

AVR259 : ATtiny40 QTouch実演キット

要点

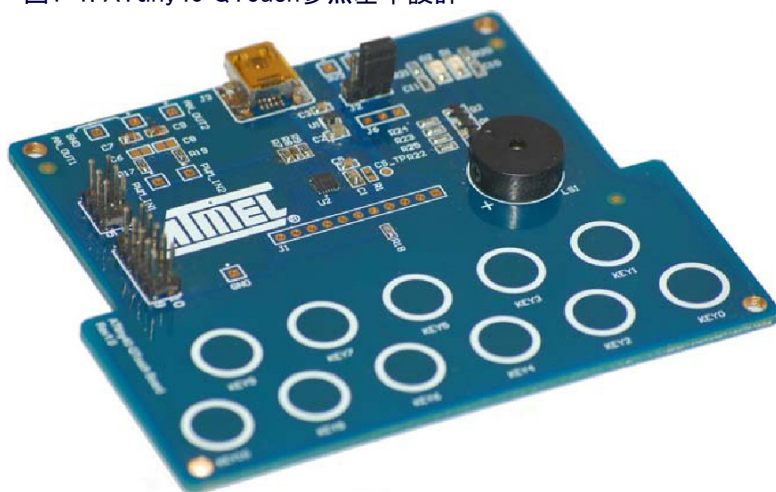
- 11個までのATMEL® AVR® QTouch®チャンネル
 - ・チャンネル毎に単一の入出力ピン
- TWI(Philips I²C適合)インターフェース
- SPIインターフェース
- 2つのアナログ出力(PWM)
- 警報器
- LED
- QTTM600 Touchテック インターフェース支援
- TPIプログラミング インターフェース
- USBまたは外部電源からの給電
- 特許と登録商標
 - ・ ATMEL QTouch® (充電移転法特許)
 - ・ AKS®(隣接キー抑制®特許)技術

1. 序説

この参照基準設計は最終的な容量性接触技術のQTouchを支援する、ATMEL ATtiny40の能力を実演することです。この技術は接触チャンネル毎に単一の入出力ピンだけが必要です。参照基準設計はPWMを用いた2つのアナログ出力チャンネル、接触を示す警報器とLED、そしてTWIまたはSPIの通信インターフェースを支援します。これは完全な形態設定能力を許し、基板外のQTouch感知器を支援します。

この応用記述は6つのキー、TWI従装置インターフェース(I²C適合)、警報器、LED制御を支援するファームウェアを含みます。

図1-1. ATtiny40 QTouch参照基準設計



2. ハードウェア

参照基準設計は11個のATMEL AVR QTouchキーを支援し、キー、滑動部、輪のような基板外接感知器を繋ぐ備えを持ちます。最終的なQTouch技術はチャンネル毎に1つの入出力ピンだけが必要で、どんな外部部品も必要ありません。QTouchチャンネルはADCピン上です。

2つのアナログチャンネルはATMEL ATtiny40からPWM経由で制御されます。これらのチャンネルは対応するキーが接触された時にアナログ値を提供します。これは既存の触感スイッチ(抵抗性梯子状)TV制御キーボード、またはキーの接触または開放を表示するのに通信インターフェースが必要ない他の何れかの応用を置き換える考えです。

警報器とLEDは通電またはキーの接触/開放状態の表示を提供するように形態設定することができます。



8ビット ATMEL
マイクロコントローラ

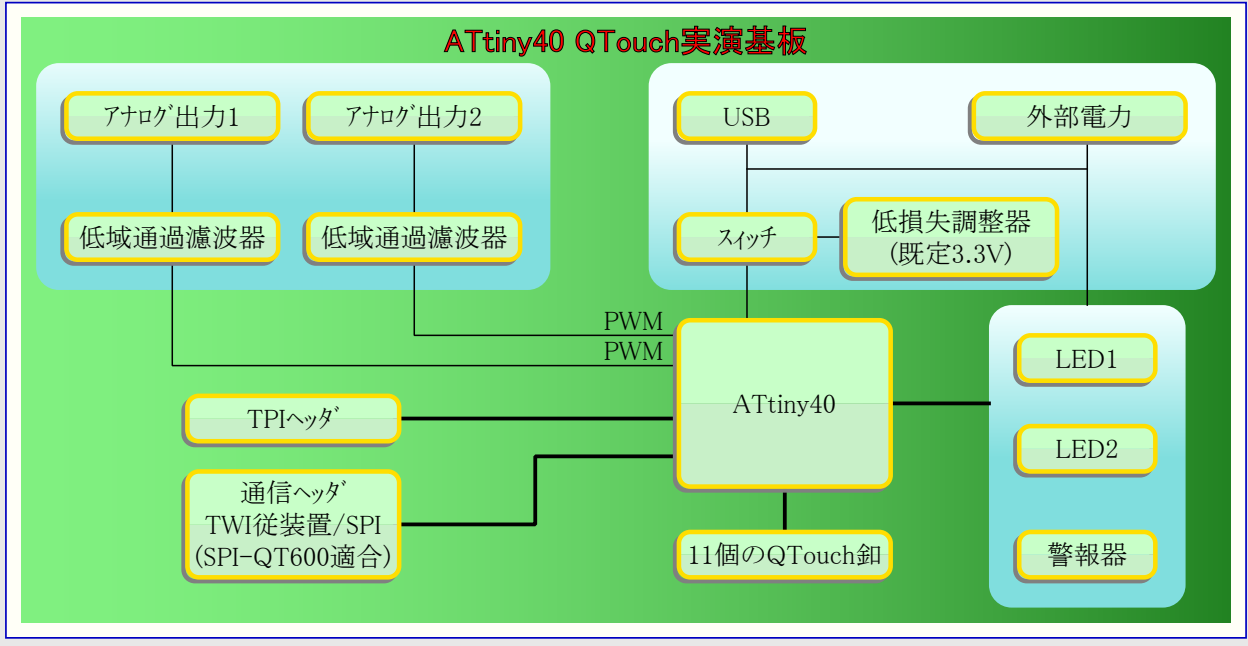
応用記述

暫定

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. 8332B-03/12, 8332BJ1-12/13

図2-1. システム構成図



2.1. QTouch

最終的な取得方法がATMEL AVR259で積分され、測定のためにA/D変換器(ADC)が必要です。接触ピンは容量性測定のためにADCピンであるべきです。ADC部署は使用者応用によってライブラリ操作のために開放されるべきです。ライブラリはADCの使用者設定を内部的に控え、接触測定のためにそれを改めて構成設定し、一旦全チャネルの測定が完了されると、控えでレジスタを復元します。

取得は非常に速く、代表的に4MHzでチャネル毎に100~300 μ sの範囲です。取得方法は一度に単一のチャネルです。

容量性取得が最初に積分されるため、Csコンデンサは旧来のQTouch法のように必要ありません。直列抵抗が必要ですが、エッジ速度制御のためとEMC/ESDの理由のためです。感知器設計は旧来のQTouch感知器と同じままです。

最終的なQTouch法は電位を知るために未知の容量の感知電極を充電します。電極は代表的に印刷回路基板(PCB)上の銅箔領域です。充電の結果は内部測定回路内に移転されます。充電と移転の1対の周回後に充電を測定することによって、感知面の容量を測定することができます。接触面上に指を置くことはその点に於ける充電の流れに影響を及ぼす外部容量を導入します。これを接触として記録します。

図2-2. QTuch回路構成図

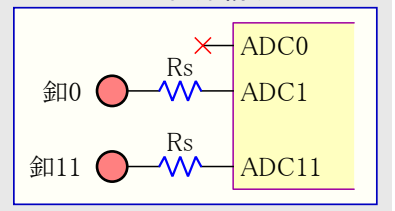
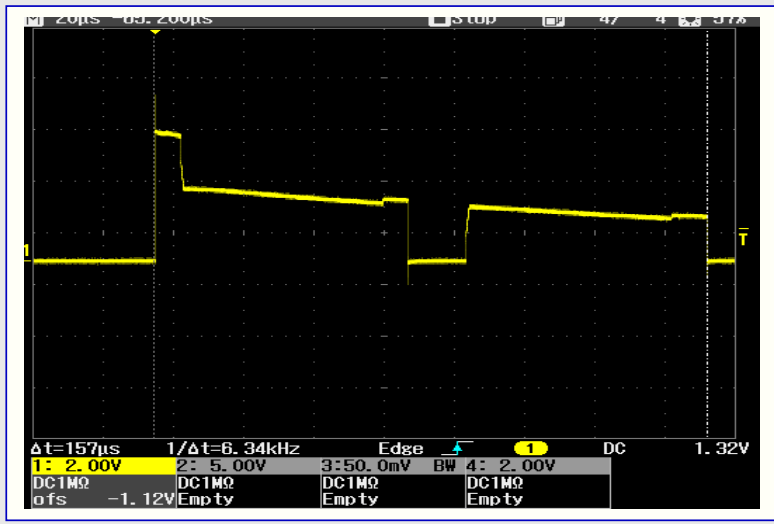


図2-3. 充電移転波形



判定論理内の信号処理が強力で信頼性のあるQTouchを作ります。静電気のスィーク、瞬間的な予期せぬ接触、または近接のための誤った起動は除去されます。QTouch感知器は単一または複数のキーを駆動することができます。機能と美的の両方の必要条件にあわせるために異なる大きさや形状のキーを使用することができます。

QTouchデバイスは経年や充電環境条件のために緩やかな変化に責任を持つように自動的な変動補償が特徴です。それらは数十年間の動的範囲を持ち、コイル、発振器、RF部品、特殊ケーブル、RC網、または幾つものディスクリート部品は必要ありません。工学的解決として、QTouchは簡単、強力、そして手頃です。

多数の接触キーが共に近接する場所では、指の接近が複数のキー周辺で容量変化を引き起こします。ATMELの特許の隣接キー抑制 (AKS:Adjacent Key Suppression)は各キーでの容量変化を繰り返して測定し、その結果を比較して使用者が接触を意図したのがどのキーかを定める反復技術を使用します。AKSは選択したキーからの信号が閾値以上に留まる場合に、他の全てのキーからの信号を抑制または無視します。これは隣接キーでの誤った接触検出を防ぎます。AKSはシステム設計者によって選択できます。

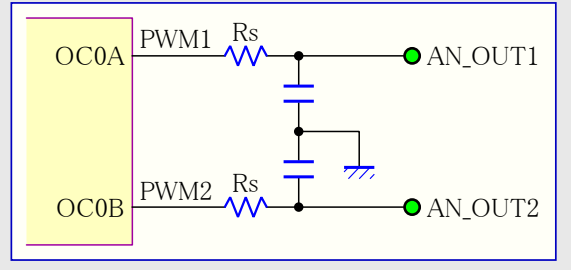
注: この測定法では次の通りです。

- 接触 : 信号>参照基準
- 非接触 : 信号<参照基準
- △ : 信号=参照基準

2.2. アナログ チャネル

PWM信号から配給されるアナログ出力は1次RC低域通過濾波器(LPF:Low Pass Filter)に供給されます。

図2-4. アナログ チャネル



2.2.1. PWM入力

PWM_IN1(PWM1)とPWM_IN2(PWM2)は既定で各々、ATtiny40のOC0A(PC0)とOC0B(PA5)のピンによって駆動されます。

任意選択で、それらは他の何れかのATtiny40ピンまたはPWM_IN1またはPWM_IN2の検査点への接続を経由した外部供給元によって駆動することができます。

2.2.2. アナログ出力

アナログ出力を引き出すために検査点が提供されます。これは低インピーダンス負荷を駆動するために外部緩衝器または電圧追跡器に供給することができます。

2.3. 警報器

基板上的の警報器(ブザー)はキーの接触/開放での告知のような使用者警報発令として使用することができます。基板は警報器を駆動するための外部駆動部回路を含みます。警報器は既定でATtiny40のPB1ピンによって駆動されます。任意選択で他のATtiny40ピンのどれでも、またはJ6ヘッダの3番ピンへの接続経由で外部供給元によって駆動することができます。

2.4. LED

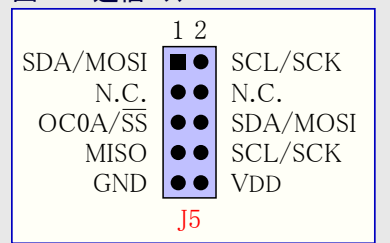
LED1とLED2は既定で各々、ATtiny40のPB2とPB3によって制御されます。しかし、警報器と同様に、他の何れかのATtiny40ピンまたはJ6ヘッダの2番ピンと1番ピンへの接続経由で外部供給元によって駆動することができます。

基板上的のLED駆動回路はいくつかの白色LEDのような高電流LEDでの試験に於いて手助けします。

2.5. 通信インターフェース

通信インターフェースはJ5ヘッダで利用可能なTWIとSPIの線を持ちます。ヘッダ配列はQT600デヘッダインターフェース互換です。

図2-5. 通信ヘッダ



2.6. 電源

2.6.1. 電力任意選択

2.6.1.1. USBコネクタ

5Vの電力はUSBコネクタ経由で基板に都合よく提供することができます。

2.6.1.2. 基板上の調整器

直線降下調整器はATtiny40/接触インターフェースに調整された電力を配給するために基板上(U1)に装着することができます。LEDや警報器のような外部負荷はこの調整された電圧によって給電されません。

推奨される調整器は以下です。

- Torex (XC6215系列)
- SEIKO (S817系列)
- BCDセミコンダクタ (AP2121系列)

2.6.1.3. 外部供給

1.8~5Vの範囲で外部電力を供給するのに基板上で検査点が提供されます。

2.6.2. 電力の考察

電源が温度で緩やかに変化する場合、デバイスはそれらの変化を感知して感度に於いて些細な変化だけで自動的に補償します。供給電圧が急に変動または移動する場合、変動補償機構は維持することができず、変則感度や誤った検出を引き起こします。

デバイスがアナログ基準電圧としてそれ自身を使用するため、その電力は非常に綺麗で独立した調整器から来るべきです。標準的で安価な低損失(LDO)形式の調整器が使用されるべきで、またLED、継電器、または他の高電流デバイスのような他の負荷の電力に使用されるべきではありません。LDOの出力での負荷変動はVCCに誤った検出または感度変動を引き起こすのに十分な変動を引き起こします。

警告: 他の論理デバイスと共用される調整器ICは突飛な動作に終わるかもしれず、推奨されません。

短い線の単独の0.1μFセラミック迂回コンデンサは供給ピンの非常に近くへ配置されるべきです。

供給波動と雑音は±25mVよりも多くないことが推奨されます。

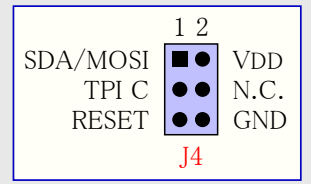
3. フログラミングとデバッグ

3.1. TPIフログラミング

デバイスはTPIインターフェースを通してフログラミングすることができます。基板上のJ4ヘッダがTPIフログラミングインターフェースです。

注: TPIフログラミングは5Vでだけ可能です。

図3-1. フログラミング ヘッダ



3.2. デバッグ

ATtiny40はチップ上デバッグを支援せず、コードはATMEL AVR Studio®またはIAR™でのシミュレータを使用するのみデバッグをすることができます。

4. ファームウェア

接触基板のファームウェアは独自化されたATMEL QTouchライブラリと使用者応用コードの2つの部分から成ります。接触関数は使用者応用コードから呼び出されます。ATMEL QTouchライブラリは容量性感知と後処理だけです。汎用入出力駆動、休止、通信などの残りの機能は使用者応用コードによって実行されなければなりません。ATMEL QTouchライブラリは特許のAKS(Adjacent Key Suppression (隣接キー抑制))を組み入れ、この基板は容量性接触感知用の特許QTouch技術で設計されています。

ATMEL QTouchライブラリ使用者の手引きについては「[追補](#)」章を参照してください。

4.1. 使用者応用コード

4.1.1. 使用者応用必要条件

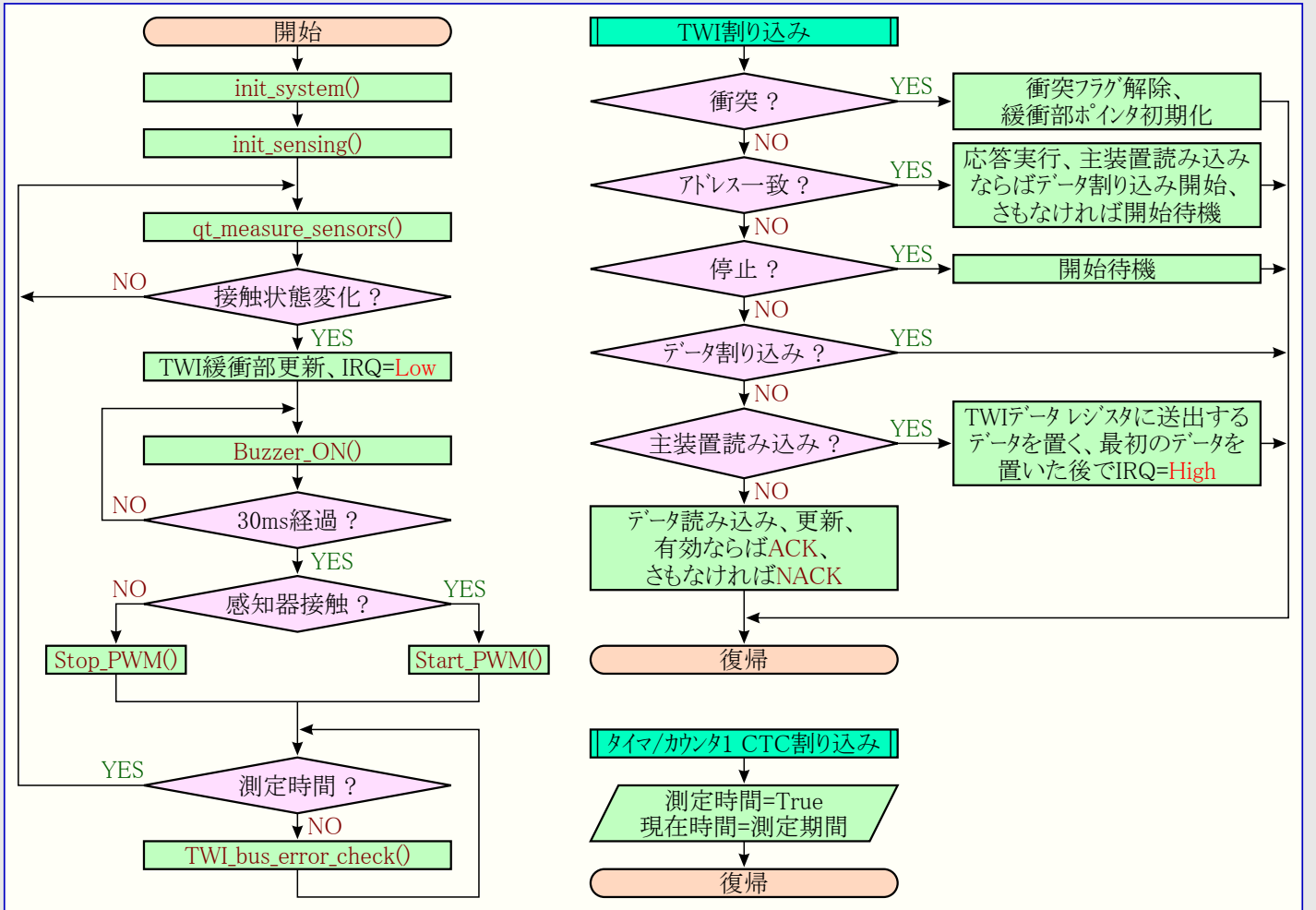
ライブラリは以下の必要条件に合うような使用者応用が必要です。

1. 現在の時間を感知しなければなりません。この情報は`qt_measure_channels()`関数への引数としてコードライブラリへ渡されます。これは変動のような時間に基づくライブラリ操作に使用されます。
2. 汎用入出力の内部プルアップはライブラリを呼ぶ時に禁止されなければなりません。ATtiny40ではプルアップがリセット後に既定で禁止されます。
3. ライブラリは使用者の接触に対して妥当な応答時間を提供するのに足るだけ度々呼ばれなければなりません。ライブラリ関数呼び出し中、使用者主応用コードは動きません。従って使用者応用に利用可能なプロセッサ時間、系の電力使用、系応答性間に二律背反があります。
4. それ自身とライブラリの両方に対する十分なスタック量。使用者応用スタックはライブラリ、加えてライブラリ関数を呼ぶ時のそれ自身の動作、加えてライブラリ関数呼び出し中に処理されるかもしれない許可された割り込みのどれに対しても充分大きくなければなりません。

使用者応用は未使用感知ピンを汎用入出力として使用することができます。感知器はそれらが許可される順番で番号付けされます。ライブラリは接触感知中で時間が重要な期間に対して割り込みを禁止します。それらの期間は一般的に数周期長だけで、故に使用者応用の割り込みは接触感知中に反応可能に留まるでしょう。けれども、接触感知中のどの割り込み処理ルーチン(ISR)も、接触測定や応用応答性に影響を及ぼすのを避けるために、可能な限り短くあるべきです。経験則として、容量性測定中のどのISRの組み合わせた期間でも1ms未満であるべきです。これはオシロスコープで接触チャネルでの集中区間を調べることで、系開発中に検査することができます。使用者がどの感知器にも接触しない時に集中区間が1msよりも多く変化する場合、逆にISRは測定に影響を及ぼし得ます。使用者応用構築時、ライブラリ関数はそれらが実際に呼ばれた場合にだけリンクされます。

4.1.2. 応用コードの流れ

図4-1. 流れ図



- 系の初期化は適切なデータ方向レジスタ設定によるMCUポート初期化、High不活性への初期PWM出力設定、全域変数初期化、それとTWIインターフェースまたはデバッグインターフェースの許可から成ります。
- ATMEL QTouchライブラリは"qt_enable_key()"を呼んで閾値、ヒステシス、AKS群を設定することによって適切なチャネル上のキーを許可することで初期化されます。全てのキーは異なるAKS群です。
- 使用者応用は許可されたチャネル全てを測定するために"qt_measure_sensors()"を呼びます。
- 感知器状態で変化があれば、IRQ線はLowを主張され、感知器の状態がTWI緩衝部に複写されます。
- 感知器状態で変化がある時に、30ms間警報器がONに切り換えられます。PWM出力は安定したLowを保たれ、警報器は4kHzの方形波を生成することによってONに切り換えられます。
- 30ms後、警報器はOFFに切り換えられ、感知器が接触されていれば、キーが接触されている限り、接触されたキーに対応するアナログ出力ピンでPWM信号が出力されます。
- 感知器が接触されている時にLEDもONに切り換えられ、感知器が開放された時にOFFへ切り換えられます。
- 各々の割り込み処理ルーチン(ISR)はTWI、PWM、現在の時間探知を処理します。
- 測定時間の場合、qt_measure_sensors()から続けます。

4.2. TWIインターフェース動作

TWIインターフェースはI²CとSMBus適合です。ATtiny40内のTWI部署は従装置機能だけを実装します。バス上での調停喪失、異常、衝突とクロック保持がハードウェアで検出されて示されます。従装置論理回路はパワーダウンを含む全ての休止形態で動作を継続します。

この応用記述で実装されるコードはこの接触応用のための独自TWI従装置ドライバです。

この従装置ドライバ Cコードは以下の2つのファイルから成ります。

- TWI_Slave.c
- TWI_Slave.h

main.cファイルはこのドライバの使用法の例を含みます。TWI_Slave.hファイルは主応用に含まれなければならない、全ての関数宣言と全てのTWI状態符号に関する定義を含みます。TWI状態符号は異常メッセージを評価して適切な活動を取るのに使用することができます。TWI_Slave.cファイルは全てのドライバ関数を含みます。

いくつかのデバイスは多数のTWI従装置アドレスへの応答をデバイスに許す、追加のTWIアドレス遮蔽レジスタ(TWIAMR)を持ちます。ここで記述される標準実装の独自化版は応用記述付属物に含まれます。

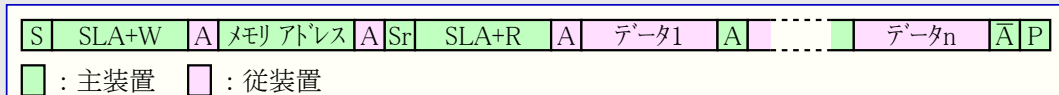
表4-1. TWI従装置ドライバ関数

関数	説明
TWI_slave_initialize()	TWI従装置をそのの初期待機状態に構成設定するためにこの関数を呼んでください。全てのTWI割り込みが許可されます。TWI初期化後に主応用から全体割り込みを許可することを忘れないでください。従装置アドレスはTWI_Slave.hファイルで定義されます。 #define TWI_ADDR 0x10
TWI_touch_status()	送り出しのために接触感知器状態をTWI緩衝部へ複写するのにこの関数を呼んでください。状態に於ける変化を主装置に告示するためにIRQ線はLowを表明されます。
_interrupt void TWI_ISR()	これは割り込み処理ルーチン(ISR)関数で、TWI割り込みが起動される時、即ちTWI事象が起こされた時には必ず自動的に呼ばれます。この関数は主応用から直接呼ばれるべきではありません。TWI ISRはTWI緩衝部の送信と主装置からのアドレス位置の受信の状態機構を処理します。
TWI_bus_error_check()	転送またはその修正中に不正なバス状態が起こされたかを調べるにはこの関数を呼んでください。

4.2.1. データ読み込み

デバイスからデータを読むための事象の流れが図4-2.で示されます。

図4-2. TWIバイト体裁



ホストは**開始条件**を送ることによって転送を始め、書き込み(W)ビットと共に装置の従装置アドレスを送ることによってこれに後続します。装置は**ACK**を送ります。ホストはその後にそれから読もうとする装置内のメモリ アドレスを送ります。装置は**ACK**を送ります。

ホストはその後に再び従装置アドレスによって後続される**再送開始条件**を送りますが、この時は読み込み(R)ビットが伴われます。装置は**ACK**または**NACK**のどちらかを返します。ホストが**ACK**を返す場合、装置は次のアドレスからのデータ バイトをその後に送信します。データ バイトが送信される毎に、装置は内部アドレスを自動的に増やします。装置はホストが**NACK**で応答するまでデータ バイトを返し続けます。ホストは**停止条件**を発行することによって転送を終わります。

表4-2. 通信バイト

アドレス	用途	アクセス
0	基板ID	読み込み
1	ファームウェア版番号	読み込み
2	感知器状態	読み込み

4.2.2. SDA,SCL

ピンはオープンドレインで、故に装置はそれらの線をLowに引く、またはそれらを開放に解放することだけができます。終端抵抗は装置が線を引き下げない場合に線をVDDに引き上げます。

終端抵抗は一般的に1kΩ ~10kΩ の範囲です。

4.2.3. IRQ線

IRQ線はLow活性の線で、感知器の状態で変化がある時にLowを表明されます。IRQ線は転送されるためにTWIバス上へ最低1バイトが置かれる時にリセットされます。

4.3. アナログインターフェース動作

基板は各々のキーに対して安定した状態の異なるアナログ電圧経由でホスト制御器へ接触状態を通信することができます。アナログ出力はMCUから生成されてRC濾波されたPWM信号です。表4-3.は5Vの供給電圧に対する様々なアナログ出力電圧を一覧にします。

表4-3. 各キーに対するアナログ値

状態	アナログチャンネル1 (AN_OUT1)	アナログチャンネル2 (AN_OUT2)
アイドル	5.0V	5.0V
KEY0	2.5V	0.0V
KEY1	2.0V	0.0V
KEY2	1.5V	0.0V
KEY3	1.0V	0.0V
KEY5	0.0V	2.5V
KEY6	0.0V	2.0V
KEY7	0.0V	1.5V
警報器ON	0.0V	0.0V

注: ・ KEY4はそれがPWMチャンネルとして使用されるために許可されません。
 ・ 警報器ON期間は30msです(両PWMチャンネル=0V)。

アナログ出力信号ピンで観測される波形は使用者インターフェースの2つの不時の事象によって告知されます。即ち、

- ・ 接触されつつある接触制御キー
- ・ 開放(非接触)されつつある接触制御キー

アイドル(IDLE)状態、即ち接触制御キーが接触/開放されていない時に於いて、PWM出力は(継続的な)Highを保ちます。

接触または開放された時に、警報器はどれかのキーが接触された時から30msの期間活性です。PWM出力は警報器が活性の期間、即ち何れかのキーが接触された時から30ms、Lowを保たれます。この期間中に接触測定は全く起こりません。これはATMEL AVR259と同じ供給から重たい負荷を駆動する時の接触測定を避けることを手助けします。

能動RC低域通過濾波器を使用してアナログレベルを生成する各々のキーに対する仕様によって、PWMは何れかのキーが接触された後の30ms後から、キーが接触されなくなるまで活性です(無限最大ON期間)。キーが開放されると、警報器は30ms間活性で、その時間中のアナログレベルは無効です。30msの期間の後で、それはVCCです。

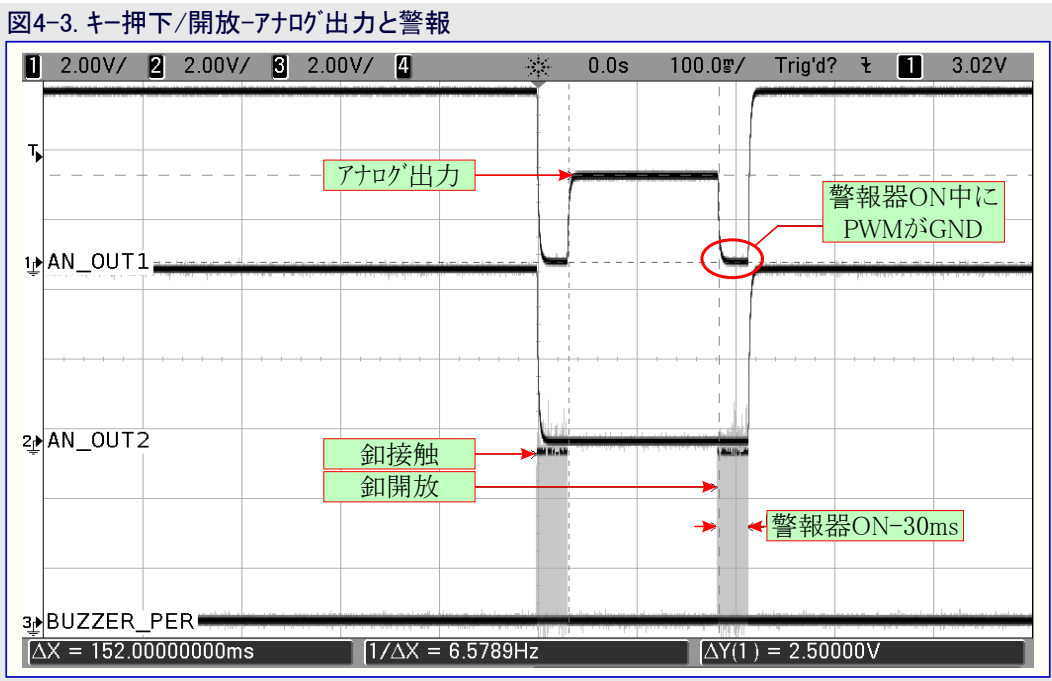
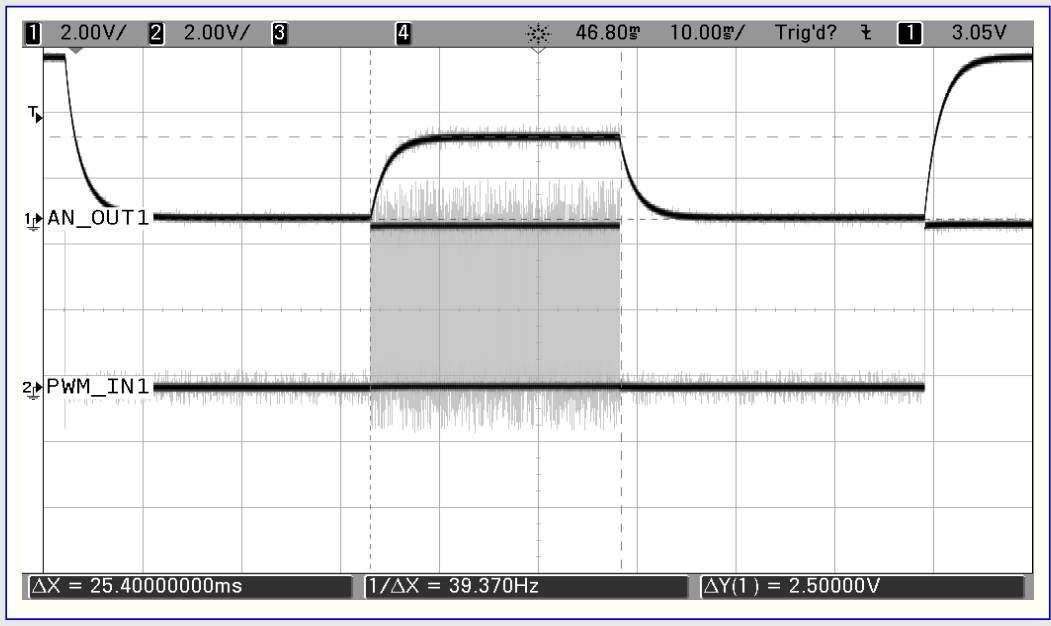


図4-4. PWMとアナログ出力レベル



4.4. 警報器とLEDの動作

基板上の警報器はどれかのキーの接触/開放での告知に使用されます。警報器はどれかのキーの接触/開放で30ms間(4kHzで)活性です。

LED2はチップが給電される時に点灯し、LED1はどれかのキーが接触した時に点灯して接触が開放されるまでONに留まります。

表4-4. 主な関数

関数	説明
init_system()	この関数はクロック、各種ポート方向、状態を初期化します。
init_timer0_isr()	この関数はタイマ/カウンタ1をPWM動作用に初期化します。
init_timer1_isr()	この関数はタイマ/カウンタ0を実時間計算用に初期化します。
pwm_idle_state()	この関数はPWMピンをアイドル状態用に形態設定します。
buzzer_pwm_control()	この関数は接触または開放の事象中に警報器、LED、PWMを制御します。
<u>__interrupt void timer1_isr(void)</u>	タイマ/カウンタ1割り込み処理ルーチン
report_debug_data()	この関数は接触デバッグ動作形態でデバッグデータを報告します。

4.5. ピン配置

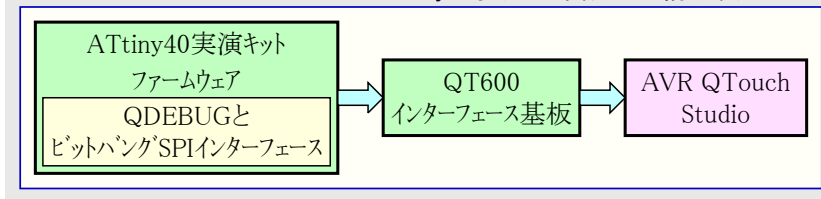
表4-5. ピン配置

機能	AVRピン	ヘッダ - ピン
電力LED	PB3	-
接触LED	PB2	-
警報器	PB1	-
PWM出力1	PC0	-
PWM出力2	PA5	-
SDA	PC4	J5-1番ピン
SCL	PC1	J5-2番ピン
IRQ	PC2	J5-7番ピン
MOSI (注)	PC4	J5-6番ピン
SCK (注)	PC1	J5-8番ピン

注: QT600接触デバッグ インターフェース。TWI禁止。

5. QT600接触デバッグ インターフェース

図5-1. AVR QTouch StudioとのATtiny40参照基準設計の構成展望図



ATtiny40参照基準設計ファームウェアはAVR QTouch Studioでの接触データ分析のためのQDebug規約を支援します。QDebug規約はATtiny40参照基準設計とQT600インターフェース基板間の通信規約です。QT600インターフェース基板はATtiny40参照基準設計とAVR QTouch Studio間の通信インターフェースです。QTouch StudioとのATtiny40参照基準設計インターフェースの構成展望図については図5-1を参照してください。

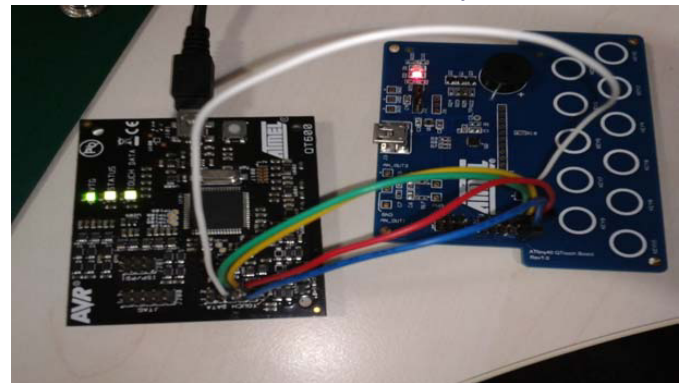
ATtiny40参照基準設計はQDebug規約を用いて1方向SPI(ビットバンク)インターフェースでQT600インターフェース基板を通してQTouch Studioへデータを送ります。使用者はQTouch Studioでチャネル信号、参照基準、変量の値を可視化することができます。

QTouch StudioとQT600インターフェース基板を使用して接触データを分析する間、図5-2.で示されるようにQT600インターフェース基板を通してATtiny40参照基準設計に電力を提供してください。QTouch Studioとのインターフェースに関して、接続は表5-1.で示されるようにATtiny40参照基準設計とQT600インターフェース基板間が作られなければなりません。

表5-1. ATtiny40参照基準設計とQT600インターフェース基板間の接続

インターフェース	ATtiny40参照基準設計 (J5コネクタ)	QT600インターフェース基板 (TOUCH DATAコネクタ)
1方向SPI (ビットバンク)	GND (検査点)	SS (5番ピン)
	MOSI (6番ピン)	MOSI (6番ピン)
	SCK (8番ピン)	SCK (8番ピン)
	GND (9番ピン)	GND (9番ピン)
	VDD (10番ピン)	VDD (10番ピン)

図5-2. QT600インターフェース基板でインターフェースされる
ATtiny40参照基準設計



注: QT600デバッグ インターフェース動作形態では以下です。

1. TWIはファームウェアで禁止されます。
2. 接触データ分析中、QT600インターフェース基板を使用してATtiny40参照基準設計に電力を投入してください。

注: ファームウェアは2つの形態設定形態から成ります。1つ目は公開形態で2つ目はデバッグ形態です。

公開形態はTWIインターフェース動作とQT600接触デバッグインターフェース動作の両方を支援します。既定によってTWIインターフェース動作が許可されます。QT600接触デバッグ インターフェース動作を許可するために、プロジェクト任意選択のC compiler⇒Preprocessor下で“TWI_INTERFACE”マクロが禁止されなければなりません。

デバッグ形態はQT600接触デバッグ インターフェース動作と(AVR Studioでのシミュレータを使用する)コード デバッグの両方を支援します。デバッグ形態ではTWIインターフェースが禁止されます。

6. 追補

6.1. ATMEL QTouchライブラリ

ATMEL QTouchライブラリは予めコンパイルされた保管庫として利用可能で、リンクに必要なCヘッダファイルと共にIARコンパイラでの使用が可能な使用料無料の静的ライブラリです。使用者応用はATtiny40を含む支援される容量性MCUの多くに対してATMELのQTouchライブラリの最上部で開発することができます。このライブラリはA/D変換器(ADC)、RAM、ROM、いくつかのレジスタ変数、汎用入出力を除いて、タイマ/カウンタ、割り込み、または他のチップ資源を全く使用しません。ライブラリは接触感知だけを提供し、使用者応用は必要な他のどの機能も提供しなければなりません。既存ファームウェアに使用されたATMEL QTouchライブラリはlibtiny40_8qt_k_Ors.r90です。

ATMEL QTouchライブラリを使用するための一般的な流れは以下のように記述することができます。

- ・使用者応用は全てのチャンネルと接触感知パラメータをそれらの既定状態にリセットするために(任意選択で)"qt_reset_sensing()"を呼びます。この段階は使用者が走行時にライブラリの動的な再形態設定を望む場合にだけ必要とされます。
- ・使用者応用は接触感知器の形態設定を必要とした時に"qt_enable_key()"、"qt_enable_rotor()"、"qt_enable_slider()"を呼びます。
- ・使用者応用はライブラリを初期化するために"qt_init_sensing()"を呼びます。
- ・その後、使用者応用は容量性測定を行うために定常的に"qt_measure_channels()"を呼びます。各呼出し後、どれかの感知器が検出されたか、許可された何れかの回転部または摺動部の角度や位置を見るために"qt_touch_status"全域変数を調べることができます。

6.2. 応用プログラミング インターフェース

6.2.1. 明示定数

APIはライブラリに提供する表6-1.で一覧にされる明示定数を定義します。ライブラリはこれらの値を用いて構築され、それらは変更されるべきではありません。

表6-1. 明示定数

明示定数	注記
QT_NUM_CHANNELS	ライブラリで支援される接触チャンネル数
QT_MAX_NUM_ROTORS_SLIDERS	ライブラリで支援される回転部と摺動部の最大番号

6.2.2. 型定義

APIは表6-2.で一覧にされる型定義を定義します。

表6-2. 型定義

型定義	注記
uint8_t	符号なし8ビット数値
uint16_t	符号なし16ビット数値
int16_t	符号付き16ビット数値
threshold_t	感知器検出閾値を設定する符号なし8ビット数値

6.2.3. 構造体

APIは表6-3.で一覧にされる構造体を使用します。この型の"qt_touch_status"全域変数が宣言され、許可された全ての感知器の現在の状態を示します。

表6-3. 構造体

構造体	領域	注記
qt_touch_status_t	sensor_states	ライブラリ感知器の状態(ON/OFF) ビット"n"=感知器"n"の状態 0=検出なし、1=検出
	rotor_slider_values[]	回転部角度または摺動部位置。これらの値は対応する回転部または摺動部を示す"sensor_states"が検出の時に有効です。

6.2.4. 列挙

APIは表6-4.で一覧にされる列挙を使用します。

表6-4. 列挙

名前	値	注記
aks_group_t	NO_AKS_GROUP AKS_GROUP_1 AKS_GROUP_2 AKS_GROUP_3 AKS_GROUP_4 AKS_GROUP_5 AKS_GROUP_6 AKS_GROUP_7	感知器がどのAKS群に入るかを示します。NO_AKS_GROUP=感知器はAKS群に入らず、抑制することができません。 AKS_GROUP_x=感知器はAKS群xに入ります。
channel_t	CHANNEL_0 CHANNEL_1 CHANNEL_2 CHANNEL_3 CHANNEL_4 CHANNEL_5 CHANNEL_6 CHANNEL_7	感知器内のチャネル。
hysteresis_t	HYST_50 HYST_25 HYST_12_5 HYST_6_25	感知器検出ヒステリシス値。これは感知器検出閾値の%として表されます。 HYST_x=ヒステリシス値は検出閾値のx%です(切り捨て丸め)。 強制限界として最小値の2が使用されることに注意してください。 例: 検出閾値=20ならば、 HYST_50=10(20の50%) HYST_25=5(20の25%) HYST_12_5=2(20の12.5%) HYST_6_25=2(20の6.25%=1ですが、強制限界の2が設定されます。
recal_threshold_t	RECAL_100 RECAL_50 RECAL_25 RECAL_12_5 RECAL_6_25	感知器再校正閾値。これは感知器検出閾値の%として表されます。 RECAL_x=再校正閾値は検出閾値のx%です(切り捨て丸め)。 注: 最小値は2が使用されます。 例: 検出閾値=40ならば、 RECAL_100=40(40の100%) RECAL_50=20(40の50%) RECAL_25=10(40の25%) RECAL_12_5=5(40の12.5%) RECAL_6_25=4(40の6.25%=2ですが、値は4に制限されます。
resolution_t	RES_1_BIT RES_2_BIT RES_3_BIT RES_4_BIT RES_5_BIT RES_6_BIT RES_7_BIT RES_8_BIT	回転部と摺動部に関して報告される角度と位置の分解能。 RES_x_BIT=回転部/摺動部はxビット値を報告します。 例: 摺動部の分解能がRES_7_BITならば、報告される位置は0~127の範囲です。

6.2.5. 全域接触感知状態

全域接触感知状態は表6-5.で一覧にされる変数を通して使用者応用のために利用可能です。

表6-5. 全域接触感知状態

変数	型	注記
qt_touch_status	qt_touch_status_t	ライブラリ感知器の状態

6.2.6. 全域接触感知形態設定

接触感知は表6-6.で一覧にされるパラメータで全体的に形態設定されます。

表6-6. 全体接触感知形態設定

変数	型	注記
qt_di	uint8_t	感知器検出積分(DI)限度。既定値:4
qt_drift_hold_time	uint8_t	200ms単位での感知器変動保持時間。 既定値:20(20×200ms=4s)、即ち検出抜け出し後4秒間変動を防ぎます。
qt_max_on_duration	uint8_t	200ms単位での感知器最大持続時間。 例:150=(150×200ms)30s後に再校正 0=再校正禁止 既定値:0(再校正禁止)
qt_neg_drift_rate	uint8_t	200ms単位での感知器負変動速度。 既定値:20(LSB毎に4s=20×200ms)
qt_pos_drift_rate	uint8_t	200ms単位での感知器正変動速度。 既定値:5(LSB毎に1s=5×200ms)
qt_recal_threshold	recal_threshold_t	感知器再校正閾値。 既定値:RECAL_50(再校正閾値=検出閾値の50%)

6.2.7. 接触感知データ

表6-7.で一覧にされるデータ配列はAPI内で利用可能です。これらはシステム開発中に接触感知が期待したように動いていることを調べるのに有用です。

表6-7. 接触感知データ配列

配列	要素型	注記
channel_signals[]	uint16_t	各チャンネルで測定した信号
channel_references[]	uint16_t	各チャンネルに対する基準信号
sensor_deltas[]	int16_t	(複数チャンネルから成るかもしれない)各感知器での信号差(Δ)

6.2.8. 使用者関数用引用

使用者が供給する濾波器関数用の引用点として“qt_filter_callback”関数ポインタが提供されます。この関数はライブラリが容量性測定を行った後、けれどもそれら进行处理する前に呼ばれます。使用者は測定した信号値に濾波器関数を提供するのにこの引用を使用することができます。

既定でのこのポインタはNULLで、関数は全く呼ばれません。

6.2.9. 感知器形態設定

6.2.9.1. 形態設定関数

表6-8.で一覧にされる関数は感知器にチャンネルを割り当てて感知器パラメータを形態設定するのに使用されます。

表6-8. 形態設定関数

関数	注記
qt_enable_key()	キー感知器を許可します。
qt_enable_rotor()	回転部感知器を許可します。
qt_enable_slider()	摺動部感知器を許可します。

6.2.9.2. qt_enable_key()

この関数はキー感知器を許可します。

```
void qt_enable_key (
    channel_t channel,
    aks_group_t aks_group,
    threshold_t detect_threshold,
    hysteresis_t detect_hysteresis );
```

パラメータは次の通りです。

- channel = キー感知器が使用する接触チャンネル
- aks_group = (どれかならば)感知器が入るAKS群
- detect_threshold = 感知器検出閾値
- detect_hysteresis = 感知器検出ヒステリシス値

キーに対応する感知器番号は感知器が許可された順番に依存します。最初に許可された感知器が感知器0で、2番目が感知器1、以下同様です。キーの現在の状況(ONまたはOFF)は"qt_touch_status.sensor_states"で調べることができます。

6.2.9.3. qt_enable_rotor()

この関数は回転部感知器を許可します。

```
void qt_enable_rotor (
    channel_t from_channel,
    channel_t to_channel,
    aks_group_t aks_group,
    threshold_t detect_threshold,
    hysteresis_t detect_hysteresis,
    resolution_t angle_resolution,
    uint8_t angle_hysteresis );
```

パラメータは次の通りです。

- from_channel = 回転部感知器内の最初のチャンネル
- to_channel = 回転部感知器内の最後のチャンネル
- aks_group = (どれかならば)感知器が入るAKS群
- detect_threshold = 感知器検出閾値
- detect_hysteresis = 感知器検出ヒステリシス値
- angle_resolution = 報告される角度の分解能値
- angle_hysteresis = 報告される角度のヒステリシス値

回転部に対応する感知器番号は感知器が許可された順番に依存します。最初に許可された感知器が感知器0で、2番目が感知器1で、以下同様です。回転部の現在の状況(ONまたはOFF)は"qt_touch_status.sensor_states"で調べることができます。回転部の値は"qt_touch_status.rotor_slider_value[]"です。どの配列の要素が使用されるかは感知器が許可された順番に依存し、最初に許可された回転部または摺動部が"rotor_slider_value[0]"を使い、2番目は"rotor_slider_value[1]"を、そして以下同様です。

報告される回転部値は回転部がONの時に有効です。

6.2.9.4. qt_enable_slider()

この関数は摺動部感知器を許可します。

```
void qt_enable_slider (
    channel_t from_channel,
    channel_t to_channel,
    aks_group_t aks_group,
    threshold_t detect_threshold,
    hysteresis_t detect_hysteresis,
    resolution_t position_resolution,
    uint8_t position_hysteresis );
```

パラメータは次の通りです。

- from_channel = 摺動部感知器内の最初のチャンネル
- to_channel = 摺動部感知器内の最後のチャンネル
- aks_group = (どれかならば)感知器が入るAKS群
- detect_threshold = 感知器検出閾値
- detect_hysteresis = 感知器検出ヒステリシス値
- position_resolution = 報告される位置の分解能値
- position_hysteresis = 報告される位置のヒステリシス値

摺動部に対応する感知器番号は感知器が許可された順番に依存します。最初に許可された感知器が感知器0で、2番目が感知器1で、以下同様です。摺動部の現在の状況(ONまたはOFF)は"qt_touch_status.sensor_states"で調べることができます。摺動部の値は"qt_touch_status.rotor_slider_value[]"です。どの配列の要素が使用されるかは感知器が許可された順番に依存し、最初に許可された回転部または摺動部が"rotor_slider_value[0]"を使い、2番目は"rotor_slider_value[1]"を、そして以下同様です。報告される摺動部値は摺動部がONの時に有効です。

6.2.10. 接触状態の測定と調査

6.2.10.1. 接触状態関数

必要とした全てのチャンネルがキー、回転部、摺動部として一旦形態設定されてしまうと、接触感知器は“qt_init_sensing()”関数を呼ぶことによって初期化されます。使用者応用はその後にパラメータとしてmsでの現在時間を渡して“qt_measure_sense()”関数を呼ぶことによって接触測定を実行することができます。ライブラリはどれ程長く感知器が検出されていたかを計算することのような時間の事象に関してこの情報を使用します。

“qt_measure_sensors()”呼出し後、使用者応用は“qt_touch_status”変数を読むことによって許可された感知器の状態を調べることができます。使用者応用はどの使用者接触も即座に検出され、どんな環境的な変化も流されるように定常体制で“qt_measure_sensors()”を呼ぶべきです。

6.2.10.2. 追加感知命令

“qt_init_sensing()”と“qt_measure_sensors()”の関数に加えて、使用者応用のために利用可能な2つの追加接触感知命令があります。これらは“qt_calibrate_sensing()”と“qt_reset_sensing()”の関数です。

6.2.10.3. qt_init_sensing()

この関数は接触感知を初期化します。

```
void qt_init_sensing( void );
```

必要とされるどの感知器もこの関数を呼ぶ前に(適切な“qt_enable_xxx()”関数を使用して)許可されなければなりません。この関数は内部ライブラリ変数を計算して接触チャンネルを形態設定し、そしてこれは“qt_measure_sensors()”を呼ぶ前に呼ばれなければなりません。

6.2.10.4. qt_mesure_sensors()

この関数は許可された全ての感知器で容量性測定を実行します。各感知器に対して測定された信号はその後に使用者の接触、開放、回転部角度の変更、摺動部位置の変更などを調べるために処理されます。

```
void qt_measure_sensors( uint16_t current_time_ms );
```

パラメータは次の通りです。

- current_time_ms = msでの現在時間

許可された全ての感知器の現在の状態は“qt_touch_status”構造体で報告されます。この関数を呼ぶ前に、(適切な“qt_enable_xxx()”関数を使用して)1つ以上の感知器が許可されて、“qt_init_sensing()”が呼ばれていなければなりません。

6.2.10.5. qt_calibrate_sensors()

この関数は許可された感知器の再校正を強制します。これは例えば応用の動作形態変更での全感知器の全体的な再校正を望む場合に有用であり得ます。

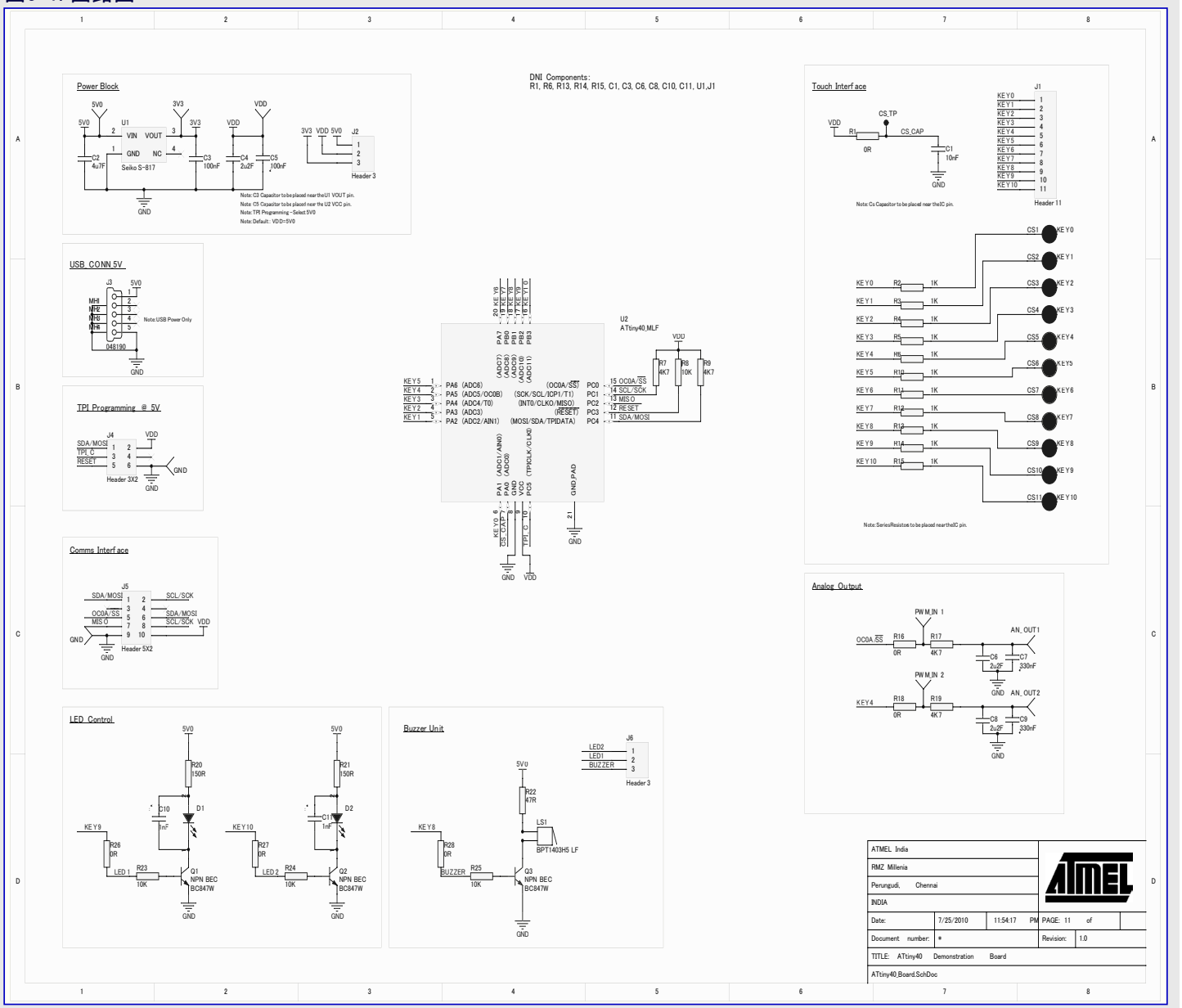
```
void qt_calibrate_sensing( void );
```

6.2.10.6. qt_reset_sensing()

この関数は全ての感知器を禁止して全てのライブラリ変数(例えば、“qt_di”)をそれらの既定値にリセットします。これは動的再構成設定感知を望む場合に有用であり得ます。この関数呼び出し後、必要とされるどの感知器も再許可され、再び“qt_measure_sensors()”が呼ばれる前に“qt_init_sensing()”が呼ばれなければなりません。

6.3. 回路图

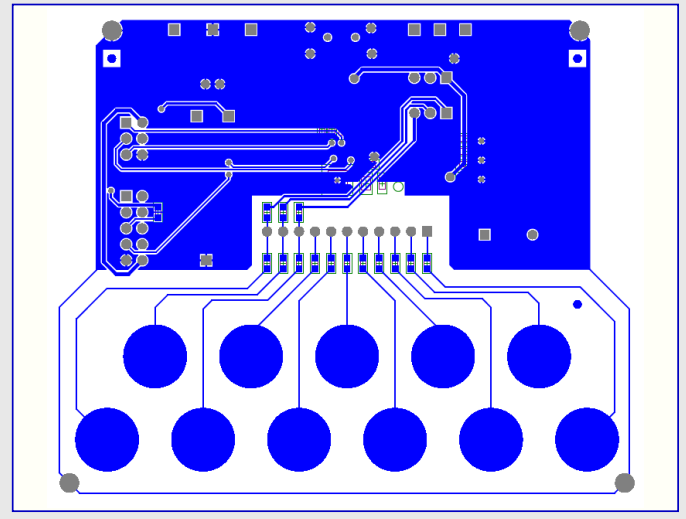
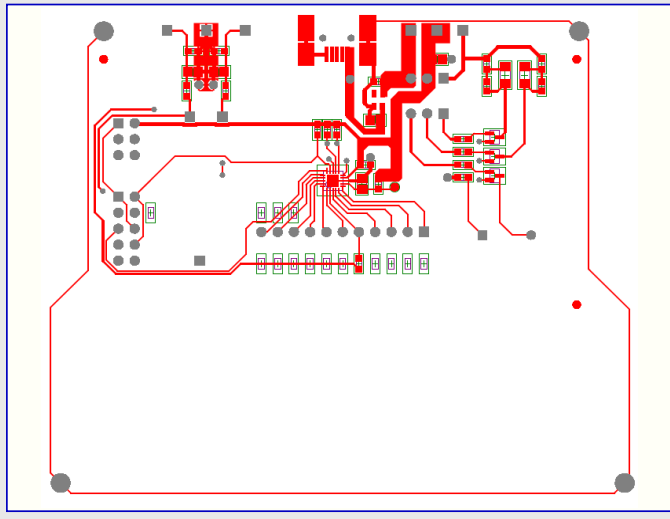
图6-1. 回路图



6.4. PCB

图6-2. 表面层

图6-3. 裏面層



6.5. 部品表

表6-1. 実装部品

通番	説明	供給社	供給社部品番号	指示子	数量
1	セラミック 4.7 μ F 16V 20% 0805 X5R			C2	1
2	セラミック 2.2 μ F 16V 20% 0805 X5R			C4	1
3	セラミック 0.1 μ F 16V 5% 0603 X7R			C5	1
4	セラミック 0.33 μ F 16V 5% 0603 X7R			C7,9	2
5	コネクタ ヘッド 3 \times 1 3ピン 2.54mm			J2	1
6	コネクタ ヘッド 3 \times 2 6ピン 2.54mm			J4	1
7	コネクタ ヘッド 5 \times 2 10ピン 2.54mm			J5	1
8	コネクタ USBミニB SMD	Würth Elektronik	65100516121	J3	1
9	LED 赤 1206 SMD	Everlight	15-21SURC/S530-A2/TR8	D1,2	2
10	プサナー 4KHz \pm 0.5 3.3VP-P	Bestar Acoustic	BPT1403H5LF	LS1	1
11	BC847B NPN SOT-23	NXP	BC847B	Q1,2,3	3
12	抵抗 0 Ω 50V 5% 0603			R16,18,26,27,28	5
13	抵抗 47 Ω 50V 5% 0603			R22	1
14	抵抗 150 Ω 50V 5% 0603			R20,21	2
15	抵抗 1k Ω 50V 5% 0603			R2,3,4,5,10,11,12	7
16	抵抗 4.7k Ω 50V 5% 0603			R7,9,17,19	4
17	抵抗 10k Ω 50V 5% 0603			R8,23,24,25	4
18	ATtiny40 20ピン MLF	ATMEL	ATtiny40-MMH	U2	1

表6-2. 未実装部品

通番	説明	供給社	供給社部品番号	指示子	数量
1	セラミック 2.2 μ F 16V 20% 0805 X5R			C6,8	2
2	セラミック 0.1 μ F 16V 5% 0603 X7R			C3	1
3	セラミック 0.001 μ F 16V 5% 0603 X7R			C10,11	2
4	抵抗 0 Ω 50V 5% 0603			R1	1
5	抵抗 1k Ω 50V 5% 0603			R6,13,14,15	4
6	コネクタ ヘッド 11ピン 2.54mm			J1	1
7	CMOS電圧調整器IC	SEIKO	S-817A33ANB-CUW-T2	U1	1

6.6. 参照

1. ATtiny40データシート
<http://www.atmel.com/Images/doc8263.pdf>
2. QTouchライブラリ
<http://www.atmel.com/qtouchlib>
3. 接触感知器設計の手引き
http://www.atmel.com/dyn/products/app_notes.asp?family_id=697
4. AVR042: AVRハードウェア設計の考察
<http://www.atmel.com/Images/doc2521.pdf>
5. AVR130: AVRタイマの初期設定と使い方
<http://www.atmel.com/Images/doc2505.pdf>

7. 評価基板/キット重要通知

この評価基板/キットは**工作、開発、実船を促進する、または評価目的だけ**の使用を意図されています。これは完成された製品ではなく、(基板/キットに於いて他の方法で注記されるかもしれないのを除き、)リサイクル(WEEE)、FCC、CE、またはULの電磁適合性に関連する制限や指令なしで完成製品へ応用できる、含めることの何かまたは何れかの技術的または法律上の必要条件に(未だ)適合しないかもしれません。ATMELは販売者と更にその先の使用者**単独の危険**に於いて、**全ての障害と共に何の保証もなく**、“現状そのまま”でこの基板/キットを供給しました。使用者は商品の適切で安全な取り扱いのために**全ての義務と責任を負います**。また使用者は商品の使用や取り扱いから起こる**全ての請求からATMELを保護**します。製品の開放構造のため、**静電放電と他のどんな技術的または法的な利害関係**に関して何れか若しくは**全ての適切な予防処置**を取るのは**使用者の責任**です。

上で述べる保障の範囲までを除き、**使用者とATMELは間接、特別、付帯的、または必然的な損害**に関して**互いに責任がない**でしょう。

そのようなATMELの製品やサービスがあるかもしれない、または使用されることに於いて、**どんな機械、処理、または組み合わせ**に関連または網羅するATMELの**どんな特許権**や他の**知的財産**の下でも**承諾は全く受けられません**。

郵便住所: Atmel Corporation, 2325 Orchard Parkway, San Jose, CA 95131



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

© 2012 Atmel Corporation. 全権利予約済

ATMEL[®]、ロゴとそれらの組み合わせ、それとAdjacent Key Suppression[®]、AKS[®]、AVR[®]、AVR Studio[®]、QTouch[®]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 2013.

本応用記述はATMELのAVR259応用記述(doc8332.pdf Rev.8332B-03/12)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。