	マストレージハンドル
pucSubClass	サブクラスコードを格納する領域のポインタ

【出力パラメータ】

pucSubClass	サブクラスコード 表 2-6 サブクラスコード を参照ください。
-------------	---

【返却値】

GRP_MSC_OK	正常終了
------------	------

【機能】

本関数では、マストレージハンドルで表されるデバイスのサブクラスコードの取得を行います。

grp_msc_ReadSector

READ SECTOR コマンド要求

【構文規則】

```
grp_s32    grp_msc_ReadSector( grp_msc_cmd *ptCmd, grp_u32 ulStartLba,  
                                grp_u32 ulSectorSize )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	読み込んだデータを格納するバッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト) セクタ長の倍数を設定ください。
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ
ulStartLba	読み込み開始論理セクタ番号
ulSectorSize	読み込みセクタ数

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、READ SECTOR コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	読み込んだデータ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

grp_msc_WriteSector

WRITE SECTOR コマンド要求

【構文規則】

```
grp_s32      grp_msc_WriteSector( grp_msc_cmd *ptCmd, grp_u32 ulStartLba,  
                                   grp_u32 ulSectorSize )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	書き込みデータを格納しているバッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト) セクタ長の倍数を設定ください。
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ
ulStartLba	書き込み開始論理セクタ番号
ulSectorSize	書き込みセクタ数

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、WRITE SECTOR コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfncallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd

コマンド情報

- lStatus

通信完了ステータス

- ulActualLength

実際の転送データ長 (単位 : バイト)

- pvUserRef

要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

grp_msc_Inquiry

INQUIRY コマンド要求

【構文規則】

```
grp_s32    grp_msc_Inquiry( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	取得した INQUIRY データを格納するバッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、INQUIRY コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

```
grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)
```

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	INQUIRY データ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本コマンドでは、36 バイト以上のバッファを用意してください。

grp_msc_ReadFormatCapacity

READ FORMAT CAPACITIES コマンド要求

【構文規則】

```
grp_s32    grp_msc_ReadFormatCapacity( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル
- ucLun	論理ユニット番号 grp_msc_hdr 構造体
- pucReqBuffer	取得した READ FORMAT CAPACITIES データを格納するバッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、READ FORMAT CAPACITIES コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfncallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	READ FORMAT CAPACITIES データ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本コマンドは、データ長が可変となります。最初の 4 バイトのデータを取得し、データ長を確認してください。

grp_msc_ReadCapacity

READ CAPACITY コマンド要求

【構文規則】

```
grp_s32    grp_msc_ReadCapacity( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	取得した READ CAPACITY データを格納するバッファ の先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、READ CAPACITY コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfncallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	READ CAPACITY データ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本コマンドでは、4 バイト以上のバッファを用意してください。

grp_msc_ModeSense

MODE SENSE コマンド要求

【構文規則】

grp_s32 grp_msc_ModeSense(grp_msc_cmd *ptCmd, grp_u8 ucPage)

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	取得した MODE SENSE データを格納するバッファの 先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領 域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ
ucPage	Page Control & Page Code Bit7-6 : Page Control Bit5-0 : Page Code

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、MODE SENSE コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	MODE SENSE データ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本コマンドは、データ長が可変となります。最初の 4 バイトのデータを取得し、データ長を確認してください。

本コマンドに対して、STALL で応答するデバイスが存在します。このようなデバイスは、その後のコマンドに対して正常に応答する場合がありますので、出力パラメータ lStatus (通信完了ステータス) が GRP_MSC_STALL だった場合には、そのまま処理を続けてください。

grp_msc_TestUnitReady

TEST UNIT READY コマンド要求

【構文規則】

```
grp_s32    grp_msc_TestUnitReady( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、TEST UNIT READY コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信

のステータスはコマンド情報のメンバ `lStatus` に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

`grp_s32 (*pfnCallback)(grp_msc_cmd* ptUsrCmd)`

< 出力パラメータ >

<code>ptUsrCmd</code>	コマンド情報
- <code>lStatus</code>	通信完了ステータス
- <code>pvUserRef</code>	要求時に設定したポインタ

< 返却値 >

`GRP_MSC_OK` を返してください。

grp_msc_RequestSense

REQUEST SENSE コマンド要求

【構文規則】

```
grp_s32    grp_msc_RequestSense( grp_msc_cmd *ptCmd )
```

【入力パラメータ】

ptCmd	コマンド情報
- hMscHdr	マストレージハンドル grp_msc_hdr 構造体
- ucLun	論理ユニット番号
- pucReqBuffer	取得した SENSE データを格納するバッファの先頭アドレス (共通メモリ管理モジュール (CMEM) から確保した領域をご利用ください)
- ulReqLength	転送データ長 (単位 : バイト)
- pfnCallback	通信完了時に呼び出されるコールバック関数 後述のコールバック関数参照
- pvUserRef	自由に使用可能なポインタ

【出力パラメータ】

なし

【返却値】

GRP_MSC_OK	正常終了
GRP_MSC_ERROR	異常終了
othres	下位モジュールでのエラー

【機能】

本関数ではデバイスに対して、REQUEST SENSE コマンド要求を行います。

本関数では、パラメータの情報を基にデバイスのサブクラスに準じたコマンドブロックを生成し、コマンド要求 (grp_msc_ReqCmd) を行います。

本関数は通信の結果如何にかかわらず即座に復帰します。

通信完了後、コマンド情報で指定したコールバック関数が呼び出されます。なお、その際の通信のステータスはコマンド情報のメンバ lStatus に格納されます。

【コールバック関数】

通信完了時に呼び出されるコールバック関数の形式は以下の通りです。

grp_s32 (*pfncallback)(grp_msc_cmd* ptUsrCmd)

< 出力パラメータ >

ptUsrCmd	コマンド情報
- lStatus	通信完了ステータス
- pucReqBuffer	SENSE データ
- ulActualLength	実際の転送データ長 (単位: バイト)
- pvUserRef	要求時に設定したポインタ

< 返却値 >

GRP_MSC_OK を返してください。

【備考】

本コマンドでは、18 バイト以上のバッファを用意してください。

**GR-USB/HOST#
マスタストレージクラスドライバ
API 仕様書**

発行年月：2008 年 10 月 第 1.01 版

発行：株式会社グレースシステム

E-Mail : gr@support.grape.co.jp

URL : <http://www.grape.co.jp>

Copyright (C) 2007-2008 Grape Systems, Inc.

All rights reserved.