USB概説

HERO and
heavy friends

Rev.1.0C-4/05

目 次

USB(Universal Serial Bus)とは

USB1.1は12Mbpsのフルスピード動作と1.5Mbpsのロースピード動作の2つの速度を支援します。1.5Mbps動作は低速でEMIの影響を受けにくいため、部品性能とフェライトビーズの費用を削減します。例えばクリスタル発振子がより安いセラミック振動子に置換できます。USB2.0はFirewire(IEEE1493a)に匹敵するよう480Mbpsのハイスピード動作が追加され、今日殆どのPCで採用されています。

USBのバスはホストにより制御されます。バスには唯一のホストだけが存在できます。仕様で如何なる形式での複数マスタ(ホスト)も支援しませんが、ホストの規則について2つの装置での調停を許すホスト調停規約を導入するために、USB2.0に標準でOn-The-Go仕様が追加されました。これは単一のポイント-ポイント間接続に限定した、制限されたホスト機能を意図したものです。ホストは帯域幅の使用方法と全ての処理(転送)の保証について責任があります。データは指示票(Token:トークン)を基準とした規約を使用し、様々な処理方法により送信できます。

USBの接続形態は順次結線(Daisy chain)ではなく、イーサネットのような星状結線(Star chain)です。これは拡張に際して何処かに費用を増大させるハブ(HUB)の使用を生じさせます。けれども現状のPCにおけるUSBコネクタ数の多さと、ハブ内蔵機器により、これが問題になるとは思われません。星状結線は装置間を単に順次結線するより、いくつかの利点があります。各装置の電力を監視し、過電流状態が起きた場合に他のUSB装置を分断(切断)することなくOFFに切り替えられます。Low-Speed装置がHigh-SpeedとFull-Speedの通信を受信しないようにハブがそれらを止めることで、High-Speed,Full-SpeedとLow-Speed装置を支援できます。

1つのUSBバスには127個までの装置が接続できます。初期のUSBは1つのホスト制御器と2つのコネクタで構成されたものが多く、従ってこれらは利用可能な同じUSB帯域幅を2つのコネクタで共用していました。現在では4,5個のホスト制御器と各制御器に対応した1つのコネクタの形式が一般的です。従って各コネクタに対してUSB帯域幅がそれぞれ最大限に利用できます。

USBホスト制御器にはそれ自体の仕様があります。USB1.1ではハードが軽くソフトが重いUHCI(Universal Host Contoller Interface)とソフトが軽くハードが重いOHCI(Open Host Controller Interface)の2つのホスト制御器仕様があります。USB2.0の導入においてUSB2.0の詳細仕様をレジスタ段階で記述するために新しいホスト制御器仕様が必要とされ、EHCI(Enhanced Host Controller Interface)が生まれました。主要な制御器はこれに従い、故に新しい1つのドライバのみの実装で済むようになります。

USBはシリアルバスと言う名前のように、2本の電源線(VBUSとGND)と2本のツイスト対による差動データ信号線の合計4本を使用します。信号線ではホストと装置のクロックを同期させる同期(SYNC)領域と送出データに対してNRZI(Non Return to Zero Invert)符号化が使用されます。

USBは動的に読み込み/解除可能なドライバでプラグ&プレイを支援します。使用者は単に装置をバスに(プラグを)接続するだけです。ホストはこの追加を検知し、新規に挿入された装置に問い合わせ、適切なドライバを読み込むでしょう。初回接続時、その装置に対して提供されたドライバが要求されるかもしれません。使用者はIRQやI/Oアドレスのような用語や、コンピュータの再起動、終了について悩む必要はありません。その装置の使用が終われば、単にプラグを抜きケーブルを取り外せ、ホストはそれが存在しないことを検知して自動的に読み込まれたドライバを解除します。

適切なドライバの読み込みは供給者識別(VID:Vendor ID)と製品識別(PID:Product ID)の組み合わせを使用して行われます。供給者識別(VID)は有償でUSB Implementor's Forumから供給されます。最新の情報は http://www.usb.org/developers/vendor/ で得られます。

他の標準化機構では教育、研究または趣味のような非商業活動に余分なVIDを供給しています。しかしUSB Implementor's Forumは未だこのサービスを提供していません。このような状態では、開発システム製造業者に割り当てられたものを使用したいと思うかもしれません。例えば殆どのチップ製造業者は、商用装置として存在しないことが明白で、外部から依頼されたチップに使用可能なVIDとPIDの組み合わせを持っているでしょう。チップ製造業者はこの商用装置用にそれらのVIDと共に使用するPIDを更に売ることができます。

特記すべきUSBの他の特徴として転送種別があります。USBは制御(Control)転送、割り込み(Interrupt)転送、等時(Isochronous)転送、大量(Bulk)転送を支援します。等時(Isochronous)転送は遅れを保証するために、定義された帯域幅での巡回予約を装置に許します。混雑がデータ損失やフレーム落ちを起こすかもしれないオーディオやビデオの応用でこれは理想的です。各転送種別は保証された遅延と帯域幅と、エラーの検出/回復のような領域の使用との間で交換条件的選択を設計者に提供します。

USB構成概要

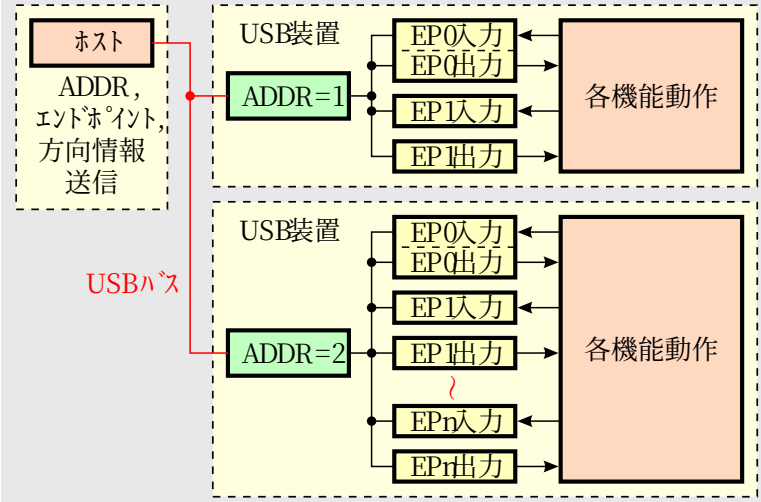
USB装置について考察すると、主にUSBﾊﾞｽ側の制御とUSB装置としての本来の動作対象制御が考えられます。ここではﾌﾟﾘﾝﾀなどの周辺装置の機能や能力を提供するUSB装置と見える標準化された仕組みについて記述します。

殆どのUSB機能はUSB制御器IC内で処理(Transaction)層までの低位USB規約が扱われます。けれども多くのUSB制御器はﾊﾟｹｯﾄ識別(PID)符号復号化異常のようなｴﾗｰを報告してきます。従って最低限の下位層の知識も必要です。

多くのUSB機能は通常8ﾊﾞｲﾄ長の直列ﾊﾞｯﾌｧを持ちます。各ﾊﾞｯﾌｧは(EP0 IN,EP0 OUT等の)ｴﾝﾄﾞﾎﾟｲﾝﾄに所属します。例えばﾎｽﾄが装置記述要求を送信した場合を考えます。装置側では設定(Setup)ﾊﾟｹｯﾄを受信して、そのﾊﾟｹｯﾄが自分宛かをｱﾄﾞﾚｽ領域から判断し、そうならばその設定(SETUP)指示票(ﾄｰｸﾝ)のｴﾝﾄﾞﾎﾟｲﾝﾄ領域の値で指定された適切なｴﾝﾄﾞﾎﾟｲﾝﾄのﾊﾞｯﾌｧへ後続するﾃﾞｰﾀﾊﾟｹｯﾄ内のﾃﾞｰﾀ部を複写します。そしてこのﾃﾞｰﾀ受信に応答するためのﾊﾝﾄﾞｼｪｰｸﾊﾟｹｯﾄを送信し、ﾊﾟｹｯﾄが受信されたことを示すｴﾝﾄﾞﾎﾟｲﾝﾄに対応したﾏｲｸﾛｺﾝﾄﾛｰﾗへの割り込みを生成します。これは通常、USB制御器などのﾊｰﾄﾞｳｪｱによって行われます。

ｿﾌﾄｳｪｱはこの割り込みを受け、そのｴﾝﾄﾞﾎﾟｲﾝﾄのﾊﾞｯﾌｧ内容を読み、その内容の装置記述要求を解析すべきです。

図1.ｴﾝﾄﾞﾎﾟｲﾝﾄ経由でのﾃﾞｰﾀの流れ



ｴﾝﾄﾞﾎﾟｲﾝﾄ

ｴﾝﾄﾞﾎﾟｲﾝﾄはﾃﾞｰﾀの供給元または受け取り先として記す事ができます。ﾊﾞｽがﾎｽﾄ中心のため、ｴﾝﾄﾞﾎﾟｲﾝﾄはUSB機能において通信路の終端とみなせます。例えばｿﾌﾄｳｪｱ層でﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞがUSB装置のEP1へﾊﾟｹｯﾄを送信したとします。ﾎｽﾄから続いてﾃﾞｰﾀが送られると、それはUSB装置のEP1出力ﾊﾞｯﾌｧに行き着きます。その後USB装置のﾌｧｰﾑｳｪｱが余裕のある時にそのﾃﾞｰﾀを読み込むでしょう。それがﾃﾞｰﾀ送信を要求している場合、ﾎｽﾄがﾊﾞｽを制御しているため、単にﾊﾞｽへ書く訳には行きません。従ってﾃﾞｰﾀをEP1入力ﾊﾞｯﾌｧに書き、このﾃﾞｰﾀはﾎｽﾄがﾃﾞｰﾀを要求したｴﾝﾄﾞﾎﾟｲﾝﾄに入力(IN)ﾊﾟｹｯﾄを送信するまで、そのﾊﾞｯﾌｧ内に保持されることになります。このようにｴﾝﾄﾞﾎﾟｲﾝﾄは装置機能のﾊｰﾄﾞｳｪｱと装置機能上で走行するﾌｧｰﾑｳｪｱ間のｲﾝﾀｰﾌｪｰｽとして見ることもできます。

全ての装置はｴﾝﾄﾞﾎﾟｲﾝﾄ0(EP0)を支援しなければなりません。これは全ての装置においてﾊﾞｽ上で装置が使用可能な間の列挙中の制御と状態情報の要求が受信されるｴﾝﾄﾞﾎﾟｲﾝﾄです。

ｲﾝﾀｰﾌｪｰｽ

ｲﾝﾀｰﾌｪｰｽは装置において特定動作を司るｴﾝﾄﾞﾎﾟｲﾝﾄ群の集合体です。従ってｲﾝﾀｰﾌｪｰｽ自体はﾊﾞｯﾌｧなどの資源を直接持たずに、ｴﾝﾄﾞﾎﾟｲﾝﾄを保持することで間接的にﾊﾞｯﾌｧを保持します。通常、このｲﾝﾀｰﾌｪｰｽと言う単位は、この上層である設定層内で異なる設定間を切り替える時の単位に使用します。

ﾊﾟｲﾌﾟ

装置側が直列のｴﾝﾄﾞﾎﾟｲﾝﾄでﾃﾞｰﾀを送受信する一方、ﾎｽﾄ側ﾌﾟﾛｸﾞﾗﾑはﾊﾟｲﾌﾟを通じてﾃﾞｰﾀを転送します。このﾊﾟｲﾌﾟはﾎｽﾄとｴﾝﾄﾞﾎﾟｲﾝﾄ間の論理的な接続です。ﾊﾟｲﾌﾟは割り当てられた帯域幅、使用する転送種別(制御(Control),割り込み(Interrupt),大量(Bulk),等時(Isochronous))、ﾃﾞｰﾀの転送方向、ﾊﾟｹｯﾄやﾊﾞｯﾌｧの最大容量などに関連したﾊﾟﾗﾒｰﾀ群も持ちます。例えば既定のﾊﾟｲﾌﾟはｴﾝﾄﾞﾎﾟｲﾝﾄ0入力とｴﾝﾄﾞﾎﾟｲﾝﾄ0出力で構成された制御(Control)転送種別の双方向ﾊﾟｲﾌﾟです。

USBでは2種類のﾊﾟｲﾌﾟが定義されています。

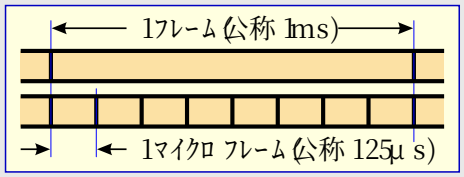
ﾒｯｾｰｼﾞﾊﾟｲﾌﾟ ﾒｯｾｰｼﾞﾊﾟｲﾌﾟには定義されたUSB形式があります。これらはﾎｽﾄから送信された要求で始まり、ﾎｽﾄにより制御されます。そしてﾃﾞｰﾀはその要求により命令され、望まれた方向に転送されます。従ってﾒｯｾｰｼﾞﾊﾟｲﾌﾟは双方向の転送をﾃﾞｰﾀに許しますが、これは制御(Control)転送だけを支援します。

ｽﾄﾘｰﾑﾊﾟｲﾌﾟ ｽﾄﾘｰﾑﾊﾟｲﾌﾟには定義されたUSB形式がなく、どのﾃﾞｰﾀ形式でもｽﾄﾘｰﾑﾊﾟｲﾌﾟへ送信でき、他方からのﾃﾞｰﾀを取得できます。ﾃﾞｰﾀは入力(IN)または出力(OUT)の予め定義された方向で順次転送されます。ｽﾄﾘｰﾑﾊﾟｲﾌﾟは割り込み(Interrupt),大量(Bulk),等時(Isochronous)転送種別を支援します。ｽﾄﾘｰﾑﾊﾟｲﾌﾟはﾎｽﾄまたは装置のどちらかで制御できます。

ﾃﾞｰﾀ転送の構造

USBでは単一ﾊﾞｽ上で複数の装置機能を扱うため、多くの工夫が成されています。このため 1対 1の RS-232のように簡単な手順での通信は望めません。最初に理解しなければならないのは、複数の転送要求を見かけ上同時処理するためにﾊﾞｽが時分割で使用されることです。これは フレームと言う単位で処理されます。High-Speedﾊﾞｽの場合は更に 1ﾌﾚｰﾑが 8つの ﾏｲｸﾛ ﾌﾚｰﾑに分割されます。従って Low/Full-Speedﾊﾞｽでは (公称)1ms毎、High-Speedﾊﾞｽでは (公称)125μ s毎にﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑ処理が繰り返されます。

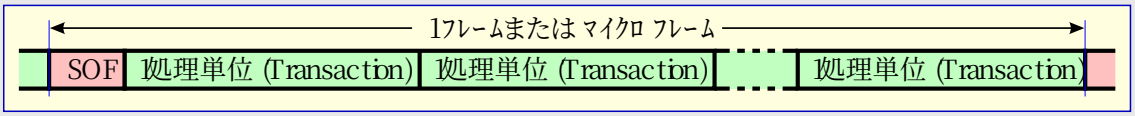
図 2.ﾌﾚｰﾑとﾏｲｸﾛ ﾌﾚｰﾑの関係



これらの ﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑは フレーム開始 (SOF)ﾊﾟｹｯﾄで実際の処理が開始されます。SOFﾊﾟｹｯﾄは単独で用いられ、後述の一般的な処理単位 (Transaction)構造に従いません。即ち指示票 (Token)ﾊﾟｹｯﾄ単独で使用されます。

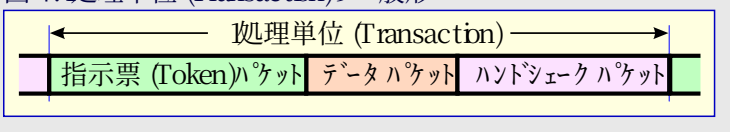
SOFﾊﾟｹｯﾄに続いて処理単位 (Transaction)群の処理が行われます。応用においての或るﾃﾞｰﾀ転送は、一般的にﾄﾞﾗｲﾊﾞなどの下層において制御部やﾃﾞｰﾀ部などの複数部分で構成され、それらの転送によって応用でのﾃﾞｰﾀが転送されます。USBではこれら下層での 1転送を転送 (Transfer)単位とし、1転送単位は複数の処理 (Transaction)単位で構成されると考えます。従ってこの処理単位が各 ﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑ内で順次転送処理され、いくつかの ﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑで 1つの転送が完了します。1ﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑ内の構成例を次に示します。

図 3.1ﾌﾚｰﾑまたは ﾏｲｸﾛ ﾌﾚｰﾑの構成



1処理単位は基本的に指示票 (Token)ﾊﾟｹｯﾄ、ﾃﾞｰﾀ ﾊﾟｹｯﾄ、ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄの 3種類で構成されます。

図 4. 1処理単位 (Transaction)の一般形



転送単位は転送種別によって異なり、制御 (Contoro1)転送のみが複雑な構造です。制御 (Contorl)転送は設定段階、ﾃﾞｰﾀ段階、状態段階の 3段階から成ります。設定段階と状態段階は各々 1つの処理単位 (Transaction)で構成されます。ﾃﾞｰﾀ段階は 1つ以上の処理単位 (Transaction)で構成されます。他の転送種別では基本的にﾃﾞｰﾀ段階のみが用いられます。

ﾌﾚｰﾑ /ﾏｲｸﾛ ﾌﾚｰﾑと転送単位は直接的な関係はありません。転送単位が ﾌﾚｰﾑ /ﾏｲｸﾛ ﾌﾚｰﾑ内で完了するとは限りません。その場合は複数の ﾌﾚｰﾑ /ﾏｲｸﾛ ﾌﾚｰﾑを使用して実行されます。ﾌﾚｰﾑ /ﾏｲｸﾛ ﾌﾚｰﾑでの実際の処理はﾊﾞｽ速度に対応する帯域幅や転送種別などによってﾎｽﾄにより決定されます。

図 5. 1転送単位との関係

帯域幅の管理

ﾎｽﾄにはﾊﾞｽの帯域幅を管理する責任があります。これは割り込みと等時の ｴﾝﾄﾞﾎﾟｲﾝﾄ設定時の列挙により、ﾊﾞｽ動作全体に渡って行われます。仕様は Full-Speedﾊﾞｽ上での周期的 (割り込みと等時)転送に対して、どの ﾌﾚｰﾑでも 90%を越えて割り当てられないように制限が設けられています。High-Speedﾊﾞｽでのこの制限は ﾏｲｸﾛ ﾌﾚｰﾑの 80%を越えないことに減らされており、これを周期的転送に割り当てることができます。

高度に飽和しているﾊﾞｽでも、残りの 10%が制御 (Control)転送と、一旦割り当てられた大量 (Bulk)転送に残され、大量転送は細切れに転送されることになるでしょう。

装置接続と列挙 (Enumeration)

列挙はどの装置が今 ﾊﾞｽに接続されたか、またそれが消費電力、ｴﾝﾄﾞﾎﾟｲﾝﾄの種別や数、製品の ｸﾗｽなどのようなどの ﾊﾟﾗﾒｰﾀを必要とするかを判定する過程の処理です。その後 ﾎｽﾄは装置にｱﾄﾞﾚｽを割り当て、ﾊﾞｽにﾃﾞｰﾀを転送することを装置に許す設定を許可します。この詳細に関しては USB仕様書で詳述されますが、ここでは列挙中にﾎｽﾄがどう応答するかを中心に簡潔に概要を示します。

Windowsの列挙は共通で、次の手順で進められます。

1. ﾎｽﾄ(または ﾊﾌﾞ)はﾃﾞｰﾀ信号線上で装置の ﾌﾟﾙｱｯﾌﾟ抵抗により新規装置の接続を検知します。ﾎｽﾄは ﾌﾟﾗｸﾞが完全に挿入され、装置への安定給電を可能とするため、最低 100ms待機します。
2. ﾎｽﾄは装置を既定状態に置くために ﾘｾｯﾄを送出します。これで装置は既定の ｱﾄﾞﾚｽ0で応答できるようになります。
3. Windowsの ﾎｽﾄは装置記述 (Device Descriptor)の先頭 64ﾊﾞｲﾄを要求します。
4. 装置記述 (Device Descriptor)の最初の 8ﾊﾞｲﾄ受信後直後、更にﾊﾞｽ ﾘｾｯﾄを行います。
5. ﾎｽﾄは装置をｱﾄﾞﾚｽ指定状態にするｱﾄﾞﾚｽ設定 (SET_ADDRESS)命令を送出します。
6. ﾎｽﾄは装置記述 (Device Descriptor)の 18ﾊﾞｲﾄ全体を要求します。
7. そして容量関係を決めるために設定記述 (Configuration Descriptor)の 9ﾊﾞｲﾄを要求します。
8. ﾎｽﾄは設定記述 (Configuration Descriptor)の 255ﾊﾞｲﾄを要求します。
9. それらで指定されているなら、ﾎｽﾄは対応する文字列記述 (String Descriptor)を要求します。

手順 9の後、Windowsは装置に対するﾄﾞﾗｲﾊﾞについて要求します。その後、設定選択 (SET_CONFIGURATION)命令を行う前に全ての記述を要求すると考えるのが一般的です。

上記の列挙手順はWindows 98SE, 2000, XPで共通です。手順 4は初めてﾌｧｰﾑｳｪｱを作成する場合に度々混乱させます。ﾎｽﾄが装置記述 (Device Descriptor)の先頭 64ﾊﾞｲﾄを要求し、その先頭 8ﾊﾞｲﾄ受信後にﾎｽﾄが装置をﾘｾｯﾄするのは、装置記述 (Device Descriptor)に何か不正はないか、装置の ﾌｧｰﾑｳｪｱが要求をどう扱うかを調べるためだけと考えるのが自然です。実際の装置記述 (Device Descriptor)取得は手順 6で行われます。

通常、記述や送出法で何か不正があると、ﾎｽﾄは長い要求間隔で 3回の読み込みを試みます。ﾎｽﾄは 3度目の試行後に諦めて装置の異常を報告します。

電気的な仕様

USBホストやハブまたはUSB制御器やトランシーバの設計を行わない限り、USB2.0仕様の電気的特性を全て知る必要はありません。ここでは要点のみを記述します。

USBはデータ用に1対の差動送信を使用します。これはNRZ符号化され、データ列内に充分なレベル遷移を保証するためにビット挿入が行われます。Low-SpeedとFull-Speedの装置での差動1はGNDへの15kΩプルダウン抵抗付きでDATA+を2.8Vより高く、GNDへの15kΩプルダウン抵抗と3.6Vへの1.5kΩプルアップ抵抗付きでDATA-を0.3Vより低くにすることで送出されます。他方の差動0は同様のプルアップ/プルダウン抵抗付きでDATA+が0.3Vより低く、DATA-が2.8Vより高くします。

受信側ではDATA+がDATA-より0.2Vより高い時に差動1、DATA+がDATA-より0.2V以上低い時に差動0として定義されます。信号の極性はバス速度に依存して反転されます。従って用語としてJとK状態が論理レベルの指示に使用されます。Low-Speedでは J状態が差動0で、High-Speedでは J状態が差動1です。

USBトランシーバは シングル エンドと差動の両出力を持ちます。或るバス状態はDATA+,DATA-または両方の シングル エンド信号で示されます。例えば シングル エンドの0または SE0は10ms以上保持した場合に装置 リセットの指示に使用され得ます。SE0はDATA+とDATA-両方をLow(< 0.3V)に保持することによって生成されます。

Low-SpeedとFull-Speedの バスは90Ω ± 15%の特性 インピーダンスを持ちます。従ってDATA+とDATA-に対して インピーダンス整合の直列抵抗を選択するときには、データシートでの これらの確認が重要です。

High-Speed動作では ノイズを削減するために17.78mAの定電流駆動を使用します。

表1.Low/Full-Speed信号 レベル一覧

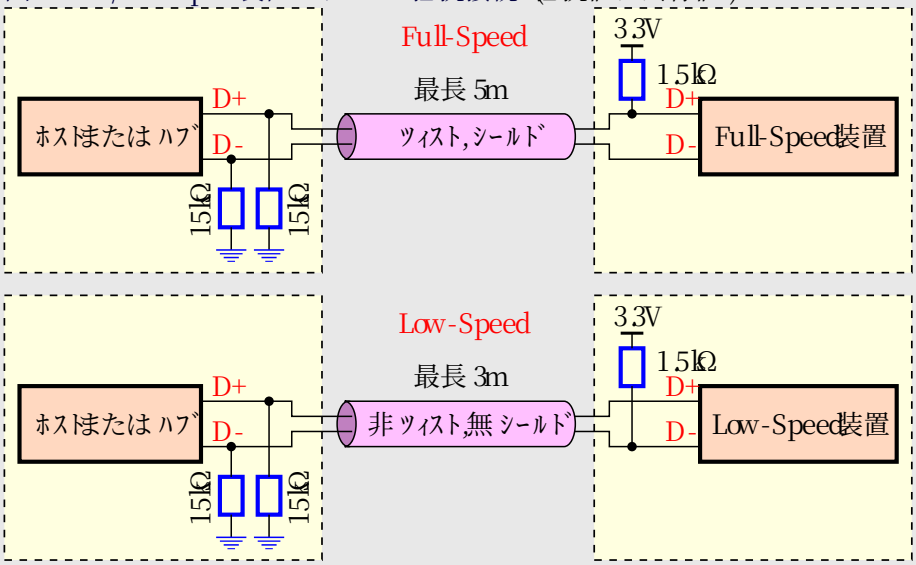
バス状態 / Speed		出力条件	入力条件	
			必要条件	許容条件
差動 1		D+>VOH(min)且つD-<VOL(max)	D+>VIH(min)且つ (D+)-(D-)>0.2V	(D+)-(D-)>0.2V
差動 0		D+<VOL(max)且つD->VOH(min)	D->VIH(min)且つ (D-)-(D+)>0.2V	(D-)-(D+)>0.2V
SE0 (シングル エンド 0)		D+とD-<VOL(max)	D+とD-<VIL(max)	D+とD-<VIH(min)
SE1 (シングル エンド 1)		D+とD->VOSE1(min)	D+とD->VIL(max)	
データ J状態	Low		差動 0	
	Full		差動 1	
データ K状態	Low		差動 1	
	Full		差動 0	
アイドル状態	Low	駆動なし	D->VHZ(min)且つD+<VIL(max)	D->VHZ(min)且つD+<VIH(min)
	Full	駆動なし	D+>VHZ(min)且つD-<VIL(max)	D+>VHZ(min)且つD-<VIH(min)
回復 (Resume)状態			データK状態	
SOP (パケット開始)			アイドルからK状態への遷移	
EOP (パケット終了)		約2ビットの SE0+1ビットの J状態	1ビット以上の SE0+1ビットの J状態	1ビット以上の SE0+J状態
切断		駆動なし	2.5μs以上の SE0	
接続		駆動なし	2msec以上の アイドル状態	2.5μs以上の アイドル状態
リセット		10ms以上のD+とD-<VOL(max)	10ms以上のD+とD-<VIL(max)	2.5μs以上のD+とD-<VIL(max)

注 :D+,D-は各々DATA+,DATA-を表します。各指標に対する具体的な数値についてはUSB2.0仕様書をご覧ください。

ﾊﾞｽ速度の識別

USB装置はDATA+またはDATA-のどちらかの信号線を3.3Vにﾌﾟﾙｱｯﾌﾟすることでその速度を示さなければなりません。Full-Speedの装置は右図で示されるように自身がFull-Speed装置であることを示すためにDATA+へ接続されたﾌﾟﾙｱｯﾌﾟ抵抗を使用します。装置端におけるこれらのﾌﾟﾙｱｯﾌﾟ抵抗は接続された装置の存在を検知するためにﾎｽﾄやﾊﾌﾞによって使用されます。ﾌﾟﾙｱｯﾌﾟ抵抗がない場合、ﾊﾞｽに何も接続されていないとみなされます。いくつかの装置ではﾌｧｰﾑｳｪｱ下でON/OFFできる抵抗が組み込まれたﾁｯﾌﾟを持ちますが、それ以外では外部抵抗が必要になります。

図6.Low/Full-Speed装置のｹｰﾌﾞﾙと抵抗接続（抵抗値は公称値）



ﾌｧｰﾑｳｪｱ下でON/OFF可能なﾌﾟﾙｱｯﾌﾟ抵抗の場合、接続時において電源供給開始で装置の初期化を行った後にﾌﾟﾙｱｯﾌﾟ抵抗をONにすることで自身の初期化時間を充分に取れます。また接続したままで一度OFFにしてからONにすることで、新規装置の認証となりますので、例えば装置機能を変更し、別の機能の装置としての使用が接続したままで行えます。

High-Speed動作についてはここで示されていません。High-Speed装置はFull-Speed装置（DATA+の1.5kΩでの3.3Vへのﾌﾟﾙｱｯﾌﾟ）としての接続で始まります。そしてﾘｾｯﾄ中にHigh-Speed動作を要求し、ﾎｽﾄまたはﾊﾌﾞがHigh-Speed動作を支援するなら、High-Speed接続を確立します。そして装置がHigh-Speedで動作する場合、ﾌﾟﾙｱｯﾌﾟ抵抗は取り去られ、平衡信号路になります。

USB2.0適合装置は必ずしもHigh-Speed動作を支援する必要はありません。これは速度が重要でない安価な装置の製造を可能にします。これはFull-Speedを支援しないUSB1.1適合のLow-Speed装置の場合にも言えます。

High-Speed装置はLow-Speed動作を支援する必要はありません。接続の初めでFull-Speed動作が必要なだけで、交渉が成立すると、その後はHigh-Speedで動作します。USB2.0適合の下方向（Downstream）装置（ﾎｽﾄまたはﾊﾌﾞ）はHigh-Speed, Full-Speed, Low-Speedの3つ全ての動作種別を支援しなければなりません。

電力供給（VBUS）

USBの利点の1つはﾊﾞｽ電力供給装置（追加ｹｰﾌﾞﾙや外部電源の必要がなく、ﾊﾞｽから電力を得る装置）です。多くの人々は必要な判断基準の全てを最初に考慮することなく、この選択に飛び付くでしょう。

USB装置は後述される設定記述内において2mA単位で表される消費電力を指定します。装置は例え外部電源を失ったとしても、列挙中に指定した値より大きく消費電力を増やせません。これらはUSB装置の3つ種別です。

- 低電力ﾊﾞｽ給電装置（Low-power bus powerd functions）
- 高電力ﾊﾞｽ給電装置（High-power bus powerd functions）
- 自己給電装置（Self-powerd functions）

低電力ﾊﾞｽ給電装置はV_{BUS}からその電力の全てを得ますが、1負荷単位を越える電力を得ることはできません。USB仕様で1負荷単位は100mAとして定義されます。低電力ﾊﾞｽ給電装置は装置からの上方向（Upstream）側ﾌﾟﾗｸﾞで4.40～5.25Vの範囲で動作するように設計されなければなりません。従って多くの3.3V動作装置では低ﾄﾞﾛｯﾌﾟﾚｷﾞｭﾚｰﾀが必須となるでしょう。

高電力ﾊﾞｽ給電装置はV_{BUS}からその電力の全てを得ますが、初期設定が成されてしまうまで、1負荷単位を越えて電力を得ることはできません。状態情報（設定）記述の問い合わせ後、5負荷単位（最大500mA）が提供され、それを得ることができます。高電力ﾊﾞｽ給電装置は最小4.40Vで検出と列挙が出来なければなりません。全負荷単位で動作する場合、4.75Vの最小V_{BUS}は最大の5.25Vで指定されます。これらは装置からの上方向（Upstream）側ﾌﾟﾗｸﾞ上での値です。

自己給電装置はV_{BUS}から1負荷単位までの電力を得、残りは外部電源から得ます。この外部電力が不足または失われた場合、ﾊﾞｽからの給電が1負荷単位を越えない処置が成されなければなりません。自己給電装置は消費電力についての諸問題がないため、仕様に対する設計が容易です。このﾊﾞｽからの1負荷単位供給は他の電源なしでの装置の検出と列挙を可能にします。

ﾊﾞｽ給電または自己給電の何れでもないUSB装置は上方向（Upstream）側ﾊﾞｽのV_{BUS}を駆動できます。V_{BUS}が失われたとき、速度識別に使用したDATA+/DATA-のﾌﾟﾙｱｯﾌﾟ抵抗からの電力がかなりの時間残留することに注意してください。他の考慮点として突入電流は制限されなければなりません。これはUSB仕様書の7.2.4.1節に概要がありますが、一般的に見落とされがちです。突入電流は装置におけるV_{BUS}とGND間の容量によって引き起こされます。故に仕様で最大容量（10μF）が指定されます。ｲﾝﾀﾞｸﾀﾝｽ性USBｹｰﾌﾞﾙを通して電流が流れた後に装置を切断するとｹｰﾌﾞﾙ端に反転電圧が発生し得ます。これを防ぐために最小1μFのV_{BUS}ﾃﾞｶｯﾌﾟ容量が指定されています。

一時停止 (Suspend)電流

一時停止動作は全ての装置で必須です。一時停止中は追加の制限が強制されます。最大一時停止電流は負荷単位に比例し、1負荷単位の装置 (既定)は 0.5mAです。これにはﾊﾞｽの ﾌﾟﾙｱｯﾌﾟ抵抗からの電流も含みます。ﾎｽﾄや ﾊﾌﾞは DATA+と DATA-に 15kΩのﾌﾟﾙﾀﾞｳﾝ抵抗を持ちます。このﾌﾟﾙﾀﾞｳﾝ抵抗は消費電力に関して装置の 1.5kΩ ﾌﾟﾙｱｯﾌﾟ抵抗と直列になり、代表値 3.3Vの VTERMで 16.5kΩの合計負荷になります。従って一時停止開始以前からこの抵抗が 0.2mA電力を消費しています。

多くの装置での他の考慮点は 3.3V電圧安定器です。USB装置の多くは 3.3Vで走行します。ｼﾘｰｽﾞﾊﾞｽ型安定器は平均的な静止時電流が 0.6mA程度と一般的にかなり非効率で、故により効率の良い、即ち高価な安定器が必要になります。多くの場合で 0.5mAの制限内に抑えるため、ﾏｲｸﾛ ｺﾝﾄﾛｰﾗの ｸﾛｯｸを低下または停止させなければならないでしょう。

殆どの ﾎｽﾄや ﾊﾌﾞは数mA程度のような過負荷を検知する能力がありませんので、USB仕様に違反してそのようにした場合でも動作を維持できるでしょう。けれども 100mAまたは許可された負荷を超える試みは、ﾎｽﾄまたは ﾊﾌﾞがこれを検知してﾊﾞｽの保全のためにその装置を切断すると予想されます。

勿論自己給電装置としての設計を選択すれば、これら設計上の問題点は避けることができます。この一時停止電流はﾃﾞｽｸﾄｯﾌﾟｺﾝﾋﾟｭｰﾀについてはあまり関係ありませんが、On-The-Go仕様で USBﾎｽﾄが組み込まれた小型機器にとっては重要なことです。これらの機器から引き出される消費電力は電池での動作時間に大きく影響するでしょう。

一時停止 (Suspend)動作への移行

USB装置はﾊﾞｽ上の動きが 3.0ms以上なければ一時停止動作へ移行するでしょう。その後停止 (Shutdown)まで更に 7.0msあり、従ってﾊﾞｽの動き停止後の 10ms間は、指定された一時停止電流だけで動作しなければなりません。一時停止されたﾎｽﾄまたは ﾊﾌﾞとの接続を維持するため、装置は一時停止中も速度選択用の ﾌﾟﾙｱｯﾌﾟ抵抗に電力を供給しなければなりません。

USBは フレーム開始 (SOF)ﾊﾟｹｯﾄや周期的なﾊﾞｽ送信でﾊﾞｽの活性状態を維持します。これはﾃﾞｰﾀがない状態でｱｲﾄﾞﾙ状態のﾊﾞｽが一時停止動作へ移行するのを防ぎます。High-Speedﾊﾞｽでは 125.0μs± 62.5ns毎にﾏｲｸﾛ ﾌﾚｰﾑを送信します。Full-Speedﾊﾞｽでは 1ms± 500ns毎にﾌﾚｰﾑを送信します。Low-Speedﾊﾞｽではどの Low-Speedﾃﾞｰﾀもない状態でだけの 1ms毎の ﾊﾟｹｯﾄ終了 (EOP)がﾊﾞｽ活性を維持します。

用語の全一時停止 (Global Suspend)は USBﾊﾞｽ全体を纏めて一時停止動作へ移行するときに使用されます。また装置はそれが接続されているﾎｽﾄまたは ﾊﾌﾞからの命令によって一時停止することもできます。これは選択的一時停止 (Selective Suspend)として参照されます。

装置はｱｲﾄﾞﾙ状態でない何かの受信時にその動作を再開します。装置が遠隔起動許可なら、装置は一時停止からの再開をﾎｽﾄへ告げるかもしれません。

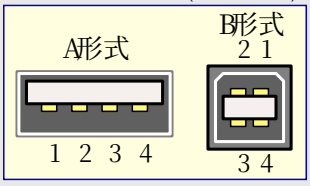
ﾃﾞｰﾀ転送速度

見落とされがちな項目として USBｸﾛｯｸの許容誤差があります。これは USB仕様書の 7.1.11節で規定されます。High-Speedﾃﾞｰﾀは 480Mbps± 0.05% Full-Speedﾃﾞｰﾀは 12Mbps± 0.25% Low-Speedﾃﾞｰﾀは 1.5Mbps± 1.5%で転送されます。これは低価格 Low-Speed装置用にｾﾗﾐｯｸ振動子の使用を可能にしますが、Full-Speedや High-Speed装置に対しては許されません。

USBｺﾈｸﾀと信号配置

全ての装置はﾎｽﾄへの上方向 (Upstream)接続で、全ての ﾎｽﾄは装置への下方向 (Downstream)接続になります。上方向と下方向の ｺﾈｸﾀは構造的に交換できず、従って下方向ﾎﾟｰﾄへ接続した下方向ﾎﾟｰﾄのようにﾊﾌﾞでの不正なﾙｰﾌﾟ接続をなくします。これらは右で示されるような俗に A形式と B形式と呼ばれる 2つの形式の ｺﾈｸﾀです。

図 7.USBｺﾈｸﾀ (ﾚｾｸﾀﾌﾟﾙ)



A形式 ﾌﾟﾗｸﾞは常に上方向側です。A形式 ｿｹｯﾄは一般的にﾎｽﾄとﾊﾌﾞで見つかるでしょう。A形式 ｿｹｯﾄはﾎｽﾄとなる PCとﾊﾌﾞで共通です。B形式 ﾌﾟﾗｸﾞは常に下方向側で、その結果 B形式 ｿｹｯﾄは装置で見れます。

いくつかの販売店で見られる A形式と A形式の直結 ｹｰﾌﾞﾙとｵｽ/ﾒｽ (ﾌﾟﾗｸﾞ/ｿｹｯﾄ)変換器は面白い構成です。これは USB仕様に矛盾します。A形式 ﾌﾟﾗｸﾞと A形式 ﾌﾟﾗｸﾞの装置は 2つの PC間の接続に使用するﾌﾞﾘｯｼﾞだけです。その他の禁制 ｹｰﾌﾞﾙは一端がﾌﾟﾗｸﾞで多端がｿｹｯﾄ(A形式または B形式のどちらか)を持つ USB延長 ｹｰﾌﾞﾙです。これらの ｹｰﾌﾞﾙは USBの ｹｰﾌﾞﾙ長の必要条件に違反するかもしれません。

USB2.0には USBﾐﾆ Bｺﾈｸﾀ導入の障害情報を含みます。これらの ｺﾈｸﾀの詳細は http://www.usb.orgの Mini-B Connector Engineering Change Noticeで得られます。ﾐﾆ ｺﾈｸﾀが遅れた理由は携帯電話などのような小型電子装置の領域から来ました。現在の B形式 ｺﾈｸﾀはこれらの装置内に統合するには大きすぎます。

表 2.USBﾋﾟﾝの配色と機能

番号	配色	機能
1	赤	VBUS (+5V)
2	白	DATA-
3	緑	DATA+
4	黒	GND

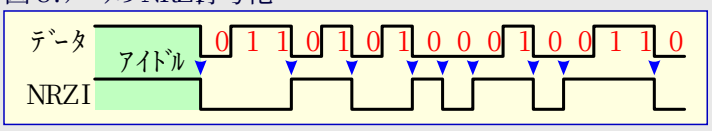
USBにﾋﾟｱ toﾋﾟｱを付加する On-The-Go仕様が公開されています。これは携帯電話などの小型機器に USBﾎｽﾄを導入し、それ故ﾐﾆ Aｺﾈｸﾀ仕様を含みます。

標準内部線色は USBｹｰﾌﾞﾙで使用され、製造時の線識別を容易にします。標準仕様にはｹｰﾌﾞﾙについて様々な電気的特性が示されています。元となる USB1.0仕様書に含まれる詳細を読むことは重要です。

NRZI(Non Return to Zero Invert)符号化

USB信号線上の信号はNRZIで符号化されます。NRZIでは各 0 が現在の信号 ﾚﾍﾞﾙを変更することで表され、各 1は現在 ﾚﾍﾞﾙを保持することで表されます。従ってﾋﾞｯﾄ挿入に関しては論理ﾃﾞｰﾀ列内の 6つの連続する論理 1後に 1つの 0ﾋﾞｯﾄが挿入されることを意味します。

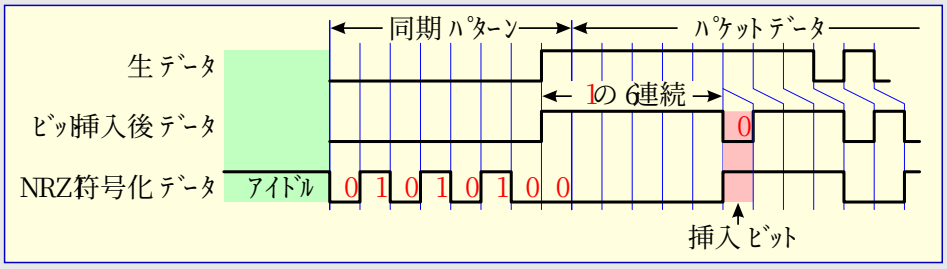
図 8.ﾃﾞｰﾀの NRZI符号化



ﾋﾞｯﾄ挿入 (0挿入)

USBでのﾃﾞｰﾀ受信は全ての時間で送信部と受信部が同期状態であることを満たさなければならず、従ってﾃﾞｰﾀ列で連続する 0または 1の流れを送ることは許されません。これは信号線上の ﾚﾍﾞﾙ遷移 (ｴｯｼﾞ)で送受信部が同期を取るためです。従ってﾚﾍﾞﾙ遷移間が長いと同期ずれの危険が増すことになります。これから考えるとﾋﾞｯﾄ単位でﾚﾍﾞﾙ遷移があればよさそうですが、単純にこれを行うとEMIの増加や逆にﾉｲｽﾞの影響を受け易くなります。USB規約ではﾋﾞｯﾄ挿入により同期を保証します。これはﾃﾞｰﾀ列の 6つの連続する 0または 1の後に 1つの単一変更 (1ﾋﾞｯﾄ)が挿入されることを意味します。USBではNRZI符号化が行われますので、元々のﾃﾞｰﾀでの連続 0はﾊﾞｽ上で 0/1の交互繰り返しになります。従って、このﾋﾞｯﾄ挿入処理は元々のﾃﾞｰﾀで連続する 6つの 1の後に 1ﾋﾞｯﾄの 0を追加する操作になります。

図 9.ﾋﾞｯﾄ挿入とﾃﾞｰﾀ符号化手順



USB規約 (Protocols)

送出データ形式が定義されていないRS-232や同様のシリアル インターフェースと異なり、USBは種々の規約階層が定義されています。一旦理解してしまえば、実際は上位階層についての心配だけです。現実に殆どのUSB制御器 ICが下位階層の処理を行うため、最終製品設計者からそれらを隠します。USBの各処理単位 (Transaction)は次の3部分からなります。

- 指示票 (Token)パケット ・・・本処理 (Transaction)が何を行うかを装置へ通知するヘッダ
- データ(Data)パケット ・・・・・・一般で言う実データ部の転送 (任意回数)
- 状態 (Status)パケット ・・・・本処理 (Transaction)の応答と誤り訂正の意味提供に使用

既に説明されたようにUSBはホストが中心のバスです。従ってホストが全ての処理単位 (Transaction)を開始します。この処理単位での最初のパケットは指示票 (Token:トークン)と呼ばれ、どの装置アドレスのどのエンドポイントが指示され、その処理が読み込みか書き込みのどちらか、何が続くのかが記述されており、これはホストにより生成されます。次のパケットは一般的に実データ部を運ぶデータ(Data)パケットで、指示やデータが正しく受信されたか、エンドポイントが使用不能、または受け入れデータが使用不能などを報告するハンドシェークパケットがそれに続きます。

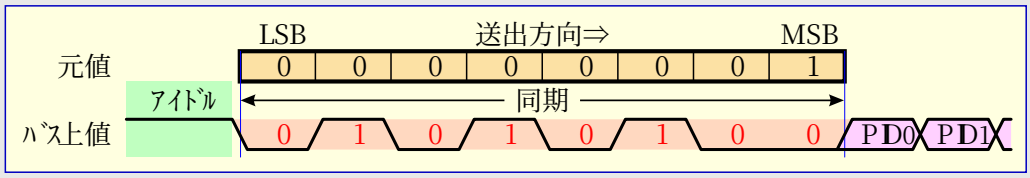
USBパケット共通の領域 (Fields)

バス上のデータはLSBが先に送信されます。USBパケットは次の各領域から成ります。

同期 (SYNC)

全てのパケットはSYNC領域から始まらなければなりません。SYNC領域は8ビット長で、送信側と受信側の同期化に使用されます。バス上での最後の2ビットはSYNC領域の最後 即ち次がPID領域であることを示します。下図においてバス上での論理値は装置速度によっては逆論理になります (Full-SpeedのDATA+として記載)。

図 10.同期値とバス上信号の関係



パケット識別 (PID)

PIDはPakect IDの略称です。本領域は送られるパケットの種別を示します。右表はこれに使用される値を示します。これらのPIDは4ビットですが、受信での正確さを保証するために論理反転値が繰り返され、合計8ビットのPIDになります。この結果の形態は以下で示されます。

図 11.PID領域形式

表 3.PID種別一覧

目的別	PID値	パケット種別
指示票 (トークン)	0001	出力 (OUT指示)
	1001	入力 (IN指示)
	0101	フレーム開始 (SOF指示)
	1101	設定 (SETUP指示)
データ	0011	データ0 (DATA0)
	1011	データ1 (DATA1)
	0111	データ2 (DATA2)
	1111	複データ (MDATA)
ハンドシェーク	0010	肯定応答 (ACK)
	1010	否定応答 (NAK)
	1110	不能応答 (STALL)
	0110	未応答 (NYET)
特殊	1100	前置部 (PRE) トークンで使用
	1100	誤り (ERR)応答で使用
	1000	分割 (SPLIT) トークンで使用
	0100	確認 (Ping) トークンで使用
	0000	予約

アドレス (ADDR)

アドレス領域はパケットがどの装置に対して指示したかを示します。7ビット長で127装置の支援を可能にします。アドレス0はアドレスがまだ割り当てられていない装置がこのアドレスへ送られるパケットに応答しなければならないため、装置へのアドレス0割り当ては無効です。本形式は以降のエンドポイント、CRCと併せ、図 12に示されます。

エンドポイント (ENDP)

エンドポイント領域は4ビットで構成され、16個のエンドポイントの使用を可能にします。Low-Speed装置では先頭の2つ (既定、最大4つ)の追加エンドポイントだけを持てます (エンドポイント0は既定且つ必須のため、これを除いての意)。本形式は次の図 12で示されます。

巡回冗長検査 (CRC)

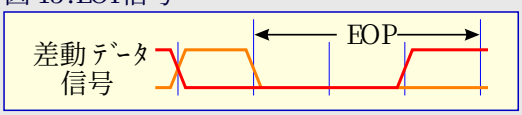
巡回冗長検査はパケット内のデータ部分の検査を行います。全ての指示票 (トークン)パケットは5ビットのCRCを、他方データパケットは16ビットのCRCを持ちます。

図 12.ADDR,ENDP,CRC領域形式

パケット終了 (EOP)

パケットの終了を示します。これは概ね2ビット時間のシングルエンド0 (SE0)と後続する1ビット時間のJ状態で表されます。

図 13.EOP信号



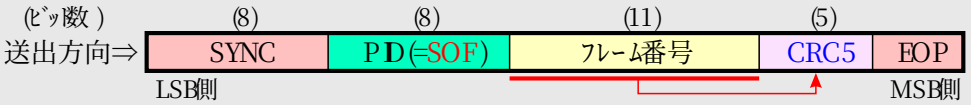
処理単位 (Transaction)とUSBﾊﾟｹｯﾄ種別

USBには 4種類の ﾊﾟｹｯﾄ種別があります。ﾌﾚｰﾑ開始 ﾊﾟｹｯﾄは新規 ﾌﾚｰﾑの開始を示します。指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄは後続するやり取りの種別を示し、ﾃﾞｰﾀ ﾊﾟｹｯﾄはそのやり取りに必要なﾃﾞｰﾀを含み、ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄはそのﾃﾞｰﾀに対する応答やｴﾗｰの報告に使用されます。一般的にUSBでは指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄ、ﾃﾞｰﾀ ﾊﾟｹｯﾄ、ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄの組で或る意味を持つ通信を行い、この通信を処理単位 (Transaction)と呼びます。

ﾌﾚｰﾑ開始 (SOF) ﾊﾟｹｯﾄ

SOFﾊﾟｹｯﾄはﾎｽﾄにより 1± 0.5ms毎に送信される 11ﾋﾞｯﾄ長の ﾌﾚｰﾑ番号を含むﾊﾟｹｯﾄです。

図 14.SOFﾊﾟｹｯﾄ形式



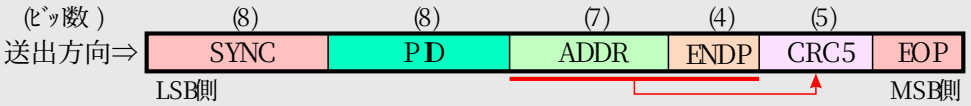
指示票 (ﾄｰｸﾝ) ﾊﾟｹｯﾄ

指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄにはﾊﾟｹｯﾄ識別 (PID)の内容により、次の 3種類があります。

- **設定 (SETUP)** ･･･ 必須の制御 (Control)転送を開始するために使用されます。
- **入力 (IN)** ･･･････ ﾎｽﾄが情報を送出したいことをUSB装置へ通知します。
- **出力 (OUT)** ･････ ﾎｽﾄが情報を取得したいことをUSB装置へ通知します。

指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄは次の形式に従わなければなりません。

図 15.指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄ形式



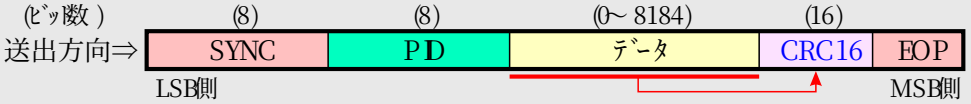
ﾃﾞｰﾀ ﾊﾟｹｯﾄ

ﾃﾞｰﾀ ﾊﾟｹｯﾄには通常、PIDの内容により、次の 2種類があります。

- **DATA0**
- **DATA1**

これらは複数ﾃﾞｰﾀ ﾊﾟｹｯﾄ時、交互に使用され欠落監視に使用されます。ﾊﾟｹｯﾄ内のﾃﾞｰﾀ部はﾊﾞｲﾄ単位で 0～ 1023ﾊﾞｲﾄの範囲ですが、Low-Speed装置にあっては 8ﾊﾞｲﾄ以内でなければなりません。ﾃﾞｰﾀ ﾊﾟｹｯﾄは次の形式です。

図 16.ﾃﾞｰﾀ ﾊﾟｹｯﾄ形式



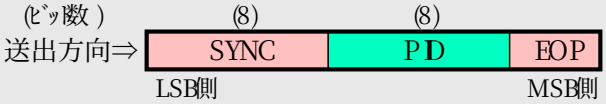
ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄ

ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄは単にPIDから成るﾊﾟｹｯﾄで、次の 3種類があります。

- **肯定応答 (ACK)** ･････ ﾊﾟｹｯﾄが成功裏に受信されたことを示す応答です。
- **否定応答 (NAK)** ･････ 装置の一時的なﾃﾞｰﾀ送受信不能を報告します。割り込み (Interrupt)転送中に送信ﾃﾞｰﾀがないことをﾎｽﾄに通知するのにも使用されます。
- **不能応答 (STALL)** ･･･ 装置が判断不能でﾎｽﾄの介在を必要とする状態であることを通知します。

ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄは次の形式です。

図 17.ﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄ形式



転送種別

USB仕様では次の 4つの転送種別が定義されています。

- 制御 (Control)転送
- 割り込み (Interrupt)転送
- 等時 (Isochronous)転送
- 大量 (Bulk)転送

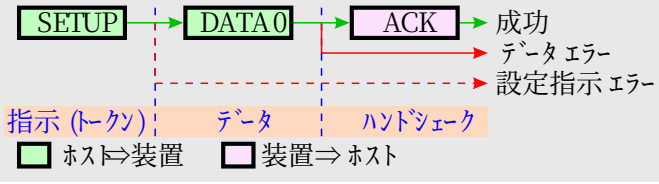
制御 (Control)転送

制御転送は通常、命令と状態の操作用に使用されます。これらは制御転送を使用して行われる全機能の列挙でUSB装置を初期設定するために必須です。これらは最善効率配給を用いてホストにより開始される集合的または離散的なパケット群です。Low-Speed装置の制御転送パケットのデータ長は 8バイト、Full-Speed装置は 64バイトでなければならず、High-Speed装置は 8, 16, 32, 64バイト長が許されます。制御転送は 3つの段階 (Stage)から成ります。

設定 (Setup)段階

設定段階は要求が送信される段階です。これは 3つのパケットから成ります。アドレスとエンドポイント番号を含む設定 (Setup)指示票 (トークン)パケットが最初に送信されます。次に設定 (Setup)パケットの要求種別詳細を含み、PID種別が常にDATA0のデータパケットが送信されます。最後のパケットは受信成功の応答またはエラーを示すのに使用されるハンドシェークパケットです。設定 (Setup)データが正しく(PIDや CRC検査が正常に)受信されたならACKで応答し、さもなくばそのデータを無視し、ハンドシェークパケットを送信してはいけません。設定 (Setup)パケットの応答ではNAKや STALLのハンドシェークパケットを使用できません。

図 18.設定 (Setup)段階の流れ



データ(Data)段階

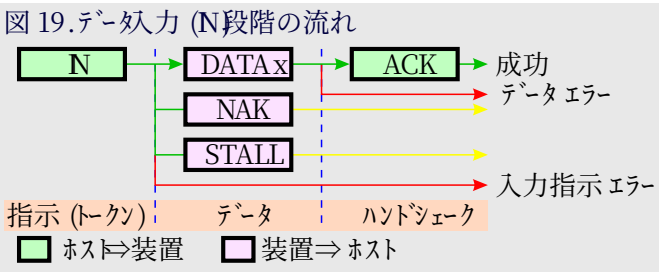
任意のデータ段階は 1つ以上の入力 (IN)または出力 (OUT)パケット転送から成ります。設定 (Setup)要求は本段階で送信されるべきデータ量を指示します。これが 1パケットの最大データ容量を越える場合、送信すべきデータは最終パケットを除いて各々のパケットが最大データ長の複数パケットで送信されます。

データ段階はデータの転送方向に依存した 2つの異なる手順があります。

入力 (IN)

ホストは制御データを受信する準備が整うと入力 (IN)パケットを送信します。装置は誤り(例えばPIDビットと反転PIDビットの不一致)と共に入力 (IN)指示票 (トークン)を受信した場合、そのパケットを無視します。その指示票 (トークン)が正しく受信されると、装置は送るべき制御データを含むデータ(DATAx)パケットか、またはエンドポイントのエラーを示す不能 (STALL)パケット、またはエンドポイントが作業中で一時的に送るデータがないことをホストへ示す否定応答 (NAK)パケットの何れかを返すことができます。

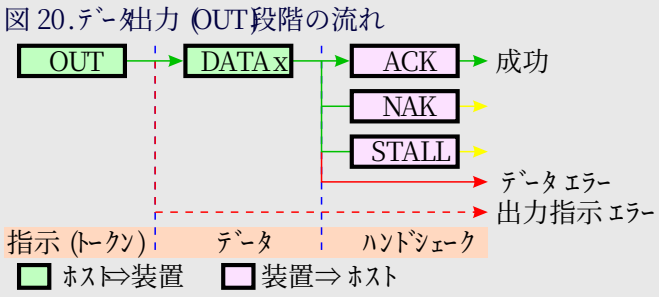
図 19.データ入力 (IN)段階の流れ



出力 (OUT)

ホストが装置へ制御データパケットを送る必要があるとき、ホストは出力 (OUT)指示票 (トークン)パケットに続きデータ部として制御データを含むデータパケットを送信します。出力 (OUT)指示票 (トークン)パケットまたはデータパケットの何れかの部分に誤りがあれば、装置はそのパケットを無視します。装置のエンドポイントバッファが空で制御データをそのバッファへ格納したなら、データの受信が成功したことをホストへ通知するために肯定応答 (ACK)パケットを送信します。直前のパケットの処理中のためにエンドポイントバッファが空でない場合、装置は否定応答 (NAK)パケットを返します。但しエンドポイントが何らかのエラーで停止(または中止)状態の場合不能応答 (STALL)パケットを返します。

図 20.データ出力 (OUT)段階の流れ

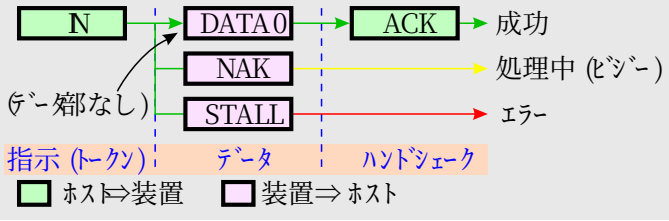


状態 (Status)段階　状態段階はﾃﾞｰﾀ段階全体に対する状態を報告しますが、これは転送方向によって形態が異なり、入出力 (IN/OUT)指示票 (ﾄｰｸﾝ)がﾃﾞｰﾀ段階と逆になります。この状態報告は常に装置側により行われます。従って入力 (IN)指示で非正常終了を報告する場合、本来ﾃﾞｰﾀ ﾊﾟｹｯﾄを返す位置で対応するﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄにより応答することに注意してください。

入力 (IN)
ﾃﾞｰﾀ段階 =出力

ﾎｽﾄがﾃﾞｰﾀ段階中に出力 (OUT)ﾊﾟｹｯﾄを送信してﾃﾞｰﾀを送信したなら、装置は入力 (IN)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄに応じてﾃﾞｰﾀ部のない (ﾃﾞｰﾀ長 =0)ﾃﾞｰﾀ ﾊﾟｹｯﾄを送信することにより、そのﾃﾞｰﾀの受信成功で応答するでしょう。然しながら、ｴﾗｰが起きた場合は不能応答 (STALL)ﾊﾟｹｯﾄで、未だﾃﾞｰﾀ処理中の場合は以降の状態段階での再試行をﾎｽﾄに頼む否定応答 (NAK)ﾊﾟｹｯﾄで応答すべきです。

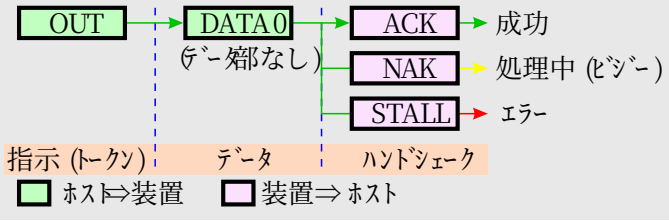
図 21.状態入力 (IN)段階の流れ



出力 (OUT)
ﾃﾞｰﾀ段階 =入力

ﾎｽﾄはﾃﾞｰﾀ段階中に入力 (IN)ﾊﾟｹｯﾄを送信してﾃﾞｰﾀを受信したなら、そのﾃﾞｰﾀの受信成功で応答しなければなりません。これは出力 (OUT)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄとそれに続くﾃﾞｰﾀ部のない (ﾃﾞｰﾀ長 =0)ﾃﾞｰﾀ ﾊﾟｹｯﾄをﾎｽﾄが送信することで行われます。装置はﾊﾝﾄﾞｼｪｰｸ ﾊﾟｹｯﾄで直ぐに状態を報告することができます。肯定応答 (ACK)は装置が命令を完了し、直ぐに他の命令の受付準備ができていることを示します。この命令の処理中にｴﾗｰが起きた場合、装置は不能応答 (STALL)で応答します。未だ処理中の場合は以降の状態段階での再試行をﾎｽﾄに頼む否定応答 (NAK)で応答します。

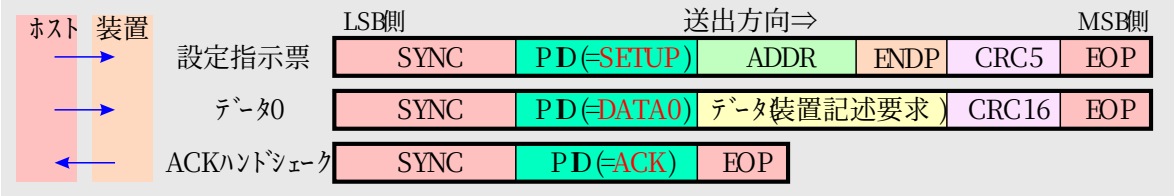
図 22.状態出力 (OUT)段階の流れ



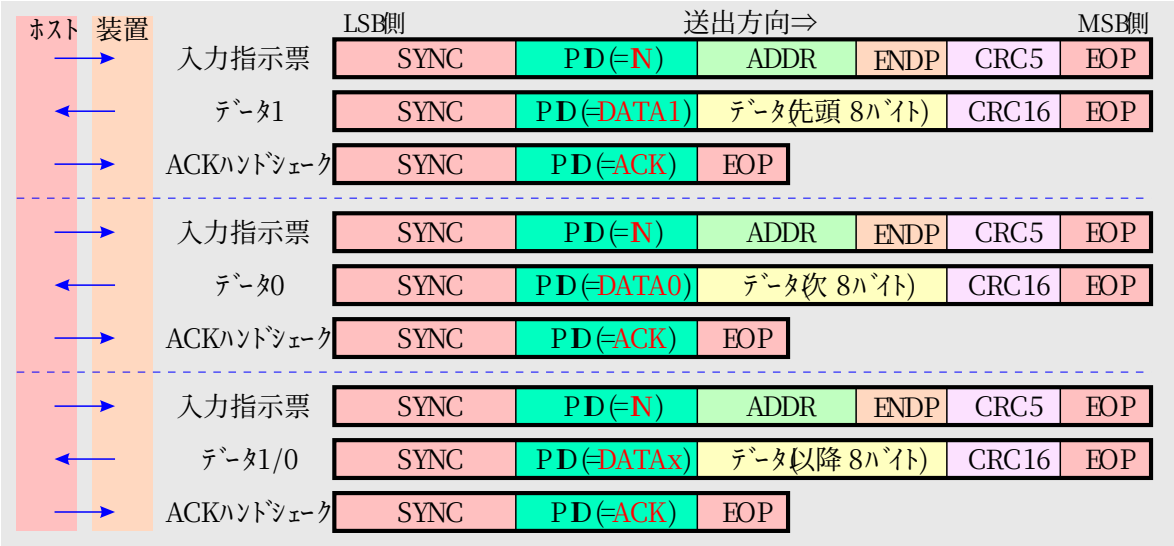
制御 (Contro)転送における大量ﾃﾞｰﾀ

例えばﾎｽﾄから列挙中に装置記述要求があり、その内容が 1つのﾃﾞｰﾀﾊﾟｹｯﾄに収まらない場合、それらのﾃﾞｰﾀは次の手順で送信されます。

ﾎｽﾄは後続のﾊﾟｹｯﾄが装置記述要求を含む設定 (SETUP)ﾊﾟｹｯﾄであることを装置に通知する設定 (SETUP)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄを送信します。このﾊﾟｹｯﾄのｱﾄﾞﾚｽ (ADDR)領域には装置記述を要求する装置のｱﾄﾞﾚｽが保持されます。ｴﾝﾄﾞﾎﾟｲﾝﾄ番号は既定ﾊﾟｲﾌﾟを指示するため、0であるべきです。その後ﾎｽﾄはﾃﾞｰﾀ(DATA0)ﾊﾟｹｯﾄを送信します。このﾊﾟｹｯﾄはﾃﾞｰﾀ部に装置記述要求を示す 8ﾋﾞｲﾄの値を持ちます。装置はｴﾗｰなしで設定 (SETUP)ﾊﾟｹｯﾄが正しく読み込まれたことを応答で返します。そのﾊﾟｹｯﾄが正しく読み込まれなかった場合、そのﾊﾟｹｯﾄを無視します。その後、ﾎｽﾄは一定時間経過後にそのﾊﾟｹｯﾄを再送信するでしょう。

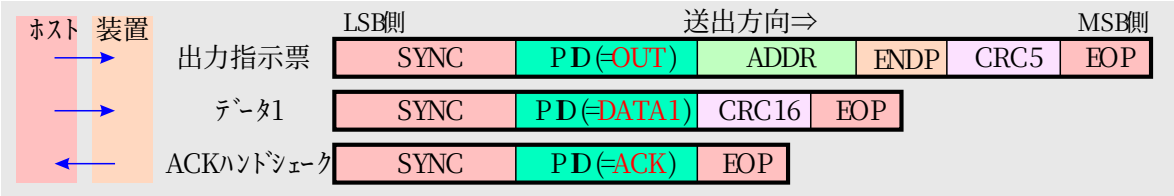


上の 3つのﾊﾟｹｯﾄは最初のUSB処理単位 (Transaction)を表します。装置は受信した 8ﾋﾞｲﾄを調べ、それが装置記述要求であることを知ります。そして装置は次からのUSB処理単位 (Transaction)で装置記述の送信を試みます。



上記のﾃﾞｰﾀﾊﾟｹｯﾄのﾃﾞｰﾀ部最大容量は 8ﾊﾞｲﾄと仮定しています。ﾎｽﾄはｴﾝﾄﾞﾎﾟｲﾝﾄからのﾃﾞｰﾀを直ぐに送信できることを装置に告げるために入力 (IN)指示票 (ﾄｰｸﾝ)を送信します。8ﾊﾞｲﾄを越えるﾃﾞｰﾀは最終ﾃﾞｰﾀを含む処理単位 (Transaction)を除いて、入力 (IN)指示票 (ﾄｰｸﾝ)に対応して 8ﾊﾞｲﾄ毎に送信されなければなりません。ﾎｽﾄはこの送信ﾃﾞｰﾀ毎に応答します。

全ﾃﾞｰﾀが送信されてしまうと、状態段階の処理単位 (Transaction)が続きます。全転送が成功の場合、ﾎｽﾄはこれを示すﾃﾞｰﾀ部なし(ﾃﾞｰﾀ長 =0)のﾃﾞｰﾀﾊﾟｹｯﾄを送信します。そして装置はこのﾃﾞｰﾀﾊﾟｹｯﾄに対してﾊﾝﾄﾞｼｪｰｸﾊﾟｹｯﾄで返答します。



割り込み (Interrupt)転送

割り込み転送の原則的な考え方はﾏｲｸﾛ ｺﾝﾄﾛｰﾗなどの割り込みと同様で、USB装置側が割り込みを生成します。然しながら、USB下ではﾎｽﾄに注意を促したくても、その前にﾎｽﾄによる巡回問い合わせ (ﾎﾟｰﾘﾝｸﾞ)まで待たなければなりません。割り込み転送には他の転送と比較して次の特徴があります。

- 保証された遅延
- 単方向のｽﾄﾘｰﾑ ﾊﾟｲﾌﾟ
- ｴﾗｰ検出と次周期での再試行

割り込み転送は、或る時点から一定の時間内に通信を開始しなければならない、一般的に非周期性のもので使用されます。割り込み要求はﾎｽﾄがﾃﾞｰﾀについて装置に問い合わせるまで装置によって順に記録保存されます。

割り込み転送でのﾊﾟｹｯﾄ内ﾃﾞｰﾀ部最大容量は、Low-Speed装置が 8ﾊﾞｲﾄ、Full-Speed装置が 64ﾊﾞｲﾄ、High-Speed装置が 1024ﾊﾞｲﾄです。

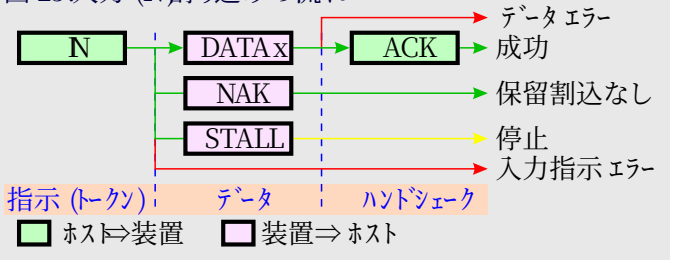
割り込み転送には割り込み入力 (IN)と割り込み出力 (OUT)の 2種類の処理単位 (Transaction)があります。

入力 (IN)

ﾎｽﾄは周期的に割り込みｴﾝﾄﾞﾎﾟｲﾝﾄを調べます。この周期の間隔は後述のｴﾝﾄﾞﾎﾟｲﾝﾄ記述で指定されます。各周期での調査はﾎｽﾄが入力 (IN)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄを送信することで始まります。この入力 (IN)指示票 (ﾄｰｸﾝ)が誤りの場合、装置はこのﾊﾟｹｯﾄを無視し、ﾊﾞｽ上で新規ﾊﾟｹｯﾄの監視を継続します。

割り込みが装置内に記録保存されている場合、入力 (IN)指示票 (ﾄｰｸﾝ)を受信した時にその割り込みに関連したﾃﾞｰﾀを含むﾃﾞｰﾀﾊﾟｹｯﾄを送信します。ﾎｽﾄでの受信が成功するとﾎｽﾄは肯定応答 (ACK)を返します。ﾃﾞｰﾀが不正な場合、ﾎｽﾄは情報を返しません。ﾎｽﾄが入力 (IN)指示票 (ﾄｰｸﾝ)で割り込みｴﾝﾄﾞﾎﾟｲﾝﾄを調査する時に割り込み状態が存在しない場合、装置は否定応答 (NAK)を送信することで、この状態を示します。割り込みｴﾝﾄﾞﾎﾟｲﾝﾄでｴﾗｰが発生している場合は、入力 (IN)指示票 (ﾄｰｸﾝ)の応答で代わりに不能応答 (STALL)を送信します。

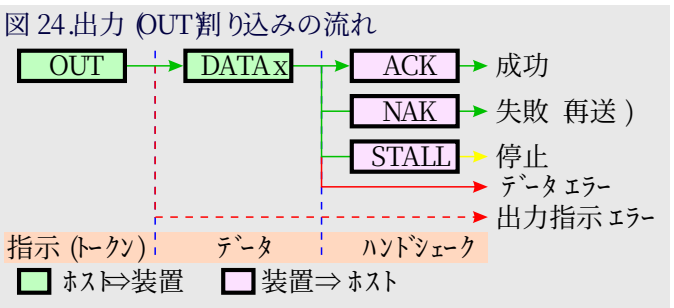
図 23.入力 (IN)割り込みの流れ



出力 (OUT)

ﾎｽﾄが装置への割り込みﾃﾞｰﾀ送信を欲するとき、出力 (OUT)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄとそれに続けて割り込みﾃﾞｰﾀを含むﾃﾞｰﾀﾊﾟｹｯﾄを送信します。出力 (OUT)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄまたはﾃﾞｰﾀﾊﾟｹｯﾄの何れかの部分が不正なら、装置はそのﾊﾟｹｯﾄを無視します。装置のｴﾝﾄﾞﾎﾟｲﾝﾄﾊﾞｯﾌｧが空で、ﾃﾞｰﾀがそのﾊﾞｯﾌｧ内に格納されたなら、そのﾃﾞｰﾀ受信が成功したことをﾎｽﾄに通知する肯定応答 (ACK)を送信します。そのｴﾝﾄﾞﾎﾟｲﾝﾄﾊﾞｯﾌｧが直前ﾊﾟｹｯﾄの処理中のために空でなかった場合、装置は否定応答 (NAK)を返します。但しｴﾝﾄﾞﾎﾟｲﾝﾄが何らかのｴﾗｰで停止 (または中止)状態の場合不能応答 (STALL)ﾊﾟｹｯﾄを返します。

図 24.出力 (OUT)割り込みの流れ



等時 (Isochronous)転送

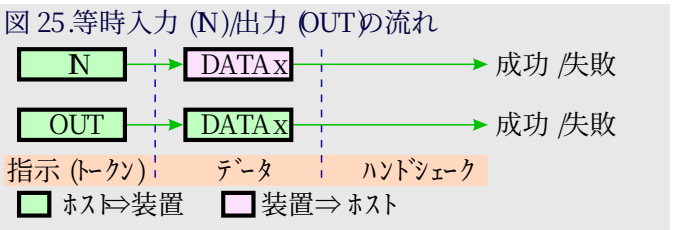
等時転送は周期的且つ継続的に起きます。通常、これらはｵｰﾃﾞｨｵやﾋﾞﾃﾞｵのように、時間に敏感な情報を含みます。これらでは誤りによる部分欠落が時間的に瞬間のため、多くの場合、気付かれないでしょう。再送の繰り返しによる時間軸の乱れの方がより深刻です。等時転送は他の転送と比較して次の特徴があります。

- USB帯域幅内での保証されたｱｸｾｽ
- 単方向のｽﾄﾘｰﾑ ﾊﾟｲﾌﾟ
- Full/High-Speedでのみ可能
- 一定時間内の遅延
- CRCでのｴﾗｰ検出、再試行なし
- PIDのDATA1/0交互使用なし

ﾃﾞｰﾀﾊﾟｹｯﾄのﾃﾞｰﾀ部最大容量は等時ｴﾝﾄﾞﾎﾟｲﾝﾄのｴﾝﾄﾞﾎﾟｲﾝﾄ記述内で指定され、Full-Speedでは最大 1023ﾊﾞｲﾄまで、High-Speedでは 1024ﾊﾞｲﾄまでです。この最大容量はﾊﾞｽの必要条件である帯域幅に影響するため、控えめな値を指定します。大容量指定は様々な容量指定での一連の代替ｲﾝﾀｰﾌｪｰｽにおいて優位になるかもしれません。列挙中にﾎｽﾄが帯域幅制限のために希望する等時ｴﾝﾄﾞﾎﾟｲﾝﾄを許可できない場合、単に失敗で終わらせず、縮小調停 (ﾌｫｰﾙﾊﾞｯｸ)が行われます。等時ｴﾝﾄﾞﾎﾟｲﾝﾄで送られるﾃﾞｰﾀは予め調停された容量より少なくでき、処理単位 (Transaction)間で長さを変えることができます。

右図は等時入力 (IN)/出力 (OUT)転送の処理単位 (Transaction)形式を示します。等時転送にはﾊﾝﾄﾞｼｪｰｸ段がなく、ｴﾗｰや停止状態の報告ができません。

図 25.等時入力 (IN)/出力 (OUT)の流れ



大量 (Bulk)転送

大量転送は突発的大量ﾃﾞｰﾀに使用されます。これにはﾌﾟﾘﾝﾀやｽｷｬﾅのﾃﾞｰﾀ転送などが該当します。大量転送ではﾃﾞｰﾀ部に対するCRC16領域の形式でのｴﾗｰ訂正と、ｴﾗｰなしでﾃﾞｰﾀが送受信されることを保証するｴﾗｰ検出/再送機構が提供されます。

大量転送は他の全ての処理単位 (Transaction)が割り当てられてしまった後の余った未割り当て帯域幅を使用します。割り込み(Interrupt)転送や等時 (Isochronous)転送でﾊﾞｽが忙しい場合、大量ﾃﾞｰﾀはﾊﾞｽ上で少しずつゆっくりと転送されるかもしれません。この結果、遅れが保証されないため、大量転送は時間に敏感でない通信に対してだけ使用されるべきです。大量転送には他の転送と比較して次の特徴があります。

- 突発的大量ﾃﾞｰﾀ転送に使用
- 単方向のｽﾄﾘｰﾑ ﾊﾟｲﾌﾟ
- Full/High-Speedでのみ可能
- 帯域幅や最小遅延の保証なし
- CRCでのｴﾗｰ検出と配給保証

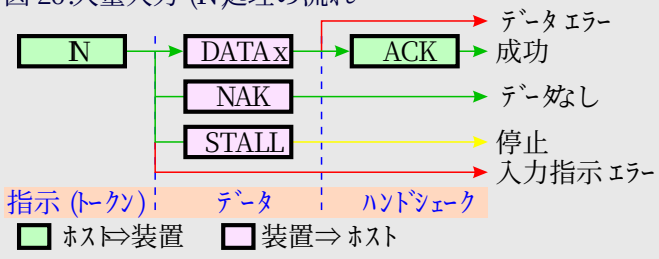
大量転送はFull-Speed装置とHigh-Speed装置でだけ支援されます。Full-Speedでのﾃﾞｰﾀ部最大容量は8,16,32,64ﾊﾞｲﾄの何れかで、High-Speedでは512ﾊﾞｲﾄまでの長さにできます。ﾃﾞｰﾀがこの最大容量未満の場合、残り部分を0で埋める必要はありません。大量転送は、正確な要求ﾃﾞｰﾀ量のﾊﾟｹｯﾄ転送、最大容量未満のﾊﾟｹｯﾄ転送、ﾃﾞｰﾀなし(ﾃﾞｰﾀ長=0)で完了すると考えられます。

大量転送には大量入力 (IN)と大量出力 (OUT)の2種類の処理単位 (Transaction)があります。

入力 (IN)

ﾎｽﾄは大量ﾃﾞｰﾀの受信準備が整うと入力 (IN)指示票 (ﾄｰｸﾝ)を送信します。装置がｴﾗｰと共にこの入力 (IN)指示票 (ﾄｰｸﾝ)を受信した場合、そのﾊﾟｹｯﾄを無視します。その入力 (IN)指示票 (ﾄｰｸﾝ)が正しく受信されると、装置は送るべき大量ﾃﾞｰﾀを含むﾃﾞｰﾀ(DATAx)ﾊﾟｹｯﾄか、またはｴﾝﾄﾞﾎﾟｲﾝﾄのｴﾗｰを示す不能 (STALL)ﾊﾟｹｯﾄ、またはｴﾝﾄﾞﾎﾟｲﾝﾄが作業中で一時的に送るﾃﾞｰﾀがないことをﾎｽﾄへ示す否定応答 (NAK)ﾊﾟｹｯﾄの何れかを返すことができます。

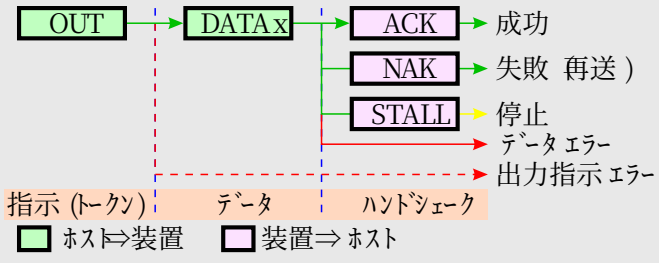
図 26.大量入力 (IN)処理の流れ



出力 (OUT)

ﾎｽﾄが装置への大量ﾃﾞｰﾀ送信を欲するとき、出力 (OUT)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄとそれに続けて大量ﾃﾞｰﾀを含むﾃﾞｰﾀﾊﾟｹｯﾄを送信します。出力 (OUT)指示票 (ﾄｰｸﾝ)ﾊﾟｹｯﾄまたはﾃﾞｰﾀﾊﾟｹｯﾄの何れかの部分が不正なら、装置はそのﾊﾟｹｯﾄを無視します。装置のｴﾝﾄﾞﾎﾟｲﾝﾄﾊﾞｯﾌｧが空で、ﾃﾞｰﾀがそのﾊﾞｯﾌｧ内に格納されたなら、そのﾃﾞｰﾀ受信が成功したことをﾎｽﾄに通知する肯定応答 (ACK)を送信します。そのｴﾝﾄﾞﾎﾟｲﾝﾄﾊﾞｯﾌｧが直前ﾊﾟｹｯﾄの処理中のために空でなかった場合、装置は否定応答 (NAK)を返します。但しｴﾝﾄﾞﾎﾟｲﾝﾄが何らかのｴﾗｰで停止 (または中止)状態の場合不能応答 (STALL)ﾊﾟｹｯﾄを返します。

図 27.大量出力 (OUT)処理の流れ



USB記述子 (Descriptor)

全USB装置はどんな装置で、誰が作り、どのUSBﾊﾞｰｼﾞｮﾝを支援し、何種類の方法で設定でき、ｴﾝﾄﾞﾎﾟｲﾝﾄの数とその種別は、などのような情報をﾎｽﾄに告げるため記述子の階層を持ちます。多くで共通するUSB記述子には次のものがあります。

- 装置記述 (Device Descriptor)
- 設定記述 (Configuration Descriptor)
- ｲﾝﾀｰﾌｪｰｽ記述 (Interface Descriptor)
- ｴﾝﾄﾞﾎﾟｲﾝﾄ記述 (Endpoint Descriptor)
- 文字列記述 (Strings Descriptor)

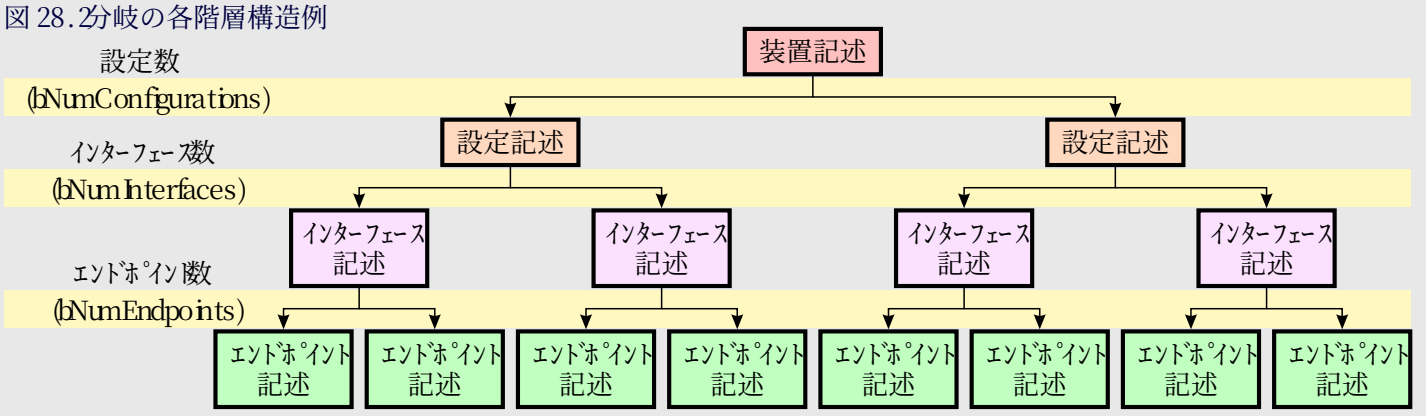
USB装置は 1つの装置記述しか持つことができません。装置記述は装置が適合するUSBﾊﾞｰｼﾞｮﾝ、適切なﾄﾞﾗｲﾊを読み込むために使用される供給者識別 (VID)と製品識別 (PID)、装置が持っている利用可能な設定数のような情報を含みます。この設定数は設定記述が何個かに別れて続くことを示します。

設定記述は、この特定設定で使用する電力量、装置がﾊﾞｽ給電か自己給電か、この設定が持つｲﾝﾀｰﾌｪｰｽ数のような値を指示します。装置が列挙されるとき、ﾎｽﾄは装置記述を読み、どの設定を許可すべきかを決定できます。これは一度に 1つの設定だけを許可できます。

例えば、高電力ﾊﾞｽ給電設定と自己給電設定を持つことが可能な場合に、装置が主電力供給元と共にﾎｽﾄへ接続されると、ﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊは主電源なしで給電される装置を許可する高電力ﾊﾞｽ給電の許可を選択するかもしれません。けれどもﾉｰﾄPCに接続された場合、主電源に装置を接続する必要がある第 2の設定 (自己給電)を許可にできます。

設定の指定は電力の違いに制限される訳ではありません。各設定は同じ方法、同じ電流での給電が有り得ますが、まだ下層に異なるｲﾝﾀｰﾌｪｰｽまたはｴﾝﾄﾞﾎﾟｲﾝﾄの組み合わせを持ちます。けれども設定変更が各ｴﾝﾄﾞﾎﾟｲﾝﾄの全動作の停止を要求することに注目すべきです。USBがこの柔軟性を提供する一方、複数の設定を持つ実際の装置は非常に僅かです。

図 28.2分岐の各階層構造例



ｲﾝﾀｰﾌｪｰｽ記述は装置の単一機能を実行する機能動作群内のｴﾝﾄﾞﾎﾟｲﾝﾄ群またはﾍｯﾀﾞと見ることができます。例えば多機能ﾌﾟﾘﾝﾀ/ﾌｧｯｸｽ/ｽｷｬﾅの場合、最初のｲﾝﾀｰﾌｪｰｽ記述はﾌﾟﾘﾝﾀ機能のｴﾝﾄﾞﾎﾟｲﾝﾄを、2つ目のｲﾝﾀｰﾌｪｰｽ記述はﾌｧｯｸｽ機能を、3つ目はｽｷｬﾅ機能を記述できます。設定記述と異なり、これらは一度に 1つしか許可できないと言う制限はありません。装置は一度に許可された複数のｲﾝﾀｰﾌｪｰｽ記述を持てます。

ｲﾝﾀｰﾌｪｰｽ記述はｲﾝﾀｰﾌｪｰｽ番号を指示するbInterfaceNumber領域と動作中の設定変更をｲﾝﾀｰﾌｪｰｽに許すbAlternativeSetting領域を持ちます。例えば装置がｲﾝﾀｰﾌｪｰｽ 0とｲﾝﾀｰﾌｪｰｽ 1を持っている場合、ｲﾝﾀｰﾌｪｰｽ 0は最初のｲﾝﾀｰﾌｪｰｽ記述)を表す 0に設定されたbInterfaceNumberと (既定を表す)0のbAlternativeSettingで示されます。

ｲﾝﾀｰﾌｪｰｽ 1は 2番目のｲﾝﾀｰﾌｪｰｽを表す 1に設定されたbInterfaceNumberと (既定を表す)0のbAlternativeSettingで示されます。このｲﾝﾀｰﾌｪｰｽ 1に別の代替設定を付加する場合は、ｲﾝﾀｰﾌｪｰｽ 1を示す 1に設定されたbInterfaceNumberと、今回は代替設定であることを表す 1をbAlternativeSettingに設定します。

これらの設定が許可されるとbAlternativeSettingが既定を表す 0である最初の 2つのｲﾝﾀｰﾌｪｰｽ記述が使用されます。然しながら、操作中にﾎｽﾄは別のｲﾝﾀｰﾌｪｰｽ記述を許可するため、代替設定を持つ (=1)ｲﾝﾀｰﾌｪｰｽ 1へ直接ｲﾝﾀｰﾌｪｰｽ設定 (SET_INTERFACE)要求を送信できます。

これはｲﾝﾀｰﾌｪｰｽ 0でﾃﾞｰﾀを送信しながら、一方でｲﾝﾀｰﾌｪｰｽ 0への影響なしにｲﾝﾀｰﾌｪｰｽ 1に関連したｴﾝﾄﾞﾎﾟｲﾝﾄ設定を変更できることで、2つの設定を持つ以上の優位性を与えます。

各ｴﾝﾄﾞﾎﾟｲﾝﾄ記述は各々のｴﾝﾄﾞﾎﾟｲﾝﾄに対する転送種別、方向、巡回検査 (ﾎﾟｰﾘﾝｸﾞ)間隔及び最大ﾃﾞｰﾀ容量を指定するために使用されます。既定の制御 (Contorol)用ｴﾝﾄﾞﾎﾟｲﾝﾄ(ｴﾝﾄﾞﾎﾟｲﾝﾄ0)は常に制御ｴﾝﾄﾞﾎﾟｲﾝﾄであると仮定され、このようなものは記述を持ちません。

USB記述子 (Descriptor)の構成

全ての記述子は共通形式で作成されます。最初のﾊﾞｲﾄは記述子の長さ(ﾊﾞｲﾄ数)を示し、第2ﾊﾞｲﾄは記述子種別を表します。記述子の長さが仕様での定義より短い場合、ﾎｽﾄはこれを無視します。けれどもその長さが予測より大きい場合、ﾎｽﾄは余分なﾊﾞｲﾄを無視し、実際に返された長さの終了後に次の記述子の検索を始めます。

表4.記述子先頭の共通部分

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味
0	bLength	1	数値	記述子全体のﾊﾞｲﾄ数
1	bDescriptorType	1	定数	記述子種別を表す数値
2	～	～	～	以降記述子内容(種別により変化)

装置記述子 (Device Descriptor)

USB装置の装置記述は装置全体を表します。USB装置は1つの装置記述しか持てないので、いくつかの基本的な事しか示しませんが、それらは支援するUSBﾊﾞｰｼﾞｮﾝ、ﾊﾟｹｯﾄのﾃﾞｰﾀ部最大容量、供給者識別(VID)と製品識別(PID)、装置で利用可能な設定数のように装置についての重要な情報です。

表5.装置記述(Device Descriptor)の構成

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味
0	bLength	1	数値	本記述全体のﾊﾞｲﾄ数
1	bDescriptorType	1	定数	装置記述を表す$01
2	bcdUSB	2	BCD	装置が適合するUSBﾊﾞｰｼﾞｮﾝ(例:USB2.0の場合$0200)
4	bDeviceClass	1	ｸﾗｽｺｰﾄﾞ	この値が0の場合は各ｲﾝﾀｰﾌｪｰｽが自身のｸﾗｽｺｰﾄﾞを指定します。$FFの場合、ｸﾗｽｺｰﾄﾞは供給者によって指定されます。他の値は有効なｸﾗｽｺｰﾄﾞです。
5	bDeviceSubClass	1	補助ｸﾗｽｺｰﾄﾞ	USB協会により割り当てられた補助ｸﾗｽｺｰﾄﾞ
6	bDeviceProtocol	1	規約ｺｰﾄﾞ	USB協会により割り当てられた規約ｺｰﾄﾞ
7	bMaxPacketSize	1	数値	ｴﾝﾄﾞﾎﾟｲﾝﾄ0の最大ﾃﾞｰﾀ容量(有効値は8,16,32,64)
8	idVendor	2	識別ｺｰﾄﾞ	USB協会により割り当てられた供給者識別ｺｰﾄﾞ
10	idProduct	2	識別ｺｰﾄﾞ	製造(供給)者により割り当てられた製品識別ｺｰﾄﾞ
12	bcdDevice	2	BCD	装置ﾊﾞｰｼﾞｮﾝ番号
14	iManufacturer	1	指標番号	製造(供給)者文字列記述の指標番号
15	iProduct	1	指標番号	製品文字列記述の指標番号
16	iSerialNumber	1	指標番号	製品通し番号文字列記述の指標番号
17	bNumConfigurations	1	数値	使用可能な設定数

- bcdUSBは装置が支援する最上位のUSBﾊﾞｰｼﾞｮﾝを報告します。この値は$VVMS形式の2進化10進値で、VVが主ﾊﾞｰｼﾞｮﾝ、Mが補助ﾊﾞｰｼﾞｮﾝ、Sが補足ﾊﾞｰｼﾞｮﾝ番号です。例えばUSB1.0は$0100、USB1.1は$0110、USB2.0は$0200として報告されます。
- bDeviceClass,bDeviceSubClass,bDeviceProtocolは装置に対応するｸﾗｽﾄﾞﾗｲﾊﾞを探すためにｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑによって使用されます。多くのｸﾗｽ仕様がｲﾝﾀｰﾌｪｰｽ段階でのｸﾗｽ確認を選ぶため、通常はbDeviceClassだけが装置段階で$00として設定されます。これは複数ｸﾗｽの支援を1つの装置に許します。
- bMaxPacketSizeはｴﾝﾄﾞﾎﾟｲﾝﾄ0の最大ﾃﾞｰﾀ容量を報告します。全ての装置はｴﾝﾄﾞﾎﾟｲﾝﾄ0を支援しなければなりません。
- idVendor,idProductは装置に対応するﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞを探すためにｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑによって使用されます。供給者識別ｺｰﾄﾞはUSB協会によって割り当てられます。
- bcdDeviceはbcdUSBと同じ形式で、装置のﾊﾞｰｼﾞｮﾝ番号を提供するために使用されます。この値は開発業者によって割り当てられます。
- iManufacturer,iProduct,iSerialNumberは各々製造業者、製品、通し番号の詳細を提供するために存在します。これらは必ずしも対応する文字列記述子を持つ必要はなく、その場合は値として0が使用されるべきです。
- bNumConfigurationsは装置が現在の速度で支援する設定数を定義します。

設定記述子 (Configuration Descriptor)

多くの装置は単純で1つの設定しか持ちませんが、USB装置はいくつかの異なる設定を持つことができます。この設定記述は装置がどう給電され、どれくらいの最大消費電力で、いくつのインターフェース数を持つかを指定します。このため装置がバス給電される場合と外部電力の場合で2つの設定を持つことが可能です。これはインターフェース記述へのヘッダであるため、異なる転送種別を使用する別の設定を持つこともできます。

一旦ホストにより全ての設定が調べられてしまうと、ホストは設定記述の1つのbConfigurationValueと一致した0以外の値で設定選択(SET_CONFIGURATION)命令を送信します。これは希望する設定の選択に使用されます。

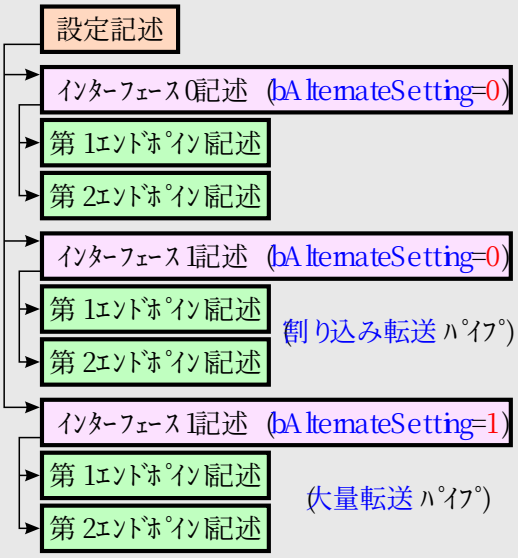
表6.設定記述(Configuration Descriptor)の構成

オフセット	領域名	バイト数	値種別	意味	
0	bLength	1	数値	本記述全体のバイト数	
1	bDescriptorType	1	定数	設定記述を表す $02	
2	wTotalLength	2	数値	返されるデータの総バイト数	
4	bNumInterfaces	1	数値	インターフェース数	
5	bConfigurationValue	1	数値	この設定選択の引数として使用される値	
6	iConfiguration	1	指標番号	本設定を記述する文字列記述の指標番号	
7	bmAttributes	1	ビット値	ビット7	バス給電
				ビット6	自己給電
				ビット5	遠隔起動
				ビット4～0	予約(=0)
8	bMaxPower	1	数値	2mA単位の最大消費電流値	

設定記述が読まれるとき、その設定階層内のインターフェースとエンドポイントに関連する全記述を含む階層全体(右図参照)を返します。wTotalLengthはその階層内のバイト数を反映します。

- bNumInterfacesはこの設定に存在するインターフェース数を示します。
- bConfigurationValueはこの設定を選択する設定選択(SET_CONFIGURATION)要求で使用されます。
- iConfigurationは人間が読める形で記載する文字列記述の指標番号です。
- bmAttributesはこの設定についての電源種別値を指定します。装置が自己給電なら、ビット6を設定(=1)します。ビット7はバス給電装置であることを示すためにUSB1.1で使用されますが、現在ではこれがbMaxPowerで行われます。装置が何らかの電力をバスから使用する場合、自己給電またはバス給電のどちらかとし、bMaxPowerで消費電力を報告しなければなりません。
- bMaxPowerは装置がバスから得る最大電力を定義します。これは2mA単位で行い、従って最大約500mAが指定できます。この仕様はバス(VBUS)から500mAを越えない範囲で電流を流すことを高電力バス給電装置に許します。装置が外部電力を失った場合であっても、bMaxPowerでの指定より多くの電流を流してはなりません。

図29.1設定全体階層構成例



ｲﾝﾀｰﾌｪｰｽ記述子 (Interface Descriptor)

ｲﾝﾀｰﾌｪｰｽ記述は装置の単一機能を実行する起動動作群内の ｴﾝﾄﾞﾎﾟｲﾝﾄ群または ﾍｯﾀﾞと見ることができます。ｲﾝﾀｰﾌｪｰｽ記述は次の形式に従います。

表 7.ｲﾝﾀｰﾌｪｰｽ記述 (Interface Descriptor)の構成

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味
0	bLength	1	数値	本記述全体の ﾊﾞｲﾄ数
1	bDescriptorType	1	定数	ｲﾝﾀｰﾌｪｰｽ記述を表す $04
2	bInterfaceNumber	1	数値	ｲﾝﾀｰﾌｪｰｽ番号
3	bAlternateSetting	1	数値	代替設定の選択で使用される代替番号値
4	bNumEndpoints	1	数値	この ｲﾝﾀｰﾌｪｰｽで使用されるｴﾝﾄﾞﾎﾟｲﾝﾄ数
5	bInterfaceClass	1	ｸﾗｽ ｺｰﾄﾞ	USB協会により割り当てられた ｸﾗｽ ｺｰﾄﾞ
6	bInterfaceSubClass	1	補助 ｸﾗｽ ｺｰﾄﾞ	USB協会により割り当てられた補助 ｸﾗｽ ｺｰﾄﾞ
7	bInterfaceProtocol	1	規約 ｺｰﾄﾞ	規約 ｺｰﾄﾞ
8	iInterface	1	指標番号	この ｲﾝﾀｰﾌｪｰｽを記載する文字列記述の指標番号

- bInterfaceNumberはこの ｲﾝﾀｰﾌｪｰｽの番号を示します。
- bAlternativeSettingは代替 ｲﾝﾀｰﾌｪｰｽの指定に使用できます。これらの代替 ｲﾝﾀｰﾌｪｰｽは ｲﾝﾀｰﾌｪｰｽ選択 (SET_INTERFACE要求で選択することができます。
- bNumEndpointsはこの ｲﾝﾀｰﾌｪｰｽで使用されるｴﾝﾄﾞﾎﾟｲﾝﾄ数を示します。この値はｴﾝﾄﾞﾎﾟｲﾝﾄ0を除外すべきで、この ｲﾝﾀｰﾌｪｰｽに連なるｴﾝﾄﾞﾎﾟｲﾝﾄ記述の数を示すために使用されます。
- bInterfaceClass,bInterfaceSubClass,bInterfaceProtocolは支援する ｸﾗｽ(例えば、HID,通信 (communications),大容量記憶装置 (mass storage)など)の指定に使用できます。これはその装置に対する特定 ﾄﾞﾗｲﾊﾞの作成なしで、ｸﾗｽ ﾄﾞﾗｲﾊﾞの使用を多くの装置に許します。
- iInterfaceはこの ｲﾝﾀｰﾌｪｰｽの文字列記述を許します。

ｴﾝﾄﾞﾎﾟｲﾝﾄ記述子 (Endpoint Descriptor)

ｴﾝﾄﾞﾎﾟｲﾝﾄ記述はｴﾝﾄﾞﾎﾟｲﾝﾄ0以外のｴﾝﾄﾞﾎﾟｲﾝﾄ記述に使用されます。ｴﾝﾄﾞﾎﾟｲﾝﾄ0は常に制御 (Contorol)ｴﾝﾄﾞﾎﾟｲﾝﾄとの認識で、どんな記述が要求されるよりも前にそのように設定されます。

表 8.ｴﾝﾄﾞﾎﾟｲﾝﾄ記述 (Endpoint Descriptor)の構成

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味			
0	bLength	1	数値	本記述全体のﾊﾞｲﾄ数			
1	bDescriptorType	1	定数	ｴﾝﾄﾞﾎﾟｲﾝﾄ記述を表す $05			
2	bEndpointAddress	1	ﾋﾞｯﾄ分割値	ﾋﾞｯﾄ7	方向 0=出力 ,1=入力 (制御ｴﾝﾄﾞﾎﾟｲﾝﾄ時無効)		
				ﾋﾞｯﾄ6～4	予約 (=0)		
				ﾋﾞｯﾄ3～0	ｴﾝﾄﾞﾎﾟｲﾝﾄ番号		
3	bmAttributes	1	ﾋﾞｯﾄ分割値	ﾋﾞｯﾄ7,6	予約 (=0)		
				ﾋﾞｯﾄ5,4 (使用種別)	11	予約	等時転送以外時は予約 (=0)
					10	明示逆送ﾃﾞｰﾀｴﾝﾄﾞﾎﾟｲﾝﾄ	
					01	暗黙逆送ﾃﾞｰﾀｴﾝﾄﾞﾎﾟｲﾝﾄ	
					00	通常ﾃﾞｰﾀｴﾝﾄﾞﾎﾟｲﾝﾄ	
				ﾋﾞｯﾄ3,2 (同期種別)	11	同期	
					10	適応	
					01	非同期	
					00	同期なし	
				ﾋﾞｯﾄ1,0 (転送種別)	11	割り込み (Interrupt)	
					10	大量 (Bulk)	
					01	等時 (Isochronous)	
					00	制御 (Contorol)	
4	wMaxPacketSize	2	数値	このｴﾝﾄﾞﾎﾟｲﾝﾄの最大ﾃﾞｰﾀ容量 (ﾊﾞｲﾄ数)			
6	bInterval	1	数値	ﾌﾚｰﾑ単位のｴﾝﾄﾞﾎﾟｲﾝﾄ調査間隔 (制御と大量転送時は無効) 等時転送時は常に 1、割り込み転送時は 1～255			

- bEndpointAddressはこの記述でどんなｴﾝﾄﾞﾎﾟｲﾝﾄが記述されるかを示します。
- bmAttributesは転送種別を指定します。これは制御 (Contorol)、割り込み (Interrupt)、等時 (Isochronous)、大量 (Bulk)の何れかにできます。等時ｴﾝﾄﾞﾎﾟｲﾝﾄが指定されたなら、同期種別や使用種別のような付加属性が選択できます。
- wMaxPacketSizeはこのｴﾝﾄﾞﾎﾟｲﾝﾄの最大ﾃﾞｰﾀﾊﾞｲﾄ数を示します。
- bIntervalは転送を行うための調査間隔を指定します。この単位はﾌﾚｰﾑ/ﾏｲｸﾛﾌﾚｰﾑで表され、従ってLow/Full-Speed装置では1ms、High-Speed装置では125μsのどちらかと等しくなります。

文字列記述子 (String Descriptor)

文字列記述は任意使用で、人間が読める情報を提供します。これらが使用されない場合、どの記述の文字列指標番号も文字列記述を利用できないことを示す 0に設定されなければなりません。

文字列は ﾕﾆｺｰﾄ形式で符号化され、複数言語の支援ができます。文字列指標番号 0は支援する言語一覧を返すために使用されます。USB言語識別 (ID)の一覧は付録を参照してください。

表 9.文字列指標番号 0の文字列記述構造

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味
0	bLength	1	数値	本記述全体のﾊﾞｲﾄ数
1	bDescriptorType	1	定数	文字列記述を表す $03
2	wLANGID(0)	2	識別ｺｰﾄﾞ	支援する言語番号 0の言語識別 (ID) (例 :$0409=米英語)
4	wLANGID(1)	2	識別ｺｰﾄﾞ	支援する言語番号 1の言語識別 (ID) (例 :$0411=日本語)
～	～	～	～	～
2n+2	wLANGID(n)	2	識別ｺｰﾄﾞ	支援する言語番号 nの言語識別 (ID)

上の文字列記述は文字列記述 0の形式を示します。ﾎｽﾄはどの言語が利用可能かを調べるために、この記述を読むべきです。希望する言語が支援されていれば、文字列記述を得る記述取得 (GET_DESCRIPTOR)要求の wIndex領域で言語識別 (ID)を送ることによりそれを参照できます。

それに続く文字列は以下の形式で運ばれます。

表 10.文字列記述構造

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味
0	bLength	1	数値	本記述全体のﾊﾞｲﾄ数
1	bDescriptorType	1	定数	文字列記述を表す $03
2	bString	n	ﾕﾆｺｰﾄﾞ	ﾕﾆｺｰﾄﾞで符号化された文字列

設定 (Setup)ﾊﾟｹｯﾄ

全ての装置は既定 ﾊﾟｲﾌﾟ上の設定 (Setup)ﾊﾟｹｯﾄに応答しなければなりません。この設定 (Setup)ﾊﾟｹｯﾄは装置の検出や設定に使用され、USB装置の ｱﾄﾞﾚｽ設定、装置記述の要求、ｴﾝﾄﾞﾎﾟｲﾝﾄの状態検査のような共通機能を実行します。

USBに適合したﾎｽﾄは処理されるべき全ての要求について最大 5秒間待ちます。また特定の要求に対しては厳密なﾀｲﾐﾝｸﾞも指定されています。

- ﾃﾞｰﾀ段階を除く標準装置要求 (Standard Device Request)は 50ms以内に完了しなければなりません。
- 標準装置要求 (Standard Device Request)の ﾃﾞｰﾀ段階はその要求後または直前の送出 ﾊﾟｹｯﾄ完了後から500ms以内にﾃﾞｰﾀ ﾊﾟｹｯﾄの送信を開始しなければなりません。
- 標準装置要求 (Standard Device Request)の状態 (Status)段階は最後の ﾃﾞｰﾀ ﾊﾟｹｯﾄ送信完了後、50ms以内に完了されなければなりません。
- ｱﾄﾞﾚｽ設定 (SET_ADDRESS)命令は状態応答までを50ms以内に処理しなければなりません。装置はその後、次の命令が送られる前にｱﾄﾞﾚｽを変更するために 2msの余裕が与えられます。

これらの制限時間は低速装置であっても充分に受け入れ可能ですが、ﾃﾞﾊﾞｯｸﾞが制限され得ます。この 50ms制限時間は、内部ﾚｼﾞｽﾀを調べるための ICEによる一時停止や 1命令実行、非同期 ｼﾘｱﾙ ﾎﾟｰﾄの 9600bpsでの ﾃﾞｰﾀ送信のような多くの ﾃﾞﾊﾞｯｸﾞについて問題になります。従ってUSBの ﾃﾞﾊﾞｯｸﾞでは一般的なﾏｲｸﾛ ｺﾝﾄﾛｰﾗでの ﾃﾞﾊﾞｯｸﾞと異なる何らかの ﾃﾞﾊﾞｯｸﾞ方法が必要となります。

各要求は以下の形式を持つ 8ﾊﾞｲﾄ長の設定 (Setup)ﾊﾟｹｯﾄで始まります。

表 11.設定 (Setup)ﾊﾟｹｯﾄ構成

ｵﾌｾｯﾄ	領域名	ﾊﾞｲﾄ数	値種別	意味		
0	bmRequestType	1	ﾋﾞｯﾄ割数値	要求の特性種別		
				ﾋﾞｯﾄ7 (ﾃﾞｰﾀ転送方向)	0	ﾎｽﾄ⇦装置
					1	装置⇨ ﾎｽﾄ
				ﾋﾞｯﾄ6,5 (種別)	0	標準 (Standard)
					1	ｸﾗｽ (Class)
					2	供給者 (Vendor)
					3	予約
				ﾋﾞｯﾄ4～0 (受け取り部)	0	装置 (Device)
					1	ｲﾝﾀｰﾌｪｰｽ (Interface)
					2	ｴﾝﾄﾞﾎﾟｲﾝﾄ (Endpoint)
					3	その他
					4～31	予約
1	bRequest	1	値	要求指定		
2	wValue	2	値	要求に従って変化するﾜｰﾄ領域		
4	wIndex	2	指標または相対位置	要求に従って変化するﾜｰﾄ領域 (通常、指標または相対位置の進行に使用)		
6	wLength	2	数値	ﾃﾞｰﾀ状態の場合、転送 ﾊﾞｲﾄ数		

bmRequestTypeは要求に対する受け取り部、方向、要求種別を決めます。

bRequestは要求種別を決めます。通常、bmRequestTypeは解析され、標準装置要求 (Standard Device Request)処理、標準 ｲﾝﾀｰﾌｪｰｽ要求 (Standard Interface Request)処理、標準 ｴﾝﾄﾞﾎﾟｲﾝﾄ要求 (Stabdard Endpoint Request)処理、ｸﾗｽ装置要求 (Class Device Request)処理などのようないくつかの処理に分岐して実行されます。設定 (Setup)ﾊﾟｹｯﾄをどう解析するかは自由です。例えば初めに bRwquestを解析し、その後要求毎に種別と受け取り部を調べることもできます。

標準要求 (Standard Request)は全USB装置で共通です。ｸﾗｽ要求 (Class Request)はﾄﾞﾗｲﾊﾞの ｸﾗｽに対して共通です。例えばHIDｸﾗｽに適合する全装置はその ｸﾗｽで共通の特有の要求群を持ちます。これらは通信 (Communications)ｸﾗｽに適合する装置とも、大容量記憶装置 (Mass storage)ｸﾗｽに適合する装置とも異なります。

この外に供給者定義の要求 (Request)があります。これらは設計 (開発)者が割り当てることができる要求です。通常、これらは装置毎に異なり、独自機能の構築に使用されます。

共通の要求は各種機能を実行する各受け取り部を基準に直接行えます。例えば状態取得 (GET_STATUS)標準要求は装置、ｲﾝﾀｰﾌｪｰｽ、ｴﾝﾄﾞﾎﾟｲﾝﾄに対して直接行えます。装置に対して行われた場合、装置は自己給電かどうかと遠隔起動の状態を示すﾌﾗｸﾞを返します。けれども同じ要求がｲﾝﾀｰﾌｪｰｽに行われた場合は常に 0値を返し、ｴﾝﾄﾞﾎﾟｲﾝﾄについてはそのｴﾝﾄﾞﾎﾟｲﾝﾄに対する停止 ﾌﾗｸﾞを返します。

wValueとwIndexは要求に関するﾊﾟﾗﾒｰﾀを許し、wLengthはﾃﾞｰﾀ段階で転送されるべきﾊﾞｲﾄ数を指定するために使用されます。

標準要求 (Standard Request)

標準装置要求 (Standard Device Request)は全てのUSB装置で実装が必要です。多くのファームウェアでは受け取り部で設定 (Setup)パケットの解析を行うと思われ、以降はこれらに沿って記述されます。

標準装置要求 (Standard Device Request)

これらは現在8つの標準装置要求があり、下表で示されます。

表 12.標準装置要求 (Stabdard Device Request)一覧

bmRequestType	bRequest	wValue	wIndex	wLength	データ
1 00 00000	GET_STATUS ($00)	0	0	2	装置状態値
0 00 00000	CLEAR_FEATURE ($01)	機能選択値	0	0	なし
0 00 00000	SET_FEATURE ($03)	機能選択値	0	0	なし
0 00 00000	SET_ADDRESS ($05)	装置アドレス	0	0	なし
1 00 00000	GET_DESCRIPTOR ($06)	記述銃別/指標番号	0または言語ID	記述長	記述内容
0 00 00000	SET_DESCRIPTOR ($07)	記述銃別/指標番号	0または言語ID	記述長	記述内容
1 00 00000	GET_CONFIGURATION ($08)	0	0	1	設定値(番号)
0 00 00000	SET_CONFIGURATION ($09)	設定番号	0	0	なし

■ 装置を指示した状態取得 (GET_STATUS)要求はデータ段階で次の形式の2バイトを返します。

図 30.状態取得要求に対するデータ

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
内容	予約														遠隔起動	自己給電

ビット0が1ならば装置が自己給電であることを示します。0ならば装置はバス給電です。ビット1が1ならば装置が遠隔起動可能なことを示し、ホストは一時停止 (Suspend)中の装置を通常動作へ起動できます。この遠隔起動ビットは機能設定 (SET_FEATURE)と機能解除 (CLEAR_FEATURE)の要求でDEVICE_REMOTE_WAKEUP ($01)機能選択値により設定/解除が行えます。

■ 機能解除 (CLEAR_FEATURE)と機能設定 (SET_FEATURE)要求はビット番号での機能設定に使用できます。装置 (Device)が受け取り部の場合、利用可能な機能選択はDEVICE_REMOTE_WAKEUP (遠隔起動)とTEST_MODE (試験動作)の2つだけです。これらの詳細についてはUSB2.0仕様書をご覧ください。

■ アドレス設定 (SET_ADDRESS)要求はUSB装置に唯一のアドレスを割り当てるために列挙中に使用されます。このアドレスはwValueで指定され、1～127の最大127個です。本要求は装置において特異で、状態 (Status)段階 (制御 (Control)転送参照)が完了するまで、そのアドレスを設定してはなりません。他の全ての要求は状態 (Status)段階前にその処理を完了しなければなりません。

■ 記述取得 (GET_DESCRIPTOR)と記述設定 (SET_DESCRIPTOR)はwValueで指定した記述内容を返すために使用されます。設定記述に対する要求は1つの要求で装置記述と全てのインターフェースとエンドポイント記述を返します。インターフェース記述とエンドポイント記述はこれらの要求で直接アクセスできません。文字列記述は複数言語支援を許すためにwIndexでの言語識別 (ID)を含みます。

■ 設定取得 (GET_CONFIGURATION)と設定選択 (SET_CONFIGURATION)は現在の装置設定の選択または取得に使用されます。設定取得要求の場合、装置状態を示すバイトがデータ段階中に返されます。0値は装置が未設定で、非0値は装置が設定済みであることを意味します。設定選択要求は装置を許可するために使用されます。許可する設定を選択するためにwValueの下位バイトに希望する設定記述のbConfigrationValue値を含むべきです。

標準 ｲﾝﾀｰﾌｪｰｽ要求 (Standard Interface Request)

これらは現在 5つの標準 ｲﾝﾀｰﾌｪｰｽ要求があり、下表で示されます。

表 13.標準 ｲﾝﾀｰﾌｪｰｽ要求 (Stabdard Interface Request)一覧

bmRequestType	bRequest	wValue	wIndex	wLength	ﾃﾞｰﾀ
1 00 00001	GET_STATUS ($00)	0	ｲﾝﾀｰﾌｪｰｽ番号	2	ｲﾝﾀｰﾌｪｰｽ状態値
0 00 00001	CLEAR_FEATURE ($01)	機能選択値	ｲﾝﾀｰﾌｪｰｽ番号	0	なし
0 00 00001	SET_FEATURE ($03)	機能選択値	ｲﾝﾀｰﾌｪｰｽ番号	0	なし
1 00 00001	GET_INTERFACE ($0A)	0	ｲﾝﾀｰﾌｪｰｽ番号	1	代替 ｲﾝﾀｰﾌｪｰｽ値
0 00 00001	SET_INTERFACE ($11)	代替設定値	ｲﾝﾀｰﾌｪｰｽ番号	0	なし

通常、wIndexは要求対象 ｲﾝﾀｰﾌｪｰｽを指定するために使用されます。この形式は以下で示されます。

図 31.標準 ｲﾝﾀｰﾌｪｰｽ要求時のwIndex値構成

ﾋﾞｯﾄ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
内容	予約								ｲﾝﾀｰﾌｪｰｽ番号							

- 状態取得 (GET_STATUS)要求は ｲﾝﾀｰﾌｪｰｽの状態を返すために使用されます。ｲﾝﾀｰﾌｪｰｽへのこの要求には 2ﾊﾞｲﾄの $00を返すべきです (これらは将来用に予約されています)。
- 機能解除 (CLEAR_FEATURE)と機能設定 (SET_FEATURE)要求はﾋﾞｯﾄ番号での機能設定に使用できます。ｲﾝﾀｰﾌｪｰｽ (Interface)が受け取り部の場合、現在の USB2.0仕様において利用可能な ｲﾝﾀｰﾌｪｰｽ機能はありません。
- ｲﾝﾀｰﾌｪｰｽ取得 (GET_INTERFACE)と ｲﾝﾀｰﾌｪｰｽ設定 (SET_INTERFACE)は ｲﾝﾀｰﾌｪｰｽ記述下でその詳細が記述されている代替 ｲﾝﾀｰﾌｪｰｽ設定の選択を行います。

標準 ｴﾝﾄﾞﾎﾟｲﾝﾄ要求 (Standard Endpoint Request)

これらは現在 4つの標準 ｲﾝﾀｰﾌｪｰｽ要求があり、下表で示されます。

表 14.標準 ｴﾝﾄﾞﾎﾟｲﾝﾄ要求 (Stabdard Endpoint Request)一覧

bmRequestType	bRequest	wValue	wIndex	wLength	ﾃﾞｰﾀ
1 00 00010	GET_STATUS ($00)	0	ｴﾝﾄﾞﾎﾟｲﾝﾄ番号	2	ｴﾝﾄﾞﾎﾟｲﾝﾄ状態値
0 00 00010	CLEAR_FEATURE ($01)	機能選択値	ｴﾝﾄﾞﾎﾟｲﾝﾄ番号	0	なし
0 00 00010	SET_FEATURE ($03)	機能選択値	ｴﾝﾄﾞﾎﾟｲﾝﾄ番号	0	なし
1 00 00010	SYNCH_FRAME ($12)	0	ｴﾝﾄﾞﾎﾟｲﾝﾄ番号	2	ﾌﾚｰﾑ番号

通常、wIndexは要求対象 ｴﾝﾄﾞﾎﾟｲﾝﾄを指定するために使用されます。この形式は以下で示されます。

図 32.標準 ｴﾝﾄﾞﾎﾟｲﾝﾄ要求時のwIndex値構成

ﾋﾞｯﾄ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
内容	予約								方向	予約			ｴﾝﾄﾞﾎﾟｲﾝﾄ番号			

- 状態取得 (GET_STATUS)要求は ｴﾝﾄﾞﾎﾟｲﾝﾄの状態 (停止 /動作不能)を示す 2ﾊﾞｲﾄを返します。この 2ﾊﾞｲﾄの形式は以下で示されます。

図 33.標準 ｲﾝﾀｰﾌｪｰｽ状態取得要求時の戻り値構成

ﾋﾞｯﾄ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
内容	予約															停止

- 機能解除 (CLEAR_FEATURE)と機能設定 (SET_FEATURE)要求はﾋﾞｯﾄ番号での機能設定に使用されます。現在標準で定義されている ｴﾝﾄﾞﾎﾟｲﾝﾄ機能選択は、ｴﾝﾄﾞﾎﾟｲﾝﾄの停止と解除をﾎｽﾄに許す ENDPOINT_HALT ($00)、1つだけです。既定の ｴﾝﾄﾞﾎﾟｲﾝﾄ(ｴﾝﾄﾞﾎﾟｲﾝﾄ0)以外の ｴﾝﾄﾞﾎﾟｲﾝﾄだけがこの機能を持つことを推奨されます。
- 同期 ﾌﾚｰﾑ (SYNCH_FRAME)要求は ｴﾝﾄﾞﾎﾟｲﾝﾄの同期 ﾌﾚｰﾑ番号を報告するために使用されます。

引用文献

- usb_20.pdf　http://www.usb.org　USB協会著
- USB_LANGIDs.pdf　http://www.usb.org　USB協会著
- usb-in-a-nutshell.pdf　http://www.beyondlogic.org　Craig Peacock著

本書はUSBの概要説明のために引用文献を基に和訳再構成したものです。一部意味が想像し易いように本来の用語とは異なる用語で記述されています。本書未記載部分に関してはUSB2.0仕様書を参照してください。

必要と思われる部分には ( )内に英語表記や略称などを残す形で表記しています。

青字の部分は ﾘﾝｸとなっています。一般的に赤字の 0,1は論理 0,1を表します。その他の赤字は重要な部分を表します。

付録 -A :CRC (Cyclic Redundant Check)

CRCはﾋﾞｯﾄ列ﾃﾞｰﾀの誤り検出/訂正に用いられる方法の1つです。誤り検出についてはﾊﾟﾘﾃｨやﾁｪｯｸｻﾑなどの簡単な方法が多く用いられますが、これらは誤り検出能力が低く、訂正に関してはその機能を持ちません。これらに対してCRCは誤り検出能力が高く、基本的に誤り訂正能力があります。けれども誤り訂正能力を高めると、CRC用の冗長ﾋﾞｯﾄも増えますので、一般的に用いられているCRCは誤り検出能力に力点をおいて策定されています。USBで用いられているのはCRC5とCRC16の2種類です。

ここではｿﾌﾄｳｪｱによるCRCの扱いに関する情報を記述します。

多くのCRC関連情報では最初に多項式での説明がなされています。これらで記述される多項式はn進数に対する表現形式の1つで、以下のχは基数(2進数ならば2、10進数ならば10)を示します。例えば$4321は次のように表現されます。

$4321 = 0100 0011 0010 0001
= $0\times\chi^{15}+1\times\chi^{14}+0\times\chi^{13}+0\times\chi^{12}+0\times\chi^{11}+0\times\chi^{10}+1\times\chi^{9}+1\times\chi^{8}+0\times\chi^{7}+0\times\chi^{6}+0\times\chi^{5}+1\times\chi^{4}+0\times\chi^{3}+0\times\chi^{2}+0\times\chi^{1}+1\times\chi^{0}$
= $\chi^{14}+\chi^{9}+\chi^{8}+\chi^{4}+\chi^{0}$

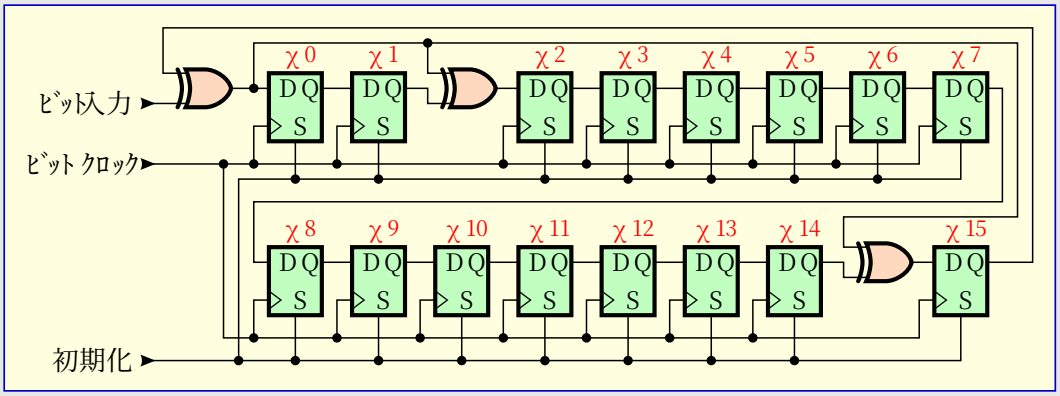
CRC処理は簡単に言えば対象領域を特定の値で割った剰余値を誤り検出/訂正値として送信時に付加し、受信側では同様の演算結果と付加された誤り検出/訂正値の比較によって誤りの検出と訂正を行うことです。USBにおける、この特定の値は次の2種類です。

CRC5 ・・・・ $\chi^{5}+\chi^{2}+\chi^{0}$ = 10 0101 (5ﾋﾞｯﾄ剰余)
CRC16 ・・・ $\chi^{16}+\chi^{15}+\chi^{2}+\chi^{0}$ =1 1000 0000 0000 0101 (16ﾋﾞｯﾄ剰余)

対象領域のﾋﾞｯﾄ列を上記の値で割った剰余が付加されるCRC値です。

USB仕様では上記の基本に対していくつか異なる点があります。またUSBではこれらのCRC計算がﾊｰﾄﾞｳｪｱで行われるのが前提ですので、ﾊｰﾄﾞｳｪｱでの実装に都合よく策定されています。USBにおけるﾊｰﾄﾞｳｪｱによる実際のCRC16計算回路を以下に示します。(以下の記述ではCRC16を基本にしますが、CRC5も同様の手順で処理されます。)

図A-1.ﾊｰﾄﾞｳｪｱCRC16計算回路



- 除算は通常の筆算による方法と同一で、最上位桁側から比較桁数を増やしながら除数より大きな数値として、その数値÷除数を商の或る桁値とし、以降最終桁まで繰り返します。2進数の場合は「商の或る桁値」が0または1しか有り得ませんので、「その数値÷除数」は単純に「その数値-除数」、即ち減算で処理できます。この演算は2を法とするため、桁上がりや桁下がりがありません。このためﾋﾞｯﾄ単位に減算され、他のﾋﾞｯﾄに影響を及ぼさないと言えます。従ってこの「減算」は「排他的論理和(EOR)」で実現されます。
- 上図の各ﾚｼﾞｽﾀの初期値は全1です。これは0が先行し、それ以外が同一値のﾋﾞｯﾄ列において、先行連続0の数に拘らず同一CRC値となるのを防止します。
- 上図で得られた剰余値(χ^{15}～χ^{0})は論理反転され、且つﾋﾞｯﾄ位置が反転(χ^{0}～χ^{15})されて最終的なCRC値になります。このﾋﾞｯﾄ位置反転はEOPの00の影響を排除するためです。受信側で同一の方法によってこのCRC値までを計算すると、上図での剰余値(χ^{15}～χ^{0})は正常に受信された場合、常に次の値になります。

CRC5 ・・・・ $\chi^{3}+\chi^{2}$ = 0 1100 (5ﾋﾞｯﾄ剰余)
CRC16 ・・・ $\chi^{15}+\chi^{3}+\chi^{2}+\chi^{0}$ =1000 0000 0000 1101 (16ﾋﾞｯﾄ剰余)

付録 -B :言語識別子

言語識別子は 16ﾋﾞｯﾄで構成されます。これは ﾋﾞｯﾄ9～ 0の 10ﾋﾞｯﾄで表される主言語識別子とﾋﾞｯﾄ15～ 10の 6ﾋﾞｯﾄで表される補助言語識別子の組み合わせです。

表 B-1.言語識別 (D)ｺｰﾄﾞ一覧

D	言語	D	言語	D	言語	D	言語
$0401	ｱﾗﾋﾞｱ語 (ｻｳｼﾞｱﾗﾋﾞｱ)	$080A	ｽﾍﾟｲﾝ語 (ﾒｷｼｺ)	$081A	ｾﾙﾋﾞｱ語 (ﾗﾃﾝ文字)	$0441	ｽﾜﾋﾘ語 (ｹﾆｱ)
$0801	ｱﾗﾋﾞｱ語 (ｲﾗｸ)	$0C0A	ｽﾍﾟｲﾝ語 (近代)	$0C1A	ｾﾙﾋﾞｱ語 (ｷﾘﾙ文字)	-	-
$0C01	ｱﾗﾋﾞｱ語 (ｴｼﾞﾌﾟﾄ)	$100A	ｽﾍﾟｲﾝ語 (ｸﾞｱﾃﾏﾗ)	$041B	ｽﾛﾊﾞｷｱ語	$0443	ｳｽﾞﾍﾞｸ語 (ﾗﾃﾝ文字)
$1001	ｱﾗﾋﾞｱ語 (ﾘﾋﾞｱ)	$140A	ｽﾍﾟｲﾝ語 (ｺｽﾀﾘｱ)	$041C	ｱﾙﾊﾞﾆｱ語	$0843	ｳｽﾞﾍﾞｸ語 (ｷﾘﾙ文字)
$1401	ｱﾗﾋﾞｱ語 (ｱﾙｼﾞｪﾘｱ)	$180A	ｽﾍﾟｲﾝ語 (ﾊﾟﾅﾏ)	$041D	ｽｳｪｰﾃﾞﾝ語	$0444	ﾀﾀｰﾙ語 (ﾀﾀｰﾙｽﾀﾝ)
$1801	ｱﾗﾋﾞｱ語 (ﾓﾛｯｺ)	$1C0A	ｽﾍﾟｲﾝ語 (ﾄﾞﾐﾆｶ)	$081D	ｽｳｪｰﾃﾞﾝ語 (ﾌｨﾝﾗﾝﾄﾞ)	$0445	ﾍﾞﾝｶﾞﾙ語
$1C01	ｱﾗﾋﾞｱ語 (ﾁｪﾆｼﾞｱ)	$200A	ｽﾍﾟｲﾝ語 (ﾍﾞﾈｽﾞｴﾗ)	$041E	ﾀｲ語	$0446	ﾊﾟﾝｼﾞｬﾌﾞ語
$2001	ｱﾗﾋﾞｱ語 (ｵｰﾏﾝ)	$240A	ｽﾍﾟｲﾝ語 (ｺﾛﾝﾋﾞｱ)	$041F	ﾄﾙｺ語	$0447	ｸﾞｼﾞｬﾗｰﾄ語
$2401	ｱﾗﾋﾞｱ語 (ｲｴﾒﾝ)	$280A	ｽﾍﾟｲﾝ語 (ﾍﾟﾙｰ)	$0420	ｳﾙﾄﾞｩｰ語 (ﾊﾟｷｽﾀﾝ)	$0448	ｵﾘﾔｰ語
$2801	ｱﾗﾋﾞｱ語 (ｼﾘｱ)	$2C0A	ｽﾍﾟｲﾝ語 (ｱﾙｾﾞﾝﾁﾝ)	$0820	ｳﾙﾄﾞｩｰ語 (ｲﾝﾄﾞ)	$0449	ﾀﾐﾙ語
$2C01	ｱﾗﾋﾞｱ語 (ﾖﾙﾀﾞﾝ)	$300A	ｽﾍﾟｲﾝ語 (ｴｸｱﾄﾞﾙ)	$0421	ｲﾝﾄﾞﾈｼｱ語	$044A	ﾃﾙｸ語
$3001	ｱﾗﾋﾞｱ語 (ﾚﾊﾞﾉﾝ)	$340A	ｽﾍﾟｲﾝ語 (ﾁﾘ)	$0422	ｳｸﾗｲﾅ語	$044B	ｶﾝﾅﾀ語
$3401	ｱﾗﾋﾞｱ語 (ｸｳｪｰﾄ)	$380A	ｽﾍﾟｲﾝ語 (ｳﾙｸﾞｱｲ)	$0423	ﾍﾞﾗﾙｰｼ語	$044C	ﾏﾗﾔﾗﾑ語
$3801	ｱﾗﾋﾞｱ語 (U.A.E.)	$3C0A	ｽﾍﾟｲﾝ語 (ﾊﾟﾗｸﾞｱｲ)	$0424	ｽﾛﾍﾞﾆｱ語	$044D	ｱｯｻﾑ語
$3C01	ｱﾗﾋﾞｱ語 (ﾊﾞｰﾚｰﾝ)	$400A	ｽﾍﾟｲﾝ語 (ﾎﾞﾘﾋﾞｱ)	$0425	ｴｽﾄﾆｱ語	$044E	ﾏﾗﾃｨｰ語
$4001	ｱﾗﾋﾞｱ語 (ｶﾀｰﾙ)	$440A	ｽﾍﾟｲﾝ語 (ｴﾙｻﾙﾊﾞﾄﾞﾙ)	$0426	ﾗﾄﾋﾞｱ語	$044F	ｻﾝｽｸﾘｯﾄ語
$0402	ﾌﾞﾙｶﾞﾘｱ語	$480A	ｽﾍﾟｲﾝ語 (ﾎﾝｼﾞｭﾗｽ)	$0427	ﾘﾄｱﾆｱ語	-	-
$0403	ｶﾀﾛﾆｱ語	$4C0A	ｽﾍﾟｲﾝ語 (ﾆｶﾗｸﾞｱ)	$0827	ﾘﾄｱﾆｱ語 (日)	-	-
$0404	中国語 (繁体)	$500A	ｽﾍﾟｲﾝ語 (ﾌﾟｴﾙﾄﾘｺ)	-	-	-	-
$0804	中国語 (簡体)	$040B	ﾌｨﾝﾗﾝﾄ語	$0429	ﾍﾟﾙｼｬ語	-	-
$0C04	中国語 (ﾎﾝｺﾝ)	$040C	仏語 (標準)	$042A	ﾍﾞﾄﾅﾑ語	-	-
$1004	中国語 (ｼﾝｶﾞﾎﾟｰﾙ)	$080C	仏語 (ﾍﾞﾙｷﾞｰ)	$042B	ｱﾙﾒﾆｱ語	$0455	ﾋﾞﾙﾏ語
$1404	中国語 (ﾏｶｵ)	$0C0C	仏語 (ｶﾅﾀﾞ)	$042C	ｱｾﾞﾘ語 (ﾗﾃﾝ文字)	-	-
$0405	ﾁｪｺ語	$100C	仏語 (ｽｲｽ)	$082C	ｱｾﾞﾘ語 (ｷﾘﾙ文字)	$0457	ｺﾝｶﾆ語
$0406	ﾃﾞﾝﾏｰｸ語	$140C	仏語 (ﾙｸｾﾝﾌﾞﾙｸﾞ)	$042D	ﾊﾞｽｸ語	$0458	ﾏﾆﾌﾟﾘ語
$0407	独語 (標準)	$180C	仏語 (ﾓﾅｺ)	-	-	$0459	ｼﾝﾄ語
$0807	独語 (ｽｲｽ)	$040D	ﾍﾌﾞﾗｲ語	$042F	ﾏｹﾄﾞﾆｱ語	-	-
$0C07	独語 (ｵｰｽﾄﾘｱ)	$040E	ﾊﾝｶﾞﾘｰ語	$0430	ｽｰﾄ語	-	-
$1007	独語 (ﾙｸｾﾝﾌﾞﾙｸﾞ)	$040F	ｱｲｽﾗﾝﾄ語	-	-	-	-
$1407	独語 (ﾘﾋﾃｨﾝｼｭﾀｲﾝ)	$0410	伊語 (標準)	-	-	-	-
$0408	ｷﾞﾘｼｬ語	$0810	伊語 (ｽｲｽ)	-	-	-	-
$0409	英語 (米国)	$0411	日本語	-	-	-	-
$0809	英語 (英国)	$0412	韓国語	-	-	$0860	ｶｼﾐｰﾙ語 (ｲﾝﾄﾞ)
$0C09	英語 (豪州)	$0812	韓国語 (ﾊﾝｸﾞﾙ)	$0436	公用 ｵﾗﾝﾀ語 (南 ｱﾌﾘｶ)	$0861	ﾈﾊﾟｰﾙ語 (ｲﾝﾄﾞ)
$1009	英語 (ｶﾅﾀﾞ)	$0413	ｵﾗﾝﾀ語 (ｵﾗﾝﾀﾞ)	$0437	ｸﾞﾙｼﾞｱ語	-	-
$1409	英語 (ﾆｭｰｼﾞｰﾗﾝﾄﾞ)	$0813	ｵﾗﾝﾀ語 (ﾍﾞﾙｷﾞｰ)	$0438	ﾌｪﾛｰ語	-	-
$1809	英語 (ｱｲﾙﾗﾝﾄﾞ)	$0414	ﾉﾙｳｪｰ語 (ﾌﾞｰｸﾓｰﾙ)	$0439	ﾋﾝﾃﾞｨｰ語	-	-
$1C09	英語 (南 ｱﾌﾘｶ)	$0814	ﾉﾙｳｪｰ語 (ﾆｰﾉｼｸ)	-	-	-	-
$2009	英語 (ｼﾞｬﾏｲｶ)	$0415	ﾎﾟｰﾗﾝﾄ語	-	-	-	-
$2409	英語 (ｶﾘﾌﾞ)	$0416	ﾎﾟﾙﾄｶﾞﾙ語 (ﾌﾞﾗｼﾞﾙ)	-	-	-	-
$2809	英語 (ﾍﾞﾘｰｽﾞ)	$0816	ﾎﾟﾙﾄｶﾞﾙ語 (標準)	-	-	$04FF	HID (ﾃﾞｰﾀ記述子使用)
$2C09	英語 (ﾄﾘﾆﾀﾞｰﾄﾞﾄﾊﾞｺﾞ)	-	-	$043E	ﾏﾗﾔ語 (ﾏﾚｰｼｱ)	$F0FF	HID (供給者定義 1)
$3009	英語 (ｼﾞﾝﾊﾞﾌﾞｴ)	$0418	ﾛﾏﾝｼｭ語	$083E	ﾏﾗﾔ語 (ﾌﾞﾙﾈｲ)	$F4FF	HID (供給者定義 2)
$3409	英語 (ﾌｨﾘﾋﾟﾝ)	$0419	露語	$043F	ｶｻﾞﾌ語	$F8FF	HID (供給者定義 3)
$040A	ｽﾍﾟｲﾝ語 (伝統的)	$041A	ｸﾛｱﾁｱ語	-	-	$FEFF	HID (供給者定義 4)