

AVR476：接触遠隔制御

要点

- ATtiny88中心のQMatrix[®](容量性接触)設計
 - ・ QMatrixハードウェアと提携する回路
 - ・ 使用者インターフェースを処理するための独自化されたファームウェア
- 11個のQMatrixキー
 - ・ 背面照明付き4つのキー
- 8ビット分解能(256段階)を持つQWheel[™]
- 起き上がり機能
- (RF部とインターフェースする)SPIインターフェース
- 警報器と背面照明制御
- ISPプログラミング
- 特許と登録商標
 - ・ QMatrix (充電移転法特許)
 - ・ AKS[®](隣接キー抑制[®]特許)技術
 - ・ QWheel (充電移転法特許)

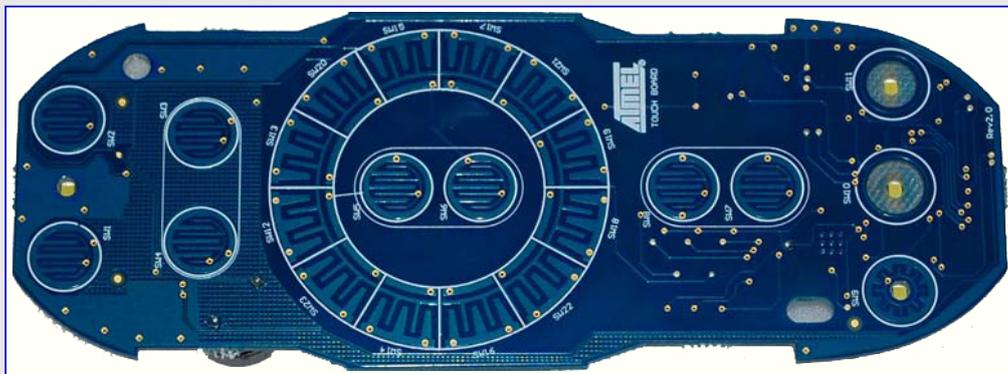
1. 序説

接触遠隔制御は人-機械/使用者インターフェース用に進化した容量性接触感知技術に基づいて独自設計された遠隔制御応用です。接触(遠隔制御)基板はSPIインターフェース経由でどんなホスト応用ともインターフェースすることができます。そのような応用の例は接触に基づくRF遠隔制御のためにRF単位部とインターフェースすることです。説明されたように、それは主に以下の2つの部分から成ります。

- ・ (接触制御と提携する回路を持つ)接触遠隔制御基板
- ・ (RF単位部、電源と提携する回路を持つ)RF基板

この応用記述は接触遠隔制御基板だけを扱い、ホスト(RF基板)を扱いません。

図1-1. 接触遠隔制御基板



この接触遠隔制御の代表的な応用は民生電子機器応用、マルチメディア装置、家電/自動装置、光制御システムなどで有り得ます。これらの応用での様々な機能は11個の接触キーを用いて切り換えることができ、同時にパラメータは(256段階の)接触輪で変更することができます。



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. 8251A-08/09, 8251AJ2-03/21

2. 接触基板 - ハードウェア

接触遠隔制御基板は対応するキー押下または輪の位置を検出するのに使われ、SPIバス経由でホストにキー/輪のデータを伝達します。容量性接触キーとQWheelはAtmel®の特許QMatrix技術に基づきます。システムは使用者入力用に11個の接触キーとQWheelを持ちます。QWheelは8ビットの分解能を持ち、輪上の接触位置に依存して\$00~\$FFの値を送り出します。

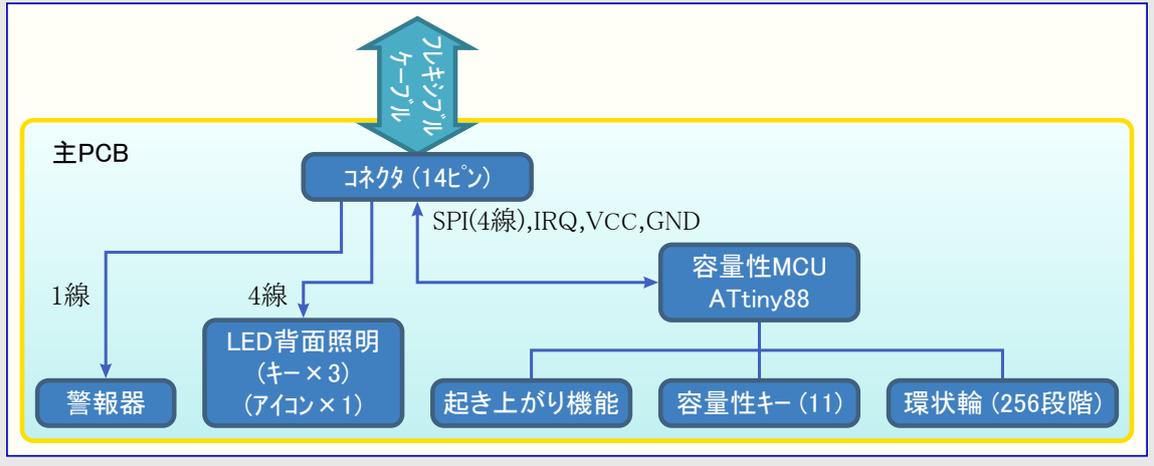
基板の一部である背面照明LEDと警報器は外部応用によって制御されます。使用者が接触したことを示すキーのために3つの背面照明LEDが提供されます。キーパッドの下付近をより良く光を拡散するために背面照明LEDに対して特別な凹部が提供されます。

キーの押下についてホストへ示すためにIRQが生成され、SPIバスが対応するキーを伝達します。

この基板は、より弱い応答を持つキーからの接触を抑制して優性なキーに接触の登録を許す、特許AKS(Adjacent Key Suppression、隣接キー抑制)が組み入れられます。この特許AKS(Adjacent Key Suppression)の詳細については以下で「QTAN0031B:誤った接触入力の無効化」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

図2-1. システム構成図



2.1. QMatrix

QMatrixデバイスは単一チップによって駆動される多数の接触キーを達成するための電極の組の受動配列の走査を使って接触を検知するように設計されたデジタル充電移転(QT)ICです。この構成設定は、各キーに対して1つの感知接続が必要なQTouch®のような感知技術と比べた時に、感知チップでの効率的な高ピン数に至り、チップとキー配列間の少ない接続数を許します。これらのデバイスは最低の部品数でキーに対して可能な最低の費用を達成します。

QMatrix回路はもの凄い信号対雑音比、高水準の耐水性フィルム、極端な水準の温度安定性、素晴らしい低電力特性、容易な配線、そして与えられたキー数に対して小さなIC外圍器の大きさを提供します。これらの理由のため、QMatrix回路は車載、台所用品、モバイルアプリに関して高く評価されます。

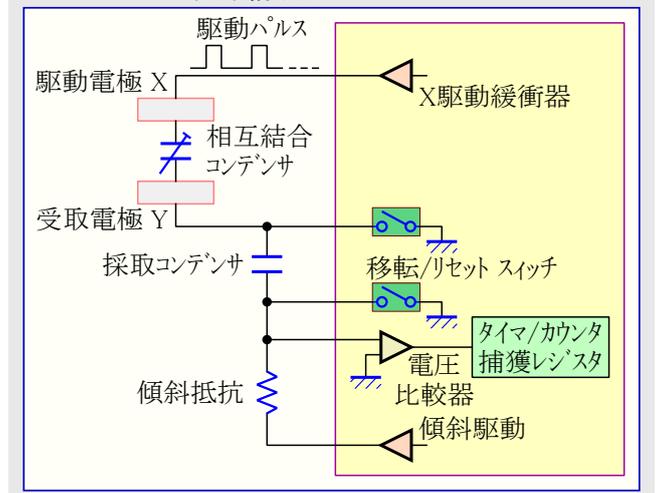
QMatrix技術の基本的な恩恵は以下のように要約することができます。

- チップのピン数に対して効率的な多いキー数
- 少ない部品数
- 容易な配線
- 安価な片面PCB材料に適合
- 温度安定性
- 自然の湿気の抑制
- 外部のRF域に対する高い耐性
- 低電力動作形態
- 透明な接触膜を作るのに使用可能
- キー当たりの非常に低い費用

照らされる、または背面照明されるキーは設計者にとって常に興味あるものです。LEDはPCBに穴を作ってLEDが載る面の周辺に電極を作ってそのパッドを配線することによって簡単にキーの中央に配置することができます。

QMatrixを使う容量性接触感知の一般的な概念は右で図解されます。

図2-2. QMatrix回路構成図



2.2. QMatrix電極設計

容量性接触感知は基準容量に比例する容量の感知を信頼します。接触前に測定した容量が大きければ、接触によって追加された容量(概ね2~5pF)は検出されないでしょう。

接触と非接触に対して信頼に足る信号の違いを得るのに真に十分に集中を感知するために、感知容量よりも相当量大きくなければなりません。感知容量に対する代表的な値は0.002~0.05 μ Fです。

この応用については、壁の照明スイッチに容易に合うように小さな接触電極が使われます。

感度を調整するために傾斜抵抗(Rsmp)を変更することができます。Rsmp増加は採取コンデンサ(Cs)の放電時間を長くし、従って放電傾斜はより浅くなってより良い分解能を提供します。代わりにより高い終了電圧を得るために集中長を増やすことができます。

集中と、従って電極上の接触回路からの計数はオシロスコープで見ることができます。接触と非接触での変化を見ることも可能です。

2.2.1. キー

XとYの電極は一般的に相互組み入れされ、それはそれらが指を組み合わせた形式です。代表的にX電極は2つが持つ領域を含むようにY電極を囲います。キーは9mmの直径を持ちます。

2.2.1.1. Y電極の幅

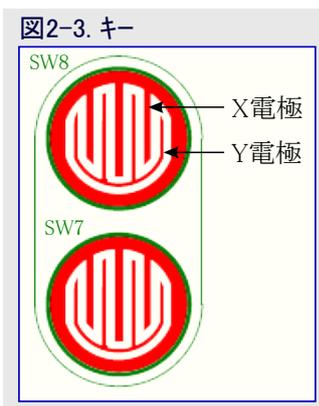
接触中に感知器に対する雑音結合の可能性をこれが最小にするため、Y電極は指状部に対して可能な最も細い(0.1~0.5mm)ものを使うべきです。

2.2.1.2. X電極の幅

X電極については、それらがY線に対する部分的な防御物として働くのに役立つため、一般的により広い電極が望まれます。Xの指状部の幅は上に被さるパネルの厚さ(T)から計算されるべきです。一般的に、関数T/2が用いられます。

2.2.1.3. 電極間の空間

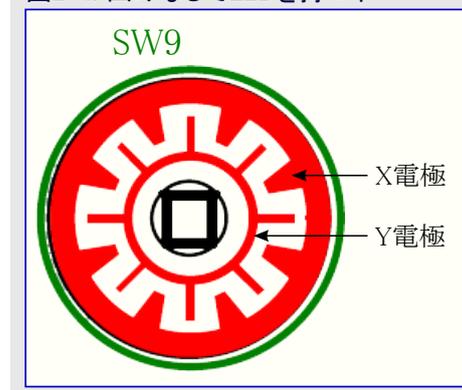
X指状部の幅として、XとYの電極間の空間はT/2であるべきです。



2.2.2. 凹みなしでLEDを持つキー

XとYの電極は相互組み入れされ、それはそれらが指を組み合わせた形式です。X電極は2つが持つ領域を含むようにY電極を囲います。穴は逆装着LEDのためにキーの中央で提供されます。この設計の目的はキーに対して割り当てられた空間の境界内でXとYの電極間の結合長を最大にすることによってSNRを最適化することです。多数の細い指状部は少数の幅のものよりもより長い結合長を意味し、それはより良いSNRに帰着します。

図2-4. 凹みなしでLEDを持つキー

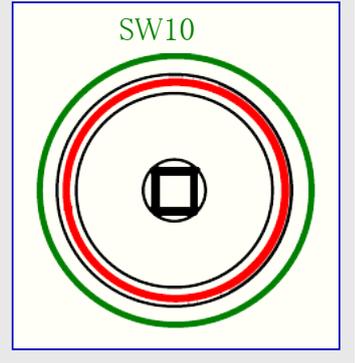


2.2.3. 凹み有りLEDを持つキー

キーは2つの基材の層に渡ってXとYの電極を分配する浸水X2層法を使います。この設計はより少ない空間でキーに対する感度を改善し、またX電極は後ろからY電極を完全に防御し、感知器は後ろからの接触感受性がありません。Y電極は4mmの直径を持ち、(裏側の)X電極は5mmの直径を持ちます。最大の結合と絶縁を提供するため、X電極の円周はY電極から両方の側で2mmです。

凹みはキーパッドの下付近の最大拡散光用に提供されます。2つの異なる大きさ、直径6.5mmと7.5mmが提供されます。

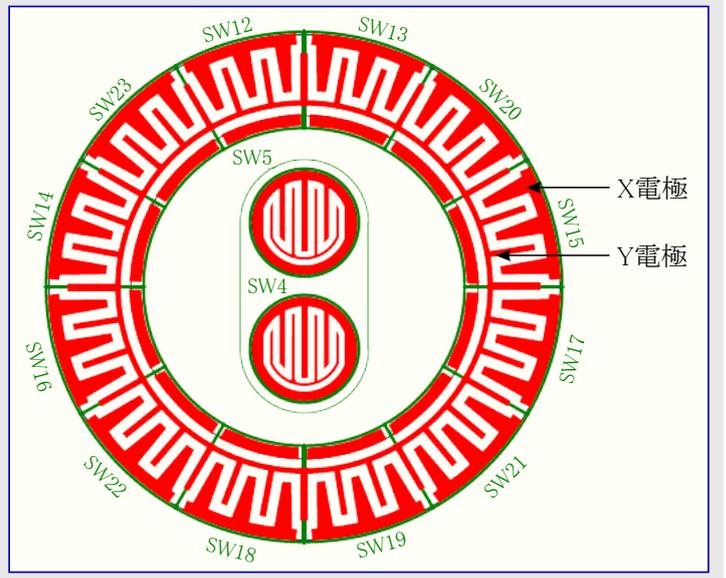
図2-4. 凹みとLEDを持つキー



2.2.4. QWheel

キーの配列は円形式で整列され、境界の終わりがありません。感知器の直径を増すために、X線上で抵抗分圧器を使って追加セグメントが作成されます。輪は4つのX線とY線に接続されます。これはより大きな量のキーを支援するために、12個の抵抗性内挿セグメントを持ちます。QWheelは44mmの外周直径と28mmの内周直径を持ちます。より良い電界の結合と制限のためにY電極はX電極内で防御されます。設計と空間の必要条件はキーと同様です。

図2-6. 中央キーを持つQWheel™



2.2.5. LED/警報器の設計

LEDが容量性感知器に近い(4mmよりも離れていない)場合、それらのON/OFF状態間の容量での変化が考慮されるべきです。それらの駆動回路の性質での変化も考慮が必要です。

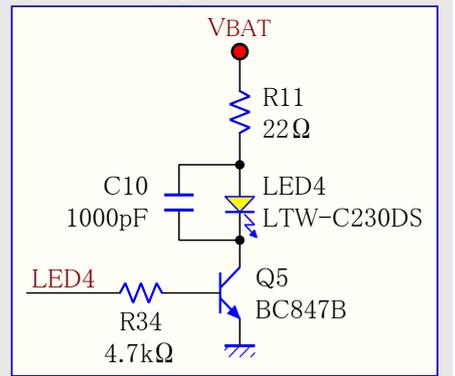
近いと判定されるLEDは代表的に1000pFの値を持つコンデンサで迂回されなければなりません。

迂回コンデンサがLEDそれ自身の物理的に近くでないことに注意してください。例えそのコンデンサがLEDから数cm離れていても、その目的のために働きます。これは感知器周辺空間に対して窮屈な配置を助け得ます。

背面照明LEDはキーの内で、従ってコンデンサが提供されます。LEDはトランジスタによって駆動され、図2-7はこのLEDに対する迂回コンデンサの配置を示します。

警報器もトランジスタによって駆動され、外部供給元によって駆動することができます。

図2-7. LED用迂回コンデンサ



2.3. 電力供給

2.3.1. VCCとVBAT

接触基板用のVCCは2.7～5Vです。この応用に関する動作VCCは3.3Vです。VCCはATtiny88デバイス、従って接触とSPIのインターフェースに給電します。

VBATは電池電圧で、基板上のLEDと警報器に給電します。

2.3.2. 電力供給の考慮

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探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うため、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)形式の調整器が使われるべきで、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わ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。

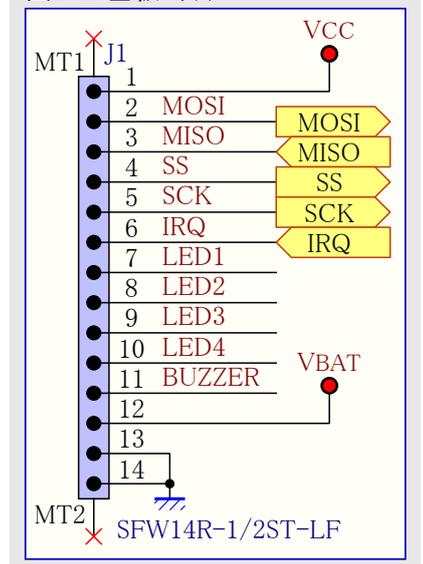
短い線の単独の0.1μFセラミック迂回コンデンサは供給ピンの非常に近くへ配置されるべきです。

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。

2.4. 基板コネクタ

14ピンのFPC/FFCコネクタがホストから接触基板への信号をインターフェースします。

図2-8. 基板コネクタ

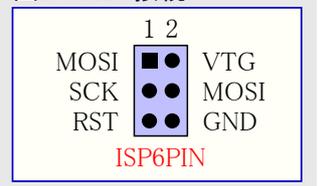


3. プログラミングとデバッグ

3.1. ISPプログラミング

AVR476はISPインターフェースを通してプログラミングすることができます。ISP用の配線のために基板の裏側で検査点が提供されます。

図3-1. ISP接続



3.2. 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。

4. 接触基板 – 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 of 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**ライブラリは標準Atmel AVRマイクロコントローラ上で接触応用を開発するための使用料無料のソフトウェアライブラリ(IAR™)です。使用者はそれらのプロジェクトに接触感知能力を提供するためにそれらのファームウェア内へライブラリをリンクすることができます。ライブラリは多くの制御応用に対する単一チップ解決策に、またはもっと複雑な応用でチップ数を減らすのに使うことができます。

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”_”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**
- 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。

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。

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

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。

4.2.2. 型定義

4.2.2.1. 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

4.2.2.2. 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。

4.2.2.3. 列挙

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	感知器内のチャンネル。
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。

4.2.3. 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

4.2.4. 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%)

4.2.5. チャネル単位接触感知構成設定

QMatrix技術に基づくライブラリの違いで、使用者応用に対して表4-7.で一覧にされるデータ配列が利用可能です。

表4-7. チャネル単位接触感知構成設定

配列	要素型	注記
qt_burst_length[]	uint16_t	各QMatrixチャネルでのパルス単位での集中長 既定値:64パルス

4.2.6. 接触感知データ

表4-8.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4-8. 接触感知データ配列

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

4.2.7. 使用者関数引用

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。

4.2.8. 感知器構成設定

表4-9.で一覧にされる関数は感知器にチャ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。

4.2.8.1. 構成設定関数

表4-9. 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。

4.2.8.2. 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。

4.2.8.3. 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。

4.2.8.4. 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。

4.2.9. 接触状態の測定と調査

4.2.9.1. 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。

ホスト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、ホスト応用は"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。

ホスト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。

4.2.9.2. 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。

4.2.9.3. 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ンネルを構成設定し、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。

4.2.9.4. 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_mesure_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。

4.2.9.5. 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 );
```

4.2.9.6. 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

使用者応用は正しく動作するために接触感知に関して以下のような判断基準に合致しなければなりません。

- 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。
- 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。
- 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、ホスト主応用コードは走行しません。従ってホスト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。
- 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。ホスト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。

4.3.2. SPI通信

SPI構成設定

1. 接触基板はSPI従装置です。
2. SCKはアイドル時にHighです。
3. データは後行端で採取されます。
4. データバイトはMSB先行です。

4.3.3. 変化線(IRQ)

変化線は有効な接触状態があることをSPI主装置に通知するのに使われます。

変化線はHigh論理活性で、従って接触状況に於いて有効な変化がある時にHighになります。

4.3.4. 休止

デバイスは測定と通信を実行する固定された時間の区間で動作し、その後に電力を保護するために周回の残りの間を休止します。休止に於いて、デバイスは測定が全く実行されないパワーダウン動作形態です。休止では125ms後にデバイスを起き上がらせるためにウォッチドッグ タイマが許可されます。デバイスはSPIのSS入力とハードウェア リセットからも起き上がらせることができます。休止での消費電力は3.3Vの電力供給で<5μAです。

4.3.5. 応用コードの流れ

1. 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"qt_reset_sensing()"を呼びます。
2. ホスト応用は接触感知器を構成設定するために"qt_enable_key()"と"qt_enable_rotor()"を呼びます。
3. ホスト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。
4. その後、ホスト応用は容量性測定を行うために定期的に"qt_mesure_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、許可された回転部と摺動部のどれかの角度または位置を見るために"qt_touch_status"全域変数を調べることができます。
5. その状態に於いてどれかの変化があった場合、SPI主装置に対して割り込みが開始され、デバイスは休止になります。
6. SSピンがLowになるためにピン変化割り込みが起こる場合、デバイスが起き上がり、SPIを通して要求されたデータを送ります。
7. 125msに構成設定されたウォッチドッグ タイマ割り込みが起こる場合、デバイスは起き上がり、チャンネルの測定を開始して手順4.からを続けます。

図5-1. 流れ図

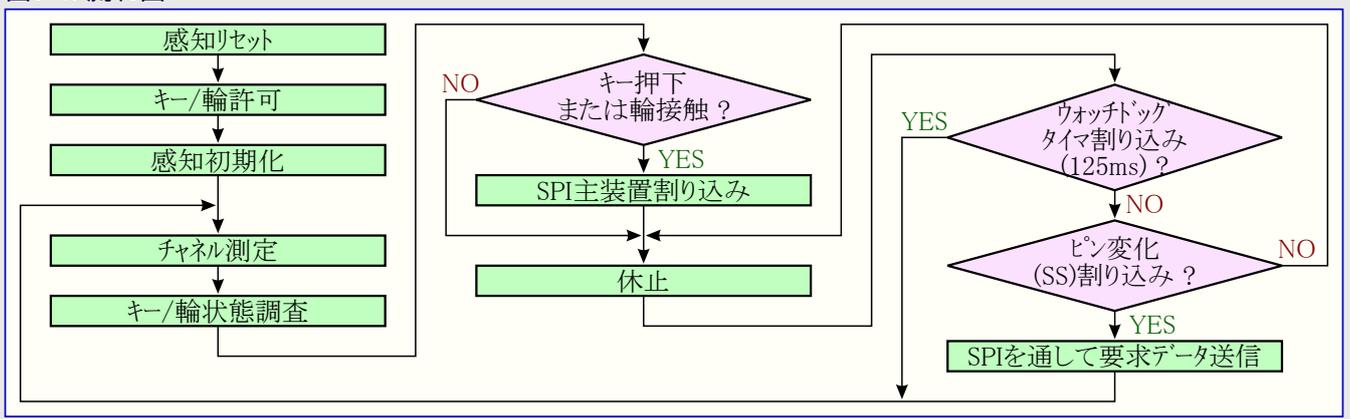


表5-1. SPIバイト

アドレス	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
0	規約ID (\$01)							
1	リセットビット	0	0	0	0	キービット 遮蔽有効	輪接触	接触ハット 接触
2	X位置 (予約)							
3	Y位置 (予約)							
4	輪位置							
5	0	0	キー5	キー4	キー3	キー2	キー1	キー0
6	0	0	キー13	キー12	キー11	キー10	キー9	キー8
7	主改訂番号				副改訂番号			

リセットビット:

リセット後の状態バイトの最初の読み込みは設定(1)されたこのビットを持ち、後続する読み込みは解除(0)されたこのビットを持ちます。

校正:

\$A5を送ってください。これは許可された感知器を校正して\$A5を送り返します。

キー (バイト5):

- キー0 : 濃さ(SATURATION)+
- キー1 : ON
- キー2 : OFF
- キー3 : 明るさ(BRIGHTNESS)-
- キー4 : 濃さ(SATURATION)-
- キー5 : 明るさ(BRIGHTNESS)+
- キー6 : (未使用、0として読めます。)
- キー7 : (未使用、0として読めます。)

キー (バイト6):

- キー8 : 次を選択
- キー9 : 直前を選択
- キー10 : 接触接続(AVR476で不使用)
- キー11 : 場面1
- キー12 : 場面2
- キー13 : 場面3
- キー14 : (未使用、0として読めます。)
- キー15 : (未使用、0として読めます。)

4.3.6. 応用コードへのライブラリ追加

1. IAR Embedded Workbench®を開いて“m88_32qm_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。

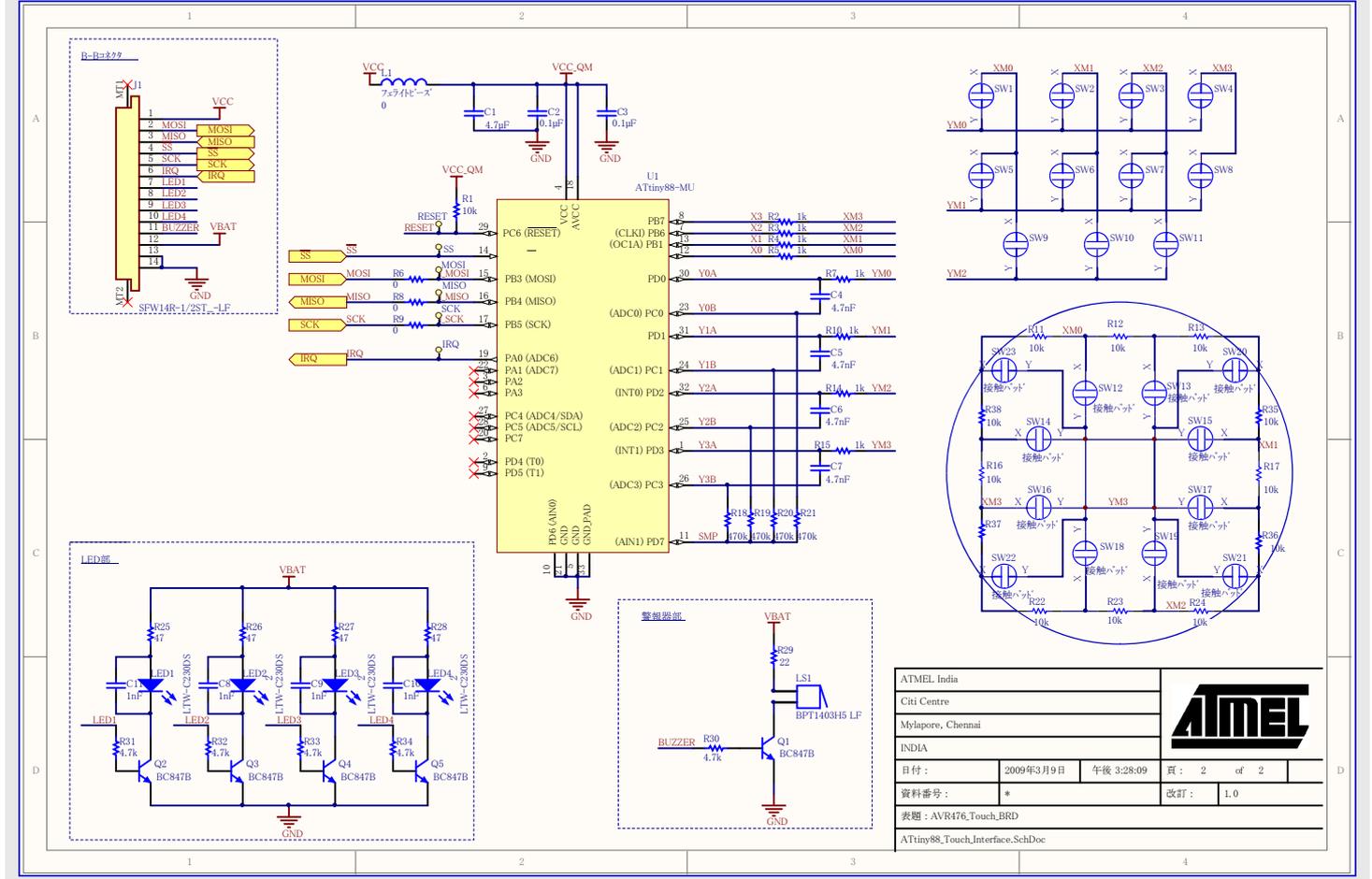
5. 追補

以降で以下の資料が示されます。

- 回路図
- 組立図
- 基板配置
- 部品表

5.1. 回路図

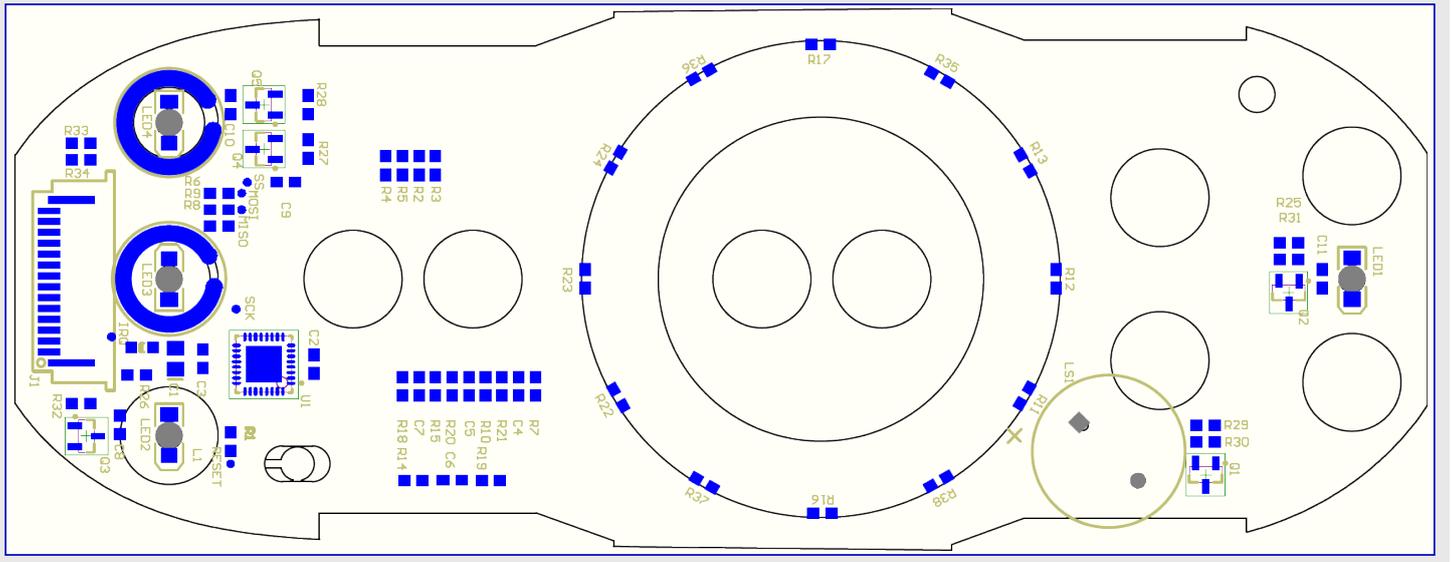
図5-1. 回路図



ATMEL India			
Citi Centre			
Mylapore, Chennai		INDIA	
日付:	2009年3月9日	午後 3:28:09	頁: 2 of 2
資料番号:	*	改訂:	1.0
表題: AVR476_Touch_BRD			
ATtiny88_Touch_Interface.SchDoc			

5.2. 組立図

図5-2. 部品組み立て(裏面)



5.3. 基板配置

図5-3. 表面

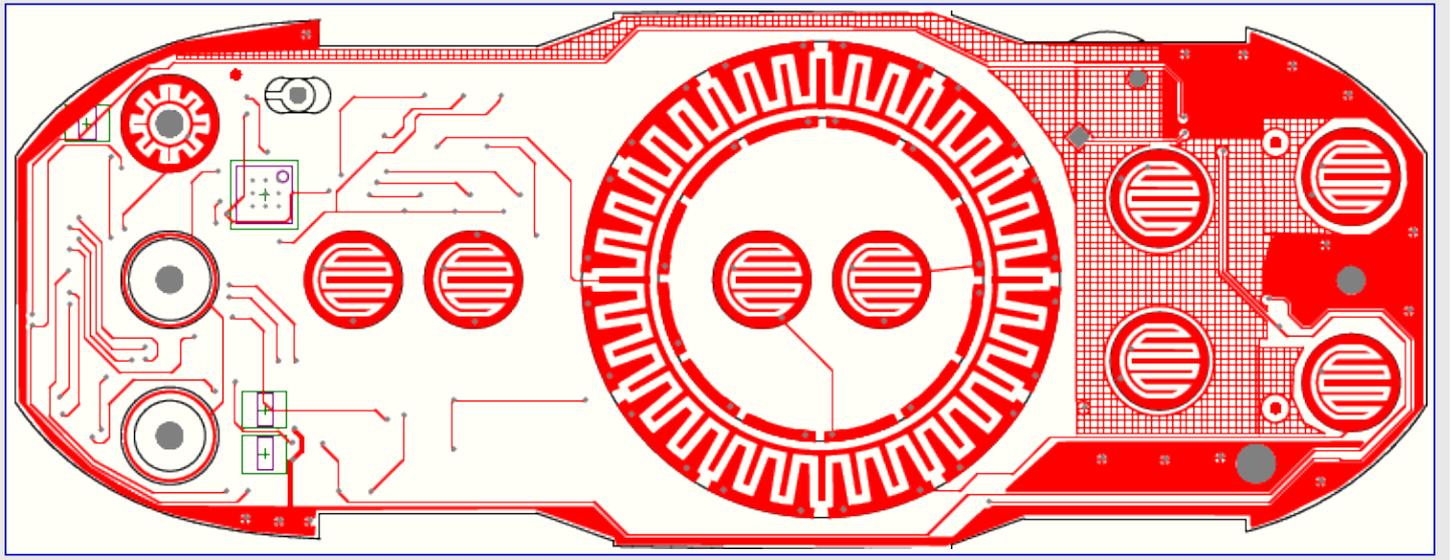
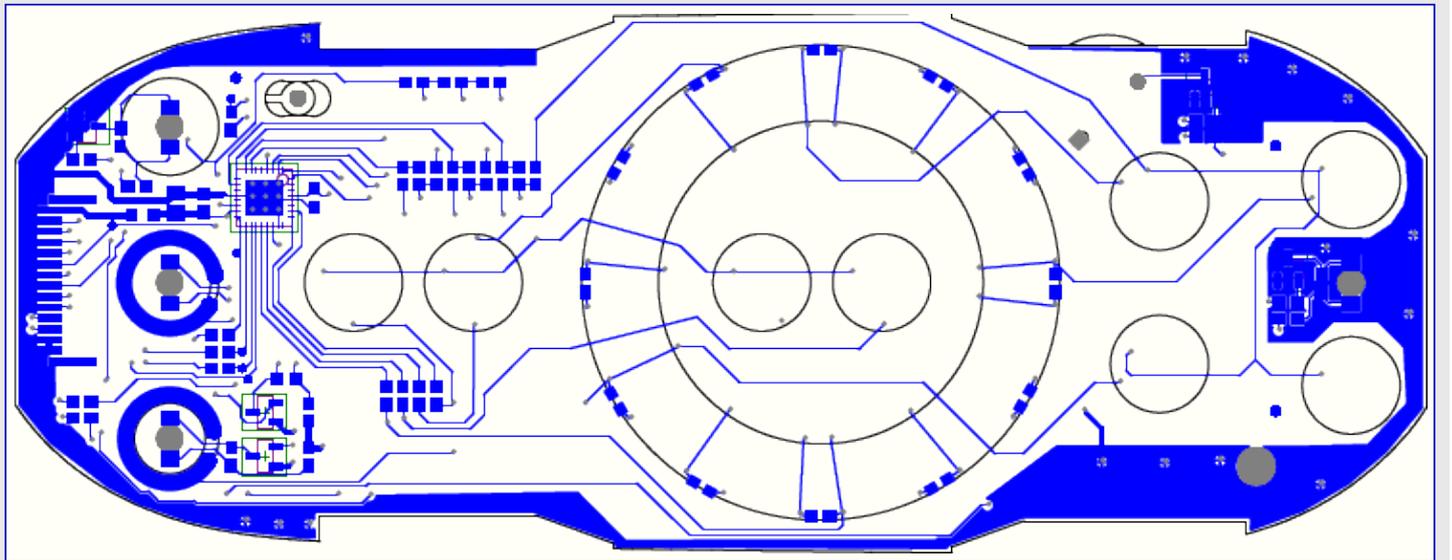


図5-4. 裏面



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			C1	1
2	セラミック 0.1 μ F 50V 5% 0603 X7R			C2,3	2
3	セラミック 8200pF 16V 2% 0603 NPO			C4,5,6,7	4
4	セラミック 1000pF 50V 5% 0603 X7R			C8,9,10,11	4
5	コネクタ FPC/FFC 14P+2 1mmピッチ	FCI	SW14R-2STE1-LF	J1	1
6	LED黄色 逆実装 1206	Liteon	LTW-C230DS	LED1,2,3,4	4
7	ブザー 4kHz \pm 0.5 3.3VP-P	Bestar Acoustic	BPT1403H5LF	LS1	1
8	BC847B NPN SOT-23	NXP	BC847B	Q1,2,3,4,5	5
9	フェライトビーズ			L1	1
10	抵抗 10k Ω 50V 5% 0603			R1	1
11	抵抗 1k Ω 50V 1% 0603			R2,3,4,5,7,10,14,15	8
12	抵抗 0 Ω 50V 5% 0603			R6,8,9	3
13	抵抗 10k Ω 50V 1% 0603			R11,12,13,15,17,22,23,24,35,36,37,38	12
14	抵抗 1M Ω 50V 1% 0603			R18,19,20,21	4
15	抵抗 47 Ω 50V 5% 0603			R25,26,27,28	4
16	抵抗 22 Ω 50V 5% 0603			R29	1
17	抵抗 4.7k Ω 50V 5% 0603			R30,31,32,33,34	5
18	ATtiny88 32ピン MLF	Atmel	ATtiny88-MU	U1	1

私たちのウェブサイトで利用可能な「[接触感知器設計の手引き](#)」を参照することも推奨されます。

6. 評価基板/キット重要通知

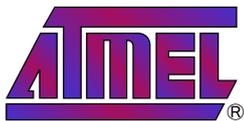
この評価基板/キットは**工作、開発、実演を促進する、または評価目的だけ**の使用を意図されています。これは完成された製品ではなく、(基板/キットに於いて他の方法で注記されるかもしれないのを除き、)リサイクル(WEEE)、FCC、CE、またはULの電磁適合性に関連する制限や指令なしで完成製品へ応用できる、含めることの何かまたは何れかの技術的または法律上の必要条件に(未だ)適合しないかもしれません。Atmelは販売者と更にその先の使用者単独の危険に於いて、全ての障害と共に何の保証もなく、“現状そのまま”でこの基板/キットを供給しました。使用者は商品の適切で安全な取り扱いのために全ての義務と責任を負います。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 © 2009, 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®ロコとそれらの組み合わせ、AVR®、AVR®ロコ、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 2013.

本応用記述はAtmelのAVR476応用記述(doc8251.pdf Rev.8251A-08/09)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。