

AVR271 : USBキーボード実演

要点

- Windows®98またはそれ以降)、Linux、Mac OSによる支援
- ドライバのインストールなし
- 単純なテキストメッセージ表示
- キーボードLED支援なし

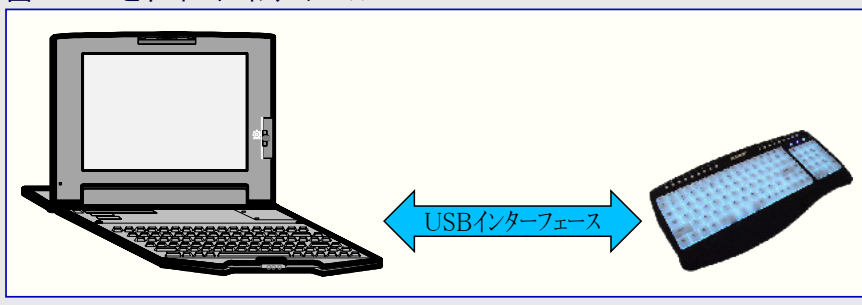
1. 序説

PS/2インターフェースはPCと周辺装置間の標準インターフェースとなるUSBインターフェースに置き換えられつつあり、新しい世代のPCから無くなっています。この変更はキーボード設計者によって追従されなければならず、PCにキーボードを接続するためにUSBインターフェースを統合しなければなりません。

本資料の目的はSTK525スタートキットとFLIP実装書き込みソフトウェアを使ってUSBキーボード実装を開始して実装する方法を記述することです。

(CD-ROMとAtmelのウェブサイトに含まれるdoc7675)AT90USBxxxマイクロコントローラ用USBソフトウェアライブラリとHID仕様(<http://www.usb.org/developers/hidpage>)の熟知が仮定されています。

図1-1. PCとキーボードインターフェース



2. ハードウェアの必要条件

USBキーボード応用は以下のハードウェアが必要です。

- AVR USB評価基板(STK525,AT90USBKey,STK526,～または使用者自身の基板)
- AVR USBマイクロコントローラ
- USBケーブル(標準A-ミニB)
- USB1.1または2.0ホストを持つWindows®(98SE,ME,2000,XP)、Linux®またはMAC OS®で走行するPC

3. 実装書き換えとデバイスファームウェア更新

デバイスをプログラミングするのに以下の方法を用いることができます。

- JTAGICEmk IIを用いるJTAGインターフェース
- AVRISPmk IIを用いるSPIインターフェース
- 工場DFUブートローダとFlipソフトウェアによるUSBインターフェース
- STK500またはSTK600を用いる並列プログラミング

これらの各種方法を用いてデバイスをプログラミングする方法を見るには、(Atmelのスタートキットを使う場合、)使う基板のハードウェア使用者の手引きを参照してください。

USBドライバのインストールとUSBインターフェースを通じたデバイスのプログラミングの方法を見るには、Flip(注)のヘルプ内容を参照してください。

注: Flipは工場DFUブートローダによって(外部ハードウェアを全く必要とせず)USBインターフェースを通してAVR USBデバイスのプログラミングを使用者に許すため、Atmelによって提供されるソフトウェアです。



8ビット **AVR**[®]
マイクロコントローラ

応用記述

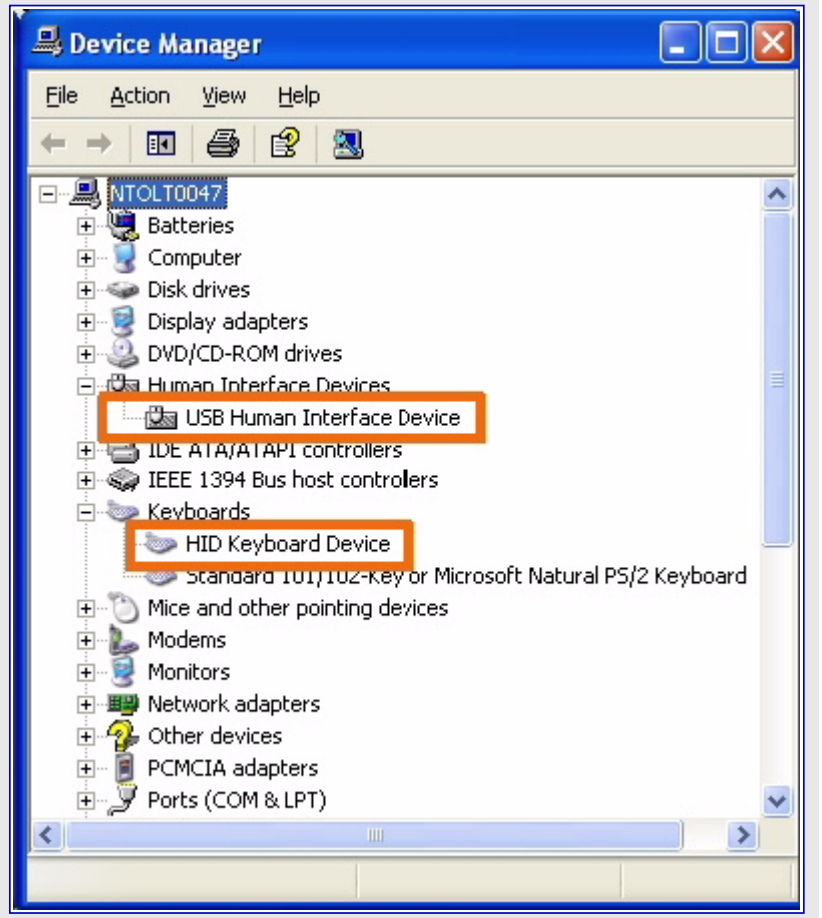
本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 7602B-07/08, 7602BJ2-05/21

4. 即時開始

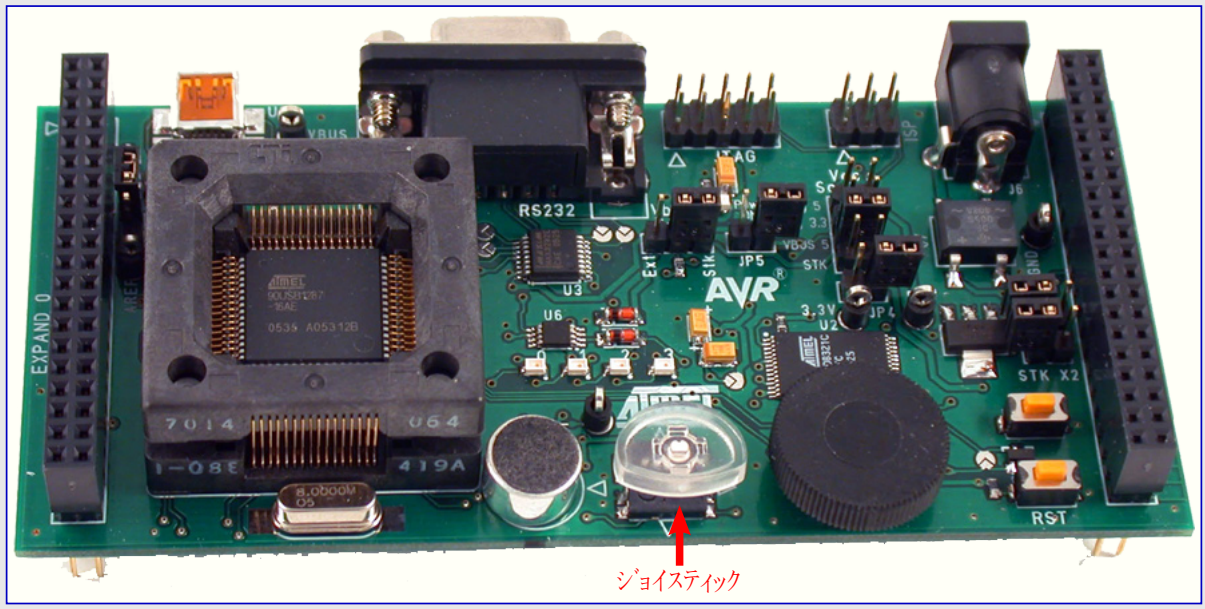
一旦デバイスがusb_keyboard.a90ファイルでプログラムされると、キーボード実演を始められます。デバイスがキーボードとして列挙(認識)されていることを調べてください(図4-1.をご覧ください)。その後にPCへ文字を送るのにキットを使うことができます。

図4-1. キーボード列挙(認識)



下図は実演によって使われるSTK525を示します(動かすAVR USB製品に依存して、使用者はAT90USBKeyやSTK526など、別のキットを使うかもしれません)。

図4-2. STK525キット

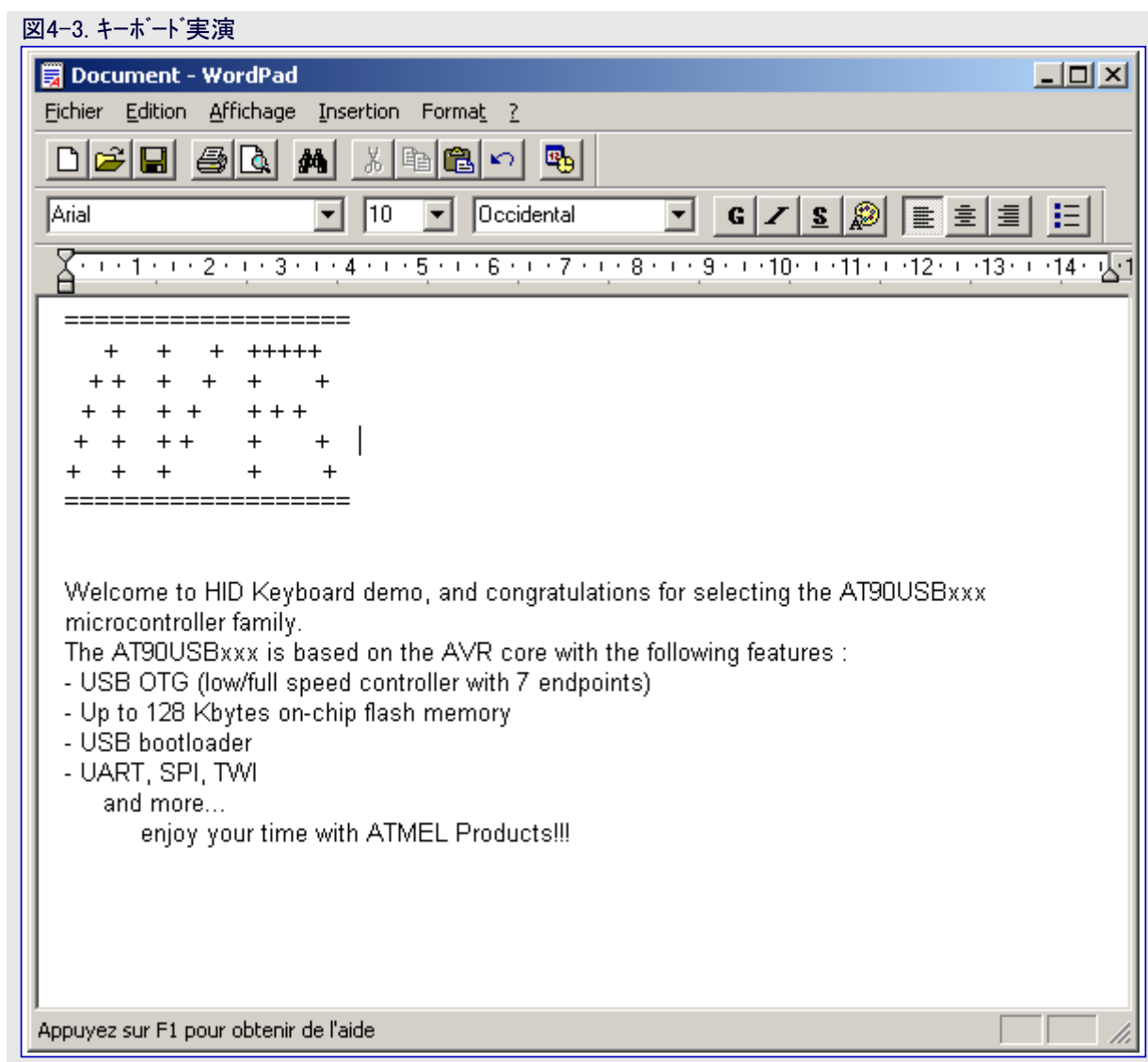


キーボード実演の目的はPCへ文字列を送る事です。

実演を始めるには以下の指示に従ってください。

1. ノートパッド(Notepad)応用またはテキスト エディタを開いてください。
2. キーボードをQWERTY形態(配列)に設定してください(さもないければ、テキスト エディタ上で不正な文字を見るでしょう)。
3. STK525を接続してください。
4. ジョイスティック釦を押してください。

図4-3. キーボード実演



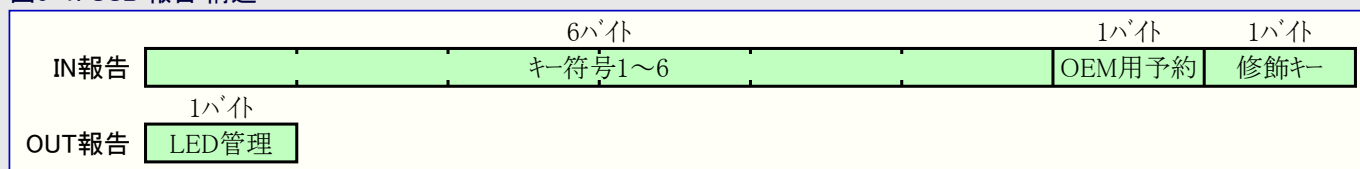
5. 応用概要

USBキーボード応用はPCとキーボード間の簡単なデータ交換です。

PCは各P時間(ポーリング間隔時間)で利用可能な新しいデータがあるかをキーボードに問い、キーボードはそれが利用可能ならばデータを送り、さもなければ利用可能なデータが無いことをPCへ告げるためにNAK(No Acknowledge)を送ります。

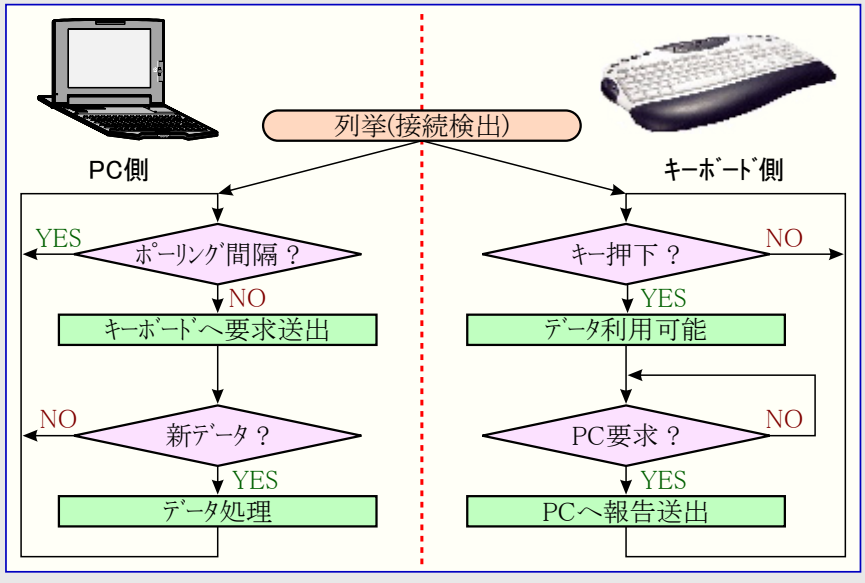
PCとキーボード間で送られるデータは'報告'と呼ばれます。押下キーを含む報告はIN報告(キーボードからPCへ)です。LED(NUM LOCK, CAPS LOCK, SCROLL LOCKなど)状態を含む報告はOUT報告(PCからキーボードへ)です。下図はこれらの報告の構造を示します。

図5-1. USB'報告'構造



注: 本実演はIN報告だけを管理します。

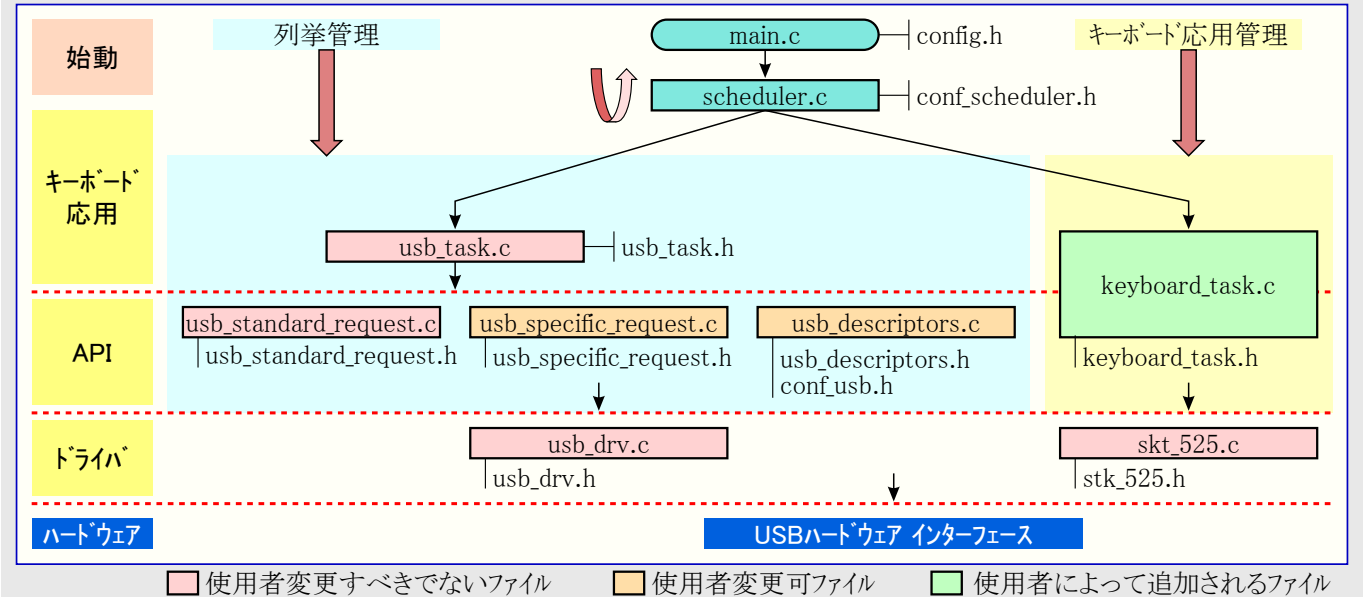
図5-2. 応用概要



6. ファームウェア

AT90USBxxxマイクロ コントローラ用USBソフトウェア ライブラリ資料で説明されるように全てのUSBファームウェア一式は同じ基本設計に基づきます(より多くの詳細については本資料を参照してください)。

図6-1. USBキーボード ファームウェア基本設計

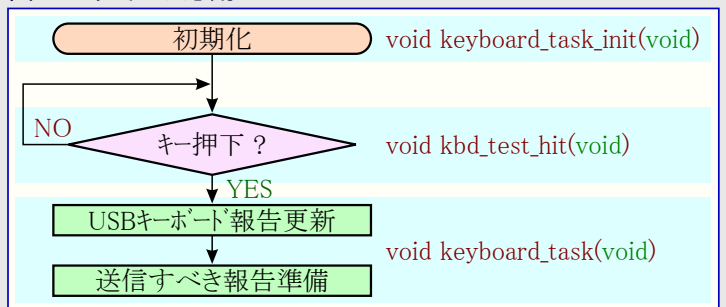


本章はキーボード部だけの専用です。これより後で記述されるファイルの独自設定は使用者に使用者自身のキーボード応用構築を許します。

6.1. keyboard_task.c

このファイルはキーボードとして使われるハードウェアを初期化し、報告データを収集してPCへ送信すべく用意されたエンドポイントFIFOに置く関数を含みます。

図6-2. キーボード応用



6.1.1. keyboard_task_init

この関数はキーボードパラメータとハードウェア資源(ジョイスティックなど)の初期化を実行します。

6.1.2. kbd_test_hit

この関数はキーが押されたかを調べてkey_hit変数に真(true)を設定します。

6.1.3. keyboard_task

この関数はどれかのキーが押された(key_hit=true)かを調べます。その(真の場合、IN報告が関連値で満たされ、ホストへ送信されるべきUSBエンドポイントFIFOに格納されます。

6.2. stk_52x.c

このファイルはSTK52x基板の資源(ジョイスティック、可変抵抗器、温度感知器、LEDなど)を管理するための全ルーチンを含みます。STK52x基板使用時に使用者はこのファイルを変更すべきではありません。そうでなければ自身のハードウェア管理ファイルを構築しなければなりません。

6.3. CAPS, NUMLOCKなどのLEDの管理方法

キーボードのLED(CAPS, NUMLOCKなど)は対応するキーが押された時にホストによって管理されます。CAPSやNUMLOCKなどのキー符号受信時に、ホストはキーボードの関連LEDをON/OFFするためにSet_Report要求(OUT報告)を送出します。

この要求はエンドポイント0(制御転送)を通して送られ、以下で示されるようにSet_Configuration要求として管理されなければなりません。

最初にホストは以下に示すようにset_reportを送ります。

bmRequestType	00100001
bRequest	SET_REPORT (0x09)
wValue	Report Type (0x02) and Report ID 0x00
wIndex	Interface (0x00)
wLength	Report Length (0x0004)
Data	Report (1バイト)

この要求はHIDクラス特有で、これがusb_standard_request.cによってではなく、usb_specific_request.cで管理される理由です。このファイルでの要求はusb_user_read_request()関数を用いて後続するbmRequestとbRequestが復号されます。報告形式(Report Type)の0x02はOUT報告に対応します。この要求を処理するためにusb_user_read_request()はhid_set_report()関数を呼びます。この関数は構成設定(setup)要求を応答し、そしてどのLEDがON/OFF切り替えされなければならないかを知るために、1バイトデータ(wLength項目を用いて量を調べられます)の取得を使用者に許します(LED使用法の値に関する更なる情報についてはHID仕様を参照してください)。

```
void hid_set_report (void)
{
    U16 wLength;
    U8 CAPS_LED = 0;
    U8 REPORT_ID;

    LSB(wInterface)=Usb_read_byte();
    MSB(wInterface)=Usb_read_byte();

    LSB(wLength) = Usb_read_byte(); // !< wLength読み込み
    MSB(wLength) = Usb_read_byte();
    Usb_ack_receive_setup();

    while(!Is_usb_receive_out());
    REPORT_ID = Usb_read_byte();
    CAPS_LED = Usb_read_byte(); // ホストによって送られたCPAS LED状態値取得
    Usb_ack_receive_out();
    Usb_send_control_in();
    while(!Is_usb_in_ready());
    // CAPS要求を解除するために報告送信
    Usb_select_endpoint (EP_KBD_IN);
    Usb_write_byte(0); // Byte0: 修飾子
    Usb_write_byte(0); // Byte1: (予約)
    Usb_write_byte(0); // Byte2: キー符号0
    Usb_write_byte(0); // Byte2: キー符号1
    Usb_write_byte(0); // Byte2: キー符号2
    Usb_write_byte(0); // Byte2: キー符号3
    Usb_write_byte(0); // Byte2: キー符号4
    Usb_write_byte(0); // Byte2: キー符号5
}
```

```

    Usb_ack_in_ready();
    // ホスト要求に従ってLED0をON/OFF切り替え
    if (CAPS_LED == 0)
        Led3_off();
    else
        Led3_on();
}

```

6.4. ブート不可からブート可の装置への変更方法

HID装置がブート可またはブート不可であるかもしれないことに注意してください。既定により、Atmelによって提供されるHID実演はブート不可装置です。応用がブート可を必要とする場合、([usb_descriptors.h](#)の副級(sub-class)項目を変更しなければなりません。

```

// USBインターフェース記述子 キーボード
#define INTERFACE_NB_KEYBOARD 0
#define ALTERNATE_KEYBOARD 0
#define NB_ENDPOINT_KEYBOARD 1
#define INTERFACE_CLASS_KEYBOARD 0x03 // HIDクラス
#define INTERFACE_SUB_CLASS_KEYBOARD 0x00 // ブート不可
#define INTERFACE_PROTOCOL_KEYBOARD 0x01 // キーボード
#define INTERFACE_INDEX_KEYBOARD 0

```

マウスをブート可装置へ変換するには、`INTERFACE_SUB_CLASS_KEYBOARD`を1に設定してください。

7. PCソフトウェア

USBキーボード応用はどんなPCソフトウェアも必要としません。

8. 制限

本実演はOUT報告を管理しません。この機能を処理するには必要なコードを追加しなければなりません(更なる情報については[6.3.項](#)を参照してください)。

9. 関連文書

- (使うデバイス番号に関連する)AVR USBデータシート
- AT90USBxxxマイクロ コントローラ用USBソフトウェア ライブラリ (doc7675)
- USB HIDクラス仕様



本社

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA

TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong

TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France

TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan

104-0033 東京都中央区
新川1-24-8
東熱新川ビル 9F

アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ウェブサイト

www.atmel.com

技術支援

avr@atmel.com

販売窓口

www.atmel.com/contacts

文献請求

www.atmel.com/literature

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。たとえば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、または利益の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。

© Atmel Corporation 2008. 不許複製 Atmel[®]、ロゴとそれらの組み合わせ、AVR[®]、STK[®]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。

© HERO 2021.

本応用記述はAtmelのAVR271応用記述(doc7602.pdf Rev.7602B-07/08)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。