AVR4901 : ASF – USB供給者ｸﾗｽ応用

要点

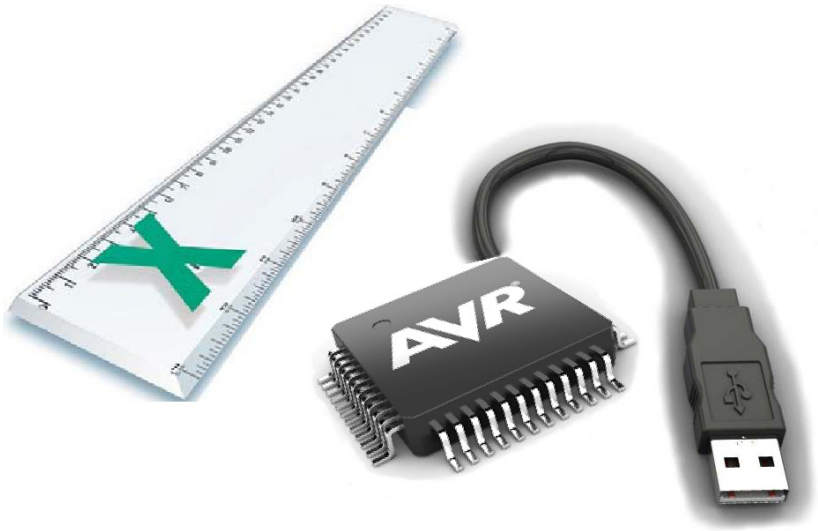
- USB 2.0適合
  - USB第9章適合
  - 全速(Full,12Mﾋﾞｯﾄ/秒)、高速(High,480Mﾋﾞｯﾄ/秒)のﾃﾞｰﾀ速度
- 制御、大量(Bulk)、等時(Isochronous)、割り込みの転送形式
- 小さなｽﾀｯｸの大きさが主応用の空間を空ける
- 遠隔起動支援
- USBﾊﾞｽ給電支援
- 実時間(OS適合、割り込み駆動)
- 8ﾋﾞｯﾄと32ﾋﾞｯﾄのAVR®を支援

8/32ﾋﾞｯﾄ ATMEL
マイクロ コントローラ

応用記述

1. 序説

この資料の狙いは新規または既存ﾌﾟﾛｼﾞｪｸﾄでUSB供給者ｸﾗｽを統合する簡単な方法を提供することです。



2. 略語

- ASF : AVRｿﾌﾄｳｪｱ枠組み(AVR Software Framework)
- CD : 複合装置(複数のｲﾝﾀｰﾌｪｰｽを持つUSB装置)(Composite device)
- CDC : 通信装置ｸﾗｽ(Communication Device Class)
- 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)
- 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 : 万能直列ﾊﾞｽ(Universal Serial Bus)
- SOF : ﾌﾚｰﾑ開始(Start of frame)

Rev. 8481A-01/12, 8481AJ1-03/14

3. 概要

この資料はUSB装置供給者応用を構築する時の必要条件の全ての形式に対する以下の6つの部分を含みます。

- **供給者ｸﾗｽ仕様**
  供給者ｸﾗｽ仕様を持つ使用者を手助けし、それらがそのような解決策を必要とする時の情報を提供します。
- **即時開始**
  供給者ｸﾗｽ装置例を使用するための準備を開始する方法を記述します。
- **例の説明**
  供給者ｸﾗｽ装置例を説明します。
- **USB装置供給者の構築**
  ﾌﾟﾛｼﾞｪｸﾄにUSB装置供給者を追加する方法を記述します。
- **USB供給者ｸﾗｽ応用の構築**
  USB供給者ｸﾗｽ応用を追加する方法を記述します。
- **USB複合装置での供給者ｸﾗｽ**
  複合装置ﾌﾟﾛｼﾞｪｸﾄに供給者ｸﾗｽ ｲﾝﾀｰﾌｪｰｽを統合する方法を記述します。

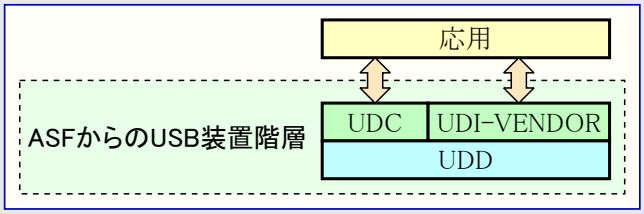
これらの部分の全てに対して、以下のような供給者ｸﾗｽ応用の主な単位部構成を知ることが推奨されます。

- 使用者応用
- USB装置ｲﾝﾀｰﾌｪｰｽVENDOR (UDI-VENDOR)
- USB装置制御部 (UDC)
- USB装置ﾄﾞﾗｲﾊﾞ (UDD)

USB階層実装に関するもっと高度な情報についてはATMEL®「AVR4900:ASF-USB装置階層」応用記述を参照してください。

注: USB装置階層はcommon¥services¥usbﾃﾞｨﾚｸﾄﾘに於いてASFで利用可能です。

図3-1. USB供給者ｸﾗｽ解決策基本構造



4. 供給者ｸﾗｽ仕様

供給者ｸﾗｽの開発は既存ｸﾗｽ(http://www.usb.org/developers/defined_class)が使用者仕様に対応しない時に必要とされ得ます。
供給者ｸﾗｽ仕様開始に先立ち、本来のUSBｸﾗｽがあなたのUSB応用に利用不能であることを表4-1.で調べてください。

表4-1. 本来ｸﾗｽの分析

ｸﾗｽ	有利な点	不利な点
HID	● 本来のﾄﾞﾗｲﾊﾞ ● USB低速(Low-Speed)支援 ● 応用の制御または監視に適合	● 割り込みｴﾝﾄﾞﾎﾟｲﾝﾄ: ○ 低速(Low-Speed))と全速(Full-Speed)=64Kﾊﾞｲﾄ/s ○ 高速(High-Speed)=2.93Mﾊﾞｲﾄ/s
CDC	● 高転送速度 ● 簡単なﾎｽﾄ応用(端末) ● USARTに基づく応用へ容易に移植	● Windows®下で*.infﾌｧｲﾙが必要 ● WHQL認証されないﾄﾞﾗｲﾊﾞ
MSC	● 本来のﾄﾞﾗｲﾊﾞ	● 大容量記憶に対してだけ適応されたSCSI規約
AUDIO	● 本来のﾄﾞﾗｲﾊﾞ ● ﾃﾞｰﾀの流れの記録または送出への興味	● ﾊﾝﾄﾞｼｪｰｸ ﾃﾞｰﾀ転送なし(等時)

供給者ｸﾗｽ構築時、指定のための主な重要なものはｴﾝﾄﾞﾎﾟｲﾝﾄによって使用される転送形態です。
それらは表4-2.で与えられる記述に基づいて選ぶことができます。

表4-2. ｴﾝﾄﾞﾎﾟｲﾝﾄ形式記述

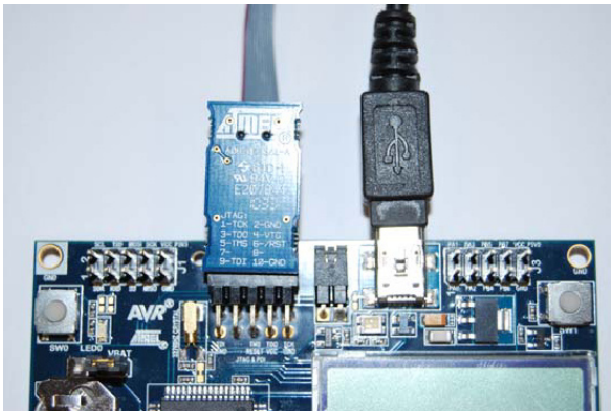
ｴﾝﾄﾞﾎﾟｲﾝﾄ形式	説明	使用事例
制御 (Control)	● 低転送速度 ● 応用使用量を減らし得る新規ｴﾝﾄﾞﾎﾟｲﾝﾄ不要	● 装置の形態設定
割り込み (Interrupt)	● 非常に遅い転送 ● 保証された周波数(繰り返し頻度)	● 実時間制限と低帯域転送
大量 (Bulk)	● 高転送速度 ● 保証された転送(ACK)	● 大量ﾃﾞｰﾀの転送
等時 (Isochronous)	● 非常に高速な転送速度 ● 保証されない転送	● ﾃﾞｰﾀの流れの応用

5. 即時開始

USB装置供給者例はATMEL AVR Studio® 5とASFで利用可能です。下記はUSB装置供給者ｸﾗｽ例を素早く開始するための手順です。

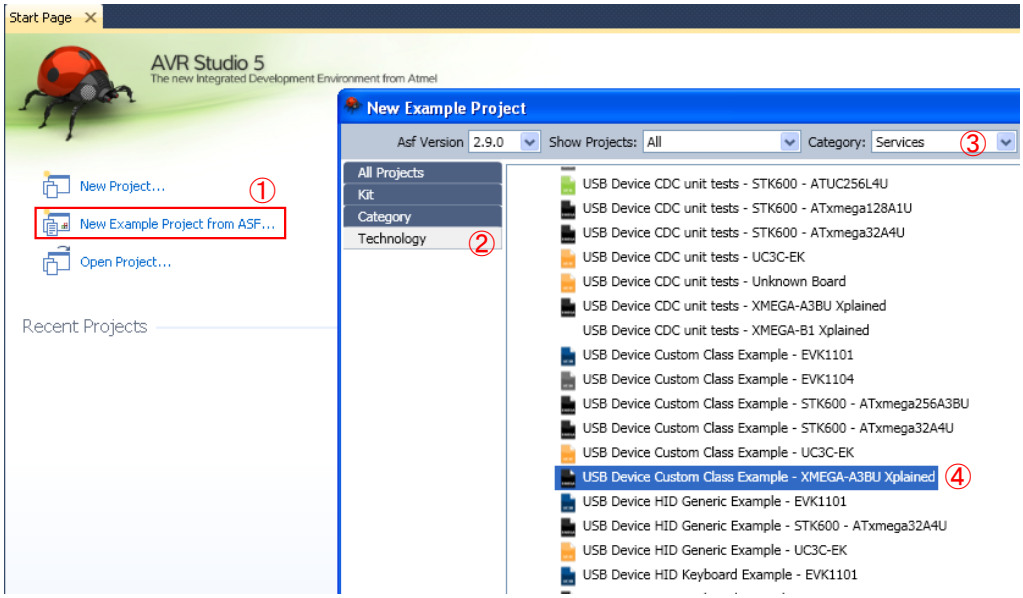
1. USBｹｰﾌﾞﾙを使用して基板(例えばATMEL AVR XMEGA®-A3BU Xplainｷｯﾄ)をPCに接続してください。
2. ﾃﾞﾊﾞｲｽをﾌﾟﾛｸﾞﾗﾐﾝｸﾞするためにATMELのﾃﾞﾊﾞｯｶﾞ(今後はJTAGICE3)を接続してください。

図5-1. 基板接続



3. AVR STudio 5を開始し、"New Example Project from ASF"をｸﾘｯｸしてください。

新しいﾌﾟﾛｼﾞｪｸﾄ例(New Example Project)の一覧で、使用する基板に基づく"USB Device Vendor Class Example"を選んでください。例を素早く見つけ出すために濾過一覧を使用することができます。



4. ｺﾝﾊﾟｲﾙして書き込み設定し、そして実行してください。

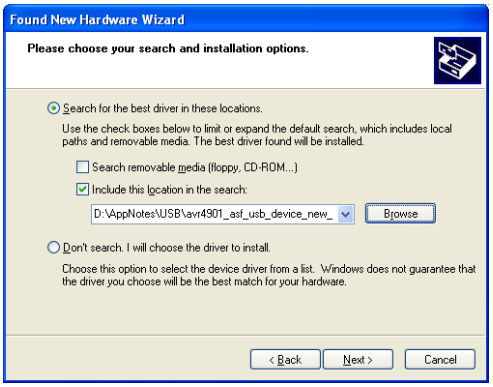
ﾌﾟﾛｼﾞｪｸﾄはどんな変更も必要なく、ｺﾝﾊﾟｲﾙされて書き込み設定し、そして走行することだけが必要です。基板によって支援されるATMELのﾃﾞﾊﾞｯｶﾞを接続してF5を押してください。

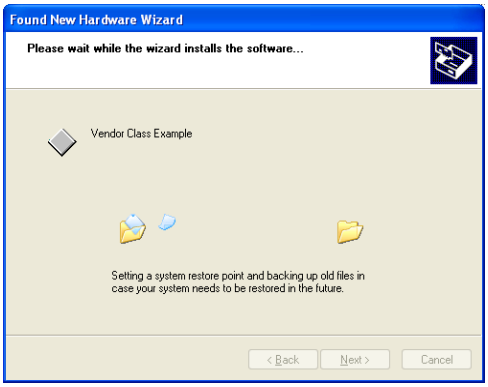
初回の応用走行時に右で示すような新規ﾊｰﾄﾞｳｪｱ発見ｳｨｻﾞｰﾄﾞ(Found New Hardware Wizard)が起き上がるので、「今回は行いません。(No, not this time)」を選択して「次へ(Next)」をｸﾘｯｸしてください。

5. infファイルを使用してドライバを設定してください。

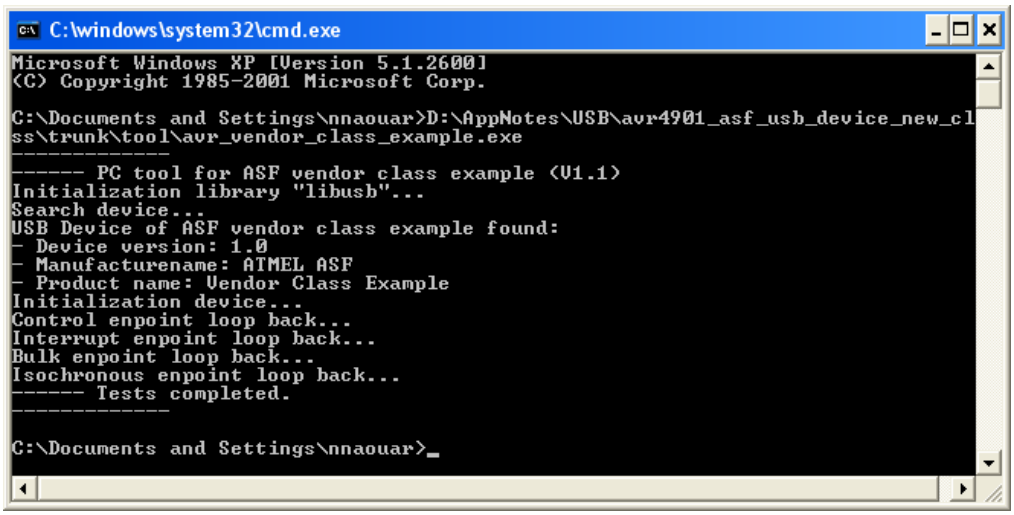
注: USB供給者クラスはどの本来のドライバによっても支援されないため、使用者は専有ドライバを提供するか、または第三者の解決策を使用することが必要です。

この例については、この装置を支援するのに開放ソース解決策の"libusb"が使用されます。infファイルはlibusbツールによって生成されました。このドライバをインストールするにはavr4901¥driver¥フォルダを指示してください。





一旦装置が正しくインストールされたなら、avr4901¥tool¥フォルダからavr_vendor_class_example.exeを開始してください。このプログラムは全てのエンドポイントで送り返しを実行します。応用の正しい実行を示すために状態が表示されます。

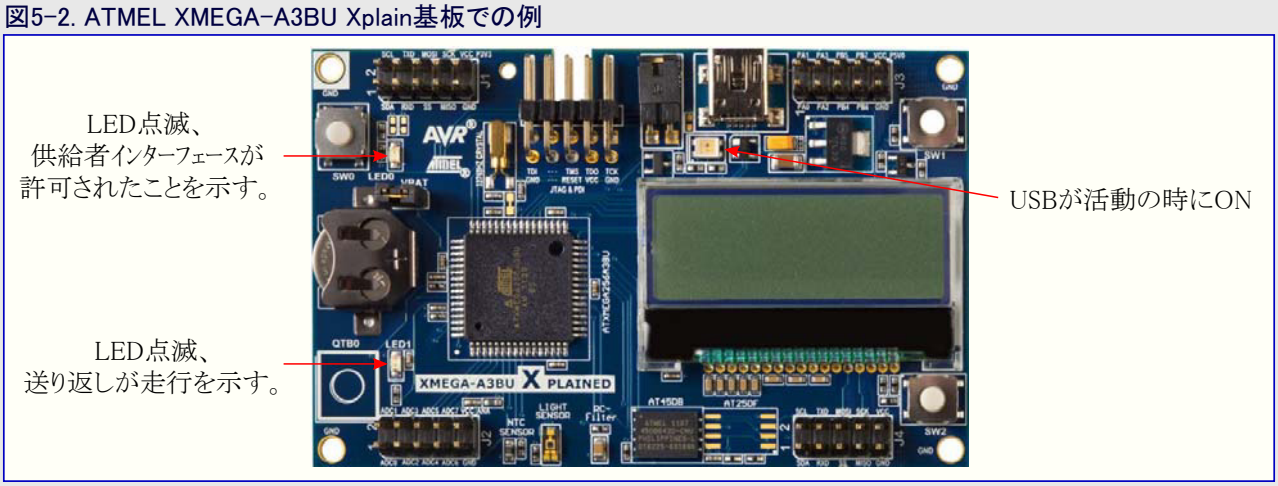


5.1. ハードウェアの動き

応用実行中にハードウェア(使用基板)は以下のように動くべきです。

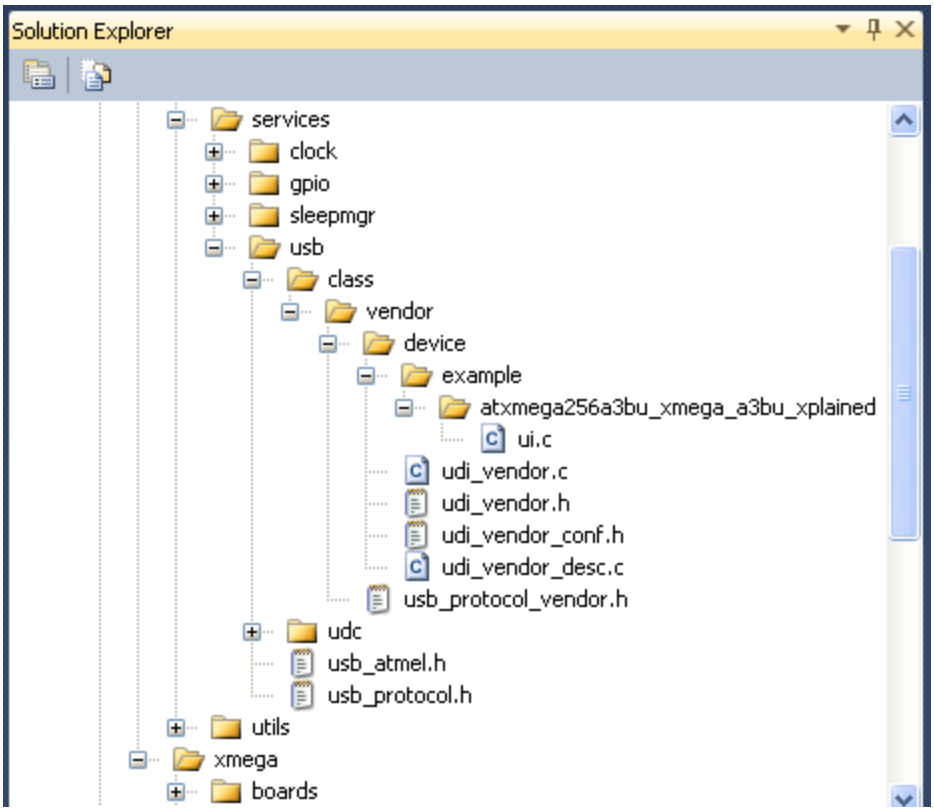
- LEDはUSB装置が活動形態の時にON、休止(Suspend)形態でOFFです。
- LEDはUSBホストによって供給者クラス インターフェースが許可されたことを示すのに点滅します。
- LEDは送り返しが走行している時にONです。

図5-2. ATMEL XMEGA-A3BU Xplain基板での例



(基板のために指定する)使用者インターフェース記述子はui.cソース ファイルの終りで定義されます。このファイルは"common¥services¥usb¥class¥vendor¥device¥example¥part_number_board¥"下の"Solution Explorer"内で利用可能です。

図5-3. ui.cファイル位置



6. 例の説明

6.1. 例の内容

ASFは様々なATMEL AVR製品に対してUSB装置供給者クラス例を提供します。これら全ての例は共通ファイルを共用し、同じ応用を実装します。

表6-1.はUSB装置供給者例内に含まれる主なファイルの要約を紹介します。これらのファイルは図3-1.で記述される単位部と連携します。

表6-1. USB装置供給者例のファイル

単位部	ファイル	ASFパス	説明
応用	main.c ui.c conf_usb.h	Examplesフォルダ	主繰り返し スイッチと動作を示すためのLEDのハードウェア構成設定 USB装置形態設定
UDI-VENDOR	udi_vendor.c/h	common¥services¥usb¥class¥vendor¥device¥	供給者クラス実装
	udi_vendor_desc.c udi_vendor_conf.h	common¥services¥usb¥class¥vendor¥device¥	供給者インターフェースを持つUSB装置用USB記述子 (USB複合装置に適用不可)
UDC	udc.c/h udc_desc.h udi.h udd.h	common¥services¥usb¥udc¥	USB装置核
	usb_protocol.h usb_atmel.h	common¥services¥usb¥	USB規約定数
UDD	usbb_device.c/h usbc_device.c/h usb_device.c/h	avr32¥drivers¥usbb¥ avr32¥drivers¥usbc¥ xmega¥drivers¥usb¥	USBドライバ

6.2. 例の動き

main.cとui.cのファイルが供給者応用の使用者インターフェースを実装します。それは3つの段階に基づきます。

1. USB装置を開始します。

```
udc_start();
udc_attach();                       // USBケーブルが接続された時に呼ばれなければなりません。
                                    // 接続されたケーブルはVBus事象経由で検知されます。
```

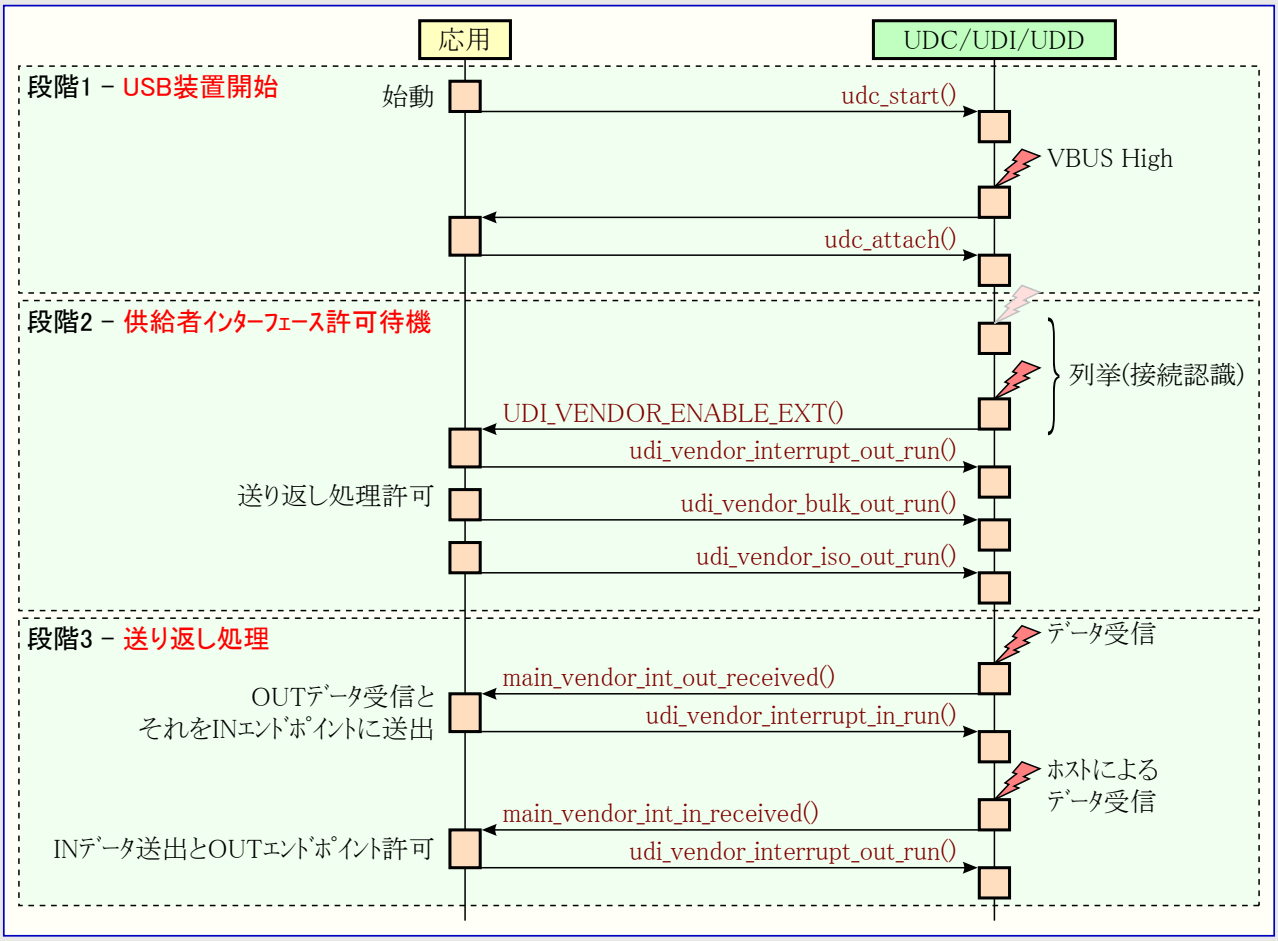
2. 呼び戻し経由での供給者インターフェースの許可と送り返し処理の許可を待ちます。

```
UDI_VENDOR_ENABLE_EXT()             // 許可されたインターフェース呼び戻し
呼び戻しは以下を呼びます。:
- udi_vendor_bulk_out_run()         // 大量(Bulk)OUTエンドポイントでの転送許可
- udi_vendor_interrupt_out_run()    // 割り込みOUTエンドポイントでの転送許可
- udi_vendor_iso_out_run()          // 等時(Isochronous)OUTエンドポイントでの転送許可
```

3. 全てのエンドポイント(制御、割り込み、大量(Bulk)、等時(Isochronous))で送り返しを実行します。

```
// OUTデータを受信すると、その後にこのデータをINで送出
void main_vendor_int_out_received(udd_ep_status_t status, iram_size_t nb_transfered)
{
    if (UDD_EP_TRANSFER_OK != status) {
        return;                     // 転送中止、そして送り返し停止
    }
    ui_loop_back_state(true);
    // OUTエンドポイントで受信したデータをINエンドポイントで送出
    udi_vendor_interrupt_in_run(main_buf_loopback, nb_transfered, main_vendor_int_in_received);
}
// INデータが送られてしまうと、OUT転送が再び許可されます。
void main_vendor_int_in_received(udd_ep_status_t status, iram_size_t nb_transfered)
{
    if (UDD_EP_TRANSFER_OK != status) {
        return;                     // 転送中止、そして送り返し停止
    }
    ui_loop_back_state(false);
    // 緩衝部満杯待ち
    udi_vendor_interrupt_out_run(main_buf_loopback, sizeof(main_buf_loopback),
                                 main_vendor_int_out_received);
}
```

図6-1. 例の動きの流れ



7. USB装置供給者の構築

USB装置供給者単位部はUSB供給者クラス応用を構築するのに使用することができるUSB供給者インターフェースを提供します。

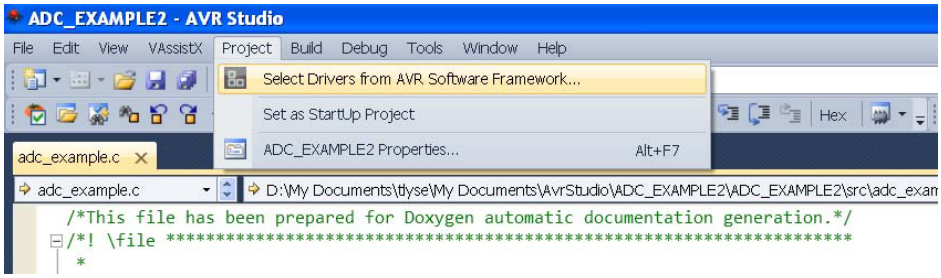
これらの単位部はATMEL AVR Studio 5で利用可能で、AVR Studio 5プロジェクトに取り込む(インポートする)ことができます。本章はプロジェクトにUSB装置供給者を追加する方法を記述します。

1. USB供給者単位部をインポートしてください。
2. 私的なUSBパラメータを形態設定してください。
3. USB装置を走らせるためにUSBルーチンを呼んでください。

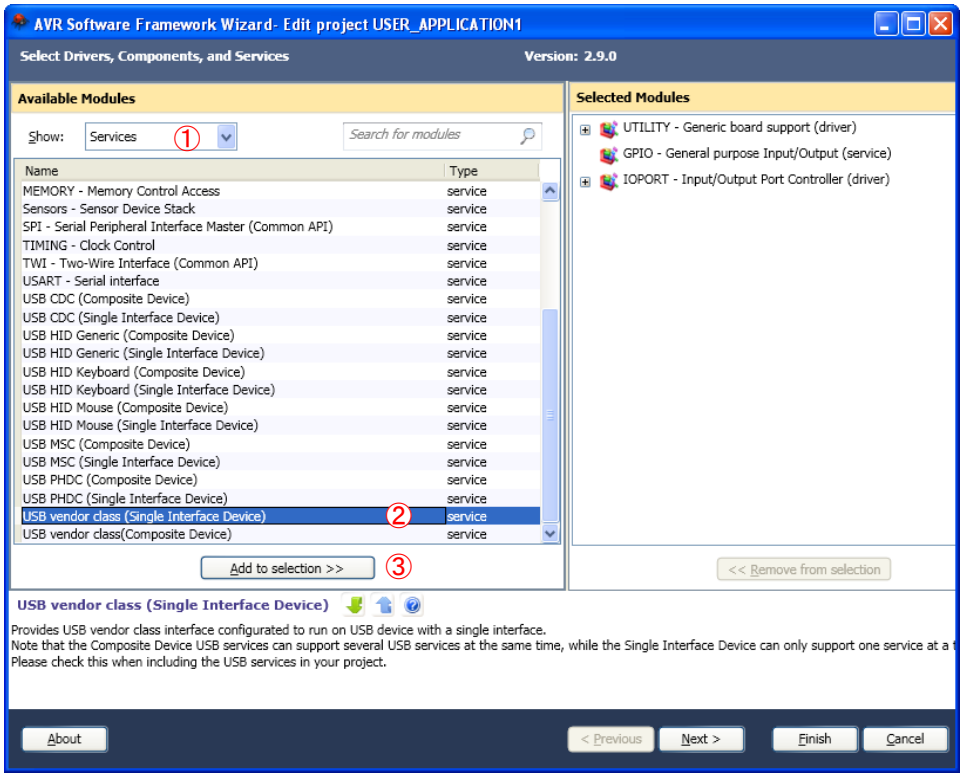
7.1. USB単位部のインポート

USB供給者単位部をインポートするには、下の指示に従ってください。

1. あなたのプロジェクトを開く、または作成してください。
2. Projectメニュから、"Select Drivers from AVR Software Framework"を選択してください。



3. Services(①)を選んで、USB vendor class (Single Interface Device)(②)を選択して"Add to selection"釦(③)をｸﾘｯｸしてください。



7.2. USB形態設定

全てのUSB階層形態設定は応用単位部内のconf.usb.hﾌｧｲﾙに格納されます。これらの形態設定は簡単で、どんな特別なUSBの知識も必要ありません。

これらはUDC,UDI,UDDの各USB単位部に対して1つの形態設定項目があります。

UDC形態設定のありそうなことは「AVR4900:ASF-USB装置階層」応用記述の「7.1.1. USB装置形態設定」で記述されます。

UDD形態設定のありそうなことは「AVR4900:ASF-USB装置階層」応用記述の「7.1.3. USBﾄﾞﾗｲﾊﾞ形態設定」で記述されます。

供給者ｲﾝﾀｰﾌｪｰｽのUDIは表7-1.で記述されるいくつかの形態設定が必要です。

表7-1. UDI供給者-形態設定

定義名	形式	説明
UDI_VENDOR_ENABLE_EXT	呼び戻し関数	供給者ｲﾝﾀｰﾌｪｰｽ許可時に呼ばれる呼び戻し関数
UDI_VENDOR_DISABLE_EXT	呼び戻し関数	供給者ｲﾝﾀｰﾌｪｰｽ禁止時に呼ばれる呼び戻し関数
UDI_VENDOR_SETUP_OUT_RECEIVED	呼び戻し関数	OUT構成設定(SETUP)要求受信時に呼ばれる呼び戻し関数
UDI_VENDOR_SETUP_IN_RECEIVED	呼び戻し関数	IN構成設定(SETUP)要求受信時に呼ばれる呼び戻し関数
UDI_VENDOR_EPS_SIZE_INT_FS UDI_VENDOR_EPS_SIZE_BULK_FS UDI_VENDOR_EPS_SIZE_ISO_FS	定数	全速(Full-Speed)に対する割り込みｴﾝﾄﾞﾎﾟｲﾝﾄ容量(～64) 全速に対する大量(Bulk)ｴﾝﾄﾞﾎﾟｲﾝﾄ容量(8,16,32,64) 全速に対する等時(Isochronous)ｴﾝﾄﾞﾎﾟｲﾝﾄ容量(～1023)
UDI_VENDOR_EPS_SIZE_INT_HS UDI_VENDOR_EPS_SIZE_BULK_HS UDI_VENDOR_EPS_SIZE_ISO_HS	定数	高速(High-Speed)に対する割り込みｴﾝﾄﾞﾎﾟｲﾝﾄ容量(～1024) 高速に対する大量(Bulk)ｴﾝﾄﾞﾎﾟｲﾝﾄ容量(8,16,32,～,512) 高速に対する等時(Isochronous)ｴﾝﾄﾞﾎﾟｲﾝﾄ容量(～1024)

注: USBﾊｰﾄﾞｳｪｱは特定のｸﾛｯｸ周波数が必要なので、conf.clock.hﾌｧｲﾙで定義される形態設定を確認することが重要です(conf.clock.hﾌｧｲﾙ内の注釈をご覧ください)。

7.3. USB実装

この項はUSB装置供給者応用を走らせるため追加するｿｰｽ ｺｰﾄﾞを記述します。

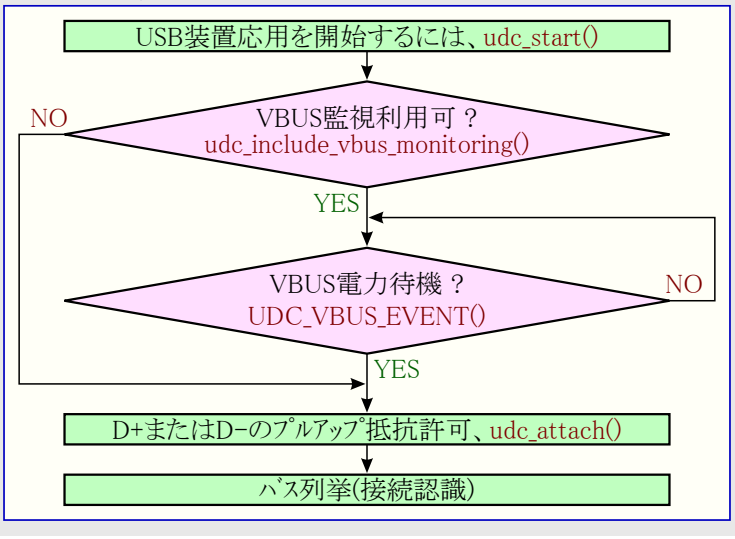
実装は以下の3つの段階から成ります。

1. USB装置を開始します。
2. ﾎｽﾄによる供給者ｲﾝﾀｰﾌｪｰｽの許可を待ちます。
3. USBﾊﾞｽ上でﾃﾞｰﾀを転送します。

7.3.1. USB装置制御

USB装置応用を開始するには2つの関数呼び出しだけが必要とされます(**図7-1.**をご覧ください)。

図7-1. USB装置応用手順



注: 新規ﾌﾟﾛｼﾞｪｸﾄの場合、USB階層は割り込みを許可してｸﾛｯｸと休止管理部(sleepmgr)のｻｰﾋﾞｽを初期化することが必要です。

例:

```
<conf_usb.h>
#define UDC_VBUS_EVENT(b_vbus_high) ¥
          vbus_event(b_vbus_high)

<主Cﾌｧｲﾙ>:
main() {
     // 割り込み認可
     irq_initialize_vectors();
     cpu_irq_enable();
     // 休止管理部ｻｰﾋﾞｽ初期化
     sleepmgr_init();
     // ｸﾛｯｸ ｻｰﾋﾞｽ初期化
     sysclk_init();
     // USB階層装置許可
     udc_start();
     if (!udc_include_vbus_monitoring()) {
          // この製品でVBUS監視が利用不可
          // これによってVBUSは存在するものとして考慮されなれなければなりません。
          vbus_event (true);
     }
}

vbus_event(b_vbus_high) {
     if (b_vbus_high) {
          // USB装置接続
          udc_attach();
     }else{
          // USB装置切断
          udc_detach();
     }
}
```

7.3.2. USBインターフェース制御

装置列挙(USB装置の検出と識別)後、USBホストは装置形態設定を開始します。USB供給者クラス インターフェースが受け入れられると、USBホストはこのインターフェースを許可してUDI_VENDOR_ENABLE_EXT()呼び戻し関数が呼ばれます。

USB装置が切断、またはUSBホストによってリセットされると、USBインターフェースは禁止され、UDI_VENDOR_DISABLE_EXT()呼び戻し関数が呼ばれます。

例:

```
<conf_usb.h>
#define UDI_VENDOR_ENABLE_EXT() ¥
        vendor_enable()
#define UDI_VENDOR_DISABLE_EXT() ¥
        vendor_disable()

<使用者Cファイル>:
vendor_enable() {
    // A/D変換開始
    ～
    return true;
}
vendor_disable() {
    // A/D変換停止
    ～
}
```

7.3.3. USB供給者関数

表7-2.で記述されるUSB供給者クラス関数は応用にデータの送出または受け取りを許します。

表7-2. UDI供給者クラスーデータ関数

宣言	説明
udi_vendor_interrupt_in_run()	IN割り込み(Interrupt)エンドポイントで転送開始
udi_vendor_interrupt_out_run()	OUT割り込みエンドポイントで転送開始
udi_vendor_bulk_in_run()	IN大量(Bulk)エンドポイントで転送開始
udi_vendor_bulk_out_run()	OUT大量エンドポイントで転送開始
udi_vendor_iso_in_run()	IN等時(Isochronous)エンドポイントで転送開始
udi_vendor_iso_out_run()	OUT等時エンドポイントで転送開始

7.4. 供給者クラスとUSBホスト支援

USB供給者クラスが使用者によって指定されるため、どのオペレーティング システムでも本来での支援はありません。使用者は自身のドライバとホスト応用を構築しなければなりません。けれども最速の解決策を構築するのを手助けするためにいくつかの解決策が存在します。最も一般的で無料なものは次の通りです。

- Libusb：この解決策は複数オペレーティング システム基盤(Windows、Linux®、Mac OS®)を支援し、開放ソース解決策を提供する優位点を持ちます。ATMELはATMEL AVR4901応用記述とで利用可能なこの解決策を使用する例を提供します。libusbに関する更なる情報については http://www.libusb.org/ を参照してください。
- WinUSB：この解決策はWindows基盤専用で(Windows Vista®から固有でそれによって)提供されます。WinUSBは等時(Isochronous)転送動作を支援しません。更なる情報については http://msdn.microsoft.com/en-us/library/windows/hardware/ff540196(v=vs.85).aspx を参照してください。

8. USB複合装置内の供給者クラス

複合装置を構築するのに必要とする情報はATMELの「AVR4902:ASF-USB複合装置」応用記述で入手可能です。この応用記述の熟知は必須です。

本章は供給者クラス インターフェースを持つ複合装置の構築に必要とする特定情報だけを紹介します。

8.1. USB形態設定

7.2.項で記述されたUSB形態設定に加えて、以下の値がconf_usb.hﾌｧｲﾙで定義されなければなりません。

- USB_DEVICE_EP_CTRL_SIZE：ｴﾝﾄﾞﾎﾟｲﾝﾄ制御容量、これは以下でなければなりません。
  - ○ 全速(Full Speed)装置に対して8,16,32または64 (RAMを節約するために8が推奨されます。)
  - ○ 高速(High Speed)装置に対して64
- UDI_VENDOR_EP_INTERRUPT_IN：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるIN割り込みｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_EP_INTERRUPT_OUT：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるOUT割り込みｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_EP_BULK_IN：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるIN大量(Bulk)ｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_EP_BULK_OUT：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるOUT大量ｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_EP_ISO_IN：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるIN等時(Isochronous)ｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_EP_ISO_OUT：供給者ｲﾝﾀｰﾌｪｰｽによって使用されるOUT等時ｴﾝﾄﾞﾎﾟｲﾝﾄ番号(UDI_VENDOR_EPS_SIZE_INT_FSが0の場合に無視)。
- UDI_VENDOR_IFACE_NUMBER：供給者ｲﾝﾀｰﾌｪｰｽのｲﾝﾀｰﾌｪｰｽ番号。
- USB_DEVICE_MAX_EP：応用内の総ｴﾝﾄﾞﾎﾟｲﾝﾄ数。これは供給者ｲﾝﾀｰﾌｪｰｽによって使用されるｴﾝﾄﾞﾎﾟｲﾝﾄ数を含まなければなりません(UDI_VENDOR_EPS_SIZE_X定義に依存して0～6)。

8.2. USB記述子

conf_usb.hﾌｧｲﾙで定義される複合装置のUSB装置記述子は供給者ｲﾝﾀｰﾌｪｰｽを含めなければなりません。

```
//! 複合ｲﾝﾀｰﾌｪｰｽ記述子の構造体定義
#define UDI_COMPOSITE_DESC_T                ¥
    udi_vendor_desc_t udi_vendor;           ¥
    ～

//! 全速(Full Speed)用複合ｲﾝﾀｰﾌｪｰｽ記述子を満たす。
#define UDI_COMPOSITE_DESC_FS                          ¥
    .udi_vendor             = UDI_VENDOR_DESC_FS,      ¥
    ～

//! 高速(High Speed)用複合ｲﾝﾀｰﾌｪｰｽ記述子を満たす。
#define UDI_COMPOSITE_DESC_HS                          ¥
    .udi_vendor             = UDI_VENDOR_DESC_HS,      ¥
    ～

//! ｲﾝﾀｰﾌｪｰｽ記述子に対応するｲﾝﾀｰﾌｪｰｽAPIを満たす。
#define UDI_COMPOSITE_API    ¥
    &udi_api_vendor,         ¥
    ～
```

9. 目次

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
新大崎勧業ﾋﾞﾙ 16F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

本応用記述はATMELのAVR4901応用記述(doc8481.pdf Rev.8481A-01/12)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訳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。