

# GR-USB/HOST# FSIF モジュール ユーザーズガイド

第 1.02 版

2015 年 12 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) 2007-2015 Grape Systems, Inc. All Rights Reserved.

## はじめに

本書は、GR-USB/HOST#におけるファイルシステムインタフェース（以下、FSIF）モジュールの概要や導入手順、および使用方法について記述したものです。

## 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_fsif_init_prm 構造体 pfnFsifNotification ← pfnEventNotification
1.02	2015 年 12 月	定義値の初期値を変更 OS 資源の修正 誤字修正

## 用語

用語	説明
デバイス ID	USB デバイスアドレスとユニークな数値を組み合わせた弊社ドライバ独自のデバイス識別子
ATAPI	AT Attachment Packet Interface サブクラスの一つ
BOT	Bulk Only Transport プロトコルの一つ
CB	CBI (with no command completion interrupt) プロトコルの一つ
CBI	Control/Bulk/Interrupt Transport プロトコルの一つ
CMEM	共通メモリ管理モジュールの略 弊社ドライバ独自の処理でキャッシュやバウンダリ問題などの対応に利用される
CRC	Cyclic Redundancy Check の略
EOP	End-of-Packet の略
FS	Full Speed (max 12Mbps) の略
FSIF	File System InterFace の略 弊社 MSC ドライバ上に存在するモジュール
HC	Host Controller の略
HCD	Host Controller Driver の略
HCM	Host Controller Driver Modules の略
HS	High Speed (max 480Mbps) の略
LS	Low Speed (max 1.5Mbps) の略
MSC	マストレージクラスの略
OVC	Over Current の略 過電流のこと
PID	Packet ID の略
RHUB	Root HUB の略 ホストコントローラのポートを監視/制御する
SCSI	Small Computer System Interface の略 サブクラスの一つ
SFF-8070i	ATAPI for Floppies サブクラスの一つ
UFI	USB Floppy Interface の略 サブクラスの一つ
USB	Universal Serial Bus の略
USB マストレージ統合キット	弊社が提供するファイルシステムから USB ホストコントローラドライバまでのパッケージ製品の名称

## 参照ドキュメント

以下のドキュメントも併せて参照ください。

### FSIF

『GR-USB/HOST# FSIF モジュール API 仕様書』

『File System Device I/O モジュール リファレンスガイド』

### ファイルシステム

『組込み向けファイルシステム GR-FILE』

### MSCドライバ

『GR-USB HOST# MSC ユーザーズガイド』

GR-USB/HOST#

『GR-USB HOST# ユーザーズガイド』

GR-VOS

『GR-VOS ポーテイングガイド』

仕様書

『Universal Serial Bus Specification Revision 2.0』

## 目次

1 FSIFの概要 .....	1
1.1 FSIFとは .....	1
1.2 他モジュールとの関連 .....	1
1.3 機能 .....	2
2 開発手順 .....	3
3 インストール手順 .....	4
4 フォルダ構成 .....	5
5 準備 .....	6
5.1 動作環境の準備 .....	6
5.2 ポーティング手順 .....	8
5.2.1 ポーティング対象ファイル .....	8
5.2.2 定義値の設定 .....	8
5.3 ライブラリファイル作成手順 .....	11
6 使用方法 .....	13
6.1 API仕様 .....	13
6.2 アプリケーション作成方法 .....	14
6.2.1 必要なヘッダファイル .....	14
6.2.2 API関数の使い方 .....	14
6.3 実行ファイル作成手順 .....	18
6.4 サンプルアプリケーション .....	20
6.4.1 ファイル構成 .....	20
6.4.2 使用方法 .....	20
6.4.3 動作内容 .....	21
7 動作確認方法 .....	22
7.1 初期化処理の確認 .....	22
7.2 最初のデバイス接続処理の確認 .....	22
Appendix A 内部構造 .....	23
Appendix B カスタマイズ .....	27
Appendix C 注意事項 .....	29
Appendix D トレースログ機能 .....	30
Appendix E Grapeware技術サポート .....	31

## 1 FSIFの概要

FSIF の概要について説明します。

### 1.1 FSIFとは

FSIF（File System InterFace）とは、上位ファイルシステムに通常のデバイスドライバとしての I/F を提供するモジュールです。

マルチユニットデバイスにも対応し、各メディア毎に情報を管理します。

### 1.2 他モジュールとの関連

以下に、FSIF と他モジュールとの関連および USB マスストレージ統合キット内の位置付けについて図示します。

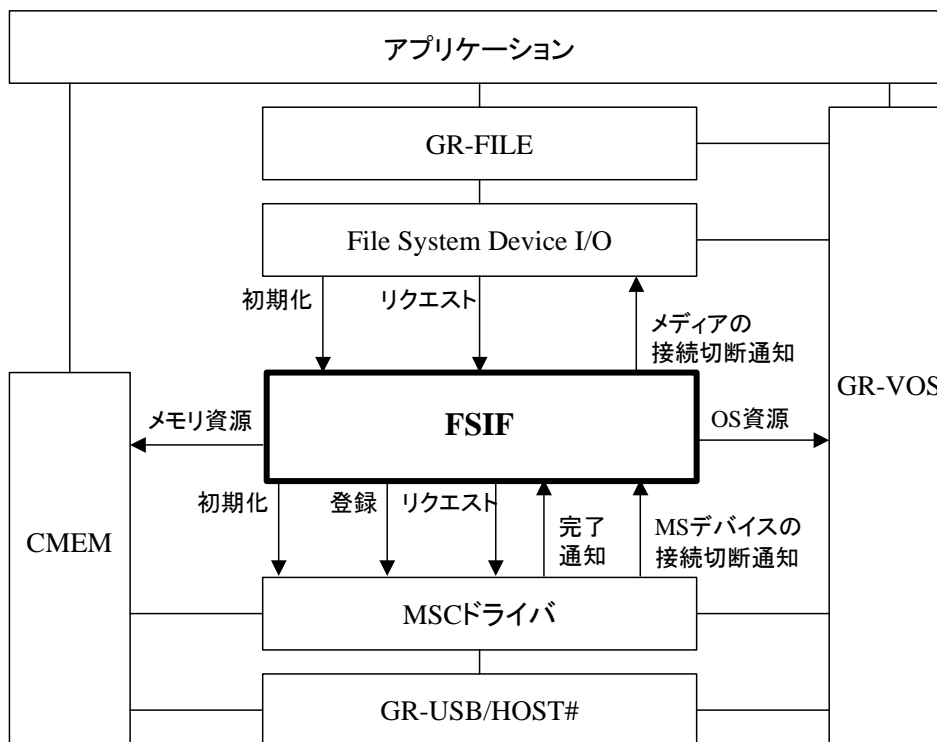


図 1-1 FSIF 関連図

#### (1) GR-FILE

弊社の提供する FAT12/16/32 に対応したファイルシステムです。

#### (2) File System Device I/O

FSIF モジュールの I/F を GR-FILE 用の I/F に変換するラップです。FSIF では

“grp\_fsif\_dev\_io.c”として提供しています。詳細につきましては『File System Device I/O モジュール リファレンスガイド』を参照ください。

※他のファイルシステムをご利用の場合、本ラップ部を作成していただく必要があります。

(3) MSC ドライバ

GR-USB/HOST#向け MSC (マストレージクラス) ドライバです。

(4) GR-USB/HOST#

USB に関する通信要求、ドライバ制御要求の API を提供するモジュールです。

(5) CMEM

CMEM (共通メモリ管理モジュール) は、GR-USB/HOST#で用いるスケジューリングリスト、制御データ・バッファ、アプリケーション・データ・バッファの領域を管理することを目的にした、共通インタフェースです。

(6) GR-VOS

仮想 OS モジュールです。

Grapeware は全て、GR-VOS の API を介して OS の資源を使用します。

ご使用される OS、環境向けにポーティングする必要があります。

## 1.3 機能

FSIF は以下の機能を有します。

(1) プラグアンドプレイ機能

FSIF では、メディアの接続を検知すると、該当するサブクラスの I/F を用いて接続処理を開始します。必要な情報を取得し、使用できる状態を確認した後に接続の通知を上位へ行います。

(2) マルチユニットデバイスのサポート

GetMaxLUN デバイスリクエストにより Unit 数を知り、全 Unit に対する接続/切断処理を行います。メディア毎に情報を管理しますので、複数 I/F を持つデバイスも個々に接続/切断の通知を行います。

(3) メディアチェック機能

リムーバブルメディアに対して、メディアの有無を確認する機能を提供します。メディアカードなどの抜き差しを自動的に検出します。

(4) セクタ単位でのリード/ライト

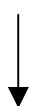
メディアに対しセクタ単位でのアクセス機能を提供します。



## 2 開発手順

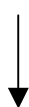
本モジュールの使用方法の基本的な流れを以下に記します。

### インストール



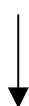
納品された CD から、開発用 PC へインストールします。  
インストールの手順に関しましては「3インストール手順」を参照ください。

### 動作環境の構築



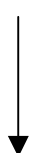
本モジュールを使用するのに必要な周辺モジュールの準備をします。  
詳しくは「5.1動作環境の準備」を参照ください。

### ポーティング



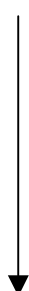
本モジュールを使用する環境に合わせてポーティングします。  
ポーティングする項目、手順などに関しましては、「5.2ポーティング手順」を参照ください。

### ライブラリファイル作成



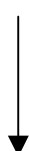
本モジュールのライブラリファイルを作成します。  
ライブラリファイルの作成手順などに関しましては、「5.3ライブラリファイル作成手順」を参照ください。

### サンプルアプリケーション動作



本モジュールを使用したアプリケーションを作成するためのサンプルとして、サンプルアプリケーションが付属しています。これは同時に、ポーティングした本モジュールの基本的な動作確認も行ないます。従って、アプリケーションを作成する前に、一度サンプルアプリケーションを動作させて頂くことを推奨します。  
サンプルアプリケーションの実行ファイル作成手順、内容などに関しましては、「6.3実行ファイル作成手順」、「6.4サンプルアプリケーション」を参照ください。

### アプリケーション作成



本モジュールを使用したアプリケーションを作成します。  
アプリケーションに提供される API 仕様、アプリケーションの作成方法に関しましては、「6.1API仕様」、「6.2アプリケーション作成」を参照ください。

### 動作確認

実装した本モジュールの動作確認を行います。  
動作確認方法に関しましては、「7動作確認方法」を参照ください。

### 3 インストール手順

CD 中にある” fsif”フォルダ以下を、開発に使用する PC の適切な場所にコピーして頂くだけで、インストールは完了です。

## 4 フォルダ構成

FSIF のフォルダ構成は以下のとおりです。

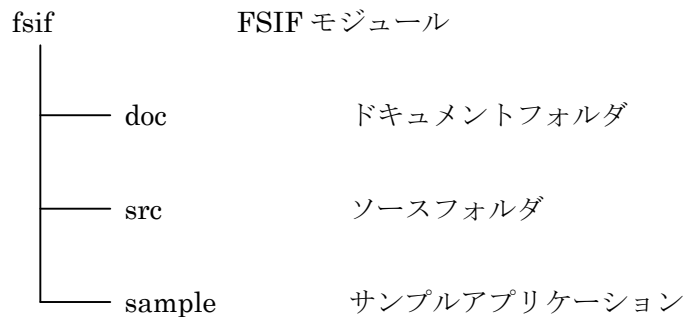


表 4-1 ファイル一覧

fsif	
relnote.txt	リリースノート
GR-USB HOST# FSIF モジュール ユーザーズガイド.pdf (本ファイル)	
doc	
GR-USB HOST# FSIF モジュール API 仕様書.pdf	
File System Device IO モジュール リファレンスガイド.pdf	
src	
grp_fsif.h	FSIF ヘッダファイル
grp_fsif.h	FSIF モジュール
grp_fsif_config.h	FSIF 設定ファイル
grp_fsif_debug.h	FSIF デバッグヘッダファイル
grp_fsif_local.h	FSIF ローカルヘッダファイル
grp_fsif_dev_io.c	GR-FILE 用 File System Device I/O (サンプル)
sample	
fsiftest.c	サンプルアプリケーション
fsiftest.h	サンプルアプリケーションヘッダファイル

## 5 準備

本章では、FSIF を使用するための準備について説明します。

### 5.1 動作環境の準備

FSIF を使用するには、以下の周辺モジュールの準備が必要となります。それぞれのモジュールの準備について説明します。

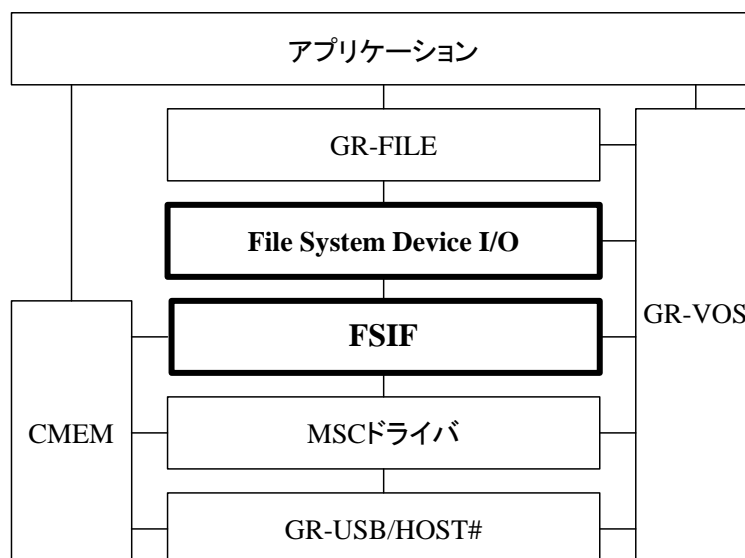


図 5-1 周辺モジュール

#### (1) GR-VOS

使用する OS、環境向けにポーティングし、GR-VOS に付属のテストアプリケーションを用いて事前に動作確認を済ませておく必要があります。

また、システム全体で使用する OS 資源の数を把握し、それを満たすだけの OS 資源を提供できるよう設定しておく必要があります。

FSIFが使用するOS資源に関しましては、「OS資源」を参照ください。

MSC ドライバ、GR-USB/HOST#が使用する OS 資源に関しましては、それぞれのユーザーズガイドを参照ください。

#### (2) CMEM

CMEM（共通メモリ管理モジュール）は関数のインタフェースのみを提供しているため、内部処理は全てポーティングを行う必要があります。

また、システム全体で使用する CMEM のメモリ資源のサイズ・数を把握し、それを満たすだけのメモリ資源を提供できるよう設定しておく必要があります。

FSIFが使用するCMEMのメモリ資源に関しましては、「メモリ資源」を参照ください。

MSC ドライバ、GR-USB/HOST#が使用する CMEM のメモリ資源に関しましては、それぞれのユーザーズガイドを参照ください。

(3) GR-USB/HOST#

使用する環境向けにポーティングし、GR-USB/HOST#に付属のテストアプリケーションを用いて事前に動作確認を済ませておく必要があります。

(4) MSC ドライバ

使用する環境向けにポーティングし、MSC ドライバに付属のテストアプリケーションを用いて事前に動作確認を済ませておく必要があります。

※File System Device I/O

GR-FILE を使用する場合には付属のファイルをご利用ください。

GR-FILE 以外のファイルシステムをお使いの場合、ファイルシステム付属のドキュメントにてデバイスドライバの作りをご確認いただき、FSIF の提供する API にて専用のファイルを作成ください。

## 5.2 ポーティング手順

FSIF を使用するには、FSIF 自身を環境に合わせてポーティングする必要があります。ここでは、FSIF のポーティング項目、仕様などについて説明します。

### 5.2.1 ポーティング対象ファイル

FSIF のポーティング対象となるファイルを以下に記します。

・ grp\_fsif\_config.h

### 5.2.2 定義値の設定

#### (1) 最大ユニット数

FSIF で管理できる最大ユニット数の設定を行います。ご利用される環境に合わせて設定してください。

表 5-1 最大ユニット数

#	定義名	初期設定	内容
1	GRP_FSIF_MAX_UINT	1	最大ユニット数 (0 より大きい値を設定してください)

※CMEM モジュール内の GRP\_CMEM\_BNUM\_FSIF は本定義値以上に設定してください。

#### (2) 最大アクセス数

同時にアクセス可能な最大タスク数の設定を行います。

ご利用される環境にあわせて設定してください。

表 5-2 最大アクセス数

#	定義名	初期設定	内容
1	GRP_FSIF_MAX_ACCESS	1	最大アクセス数 (0 より大きい値を設定してください)

#### (3) タスクスタックサイズ

FSIF 内部で使用するタスクのスタックサイズの設定を行います。

基本的には変更する必要はありませんが、ご利用される環境により消費するスタックサイズが異なりますので、必要に応じて再設定してください。

表 5-3 タスクスタックサイズ

#	定義名	初期設定	内容
1	GRP_FSIF_MAIN_TASK_STK	2048	FSIF メインタスク (_grp_fsif_MainTask) のスタックサイズ

(4) サブクラス・プロトコル使用設定

マストレージクラスのサブクラス、プロトコルの組み合わせについて、使用する／使用しないの設定を行います。

表 5-4 サブクラス・プロトコル使用設定

#	定義名	初期設定	内容
1	GRP_FSIF_SCS_BOT	GRP_USB_TRUE	サブクラス : SCSI プロトコル : BOT 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
2	GRP_FSIF_ATA_BOT	GRP_USB_FALSE	サブクラス : ATAPI プロトコル : BOT 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
3	GRP_FSIF_SFF_BOT	GRP_USB_FALSE	サブクラス : SFF8070-i プロトコル : BOT 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
4	GRP_FSIF_UFI_BOT	GRP_USB_FALSE	サブクラス : UFI プロトコル : BOT 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
5	GRP_FSIF_SCS_CBI	GRP_USB_FALSE	サブクラス : SCSI プロトコル : CBI 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
6	GRP_FSIF_ATA_CBI	GRP_USB_FALSE	サブクラス : ATAPI プロトコル : CBI 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
7	GRP_FSIF_SFF_CBI	GRP_USB_FALSE	サブクラス : SFF8070-i プロトコル : CBI 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
8	GRP_FSIF_UFI_CBI	GRP_USB_FALSE	サブクラス : UFI プロトコル : CBI 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない

<次ページへ続く>

9	GRP_FSIF_SCS_CB	GRP_USB_FALSE	サブクラス : SCSI プロトコル : CB 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
10	GRP_FSIF_ATA_CB	GRP_USB_FALSE	サブクラス : ATAPI プロトコル : CB 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
11	GRP_FSIF_SFF_CB	GRP_USB_FALSE	サブクラス : SFF8070-i プロトコル : CB 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない
12	GRP_FSIF_UFI_CB	GRP_USB_FALSE	サブクラス : UFI プロトコル : CB 上記組み合わせを使用するか設定してください GRP_USB_TRUE : 使用する GRP_USB_FALSE : 使用しない

※MSC ドライバの「GRP\_MSC\_REG\_MAX」は、上記の GRP\_USB\_TRUE に設定した合計数以上に設定する必要があります。

※GR-USB/HOST#への登録数である「GRP\_USBD\_HOST\_MAX\_REGISTER (grp\_usbc\_cfg.h)」にも影響がありますのでご注意ください。



### 5.3 ライブラリファイル作成手順

FSIF を使用する場合、まずは FSIF だけで 1 つのライブラリファイルを作成し、それをアプリケーションとリンクさせるようにすることで、整理され、快適に使用することができます。

(もちろんライブラリファイル化せず、直接アプリケーションとリンクしても構いません。)

ここでは、FSIF のライブラリファイルを作成する手順について説明します。

#### (1) コンパイル対象のファイル

fsif フォルダ以下の、以下の C ソースファイルが対象となります。

grp\_fsif.c

File System に GR-FILE をご利用の場合は、以下の C ソースファイルも対象となります。

grp\_fsif\_dev\_io.c

#### (2) インクルードパス

FSIF や、周辺モジュールの必要なファイルが参照できるよう、以下のパスを設定してください。

##### FSIF

<インストール先>%fsif%src

##### MSCドライバ

<インストール先>%ms\_c%src%com

<インストール先>%ms\_c%src%bot

<インストール先>%ms\_c%src%cbi

<インストール先>%ms\_c%src%scsi

<インストール先>%ms\_c%src%atapi

<インストール先>%ms\_c%src%ff8070i

<インストール先>%ms\_c%src%ufi

##### GR-USB/HOST#

<インストール先>%grp\_usb\_h%src%usbm%common

<インストール先>%grp\_usb\_h%src%usbm%usbd

<インストール先>%grp\_usb\_h%src%usbm%cnf\_sft

<インストール先>%grp\_usb\_h%src%hcm%cmem

##### GR-VOS

<インストール先>%gr\_vos%common

<インストール先>%gr\_vos%inc

#### (3) マクロ定義

FSIF では、コンパイル時に定義が必要なマクロは特にありません。

(4) 出力ファイル

出力ファイルを”ライブラリファイル”形式に設定してください。

ライブラリファイルの名前は特に指定はありませんが、どのモジュールのライブラリファイルか判別しやすい名前にすることを推奨します。(例 : `grp_fsif.lib`)

(5) コンパイル・リンク

(1) で示した対象ファイルを全てコンパイル、リンクし、1つのライブラリファイルを生成してください。

## 6 使用方法

本章では、FSIF を使用する方法について説明します。

### 6.1 API仕様

FSIF がアプリケーションに対して提供する API 関数を以下に記します。

#### 初期化関数

grp_fsif_Init	FSIF の初期化
---------------	-----------

#### コマンド関数

grp_fsif_WriteSector	データ書き込み処理
grp_fsif_ReadSector	データ読み込み処理

#### メディア情報取得関数

grp_fsif_GetMediaInfo	メディア情報取得処理
-----------------------	------------

#### リセット関数

grp_fsif_Reset	リセット処理
----------------	--------

#### CMEM領域管理関数

grp_fsif_GetNonCacheBuffer	CMEM 領域取得処理
grp_fsif_RelNonCacheBuffer	CMEM 領域解放処理

API 仕様の詳細に関しましては、別冊『FSIF API 仕様書』を参照ください。

## 6.2 アプリケーション作成方法

FSIF の提供する API を使用してアプリケーションを作成する方法について説明します。

### 6.2.1 必要なヘッダファイル

FSIF を使用するアプリケーションは、以下のヘッダファイルをインクルードする必要があります。

<インストール先>¥gr\_vos¥inc¥grp\_vos.h

<インストール先>¥fsif¥src¥grp\_fsif.h

<インストール先>¥ms\_c¥src¥com¥grp\_msc.h

<インストール先>¥gr\_usb\_h¥src¥usbm¥common¥grp\_usbc.h

### 6.2.2 API関数の使い方

ここでは基本的な API 関数の使い方について、サンプルアプリケーションのプログラムを基に説明します。

サンプルアプリケーションに関しましては、「6.4 サンプルアプリケーション」を参照ください。

#### (1) 初期化

```
static grp_s32 _UsbHTest_Initialize(void)
{
    (中略)

    /* Initialize GR-VOS */
    lStat = grp_vos_Init();
    if (lStat != GRP_VOS_POS_RESULT)
    {
        return 1;
    }

    (中略)

    /* Initialize GR-USB/Host */
    lStat = grp_usbc_HostInit();
    if (lStat != GRP_USBC_OK)
    {
        return 2;
    }

    (中略)

    /* FSIF Initialize */
    tInitPrm.pfnFsifNotification = _FsifTest_EventNotice_Callback;
    lStat = grp_fsif_Init(&tInitPrm);
    if (lStat != GRP_FSIF_OK)
    {
        return 4;
    }
}
```

①

```

(中略)

/* Set unregister device callback */
lStat = grp_usbc_Enable();
if( lStat != GRP_USBC_OK )
{
    return 5;
}

return USBHTEST_OK;
}

```

← ②

周辺モジュールと FSIF の初期化を行います (①)。FSIF は GR-VOS や GR-USB/HOST# の機能を使用するため、GR-VOS や GR-USB/HOST#の初期化を最初に行う必要があります。MSC ドライバの初期化や登録処理は、FSIF 初期化処理内で実行するため、アプリケーションから直接実行する必要はありません。

初期化が終わりましたら、GR-USB/HOST#の起動処理を行います (②)。この処理により、USB/Host の割込みが許可されます。

## (2) 接続・切断

```

static void _FsifTest_EventNotice_Callback(grp_s32 ulEvtnt, void *pvHdr, grp_si iIdx)
{
    if( ulEvtnt == GRP_FSIF_ATTACHED_MEDIA )
    {
        /* connect process */
        l_tDevInfo.iDevId    = iIdx;
        l_tDevInfo.pvHdr     = pvHdr;
        l_ulConnectFlag     = FSIFTEST_FLG_ON;
    }
    else
    {
        /* disconnect process */
        l_ulConnectFlag = FSIFTEST_FLG_OFF;
    }
}

```

} ③

「5.2.2定義値の設定」の「サブクラス・プロトコル使用設定」で”使用する”と設定したサブクラスとプロトコルの組み合わせのマスストレージデバイスが接続・切断されると、初期化時に指定したコールバック関数が呼び出されます。

接続時には、パラメータで渡されるメディア情報を保存する必要があります (③)。この情報は以降の転送などで使用します。

### (3) メディア情報取得

```
static grp_s32 _FsifTest_GetMediaInfo(void)
{
    grp_s32          lStat;
    grp_fsif_media_info  tMediaInfo;

    /* Get Media Info */
    lStat = grp_fsif_GetMediaInfo(l_tDevInfo.pvHdr, &tMediaInfo); ← ④
    if (lStat == GRP_FSIF_OK)
    {
        (中略)

        /* OK */
        l_tDevInfo.ulSectorSize = tMediaInfo.ulSectorSize;
        l_tDevInfo.ulSectorNum  = tMediaInfo.ulSectorNum;

        (中略)
    }
}
```

メディアに関する情報を取得するために、`grp_fsif_GetMediaInfo` が用意されています (④)。この API 関数を使用することで、指定したメディアの 1 セクタサイズ、総セクタ数を取得することができます。

### (4) リード

```
static grp_s32 _FsifTest_ReadWrite(void)
{
    (中略)

    /* Get data buffer for save */
    lStat = grp_cmem_BlkJGet(GRP_CMEM_ID_TEST, (void **)&pucSaveDataBuf); ← ⑤
    if (lStat == GRP_CMEM_OK)
    {
        /* Read data */
        lStat = grp_fsif_ReadSector(l_tDevInfo.pvHdr, TEST_START_SECNUM, ulSecCnt, ← ⑥
                                   pucSaveDataBuf);
        if (lStat == GRP_FSIF_OK)
        {
            (中略)

            }

        grp_cmem_BlkJRel(pucSaveDataBuf); ← ⑦
    }
}
```

FSIF では、通信に使用するバッファ領域を CMEM から確保する必要があります (⑤)。データ転送を伴う API 関数 (`grp_fsif_ReadSector`、`grp_fsif_WriteSector`) を発行する際は、その前に CMEM からバッファ領域を確保してください。

**READ** ではセクタ単位でメディアからデータを読み出すことができます。メディアハンドルで指定したメディアの指定したセクタ番号から、指定したセクタ数分のデータを、指定したバッファ領域に読み出します。データ読み出しが完了するまでは、本関数から戻りません (⑥)。

**READ** が完了したら、不要になったバッファ領域は必ず解放してください (⑦)。

### 6.3 実行ファイル作成手順

ここでは FSIF を使用したアプリケーションの実行ファイルを作成する手順について説明します。  
なお、「ライブラリファイル作成手順」に従って、既に FSIF のライブラリファイルを作成していることを前提とします。また、FSIF を使用するために必要な周辺モジュールに関しても、それぞれのモジュール毎にライブラリファイルが作成されていることを前提とします。

#### (1) リンク対象のファイル

以下のファイルがリンクの対象となります。

- ・アプリケーションソースファイル
- ・FSIF のライブラリファイル
- ・各周辺モジュールのライブラリファイル (GR-VOS や GR-USB/HOST#、MSC ドライバなど)
- ・その他のライブラリ (OS や標準ライブラリなど)

#### (2) インクルードパス

アプリケーション、FSIF、周辺モジュールのそれぞれの必要なヘッダファイルが参照できるよう、以下のパスを設定してください。

##### FSIF

<インストール先>¥fsif¥src

##### MSCドライバ

<インストール先>¥ms\_c¥src¥com

<インストール先>¥ms\_c¥src¥bot

<インストール先>¥ms\_c¥src¥cbi

<インストール先>¥ms\_c¥src¥scsi

<インストール先>¥ms\_c¥src¥atapi

<インストール先>¥ms\_c¥src¥sff8070i

<インストール先>¥ms\_c¥src¥ufi

##### GR-USB/HOST#

<インストール先>¥grp\_usb\_h¥src¥usbm¥common

<インストール先>¥grp\_usb\_h¥src¥usbm¥usbd

<インストール先>¥grp\_usb\_h¥src¥usbm¥cnf\_sft

<インストール先>¥grp\_usb\_h¥src¥hcm¥cmem

##### GR-VOS

<インストール先>¥gr\_vos¥common

<インストール先>¥gr\_vos¥inc

#### (3) 出力ファイル

出力ファイル形式を、環境に合わせた実行ファイルの形式に設定してください。



(4) リンク

アプリケーションをコンパイルしその他のライブラリとリンクして、1つの実行ファイルを生成してください。

## 6.4 サンプルアプリケーション

FSIF に付属のサンプルアプリケーションについて説明します。

### 6.4.1 ファイル構成

サンプルアプリケーションファイルを以下に記します。

- fsiftest.c
- fsiftest.h

### 6.4.2 使用方法

アプリケーションタスクから、サンプルアプリケーションの **FSIF** テストメイン関数 (**FsifTest\_Main**) を呼び出してください。

サンプルアプリケーション内部では、**GR-VOS** の初期化と自タスクの登録、**GR-USB/HOST#**の初期化も行っております。

また、**HUB** クラスドライバ（オプションにて提供）を使用される場合は、コンパイル時のマクロ定義に”**FSIF\_TEST\_USE\_HUB**”を追加してください。サンプルアプリケーション内部で **HUB** クラスドライバの初期化処理を行います。

### 6.4.3 動作内容

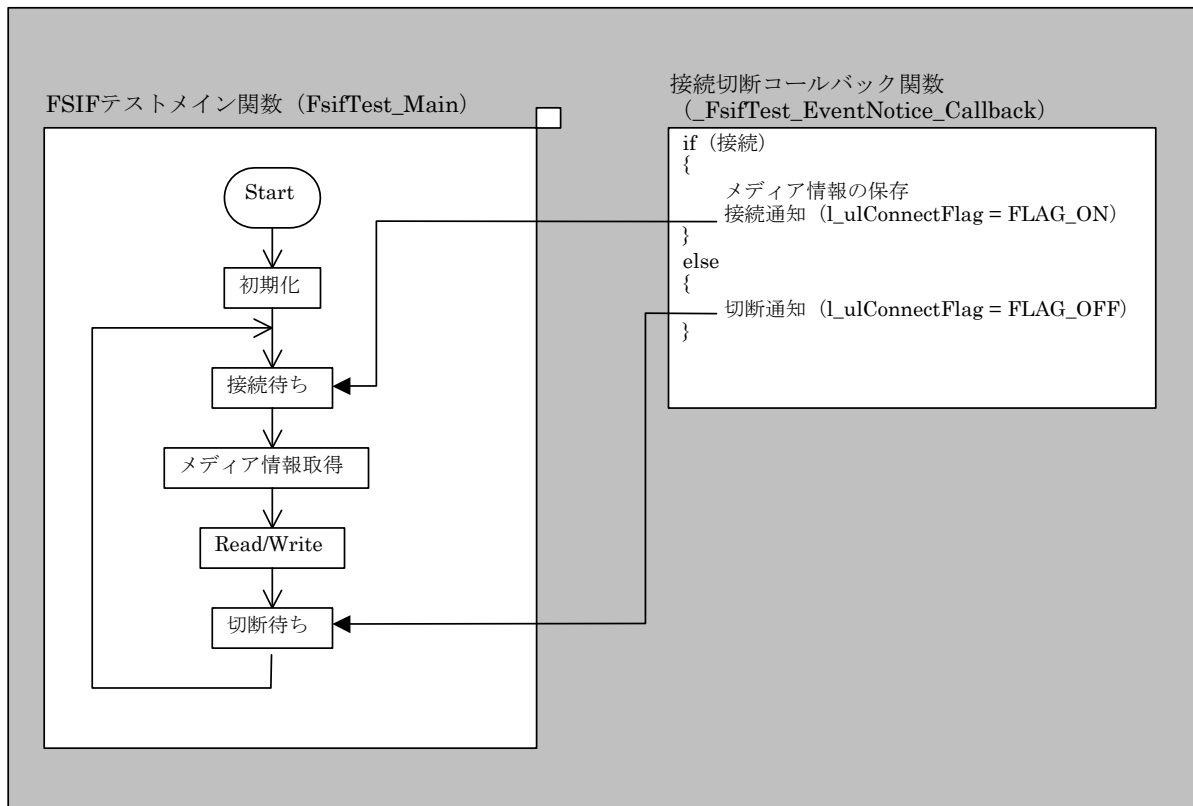


図 6-1サンプルアプリ動作内容

FSIF テストメイン関数は、初期化、メディア情報取得、Read/Write と一連の動作のテストを行います。途中でエラーが発生しますと、そこで無限ループに入り止ります。従って、Read/Write テストが正常に終了したことは、一連のテストが正常に終了したことを意味します。

一連のテストが終了すると切断待ちで止まりますので、メディアを切断・再接続して頂くことで、再度テストを実施することができます。

## 7 動作確認方法

本章では、ポーティングした FSIF モジュールの動作確認方法について説明します。

### 7.1 初期化処理の確認

FSIF モジュールでは、`grp_fsif_Init` (`grp_fsif.c`) にて初期化が行われます。

以下の項目について正常に動作が行われているかご確認ください。

#### Check

- ☐ `_grp_fsif_InitUnitInfo` でエラーとなっていないか  
⇒エラーとなる場合、CMEM モジュールの設定についてご確認ください。
- ☐ `_grp_fsif_AccessQueInit` でエラーとなっていないか  
⇒エラーとなる場合、リソース数が足りているか GR-VOS の設定をご確認ください。
- ☐ `_grp_fsif_CreateVos` でエラーとなっていないか  
⇒エラーとなる場合、リソース数が足りているか GR-VOS の設定をご確認ください。
- ☐ `_grp_fsif_RegistSubClass` でエラーとなっていないか  
⇒エラーとなる場合、MSC ドライバのポーティングをご確認ください。

### 7.2 最初のデバイス接続処理の確認

初期化、登録、GR-USB/HOST#の起動処理 (`grp_usbc_Enable`) 後、最初にデバイスを接続した際に、以下の項目について正常に動作が行われているかご確認ください。

#### Check

- ☐ `_grp_fsif_Notification` (`grp_fsif.c`) が呼び出されるか  
⇒呼び出されない場合、接続したデバイスのプロトコルおよびサブクラスを確認し、同組み合わせを設定しているかご確認ください。また、GR-USB/HOST#のポーティングもご確認ください。

## Appendix A 内部構造

FSIF の内部構造について説明します。

### A.1 内部構成

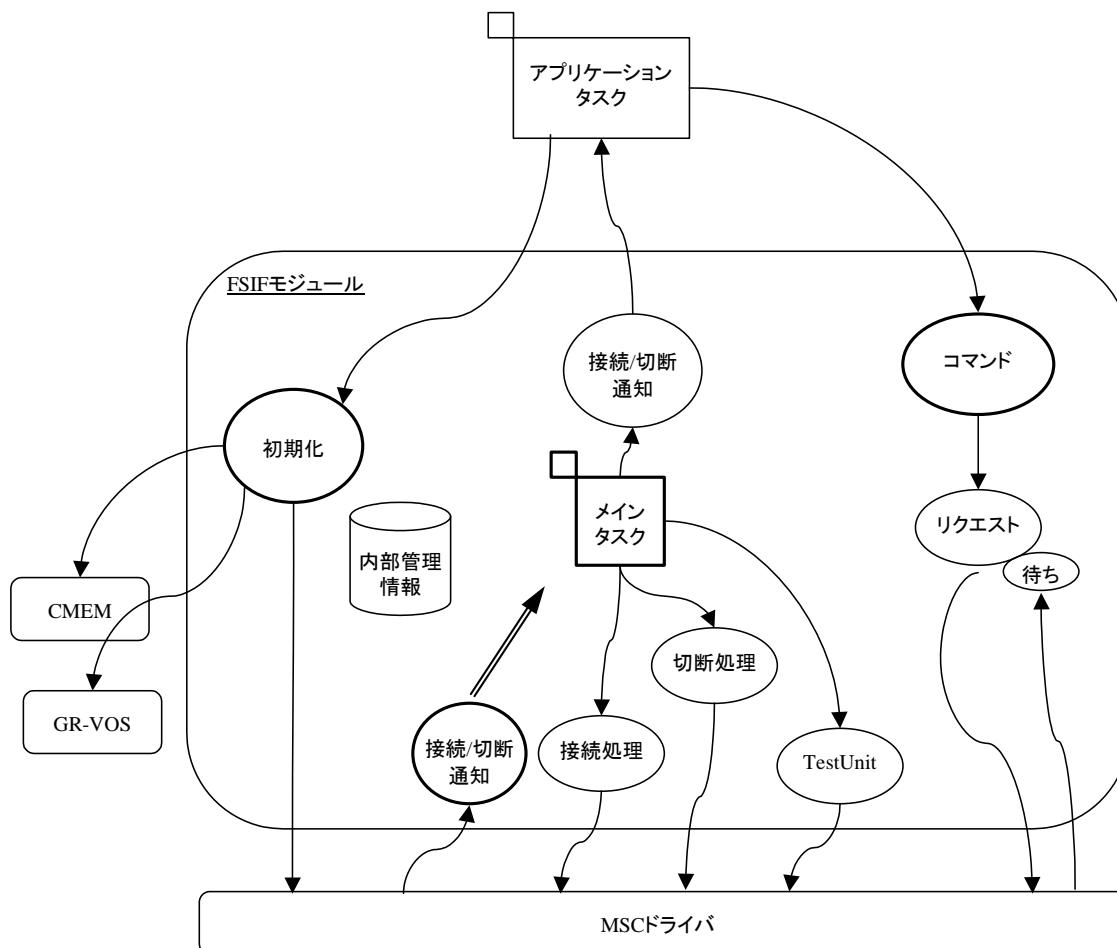


図 A-1 FSIF モジュールの構成

#### (1) 初期化

内部管理情報の初期化および CMEM、VOS の資源の生成などを行います。また MSC ドライバの初期化関数の実行および登録処理を行います。

登録処理は BOT/CBI/CB と上位のサブクラスの必要な組み合わせ分実行されます。

#### (2) 接続/切断通知

MS デバイスが接続または切断されると、コールバック通知により処理が実行されます。

通知を受けた後は、メインタスクにて処理が実行されます。

### (3) メインタスク

メインタスクは MSC ドライバによる接続/切断の通知により起動します。

接続であれば、一連の処理 (Inquiry、ReadCapacity など) を実行し、また複数ユニットを持つ場合には、全てのユニットに対して同様の処理を行います。全てを確認したところで、メディアがある場合には上位へ接続を通知します。

切断については、接続中のフラグが設定されている場合のみ、上位へ切断を通知します。

なお接続されているユニット (RemovableMedia のみ) 全てに TestUnitReady コマンドを定期的に発行します。

この時、接続状態でありながらメディアなし、またその逆となった場合に上位へ接続や切断を通知します。

### (4) コマンド

上位からの各コマンドは、FSIF モジュール内で完了通知を待ちます。またエラーとなった場合には、RequestSense コマンドを発行し、その結果を上位へ返します。

## A.2 接続時のフローチャート

弊社ドライバでは、MS デバイスの接続を確認した後に、以下の処理を行います。

下図のとおり、TestUnitReady コマンドを発行した後にアクセスできるメディアとして上位へ接続を通知します。

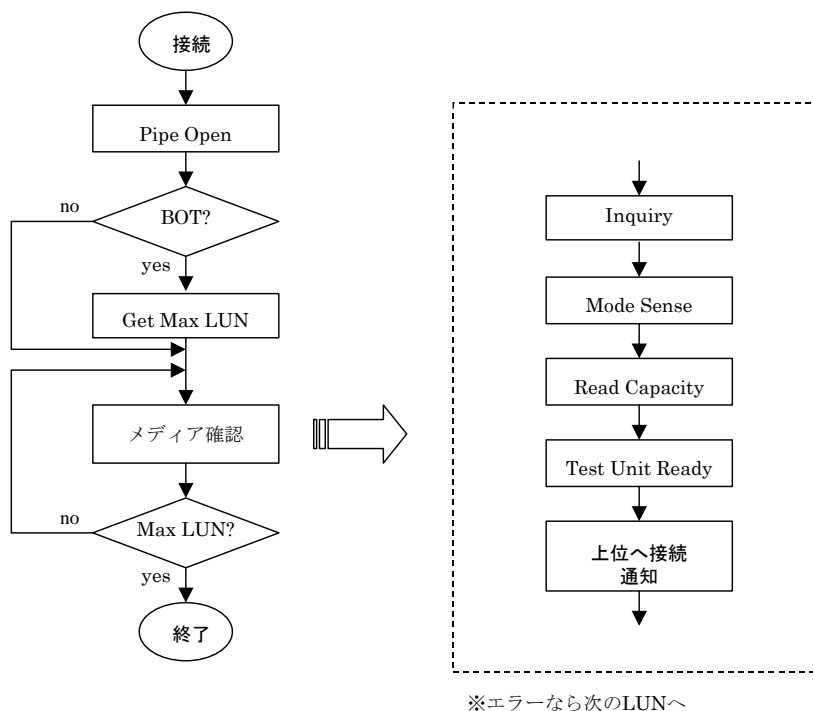


図 A-2 接続時のフローチャート

### A.3 OS資源

FSIF で使用する OS 資源は以下のとおりです。

セマフォ : 6  
イベントフラグ : 1  
キュー : 2+GRP\_FSIF\_MAX\_ACCESS  
タスク : 1

#### (1) セマフォ

#	名前	初期カウント値	用途
1	"sFSIF1"	1	イベント情報リスト排他制御用
2	"sFSIF2"	1	メディア情報リスト排他制御用
3	"sFSIF3"	1	メディアアクセス排他制御用
4	"sFSIF4"	1	メディア情報排他制御用
5	"sFSIF5"	1	非キャッシュバッファ領域排他制御用
6	"sFSIF6"	GRP_FSIF_MAX_ACCESS	上位アプリアクセス制御用

#### (2) イベントフラグ

#	名前	用途
1	"fFSIF"	接続切断イベント通知用

#### (3) キュー

#	名前	メッセージ サイズ (バイト)	メッセージ数	用途
1	"qFSIF1"	4	1	メインタスク通信完了通知用
2	"qFSIF2"	4	1	ポーリングタスク通信完了通知用
3	"qFSIFMX0" ※1	4	1	上位アプリ通信完了通知用

※1 GRP\_FSIF\_MAX\_ACCESS の定義されている設定値分のキューを、名前を変えて作成します。

キューの名前は末尾の「0」を、「0」～「9」、「A」～のように変更し作成されます。

#### (4) タスク

#	名前	エントリ関数	優先度	スタックサイズ (バイト)
1	"tFSIF1"	_grp_fsif_MainTask	GRP_VOS_PRI_NORMAL	2048

#### A.4 メモリ資源

FSIF で使用する CMEM のメモリ資源は以下のとおりです。

#	名前	ID	サイズ	個数
1	GRP_CMEM_xxx_FSIF	0x80000009	36	5
2	GRP_CMEM_xxx_FSIF_MAX_LUN	0x8000000A	4	1
3	GRP_CMEM_xxx_FSIF_NC_BUF	0x8000000B	2048	1

※xxx には、ID、BSIZE（サイズ）、BNUM（個数）が入ります。

※#1 の個数は GRP\_FSIF\_MAX\_UNIT 以上の値に設定してください。



## Appendix B カスタマイズ

FSIF のカスタマイズについて説明します。

### (1) GRP\_FSIF\_MAX\_ACCESS

上位アプリケーションから同時にコマンドを実行できる数を設定します。初期設定値で 2 つのタスクが同時にコマンドを実行すると、一つのタスクはセマフォによりブロックされます。

表 A2-1 同時アクセス数

#	定義名	初期設定	備考
1	GRP_FSIF_MAX_ACCESS	1	最大アクセス数 (0 より大きい値を設定してください)

※同じデバイスへのアクセスは MSC ドライバにてキューイングされるため、回線上では 1 アクセスとなります。

### (2) GRP\_FSIF\_MAX\_NTC\_QUEUE

MSC のイベント通知 CB から、FSIF の MainTask へイベント情報を受け渡す際に使用するキューのメッセージ数を設定します。設定値を超えた接続通知が発生した場合、その通知は無視されます。

表 A2-2 キューのメッセージ数

#	定義名	初期設定	備考
1	GRP_FSIF_MAX_NTC_QUEUE	1	同時アクセスデバイス数 (0 より大きい値を設定してください)

### (3) GRP\_FSIF\_MAX\_DATA\_LEN

FSIF から MSC ヘデータ転送要求をかける時のサイズを設定します。FSIF 内でこの分の領域を確保している訳ではなく、単にこのサイズに分けて要求をかけています。

表 A2-3 最大データ長

#	定義名	初期設定	備考
1	GRP_FSIF_MAX_DATA_LEN	64*1024	最大データ長 (0 より大きい値とし、またセクタ長の倍数としてください)

(4) GRP\_FSIF\_TEST\_UNIT\_WAIT\_TIME

FSIF タスクで接続/切断状態のメディアを検知するためのコマンドを発行する間隔を設定します。

表 A2-4 ポーリング間隔

#	定義名	初期設定	備考
1	GRP_FSIF_TEST_UNIT_WAIT_TIME	2000 (2 秒)	ポーリング間隔 (0 より大きい値を設定してください)

※他のコマンドが実行されている場合、それらの処理を優先させるため、設定した時間よりポーリング間隔が長くなる場合があります。

## Appendix C 注意事項

FSIF をご利用にあたっての注意事項について記します。

### C.1 データバッファ領域

FSIF モジュールでデータ転送を伴う通信を要求する場合、ユーザーは **CMEM**（共通メモリ管理モジュール）よりバッファ領域を確保し、その領域を使用する必要があります。

※**CMEM** では、バッファ領域のキャッシュ/非キャッシュの問題や、特定のホストコントローラに存在するバウンダリ問題の対策などのメモリ管理を対応するため、内部ロジックをポーティングしていただく必要があります。

FSIF モジュールでは、**CMEM** 領域の取得/解放の API を用意していますので、セクタのリード/ライトを行う際には、こちらの関数もご利用いただけます。

## Appendix D トレースログ機能

弊社ドライバには、トレースログ機能がございます。

トレースログ機能は、弊社ドライバ内の動きを関数単位で記録する機能です。

不具合などが発生した際にログの取得をお願いする場合がございます。

### ログ取得手順

- (1) `grp_usbc_dbg.h` の「GRP\_DBG\_MAX\_TRACE」を設定してください。デフォルトでは「1024」となっています。
- (2) マクロ定義に「GRP\_USBC\_DBG\_MDL」および「GRP\_FSIF\_DEBUG\_LOGGING」を追加し、ドライバのライブラリファイル、およびアプリケーションの実行ファイルを作成します。  
(`grp_usbc_dbg.h`、`grp_fsif_debug.h` ファイル内にそれぞれ定義を追加してもかまいません。)
- (3) プログラムを実行し、問題発生後に停止します（この間のトレースログが記録されます）。
- (4) `grp_usbc_dbg.c` 内の「`g_iDbgMdlCnt`」の値と、「`g_aucDbgMdlTbl`」の値をダンプし、ご連絡ください。（デフォルトの設定ですと、 $4 \times 1024 = 4096$  バイトの情報量となります）

他のモジュールのログを取得する場合は、各ユーザーズガイドを参照ください。

## Appendix E Grapeware技術サポート

### E.1 サポート時間

午前 10 : 00 より午後 5 : 30 まで  
(土日祝日、年末年始休日、指定休日を除く)

### E.2 質問手順

#### 問題発生



製品、マニュアルについて不明な点がある。あるいは製品が仕様どおりに動作しない、不具合がある、などの問題に直面された場合。

#### 保守サービス有効期限の確認



Grapeware 技術サポートに問い合わせる前に、保守サービス有効期限を確認ください。  
保守サービス有効期限内である場合、次のステップに進んでください。  
保守サービス有効期限を過ぎている場合、弊社営業担当者まで問い合わせください。

#### Grapeware技術サポートに問い合わせ

Grapeware 技術サポートへの問い合わせはメール・電話・FAX で受け付けております。

メール : [gr@support.grape.co.jp](mailto:gr@support.grape.co.jp)

電話 : 045-222-3762

FAX : 045-222-3760

※問題や、その対応の正確なログを残すために、弊社ではメールを推奨しております。

問い合わせの際は、以下の情報をお知らせください。

#### <情報>

- ・ カスタマー番号
- ・ 御社名、担当者様名
- ・ 問題の発生する製品名、バージョン
- ・ 問題の内容 (問題発生手順、頻度など)

また、開発環境 (CPU、ツールなど) に関する情報も併せてお知らせ頂けると、サポートをよりスムーズに進められる場合がございます。

別紙『質問フォーム.txt』もご利用ください。

問い合わせに対し、2 営業日以内に回答、もしくは (回答まで至らなかった場合は) 状況報告をさせていただきます。

**GR-USB/HOST#  
FSIF モジュール  
ユーザーズガイド**

発行年月：2015 年 12 月 第 1.02 版

発行：株式会社グレープシステム

E-Mail : [gr@support.grape.co.jp](mailto:gr@support.grape.co.jp)

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

Copyright (C) 2007-2015 Grape Systems, Inc.

All rights reserved.