

AVR4950 : ASF – USBホスト階層

要点

- USB2.0適合
 - ・ 第9章
 - ・ 制御(Control)、大量(Bulk)、等時(Isochronous)、割り込み(Interrupt)の転送形式
 - ・ 低速(Low Speed:1.5Mbps)、全速(Full Speed:12Mbps)、高速(High Speed:480Mbps)のデータ速度
- 低いスタック量
- 実時間(OS適合、遅れなし)
- 割り込み駆動
- USB DMAを用いた速度性能
- 低電力動作形態
- USBハブ
- 主要なUSBクラスと使用準備済み(HID, CDC, MSC, PHDC, AUDIO)
- 複合装置
- 8ビットと32ビットのAVR®基盤
- GCCとIAR™のコンパイラ

1. 序説

この資料はUSBホスト階層を紹介しします。この階層はATMEL® AVRソフトウェア枠組み(ASF)に含まれ、USB組み込みホスト応用を構築する最も素早く容易な方法をお客様に提供することが狙いです。また、このUSBホスト階層は低専有量と低電力志向です。この階層の完全な記述はこの資料で利用可能です。

2. 略語

ASF	: AVRソフトウェア枠組み(AVR Software Framework)
CBW	: 命令部被い部(Command Block Wrapper) (大容量記憶クラス)
CDC	: 通信装置クラス(Communication Device Class)
CSW	: 命令状態被い部(Command Statud Wrapper) (大容量記憶クラス)
DPまたはD+	: データ+差動線
DMまたはD-	: データ-差動線
EHCI	: 強化ホスト制御器インターフェース(Enhanced Host Controller Interface)
FS	: USB全速(Full Speed)
HID	: 対人インターフェース装置(Human interface device)
HS	: USB高速(High Speed)
LS	: USB低速(Low Speed)
MSC	: 大容量記憶クラス(Mass Storage Class)
OHCI	: 開放ホスト制御器インターフェース(Open Host Controller Interface)
PHDC	: 周辺健康装置クラス(Peripheral Health Device Class)
PIPE	: ホスト動作でエンドポイント転送を管理するためのデータ緩衝部
sleepmgr	: ASFからの休止管理サービス
UDC	: USB装置制御部(USB Device Controller)
UDD	: USB装置記述子(USB Device Descriptor)
UDI	: USB装置インターフェース(USB Device Interface)
UHC	: USBホスト制御器(USB Host Controller)
UHD	: USBホスト記述子(USB Host Descriptor)
UHI	: USBホスト インターフェース(USB Host Interface)
USB	: 万能直列バス(Universal Serial Bus)
USBB	: 32ビットATMELコア用USBハードウェア インターフェースB版
USBC	: 32ビットATMELコア用USBハードウェア インターフェースC版
STALL	: USB単位転送を停止するのに使用されるUSB通票(トークン)
ZLP	: 0長パケット(Zero length pakect)



8ビット ATMEL
マイクロコントローラ

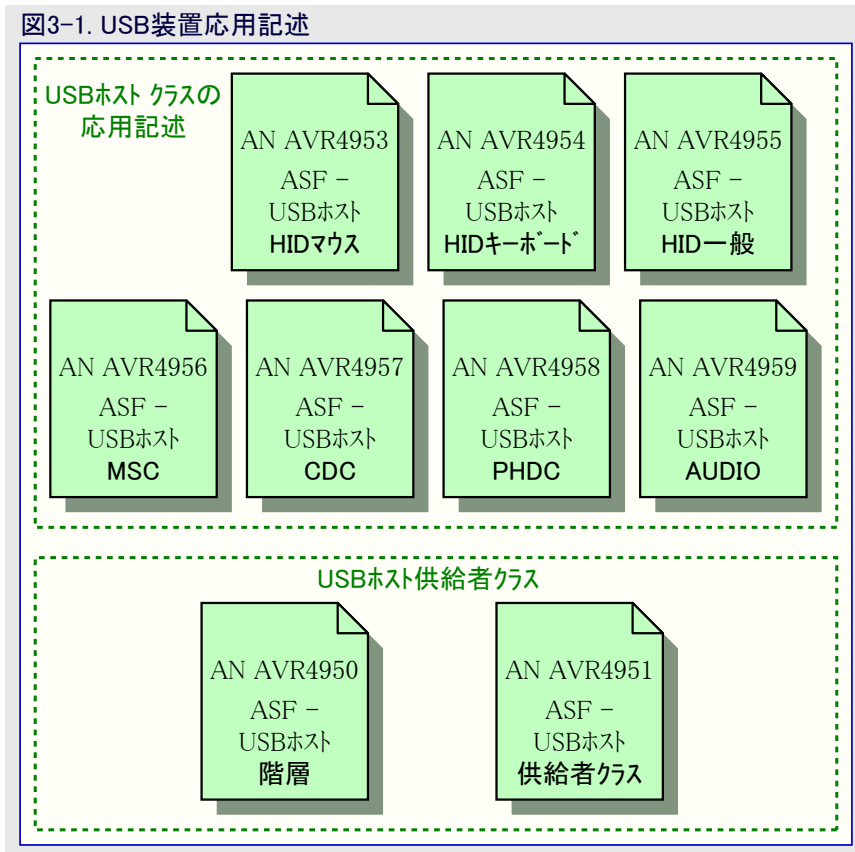
応用記述

本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 8486A-02/12, 8486AJ1-03/14

3. USBホスト応用記述

ATMELによって多数のUSBホスト例が提供されます。各例に関して、ATMELは様々な応用記述を提供します。



USBホスト クラス応用記述(クラス:HID,CDC,MSC,PHDC,AUDIO)を理解するのに基本的なUSBの知識が必要です。ASFで提供されるクラスの1つでUSBホストを作成するには、そのUSBクラスに関連する応用記述を直接参照してください。USBホスト階層とUSBホスト供給者クラスの応用記述は高度なUSB開発者用に設計されています。

4. 構成

4.1. 概要

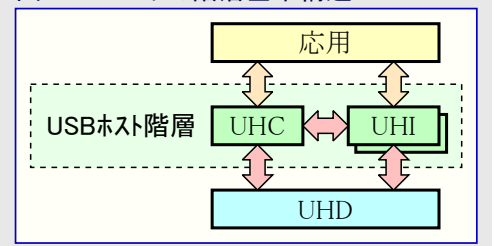
USBホスト階層は以下の3つの部分に分けられます。

- USBホスト制御部(UHC)はUSB第9章適合を提供します。
- USBホスト インターフェイス(UHI)はUSBクラス適合を提供します。
- USBホストドライバ(UHD)は各AVR製品に対するUSBインターフェースを提供します。

注: USBホストドライバは完全な割り込み動作形態で実装され、従ってこのUHDは第三者のUSB階層に対してUSBドライバを作成するための完璧な基礎です。これは独自インターフェースまたはOHCI/EHCIインターフェースのどちらに対しても適用できます。

2重(装置/ホスト)動作形態の場合、USB装置階層を記述するのに「AVR4900:ASF - USB装置階層」応用記述が提供されます。

図4-1. USBホスト階層基本構造



4.2. メモリ使用空間

USBホスト階層のメモリ使用空間は以下に依存します。

- ATMELコア(megaAVR®、UC3)
- USBハードウェア版
- 使用されるUSBクラス
- コンパイラと最適化レベル

これらのパラメータは異なる使用空間値が与えられますが、平均でUSBホスト階層は最高最適化レベルのコンパイラを使用して10Kバイトのフラッシュメモリと1KバイトのSRAMを超えません。

4.3. USBホスト階層ファイル

USBホスト階層ファイルは新しい例プロジェクト(New Example Project)ウィザードとASF単位部選択(Select ASF Modules)ウィザードを通してATMEL AVR Studio® 5で利用可能です。

例のアクセスは例一覧の閲覧または単純に検索領域で“host”を入力することによって行われます。

注: 助言: AVR Studio 5で例一覧を減らすために技術USB(Technology USB)を選択してください。

注: <<From ASF V1>>を含む例の名前は、この応用記述の範囲外であるASF v1.7から前のUSBホスト階層の実装を参照します。

4.3.1. 全製品用共通ファイル

ファイル	パス
<ul style="list-style-type: none"> USB定数定義 usb_protocol.h (usb.orgから) usb_atmel.h (ATMELから) 	common¥services¥usb¥
<ul style="list-style-type: none"> UHCファイル uhc.c/h uhi.h uhd.h 	common¥services¥usb¥uhc¥
<ul style="list-style-type: none"> クラス規約ファイル usb_protocol_<クラス名>.h [*] 	common¥services¥usb¥class¥<クラス名>¥
<ul style="list-style-type: none"> UHIファイル uhi_foo.c/h 	common¥services¥usb¥class¥foo¥host¥

注[*]: これらのファイルはUSB装置階層と共通です。

4.3.2. 選択したAVR製品に依存するUHDファイル

- avr32¥drivers¥usbb¥usbb_host.c/h
- avr32¥drivers¥usbb¥usbb_otg.h [*]
- avr32¥drivers¥usbc¥usbc_host.c/h
- avr32¥drivers¥usbc¥usbc_otg.h [*]

注[*]: これらのファイルはUSB装置階層と共通です。

4.3.3. 各応用に対する特殊ファイル

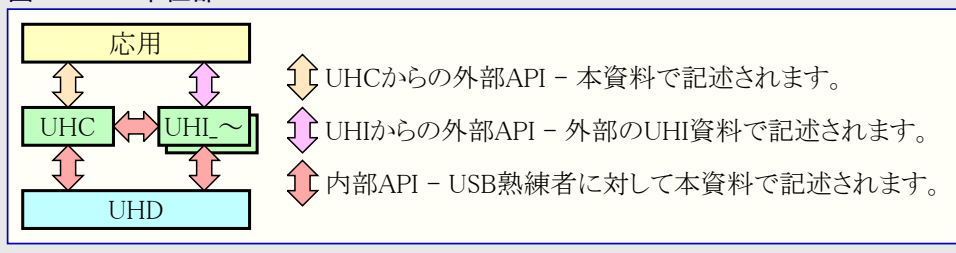
- USBホスト階層を形態設定する応用ファイル(7. 形態設定章と表5-3をご覧ください。)

usb_conf_host.h

5. 応用プログラミング インターフェース

本章は独立した資料で記述されるUHI APIを除く全てのAPIを記述します。

図5-1. USB単位部



5.1. UHCからの外部API

外部UHC APIは一般的なUSBホストの動きの管理と一般的なUSBホスト事象の受け取りを応用に許します。これらの制御と事象はどのUSB応用に対しても共通です。

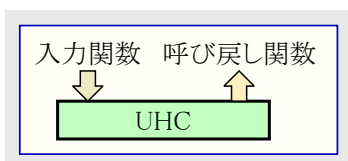


表5-1. UHCからの外部API - 入力関数

宣言	説明
uhc_start()	ホスト動作開始
uhc_stop(bool b_id_stop)	ホスト動作停止、b_id_stopによる要求の場合にUSB IDピンの制御を停止
uhc_suspend(bool b_remotewakeup)	USB線停止。これは各装置の遠隔起動機能を認定するかまたはしません。
uhc_is_suspend()	USB線で停止状態が許可されているか調査
uhc_resume()	USB線再開

表5-2. UHCからの外部API - 接続された装置に関する入力関数

宣言	説明
<code>uint8_t uhc_get_device_number()</code>	接続された装置数を返します。
<code>char*uhc_dev_get_str_manufacturer(uhc_device_t*dev)</code>	USB装置に対応する製造業者ASCII文字列を返します。
<code>char*uhc_dev_get_str_product(uhc_device_t*dev)</code>	USB装置に対応する製品ASCII文字列を返します。
<code>char*uhc_dev_get_str_serial(uhc_device_t*dev)</code>	USB装置に対応する通番ASCII文字列を返します。
<code>char* uhc_dev_get_str(uhc_device_t*dev, uint8_t str_id)</code>	文字列IDに対応するUSB ASCII文字列を返します。
<code>uint16_t uhc_dev_get_power(uhc_device_t*)</code>	装置の最大消費(mA)を得ます。
<code>uhd_speed_t uhc_dev_get_speed(uhc_device_t*)</code>	現在の装置速度を返します。
<code>bool uhc_dev_is_high_speed_support(uhc_device_t*)</code>	装置が高速(High speed)を支援する場合にTRUEを返します。

全てのUHC呼び戻しは任意選択で、各応用に対して`usb_conf_host.h`ファイルで使用者によって定義されます。

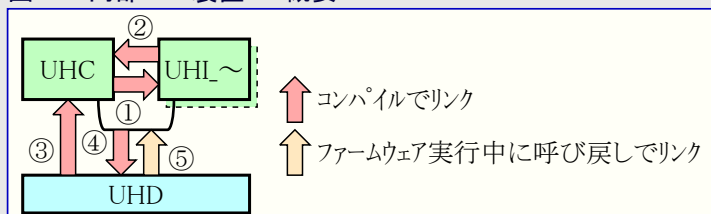
表5-3. UHCからの外部API - 呼び戻し関数

定義名	説明
<code>UHC_MODE_CHANGE(bool b_host_mode)</code>	USB動作形態が自動的に切り替えられることを通知するため。これはIDピンが利用可能な時にだけ可能です。
<code>UHC_VBUS_CHANGE(bool b_present)</code>	VBUSレベルが変更されたことを通知するため(VBUS監視を持つUSBハードウェアでだけ利用可能)。
<code>UHC_VBUS_ERROR()</code>	VBUS異常が起きたことを通知するため(VBUS監視を持つUSBハードウェアでだけ利用可能)。
<code>UHC_CONNECTION_EVENT(uhc_device_t* dev, bool b_present)</code>	装置が接続または切断されたことを通知するため。
<code>UHC_WAKEUP_EVENT()</code>	USB装置またはホストがUSB線を起こす(起動する)時に呼ばれます。
<code>UHC_SOF_EVENT()</code>	1ms毎の各SOF受信に対して呼ばれます。 注: 高速(High speed)と全速(Full speed)の動作形態で利用可能。
<code>UHC_DEVICE_CONF(uhc_device_t* dev)</code>	USB装置形態設定が選ばれなければならない時に呼ばれます。故に、応用はこの装置に対する形態設定番号、またはそれを拒否するために形態設定番号0のどちらかを選ぶことができます。呼び戻しが定義されていない場合、形態設定1が選ばれます。
<code>UHC_ENUM_EVENT(uhc_device_t* dev, uhc_enum_status_t status)</code>	USB装置列挙(接続認証)が完了される時に呼ばれます。状態は以下で有り得ます。 <ul style="list-style-type: none"> • <code>UHC_ENUM_SUCCESS</code> 装置は最低1つの支援されるインターフェースでバス列挙(接続認証)されます。 • <code>UHC_ENUM_UNSUPPORTED</code> 全てのインターフェースがUHIによって支援されません。 • <code>UHC_ENUM_OVERCURRENT</code> 装置電力を支援することができません。 • <code>UHC_ENUM_FAIL</code> USB列挙(接続認証)中に問題が起きました。 • <code>UHC_ENUM_HARDWARE_LIMIT</code> USBハードウェアはそれを支援できません。十分な空きパイプ無し。 • <code>UHC_ENUM_SOFTWARE_LIMIT</code> USBソフトウェアはそれを支援できません。実装制限。 • <code>UHC_ENUM_MEMORY_LIMIT</code> USBソフトウェアはそれを支援できません。十分なメモリ無し。

5.2. 内部API

以下の定義はASFで提供されない特別なUSBホストの開発を意図する高度なUSB使用者用に定義されます。

図5-2. 内部USB装置API概要



注: 番号は以下の表に対する参照です。

表5-4. UHCからのUHI入力 - ①

宣言	説明
<code>uhc_enum_status_t (*install)(uhc_device_t*)</code>	新しい装置によって支援される場合にインターフェースを取り付け(エンドポイント割り当て)
<code>void (*uninstall)(uhc_device_t*)</code>	この装置に取り付けられていればインターフェースを取り外し
<code>void (*enable)(uhc_device_t*)</code>	装置のインターフェースを開始するために呼ばれます。
<code>void (*sof_notify)(bool b_micro)</code>	許可したUSBインターフェースでのSOF事象を通知するのにUHCによって呼ばれます。

注: UHI APIはUSB_HOST_UHI配列に格納されます。表7-1をご覧ください。

表5-5. UHDからのUHC入力 - ③

宣言	説明
<code>uhc_start()</code>	ホスト動作を開始します。
<code>uhc_stop(bool b_id_stop)</code>	ホスト動作と、 <code>b_id_stop</code> によって要求された場合にUSB IDピン管理を停止します。
<code>uhc_notify_connection(bool b_plug)</code>	装置が接続または切断された時に呼ばれます。
<code>uhc_notify_sof(bool b_micro)</code>	SOFが送られる時に呼ばれます。
<code>uhc_notify_resume()</code>	再開バス状態発生時に呼ばれます。下または上方向流れ再開後

表5-6. UHD入力 - ④

宣言	呼出者	説明
<code>uhd_enable()</code>	UHC	USBホスト動作と、IDピンが利用可能ならばID管理を許可
<code>uhd_disable(bool b_id_stop)</code>	UHC	USBホスト動作と、 <code>b_id_stop</code> で要求された場合にUSB IDピン管理を禁止
<code>uhd_speed_t uhd_get_speed()</code>	UHC/UHI	(USBハブに接続された装置ではなく)USBインターフェースに接続された装置の速度を調べるため
<code>uint16_t uhd_get_frame_number()</code>	応用	現在のフレーム開始(SOF)番号を返します。
<code>uint16_t uhd_get_microframe_number()</code>	応用	現在のマイクロ フレーム開始番号を返します。
<code>uhd_send_reset(uhd_callback_reset_t callback)</code>	UHC	USBドライバはUSB線でリセット状態を許可
<code>uhd_suspend()</code>	UHC	USBドライバはUSB線で一時停止状態を許可
<code>uhd_is_suspend()</code>	UHC	USB線で一時停止状態が許可されているか調査
<code>uhd_resume()</code>	UHC	USBドライバはUSB線でアイドル状態を許可(“下方向流れ再開”)
<code>bool uhd_setup_request(usb_add_t add, usb_setup_req_t req, uint8_t *payload, uint16_t payload_size, uhd_callback_setup_run_t callback_run, uhd_callback_setup_end_t callback_end)</code>	UHC/UHI	制御エンドポイント構成設定待ち行列にこの構成設定(SETUP)要求を追加 (要求制限時間は5秒)
<code>bool uhd_ep0_alloc(usb_add_t add, uint8_t ep_size)</code>	UHC	制御エンドポイント0を許可
<code>bool uhd_ep_alloc(usb_add_t add, usb_ep_desc_t *ep_desc)</code>	UHC	(制御エンドポイントを除く)エンドポイントを許可
<code>uhd_ep_free(usb_add_t add, usb_ep_t endp)</code>	UHC	装置の1つのエンドポイントまたは全てのエンドポイントを禁止
<code>bool uhd_ep_run(usb_add_t add, usb_ep_t endp, bool b_shortpacket, uint8_t *buf, iram_size_t buf_size, uint16_t timeout, uhd_callback_trans_t callback)</code>	UHI	エンドポイントでINまたはOUTのデータ転送を開始/停止。転送の計画はUHDによって管理されます。 注: 制御エンドポイントはここで認定されません。
<code>uhd_ep_abort(usb_add_t add, usb_ep_t endp)</code>		

表5-7. UHD呼び戻し - ⑤

宣言	説明
<code>typedef void (*uhd_callback_reset_t)(void)</code>	<code>uhd_send_reset()</code> 経路で要求されたリセット事象が完了される時に呼ばれます。
<code>typedef bool (*uhd_callback_setup_run_t)(usb_add_t add, uint8_t **payload, uint16_t *payload_size)</code>	<code>uhd_setup_request()</code> 経路で割り当てられた(データ本体)緩衝部が満杯または空の時に呼ばれます。その後、構成設定(SETUP)要求は持続します。SETUP要求を再開するために新しい緩衝部を提供することができ、さもなければSETUP要求は中止されるでしょう。
<code>typedef void (*uhd_callback_setup_end_t)(usb_add_t add, uhd_trans_status_t status, uint16_t payload_trans)</code>	構成設定(SETUP)要求が完了される時に呼ばれます。この呼び戻しは <code>uhd_setup_request()</code> ルーチン経路で与えられます。
<code>typedef void (*uhd_callback_trans_t)(usb_add_t add, uhd_trans_status_t status, iram_size_t nb_transferred)</code>	転送要求が完了または取り消される時に呼ばれます。この要求は <code>uhd_ep_run()</code> 経路で登録されます。
<code>uhd_trans_status_t</code>	この構造体は成功か否か何れにしても毎回の試みた単位転送後に更新されます。単位転送が成功の場合、状態領域は以下に設定されます。 <ul style="list-style-type: none"> • <code>UHD_TRANS_NOERROR</code> さもなければ、それは以下の転送異常形式に従って設定されます。 <ul style="list-style-type: none"> • <code>UHD_TRANS_CRC</code> データ パケットでのCRC異常 • <code>UHD_TRANS_DISCONNECT</code> 装置が切断されています。 • <code>UHD_TRANS_MISMATCH</code> データ交互PIDが予測される値と一致しません。 • <code>UHD_TRANS_STALL</code> エンドポイントが機能停止(STALL)PIDを返しました。 • <code>UHD_TRANS_NOT_RESPONDING</code> 装置が通票(トークン)に回答しない(IN)、またはハンドシェイクを提供しない(OUT)。 • <code>UHD_TRANS_PID_FAILURE</code> エンドポイントからのPIDの検査ビットが誤り。 • <code>UHD_TRANS_TIMEOUT</code> 制限時間前にデータ送信が完了しません。 • <code>UHD_TRANS_ABORTED</code> データ送信が中止されました。

表5-8. 高速(High Speed)応用専用UHD入力 - ④

宣言	呼出者	説明
<code>uhd_test_mode_j()</code>	UHC	USB HSポート上の特定信号を許可するための機能。これらはUSBホストHS検定を動かすために要求されます。
<code>uhd_test_mode_k()</code>		
<code>uhd_test_mode_se0_nak()</code>		
<code>uhd_test_mode_packet()</code>		

5.3. 装置接続情報

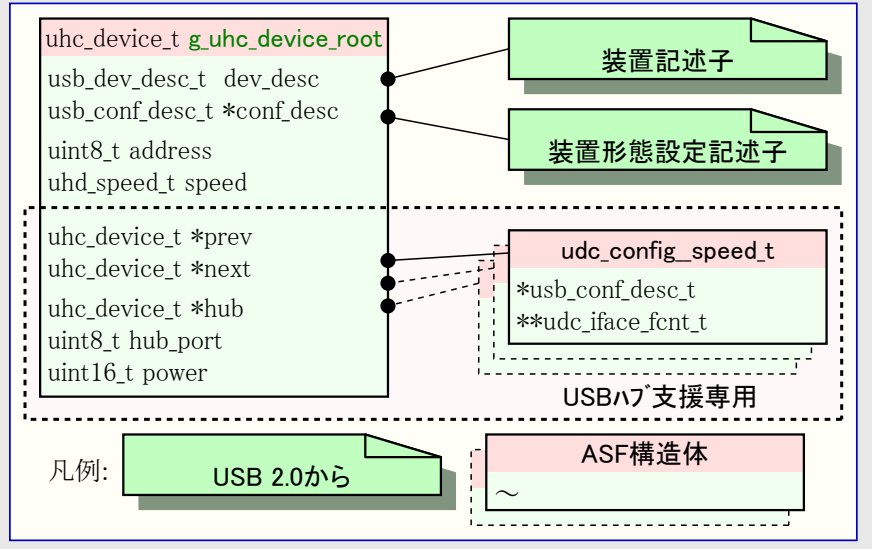
UHCはUSB樹状系図で接続された装置について多くの情報を格納する、`uhc_device_t`と呼ばれる構造体経由で装置を管理します。`g_uhc_device_root`全域変数は最初に接続された装置についての情報を含み、それは装置一覧の最初の要素です。ユーザー応用は現在のUSB樹状系図の情報を得るのにこの一覧を読むことができます。

注: 装置位置とUSB記述子の領域割り当てが動的で標準割り当てライブラリ(例えば、`alloc()`)が必要なので、USB樹状系図制限はありません。

`uhc_device_t`構造体は以下を含みます。

- USB装置記述子のポインタ
- USB形態設定記述子のポインタ
- USB装置アドレス
- USB装置速度
- 親USBハブ装置のポインタ(USBハブ支援専用)
- ハブポート番号接続(USBハブ支援専用)

図5-3. `uhc_device_t`定義



6. USBホスト階層の動き

本章は高度なUSBの使い方の疑問に答えるのとUSBホスト階層の動きを紹介することが狙いです。

以下の項はそれらの段階に関する各種階層間の相互作用を記述します。

- USB 2重役割管理
- USB VBUS管理
- USB装置接続管理
- USB装置列挙(接続認証)管理

6.1. 2重役割

USBハードウェアがUSB装置とUSBホストで動く可能性を持つ時に、使用者は2重役割応用を実装することができます。この応用は(OTG仕様からの)USB IDピンレベルまたは独自処理に従ってUSB装置とUSBホストの作業間を切り替えます。

USBホスト階層は2重役割を管理するのに、**手動**、**自動**、**半自動**の3つの可能性を提供します。

手動2重役割はIDピンが利用不能の時に使用されなければなりません。この場合、独自処理が2重役割を管理します。

自動と**半自動**の2重役割はIDピンがUSBハードウェア インターフェースによって管理される時に使用されなければなりません。

半自動2重役割はIDピン変化時にUSB装置とUSBホストの動作開始の遅延を許します。この可能性は`UHD_START_MODE_MANUAL`が定義された時に許可されます(表7-2をご覧ください)。

図6-1. USB手動2重役割

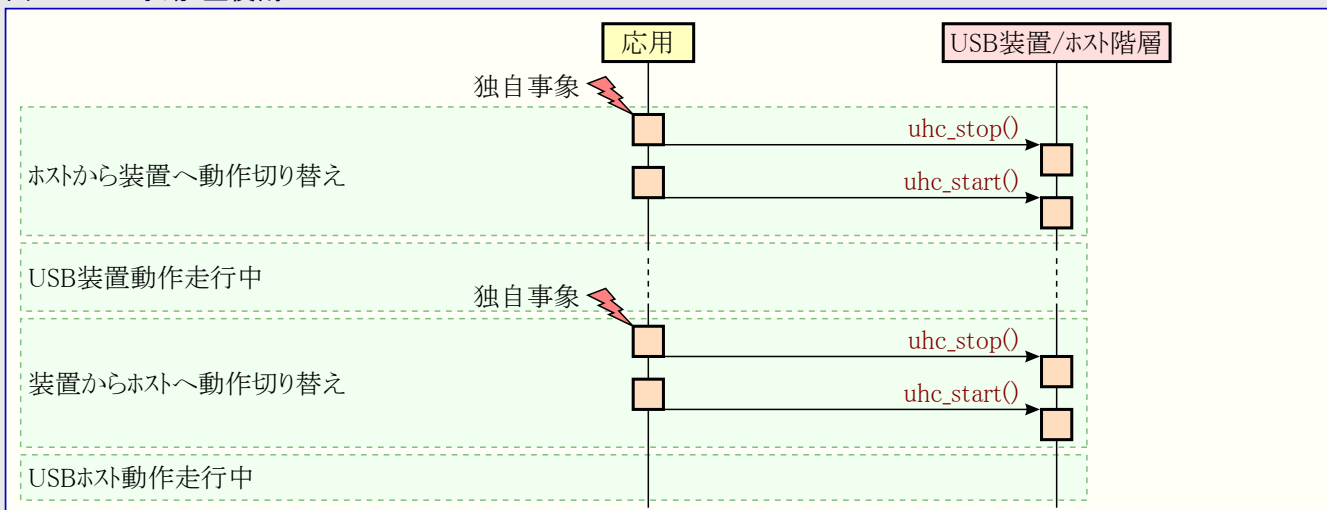


図6-2. USB半自動2重役割

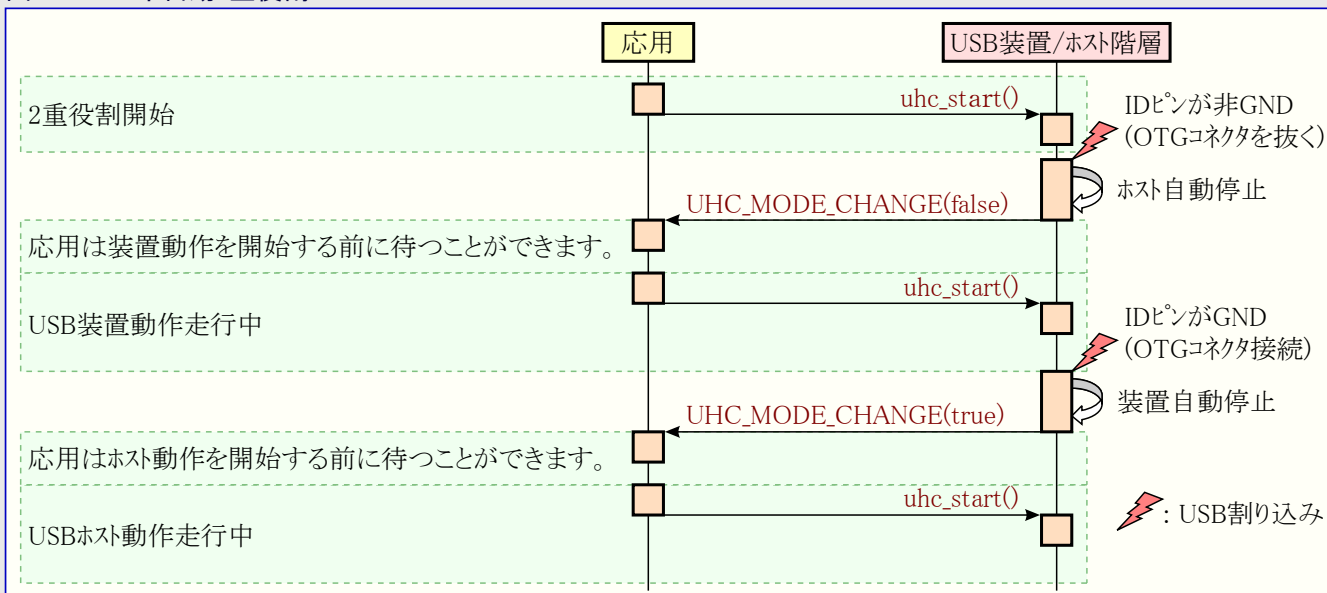
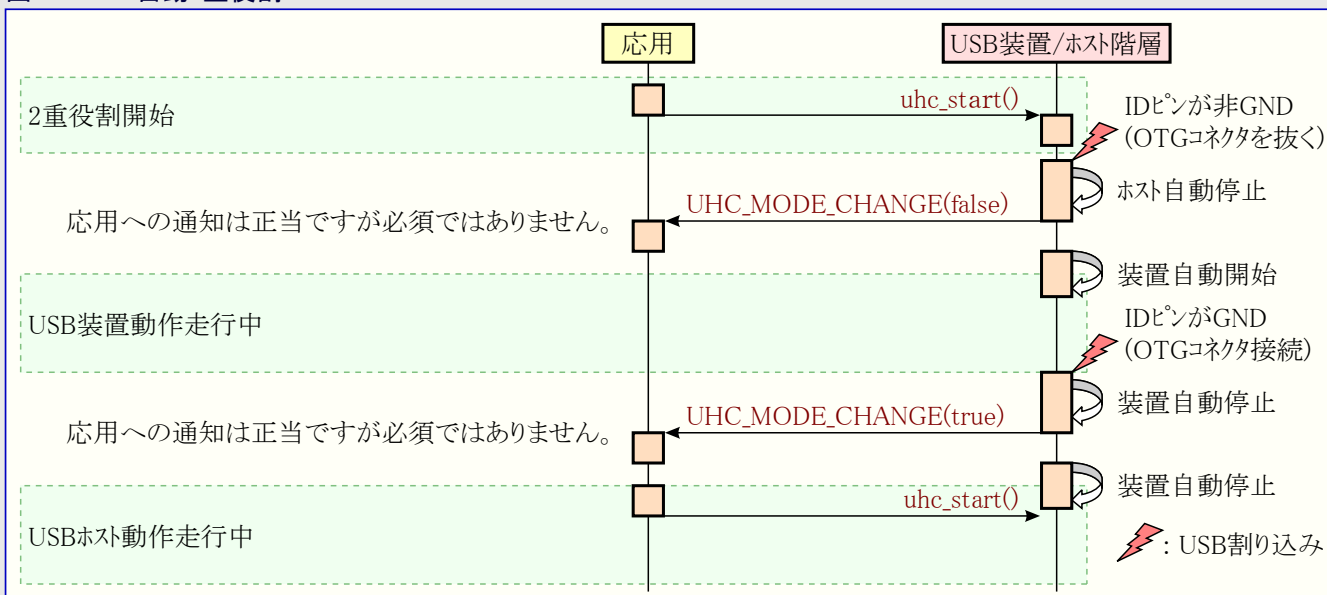
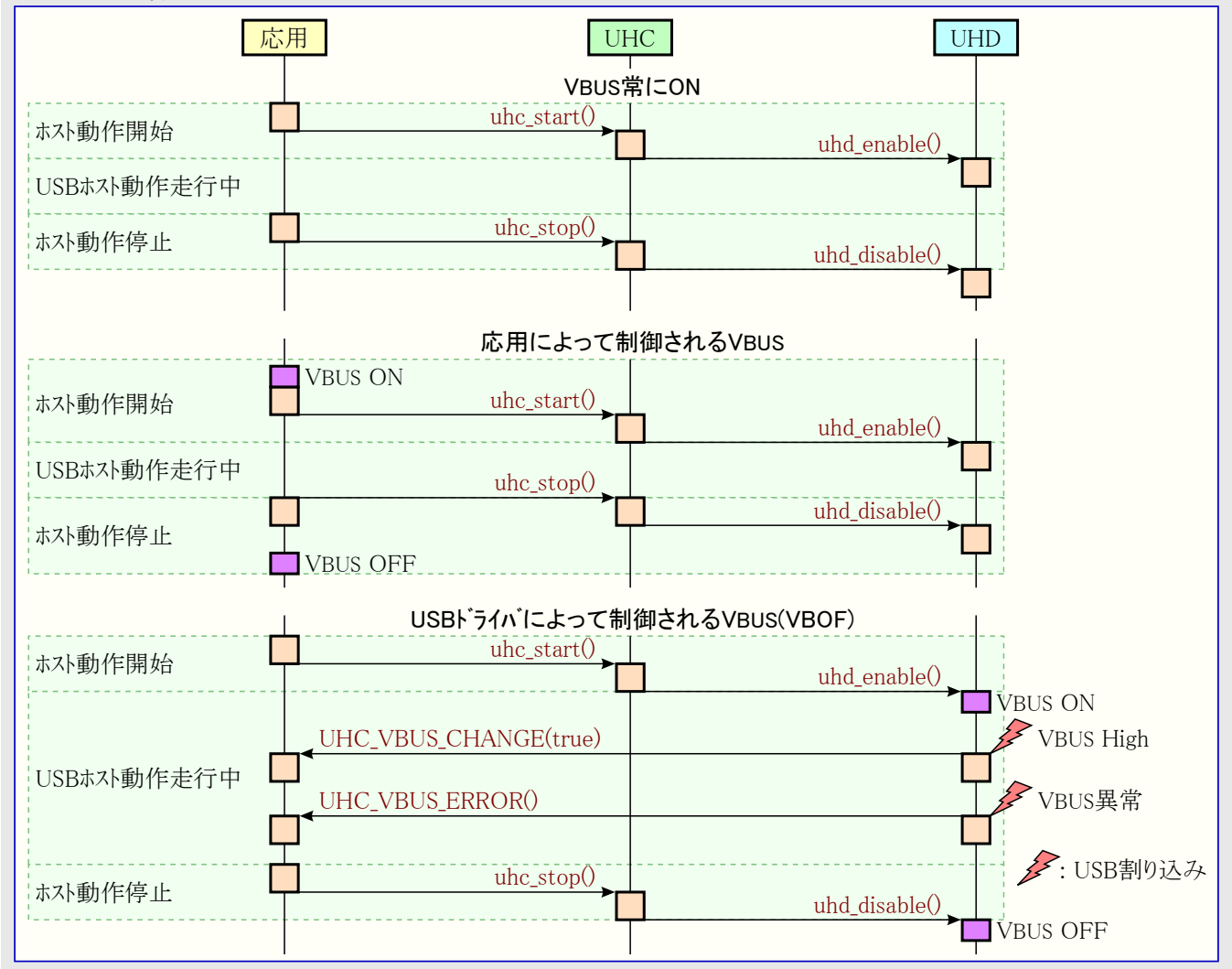


図6-3. USB自動2重役割



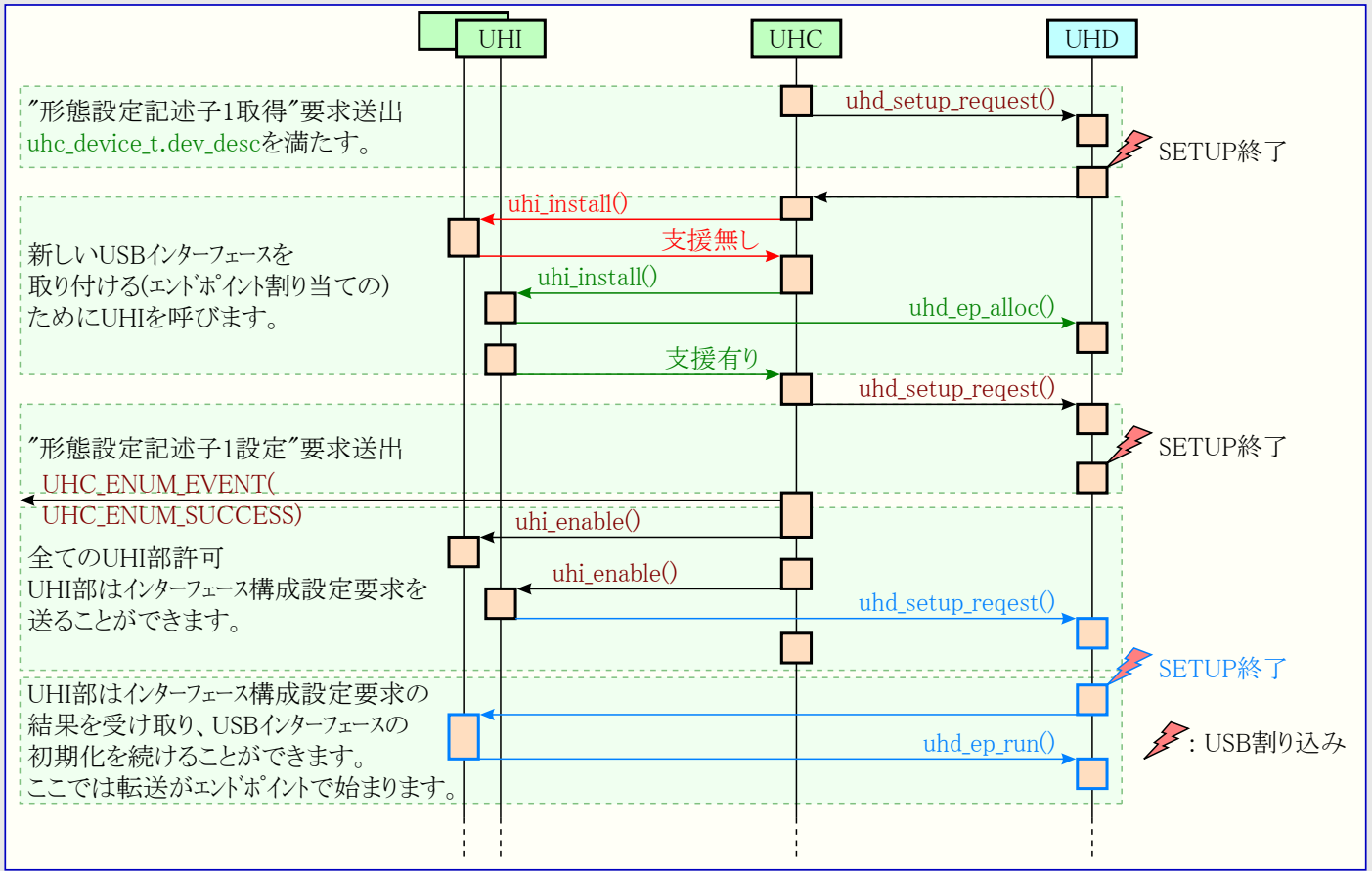
6.2. VBUS管理

図6-4. VBUS管理



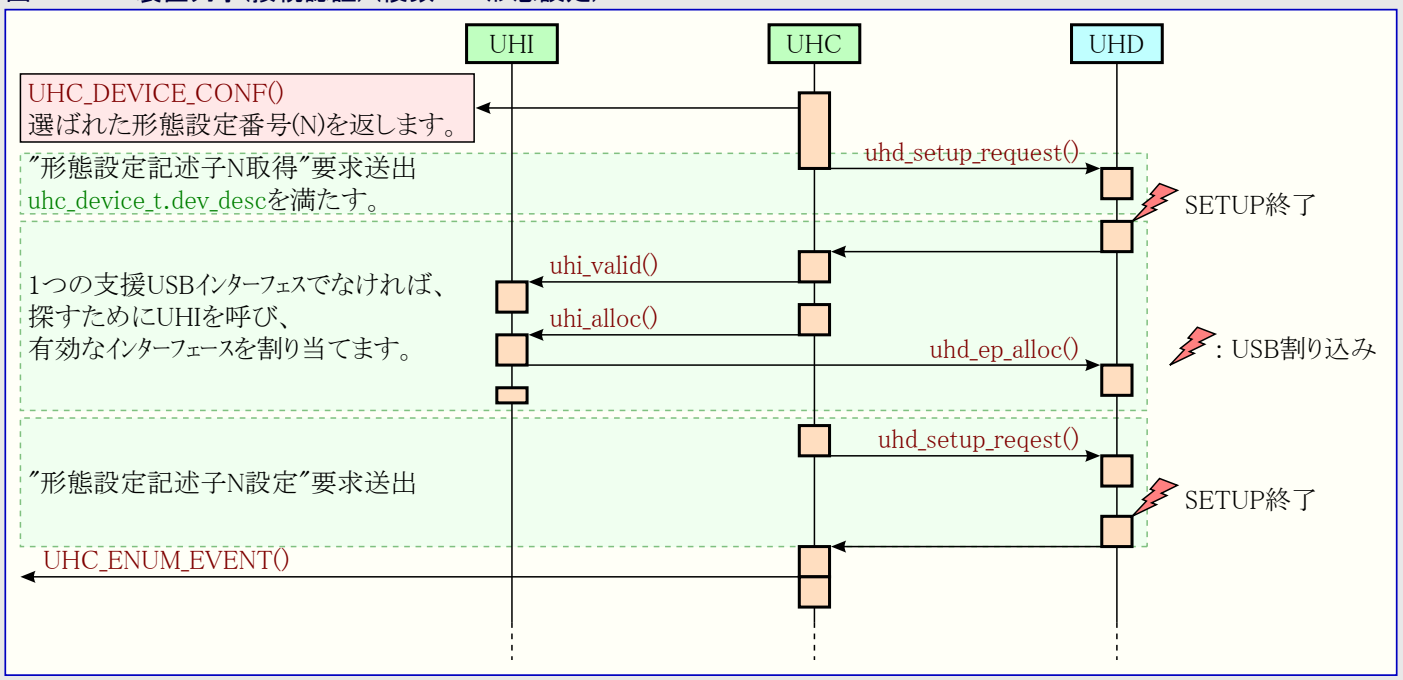
6.4. 装置列挙(接続認証)

図6-6. USB装置列挙(接続認証) (単一USB形態設定)



注: ここで、`UHC_DEVICE_CONF()`呼び戻し関数は定義されず、従って規定によって装置形態設定1が選ばれます。

図6-7. USB装置列挙(接続認証) (複数USB形態設定)



複数形態設定を持つ既知の装置を支援するための使用者コード例は次の通りです。

```
<conf_usb.h>
#define UHC_DEVICE_CONF(dev) user_choose_usb_conf(dev)

<使用者Cファイル>:
uint8_t user_choose_usb_conf(uhc_device_t* dev) {
    if ((dev->dev_desc.idVendor == 0x1234)
        && (dev->dev_desc.idProduct == 0x5678)) {
        Assert(dev->dev_desc.bNumConfigurations>=2);
        return 2; // 形態設定2選択
    }
    return 0; // 未知の装置なので選択形態設定なし
```

7. 形態設定

応用の形態設定は`conf_usb_host.h`ファイルで定義されます。このファイルは各応用に対して作成されなければならない、この実行には基本的なUSBの知識が必要です。

`conf_usb_host.h`ファイルは以下の形態設定を定義しなければなりません。

- USBホスト形態設定
- USBホスト インターフェース形態設定
- USBホストドライバ形態設定

7.1. USBホスト形態設定

以下の形態設定は主USB装置形態設定である、応用の`conf_usb.h`ファイルに含められなければなりません。

表7-1. USBホスト形態設定

定義名	型	説明
USB_HOST_UHI	UHI APIの配列	USBホストによって支援されるUHIの一覧を定義 例: #define USB_HOST_UHI UHI_MSC, UHI_HID_MOUSE
USB_HOST_POWER_MAX	mA	VBUSで許される最大電流
USB_HOST_HUB_SUPPORT (注)	(定義のみ)	USBハブ支援認定 (ASF3公開に対して利用可能)
USB_HOST_HS_SUPPRT (注)	(定義のみ)	高速(High speed)での走行をUSBホストに認定

注: 任意選択形態設定。これを禁止するには定義命令文を注釈にしてください(例: // #define USB_HOST_X)。

7.2. USBホスト インターフェース形態設定

UHI形態設定はUSBホスト クラス応用記述で記述されます。

7.3. USBホストドライバ形態設定

USBハードウェア インターフェースは`conf_usb_host.h`ファイル、`conf_board.h`ファイル、プロジェクト任意選択で許可することができる特別な機能を提供することができます。

表7-2. `conf_usb_host.h`でのUSBホストドライバ形態設定

定義名	値	UHD	説明
UHD_NO_SLEEP_MGR	(定義のみ)	全て	休止管理サービスの管理を取り去ります。
UHD_ISOCHRONOUS_NB_BANK	1,2,3	USBB	等時エンドポイント緩衝を減少または増加 (未定義時の既定値:2)
UHD_BULK_NB_BANK	1,2,3	USBB	大量(バルク)エンドポイント緩衝を減少または増加 (未定義時の既定値:2)
UHD_INTERRUPT_NB_BANK	1,2,3	USBB	割り込みエンドポイント緩衝を減少または増加 (未定義時の既定値:1)
UHD_INT_LEVEL	0~3	USBB USBC	AVR32コアのUSB割り込みレベルを設定 (未定義時の既定値:0(推奨))
UHD_START_MODE_MANUAL	(定義のみ)	USBB USBC	既定により、IDピンが利用可能な(USB_ID定義)時に、ホストまたは装置の動作はIDピン変化時に自動的に停止されて開始されます。この自動開始は禁止することができますが、自動停止動作は存続します。

表7-3. conf_board.hでのUSBホストドライバ形態設定

定義名	値	UHD	説明
USB_ID	AVR32_USBB_USB_ID_x_x	AVR32 - USBB	USBコネクタからIDピンに接続される入力ピンを定義します。これはOTGコネクタからUSB IDピンの使用を許可します。未定義なら、IDピンの使用は禁止されます。
	AVR32_USBB_USB_または AVR32_USBC_USB_x	AVR32 - USBC	
USB_VBOF	AVR32_USBB_USB_VBOF_x_x	AVR32 - USBB	VBUS生成器の許可ピンに接続される出力ピンを定義します。これはUSBハードウェア インターフェースによるVBUS制御の使用を許可します。未定義ならば、VBUS制御は禁止されます。
	AVR32_USBB_USB_VBOFまたは AVR32_USBB_USB_VBOF_x	AVR32 - USBC	
USB_VBOF_ACTIVE_LEVEL	LOWまたはHIGH	AVR32 - USBB AVR32 - USBC	USB_VBOF出力ピンの活性レベル

表7-4. プロジェクトでのUSBホスト装置ドライバ形態設定

定義名	値	UHD	説明
UHD_ENABLE	(定義のみ)	USBB	これらの定義は2重役割(ホストと装置)が許可されたかを知ることを許します。
UDD_ENABLE	(定義のみ)	USBC	

8. USBハブ支援

USBホスト階層はUSBハブを通してUSB樹状系図を支援することができます。けれども、ATMELのデバイスで提供されるUSBハードウェア インターフェースはUSB樹状系図を制限し得ます。

表8-1. USBハブ ハードウェア制限

USBハードウェア インターフェース	説明
USBB	パイプ数が7または8に制限されますが、制御エンドポイントは1つのパイプで多重化されます。USB樹状系図上で同時に複数のUSB速度を支援しません。
USBC	パイプ数が7に制限されますが、制御エンドポイントは1つのパイプで多重化されます。パイプ多重化を支援します。(注) USB樹状系図上で同時に複数のUSB速度を支援しません。

注: パイプ数のための制限を避けるためにパイプの多重化が可能です。けれども、転送計画はUSBハードウェア代わってソフトウェア(UHDドライバ)によって行わなければなりません。

9. 電力消費

ATMEL製品で利用可能な電力動作形態はUSBハードウェアによって支援されます。全てのUSBドライバは休止管理からの機能を実装します。従って、USB階層へのどの呼び出しにも先立って、応用によって休止管理サービス初期化(sleepmgr_init())が呼ばれなければなりません。

9.1. AVR32コア

AVR32コアではCPUで休止命令を使用することによってアイドル区間の間、クロックと発振器を自動的にOFFへ切り替えることができます。

表9-1. USBBとUSBCのホストドライバによって支援される休止レベル

USB状態	認定される最大休止動作形態	
IDピン不活性(ホスト動作形態待ち)	静止(STATIC)	
VBUS無し(VBUS事象待ち)	停止(STOP)またはUTMI部での凍結(FROZEN) (UTMI=高速(High Speed)能力)	
VBUS存在且つ装置切断	アイドル(IDLE)	
VBUS存在 且つ 装置接続	一時停止(Suspend)	静止(STATIC)
	DMAなしアイドル(IDLE)	アイドル(IDLE)
	DMA走行中+アイドル(IDLE)	アイドル(IDLE)

注: USB階層以外の他のソフトウェア単位部がより深くない休止動作形態を必要とする場合に、ここで一覧にされた休止動作形態に達しないかもしれません。

10. 目次

AVR4950 : ASF – USBホスト階層	1
要点	1
1. 序説	1
2. 略語	1
3. USBホスト応用記述	2
4. 構成	2
4.1. 概要	2
4.2. メモリ使用空間	2
4.3. USBホスト階層ファイル	3
4.3.1. 全製品用共通ファイル	3
4.3.2. 選択したAVR製品に依存するUHDファイル	3
4.3.3. 各応用に対する特殊ファイル	3
5. 応用プログラミング インターフェース	3
5.1. UHCからの外部API	3
5.2. 内部API	5
5.3. 装置接続情報	8
6. USBホスト階層の動き	8
6.1. 2重役割	8
6.2. VBUS管理	10
6.3. 装置接続	11
6.4. 装置列挙(接続認証)	12
7. 形態設定	13
7.1. USBホスト形態設定	13
7.2. USBホスト インターフェース形態設定	13
7.3. USBホストドライバ形態設定	13
8. USBハブ支援	14
9. 電力消費	14
9.1. AVR32コア	14
10. 目次	15



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA
TEL (+1)(408) 441-0311
FAX (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan

141-0032 東京都品川区
大崎1-6-4
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2012 Atmel Corporation. 全権利予約済

ATMEL®、ATMELロゴとそれらの組み合わせ、AVR®、AVR Studio®、megaAVR®、それとその他はATMEL Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はATMEL製品と関連して提供されています。本資料またはATMEL製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。ATMELのウェブサイトに表示する販売の条件とATMELの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、ATMELはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえATMELがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してATMELに責任がないでしょう。ATMELは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。ATMELはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、ATMEL製品は車載応用に対して適当ではなく、使用されるべきではありません。ATMEL製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© HERO 2014.

本応用記述はATMELのAVR4950応用記述(doc8486.pdf Rev.8486A-02/12)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

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。