

AVR262 : USB HIDでのAtmel QTouch

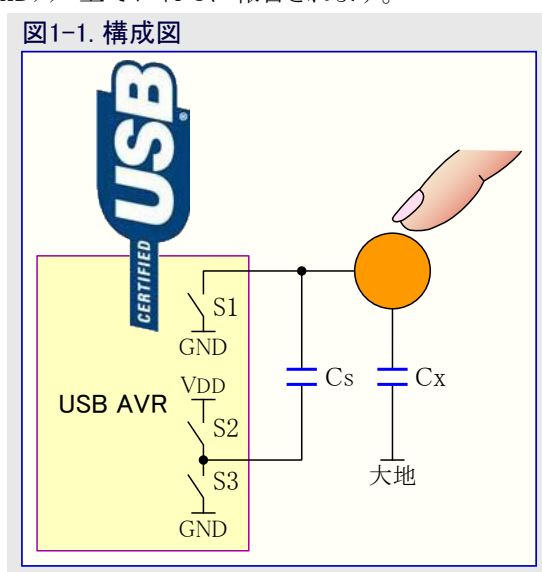
要点

- 単一チップ解決策
 - ・ Atmel® AVR® AT90USB646でのAtmel QTouch®とUSB HID
- QTouchライブラリ使用
 - ・ 8つのQTouchキー
 - ・ AtmelのQTouchライブラリを用いた独自ファームウェア
- 違うAVR USBデバイスに対する容易な可搬性
- IAR™とGCCを支援

1. 序説

この応用記述はUSBインターフェースを通してQTouch感知に関する単一チップ解決策を提供するためのAT90USB646マイクロコントローラでのQTouchライブラリ使用方法を説明します。接触感知器状態が採取され、USB HIDクラス上でホストPCに報告されます。

図1-1. 構成図



2. ハードウェア必要条件

- ・ AtmelのSTK®600、STK500または独自基板
- ・ AtmelのQTouch感知器基板
- ・ USBホスト



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。

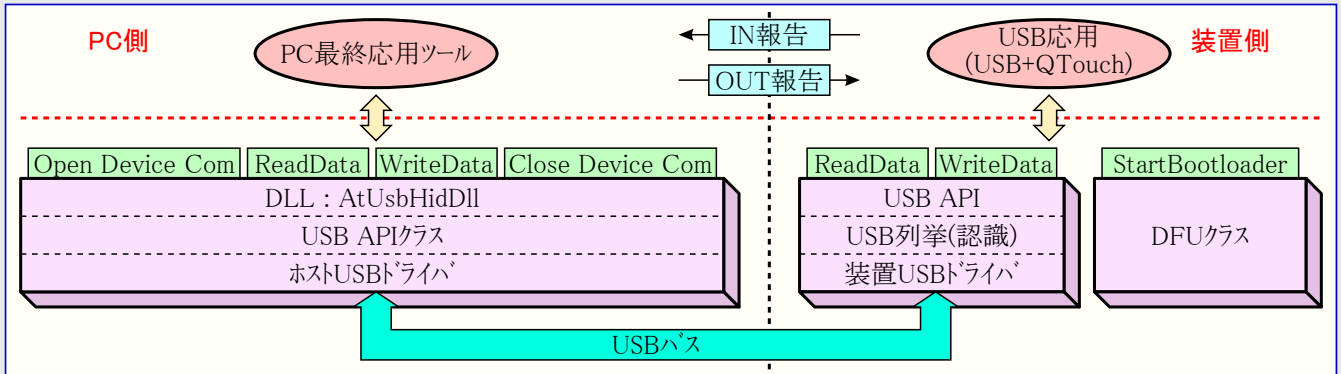
Rev. 8375A-03/11, 8375AJ2-05/21

3. 応用概要

この実演は感知されたQTouchキーを採取し、報告を生成し、要求に於いてそれをPC(ホスト)に送ります。応用は標準的なUSB HID応用上で構築され、そしてそれはPC(ホスト)と装置間の簡単なデータ交換を許します。PCは定期的なポーリング間隔で装置からの利用可能な新しいQTouchデータを要求します。装置は利用可能ならばデータを送ります。さもなければ、利用可能なデータが全く無いことをPCに通知するためにNAK(No Acknowledge)が送られます。装置からPCへ送られる一括データはIN報告として引用されます。

図3-1.は応用の全般概要を与えます。

図3-1. 応用概要



ホストへ送られる報告は使用者応用によって使うことができるバイトの組を含みます。この応用が丁度8つのQTouchキーを使うので、接触状態を転送するのに1バイトで充分ですが、これは必要条件に依存して変更することができます。

この応用では接触感知にポートAとポートCが使われます。SNSとSNSKのピンが同じポートに接続される時に偶数ピン番号がSNSピンとして、奇数ピン番号がSNSKピンとして使われます。

- ・ PA0,PA2,PA4,PA6 ⇒ SNS0,SNS1,SNS2,SNS3
- ・ PA1,PA3,PA5,PA7 ⇒ SNSK0,SNSK1,SNSK2,SNSK3

報告バイトの最下位ビット(LSB)はPA0とPA1に接続された感知器の値を持ちます。同様に報告バイトの次に高いビットはPA2とPA3に接続された感知器の値を持ちます。報告バイト内の感知器の位置は図3-2.で示されます。

図3-2. 接触データIN報告バイト

ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
PC7/PC6	PC5/PC4	PC3/PC2	PC1/PC0	PA7/PA6	PA5/PA4	PA3/PA2	PA1/PA0

PA1/PA0に接続された感知器上で接触が検出されると、報告バイトの値は\$01で、PC5/PC4で検出されるとその値は\$40です。残りの感知器に対する報告バイト値も同様に扱われます。

4. Touchライブラリ統合

応用を構築するには、他のソースファイルと統合されるべき(デバイスとコンパイラ特有の)適切なTouchヘッダファイル、アセンブリ言語ファイル、ライブラリファイルが必要です。正しいデバイスと応用特有ライブラリファイルを選択するにはAtmel QTouchライブラリの根フォルダ内のLibrary_Selection_Guide.xlsファイルを参照してください。ライブラリ選択の手引きはAtmelのウェブサイトでも入手可能です。

技術を"QTouch"、MCUシステムを"USB AVR"、MCU形式を"8-bit"、MCUを"Atmel AT90USB646"、ツールチェーンを"GCC/IAR"、QTouchに利用可能なポートを"A,B,C,D,E,F"、チャンネルの最大数を"8"として選択してください。これらの項目を選ぶことによってインクルードされるべき適切なライブラリファイルを見つけることができます。

- ・ libv3g1-8qt-k-0rs.r90 - 各々のデバイス用のIARライブラリファイル
- ・ libavr5g1-8qt-k-0rs.a - 各々のデバイス用のGCCライブラリファイル
- ・ qt_asm_tiny_mega.S - アセンブリ言語ファイル
- ・ touch_qt_config.h
- ・ touch_api.h

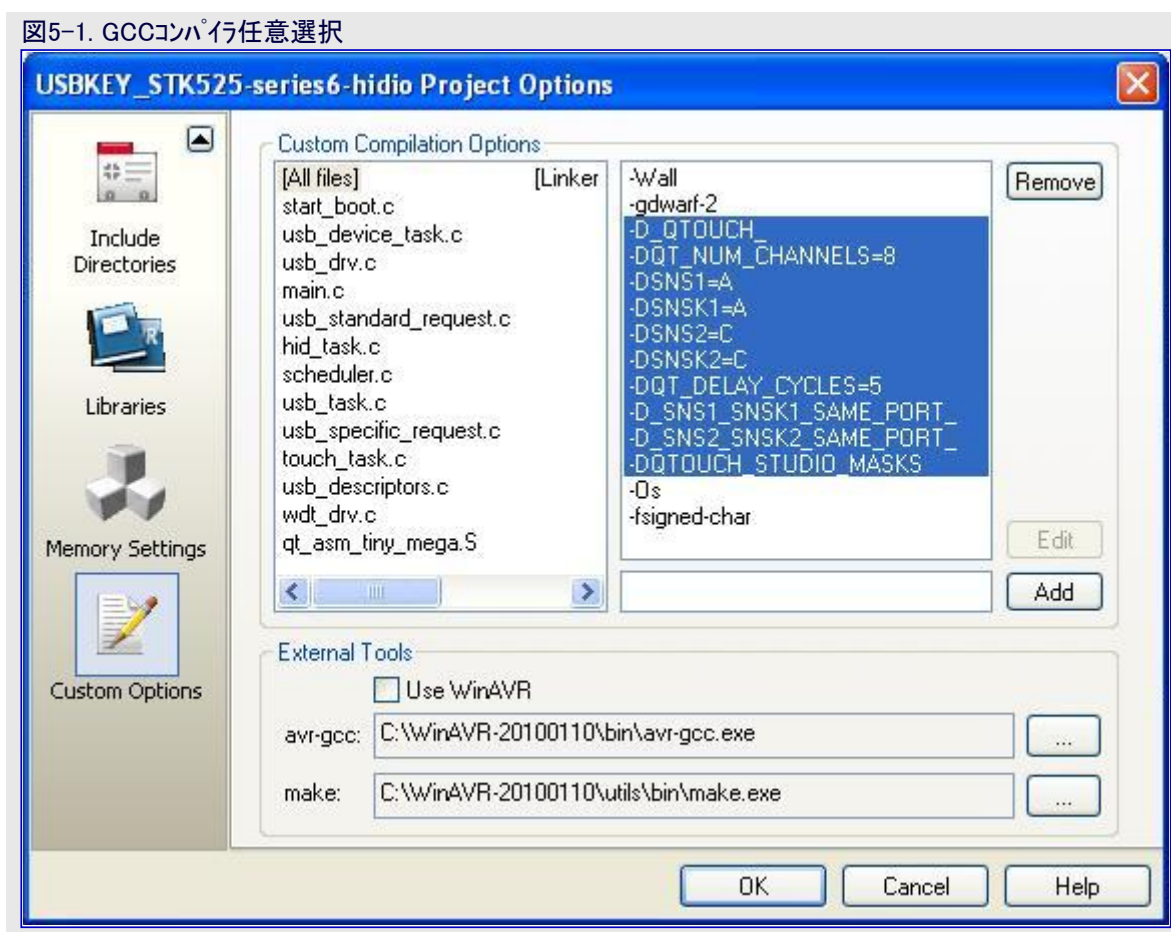
5. コンパイラとアセンブラの任意選択

IARとGCCで応用を構築するには設定されるべき或る任意選択が必要です。両コンパイラがそれら自身の方法で固有なため、それぞれに少しの違いが必要とされます。本章は各々のコンパイラに対して必要とされる必要な任意選択設定を説明します。

GCCでは**Project Options**を開き、そして**Custom Options**をクリックしてください。これらの独自コンパイル任意選択を1つずつ追加してください(図5-1.参照)。

```
-D_QTOUCH_
-DQT_NUM_CHANNELS=8
-DSNS1=A
-DSNSK1=A
-DSNS2=C
-DSNSK2=C
-DQT_DELAY_CYCLES=5
-D_SNS1_SNSK1_SAME_PORT_
-D_SNS2_SNSK2_SAME_PORT_
-DQTOUCH_STUDIO_MASKS
```

図5-1. GCCコンパイラ任意選択



IARについてはコンパイラだけでなくアセンブラでも任意選択が設定されなければなりません。IAR Project OptionsでC/C++ Compiler分類を選び、Preprocessorタブ上で図5-2.で示されるように以下の任意選択が提供されるべきです。図5-3.で示されるように、Assembler分類のPreprocessorタブ上で同じ任意選択が追加されるべきです。

図5-2. IARコンパイラ任意選択

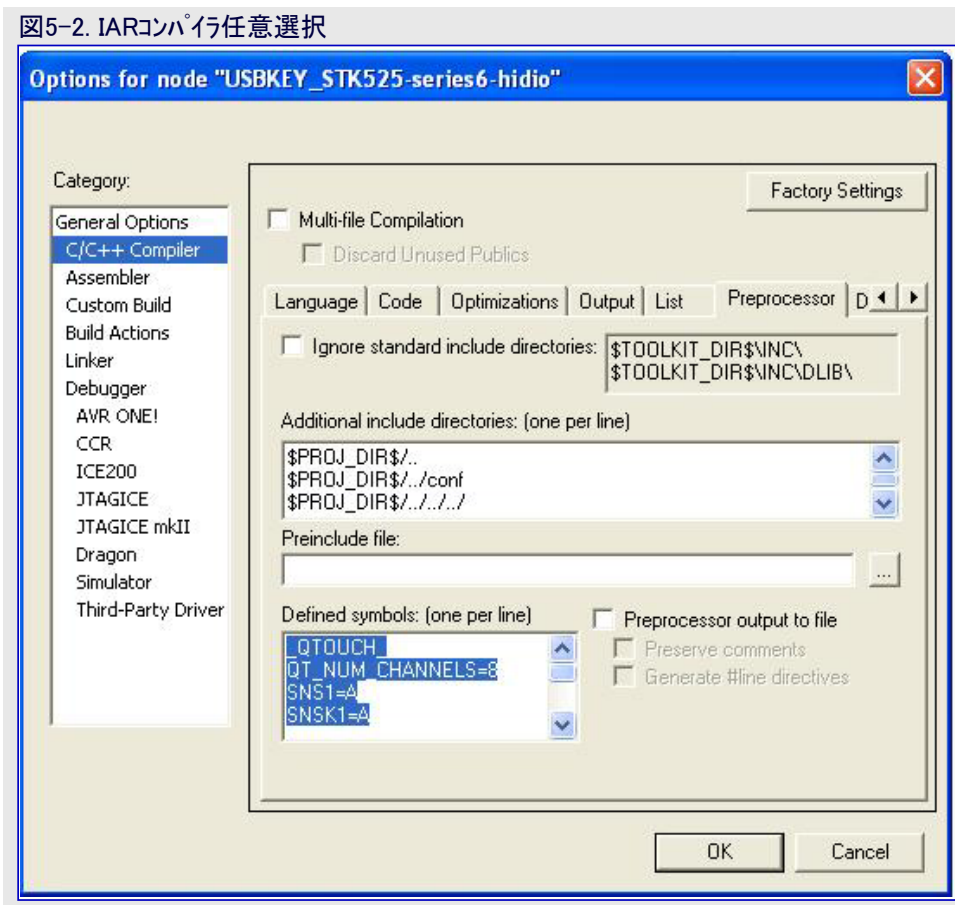
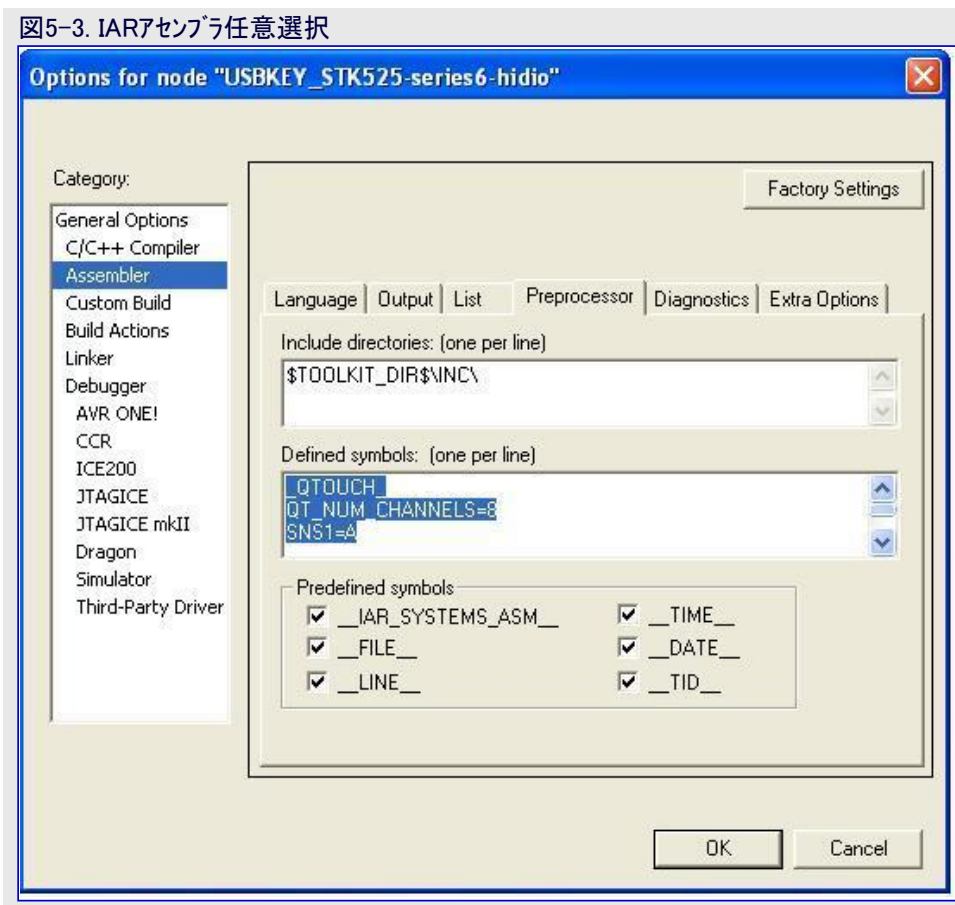


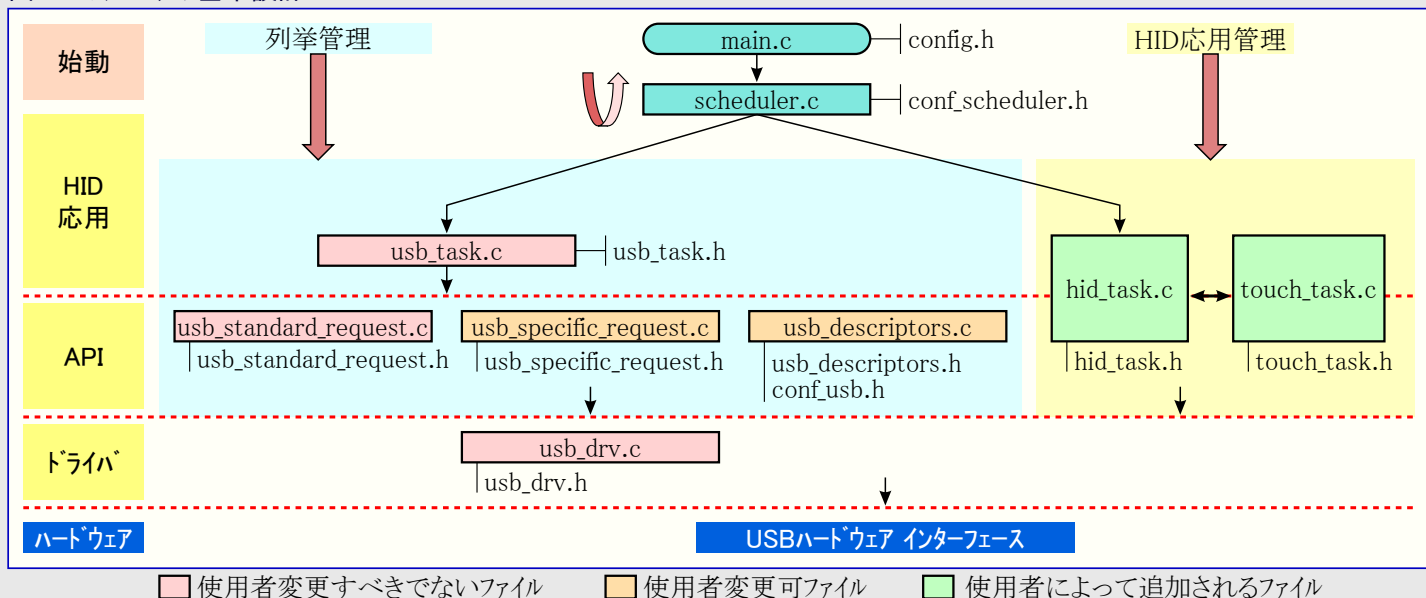
図5-3. IARアセンブラ任意選択



6. ファームウェア

この応用は`touch_task.c`と`hid_task.c`を含むHID応用管理部を使います。図6-1は構成図の手助けと共にファームウェアの基本設計を示します。

図6-1. ファームウェア基本設計



接触検知と報告生成は`hid_task.c`と`touch_task.c`、それとそれら各々のヘッダファイルで扱われます。

様々な関連する関数は3つの主要分野に群化することができます。必要な関数は以下の各関連分野で説明されます。

- ・ 初期化に関連する関数
- ・ 接触ライブラリAPI
- ・ HID報告生成

6.1. 初期化に関連する関数

6.1.1. config_keys

`config_key`関数には感知器をキーとして各々のチャネルに構成設定する責任があります。

6.1.2. CHANNEL_t

`CHANNEL_t`関数はキー用に構成設定されるべきチャネル番号を指定します。

6.1.3. AKS_GROUP

`AKS_GROUP`関数は“キー”として構成設定されつつある感知器と連携する隣接キー消去(AKS®:Adjacent Key Suppression)群を指定します。

6.1.4. 10u

`10u`関数は感知器に対する検出閾値を指定します。

6.1.5. HYST_6_25

`HYST_6_25`関数は感知器に対する検出ヒステリシスを指定します。

```
qt_enable_key( CHANNEL_0, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_1, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_2, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_3, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_4, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_5, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_6, AKS_GROUP_1, 10u, HYST_6_25 );
qt_enable_key( CHANNEL_7, AKS_GROUP_1, 10u, HYST_6_25 );
```

この関数のより多くの情報についてはAtmel QTouch使用者の手引きを参照してください。

6.1.6. qt_init_sensing

`qt_init_sensing`関数は全ての許可されたチャネルに対して接触検知を初期化するのに使われます。必要とされる全ての感知器はこの関数を呼び出す前に構成設定されるべきです。

```
#define qt_init_sensing( ) qt_init_sensing_with_burst( BURST_FUNC_NAME , CALCULATE_MASKS)
```

6.1.7. qt_set_parameters

`qt_set_parameters`関数はQTouch採取法ライブラリの`qt_config_data`変数内の全域構成設定を初期化するのに使われます。別の言葉で、これは構成設定データ構造体内の既定閾値を満たします。

6.1.8. touch_init_HID

`touch_init_HID`関数には接触検知に関連する初期化ルーチンを呼び出す責任があります。

```
/* 感知器をキーとして構成設定 */
config_keys();

/* 接触感知初期化 */
qt_init_sensing();

/* 使用者によって本関数内で再校正閾値、Max_On_Durationなどのようなパラメータを設定 */
qt_set_parameters();

/* 定期的に発生するように計時器割り込み処理ルーチン(ISR)を構成設定 */
init_timer_isr();
```

6.2. 接触ライブラリAPI

6.2.1. touch_measure

`touch_measure`関数には接触感知を測定する責任があります。

```
/* ライブラリに再び現れたことを示すための状態フラグ */
uint16_t status_flag = 0u;
uint16_t burst_flag = 0u;
if( time_to_measure_touch )
{
    /* フラグ解除:これは接触を測定するための時間 */
    time_to_measure_touch = 0u;
    do
    {
        /* 接触感知器1回測定 */
        status_flag = qt_measure_sensors( current_time_ms_touch );
        burst_flag = status_flag & QTLIB_BURST_AGAIN;
    }while ( burst_flag );
}
```


6.3. HID報告生成

6.3.1. hid_report_in

ホストがデータを要求すると必ず、それによって装置が応答します。新しいデータが全く無ければ、装置はNAK応答を送ります。新しいデータが利用可能な時に、装置はホストに情報を送ることが必要です。この関数にはUSB経由で送るべき報告を生成する責任があります。

接触または開放が検出されると必ず、touch_data変数の値が変わり、ホストが新しい状態で更新されます。

```

    Usb_select_endpoint(EP_HID_IN);
    if(!Is_usb_write_enabled())
        return;                                // 報告送出準備不可

    if (touch_data == old_touch_data)
        return;
    old_touch_data = touch_data;

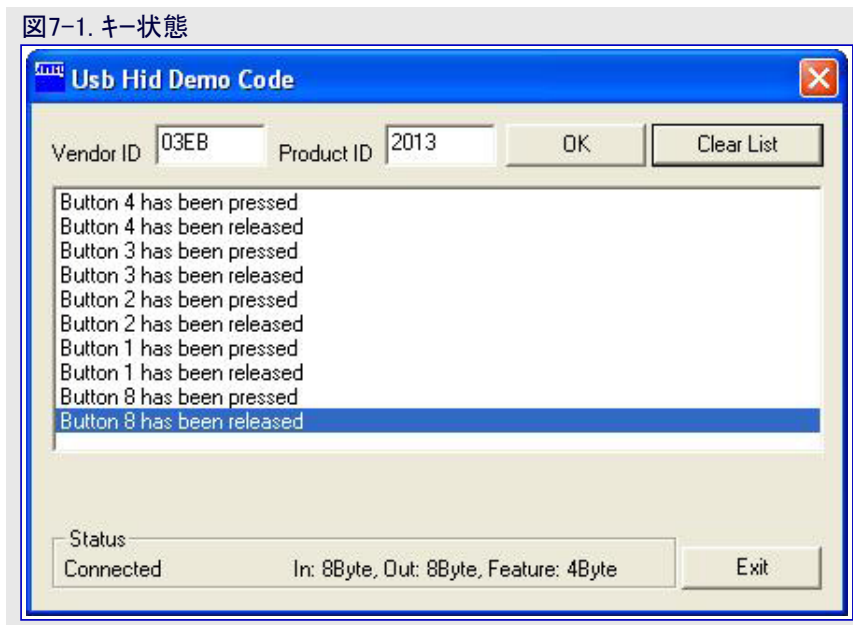
    // Send report
    Usb_write_byte(touch_data);                // 接触情報
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_write_byte(GPIOR1);                    // 擬態(未使用)
    Usb_ack_in_ready();                        // USB上でデータ送出

```

7. ホスト応用

この応用記述はどのキーが押された、または開放されたかを表示するのに使うことができるGUIに基づくPCホスト応用を含みます。このPC実行物は連携する配布物で入手可能です。各々のキー押下とキー解放は図7-1.で示されように表示されます。

図7-1. キー状態



8. 関連項目

- Atmel AT90USB646系データシート http://www.atmel.com/dyn/products/product_card.asp?part_id=3877
- Atmel AVR Studio® http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- Atmel AVR QTouchライブラリ 4.3 http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4627
- Atmel QTouchライブラリ使用者の手引き http://www.atmel.com/dyn/resources/prod_documents/doc8207.pdf
- AVR328:megaAVR®デバイスでのUSB標準HID実装 http://www.atmel.com/dyn/resources/prod_documents/doc7599.pdf
- AVR1511:QT600-ATxmega128A1練習の手引き http://www.atmel.com/dyn/resources/prod_documents/doc8350.pdf

9. 目次

要点	1
1. 序説	1
2. ハードウェア必要条件	1
3. 応用概要	2
4. Touchライブラリ統合	2
5. コンパイラとアセンブラの任意選択	3
6. ファームウェア	5
6.1. 初期化に関連する関数	5
6.1.1. config_keys	5
6.1.2. CHANNEL_t	5
6.1.3. AKS_GROUP	5
6.1.4. 10u	5
6.1.5. HYST_6_25	5
6.1.6. qt_init_sensing	6
6.1.7. qt_set_parameters	6
6.1.8. touch_init_HID	6
6.2. 接触ライブラリAPI	6
6.2.1. touch_measure	6
6.3. HID報告生成	7
6.3.1. hid_report_in	7
7. ホスト応用	7
8. 関連項目	7
9. 目次	8



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. 不許複製

Atmel®、ロゴとそれらの組み合わせ、それとAVR®、AVR®ロゴ、AVR Studio®、megaAVR®、AKS®、QTouch®、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。

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のウェブサイトにある販売の条件と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 2021.

本応用記述はAtmelのAVR262応用記述(doc8375.pdf Rev.8375A-03/11)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。