

AVR4900 : ASF – USB装置階層

要点

- USB 2.0適合
 - ・ USB第9章保証
 - ・ 制御(Control)、大量(Bulk)、等時(Isochronous)、割り込み(Interrupt)の転送形式
 - ・ 低速(Low,1.5Mビット/秒)、全速(Full,12Mビット/秒)、高速(High,480Mビット/秒)のデータ速度
- 主応用のための空間を空ける小さな階層
- 実時間(OS適合、遅延なし)
- 8ビットと32ビットのAVR[®]基盤を支援
- 速度性能を増すUSB DMA支援
- 殆どのUSBクラスを支援、使用準備済み(HID,CDC,MSC,PHDC,AUDIO)

1. 序説

この資料はUSB装置階層を紹介します。この階層はATMEL[®] AVRソフトウェア枠組み(ASF:AVR Software Framework)に含まれ、最高に素早く最も簡単なUSB応用構築方法をお客様に提供するのが狙いです。この階層の完全な記述がこの資料で利用可能です。基本的なUSBの知識を除いて特別な知識は全く必要とされません。

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
- ・ 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)
- ・ 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)
- ・ USB : 万能直列バス(Universal Serial Bus)
- ・ MSC : 大容量記憶クラス(Mass Storage Class)
- ・ PHDC : 周辺健康装置クラス(Peripheral Health Device Class)
- ・ sleepmgr : ASFからの休止管理サービス
- ・ ZLP : 0長パケット(Zero length pakect)



ATMEL

マイクロコントローラ

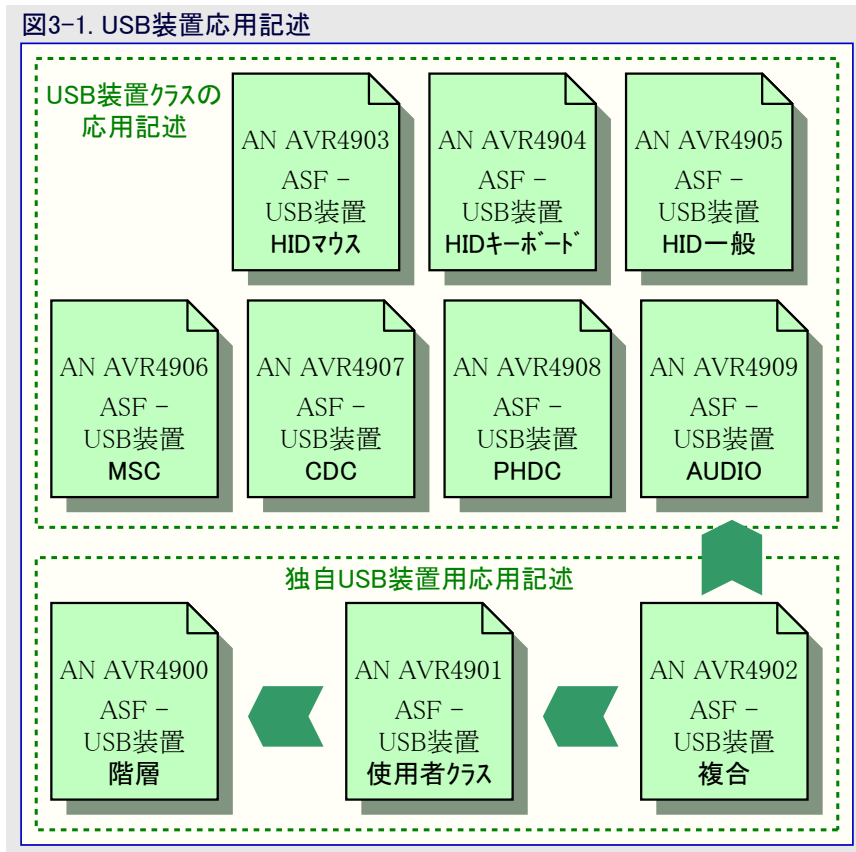
応用記述

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. 8360C-09/11, 8360CJ2-03/14

3. USB装置応用記述

ATMELによって多数のUSB装置例が提供されます。各例に関して、ATMELは応用記述を提供します。



USB装置クラス応用記述(クラス:HID,CDC,MSC,PHDC)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を作成するには、そのUSBクラスに対して関連する応用記述を直接参照してください。新しいクラスと複合USB装置の応用記述は高度なUSB開発者用に設計されています。

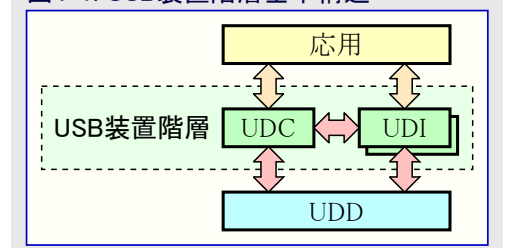
4. 構成

4.1. 概要

USB装置階層は以下の3つの部分に分けられます。

- USB装置制御部(UDC)はUSB第9章適合を提供します。
- USB装置インターフェース(UDI)はUSBクラス適合を提供します。
- USB装置ドライバ(UDD)は各AVR製品に対するUSBインターフェースを提供します。

図4-1. USB装置階層基本構造



4.2. メモリ使用空間

USB装置階層のメモリ使用空間は以下に依存します。

- AVRコア(XMEGA[®]、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多くの値が与えられ、平均的なUSB装置階層では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装置階層ファイル

USB装置階層ファイルはATMEL® AVR Studio® 5で利用可能です。

ATMEL® AVR Studio® 5は新しい例プロジェクトの作成(New Example Project)を許します。故に、例でUSB装置(USB Device)によって始まるUSB装置例を一覧にしてください(例一覧を減らすために技術USB(Technology USB)を選択してください)。注意、(ASF V1からの)文での例はこのUSB装置階層に対応しません。

4.3.1. 全AVR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"> UDCファイル
udc.c/h
udc_desc.h
udi.h
udd.h | common¥services¥usb¥udc¥ |
| <ul style="list-style-type: none"> クラス規約ファイル
usb_protocol_foo.h | common¥services¥usb¥class¥foo¥ |
| <ul style="list-style-type: none"> UDIファイル
udi_foo.c/h
udi_foo_desc.c
udi_foo_conf.h | common¥services¥usb¥class¥foo¥device¥ |

4.3.2. 選択したAVR製品に依存するUDDファイル

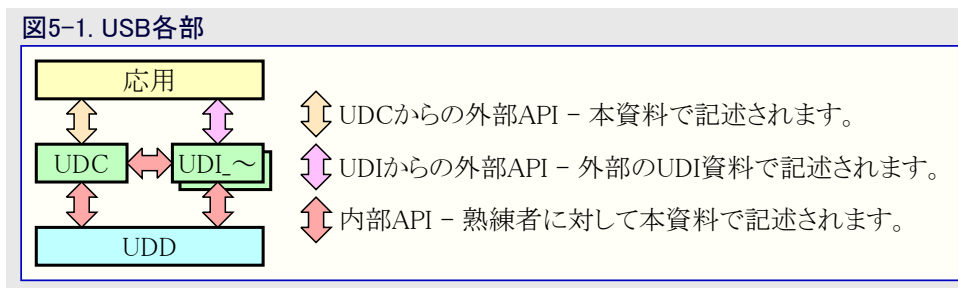
- avr32¥drivers¥usbb¥usbb_device.c/h
- avr32¥drivers¥usbb¥usbb_otg.h
- avr32¥drivers¥usbc¥usbc_device.c/h
- avr32¥drivers¥usbc¥usbc_otg.h

4.3.3. 各応用に対する特殊ファイル

- 応用ファイル
usb_conf.h (この形態設定ファイルは必須)

5. 応用プログラミング インターフェース

本章はそれ自身が資料を持つUDI APIを除いた全てのAPIを記述します。



5.1. UDCからの外部API

外部UDC APIは一般的なUSB装置の動きの管理と一般的なUSB装置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。

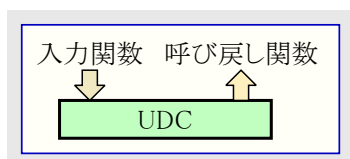


表5-1. UDCからの外部API - 入力

宣言	説明
udc_start()	USB装置階層開始
udc_stop()	USB装置階層停止
udcattach() udc_detach()	装置列挙(接続認識)認証またはDMかDPでのプルアップ不許可
udc_wakeup()	USB装置起き上がり

全てのUDC呼び戻しは任意選択で、各応用に対してusb_conf.hで使用者によって定義されます。

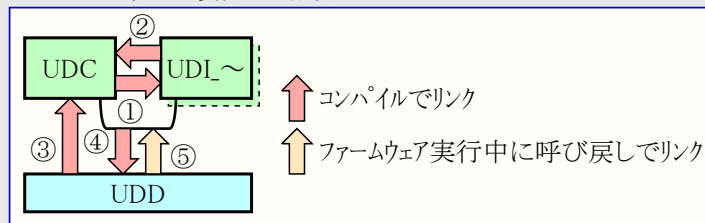
表5-2. UDCからの外部API - 呼び戻し

定義名	説明
UDC_VBUS_EVENT(bool b_present)	VBUSレベル変更通知のため(USBハードウェアがVBUS監視を含む場合のみ)
UDC_SUSPEND_ENENT()	USBが休止(サスペンド)形態移行時に呼ばれます。
UDC_RESUME_ENENT()	USB起き上がり時に呼ばれます。
UDC_SOF_EVENT()	1ms毎に受信した各SOFに対して呼ばれます。 注: 高速(High Speed)と全速(Full Speed)動作で利用可能
UDC_REMOTEWAKEUP_ENABLE()	装置がそれを支援している場合、USBホストが遠隔起動許可/禁止を要求する時に呼ばれます。
UDC_REMOTEWAKEUP_DISABLE()	
UDC_GET_EXTRA_STRING()	(製造業者、製品、通番の文字列以外に)特別な文字列記述子が支援されなければならない場合。
UDC_SPECIFIC_REQUEST()	特定装置構成設定要求が支援されなければならない場合。

5.2. 内部API

以下の定義はASFで提供されない特別なUSB装置を開発する高度な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5-3. UDCからのUDI入力 - ①

宣言	説明
bool (*enable)()	USBインターフェースを許可/禁止するためにUDCによって呼ばれます。
void (*disable)()	
bool (*setup)()	USB構成設定インターフェース(setup interface)要求が受信された時に呼ばれます。
uint8_t (*getsetting)()	USBインターフェースの現在の代替設定を得るためにUDCによって呼ばれます。
uint8_t (*sof_notify)()	許可されたUSBインターフェースでSOF事象を通知するためにUDCによって呼ばれます。

注: UDI APIはUDC記述子形態設定ファイル経由でUDC単位部と繋がられます(7.2項をご覧ください)。

表5-4. UDIからのUDC入力 - ②

宣言	説明
usb_iface_desc_t* udc_getiface()	UDIが呼ばれる(表5-3. UDI入力)時にUDCによって選択されたUSBインターフェース記述子を与えます。

表5-5. UDDからのUDC入力 - ③

宣言	説明
void udc_reset()	バス状態リセットが起きた時に呼ばれます。
bool udc_process_setup()	構成設定(setup)パケットが受信された時に呼ばれます。

表5-6. UDD入力 - ④

宣言	呼び出し者	説明
<code>void udd_enable()</code>	UDC	USB装置動作形態を許可/禁止
<code>void udd_disable()</code>		
<code>void udd_attach()</code> <code>void udd_detach()</code>	UDC	USB線路上にプルアップを挿入または取り外し
<code>void udd_set_address(uint8_t add)</code> <code>uint8_t udd_getaddress()</code>	UDC	USB装置アドレスを変更/戻す
<code>bool udd_is_high_speed()</code>	UDC/UDI	USB HS装置の場合に、列挙(接続認識)中に選ばれた速度を調査
<code>uint16_t udd_get_frame_number()</code>	応用	現在のフレーム開始(SOF)番号を返します。
<code>udd_send_wake_up()</code>	応用	USBドライバは“上方向再開”と呼ばれる再開信号を送ります。
<code>bool udd_ep_alloc(usb_ep_id_t ep, uint8_t bmAttributes, uint16_t wMaxPacketSize)</code> <code>udd_ep_free(usb_ep_id_t)</code>	UDC	エンドポイントを許可/禁止
<code>bool udd_ep_clear_halt(usb_ep_id_t)</code> <code>bool udd_ep_set_halt(usb_ep_id_t)</code> <code>bool udd_ep_is_halted(usb_ep_id_t)</code>	UDC/UDI	エンドポイントの状態(停止か否か)を解除/設定/取得
<code>bool udd_ep_wait_stall_clear(udd_ep_id_t endp, udd_callback_nohalt_t callback)</code>		エンドポイント停止が取り除いた時に呼ぶための呼び戻しを登録
<code>bool udd_ep_run(usb_ep_id_t endp, bool b_shortpacket, uint8_t *buf, Uint32_t u#"_size_buf, udd_callback_trans_t callback)</code> <code>udd_ep_abort(usb_ep_id_t endp)</code>	UDI	エンドポイントでのデータ転送入力または出力を開始/停止 注: 制御エンドポイントはここで認可されません。

表5-7. UDD呼び戻し - ⑤

宣言	説明
<code>typedef void (*udd_callback_nohalt_t)(void);</code>	エンドポイントでの停止が取り除かれた時に呼ばれます。 このものは <code>udd_ep_wait_stall_clear()</code> 経由で登録されます。
<code>typedef void (*udd_callback_trans_t)(udd_ep_status_t status, iram_size_t nb_transferred)</code>	転送要求が終了または取り消された時に呼ばれます。 このものは <code>udcdrv_ep_run()</code> 経由で登録されます。

表5-8. 高速(High Speed)応用専用UDD入力 - ④

宣言	呼び出し者	説明
<code>uint16_t udd_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>udd_test_mode_j()</code> <code>udd_test_mode_k()</code> <code>udd_test_mode_se0_nak()</code> <code>udd_test_mode_packet()</code>	UDC	USB HS装置を検査するための機能。これらはUSB検定を動かすために要求されます。

全域変数`udd_g_ctrlreq`はUDDによって宣言され、次のような2つの部分を含みます。

- UDDによって更新され、UDCとUDIによって使用される値(表5-9)
- UDCとUDIによって更新され、UDDによって使用される値(表5-10)

UDDの外側では、この変数がUDCに関して`udc_process_setup()`、UDIに関して`*setup()`によって処理されます(図6-2をご覧ください)。

表5-9. UDDによって更新されるudd_g_ctrlreq領域

宣言	説明
usb_setup_req_t req	構成設定(SETUP)パケット内に含まれ、要求の復号に使用される値
uint8_t *payload	緩衝部の内容がUDDによって送られる、または満たされる。 u16_sizeが0と等しい場合、NULLにすることができます。

表5-10. UDCまたはUDIIによって更新されるudd_g_ctrlreq

宣言	説明
uint8_t *payload	送るまたは満たすための緩衝部の位置指示値
uint16_t u16_size	送るまたは満たすための緩衝部の大きさ DATA段階が必要とされない時には0にすることができます。
bool over_under_run(void)	与えられた緩衝部(.payload)が一杯または空の時にUDDによって呼ばれます。 NULLにすることができます。
void *callback(void)	構成設定要求が終わった(SETUP+DATA+ZLP)時にUDDによって呼ばれます。 NULLにすることができます。

6. 動き

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

階層実装はUSB割り込みによってのみ管理されるUDC階層を提供します。この解決策は最小遅延を保証し、どんな待機繰り返しも必要とせず、OS適合性を保証します。

USB割り込みルーチンはどれ程の長さ妨げられますか？

USB割り込みルーチンは低優先権で許可され、より高い優先権を持つ別の割り込みルーチンや重要なコード領域によって妨げられ得ます。USB仕様に従い、AVRのUSBハードウェアとソフトウェアは(USBアドレスを割り当てるためにUSB列挙(接続認識)段階中に実行される)USBアドレス設定(Set Address)要求を除いてどんな時間制限も持ちません。

表6-1. アドレス設定(Set Address)タイミング

USBホスト	最大時間(注)
仕様	2ms
USB org検証ツール	12ms
Windows® XP	48ms
Windows Vista, 7	32ms
Mac Mini OSX 10.5.8	77ms
Ubuntu 8.04, Ubuntu 9, Open Suse 11.1	29ms
Fedora 9, Fedora 10	24ms

注: これらのタイミングはUSBホストハードウェアに依存し得ますが、その範囲を与えます。また、この時間は構成設定再試行時間を含みます。

USBホストは無応答USB装置をリセットするために時間制限を用品(この時間はUSB仕様によって指定されません)。

以下はオペレーティングシステムの時間制限の例です。

表6-2. OS時間制限

USBホスト	制限時間				
	制御エンドポイント		大容量記憶		
	データ段階	ZLP段階	CBW	データ読み込み	CSW
仕様	制限時間なし				
Windows® XP	5.3s	5.3s	19s	9.3s	9.3s
Windows Vista, 7	5.3s	5.3s	19s	160s/60s	160s/60s
Mac Mini OSX 10.5.8	5.9s	5.6s	11s	31s	22s
Ubuntu 8.04, Ubuntu 9, Open Suse 11.1	5s	5s	30s	30s	30s
Fedora 9, Fedora 10	5s	5s	30s	60s	30s

以下の図は各種層間での相互作用を記述します。

図6-1. USB装置の始動と停止

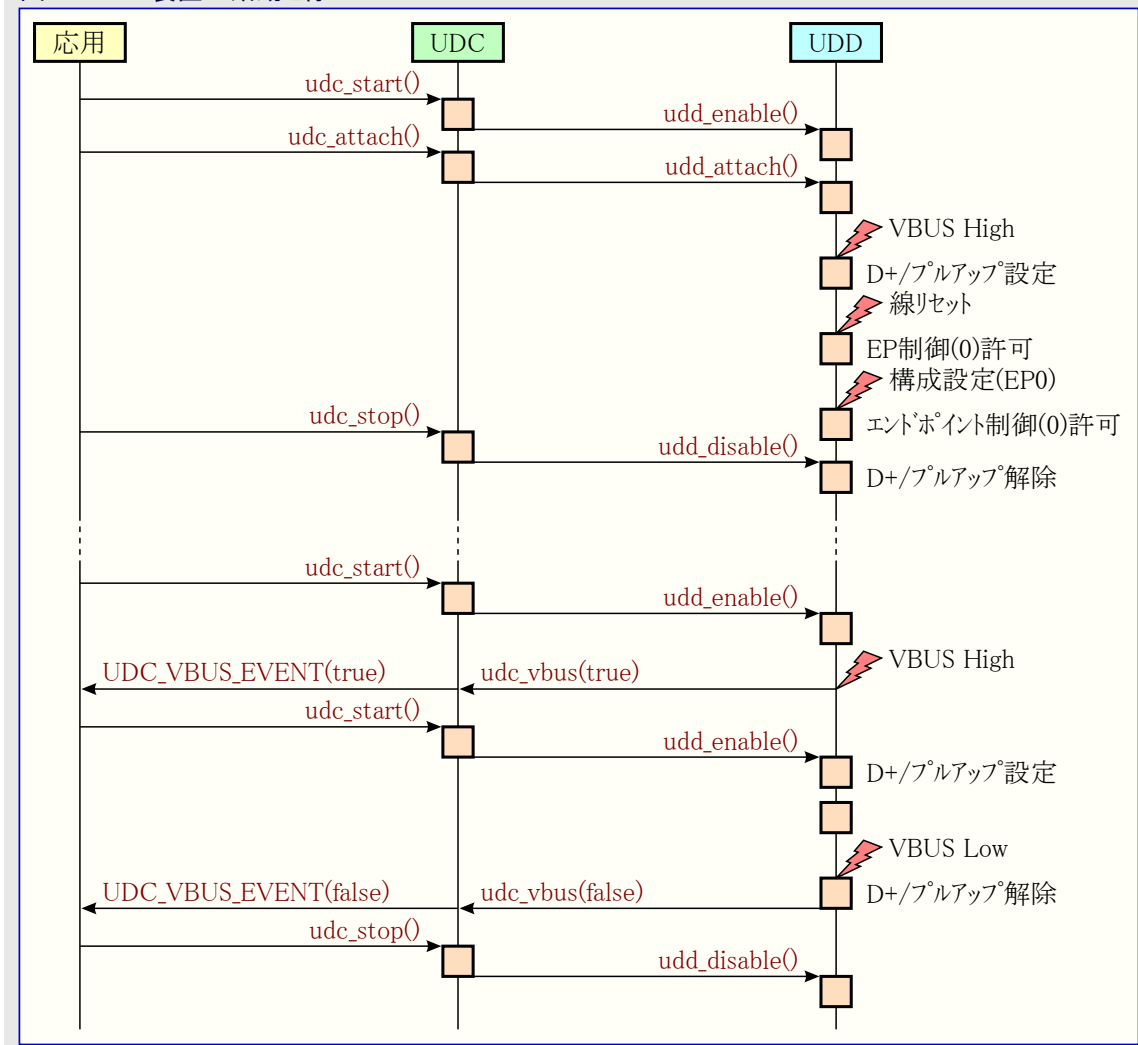
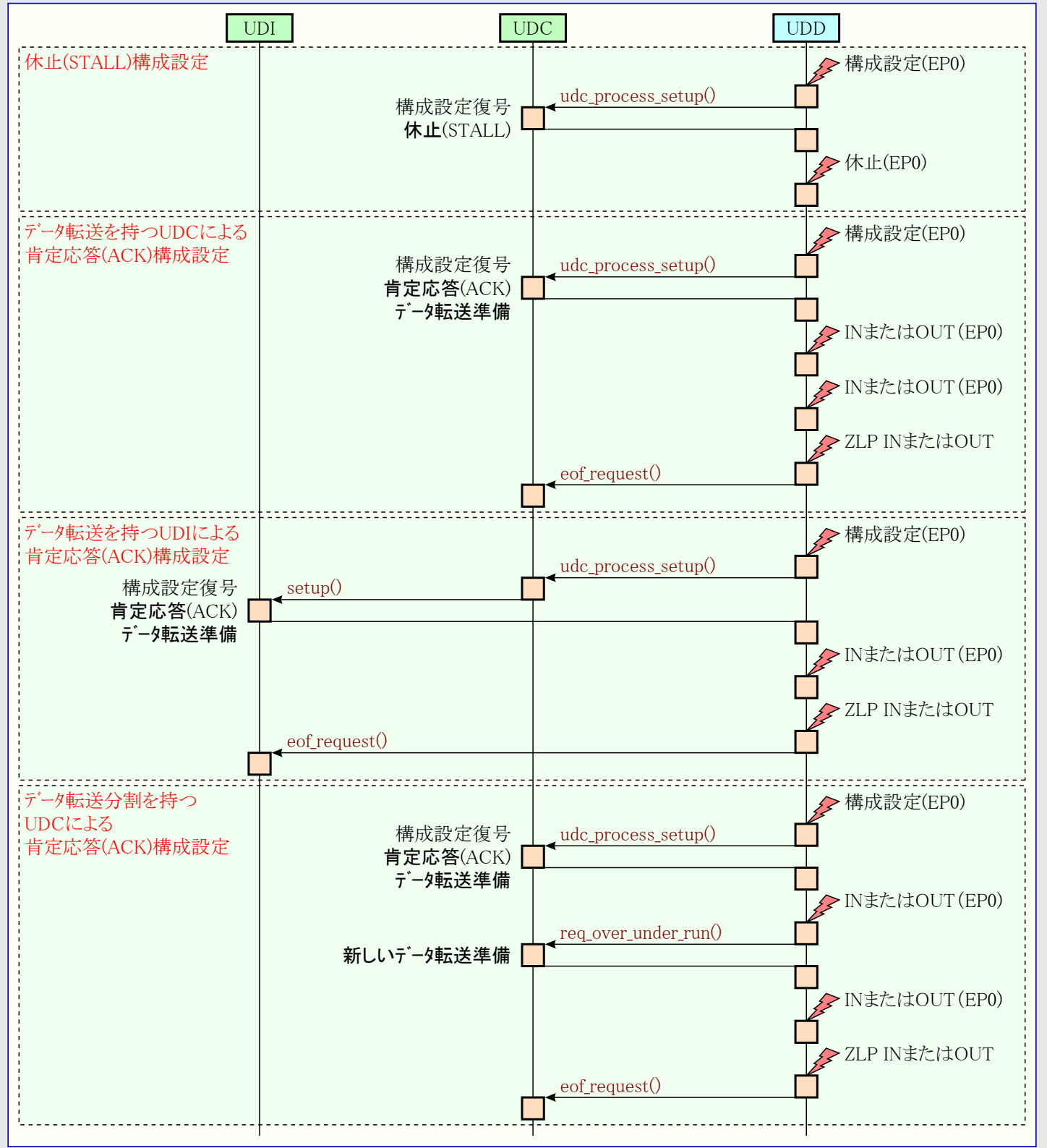
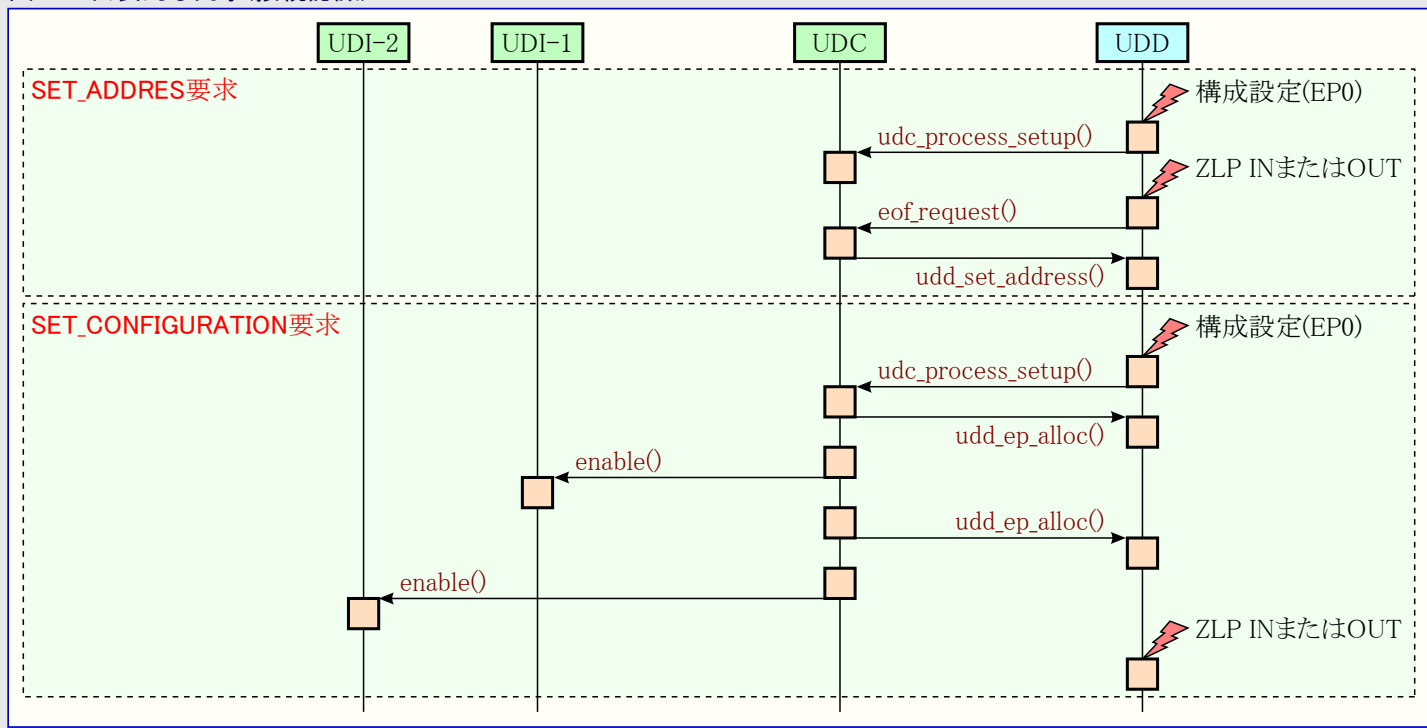


図6-2. 制御エンドポイントの管理



注: `udd_g_ctrlreq`変数はUDDとUDC/UDI間の通信に使用されます(表5-9と表5-10をご覧ください)。

図6-3. 代表的な列挙(接続認識)



7. 形態設定

形態設定は応用とUSB記述子の2つの部分に分けられます。

応用の形態設定は`conf_usb.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.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装置形態設定
- USBインターフェース形態設定
- USBドライバ形態設定

USB記述子形態設定はATMELによって提供された既定形態設定が使用されない時に要求されます。この形態設定情報は高度なUSB使用者に関して利用可能です。

7.1. USB形態設定

7.1.1. USB装置形態設定

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

表7-1. USB装置形態設定

定義名	型	説明
USB_DEVICE_VENDOR_ID	語(ワート)	USB.orgによって提供される供給者ID(ATMEL=\$03EB)
USB_DEVICE_PRODUCT_ID	語(ワート)	(<code>usb_atmel.h</code> で参照される)製品ID
USB_DEVICE_MAJOR_VERSION	バイト	装置の主版番号
USB_DEVICE_MINOR_VERSION	バイト	装置の副版番号
USB_DEVICE_MANUFACTURE_NAME (注2)	文字列 (注1)	製造業者の静的ASCII名
USB_DEVICE_PRODUCT_NAME (注2)	文字列 (注1)	製品の静的ASCII名
USB_DEVICE_SERIAL_NAME (注2)	文字列 (注1)	通番を許可して設定するための静的ASCII名
USB_DEVICE_GET_SERIAL_NAME_POINTER() (注2)	const uint8_t 関数(void)	通番を許可して設定するための動的ASCII名での位置指示子を与えます。 USB_DEVICE_GET_SERIAL_NAME_LENGTHが必要で、USB_DEVICE_SERIAL_NAMEを無効にします。
USB_DEVICE_GET_SERIAL_NAME_LENGTH() (注2)	uint8_t 関数(void)	通番を許可するのに使用される動的ASCII名の長さを与えます。
USB_DEVICE_POWER	数値	最大装置電力(mA)
USB_DEVICE_ATTR	バイト	以下の機能を許可するために追加するUSB属性 <ul style="list-style-type: none"> • USB_CONFIG_ATTR_SELF_POWERED • USB_CONFIG_ATTR_REMOTE_WAKEUP 注: 遠隔起動機能が許可される場合、遠隔起動(remotewakeup)呼び戻しを定義します(表5-2をご覧ください)。
USB_DEVICE_LOW_SPEED (注2)	(定義のみ)	低速(Low Speed)での走行をUSB装置に強制します。
USB_DEVICE_HS_SUPPORT (注2)	(定義のみ)	高速(High Speed)での走行にUSB装置を認定します。
USB_DEVICE_MAX_EP	バイト	装置によって使用される最大エンドポイント数を定義します(制御エンドポイントは含みません)。

注1: 文字列構文法の例:

```
#define USB_DEVICE_MANUFACTURE_NAME "ATMEL"
```

定義は省略でき、従って文字列はUSB列挙(接続認識)で取り去られます。

注2: 任意選択形態設定。それを禁止するにはその定義命令行を注釈にしてください(例: // #define USB_DEVICE_X)。

7.1.2. USBインターフェース形態設定

UDI形態設定はUSB装置クラス応用記述で記述されます。

7.1.3. USBドライバ形態設定

以下の形態設定は応用の`conf_usb.h`ファイルに含まれなければなりません。

AVR製品はここで許可することができる特別なハードウェア機能を提供します。

表7-2. USB装置ドライバ形態設定

定義名	値	UDD	説明
UDD_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。
UDD_ISOCHRONOUS_NB_BANK	1,2,3	AVR32 - USBB	等時エンドポイント緩衝を減少または増加 (既定値:2)
UDD_BULK_NB_BANK	1,2,3	AVR32 - USBB	大量(バルク)エンドポイント緩衝を減少または増加 (既定値:2)
UDD_INTERRUPT_NB_BANK	1,2,3	AVR32 - USBB	割り込みエンドポイント緩衝を減少または増加 (既定値:1)
UDD_USB_INT_LEVEL	0~3	AVR32 - USBB AVR32 - USBC	AVR32コアでUSB割り込みレベルを設定 (既定値:0 (推奨))

7.2. USB記述子

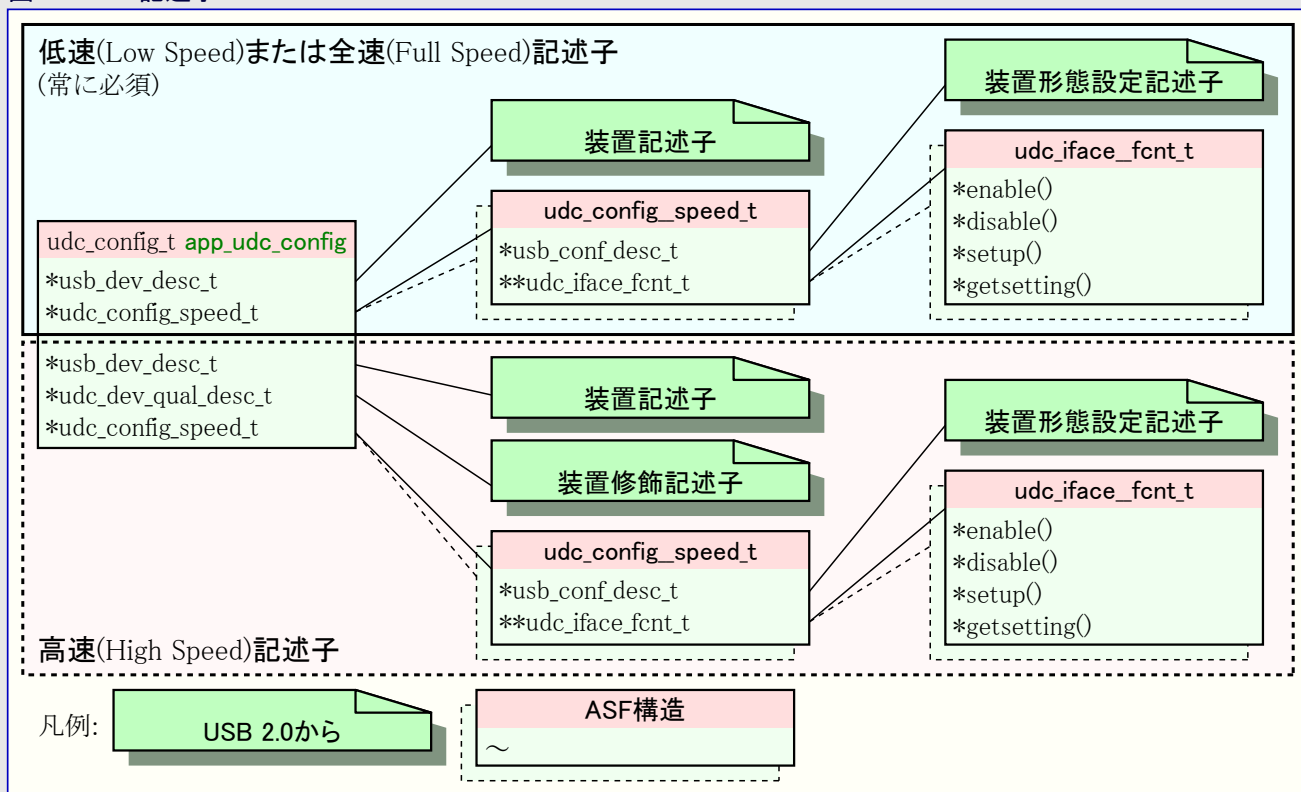
この項は新しいUDIまたは複合USB装置の作成を望むUSB開発者に向けられています。

ASFによって既に提供されたUSBクラスは既定USB装置記述子を含みます。これらの記述子は`udi_foo_desc.c`と`udi_foo_conf.h`のUDIファイルで定義され、全てのUDI応用記述で記述される容易な実装を許します。

記述子ファイルは以下を含む`app_udc_config`全域変数を宣言します。

- 可能な各速度に対する装置記述子 (`usb_dev_desc_t`)
- 高速(High Speed)装置の場合での1つの装置修飾 (`usb_dev_qual_desc_t`)
- 各形態設定に対する形態設定記述子 (`usb_conf_desc_x_t`)
- UDIと形態設定記述子間の繋げ (`udc_iface_fcnt_t`)

図7-1. USB記述子



より多くの情報についてはATMELの「ASF - USB装置新規クラス」応用記述をご覧ください。

8. 電力消費

AVR製品で利用可能な電力形態はUSB線状態に従ってUSBハードウェアによって支援することができます。USBドライバはそれら全ての節電動作形態を管理するのに休止管理(sleepmgr)サービスを使います。USB応用が作成される時に休止管理サービス初期化(`sleepmgr_init()`)が必要とされます。

8.1. USBBとUSBCの休止形態

全てのAT32UC3休止形態は各AT32UC3データシートの「電力管理部」章で記述されます。休止形態はUSBBとUSBCのドライバが以下のようなことによって支援されます。

- USBアイドル(IDLE)状態で、ドライバはUSBクロックが必要で、アイドル(IDLE)形態まで認可します。
- USB休止(SUSPEND)状態で、ドライバはUSBクロックが必要ではありませんが、クロック再開の間の最小タイミングを要求します。従って、それは静止(STATIC)または待機(STANDBY)の形態まで支援されます。
- USB自己給電形態で使用されるVBUS監視は停止(STOP)形態まで認可します。

表8-1. AT32UC3でのUSB休止(サスペンド)状態で支援される最大休止レベル

USB電力形態	USB速度形態	USBクロック始動	休止形態認可
バス給電と自己給電	LS, FS	> 10ms	待機(STANDBY)
バス給電と自己給電	HS	> 3ms	待機(STANDBY)
自己給電	LS, FS	≤ 10ms	停止(STOP)
自己給電	HS	≤ 3ms	停止(STOP)
バス給電	LS, FS	≤ 10ms	静止(STATIC)
バス給電	HS	≤ 3ms	静止(STATIC)

注: USBクロックを生成するのに度々外部発振器が使用されます。従って、USBクロック始動タイミングは発振器の始動タイミングに相当します。

AT32UC3系統はUSB休止(サスペンド)形態での電力制限(2.5mA)を容易に支援します。

従って、どんなバス給電装置応用に対しても、以下が必要です。

- `conf_usb.h`ファイルからUSB_DEVICE_ATTR定義内のUSB_CONFIG_ATTR_SELF_POWEREDビットを取り去ってください。
- 低始動時間を持つ外部発振器を使用してください。この値はBOARD_OSC0_STARTUP_US定義経由で基板ヘッダ内で指定されます。始動時間の可能性で取り計らい、AT32UC3データシートでOSCCTRL0レジスタの可能性をご覧ください。
- 3.3VのAT32UC3電源生成に外部低損失調整器が必要とされます。この調整器を選択する時に、静止電流が全体2.5mA休止(サスペンド)電流の大きすぎる割合を消費しないことを確実にしてください。

8.2. ATxmega休止形態

全てのATxmega休止形態は各ATxmegaデータシートの「電力管理部」章で記述されます。休止形態はUSBドライバが以下のようになることによって支援されます。

- USBアイドル(IDLE)状態で、ドライバはUSBクロックが必要で、アイドル(IDLE)形態まで認可します。
- USB休止(SUSPEND)状態で、ドライバはUSBクロックが必要ではありませんが、クロック再開の間の最小タイミングを要求します。従って、それは電力停止(POWER DOWN)または待機(STANDBY)の形態まで支援されます。

表8-2. ATxmegaでのUSB休止(サスペンド)状態で支援される最大休止レベル

USB電力形態	USB速度形態	USBクロック始動	休止形態認可
バス給電と自己給電	LS, FS	> 10ms	待機(STANDBY)
バス給電と自己給電	LS, FS	≤ 10ms	電力停止(POWER DOWN)

注: USBクロックを生成するのに度々内部発振器が使用されます。従って、USBクロック始動タイミングは<10msです。

ATxmega系統はUSB休止(サスペンド)形態での電力制限(2.5mA)を容易に支援します。

従って、どんなバス給電装置応用に対しても、以下が必要です。

- `conf_usb.h`ファイルからUSB_DEVICE_ATTR定義内のUSB_CONFIG_ATTR_SELF_POWEREDビットを取り去ってください。
- 低始動時間を持つ内部発振器を使用してください。
- 3.3VのATxmega電源生成に外部低損失調整器が必要とされます。この調整器を選択する時に、静止電流が全体2.5mA休止(サスペンド)電流の大きすぎる割合を消費しないことを確実にしてください。

9. 変更記録

9.1. 8360A-12/10 初版

9.2. 8360B-04/11

1. 電力消費に関する全ての項を更新
2. UDIとUDDのAPIを更新
"sof_notify()"とudd_get_microframe_number()記述をご覧ください。

9.3. 8360C-09/11

1. 動的通番実装のための新しい任意選択を追加
2. ATxmega製品情報を追加
3. 要点一覧で、高速(High Speed) 48Mビット/秒を480Mビット/秒によって修正

10. 目次

要点	1
1. 序説	1
2. 略語	1
3. USB装置応用記述	2
4. 構成	2
4.1. 概要	2
4.2. メモリ使用空間	2
4.3. USB装置階層ファイル	3
4.3.1. 全AVR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UDD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. UDCからの外部API	3
5.2. 内部API	4
6. 動き	6
7. 形態設定	9
7.1. USB形態設定	10
7.1.1. USB装置形態設定	10
7.1.2. USBインターフェース形態設定	10
7.1.3. USBドライバ形態設定	10
7.2. USB記述子	11
8. 電力消費	11
8.1. USBとUSBCの休止形態	11
8.2. ATxmega休止形態	12
9. 変更記録	12
9.1. 8360A-12/10	12
9.2. 8360B-04/11	12
9.3. 8360C-09/11	12
10. 目次	13



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

© 2011 Atmel Corporation. 全権利予約済

ATMEL®、ATMELロゴとそれらの組み合わせ、それとAVR®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。Windows®とその他は米国とその他の国に於いてMicrosoft Corporationの登録商標です。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のウェブサイトに位置する販売の条件と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AVR4900応用記述(doc8360.pdf Rev.8360C-09/11)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。