

## V850E2/ML4

R01AN1217JJ0102

Rev.1.02

## USB ホストソフトウェア

2013.01.10

### 要旨

本マニュアルでは、USB ホスト・コントローラ (USBH) を使用した、USB デバイス制御例を説明します。

### 動作確認デバイス

- V850E2/ML4 グループ(型名 uPD70F4022)

### ターゲットボード

- V850E2/ML4 CPU ボード(型番 : R0K0F4022C000BR)

### 注意事項

- 制御可能な USB デバイスはハブを含み最大 3 個です。
- 最大 7 ポートまでのハブを 1 段のみ接続可能です(ルートハブは除く)。
- マスストレージクラスの Bulk Only Transport、コミュニケーションクラスの Abstract Control Model、及びハブクラスに対応しています。

## 目次

1. はじめに.....	3
2. 関数相関関係.....	5
3. 初期化.....	6
4. コントロール転送.....	10
5. インタラプトイン転送.....	21
6. バルク転送（マスタストレージクラス）.....	25
7. バルク転送（コミュニケーションクラス）.....	32
8. ディスクリプタ生成.....	35
9. アプリケーション（マスタストレージクラス）.....	41
10. アプリケーション（コミュニケーションクラス）.....	45
11. ハブクラスドライバ.....	50
12. 階層構造.....	52
13. 割り込み.....	55
14. データ構造.....	58
15. 参考ドキュメント.....	62

## 1. はじめに

### 1.1 仕様

- 制御可能な USB デバイスはハブを含み最大 3 個です。
- 最大 7 ポートまでのハブを、1 段のみ接続可能です(ルートハブは除く)。
- マスストレージクラスの Bulk Only Transport、コミュニケーションクラスの Abstract Control Model、及びハブクラスに対応しています。
- 他のクラスが必要な場合は、お客様で対応お願いいたします。
- マウント可能なメモリ (USB メモリ、SD カード等) の容量は最大 32G バイトです。
- ファイル操作のサンプルアプリケーションではメモリ、V850E2/ML4 USB ファンクションサンプル<sup>[1]</sup>、SH7216 USB ファンクションサンプル<sup>[4]</sup>に対し下記の動作をしております。
  - ルートに 10 個のファイル作成
  - ディレクトリを作成し、作成したディレクトリに 10 個のファイル作成
  - ブロック(512 バイト)単位で、1 ブロック~40 ブロックの書き込み、読み出し、比較
  - バイト単位で 1 バイト~512 バイトの書き込み、読み出し、比較
- FAT ファイルシステムはルネサス製ファイルシステムである M3S-TFAT-Tiny を使用しております。
- コミュニケーションクラスの通信では V850E2/ML4 USB ファンクションサンプル<sup>[2]</sup>のみで動作確認しております。
- コミュニケーションクラス通信のサンプルアプリケーションでは V850E2/ML4 USB ファンクションサンプル<sup>[2]</sup>に対し下記の動作をしております。
  - 通信速度、ストップビット、パリティ、データビット数等の設定
  - 通信速度、ストップビット、パリティ、データビット数等の取得
  - バイト単位で 1 バイト~64 バイトの書き込み、読み出し、比較
- 複数インタフェース (最大 3 個) を持ち、マスストレージクラスとコミュニケーションクラス対応の V850E2/ML4USB マルチファンクションサンプル<sup>[3]</sup>を制御可能です。

### 1.2 使用機能

- H バス
- USB ホスト・コントローラ
- 割り込み

### 1.3 適用条件

- マイコン: V850E2/ML4(uPD70F4022)
- 評価ボード: V850E2/ML4CPU ボード(型番 : R0K0F4022C000BR)
- 動作周波数:
  - 入力クロック: 10MHz
  - 内部システムクロック (fCLK): 200MHz
  - H バスクロック (fHCLK)33.3MHz
- 動作モード:
  - 通常動作モード
- 統合開発環境: ルネサス エレクトロニクス製 CubeSuite+ V1.02.00 [12 Apr 2012]
- C コンパイラ: ルネサス エレクトロニクス製 CX V1.21
- エミュレータ: ルネサス エレクトロニクス製 E1 エミュレータ

### 1.4 関連アプリケーションノート

- [1] V850E2/ML4 マイクロコンピュータ USB MSC(マスタストレージクラス)ドライバ編 アプリケーションノート(R01AN0011JJ)
- [2] V850E2/ML4 マイクロコンピュータ USB CDC(コミュニケーションクラス)ドライバ編 アプリケーションノート(R01AN0010JJ)
- [3] V850E2/ML4 マイクロコンピュータ USB マルチファンクション動作例 アプリケーションノート (R01AN1037JJ)
- [4] SH7216 グループ USB ファンクションモジュール USB マスタストレージクラス アプリケーションノート (RJJ06B1071)
- [5] V850E2M FAT ファイルシステムソフトウェア M3S-TFAT-Tiny: 導入ガイド(R01AN0128JJ)

## 2. 関数相関関係

関数の相関関係を示す。各関数の詳細は第3章以降で示す。

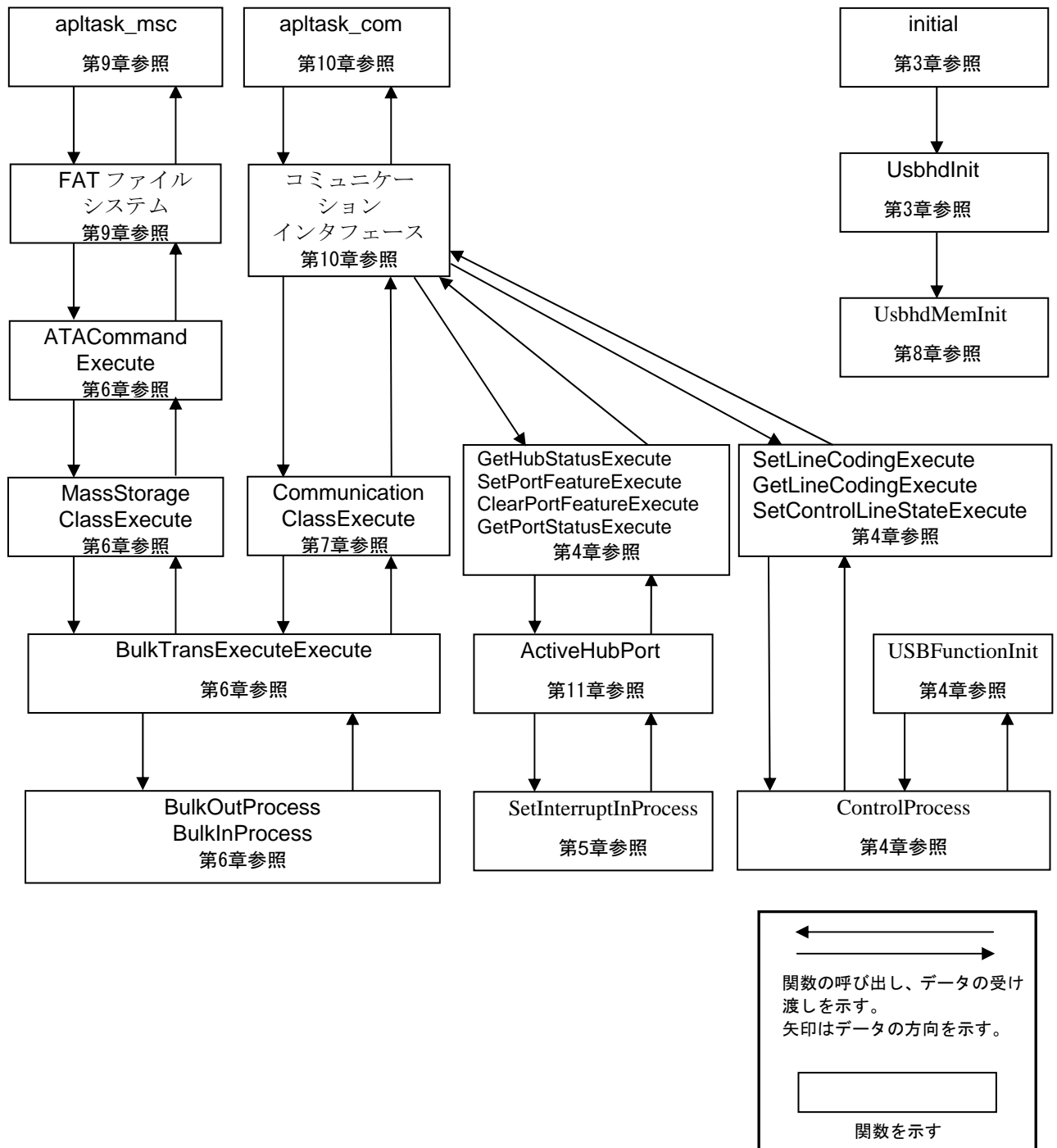


図 1 関数の相関関係

### 3. 初期化

初期化時における関数の相関図、仕様、動作を示す。

#### 3.1 初期化関数相関図

initial 関数が各関数を呼び出す。各関数は周辺モジュール毎に初期化を行う。

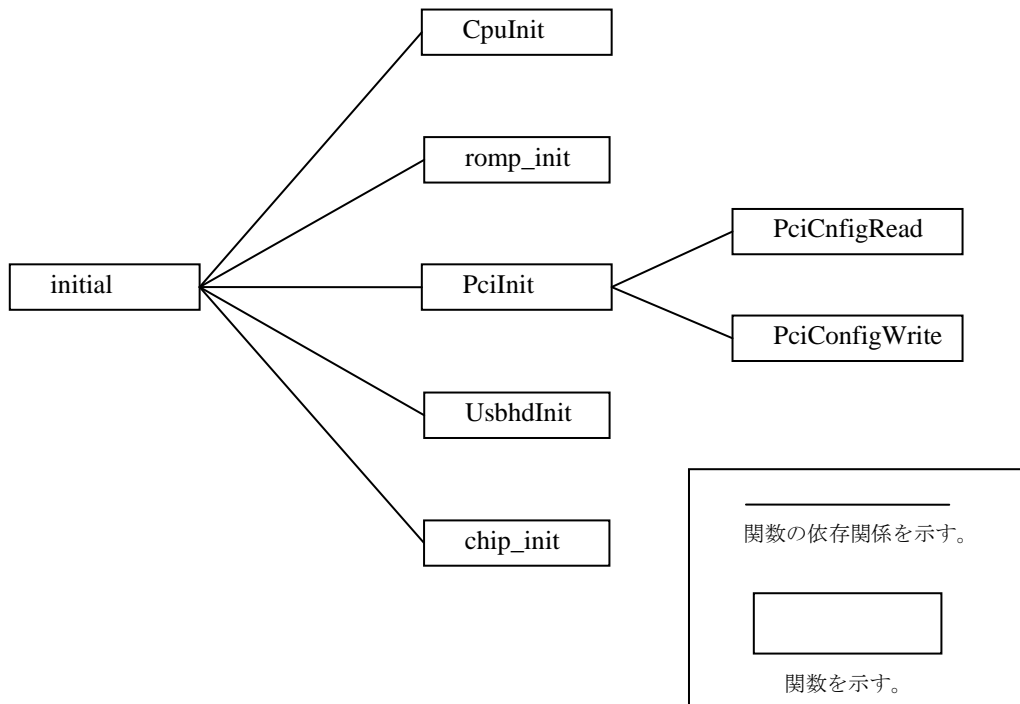


図 2 初期化時の関数相関関係

### 3.2 関数説明

関数名と処理内容を示す。

関数名	内容
initial	CPU 電源投入時に呼び出され、各種初期化関数群を呼び出す。実際のハードウェア初期化処理は下記の xxxInit 関数で実行する。この関数自体は処理を呼び出すのみとなる。
Cpulnit	V850E2/ML4 CPU の初期化を行う。
romp_init	ROM 化パッケージ用初期化処理 コピー関数 (_rcopy) を呼び出し、指定されたアドレスに格納されている情報を RAM 領域に 1 バイトずつコピーする。
Pcilnit	USB ホスト・コントローラのレジスタにアクセスするために PCI ブリッジの初期化を行う。
PciConfigRead	PCI コンフィギュレーション空間を通して USB ホストドライバレジスタのベースアドレスを読み出す
PciConfigWrite	PCI コンフィギュレーション空間を通して USB ホストドライバレジスタのベースアドレスを書き込む
UsbhdInit	USB ホスト・コントローラの初期化を行う。USB ホスト・コントローラのレジスタへのアクセスは PCI ブリッジを介して行うため、Pcilnit 処理後にこの関数を呼び出す必要がある。
UsbhdMemInit	ED(Endpoint Descriptor)、TD(Transfer Descriptor)で使用するグローバルメモリの初期化を行う。
chip_init	各ポートの初期設定を行う。

### 3.3 関数インタフェース

各関数のインタフェースを示す。

関数名	initial
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

関数名	Cpulnit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

関数名	romp_init
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

関数名	Pcilnit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

関数名	PciConfigRead	
パラメータ	RegisterNumber	ReadData
パラメータの内容	PCI コンフィギュレーションレジスタ番号	読み出したデータ
パラメータの範囲	0x10	32bit データ
戻り値	Status	
戻り値の内容	PciConfigRead が正常終了したかを判断する	
戻り値の範囲	0→正常終了 1→異常終了	

関数名	PciConfigWrite	
パラメータ	RegisterNumber	WriteData
パラメータの内容	PCI コンフィギュレーションレジスタ番号	USB ホスト・コントローラ・ベースアドレス
パラメータの範囲	0x10	32bit データ
戻り値	Status	
戻り値の内容	PciConfigWrite が正常終了したかを判断する	
戻り値の範囲	0→正常終了 1→異常終了	

関数名	UsbhdInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

関数名	UsbhdMemInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	UsbhdMemInit が正常終了したかを判断する
戻り値の範囲	0→正常終了 (エラー通知無し)



関数名	chip_init
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	void
戻り値の内容	無し
戻り値の範囲	無し

### 3.4 初期化動作

初期化時の関数の動作を示す。各関数はinitial関数からコールされる。各関数の戻り値はvoidであり、戻りメッセージは省略している。

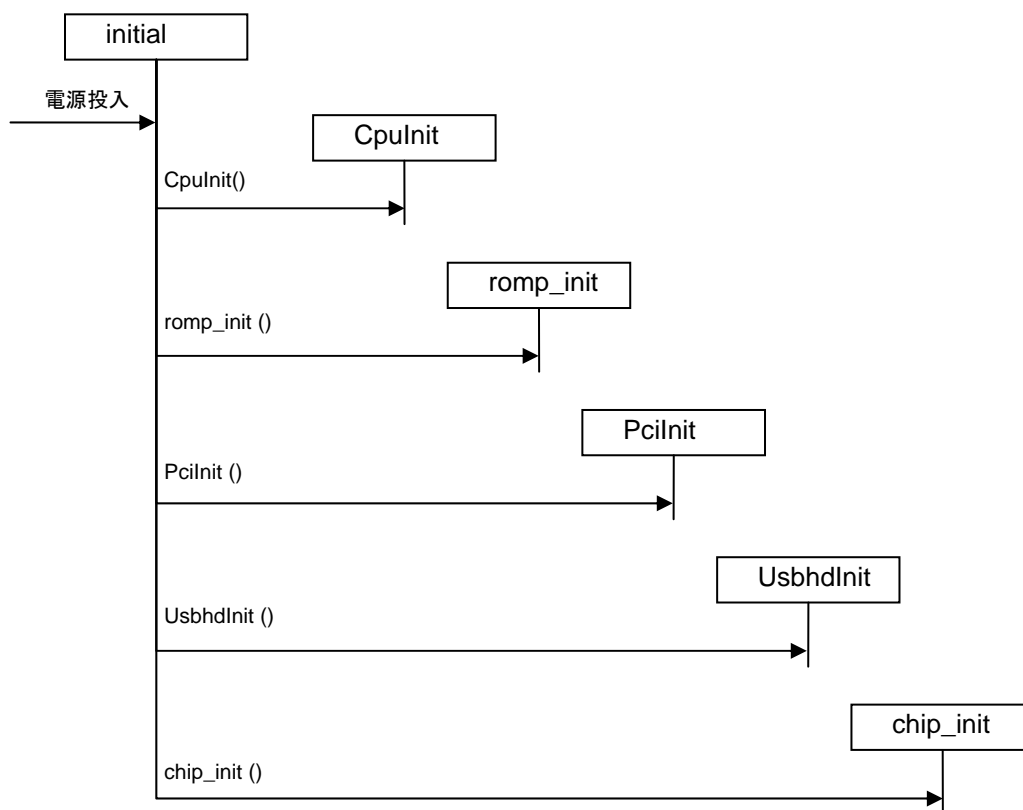


図 3 初期化関数動作

## 4. コントロール転送

コントロール転送における関数の相関図、仕様、動作を示す。

### 4.1 コントロール転送関数相関図

1 個目の USB デバイス接続、または、ユーザアプリケーションがハブのダウンストリームポートに接続される 3 個目の USB デバイスにアクセスする場合、UsbFunctionInit 関数を実行する。そして、エニユメレーションで各関数を実行する。

ハブクラスのクラスコマンドはハブのエニユメレーション完了後、またはハブのダウンストリームポートへの USB デバイス検出待ち状態で、ハブの状態取得、ハブのダウンストリームポートのリセット、有効化、状態取得時等に行われる。

マスストレージクラスのクラスコマンドはエニユメレーション完了後、マスストレージデバイスの初期化時に実行される。

コミュニケーションクラスのクラスコマンドはコミュニケーションクラスレイヤーからの要求で実行される。

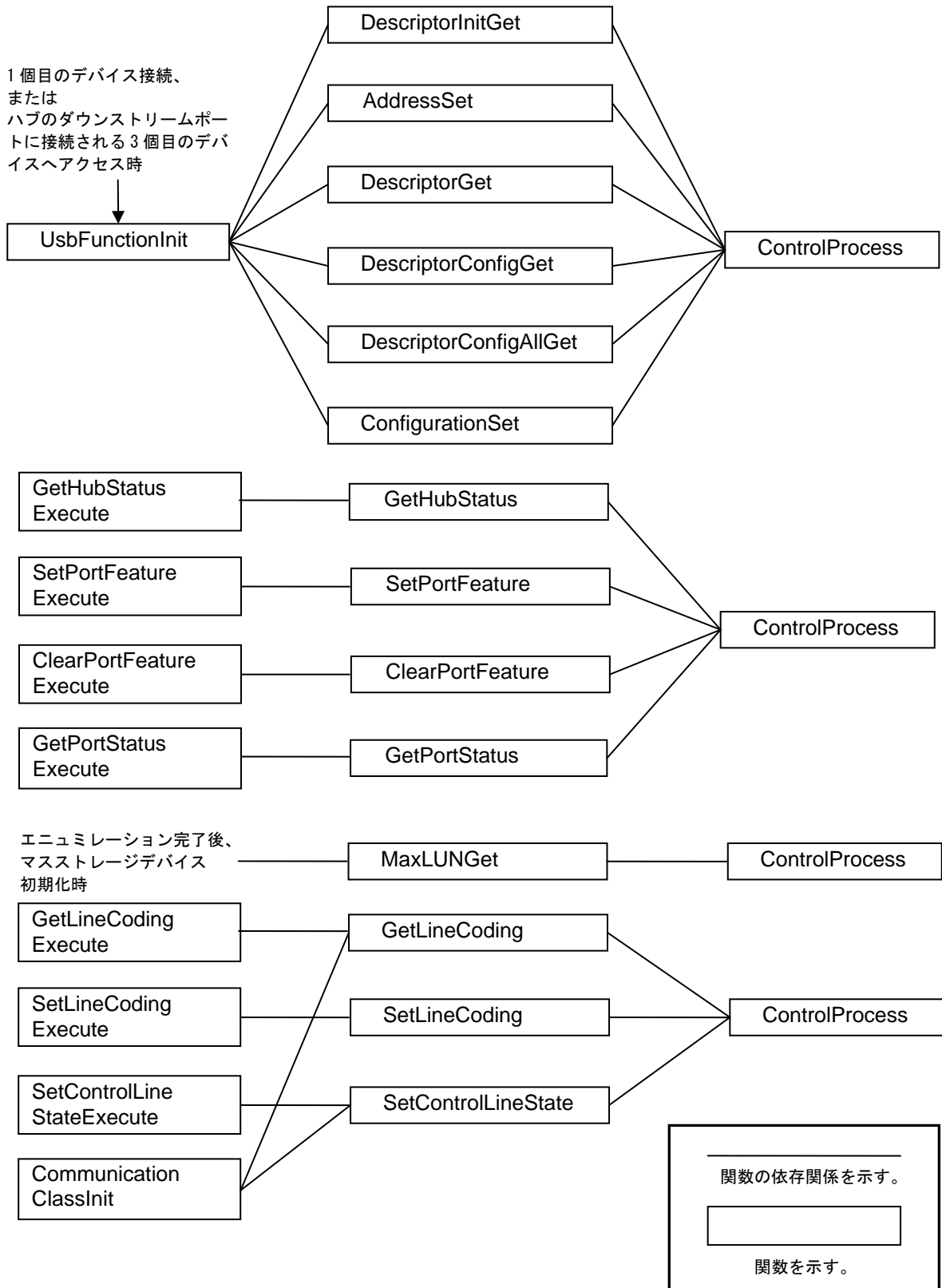


図 4 コントロール転送の関数相関関係

## 4.2 関数説明

関数名と処理内容を示す。

関数名	内容
UsbFunctionInit	<p>1 個目の USB デバイス接続、またはユーザアプリケーションがハブのダウンストリームポートに接続される 3 個目の USB デバイスにアクセスする場合、呼び出される。</p> <p>USB デバイスに対するアドレス設定、ディスクリプタ情報を取得する関数を呼び出す。また、ハブのデバイス接続先のダウンストリームポートを有効にする。</p> <ol style="list-style-type: none"> <li>標準デバイスリクエストデータ(デバイス)を取得し、ドライバ構造体(USBHD_DEV_INFO)にエンドポイント 0 の最大パケットサイズを設定する。</li> <li>USB デバイスにアドレスを割り当てる。</li> <li>標準デバイスリクエストデータ(デバイス)を取得する。</li> <li>標準デバイスディスクリプタ(コンフィギュレーション)を取得し、ドライバ構造体に標準デバイスディスクリプタ(コンフィギュレーション)長、インタフェース数、コンフィギュレーション値を設定する。</li> <li>標準デバイスディスクリプタ(コンフィギュレーション)の全てを取得し、ドライバ構造体に全インタフェース(最大 3 個)のエンドポイント情報を設定する。</li> <li>USB デバイスのエミュレーションを完了する。</li> <li>ハブのダウンストリームポートに接続された USB デバイスの検出を待つ。USB デバイスを検出した場合、対応するダウンストリームポートを有効にする。</li> </ol>
ControlProcess	<p>セットアップデータ転送処理時呼び出される。</p> <ol style="list-style-type: none"> <li>送受信の ED、セットアップステージ用 TD、データステージ用 TD、ステータスステージ用 TD 領域を取得する。</li> <li>ED に各種パラメータを設定し、セットアップステージ用 TD の Current Buffer Pointer に標準デバイスリクエストデータ(8byte)を設定する。</li> <li>データステージが存在しイン転送の場合、データステージ用 TD にデータ受信のバッファを設定する。</li> <li>データステージが存在しアウト転送の場合、データステージ用 TD にデータ送信のバッファを設定する。</li> <li>ED と TD をリンクさせ、ED のアドレスを ControlHeadED レジスタに設定し、HcCommandStatus レジスタに 0x02 を書き込み、コントロール転送を開始する。</li> <li>転送終了後 Done キューを確認し、正常に転送が終了していることを確認する。</li> </ol>
DescriptorInitGet	<p>USB デバイスへのアドレス割り当て前に、USB デバイスから標準デバイスディスクリプタ(デバイス)を最大 64 バイト取得する。</p> <ol style="list-style-type: none"> <li>標準デバイスリクエストデータ(8byte)を作成し、リクエストに Get Descriptor (0x06)を設定する。</li> <li>ControlProcess を呼び出し、転送処理を実行する。</li> </ol>

関数名	内容
AddressSet	USB デバイスにアドレスを割り当てる。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに Set Address(0x05)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
DescriptorGet	USB デバイスから標準デバイスディスクリプタ(デバイス)を取得する。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに Get Descriptor (0x06)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
DescriptorConfigGet	USB デバイスから標準デバイスディスクリプタ(コンフィギュレーション)を取得する(9byte)。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに Get Descriptor (0x06)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
DescriptorConfigAllGet	USB デバイスから標準デバイスディスクリプタ(コンフィギュレーション)を取得する(全コンフィギュレーションディスクリプタ)。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに Get Descriptor (0x06)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
GetHubDescriptor	USB デバイスからハブディスクリプタを取得する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに Get Descriptor (0x06)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
DeviceStatusGet	USB デバイスから DeviceStatus を取得する。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに GetStatusRequest(0x00)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
ConfigurationSet	USB デバイスに Set Configuration を発行する。 1. 標準デバイスリクエストデータ(8byte)を作成し、リクエストに Set Configuration(0x09)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
GetHubStatus	USB デバイスから HubStatus を取得する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに GetStatusRequest(0x00)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
SetPortFeature	USB デバイスに SetPortFeature を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに SetFeatureRequest(0x03)、インデックスに対象ポート番号を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
ClearPortFeature	USB デバイスに ClearPortFeature を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに ClearFeatureRequest(0x01)、インデックスに対象ポート番号を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。

関数名	内容
GetPortStatus	USB デバイスに GetPortStatus を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに GetStatusRequest(0x00)、インデックスに対象ポート番号を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
MaxLUNGet	USB デバイスに MaxLUNGet を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに GetMaxLun (0xFE)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
SetLineCoding	USB デバイスに SetLineCoding を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに SetLineCoding (0x20)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
GetLineCoding	USB デバイスに GetLineCoding を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに GetLineCoding (0x21)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。
SetControlLineState	USB デバイスに SetControlLineState を発行する。 1. クラスリクエストデータ(8byte)を作成し、リクエストに SetControlLineState (0x22)を設定する。 2. ControlProcess を呼び出し、転送処理を実行する。

#### 4.3 関数インタフェース

各関数のインタフェースを示す。ドライバ構造体(=UsbDevInfo)の詳細は第14章で示す。

関数名	UsbFunctionInit	
パラメータ	UsbDevInfo	dev_num
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	デバイス番号
パラメータの範囲	NULL 以外	0, 1, 2
戻り値	STATUS	
戻り値の内容	UsbFunctionInit が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	ControlProcess	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	ControlProcess が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	DescriptorInitGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DescriptorGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	AddressSet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	AddressSet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DescriptorGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DescriptorGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DescriptorConfigGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DescriptorConfigGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DescriptorConfigAllGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DescriptorConfigAllGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	GetHubDescriptor
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	GetHubDescriptor が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DeviceStatusGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DeviceStatusGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	ConfigurationSet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ConfigurationSet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	GetHubStatus
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	GetHubStatus が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	SetPortFeature
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	SetPortFeature が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了



関数名	ClearPortFeature
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ClearPortFeature が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	GetPortStatus
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	GetPortStatus が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	MaxLUNGet
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	MaxLUNGet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	SetLineCoding
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	SetLineCoding が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	GetLineCoding
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	GetLineCoding が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	SetControlLineState
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	SetControlLineState が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

#### 4.4 コントロール転送動作

コントロール転送時の関数の動的な振る舞いを示す。

コントロール転送には、エニューメレーション時の動作とクラスコマンド発行時の動作がある。

##### (1) エニューメレーション時

1 個目の USB デバイス接続、または、ユーザアプリケーションがハブのダウンストリームポートに接続される 3 個目の USB デバイスにアクセスする場合、UsbFunctionInit が実行される。デバイス接続時の割り込み発生時の詳細は、第13章を参照してください。

エニューメレーション時の動作を

図 5に示す。

DescriptorInitGet 関数で USB デバイスの最大パケットサイズを取得後、AddressSet 関数で USB デバイスに対するアドレスを設定し、DescriptorGet 関数などで USB デバイスの情報を取得する。また、USB デバイスがハブの場合、ハブディスクリプタも取得する。取得した情報はドライバ構造体に格納する。

##### (2) クラスコマンド発行時

クラスコマンド発行時の動作をコミュニケーションクラスの SetLineCoding と GetLineCoding の場合について図 6に示す。

アプリケーションから転送速度、ビット長等の通信パラメータ設定要求があると、SetLineCoding コマンドで通信パラメータを USB デバイスに設定する。

アプリケーションから転送速度、ビット長等の通信パラメータ取得要求があると、GetLineCoding コマンドで通信パラメータを USB デバイスから取得する。

他のハブクラス、マストレージクラス、コミュニケーションクラスのクラスコマンドも同様な動作であり、説明は省略する。

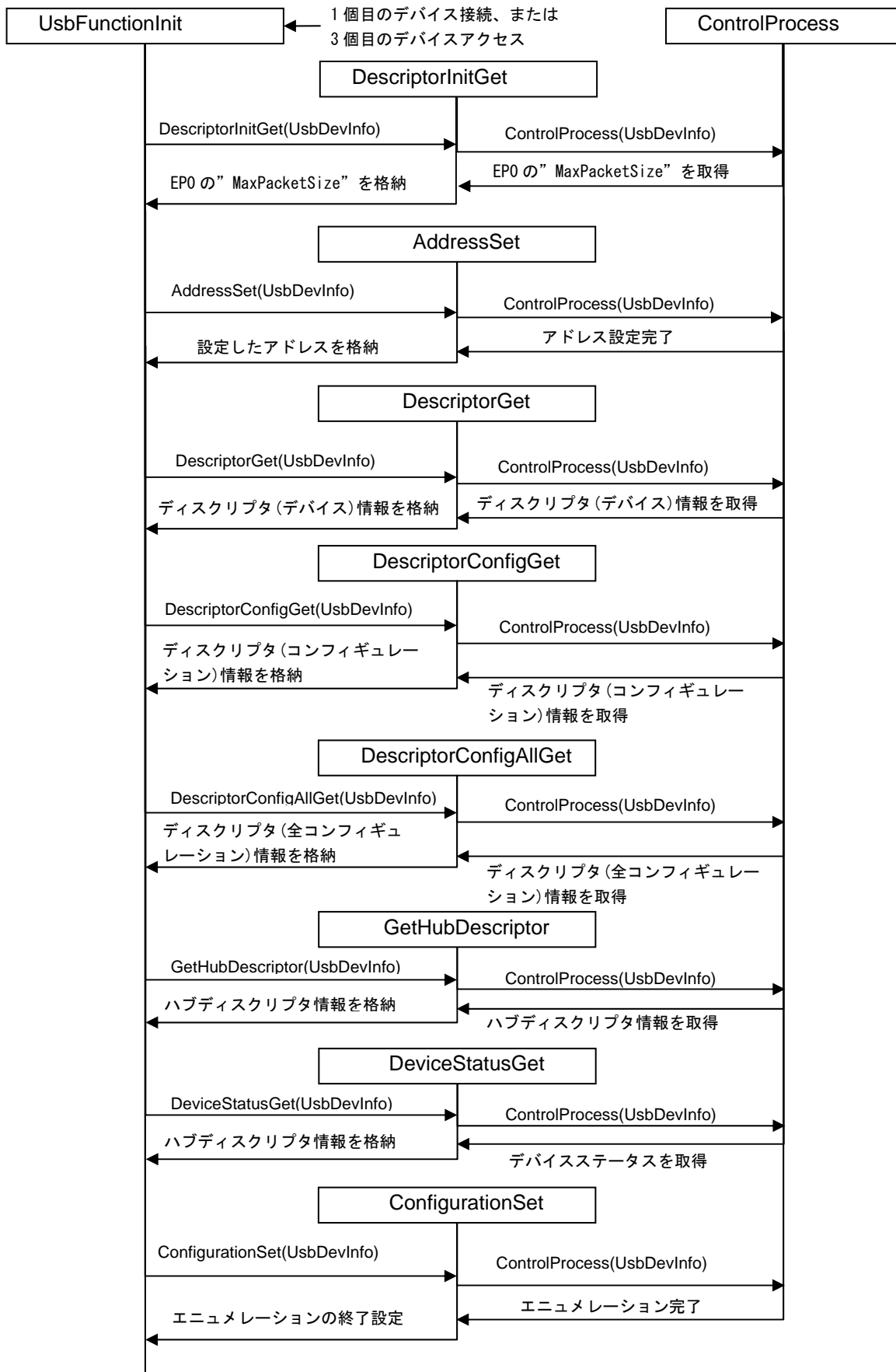


図 5 コントロール転送関数動作(エnumレーション時)

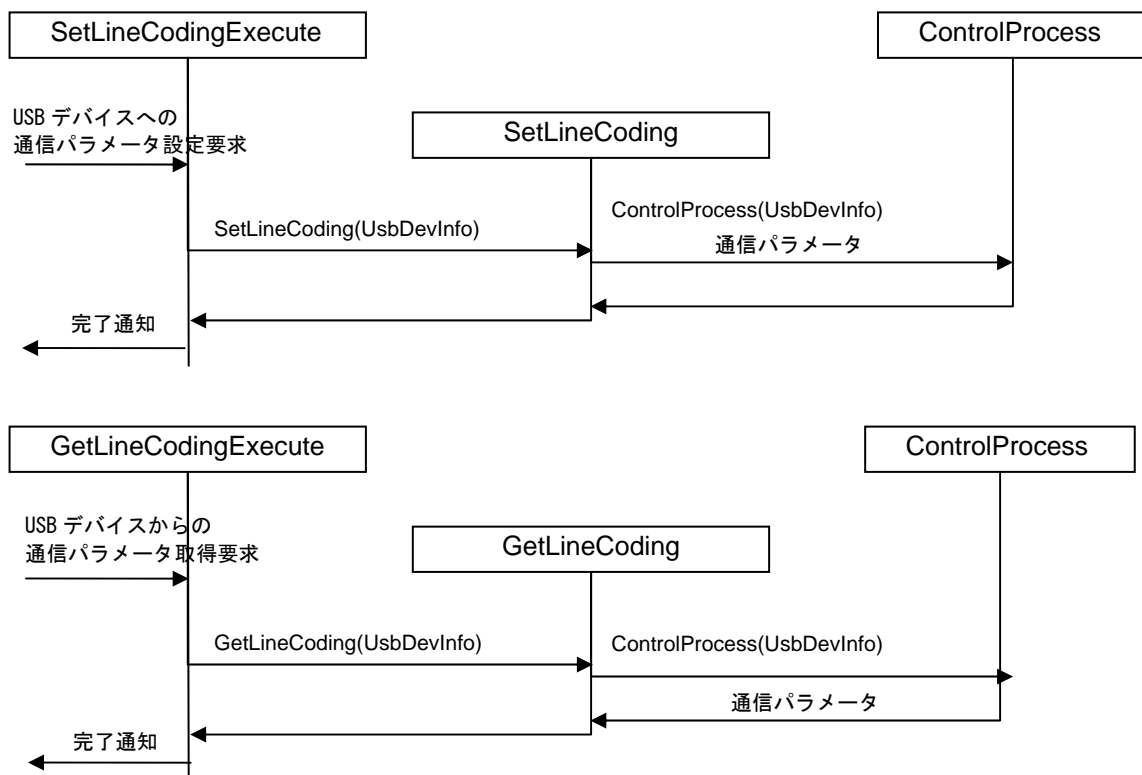


図 6 コントロール転送関数動作(クラスコマンド発行時)

## 5. インタラプトイン転送

インタラプトイン転送における関数の相関図、仕様、動作を示す。

### 5.1 インタラプトイン転送関数相関図

ハブのエnumレーション完了後、ハブのダウンストリームポートを有効にする ActiveHubPort 関数を実行する。そして、ActiveHubPort 関数内の SetInterruptInProcess 関数でハブの状態変化を検出するインタラプトイン転送を開始する。

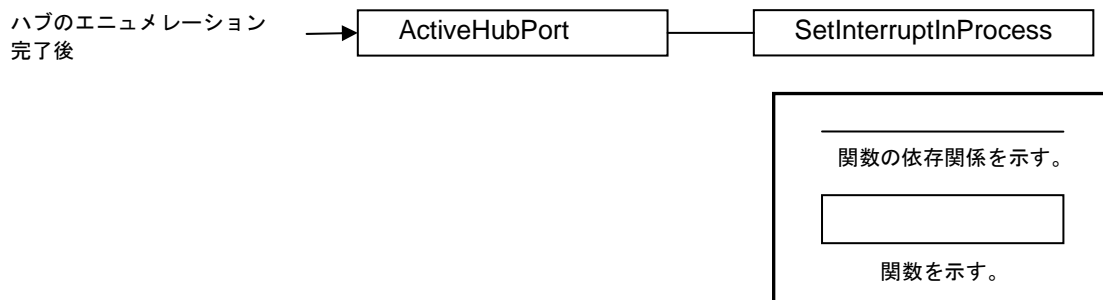


図 7 インタラプトイン転送の関数相関関係(ハブクラス)

### 5.2 関数説明

関数名と処理内容を示す。

関数名	内容
ActiveHubPort	<p>ハブのエnumレーション完了後、呼び出される。 ハブの状態変化を検出し、ハブのデバイス接続先のダウンストリームポートを有効にする。</p> <ol style="list-style-type: none"> <li>1. ハブの状態変化を検出するインタラプトイン転送を開始する。</li> <li>2. デバイス接続済みのダウンストリームポートを全て Powered-off 状態から Disconnected 状態に遷移させる。</li> <li>3. インタラプトイン転送によりハブの状態変化を待つ。</li> <li>4. デバイス接続済みのダウンストリームポートの 1 個をリセットする。</li> <li>5. インタラプトイン転送によりハブの状態変化を待つ。</li> <li>6. 4. でリセットしたダウンストリームポートが Enable 状態であることを確認する。</li> </ol>
SetInterruptInProcess	<p>ハブの状態変化を検出するインタラプトイン転送設定時呼び出され、インタラプトイン転送を開始する。</p> <ol style="list-style-type: none"> <li>1. 32 個の ED 領域、1 個の TD 領域を取得する。</li> <li>2. ED に各種パラメータを設定する。</li> <li>3. TD に各種パラメータを設定する。</li> <li>4. 32 個の ED を TD とリンクさせる。</li> <li>5. エンドポイントディスクリプタのインターバルを判定し、ED の Skip ビットを設定する。</li> <li>6. 32 個の ED のアドレスを HCCA の HccaInterruptTable に設定し、インタラプトイン転送を開始する。</li> </ol>

### 5.3 関数インタフェース

各関数のインタフェースを示す。ドライバ構造体(=UsbDevInfo)の詳細は第14章で示す。

関数名	ActiveHubPort
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ActiveHubPort が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	SetInterruptInProcess	
パラメータ	UsbDevInfo	Num
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	インタフェース番号
パラメータの範囲	NULL 以外	0, 1, 2
戻り値	STATUS	
戻り値の内容	SetInterruptInProcess が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

### 5.4 インタラプトイン転送動作

インタラプトイン転送は、**図 8**に示すように HCCA (Host Controller Communication Area)の HccaInterruptTable フィールドへ 32 個の ED (Endpoint Descriptor)アドレスを設定することで開始する。HC (Host Controller)は各 ED アドレスを 32msec 間隔でポーリングする。そして、ポーリングした ED に TD (Transfer Descriptor)がリンクされていれば ED と TD に従いインタラプトイン転送を行う。**図 8**は、インターバルが 8msec の場合を示しており、4 個の ED である IntInED[0]~IntInED[3]を IntInTD とリンクさせ、その後、ED を均等に配置する置換をすることで、インターバルが 8msec のインタラプトイン転送が HC により行われる。

インタラプトイン転送時の関数の動作を**図 9**に示す。

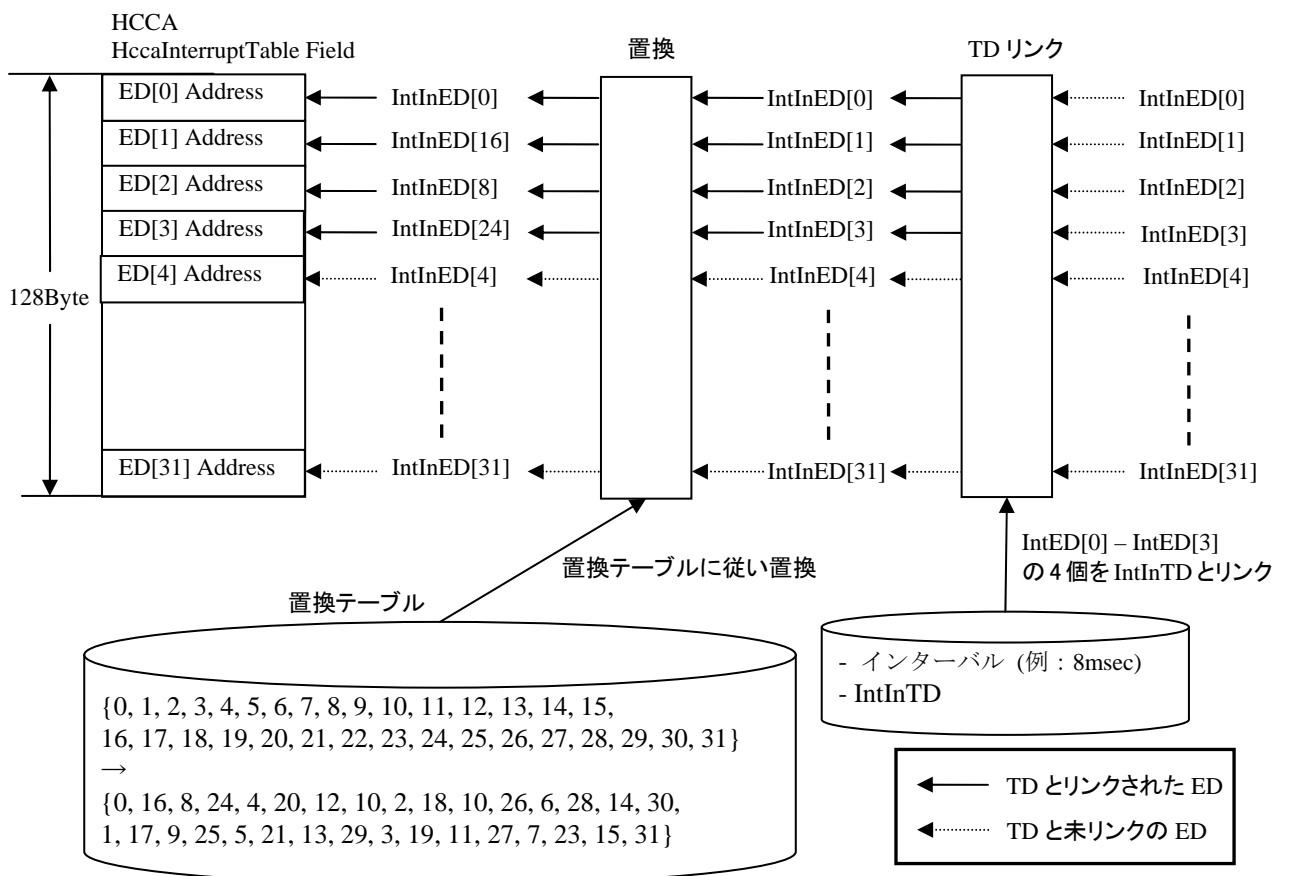


図 8 インタラプトイン転送での ED/TD 設定

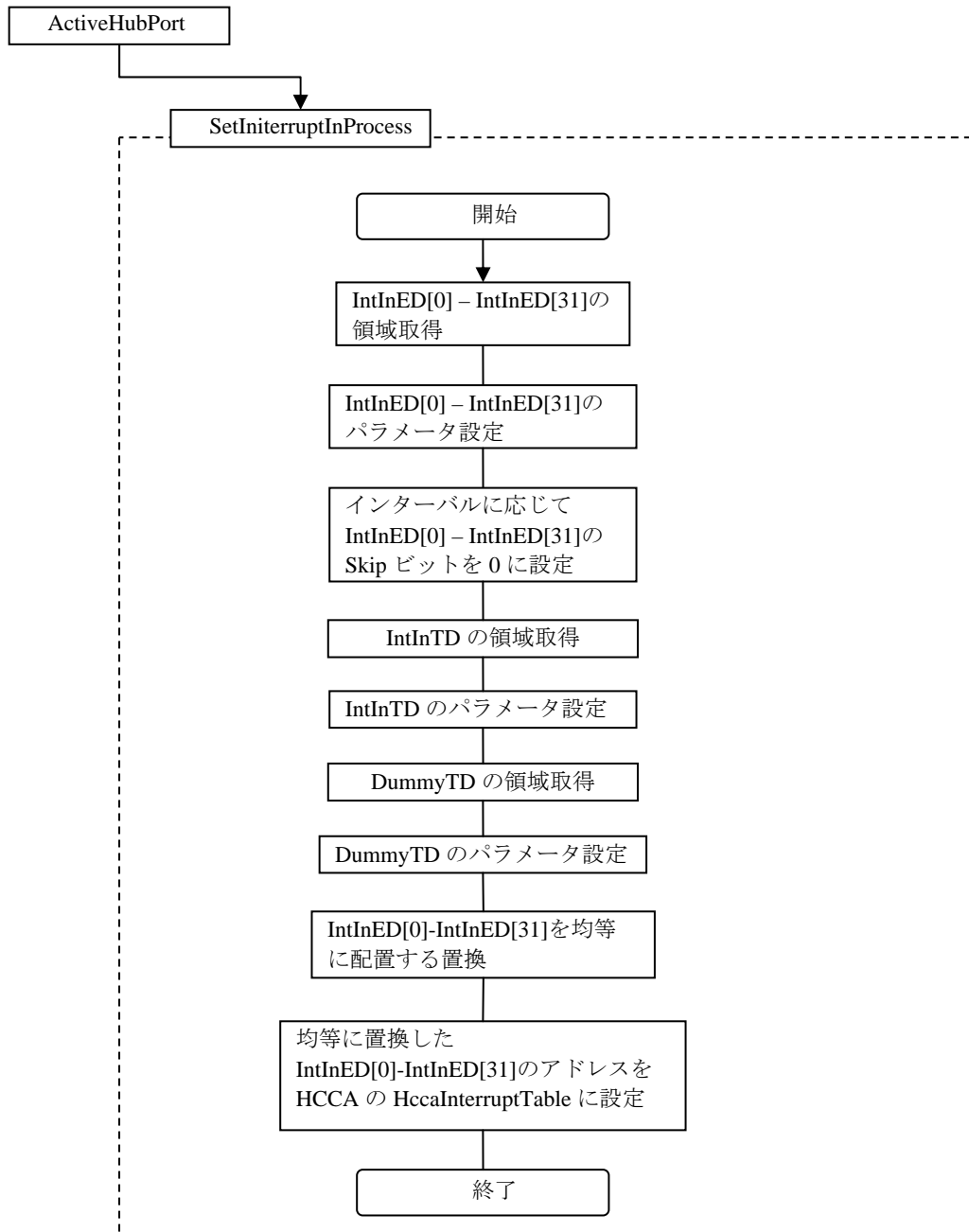


図 9 インタラプトイン転送関数動作



## 6. バルク転送（マストレージクラス）

マストレージクラスでのバルク転送における関数の相関図、仕様、動作を示す。

### 6.1 バルク転送関数相関図

ファイルシステムから ATACommandExecute 関数が呼び出され、ATAPI コマンドに変換し、バルク転送を実行する。

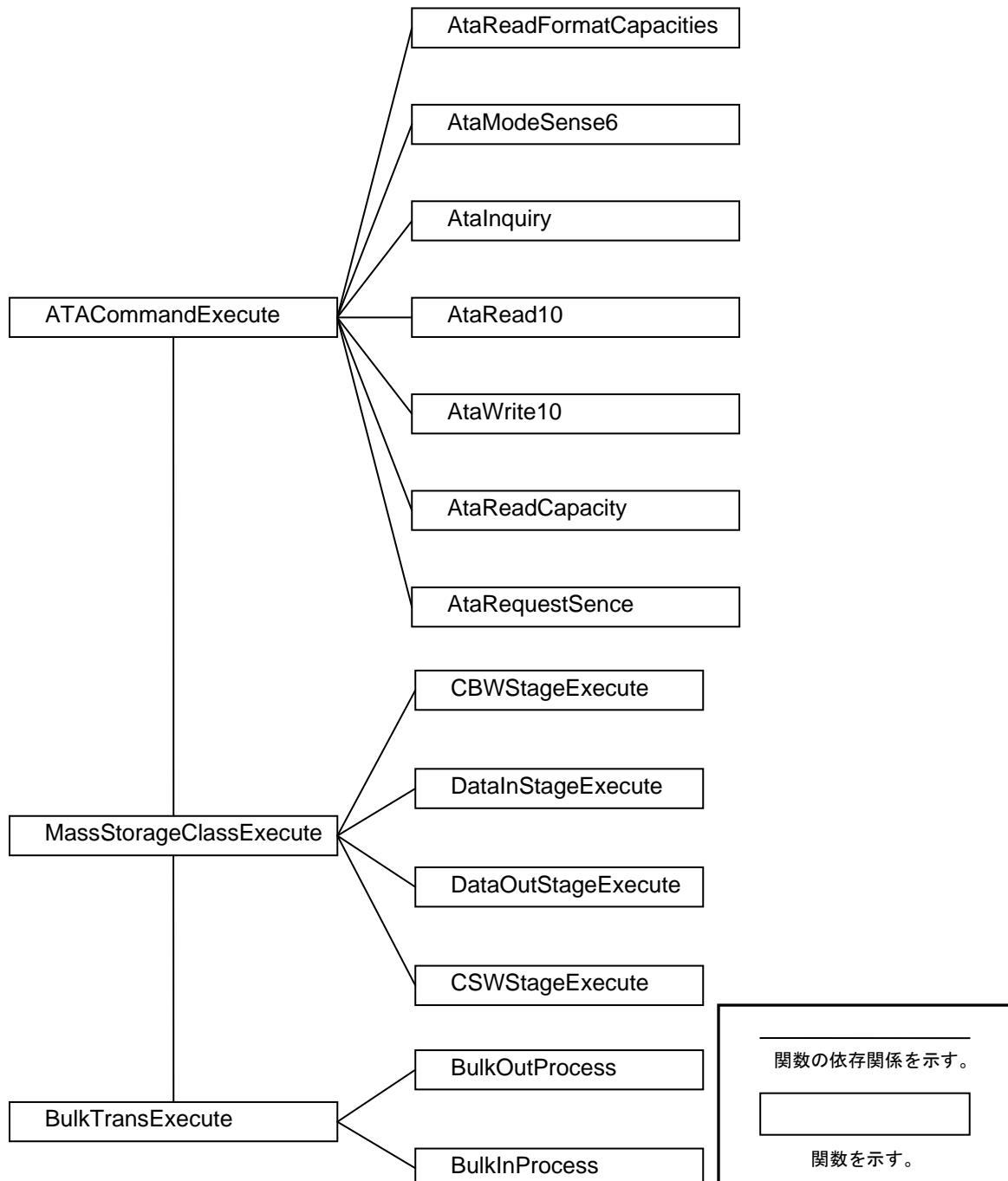


図 10 バルク転送の関数相関関係(マストレージクラス)

## 6.2 関数説明

関数名と処理内容を示す。

関数名	内容
ATACommandExecute	FATfilesystem から呼び出される。 Filesystem からのコマンドにより、AtaWrite10 等の関数を実行する。
AtaRead10	読み出すブロック番号、受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaWrite10	書き込むブロック番号、データ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaModeSense6	モードセンステーブルを読み込む。 受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaInquiry	Inquiry テーブルを読み込む。 受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaRequestSence	センスデータを読み込む。 受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaReadFormatCapacities	ReadFormatCapacity テーブルを読み込む。 受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
AtaReadCapacitys	ReadCapacity テーブルを読み込む。 受信バッファ領域等を格納するデータ構造体を作成し、MassStorageClassExecute 関数を呼び出す。
MassStorageClassExecute	ドライバ構造体を解析し、CBWStageExecute、DataInStageExecute、DataOutStageExecute、CSWStageExecute 関数を呼び出す。 転送終了後、正常に転送が終了できたことを確認する。 転送が異常終了した場合、再送処理等を行う。
CBWStageExecute	CBW データを作成し、BulkTransExecute 関数を呼び出す。
DataInStageExecute	データ受信用領域を設定し、BulkTransExecute 関数を呼び出す。
DataOutStageExecute	データ送信用領域の設定をし、BulkTransExecute 関数を呼び出す。
CSWStageExecute	CSW データを作成し、BulkTransExecute 関数を呼び出す。
BulkTransExecute	コマンドを解析し、BulkOutProcess、BulkInProcess 関数を呼び出す。 転送終了後、終了ステータスを確認する。
BulkOutProcess	1. バルクアウト用 ED、データ用 TD 領域を取得する。 2. Current Buffer Pointer にデータ送信用アドレスを設定する。 3. ED と TD をリンクさせ、ED のアドレスを HcBulkHeadED レジスタに設定し、HcCommandStatus レジスタに 0x04 を書き込み、バルク転送を開始する。 4. 転送終了後 Done キューを確認し、正常に転送が終了できたことを確認する。

関数名	内容
BulkInProcess	1. バルクイン用 ED、データ用 TD 領域を取得する。 2. Current Buffer Pointer にデータ受信アドレスを設定する。 3. ED と TD をリンクさせ、ED のアドレスを HcBulkHeadED レジスタに設定し、HcCommandStatus レジスタに 0x04 を書き込み、バルク転送を開始する。 4. 転送終了後 Done キューを確認し、正常に転送が完了できたことを確認する。

### 6.3 関数インタフェース

各関数のインタフェースを示す。USBDevInfo の詳細は第14章で示す。

関数名	ATACommandExecute	
パラメータ	UsbDevInfo	Command
パラメータの内容	ドライバ構造体	コマンド種別
パラメータの範囲	NULL 以外	ATA_INQUIRY、ATA_RREAD_FORMAT、ATA_RREAD_CAPACITY、ATA_RREAD_10、ATA_MODE_SENCE6、ATA_REQUEST_SENCE、ATA_WRITE_10
戻り値	STATUS	
戻り値の内容	ATACommandExecute が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	AtaRead10	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	AtaRead10 が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	AtaWrite10	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	AtaWrite10 が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	AtaModeSense6	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	AtaModeSense6 が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	AtalInquiry
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	AtalInquiry が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	AtaRequestSence
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	AtaRequestSence が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	AtaReadFormatCapacities
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	AtaReadFormatCapacities が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	AtaReadCapacity
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	AtaReadCapacity が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	MassStorageClassExecute
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	MassStorageClassExecute が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	CBWStageExecute
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	CBWStageExecute が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DataInStageExecute
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DataInStageExecute が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	DataOutStageExecute
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DataOutStageExecute が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	CSWStageExecute
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	CSWStageExecute が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	BulkTransExecute		
パラメータ	UsbDevInfo	Command	Num
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	コマンド種別	インタフェース番号
パラメータの範囲	NULL 以外	BULK_OUT_COMMAND、 BULK_IN_COMMAND	0, 1, 2
戻り値	STATUS		
戻り値の内容	BulkTransExecute が正常終了したかを判断する		
戻り値の範囲	0 →正常終了 0 以外 →異常終了		

関数名	BulkOutProcess	
パラメータ	UsbDevInfo	Num
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	インタフェース番号
パラメータの範囲	NULL 以外	0, 1, 2
戻り値	STATUS	
戻り値の内容	BulkOutProcess が正常終了したかを判断する	
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了	

関数名	BulkInProcess	
パラメータ	UsbDevInfo	Num
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	インタフェース番号
パラメータの範囲	NULL 以外	0, 1, 2
戻り値	STATUS	
戻り値の内容	BulkInProcess が正常終了したかを判断する	
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了	

## 6.4 Read/Write 動作

AtaRead10/AtaWrite10 実行時の動作を示す。

ファイルシステムからのアクセスで ATACommandExecute 関数が実行され、アクセスの種類により AtaRead10/Write10 関数等が実行される。AtaRead10/Write10 関数等から MassStorageClassExecute 関数が呼び出され、バルク転送を行う。

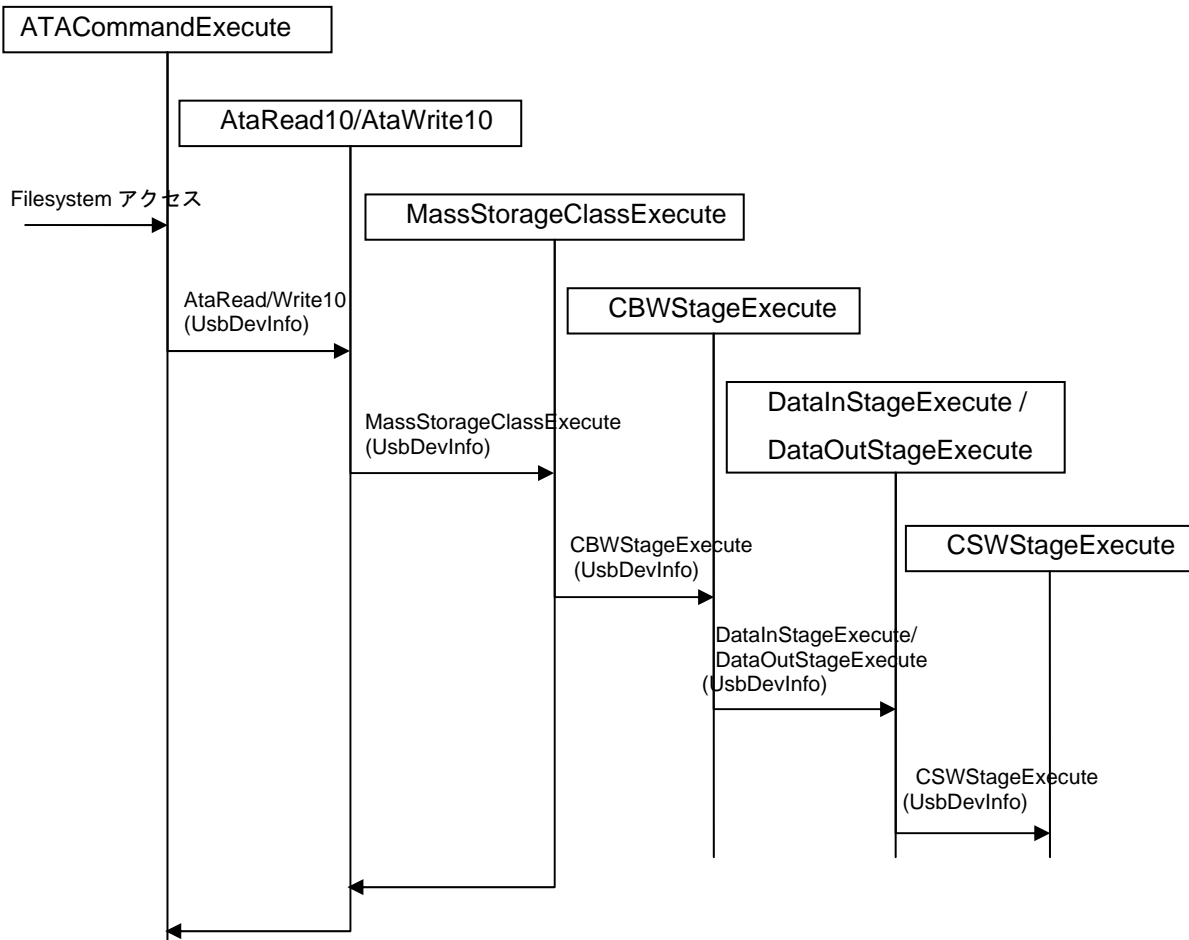


図 11 Read/Write 関数動作

## 7. バルク転送（コミュニケーションクラス）

コミュニケーションクラスでのバルク転送における関数の相関図と仕様を示す。

### 7.1 バルク転送関数相関図

コミュニケーションクラスインタフェースの ComRead、ComWrite 関数から CommunicationClassExecute 関数が呼び出され、バルク転送を実行する。

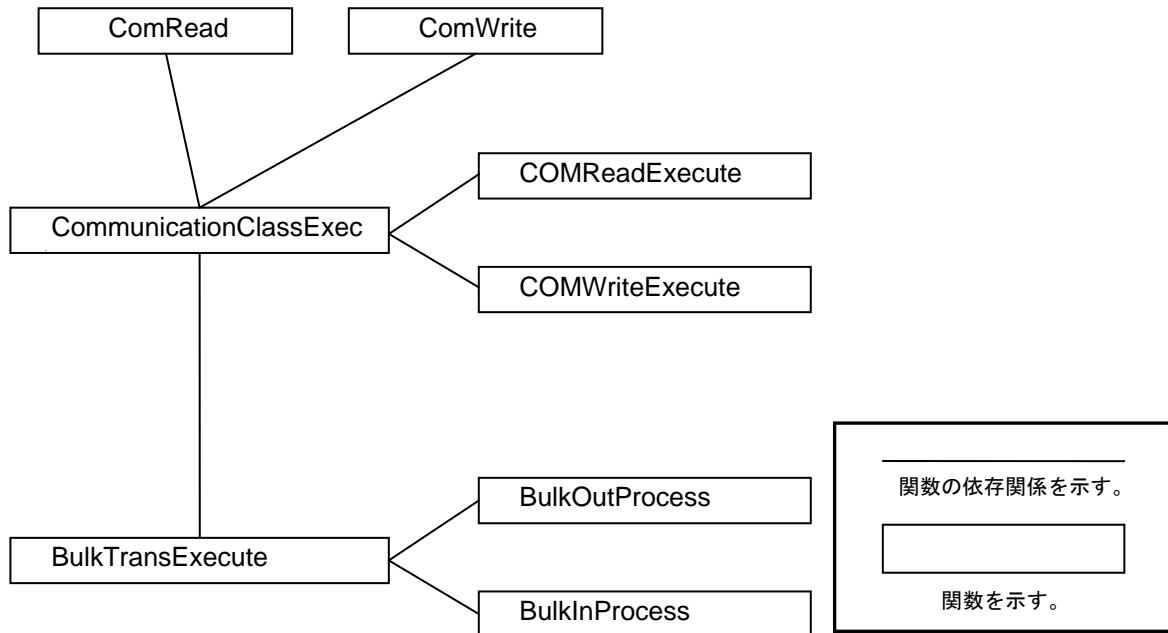


図 12 バルク転送の関数相関関係(コミュニケーションクラス)



## 7.2 関数説明

関数名と処理内容を示す。

関数名	内容
ComRead	ユーザアプリケーションから読み出し要求として呼び出される。 CommunicationClassExec の関数を実行する。
ComWrite	ユーザアプリケーションから書き込み要求として呼び出される。 CommunicationClassExec の関数を実行する。
CommunicationClassExecute	ドライバ構造体を解析し、COMReadExecute、COMWriteExecute 関数を呼び出す。 転送終了後、正常に転送が終了できたことを確認する。 転送が異常終了した場合、再送処理等を行う。
COMReadExecute	データ受信領域を設定し、BulkTransExecute 関数を呼び出す。
COMWriteExecute	データ送信領域の設定をし、BulkTransExecute 関数を呼び出す。
BulkTransExecute	コマンドを解析し、BulkOutProcess、BulkInProcess 関数を呼び出す。 転送終了後、終了ステータスを確認する。
BulkOutProcess	1. バルクアウト用 ED、データ用 TD 領域を取得する。 2. Current Buffer Pointer にデータ送信アドレスを設定する。 3. ED と TD をリンクさせ、ED のアドレスを HcBulkHeadED レジスタに設定し、HcCommandStatus レジスタに 0x04 を書き込み、バルク転送を開始する。 4. 転送終了後 Done キューを確認し、正常に転送が終了できたことを確認する。
BulkInProcess	1. バルクイン用 ED、データ用 TD 領域を取得する。 2. Current Buffer Pointer にデータ受信アドレスを設定する。 3. ED と TD をリンクさせ、ED のアドレスを HcBulkHeadED レジスタに設定し、HcCommandStatus レジスタに 0x04 を書き込み、バルク転送を開始する。 4. 転送終了後 Done キューを確認し、正常に転送が終了できたことを確認する。

## 7.3 関数インタフェース

各関数のインタフェースを示す。USBDevInfo の詳細は第14章で示す。なお、BulkTransExecute、BulkOutProcess、BulkInProcess については第6.3節を参照ください。

関数名	ComRead	
パラメータ	buffer	CntByte
パラメータの内容	受信データバッファ	受信要求バイト数
パラメータの範囲	NULL 以外	1 ~ 64
戻り値	読み出し完了バイト数	
戻り値の内容	ComRead の正常終了と読み出し完了バイト数の情報	
戻り値の範囲	1 以上	読み出し完了バイト数
	-1 以下	→異常終了

関数名	ComWrite	
パラメータ	buffer	CntByte
パラメータの内容	送信データバッファ	送信要求バイト数
パラメータの範囲	NULL 以外	1 ~ 64
戻り値	書き込み完了バイト数	
戻り値の内容	ComWrite の正常終了と読み出し完了バイト数の情報	
戻り値の範囲	1 以上 書き込み完了バイト数 -1 以下 →異常終了	

関数名	CommunicationClassExecute	
パラメータ	UsbDevInfo	Command
パラメータの内容	ドライバ構造体 (USB デバイス情報、転送情報)	コマンド種別
パラメータの範囲	NULL 以外	COMReadExecute、 COMWriteExecute
戻り値	STATUS	
戻り値の内容	CommunicationClassExecute が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	COMReadExecute	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	COMReadExecute が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	COMWriteExecute	
パラメータ	UsbDevInfo	
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	
パラメータの範囲	NULL 以外	
戻り値	STATUS	
戻り値の内容	COMWriteExecute が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

## 8. ディスクリプタ生成

ディスクリプタ生成における関数の相関図、仕様、動作を示す。

### 8.1 ディスクリプタ生成関数相関図

ディスクリプタ生成関数相関図を示す。ED、TD は UsbhdMemInit 関数で生成される。

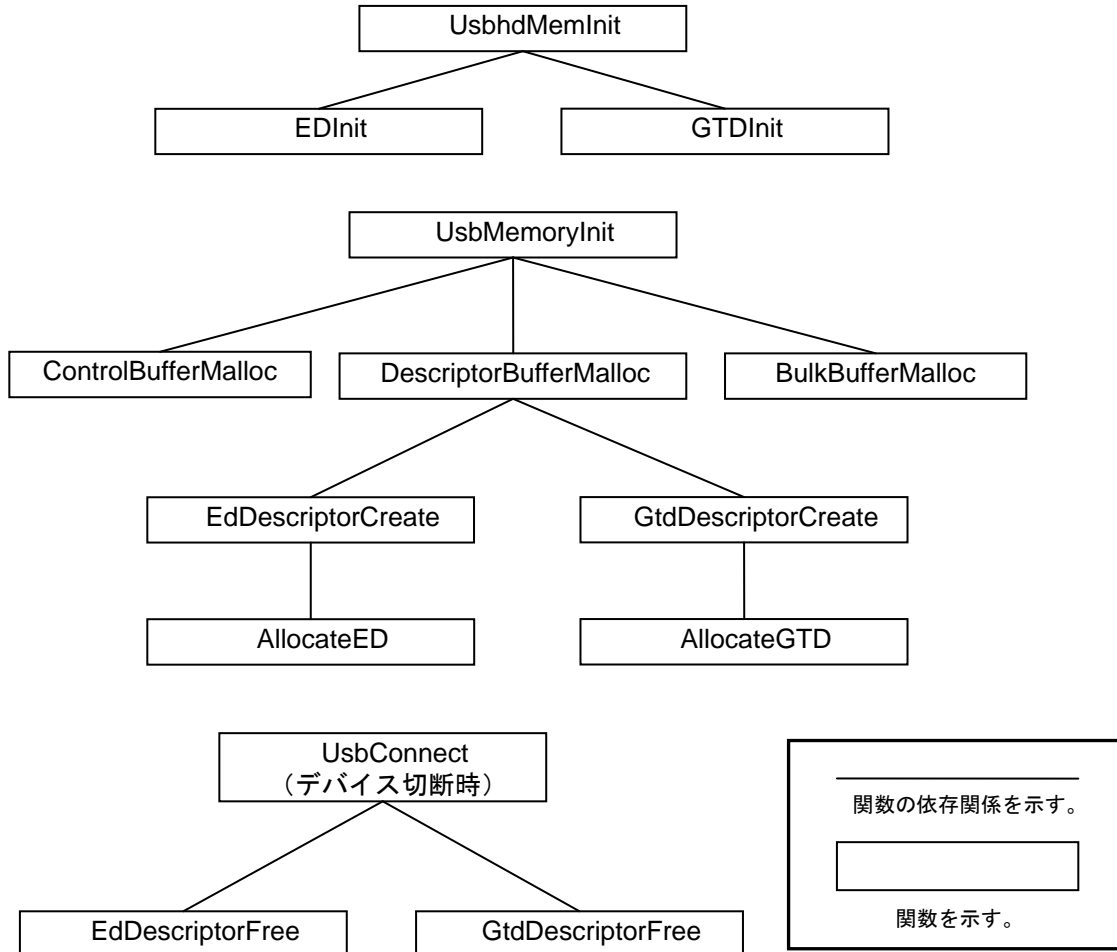


図 13 ディスクリプタ生成の関数相関関係

## 8.2 関数説明

関数名と処理内容を示す。

関数名	内容
UsbhdMemInit	ED、TD のパラメータを取得し、EDInit 関数、GTDInit 関数を呼び出す。
EDInit	ED 用ディスクリプタ領域を初期化する。
GTDInit	TD 用ディスクリプタ領域を初期化する。
UsbMemoryInit	USB デバイス接続時、コントロール転送、バルク転送用バッファを初期化する。
ControlBufferMalloc	コントロール転送で使用するデータ領域の確保を行う。
BulkBufferMalloc	バルク転送とインタラプトイン転送で使用するデータ領域の確保を行う。
DescriptorBufferMalloc	ED と TD の設定を行う。 コントロール転送は、ED が 1 個、TD が 4 個となる。 バルク転送は、インタフェース毎 (3 個) に ED が 2 個、TD が 4 個となる。 インタラプトイン転送は、ED が 32 個、TD が 2 個となる。
EdDescriptorCreate	ED 領域から 1 つの ED を割り当て、そのアドレスを返す。
GtdDescriptorCreate	TD 領域から 1 つの TD を割り当て、そのアドレスを返す。
AllocateED	ED 領域から未使用の ED を検索し、そのアドレスを返す。
AllocateGTD	TD 領域から未使用の TD を検索し、そのアドレスを返す。
EdDescriptorFree	指定された ED をチェーンから外し、未使用状態に更新する。
GtdDescriptorFree	指定された TD をチェーンから外し、未使用状態に更新する。

### 8.3 関数インタフェース

各関数のインタフェースを示す。

関数名	UsbhdMemInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	UsbhdMemInit が正常終了したかを判断する
戻り値の範囲	0 → 正常終了

関数名	EDInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	EDInit が正常終了したかを判断する
戻り値の範囲	0 → 正常終了

関数名	GTDInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	GTDInit が正常終了したかを判断する
戻り値の範囲	0 → 正常終了

関数名	UsbMemoryInit
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	UsbMemoryInit が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了

関数名	ControlBufferMalloc
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ControlBufferMalloc が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了

関数名	BulkBufferMalloc
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	BulkBufferMalloc が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了

関数名	DescriptorBufferMalloc
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	DescriptorBufferMalloc が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了

関数名	EdDescriptorCreate
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	USBHD_ED_DESC
戻り値の内容	割り当てる ED のアドレス
戻り値の範囲	NULL 以外 → 正常終了 NULL → 空き ED 無し

関数名	GtdDescriptorCreate
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	USBHD_GTD_DESC
戻り値の内容	割り当てる TD のアドレス
戻り値の範囲	NULL 以外 → 正常終了 NULL → 空き TD 無し

関数名	AllocateED
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	DESC_ED_MEMORY
戻り値の内容	未使用 ED のアドレス
戻り値の範囲	NULL 以外 → 正常終了 NULL → 空き ED 無し

関数名	AllocateGTD
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	DESC_GTD_MEMORY
戻り値の内容	未使用 TD のアドレス
戻り値の範囲	NULL 以外 → 正常終了 NULL → 空き TD 無し

関数名	EdDescriptorFree
パラメータ	ED
パラメータの内容	開放する ED のアドレス
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	EdDescriptorFree が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 -1 → 異常終了

関数名	GtdDescriptorFree
パラメータ	GTD
パラメータの内容	開放する TD のアドレス
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	GtdDescriptorFree が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 -1 → 異常終了

#### 8.4 ED/TD 取得動作

ED/TD 取得時の動作を示す。

UsbhdMemInit で DE、TD のベースアドレスを設定し、EDInit/GTDInit を呼び出す。

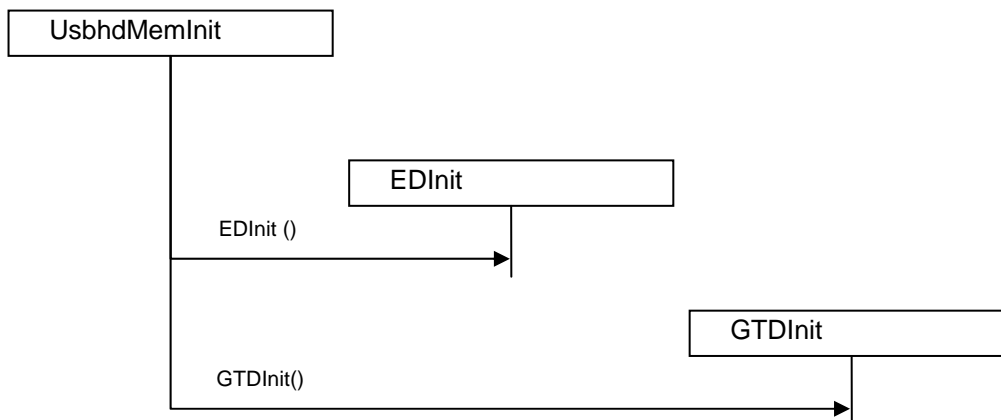


図 14 ED/TD 取得関数動作

### 8.5 ED/TD 設定動作

ED/TD 設定時の動作を示す。

UsbMemoryInit でコントロール転送、バルク転送、インタラプトイン転送の ED、TD 領域を設定する。また、コントロール転送、バルク転送、インタラプトイン転送で使用するデータ領域の確保をする。

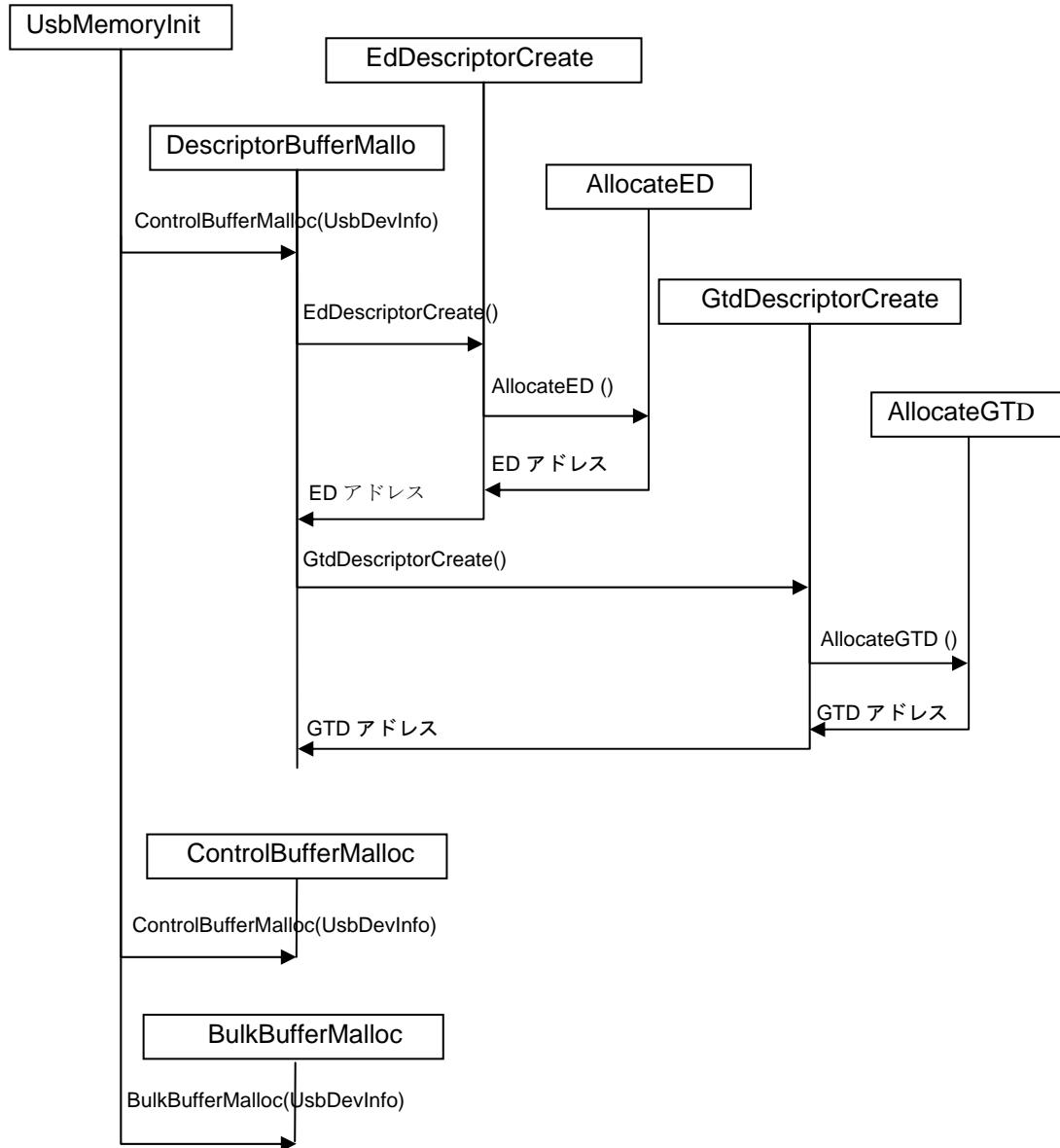


図 15 ED/TD 設定関数動作



## 9. アプリケーション（マスタレイジクラス）

アプリケーション（マスタレイジクラス）における関数の相関図、仕様、動作を示す。

### 9.1 アプリケーション相関図

アプリケーションの関数相関図を示す。

FAT ファイルシステムはルネサス製ファイルシステムである M3S-TFAT-Tiny（以下、TFAT）を使用しています。

本章ではアプリケーションで使用する代表的な関数の使用方法を説明します。その他関数の使用方法については TFAT アプリケーションノート<sup>[5]</sup>を参照ください。

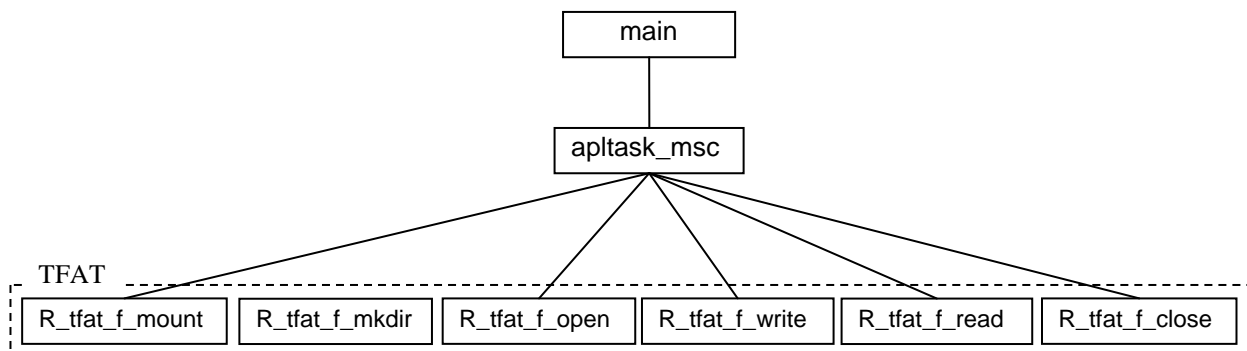


図 16 アプリケーションの関数相関関係

### 9.2 関数説明

関数名と処理内容を示す。

関数名	内容
apltask_msc	main 関数から呼ばれ、アプリケーション処理を実行する。
R_tfat_f_mount	FAT ファイルシステム作業領域を登録し、下位のドライバと関連付けます。
R_tfat_f_mkdir	ディレクトリを作成します。
R_tfat_f_open	既存のファイルを開きます。または、新規にファイルを作成します。
R_tfat_f_write	ファイルにデータを書き込みます。
R_tfat_f_read	ファイルからデータを読み出します。
R_tfat_f_close	開いているファイルを閉じます。

### 9.3 関数インタフェース

各関数のインタフェースを示す。

関数名	apltask_msc	
パラメータ	void	
パラメータの内容	無し	
パラメータの範囲	無し	
戻り値	STATUS	
戻り値の内容	apltask_msc が正常終了したかを判断する	
戻り値の範囲	0→正常終了 -1→異常終了	

関数名	R_tfat_f_mount	
パラメータ	論理ドライブ番号	作業領域
パラメータの内容	作業領域を登録または解除する論理ドライブ番号	作業領域へのポインタ
パラメータの範囲	常に 0	NULL 以外
戻り値	登録または解除の結果	
戻り値の内容	R_tfat_f_mount が正常終了したかを判断する	
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了	

関数名	R_tfat_f_mkdir	
パラメータ	ディレクトリ名	
パラメータの内容	作成するディレクトリ名へのポインタ	
パラメータの範囲	ASCII コード、ショートファイル名 (8.3 形式)	
戻り値	ディレクトリ作成結果	
戻り値の内容	R_tfat_f_mkdir が正常終了したかを判断する	
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了	

関数名	R_tfat_f_open		
パラメータ	ファイルポインタ	ファイル名	属性
パラメータの内容	開いたファイルの情報を格納する構造体へのポインタ	開くファイル名	ファイルの属性
パラメータの範囲	NULL 以外	ASCII コード、ショートファイル名 (8.3 形式)	例： TFAT_FA_OPEN_ALWAYS：ファイルを開く 存在しない場合、新規に作成する TFAT_FA_WRITE：書き込み可能 TFAT_FA_READ：読み出し可能
戻り値	ファイルを開いた（または作成）結果		
戻り値の内容	R_tfat_f_open が正常終了したかを判断する		
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了		

関数名	R_tfat_f_write			
パラメータ	ファイルポインタ	バッファ	要求サイズ	完了サイズ
パラメータの内容	開いたファイルの情報を格納する構造体へのポインタ	書き込みデータバッファ	書き込み要求サイズ (バイト数)	書き込み完了サイズ (バイト数)
パラメータの範囲	R_tfat_f_open で取得したファイルポインタ	NULL 以外	NULL 以外	-
戻り値	データ書き込み結果			
戻り値の内容	R_tfat_f_write が正常終了したかを判断する			
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了			

関数名	R_tfat_f_read			
パラメータ	ファイルポインタ	バッファ	要求サイズ	完了サイズ
パラメータの内容	開いたファイルの情報を格納する構造体へのポインタ	読み出しデータバッファ	読み出し要求サイズ (バイト数)	読み出し完了サイズ (バイト数)
パラメータの範囲	R_tfat_f_open で取得したファイルポインタ	NULL 以外	NULL 以外	-
戻り値	データ読み出し結果			
戻り値の内容	R_tfat_f_read が正常終了したかを判断する			
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了			

関数名	R_tfat_f_close	
パラメータ	ファイルポインタ	
パラメータの内容	開いたファイルの情報を格納する構造体へのポインタ	
パラメータの範囲	R_tfat_f_open で取得したファイルポインタ	
戻り値	ファイルを閉じた結果	
戻り値の内容	R_tfat_f_close が正常終了したかを判断する	
戻り値の範囲	TFAT_FR_OK→正常終了 TFAT_FR_OK 以外→異常終了	

### 9.4 アプリケーション動作

サンプルアプリケーションは下記の動作をする。

- ファイルをルートに 10 個のファイル作成
- ディレクトリを作成し、作成したディレクトリに 10 個のファイル作成
- ブロック(512 バイト)単位で、書き込み後、読み出してデータ比較 (1 ブロック~40 ブロック)
- バイト単位で、書き込み後、読み出してデータ比較 (1 バイト~1024 バイト)

図 17には書き込み後、読み出してデータ比較をする例を示しています。

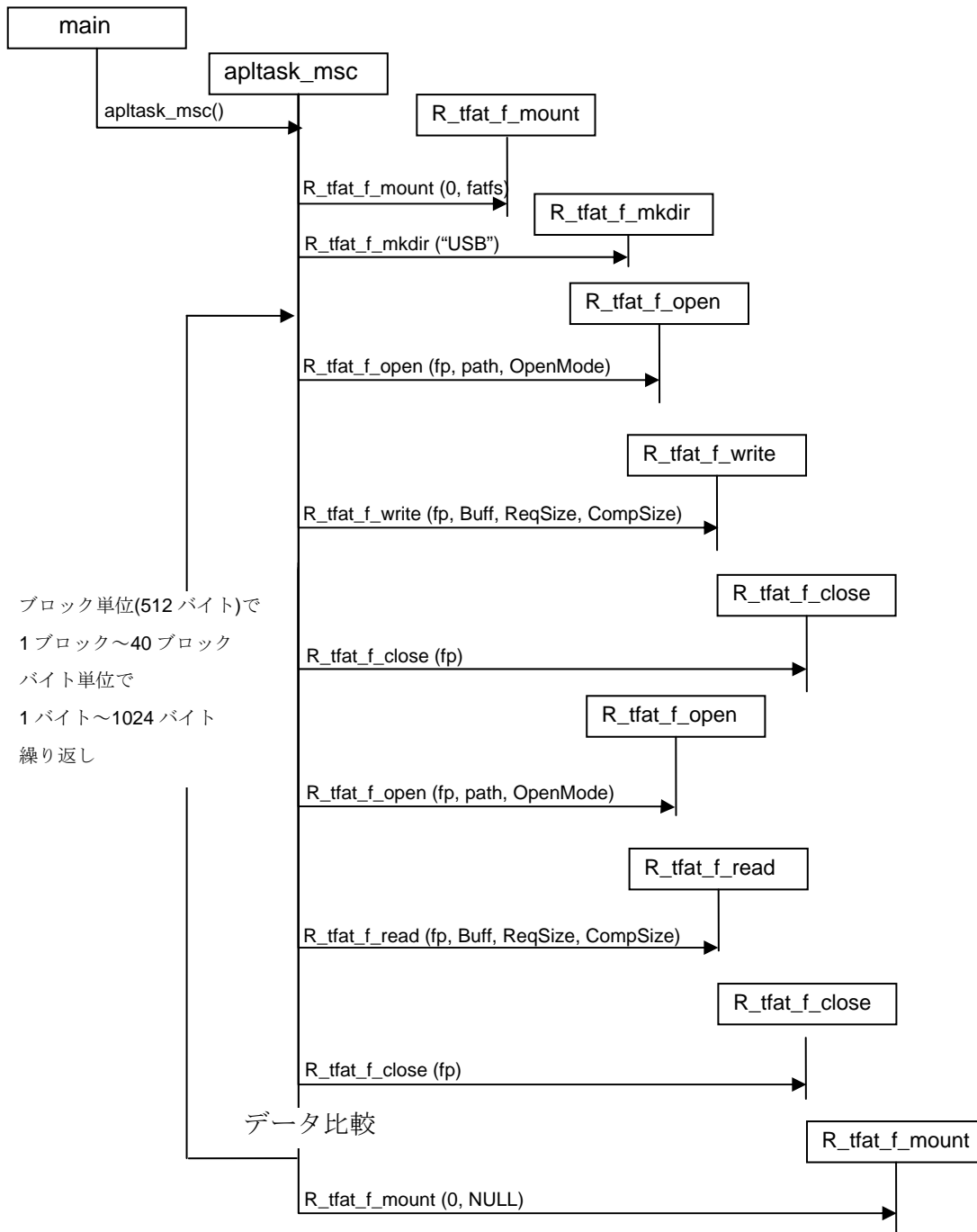


図 17 アプリケーション動作

## 10. アプリケーション（コミュニケーションクラス）

アプリケーション（コミュニケーションクラス）における関数の相関図、仕様、動作を示す。

### 10.1 アプリケーション相関図

アプリケーションの関数相関図を示す。

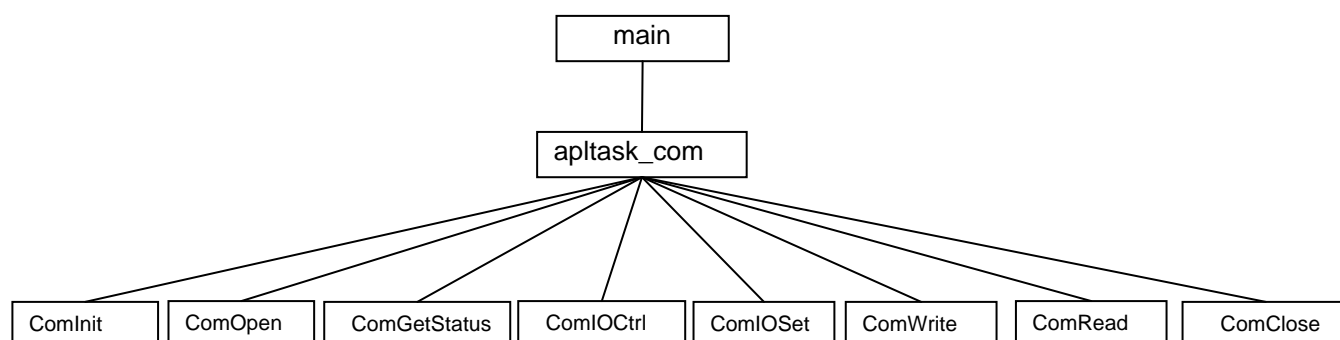


図 18 アプリケーションの関数相関関係

## 10.2 関数説明

関数名と処理内容を示す。

関数名	内容
apltask_com	main 関数から呼ばれ、アプリケーション処理を実行する。
ComInit	USB デバイスから初期通信状態を取得する。
ComOpen	開始フラグを設定する。
ComClose	開始フラグを解除する。
ComGetStatus	USB デバイスから通信パラメータを取得する。
ComIOCtrl	USB デバイスへ通信パラメータを設定する。
ComIOSet	USB デバイスへキャリア制御信号を送信する。
ComWrite	書き込みバッファのポインタ、書き込み要求バイト数を指定し書き込みを実行する。
ComRead	読み出しバッファのポインタ、読み出し要求バイト数を指定し読み出しを実行する。

## 10.3 関数インタフェース

各関数のインタフェースを示す。

関数名	apltask_com
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	apltask_com が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	ComInit
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	ComInit が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	ComOpen
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	ComOpen が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	ComClose
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	無し

関数名	ComGetStatus
パラメータ	param
パラメータの内容	通信パラメータ構造体(転送速度、ストップビット、パリティ、データビット)
パラメータの範囲	転送速度 : 115200 (115200bps) 、 57600 (57600bps) 、 38400 (38400bps) 、 14400 (14400bps) 、 9600 (9600bps) 、 4800 (4800bps) ストップビット : 0 (1ビット) 、 1 (1.5ビット) 、 2 (2ビット) パリティ : 0 (None) 、 1 (Odd) 、 2 (Even) 、 3 (Mark) 、 4 (Space) データビット : 5 (5ビット) 、 6 (6ビット) 、 7 (7ビット) 、 8 (8ビット) 、 16 (16ビット)
戻り値	STATUS
戻り値の内容	ComGetStatus が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0以外 →異常終了

関数名	ComIOCtrl
パラメータ	param
パラメータの内容	通信パラメータ構造体(転送速度、ストップビット、パリティ、データビット)
パラメータの範囲	転送速度 : 115200 (115200bps) 、 57600 (57600bps) 、 38400 (38400bps) 、 14400 (14400bps) 、 9600 (9600bps) 、 4800 (4800bps) ストップビット : 0 (1ビット) 、 1 (1.5ビット) 、 2 (2ビット) パリティ : 0 (None) 、 1 (Odd) 、 2 (Even) 、 3 (Mark) 、 4 (Space) データビット : 5 (5ビット) 、 6 (6ビット) 、 7 (7ビット) 、 8 (8ビット) 、 16 (16ビット)
戻り値	STATUS
戻り値の内容	ComIOCtrl が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0以外 →異常終了

関数名	ComIOSet
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	ComIOSet が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0以外 →異常終了

関数名	ComWrite	
パラメータ	buffer	CntByte
パラメータの内容	送信データバッファ	送信要求バイト数
パラメータの範囲	NULL 以外	1 ~ 64 (バイト)
戻り値	書き込み完了バイト数	
戻り値の内容	ComWrite の正常終了と読み出し完了バイト数の情報	
戻り値の範囲	1 以上 書き込み完了バイト数 -1 以下 →異常終了	

関数名	ComRead	
パラメータ	buffer	CntByte
パラメータの内容	受信データバッファ	受信要求バイト数
パラメータの範囲	NULL 以外	1 ~ 64 (バイト)
戻り値	読み出し完了バイト数	
戻り値の内容	ComRead の正常終了と読み出し完了バイト数の情報	
戻り値の範囲	1 以上 読み出し完了バイト数 -1 以下 →異常終了	

#### 10.4 アプリケーション動作

アプリケーションは下記の動作をする。また、動作フローを図 19 に示す。

1. USB デバイスへのアクセスを開始する。(ComOpen 実行)
  2. 通信パラメータを設定する。(ComIOCtrl 実行)
  3. 2. で設定した通信パラメータを取得し、確認する。(ComGetStatus 実行)
  4. キャリア制御信号を送信する。(ComIOSet 実行)
  5. 再度、通信パラメータを設定する。(ComIOCtrl 実行)
  6. 5. で設定した通信パラメータを取得し、確認する。(ComGetStatus 実行)
  7. データを書き込む。(ComWrite 実行)
  8. データ書き込み後の通信パラメータを取得し、確認する。(ComGetStatus 実行)
  9. データを読み出す。(ComRead 実行)
  10. アクセスを終了する。(ComClose 実行)
1. から 10. の動作を、データサイズを 1~64 バイト、通信パラメータを表 1 に示す範囲で変更して繰り返す。

表 1 通信パラメータ

通信パラメータ	値 (内容)
転送速度	115200 (115200bps) 、 57600 (57600bps) 、 38400 (38400bps) 、 14400 (14400bps) 、 9600 (9600bps) 、 4800 (4800bps)
ストップビット	0 (1 ビット) 、 1 (1.5 ビット) 、 2 (2 ビット)
パリティ	0 (None) 、 1 (Odd) 、 2 (Even) 、 3 (Mark) 、 4 (Space)
データビット	5 (5 ビット) 、 6 (6 ビット) 、 7 (7 ビット) 、 8 (8 ビット) 、 16 (16 ビット)



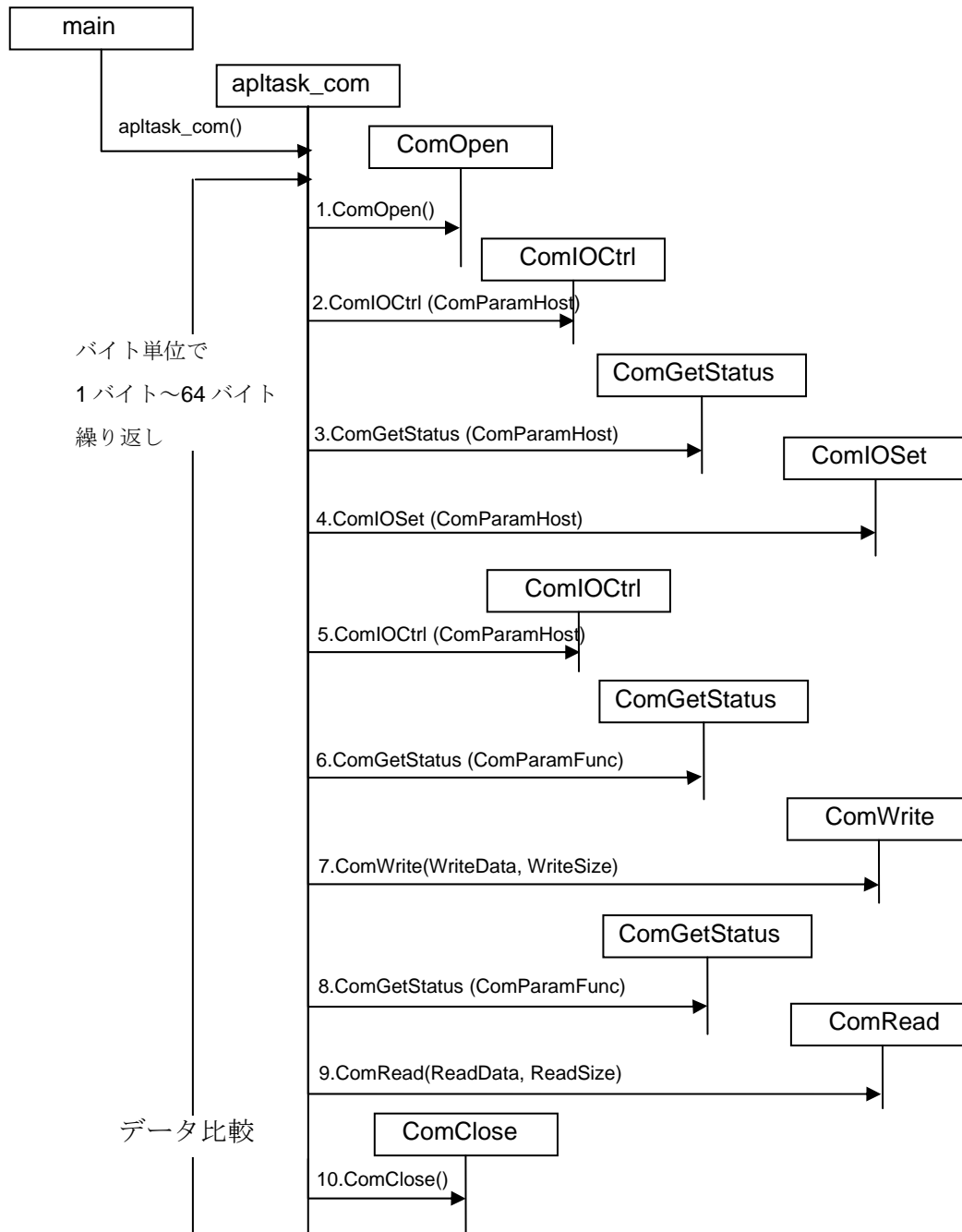


図 19 アプリケーション動作

## 11. ハブクラスドライバ

ハブクラスドライバの関数仕様を示す。

### 11.1 関数説明

関数名と処理内容を示す。

関数名	内容
HubClassInit	デバイス接続済みのダウンストリームポートを全て Powered-off 状態から Disconnected 状態に遷移させる。
WaitHubChange	インタラプトイン転送によりハブの状態変化を待つ。
ResetHubPort	デバイス接続済みのダウンストリームポートの1個をリセットし、インタラプトイン転送によりリセットの完了を待つ。
ReadHubStatus	インタラプトイン転送で受信したデータを格納する。

## 11.2 関数インタフェース

各関数のインタフェースを示す。

関数名	HubClassInit
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	HubClassInit が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

関数名	WaitHubChange	
パラメータ	UsbDevInfo	port
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	ポート番号
パラメータの範囲	NULL 以外	0 ~ 6 (最大ポート数:7)
戻り値	STATUS	
戻り値の内容	WaitHubChange が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	ResetHubPort	
パラメータ	UsbDevInfo	port
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)	ポート番号
パラメータの範囲	NULL 以外	0 ~ 6 (最大ポート数:7)
戻り値	STATUS	
戻り値の内容	ResetHubPort が正常終了したかを判断する	
戻り値の範囲	0 →正常終了 0 以外 →異常終了	

関数名	ReadHubStatus
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ReadHubStatus が正常終了したかを判断する
戻り値の範囲	0 →正常終了 0 以外 →異常終了

## 12. 階層構造

USB ホストソフトウェアの階層構造を示す。

### 12.1 階層構造図

階層構造を図 20に示す。

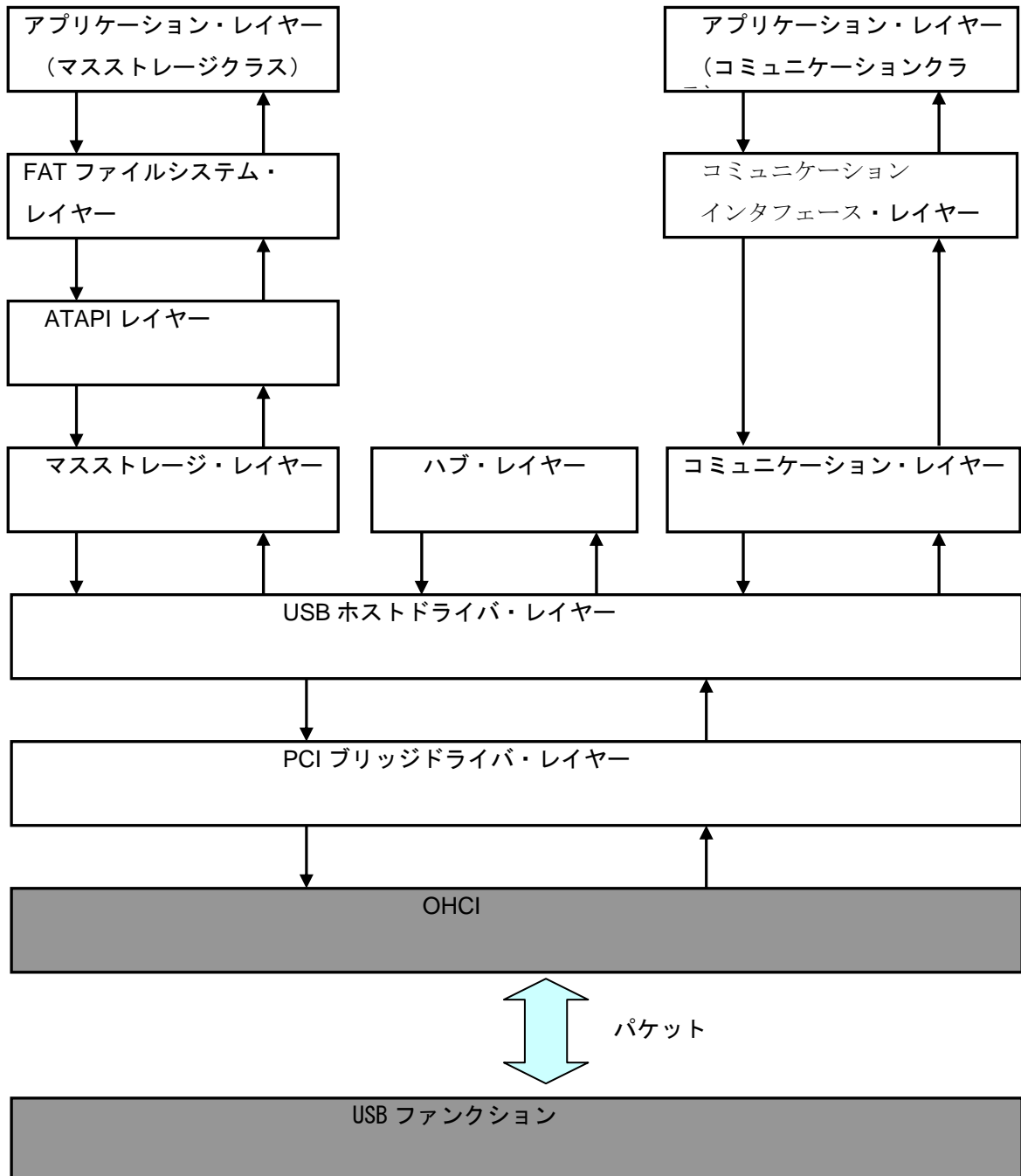


図 20 階層図

## 12.2 階層説明

階層毎の関数名と処理内容を示す。

階層	主な関数名	説明
アプリケーション	apltask_msc apltask_com	アプリケーション・レイヤーは、下位レイヤーにデータを送信、受信する。
FAT ファイルシステム	cfs_mountfs cfs_open cfs_write cfs_read cfs_close cfs_umountfs	アプリケーション・レイヤーからの要求でファイルのオープン、読み出し、書き込み等の操作を実行する。ブロック番号、受信バッファのアドレス、送信バッファアドレスを下位レイヤーに渡す。
コミュニケーション インタフェース	ComInit ComOpen ComClose ComRead ComWrite ComGetStatus ComIOCtrl ComIOSet	アプリケーション・レイヤーからの要求で USB デバイスのオープン、制御、読み出し、書き込み等の操作を実行する。通信パラメータ、受信バッファのアドレス、送信バッファアドレスを下位レイヤーに渡す。
ATAPI	ATACommandExecute AtaReadFormatCapacities AtaModeSense6 AtaInquiry AtaRead10 AtaWrite10 AtaReadCapacitys AtaRequestSence	ファイルシステムからの Read、Write コマンドの要求を ATAPI コマンドに変換し、下位レイヤーに渡す。
マスタストレージ	MassStorageClassExecute CBWStageExecute DataInStageExecute DataOutStageExecute CSWStageExecute	ATAPI レイヤーからの ATAPI コマンド、書き込み、読み出し用アドレスを受け取り、CBW、データ、CSW のバルク転送を実行する。
コミュニケーション	CommunicationClassExecute COMReadExecute COMWriteExecute GetLineCodingExecute SetLineCodingExecute SetControlLineStateExecute	コミュニケーションインタフェースレイヤーからの USB デバイス制御要求を受け取り、コミュニケーションクラスコマンドを実行する。また、書き込み、読み出し要求を受け取りバルク転送を実行する。

階層	主な関数名	説明
ハブ	HubClassInit GetHubStatusExecute SetPortFeatureExecute ClearPortFeatureExecute GetPortStatusExecute WaitHubChange RsetHubPort ReadHubStatus	ハブの制御でハブクラスコマンドを実行する。また、インタラプトイン転送で受信したデータを格納する。
USB ホストドライバ	ControlProcess DescriptorInitGet AddressSet DescriptorGet DescriptorConfigGet DescriptorConfigAllGet GetHubDescriptor DeviceStatusGet ConfigurationSet GetHubStatus SetPortFeature ClearPortFeature GetPortStatus MaxLUNGet SetLineCoding GetLineCoding SetControlLineState BulkTransExecute BulkOutProcess BulkInProcess UsbhdInit ActiveHubPort SetInterruptInProcess	コントロール転送を使用し、USB デバイス接続時のエニュメレーションやクラスコマンドを実行する。 マスタストレージレイヤーやコミュニケーションクラスレイヤーからのデータ送受信要求に従い、バルク転送を実行する。 ハブデバイス接続時にはインタラプトイン転送を実行する。 PCI ブリッジドライバを使用し、ホストドライバの初期化を行う。
PCI ブリッジドライバ	PciInit PciConfigRead PciConfigWrite	PCI ブリッジを介して OHCI ホストブリッジにアクセスする。
メモリモジュール	UsbhdMemInit EDInit GTDInit UsbMemoyInit EdDescriptorCreate GtdDescriptorCreate AllocateED AllocateGTD EdDescriptorFree GtdDescriptorFree	ED、TD 用メモリ領域を確保する。

### 13. 割り込み

割り込み発生時の関数相関図、仕様、動作を示す。

#### 13.1 アプリケーション相関図

割り込み発生時の関数相関図を示す。

USB デバイス接続され、RootHubStatusChange イベント発生した場合（ルートハブがポートの状態変化を検出）、または、TD 処理が完了し WritebackDoneHead イベントが発生した場合、割り込みにより CPU に通知される。

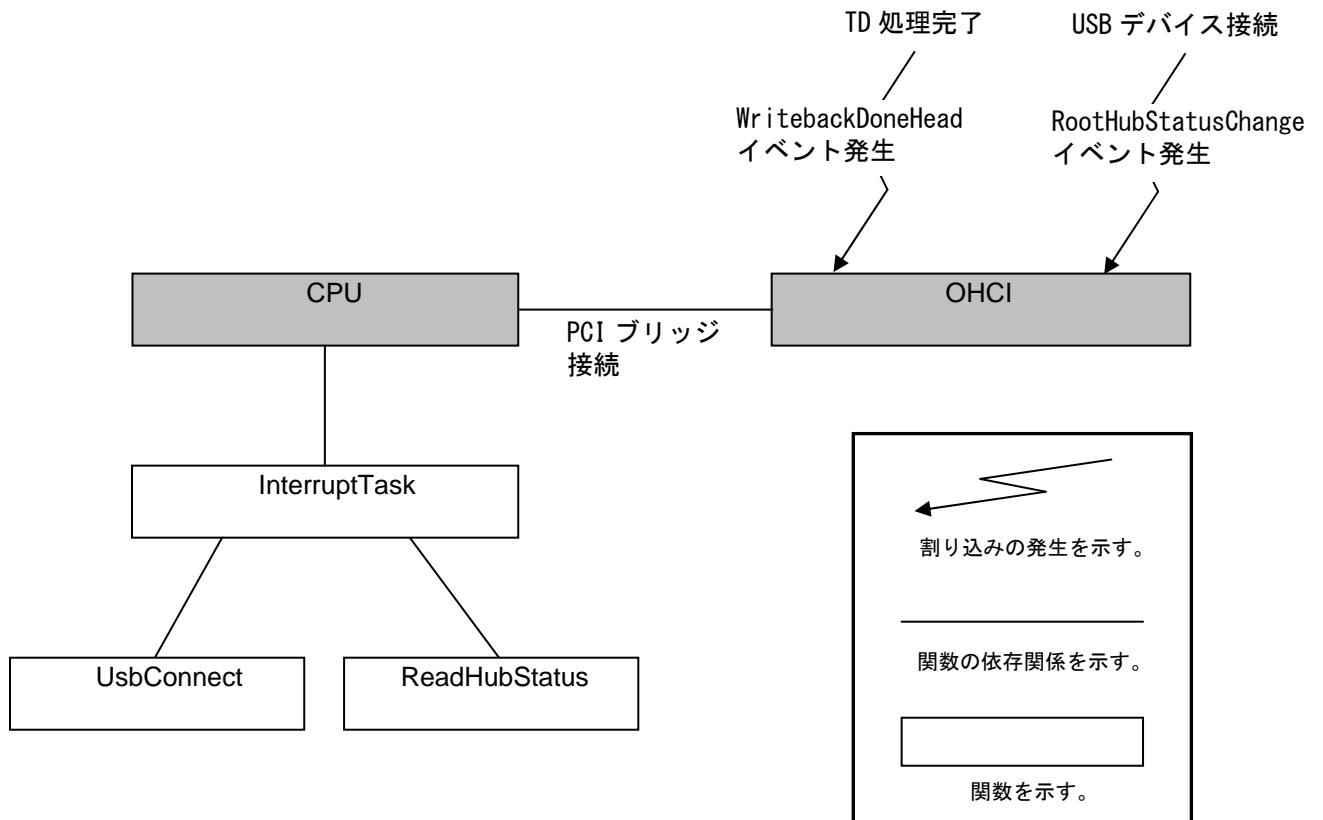


図 21 割り込み発生時の関数相関関係

## 13.2 関数説明

関数名と処理内容を示す。

関数名	内容
InterruptTask	CPU からの INTUSBH0 割り込みで呼び出される。 OHCI の割り込みレジスタ(HcInterruptStatus レジスタ)を読み出す。 USB デバイスが接続された場合(RootHubStatusChange ビットが 1'b の場合)、UsbConnect 関数を実行する。 TD 処理が完了した場合(WritebackDoneHead ビットが 1'b の場合)、OHCI の HccaDoneHead レジスタを読み出し完了した TD の内容を確認する。 完了した TD がインタラプトイン転送の TD であれば、ReadHubStatus 関数を実行する。完了した TD がインタラプトイン転送以外の TD であれば write_back_done フラグを 1'b に設定する。 なお、割り込みを CPU への通知には RootHubStatusChange ビットと WritebackDoneHead ビットの OHCI、PCI、CPU での割り込みマスクの解除が必要です。
UsbConnect	OHCI のポート状態レジスタ(HcRhPortStatus レジスタ)を読み出し、ルートハブのポートに USB デバイスが接続されていることを確認する。
ReadHubStatus	インタラプトイン転送で受信したデータを格納する。

## 13.3 関数インタフェース

関数のインタフェースを示す。

関数名	InterruptTask
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	無し
戻り値の内容	無し
戻り値の範囲	無し

関数名	UsbConnect
パラメータ	void
パラメータの内容	無し
パラメータの範囲	無し
戻り値	STATUS
戻り値の内容	UsbConnect が正常終了したかを判断する
戻り値の範囲	0 → 正常終了

関数名	ReadHubStatus
パラメータ	UsbDevInfo
パラメータの内容	ドライバ構造体(USB デバイス情報、転送情報)
パラメータの範囲	NULL 以外
戻り値	STATUS
戻り値の内容	ReadHubStatus が正常終了したかを判断する
戻り値の範囲	0 → 正常終了 0 以外 → 異常終了



### 13.4 割り込み対応動作

割り込み発生時の動作を示す。USB デバイスが接続されると OHCI のルートハブがポートの状態変化を検出、または、TD 処理が完了し WritebackDoneHead イベントが発生したことを CPU に通知する。CPU は INTUSBH0 割り込みベクタに登録されている InterruptTask を実行する。

OHCI の割り込みレジスタを読み出し、ポートの状態が変化を確認できた場合、UsbConnect 関数を実行する。

OHCI の割り込みレジスタを読み出し、WritebackDoneHead イベントが発生したことを確認できた場合、さらに、OHCI の HccaDoneHead レジスタを読み出し完了した TD の内容を確認する。完了した TD がインタラプトイン転送の TD であれば、ReadHubStatus 関数を実行する。また、完了した TD がインタラプトイン転送以外の TD であれば write\_back\_done フラグを 1'b に設定する。

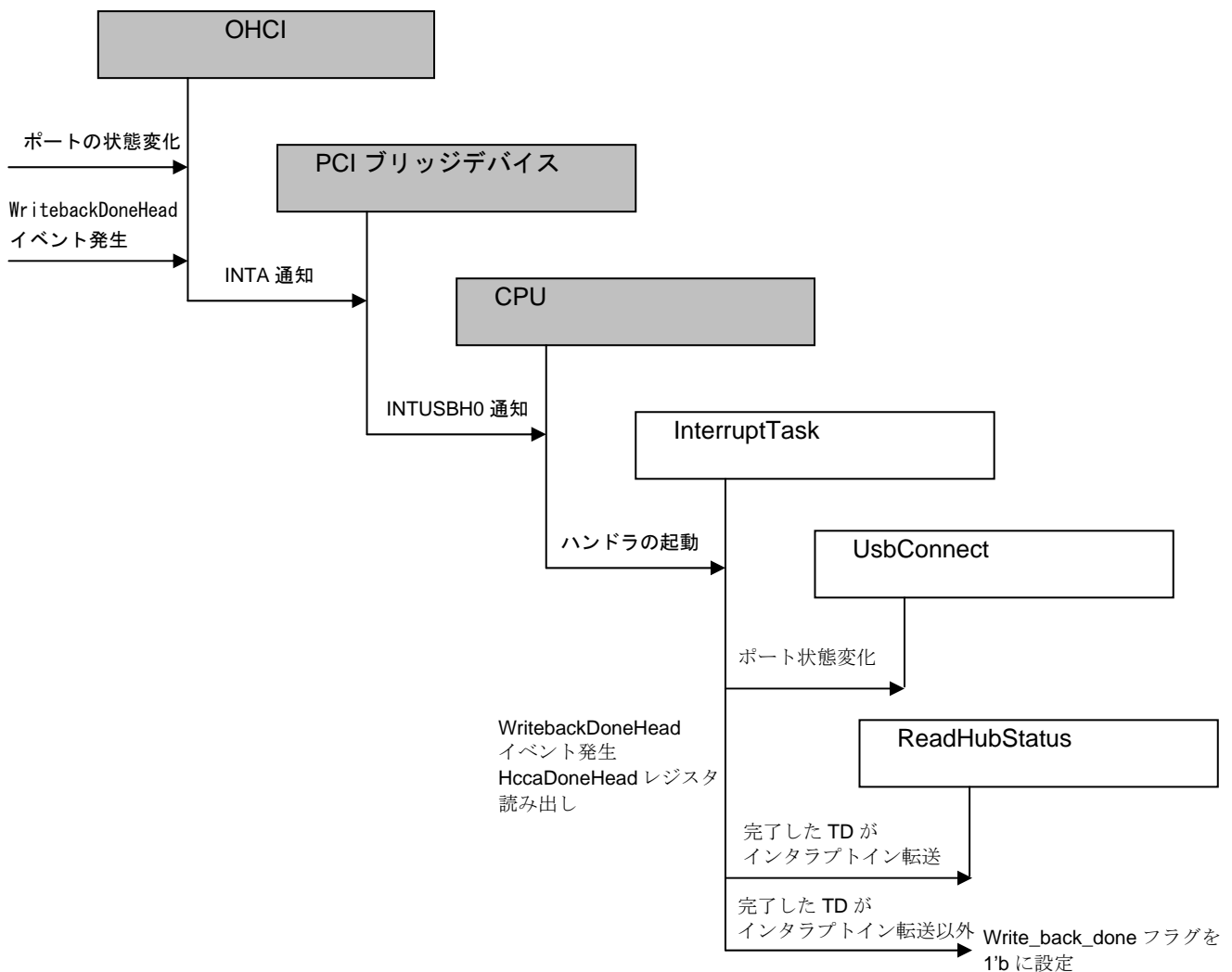


図 22 割り込み対応動作

## 14. データ構造

USB ホストドライバで使用するデータ構造を示す。

構造体名	内容
USBHD_DEV_INFO	エnumレーション終了フラグ ポート番号 USB デバイスアドレス 新規に接続された USB デバイスに割り当てるデバイスアドレス エndポイント0の最大パケットサイズ Get Configuration Descriptor で返されるディスクリプタの全サイズ インタフェース数 コンフィギュレーション値 インタフェースのエndポイント数(インタフェース : 0~2) インタフェース・クラス(インタフェース : 0~2) インタフェース・サブクラス(インタフェース : 0~2) インタフェース・プロトコル(インタフェース : 0~2) インタラプト転送用エndポイント番号(インタフェース : 0~2) インタラプト転送用エndポイントの属性(インタフェース : 0~2) インタラプト転送用エndポイントの最大パケットサイズ(インタフェース : 0~2) インタラプト転送用エndポイントの転送間隔(インタフェース : 0~2) バルクイン転送用エndポイント番号(インタフェース : 0~2) バルクイン転送用エndポイントの属性(インタフェース : 0~2) バルクイン転送用エndポイントの最大パケットサイズ(インタフェース : 0~2) バルクアウト転送用エndポイント番号(インタフェース : 0~2) バルクアウト転送用エndポイントの属性(インタフェース : 0~2) バルクアウト転送用エndポイントの最大パケットサイズ(インタフェース : 0~2) HCCA ベースアドレス デバイスステータス マスストレージクラスのインタフェース番号 コミュニケーションクラスクラスのインタフェース番号 データインタフェースクラスのインタフェース番号 ハブクラスのインタフェース番号 MaxLUN 値 CBW タグ コミュニケーションクラスデバイスの開始フラグ ハブのダウンストリーム数 インタラプトイン転送で取得するハブの状態変化ビットマップデータ ハブステータス ポートステータス ポート状態変化ステータス デバイス・リクエスト用アドレス コントロール転送用受信データアドレス コントロール転送用受信データサイズ コントロール転送用送信データアドレス コントロール転送用送信データサイズ バルク転送用データアドレス バルクアウト転送用データサイズ バルクイン転送用データサイズ ディスクリプタ(ED, TD)情報 CBW データアドレス マスストレージ用データ格納アドレス CSW データ格納アドレス

構造体名	内容
	コミュニケーションクラス用データ格納アドレス コミュニケーションクラス用パラメータ格納アドレス ハブクラス用パラメータ格納アドレス 受信データ用バッファアドレス 送信データ用バッファアドレス データサイズ FileInfo 構造体へのポインタ
USBHD_DESC_INFO	コントロール転送用 ED アドレス セットアップデータ用 TD アドレス データステージ用 TD アドレス ステータスステージ用 TD アドレス コントロール転送用ダミーTD アドレス バルクアウト用 ED アドレス(インタフェース : 0~2) バルクイン用 ED アドレス(インタフェース : 0~2) バルクアウト用 TD アドレス(インタフェース : 0~2) バルクイン用 TD アドレス(インタフェース : 0~2) バルクアウト用ダミーTD アドレス(インタフェース : 0~2) バルクイン用ダミーTD アドレス(0~2) インタラプトイン用 ED アドレス(0~31) インタラプトイン用 TD アドレス インタラプトイン用ダミーTD アドレス
FILEINFO	論理ブロック番号 データサイズ データ数 ATAPI コマンド セクタサイズ セクタ数 書き込み用データポインタ 読み出し用データポインタ データポインタ ファイル名 モード
LINE_CODING_STRUCT	転送速度(bps) ストップビット パリティ データビット
ED	データ詳細は表 2、表 3を参照。
TD	データ詳細は表 4、表 5を参照。

表 2 エンドポイントディスクリプタ(ED)フォーマット概要

	31 27	26 16	15	14	13	12 11	10 7	6 4	3 2	1	0
Word0	—	MPS	F	K	S	D	EN	FA			
Word1	TD Queue Tail Pointer(TailP)								—		
Word2	TD Queue Head Pointer(HeadP)								0	C	H
Word3	Next Endpoint Descriptor(NextED)								—		

表 3 エンドポイントディスクリプタ(ED)フォーマット詳細

名前	ホスト・コントローラからの R/W	定義
MPS	R	対象となるファンクション側エンドポイントが 1 つの USB パケットで受信または送信できる最大 byte 値を示す
F	R	コントロール/バルク/インタラプト転送→1 アイソクロナス転送→0
K	R	このビットをセットすることで、このエンドポイントに関する通信を行わずに次の ED へと制御を移すことができる
S	R	フルスピード→0 ロースピード→1
D	R	通信方向 00 : TD のフィールドで設定 01 : OUT 10 : IN
EN	R	通信対象となるエンドポイント番号
FA	R	通信対象となる USB デバイスアドレス
C	R/W	TD がリタイアする際、最後のトグル情報がセットされる
H	R/W	HC がこの ED の処理を停止させたいときにセットされる 通常 TD の処理がエラーだった場合セットされる
TailP	R	この ED にリンクされている最後の TD のアドレスが格納されている TailP と HeadP が同一の場合、この ED には処理すべき TD が存在しない
HeadP	R/W	この ED にリンクされている最初の TD このフィールドが示す TD の処理が完了するたびに次の TD のアドレスに自動更新される
NextED	R	この ED にリンクされている ED へのアドレス リンクする ED がない場合 0 をセットする

表 4 トランスファーディスクリプタ(TD)フォーマット概要

	31 28	27 26	25 24	23 21	20 19	18	17 4	3 0
Word0	CC	EC	T	DI	DP	R	—	
Word1	Current Buffer Pointer(CBP)							
Word0	Next TD(NextTD)							
Word2	Buffer End(BE)							

表 5 トランスファーディスクリプタ(TD)フォーマット詳細

名前	ホスト・コントローラからの R/W	定義
CC	R/W	この TD から生成された最後の転送ステータスを示す
EC	R/W	転送エラーが発生するたびにこの値がインクリメントされる この値が 2 で転送エラーが発生した場合、エラーの種類を CC に記録し、TD を Done キューに移動する
T	R/W	データパケットの PID(DATA0/DATA1)の生成と比較に使用する。 送信/受信が成功するたびに更新する
DI	R	TD の処理を完了した際に発生する割り込みのタイミングを規定する 0 : 処理完了後、直ちに発生 1 : TD の処理が完了したフレームの後に発生 111 : 割り込みは発生しない
DP	R	転送方向と PID を設定する 00 : SETUP エンドポイントへ 01 : OUT エンドポイントへ 10 : IN エンドポイントから
R	R	この TD から生成される最後のパケットがショートパケットになるとき、エラーが発生する 1 でエラーを無視する
CBP	R/W	エンドポイントと転送するためのバッファ領域を指す 常に次にアクセスすべきバッファ領域のアドレスを示す 0 の場合はサイズ 0 のデータの転送、もしくは完了した場合
NextTD	R/W	次の TD へのポインタ
BE	R	バッファ領域の末尾のアドレス

## 15. 参考ドキュメント

- ハードウェアマニュアル  
V850E2/ML4 ユーザーズマニュアルハードウェアマニュアル [R01UH0262JJ]  
(最新版はルネサスエレクトロニクスのホームページから入手してください)
- ソフトウェアマニュアル  
V850E2M ユーザーズマニュアルアーキテクチャ編 [R01US0001JJ]  
(最新版はルネサスエレクトロニクスのホームページから入手してください)
- Universal Serial Bus Revision 2.0 specification
- OpenHCI Open Host Controller Interface specification for USB Release 1.0a
- Universal Serial Bus Mass Storage Class Bulk-Only Transport Revision 1.0
- Universal Serial Bus Class Definitions for Communications Devices Revision 1.2
- Universal Serial Bus Class Subclass Specification for PSTN Devices s Revision 1.2

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.06.22	—	初版発行
1.01	2012.09.14	—	M3S-TFAT-Tiny (ルネサス製 FAT ファイルシステム) を搭載
1.02	2013.01.10	—	ハブクラス対応 (複数デバイス制御)



## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認ください。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更することがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>