

AVR252 : TV制御接触キーボード

要点

- ATtiny48/88中心のQTouch[®]設計
 - ・ QTouchハードウェアと提携する回路
 - ・ Atmel QTouchライブラリを使って独自化されたファームウェア
- 6つの容量性接触キー
- パネル厚 : 最大50mmの硝子または20mmのプラスチック (電極の大きさとCsに依存)
- 全てのキーに対するアナログ出力
- キー接触での警報告知
- 起き上がり機能
- ISPプログラミング
- デバッグ支援
- 3.3Vと4.2Vでの動作
- 特許と登録商標
 - ・ QTouch[®] (充電移転法特許)
 - ・ QMatrix[®] (充電移転法特許)
 - ・ AKS[®](隣接キー抑制[®]特許)技術

1. 序説

TV制御接触キーボードは既存の電気機械(抵抗性梯子)式TV制御キーボードに対するちょっとした置き換えのために設計されています。これは人-機械/使用者インターフェースに関する最新の容量性接触に基づきます。TV制御応用のために設計されましたが、6,7個の接触キーを持つ他のどんな応用にも使うことができます。接触キーボードは外部的に供給される供給(VCC)電圧に基づいて、どのキーの接触でも各々のアナログ(DC)出力を出力します。

図1-1.と図1-2.は部品と共に組み立てられたAVR252基板の各々、部品側と接触キーパッド側を示します。示されるように、これはON/OFF、TV/AV、PROG+、PROG-、VOL+、VOL-、MENU (MENUは物理的な接触キーではありませんが、VOL+とVOL-のキーを共に同時に接触することがメニューキーとして動きます。)のような様々なTVパラメータを制御する7つのキーを持ちます。

図1-1. PCBA部品側視

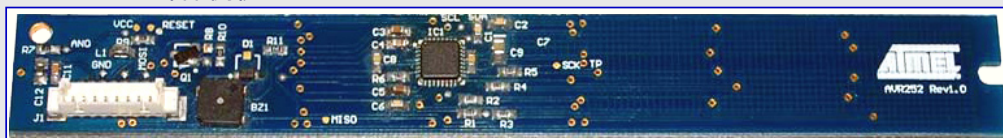


図1-2. PCBA接触キーパッド側視



8ビット **AVR[®]**
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. 8264A-09/09, 8264AJ3-03/21

2. 接触キーボード - ハードウェア

システム構成図は下の図2-1.で図解されるとおりです。制御キーパッドはキー接触を示すための基板上の警報器、予め定義された期間の間にキーが全く操作されない時のパワーダウン動作形態からの回復のための起き上がり機能を持ちます。アナログ出力(ANO)はそれによって各々のキー接触での決定をするTV制御器またはホスト制御器のA/D変換器(ADC)に供給されます。キーボードは8ピンコネクタで利用可能なプログラミングとデバッグ用インターフェースと検査点を持ちます。図2-2.は接触制御基板上の重要な部品とキーの配置を示します。

図2-1. システム構成図

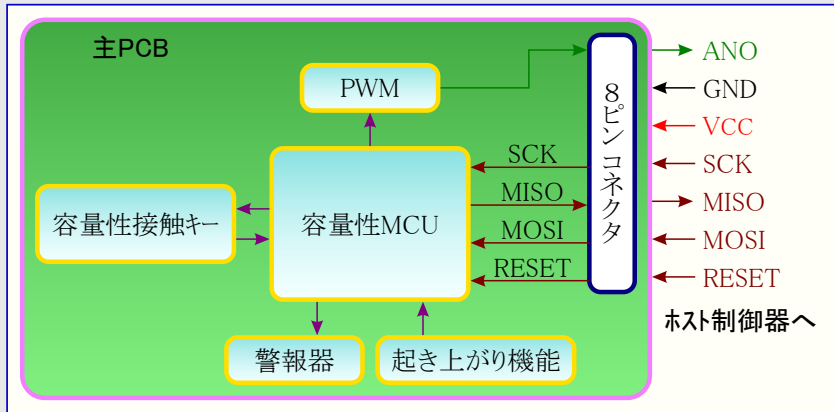
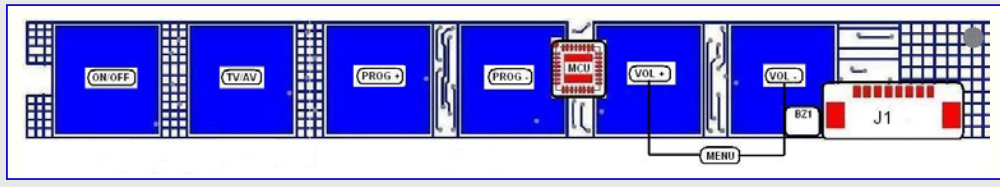


図2-2. キーボード配列と重要な部品配置



この応用記述の最後の追補に含まれる図5-1.はTV制御接触キーボード用の回路図を示します。キーボードは外部的に良く調整された3.3Vまたは4.2Vの供給電圧で給電されなければなりません。

警告: 通常動作中、裏面は絶縁材料と露出電極を使って絶縁されることであり、ピアは電気的接触に接触または作られてはなりません。上の必要条件なしでの通常動作は保証されません。

2.1. QTouch

QTouchデバイスは未知の容量の感知電極を既知の電位へ充電します。電極は代表的に印刷回路基板上の銅箔領域です。充電の結果は測定回路内へ転送されます。1つ以上の充電移転周回後の充電を測定することによって、感知面の容量を決めることができます。接触面上に指を置くことはその点での充電の流れに影響を及ぼす外部容量を持ち込みます。これは接触として記録します。

決定論理内の信号処理はQTouchを頑強かつ確かにします。静電気のスプイクまたは瞬間的な故意でない接触や接近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多数のキーを駆動することができます。複数キーが使われる場所では、各キーを個別の感性レベルに設定することができます。機能と美的の両方の必要条件に合わせるのに、異なる大きさや形状のキーを使うことができます。

優秀な電磁気適合のため、QTouch感知器はスペクトラム拡散変調と、集中と集中の間の長い遅延で乱的に散在した充電パルスを使います。個別パルスは相互集中パルスの空きの5%またはそれ未満程に短くすることができます。この策の恩恵は、より低い感知器間妨害、低減されたRF放射と感受性、低消費電力を含みます。

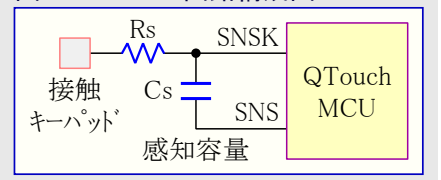
QTouchデバイスは経年や環境条件の変化のための緩やかな変化の責任を取るための自動変動補償が特徴です。それらは数10の動的範囲を持ち、コイル、発振器、RF部品、特殊ケーブル、RC網、または多数の個別部品が必要ありません。工学的解決策としてのQTouchは簡単、頑強、上品、そして手頃です。

多数の接触キーが共に近い場所では、指の接近が1つのキーよりも多くの周りの容量での変化を引き起こします。Atmel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、その結果と比較して使用者が接触を意図したのがどのキーかを決定、反復技術を使います。選択したキーからの信号が閾値レベル以上に留まる場合に、AKSは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可能です。この特許AKS隣接キー抑制の詳細については以下の「QTAN0031B:誤った接触入力無効化」応用記述を参照してください。

http://www.atmel.com/dyn/products/app_notes.asp?family_id=697

QTouchを使った容量性接触感知の一般的な概念は右で図解されます。

図2-3. QTouch回路構成図



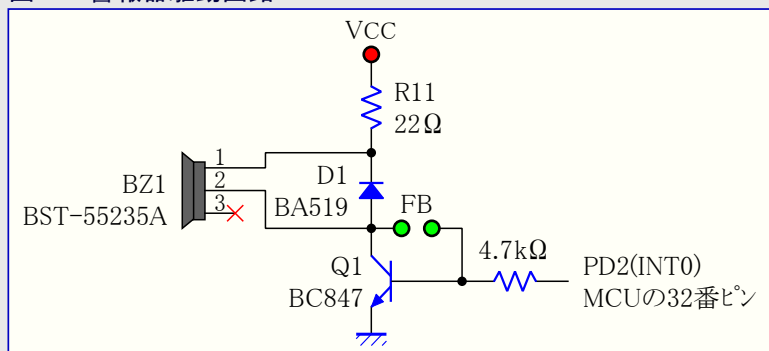
2.2. 感度

それらのチャネルでの感度を増すより大きな接触キーまたはより大きな感知容量のどちらかを使って、多重化されたチャネルでQTouch技術でのより高い接触感度を達成することができます。より大きな領域を持つ接触キーの使用または感知容量がより高い単位の場合に消費電力が増加されます。前面パネルまたは絶縁面で指の接触領域面よりも大きくない領域を持つ接触キーを使うことが推奨されます。これらの技法のどちらかを使った感度増加は雑音に対する感受性のような不利な影響も導き得ます。多重化されたチャネルに対して全てのチャネルに渡って標準化された感度の値を持つために、上で言及した理由のために、0.022μFの既存感知容量は0.047μFまで増やすことができます。感度調整とハードウェア変更を考慮に入れるために、閾値は基板ファームウェアで調節することもできます。これらの変更はコンパイル時間中に実施されるだけで、ファームウェア更新後にハードウェアで反映されます。

2.3. 警報器駆動部

どれかのキー接触の告知に基板上の圧電プザァーが使われます。警報器は何れかのキー接触で(2.2Vで4kHzで)30ms間だけ鳴ります。下の図2-4はAVR252基板上の活性な警報器駆動部の回路図を示します。警報器駆動部回路は容量性MCU ATtiny48/88のポートDのPD2であるMLF/QFN外周器の32番ピンによって駆動されます。R11抵抗は電流制限抵抗、一方D1は還流ダイオードとして使われます。

図2-4. 警報器駆動回路



2.4. 電力供給

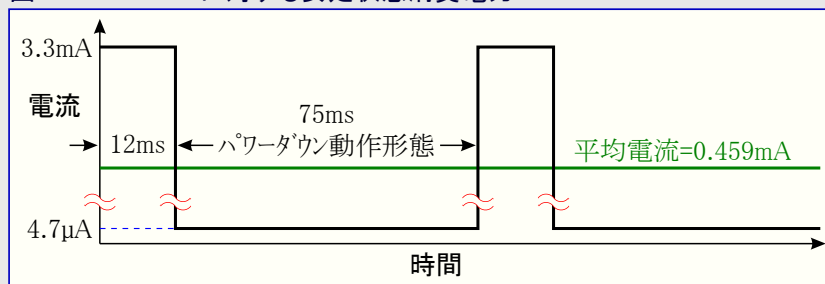
2.4.1. 電力供給の考慮

基板は3.3Vまたは4.2Vの良く調整されたDC供給電圧から外部的に給電されなければなりません。ATtiny48/88 MCUのVCCピンは0.1μFと4.7μFの値の2つの外部コンデンサを使って雑音分離(デカップ)されます。図5-1の回路図は電源での高周波数電流スパイクからの保護を提供することができる、電源と直列のフェライトビーズを示します。これは電源の波動と雑音が±25mVよりも小さく、負荷調整が基板の標準動作に対して1%を超えないことが推奨されます。供給電圧は外部的にVCCへ結ばれるAVCC基準ピンでの1.1V内部バンドギャップ電圧測定のため、始動中に周期的な間隔で10回採取されます。これらの10採取後に電源は監視されず、電源での変化は今後予期せぬ結果を導くかもしれません。この採取動作の結果はアナログ出力用の値の組を決めます。

2.4.2. 消費電力

ファームウェアは起き上がり機能を提供するためにATtiny48/88容量性MCUの電力管理機能を利用します。標準動作またはPWM静止中の電源からの電流流出は、RC受動低域通過(LP)濾波器応答に一致する117kHzで一定を保持されるPWM周波数としての接触されたキーの機能です。高周波数は物理的に小さな大きさの部品(コイル、コンデンサ)を許します、従って総システム費用を減らします。アイドル動作形態の間、または各容量性測定の組後に接触キーが全く接触されない時にMCUは75ms間パワーダウン休止形態へ移行し、起き上がりで容量性機能を再開します。パワーダウン休止形態中、非同期単位部だけが活動で、128kHzの周波数の独立した発振器で走行するウォッチドッグ タイマ(WDT)が標準動作状態で走行します。WDT割り込みが許可され、予め定義された期間後にWDT副システムは容量性機能を続けるためにMCUを起き上がらせる割り込みを生成します。追跡されている時間が起き上がりで更新されます。同様により大きな値の感知容量も集中期間中に電流流出に影響を及ぼします。

図2-5. VCC=3.3Vに対する安定状態消費電力



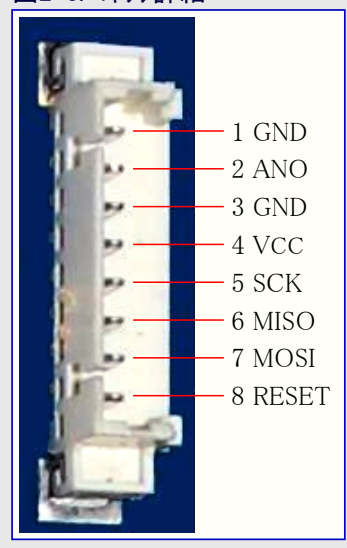
2.5. 基板コネクタ

図2-6.はTVとのインターフェースとプログラミングのために基板上で利用可能なJ1の8ピン コネクタの詳細を示します。

- GND 接地
- ANO アナログ出力
- GND 接地
- VCC 供給電圧
- SCK SPIクロック
- MISO 主装置入力従装置出力
- MOSI 主装置出力従装置入力
- RESET リセット

注: MOSI,MISO,SCKのピンが各々、チャンネル3,4,5と多重化されているため、ファームウェア更新のためのSPIプログラミング後に8線を持つ8ピン コネクタを取り去ることが推奨され、3線即ちANO,GND,VCC (換言すると2,3,4番ピン)だけを持つ8ピン コネクタの使用が望まれます。線はチャンネル上の浮遊容量効果を引き起こし得て、容量性測定結果に影響を及ぼすだけでなく、雑音も加え、結局これらのピンを使う許可されたチャンネルでの接触感度を減らします。

図2-6. コネクタ詳細

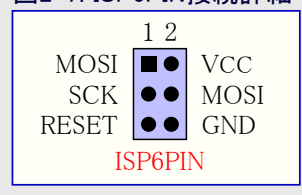


2.6. プログラミングとデバッグ

2.6.1. ISPプログラミング

基板上でのファームウェア更新はISPの設備と共に何れかの標準的なプログラミング ツールを使って行うことができます。図2-7.は1つ毎の信号割り当てを使うSPIポートと何れかのプログラミング ツールで基板を接続するのに必要とされる接続詳細を示します。

図2-7. ISP6PIN接続詳細



2.6.2. デバッグWIRE

チップ上デバッグ システムのデバッグWIREはCPUでのプログラムの流れ制御、AVR命令実行と各種不揮発性メモリをプログラミングするための単線、双方向インターフェースです。デバッグWIRE許可時、RESETポート ピンは許可されたプルアップを持つワイヤードAND(オープンドレイン)双方向入出力ピンとして構成設定され、目的対象とエミュレータ間の通信交換機になります。デバイスのデータシートで言及されるように、この使用方法に関する指針を保証してください。

3. アナログ出力

基板は各々のキーに対してアナログ(DC)電圧を出力することによってJ1(8ピン コネクタ)の2番ピンでホスト制御器に接触状態を伝えます。アナログ出力はRC濾波されたPWM信号です。表3-1.は両方の供給電圧(VCC)の値に対する様々なアナログ出力電圧を一覧にします。

表3-1. アナログ出力値

キー	アナログ出力 (VCC=3.3V)	アナログ出力 (VCC=4.2V)
アイドル状態	3.3V	4.2V
ON/OFF	2.9V	0.6V
TV/AV	2.2V	3.0V
PROG+	1.3V	1.5V
PROG-	2.5V	3.5V
VOL+	1.9V	2.5V
VOL-	0.3V	2.0V
MENU (注)	0.8V	1.1V

注: MENU機能を活性にするにはVOL+とVOL-の接触キーが共に接触されることが必要です。

図3-1.と図3-2.は両供給電圧(3.3Vと4.2V)に対するタイミング詳細と共にJ1の2番ピンでのPWM波形出力を示します。

アナログ出力信号ピンで観測される波形は2つの不時の使用者相互作用の事象(換言すると以下)によって告知されます。

- ・ 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

安定状態、換言すると接触されている接触制御キーが全くない時に、PWM出力は(継続)Highを保ちます。警報器はどれかのキーが接触された時から30msの期間、活性になります。PWM出力は警報器が活性な期間、換言するとどれかのキーが接触された時から30msの間Lowを保ちます。PWM出力は各々のキー、換言するとON/OFF、TV/AV、PROG+、PROG-、VOL+、VOL-、MENUに対する指定によって活性にして、どれかのキーが接触された30ms後からキーが接触されなくなるまで(無限最大ON期間)、RC低域通過濾波器を用いてDC電圧レベルを生成します。PWM出力はキーが接触されない時から不活性Highの安定状態値で再び安定します。

図3-1. VCC=3.3V用PWM波形

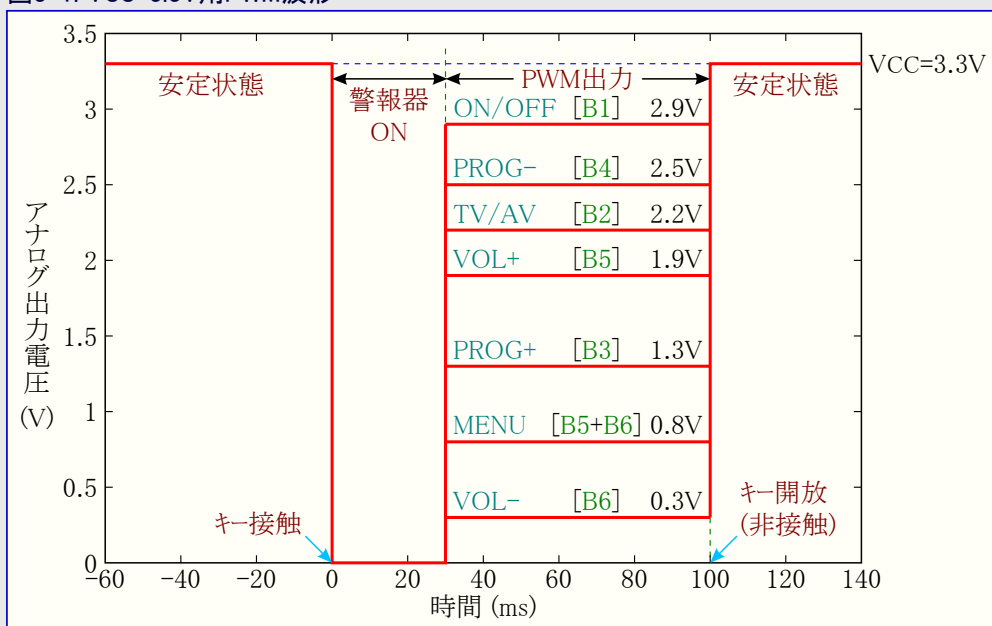
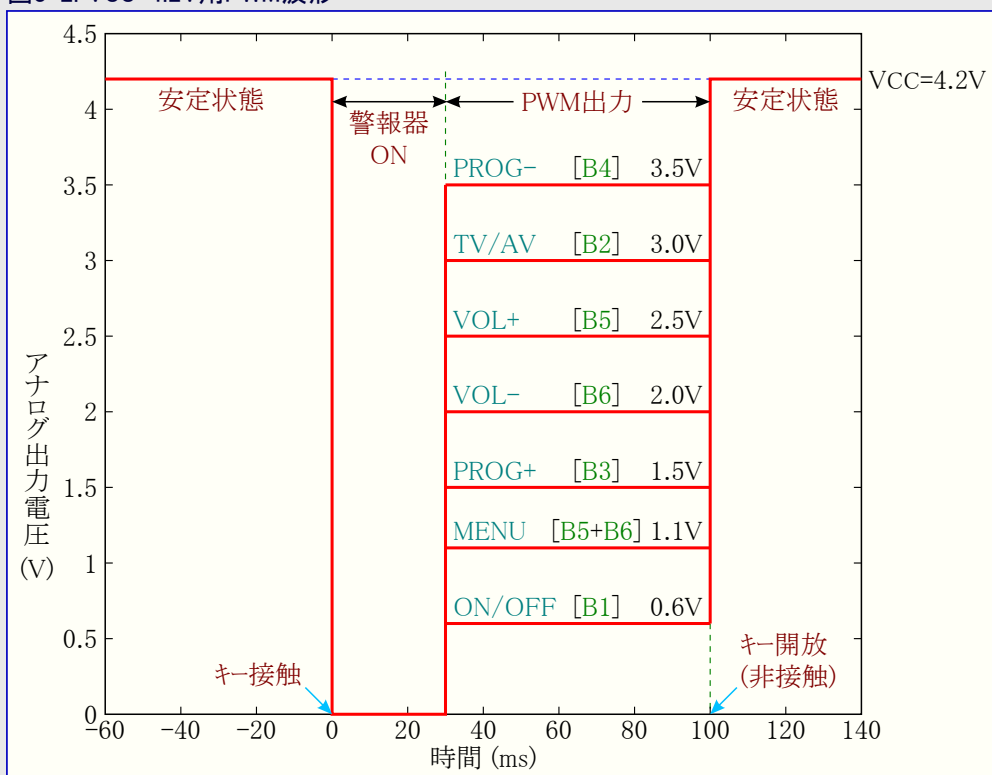


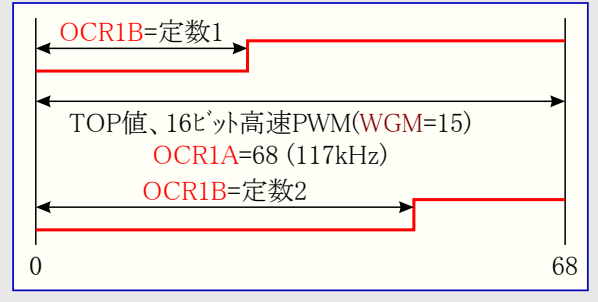
図3-2. VCC=4.2V用PWM波形



3.1. PWM信号の生成

PWM信号はATtiny48/88チップ上でPWM付きの16ビットのタイマ/カウンタ1を使って生成されます。図3-3は指定された周波数(117kHz)のPWM信号の生成に使われる各種レジスタ、動作形態、定数の値を示します。PWM信号生成に比較出力動作形態3(比較一致で設定(H)、TOPで解除(L))での波形生成動作形態15(16ビット高速PWM動作)が使われます。タイマ/カウンタ1のOCR1Bレジスタに設定するために、供給された電圧に依存して2つの参照表の1つが選択されます。各参照表は7つのキー機能の1つを示す位置指示子を使って指標化される表内の位置のPWM波形を生成するための定数を含みます。

図3-3. タイマ/カウンタ1を使ったPWMの生成



3.2. PWM参照表

システムクロックはATtiny48/88のヒューズによって設定される6CK/14CK+65nsとしての始動時間(パワーダウン/リセット)を持つチップ上の8MHz校正されたRC発振器です。

PWM周期をシステムクロック刻みの点から見ると、波形生成動作形態15(16ビット高速PWM)に関してTOP値はOCR1Aに対して使われず。

$$\begin{aligned} \text{PWM周期} &= (1/\text{PWM周波数})/(\text{システムクロック周期}) \\ &= (1/117\text{kHz})/(1/8\text{MHz}) \\ &= 68 \end{aligned}$$

OCR1BでのPWMチャネルBを使って、望むアナログ(DC)電圧の生成のために比較値として使われるべき定数は以下です。

$$\text{定数} = 68 - \frac{V \times 68}{V_{CC}}$$

上の方法を使って参照表は両方の動作電圧レベル、即ちVCC=3.3V(Low)とVCC=4.2V(High)に対して存在されます。

表3-2. PWM生成用参照表

VCC	低DC電圧 (VCC=3.3V)		高DC電圧 (VCC=4.2V)	
接触制御パラメータ	アナログ出力	定数	アナログ出力	定数
アイドル状態	3.3V	-	4.2V	-
ON/OFF	2.9V	8	0.6V	58
TV/AV	2.2V	23	3.0V	19
PROG+	1.3V	41	1.5V	44
PROG-	2.5V	16	3.5V	11
VOL+	1.9V	29	2.5V	28
VOL-	0.3V	62	2.0V	36
MENU	0.8V	52	1.1V	50

アイドル状態の間、ポートが出力として構成設定され、それによって設定される時に、何も落とすことなくPWM出力ピンで必要とされる安定したHighまたはLowのレベルを生じる標準ポート動作のために、比較出力動作はOC1A/OC1Bが切断される非PWM動作0に切り換えられます。

3.3. アナログ(DC)出力電圧に対するPWM分解能

PWM分解能はシステムクロック周波数、使う波形生成動作形態、クロック前置分周器、TOP値のような多くの要素に依存します。より高いクロック周波数は、この場合での117kHzの指定周波数に対して必要とされる与えられた時間の区間内で、より多くのクロック刻み数を許します。波形生成動作形態は8,9,10または16ビットのPWM分解能が達成できるかどうかを定義し、クロック前置分周器は予め決められた尺度でタイマ/カウンタクロックを分周することができ、そしてTOP値は各計数周期の最後を決めます。PWM出力の分解能は従って次のように計算することができます。

$$\text{分解能} = \frac{1}{68} \times V_{CC}$$

$$\begin{aligned} \text{分解能} &= 0.0485\text{V (VCC=3.3Vに対して)} \\ &= 0.0618\text{V (VCC=4.2Vに対して)} \end{aligned}$$

3.4. PWM出力に対する雑音余裕

アナログ出力に対する雑音余裕は最悪の筋書きを考慮して計算することができます。出力信号で利用可能な最小変量の33%の余裕にすると、基板設計に対して0.1Vの雑音余裕が得られます。図3-4.は最悪の考慮下での雑音余裕の計算の図形表現を示します。

図3-4. PWMアナログ出力に対する雑音余裕



4. ファームウェア

接触基板のファームウェアは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発動することができます。Atmel QTouchライブラリは容量性感知と後処理だけを行い、汎用入力の駆動、休止、通信などのような関数の残り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AKS(Adjacent Key Suppression、隣接キー抑制)を組み入れ、基板は容量性接触感知用の特許QTouch技術で設計されます。この応用記述はAtmel QTouchライブラリの詳細を徹底的に記述せず、このライブラリの使用者の手引きと共に使われなければなりません。

4.1. Atmel QTouchライブラリ

Atmel QTouchライブラリ1.1はリンクに必要なCヘッダファイルと共にIAR TMコンパイラで使うために利用可能な使用料無料の静的ライブラリである、予めコンパイルされた格納庫の形式で利用可能です。使用者応用はATtiny48/88を含む支援される容量性MCUの多くに対してAtmel QTouchライブラリの最上部で開発することができます。ライブラリは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、他のチップ資源を全く使いません。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んな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対して使われるAtmel QTouchライブラリはt88_8qt_BD_c5_k.r90です。

ライブラリファイルの命名法は次のとおりです。

“デバイス”_“チャンネル数”+“技術”_“SNSKポート+SNSポート”_“cN”_“kまたはkrs”.r90

- ・デバイス : デバイス名 (mはATmega、tはATtiny)
- ・チャンネル数 : 支援されるチャンネル数
- ・技術 : qt=QTouch、qm=QMatrix
- ・SNSKポート : 接触感知器のSNSK電極に使われるポート
- ・SNSポート : 接触感知器のSNS電極に使われるポート
- ・cN : NはQTouchに対する充電周回数、QMatrixに対する持続時間
- ・kまたはkrs : k=キーだけに対するライブラリ。krs=キー、回転部、摺動部に対するライブラリ。

Atmel QTouchライブラリを使うための全般的な流れは次のように記述することができます。

- ・使用者応用は(任意選択で)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“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構成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を構成設定するのに必要とされる、“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のに“qt_init_sensing()”を呼びます。
- ・その後、使用者応用は容量性測定をするために定期的に“qt_measure_channels()”を呼びます。各呼び出し後、どれかの感知器が検出されたかと、どれかの許可された回転部や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。

4.2. 応用プログラミング インターフェース

4.2.1. 明示定数

APIはライブラリに提供する表4-1.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4-1. 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

4.2.2. 型定義

APIは表4-2.で一覧にされる型定義を定義します。

表4-2. 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

4.2.3. 構造体

APIは表4-3.で一覧にされる構造体を使います。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4-3. 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。

4.2.4. 列挙

APIは表4-4.で一覧にされる列挙を使います。

表4-4. 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が使わ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が使われます。 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。

4.2.5. 全域接触感知状態

全域接触感知状態は表4-5.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4-5. 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

4.2.6. 全域接触感知構成設定

接触感知は表4-6.で一覧にされるパラメータで全体的に構成設定されます。

表4-6. 全体接触感知構成設定

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%)

4.2.7. 接触感知データ

表4-7.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4-7. 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)

4.2.8. 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うことができます。

既定でのこのポインタはNULLで、関数は全く呼ばれません。

4.2.9. 感知器構成設定

4.2.9.1. 構成設定関数

表4-8.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構成設定するのに使われます。

表4-8. 構成設定関数

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。

4.2.9.2. 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 );
```

パラメータは次のとおりです。

- channel = キー感知器が使う接触チャンネル
- 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。

4.2.9.3. 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 );
```

パラメータは次のとおりです。

- 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わ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。

4.2.9.4. 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 );
```

パラメータは次のとおりです。

- 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わ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。

4.2.10. 接触状態の測定と調査

4.2.10.1. 接触状態関数

必要とした全てのチャネルがキー、回転部、摺動部として一旦構成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います。

“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。

4.2.10.2. 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。

4.2.10.3. 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()”関数を使って)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ネルを構成設定し、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。

4.2.10.4. 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 );
```

パラメータは次のとおりです。

- 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()”関数を使って)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。

4.2.10.5. 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 );
```

4.2.10.6. 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。

4.3. 使用者応用コード

4.3.1. 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. 現在の時間を探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われます。
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。MCU制御レジスタ(MCUCR)のプルアップ禁止(PUD)ビットの設定(1)がこれを行います。
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充分なスタック量。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うことができます。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。

4.3.2. 応用コードの流れ

図4-1. は基板ファームウェアの流れ図を示します。

1. 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、プルアップの禁止、どの未使用資源も禁止と停止、それとデバッグインターフェースの初期化から成ります。
2. Atmel QTouchライブラリは“qt_enable_key()”を呼んで閾値、ヒステリシス、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。VOL+とVOL-のキーはMENU機能のために異なるAKS群に割り当てられます。
3. A/D変換器(ADC)は外部的にVCCに結ばれたAVCC基準を用いて1.1Vの内部バンドギャップ電圧を測定します。これはPWM生成用の2つの参照表の1つを選択するための供給電圧測定に使われます。
4. 規則的な動作で500msよりも大きな妨害または遅延を引き起こし、走行時に生成されるかもしれない処理されないどんな例外も処理するために、ウォッチドッグタイマがリセットされてシステムリセット動作が500ms間で許可されます。各測定周回後に通常動作がウォッチドッグをリセットし、それによって異常無し動作の場合にリセットなしで動作を続けます。
5. 使用者応用は許可された全てのチャネルを測定するために“qt_mesure_channels()”を呼びます。
6. 各キーは接触に関して調べられ、接触の場合に30ms間、次の状態に入ります。PWM出力は安定Lowを保たれ、警報器が4kHzの方形波生成によってONに切り換わります。
7. 30ms後にライブラリに対して現在の時間が更新され、警報器がOFFに切り換えられ、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VCCに関するPWM信号が生成されます。
8. キーが開放または非接触にされると、PWM出力は安定Highの非活性状態に設定されます。ライブラリに対して現在の時間が更新されます。
9. どのキーも現在接触されつづなければ、ウォッチドッグタイマがリセットされて75ms間、パワーダウン休止動作形態に移行されます。
10. パワーダウン休止動作形態からの起き上がりで、現在の時間が更新されて、同じ手順の流れが手順4.から繰り返されます。

以降の表4-10、表4-11、表4-12、表4-13. はマクロ変数、マクロ関数、外部全域変数、Atmel QTouchライブラリのAPIとファームウェアで使われる関数呼び出しを一覧にします。

図4-1. 基板ファームウェア用流れ図

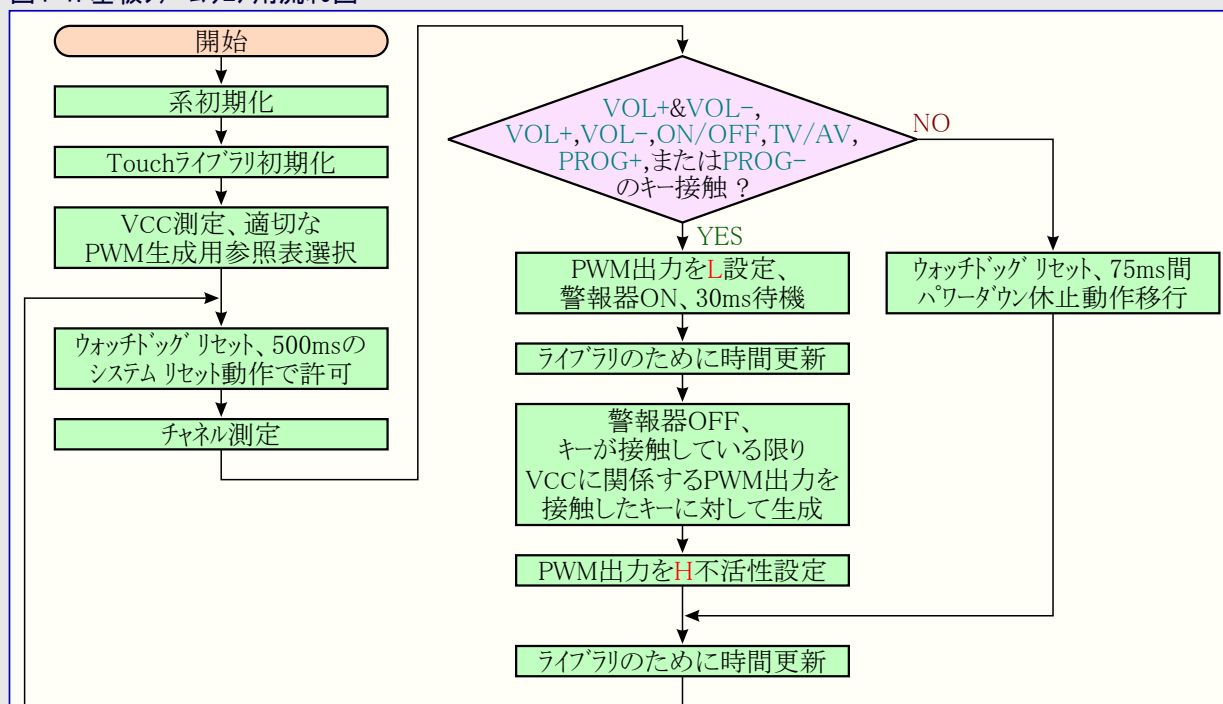


表4-10. ファームウェアで使われるマクロ変数

名前	値	説明
SNSK	B	各チャネルのSNSK電極に割り当てられるべきポート
SNS	D	各チャネルのSNS電極に割り当てられるべきポート
QT_DELAY_CYCLES	5	各集中期間に対する滞在時間
QT_NUM_CHANNELS	8	容量性測定チャネル数
QT_MAX_NUM_ROTORS_SLIDERS	0	回転部と摺動部の数
POWER_DOWN_PERIOD_MS	75	パワーダウン休止動作形態時間
POWER_ON_PERIOD_MS	20	測定と測定の間電源ON時間
QT_NUM_SENSOR_STATE_BYTES	1	感知器状態に必要とされるバイト数
QT_SNS_PORT	SNS	Atmel QTouchライブラリ用SNSポート
QT_SNSK_PORT	SNSK	Atmel QTouchライブラリ用SNSKポート
DBG_CLK_PORT	C	クロックに使われるべきデバッグポート
DBG_DATA_PORT	C	データに使われるべきデバッグポート
DBG_CLK_BIT	5	クロック用デバッグポートで使われるべきビット
DBG_DATA_BIT	4	データ用デバッグポートで使われるべきビット

表4-11. ファームウェアで使われるマクロ関数

名前	説明
JOIN	変数名を作成するために2つの字句単位(トークン)を接続
REG	字句単位(トークン)を使用してレジスタ名を作成

表4-12. ファームウェアで使われる外部全域変数

名前	説明
qt_touch_status	ライブラリ感知器の状態
channel_references	各チャネルに対する基準信号
sensor_deltas	(複数チャネルを含むかもしれない)各チャネルでの信号差(Δ)

表4-13. ファームウェアで使われるAPIと関数呼び出し

名前	説明
init_system	ホスト応用、ピン、ウォッチドッグなどを初期化
init_debug_if	デバッグ インターフェース初期化
qt_init_globals	全域的な閾値パラメータ初期化
qt_enable_key	キー感知器許可
qt_init_sensing	接触感知初期化
qt_measure_sensors	許可された全ての感知器で容量性測定実行
report_debug_data	ホストへデバッグ データを報告
output_to_debugger	デバッグ インターフェース上に複数バイト送信
send_debug_byte	デバッグ インターフェース上に1バイト送出

4.4. 応用コードへのライブラリ追加

1. IAR® Embedded Workbenchを開いて“t88_8qt_example_chip”作業空間を開いてください。
2. プロジェクトのルート ディレクトリ内へライブラリを複製してください。ライブラリ オブジェクト ファイルは“r90”の拡張子を持ちます。
3. プロジェクト上で右クリックし、ショートカット メニュー上でAddを選択し、その後にFilesを選択してください。
4. ライブラリに対して“r90”ファイルを選択してください。
5. ホストの必要条件によってmainファイルを変更してhexファイルを生成するためにコンパイルしてください。

5.3. 基板配置

図5-3. 表面

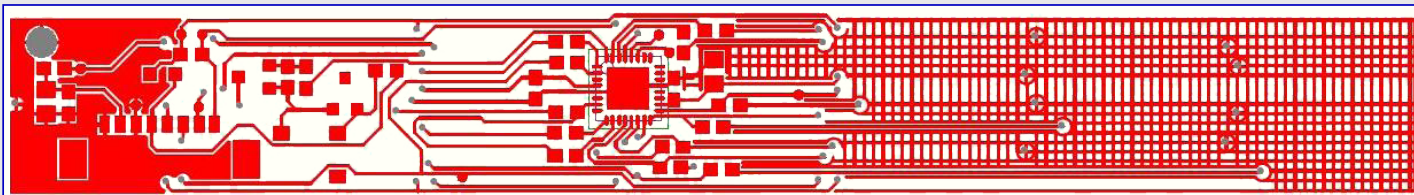
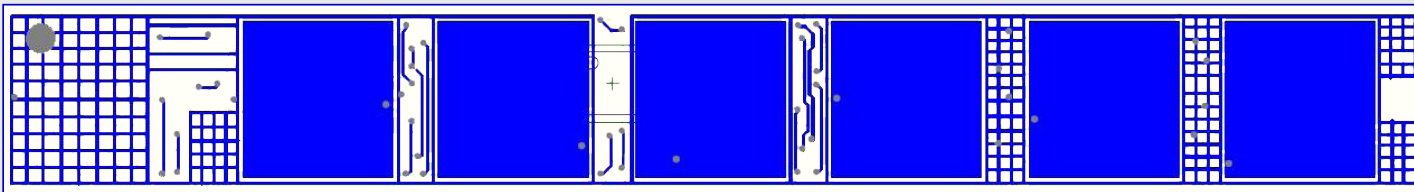


図5-4. 裏面



5.4. 部品表

表5-1. 実装部品

通番	説明	供給会社	供給会社部品番号	指示子	数量
1	抵抗 1kΩ 50V 1% 0603			R1,2,3,4,5,6	6
2	抵抗 330Ω 50V 5% 0603			R7	1
3	抵抗 10kΩ 50V 1% 0603			R9	1
4	抵抗 4.7kΩ 50V 5% 0603			R10	1
5	抵抗 22Ω 50V 1% 0603			R11	1
6	BC847B NPN SOT-23	NXP	BC847B	Q1	1
7	1A整流ダイオード	Fairchild	BAS19	D1	1
8	セラミック 0.022μF 50V 10% 0603 X7R			C1,2,6	2
9	セラミック 0.047μF 50V 10% 0603 X7R			C3,4,5	2
10	セラミック 4700pF 16V 10% 0603 X5R			C7	1
11	セラミック 0.1μF 50V 10% 0603 X7R			C8,9,11	3
12	セラミック 1μF 16V 10% 0603 X7R			C10	1
13	ブザー 4KHz±0.5 3VP-P	Bosan Hitech	BST-5523SA	BZ1	1
14	コネクタ 8P 1.25mmピッチ	MOLEX inc.	53398-0819	J1	1
15	ATtiny48/88 32ピン MLF	Atmel	ATtiny48/88	IC1	1
16	フレイトピース 0E			L1	1
合計					27

6. 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、実演を促進する、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、付带的、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、または使われることに於いて、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授けられません。

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

Copyright © 2008, Atmel Corporation



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 2009. 不許複製 Atmel®、Atmel®ロゴとそれらの組み合わせ、AVR®、AVR®ロゴ、QTouch®、QMtrix®、Adjacent Key suppression®、AKS®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AVR252応用記述(doc8264.pdf Rev.8264A-10/09)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。