



MICROCHIP

注意：この日本語版文書は参考資料としてご利用ください。最新情報は必ずオリジナルの英語版をご参照願います。

# AN1142

## 組み込み USB ホストにおけるマストレージクラス

Author: Kim Otten  
Microchip Technology Inc.

### はじめに

Microchip 社の USB OTG モジュール内蔵マイクロコントローラを使うと、マイクロコントローラ アプリケーションで USB 組み込みホスト機能を容易にサポートできます。この機能の代表的な用途の 1 つは、USB フラッシュ ドライブやメモ리카ード リーダ等のマストレージ デバイスへの接続です。これらのデバイスは USB マストレージ クラスを使います。

### USB マストレージ クラス

#### 概要

USB がサポートする 4 種類の転送タイプの内、大きなサイズ of データ転送に最も適しているのがバルク転送です。バルク転送はコントロール、インタラプト、アイソクロナス転送の完了後にフレームの残りの帯域幅の全てを使う事で、USB の帯域幅を効率的に使います。バルク転送のフレームあたりの転送バイト数は可変です。また、バルク転送はデータの精度を確保するためにエラーチェック機能を備えています。バルク転送に使える正確な時間は、バス上の他のトラフィックの量によって決まります。各種の他の転送も実行する必要がある場合、フレーム内でバルク転送に使える帯域幅はほとんど残らない事もあります。従って、バルク転送はタイミングが重要ではない動作向けにのみ使います。

マストレージ デバイス向けのクラス / サブクラス / プロトコル識別子は、デバイス ディスクリプタの bDeviceClass、bDeviceSubClass、bDeviceProtocol フィールドに格納されません。これらのフィールドは全て 0x00 に設定され、代わりにインターフェイス ディスクリプタの bInterfaceClass、bInterfaceSubClass、bInterfaceProtocol フィールドで識別子を指定します。USB マストレージ デバイスの代表的な設定は以下の通りです。

- bInterfaceClass - 0x08 (マストレージ クラス)
- bInterfaceSubClass - 0x06 (SCSI Primary Command-2 (SPC-2))
- bInterfaceProtocol - 0x50 (バルク転送専用)

マストレージ デバイスは複数の論理ユニットを持つ事ができ、各ユニットは論理ユニット番号 (LUN) で表します。デバイス上の全ての論理ユニットは同じデバイス特性を有しますが、LUN を使って別々にアドレ

ス指定できます。LUN の値は 0 ~ 15 です。デバイスが 1 つの論理ユニットしかサポートしない場合、LUN は 0 です。

上記の代表的なコンフィグレーションのマストレージクラスはバルク転送のみをサポートします。このプロトコルでは、以下の 3 つのエンドポイントを使います。

- エンドポイント 0 (コントロール転送用)
- 1 つのバルク IN エンドポイント
- 1 つのバルク OUT エンドポイント

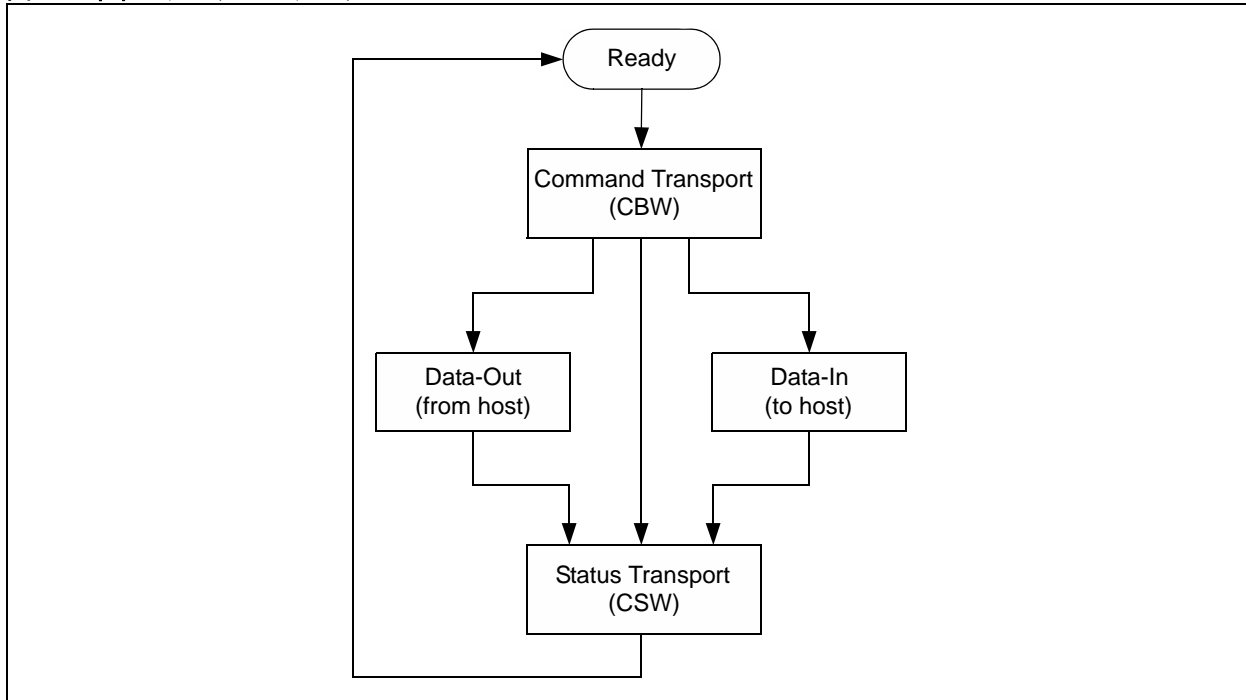
バルク転送は以下の 3 つのステージで構成されます。

- 命令
- データ (必須ではない)
- ステータス

命令ステージは、バルク OUT エンドポイントを介してホストからペリフェラルへ送信します。データステージが存在する場合、データをペリフェラルからホストへ転送する場合はバルク IN エンドポイントを使い、ホストからペリフェラルへ転送する場合はバルク OUT エンドポイントを使います。ステータスステージでは、ホストはバルク IN エンドポイントを介してペリフェラルから転送に関するステータス情報を受信します。ステージのフローを図 1 に示します。

# AN1142

図 1: 命令 / データ / ステータスのフロー



## CBW (COMMAND BLOCK WRAPPER)

CBW (Command Block Wrapper)は、転送の命令ステージ中にペリフェラルへ送信されます。CBW は以下の情報を含む 31 バイトのパケットです。

- 転送を識別するためのタグ
- データステージ中に転送するバイト数
- 転送に使う論理ユニットの LUN
- デバイスが実行する命令ブロック

CBW のフォーマットを表 1 に示します。

表 1: CBW (COMMAND BLOCK WRAPPER)

CBW (Command Block Wrapper)								
バイト	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 ~ 3	dCBWSignature							
4 ~ 7	dCBWTag							
8 ~ 11	dCBWDataTransferLength							
12	bmCBWFlags							
13	Reserved (0)				bCBWLUN			
14	Reserved (0)			BCBMCBLength				
15 ~ 30	CBWCB							

CBW はマスタストレージ クライアント ドライバによって内部で生成されます。

## CSW (COMMAND STATUS WRAPPER)

CSW (Command Status Wrapper) はペリフェラルからホストへ送信されます。CSW は以下の情報を含む 13 バイトのパケットです。

- 転送を識別するためのタグ (CBW 内のタグと一致する必要がある)
- 期待されたデータバイト数と実際に転送されたバイト数の差
- 命令の成功または失敗

CSW のフォーマットを表 2 に示します。

表 2: CSW (COMMAND STATUS WRAPPER)

CSW (Command Status Wrapper)								
バイト	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0-3	dCBWSignature							
4 ~ 7	dCSWTag							
8 ~ 11	dCSWDataResidue							
12	bCSWStatus							

CSW はマスタストレージ クライアント ドライバによって内部で受信およびチェックされます。

## マスタストレージクライアントドライバの使用

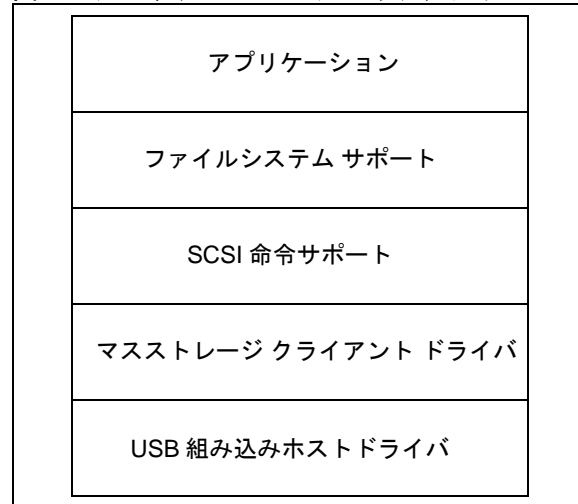
### マスタストレージクライアントドライバのインストール

マスタストレージクライアントドライバは、USB 組み込みホスト サポート パッケージ (Microchip 社ウェブサイト (<http://www.microchip.com/usb>) から入手可能) の一部としてインストールされます。インストールの詳細はアプリケーションノート『AN1140 - USB 組み込みホストスタック』を参照してください。

### アプリケーションアーキテクチャ

ほとんどのアプリケーションは、USB ホスト マスタストレージクライアントドライバと直接連携しません。代わりに、メディア インターフェイス層を使います。このレイヤがクライアントドライバと連携し、アプリケーションに代わってホストスタックドライバを使います。例えば、『AN1145 - 組み込みホストにおける USB フラッシュドライブの使用』に記載したアプリケーションは、図 2 に示すように、アプリケーション層を含む 5 つのレイヤを使います。

図 2: アプリケーションアーキテクチャ



「SCSI 命令サポート」層はメディア インターフェイス層であり、ファイルシステム命令を SCSI 命令に変換した後に USB マスタストレージクラスを使う USB ペリフェラルへ伝送します。

Note: USB ホスト マスタストレージ クラス ドライバ API に関する詳細は、アプリケーションノート『AN1141 - USB 組み込みホストスタック プログラマガイド』と「Help」ディレクトリ内の API 関連文書を参照してください。

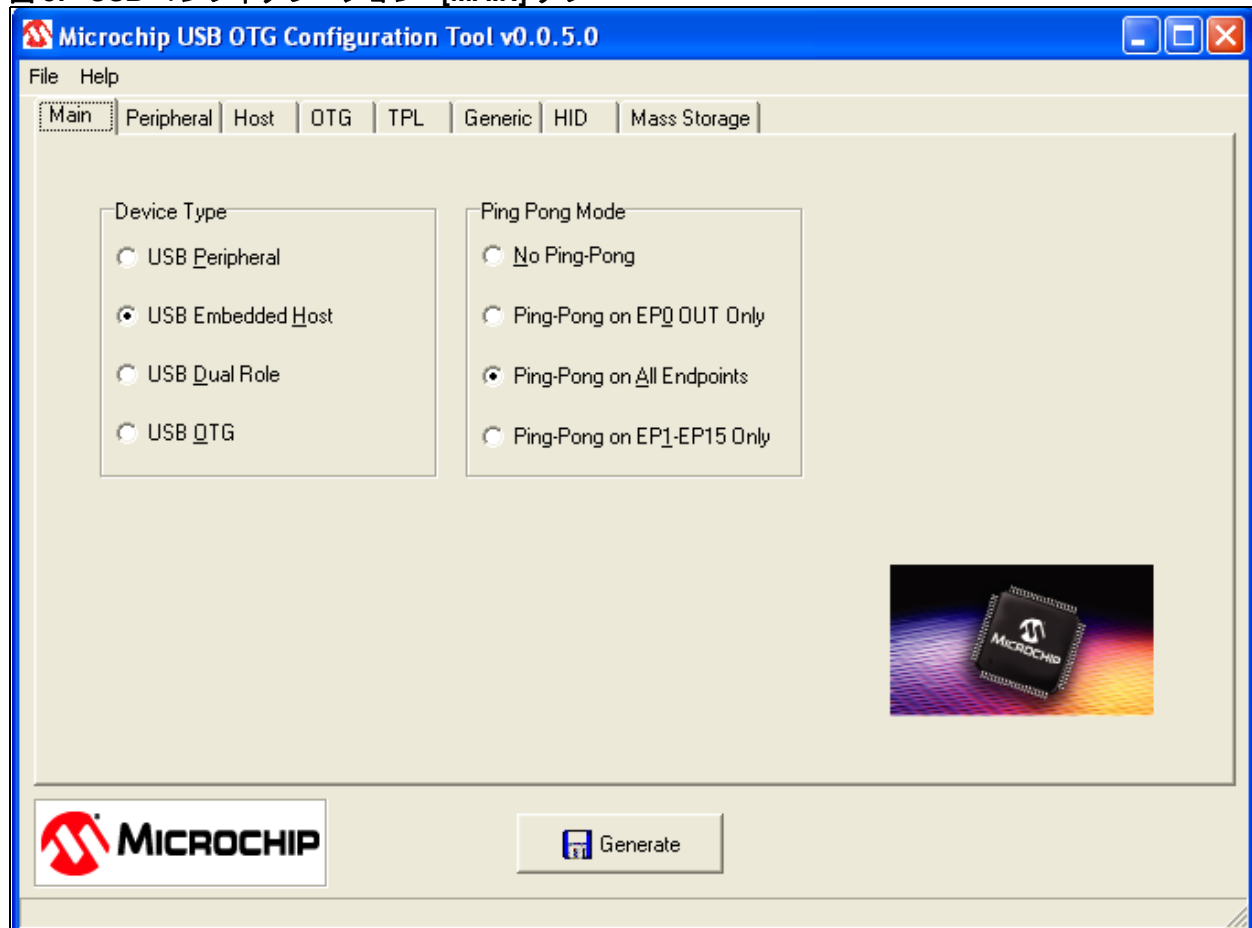
## クラスの設定

### USB コンフィグレーション ツールの使用方法

アプリケーション向けにマスストレージ クライアント ドライバを設定するために、USB コンフィグレーション ツール (USBConfig.exe) を使います。このツールは、インストール ディレクトリ内の「\Microchip\USB」サブディレクトリにインストールされます。

USB 組み込みホスト向けにマスストレージ クライアント ドライバを使うには、[Main] タブ内でアプリケーションの USB デバイスタイプとして「USB Embedded Host」を選択します。

図 3: USB コンフィグレーション - [MAIN] タブ



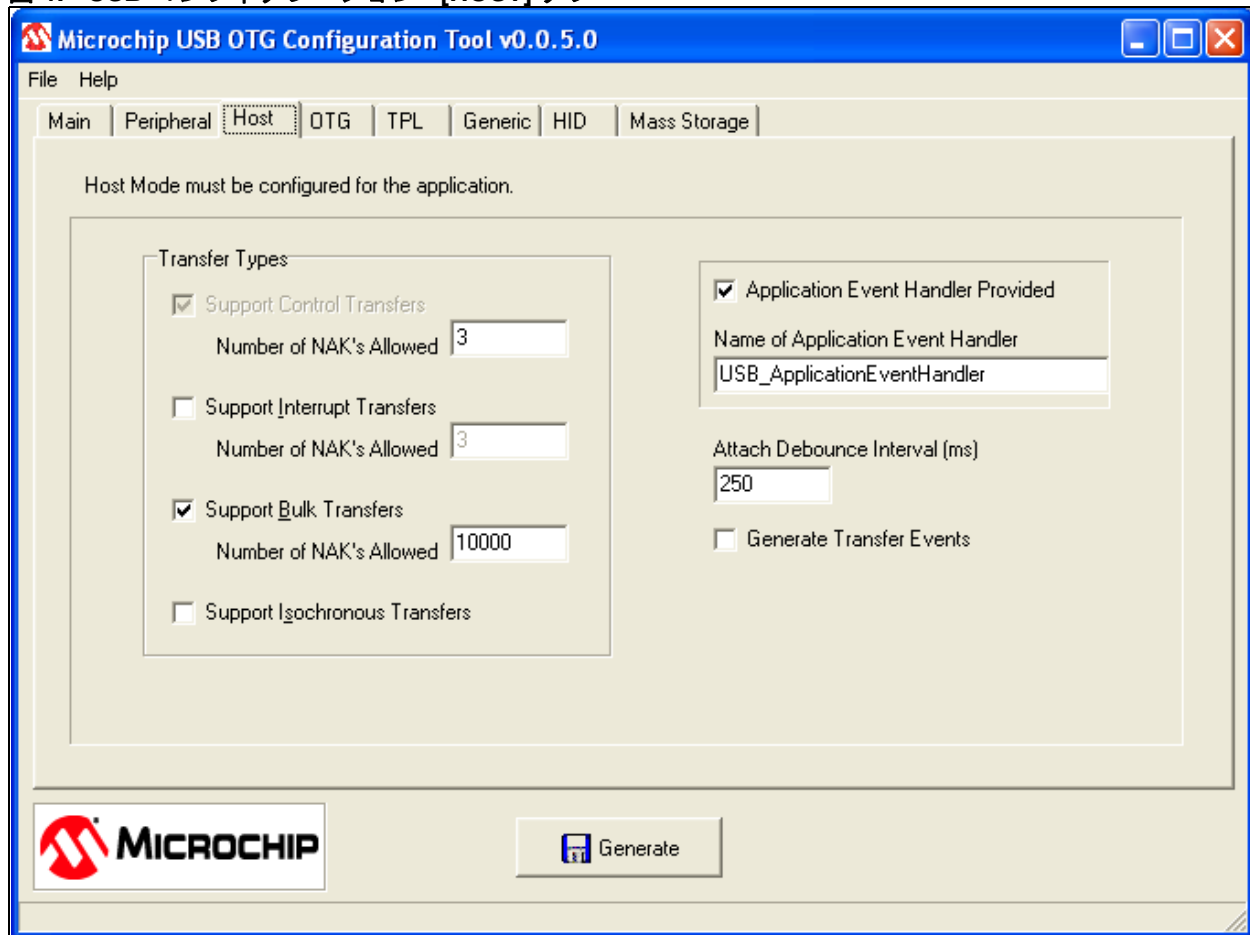
[Host] タブを開いて、図 4 に示すように基本的なホスト動作を設定します。マスストレージ クライアントドライバは、コントロールおよびバルク エンドポイントのサポートを要求します。アプリケーションがインタラプトまたはアイソクロナス エンドポイントを要求するクラスを含まない場合、それらのエンドポイントタイプは無効にできません。

マスストレージ デバイスの応答遅延は、USB の通信フレーム周期 (1 ms) に比べてかなり長くなる場合があります。従って、通信の試行を終了するまでに許容する NAK 応答の数「Number of NAK's Allowed」を大きくする事を推奨します。また、デバイスによっては、電源投入後の初期化に USB 仕様値 (100 ms) よりも長

い時間を要する場合があります。従って、接続デバウンス期間「Attach Debounce Interval (ms)」を大きくする事を推奨します。次に、[Name of Application Event Handler] に、アプリケーション レベルのイベントハンドラとして機能するメイン ソースファイル内の関数の名前を入力します。

USB マスストレージ クライアント ドライバは、USB ホストドライバをポーリングする事で転送ステータスを確認するか、あるいは USB ホストドライバ転送イベントに応答する事ができます。この選択については、後述の「イベントの生成」を参照してください。

図 4: USB コンフィグレーション - [HOST] タブ



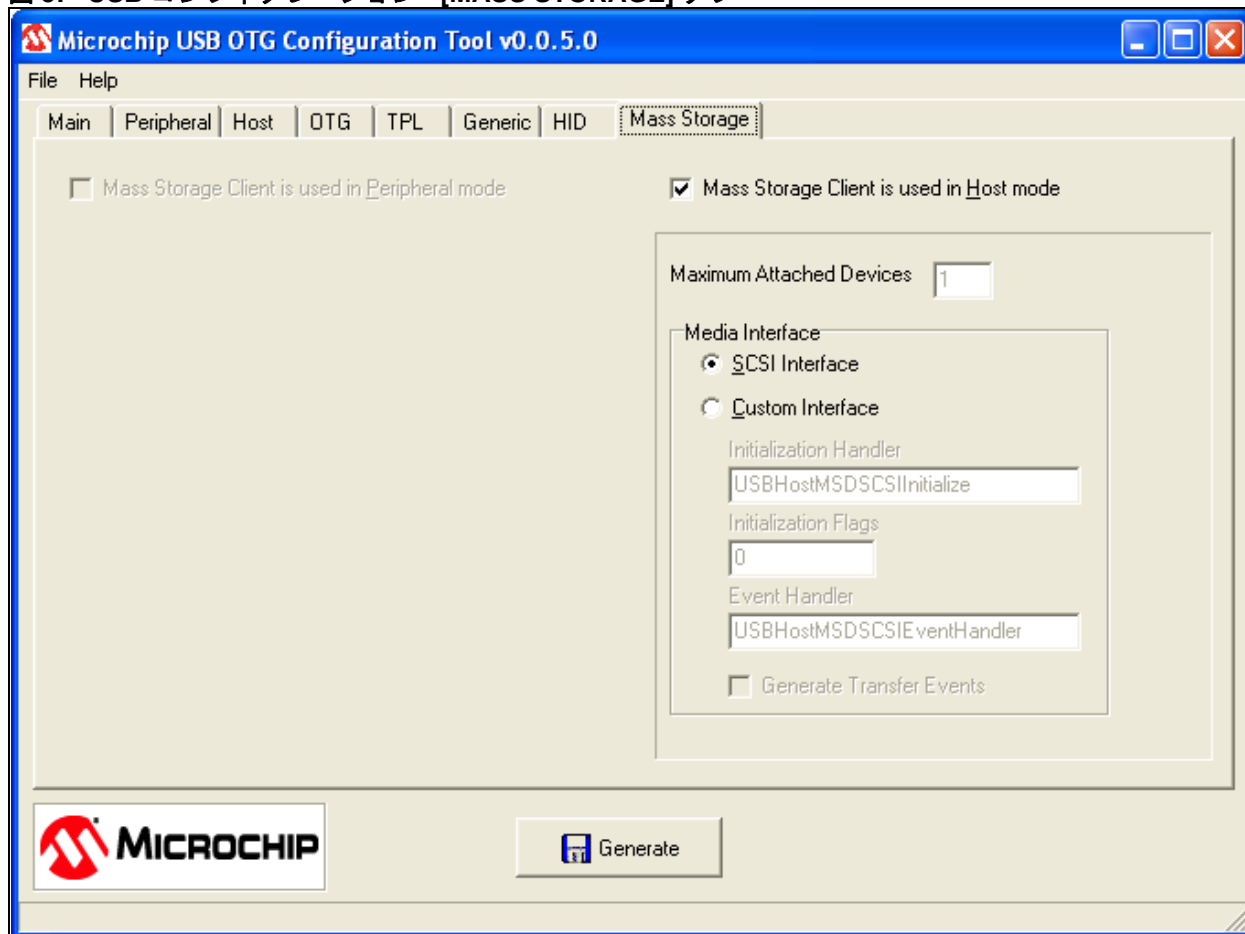
[Mass Storage] タブを開き、図 5 に示すように、[Mass Storage Client is used in Host Mode] チェックボックスにチェックマークを付ける事で、マストレージ組み込みホストに対するサポートを有効にします。

多くのマストレージ デバイスは SCSI インターフェイス プロトコルを使います。このプロトコルのサポートは、マストレージ クライアント ドライバと一緒に提供されます。

このレイヤ内の各関数は次の動作が始まる前に完了する必要があるため、マストレージ転送イベントを使用しません。

[Generate] ボタンをクリックしてコンフィギュレーション ファイル「usb\_config.c and usb\_config.h」を生成し、プロジェクト ディレクトリに保存します。

図 5: USB コンフィギュレーション - [MASS STORAGE] タブ



### インターフェイス関数の定義

クライアント ドライバは、メディア インターフェイス層内で 2 つのインターフェイス関数を必要とします。1 つは初期化ハンドラです。これはペリフェラルのエnumレーションおよび初期化後に、マストレージ クライアント ドライバによって呼び出されます。初期化ハンドラは、以下のように typedef で定義された関数である必要があります。

```
typedef BOOL (*USB_CLIENT_INIT) (BYTE
address, DWORD flags);
```

この関数は、メディア インターフェイスに固有の初期化を実行します。初期化中にエラーが発生しなかった場合、このルーチンは TRUE を返します。エラーに遭遇した場合、このルーチンは FALSE を返し、ペリフェラルへの転送を許容しません。

もう 1 つのインターフェイス関数は、通常動作中に発生するイベントを処理するために必要です。このイベントハンドラは、以下のように typedef で定義された関数である必要があります。

```
typedef BOOL (*USB_CLIENT_EVENT_HANDLER)
(BYTE address, USB_EVENT event, void
*data, DWORD size);
```

発生する可能性のあるイベントの例として、EVENT\_DETACHがあります。このイベントはデバイスがバスから切断された時に発生します。この場合、メディア インターフェイス層は、自身の接続メディア リストからそのデバイスを削除する等の動作によって、自身のステータスを更新する必要があります。

全てのイベントを含むリストについては、「Help」ディレクトリ内の API 関連文書を参照してください。

クライアント ドライバは、メディア インターフェイスに必要なペリフェラル初期化ハンドラとイベントハンドラのリストを要求します。このリストは、スタックと一緒に提供されるコンフィグレーション ツール (USBConfig.exe) を使って定義します。

## イベントの生成

クライアント ドライバの設定により、USB ホスト層から転送イベント (EVENT\_TRANSFER) を使うことができます。同じくクライアント ドライバの設定により、メディア インターフェイス層向けの転送イベント (EVENT\_MSD\_TRANSFER) を生成する事もできます。これら 2 つのイベントは互いに独立して設定でき、下の表 3 に示す 4 通りの組み合わせが可能です。

表 3: イベント コンフィグレーション

USB ホストドライバ	USB ホスト MSD ハンドラ
転送ステータスをポーリングする	MSD 転送ステータスをポーリングする
転送ステータスをポーリングする	MSD 転送イベントを生成する
転送イベントを生成する	MSD 転送ステータスをポーリングする
転送イベントを生成する	MSD 転送イベントを生成する

USB 組み込みホスト転送イベントを使う場合、アプリケーションが要求するプログラムメモリおよびデータメモリの容量は増加しますが、アプリケーション処理をより効率的に実行できます。USB 組み込みホスト転送イベントの設定は、メディア インターフェイス層に対して透過的です。

USB 組み込みホスト MSD イベントを使う場合、プログラムメモリの要求量が増加し、これらのイベントを処理するメディア インターフェイス層を適切に構成する必要があります。一般的に、転送イベントを使うとコード アーキテクチャが複雑化し、C プログラミングの初心者には設計 / 開発 / デバッグ / 保守がより難しくなります。

USB 組み込みホスト MSD 転送イベントを使うかどうかの選択は、アプリケーション内の他のレイヤの実装によっても影響を受けます。例えば、アプリケーションノート『AN1045 - Implementing File I/O Functions using Microchip's Memory Disk Drive File System Library』は、PC で使用可能なフォーマットでファイルをオープン / クローズ / 読み書きするための関数を

提供します。ユーザはファイルを開かない限りそこに書き込む事はできないため、メモリ ディスクドライブ ファイルシステムは、要求された動作が完了するまで他のタスクの実行をブロックします。ファイルシステム層は実行をブロックするため、USB 組み込みホスト MSD 転送イベントを使えるようにメディア インターフェイス層を構成してもメリットはありません。従って、より単純なポーリング方式を使います。

Note 1: USB 組み込みホストは USB 割り込みを使いますが、ホストドライバ層からクライアントドライバへ向けて生成される転送イベントは、ポーリングによってトリガされます。これは、USB ISR がタイムリーに完了する事を確実にするためです。ホストドライバに関する詳細は、アプリケーション ノート『AN1140 - USB 組み込みホストスタック』と『AN1141 - USB 組み込みホストスタック プログラマガイド』を参照してください。

2: USB 組み込みホスト MSD 転送イベントを使うかどうかに関係なく、メディア インターフェイス層は他のシステムイベントを処理するためのイベントハンドラを含む必要があります。

## クライアント ドライバの初期化

ホスト マスタストレージクライアント ドライバの初期化には以下の関数を使います。

```
BYTE USBHostMSDInit(void);
```

この関数は動作の全ての内部変数を初期化します。この関数はアプリケーションの実行中に 1 度だけ呼び出します。

USB コンフィグレーション ツールは USB 組み込みホストドライバ、サポートするクライアント ドライバ、メディア インターフェイスが要求する全ての初期化ルーチンを呼び出すためのマクロ (USBInitialize()) を提供します。

## クライアントドライバの通常動作

通常のバックグラウンド動作の実行には以下の関数を使います。

```
void USBHostMSDTasks(void);
```

デバイスの動作を可能にするには、このルーチンを周期的に呼び出す必要があります。情報の転送のほとんどは USB 割り込みを使って処理されるため、ポーリング周期は重要ではありません。アプリケーションは複数のクラスをサポートする場合があるため、この関数は USBHostTasks() 関数 (これも周期的に呼び出す必要がある) を呼び出しません。

USB コンフィグレーション ツールは、USB ホストドライバとサポートするクライアント ドライバが要求する全てのバックグラウンド タスク ルーチンを呼び出すためのマクロ (USBTasks()) を提供します。



## サポートする論理ユニット番号 (LUN)

メディア インターフェイスの初期化に成功すると、マスタストレージ クラスドライバは直ちに EVENT\_MSD\_MAX\_LUN イベントを介してデバイスの最大論理ユニット番号(LUN)をメディアインターフェイス層に知らせます。これにより、後続の全ての転送リクエストは、有効な LUN を参照するために、この値と照合されます。

Note: メディア インターフェイスおよびファイルシステム層は、複数の LUN をサポートできない場合があります。

## 転送の実行

ペリフェラルとの通信の初期化には、以下の 2 つの関数を使います。

```
BYTE USBHostMSDRead(
    BYTE deviceAddress,
    BYTE deviceLUN,
    BYTE *commandBlock,
    BYTE commandBlockLength,
    BYTE *data,
    DWORD dataLength);
```

```
BYTE USBHostMSDWrite(
    BYTE deviceAddress,
    BYTE deviceLUN,
    BYTE *commandBlock,
    BYTE commandBlockLength,
    BYTE *data,
    DWORD dataLength);
```

commandBlock は、実行する動作をペリフェラルに伝えるための最大 16 バイトのブロックです。SCSI メディア インターフェイス層を使う場合、このブロックは要求された動作を実行するための SCSI 命令を格納します。

USB\_SUCCESS (0x00) の戻り値は、動作が正常に始まった事を示します。

通信が始まった後は、動作の完了を待機しながら USBHostTasks() と USBHostMSDTasks() を実行するよう注意してください。動作のステータスは以下の関数を呼び出す事で判定できます。

```
BOOL USBHostMSDTransferIsComplete(
    BYTE deviceAddress,
    BYTE *errorCode,
    DWORD *byteCount)
```

この関数が FALSE を返した場合、転送は未完了であり、返されたエラーコードとバイトカウントは無効です。この関数が TRUE を返した場合、返されたエラーコードは動作のステータスを示し、返されたバイトカウントは転送されたバイトの数を示します。

ホストからマスタストレージ ペリフェラルへのデータ転送の例を例 1 に示します。

### 例 1: マスタストレージ データ転送 (ペリフェラルからホストへの転送)

```
error = USBHostMSDRead( device, 0, command, 10, buffer, size );

if (!error)
{
    while (!USBHostMSDTransferIsComplete( device, &error, &count ))
    {
        USBHostTasks();
        USBHostMSDTasks();
    }
}
```

# AN1142

---

## まとめ

USB 組み込みホスト マスストレージ クラスは、広く普及している USB マスストレージ デバイスに簡潔なインターフェイスを提供します。これにより、柔軟で幅広く使われているストレージ媒体を組み込みアプリケーションで容易に利用できるようになります。

## 資料

アプリケーション ノート、AN1045 -Implementing File I/O Functions using Microchip's Memory Disk Drive File System Library

- <http://www.microchip.com>

アプリケーション ノート、AN1140 - USB 組み込みホストスタック

- <http://www.microchip.com>

アプリケーション ノート、AN1141 - USB 組み込みホストスタック プログラマガイド

- <http://www.microchip.com>

USBUniversal Serial Bus ウェブサイト :

- <http://www.usb.org>

Microchip 社ウェブサイト :

- <http://www.microchip.com>

---

---

**Microchip 社製デバイスのコード保護機能に関して次の点にご注意ください。**

- Microchip 社製品は、該当する Microchip 社データシートに記載の仕様を満たしています。
- Microchip 社では、通常の条件ならびに仕様に従って使用した場合、Microchip 社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解ではこうした手法は、Microchip 社データシートにある動作仕様書以外の方法で Microchip 社製品を使用する事になります。このような行為は知的財産権の侵害に該当する可能性が非常に高いと言えます。
- Microchip 社は、コードの保全性に懸念を抱くお客様と連携し、対応策に取り組んでいきます。
- Microchip 社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。

**コード保護機能は常に進歩しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。Microchip 社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著**

---

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。Microchip 社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。Microchip 社は、本書の情報およびその使用に起因する一切の責任を否認します。Microchip 社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途に Microchip 社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。暗黙的あるいは明示的を問わず、Microchip 社が知的財産権を保有しているライセンスは一切譲渡されません。

#### 商標

Microchip 社の名称と Microchip ロゴ、dsPIC、FlashFlex、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> ロゴ、rfPIC、SST、SST ロゴ、SuperFlash、UNI/O は、米国およびその他の国における Microchip Technology Incorporated の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL、Embedded Control Solutions Company は、米国における Microchip Technology Incorporated の登録商標です。

Silicon Storage Technology は、その他の国における Microchip Technology Incorporated の登録商標です。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT ロゴ、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPF、MPLAB 認証ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rLAB、Select Mode、SQL、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA、Z-Scale は、米国およびその他の国における Microchip Technology Incorporated の登録商標です。

SQTP は、米国における Microchip Technology Incorporated のサービスマークです。

GestIC と ULPP は、その他の国における Microchip Technology Germany II GmbH & Co. & KG (Microchip Technology Incorporated の子会社) の登録商標です。

その他、本書に記載されている商標は各社に帰属します。

©2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63276-121-7

**QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
= ISO/TS 16949 =**

Microchip 社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州)の本部、設計部およびウェハ製造工場そしてカリフォルニア州とインドのデザインセンターが ISO/TS-16949:2009 認証を取得しています。Microchip 社の品質システム プロセスおよび手順は、PIC<sup>®</sup> MCU および dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> コード ホッピング デバイス、シリアル EEPROM、マイクロペリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関する Microchip 社の品質システムは ISO 9001:2000 認証を取得しています。



# MICROCHIP

## 各国の営業所とサービス

### 北米

**本社**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel:480-792-7200  
Fax:480-792-7277  
技術サポート :  
<http://www.microchip.com/support>  
URL:  
[www.microchip.com](http://www.microchip.com)

**アトランタ**  
Duluth, GA  
Tel:678-957-9614  
Fax:678-957-1455

**オースティン (TX)**  
Tel:512-257-3370

**ボストン**  
Westborough, MA  
Tel:774-760-0087  
Fax:774-760-0088

**シカゴ**  
Itasca, IL  
Tel:630-285-0071  
Fax:630-285-0075

**クリーブランド**  
Independence, OH  
Tel:216-447-0464  
Fax:216-447-0643

**ダラス**  
Addison, TX  
Tel:972-818-7423  
Fax:972-818-2924

**デトロイト**  
Novi, MI  
Tel:248-848-4000

**ヒューストン (TX)**  
Tel:281-894-5983

**インディアナポリス**  
Noblesville, IN  
Tel:317-773-8323  
Fax:317-773-5453

**ロサンゼルス**  
Mission Viejo, CA  
Tel:949-462-9523  
Fax:949-462-9608

**ニューヨーク (NY)**  
Tel:631-435-6000

**サンノゼ (CA)**  
Tel:408-735-9110

**カナダ - トロント**  
Tel:905-673-0699  
Fax:905-673-6509

### アジア / 太平洋

**アジア太平洋支社**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel:852-2943-5100  
Fax:852-2401-3431

**オーストラリア - シドニー**  
Tel:61-2-9868-6733  
Fax:61-2-9868-6755

**中国 - 北京**  
Tel:86-10-8569-7000  
Fax:86-10-8528-2104

**中国 - 成都**  
Tel:86-28-8665-5511  
Fax:86-28-8665-7889

**中国 - 重慶**  
Tel:86-23-8980-9588  
Fax:86-23-8980-9500

**中国 - 杭州**  
Tel:86-571-8792-8115  
Fax:86-571-8792-8116

**中国 - 香港 SAR**  
Tel:852-2943-5100  
Fax:852-2401-3431

**中国 - 南京**  
Tel:86-25-8473-2460  
Fax:86-25-8473-2470

**中国 - 青島**  
Tel:86-532-8502-7355  
Fax:86-532-8502-7205

**中国 - 上海**  
Tel:86-21-5407-5533  
Fax:86-21-5407-5066

**中国 - 瀋陽**  
Tel:86-24-2334-2829  
Fax:86-24-2334-2393

**中国 - 深圳**  
Tel:86-755-8864-2200  
Fax:86-755-8203-1760

**中国 - 武漢**  
Tel:86-27-5980-5300  
Fax:86-27-5980-5118

**中国 - 西安**  
Tel:86-29-8833-7252  
Fax:86-29-8833-7256

**中国 - 厦門**  
Tel:86-592-2388138  
Fax:86-592-2388130

**中国 - 珠海**  
Tel:86-756-3210040  
Fax:86-756-3210049

### アジア / 太平洋

**インド - バンガロール**  
Tel:91-80-3090-4444  
Fax:91-80-3090-4123

**インド - ニューデリー**  
Tel:91-11-4160-8631  
Fax:91-11-4160-8632

**インド - プネ**  
Tel:91-20-3019-1500

**日本 - 大阪**  
Tel:81-6-6152-7160  
Fax:81-6-6152-9310

**日本 - 東京**  
Tel:81-3-6880-3770  
Fax:81-3-6880-3771

**韓国 - 大邱**  
Tel:82-53-744-4301  
Fax:82-53-744-4302

**韓国 - ソウル**  
Tel:82-2-554-7200  
Fax:82-2-558-5932 または  
82-2-558-5934

**マレーシア - クアラルンプール**  
Tel:60-3-6201-9857  
Fax:60-3-6201-9859

**マレーシア - ペナン**  
Tel:60-4-227-8870  
Fax:60-4-227-4068

**フィリピン - マニラ**  
Tel:63-2-634-9065  
Fax:63-2-634-9069

**シンガポール**  
Tel:65-6334-8870  
Fax:65-6334-8850

**台湾 - 新竹**  
Tel:886-3-5778-366  
Fax:886-3-5770-955

**台湾 - 高雄**  
Tel:886-7-213-7830

**台湾 - 台北**  
Tel:886-2-2508-8600  
Fax:886-2-2508-0102

**タイ - バンコク**  
Tel:66-2-694-1351  
Fax:66-2-694-1350

### ヨーロッパ

**オーストリア - ヴェルス**  
Tel:43-7242-2244-39  
Fax:43-7242-2244-393

**デンマーク - コペンハーゲン**  
Tel:45-4450-2828  
Fax:45-4485-2829

**フランス - パリ**  
Tel:33-1-69-53-63-20  
Fax:33-1-69-30-90-79

**ドイツ - デュッセルドルフ**  
Tel:49-2129-3766400

**ドイツ - ミュンヘン**  
Tel:49-89-627-144-0  
Fax:49-89-627-144-44

**ドイツ - プフォルトツハイム**  
Tel:49-7231-424750

**イタリア - ミラノ**  
Tel:39-0331-742611  
Fax:39-0331-466781

**イタリア - ベニス**  
Tel:39-049-7625286

**オランダ - ドリユーン**  
Tel:31-416-690399  
Fax:31-416-690340

**ポーランド - ワルシャワ**  
Tel:48-22-3325737

**スペイン - マドリード**  
Tel:34-91-708-08-90  
Fax:34-91-708-08-91

**スウェーデン - ストックホルム**  
Tel:46-8-5090-4654

**イギリス - ウォーキングム**  
Tel:44-118-921-5800  
Fax:44-118-921-5820

03/25/14