



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。

AN1163

USB 組み込みデバイスにおける HID クラス

Author: Sean Justice
Microchip Technology Inc.

はじめに

HID (Human Interface Device) は USB (Universal Serial Bus) で使われるクラスの 1 つです。HID クラスには、人がコンピュータ システムを操作するために使うデバイスが含まれます。これらのデバイスには例えばマウス、キーボード、ジョイスティック、ボタン、スイッチ等があります。HID クラスのデバイスはヒューマン インターフェイスからの情報をコンピュータ システムへ提供するだけでなく、コンピュータ システムの動作を示す各種の出力をユーザに提供する機能も備える事ができます。

本書では、USB モジュールを内蔵した Microchip 社製 32 ビット PIC[®] マイクロコントローラ上で動作するほとんどのアプリケーションに組み込み可能な HID ファンクション ドライバについて説明します。HID ファンクション ドライバは Microchip 社の PIC32 USB デバイスタックを使います。

本書では、HID ファンクション ドライバを使って動作するデモ アプリケーションについても説明します。この HID デモ アプリケーションはマウスをシミュレートします。

前提条件

本書は、読者が Microchip 社製開発ツール MPLAB[®] IDE と MPLAB REAL ICE[™] インサーキット エミュレータを使い慣れており、C 言語と USB デバイス プロトコルおよびディスクリプタに関する十分な知識を有している事を前提としています。本書では、これらの技術に関する専門用語を使いますが、それらの概念について詳しく解説しません。より詳細な情報が必要な読者は、関連する仕様書もお読みください。

特長

本書では、HID ファンクション ドライバの主要構成要素に関して説明します。Microchip 社の HID ファンクション ドライバは以下の特長を備えます。

- RTOS またはアプリケーションに依存せず機能
- Microchip 社の MPLAB IDE ツール群をサポート
- 『Universal Serial Bus (USB) Device Class Definition for Human Devices (HID), Version 1.11』が定義する HID 1.1 仕様をサポート (仕様書は <http://www.usb.org/developers/hidpage/> から入手可能)
- USB エンドポイントを 1 つだけ使用
- HID レポートを生成するためのマクロを提供
- 複数の HID レポートに対応
- 『Universal Serial Bus Specification, Revision 2.0』の Chapter 9 に記載されている標準 HID USB コンフィグレーション要求に対応 (仕様書は <http://www.usb.org/developers/docs/> から入手可能)

制限事項

この HID ファンクション ドライバは組み込みシステムでの使用を前提に開発されたため、USB デバイスタックの制限事項を受け継ぎます (Microchip 社アプリケーション ノート 『AN1176 - USB Device Stack for PIC32 Programmer's Guide』参照)。

システム ハードウェア

このアプリケーションとファームウェアは以下のハードウェア向けに開発されました。

- USB 対応 PIC32 ファミリー マイクロコントローラ PIM (Plug-In Module)
- Microchip 社製 Explorer 16 開発ボード
- USB PICtail[™] Plus ドータボード

USB デバイスタックと HID ファンクション ドライバのソースファイルを修正する事で他の開発ボードを使える他、ほとんどのハードウェアの違いに対応できます。

PIC[®] MCU のメモリリソース要件

HID ファンクション ドライバには、USB デバイスタックの全てのメモリ要件がそのまま適用されます。詳細はアプリケーション ノート『AN1176 - USB Device Stack for PIC32 Programmer's Guide』内の「Memory Resource Requirements」を参照してください。

HID ファンクション ドライバが消費するフラッシュメモリと RAM の容量を下表に示します。

表 1: メモリ必要量

メモリ	容量
フラッシュ	3,284 Bytes
RAM	160 Bytes

HID ファンクション ドライバ アプリケーションは以下を定義します。

- USB ディスクリプタ テーブル
- HID レポート構造体

USB ディスクリプタ テーブルと HID レポート構造体は全ての HID ファンクション ドライバ アプリケーションに必要です。

USB ディスクリプタ テーブルのメモリ要求量を下表に示します。

表 2: USB ディスクリプタ テーブル

メモリ	容量
フラッシュ	56 Bytes

HID レポート構造体のメモリ要求量を下表に示します。

表 3: HID レポート構造体

メモリ	容量
フラッシュ	1076 Bytes

USB ディスクリプタ テーブルと HID レポートが消費するメモリリソースの容量は、以下を含む (しかしこれらに限定されない) 各種の要因によって変化します。

- ユーザが必要とする USB ファンクション ドライバの数
 - コンフィグレーション、インターフェイス、エンドポイントコンフィグレーションの数
- HID レポートのサイズと定義

ソースファイルのインストール

HID ファンクション ドライバのソースファイルは Microchip 社ウェブサイト (補遺 F: 「HID ファンクション ドライバのソースコード」) からダウンロードできます。ソースコードは 1 つの Microsoft Windows[®] 用インストール ファイルに収められています。

インストール手順は以下の通りです。

1. インストール ファイルを実行します。Windows のインストール ウィザードがインストール手順を指示します。
2. インストールを続けるには、ソフトウェア ライセンス使用許諾契約に対して [同意する] をクリックする必要があります。
3. インストールが完了すると、以下のディレクトリが作成されます。
 - a) 「hid_device_driver」ディレクトリ
(\PIC32 Solutions\Microchip\USB)
このディレクトリは、HID ファンクション ドライバのソースファイルと関連文書を格納します。
 - b) 「usb_func_hid」ディレクトリ
(PIC32 Solutions\Microchip\Include\USB)
このディレクトリは、PIC32 USB デバイスタックと HID ファンクション ドライバ用のインクルード ファイルを格納します。
 - c) 「usb_hid_mouse_device_demo」ディレクトリ
(\PIC32 Solutions)
このディレクトリは、HID ファンクション ドライバを使ったマウスデモ用のデモプロジェクトとソースファイルを格納します。
4. 最新バージョンに固有の機能と制限事項については、リリースノートを参照してください。

ソースファイルの構成

HID ファンクション ドライバは、複数のディレクトリに格納された複数のファイルによって構成されます。表 4 に、ソースファイルとディレクトリの一覧を示します。

表 4: HID ソースファイルのディレクトリ構造

ファイル	ディレクトリ	内容
hid.c	\PIC32 Solutions\Microchip\USB\hid_device_driver	USB HID デバイスクラス ドライバ
hiddsc.tmpl	\PIC32 Solutions\Microchip\USB\hid_device_driver	HID ディスクリプタ テンプレート
hidreport.tmpl	\PIC32 Solutions\Microchip\USB\hid_device_driver	HID レポート テンプレート
usb_device_hid.h	\PIC32 Solutions\Microchip\Include\USB	API 定義と 変更可能なマクロ
hidpri.h	\PIC32 Solutions\Microchip\USB\hid_device_driver	プライベート関数と マクロ定義
hiddesc.h	\PIC32 Solutions\Microchip\Include\USB	USB HID ディスクリプタ
hidreport.h	\PIC32 Solutions\Microchip\Include\USB	HID レポート定義

デモ アプリケーション

Microchip 社の HID ファンクション ドライバには、コンピュータ用マウスをシミュレートするデモ アプリケーションが含まれています。このアプリケーションは、Explorer 16 開発ボード上で Microchip 社の USB デバイスタック ソフトウェアを使って動作するように設計されています。しかし、アプリケーションを修正して他のボードを使う事もできます。

マウス シミュレーション デモ アプリケーションは以下を実行します。

- HID ファンクション ドライバ向けの USB デバイス エnumレーション
- Explorer 16 ボード上のスイッチ SW3 (RD6) が押された時にマウスをエミュレート
 - 以下のスイッチ操作によって特定のマウス動作を実行
 - スイッチ SW6 (RD7) を押すとマウスカーソルは右へ移動する
 - スイッチ SW6 (RD7) と SW4 (RD13) を押すとマウスカーソルは左へ移動する
 - スイッチ SW5 (RA7) を押すとマウスカーソルは上へ移動する
 - スイッチ SW5 (RA7) と SW4 (RD13) を押すとマウスカーソルは下へ移動する

デモ アプリケーションのプログラミング

ターゲットにデモ アプリケーションをプログラミングするには、MPLAB REAL ICE インサーキット エミュレータにアクセスする必要があります。以下の手順は、MPLAB IDE の使用を前提とします。これらを使わない場合、使用プログラムの説明書を参照してください。

1. MPLAB REAL ICE インサーキット エミュレータを Explorer 16 ボード (または他のターゲットボード) に接続します。
2. ターゲットボードに電源を投入します。
3. MPLAB IDE を起動します。
4. 使用する USB 対応 PIC32 デバイスを選択します (この手順は、既にビルド済みの HEX ファイルをインポートする場合にのみ必要です)。
5. MPLAB REAL ICE をプログラマとして有効にします。
6. ビルド済みの HEX ファイルを使う場合、そのファイルを MPLAB にインポートします。
7. HEX ファイルをリビルドする場合、プロジェクト ファイルを開き、ビルド手順に従ってアプリケーション HEX ファイルを作成します。
8. このデモ アプリケーションには Explorer 16 ボード向けのオプションが設定されています。別のボードにプログラミングする場合、MPLAB IDE のコンフィグレーション設定メニューから適切なオシレータモードを選択する必要があります。
9. MPLAB IDE で [Programmer] メニューを選択し、[Select Programmer]->[6 REAL ICE] をクリックします。
10. MPLAB IDE が REAL ICE インサーキット エミュレータと PIC MCU を検出したら、[Programmer] メニューを選択して [Program] をクリックすると、デバイスのプログラミングが始まります。
11. 数秒後にメッセージ「Programming successful」が表示されます。このメッセージが表示されない場合、ボードと MPLAB REAL ICE の接続を確認してください。問題が解消しない場合、MPLAB IDE と REAL ICE のオンラインヘルプを参照してください。
12. ターゲットボードの電源を OFF にし、ボードから MPLAB REAL ICE ケーブルを切り離します。
13. ボードに電源を再投入し、LCD に「PIC32 HID Device」と表示される事を確認します。表示されない場合、プログラミング手順を確認し、必要に応じてプログラミングをやり直してください。

デモ アプリケーションのビルド

本書のデモ アプリケーションは、Microchip 社の C32 C コンパイラを使ってビルドできます。必要に応じて、Microchip 社製マイクロコントローラ向けに普段お使いのコンパイラヘソースを移植してください。

本書のアプリケーションには、MPLAB IDE 向けに定義済みのマウス HID デモ用プロジェクト ファイルが含まれています。このプロジェクトは USB 対応 PIC32 デバイスを使って作成されました。異なるデバイスを使う場合、MPLAB IDE メニューコマンドを使って適切なデバイスを選択する必要があります。

加えて、このデモ アプリケーション プロジェクトは、MPLAB IDE の [Build Options] で定義されている追加のインクルードパスを使います。

以下のインクルードパスが必要です。

- .\
- ..\Microchip\Include
- ..\..\Microchip\Include

表 5 に、デモ アプリケーションのビルドに必要なソースファイルの一覧を示します。

表 5: デモ アプリケーションのプロジェクト ファイル

ファイル	ディレクトリ	内容
mouse_demo.c	\PIC32 Solutions\usb_hid_mouse_device_demo	メインのデモ ソースファイル
mouse_dsc.c	\PIC32 Solutions\usb_hid_mouse_device_demo	マウス HID USB ディスクリプタ
mouse_report.c	\PIC32 Solutions\usb_hid_mouse_device_demo	マウスデモ向け HID レポート
HardwareProfile.h	\PIC32 Solutions\usb_hid_mouse_device_demo	PIC32 向けハードウェア定義
usb_config.h	\PIC32 Solutions\usb_hid_mouse_device_demo	USB 固有のヘルパー関数向け定義
hid.c	\PIC32 Solutions\Microchip\USB\hid_device_driver	USB HID ソースファイル
hidpri.h	\PIC32 Solutions\Microchip\USB\hid_device_driver	プライベート関数とマクロ定義
hid.h	\PIC32 Solutions\Microchip\Include\USB	USB HID インクルードファイル
hiddesc.h	\PIC32 Solutions\Microchip\Include\USB	HID 固有ディスクリプタの定義
hidreport.h	\PIC32 Solutions\Microchip\Include\USB	HID レポート構造体マクロと定義
usb_device.c	\PIC32 Solutions\Microchip\USB	USB デバイス API
usb_hal.c	\PIC32 Solutions\Microchip\USB	USB ハードウェア API
usb_hal_core.c	\PIC32 Solutions\Microchip\USB	USB ハードウェア コア API
usb.h	\PIC32 Solutions\Microchip\USB	USB トップレベルインクルードファイル
usb_ch9.h	\PIC32 Solutions\Microchip\Include\USB	USB 定義とサポート (『Universal Serial Bus Specification, Revision 2.0』の Chapter 9 に従う)
usb_common.h	\PIC32 Solutions\Microchip\Include\USB	USB 一般定義
usb_device.h	\PIC32 Solutions\Microchip\Include\USB	USB デバイス定義と API プロトタイプ
usb_hal.h	\PIC32 Solutions\Microchip\Include\USB	USB ハードウェアサポート
mstimer.c	\PIC32 Solutions\Microchip\Common	1 ms タイマ
ex16lcd.c	\PIC32 Solutions\Microchip\Common	Explorer 16 開発ボード (DM2400x)
mstimer.h	\PIC32 Solutions\Microchip\Include	1 ms タイマ定義
ex16lcd.h	\PIC32 Solutions\Microchip\Include	Explorer 16 開発ボード LCD 定義

以下では、デモ アプリケーションをビルドするための手順について説明します。この手順は MPLAB IDE の使用を前提とし、MPLAB IDE を使い慣れた読者向けに書かれています。MPLAB IDE を使わない場合、ご使用になるプログラムの説明書を参照してプロジェクトを作成およびビルドしてください。

1. Microchip 社製 HID ファンクション ドライバのソースファイルをあらかじめインストールしておく必要があります。インストール方法については、「**ソースファイルのインストール**」を参照してください。
2. MPLAB IDE を起動し、プロジェクトファイルを開きます。
3. MPLAB IDE のメニューコマンドを使ってプロジェクトをビルドします。このデモ プロジェクトは、ソースファイルがインストール ウィザードの推奨ディレクトリ構造に保存されている場合に正しくコンパイルするよう作成されています。ソースファイルを異なるディレクトリにインストールした場合、またはインストール後に移動した場合、プロジェクトのビルド設定を変更するか作成し直す必要があります。詳細は「**デモ アプリケーションのビルド**」を参照してください。
4. 正しくビルドされなかった場合、MPLAB IDE とコンパイラの設定を確認してください。

アプリケーションに固有の USB サポート

Microchip 社の PIC32 向け USB デバイス ファームウェア スタックを使用するにあたり、HID デモは以下のアプリケーションに固有のテーブルを実装します。

- USB ディスクリプタ テーブル
- エンドポイント コンフィグレーション テーブル
- ファンクション ドライバテーブル

USB ディスクリプタ テーブル

全ての USB デバイスは、そのデバイスに関する詳細情報（使用するクラスドライバ等）を USB ホストに提供するために、一連のディスクリプタ（データ構造体）を提供する必要があります。これらのディスクリプタの詳細（格納する情報等）は、『Universal Serial Bus Specification, Revision 2.0』の Chapter 9 と『Universal Serial Bus (USB) Device Class Definition for Human Devices (HID), Version 1.11』で明確に定義されています。詳細については、これらの文書を参照してください。

USB デバイス ディスクリプタは以下の 3 つのグループに分類できます。

- デバイス
- コンフィグレーション
- スtring

デバイス ディスクリプタは、デバイスのタイプと選択可能なコンフィグレーションの数を指定します。

コンフィグレーション ディスクリプタは、インターフェイスのタイプと使用するエンドポイントを記述します。このグループには、クラス別ディスクリプタも含まれます。

String ディスクリプタは、ホストが表示するユーザ向け情報を提供します。通常、このディスクリプタの使用は必須ではありません。

デモ アプリケーションのディスクリプタ テーブル

デモ アプリケーションが提供するディスクリプタ テーブルはソースファイル `mouse_dsc.c` に含まれています（このテーブルの概要は補遺 E: 「**USB ディスクリプタ テーブル定義**」に記載しています）。

デモ アプリケーションのディスクリプタ テーブルを変更する事で、他のインターフェイスまたはコンフィグレーションを追加できます。しかし、『Universal Serial Bus Specification, Revision 2.0』の Chapter 9 と、その他の適用されるデバイス ファンクション ドライバに固有の仕様について完全に理解してからディスクリプタ テーブルの変更を試みる事を推奨します。

エンドポイント コンフィグレーション テーブル

USB デバイスタックはエンドポイント コンフィグレーション テーブルを使う事で、ディスクリプタ テーブルの定義に従うインターフェイスと代替設定によって、全てのエンドポイントを正しく設定します。このテーブルは、各エンドポイントで発生するイベントを処理するためのファンクション ドライバを指定します。

各テーブルエントリは以下の情報を格納します。

- 最大パケットサイズ
- コンフィグレーション フラグ
- コンフィグレーション番号
- エンドポイント番号
- インターフェイス番号
- 代替設定
- デバイス ファンクション テーブル内のエンドポイントハンドラに対するインデックス

HID ファンクション ドライバは1つのエンドポイント (インタラプト IN) だけを使うため、このデモ アプリケーションのエンドポイント コンフィグレーション テーブルにはエントリが1つしかありません。ソース ファイル `mouse_dsc.c` 内のこのテーブルを例1に示します。エンドポイント コンフィグレーション テーブルの詳細については、アプリケーション ノート『AN1176 - USB Device Stack for PIC32 Programmer's Guide』を参照してください。

例 1: エンドポイント コンフィグレーション テーブル

```
const EP_CONFIG _EpConfigTbl[] =
{
    {
        HID_MAX_REPORT_SIZE, // max pack size
        USB_EP_TRANSMIT|USB_EP_HANDSHAKE, // configure for Tx and enable
        // handshaking
        1, // configuration number
        1, // endpoint number
        0, // interface number
        0, // alternate setting
        0 // handler funciton index
    }
};
```

AN1163

ファンクション ドライバテーブル

デバイスは複数のクラスまたはベンダー固有 USB デバイス ファンクション ドライバを実装する場合があります。Microchip 社の PIC32 USB デバイスタックはテーブルを使ってアクセスを管理する事で、それらのファンクション ドライバをサポートします。1つのテーブルエントリは、1つのファンクション ドライバを管理するための情報を格納します。

各テーブルエントリは以下の情報を格納します。

- 初期化ルーチン
- イベントハンドラ ルーチン
- 初期化フラグ

このデモ アプリケーションのファンクション ドライバテーブルにはエントリが1つしかありません(1つのHIDファンクション ドライバしか存在しないため)。ソースファイル `mouse_dsc.c` 内のこのテーブルを例 2 に示します。ファンクション ドライバ テーブルの詳細については、アプリケーション ノート『AN1176 - USB Device Stack for PIC32 Programmer's Guide』を参照してください。

例 2: ファンクション ドライバテーブル

```
const FUNC_DRV _DevFuncTbl[] =
{
    {HIDInit, // Initialization routine
    HIDEventHandler, // Event handler routine
    0 // Initialization flags
    }
};
```

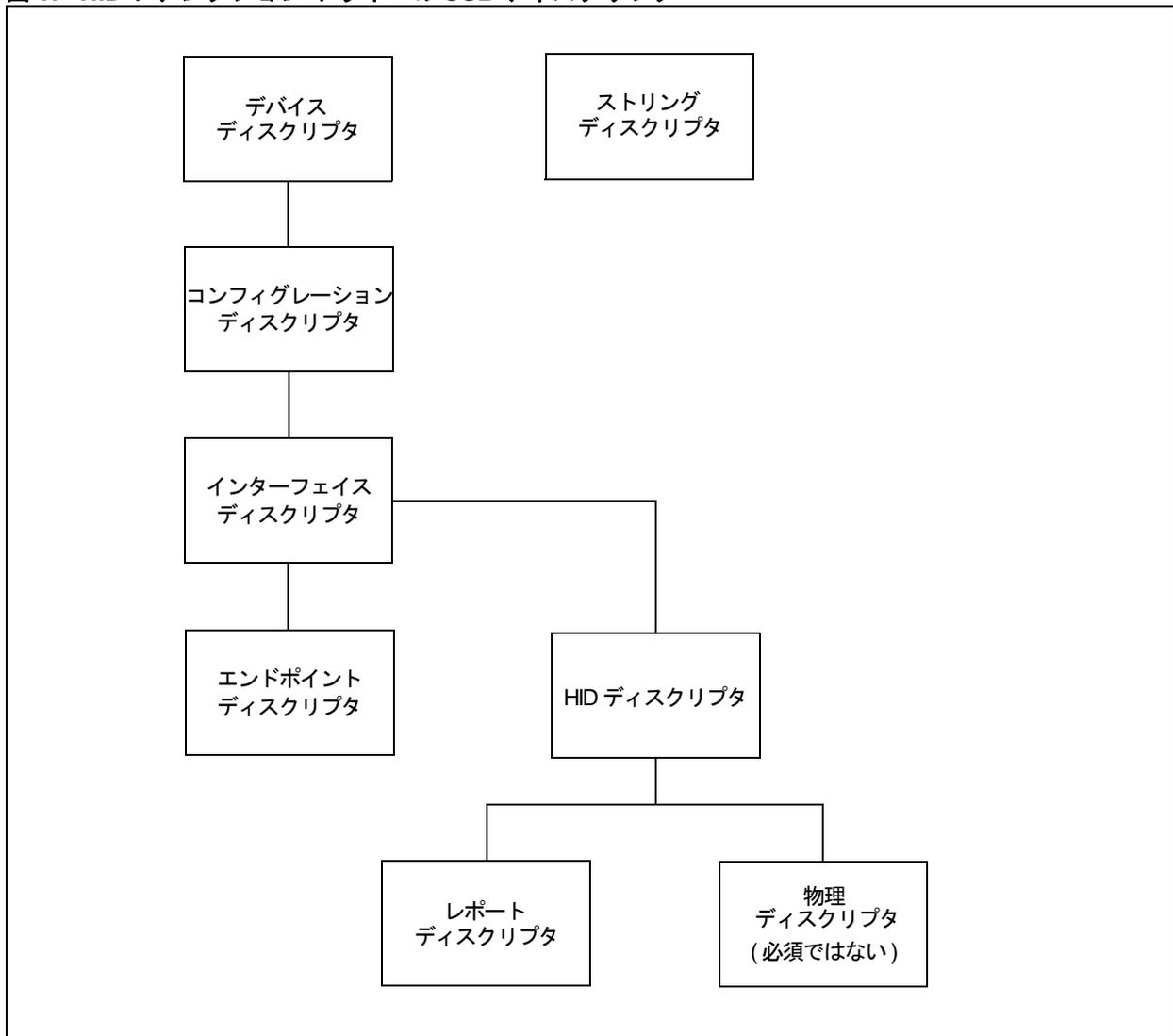
HID ファンクション ドライバの概要

HID ディスクリプタ

全ての USB デバイスは、そのデバイスに対応するディスクリプタ構造体を有します。各デバイスは、インターフェイス層で定義される複数のクラス (HID 等) を含む事ができます。図 1 に、HID クラスデバイスのディスクリプタ構造体のツリー図を示します。

HID ディスクリプタは、その下に存在する他の HID クラス固有ディスクリプタの個数を示します。少なくとも 1 つのレポート ディスクリプタが必要です。物理ディスクリプタは必須ではありません。レポート ディスクリプタは、デバイスが生成するデータレポートの書式と意味を記述します。ホストの HID クラスドライバは、クラス固有リクエストを使って HID レポート ディスクリプタを読み込みます。初期化後、ユーザがデバイスを操作すると、デバイスはそれをホストに示すためにレポートを生成します。

図 1: HID ファンクション ドライバの USB ディスクリプタ



AN1163

HID ファンクション ドライバ (USB デバイス) は、既定値パイプ (コントロール) またはインタラプト IN パイプを使って HID クラスドライバ (USB ホスト) と通信します。HID ファンクション ドライバは、データを送信するために必ずインタラプト IN パイプを使いますが、必要に応じてインタラプト OUT エンドポイントも使えます。

ホストがデバイスに向けて生成するレポートは、Set Report リクエストを使って、インタラプト OUT エンドポイントまたは既定値エンドポイント (コントロール) を介して送信されます。

表 6: HID エンドポイント コンフィグレーション

パイプ	概要	必要性
コントロール (エンドポイント 0)	USB 制御、クラスリクエストコード、ポーリングしたデータ (メッセージデータ)	必須
インタラプト IN	デバイスからホストへのデータ入力	必須
インタラプト OUT	ホストからデバイスへのデータ出力	任意

HID レポート ディスクリプタ

HID レポート ディスクリプタは、「アイテム」と呼ぶ各種の情報項目によって構成されます。各アイテムは、レポートデータの 1 つの特性を表します。

レポートアイテムは、1 バイトの接頭辞の後にペイロードが続く一般的なフォーマットに従います。接頭辞バイトはペイロードのタグ、タイプ、サイズを格納します。図 2 に、アイテムの接頭辞バイトを示します。

レポートアイテムのタイプ

アイテムには以下のタイプがあります。

- ショートアイテム
- ロングアイテム

ショートアイテムのペイロードのサイズは 0、1、2、4 バイトのいずれかです。図 3 に、ショートアイテムの例を示します (ペイロードサイズは 2 バイト)。

ロングアイテムのサイズは最大で 255 バイトです。ロングアイテムは 1 バイトの接頭辞 (0xFE にハードコードされる)、1 バイトのペイロードサイズ値、1 バイトのタグ、ペイロードによって構成されます。図 4 に、ロングアイテムのフォーマットを示します。

図 2: HID レポートアイテムの接頭辞バイト

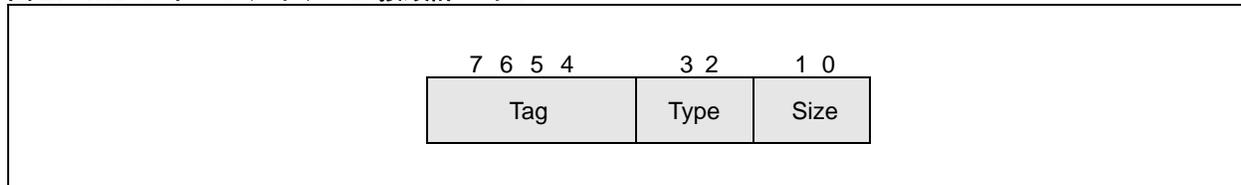


図 3: HID レポートのショートアイテムのフォーマット (ペイロードサイズ = 2 バイト)

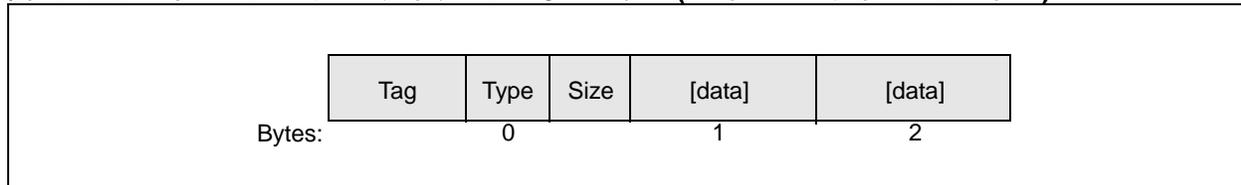
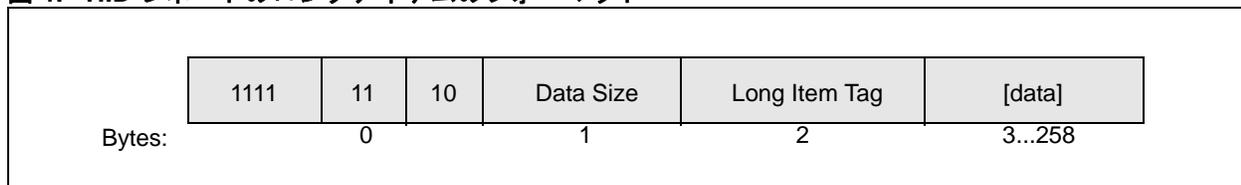


図 4: HID レポートのロングアイテムのフォーマット



レポートアイテム

アイテムは以下のクラスに分類されます。

- Input アイテム
- Output アイテム
- Feature アイテム

Input アイテムクラスは、キーボードやマウス等の物理的操作によって提供されるデータに関する情報を記述します。

Output アイテムクラスは、デバイスに向けて送信されるデータ (例: キーボードの Caps Lock キーの ON/OFF を示す LED へのデータ等) を記述します。

Feature アイテムクラスは、デバイスに向けて送信されるコンフィグレーション情報を記述します。

HID ファンクション ドライバデモに含まれるレポート ディスクリプタを以下に示します。

```
Usage Page (Generic Desktop),
Usage (Mouse),
Collection (Application),
    Usage (Pointer),
    Collection (Physical),
        Usage Page (Button Page),
        Usage Minimum (1),
        Usage Maximum (3),
        Logical Minimum (0),
        Logical Maximum (1),
        Report Count (3),
        Report Size (1),
        Input (Data, Variable, Absolute),
        Report Count (1),
        Report Size (5),
        Input (Constant),
        Usage Page (Generic Desktop),
        Usage (X),
        Usage (Y),
        Logical Minimum (-127),
        Logical Maximum (127),
        Report Count (2),
        Report Size (8),
        Input (Data, Variable, Relative),
    End Collection,
End Collection
```

1 つの行は 1 つのレポートアイテムを表します。HID ファンクション ドライバはこれらのレポートアイテムを使って、図 5 に示すレポート構造体を定義します。

- レポートアイテム **Usage Page (Generic Desktop)** と **Usage Page (Mouse)** は、これから定義するレポート構造体がデスクトップ向けである事と、デバイスがマウスであることを HID ホストに知らせます。
- **Collection (Physical)** セクション内のアイテムは、レポート内のデータを定義します。

3 個のボタン (図 5 の B1、B2、B3) はレポート内で以下によって定義します。

- **Usage Minimum (1)** と **Usage Maximum (3)** は、マウスが最小 1 個、最大 3 個のボタンを使う事を表します。
- **Logical Minimum (0)** と **Logical Maximum (1)** は、レポートが格納する値のレンジを表します。この場合、ボタンの値は ON 時に (1)、OFF 時に (0) です。
- **Report Count (3)** と **Report Size (1)** は、ボタンの個数と、各ボタンのデータサイズを表します。レポートは 3 個のボタンを含み、各ボタンの値は 1 ビットで表します (レポート内のバイト 0 の bit 0-2)。
- **Input (Data, Variable, Absolute)** は、送信データのタイプを指定します。ボタンはホストへの入力データであり、それらの値は変化し、絶対的 (ON または OFF) です。

バイト 0 の残りのビットは 0 にハードコードします。

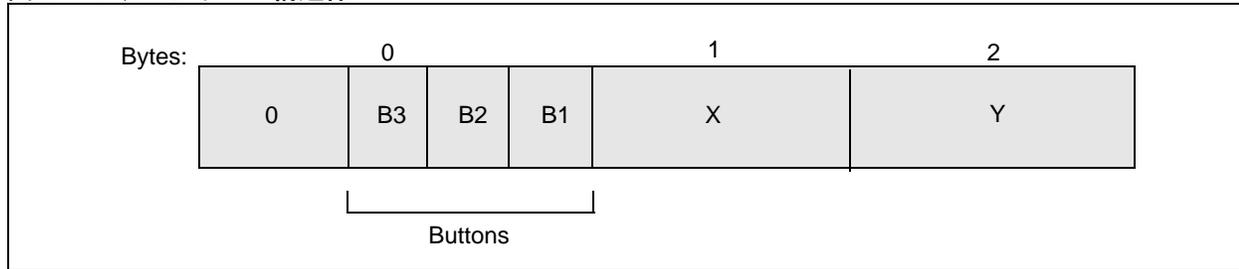
- **Report Count (1)**
- **Report Size (5)**
- **Input (Constant)**

マウスの相対位置はバイト 1 (X) とバイト 2 (Y) によって表します。

- **Usage (X)** と **Usage (Y)** は、次の 2 つの入力を位置として識別します。
- **Logical Minimum (-127)** と **Logical Maximum (127)** は、レポートが格納する値のレンジを表します。
- **Report Count (2)** と **Report Size (8)** は、1 つの位置を表すためのデータの個数と、各データのサイズを定義します。
- **Input (Data, Variable, Relative)** は、送信データのタイプを指定します。位置はホストへの入力データであり、それらの値は変化し、直前のレポートに対する相対位置を表します。

AN1163

図 5: マウスレポート構造体



レポート

HID ファンクション ドライバは、1 ms 周期で USB トランザクションを送信または受信できます。1 つまたは複数の USB トランザクションは、デバイスへの有意なデータの転送を形成します。

表 7 に、HID レポートのタイプを示します。

表 7: データ転送レポート

レポートタイプ	パイプ
Input	インタラプト IN
Output	コントロール、 インタラプト OUT も任意に使用可能
Feature	コントロール、 インタラプト OUT も任意に使用可能

HID レポートにはレポート ID (1 バイト接頭辞) を割り当てる必要がある場合とそうでない場合があります。HID ファンクション ドライバが複数のレポートを生成する場合、レポート ID を使って、適正なデータ処理を実行するためにレポートにタグを付ける必要があります。レポート ディスクリプタ内にレポート ID アイテムタグが存在しない場合、Input/Output/Feature レポート構造体が 1 つしか存在しないと見なされます。レポート内の全てのデータはリトルエンディアン形式で表現されます (図 5 参照)。

Input レポートは、ユーザ操作に関するデータをデバイスからホストへ送信するために使います。Output レポートは、ホストからデバイスへ制御データ (キーボードの Caps Lock インジケータ LED を点灯するための命令等) を送信するために使います。Feature レポートは、ホストがデバイスを設定するために使います。このデモ アプリケーションは、Input レポートを 1 つだけサポートします。

まとめ

本書では、HID ファンクション ドライバを使ってマウスをシミュレートするカスタム デモ アプリケーションについて説明しました。

通常、USB を制御するにはデバイス識別、制御、データ転送に関するプロトコルに対応する必要があります。しかし、Microchip 社が USB プロトコルを細部まで配慮して設計した簡潔なHIDファンクションドライバを使う事で、弊社製 USB 対応マイクロコントローラを使ったアプリケーションを容易に実装できます。

参考資料

- Microchip 社アプリケーションノート、AN1176 - USB Device Stack for PIC32 Programmer's Guide
- Microchip MPLAB® IDE
インサーキット開発環境 (以下で無償提供)
www.microchip.com/mplabide
- Universal Serial Bus Specification, Revision 2.0
<http://www.usb.org/developers/docs>
- Universal Serial Bus (USB) Device Class Definition for Human Devices (HID), Version 1.11
http://www.usb.org/developers/docs/HID1_11.pdf

補遺 A: MICROCHIP 社製 HID ファンクション ドライバの依存性

HIDEventHandler - USB デバイス イベントハンドラ関数

本書の HID ファンクション ドライバ アプリケーションは、Microchip 社アプリケーション ノート『AN1176 - USB Device Stack for PIC32 Programmer's Guide』に準拠するイベントハンドラを提供します。このルーチンは USB デバイス ファンクション ドライバテーブルに含める必要があります。

構文

```
PUBLIC BOOL HIDEventHandler(USB_EVENT event, void *data, UINT size)
```

パラメータ

event - 発生したイベントを特定する列挙データ型

data - イベント依存データを指すポインタ

size - イベント依存データのサイズ (byte)

戻り値

TRUE (実行に成功した)、FALSE (実行に失敗した)

備考

HIDInit - USB デバイス初期化ハンドラ関数

このルーチンは、HID ファンクション ドライバに関連する全てのデータ構造体を初期化します。このルーチンは USB デバイス ファンクション ドライバテーブルに含める必要があります。

構文

```
PUBLIC BOOL HIDInit(unsigned long flags)
```

パラメータ

flags - 予約済み (0 を引き渡す)

戻り値

TRUE (初期化に成功した)、FALSE (初期化に失敗した)

備考

「flags」パラメータは、USB デバイス ファンクション ドライバ テーブルの初期化フラグ パラメータとして引き渡されます。

Note: HIDEventHandler と HIDInit の詳細については、アプリケーション ノート『AN1176 - USB Device Stack for PIC32 Programmer's Guide』を参照してください。

補遺 B: HID ファンクション レポート ハンドラ テーブル

HID クラスドライバからレポートを出力するために、レポートを処理および取得するためのルーチンを提供する必要があります。HID ソースファイル (hid.c) にはテーブル `_HidReportHandler` が含まれています。

各テーブルエントリは以下の情報を格納します。

- Report Handler ルーチン
- Get Report ルーチン

API 定義

Report Handler レ ルーチン

このルーチンは出力レポートを処理するために呼び出します。

構文

```
BOOL < Report Handler Routine >(void *data, unsigned int size)
```

パラメータ

data - HID クラスドライバから受信したデータを指すポインタ

size - 引き渡されたデータのサイズ (byte)

戻り値

TRUE (処理に成功した)、FALSE (処理に失敗した)

前提条件

HID ソースファイル内の `_HidReportHandler` 構造体に含まれている事

副次的影響

なし

AN1163

Get Report ルーチン

このルーチンは現在の入力レポートを取得します。

構文

```
UINT < Get Report Routine>(void *data)
```

パラメータ

data - このルーチン用にレポートを格納するデータバッファを指すポインタ

戻り値

0 の場合は ERROR、0 以外の値はレポートのサイズ (byte) を表す

前提条件

HID ソースファイル内の `_HidReportHandler` 構造体に含まれている事

副次的影響

なし

補遺 C: HID ファンクション ドライバ マクロ

本書の HID アプリケーションは、カスタマイズ用に各種のファンクション ドライバマクロを提供します。

以下の HID ファンクション ドライバマクロが利用できません。

- mHIDOpenTimer
- HID_TIMER_CONFIG
- mHIDConfigIntTimer
- HID_INT_CONFIG
- mHIDEnableTimerInt
- mHIDDisableTimerInt
- mHIDTimerIntHandler
- mHIDClearInt
- HID_EP_IN
- HID_EP_IN_SIZE
- HID_MAX_REPORT_SIZE
- HID_NUM_REPORTS

AN1163

mHIDOpenTimer

目的： タイマのペリフェラル ライブラリマクロを使ってタイマを開きます。
既定値： OpenTimer1

HID_TIMER_CONFIG

目的： タイマ設定値
既定値： (T1_ON)

mHIDConfigIntTimer

目的： タイマのペリフェラル ライブラリマクロを使ってタイマ割り込みを設定します。
既定値： ConfigIntTimer1

HID_INT_CONFIG

目的： タイマ割り込み設定値
既定値： (T1_INT_OFF | T1_INT_PRIOR_2 | T1_INT_SUB_PRIOR_1)

mHIDEnableTimerInt

目的： タイマのペリフェラル ライブラリマクロを使ってタイマ割り込みを有効にします。
既定値： EnableIntT1

mHIDDisableTimerInt

目的： タイマのペリフェラル ライブラリマクロを使ってタイマ割り込みを無効にします。
既定値： DisableIntT1

mHIDTimerIntHandler

目的： 割り込みハンドラ関数を定義します。
既定値： Timer1IntHandler

mHIDClearInt

目的： タイマのペリフェラル ライブラリマクロを使ってタイマ割り込みフラグをクリアします。
既定値： mT1ClearIntFlag

HID_EP_IN

目的： HID ファンクション ドライバが入力レポートを送信する際に送信元となるエンドポイント
既定値： 1

HID_EP_IN_SIZE

目的： HID ファンクション ドライバ エンドポイントのサイズ (byte)
既定値： 16

HID_MAX_REPORT_SIZE

目的： レポート構造体の総サイズ (byte)
既定値： 64

HID_NUM_REPORTS

目的： サポートする HID レポートの数
既定値： 1

補遺 D: USB HID ファンクション ドライバ API

以下では、HID ファンクション ドライバ API について説明します。

API - HIDSendReport

このルーチンは入力レポートを HID クラスドライバへ送信します。

構文

```
BOOL HidSendReport(BYTE id, void *data, UINT size, BOOL change)
```

パラメータ

id - レポート ID (レポート ID が存在しない場合、値は 0)

data - 送信するレポートデータを指すポインタ

size - 送信データのサイズ

change - データが変更されたかどうかを示す

戻り値

TRUE (データ送信要求は処理された)、FALSE (処理に失敗した)

前提条件

USB デバイスタックと HID が初期化済みである事

副次的影響

なし

例

```
// send a report
BYTE report[4];
report[0] = 'M';
report[1] = 'C';
report[2] = 'H';
report[3] = 'P';
if(!HidSendReport(0, report, 4, FALSE))
{
    // handle send error
}
```

補遺 E: USB ディスクリプタ テーブル定義

HID ファンクション ドライバは、「HID ディスクリプタ」で説明したように、下表の値を使って自身のディスクリプタ テーブルを定義します。

表 E-1: デバイス ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bLength	このディスクリプタのサイズ	12h
bDescriptorType	タイプ (常に USB デバイス ディスクリプタ)	1
bcdUSB	USB 仕様バージョン (BCD)	200h
bDeviceClass	デバイスクラス コード	0
bDeviceSubClass	デバイス サブクラス コード	0
bDeviceProtocol	デバイス プロトコル	0
bMaxPacketSize	エンドポイント 0 の最大パケットサイズ	10h
idVendor	ベンダー ID (VID)	4D8h
idProduct	製品 ID (PID)	Dh
bcdDevice	デバイス リリース番号 (BCD)	1
iManufacturer	製造者名ストリング ディスクリプタのインデックス	1
iProduct	製品を示すストリング ディスクリプタのインデックス	2
iSerialNum	製品シリアル番号ストリング ディスクリプタのインデックス	0
bNumConfigurations	サポートするコンフィグレーションの数	1

表 E-2: コンフィグレーション ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bLength	このディスクリプタのサイズ	9
bDescriptorType	タイプ (常に USB コンフィグレーション ディスクリプタ)	2
wTotalLength	全てのディスクリプタの総長	22h
bNumInterfaces	インターフェイスの数	1
bConfiguration/Value	ID 値	1
iConfiguration	ストリング ディスクリプタのインデックス	0
bmAttributes		
reserved_zero	常に 0	0
remote_waking	1: デバイスはリモート復帰をサポートする	1
self_powered	1: デバイスは電源を内蔵する	0
reserved_one	常に 1	1
bMaxPower	mA/2 (例: 100 mA = 50)	50

表 E-3: インターフェイス ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bLength	このディスクリプタのサイズ	9
bDescriptorType	タイプ(常にUSBインターフェイス ディスクリプタ)	4
bInterfaceNumber	インターフェイスの ID 番号	0
bAlternateSetting	代替インターフェイス設定の ID 番号	0
bNumEndpoints	このインターフェイス内のエンドポイントの数	1
bInterfaceClass	USB インターフェイス クラス ID	3
bInterfaceSubClass	USB インターフェイス サブクラス ID	1
bInterfaceProtocol	USB インターフェイス プロトコル ID	2
iInterface	インターフェイスのストリング ディスクリプタのインデックス	0

表 E-4: HID インターフェイス ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bLength	このディスクリプタのサイズ	9
bDescriptorType	タイプ(常に USB HID インターフェイス ディスクリプタ)	21h
bcdHID	HID 仕様リリース (BCD)	101h
bCountryCode	ローカライズされたハードウェアの国コード	0
bNumDescriptors	クラス ディスクリプタの数 (1 以上)	1

表 E-5: HID クラス インターフェイス ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bDescriptorType	クラス ディスクリプタの ID タイプ	22h
bDescriptorLength	ディスクリプタ(すなわちHIDレポート)の総サイズ	32h

表 E-6: エンドポイント ディスクリプタ

フィールド	概要	HID ファンクション ドライバ値
bLength	このディスクリプタのサイズ	7
bDescriptorType	タイプ(常に USB エンドポイント ディスクリプタ)	5
bEndpointAddress		
ep_num	エンドポイント番号	1
reserved	常に 0	0
direction	IN または OUT	1
bmAttribute		
transfer_type	転送タイプ	1
synch_type	同期タイプ	0
usage_type	使用タイプ	0
reserved	常に 0	0
wMaxPacketSize	エンドポイントが処理可能な最大パケットサイズ	16
bInterval	このエンドポイントのデータをポーリングする間隔	10

AN1163

表 E-7: シリアル番号STRING ディスクリプタ

フィールド	概要	HID ファンクションドライバ値
bLength	このディスクリプタのサイズ	4h
bDescriptorType	タイプ (常に USB STRING ディスクリプタ)	3
wLangid[]	文字列	409h

表 E-8: 製造者STRING ディスクリプタ

フィールド	概要	HID ファンクションドライバ値
bLength	このディスクリプタのサイズ	34h
bDescriptorType	タイプ (常に USB STRING ディスクリプタ)	3
wLangid[]	文字列	Microchip Technology Inc.

表 E-9: 製品STRING ディスクリプタ

フィールド	概要	HID ファンクションドライバ値
bLength	このディスクリプタのサイズ	2Eh
bDescriptorType	タイプ (常に USB STRING ディスクリプタ)	3
wLangid[]	文字列	Mouse In a Circle Demo

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") is intended and supplied to you, the Company's customer, for use solely and exclusively with products manufactured by the Company.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

補遺 F: HID ファンクション ドライバの ソースコード

本書で説明したHIDファンクションドライバのソースコードは、無償のライセンス使用許諾契約の下に提供されます。全ての内容を収めたアーカイブファイルは、以下の Microchip 社ウェブサイトからダウンロードできます：

www.microchip.com

ダウンロード後に、リリースノート記載されている最新リビジョンレベルとソフトウェアの変更履歴を必ず確認してください。

AN1163

改訂履歴

リビジョン A (2008 年 02 月)

本書は初版です。

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、PIC³² 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、rfLAB、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: 978-1-63276-124-8

**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[®] MCU および dsPIC[®] DSC、KEELOQ[®] 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。

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

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 - ウォーキングガム
Tel:44-118-921-5800
Fax:44-118-921-5820