

# USBアプリを作る際に 知っておきたい通信の基礎

関本 健太郎

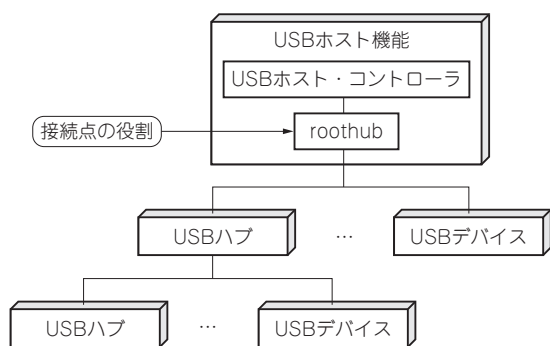


図1 USB機器の物理的な接続形態

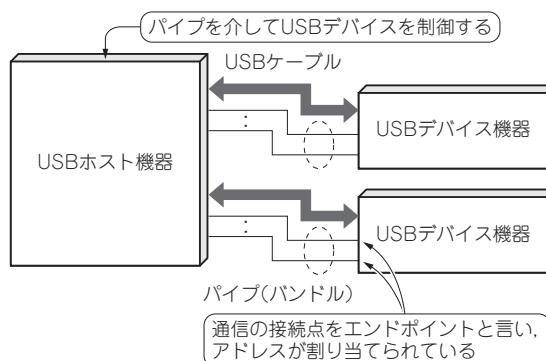


図2 USB機器の論理的な通信インターフェース…パイプ

この章では、USBアプリケーションを作成する上で知っておきたいUSB通信の概要を説明します。USBホストとUSBデバイスは、USBケーブルを通して、USBプロトコル仕様に準拠したシリアル通信でデータの転送を行います。

### 接続

#### ● 物理的な接続

USBホスト機器の内部には、USBホスト・コントローラと、USBハブやUSBデバイスの接続点となるroothubがあります(図1)。roothubには、USBハブやUSBデバイス機器が接続されます。

USBハブはさらに、USBハブやUSBデバイス機器をカスケード接続できます。最大接続数はUSBの仕様で規定されており、接続されているUSB機器のハードウェア、ファームウェアによっても制限されます。

#### ● 論理的な接続

データ通信は、パイプと呼ばれる論理的な通信インターフェースを介して行われます(図2)。USBホストは、パイプを介して、接続されたUSBデバイスの制御を行います。

図3は、右側でUSB通信モデルの基本的な流れと相互関係を、左側でホスト部分の詳細を示しています。

USBホスト側のコンポーネントは次の3つに分類されています。

- クライアント
- USBシステム(ホスト・ドライバ)
- USB BUSインターフェース(ホスト・ハードウェア)

USBシステム(ホスト・ドライバ)部は、さらに以下の3つに分類されます。

- ① USBホスト・コントローラ・ドライバ
- ② USBドライバ
- ③ USBホスト・ソフトウェア(アプリケーション)

#### ● USBホスト・コントローラ・ドライバの基本機能

図3中の①USBホスト・コントローラ・ドライバは、次の機能を提供します。

- USBデバイスの接続と取り外しの管理
- USBデバイスとの通信手順の管理
- USBデバイスとのデータの転送処理
- ステータスの収集と統計情報の管理
- 電源情報を含めたUSBデバイスとの物理的なインターフェースの管理

デバイスの検出処理(enumeration)とデバイスの設定は、デフォルトのパイプ注1を通して行われます。

注1: 全てのUSBデバイスはUSB通信の接続点となるエンドポイントを持ち、その1つにデフォルト・エンドポイント(エンドポイント・アドレス0)があります。デフォルトのエンドポイントを介した論理的な通信経路をデフォルトのパイプと呼びます。

# 特集 USBホスト&デバイス ラズパイPico 虎の巻

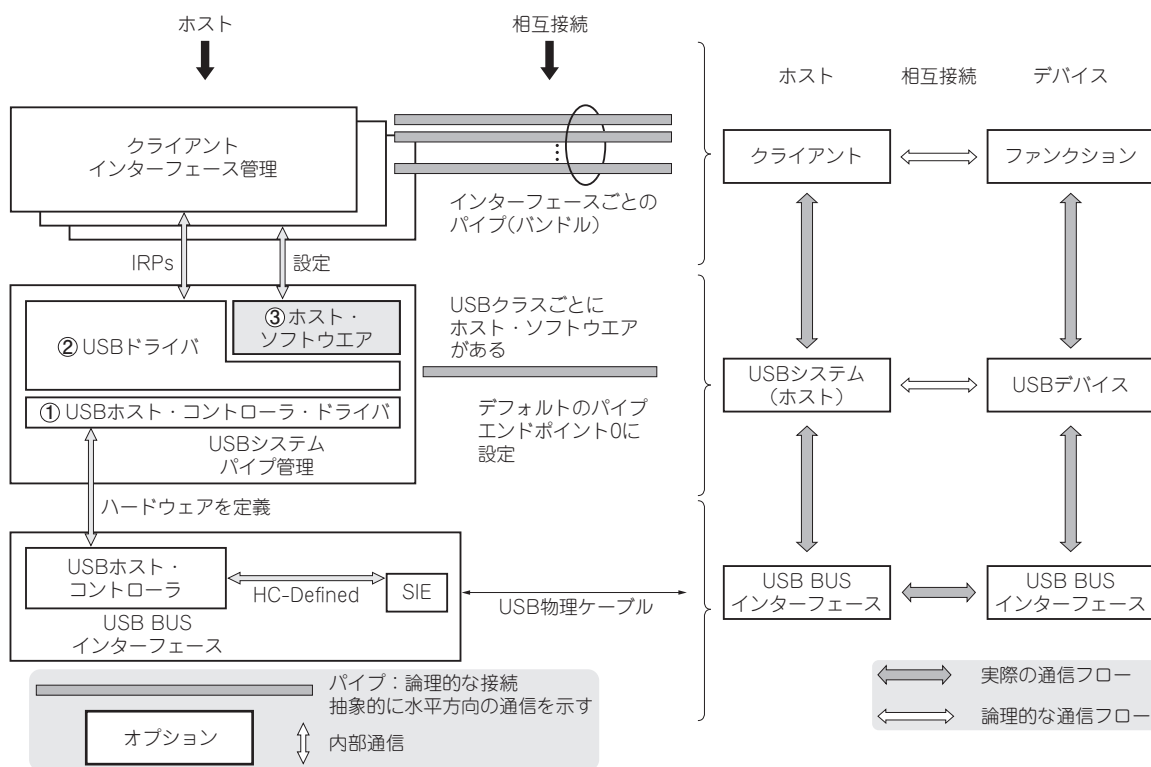


図3 USBホスト内部処理および論理的な通信フロー  
USB 2.0仕様書 10章図10-1, 図10-2からの抜粋

## USB通信の転送タイプ

USBのデータ転送のタイプは次の4つです。USBのデバイスは、この4つの転送タイプを特定の用途に応じ、選択し、目的の機能を実装します。

### ● コントロール転送

急激に負荷の高い状態となる、あるいは非定期的に発生する、ホストのソフトウェアによって開始される要求/応答通信です。通常、コマンド/ステータス操作に使われます。

### ● アイソクロナス転送

ホストとデバイス間の定期的で継続的な通信に利用されます。通常、時間的な制限を持つ情報に使われます。この転送タイプは、データ中に時間の概念も保持します。ただし、これはデータの配信ニーズが常にタイム・クリティカルであることを意味するものではありません。

### ● インタラプト転送

低頻度、あるいは遅延制限のある通信に利用されます。

### ● バルク転送

非周期的で大規模なパケットの高速通信に利用されます。通常、利用帯域幅を可変にできるデータに使用されます。帯域幅が使えるようになるまで、通信を遅らせることもできます。

## USBデバイスのフレームワーク

### ● USBディスクリプタ…デバイスの機能をUSBホストに報告する

USBデバイスは、ディスクリプタ(記述子)を使ってそのデバイスの機能をUSBホストに報告します。ディスクリプタは、USBの規格で定義された形式のデータ構造のことです。

各ディスクリプタは、ディスクリプタの合計バイト数を含むバイト幅のフィールドで始まり、その後にディスクリプタのタイプを識別するバイト幅のフィールドが続きます。USBデバイスは、次の2つの方法のどちらかでクラスまたはベンダ独自のディスクリプタを返します。

#### ▶ 標準ディスクリプタ

USBデバイスが、クラスまたはベンダ独自のディスクリプタとして、標準ディスクリプタと同じ形式を

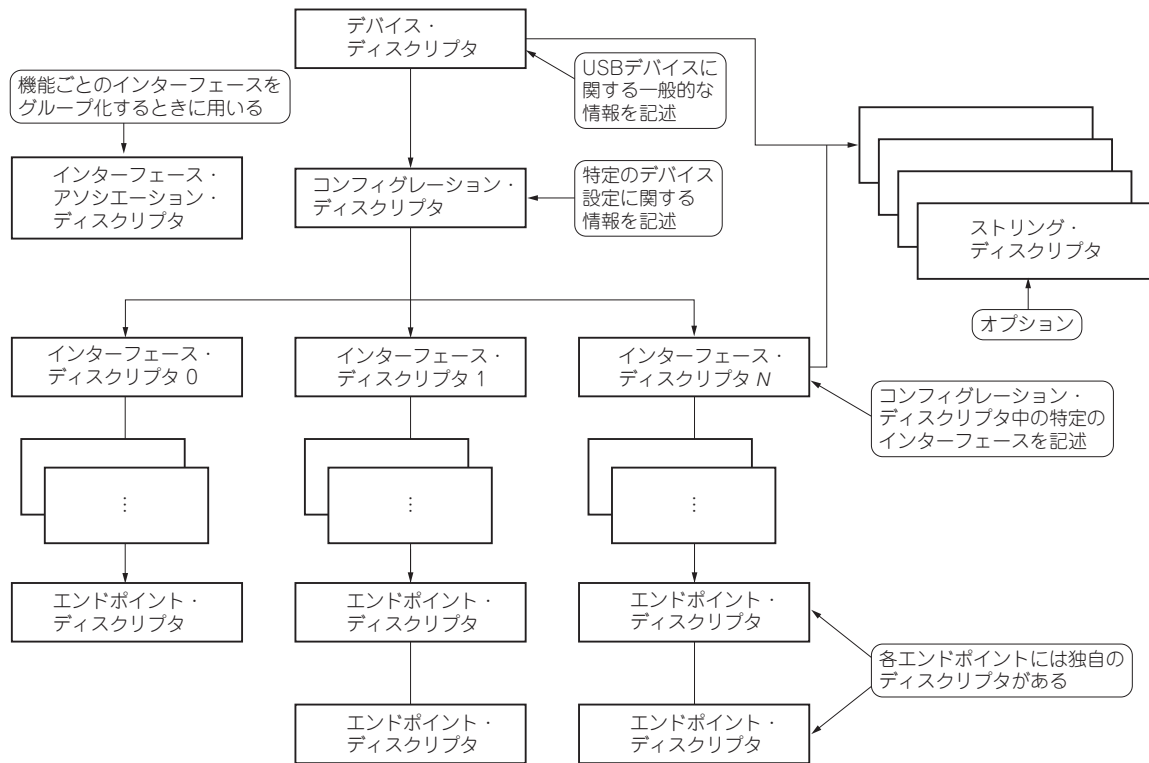


図4 USBデバイスのディスクリプタ構成例

使用する(ディスクリプタの先頭が長さバイトで始まり、タイプ・バイトが続く形式)場合、USBホストはGetDescriptor (Configuration) リクエストを行い、構成情報を標準ディスクリプタに従った形式で取得します。

▶非標準ディスクリプタ

USBデバイスがクラスまたはベンダ独自のディスクリプタとして、標準ディスクリプタと異なる形式を使用する場合、USBホストはクラスまたはベンダ独自のディスクリプタのタイプとインデックスを指定したGetDescriptorリクエストを行い、デバイスからディスクリプタを取得します。

●標準ディスクリプタを詳しく見てみる

USBデバイスの標準ディスクリプタは以下の要素から構成されます(図4)。USBデバイスは必ずこの形式に沿ったディスクリプタを保持しており、USBデバイスの機能を表現する必要があります。

- デバイス
- デバイス・クオリファイア (Device Qualifier, USB 2.0より追加)
- インターフェース・アソシエーション
- コンフィグレーション

- インターフェース
  - エンドポイント
  - ストリング
- <デバイス拡張情報>
- BOS (Binary Object Store, USB 2.1より追加)
  - デバイス・ケイパビリティ (USB 2.1より追加)

▶デバイス

デバイス・ディスクリプタは、USBデバイスに関する一般的な情報を記述しています。これにはデバイスおよびデバイスの全ての構成にグローバルに適用される情報が含まれています(表1)。USBデバイスにはデバイス・ディスクリプタが1つだけあります。

▶コンフィグレーション

コンフィグレーション・ディスクリプタは、特定のデバイス設定に関する情報を記述しています(表2)。

▶インターフェース

インターフェース・ディスクリプタは、コンフィグレーション中の特定のインターフェースを記述しています(表3)。

▶インターフェース・アソシエーション

機能ごとのインターフェースをグループ化する(コンボ・デバイスを作成する)際に用います(表4)。

**特集**

**第1部**  
**USBマイコン**  
**PiCO基礎知識**

**第2部**  
**役立ちサンプル**  
**徹底解説**

**第3部**  
**USBデバイス製作集**

**第4部**  
**USBホスト製作集**

# 特集 USBホスト&デバイス ラズパイPico 虎の巻

表1 デバイス・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	デバイス・ディスクリプタのタイプ
2	bcdUSB	2	BCD	USBの仕様のバージョン (BSD形式)
4	bDeviceClass	1	Class	クラス・コード (0xffならベンダ独自クラス)
5	bDeviceSubClass	1	SubClass	サブクラス・コード
6	bDeviceProtocol	1	Protocol	プロトコル・コード
7	bMaxPacketSize0	1	Number	エンドポイントの最大パケット・サイズ
8	idVendor	2	ID	ベンダID
10	idProduct	2	ID	プロダクトID
12	bcdDevice	2	BCD	デバイスのリリース番号 (BCD形式)
14	iManufacturer	1	Index	製造元のストリング・ディスクリプタのインデックス
15	iProduct	1	Index	製品のストリング・ディスクリプタのインデックス
16	iSerialNumber	1	Index	シリアル番号のストリング・ディスクリプタのインデックス
17	bNumConfigurations	1	Index	可能性のあるコンフィグレーションの数

表2 コンフィグレーション・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	コンフィグレーション・ディスクリプタのタイプ
2	wTotalLength	2	Number	この設定のトータル・データ・サイズ
4	bNumInterfaces	1	Number	この設定でサポートされるインターフェースの数
5	bConfigurationValue	1	Number	この設定を選択するために SetConfiguration() への引数として使用される値
6	iConfiguration	1	Index	ストリング・ディスクリプタのインデックス
7	bmAttribute	1	Bitmap	設定の特性. ビット7: 予約, ビット6: セルフ・パワー, ビット5: リモート・ウェイクアップ, ビット4~0: 予約
8	bMaxPower	1	mA	USBデバイスでの最大電力消費 (2mA単位)

表3 インターフェース・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	インターフェース・ディスクリプタのタイプ
2	bInterfaceNumber	1	Number	インターフェースの数
3	bAlternateSetting	1	Number	別の設定を選択するための値
4	bNumEndpoints	1	Number	このインターフェースで使用されるエンドポイントの数
5	bInterfaceClass	1	Class	クラス・コード (0xffならベンダ独自クラス)
6	bInterfaceSubClass	1	SubClass	サブクラス・コード
7	bInterfaceProtocol	1	Protocol	プロトコル・コード
8	iInterface	1	Index	ストリング・ディスクリプタのインデックス

表4 インターフェース・アソシエーション・ディスクリプタ

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	インターフェース・ディスクリプタのタイプ
2	bFirstInterface	1	Number	最初のインターフェース
3	bInterfaceCount	1	Number	インターフェースの数
5	bFunctionClass	1	Class	クラス・コード (0xffならベンダ独自クラス)
6	bFunctionSubClass	1	SubClass	サブクラス・コード
7	bFunctionProtocol	1	Protocol	プロトコル・コード
8	iFunction	1	Index	ストリング・ディスクリプタのインデックス

## 第2章 USBアプリを作る際に知っておきたい通信の基礎

表5 エンドポイント・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	エンドポイント・ディスクリプタのタイプ
2	bEndpointAddress	1	Endpoint	エンドポイント・アドレス
				ビット3～0: エンドポイント番号
				ビット6～4: 予約(0) ビット7: 0=Out エンドポイント, 1=In エンドポイント
3	bmAttributes	1	Bitmap	ビット1～0: 転送タイプ
				ビット3～2: 同期タイプ
				ビット5～4: 使用タイプ
4	wMaxPacketSize	2	Number	エンドポイントの最大パケット・サイズ
6	bInterval	1	Number	エンドポイントのデータ転送のポーリング間隔

表6 スtring・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	$N + 2$	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	String・ディスクリプタのタイプ
2	wLANGID [0]	2	Number	LANGID コード0
:	:	:	:	
$N$	wLANGID [ $x$ ]	2	Number	LANGID コード $x$

表7 BOSディスクリプタ形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ(5バイト)
1	bDescriptorType	1	Constant	BOSディスクリプタのタイプ
2	wTotalLength	2	Number	この設定のトータル・データ・サイズ (BOSディスクリプタ・サイズと下層のデバイス・ケイパビリティ・ ディスクリプタのサイズの合計)
3	bNumDeviceCaps	1	Constant	予約(0)

表8 プラットフォーム・ディスクリプタの形式

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	デバイス・ケイパビリティ・ディスクリプタのタイプ
2	bDevCapabilityType	1	Number	プラットフォーム・ケイパビリティ・タイプ
3	bReserved	1	Constant	予約(0)
4	PlatformCapabilityUUID	16	Number	デバイスのプラットフォーム・ケイパビリティを示す128ビットの UUID値
20	CapabilityData	可変	Binary	プラットフォームに特化した機能に関連づけられているデータを含む 可変長のフィールド。ゼロの場合もある

### ▶ エンドポイント

エンドポイント・ディスクリプタのインターフェースで使用される各エンドポイントには、独自のディスクリプタがあります(表5)。

### ▶ String

String・ディスクリプタはオプションです(表6)。デバイスがString・ディスクリプタをサポートしない場合、String・ディスクリプタへのインデックスは全てゼロにリセットされます。

### ▶ BOS (Binary Object Store)

BOSディスクリプタは、USBデバイスのデバイス・レベルの機能のディスクリプタ(デバイス・ケイパビリティ・ディスクリプタ)を子を持つディスクリプタです(表7)。

### ▶ デバイス・ケイパビリティ(プラットフォーム・ディスクリプタ)

デバイス・ケイパビリティ・ディスクリプタの仕様は、USB 2.1で使用できます。プラットフォーム/OSベンダによって個別に定義および公開され、固有のプ

特集

第1部

USBマイコン  
PIC基礎知識

第2部

役立ちサンプル  
徹底解説

第3部

USBデバイス製作集

第4部

USBホスト製作集

# 特集 USBホスト&デバイス ラズパイPico 虎の巻

表9 WebUSBデバイス・ケイパビリティ・ディスクリプタ

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ (24バイト)
1	bDescriptorType	1	Constant	デバイス・ケイパビリティ・ディスクリプタのタイプ
2	bDevCapabilityType	1	Number	プラットフォーム・ケイパビリティ・タイプ
3	bReserved	1	Constant	予約 (0)
4	PlatformCapabilityUUID	16	Number	{3408b638-09a9-47a0-8bfd-a0768815b665}
20	bcdVersion	2	BCD	プロトコル・バージョン (0x0100)
22	bVendorCode	2	Number	WebUSBリクエストを行うときのbRequestの値
23	iLandingPage	1	Number	デバイス・ランディング・ページのURLディスクリプタ・インデックス

割り当てられたUUID値  
表13のセットアップ・パケットの一部

表10 Microsoft OS 2.0デバイス・ケイパビリティ・ディスクリプタ

オフセット	フィールド	サイズ	値の形式	説明
0	bLength	1	Number	このディスクリプタのバイト・サイズ
1	bDescriptorType	1	Constant	デバイス・ケイパビリティ・ディスクリプタのタイプ
2	bDevCapabilityType	1	Number	プラットフォーム・ケイパビリティ・タイプ
3	bReserved	1	Constant	予約 (0)
4	PlatformCapabilityUUID	16	Number	{D8DD60DF-4589-4CC7-9CD2-659D9E648A9F}
20	CapabilityData	可変	Binary	プラットフォームに特化した機能に関連づけられているデータを含む可変長のフィールドでゼロの場合もある。詳細はMicrosoft OS 2.0 descriptor capability dataを参照

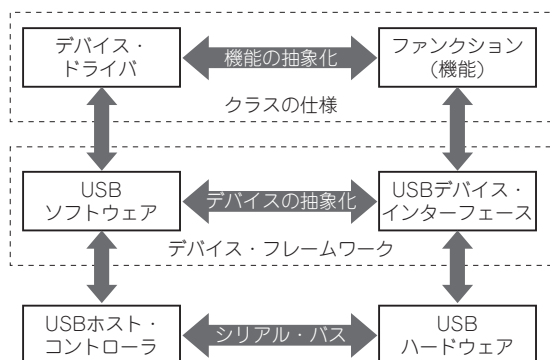


図5 USBホスト-デバイス間の層ごとの通信

ます。それぞれのクラスの仕様が定義されており、その仕様に従ってUSBデバイスの機能が実装され、USBホストとUSBデバイスはその仕様に従って通信します(図5)。

USBホストは、USBデバイス・クラス情報に基づいて、USBデバイスの機能を識別し、その機能に基づいてそのデバイスのドライバを実行します。この情報は、ベースクラス、サブクラスおよびプロトコルの3つ(各1バイト、トータル3バイト)から構成されています。

USBデバイス・クラスは、USBデバイスのディスクリプタ中の2カ所で指定されます。1つはデバイス・ディスクリプタにあり、もう1つはインターフェース・ディスクリプタにあります。定義されたクラス・コードには、

- デバイス・ディスクリプタでのみ使用できるもの
- デバイス・ディスクリプタとインターフェース・ディスクリプタの両方で使用できるもの
- インターフェース・ディスクリプタでのみ使用できるもの

があります。表11は、2022年1月現在、定義されているベースクラス値、一般的な用途、およびそのベースクラスの値を指定するディスクリプタ(デバイス・

プラットフォーム、固有のデバイス機能を識別するために使用される128ビットのUUID値が含まれています(表8)。ディスクリプタには、機能に関連付けられた1バイト以上のデータが含まれる場合もあります。WebUSBディスクリプタ(表9)やMicrosoft OS 2.0ディスクリプタ(表10)などがあります。第2部第11章のwebusb\_serialのサンプルで利用されています。

## ● USBデバイス・クラス

USBクラス<sup>注2</sup>は、同様の属性またはサービスを持つデバイスまたはインターフェースのグループを指し

注2: デバイス、ホストに依存しないコンテキストでUSBクラスとしました。

## 第2章 USBアプリを作る際に知っておきたい通信の基礎

表11 USBデバイス・クラスの種類

<https://www.usb.org/defined-class-codes>

ベース・クラス	使用ディスクリプタ	用途
00h	デバイス	インターフェースを記述するときのクラス情報に使う
01h	インターフェース	オーディオ
02h	デバイス/インターフェース	通信とCDCコントロール
03h	インターフェース	HID (ヒューマン・インターフェース・デバイス)
05h	インターフェース	物理
06h	インターフェース	画像
07h	インターフェース	プリンタ
08h	インターフェース	大容量記憶装置
09h	デバイス	ハブ
0Ah	インターフェース	CDCデータ
0Bh	インターフェース	スマート・カード
0Dh	インターフェース	コンテンツ・セキュリティ
0Eh	インターフェース	動画
0Fh	インターフェース	健康管理
10h	インターフェース	音声/動画 デバイス
11h	デバイス	USB (ビルボード・デバイス・クラス)
12h	インターフェース	USB (Type-Cブリッジ・クラス)
3Ch	インターフェース	I3C デバイス・クラス
DCh	デバイス/インターフェース	診断装置
E0h	インターフェース	ワイヤレス・コントローラ
EFh	デバイス/インターフェース	さまざまな用途
FEh	インターフェース	何かに特化した用途
FFh	デバイス/インターフェース	ベンダ固有の用途

ディスクリプタまたはインターフェース・ディスクリプタ、あるいはその両方を示しています。

### USBパケット

#### ● 3種類のパケットがある

全てのUSBパケットは、SYNCフィールドから始まり、PIDフィールドへと続きます(図6)。

パケットのタイプによって、PIDフィールドの後続のフォーマットが異なります。USBパケットには、トークン、データ、ハンドシェイクの3種類があります。

##### ▶ トークン

トークン・パケットは、SYNCフィールド、PIDフィールド、アドレス・フィールド、エンドポイント・フィールドと続き、最後にアドレス・フィールドとエンドポイント・フィールドのCRCが付加されます。PIDフィールドは、IN、OUT、SETUPあるいは

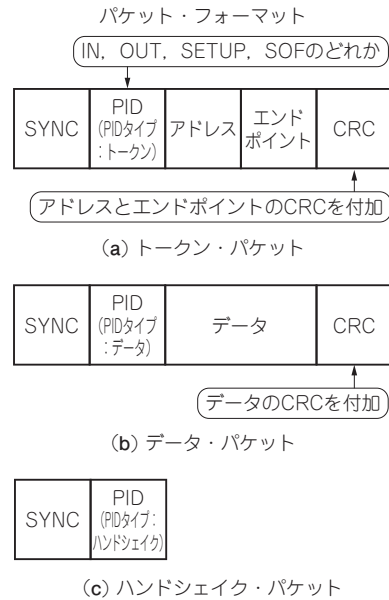


図6 全てのUSBパケットはSYNCフィールドからPIDフィールドへ続く

SOFのどれかになります(表12)。

##### ▶ データ

データ・パケットは、SYNCフィールド、PIDフィールド、データ・フィールド、CRCとなります。PIDフィールドは、DATA0、DATA1、DATA2、MDATAのどれかになります(表12)。

##### ▶ ハンドシェイク

ハンドシェイク・パケットは、SYNCフィールド、PIDフィールドで構成されます。

#### ● データ転送の際のパケット例

USBのデータ転送は、前述のパケット・タイプを定義された順番で指定することで実行されます。例えばコントロール転送のホストからデバイス方向のデータの場合には、セットアップ・ステージ、データ・ステージ、ステータス・ステージの3ステージを経て、転送が完了します。図7の例は、USBデバイスをホストに接続した際に、ホストからデバイスに対して、最初に行われるUSBデバイス・ディスクリプタのリクエストの例です。

セットアップ・ステージでは、ホストからデバイスへ、デバイスのアドレスとエンドポイント番号を指定したトークン・パケットを送信し、続いて、bRequestにGET\_DESCRIPTORを指定したセットアップ・パケット形式のデータ・パケットを送信し、最後にデバイスからホストにACKのハンドシェイク・パケットを返します。

データ・ステージでは、トークン・パケットをホス

特集

第1部

USBマイコン  
PIC基礎知識

第2部

役立ちサンプル  
徹底解説

第3部

USBデバイス製作集

第4部

USBホスト製作集

# 特集 USBホスト&デバイス ラズパイPico虎の巻

## コラム USBディスクリプタの詳細を表示するツール

関本 健太郎

### ● Windows…UsbTreeView.exe

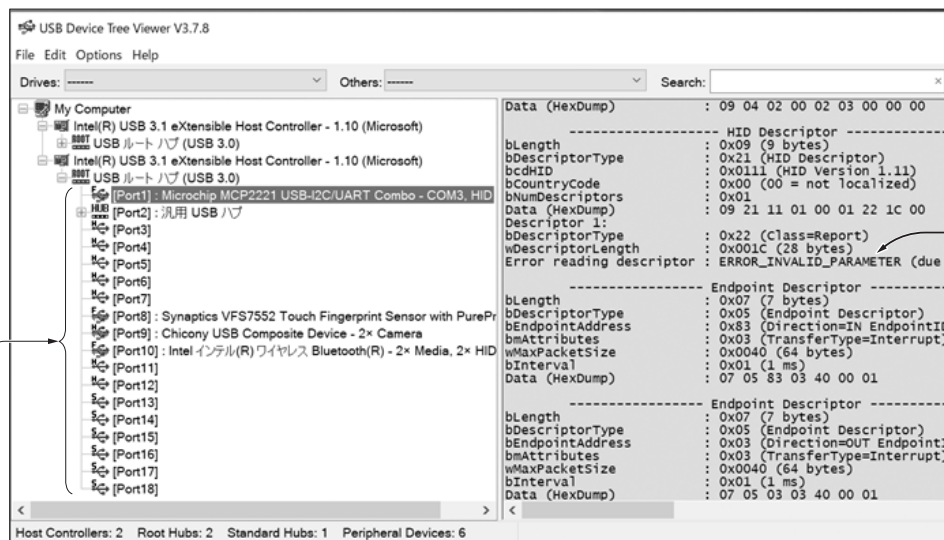
筆者お勧めのUSBディスクリプタ詳細表示ツールを紹介します。Windows環境にはUsbTreeView.exeというツールが公開されています。

[https://www.uwe-sieber.de/usbtreeview\\_e.html](https://www.uwe-sieber.de/usbtreeview_e.html)

図Aの例では、第3部で紹介するMCP2221 USB-I<sup>2</sup>C/UART Comboという製品のUSBディスクリプタ情報を取得しています。このツールを利用すると、ホストPCに接続されるUSBデバイスの詳細情報をツリー形式で表示できます。ただし、レポート・ディスクリプタについては、ERROR\_INVALID\_PARAMETERで詳細情報は取得できませんでした。

### ● Linux…lsusb -v

Linuxでは、lsusb -vでディスクリプタの詳細を表示することはできますが、後述するレポート・ディスクリプタの内容までは表示できない場合があります。その場合にはusbhid-dumpコマンドを利用するとよいでしょう。lsusb -vでは、レポート・ディスクリプタの情報を取得できませんでしたが(rootでデバイスのunbindを行えばlsusbコマンドでも表示可能)、usbhid-dumpコマンド(usbhid-dump -a <バス番号>:<デバイス番号>またはusbhid-dump -d <vid>:<pid>)を使うと、レポート・ディスクリプタの情報を表示できます(リストA)。



ホストPCに接続されているUSBデバイスの情報をツリー形式で表示できる

レポート・ディスクリプタの情報は取得できない

図A UsbTreeViewで「MCP2221 USB-I<sup>2</sup>C/UART Combo」のディスクリプタ情報を取得する

トからデバイスへ、デバイスのアドレスとエンドポイント番号を指定したトークン・パケットを送信し、デバイスからホストにデバイス・ディスクリプタを格納したデータ・パケットを送信し、最後にホストからデバイスにACKのハンドシェイク・パケットを返します。

ステータス・ステージでは、ホストからデバイスにデータ・パケットのサイズがゼロとなるデータ・パケットを送信し、最後にデバイスからホストにACKのハンドシェイク・パケットを返します。

## USBデバイス・リクエスト

### ● セットアップ・パケット

USBホストは、USBデバイスのコントロール・パイプに対して、コントロール転送でリクエストを行います。リクエストは8バイトのセットアップ・パケットに格納して行います(表13)。

例えば、GET\_DESCRIPTOR(表14)の場合、セットアップ・パケットは80 06 00 01 00 00 40 00となり



## 第2章 USBアプリを作る際に知っておきたい通信の基礎

リストA lsusbおよびusbhid-dumpによるMCP2221 USB-I2C/UART Comboのディスクリプタ表示

```
// lsusbコマンドでUSBディスクリプタを取得する
lsusb -t
:
Bus 001 Device 008: ID 04d8:00dd Microchip
Technology, Inc. MCP2221 USB-I2C/UART Combo
Device Descriptor:
  bLength 18
  bDescriptorType 1
  bcdUSB 2.00
  bDeviceClass 239 Miscellaneous Device
  bDeviceSubClass 2
  bDeviceProtocol 1 Interface Association
  bMaxPacketSize0 8
  idVendor 0x04d8 Microchip Technology, Inc.
  idProduct 0x00dd
  bcdDevice 1.00
  iManufacturer 1 Microchip Technology Inc.
  iProduct 2 MCP2221 USB-I2C/UART Combo
  iSerial 0
  bNumConfigurations 1
  :
HID Device Descriptor: // HIDディスクリプタ
  bLength 9
  bDescriptorType 33
  bcdHID 1.11
  bCountryCode 0 Not supported
  bNumDescriptors 1
  bDescriptorType 34 Report
  wDescriptorLength 28 // レポート・ディスクリプタを取得できない
Report Descriptors: ** UNAVAILABLE **
Endpoint Descriptor:
  :
Endpoint Descriptor:
  bLength 7
  bDescriptorType 5
  bEndpointAddress 0x03 EP 3 OUT
  bmAttributes 3
  Transfer Type Interrupt
  Synch Type None
  Usage Type Data
  wMaxPacketSize 0x0040 1x 64 bytes
  bInterval 1
  Device Status: 0x0000
  (Bus Powered)
// usbhid-dump \コマンドでレポート・ディスクリプタを取得する
sudo usbhid-dump -a 1:8
001:008:002:DESCRIPTOR 1656147487.964654 06 00 FF
09 01 A1 01 19 01 29 40 15 01 25 40 75 08 95 40 81
00 19 01 29 40 91 00 C0
```

表12 PIDフィールドの詳細

PIDタイプ	PID名	PID<3:0>	説明
トークン	OUT	0001B	アドレスとエンドポイントの番号(ホスト→デバイス)
	IN	1001B	アドレスとエンドポイントの番号(デバイス→ホスト)
	SOF	0101B	フレーム開始マークとフレーム番号
	SETUP	1101B	コントロール転送のSETUPフェーズ、アドレスとエンドポイントの番号(ホスト→デバイス)
データ	DATA0	0011B	データ・パケットのPID偶数
	DATA1	1011B	データ・パケットのPID奇数
	DATA2	0111B	データ・パケットのPIDハイスピード、ハイバンド、アイソクロナス転送
	MDATA	1111B	データ・パケットのPIDハイスピード、スプリットかつハイバンド、アイソクロナス転送
ハンドシェイク	ACK	0010B	受け取り側がエラーなしでパケットを受信
	NAK	1010B	受け取り側がデータを受信できない、または送信側がデータを送信できない
	STALL	1110B	エンドポイントがハングまたはコントロール転送が未サポート
	MYET	0100B	受信側からレスポンスがない
スペシャル	PRE	1100B	(トークン・パケットの)ホスト発行のプレアンブル下流バスはローデバイスを有効化
	ERR	1100B	(ハンドシェイク・パケットの)スプリット転送エラー
	SPLIT	1100B	(トークン・パケットの)ハイスピード・スプリット転送トークン
	PING	1000B	バルクおよびコントロール・エンドポイントでの(トークン・パケットの)ハイスピード・フロー制御プローブ
	Reserved	0000B	リザーブPID

ます。それぞれのフィールドは以下となります。

▶ **bmRequest=0x80**

デバイスからホスト方向なのでD7=1、標準タイプなのでD5～D6=0となります。デバイス・ディスクリプタなのでD0～D4=0となります。

▶ **bRequest=0x06**

GET\_DESCRIPTORの値です。

▶ **wValue=0x0100**

上位バイトのディスクリプタ・タイプはデバイスで0x01、下位バイトのディスクリプタ・インデックスは0x00です。

特集

第1部

USBマイコン  
PIC基礎知識

第2部

役立ちサンプル  
徹底解説

第3部

USBデバイス製作集

第4部

USBホスト製作集

# 特集 USBホスト&デバイス ラズパイPico 虎の巻

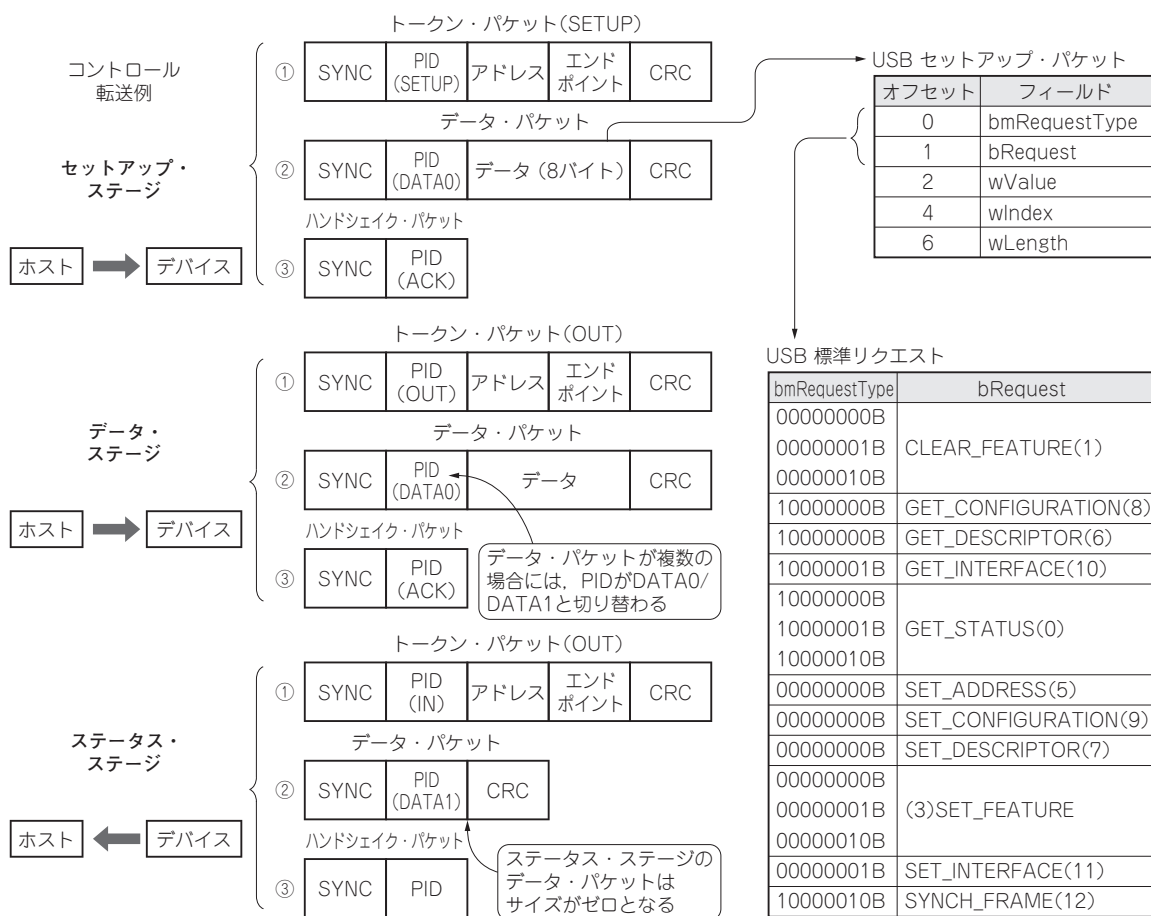


図7 コントロール転送の例 (デバイス・ディスクリプタ・リクエスト)

表13 USBデバイス・リクエストのセットアップ・パケットのフォーマット

オフセット	フィールド	サイズ	値の形式	説明
0	bmRequestType	1	Bitmap	リクエストの特性 D7: データ転送の方向 0 = ホストからデバイス 1 = デバイスからホスト D6~5: タイプ 0 = 標準 1 = クラス 2 = ベンダ 3 = 予約 D4~0: 受信者 0 = デバイス 1 = インターフェース 2 = エンドポイント 3 = その他 4~31 = 予約
1	bRequest	1	Value	リクエスト
2	wValue	2	Value	リクエストに関する値(ワード単位)
4	wIndex	1	Index of Offset	リクエストに関するインデックスまたはオフセット値(ワード単位)
6	wLength	1	Count	データ・ステージの場合には転送数(ワード単位)

▶ **wIndex=0x0000**

ゼロまたは言語IDです。

▶ **wLength=0x0040**

コントロール転送のセカンダリ・フェーズのデータ長は64バイトです。

● **標準リクエスト**

全てのUSBデバイスは、セットアップ・パケット中で指定される標準リクエスト(表14)に応答しなければなりません。標準リクエストは、bmRequest(1バイト)、bRequest(1バイト)、wValue(2バイト)、

## 第2章 USBアプリを作る際に知っておきたい通信の基礎

表14 標準リクエスト

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0000000B	CLEAR_FEATURE	機能	0	0	N1
0000001B		セクタ	インターフェース		
0000010B			エンドポイント		
1000000B	GET_CONFIGURATION	0	0	1	コンフィグレーション値
1000000B	GET_DESCRIPTOR	ディスクリプタ・タイプ とディスクリプタ・ インデックス	0または言語ID	ディスクリプタ長	ディスクリプタ
1000001B	GET_INTERFACE	0	インターフェース	1	オルタネート・ インターフェース
1000000B	GET_STATUS	0	0	2	デバイス、インターフェース、 または エンドポイント・ステータス
1000001B			インターフェース		
1000010B			エンドポイント		
0000000B	SET_ADDRESS	デバイス アドレス	0	0	N1
0000000B	SET_CONFIGURATION	コンフィグレーション 値 アドレス	0	0	N1
0000000B	SET_DESCRIPTOR	ディスクリプタ・タイプ とディスクリプタ・ インデックス	0または言語ID	ディスクリプタ長	ディスクリプタ
0000000B	SET_FEATURE	機能	0	0	N1
0000001B		セクタ	インターフェース		
0000010B			エンドポイント		
0000001B	SET_INTERFACE	オルタネート・セッテイ ング	インターフェース	0	N1
1000010B	SYNCH_FRAME	0	エンドポイント	2	フレーム・ナンバ

wIndex (2バイト), wLength (2バイト) の合計8バイトのフィールドからなり、bRequest フィールドに標準リクエストのコードを指定します。

bmRequest フィールドは表13のようにビットごとに特性が割り当てられています。wValue, wIndex, wLength のフィールドは、リクエストごとに意味が異なり、表14では表現しきれていないので、詳細はUSBの仕様書を確認する必要があります。

プロトコル・スタックのTinyUSBでは、コントロール転送が発生したときに呼び出されるコールバック関数が提供されており、USBリクエストを容易に処理できるようになっています。TinyUSBでサポートされたUSBクラスについては、多くのUSBリクエストがTinyUSBのライブラリ中で処理されており、ほとんどはTinyUSBのユーザが意識して処理を追加する必要はありません。

ただし、未サポートのUSBクラスだったり、ベンダ独自のクラスに対しては、ユーザ自身が処理する必要があります。そのような場合には、前述のUSBリクエストの形式に従って、bmRequestTypeやbRequestの値を判別して、独自の処理を追加することになります。第2部以降の活用例では、一部その処

理例を解説しました。

\* \* \*

本章では、TinyUSBを使いこなす上で必要最低限と思われる、USB通信の基礎を解説しました。さらに詳細なUSBの仕様については、参考文献に記載したUSB 2.0の仕様を参照していただくことをお勧めいたします。

#### ◆参考文献◆

- (1) USB 2.0 Specification.  
[https://www.usb.org/sites/default/files/usb\\_20\\_20211008.zip](https://www.usb.org/sites/default/files/usb_20_20211008.zip)
- (2) Microsoft OS 2.0 Descriptors Specification.  
[https://download.microsoft.com/download/3/5/6/3563ED4A-F318-4B66-A181-AB1D8F6FD42D/MS\\_OS\\_2\\_0\\_desc.docx](https://download.microsoft.com/download/3/5/6/3563ED4A-F318-4B66-A181-AB1D8F6FD42D/MS_OS_2_0_desc.docx)

せきもと・けんたろう

特集

第1部

USBマイコン  
PIC基礎知識

第2部

役立ちサンプル  
徹底解説

第3部

USBデバイス製作集

第4部

USBホスト製作集