

μ PD70F3769

マスタレージクラスサンプル・プログラム

MSC サンプル・ドライバ

ユーザーズ・マニュアル

(第 1.0 版)

テセラ・テクノロジー株式会社

- 本資料の内容は予告なく変更することがあります。
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

目次

1. 概要.....	3
1.1. 制限事項.....	3
2. 構成.....	4
2.1. システム構成.....	4
2.2. ファイル構成.....	5
3. 開発環境.....	6
4. ROM・RAM サイズ.....	7
5. マスストレージクラス (MSC)	8
5.1. MSC の位置づけ.....	11
6. SCSI コマンド処理.....	12
6.1. SCSI コマンド処理部の位置づけ.....	12
6.2. 対応 SCSI コマンド.....	12
7. 処理フロー.....	13
7.1. 初期化処理.....	13
7.1.1. CPU 初期化処理.....	14
7.1.2. USB 初期化処理.....	16
7.2. エンドポイント監視処理.....	18
7.2.1. エンドポイント0 監視処理.....	18
7.2.2. エンドポイント1 監視処理.....	20
7.3. SCSI コマンド処理.....	21
8. 関数仕様.....	23
9. レジスタ設定.....	38
9.1. CPU 関連レジスタの設定.....	39
9.2. USB 関連レジスタの設定.....	41
9.3. ディスクリプタ設定.....	47
10. マスストレージクラスサンプル・プログラムの構成.....	53
10.1. ドライバのインストール.....	54

1. 概要

本書は μ PD70F3769 内蔵の USB ファンクション・コントローラ用サンプル・プログラム、マスタトレージクラス(MSC)サンプル・ドライバについて説明するものです。

サンプル・ドライバの主な機能を以下に示します。

- ・ FS(フル・スピード:12Mbps)デバイスとして動作します。
- ・ バス・パワー・デバイスとして動作します。
- ・ ホスト接続時、マスタトレージクラス・バルクオンリーデバイスとして認識されます。
- ・ ホスト側からディスクフォーマットすることで任意のファイルシステム形式にフォーマットすることが出来ます。
- ・ ファイルやフォルダ等のデータを内蔵 RAM に書き込むことが出来ます。
- ・ 書き込んだファイルやフォルダを内蔵 RAM から読み出すことが出来ます。

注: データ領域は、内蔵 RAM に 24Kbyte 確保しています。このため、デバイス本体の電源 OFF、Reset SW 押下でデータが初期化されます。

1.1. 注意事項

本「MSC サンプル・ドライバ」をお客様の装置に組み込む場合は、お客様が所有するベンダーIDに変更してください。

2. 構成

2.1. システム構成

サンプル・プログラムのシステム構成を次に示します。

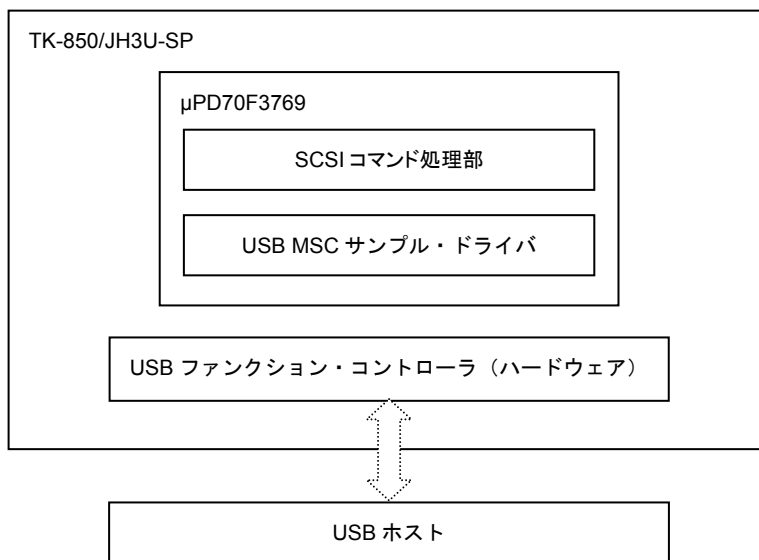


図 2-1 サンプル・プログラムのシステム構成

2.2. ファイル構成

サンプル・プログラムのファイル構成を次に示します。

【source】

ファイル名	処理内容
main.c	メインルーチン、初期化
scsi_cmd.c	SCSI コマンド処理
usbf850.c	USB 初期化、エンドポイント制御、バルク転送、コントロール転送
usbf850_storage.c	MSC 固有処理

【include】

ファイル名	処理内容
errno.h	エラーコード定義
main.h	main.c 関数プロトタイプ宣言
RegDef.h	レジスタ定義
scsi.h	SCSI 関連マクロ定義
Types.h	ユーザ型宣言
usbf850.h	usbf850.c 関数プロトタイプ宣言
usbf850_storage.h	usbf850_storage.c 関数プロトタイプ宣言
usbstrg_desc.h	ディスクリプタ定義
usbstrg_sfr.h	USB ファンクション・コントローラ用レジスタアクセス用マクロ定義

3. 開発環境

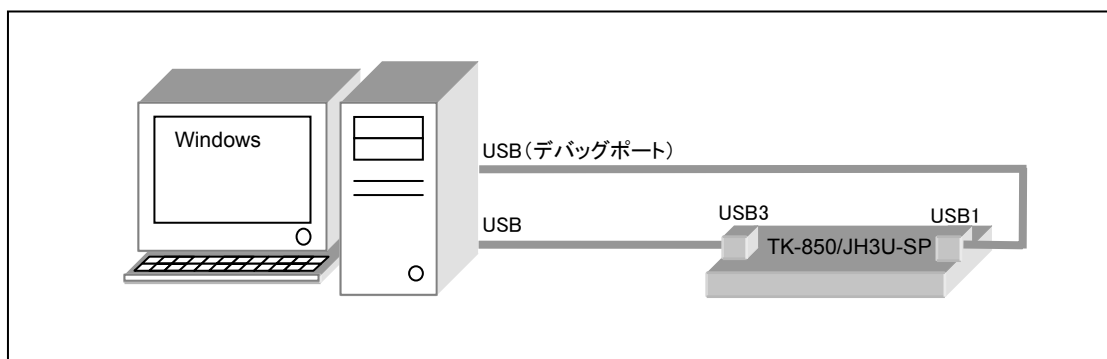


図 3-1 開発環境構築例(ハードウェア)

4. ROM・RAM サイズ

μ PD70F3769 の ROM/RAM 容量は以下の通りです。

ROM 容量: 512 [Kbyte]

RAM 容量: 56 [Kbyte]

サンプル・ドライバで使用する ROM/RAM サイズは以下の通りです。

※ベクタテーブルサイズは除く

ROM サイズ :約 7.9 [Kbyte]

RAM サイズ :約 25.9 [Kbyte]

※内 24Kbyte をストレージ用データ領域として使用。

5. マスストレージクラス(MSC)

本サンプル・ドライバはマスストレージクラス(MSC)のバルクオンリー方式です。バルクオンリー方式では、コマンドやステータス、実際のデータ転送まで全てをバルク転送で行います。コマンドはバルク OUT 転送により、ホストからデバイスに対して送信されます。これは、Command Block Wrapper(CBW)という仕様で定義されています。コマンドがデータ転送を伴うものであった場合、引き続きバルク IN、バルク OUT 転送によりデータの入出力動作が実行されます。最後に、ステータスとしてバルク IN 転送により、デバイスからホストに対してデータを送信します。これは Command Status Wrapper(CSW)という仕様で定められています。USB のマスストレージクラス(MSC)仕様に関しては、MSC 規格書「Universal Serial Bus Mass Storage Class Bulk-Only Transport Revision1.0」を参照してください。

サンプル・ドライバで対応するクラスリクエストは次の通りです。

・GET_MAX_LUN

マスストレージデバイスの論理装置数(Logical Unit Number)を取得する為のリクエストです。

・MASS_STORAGE_RESET

マスストレージデバイスと関連するインターフェースをリセットする為のリクエストです。

CBW フォーマット

Byte	Bit	7	6	5	4	3	2	1	0
0-3		dCBWSignature							
4-7		dCBWTag							
8-11		dCBWDataTransferLength							
12		bmCBWFlags							
13		Reserved				bCBWLUN			
14		Reserved			bCBWCBLength				
15-30		CBWCB							

dCBWSignature

シグネチャ。43425355H(リトルエンディアン)固定です。

dCBWTag

ホストで任意に番号を定義するタグ。コマンドとステータスを対応させます。

dCBWDataTransferLength

データフェーズで転送するデータの長さを指定します。データが無い場合は0です。

bmCBWFlags

転送方向。bit7:0=バルク OUT、1=バルク IN。他のビットは0です。

bCBWLUN

1 つの USB デバイスに複数のドライブが接続されている場合、そのドライブ番号を指定します。

bCBWCBLength

コマンドパケットの長さ。

CBWCB

コマンドパケットデータ。

CSW のフォーマット

Byte \ Bit	7	6	5	4	3	2	1	0
0-3	dCSWSignature							
4-7	dCSWTag							
8-11	dCSWDataResidue							
12	bCSWStatus							

dCSWSignature

シグネチャ。53425355H(リトルエンディアン)固定です。

dCSWTag

コマンド転送時の dCBWTag に対応させることで、ホストがフェーズの一致を確認することが出来ます。

dCSWDataResidue

「残りのデータ」を意味します。データ転送中にエラーが発生した場合など、ホストが要求したデータよりもデバイスが応答したデータの方が短い場合、ここに残りのデータ量が設定されます。このため、ステータス (bCSWStatus) が成功を示している場合、ここに 0 以外の値が設定されている場合は、デバイスからの応答データが短かったということになります。

dCSWStatus

CBW 処理結果のステータス。値の内容は以下の通りです。

値	設定内容
00H	成功
01H	失敗
02H	フェーズエラー
03H~FFH	予約

5.1. MSC の位置づけ

サンプル・ドライバは以下の様な位置づけとなっています。

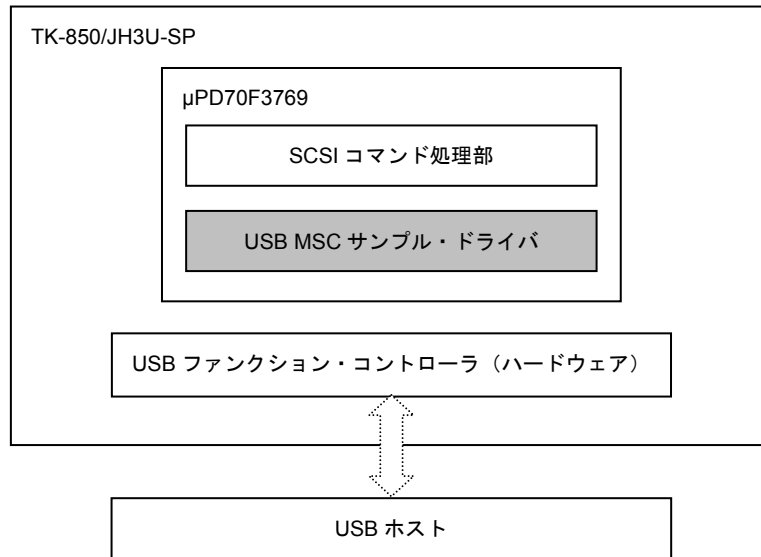


図 5-1 MSC の位置づけ

6. SCSI コマンド処理

サンプル・プログラムでは、MSC サンプル・ドライバの上に SCSI コマンド処理部があり、CBW にカプセル化された SCSI コマンドの処理を行います。

6.1. SCSI コマンド処理部の位置づけ

SCSI コマンド処理部は以下の様な位置づけになっています。

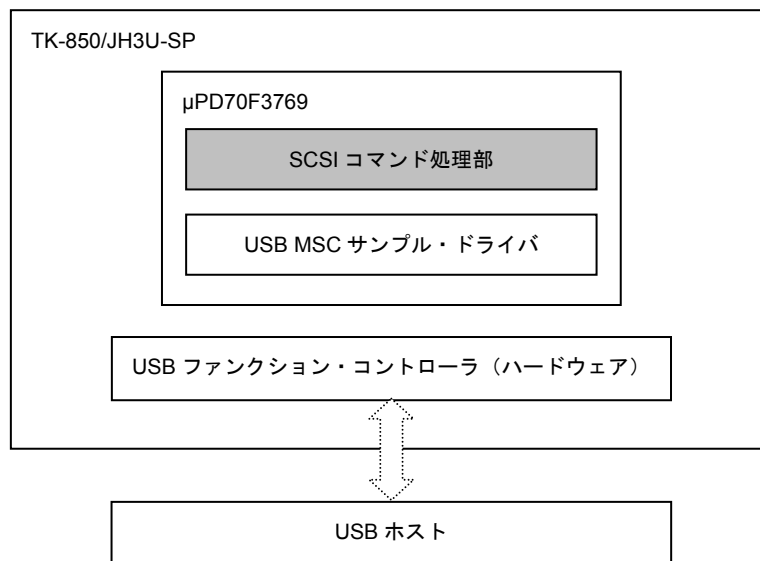


図 6-1 SCSI コマンド処理の位置づけ

6.2. 対応 SCSI コマンド

サンプル・ドライバでは、以下の SCSI コマンドに対応しています。

- ・TEST_UNIT_READY
- ・START_STOP_READY
- ・REQUEST_SENSE
- ・READ6
- ・MODE_SENSE6
- ・READ_FORMAT_CAPACITIES
- ・WRITE6
- ・MODE_SELECT
- ・VERIFY
- ・WRITE_BUFFER
- ・SEEK
- ・SYNCHRONIZE_CACHE
- ・INQUIRY
- ・READ10
- ・MODE_SENSE10
- ・READ_CAPACITY
- ・WRITE10
- ・MODE_SELECT10
- ・WRITE_VERIFY
- ・PREVENT

※ 上記コマンド以外を受信した場合は STALL 応答を行います。

7. 処理フロー

サンプル・ドライバで実装されている処理を以下に示します。

- ・ 初期化処理
- ・ エンドポイント監視処理
- ・ CBW データ処理

※サンプル・ドライバでは、エンドポイント監視を割り込みベクタではなくポーリングで各エンドポイントの割り込みフラグを監視することで行っています。

7.1. 初期化処理

初期化処理では以下の処理をおこなっています。

- ・ CPU 初期化
- ・ USB 初期化

7.1.1. CPU 初期化処理

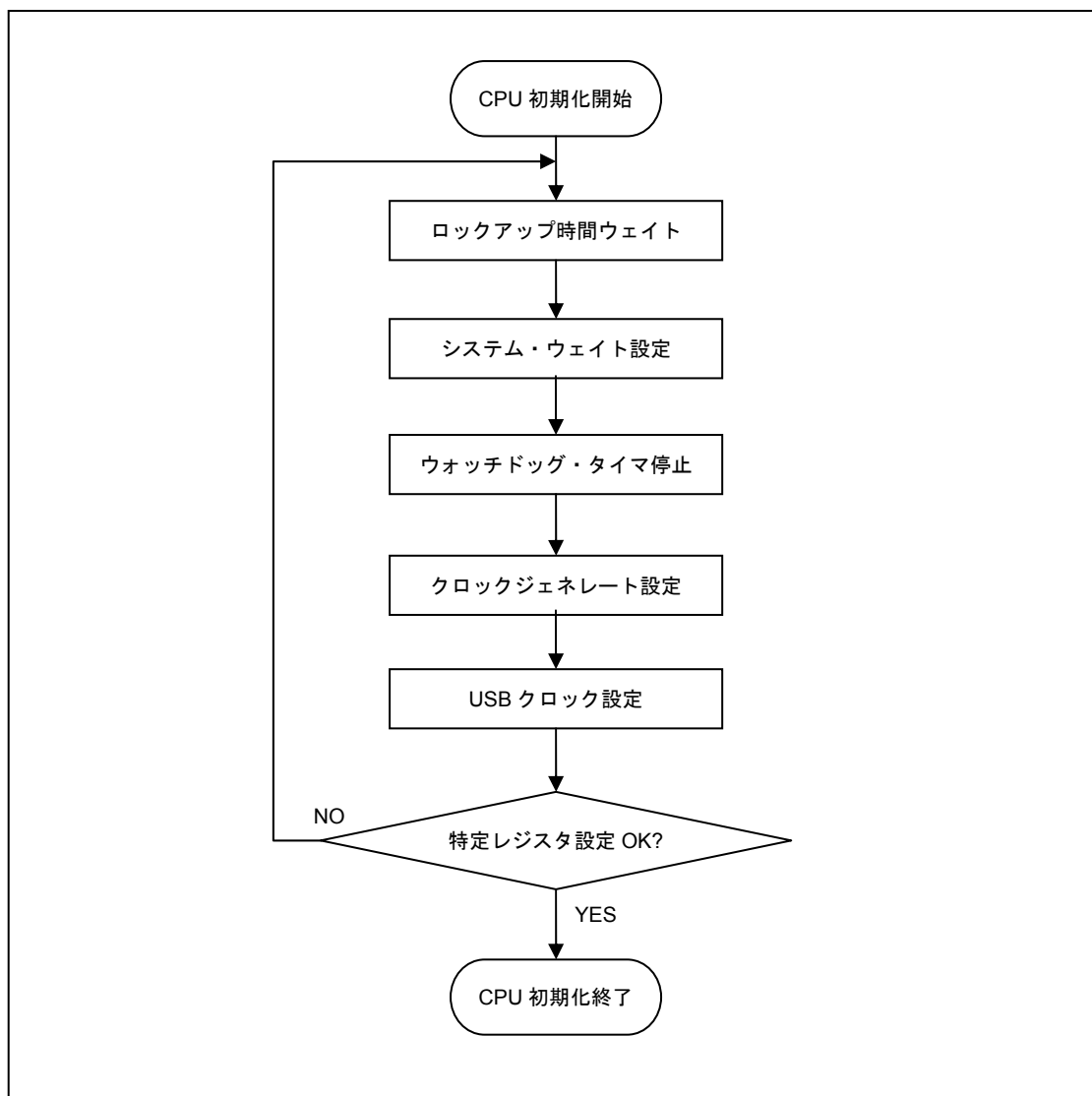


図 7-1 CPU 初期化フロー

CPU 初期化で実行する処理内容を次に示します。

- ・ ロックアップ時間ウェイト
LOCKR レジスタを監視し、周波数が安定する(ロック状態)までウェイトします。
- ・ システム・ウェイト設定
VSWC レジスタで内蔵周辺 I/O レジスタに対するバス・アクセスのウェイトを制御します。
- ・ ウォッチドッグ・タイマ停止
WDTM2 の WDM21・WDM20 ビットを”0”にすることでウォッチドッグ・タイマの動作モードを停止に設定します。
- ・ クロックジェネレート設定
CKC レジスタでメイン・クロックの逡倍率を指定します。
PLLCTL レジスタで PLL モードの ON/OFF 設定を指定します。
PCG レジスタで CPU クロックの分周率や、メイン/サブクロックでの動作を指定します。
- ・ USB クロック設定
UCKSEL レジスタ、UFCKMSK レジスタ、UHCKMSK レジスタで USB クロックを制御します。
- ・ 特定レジスタ設定時のエラー発生確認
CKC レジスタ、PCG レジスタは特定レジスタである為、操作後に SYS レジスタの PRERR ビットにてプロテクション・エラーが発生していないか確認する必要があります。

7.1.2. USB 初期化処理

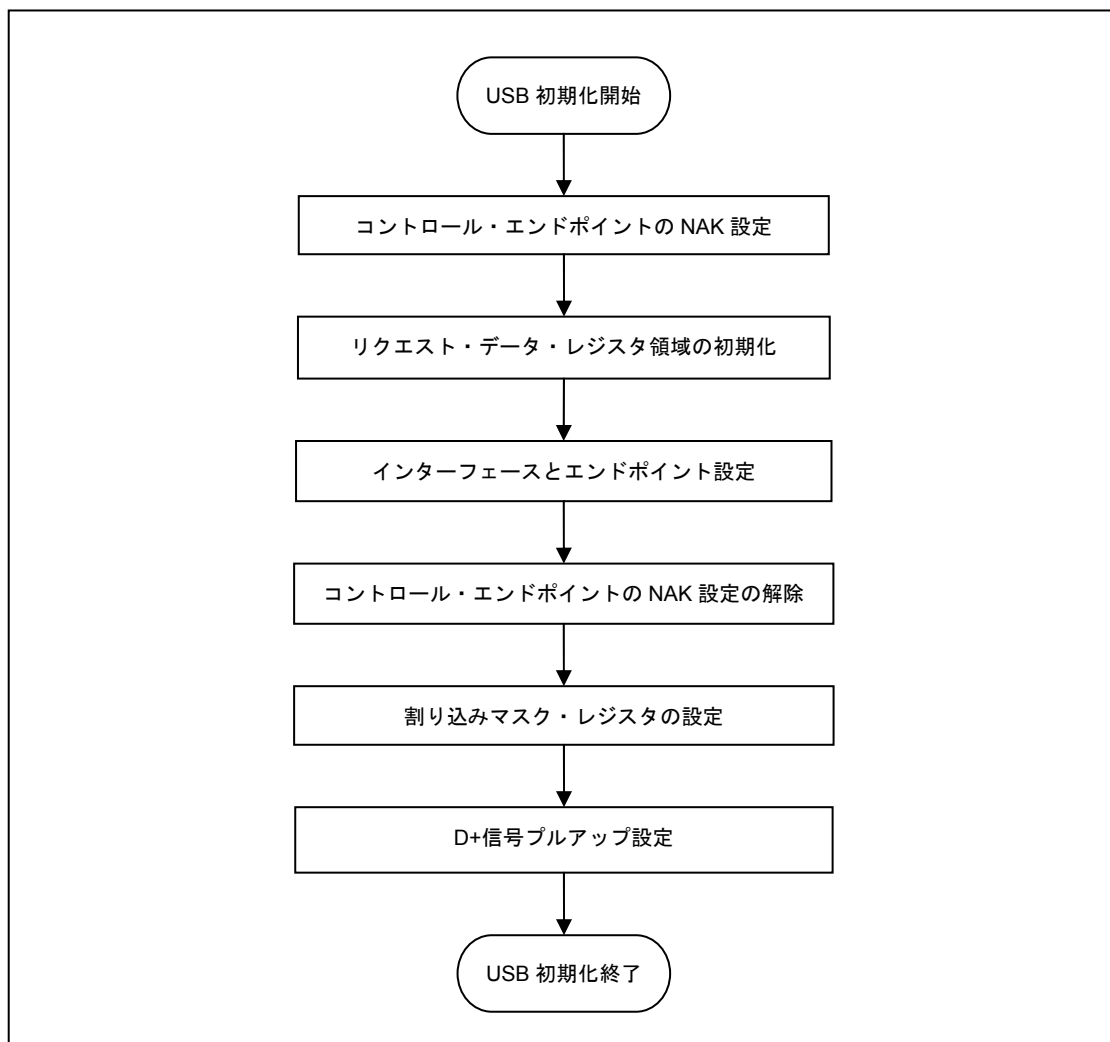


図 7-2 USB 初期化フロー

USB 初期化で実行すべき処理内容を次に示します。

- ・ コントロール・エンドポイントの NAK 設定
自動実行リクエストを含む全てのリクエストに NAK 応答します。
自動実行リクエストで使用するデータの登録が完了するまでハードウェアが自動実行リクエストに意図しないデータを返さないように設定します。
- ・ リクエスト・データ・レジスタ領域の初期化
Get Descriptor リクエストに응答するためのディスクリプタ・データなどをレジスタに登録します。
登録するデータは、デバイス・ステータス、エンドポイント 0 ステータス、Device Descriptor、Configuration Descriptor、Interface Descriptor、Endpoint Descriptor です。
- ・ インターフェースとエンドポイントの設定
サポートするインターフェースの数、オルタナティブ設定の状態、インターフェースとエンドポイントの関係などの情報をレジスタに設定します。
- ・ コントロール・エンドポイントの NAK 設定の解除
自動実行リクエスト用のデータ登録が終わったところで、コントロール・エンドポイントの NAK 設定を解除します。
- ・ 割り込みマスク・レジスタの設定
USB ファンクション・コントローラの割り込みステータス・レジスタに示される割り込み要因ごとのマスクを設定します。
- ・ D+信号プルアップ設定
D+信号をプルアップし、ホスト側にデバイスが接続されたことを認識させます。

7.2. エンドポイント監視処理

エンドポイント監視処理では、エンドポイント 0(コントロール転送用エンドポイント)、エンドポイント 1(バルクアウト用エンドポイント)の FIFO にデータあるかどうかを監視しています。

7.2.1. エンドポイント 0 監視処理

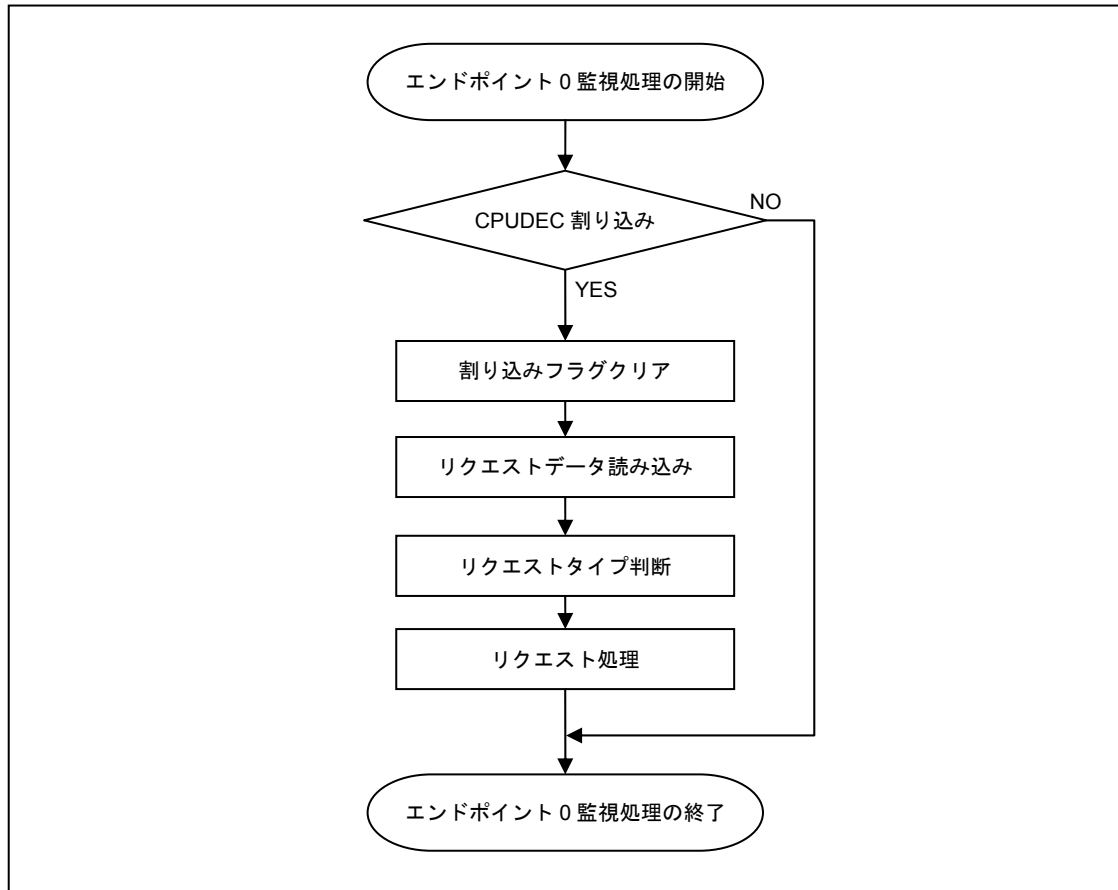


図 7-3 エンドポイント 0 監視処理フロー

エンドポイント 0 監視処理で実行すべき処理内容を次に示します。

エンドポイント 0 はコントロール転送用のエンドポイントですが、プラグイン時のエニュメレーションで使用するような標準デバイスリクエストの一部はハードウェアで自動応答する為、ここで監視する必要があるのはハードウェアで自動応答しない標準リクエスト、クラスリクエスト、ベンダリクエストです。

- ・ CPUDEC 割り込み判断
UF0IS1 の CPUDEC ビットが ON(1)していることで、割り込みが発生していると判断します。
- ・ 割り込みフラグクリア
UF0IC1 の CPUDECC ビットを OFF(0)することで割り込み要因をクリアします。
- ・ リクエストデータ読み込み
受信データを FIFO から読み込み、リクエストデータを構成します。
- ・ リクエストタイプ判断
リクエストデータが、ハードウェアで自動実行しない標準リクエストかクラスリクエストかベンダリクエストかを判断します。
- ・ リクエスト処理
リクエストタイプによりそれぞれのリクエスト処理を実行します。

7.2.2. エンドポイント 1 監視処理

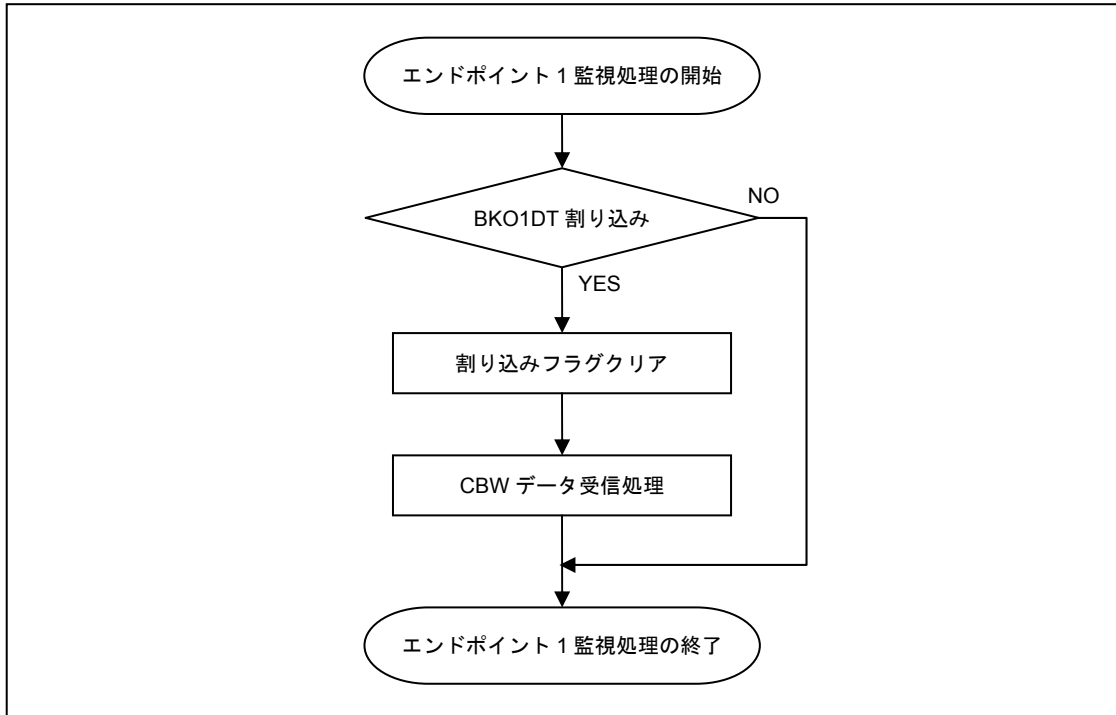


図 7-4 エンドポイント 1 監視処理フロー

エンドポイント 1 監視処理で実行すべき処理内容を次に示します。

- ・ BKO1DT 割り込み判断
UF0IS3 レジスタの BKO1DT ビットが ON(1)していることで正常受信完了割り込みが発生したと判断します。
- ・ 割り込みクリア
UF0IC3 レジスタの BKO1DTC ビットを OFF(0)することで割り込み要因をクリアします。
- ・ CBW データ受信処理
CBW データ受信処理を実行します。

7.3. SCSI コマンド処理

SCSI コマンド処理は、USB で CBW データを受信すると開始されます。

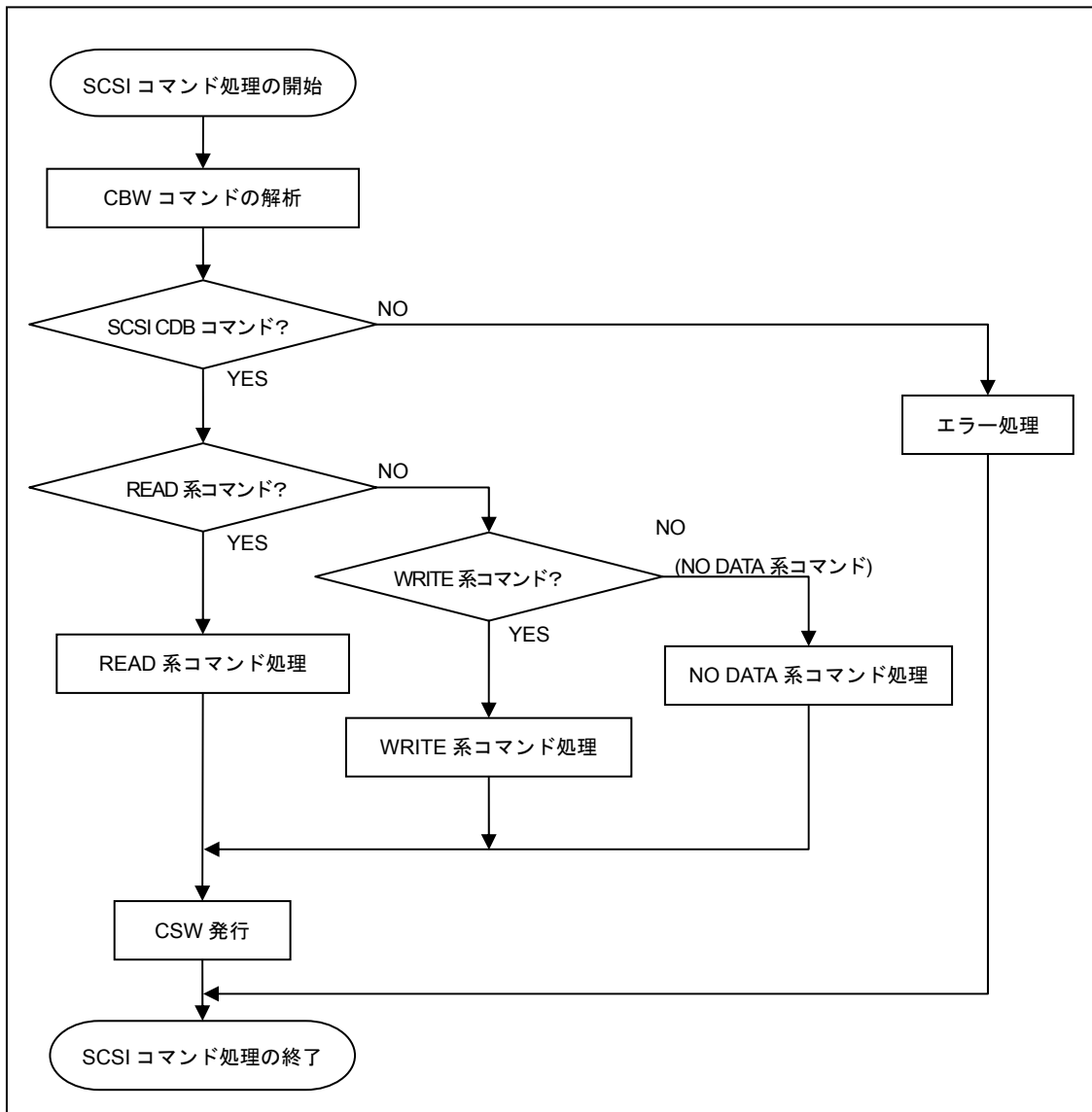


図 7-5 SCSI コマンド処理フロー

SCSI コマンド処理で実行する処理内容を次に示します。

- ・ CBW コマンドの解析
CBW データの受信後、CBW の内容について解析を行います。ここでは、CBW のタグを保存し、CBWCB の有効データ数、コマンドの方向について確認を行い、READ 系、WRITE 系、NO DATA 系のそれぞれの処理を実行します。
- ・ エラー処理
不正なコマンドなどエラー処理を行います。
- ・ READ 系コマンドの処理
SCSI コマンドの READ 系コマンド処理を行います。
- ・ WRITE 系コマンドの処理
SCSI コマンドの WRITE 系コマンド処理を行います。
- ・ NO DATA 系コマンドの処理
SCSI コマンドの NO DATA 系コマンド処理を行います。
- ・ CSW 発行
各コマンド処理の実行結果から CSW データを生成し送信処理を行います。

8. 関数仕様

サンプル・ドライバで使用する関数仕様を次に示します。

main.c

void main(void)	
関数概要	メイン処理
パラメータ	void
戻り値	void
処理概要	・初期化
	・エンドポイント 0 監視
	・エンドポイント 1 監視

void init(void)	
関数概要	初期化処理
パラメータ	void
戻り値	void
処理概要	・CPU 初期化
	・ROM 化パッケージ用初期化
	・USB 初期化

void cpu_init(void)	
関数概要	CPU 初期化処理
パラメータ	void
戻り値	void
処理概要	・ロックアップ時間ウェイト
	・システム・ウェイト設定
	・ウォッチドッグ・タイマ設定
	・PLL ON(システムクロックジェネレート)
	・クロック分周設定
	・USB クロック設定

void romp_init(void)	
関数概要	ROM 化パッケージ用初期化処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> ・コピー関数実行 _rcopy(&label, number)は、label の示すアドレス以降に存在する rompsec セクション内の情報をもとに、コピーしたいセクション番号 number の初期値データ、または RAM に配置するテキストを RAM 領域に 1 バイトずつコピーします。number に-1 を指定した場合、rompsec セクション内のすべてのセクションをコピーします。

usb850.c

void usb850_init(void)	
関数概要	USB ファンクション・コントローラ初期化処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> ・USB ファンクション・コントローラ関連レジスタの初期設定 詳細は別紙「レジスタ設定(MSC).xls」参照
	<ul style="list-style-type: none"> ・MSC クラスリクエスト関数登録
	<ul style="list-style-type: none"> ・D+信号プルアップ

void intusb0b(void)	
関数概要	エンドポイント 0 監視処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> ・CPUDEC 割り込みの監視 ・クラスリクエストの該当処理実行

void intusb1b(void)	
関数概要	エンドポイント 1 監視処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> ・BKO1DT 割り込みの監視 ・CBW データ受信処理

INT32 usbf850_data_send(UINT8* data, INT32 len, INT8 ep)			
関数概要	USB データ送信処理		
パラメータ	UINT8 *	data	:送信データバッファポインタ
	INT32	len	:データ長
	INT8	ep	:データ送信エンドポイント指定(エンドポイント番号)
戻り値	処理結果		
	DEV_OK	:正常終了	
	DEV_ERROR	:異常終了	
処理概要	・送信データバッファに格納されているデータを、1byte ずつ指定されたエンドポイント用のデータ送信 FIFO に格納します。		

INT32 usbf850_data_receive(UINT8* data, INT32 len, INT8 ep)			
関数概要	USB データ受信処理		
パラメータ	UINT8 *	data	:受信データ格納バッファポインタ
	INT32	len	:データ長
	INT8	ep	:データ受信エンドポイント指定(エンドポイント番号)
戻り値	処理結果		
	DEV_OK	:正常終了	
	DEV_ERROR	:異常終了	
処理概要	・指定されたエンドポイント用のデータ受信 FIFO から 1byte ずつデータを読み込み、受信データ格納バッファに格納します。		

void usbf850_sendnullEP0(void)	
関数概要	エンドポイント 0 用 NULL パケット送信処理
パラメータ	void
戻り値	void
処理概要	・FIFO クリア
	・エンドポイント 0 用データエンドビット ON

void usbf850_sendstallEP0(void)	
関数概要	エンドポイント 0 用 STALL 送信処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 用 STALL ビット ON

void usbf850_standardreq(void)	
関数概要	USB ファンクション・コントローラが自動応答しない標準リクエスト処理
パラメータ	void
戻り値	void
処理概要	・リクエストが GET_DESCRIPTOR の場合は GET_DESCRIPTOR 処理を実行
	・それ以外はエンドポイント 0 用 STALL 処理を実行

void usbf850_getdesc(void)	
関数概要	GET_DESCRIPTOR 処理
パラメータ	void
戻り値	void
処理概要	・StringDescriptor を要求している場合は StringDescriptor をエンドポイント 0 から送信します。
	・それ以外はエンドポイント 0 用 STALL 処理を実行

void usbf850_sstall_ctrl(void)	
関数概要	STALL 制御処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 用 STALL 処理を実行

usbf850_storage.c

void usbf850_blkonly_mass_storage_reset(void)	
関数概要	Mass Storage Reset 処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 1、エンドポイント 2 の FIFO クリア
	・エンドポイント 1、エンドポイント 2 を STALL 設定
	・エンドポイント 0 から NULL パケット送信

void usbf850_max_lun(void)	
関数概要	Get Max Lun 処理
パラメータ	void
戻り値	void
処理概要	・MAX LUN 送信

void usbf850_setfunction_storage(void)	
関数概要	クラスリクエスト関数登録処理
パラメータ	void
戻り値	void
処理概要	・各クラスリクエスト関数のアドレスを登録

void usbf850_rx_cbw(void)	
関数概要	CBW データの受信処理
パラメータ	void
戻り値	void
処理概要	・バルク IN の FIFO から CBW データ読み込み
	・CBW チェック処理実行

INT32 usbf850_storage_cbwchk(void)		
関数概要	CBW データのコマンド解析処理	
パラメータ	void	
戻り値	CBW チェック時のステータス	
	DEV_OK	: 正常終了
	DEV_ERROR	: 異常終了
処理概要	<ul style="list-style-type: none"> ・CBW データを解析し、 ・NO DATA 系コマンド ・DATA IN 系コマンド ・DATA OUT 系コマンド のいずれかを判断し、それぞれの処理を実行します。	

void usbf850_cbw_error(void)	
関数概要	CBW データのエラー処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 1、エンドポイント 2 を STALL させます。

void usbf850_no_data(void)	
関数概要	SCSI の NO DATA 系コマンド処理
パラメータ	void
戻り値	void
処理概要	・コマンド処理を実行し、処理結果を CSW にて送信します。

void usb850_data_in(void)	
関数概要	SCSI の DATA IN 系コマンド処理
パラメータ	void
戻り値	void
処理概要	・コマンド処理を実行し、処理結果を CSW にて送信します。

void usb850_data_out(void)	
関数概要	SCSI の DATA OUT 系コマンド処理
パラメータ	void
戻り値	void
処理概要	・コマンド処理を実行し、処理結果を CSW にて送信します。

INT32 usb850_csw_ret(UINT8 status)			
関数概要	CSW 応答処理		
パラメータ	UINT8	status	:コマンド処理結果
戻り値	CSW 送信処理結果		
	DEV_OK	:正常終了	
処理概要	・CSW 作成		
	・USB 送信処理		

scsi_cmd.c

INT32 scsi_command_to_ata (UINT8* ScsiCommandBuf, UINT8* pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	SCSI コマンド実行処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	SCSI コマンド処理結果		
	DEV_OK	:正常終了	
	DEV_ERR_NODATA	:NO DATA 系コマンドで転送方向エラー	
	DEV_ERR_READ	:READ 系コマンドで転送方向エラー	
	DEV_ERR_WRITE	:WRITE 系コマンドで転送方向エラー	
DEV_ERROR	:各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正		
処理概要	・SCSI コマンドを判断し、各コマンド処理を実行します。		
	・該当コマンドが無い場合は、ILLEGAL REQUEST として SENSE KEY を更新します。		

INT32 ata_test_unit_ready(INT32 TransFlag)			
関数概要	TEST UNIT READY コマンド処理		
パラメータ	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_NODATA	: NO DATA 系コマンドで転送方向エラー	
処理概要	・SENSE KEY 更新		

INT32 ata_seek(INT32 TransFlag)			
関数概要	SEEK コマンド処理		
パラメータ	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_NODATA	: NO DATA 系コマンドで転送方向エラー	
処理概要	・SENSE KEY の更新		

INT32 ata_start_stop_unit(INT32 TransFlag)			
関数概要	START STOP UNIT コマンド処理		
パラメータ	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_NODATA	: NO DATA 系コマンドで転送方向エラー	
処理概要	・SENSE KEY 更新		

INT32 ata_synchronize_cache(INT32 TransFlag)			
関数概要	SYNCHRONIZE CACHE コマンド処理		
パラメータ	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_NODATA	: NO DATA 系コマンドで転送方向エラー	
処理概要	・SENSE KEY 更新		

INT32 ata_request_sense (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	REQUEST SENSE コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK		: 正常終了
	DEV_ERR_NODATA		: NO DATA 系コマンドで転送方向エラー
	DEV_ERR_READ		: READ 系コマンドで転送方向エラー
処理概要	・SENSE KEY データ送信		

INT32 ata_inquiry (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	INQUIRY コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK		: 正常終了
	DEV_ERR_READ		: READ 系コマンドで転送方向エラー
	DEV_ERROR		: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正
処理概要	・SENSE KEY データ更新		
	・INQUIRY データ送信		

INT32 ata_mode_select (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	MODE SELECT(6)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・MODE SELECT データを受信し、MODE SELECT データテーブル更新		

INT32 ata_mode_select10 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	MODE SELECT(10)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・MODE SELECT(10)データを受信し、MODE SELECT(10)データテーブル更新		

INT32 ata_mode_sense (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	MODE SENSE(6)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・MODE SENSE データ送信		

INT32 ata_mode_sense10 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	MODE SENSE(10)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・MODE SENSE(10)データ送信		

INT32 ata_read_format_capacities (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	READ FORMAT CAPACITIES コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・FORMAT CPACITY データ送信		

INT32 ata_read_capacity (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	READ CAPACITY コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・CPACITY データ送信		

INT32 ata_read6 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 lDataSize, INT32 TransFlag)			
関数概要	READ(6)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	lDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから読み込み開始アドレス算出		
	・データ領域からデータを読み込み、USB 送信		

INT32 ata_read10 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 lDataSize, INT32 TransFlag)			
関数概要	READ(10)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	lDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから読み込み開始アドレス算出		
	・データ領域からデータを読み込み、USB 送信		

INT32 ata_write6 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 lDataSize, INT32 TransFlag)			
関数概要	WRITE(6)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	lDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから書き込み開始アドレス算出		
	・書き込みデータを読み込み、データ領域に書き込み		

INT32 ata_write10 (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 lDataSize, INT32 TransFlag)			
関数概要	WRITE(10)コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	lDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから書き込み開始アドレス算出		
	・書き込みデータを読み込み、データ領域に書き込み		

INT32 ata_verify (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	VERIFY コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_NODATA	: NO DATA 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから書き込み開始アドレス算出		
	・書き込みデータを読み込み、データ領域に書き込み		

INT32 ata_write_verify (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	WRITE VERIFY コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・LBA とブロックサイズから書き込み開始アドレス算出		
	・書き込みデータを読み込み、データ領域に書き込み		

INT32 ata_write_buff (UINT8 *ScsiCommandBuf, UINT8 *pbData, INT32 IDataSize, INT32 TransFlag)			
関数概要	WRITE BUFF コマンド処理		
パラメータ	UINT8*	ScsiCommandBuf	SCSI コマンド格納バッファポインタ
	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	IDataSize	データサイズ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_WRITE	: WRITE 系コマンドで転送方向エラー	
	DEV_ERROR	: 各コマンドの実行結果で上記 4 つのステータス以外、または、リクエストが不正	
処理概要	・SENSE KEY データ更新		
	・データ読み捨て		

INT32 scsi_to_usb(UINT8 *pbData, INT32 TransFlag)			
関数概要	(SCSI コマンド系)USB データ送信処理		
パラメータ	UINT8*	pbData	コマンド用データ格納バッファポインタ
	INT32	TransFlag	データ転送方向
戻り値	処理結果		
	DEV_OK	: 正常終了	
	DEV_ERR_READ	: READ 系コマンドで転送方向エラー	
処理概要	・SENSE KEY データ更新		
	・USB データ送信処理		

9. レジスタ設定

9.1. CPU 関連レジスタの設定

アドレス	レジスタ名称	レジスタ略称	設定値	備考
0xFFFF F820	ロック・レジスタ	LOCKR	-	LOCK(0):PLL のロック状態 0:ロック状態 1:アンロック状態
0xFFFF F06E	システム・ウェイト・コントロール レジスタ	VSWC	0x12	内蔵周辺 I/O レジスタに対するバス・アクセスのウェイト制御 12H:ウェイト数 3
0xFFFF F9FC	オンチップ・デバッグ・モード設 定レジスタ	OCDM	0x00	特定レジスタ OCDM0(0):動作モード _DRST 端子がロウ・レベルの場合: 通常動作モード(オンチップ・デバッグ・モード用端子として使用) _DRST 端子がハイ・レベルの場合: オンチップ・デバッグ・モード(オンチップ・デバッグ・モード用端子として使用)
0xFFFF F6D0	ウォッチドッグ・タイマ・モードレ ジスタ 2	WDTM2	0x00	00H:タイマ停止
0xFFFF F822	クロック・コントロール・レジスタ	CKC	0x0B	特定レジスタ CKDIV0(0):PLL モード時の内部システムクロック 1:8 通倍 ※CKC レジスタには必ず 0BH を設定すること。
0xFFFF F82C	PLL コントロール・レジスタ	PLLCTL	0x03	SELPLL(1):CPU 動作クロック選択レジスタ 1:PLL モード PLLON(0):PLL 動作停止レジスタ 1:PLL 動作(PLL 動作開始後、周波数が安定するまでの所定のロックアップ時間が必要)
0xFFFF F828	プロセッサ・クロック・コントロー	PCC	0x00	特定レジスタ

	ルレジスタ				
					FRC(7):サブクロック発振回路の内蔵帰還抵抗の制御 使用する(0)
					MCK(6):メインクロック発振回路の制御 発振する(0)
					MFRC(5):メイン・クロック発振回路の内蔵帰還抵抗の制御 使用する(0)
					CK3-0(3-0):内部システムクロック(fclk)/CPUクロック(fcpu)の選択 fxx(0000)
0xFFFF	F484	データ・ウェイト・コントロール・レジスタ 0	DWC0	0x7777	DW32-30(14-12):CS3の挿入ウェイト・ステート数 1H:7
					DW22-20(10-8):CS2の挿入ウェイト・ステート数 1H:7
					DW12-10(6-4):CS1の挿入ウェイト・ステート数 7H:7 ※CS1は内部で使用する為、値固定
					DW02-00(3-0):CS0の挿入ウェイト・ステート数 1H:7
0xFFFF	FE40	USBクロック選択レジスタ	UCKSEL	0x02	UHUCLK:UCLK、UHPCLK:UHPCLK、UFUCLK:UCLK、UHPCLK:UHUCLK
0xFFFF	FE41	UFUCLKマスクレジスタ	UFCKMSK	0x00	USB Host バッファ&許可:USBH使用、USB Function バッファ&許可:USBF使用
0xFFFF	FE42	UHPCLK/UFHUCLKマスクレジスタ	UHCKMSK	0x00	

9.2. USB 関連レジスタの設定

アドレス	マクロ名称	内容	初期化処理後	
			値	設定
0x00200002	UF0E0NA	UF0 EP0NAKALL レジスタ	0x01	[ビット 0:EP0NKA] Endpoint0 への SETUP トランザクション以外の NAK を送信する(1)
0x00200144	UF0DSTL	UF0 デバイス・ステータス・レジスタ L	0x00	[ビット 1:RMWK] デバイスによるリモート・ウェイクアップ機能の使用禁止(0)
				[ビット 0:SFPW] デバイスがバス・パワード(0)
0x0020014C	UF0E0SL	UF0 EP0 ステータス・レジスタ L	0x00	[ビット 0:E0HALT] Endpoint0 のストール状態クリア(0)
0x00200150	UF0E1SL	UF0 EP1 ステータス・レジスタ L	0x00	[ビット 0:E1HALT] Endpoint1 のストール状態クリア(0)
0x00200154	UF0E2SL	UF0 EP2 ステータス・レジスタ L	0x00	[ビット 0:E2HALT] Endpoint2 のストール状態クリア(0)
0x002001A0	UF0DSCL	UF0 ディスクリプタ・レングス・レジスタ	0x20	[ビット 7-0:DPL7-0] GET_DESCRIPTOR Configuration リクエストで返信する全ディスクリプタのバイト数-1 の値
0x002001A2	UF0DD0	UF0 デバイス・ディスクリプタ・レジスタ	0x12	bLength
0x002001A4	UF0DD1		0x01	bDescriptorType
0x002001A6	UF0DD2		0x00	bcdUSB Lo
0x002001A8	UF0DD3		0x02	bcdUSB Hi
0x002001AA	UF0DD4		0x00	bDeviceClass
0x002001AC	UF0DD5		0x00	bDeviceSubClass
0x002001AE	UF0DD6		0x00	bDeviceProtocol
0x002001B0	UF0DD7		0x40	bMaxPacketSize0
0x002001B2	UF0DD8		0x09	idVendor Lo
0x002001B4	UF0DD9		0x04	idVendor Hi
0x002001B6	UF0DD10		0xD2	idProduct Lo
0x002001B8	UF0DD11		0x01	idProduct Hi
0x002001BA	UF0DD12		0x01	bcdDevice Lo
0x002001BC	UF0DD13		0x00	bcdDevice Hi
0x002001BE	UF0DD14	0x01	iManufacture	

0x002001C0	UF0DD15		0x00	iProduct
0x002001C2	UF0DD16		0x00	iSerialNumber
0x002001C4	UF0DD17		0x01	bNumConfigurations
0x002001C6	UF0CIE0	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ (コンフィギュレーションディスクリプタ部)	0x09	bLength
0x002001C8	UF0CIE1		0x02	bDescriptorType
0x002001CA	UF0CIE2		0x20	wTotalLength Lo
0x002001CC	UF0CIE3		0x00	wTotalLength Hi
0x002001CE	UF0CIE4		0x01	bNumInterface
0x002001D0	UF0CIE5		0x01	bConfigurationValue
0x002001D2	UF0CIE6		0x00	iConfiguration
0x002001D4	UF0CIE7		0x80	bmAttributes
0x002001D6	UF0CIE8		0x1B	MaxPower
0x002001D8	UF0CIE9			0x09
0x002001DA	UF0CIE10	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ (インターフェース 0 部)	0x04	bDescriptorType
0x002001DC	UF0CIE11		0x00	bInterfacenumber
0x002001DE	UF0CIE12		0x00	bAlternateSetting
0x002001E0	UF0CIE13		0x02	bNumEndpoints
0x002001E2	UF0CIE14		0x08	bInterfaceClass
0x002001E4	UF0CIE15		0x06	bInterfaceSubClass
0x002001E6	UF0CIE16		0x50	bInterfaceProtocol
0x002001E8	UF0CIE17		0x00	iInterface
0x002001EA	UF0CIE18			0x07
0x002001EC	UF0CIE19	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ (エンドポイント1部)	0x05	bDescriptorType
0x002001EE	UF0CIE20		0x81	bEndpointAddress
0x002001F0	UF0CIE21		0x02	bmAttributes
0x002001F2	UF0CIE22		0x40	wMaxPacketSize Lo
0x002001F4	UF0CIE23		0x00	wMaxPacketSize Hi
0x002001F6	UF0CIE24		0x00	bInterval

0x002001F8	UF0CIE25	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ (エンドポイント 2 部)	0x07	bLength
0x002001FA	UF0CIE26		0x05	bDescriptorType
0x002001FC	UF0CIE27		0x02	bEndpointAddress
0x002001FE	UF0CIE28		0x02	bmAttributes
0x00200200	UF0CIE29		0x40	wMaxPacketSize Lo
0x00200202	UF0CIE30		0x00	wMaxPacketSize Hi
0x00200204	UF0CIE31		0x00	bInterval
0x00200074	UF0MODC	UF0 モード・コントロール・レジスタ	0x00	[ビット 6: CDCG DST] GET_DESCRIPTOR Configuration リクエストは自動のまま(0)
0x00200080	UF0AIFN	UF0 アクティブ・インターフェース・ナンバー・レジスタ	0x00	[ビット 7: ADDIF] Interface 0 だけをサポート(0)
				[ビット 1-0: IFNO1、0] サポートする Interface 番号
0x00200084	UF0AAS	UF0 アクティブ・オルタナティブ・セッティング・レジスタ	0x00	[ビット 3-1: AL5ST3,2,1] Alternative Setting0(0)
0x00200086	UF0E1IM	UF0 エンドポイント1 インターフェース・マッピング・レジスタ	0x20	[ビット 7-5: E1EN2,1,0] Interface 0, Alternative Setting0 とリンク(001B)
				[ビット 4: E12AL1] CONF ビット=1 で Alternate Setting1 に設定されても有効にならない(0)
				[ビット 3-0: E15AL4-1] CONF ビット=1 で Alternate Setting n に設定されても有効にならない(0)
0x00200088	UF0E2IM	UF0 エンドポイント2 インターフェース・マッピング・レジスタ	0x20	[ビット 7-5: E2EN2,1,0] Interface 0, Alternative Setting(001B)
				[ビット 4: E22AL1] CONF ビット=1 で Alternate Setting1 に設定されても有効にならない(0)
				[ビット 3-0: E25AL4-1] CONF ビット=1 で Alternate Setting n に設定されても有効にならない(0)
0x0020003C	UF0IC0	UF0 割り込みクリアレジスタ 0	0x20	[ビット 7: BUSRSTC] Bus Reset 割り込み要求クリア(0)
				[ビット 6: RSUSPDC] Resume/Suspend 割り込み要求クリア(0)
				[ビット 4: SHORTC] Short 割り込み要求をクリア(0)
				[ビット 3: DMAEDC] DMA_END 割り込み要求をクリア(0)
				[ビット 2: SETRQC] SET_RQ 割り込み要求をクリア(0)

				[ビット 1:CLRRQC] CLR_RQ 割り込み要求をクリア(0)
				[ビット 0:EPHALTC] EP_Halt 割り込み要求をクリア(0)
0x0020003E	UF0IC1	UF0 割り込みクリアレジスタ 1	0x80	[ビット 6:E0INC] EPOIN 割り込み要求をクリア(0)
				[ビット 5:E0INDTC] EPOINDT 割り込み要求をクリア(0)
				[ビット 4:E0ODTC] EPOOUTDT 割り込み要求をクリア(0)
				[ビット 3:SUCESC] Success 割り込み要求をクリア(0)
				[ビット 2:STGC] Stg 割り込み要求をクリア(0)
				[ビット 1:PROTC] Protect 割り込み要求をクリア(0)
				[ビット 0:CPUDEC] CPUDEC 割り込み要求をクリア(0)
0x00200040	UF0IC2	UF0 割り込みクリアレジスタ 2	0x0C	[ビット 7:BKI2INC] BLK21IN 割り込み要求をクリア(0)
				[ビット 6:BKI2DTC] BLK21DT 割り込み要求をクリア(0)
				[ビット 5:BKI1INC] BLK11IN 割り込み要求をクリア(0)
				[ビット 4:BKI1DTC] BLK11DT 割り込み要求をクリア(0)
				[ビット 1:IT2DTC] INT2DT 割り込み要求をクリア(0)
				[ビット 0:IT1DTC] INT1DT 割り込み要求をクリア(0)
0x00200042	UF0IC3	UF0 割り込みクリアレジスタ 3	0x00	[ビット 7:BKO2FLC] BLK21FL 割り込み要求をクリア(0)
				[ビット 6:BKO2NLC] BLK21NL 割り込み要求をクリア(0)
				[ビット 5:BKO2NAKC] BLK21NK 割り込み要求をクリア(0)
				[ビット 0:BKO2DTC] BLK21DT 割り込み要求をクリア(0)
				[ビット 3:BKO1FLC] BLK11FL 割り込み要求をクリア(0)
				[ビット 2:BKO1NLC] BLK11NL 割り込み要求をクリア(0)
				[ビット 1:BKO1NAKC] BLK11NK 割り込み要求をクリア(0)
				[ビット 0:BKO1DTC] BLK11DT 割り込み要求をクリア(0)
0x00200044	UF0IC4	UF0 割り込みクリアレジスタ 4	0xDF	[ビット 5:SETINTC] SET_INT 割り込み要求をクリア
0x00200060	UF0FIC0	UF0 FIFO クリアレジスタ 0	0xFF	[ビット 7:BKI2SC] UF0B12 レジスタの SIE 側 FIFO のみクリア(1)
				[ビット 6:BKI2CC] UF0B12 レジスタの CPU 側 FIFO のみクリア(1)
				[ビット 5:BKI1SC] UF0B11 レジスタの SIE 側 FIFO のみクリア(1)
				[ビット 4:BKI1CC] UF0B11 レジスタの CPU 側 FIFO のみクリア(1)

				[ビット 3:ITR2C] UF0INT2 レジスタクリア(1)
				[ビット 2:ITR1C] UF0INT1 レジスタクリア(1)
				[ビット 1:EP0WC] UF0E0W レジスタクリア(1)
				[ビット 0:EP0RC] UF0E0R レジスタクリア(1)
0x00200062	UF0FIC1	UF0 FIFO クリアレジスタ 1	0x0F	[ビット 3:BKO2C] UF0BO2 レジスタの SIE 側/CPU 側の両 FIFO をクリア(1)
				[ビット 2:BKO2CC] UF0BO2 レジスタの CPU 側だけの FIFO をクリア(1)
				[ビット 1:BKO1C] UF0BO1 レジスタの SIE 側/CPU 側の両 FIFO をクリア(1)
				[ビット 0:BKO1CC] UF0BO1 レジスタの CPU 側だけの FIFO をクリア(1)
0x0020002E	UF0IM0	UF0 割り込みマスクレジスタ 0	0x9F	[ビット 7:BUSRSTM] Bus Reset 割り込みをマスク(1)
				[ビット 6:RSUSPDM] Resume/Suspend 割り込みをマスクしない(0)
				[ビット 4:SHORTM] Short 割り込みをマスク(1)
				[ビット 3:DMAEDM] DMA_END 割り込みをマスク(1)
				[ビット 2:SETRQM] SET_RQ 割り込みをマスク(1)
				[ビット 1:CLRRQM] CLR_RQ 割り込みをマスク(1)
				[ビット 0:EPHALTM] EP_Halt 割り込みをマスク(1)
0x00200030	UF0IM1	UF0 割り込みマスクレジスタ 1	0x7E	[ビット 6:E0INM] EP0IN 割り込みをマスク(1)
				[ビット 5:E0INDTM] EPINDT 割り込みをマスク(1)
				[ビット 4:E0ODTM] EP0OUTDT 割り込みをマスク(1)
				[ビット 3:SUCESM] Success 割り込みをマスク(1)
				[ビット 2:STGM] Stg 割り込みをマスク(1)
				[ビット 1:PROTM] Protect 割り込みをマスク(1)
				[ビット 0:CPUDECM] CPUDEC 割り込みをマスクしない(0)
0x00200032	UF0IM2	UF0 割り込みマスクレジスタ 2	0xF3	[ビット 7:BK21INM] BLK12IN 割り込みをマスク(1)
				[ビット 6:BK12DTM] BLK2DT 割り込みをマスク(1)
				[ビット 5:BK11INM] BLK11IN 割り込みをマスク(1)
				[ビット 4:BK11DTM] BLK1DT 割り込みをマスク(1)
				[ビット 1:IT2DTM] INT2DT 割り込みをマスク(1)
				[ビット 0:IT1DTM] INT1DT 割り込みをマスク(1)

0x00200034	UF0IM3	UF0 割り込みマスクレジスタ 3	0xFE	[ビット 7: BKO2FLM] BLKO2FL 割り込みをマスク(1)
				[ビット 6: BKO2NLM] BLKO2NL 割り込みをマスク(1)
				[ビット 5: BLKO2NAKM] BLKO2NK 割り込みをマスク(1)
				[ビット 4: BKO2DTM] BLKO2DT 割り込みをマスク(1)
				[ビット 3: BKO1FLM] BLKO1FL 割り込みをマスク(1)
				[ビット 2: BKO1NLM] BLKO1NL 割り込みをマスク(1)
				[ビット 1: BLKO1NAKM] BLKO1NK 割り込みをマスク(1)
				[ビット 0: BKO1DTM] BLKO1DT 割り込みをマスクしない(0)
0x00200036	UF0IM4	UF0 割り込みマスクレジスタ 4	0x20	[ビット 5: SETINTM] SET_INT 割り込みをマスクする(1)
0xFFFFF428	PM4	ポート 4 モード・レジスタ	0xFC	[ビット 2: PM42] 入力モード(1)
				[ビット 1: PM41] 出力モード(0)
				[ビット 0: PM40] 出力モード(0)
0xFFFFF408	P4	ポート 4・レジスタ	0x02	[ビット 2: P42] (出力モード時)0 出力(0)
				[ビット 1: P41] (出力モード時)1 出力(1)
				[ビット 0: P40] (出力モード時)0 出力(0)

9.3. ディスクリプタ設定

デバイス・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x12
bDescriptor	1	ディスクリプタタイプ	0x01
bcdUSB	2	BCD 表現の USB 仕様リリース番号	0x0200
bDeviceClass	1	クラス・コード 0x00: クラス無し、0x01~0xFE: 特定、0xFF: ベンダ	0x00
bDeviceSubClass	1	サブ・クラスコード	0x00
bDeviceProtocol	1	プロトコル・コード 0x00: 固有プロトコル使用せず、0xFF: ベンダ固有	0x00
bMaxPacketSize0	1	エンドポイント 0 の最大パケットサイズ	0x40
idVendor	2	ベンダ ID(USB IF が割り当て)	0x0409
idProduct	2	プロダクト ID(ベンダが割り当てる)	0x01D2
bcdDevice	2	BCD 表現のデバイスのリリース番号	0x0001
iManufacture	1	製造者を表すstring・ディスクリプタへのインデックス	0x01
iProduct	1	製品を表すstring・ディスクリプタへのインデックス	0x00
iSerialNumber	1	デバイスの製造番号を表すstring・ディスクリプタへのインデックス	0x00
bNumConfigurations	1	構成可能な数	0x01

コンフィグレーション・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x09
bDescriptor	1	ディスクリプタタイプ	0x02
wTotalLength	2	構成全体(構成、インターフェース、エンドポイント、そのほかのディスクリプタ)の長さ	0x0020
bNumInterfaces	1	構成のもつインターフェースの数	0x01
bConfigurationValue	1	SetConfiguration リクエストで、この構成を選択するための引数値(1 以上)	0x01
iConfiguration	1	構成を表すストリング・ディスクリプタへのインデックス	0x00
bmAttributes	1	構成の特性、ビット単位で意味づけ D7: "1" D6: 自己電源 D5: リモート・ウェークアップ D4-D0: 予約(0)	0x80
bMaxPower	1	最大バス電力消費量を 2mA 単位で指定	0x1B

インターフェース・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x09
bDescriptor	1	ディスクリプタタイプ	0x04
bInterfaceNuber	1	構成の中で、このインターフェースを表すインデックス番号(0 ベース)	0x00
bAlternateSetting	1	SetInterface リクエストで、代替設定を選択するための引数値	0x00
bNumEndpoints	1	(エンドポイント 0 を除く)インターフェースの持つエンドポイント数	0x02
bInterfaceClass	1	クラス・コード 0x00: クラスなし、0x01~0xFE: 特定、0xFF: ベンダ	0x08
bInterfaceSubClass	1	サブクラス・コード	0x06
bInterfaceProtocol	1	プロトコル・コード 0x00: 固有プロトコル使用せず、0xFF ベンダ固有	0x50
iInterface	1	このインターフェースを表すストリング・ディスクリプタへのインデックス	0x00

エンドポイント・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x07
bDescriptor	1	ディスクリプタタイプ	0x05
bEndpointAddress	1	エンドポイント・アドレス。ビット単位で意味づけ D7: 方向 0: OUT 1: IN D6-D4: 予約(0) D4-D0: エンドポイント番号	0x81
bmAttributes	1	属性(ビット単位で意味づけ) D1-D0: 転送タイプ 0: コントロール 1: アイソクロナス 2: バルク 3: インタラプト ※D5~D2 はアイソクロナス・エンドポイントのみで使用 D3-D2: 同期タイプ 0: 同期なし 1: 非同期 2: アダプティブ 3: 同期 D5-D4: ユーセージ・タイプ 0: データ・エンドポイント 1: フィードバック・エンドポイント 2: 従属的なフィードバック・エンドポイント 3: (予約)	0x02

wMaxPacketSize	2	<p>ペイロード・サイズ指定(ビットで意味づけ)</p> <p>D10-D0: 最大パケット・サイズ</p> <p>D12-D11: μ フレームあたりの追加的なトランザクション数(ハイ・スピードのアイソクロナスとインタラプトのみ)</p> <p>0: 追加なし(1トランザクション/μ フレーム)</p> <p>1: 1 つ(2トランザクション/μ フレーム)</p> <p>2: 2 つ(3トランザクション/μ フレーム)</p> <p>3: 未使用(予約)</p>	0x0040
bInterval	1	<p>データ転送のエンドポイントをポーリング間隔</p> <p>フル/ロー・スピード・インタラプト: ms 単位(フレーム数)で指定</p> <p>ハイ・スピード・アイソクロナス/インタラプト: μ フレーム単位で 2 の(N-1)乗の N を指定 (例えば、bInterval が 4 の場合、8 μ フレームに 1 回ポーリング)</p> <p>フル・スピード・アイソクロナス: 1ms 単位(フレーム数)で 2 の(N-1)乗の N を指定</p> <p>ハイ・スピード・バルク/コントロール: エンドポイントの最大 NAK レートを μ フレーム単位で指定</p> <p>値 0 は OUT/DATA トランザクションで NAK 応答しないことを意味</p>	0x00

ストリングディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x07
bDescriptor	1	ディスクリプタタイプ	0x05
bString	-	Language Code: 0x09,0x04	
		Manufacture: "NEC Electronics Co."	

10. マスストレージクラスサンプル・プログラムの構成

サンプル・プログラムのディレクトリ構成を次に示します。

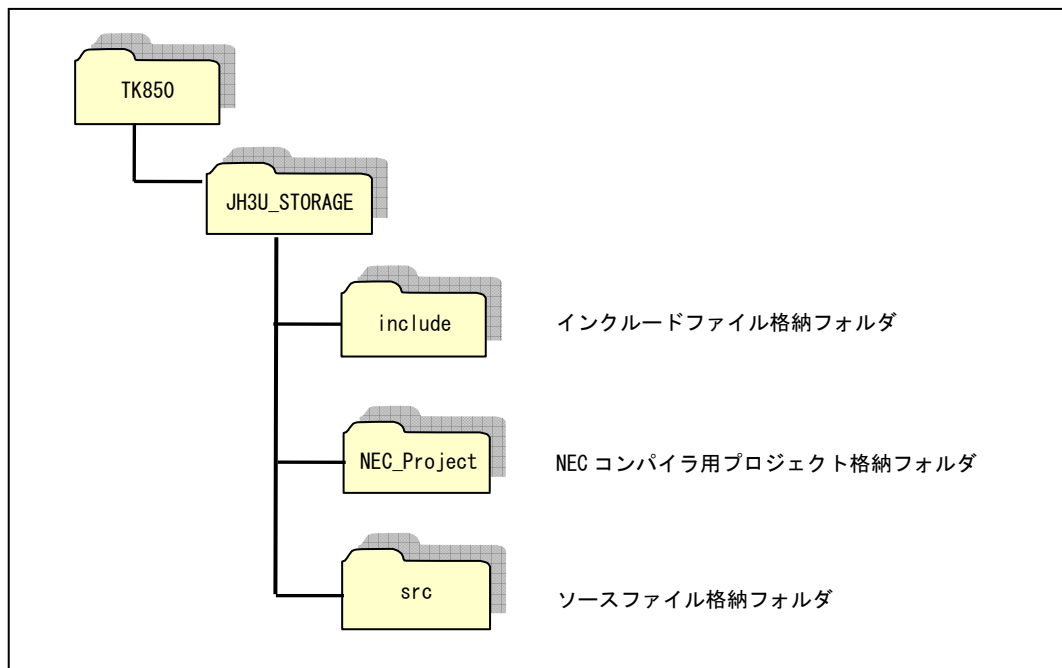


図 10-1 ディレクトリ構成例

10.1. ドライバのインストール

デバッグポート(USB1)、 μ PD70F3769 ファンクション USB ポート(USB3)はホスト PC 接続時にそれぞれ別のドライバが必要です。ドライバをインストールするために必要なファイルを次に示します。ドライバインストール時に該当するファイルを指定してください。

[デバッグポート] (USB1 コネクタ)

MQB2SALL.inf 開発ツールのインストール時に”Starter Kit USB Driver”をインストールすることで、デバッグポート用ドライバが「C:¥Program Files¥NECElectronicsTools¥TK-driver」にインストールされます。接続時には「NECElectronics Starter Kit Virtual UART」として認識されます。

[μ PD70F3769 ファンクション USB ポート] (USB3 コネクタ)

マストレージクラスサンプル・プログラムのドライバは Windows 標準のドライバを使用します。サンプルプログラムが RUN の状態でホスト PC に接続すると、自動的にドライバがインストールされて、「ディスクドライブ」に「NEC corp TK-850JG3-H/JH3- USB Device」として認識されます。

以下はハードウェアの環境構築例です。

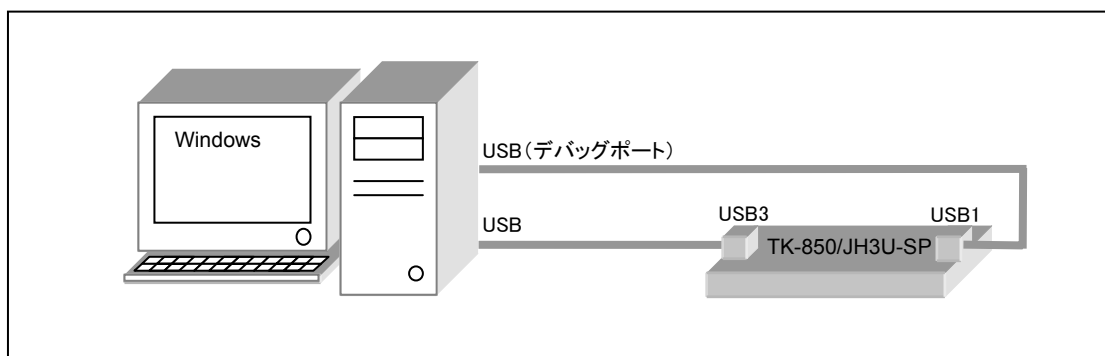


図 10-2 開発環境構築例(ハードウェア)

改版履歴

版数	日付	内容
1.0	2008 / 10 /10	初版