

AVR4907 : ASF – USB装置CDC応用

要点

- USB 2.0適合
 - ・ USB第9章保証
 - ・ 全速(Full,12Mビット/秒)、高速(High,480Mビット/秒)のデータ速度
- 基本USB CDC実装
 - ・ 送受信の流れ
 - ・ 8または9ビットのデータ
- 利用可能な制御
 - ・ DCD信号
 - ・ DSR信号
 - ・ フレーミング異常
 - ・ パリティ誤り
 - ・ オーバーラン
- 遠隔起動支援
- USBバス給電支援
- 実時間(OS適合、割り込み駆動)
- 8ビットと32ビットのAVR[®]を支援

1. 序説

この資料の狙いは新規または既存プロジェクトでUSB CDC装置応用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)
- ・ 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)
- ・ SOF : フレーム開始(Start of frame)



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. 8447A-11/11, 8447AJ1-03/14

3. 概要

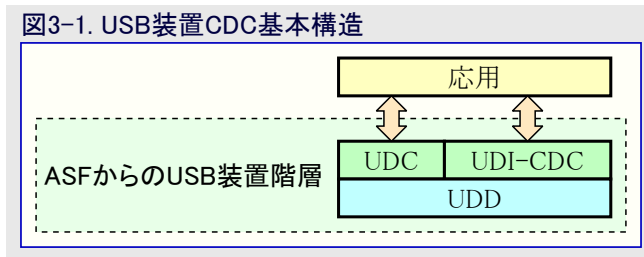
この資料はUSB装置CDC応用を構築する時の必要条件の全ての形式に対する以下の5つの部分を含みます。

- **即時開始**
使用準備済みのCDC装置例を開始する方法を記述します。
- **例の説明**
CDC装置例を説明します。
- **USB装置CDCの構築**
プロジェクトにUSB装置CDCインターフェースを追加する方法を記述します。
- **標準入出力で接続されるUSB装置CDCの構築**
標準入出力で接続されるUSB装置CDCインターフェースを追加する方法を記述します。
標準入出力：`stdio`ライブラリ、例：`printf()`
- **USB複合装置でのCDC**
複合装置プロジェクトにUSB装置CDC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CDC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CDC (UDI-CDC)
- USB装置制御部 (UDC)
- USB装置ドライバ (UDD)

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



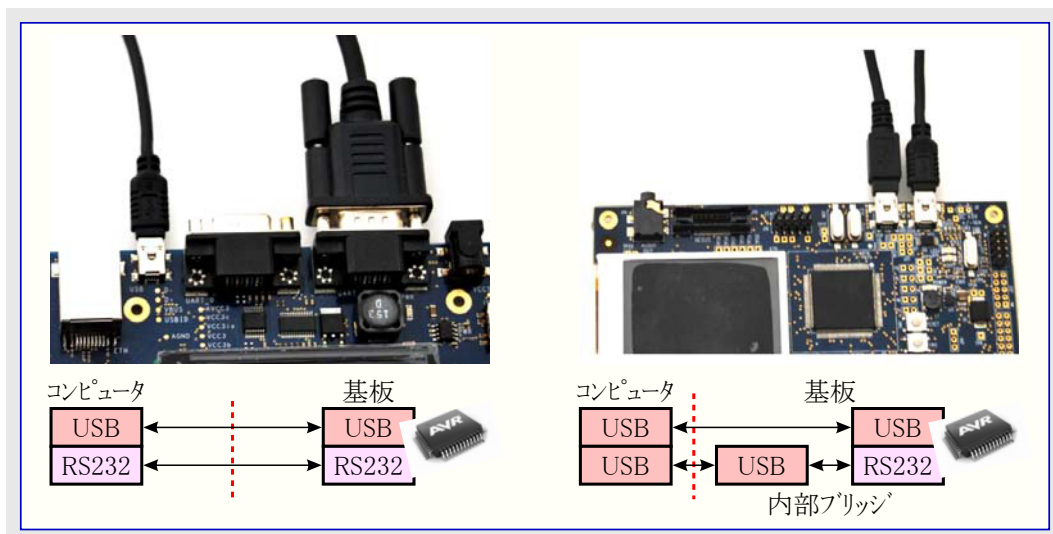
注: USB装置階層は`common¥services¥usb`ディレクトリに於いてASFで利用可能です。

4. 即時開始

USB装置CDC例はATMEL® AVR Studio® 5とASFで利用可能です。それはUARTとUSBインターフェース間のブリッジです。これはAVR製品上の真のUARTへ読み書きアクセスをするUSBホスト上の“COM”通信ポートを提供します。

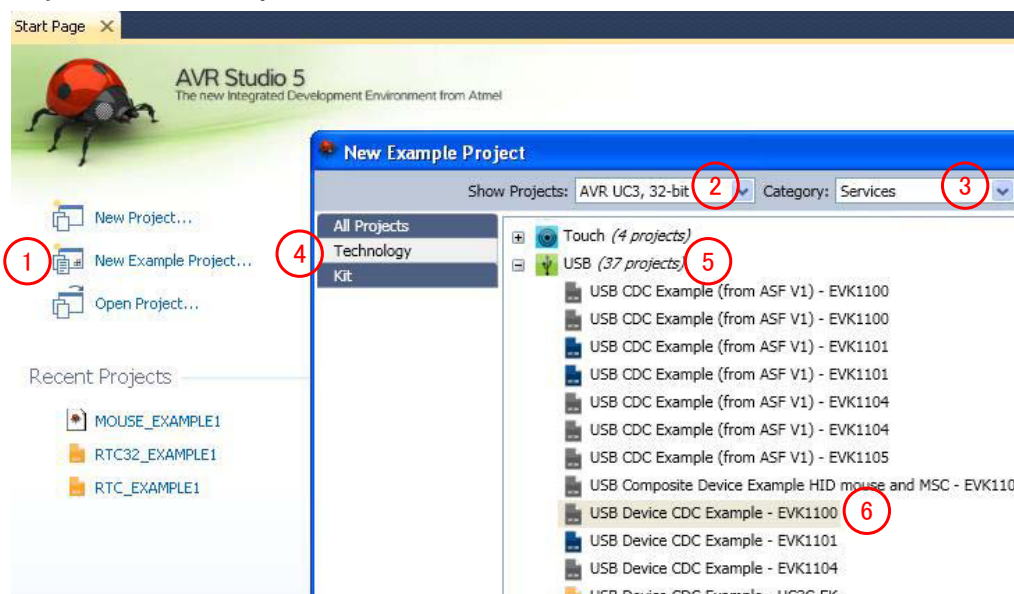
1. 基板通電と、USBとUARTの接続

基板とUSBホスト間にUSBケーブルを接続してください。このケーブルが基板に給電します。RS232のUART接続はDB9ケーブル、または基板が内部UARTブリッジを含む場合に別のUSBケーブルによって行うことができます。



2. AVR Studio 5は新しい例プロジェクト(New Example Project)の作成を許します。

例一覧に於いて、使用するATMEL基板に対応するUSB装置CDC例(USB Device CDC Example)を選択してください。例を素早く見つけるのに濾過一覧を用いてください。



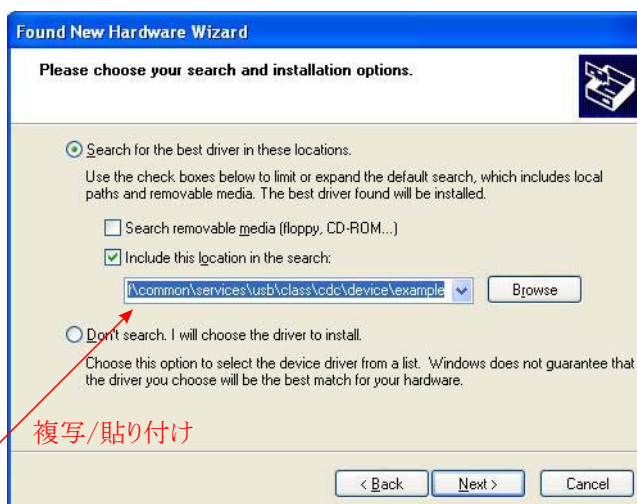
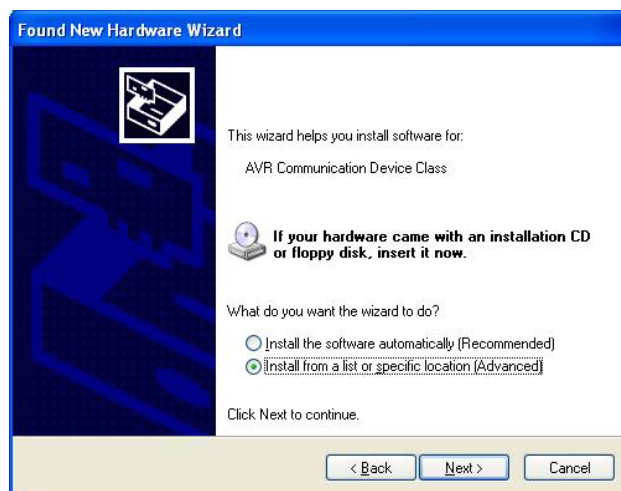
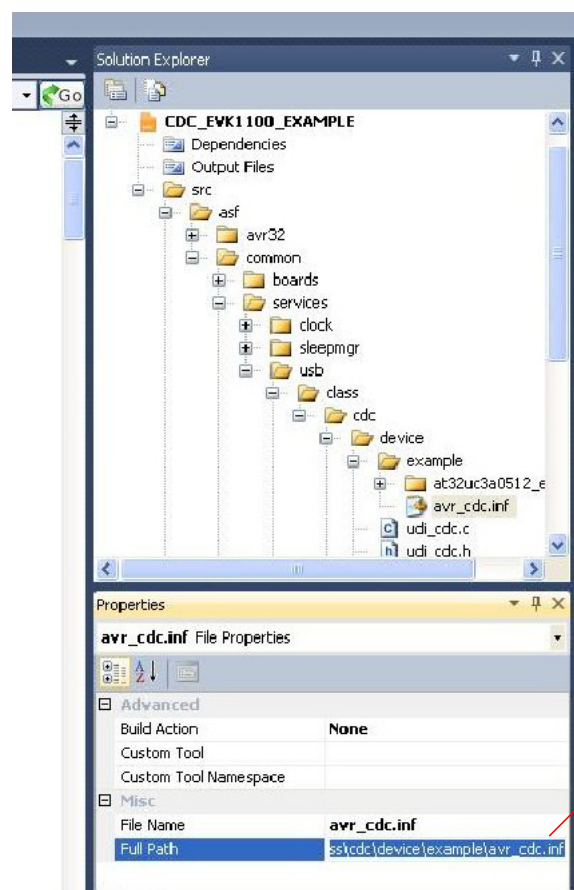
3. コンパイル、書き込み設定、実行

プロジェクトはどんな変更も必要なく、コンパイル、書き込み設定、走行だけが必要です。基板によって支援されるATMELのデバuggを接続してF5を押してください。

4. Windows®上での新しいCDC装置のインストール

初回使用の間にドライバをインストールするため、ハードウェアウィザードが直ぐに出現されます。高度な操作(Advance)形態が選択されなければなりません。

Windowsでのインストール中にexampleフォルダで利用可能なavr_cdc_infファイルが選択されなければなりません。



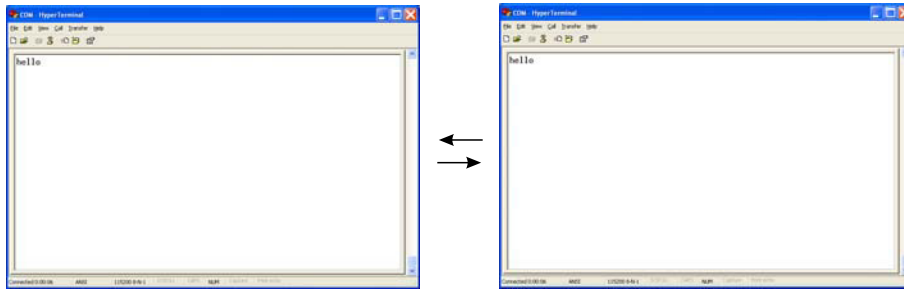
複写/貼り付け

5. 端末を開く

UARTとUSB CDCの両ポートに対してRS232端末が開かなければなりません。ポート形態設定(速度、データの大きさ、パリティ、停止ビット)は両方で同じでなければなりません。全てのAVRデバイスによってパリティなし115200ボーが支援されます。

6. データ送付

さて、端末でキーを入力してください。そのキーは他の端末で表示します。

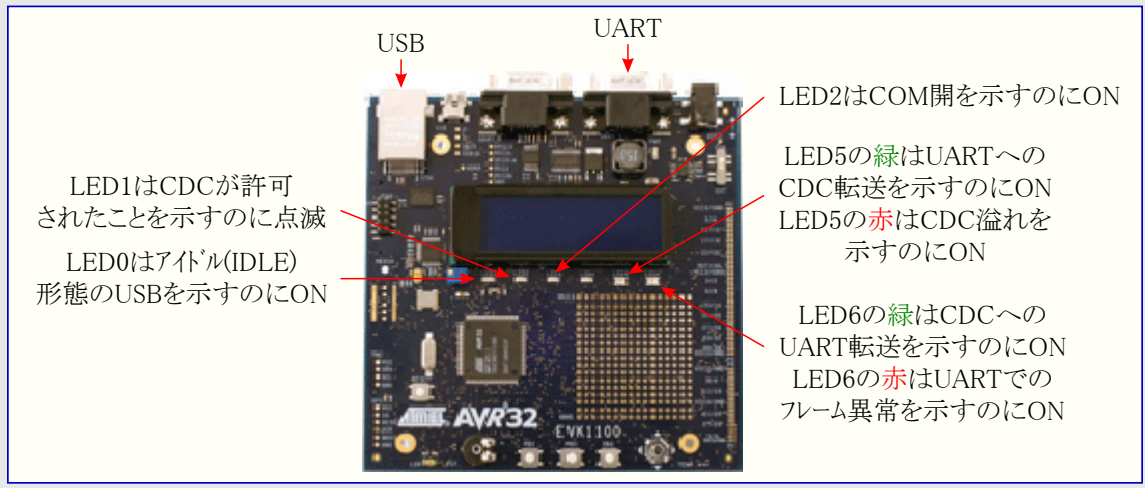


7. 流れ制御

基板のユーザーインターフェースは以下の情報を提供することができます。

- LEDはUSB装置がアイドル(IDLE)動作形態の時にON、休止(SUSPEND)動作形態の時にOFFです。
- LEDはUSBホストによってCDCインターフェースが許可されていることを示すのに点滅します。
- LEDはUSBホストによってCOMが開かれている時にONです。
- LEDはデータ転送中にONです。
- LEDはUARTでのフレーム異常を意味します。
- LEDはCDC溢れを意味します。

図4-1. ATMEL EVK1100基板での例



5. 例の説明

5.1. 例の内容

ASFは様々なATMEL AVR製品に対してUSB装置CDC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、UARTブリッジを実装します。

表5-1はUSB装置CDC応用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5-1. USB装置CDC例のファイル

単位部	ファイル	ASFパス	説明
応用	main.c ui.c uart_uc3.c uart_xmega.c conf_usb.h	Exampleフォルダ*	主繰り返し 操作を見せるためのハードウェア スイッチとLEDの構成設定 UC3 UART制御とCDCインターフェースへの接続 XMEGA® UART制御とCDCインターフェースへの接続 USB装置形態設定
UDI CDC	udi_cdc.c/h	common¥services¥usb¥class¥cdc¥device¥	CDCクラス実装
	udi_cdc_desc.c udi_cdc_conf.h	common¥services¥usb¥class¥cdc¥device¥	CDC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)
	usb_protocol_cdc.h	common¥services¥usb¥class¥cdc	CDC規約定数
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*

5.2. 例の性能

ATMEL AVR UC3でのUARTブリッジ動作形態は115200ボーレートを容易に支援することができます。例えば、48MHzのCPU周波数でアイドル時間が86%です。

ATMEL AVR XMEGAでのUARTブリッジ動作形態は115200ボーレートを支援することができます。例えば、24MHzのCPU周波数でアイドル時間が65%です。

ATMELの「AVR4920:ASF – USB装置階層 – 適合と性能係数」応用記述が利用可能で、UARTブリッジ動作形態以外のCDC性能に関するより多くの情報を提供します。

5.3. 例の動き

main.c, ui.c, uart_xmega.cとuart_uc3.cのファイルがCDC応用の使用者インターフェースとUART制御を実装します。これは以下のように5つの段階から成ります。

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. 呼び戻し経路でCDCインターフェースの許可を待ちます。

```
UDI_CDC_ENABLE_EXT() // 認可とUART通信ポートを開きます。
```

3. 通信ポートの新しい形態設定を設定します。

```
UDI_CDC_SET_CODING_EXT(cfg) // UART通信ポートの形態設定
```

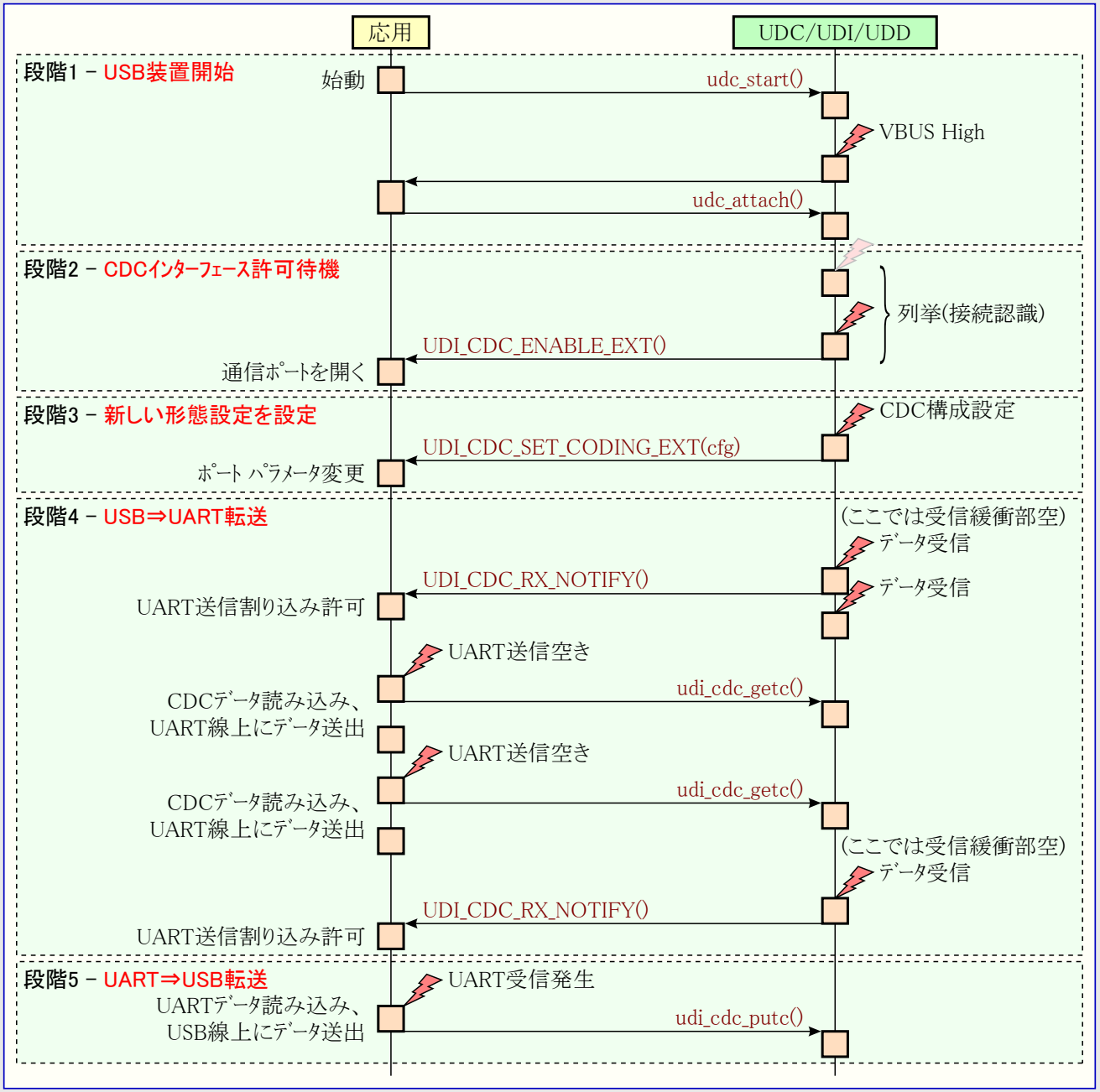
4. USBバスからのデータ転送

```
UDI_CDC_RX_NOTIFY() // CDC受信緩衝部が空でないのを通知、その後にUART送信が許可されます。
udi_cdc_getc() // CDC受信緩衝部読み込みルーチン
               // これはUART送信割り込みルーチンで呼ぶことができる短い関数です。
```

5. USBバスへデータ転送

```
udi_cdc_putc() // CDC送信緩衝部にデータを置くルーチン
               // これはUART受信割り込みルーチンで呼ぶことができる短い関数です。
```

図5-1. 例の動きの流れ



6. USB装置CDCの構築

USB装置CDC単位部はどのインターフェースにも接続することができるUSB CDCインターフェースを提供します。

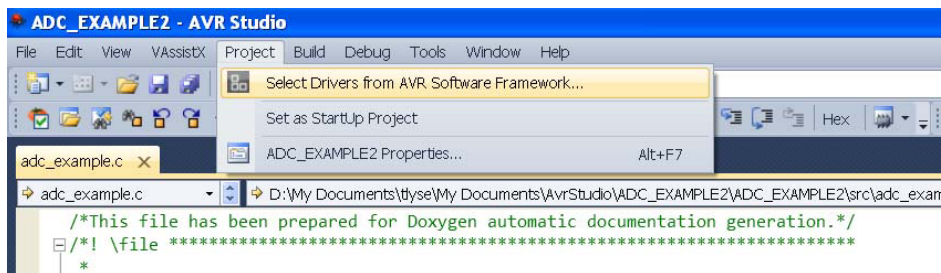
これらの単位部はATMEL AVR Studio 5で利用可能で、AVR Studio 5プロジェクトに持ち込む(インポートする)ことができます。この項はプロジェクトにUSB装置CDCを追加する方法を記述します。

1. USB CDC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。

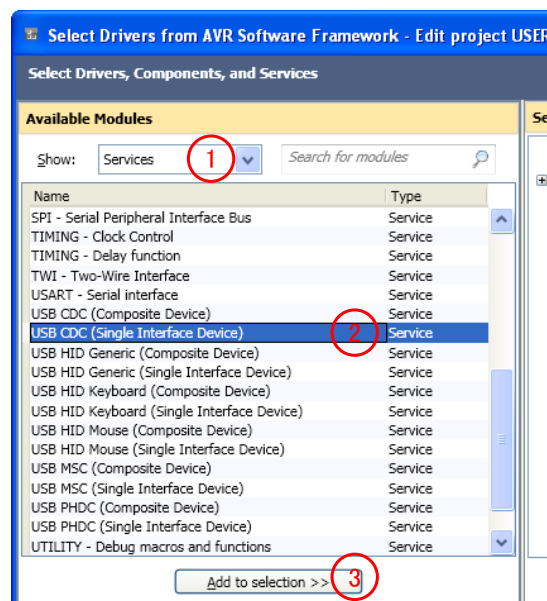
6.1. USB単位部のインポート

USB CDC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メニューから、“Select Drivers from AVR Software Framework”を選択してください。



3. Servicesを(1)選んで、USB CDC (Single Interface Device)(2)を選択して“Add to selection”鈕(3)を押してください。



6.2. 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。

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ATMEL「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ATMEL「AVR4900:ASF-USB装置階層」応用記述の「7.1.3. USBドライバ形態設定」で記述されます。

CDCインターフェースは表6-1.で記述されるいくつかの形態設定が必要です。

表6-1. UDI CDC - 形態設定

定義名	型	説明
UDI_CDC_LOW_RATE	動作形態許可のための(値なし)定義のみ	CDC装置からホストへ転送する時にCDC緩衝部容量を減らすために低速であることを定義
UDI_CDC_DEFAULT_RATE	語(ワード)32ビット	CDC始動での既定通信速度
UDI_CDC_DEFAULT_STOPBITS	CDC_STOP_BITS_1 CDC_STOP_BITS_1_5 CDC_STOP_BITS_2	CDC始動での既定停止ビット形態設定
UDI_CDC_DEFAULT_PARITY	CDC_PAR_NONE CDC_PAR_ODD CDC_PAR_EVEN CDC_PAR_MARK CDC_PAR_SPACE	CDC始動での既定パリティ形態設定
UDI_CDC_DEFAULT_DATABITS	8または9	CDC始動での既定データビット形態設定

注: USBハード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た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)。

6.3. USB実装

この項は追加してUSB装置CDC応用を走らせるためのソースコードを記述します。

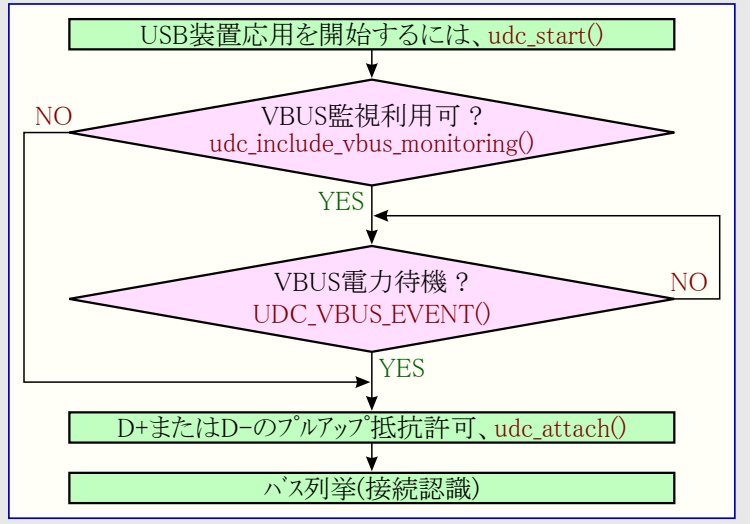
実装は以下の3つの段階から成ります。

1. USB装置を開始します。
2. ホストによってCDCインターフェースが許可されるのを待ちます。
3. USBバス上でデータを転送します。

6.3.1. 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6-1をご覧ください)。

図6-1. USB装置応用手順



注: 新規プロジェクトの場合、USB階層は割り込みを許可してクロックと休止管理部(sleepmgr)のサービ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();
    // 休止管理部サービス初期化
    sleepmgr_init();
    // クロック サービス初期化
    sysclk_init();
    // USB階層装置許可
    udc_start();
    if (!udc_include_vbus_monitoring()) {
        // この製品でVBUS監視が利用不可なら、それによってVBUSは存在すると見做さ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();
    }
}
  
```

6.3.2. 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。装置からのUSB CDCインターフェースがホストによって受け入れられると、USBホストはこのインターフェースを許可してUDI_CDC_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_CDC_DISABLE_EXT()呼び戻し関数が呼ばれます。

従って、これらの関数で通信ポートを許可/禁止することが推奨されます。

例:

```
<conf_usb.h>
#define UDI_CDC_ENABLE_EXT() ¥
    cdc_enable()
#define UDI_CDC_DISABLE_EXT() ¥
    cdc_disable()

<主Cファイル>:
cdc_enable() {
    // UARTを開き、UART送信割り込みを許可
    ~
    return true;
}
cdc_disable() {
    // UART割り込みを禁止してUARTを閉じる
    ~
}
```

6.3.3. USB CDC制御

表6-2.で記述されるUSB CDC関数は通知送出手を応用に許します。表6-3.の関数はデータの送信または受信を許します。

表6-2. UDI CDC - 関数

宣言	説明
udi_cdc_ctrl_signal_dcd(bool b_set)	DCD信号の状態変化に関する通知を送信
udi_cdc_ctrl_signal_dsr(bool b_set)	DSR信号の状態変化に関する通知を送信
udi_cdc_signal_framing_error()	フレーミング異常に関する通知を送信
udi_cdc_signal_parity_error()	パリティ誤りに関する通知を送信
udi_cdc_signal_overrun()	オーバーランに関する通知を送信

表6-3. UDI CDC - データ関数

宣言	説明
udi_cdc_is_rx_ready()	この関数はCDCでキャラクタが受信されたかを調べます。
int udi_cdc_getc()	CDCで値を待って取得します。
iram_size_t udi_cdc_read_buf(int* buf, iram_size_t size)	CDCで待機してRAM緩衝部を読みます。
udi_cdc_is_tx_ready()	この関数は新しいキャラクタの送信が可能かを調べます。
int udi_cdc_putc(int value)	CDCでバイトを出力します。
iram_size_t udi_cdc_write_buf(const int* buf, iram_size_t size)	CDCでRAM緩衝部に書きます。

例:

```

<主Cファイル>:
cdc_enable() {
    // UARTを開き、UART送信割り込みを許可
    ~
    return true;
<UART Cファイル>:
uart_process() {
    if (is_uart_framing_error())
        udi_cdc_signal_framing_error();
    if (is_data_reception()) {
        value = uart_getc();
        udi_cdc_putc(value);
    }
    if (udi_cdc_is_tx_ready())
        value = udi_cdc_getc();
        uart_putc(value);
    }
    ~
}

```

6.4. Windows OS用infファイル作成

Windows OSは新しいCDC装置をインストールするのに*.infファイルが必要です。
これは完了するためのinfファイルの雛形です(<>間の注釈をご覧ください)。

```

; Windows 2000, XP & Vista setup File for AVR CDC Device
[Version]
Signature="$Windows NT$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}
Provider=%ATMEL% <あなたの製造者名でATMELを置き換えてください。>
DriverVer=10/15/1999, 5.0.2153.1

;-----
; 目的対象
;-----

[Manufacturer]
%ATMEL%=ATMEL, NTamd64 <あなたの製造者名でATMELを置き換えてください。>
[ATMEL]
%ATMEL_CDC%=Reader, USB¥VID_03EB&PID_2404 <あなたのVIDで03EBを、あなたのPIDで2404を置き換えてください。>
[ATMEL.NTamd64]
%ATMEL_CDC%=DriverInstall, USB¥VID_03EB&PID_2404 <あなたのVIDで03EBを、あなたのPIDで2404を置き換えてください。>

[次頁へ続く]

```

```

;-----
; Windows 2K, XP, とVista32
;-----
[Reader_Install.NTx86]
[DestinationDirs]
DefaultDestDir=12
Reader.NT.Copy=12
[Reader.NT]
include=mdmcpq.inf
CopyFiles=Reader.NT.Copy
AddReg=Reader.NT.AddReg
[Reader.NT.Copy]
usbser.sys
[Reader.NT.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,usbser.sys
HKR,,EnumPropPages32,, "MsPorts.dll,SerialPortPropPageProvider"
[Reader.NT.Services]
AddService = usbser, 0x00000002, Service_Inst
[Service_Inst]
DisplayName = %Serial.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 3 ; SERVICE_DEMAND_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %12%\usbser.sys
LoadOrderGroup = Base
;-----
; Vista64
;-----
[DriverInstall.NTamd64]
include=mdmcpq.inf
CopyFiles=DriverCopyFiles.NTamd64
AddReg=DriverInstall.NTamd64.AddReg
[DriverCopyFiles.NTamd64]
usbser.sys,, 0x20
[DriverInstall.NTamd64.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,usbser.sys
HKR,,EnumPropPages32,, "MsPorts.dll,SerialPortPropPageProvider"
[DriverInstall.NTamd64.Services]
AddService=usbser, 0x00000002, DriverService.NTamd64
[DriverService.NTamd64]
DisplayName=%Serial.SvcDesc%
ServiceType=1
StartType=3
ErrorControl=1
ServiceBinary=%12%\usbser.sys

;-----
; 文字列
;-----
[Strings]
ATMEL = "ATMEL, Inc." <あなたの製造者名でATMELを置き換えてください。>
ATMEL_CDC = "AVR Communication Device Class" <あなたの装置記述を書いてください。>
Serial.SvcDesc = "USB Serial emulation driver"

```

注: この*.infファイルの例は以下のCDC例で提供されます。

`¥common¥services¥usb¥class¥cdc¥device¥example¥avr_cdc.inf`



7. 標準入出力へ接続されるUSB装置CDCの構築

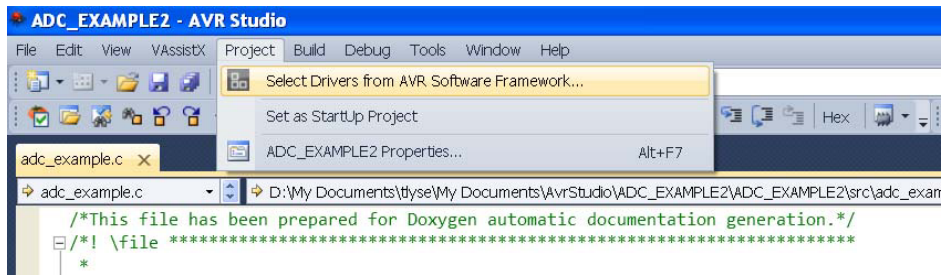
USB/CDC標準I/O単位部は標準I/Oライブラリ(例えば、`printf()`)とUSB CDCインターフェース間の接続の素早い追加を提供します。この単位部はATMEL AVR Studio 5で利用可能で、AVR Studio 5プロジェクトに持ち込む(インポートする)ことができます。この項はプロジェクトにUSB装置CDCを追加する方法を記述します。

1. USB/CDC標準I/O(stdio)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。
4. Windows OS用のinfファイルを作成してください(Linux[®]では不要です)。

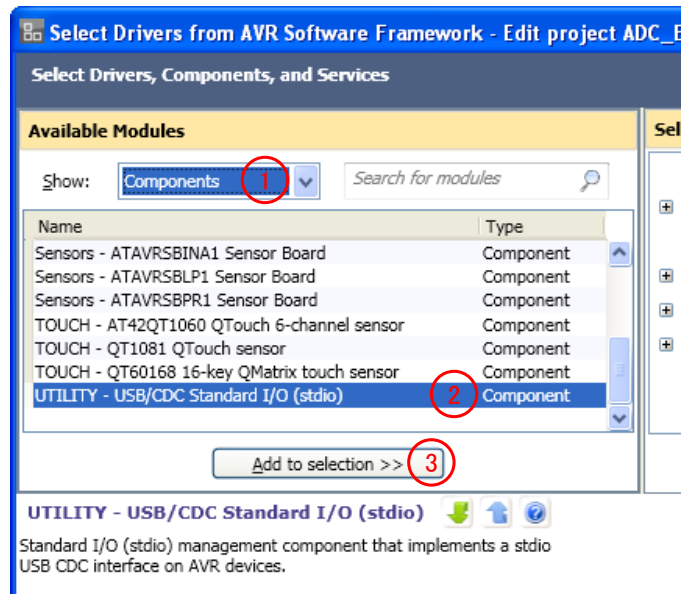
7.1. USB単位部のインポート

USB/CDC標準I/O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**メニューから、“**Select Drivers from AVR Software Framework**”を選択してください。



3. **Services**を(1)選んで、“**UTILITY - USB/CDC Standard I/O (stdio)**”(2)を選択して“**Add to selection**”鈕(3)をクリックしてください。



7.2. USB形態設定

表6-2で記述される指示に従ってください。

7.3. USB実装

USB/CDC標準I/O単位部で既に実装されている6.3.2項と6.3.3項を除き、6.3項で記述される指示に従ってください。

7.4. Windows OS用infファイル作成

6.4項で記述される指示に従ってください。

8. USB複合装置でのCDC

複合装置の構築に必要な情報はATMELの「AVR4902:ASF-USB複合装置」応用記述で利用可能です。この応用記述の熟知が必須です。

本章はCDCインターフェースで複合装置を構築するのに必要な特定情報だけを紹介しました。

8.1. USB形態設定

6.2.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

エンドポイント制御容量

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

USB_CDC_DATA_EP_IN

データ送信(TX)のためにCDCによって使用されるIN大量(バルク)エンドポイント番号。

USB_CDC_DATA_EP_OUT

データ受信(RX)のためにCDCによって使用されるOUT大量(バルク)エンドポイント番号。

USB_CDC_COMM_EP

通信事象送信のためにCDCによって使用されるIN割り込みエンドポイント番号。

USB_CDC_COMM_IFACE_NUMBER

CDC通信インターフェースのインターフェース番号。

USB_CDC_DATA_IFACE_NUMBER

CDCデータ インターフェースのインターフェース番号。

USB_DEVICE_MAX_EP

応用での総エンドポイント数。これはCDCインターフェース用の3つのエンドポイントを含めなければなりません。

8.2. USB記述子

装置記述子でのCDCインターフェース追加はCDC通信インターフェース記述子をCDCデータ インターフェース記述子と提携するためにインターフェース提携記述子(IAD:Interface Association Descriptor)の追加が必要です。

`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CDC通信インターフェース記述子、CDCデータ インターフェース記述子、インターフェース提携記述子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          ¥
    usb_iad_desc_t udi_cdc_iad;        ¥
    udi_cdc_comm_desc_t udi_cdc_comm;  ¥
    udi_cdc_data_desc_t udi_cdc_data;  ¥
    ~

//! 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_cdc_iad           = UDI_CDC_IAD_DESC,          ¥
    .udi_cdc_comm         = UDI_CDC_COMM_DESC,          ¥
    .udi_cdc_data         = UDI_CDC_DATA_DESC,          ¥
    ~

//! 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_cdc_iad           = UDI_CDC_IAD_DESC,          ¥
    .udi_cdc_comm         = UDI_CDC_COMM_DESC,          ¥
    .udi_cdc_data         = UDI_CDC_DATA_DESC,          ¥
    ~

//! 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_cdc_comm,             ¥
    &udi_api_cdc_data,             ¥
    ~
```

9. 目次

要点	1
1. 序説	1
2. 略語	1
3. 概要	2
4. 即時開始	2
5. 例の説明	5
5.1. 例の内容	5
5.2. 例の性能	5
5.3. 例の動き	5
6. USB装置CDCの構築	6
6.1. USB単位部のインポート	7
6.2. USB形態設定	7
6.3. USB実装	8
6.3.1. USB装置制御	8
6.3.2. USBインターフェース制御	9
6.3.3. USB CDC制御	9
6.4. Windows OS用infファイル作成	10
7. 標準入出力へ接続されるUSB装置CDCの構築	12
7.1. USB単位部のインポート	12
7.2. USB形態設定	12
7.3. USB実装	12
7.4. Windows OS用infファイル作成	12
8. USB複合装置でのCDC	13
8.1. USB形態設定	13
8.2. USB記述子	13
9. 目次	14



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、AVR[®]、AVR Studio[®]、XMEGA[®]、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。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AVR4907応用記述(doc8447.pdf Rev.8447A-11/11)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。