

説明

Atmel® QTouch® 周辺機能接触制御器 (PTC: Peripheral Touch Controller) は、釐、摺動子、輪として機能する感知器での静電容量接触測定用の組み込みハードウェアを提供します。PTCはどんな外部部品も必要なしで相互容量と自己容量の両測定を支援します。これは素晴らしい感度と雑音耐性、更に自己校正を提供し、使用者による感度調整労力を最小にします。

PTCは自律的に実行する静電容量接触感知器測定が意図されます。外部の静電容量接触感知部は代表的にPCBで形成され、感知部電極はデバイスの入出力ピンを用いてPTCのアナログ電荷蓄積器に接続されます。PTCは酸化インジウム錫 (ITO: Indium Tin Oxide) 感知部格子を含む、異なるX-Y構成設定で容量接触配列として構成される相互容量感知部を支援します。相互容量動作では、PTCがX線 (駆動線) 毎に1つのピンとY線 (感知線) 毎に1つのピンを必要とします。自己容量動作では、各自己容量感知部に対してPTCがY線駆動部で1つのピンだけを必要とします。

要点

- 低電力、高感度、環境的に丈夫な静電容量接触釐、摺動子、輪の実装
- 相互容量と自己容量の感知を支援
- 自己容量動作で32個までの釐
- 相互容量動作で256個までの釐
- 一括動作構成設定支援
- 電極毎に1つのピン - 外部部品なし
- 負荷補償電荷検出
- 相互容量動作寄生容量補償
- 優れた感度のための調整可能な利得
- 温度とVDDの範囲に渡る0変動
- 温度やVDDの補償に対して必要なし
- 高伝導耐性のためのハードウェア雑音濾過と雑音信号非同期化
- Atmelが提供するQTouchライブラリ ファームウェアとQTouch構成器 (Composer) ツール

製品支援

QTouch 静電容量接触感知ソフトウェア ライブラリに関連する援助と関連する問題については、最寄りのAtmel営業担当者にお問い合わせ頂くか、または支援要請を依頼するか、または包括的な知識基盤にアクセスするためにmyAtmel設計支援入り口で記入してください。

myAtmelアカウントを持って居ない場合、ページの先頭のmyAtmelメニューで **Create Account** (アカウント作成) をクリックすることによって新しいアカウントを作成するために <http://www.atmel.com/design-support/> を訪ねてください。

ログインすると、知識基盤にアクセスすることができ、myAtmelページから新しい支援事例を提出するか、または進行中の事例の状態を再調査してください。

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。

目次

説明	1
要点	1
1. 開発ツール	3
2. 支援するデバイスの種類	3
3. 静電容量接触技術	4
3.1. 静電容量接触感知器	4
3.2. 容量測定方法	4
3.3. 自己容量測定法	4
3.4. 相互容量測定法	4
3.5. 容量接触一括感知部	4
3.6. 容量接触低電力感知部	5
3.7. PTCとその恩恵	6
3.8. 自己容量法と相互容量法に対するPTC構成図	7
3.9. PTCでの設計手法	8
3.10. 容量接触開発周囲	8
4. 接触感知器デバッグと状態情報	9
4.1. 信号	9
4.2. 参照基準	9
4.3. 差	10
4.4. 接触状態と摺動子/輪の位置	10
5. QTouchライブラリ	11
5.1. 概要	11
5.2. ライブラリ パラメータ	12
5.3. 耐水性	26
5.4. 感知部状態読み取り	27
5.5. 応用の流れ	27
5.6. API手順	28
5.7. 状態機構	29
5.8. 動作形態	30
5.9. 接触ライブラリAPI異常	33
6. 雑音性能用調整	33
6.1. 雑音源	33
6.2. 雑音計数器測定	33
7. 応用設計	37
7.1. 接触ライブラリと関連ファイル	37
7.2. コードとデータのメモリの考慮	37
8. 例応用	39
8.1. Atmel基板例プロジェクト	39
8.2. 使用者基板例プロジェクト	41
8.3. 例プロジェクトでのAtmelソフトウェア枠組み(ASF)使用	42
8.4. 使用者基板をプログラミングするのにXplained Proキットの使用	42
8.5. QDebug接触データ デバッグ通信インターフェースの使用	42
8.6. 使用者基板からQDebugデータの流にXplained Proキットの使用	43
8.7. 使用者基板からQDebugデータの流にAtmel ICEの使用	44
9. 既知の問題	45
10. PTC QTouchでの良くある質問	46
11. 追補	47
11.1. マクロ	47
11.2. 型定義	48
11.3. 列挙	48
11.4. データ構造体	53
11.5. 全域変数	57
11.6. API	57
12. 改訂履歴	62

1. 開発ツール

PTCを用いてQTouchライブラリを開発するには以下の開発ツールが必要とされます。

- GCCコンパイラ用開発環境
 - QTouch構成器(Composer) 5.9.116またはそれ以降版
 - QTouchライブラリ 5.9.211またはそれ以降版
 - 注: QTouchライブラリと構成器拡張は<http://www.atmel.com/>からダウンロードすることができるAtmel Studio 7でだけ動きます。
 - 依存するAtmel Studio拡張
 - Atmelソフトウェア枠組み 3.30.1またはそれ以降版
 - Atmelキット拡張 7.0.70またはそれ以降版
- IARコンパイラ用開発環境
 - ARM®用IAR Embedded Workbench® 7.50.1.10273またはそれ以降
 - Atmel AVR®用IAR Embedded Workbench® 6.70.1またはそれ以降
 - Atmelソフトウェア枠組み 3.29.0またはそれ以降 (任意選択)
 - Atmel QTouchライブラリ 5.9.211 IARインストーラ (<http://www.atmel.com/tools/qtouchlibraryptc.aspx>で入手可能)

2. 支援するデバイスの種類

SAMとATmegaのデバイス用のQTouchライブラリは以下のデバイスの種類に対して利用可能です。

系列	品種
SAM D20 J系列	ATSAMD20J18, ATSAMD20J17, ATSAMD20J16, ATSAMD20J15,ATSAMD20J14
SAM D20 G系列	ATSAMD20G18, ATSAMD20G18U, ATSAMD20G17, ATSAMD20G17U,ATSAMD20G16, ATSAMD20G15, ATSAMD20G14
SAM D20 E系列	ATSAMD20E18, ATSAMD20E17, ATSAMD20E16, ATSAMD20E15,ATSAMD20E14
SAM D21 J系列	ATSAMD21J18A, ATSAMD21J17A, ATSAMD21J16A, ATSAMD21J15A,ATSAMD21J16B, ATSAMD21J15B
SAM D21 G系列	ATSAMD21G18A, ATSAMD21G17A, ATSAMD21G16A, ATSAMD21G15A,ATSAMD21G15B, ATSAMD21G16B, ATSAMD21G17AU, ATSAMD21G18AU
SAM D21 E系列	ATSAMD21E18A, ATSAMD21E17A, ATSAMD21E16A, ATSAMD21E15A,ATSAMD21E15B, ATSAMD21E15BU, ATSAMD21E16B, ATSAMD21E16BU
SAM D10 C系列	ATSAMD10C14A
SAM D10 D系列	ATSAMD10D14AM, ATSAMD10D14AS, ATSAMD10D14AU
SAM D11 C系列	ATSAMD11C14A
SAM D11 D系列	ATSAMD11D14AM, ATSAMD11D14AS, ATSAMD11D14AU
SAM L21 E系列	ATSAML21E15B, ATSAML21E16B, ATSAML21E17B, ATSAML21E18B
SAM L21 G系列	ATSAML21G16B, ATSAML21G17B, ATSAML21G18B
SAM L21 J系列	ATSAML21J16B, ATSAML21J17B, ATSAML21J18B
SAM R21 E系列	ATSAMR21E16A, ATSAMR21E17A, ATSAMR21E18A, ATSAMR21E19A
SAM R21 G系列	ATSAMR21G16A, ATSAMR21G17A, ATSAMR21G18A
SAM DA1 E系列	ATSAMDA1E14A, ATSAMDA1E15A, ATSAMDA1E16A
SAM DA1 G系列	ATSAMDA1G14A, ATSAMDA1G15A, ATSAMDA1G16A
SAM DA1 J系列	ATSAMDA1J14A, ATSAMDA1J15A, ATSAMDA1J16A
SAM C21 E系列	ATSAMC21E15A, ATSAMC21E16A, ATSAMC21E17A, ATSAMC21E18A
SAM C21 G系列	ATSAMC21G15A, ATSAMC21G16A, ATSAMC21G17A, ATSAMC21G18A
SAM C21 J系列	ATSAMC21J16A, ATSAMC21J17A, ATSAMC21J18A
SAM C20 E系列	ATSAMC20E15A, ATSAMC20E16A, ATSAMC20E17A, ATSAMC20E18A
SAM C20 G系列	ATSAMC20G15A, ATSAMC20G16A, ATSAMC20G17A, ATSAMC20G18A
SAM C20 J系列	ATSAMC20J16A, ATSAMC20J17A, ATSAMC20J18A
SAM L22 G系列	ATSAML22G16A, ATSAML22G17A, ATSAML22G18A
SAM L22 J系列	ATSAML22J16A, ATSAML22J17A, ATSAML22J18A
SAM L22 N系列	ATSAML22N16A, ATSAML22N17A, ATSAML22N18A
ATmega系列	ATmega328PB, ATmega324PB

3. 静電容量接触技術

3.1. 静電容量接触感知器

静電容量接触感知器は伝統的な機械的インターフェースを置き換え、機械的な摩擦なしで動作し、環境に対して閉鎖されています。これらは工業設計に於いてより大きな柔軟性を提供し、最終製品設計の差別化に帰着します。より多くの情報については「[容量接触一括感知部](#)」と「[容量接触低電力感知部](#)」を参照してください。

3.2. 容量測定方法

自己容量測定法は未知の容量の感知電極を既知電位に充電することを含みます。結果として電荷が測定回路に移動されます。1つ以上の充電と移動の周回で電荷を測定することにより、感知極の容量を測定することができます。

相互容量測定法は感知電極の対を使います。1つの電極は論理パルスから成る電荷が群発(集中)動作で駆動される放電部として働きます。他の電極は上を覆うパネル誘電体を用いて放射部と結合する受信部として働きます。指がパネルに接触すると、場の結合が減らされ、接触が検出されます。

3.3. 自己容量測定法

- ・ 単一感知電極(Y線)を使用
 - 自己容量釦を1つのチャネルとして形成することができます。
 - 自己容量の摺動子と輪は3つのチャネルを使って形成されます。
- ・ 頑丈で使い易い、少ない感知器数に対して理想的

図3-1. 感知部種類

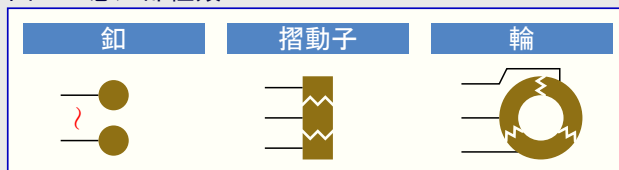


図3-2. 容量測定原理

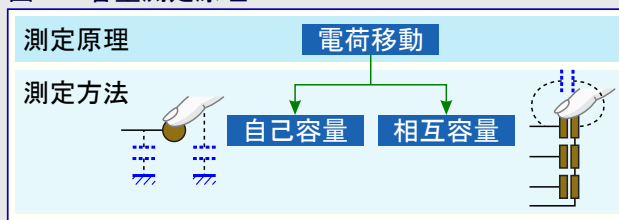
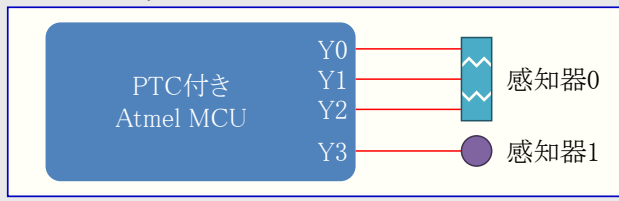


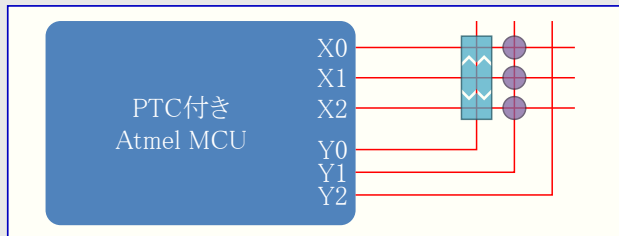
図3-3. 自己容量方式



3.4. 相互容量測定法

- ・ 感知電極の対(X-Y線)を使用
 - 相互容量釦は1つのX-Yチャネルを使います。
 - 相互容量の摺動子と輪は感知部の大きさに応じて3~8つのX-Yチャネルを使うように構成設定することができます。
- ・ 多い感知器数に適合
- ・ より良い耐水性

図3-4. 相互容量方式



3.5. 容量接触一括感知部

一括感知部構成設定は1つの単一感知器として働くための複数感知線(自己容量測定)または複数の駆動と感知の線(相互容量測定)の組み合わせです。一括動作はシステム性能全体を改善するための応用開発者用ツールとして働きます。

改善される電力効率

複数感知部が共に一括にされ1つの単一感知器として扱われると、走査を実行するのにかかる時間が減らされます。複数釦を使う電池給電応用に対して、接触感知部の群は単一の一括感知器を形成するように一纏めにされ、この感知器だけを走査することができます。これによって減らされた電力消費に帰着します。システムで構成設定された全ての感知部は、一纏めにされた感知部でのユーザー存在検出で、その後個別に走査することができます。

改善される応答時間

多いキー数の応用では、感知部接触と接触連絡間にかかなりの遅延が有り得ます。これは各測定周回で各キーの容量を順次測定するのに時間がかかるためです。一括動作実行で、感知部を群に配列することによってこの遅延を減らすことができます。それらの一纏めにされた群の1つが接触検出を示すと、どれが接触されたかを判断するのに、その群内のキーだけが個別に測定されます。

例えば、64キーから成るキーボードは8つの群を一纏めにした8つに分けられるかもしれません。

故に、各測定周回は一纏めにされた8つの感知部だけを測定するように減らされます。接触連絡が加えられると、最初に一括感知部が接触差を示し、その後8つの部分キーが走査され、位置が決定されます。伝統的な全キーの順次走査での64測定に比べて、全キーの接触状態を決定するのに16測定だけが必要とされます。

[一括構成設定で]キーの群を走査することができるため、それは低電力採取中に追加端を提供し、故に消費される電力を劇的に減らします。各感知部は自身の前置分周されたクロックと雑音耐性を改善するための直列抵抗を持ちます。

図3-5. PTCに接続された自己容量感知部

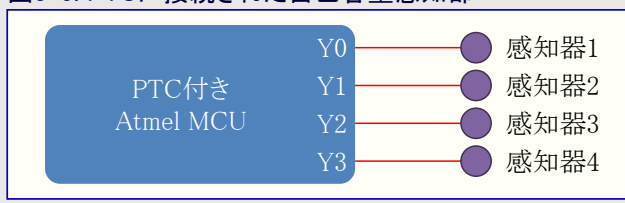
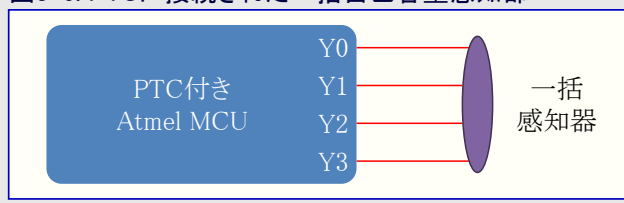


図3-6. PTCに接続された一括自己容量感知部



上の右の図では個別の鉤が自己容量配列と等価な一括と共に示されます。

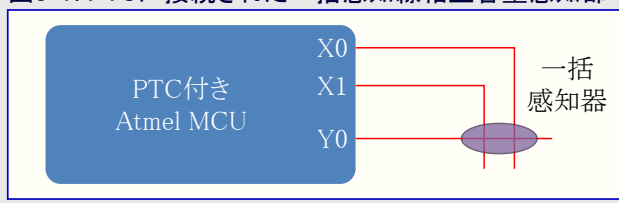
自己容量法に対する一括動作ピンと感知部の構成設定

```
#define DEF_SELF_CAP_LINES Y(5), Y(4), Y(11), Y(10), Y(13), Y(7), Y(12), Y(6), LUMP_Y(5, 4)

touch_ret = touch_selfcap_sensor_config(SENSOR_TYPE_LUMP, CHANNEL_8, CHANNEL_8, NO_AKS_GROUP, ¥
40u, HYST_6_25, RES_8_BIT, &sensor_id);
```

右図では相互容量で一括にされた感知部構成設定が表されます。

図3-7. PTCに接続された一括感知線相互容量感知部



相互容量法に対する一括動作ピンと感知部の構成設定

```
#define DEF_MUTL_CAP_NODES X(8), Y(10), X(9), Y(10), X(2), Y(12), X(3), Y(12), X(8), Y(12), ¥
X(9), Y(12), X(2), Y(13), X(3), Y(13), X(8), Y(13), X(9), Y(13), LUMP_X(2, 3, 8, 9), LUMP_Y(10, 13)

touch_ret = touch_mutlcap_sensor_config(SENSOR_TYPE_LUMP, CHANNEL_10, CHANNEL_10, ¥
NO_AKS_GROUP, 20u, HYST_6_25, RES_8_BIT, 0, &sensor_id);
```

使用制限

一括にした感知部容量負荷は相互または自己の容量法のどちらでも個別の感知部に対する最大感知部負荷を超えるべきではありません。一括動作はより大きな感知部を1つの単一の感知部として扱い、従って一括にされた最大感知部負荷はこの仕様にも注視されるべきで、さもないければこれは校正誤差に帰着するでしょう。

相互容量測定動作では各感知部の容量負荷が自己容量法のそれよりも一般的によりもっと低くなります。従って単一の一括にした感知部として、より多くの相互感知部を共に使うことが原則として可能です。使用者は以下を用いて一括にした感知部が校正誤差(0x80の値)に終わらないことを保証することができます。

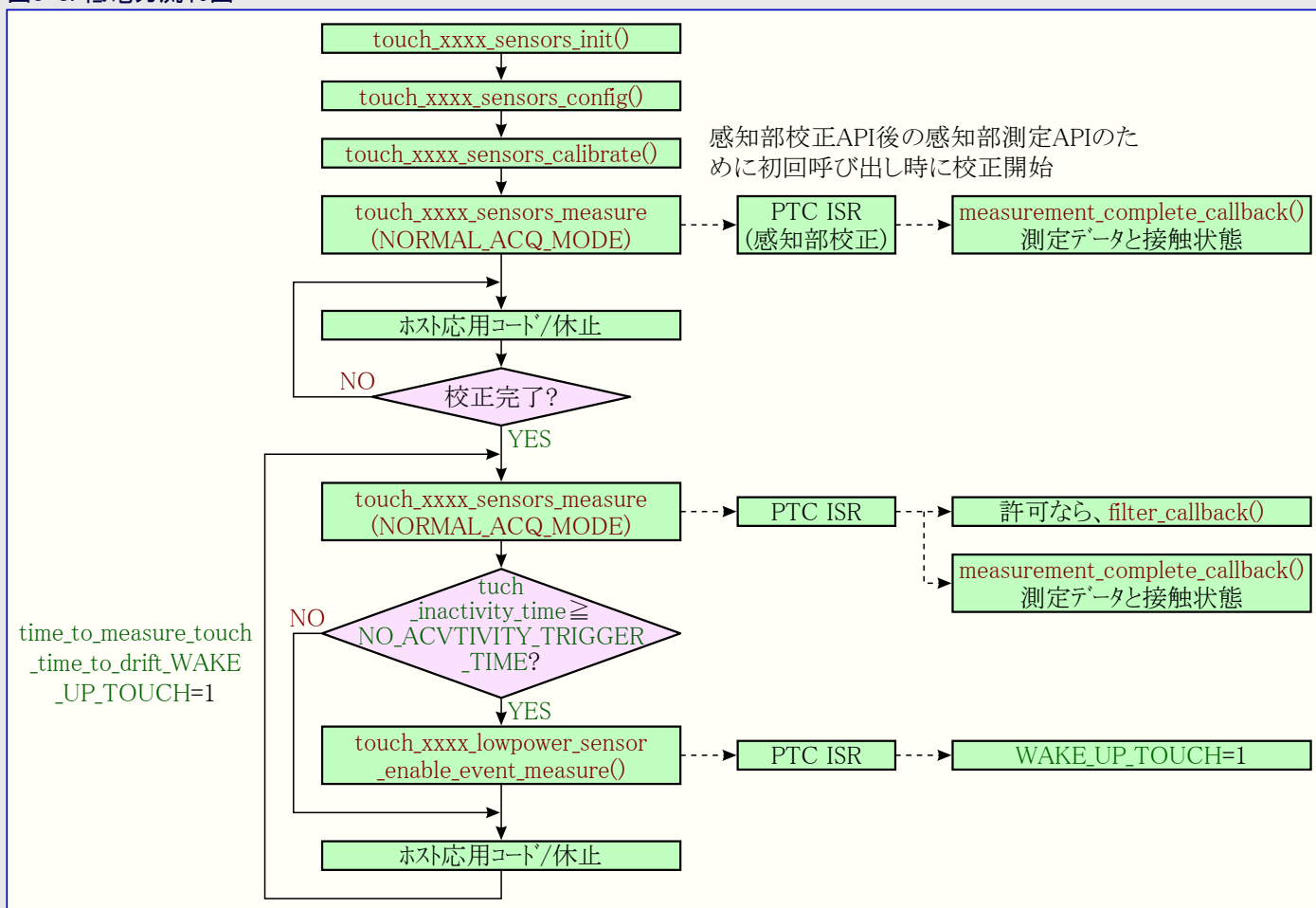
```
#p_XXXXCAP_measure_data->p_sensors[<SENSOR>].state variable.
```

3.6. 容量接触低電力感知部

QTouchライブラリは事象システムを用いて自律的にPTC接触感知を操作するように構成設定されるかもしれませんが、この動作に於いて、単一感知部は'低電力'キーとして指定され、どのCPU活動もなしに接触検出に対して周期的に測定されるかもしれません。CPUはこの動作を通して多分深い休止動作に保たれ、電力消費を最小にします。

低電力キーは自己容量に対して1つのY(感知)線、または相互容量に対して1つのX(駆動)と1つのY(感知)で分離した電極かもしれず、または一括動作感知部として複数の駆動や感知の線の組み合わせかもしれません。

図3-8. 低電力流れ図



活性測定動作

活性測定動作ではDEF_TOUCH_MEASUREMENT_PERIOD_MSms走査間隔で構成設定された全ての感知部が測定されます。使用者応用配置は接触活動がNO_ACTIVITY_TRIGGER_TIMEms間、構成設定したどの感知部も検出されない時に応用が低電力測定動作に切り替わるように設計することができます。

低電力測定動作

低電力測定動作では、指定した感知部または一括にされた感知部を単一感知部として走査することができます。この動作には、システムがスタンバイ休止動作で、SAM/megaのデバイスでRTCとPTC/WDTとPTCの単位部を除き、CPUと他の周辺機能が休止です。指定した低電力感知部での使用者接触はその接触を解決するために起き上がって活性測定を実行することをCPUにさせます。指定した低電力感知部の参照基準経緯を保つため、RTC/WDTは1つの活性測定を実行するのにDEF_LOWPOWER_SENSOR_DRIFT_PERIODICITYms毎で周期的にCPUを起こすように構成設定されます。

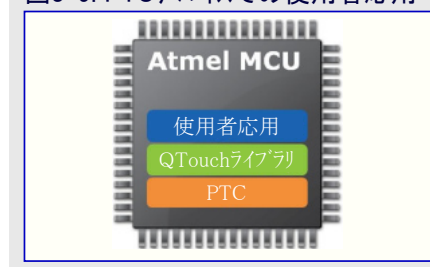
活性測定動作と低電力測定動作間の切り替え

活性から低電力の測定動作へ切り替えると、一括感知部を除く全ての感知部が禁止されます。故に、低電力動作中にそれらの感知部で参照基準追跡が全く実行されません。一括感知部で接触が検出されると、禁止された全ての感知部が今や再許可されてその感知部で測定が開始されます。デバイスが非常に長い時間の間、休止の場合、それらの感知部での正しい参照基準値を保証するために再許可された感知部での校正を強制することが推奨されます。

3.7. PTCとその恩恵

- ハードウェア+ファームウェアの混合解決策は使用者に感知部構成設定の定義を許します。
 - 周辺機能接触制御器 + QTouchライブラリ
- PTCはデータ採取を自律的に動き、低いCPU利用と電力消費に帰着します。
 - 使用者に制御された電力性能二律背反
 - CPUは電力を節約するために採取間で休止にすることができます。
 - 代わりに、CPUは接触採取間で他の時間が重要な動作を実行することができます。
- 強力な雑音性能

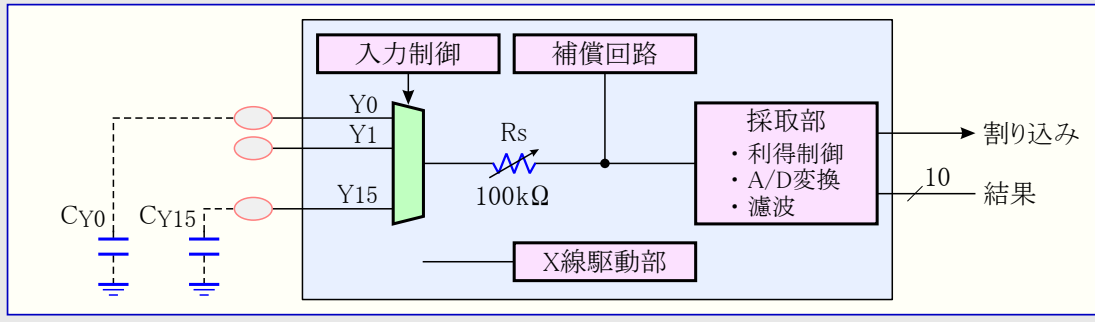
図3-9. PTCデバイスでの使用者応用



3.8. 自己容量法と相互容量法に対するPTC構成図

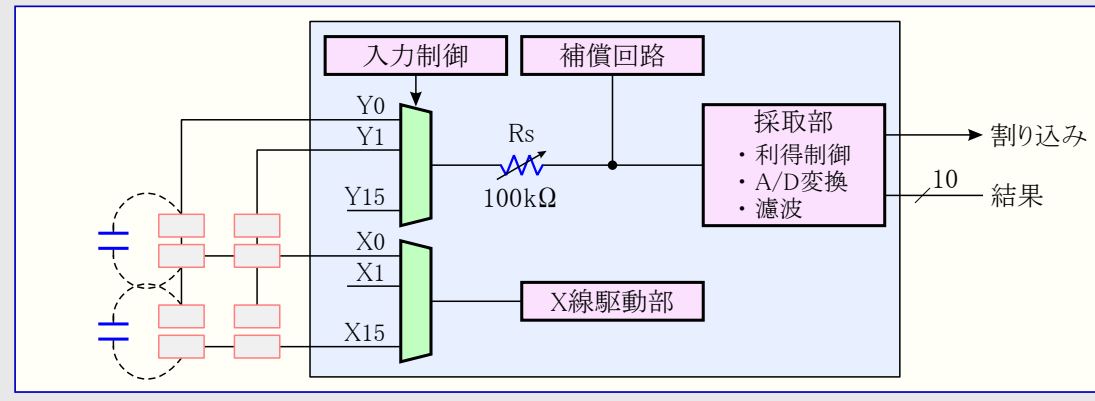
自己容量測定用PTC構成図は以下の図で示されます。自己容量感知部へY線だけを接続することができ、入力制御を用いて選択されます。X線は未使用に留まり、他のどのGPIO機能にも使うことができます。補償回路と共にこの採取単位部は使用者接触のための容量での変化の測定で役立ちます。

図3-10. PTC自己容量法 - 構成図



相互容量測定用PTC構成図は以下の図で示されます。X線とY線の両方が相互感知部に接続されるべきで、入力制御を用いて選択されます。

図3-11. PTC相互容量法 - 構成図



3.8.1. 補償回路

PTCは感知部容量を補償するのに使われる内部補償回路を持ちます。自己容量と相互容量の両感知動作は同じ補償範囲を持ちます。しかし、相互容量動作は自己容量動作と比べてより多くの寄生容量を補償することができます。

tag_touch_measure_data_t構造体は現在のチャネルの補償回路値を表すp_cc_calibration_valsパラメータを含みます。より多くの情報については測定データ型(tag_touch_measure_data_t)を参照してください。

```
pFでの使用される補償回路値 = (p_cc_calibration_vals[channel_no] & 0x0F) * 0.00675 +  
    ((p_cc_calibration_vals[channel_no] >> 4) & 0x0F) * 0.0675 +  
    ((p_cc_calibration_vals[channel_no] >> 8) & 0x0F) * 0.675 +  
    ((p_cc_calibration_vals[channel_no] >> 12) & 0x3) * 6.75
```

また、touch_XXXXcap_sensors_calibrate関数は使う感知部に従って補償回路を校正することで使用者を助けます。このルーチンが飽和のために補償回路を校正するのに失敗した場合、この測定はTOUCH_CC_CALIB_ERRORを返すでしょう。補償回路はその限界を超えるかもしれませんが。失敗した特定感知部はそれがSENSOR_CALIBRATION_ERR(0x80u)の値を含む時にp_XXXXcap_measure_data->p_sensor[<sensor>].stateを用いて判断することができます。

- 自己容量動作の代表的な補償回路値は10〜25pFの範囲で、相互容量動作でのそれは約2pFです。
- 補償回路値は感知部の大きさと感知部や布線周辺の接地によって影響を及ぼされます。補償回路値は0.00675〜31.48pFの範囲です。
- 補償回路値が限度を超える場合、感知部と接地の面に於いて平面の代わりに網目を使ってください。
- 詳細な感知部設計については<http://www.atmel.com/images/doc10752.pdf>を参照してください。

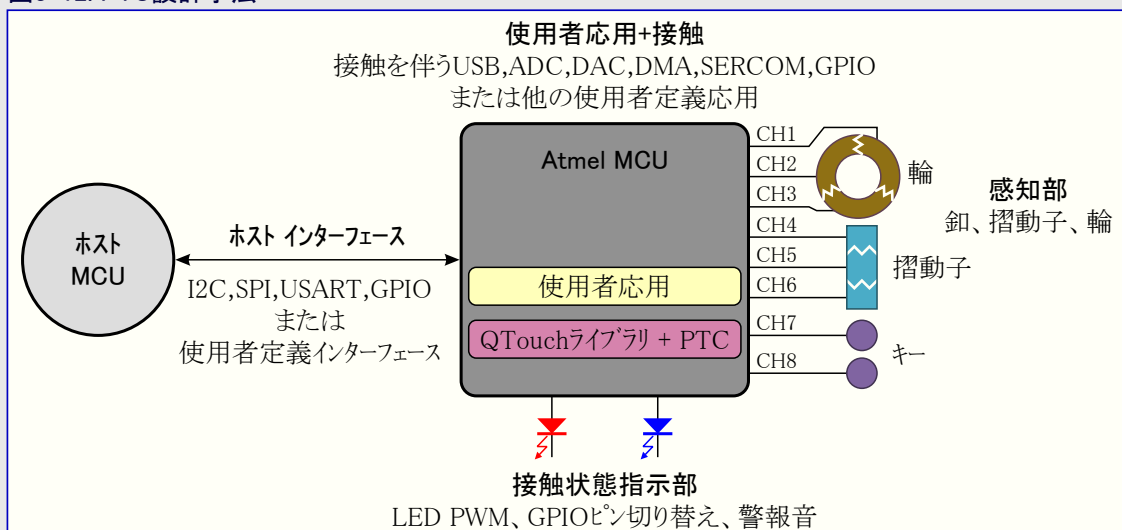
3.9. PTCでの設計手法

PTCと共にAtmel MCUを使う時に2つの設計手法が可能です。Atmel MCUは主に接触測定用MCUとして使われ得ます。さもなければ、Atmel MCUは”チップ上”接触機能に使われるPTCと共に例えばUSB、ADC、DAC、SERCOM、DMA、GPIOのような周辺機能を利用するホストMCUとして機能し得ます。

設計手法は次のとおりです。

- 接触MCUとして主に機能するPTC付きAtmel MCU
 - 接触感知部状態と回転子/摺動子の位置検出に使用
 - LED、警報器などを使って接触状態を示すための付加的使用
 - ホストMCUへ接触状態と回転子/摺動子の位置情報を送出
- チップ上接触機能を持つホストMCUとして機能するAtmel MCU
 - チップ上接触機能とで単一チップ解決策として費用節約設計が可能
 - 設計された使用者応用に対して他のチップ上周辺機能を利用

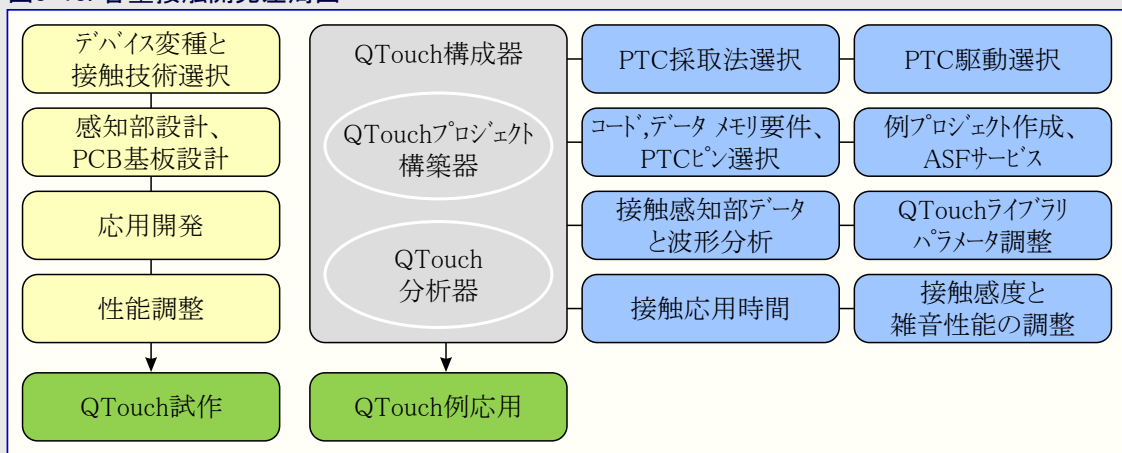
図3-12. PTC設計手法



3.10. 容量接触開発周期

容量接触開発周期は使用者インターフェース ハードウェアだけでなくファームウェア応用開発を開発するのもPCB基板設計を伴います。Atmel Studio拡張陳列室の一部として利用可能なQTouch構成器(Composer)PCソフトウェアはPTC QTouchライブラリ プロジェクトのために接触感知部に対して望む使用者構成設定を自動的に生成することを許します。QTouch構成器は接触感知部データ分析と感度と雑音に対する性能調整も許します。

図3-13. 容量接触開発差周期



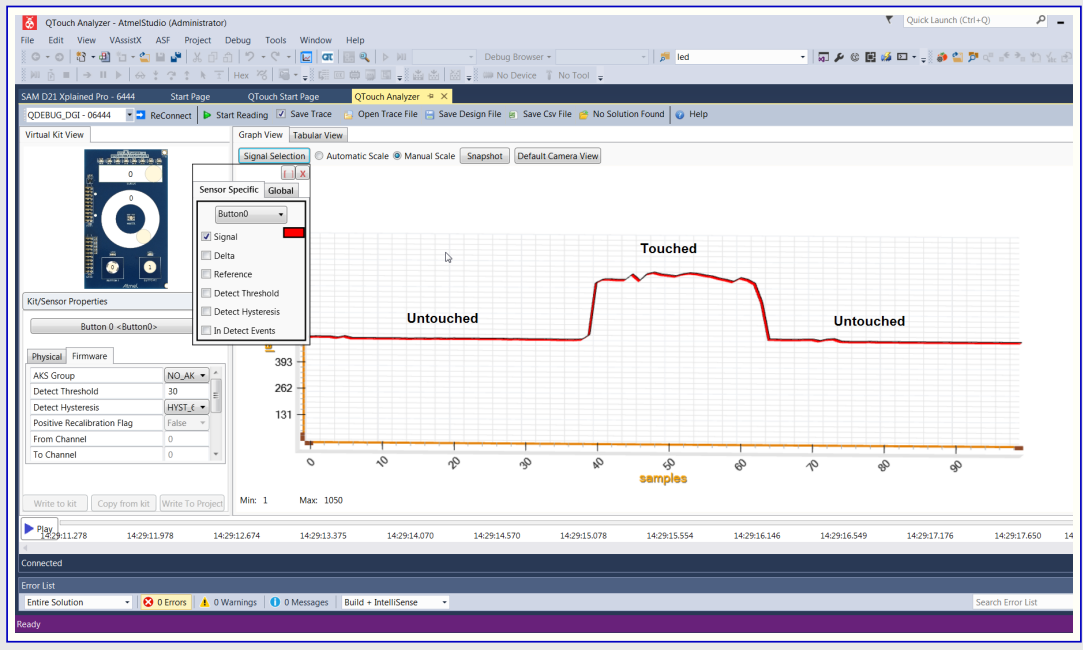
4. 接触感知器デバッグと状態情報

感知部の調整に必要な接触感知部デバッグ情報は信号、参照基準、差、補償容量です。信号、参照基準、差が感度と雑音で感知部パラメータを調整するのに役立つ一方で、補償容量は極端な感知部設計に対する指標です。感知部情報と位置情報は関連する接触活動を始めるために使用者応用によって判定されなければなりません。

4.1. 信号

信号値は与えられた接触チャネル上での生の測定データです。この値は接触で増加します。

図4-1. チャネル信号



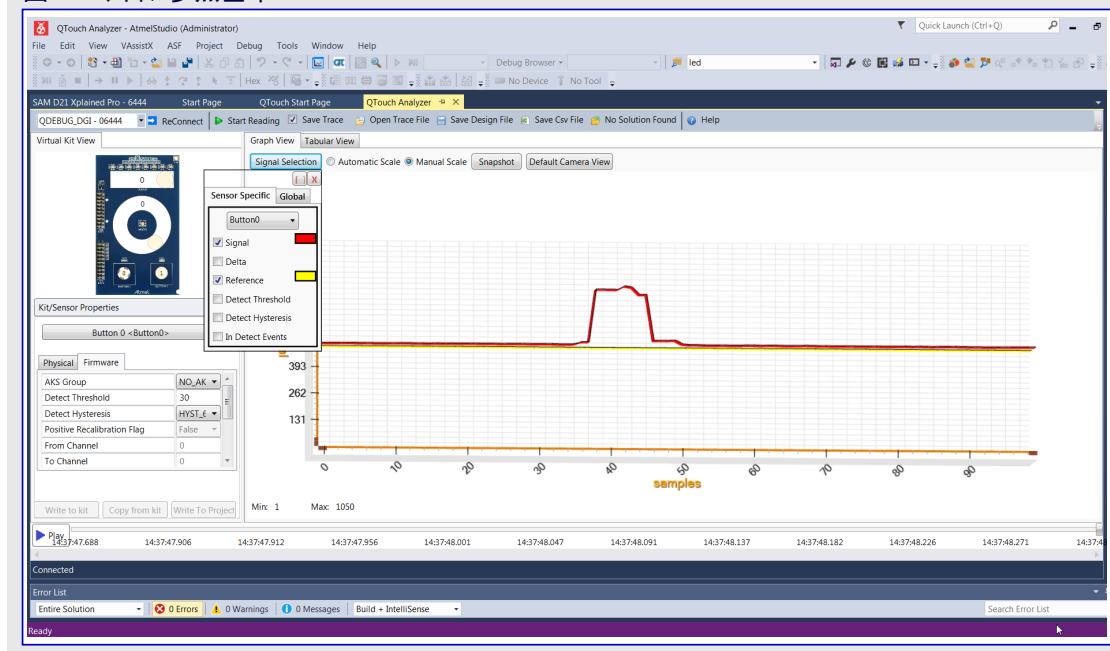
4.2. 参照基準

接触チャネルの参照基準値は特定チャネルでの長期測定平均です。

以下を表します。

- ・ 接触が全くない時にリセットする信号
- ・ 校正処理中に得られた初期値
- ・ 参照基準は変動補償算法によって適応されます。

図4-2. チャネル参照基準

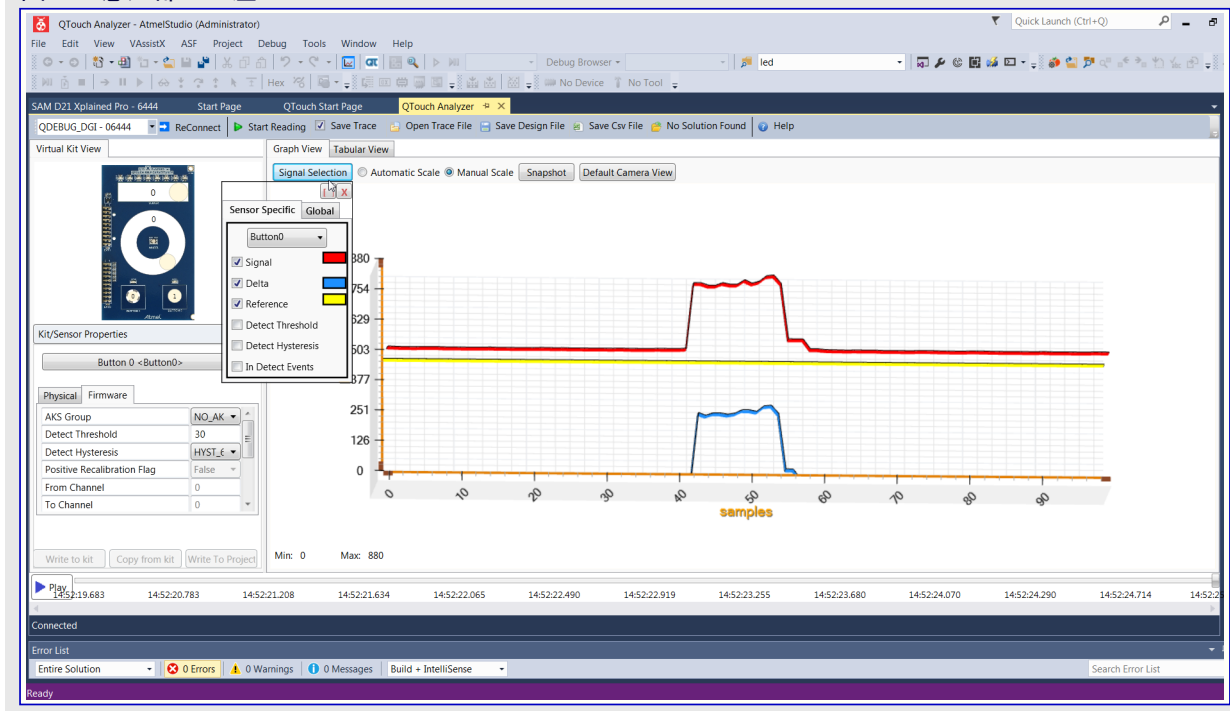


4.3. 差

接触チャネルの差の値は接触強度を表します。

- ・ 差=(信号-参照基準)
- ・ 差は接触で増加します。

図4-3. 感知部での差



4.4. 接触状態と摺動子/輪の位置

感知部接触状態は主に使用者応用によって利用される接触感知部情報です。感知部の状態はONまたはOFFのどちらかで有り得ます。摺動子と輪については、追加の接触位置が目的です。8ビット分解能については、接触位置が端から端で0～255の範囲です。接触ライブラリでの設定を構成設定することによってより低い分解能に構成設定することが可能です。感知器接触状態と摺動子/輪位置は一旦ライブラリが測定を完了するときと常に使われるでしょう。

相互容量または自己容量の感知部に対する接触感知部状態は以下のブール変数を読むことによって得ることができます。

```
bool sensor_state_self = GET_SELFAP_SENSOR_STATE (SENSOR_NUMBER);  
bool sensor_state_mutl = GET_MUTLAP_SENSOR_STATE (SENSOR_NUMBER);
```

相互容量または自己容量の感知部に対する接触感知部の回転子や摺動子の位置情報は以下のパラメータを用いて得ることができます。

```
uint8_t rotor_slider_position_self = GET_SELFAP_ROTOR_SLIDER_POSITION (ROTOR_SLIDER_NUMBER);  
uint8_t rotor_slider_position_mutl = GET_MUTLAP_ROTOR_SLIDER_POSITION (ROTOR_SLIDER_NUMBER);
```

相互容量または自己容量の感知部に対する接触感知部の雑音状態は以下のパラメータを用いて得ることができます。

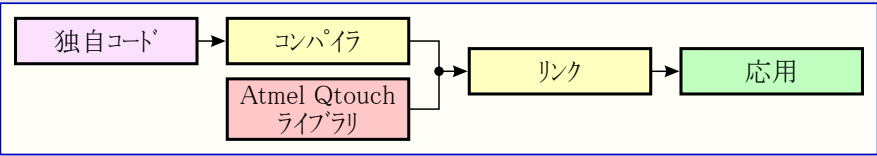
```
bool sensor_noise_state_self = GET_SELFAP_SENSOR_NOISE_STATUS (SENSOR_NUMBER);  
bool sensor_noise_state_mutl = GET_MUTLAP_SENSOR_NOISE_STATUS (SENSOR_NUMBER);
```

5. QTouchライブラリ

Atmel QTouchライブラリは汎用のAtmel SMART | ARMとAVR®マイクロ コンピュータ応用に対して組み込みの容量接触釦、摺動子、輪の機能の開発を簡単にします。使用料無料のQTouchライブラリは各デバイスに対していくつかのライブラリを提供し、異なる数の接触チャンネルを支援し、接触応用で柔軟性と効率性の両方を許します。

QTouch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の開発や、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。開発者は単一インターフェースでの各種組み合わせで釦、摺動子、輪を実装する自由を持ちます。

図5-1. QTouchライブラリ

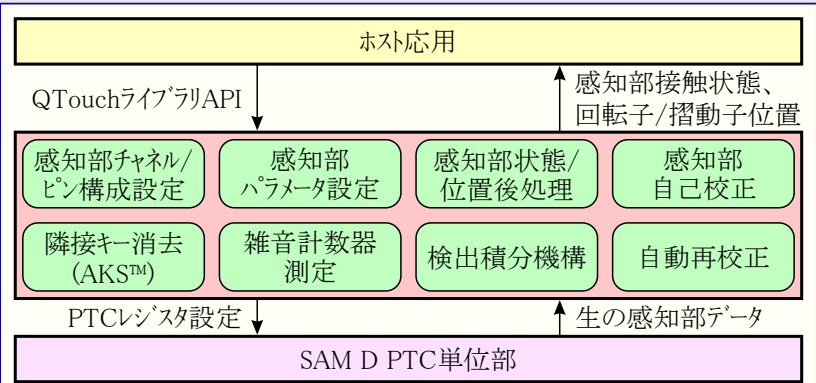


5.1. 概要

PTC用のQTouchライブラリAPIは接触感知部ピン構成設定、採取パラメータ設定だけでなく、周期的な感知部データ捕獲と状態更新操作に対しても使うことができます。QTouchライブラリは必要な活動を実行するためにPTC単位部と次々と調整します。PTC単位部は外部容量接触感知部とインターフェースして自己と相互の容量法の測定を実行することができます。このライブラリは低電力と一括動作構成設定が特徴です。

QTouchライブラリAPIは使用者応用が自立型の自己容量法か相互容量法または両方の方法を同時に使うことができるように整えられます。下表は各方法に対して利用可能なAPIを捕らえます。補助APIは使用者応用に追加の柔軟性を提供します。

図5-2. QTouchライブラリ概要



方法	標準API	補助API
相互容量	touch_mutlcap_sensors_init touch_mutlcap_sensor_config touch_mutlcap_sensors_calibrate touch_mutlcap_sensors_measure touch_mutlcap_sensors_deinit touch_mutlcap_lowpower_sensor_enable_event_measure	touch_mutlcap_sensor_get_delta touch_mutlcap_sensor_update_config touch_mutlcap_sensor_get_config touch_mutlcap_update_global_param touch_mutlcap_get_global_param touch_mutlcap_update_acq_config touch_mutlcap_get_acq_config touch_mutlcap_sensor_disable touch_mutlcap_sensor_reenable touch_multicap_mois_tolrnce_enable touch_multicap_mois_tolrnce_disable touch_mutlcap_cnfg_mois_threshold touch_mutlcap_cnfg_mois_mltchgrp touch_mutlcap_mois_tolrnce_quick_reburst_enable touch_mutlcap_mois_tolrnce_quick_reburst_disable touch_mutlcap_get_libinfo touch_library_get_version_info touch_resume_ptc touch_suspend_ptc
自己容量	touch_selfcap_sensors_init touch_selfcap_sensor_config touch_selfcap_sensors_calibrate touch_selfcap_sensors_measure touch_selfcap_sensors_deinit touch_selfcap_lowpower_sensor_enable_event_measure	touch_selfcap_sensor_get_delta touch_selfcap_sensor_update_config touch_selfcap_sensor_get_config touch_selfcap_update_global_param touch_selfcap_get_global_param touch_selfcap_update_acq_config touch_selfcap_get_acq_config touch_selfcap_sensor_disable touch_selfcap_sensor_reenable touch_selfcap_mois_tolrnce_enable touch_selfcap_mois_tolrnce_disable touch_selfcap_cnfg_mois_threshold touch_selfcap_cnfg_mois_mltchgrp touch_selfcap_mois_tolrnce_quick_reburst_enable touch_selfcap_mois_tolrnce_quick_reburst_disable touch_selfcap_get_libinfo touch_library_get_version_info touch_suspend_ptc touch_resume_ptc

5.2. ライブ러리 パラメータ

QTouch 라이브러리構成設定パラメータが下表で一覧にされます。

構成設定	相互容量	自己容量
ピン構成設定	DEF_MUTLCAP_NODES	DEF_SELFCAP_LINES
感知部 構成設定	DEF_MUTLCAP_NUM_CHANNELS DEF_MUTLCAP_NUM_SENSORS DEF_MUTLCAP_NUM_ROTORS_SLIDERS DEF_MUTLCAP_PTC_GPIO_STATE DEF_MUTLCAP_QUICK_REBURST_ENABLE	DEF_SELFCAP_NUM_CHANNELS DEF_SELFCAP_NUM_SENSORS DEF_SELFCAP_NUM_ROTORS_SLIDERS DEF_SELFCAP_PTC_GPIO_STATE DEF_SELFCAP_QUICK_REBURST_ENABLE
感知部 個別パラメータ	Detect Threshold Detect Hysteresis Position Resolution Position Hysteresis AKS group	Detect Threshold Detect Hysteresis Position Resolution AKS group
感知部 全体パラメータ	DEF_MUTLCAP_DI DEF_MUTLCAP_TCH_DRIFT_RATE DEF_MUTLCAP_ATCH_DRIFT_RATE DEF_MUTLCAP_MAX_ON_DURATION DEF_MUTLCAP_DRIFT_HOLD_TIME DEF_MUTLCAP_ATCH_RECAL_DELAY DEF_MUTLCAP_ATCH_RECAL_THRESHOLD DEF_MUTLCAP_TOUCH_POSTPROCESS_MODE DEF_MUTLCAP_AKS_ENABLE DEF_MUTLCAP_CSD DEF_MUTLCAP_AUTO_OS_SIGNAL_STABILITY_LIMIT	DEF_SELFCAP_DI DEF_SELFCAP_TCH_DRIFT_RATE DEF_SELFCAP_ATCH_DRIFT_RATE DEF_SELFCAP_MAX_ON_DURATION DEF_SELFCAP_DRIFT_HOLD_TIME DEF_SELFCAP_ATCH_RECAL_DELAY DEF_SELFCAP_ATCH_RECAL_THRESHOLD DEF_SELFCAP_TOUCH_POSTPROCESS_MODE DEF_SELFCAP_AKS_ENABLE DEF_SELFCAP_CSD DEF_SELFCAP_AUTO_OS_SIGNAL_STABILITY_LIMIT
感知部 採取パラメータ	DEF_MUTLCAP_FILTER_LEVEL_PER_NODE DEF_MUTLCAP_AUTO_OS_PER_NODE DEF_MUTLCAP_GAIN_PER_NODE DEF_MUTLCAP_FREQ_MODE DEF_MUTLCAP_HOP_FREQS DEF_MUTLCAP_CLK_PRESCALE_PER_NODE DEF_MUTLCAP_SENSE_RESISTOR_PER_NODE	DEF_SELFCAP_FILTER_LEVEL_PER_NODE DEF_SELFCAP_AUTO_OS_PER_NODE DEF_SELFCAP_GAIN_PER_NODE DEF_SELFCAP_FREQ_MODE DEF_SELFCAP_HOP_FREQS DEF_SELFCAP_CLK_PRESCALE_PER_NODE DEF_SELFCAP_SENSE_RESISTOR_PER_NODE
感知部校正 自動調節設定	AUTO_TUNE_PRSC, AUTO_TUNE_RSEL, AUTO_TUNE_NONE	
感知部 雑音測定と 締め出し パラメータ	DEF_MUTLCAP_NOISE_MEAS_ENABLE DEF_MUTLCAP_NOISE_MEAS_SIGNAL_STABILITY_LIMIT DEF_MUTLCAP_NOISE_LIMIT DEF_MUTLCAP_NOISE_MEAS_BUFFER_CNT DEF_MUTLCAP_LOCKOUT_SEL DEF_MUTLCAP_LOCKOUT_CNTDOWN	DEF_SELFCAP_NOISE_MEAS_ENABLE DEF_SELFCAP_NOISE_MEAS_SIGNAL_STABILITY_LIMIT DEF_SELFCAP_NOISE_LIMIT DEF_SELFCAP_NOISE_MEAS_BUFFER_CNT DEF_SELFCAP_LOCKOUT_SEL DEF_SELFCAP_LOCKOUT_CNTDOWN
感知部 採取周波数 自動調節 パラメータ	DEF_MUTLCAP_FREQ_AUTO_TUNE_ENABLE DEF_MUTLCAP_FREQ_AUTO_TUNE_SIGNAL_STABILITY_LIMIT DEF_MUTLCAP_FREQ_AUTO_TUNE_IN_CNT	DEF_SELFCAP_FREQ_AUTO_TUNE_ENABLE DEF_SELFCAP_FREQ_AUTO_TUNE_SIGNAL_STABILITY_LIMIT DEF_SELFCAP_FREQ_AUTO_TUNE_IN_CNT
共通パラメータ	DEF_TOUCH_MEASUREMENT_PERIOD_MS, DEF_TOUCH_PTC_ISR_LVL	
低電力 パラメータ	DEF_LOWPOWER_SENSOR_EVENT_PERIODICITY, DEF_LOWPOWER_SENSOR_DRIFT_PERIODICITY_MS, DEF_LOWPOWER_SENSOR_ID	
湿気パラメータ	DEF_MUTLCAP_MOIS_TOLERANCE_ENABLE DEF_MUTLCAP_NUM_MOIS_GROUPS DEF_MUTLCAP_MOIS_QUICK_REBURST_ENABLE	DEF_SELFCAP_MOIS_TOLERANCE_ENABLE DEF_SELFCAP_NUM_MOIS_GROUPS DEF_SELFCAP_MOIS_QUICK_REBURST_ENABLE

5.2.1. ピン、チャネル、感知部のパラメータ

相互容量法は各接触チャネルに対して感知電極の対を使います。これらの電極はX線とY線として示されます。容量測定は接触(X-Y)節がDEF_MUTLCAP_NODES構成設定パラメータで指定される順番で連続的に実行されます。相互容量接触感知部は単一のX-Yチャネルを用いて形成され、一方接触回転子または摺動子の感知部は3から8つのX-Yチャネルを用いて形成されます。

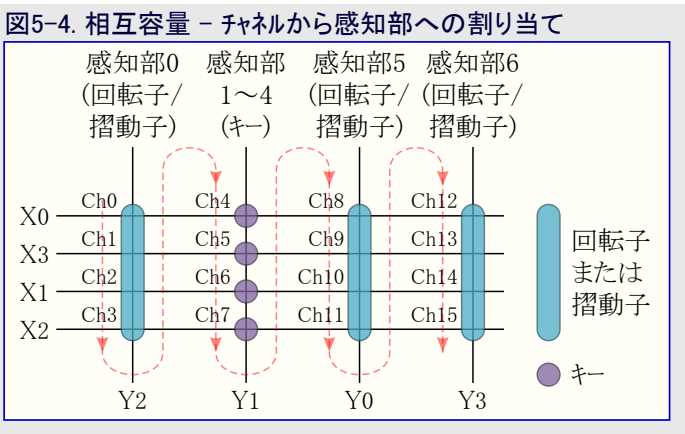
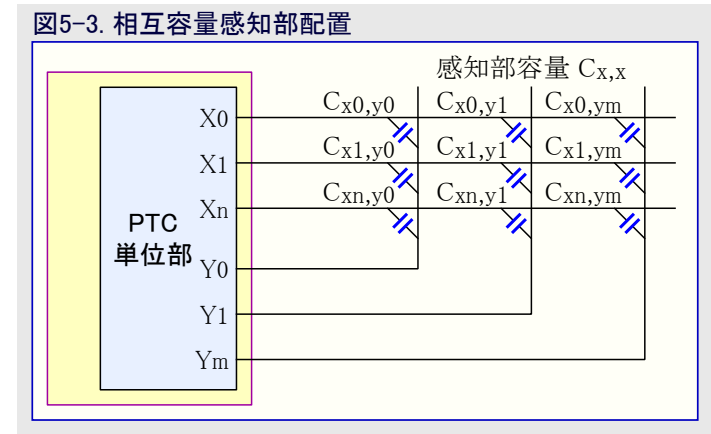
相互容量チャネル (X-Y感知節)

- SAM D20JとSAM D21J (64ピン) : 256までの接触チャネル、16本のX線と16本のY線
- SAM D20GとSAM D21G (48ピン) : 120までの接触チャネル、12本のX線と10本のY線
- SAM D20EとSAM D21E (32ピン) : 60までの接触チャネル、10本のX線と6本のY線
- SAM R21E (32ピン) : 12までの接触チャネル、6本のX線と2本のY線
- SAM R21G (48ピン) : 48までの接触チャネル、8本のX線と6本のY線
- SAM DA1J (64ピン) : 256までの接触チャネル、16本のX線と16本のY線
- SAM DA1G (48ピン) : 120までの接触チャネル、12本のX線と10本のY線
- SAM DA1E (32ピン) : 60までの接触チャネル、10本のX線と6本のY線
- SAM D21G17AUとSAM D21G18AU (45ピン) : 132までの接触チャネル、12本のX線と11本のY線
- SAM D21E15BUとSAM D21E16BU (35ピン) : 60までの接触チャネル、10本のX線と6本のY線

以下のデバイスはXとYの多重化任意選択を持ちます。

- SAM D10C14AとSAM D11C14A (14ピン) : 12までの接触チャネル、4本のX線と3本のY線
- SAM D10D14AS/AUとSAM D11D14AS/AU (20ピン) : 42までの接触チャネル、7本のX線と6本のY線
- SAM D10D14AMとSAM D11D14AM (24ピン) : 72までの接触チャネル、9本のX線と8本のY線
- SAM L21E (32ピン) : 42までの接触チャネル、7本のX線と6本のY線
- SAM L21G (48ピン) : 81までの接触チャネル、9本のX線と9本のY線
- SAM L21J (64ピン) : 169までの接触チャネル、13本のX線と13本のY線
- SAM L22G (48ピン) : 132までの接触チャネル、11本のX線と12本のY線
- SAM L22J (64ピン) : 182までの接触チャネル、13本のX線と14本のY線
- SAM L22N (100ピン) : 256までの接触チャネル、16本のX線と16本のY線
- SAM C21EとSAM C20E (32ピン) : 60までの接触チャネル、10本のX線と6本のY線
- SAM C21GとSAM C20G (48ピン) : 120までの接触チャネル、12本のX線と10本のY線
- SAM C21JとSAM C20J (64ピン) : 256までの接触チャネル、16本のX線と16本のY線
- ATmega328PB (32ピン) : 144までの接触チャネル、12本のX線と12本のY線
- ATmega324PB (44ピン) : 256までの接触チャネル、16本のX線と16本のY線

少数のピンはX線またはY線のどちらかとして使うことができます。個別デバイスのデータシートはこの多重化任意選択についてのより多くの情報を提供します。



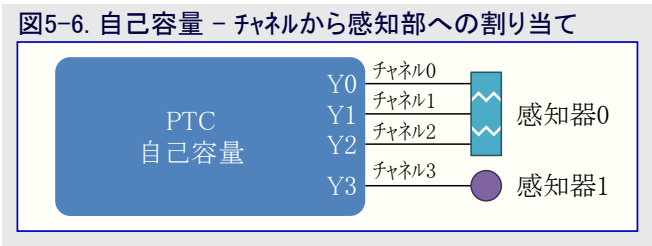
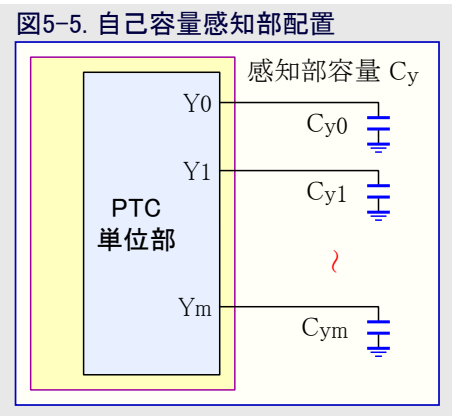
X-Y節対は不連続的な順番でDEF_MULTCAP_NODES構成設定パラメータを用いて指定することができます。チャネル番号付けはDEF_MULTCAP_NODES構成設定パラメータで指定したX-Y節点対と同じ順番で行われます。

設定	構成設定名	データ型	単位	X-Y節点対数		
				最小	最大	代表
相互容量接触チャネル節点	DEF_MUTLCAP_NODES	uint16_t配列	なし	1	256	—
相互容量チャネル数	DEF_MUTLCAP_NUM_CHANNELS	uint16_t	なし	1	256	—
相互容量感知部数	DEF_MUTLCAP_NUM_SENSORS	uint16_t	なし	1	256	—
相互容量回転子と摺動子の数	DEF_MUTLCAP_NUM_ROTORS_SLIDERS	uint8_t	なし	0	85	—

自己容量法はY線によって示される単一感知電極を使います。容量測定はDEF_SELFCAP_LINES構成設定パラメータで指定されるY線の順番で連続的に実行されます。自己容量接触卸感知部は単一線チャンネルを用いて形成され、一方接触回転子または摺動子の感知部は3つのY線チャンネルを用いて形成することができます。

自己容量チャンネル (Y感知線)

- SAM D20JとSAM D21J (64ピン) : 16までのチャンネル
- SAM D20GとSAM D21G (48ピン) : 10までのチャンネル
- SAM D20EとSAM D21E (32ピン) : 6までのチャンネル
- SAM D10C14AとSAM D11C14A (14ピン) : 7までのチャンネル
- SAM D10D14AS/AUとSAM D11D14AS/AU (20ピン) : 13までのチャンネル
- SAM D10D14AMとSAM D11D14AM (24ピン) : 16までのチャンネル
- SAM L21E (32ピン) : 7までのチャンネル
- SAM L21G (48ピン) : 10までのチャンネル
- SAM L21J (64ピン) : 16までのチャンネル
- SAM R21E (32ピン) : 2までのチャンネル
- SAM R21G (48ピン) : 6までのチャンネル
- SAM DA1J (64ピン) : 16までのチャンネル
- SAM DA1G (48ピン) : 10までのチャンネル
- SAM DA1E (32ピン) : 6までのチャンネル
- SAM C21EとSAM C20E (32ピン) : 16までのチャンネル
- SAM C21GとSAM C20G (48ピン) : 22までのチャンネル
- SAM C21JとSAM C20J (64ピン) : 32までのチャンネル
- SAM L22G (48ピン) : 15までのチャンネル
- SAM L22J (64ピン) : 19までのチャンネル
- SAM L22N (100ピン) : 24までのチャンネル
- ATmega328PB (32ピン) : 24までのチャンネル
- ATmega324PB (44ピン) : 32までのチャンネル



Y感知線は不連続的な順番でDEF_SELFCAP_LINES構成設定パラメータを用いて指定することができます。チャンネル番号付けはDEF_SELFCAP_NODES構成設定パラメータで指定したY感知線と同じ順番で行われます。

設定	構成設定名	データ型	単位	Y線数		
				最小	最大	代表
自己容量接触チャンネル線	DEF_SELFCAP_LINES	uint16_t配列	なし	1	32	–
自己容量チャンネル数	DEF_SELFCAP_NUM_CHANNELS	uint16_t	なし	1	32	–
自己容量感知部数	DEF_SELFCAP_NUM_SENSORS	uint16_t	なし	1	32	–
自己容量回転子と摺動子の数	DEF_SELFCAP_NUM_ROTORS_SLIDERS	uint8_t	なし	0	10	–

接触感知部はtouch_xx_sensor_config()APIを用いて指定したチャンネルの連続的な順番で許可されなければなりません。改善されたEMC性能のため、X線とY線で1kΩの値を持つ直列抵抗が使われなければなりません。接触感知部の設計についてのより多くの情報に関してはwww.atmel.comで入手可能な「釐、摺動子、輪接触感知部設計の手引き」を参照してください。

5.2.2. 感知部の個別パラメータ

本項は個別感知部に対して指定する設定を説明します。

検出閾値

感知部の検出閾値は可能性のある接触検出としての品質に対してその信号がどれ位その参照基準水準以上に増加しなければならないかを定義します。けれども、最終的な検出確認は検出積分(DI:Detect Integrator)限度を満足させなければなりません。より大きな閾値は信号が閾値水準を超えるためにもっと変化しなければならない(即ち、より大きな接触が必要な)ので、感知部を鈍感にします。逆に、より低い閾値は感知部をもっと敏感にします。

閾値設定は感知部が接触される時の信号の振れる量に依存します。通常、より厚い前面パネルまたはより小さな電極は接触でより小さな信号振幅を持ち、故により低い閾値水準を必要とします。代表的に、検出閾値は接触差の50%に設定されます。釘に対して望まれる接触差は概ね30～80計数で、輪や摺動子に対しては概ね50～120計数です。

設定	構成設定名	データ型	単位	最小	最大	代表
閾値	detect_threshold	threshold_t	計数	3	255	20～50 (釘用) 30～80 (摺動子/輪用)

検出ヒステリシス

この設定は感知部検出ヒステリシスです。これは感知部検出閾値設定の%として表されます。一旦感知部が検出に行くと、信号水準が基の閾値水準に近い場合に感知部が検出の出入りで狼狽えるのを避けるために、その閾値水準が(ヒステリシス値によって)減らされます。

- ・ 0の設定 = 検出閾値の50% (HYST_50)
- ・ 1の設定 = 検出閾値の25% (HYST_25)
- ・ 2の設定 = 検出閾値の12.5% (HYST_12_5)
- ・ 3の設定 = 検出閾値の6.25% (HYST_6_25)

設定	構成設定名	データ型	単位	最小	最大	代表
ヒステリシス	detect_hysteresis	uint8_t (2ビット)	列挙	HYST_6_26	HYST_50	HYST_6_26

位置分解能

回転子と摺動子は位置分解能(回転子の場合での角度分解能と摺動子での場合の直線分解能)を設定する必要があります。分可能は回転子や摺動子の位置を報告するのに必要とされるビット数です。これは2～8ビットの値を持つことができます。

設定	構成設定名	データ型	単位	ビット数 (報告位置範囲)		
				最小	最大	代表
位置分解能	position_resolution	uint8_t (3ビット)	なし	2 (0～3)	8 (0～255)	8 (0～255)

位置ヒステリシス

相互容量に於いて、回転子や摺動子は位置ヒステリシス(回転子の場合での角度ヒステリシスと摺動子の場合での直線ヒステリシス)を設定する必要があります。これは使用者が押した後でスクロールの方向が変更される時と最初にスクロールする間で接触位置が報告される前に使用者が逆行しなければならない位置数です。

ヒステリシスは0(1つの位置)～7(8つの位置)の範囲にすることができます。ヒステリシスは内部的に8ビット分解能で実行されて望む分解能に尺度調整され、従って、8ビットよりも低い分機能で、接触が検出された場所に依存して、ヒステリシス設定から1報告位置の違いが有るかもしれません。報告する位置の飛びが観測されるより低い分機能では、ヒステリシスを0(1つの位置)に設定することができます。より高い分解能(6～8ビット)では、最低2つの位置またはそれ以上のヒステリシスを持つことが推奨されるでしょう。

注: 分解能で利用可能な位置よりも大きなヒステリシス値を持つことは有効ではありません。例えば、2ビット(4つの位置)の分解能での5つの位置のヒステリシス値は無効です。位置ヒステリシスは自己容量法感知部の場合に無効(不使用)です。

設定	構成設定名	データ型	単位	最小	最大	代表
位置ヒステリシス	position_hysteresis	uint8_t (3ビット)	なし	0	7	7

隣接キー消去 (AKS®)

感知部が共に近い、または高い感度に構成設定される設計では複数の感知部が同時に検出を報告するかもしれません。意図する単一接触を決めるのを応用に許すため、接触ライブラリはAKS群で或る数の感知部を構成設定する能力を提供します。

感知部の群が同じAKS群の時に、最強の感知部だけが検出を報告します。その感知部の検出報告は例え別の感知部の差がより強くなっても検出の報告を続けます。この感知部はその差がその検出閾値よりも低く落ちるまで検出に留まります。このAKS群内のいくつかの感知部が未だ検出の場合、最強のものだけが検出を報告します。任意の時点で、各AKS群から1つの感知部だけが接触と報告されます。

AKS機能はDEF_XXXXCAP_AKS_ENABEマクロを用いて許可または禁止することができます。

- 1u = AKS群化機能が許可されます。
- 0u = AKS群化機能が禁止されます。

ライブラリは隣接キー消去群(AKS群)の1つに属するように感知部を構成設定する能力を提供します。

5.2.3. 感知部の全体パラメータ

本項は全感知部共通の設定を説明します。例えば、相互容量感知部の再校正閾値(全体設定の1つ)はRECAL_100として設定され、全ての相互容量感知部は100%の再校正閾値に構成設定されます。これらの感知部全体パラメータ設定は自己容量と相互容量の感知部に対して独立して設定することができます。

検出積分

QTouchライブラリは堅牢な環境で検出を確認する検出積分機構が特徴です。検出積分器(DI:Detect Integrator)は電気的な雑音のような偽の事象によって引き起こされる誤った検出を消去するための簡単な信号濾波器として働きます。

感知部の差がその閾値を超える毎に計数器が増され、閾値水準以下に行くのを除き、特定数の採取間それに留まります。この計数器が予め設定した限度(DI値)に達すると、感知部は最終的に接触したことを宣言されます。何れかの採取で差が閾値水準を下回った場合、計数器が解消されて、処理は初めから開始されなければなりません。DI処理は'開放'(検出外)事象にも適用されます。

例えば、DI値が10の場合、デバイスはその閾値を超えて感知部が接触したことを宣言されるのに先立って、閾値水準以下になることなく、(10+2)連続採取間そこに留まらなければなりません。

設定	構成設定名	データ型	単位	最小	最大	代表
検出積分	DEF_MUTLCAP_DI, DEF_SELFCAP_DI	uint8_t	周回	0	255	4

最大ON期間

感知部が長期間の接触検出に帰着する、物が意図せず接触する場合、数秒の遅延時間後、その機能を回復するために通常、感知部を再校正することが望まれます。

最大ON期間計時器は検出のようなものを監視し、検出が計時器の設定を超えた場合に感知部が自動的に再校正されます。再校正が行われた後、影響を及ぼされた感知部は例えそれが未だ外来物と接触していても、一旦再び正常に機能します。

最大ON期間は継続的な検出中にチャネルが決して再校正しない場合にこれを0(無制限制限時間)に設定することによって禁止することができます(しかしホストは未だ再校正を指示することができます)。

設定	構成設定名	データ型	単位	最小	最大	代表
最大ON期間	DEF_MUTLCAP_MAX_ON_DURATION, DEF_SELFCAP_MAX_ON_DURATION	uint8_t	200ms	0	255	30 (6s)

接触を離れて接触変動の方へ

一般的な感知部での変動は物理的な感知部特性での温度(または他の要素)の影響に対する補償を許すための(感知部の)参照基準水準調整を意味します。このような補償に対する参照基準水準減少は負の変動と呼ばれ、参照基準水準増加は正の変動と呼ばれます。具体的に、変動補償は信号を減らすよりも信号を増やす方により速く補償するように設定されるべきです。

信号は時間と温度に渡る物理的な感知部特性での変化のために変動し得ます。このような変動が補償されることが重要で、さもなければ、誤った検出と感度変化が起こり得ます。

変動補償は事実上、検出がない間にだけ起きます。一旦指が感知されると、信号が正当に物体を検出するため、変動補償機構は停止します。変動補償は問題の信号が'検出閾値'水準を超えていない時にだけ動きます。

変動補償機構は非対称で有り得ます。これは単に適切な構成設定パラメータを変えることによって、他のそれよりも速く或る方向で発生させることができます。

感知部の信号値は(接触)物体がそれに近接する時、または時間と温度に渡って感知部の特性変化で増加する傾向にあります。例えばチャネルに接触する(接触変動に向かう)前でも接近する指が部分的または完全に補償され得るため、増加する信号は素早く補償されるべきではありません。

けれども、感知部が既に(或る期間に渡って)完全な許容にされ、検出を起こさないチャネル上の物体は、人為的に抑制された参照基準水準を持つ感知部を残して突然に取り去られるかもしれず、故に接触に鈍感になります。後者の場合、(接触変動から離れて)相対的に素早く参照基準水準を上げることによって感知部は物体の移動に対して補償されるべきです。

設定	構成設定名	データ型	単位	最小	最大	代表
接触変動に向かう	DEF_MUTLCAP_TCH_DRIFT_RATE, DEF_SELFCAP_TCH_DRIFT_RATE	uint8_t	200ms	0	127	20 (4s)
接触変動から離れる	DEF_MUTLCAP_ATCH_DRIFT_RATE, DEF_SELFCAP_ATCH_DRIFT_RATE	uint8_t	200ms	0	127	5 (1s)

変動保持期間

変動保持時間(DHT:Drift Hold Time)は1つ以上の感知部が活性にされる間で全ての感知部での変動を制限するのに使われます。これはキーが検出された後で変動が停止される時間の長さを定義します。この機能はキーに接触する、またはキーパッド上に指を浮かせる高密度キーパッドが未接触のキーを変動させ、従って感度変化を引き起こし、最終的に接触検出を禁止する場合に有用です。

設定	構成設定名	データ型	単位	最小	最大	代表
変動保持時間	DEF_MUTLCAP_DRIFT_HOLD_TIME, DEF_SELFCAP_DRIFT_HOLD_TIME	uint8_t	200ms	0	255	20 (4s)

接触離脱再校正閾値

再校正閾値はこれを超えるレベルが自動的に再校正を引き起こします。再校正閾値は検出閾値設定の%として表されます。

この設定は列挙された値でその設定は次のとおりです。

- 0の設定 = 検出閾値の100% (RECAL_100)
- 1の設定 = 検出閾値の50% (RECAL_50)
- 2の設定 = 検出閾値の25% (RECAL_25)
- 3の設定 = 検出閾値の12.5% (RECAL_12_5)
- 4の設定 = 検出閾値の6.25% (RECAL_6_25)

けれども、絶対値4がこの設定のハード限界です。例えば、検出閾値が40で再校正閾値が4に設定される場合、これは絶対値2($40 \times 6.25\% = 2.5$)を意味しますが、これはハード限界の4に制限されます。

設定	構成設定名	データ型	単位	最小	最大	代表
接触離脱再校正閾値	DEF_MUTLCAP_ATCH_RECAL_THRESHOLD, DEF_SELFCAP_ATCH_RECAL_THRESHOLD	uint8_t	列挙	RECAL_6_25	検出閾値	RECAL_100

接触離脱再校正遅延

どれかのキーが重大な負の差を持つことを見つけた場合、それは異常状態と見做されます。この状態が接触離脱再校正遅延、即ち、`qt_pos_recal_delay`期間よりもっと持続する場合、自動再校正が実行されます。

計数器は感知部差が接触離脱再校正閾値と等しくて特定採取数間それに留まる毎に増やされます。この計数器が予め設定された限度(PRD値)に達すると、感知部は最終的に再校正されます。どれかの採取で差が接触離脱再校正閾値水準よりも大きいと気づいたなら、計数器は解消されて接触変動離脱が実行されます。

例えば、接触離脱再校正遅延設定が10の場合、感知部が再校正されたことを宣言される前に、差は再校正閾値以下に落ちて閾値水準以下になることなく連続で10採取間それに留まらなければなりません。接触離脱再校正は0の設定で禁止することができます。

設定	構成設定名	データ型	単位	最小	最大	代表
接触離脱再校正遅延	DEF_MUTLCAP_ATCH_RECAL_DELAY, DEF_SELFCAP_ATCH_RECAL_DELAY	uint8_t	周回	0	255	10

感知部後処理動作

`TOUTCH_LIBRARY_DRIVEN`動作が選択されると、ライブラリは接触の押下、開放、校正を解決するために繰り返し接触測定を自己開始します。この動作は最良の応答時間に適します。

`TOUCH_APPLN_DRIVEN`動作が選択されると、ライブラリは接触の押下、開放、校正を解決するために繰り返し接触測定を開始しません。この動作は厳重なCPU時間必要条件を要求する応用に対して確定的なPTC割り込み実行時間を適合させます。これは実行されつつある他の重要な応用コードのため、繰り返し接触測定が遅らされるためです。この動作は潜在的に接触応用時間に影響を及ぼし得ます。

`TOUCH_APPLN_DRIVEN`動作で接触応答時間を改善するため、接触測定を周期的に、または再び群発採取状態フラグが設定された時に`touch_XXXXcap_sensors_measure`API呼び出しは次のように変更されるべきです。


```

if ((touch_time.time_to_measure_touch == 1u) || (p_mutlcap_measure_data->acq_status & ¥
TOUCH_BURST_AGAIN)
{
    /* 接触感知部測定処理開始 */
    touch_ret =
touch_mutlcap_sensors_measure(touch_time.current_time_ms, NORMAL_ACQ_MODE, ¥
touch_mutlcap_measure_complete_callback);
}

```

設定	構成設定名	データ型	任意選択	代表
感知部後処理動作	DEF_MUTLCAP_TOUCH_POSTPROCESS_MODE, DEF_SELFCAP_TOUCH_POSTPROCESS_MODE	uint16_t	TOUCH_LIBRARY_DRIVEN, TOUCH_APPLN_DRIVEN	TOUCH_LIBR ARY_DRIVEN

充電共用遅延

充電共用遅延(CSD:Charge Share Delay)は接触感知部が完全に充電されることを確実にするために容量測定周回内に挿入される追加の充電周回数を示します。CSD値はY線での直列抵抗と共に感知部容量に依存します。

注: 充電共用遅延でどの増加も指定構成設定に対して測定時間を増加します。

手動調整が実行される時に、直列抵抗と共に容量の最大組み合わせでの感知部に対するCSD値が考慮されるべきです。

設定	構成設定名	データ型	任意選択	最小	最大	代表
CSD(充電共用遅延)	DEF_MUTL_CAP_CSD_VALUE, DEF_SELF_CAP_CSD_VALUE	uint8_t	PTC周回	0	250	0

手動でCSD設定を調整するには?

1. 始めに、64のような任意の大きな値を使って信号値を記録してください。大きな値は充電時間が完全な電荷移動に十分なことを保証します。
2. CSDを減らして信号が始めに使った値の概ね97～98%になるまで、信号値低下を確認してください。これは信号での主などんな損失もなしでの良好な電荷移動を保証します。
3. システムで利用可能な全ての感知部に対して同じ手順(手順1と2)を続けてください。全体設定に対してはシステムで使われたCSDの最大値を使ってください。

注: 同じCSD設定に対して、相互容量は自己容量よりも低い群発時間を持ちます。相互容量CSDでの1単位増加は約12PTC周回を費やします。一方自己容量に対するCSDでの増加は同じ設定を持つ相互容量CSD時間の概ね2倍を費やします。

自動OS信号安定限度

DEF_XXXXCAP_AUTO_OS_SIGNAL_STABILITY_LIMITパラメータは過採取を実行するための信号の安定限度を定義します。安定限度は雑音環境下の感知部信号値での偏差です。高い水準の安定限度は大きな雑音存在での過採取自動起動に設定します。これはこの設定をシステムの最低感知部検出閾値近くに保って更に雑音に基づいて調節することが推奨されます。

範囲：1～1000

5.2.4. 感知部採取パラメータ

濾波水準

濾波水準設定は各採取を解決するのにかかる採取数を制御します。より高い濾波水準設定は雑音条件下で改善された信号雑音比を提供し、一方で測定に対する総時間の増加が電力消費と応答時間の増加に帰着します。この設定は基本チャネル毎に利用可能で容易な調整を許します。

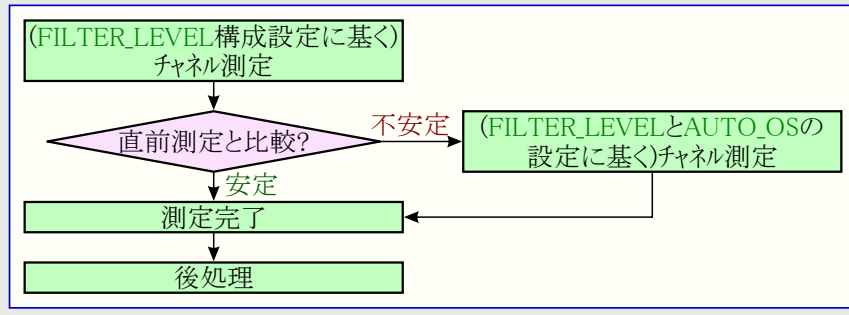
設定	構成設定名	データ型	単位	最小	最大	代表
濾波水準	DEF_MUTLCAP_FILTER_LEVEL_PER_NODE, DEF_SELFCAP_FILTER_LEVEL_PER_NODE	filter_level_t	採取数	0	64	16

自動過採取

自動過採取は'濾波水準'の既定設定で不安定な信号が検出された時の感知部チャネルの自動過採取を制御します。自動過採取の許可は不安定な信号が観測される時に、対応する感知部チャネルで取られる'濾波水準'×'自動過採取'に帰着します。'濾波水準'がFILTER_LEVEL_4に設定され、'自動過採取'がAUTO_OS_4に設定される場合、安定信号で4過採取が取られ、不安定な信号が検出される時に16過採取が取られます。この設定は基本チャネル毎に利用可能で容易な調整を許します。

より高い濾波水準設定は雑音条件下で改善された信号雑音比を提供し、一方で測定に対する総時間の増加が電力消費と応答時間の増加に帰着します。

図5-7. 自動過採取



自動過採取は最良の電力消費を達成するために禁止することができます。

設定	構成設定名	データ型	任意選択	最小	最大	代表
自動過採取	DEF_MUTLCAP_AUTO_OS_PER_NODE, DEF_SELFCAP_AUTO_OS_PER_NODE	auto_os_t	採取倍率	2	128	AUTO_OS_NONE

利得設定

利得設定は接触での接触差の拡大を許すために基本チャンネル毎に適用されます。利得設定は感知部設計と接触パネル厚に依存します。

設定	構成設定名	データ型	任意選択	最小	最大	代表
利得	DEF_MUTLCAP_GAIN_PER_NODE, DEF_SELFCAP_GAIN_PER_NODE	gain_t	利得倍率	1	32	1(自己容量) 4(相互容量)

下表は与えられた利得設定と濾波水準の設定組み合わせに対して期待される信号値を示します。提供される値は兆候だけで実際の感知部信号値はおそらく提言値に近くなるでしょう。

表5-8. FILTER_LEVELとGAINの組み合わせに対する平均設定信号値

FILTER_LEVELとGAIN の組み合わせ	平均設定信号値 (S利得=ソフトウェア利得(デジタル利得), H利得=ハードウェア利得(アナログ利得))					
	GAIN_1	GAIN_2	GAIN_4	GAIN_8	GAIN_16	GAIN_32
FILTER_LEVEL_1	512 (0,0)	512 (0,1)	512 (0,2)	512 (0,3)	512 (0,4)	512 (0,5)
FILTER_LEVEL_2	512 (0,0)	1024 (1,0)	1024 (1,1)	1024 (1,2)	1024 (1,3)	1024 (1,4)
FILTER_LEVEL_4	512 (0,0)	1024 (1,0)	2048 (2,0)	2048 (2,1)	2048 (2,2)	2048 (2,3)
FILTER_LEVEL_8	512 (0,0)	1024 (1,0)	2048 (2,0)	4096 (3,0)	4096 (3,1)	4096 (3,2)
FILTER_LEVEL_16	512 (0,0)	1024 (1,0)	2048 (2,0)	4096 (3,0)	8192 (4,0)	8192 (4,1)
FILTER_LEVEL_32	512 (0,0)	1024 (1,0)	2048 (2,0)	4096 (3,0)	8192 (4,0)	16384 (5,0)
FILTER_LEVEL_64	512 (0,0)	1024 (1,0)	2048 (2,0)	4096 (3,0)	8192 (4,0)	16384 (5,0)
推奨	優秀			良好	貧弱	

前置分周器設定

前置分周器パラメータは特定チャンネルに対するクロック分周器を示します。これは基本チャンネル毎に設定することができ、各感知部節点/チャンネルに対して独立です。このパラメータは自動調節設定に基づいて自動的に調節されます。このパラメータの調節は改善された雑音性能を許します。

設定	構成設定名	データ型	任意選択	最小	最大	代表
前置分周器	DEF_MUTLCAP_CLK_PRESCALE_PER_NODE, DEF_SELFCAP_CLK_PRESCALE_PER_NODE	prsc_div_sel_t	PRSC_DIV_SEL_1, PRSC_DIV_SEL_2, PRSC_DIV_SEL_4, PRSC_DIV_SEL_8	PRSC_DIV_SEL_1	PRSC_DIV_SEL_8	PRSC_DIV_SEL_1

直列抵抗設定

直列抵抗は採取のために特定チャネルで使われる抵抗器を示します。この値は調節可能で自動と手動の両方の調節任意選択を許します。このパラメータの調節は改善された雑音性能を許します。

設定	構成設定名	データ型	任意選択	最小	最大	代表
直列抵抗	DEF_MUTLCAP_SENSE_RESISTOR_PER_NODE, DEF_SELFCAP_SENSE_RESISTOR_PER_NODE	rsel_val_t	RSEL_VAL_0, RSEL_VAL_20, RSEL_VAL_50, RSEL_VAL_100	RSEL_VAL_0	RSEL_VAL_100	RSEL_VAL_100

ブート前置分周器設定

ブート前置分周器パラメータは特定チャネルに対するクロック分周器を示します。これは基本チャネル毎に設定することができ、各感知部節点/チャネルに対して独立です。この設定は電源ON後の感知部校正に使われます。このパラメータは自動調節が利用できない時に構成設定されなければなりません。

設定	構成設定名	データ型	任意選択	最小	最大	代表
ブート 前置分周器	DEF_MUTLCAP_CC_CAL_CLK_PRESCALE_PER_NODE, DEF_SELFCAP_CC_CAL_CLK_PRESCALE_PER_NODE	prsc_div_sel_t	PRSC_DIV_SEL_1, PRSC_DIV_SEL_2, PRSC_DIV_SEL_4, PRSC_DIV_SEL_8	PRSC_DIV_SEL_1	PRSC_DIV_SEL_8	PRSC_DIV_SEL_1

ブート直列抵抗設定

ブート直列抵抗はデバイス電源ON校正での特定チャネルで使われます。このパラメータは自動調節が利用できない時に構成設定されなければなりません。

設定	構成設定名	データ型	任意選択	最小	最大	代表
ブート 直列抵抗	DEF_MUTLCAP_CC_CAL_SENSE_RESISTOR_PER_NODE, DEF_SELFCAP_CC_CAL_SENSE_RESISTOR_PER_NODE	rsel_val_t	RSEL_VAL_0, RSEL_VAL_20, RSEL_VAL_50, RSEL_VAL_100	RSEL_VAL_0	RSEL_VAL_100	RSEL_VAL_100

周波数動作形態

周波数動作構成設定は環境雑音に対抗するためにPTC接触採取周波数特性を調節することを使用者に許します。

FREQ_MODE_HOP

周波数飛び跳ね動作任意選択が選ばれると、PTCはDEF_SELFCAP_HOP_FREQSで指定されるように3つのPTC採取周波数遅延設定を用いて行われる後続測定を持つ周波数飛び跳ね周回を進行します。この場合、測定された信号値に追加のソフトウェア中央値濾波が適用されます。

FREQ_MODE_SPREAD_MEDIAN

周波数スペクトラム中央拡散動作任意選択が選ばれると、PTCは許可されたスペクトラム拡散で動きます。この場合、測定された信号値に追加のソフトウェア中央値濾波が適用されます。

FREQ_MODE_NONE

周波数動作なし任意選択が選ばれると、PTCは定速で動きます。この動作は最善の電力消費に適します。

設定	構成設定名	データ型	任意選択	最小	最大	代表
周波数 動作形態	DEF_MUTLCAP_FREQ_MODE, DEF_SELFCAP_FREQ_MODE	freq_mode_sel_t	FREQ_MODE_NONE, FREQ_MODE_HOP, FREQ_MODE_SPREAD, FREQ_MODE_SPREAD_MEDIAN	FREQ_MODE_NONE	FREQ_MODE_SPREAD_MEDIAN	FREQ_MODE_NONE

周波数飛び跳ね遅延

周波数飛び跳ね遅延設定は周波数動作形態がFREQ_MODE_HOPに設定される時に使われます。3つの周波数飛び跳ね遅延設定の組が指定されるべきです。この遅延設定は与えられた感知部で連続する測定間にn個のPTCクロック周期を挿入し、それによってPTC採取周波数を変えます。FREQ_HOP_SEL_1設定は連続する測定間に0個のPTCクロック周期を挿入します。FREQ_HOP_SEL_16設定は15個のPTCクロック周期を挿入します。故に、より大きい遅延設定はより小さな遅延設定に比べて与えられた感知部での容量測定にかかる総時間を増します。採取周波数と同じ周波数付近の雑音を避けるために望む設定を使うことができます。

設定	構成設定名	データ型	単位	最小	最大	代表
周波数 飛び跳ね遅延	DEF_MUTLCAP_HOP_FREQS, DEF_SELFCAP_HOP_FREQS	freq_hop_sel_t	PTC クロック周期	FREQ_HOP_SEL_1	FREQ_HOP_SEL_16	FREQ_HOP_SEL_1, FREQ_HOP_SEL_2, FREQ_HOP_SEL_3

5.2.5. 感知部校正自動調節設定

自動調節パラメータ設定は電力消費または雑音性能のためにPTC単位部の調整を使用者に許すため、touch_xx_sensors_calibrate APIへ渡されます。

AUTO_TUNE_PRSC

前置分周器の自動調節が選ばれると、PTCは使用者定義内部直列抵抗設定(DEF_XXXXCAP_SENSE_RESISTOR_PER_NODE)を使い、前置分周器は完全な電荷移転を確実にするためにPTC動作を減速に調節されます。直列抵抗としてRSEL_VAL_100を持つ前置分周器の調節は最低の電力消費に帰着し、一方で電力消費と接触応答時間の増加に帰着します。

AUTO_TUNE_RSEL

直列抵抗の自動調節が選ばれると、PTCは使用者定義前置分周器設定速度(DEF_XXXXCAP_CLK_PRESCALE_PER_NODE)で動き、完全な電荷移転を許すように内部直列抵抗が自動的に最適に調節されます。PTC前置分周としてPRSC_DIV_SEL_1を持つ直列抵抗の自動調節は最良の場合の電力消費に帰着します。

AUTO_TUNE_NONE

手動調節任意選択が選ばれると、DEF_XXXXCAP_CLK_PRESCALE_PER_NODEとDEF_XXXXCAP_SENSE_RESISTOR_PER_NODEで与えられるようなPTCの前置分周器と直列抵抗の使用者定義値がPTC動作に使われます。

設定	構成設定名	データ型	単位	最小	代表
自動調節	touch_xxcap_sensors_calibrate API入力に対して提供される	auto_tune_type_t	なし	AUTO_TUNE_NONE, AUTO_TUNE_PRSC, AUTO_TUNE_RSEL	AUTO_TUNE_NONE

5.2.6. 感知部雑音測定と閉鎖パラメータ

雑音は逐次測定の移動窓での過去データを用いて各チャネル採取後に基本チャネル毎に測定されます。加えまたは取り去った接触の実体を排除するために報告された雑音、但し、雑音指示は雑音閉鎖が防止されるのに先立って偽の接触検出に充分速く反応しなければなりません。

窓緩衝部が安定限度と比較される間、信号は採取毎に変わります。雑音は窓期間内で2つ変化が起こり、その両方がDEF_XXXXCAP_NOISE_MEAS_SIGNAL_STABILITY_LIMIT限度を超える時にだけ報告されます。

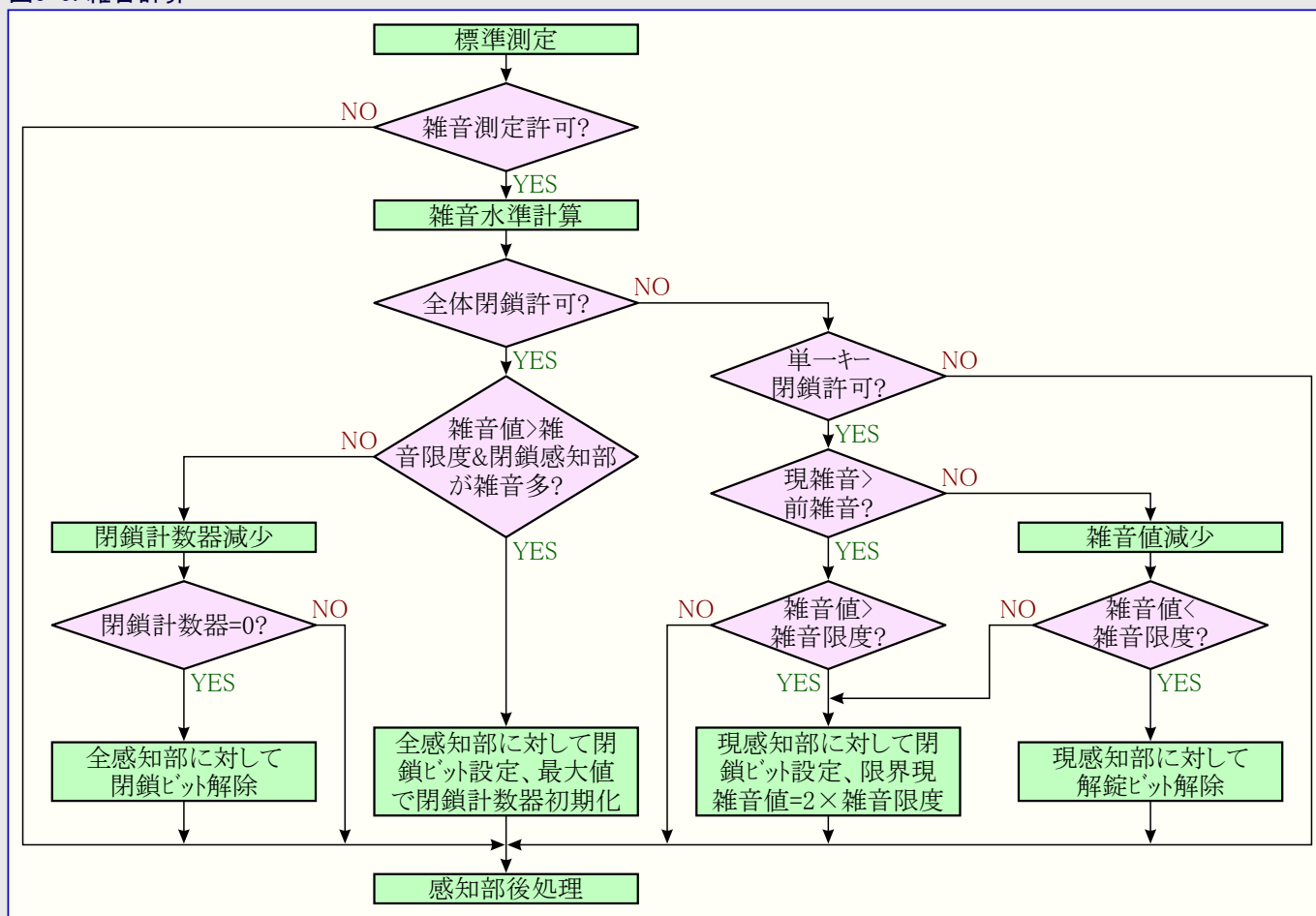
雑音は以下の算法を用いて計算されます。

```
if (swing count > 2)
{
    Nk = ((|Sn - Sn-1| > DEF_XXXXCAP_NOISE_MEAS_SIGNAL_STABILITY)) ? (0) : (|Sn-Sn-1| - ¥
        DEF_XXXXCAP_NOISE_MEAS_SIGNAL_STABILITY))
}
else
{
    Nk = 0
}
```

swing countは緩衝部窓期間中にDEF_MUTLCAP_NOISE_MES_SIGNAL_STABILITY_LIMIT限度を超えた信号変化数です。

測定した雑音がDEF_MUTLCAP_NOISE_LIMITを超えると、接触ライブラリは感知部を閉鎖し、接触検出なしを報告して変動が停止されます。雑音測定は全てのチャネルに対して提供されます。p_xxxxcap_measure_data->p_nm_ch_noise_val内の各バイトはそのチャネルに関連する雑音水準を提供します。雑音指示は応用によって構成設定された全ての感知部に対して提供されます。雑音が多いかどうかを感知部が判断するのに各感知部に対するp_xxxxcap_measure_data->p_sensor_noise_status内のビットが利用可能です。次のコード断片は特定感知部の雑音状態を読むための見本コードを提供します。

図5-9. 雑音計算



雑音測定信号安定限度

DEF_XXXXCAP_NOISE_MEAS_SIGNAL_STABILITY_LIMITパラメータは雑音環境下の感知部信号値での相違です。雑音信号安定限度を超えるなどの雑音水準も雑音限度の一因になります。

この設定をシステムの最低感知部検出閾値近くに保ち、雑音に基づいて更に調節することが推奨されます。

信号値は窓緩衝部期間中の採取から採取で変わり得ます。隣接緩衝部値間の違いは使用者が構成設定した安定限度と比較されません。

雑音は指定した窓期間内で2つの変化が起きて両方が安定限度を超える場合にだけ報告されます。

範囲：1～1000

雑音限度

DEF_XXXXCAP_NOISE_LIMITパラメータは雑音緩衝数に渡って累積された総合雑音に対する限度を指定します。累積した雑音雑音限度を超えた場合、閉鎖が起動されます。このパラメータには次の2つの目的があります。

- ・動いている窓の間で計算された雑音水準がDEF_XXXXCAP_NOISE_LIMITを超えた場合、対応する感知部は雑音を宣言され、感知部全体雑音ビットが'1'として設定されます。
- ・動いている窓の間で計算された雑音水準がDEF_XXXXCAP_NOISE_LIMITを超えた場合、システムは感知部閉鎖機能を起動します。

範囲：1～255

雑音測定緩衝数

DEF_XXXXCAP_NOISE_BUFFER_CNTパラメータは雑音測定緩衝部に対する緩衝数を選ぶのに使われます。

範囲：3～10 (N個の採取の違いを調べなければならない場合はこのパラメータを"N+1"として定義します。)

N=4ならば、DEF_XXXXCAP_NOISE_MEAS_BUFFER_CNTを5uとして設定してください。

感知部閉鎖選択

この機能は測定した雑音がDEF_XXXXCAP_NOISE_LIMITを超えて接触が報告されない時に感知部を閉鎖します。これは後処理を止めます。故に、高い水準の雑音が偽の接触変動を報告したり不正な再校正をすることをチャネルにさせ得ません。

DEF_XXXXCAP_LOCKOUT_SELパラメータは閉鎖機能方法を選ぶのに使われます。

- DEF_XXXXCAP_LOCKOUT_SELがSINGLE_SENSOR_LOCKOUTに設定されて感知部の雑音水準がDEF_XXXXCAP_NOISE_LIMITよりも大きい場合、対応する感知部が接触検出から締め出されて変動が禁止されます。
- DEF_XXXXCAP_LOCKOUT_SELがGLOBAL_SENSOR_LOCKOUTに設定されてどれかの感知部の雑音水準がDEF_XXXXCAP_NOISE_LIMITよりも大きい場合、全ての感知部が接触検出から締め出されて変動が禁止されます。
- DEF_XXXXCAP_LOCKOUT_SELがNO_LOCKOUTに設定された場合、閉鎖機能が禁止されます。

注: 全体感知部雑音ビットはSINGLE_SENSOR_LOCKOUTとGLOBAL_SENSOR_LOCKOUTに対して利用可能です。全体感知部雑音ビットはNO_LOCKOUTにに対して利用不可です。

範囲：0～2

感知部閉鎖秒読み

感知部信号が雑音性から良好な状況に移動してDEF_XXXXCAP_LOCKOUT_CNTDOWN数の測定間それに留まる場合、その感知部は解錠されて感知部は接触検出の準備が整って変動が許可されます。

注: このパラメータは全体閉鎖に対してだけ有効です。

範囲：1～255

5.2.7. 感知部採取周波数自動調節パラメータ

周波数自動調節機能はFREQ_MODE_HOP動作で最良のSNRを示す採取周波数の自動選択によって接触検出に対する信号データの最高品質を提供します。各測定周回中、同じ周波数での最後の採取からの信号変化が各感知部に対して記録されます。その周回後、全ての感知部が現在の採取周波数で測定されてしまうと、全感知部の最大信号変化がその周波数段に対する相違として格納されます。

各周波数段に対する相違はDEF_XXXXCAP_FREQ_AUTO_SIGNAL_STABILITY_LIMIT限度と比較され、この限度を超えた場合、段毎の計数器が増加されます。測定した相違が限度よりも小さい場合、これが0として設定されていないなら、この計数器は減少されます。全ての周波数が安定限度を超える雑音を示す場合、最高相違を持つ特定周波数段に対する計数器だけがその周回後に増加されます。

周波数計数器がDEF_XXXXCAP_FREQ_AUTO_TUNE_IN_CNT(自動調節計数変数)に達すると、その周波数段は自動調節用に使われます。新しい周波数選択が適用され、全ての周波数に対して計数器と相違がリセットされます。周波数が自動調節用に使われた後、その周波数段に対する計数は元の計数の半分に設定され、全ての周波数が測定されてしまうか、または自動調節を再起動しないと判定された周波数が選ばれるのどちらかまで処理が繰り返されます。

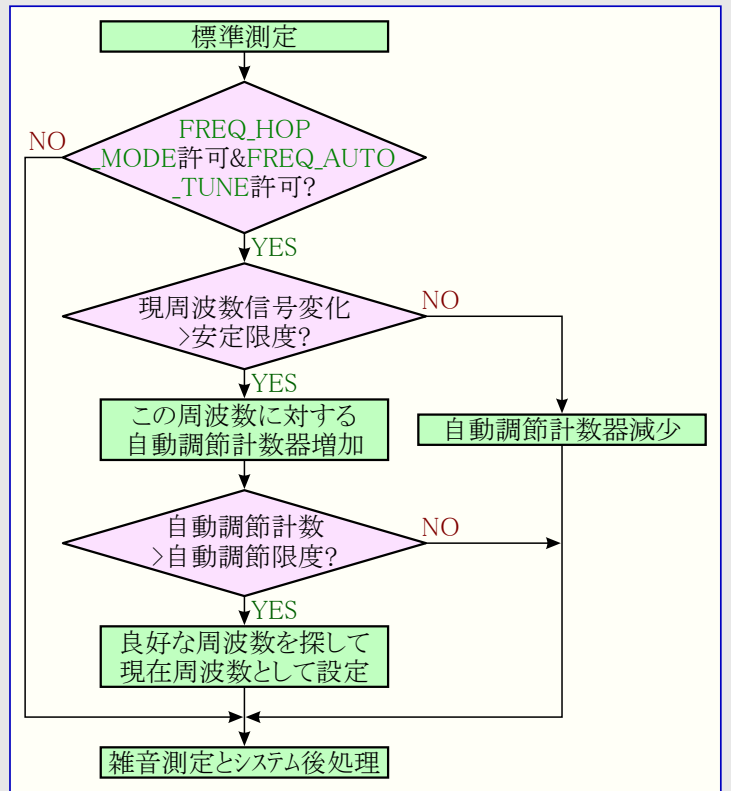
全ての周波数が調べられて相違がDEF_XXXXCAP_FREQ_AUTO_SIGNAL_STABILITY_LIMIT限度を超える場合、現在調節下の周波数段に最低相違を持つ周波数が選ばれます。自動調節処理が再初期化され、更なる自動調節は周波数段の相違計数器が再び計数限度に達するまで実行しません。

周波数自動調節信号安定性

DEF_XXXXCAP_FREQ_AUTO_SIGNAL_STABILITYは雑音環境下の感知部信号値での相違です。雑音信号安定限度水準は雑音の存在で採取周波数を自動調節するために設定されます。この設定をシステムの最低感知部検出閾値近くに保ち、雑音に基づいて更に調節することが推奨されます。

範囲：1～1000

図5-10. 周波数自動調節



周波数自動調節計数器

DEF_XXXXCAP_FREQ_AUTO_TUNE_IN_CNTパラメータは周波数自動調節を起動するのに使われます。各周波数で感知部信号変化がDEF_XXXXCAP_FREQ_AUTO_TUNE_IN_CNTの間DEF_XXXXCAP_FREQ_AUTO_SIGNAL_STABILITY_LIMITとして指定された値を超える場合、この周波数で周波数自動調節が起動されます。

範囲：1～1000

注：周波数自動調節機能と関連するパラメータはFREQ_MODE_HOP動作でだけ利用可能です。

5.2.8. 迅速再群発パラメータ

迅速再群発

このマクロは迅速再群発機能を許可または禁止するのに使われます。迅速再群発が許可されると、使用者の接触と開放で、検出積分(または跳ね返り抑制)を解決するために接触された感知部またはチャネルだけがその後に測定されます。この機能の許可は最良の接触応答時間に帰着します。

迅速再群発が禁止されると、使用者の接触と解放で、検出積分(または跳ね返り抑制)を解決するために全ての感知部またはチャネルが測定されます。この機能は使用者活動中に全ての感知部測定を伴う何れかの特別な応用を開発する時にだけ禁止されるべきです。

AKS(隣接キー消去)内では、この機能が許可または禁止にされているのと無関係にその群内の全ての感知部が使用者接触中に測定されます。

5.2.9. 共通パラメータ

測定期間

測定期間設定は接触感知部測定に対する周期的な間隔を設定するのに使われます。最小測定期間設定は全感知部で測定を完了するのにかかる時間より大きくすべきです。これはWhile繰り返してtouch_xx_sensors_measure APIを呼び、測定完了呼び戻しでGPIOピンを交互切り替えることによって簡単に決めることができます。

```
main()
{
    while(1)
    {
        touch_ret = touch_mutlcap_sensors_measure(touch_time.current_time_ms, NORMAL_ACQ_MODE, &
            touch_mutlcap_measure_complete_callback);
    }
}

void touch_mutlcap_measure_complete_callback( void )
{
    if (!(p_mutlcap_measure_data->acq_status & TOUCH_BURST_AGAIN))
    {
        /* 相互容量測定終了フラグ設定 */
        p_mutlcap_measure_data->measurement_done_touch = 1u;
        port_pin_toggle_output_level(PIN_PB00);
    }
}
```

設定	構成設定名	データ型	単位	値	最大	代表
感知部測定間隔	DEF_TOUCH_MEASUREMENT_PERIOD_MS	uint16_t	ms	GPIOピン交互切り替え手順を通して見つけれられるべきです。	65535	20

PTC割り込み優先レベル

SAMでの入れ子でベクタ型の割り込み制御器(NVIC)は4つの異なる優先レベルを持ちます。PTC変換終了ISRの優先レベルは処理時間が重要な操作に対する応用の必要条件に基づいて選ぶことができます。PTC割り込み優先レベルの最低設定は接触応用時間での影響を持つかもしれませんが、他のより高い優先割り込みによってかかる時間に依存します。

設定	構成設定名	データ型	単位	最小	最大	代表
PTC割り込み優先レベル	DEF_TOUCH_PTC_ISR_LVL	uint8_t	なし	0 (最高優先権)	3 (最低優先権)	3

スタック溢れを防ぐため、使用者応用で十分なスタックの大きさが設定されていることを保証してください。この構成設定はSAMデバイスにだけ適用できます。

touch_suspend_app_cb

接触ライブラリAPIが呼ばれる前に応用によって初期化されなければならない呼び戻し関数ポイントです。接触ライブラリは一時停止操作が応用によって続けられなければならない時にこの関数によって指示された関数を呼びます。

設定	構成設定名	データ型	返り値
一時停止呼び戻し	touch_suspend_app_cb	void(* volatile touch_suspend_app_cb) (void)	void

低電力感知部事象周期

CPUが活性からスタンバイ動作に戻ると、低電力感知器として構成設定された感知部はこの周期で走査されます。このパラメータに対する大きな値は電力消費を減らしますが、低電力感知部に対する応答時間を増します。

以下のマクロは低電力感知部事象周期を構成設定するのに使われます。

- LOWPOWER_PER0_SCAN_3_P_9_MSマクロは3.9msでの走査速度に設定します。
- LOWPOWER_PER1_SCAN_7_P_8_MSマクロは7.8msでの走査速度に設定します。
- LOWPOWER_PER2_SCAN_15_P_625_MSマクロは15.625msでの走査速度に設定します。
- LOWPOWER_PER3_SCAN_31_P_25_MSマクロは31.25msでの走査速度に設定します。
- LOWPOWER_PER4_SCAN_62_P_5_MSマクロは62.5msでの走査速度に設定します。
- LOWPOWER_PER5_SCAN_125_MSマクロは125msでの走査速度に設定します。
- LOWPOWER_PER6_SCAN_250_MSマクロは250msでの走査速度に設定します。
- LOWPOWER_PER7_SCAN_500_MSマクロは500msでの走査速度に設定します。

低電力感知部変動周期

このパラメータは低電力動作中の単一活性測定に対する走査周期を構成設定します。この活性測定は低電力感知部の参照基準経緯維持に必要とされます。

設定	構成設定名	データ型	単位	最小	最大	代表
低電力感知部変動速度	DEF_LOWPOWER_SENSOR_DRIFT_PERIODICITY_MS	uint16_t	ms	0	65535	2000

低電力感知部ID

DEF_LOWPOWER_SENSOR_IDマクロは感知器を低電力感知部として構成設定するのに使われます。1つの感知部だけが低電力感知部として構成設定することができます。低電力感知部は標準感知部または一括感知部で有り得ます。

5.2.10. 水分パラメータ

耐水性許可

DEF_XXXXCAP_MOIS_TOLERANCE_ENABLEマクロは水分検出機能の許可または禁止に使われます。

水分迅速再群発

DEF_XXXXCAP_MOIS_QUICK_REBURST_ENABLEマクロは与えられた水分群内で迅速再群発機能の許可または禁止に使われます。許可時、与えられた水分群内でどれかの感知部が接触されると、検出積分または跳ね返り抑制を解決するためにその感知部だけが繰り返し測定が行われます。禁止時、与えられた水分群内でどれかの感知部が接触されると、検出積分または跳ね返り抑制を解決するために水分群内の全ての感知部で繰り返し測定が行われます。最良の接触応答時間のためにこの機能を許可することが推奨されます。

水分群

DEF_XXXXCAP_NUM_MOIS_GROUPSマクロはシステムに存在する個別の水分群の総数を指定します。

5.2.11. PTC線接地機能

PTC GPIO状態

DEF_XXXXCAP_PTC_GPIO_STATEマクロはPTC測定周回間で測定しない自己容量/相互容量PTC線を接地/VCCに設定するのに使われます。PTC線のGND_WHEN_NOT_MEASUREDへの設定はピンが測定されない時に必ずピンの状態をLowに設定します。PTC線のPULLHIGH_WHEN_NOT_MEASUREDへの設定は測定周回で感知部測定間でPTC線を浮き状態にします。低電力を得るためにGND_WHEN_NOT_MEASUREDを設定することが推奨されます。

5.3.2. 複数接触群

使用者が特定要求を示すために耐水性群内で同時に複数の感知部の接触を望む場合、応用はそれらの感知部を単一の複数接触群に構成設定すべきです。複数の複数接触群はお客様応用によって形成することができます。ライブラリは単一の耐水性群内で最大8つの複数接触群を支援します。

耐水性機能は以下の筋書き下でシステムの性能を改善します。

- ・ 前面パネル面に吹きかけられた水滴
- ・ 前面パネル面に降り注がれた激しい水
- ・ 複数感知部での大きな水溜り
- ・ 複数感知部での滴る水

耐水性機能は以下の筋書き下でどんな重要な性能改善を提供することも意図されません。

- ・ 単一感知部での分離された大きな水溜り
- ・ 単一感知部での直接注水

同じ水分群内で、使用者は全ての感知部を単一の複数接触群に構成設定すべきではありません。

5.4. 感知部状態読み取り

雑音耐性と耐水性の機能が許可されると、感知部状態の有効性は水分状態と雑音状態に基づきます。感知部の雑音耐性と耐水性の情報については[雑音計数器測定と水分パラメータ](#)を参照してください。感知部の状態は感知部が雑音と水分によって影響を及ぼされない時にだけ有効です。感知部が雑音多または水分によって影響を及ぼされる場合、感知部の状態はOFFとして考慮されなければなりません。以下のコード断片は相互容量感知部についても同じです。

感知部がDI(検出積分)中に接触されて開放される場合、ライブラリはOFFまたはDISABLED以外の状態の感知部に対応するチャンネルで群発を行います。AKS群内の何れかの感知部がOFFまたはDISABLED以外の状態の場合、ライブラリはそのAKS群に属す感知部に対応するチャンネルで群発を行います。どれかの水分群内の感知部がOFFまたはDISABLED以外の状態の場合、ライブラリはその水分群に属す感知部に対応するチャンネルで群発を行います。

```
if(! (GET_MUTLCAP_SENSOR_NOISE_STATUS (SENSOR_NUMBER)))
{
    if(! (GET_MUTLCAP_SENSOR_MOIS_STATUS (SENSOR_NUMBER)))
    {
        /* 感知部状態は有効な読み込み感知部状態 */
    }
    else
    {
        /* 感知部は水分に影響を及ぼされています。 */
    }
}
else
{
    /* 感知部は雑音多 */
}
```

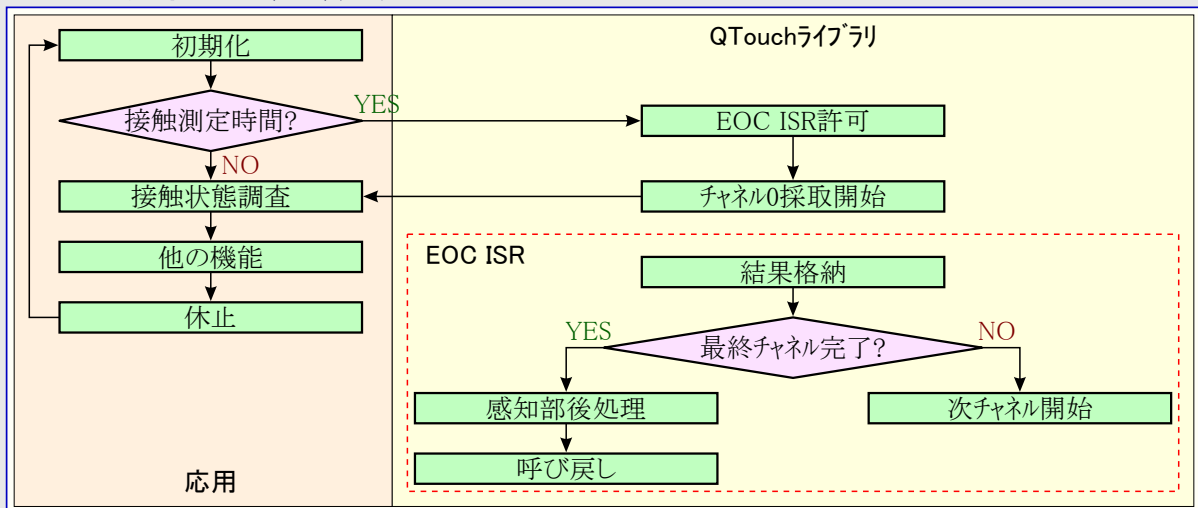
5.5. 応用の流れ

5.5.1. 応用の流れ – SAM

応用は相互容量または自己容量のどちらかの感知部で接触測定を周期的に始めます。各感知部測定の最後で、PTC単位部は変換終了(EOC:End Of Conversion)割り込みを生成します。接触測定は全ての感知部が測定されるまで順次実行されます。接触状態と回転子/摺動子の位置を決めるために測定した感知部データで追加の後処理が実行されます。測定の完了を示すために割り込み呼び戻し関数が起動されます。推奨される操作の流れはCPUを接触感知測定中に休止または他の関数を実行のどちらにも容易にします。

PTCを使う前に、PTC周辺機能用の標準クロック生成器が応用によって構成設定されるべきです。PTC標準クロックは4MHzに設定されることが推奨されます。

図5-12. 応用対QTouchライブラリの流れ

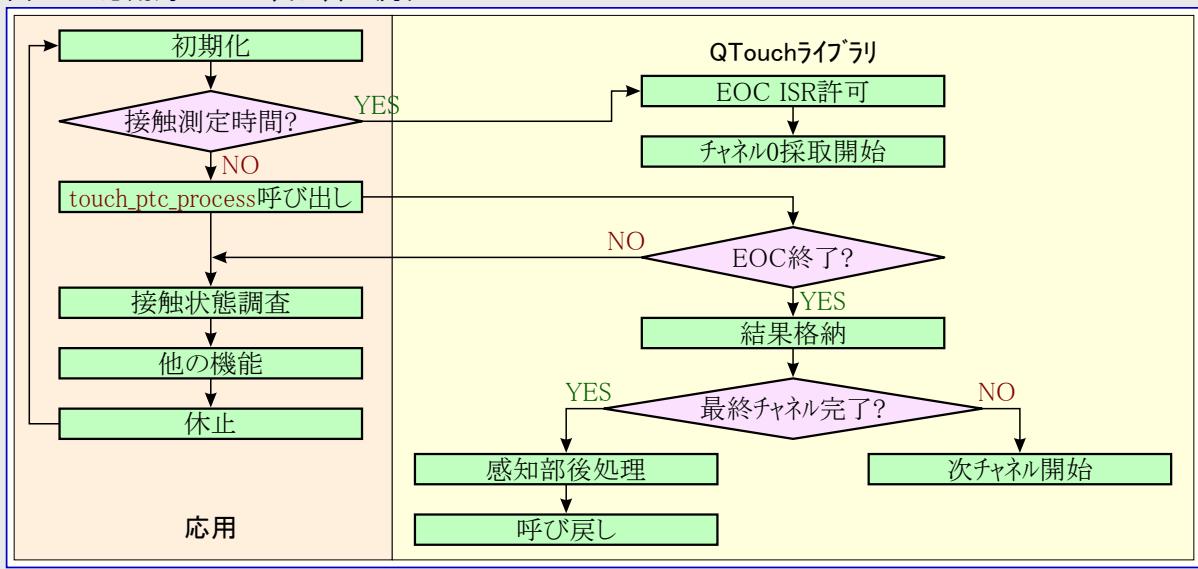


5.5.2. 応用の流れ - megaAVR

応用はポーリングまたは割り込みのどちらかで相互容量または自己容量のどちらかの感知部での接触測定を周期的に始めます。ポーリング動作に於いて、接触APIは閉塞APIで、より多くのCPU時間を消費します。ISR動作に於いて、接触APIは非閉塞で、各感知部測定の最後で変換終了(EOC:End Of Conversion)割り込みを生成します。接触測定は`touch_xxxcap_sensors_measure()` APIを呼ぶことによって最初の感知部で始められます。接触測定が順次始められ、割り込み環境に代わって応用環境で`touch_ptc_process()` APIを呼ぶことによって接触状態と回転子/摺動子の位置を決めるために測定した感知部データに於いて追加の後処理が実行されます。測定の完了を示すために割り込み呼び戻し関数が起動されます。ISR動作の操作の流れはCPUを接触感知測定中に休止または他の関数を実行のどちらも容易にします。

PTCクロックを4MHzに設定することが推奨されます。

図5-13. 応用対QTouchライブラリの流れ



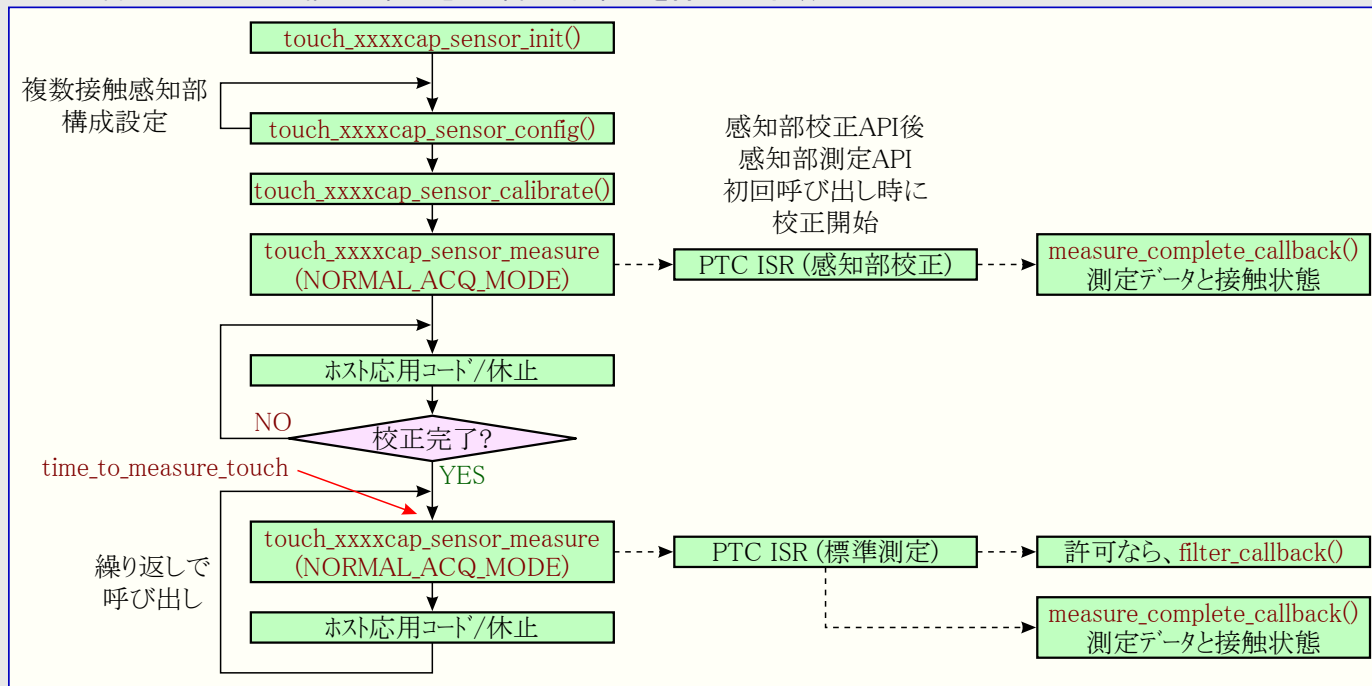
5.6. API手順

`touch_xx_sensors_init` APIはQTouchライブラリだけでなくPTC単位部も初期化します。これは相互と自己の容量法特定ピン、レジスタ、それと全体感知部構成設定も初期化します。

`touch_xx_sensor_config` APIは個別感知部を構成設定します。感知部特定構成設定パラメータはこのAPIに対する入力引数として提供されます。

`touch_xx_sensors_calibrate` APIは構成設定された全ての感知部を校正し、標準動作用に感知部の準備を整えます。`touch_xx_sensor_s_measure` APIは構成設定した全ての感知部で感知部測定を始めます。

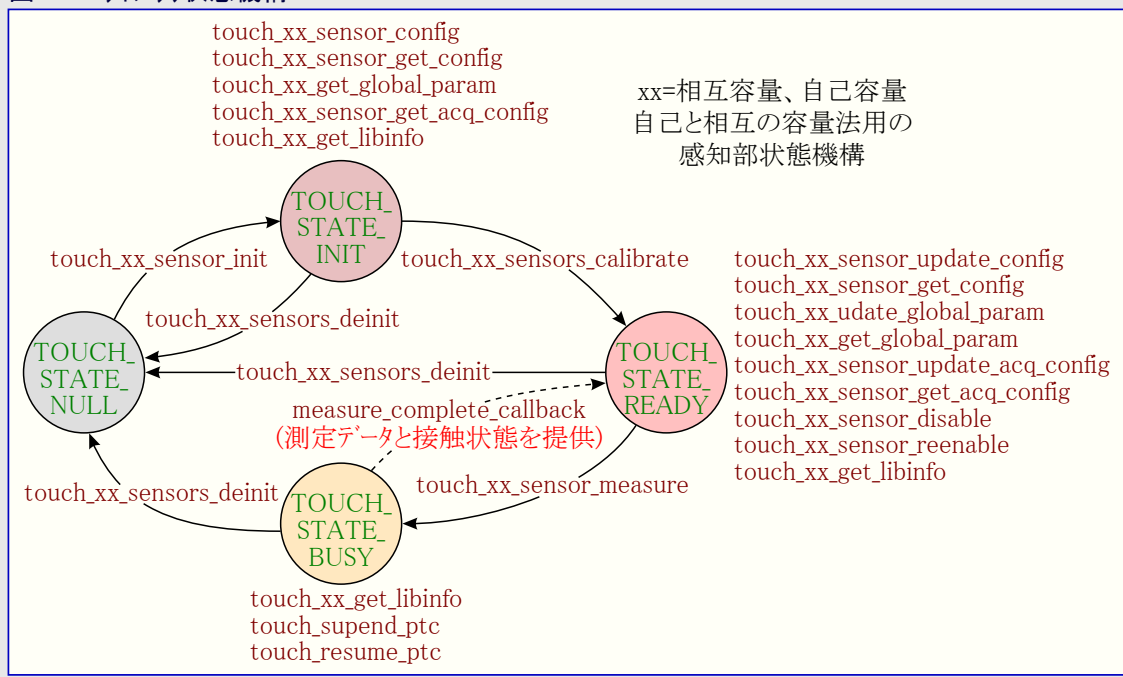
図5-14. 許可された自己と相互の容量感知部組み合わせを持つAPI手順



5.7. 狀態機構

様々なライブラリの状態と事象遷移を提供するPTC QTouchライブラリ状態機構は下図で見つけることができます。この状態機構は接触採取法の各々に対して独立して維持され、それは相互容量感知部操作の状態が自己容量の状態と異なることができ、それらの共存を許すことを意味します。

図5-15. ライブラリ状態機構



`touch_xx_sensors_init` APIはQTouchライブラリだけでなくPTC単位部も初期化します。これは相互と自己の容量法特定ピン、レジスタ、それと全体感知部構成設定も初期化します。

`touch_xx_sensor_config` APIは個別感知部を構成設定します。感知部特定構成設定パラメータはこのAPIに対する入力引数として提供されます。

`touch xx sensors calibrate` APIは構成設定された全ての感知部を校正し、標準動作用に感知部の準備を整えます。

`touch_xx_sensors_measure` APIは構成設定した全ての感知部で感知部測定を始めます。

`touch_xx_sensors_deinit`関数は初期化したライブラリ状態を解除するのに使われます。内部のライブラリ データと状態を解消するのに使われます。呼ばれると、ライブラリ状態を**TOUCH STATE NULL**に変更します。

touch_xx_lowpower_sensor_enable_event_measure APIは低電力監視部測定に基づく事象起動を開始するのに使われます。

接触ライブラリ一時停止再開操作

接触ライブラリはPTCの一時停止と再開するためにtouch_suspend_ptcとtouch_resume_ptcのAPIを提供します。

一時停止APIが呼ばれると、接触ライブラリは一時停止動作を始め、応用へ戻ります。現在のPTC変換完了後、接触ライブラリは一時停止動作を始めて応用接触一時停止呼び戻し関数ポインタを呼びます。一時停止完了呼び戻し関数ポインタは応用によって登録されなければなりません。

注: 登録されていない場合、一時停止呼び出しはTOUCH_INVALID_INPUT_PARAMを返します。

その後に応用は電力消費を減らすために対応するクロックを禁止すべきです。右の流れ図は一時停止手順を示します。

接触状態がTOUCH_STATE_BUSYなら、使用者はクロックを禁止して一時停止ルーチンを完了するのを続けることができます。

操作を再開するには、右の手順を実行してください。

SAMコントローラは事象システムを用いて自立的にPTC接触感知を操作するように構成設定されるかもしれませんが。この動作形態では、単一感知部チャネルが'低電力'キーとして支持され、接触検出のためにCPU活動なしで周期的に測定され得ます。CPUは操作を通してスタンバイ(STRANDBY)を保つことができ、電力消費を最小にします。

低電力キーは自己容量に対するY(感知)線または相互容量に対する1つのX(駆動)線と1つのY(感知)線を持つ個別の電極にでき、または記述されるような一括動作感知部として複数の駆動や感知の線の組み合わせにすることができます。

この方法では、例えば多いキー数の応用と同時に3.3Vで10μA未満を流す極端に低い電力水準でも高速応答を達成することができます。

図5-16. 一時停止手順

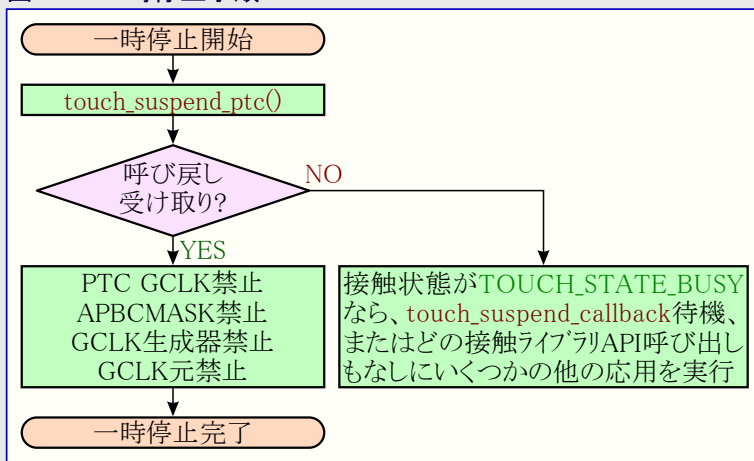
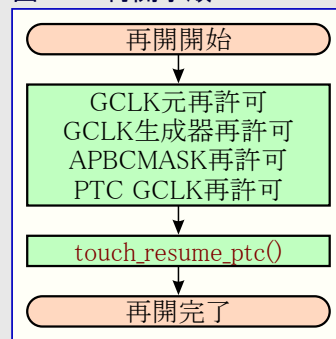


図5-17. 再開手順



5.8. 動作形態

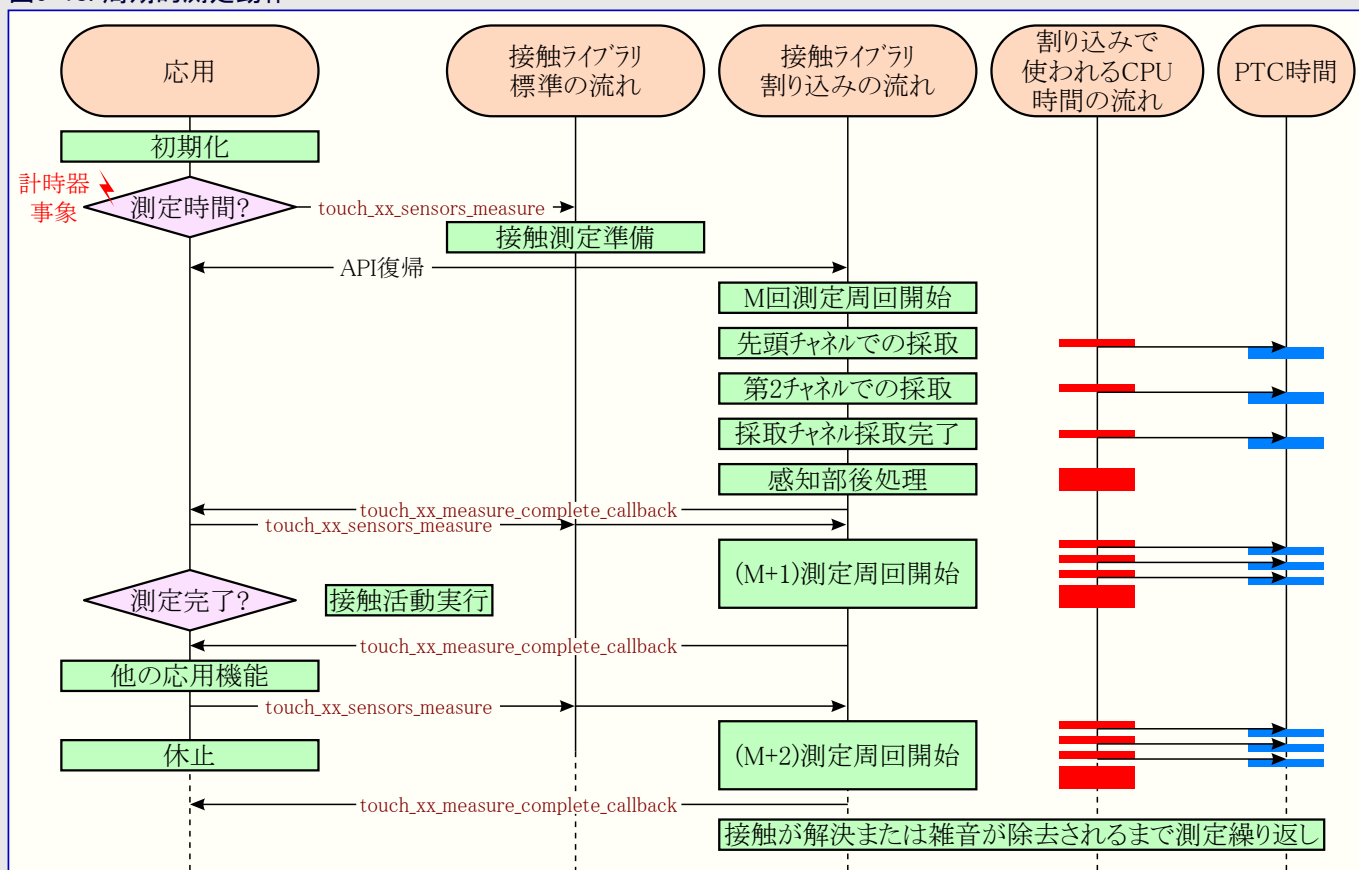
QTouchライブラリは以下の感知部測定動作で動くことができます。

- 周期的測定
- 連続的測定

5.8.1. 周期的測定

周期的測定動作では、計時器割り込みのような周期的な事象を通して応用によって感知部の測定が始められます。周期的測定動作の筋書きは感知部が全く接触されていない時です。長い測定周期はより低い電力消費を達成するのに使うことができる一方で、短い測定周期はより良い接触応用時間のために必要とされます。故に、測定周期は与えられた応用に適合するように調節されるべきです。代表的な測定周期は20～250msの範囲です。

図5-18. 周期の測定動作



5.8.2. 連続的測定

連続的測定動作では、接触ライフリから続けて感知部測定を始めることができます。この動作は以下の筋書き下で使用者の存在を解決、または校正を解決するために起動することができます。

- ・ 感知部が接触または開放される時の使用者の存在を解決
- ・ 以下の時に校正を解決
 - 感知部はtouch_xx_sensors_calibrate APIを用いて校正されます。
 - 感知部が接触再校正条件から離脱
 - 感知器が最大ON持続時間状態

連続測定動作を示すために測定データ構造体のTOUCH BURST AGAIN採取状態データビットが設定されます。

```
void touch_mutlcap_measure_complete_callback(void)
{
    if (!(p_mutlcap_measure_data->acq_status & TOUCH_BURST_AGAIN))
    {
        /* 相互容量測定終了フラグ設定 */
        p_mutlcap_measure_data->measurement_done_touch = 1u;
    }
}
```

接触ライブラリ採取状態フラグ

連続測定動作中の接触ライブラリ採取状態情報はtouch_measure_data_t接触測定データ構造体のtouch_acq_status_t acq_status要素を用いて利用可能です。

表5-1. 接触採取状態ビット領域

マクロ	ビット領域	注釈
TOUCH_NO_ACTIVITY	0x0000u	接触活動なし
TOUCH_IN_DETECT	0x0001u	最低1つの接触チャネルが検出
TOUCH_STATUS_CHANGE	0x0002u	最低1つの接触チャネルの接触状態が変更
TOUCH_ROTOR_SLIDER_POS_CHANGE	0x0004u	最低1つの回転子/摺動子の回転子/摺動子位置が変更
TOUCH_CHANNEL_REF_CHANGE	0x0008u	最低1つの接触チャネルの参照基準値が変更
TOUCH_BURST_AGAIN	0x0100u	濾波または校正状態を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_CAL	0x0200u	校正を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_FILTERIN	0x0400u	濾波を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_DI	0x0800u	検出積分を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_POS_RECAL	0x1000u	再校正を解決するのに再群発が必要なことを示します。
TOUCH_CC_CALIB_ERROR	0x2000u	最低1つのチャネルでCC校正失敗を示します。
TOUCH_AUTO_OS_IN_PROGRESS	0x4000u	安定なチャネル信号を得るための自動過採取(OS)進行中を示します。

採取状態フラグは示されるように測定完了呼び戻し内で監視されます。

```
void touch_mutlcap_measure_complete_callback(void)
{
    if ((p_mutlcap_measure_data->acq_status & TOUCH_BURST_AGAIN))
    {
        // 採取完了を示す。
    }
    if ((p_mutlcap_measure_data->acq_status & TOUCH_RESOLVE_CAL))
    {
        // 感知部校正進行中を示す。
    }
    if (!(p_mutlcap_measure_data->acq_status & TOUCH_BURST_AGAIN))
    {
        // 採取完了を示す。
        /* 相互容量測定終了フラグ設定 */
        p_mutlcap_measure_data->measurement_done_touch = 1u;
    }
}
```

連続的測定後処理動作

QTouchライブラリに対する感知部データ後処理動作はtouch.hファイルの一部として利用可能なDEF_XXXXCAP_TOUCH_POSTPROCESS_MODE構成設定項目を用いて選ぶことができます。

TOUCH_LIBRARY_DRIVEN動作が選ばれると、接触押下、開放、校正を解決するためにライブラリは繰り返し接触測定を自ら始めます。この動作は最良の応答時間に適します。

TOUCH_APPLN_DRIVEN動作が選ばれると、接触押下、開放、校正を解決するためにライブラリは繰り返し接触測定を始めません。この動作は厳しいCPU時間必要条件を必要とする応用に対する確定的なPTC割り込み実行時間に適合します。他の重要な応用コードが実行されつつあるために繰り返し接触測定が遅らされるので、この動作は潜在的に接触応答時間に影響を及ぼし得ます。

TOUCH_APPLN_DRIVEN動作での応答時間を改善するため、どんな遅延もなしに追加の測定に応えるように、感知部測定を始めるのに以下の条件が適用されるべきです。同じ条件はシステムが休止へ行き得るのに先立って完了されるべき保留採取を調べるために、休止のような他の応用の筋書きに対しても適用され得ます。

```
if ((touch_time.time_to_measure_touch == 1u) || (p_mutlcap_measure_data->acq_status & ¥
    TOUCH_BURST_AGAIN))
{
    /* 接触感知部測定処理を開始、または保留測定がある場合 */
    touch_ret = touch_mutlcap_sensors_measure(touch_time.current_time_ms, ure_complete_callback);
}
```

5.9. 接触ライブラリAPI異常

下表は接触ライブラリAPI異常符号情報を提供します。API異常符号型はtouch_ret_t列挙です。

異常符号列挙	注釈
TOUCH_SUCCESS	操作成功完了
TOUCH_ACQ_INCOMPLETE	接触ライブラリは前の保留接触測定で多忙
TOUCH_INVALID_INPUT_PARAM	無効な入力パラメータ
TOUCH_INVALID_LIB_STATE	現在の接触ライブラリの状態で許されない操作
TOUCH_INVALID_SELFCAP_CONFIG_PARAM	無効な自己容量構成設定入力パラメータ
TOUCH_INVALID_MUTLCAP_CONFIG_PARAM	無効な相互容量構成設定入力パラメータ
TOUCH_INVALID_RECAL_THRESHOLD	無効な再校正閾値入力値
TOUCH_INVALID_CHANNEL_NUM	構成設定したチャンネル総数を超えるチャンネル番号パラメータ
TOUCH_INVALID_SENSOR_TYPE	無効感知部型。感知部型はSENSOR_TYPE_UNASSIGENDにできません。
TOUCH_INVALID_SENSOR_ID	無効な感知部番号パラメータ
TOUCH_INVALID_RS_NUM	回転子/摺動子を構成設定しようとした時に回転子/摺動子数を0として設定

接触プロジェクトでの応用異常符号はDEF_TOUCH_APP_ERR_HANDLERマクロを用いて許可または禁止することができます。既定により、応用異常処理部を禁止するためにDEF_TOUCH_APP_ERR_HANDLERマクロの値は0に設定されます。応用異常処理部を許可するには、DEF_TOUCH_APP_ERR_HANDLERマクロを1として設定してください。許可されると、異常を捕らえるのにwhile(1)が用いられます。

更なる情報については[応用異常符号\(tag_touch_app_err_t\)](#)を参照してください。

6. 雑音性能用調整

PTCは容量性接触解決策を設計することを容易にし、一方同時に接触と性能の高品質を保つのに細心の注意を払って設計されています。それにもかかわらず、どの接触感知応用に於いても、システム設計者は目的対象環境での電氣的な妨害が感知部の性能にどう影響を及ぼし得るかを考慮しなければなりません。

不十分な調節の接触感知部は放射または伝導のどちらかの雑音試験で失敗を示すかもしれませんが、それは機器の環境や電力領域で起こるかもしれませんが、または標準動作中に機器自身によって生成されるかもしれません。多くの応用では明確に定義されるEMC性能基準が満たされなければならない品質規格があります。けれども、規格が最も一般的に起きる雑音の型と供給元を含むため、規格を満たすことはシステムが決してEMC問題を示さないことの証拠とは見做され得ません。

雑音耐性は接触応答時間と電力消費の増加の犠牲になります。システム設計者は最低電力消費を確実にするために接触感知部の正しい調節を行わなければなりません。PTC QTouchライブラリは接触応答時間、雑音耐性、電力消費間での最良の調和を得るために緒節することができる多数の使用者構成設定可能な機能を持ちます。

6.1. 雑音源

接触感知部性能に影響を及ぼす雑音源は次の様に分類することができます。

自己生成物

- ・ 電動機
- ・ 圧電ブザー
- ・ PWM制御の放射
- ・ 蛍光灯
- ・ 無線送信
- ・ 誘導性コンロの導電
- ・ 電源/充電器
- ・ 主電源

機器EMC規格

- ・ 伝導耐性 EN61000-4-6

6.2. 雑音計数器測定

雑音の影響は感知部で誘発または注入された雑音信号の振幅とその雑音信号の周波数分布図に大きく依存します。

一般的に、この雑音は以下として分類することができます。

- ・ 広帯域雑音
- ・ 狭帯域雑音

6.2.1. 広帯域雑音計数器測定

広帯域雑音は周波数成分がPTCの容量測定採取周波数に高調波的に関連しない雑音信号を呼びます。

雑音信号と組み合わせられた採取信号の最大と最小の電圧水準がPTCの入力範囲内で十分に多い採取数が取られれば、広帯域雑音妨害は濾波水準(DEF_MUTLCAP_FILTER_LEVEL_PER_NODEとDEF_SELFCAP_FILTER_LEVEL_PER_NODE)と自動過採取(DEF_MUTLCAP_AUTO_OS_PER_NODEとDEF_SELFCAP_AUTO_OS_PER_NODE)の設定を高い値に設定することによって平均化することができます。

雑音振幅が過大な場合、PTC部分は測定の飽和を経験します。この場合に雑音信号と組み合わせられた採取信号はPTCの入力範囲外で、測定の切り落とし(頭打ち)に帰着します。

それが測定試料の一部でだけ起こるため、この切り落としは解決された測定に於いて度々観測可能ではありませんが、切り落とされた試料の存在は採取点の効果的な平均化を妨げます。

この場合、試料の平均化は例えば大きな比率の過採取でも雑音なし測定の結果になりません。解決された信号は切り落とした信号の非対称性のため、その正しいレベルからの移動を示します。

構成設定パラメータ	設定
DEF_MUTLCAP_FILTER_LEVEL_PER_NODE, DEF_SELFCAP_FILTER_LEVEL_PER_NODE	FILTER_LEVEL_64
DEF_MUTLCAP_AUTO_OS_PER_NODE, DEF_SELFCAP_AUTO_OS_PER_NODE	AUTO_OS_DISABLE
DEF_MUTLCAP_FREQ_MODE, DEF_SELFCAP_FREQ_MODE	FREQ_MODE_NONE
DEF_MUTLCAP_CLK_PRESCALE_PER_NODE, DEF_SELFCAP_CLK_PRESCALE_PER_NODE	PRSC_DIV_SEL_1
DEF_MUTLCAP_SENSE_RESISTOR_PER_NODE, DEF_SELFCAP_SENSE_RESISTOR_PER_NODE	RSEL_VAL_100
touch_mutlcap_sensors_calibrate(), touch_selfcap_sensors_calibrate APIに対する自動調節入力	AUTO_TUNE_PRSC

段階1: 切り落とし防止

これは採取信号と組み合わせられた雑音の尺度を減らすためにハードウェア低域通過濾波器の実装が必要です。感知部容量はY(感知線)でPTC内部抵抗またはPCB上で外部的に実装のどちらかでの直列抵抗と組み合わせられます。外部直列抵抗はデバイスのY線から感知部間でデバイスピンの最も近くに実装されるべきです。

注: 切り落とし(頭打ち)を防ぐために自己容量応用に対して常に外部直列抵抗を使ってください。PTCの内部直列抵抗は100kΩに制限されます。雑音水準に依存して、外部直列抵抗は最大1MΩまで調べることができます。

段階2: 電荷移動検査

低域通過濾波器を形成するための直列抵抗追加の効果として、感知部充電用の時定数が増されます。各測定採取中に感知部容量が完全に充放電されるのを確実にすることが重要です。

不十分な充電は減少された接触差として観測され得たり、または感知電極への接続によってオシロスコープで見ることができます。

けれども、この問題は接触感知部操作に於いて明らかではないかもしれませんが、例えば低水準の雑音の存在でさえも応用は完全に上手に動くかもしれませんが、抵抗を除く構成と比べて抵抗の追加付きの雑音試験中に遥かに悪い性能を示します。

自動調節設定を通す電荷移動

QTouchライブラリ自動調節設定は許可された各キーで電荷移動試験を実行し、前置分周器を完全な電荷移動を保証する利用可能な最高速設定に設定します。

以下の設定の組み合わせが使われるべきです。

- DEF_MUTLCAP_SENSE_RESISTOR_PER_NODEとDEF_SELFCAP_SENSE_RESISTOR_PER_NODEはRSEL_VAL_100に設定されるべきです。
- 自動調節前置分周器(AUTO_TUNE_PRSC)はtouch_mutlcap_sensors_calibrate(AUTO_TUNE_PRSC)とtouch_selfcap_sensors_calibrate(AUTO_TUNE_PRSC)に対する入力パラメータとして提供されるべきです。

手動調節による電荷移動用試験

- AUTO_TUNE_NONE設定がtouch_mutlcap_sensors_calibrate(AUTO_TUNE_NONE)とtouch_selfcap_sensors_calibrate(AUTO_TUNE_NONE)の校正APIに対する入力として提供される場合、PTCはPTCクロック前置分周器(DEF_MUTLCAP_CLK_PRESCALE_PER_NODE, DEF_SELFCAP_CLK_PRESCALE_PER_NODE)と内部直列抵抗(DEF_MUTLCAP_SENSE_RESISTOR_PER_NODE, DEF_SELFCAP_SENSE_RESISTOR_PER_NODE)のユーザー定義設定を使います。
- 参照基準測定: 採取測定(信号値)は前置分周器を最大、即ち、PRSC_DIV_SEL_8に設定して取得されます。
- 検査測定: 第2測定(信号値)は減らされた前置分周器(PRSC_DIV_SEL_4)で取得されます。

2つの測定間の違いが最初の値の約3%(1/32)未満なら、結論は完全電荷移動がPRSC_DIV_SEL_4で達成されたことです。

この測定は完全電荷移動が達成される間で最高速PTC動作速度を見つけるためにPRSC_DIV_SEL_2とPRSC_DIV_SEL_1に対しても繰り返されます。

段階3：過採取調整

ハードウェア濾波と完全電荷移動によって一旦切り落とし(頭打ち)が防がれると、次の段階は濾波水準(DEF_MUTLCAP_FILTER_LEVEL_PER_NODE, DEF_SELFCAPI_FILTER_LEVEL_PER_NODE)と自動過採取(DEF_MUTLCAP_AUTO_OS_PER_NODE, DEF_SELFCAPI_AUTO_OS_PER_NODE)に対する最良設定を見つけることです。

自動過採取機能は重要な変化を示した感知部でだけ追加の採取が取られる利点を提供します。このような変化が存在しない場合、測定周回は毎回の測定で過採取速度として($\times \text{AUTO_OS}$)を適用するのに比べてもっと短くなり得ます。AUTO_OS使用時に頻繁に起き過ぎないことを確実にするよう注意が払われるべきです。

FILTER_LEVEL採取に対する採取時間は次のように表すことができます。

$$A + (B \times \text{FILTER_LEVEL})$$

ここでAはPTC構成設定と後処理に対する合計時間、Bは過採取期間(採取測定時間毎)です。

AUTO_OSが適用されると、この時間が次のように増されます。

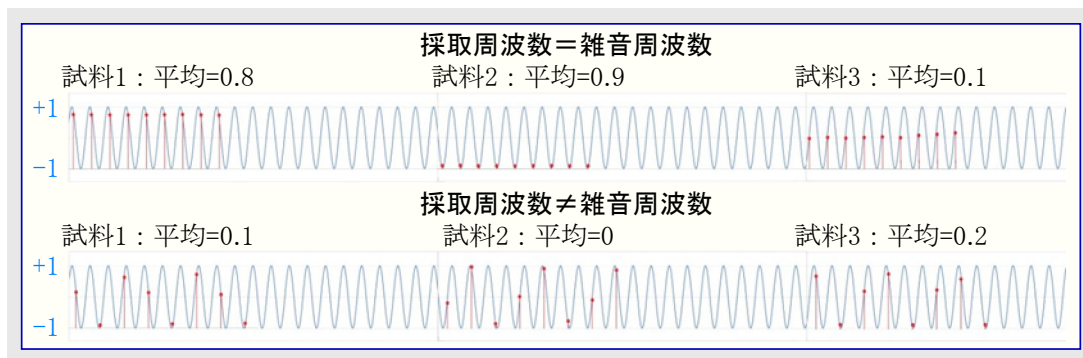
$$A + (B \times \text{FILTER_LEVEL} \times (1 + \text{AUTO_OS}))$$

FILTER_LEVELはAUTO_OSが最悪雑音状況中にだけ適用されるのを確実にするために充分大きくすべきです。

6.2.2. 狭帯域雑音計数器測定

雑音がPTC容量測定採取周波数に関連する周波数成分を含む場合、過採取の量は雑音の影響を平均化しません。同じ採取周波数を取られたどの測定採取群も測定変位に帰着します。各測定からの結果の実際の変位は雑音成分と採取周波数の相対的な位相に依存します。

この影響は下図で説明され、ここで雑音は正弦波によって表されます。



段階4：周波数動作形態選択

注: 前の項で提供された段階1、段階2、段階3は広帯域雑音と狭帯域雑音の両方を持つシステムでこの段階と組み合わせて使われるべきです。段階1の前で提供された既定設定は雑音調節開始に先立つ開始点として使われるべきです。

FREQUENCY_MODE_NONEでは、単一採取周波数が使われ、採取は与えられた前置分周器設定で可能な最速で取得されます。これは採取周波数に関連しない全ての雑音周波数で充分な過採取での素晴らしい雑音耐性で最善の応用時間を与えます。

けれども、雑音が採取周波数に関連する高/低調波(または近くの)周波数の場合、上で説明されたように、雑音問題が厳しくなります。

これはEN61000-4-6のような周波数掃引試験が必要とされる応用で特に重要です。

FREQUENCY_MODE_SPREADは採取周波数により広い分布を適用するために連続する採取間の期間が鋸歯形式で変調されるように、採取速度に変更を適用します。採取周波数(F0)は従って(F0/2~F0)範囲に渡って拡散されます。相対的に低い雑音振幅に於いて、これは応答時間での最低の犠牲で効果的な性能改善をし得ます。

FREQUENCY_MODE_HOPは影響を及ぼされた周波数で取られた測定の使用を避けるために3つの基準周波数と中央値濾波器を利用します。周波数は問題周波数帯内で交差高調波の組を最小にするように選ばれるべきです。

3つの周波数の各々は各測定周回に対して順番に使われます。例えば、

- ・ 周回1：周波数0で全感知部測定
- ・ 周回2：周波数1で全感知部測定
- ・ 周回3：周波数2で全感知部測定
- ・ 周回4：周波数0で全感知部測定
- ・ 周回5：周波数1で全感知部測定

周波数0が雑音周波数に関連する場合、F0を取る測定は高い変動を示すでしょう。中央値濾波器の使用で、これは中心を離れた測定が除外されることを保証します。

いくつかの応用で、自己生成雑音が既定飛び跳ね周波数の1つ以上に影響を及ぼす存在になるかもしれません。このような場合、この周波数を避けるために飛び跳ね周波数は変更されるべきです。

いくつかの雑音周波数は2つの飛び跳ね周波数の高/低調波近くで起き得、その場合にシステムは雑音を測定外に平均化するのに充分な採取を提供するために、より高いFILTER_LEVELやAUTO_OSの設定で調節されなければなりません。

PTC採取周波数決定

PTC採取周波数は次式によって与えられます。

PTC採取周波数 = (1/PTC採取時間)

PTC採取時間は次式によって与えられます。

PTC採取時間 = (採取当たりの周期数+飛び跳ね周波数)×PTC I/Oクロック周期

ここで、採取当たりの周期数=各採取に対して必要とされるPTCクロック周期です。これは固定値で15です。飛び跳ね周波数=PTC採取周波数遅延設定です。

このパラメータは`touch.h`ファイルで`DEF_MULTCAP_HOP_FREQS`と`DEF_SELFCAP_HOP_FREQS`のシンボルによって表されます。

PTC採取周波数はPTCへの標準クロック入力と前置分周器設定に依存します。この遅延設定は与えられた感知部での連続的な測定間にn個のPTCクロック周期を挿入し、それによってPTC採取周波数を変えます。

`FREQ_HOP_SEEL_1`設定は連続的な測定間に0個のPTCクロック周期を挿入します。`FREQ_HOP_SEL_16`設定は15個のPTCクロック周期を挿入します。

故に、より大きな遅延設定はより小さな遅延設定と比べて与えられた感知部での容量測定にかかる総時間を増します。採取周波数と同じ周波数付近の雑音を避けるような望む設定を使うことができます。

範囲：`FREQ_HOP_SEL_1`～`FREQ_HOP_SEL_16`

このパラメータに値を割り当てる時に3つの周波数飛び跳ね遅延設定を指定することが必要です。各PTCクロック周期の期間は次式によって与えられます。

ここで、

`CLKPTC` = PTCへの標準クロック入力

クロック構成設定については`touch.c`ファイルで`touch_configure_ptc_clock()` APIを参照してください。

前置分周器 = PTCクロック前置分周器設定

このパラメータは`touch.h`ファイルで`DEF_MUTLCAP_CLK_PRESCALE_PER_NODE`と`DEF_SELFCAP_CLK_PRESCALE_PER_NODE`のシンボルによって表されます。

例: PTCへの標準クロック入力=4MHzの場合、

- `PRSC_DIV_SEL_1`はPTCクロックを4MHzに設定します。
- `PRSC_DIV_SEL_2`はPTCクロックを2MHzに設定します。
- `PRSC_DIV_SEL_4`はPTCクロックを1MHzに設定します。
- `PRSC_DIV_SEL_8`はPTCクロックを500kHzに設定します。

例: `CLKPTC`=4MHz、前置分周器=`PRSC_DIV_SEL_1`の時に得られるPTC採取周波数は次のとおりです。

飛び跳ね周波数	PTC採取周波数 (kHz)	飛び跳ね周波数	PTC採取周波数 (kHz)
<code>FREQ_HOP_SEL_1</code>	66.67	<code>FREQ_HOP_SEL_9</code>	43.48
<code>FREQ_HOP_SEL_2</code>	62.50	<code>FREQ_HOP_SEL_10</code>	41.67
<code>FREQ_HOP_SEL_3</code>	58.82	<code>FREQ_HOP_SEL_11</code>	40.00
<code>FREQ_HOP_SEL_4</code>	55.56	<code>FREQ_HOP_SEL_12</code>	38.46
<code>FREQ_HOP_SEL_5</code>	52.63	<code>FREQ_HOP_SEL_13</code>	37.04
<code>FREQ_HOP_SEL_6</code>	50.00	<code>FREQ_HOP_SEL_14</code>	35.71
<code>FREQ_HOP_SEL_7</code>	47.62	<code>FREQ_HOP_SEL_15</code>	34.48
<code>FREQ_HOP_SEL_8</code>	45.45	<code>FREQ_HOP_SEL_16</code>	33.33

注: 採取周波数はクロック元の公差に基づいて変わるかもしれません。

7. 応用設計

7.1. 接触ライブラリと関連ファイル

下表はQTouchライブラリを使うのに必要とされる必須ファイルを規定します。既存のユーザー例応用にQTouch機能を追加するため、これらのファイルとコンパイラに基づく関連ライブラリがユーザープロジェクトに追加されるべきです。

表7-1. 接触ライブラリ ファイル

ファイル	説明
touch_api_ptc.h	ライブラリとインターフェースするのに使われるAPIとデータ構造体を含むQTouchライブラリAPIヘッダ ファイル
touch.h	QTouchライブラリ構成設定ヘッダ ファイル
touch.c	QTouchライブラリ初期化と感知部構成設定を実演するためのヘルパ ファイル
libsamxxx_qtouch.iar.aまたは libsamxxx_qtouch_gcc.a	自己容量と相互容量の両感知部を支援する IARまたはGCC用にコンパイルされたQTouchライブラリ

7.2. コードとデータのメモリの考慮

下表はAQTouchライブラリに必要とされる代表的なコードとデータのメモリを記録します。表で提供される代表的なメモリ必要条件は応用に於ける通常のAPI使用だけを考慮することで至ります。補助(ヘルパ)APIの使用は追加のコード メモリを消費します。

各測定法は信号、参照基準、感知部構成設定情報、接触状態を格納するために応用からの追加データ メモリを必要とします。このデータ メモリは'データ塊'配列として応用によって提供されます。このデータ塊の大きさは構成設定された感知部数に依存します。touch_api_ptc.h内のPRIV_xx_DATA_BLK_SIZEマクロはこのデータ メモリ塊の大きさを計算します。

表7-2. 相互容量法

デバイス系列	キーのみ		回転子または摺動子を持つキー	
	コード メモリ	データ メモリ	コード メモリ	データ メモリ
libsamd1x-qtouch-gcc.a	9602	845	11114	861
libsamd1x-qtouch-iar.a	9005	497	10377	513
libsamd2x-qtouch-gcc.a	9222	841	10734	857
libsamd2x-qtouch-iar.a	8881	497	10254	513
libsaml21-qtouch-gcc.a	9282	841	10794	857
libsaml21-qtouch-iar.a	9744	497	11115	513
libsamda1-qtouch-gcc.a	9222	841	10734	857
libsamda1-qtouch-iar.a	8881	497	10254	513
libsamc2x-qtouch-gcc.a	9752	841	11264	857
libsamc2x-qtouch-iar.a	9209	501	10567	517
libsamr21-qtouch-gcc.a	9246	841	10758	857
libsamr21-qtouch-iar.a	8905	497	10277	513
libsaml22-qtouch-gcc.a	9886	841	11078	857
libsaml22-qtouch-iar.a	9509	501	10981	517
libatmega328pb_qtouch_gcc.a	13338	503	15760	532
libMega328PB_qtouch.r90	10761	578	12391	607
libatmega324pb_qtouch_gcc.a	14082	482	16520	631
libMega324PB_qtouch_iar.r90	10646	441	12379	562

ATmega328PBの場合、雑音、水分、自動調節、Qデバッグの機能なしの単一接触チャネル(相互容量動作)に対してRAM使用量は503バイトです。RAM使用量は各追加チャネルに対して36バイトずつ増やされます。

表7-3. 自己容量法

デバイス系列	キーのみ		回転子または摺動子を持つキー	
	コード メモリ	データ メモリ	コード メモリ	データ メモリ
libsamd1x-qtouch-gcc.a	9576	841	10884	849
libsamd1x-qtouch-iar.a	8952	497	10216	505
libsamd2x-qtouch-gcc.a	9098	845	10506	845
libsamd2x-qtouch-iar.a	8841	497	10101	505
libsaml21-qtouch-gcc.a	9258	841	10566	845
libsaml21-qtouch-iar.a	9806	497	11070	505
libsamda1-qtouch-gcc.a	9198	845	10506	845
libsamda1-qtouch-iar.a	8841	497	10101	505
libsamc2x-qtouch-gcc.a	9716	841	11024	845
libsamc2x-qtouch-iar.a	9148	501	10400	509
libsamr21-qtouch-gcc.a	9542	841	10850	845
libsamr21-qtouch-iar.a	8851	497	10115	505
libsaml22-qtouch-gcc.a	9530	841	11158	845
libsaml22-qtouch-iar.a	9410	501	10733	509
libatmega328pb_qtouch_gcc.a	13274	500	15260	519
libMega328PB_qtouch.r90	10705	594	12041	613
libatmega324pb_qtouch_gcc.a	14026	478	16024	593
libMega324PB_qtouch_iar.r90	10596	437	12329	558

ATmega328PBの場合、雑音、水分、自動調節、Qデバッグの機能なしの単一接触チャネル(自己容量動作)に対してRAM使用量は500バイトです。RAM使用量は各追加チャネルに対して32バイトずつ増やされます。

注: 1. 指定デバイス変種によって支援される感知部の総数はコード、データ、スタックのメモリ必要条件だけでなくXY線数によっても制限されます。

2. コードとデータに使われるメモリを節約するため、GCC例プロジェクトに対して新しいLib-nano Cライブラリが使われています。

8. 例応用

8.1. Atmel基板例プロジェクト

GCC Xplained Pro例プロジェクトはAtmel StudioでFile⇒New Exxample Projectメニュー任意選択を通してアクセスすることができます。

IAR Xplained Pro例プロジェクトはAtmel QTouchライブラリPTC部パックを通してアクセスすることができます。

Xplained Proキットに対して以下の例プロジェクトが利用可能です。

- SAM D20 Xplained ProとQT1 Xplained Pro相互容量例応用
- SAM D20 Xplained ProとQT1 Xplained Pro自己容量例応用
- SAM D21 Xplained ProとQT1 Xplained Pro相互容量例応用
- SAM D21 Xplained ProとQT1 Xplained Pro自己容量例応用
- 一括低電力構成設定を持つSAM D20 Xplained ProとQT1 Xplained Pro相互容量例応用
- 一括低電力構成設定を持つSAM D20 Xplained ProとQT1 Xplained Pro自己容量例応用
- SAM D11 Xplained Pro自己容量例応用
- SAM D10 Xplained Pro自己容量例応用
- SAM D20 QTouch頑強実演水分例応用(自己+相互)
- SAM C20 QTouch頑強実演水分例応用
- 一括低電力構成設定を持つSAM D20 Xplained ProとQT3 Xplained Pro相互容量例応用
- 一括低電力構成設定を持つSAM L21 Xplained ProとQT3 Xplained Pro相互容量例応用
- SAM DA1 Xplained ProとQT4 Xplained Pro自己容量例応用
- SAM C21 Xplained ProとQT1 Xplained Pro相互容量例応用
- SAM C21 Xplained ProとQT1 Xplained Pro自己容量例応用
- SAM C21 Xplained Pro自己容量例応用(基板上感知器)
- SAM C21 Xplained ProとQT5 Xplained Pro相互容量例応用
- SAM L22 Xplained Proと接触セグメントLCD Xplained Pro相互容量例応用
- ATmega328PB Xplained Mini自己容量例応用
- ATmega324PB Xplained ProとQT5 Xplained Pro相互容量例応用

注: SAM L22についてはPTCクロックを8MHzに設定することが推奨されます。

プロジェクトでのクロック構成設定変更

- SAM C20/C21改訂Bデバイスについては、主クロックとしてDFLLが使われ、DFLLクロック元に対する参照基準クロックとしてOSC32Kが使われます。SAM C20/C21改訂Cデバイスについては、主クロックとしてOSC48MHzが使われます。これはSAM C20/C21改訂BとSAM C20/C21改訂Cの両デバイスに対して同じプロジェクトを使うことによって例プロジェクトで実演されます。
- 主クロックとしてDFLLを持つ例プロジェクトは参照基準クロックとして分周されたOSC8MHz/OSC16MHzクロックを使います。
- SAM L22例プロジェクトは16MHz用にDFLL(性能水準-PL2)に構成設定し、それを主クロックとして利用します。このクロック設定は高い性能を提供します。
- SAM L22低電力使用者基板プロジェクトは8MHz用にDFLL(性能水準-PL0)に構成設定し、それを主クロックとして利用します。このクロック設定は低い電力消費を提供します。
- SAM L21/L22低電力例プロジェクトはスタンバイ休止動作で主調整器として降圧調整器を持つPL0(低電力指向動作)で構成設定されます。これは低電力なものを達成するのに最適な構成設定です。

例プロジェクトは接触を披露するためにXplained Pro基板と拡張キットを使います。これらの拡張キットは以下の項で説明されます。

QT1 Xplained Proキット

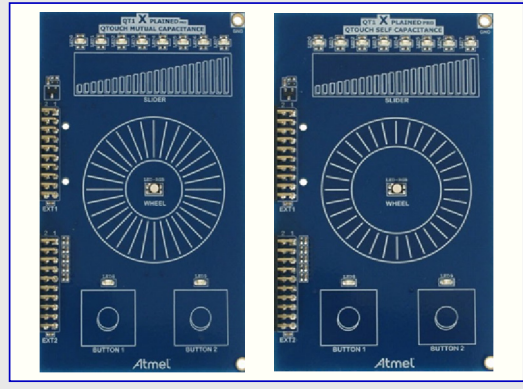
QT1 Xplained Pro自己容量と相互容量の拡張基板はSAM D20、SAM D21、SAM DA1、SAM C21、SAM L22 Xplained Pro評価キットによって支援されます。

注: SAM C21 Xplained Proは3.3Vと5.5VのVCCで動作することができ、一方QT1 Xplained Proは最大電圧3.6Vで動作することができます。SAM C21 Xplained Pro上のVCC選択ヘッダを3.3V位置に置くことを確実にしてください。

QT1 Xplained Pro基板は以下の鉤、摺動子、輪の組み合わせを実演します。

- 2つの鉤+2つの黄色LED
- 1つの摺動子+8つの黄色LED
- 1つの輪+1つのRGB LED

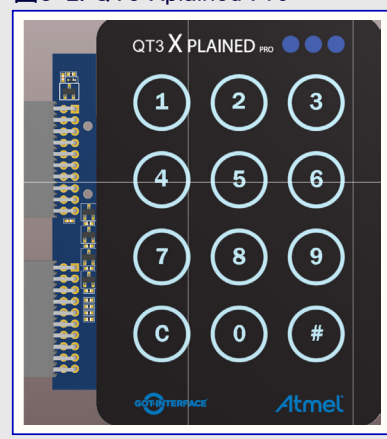
図8-1. QT1 Xplained Pro相互容量と自己容量



QT3 Xplained Proキット

QT3 Xplained Pro拡張基板はそれ上に12個の相互容量釦を持ち、SAM D20、SAM D21、SAM DA1、SAM L21、SAM L22、SAM C21 Xplained Pro評価キットによって支援されます。

図8-2. QT3 Xplained Pro

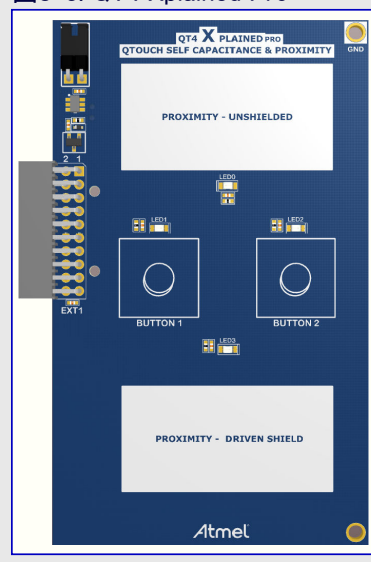


QT4 Xplained Proキット

QT4 Xplained Pro基板は以下のお膳立てを実演します。

- 2つの自己容量釦
- 1つの遮蔽されていない近接感知器
- 外部演算増幅器駆動部で駆動された遮蔽を持つ1つの近接感知器
- 各自己容量釦に対する1つのLED表示器
- 各近接感知器に対する1つのLED表示器

図8-3. QT4 Xplained Pro

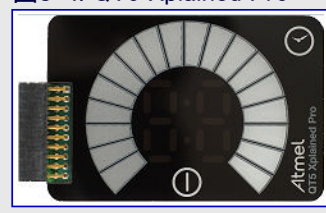


QT5 Xplained Proキット

QT5 Xplained Pro基板は以下のお膳立てを実演します。

- 1つの4チャンネル(4つのXと1つのY)相互容量曲線摺動子
- 2つの相互容量釦
- コロン(:)で分離された2つの7セグ桁として配置された16個のLED
- ISSIからのIS31FL3728 I²C LED配列制御器

図8-4. QT5 Xplained Pro



8.2. 使用者基板例プロジェクト

使用者基板の必要条件によって定義される感知部とピン構成設定に基づいてGCCプロジェクトを作成するのにAtmel Studio QTouch構成器(Composer)を使うことができます。生成された例プロジェクトはQTouch分析器(Analyzer)へ流すQDebugデータも許します。

使用者基板例プロジェクトはAtmel Studioで以下のメニュー任意選択を用いてQTouch構成器にアクセスすることによって生成することができます。

File⇒New Project⇒GCC C QTouch Execuable Project⇒Create QTouch Library Project

画面例で示されるようにQTouchプロジェクト構築部ウィザードが現れます。使用者の好みに従って感知部、デバイス、ピン、デバッグインターフェース、パラメータの調節の選択を行うことができ、プロジェクトを生成することができます。図は使用者基板の1つで生成されたプロジェクトを示します。

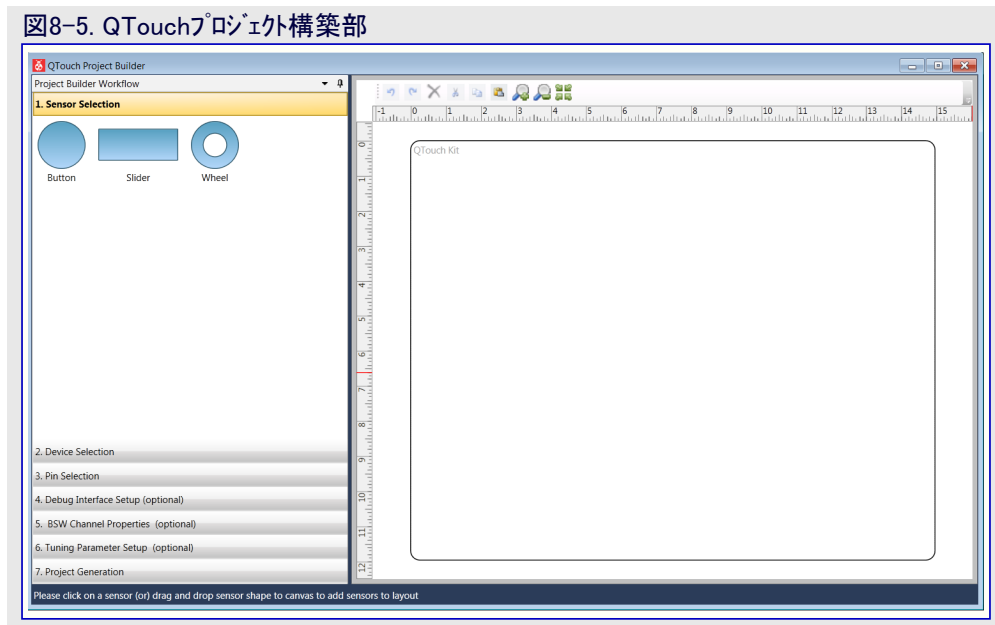
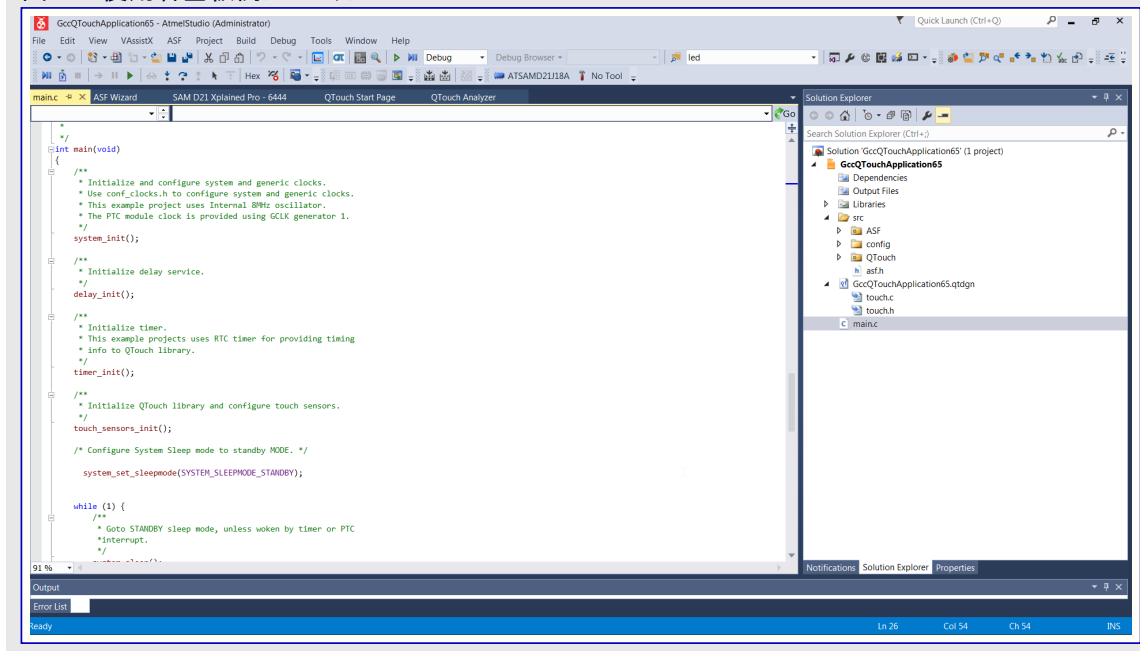


図8-6. 使用者基板例プロジェクト



8.3. 例プロジェクトでのAtmelソフトウェア枠組み(ASF)使用

例プロジェクトはAtmelソフトウェア枠組み(ASF:Atmel Software Framework)に基づきます。ASFのより多くの情報については<http://www.atmel.com/>で「Atmelソフトウェア枠組み使用者の手引き」を参照してください。

Atmel®ソフトウェア枠組み(ASF:Atmel Software Framework)はAtmelのフラッシュMCUのmegaAVR、AVR XMEGA、AVR UC3、SAMのデバイス用の組み込みソフトウェアの大きな集合を提供するMCUソフトウェア ライブラリです。

- ハードウェアと高い価値のミドルウェア(中間に位置するソフトウェア)に対する抽象化を提供してマイクロ コントローラの使用を容易にします。
- ASFは評価、試作、設計、製品段階に対して使われるように設計されています。
- ASFは画像使用者インターフェースを持つAtmel Studioで統合されるか、またはGCC、IARのコンパイラ用の独立型として利用可能です。
- ASFは無料でダウンロードすることができます。

8.4. 使用者基板をプログラミングするのにXplained Proキットの使用

SAM D20 Xplained Proは外部目的対象をプログラミングしてデバッグするためのCortex®デバッグ コネクタ(10ピン)が特徴です。このコネクタはSWDインターフェースに限定され、使用者によって開発された最終製品でSAM D20デバイスの実装プログラミングとデバッグを意図されます。より多くの情報については「SAM D20 Xplained Pro使用者の手引き」(www.atmel.com)を参照してください。

8.5. QDebug接触データ デバッグ通信インターフェースの使用

IARとGCCの例プロジェクト使用時、QDebug接触データ デバッグ インターフェースを許可することができます。これは接触デバイスとQTouch分析器(Analyzer)間の通信を許します。

QDebugを許可または禁止するには、**touch.h**ファイルで**DEF_TOUCH_QDEBUG_ENABLE**を構成設定してください。

QDebugが許可されて接触デバッグ データがQTouch分析器で更新されつつある時に、応答時間での遅延を増すデバッグ通信データ転送のために接触応答時間がより遅くなります。

QTouch分析器を用いた接触感知部調節後、最適化した接触性能のためにQDebugを禁止してください。

図8-7. QDebug用Atmel DGIインターフェース

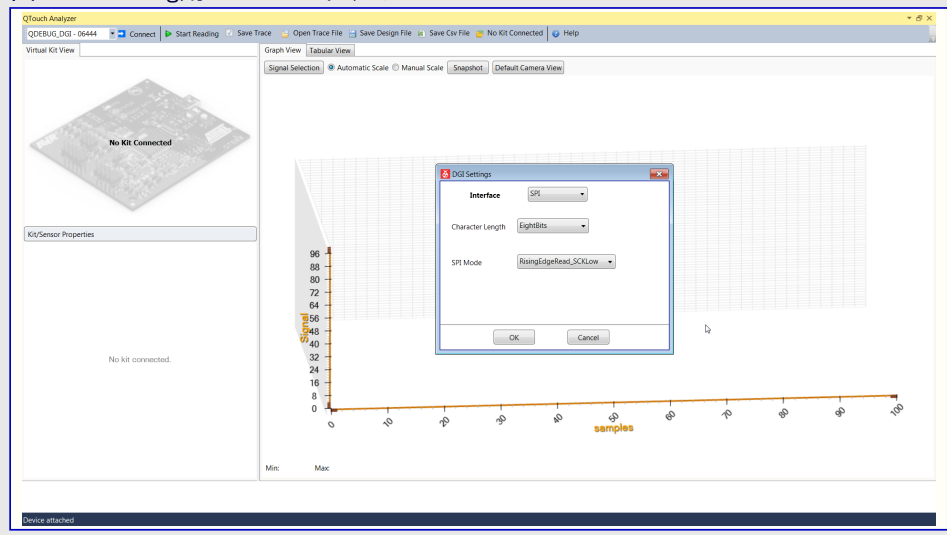
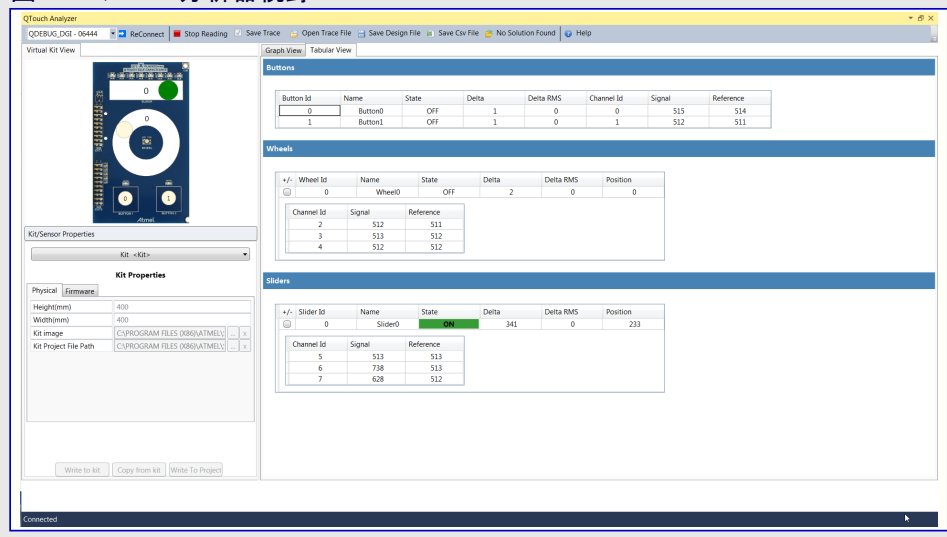
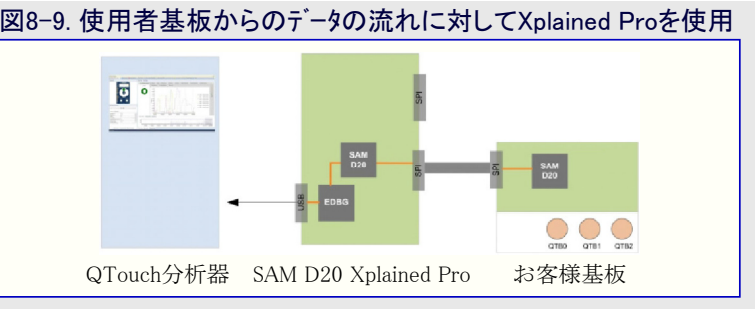


図8-8. QTouch分析器視野



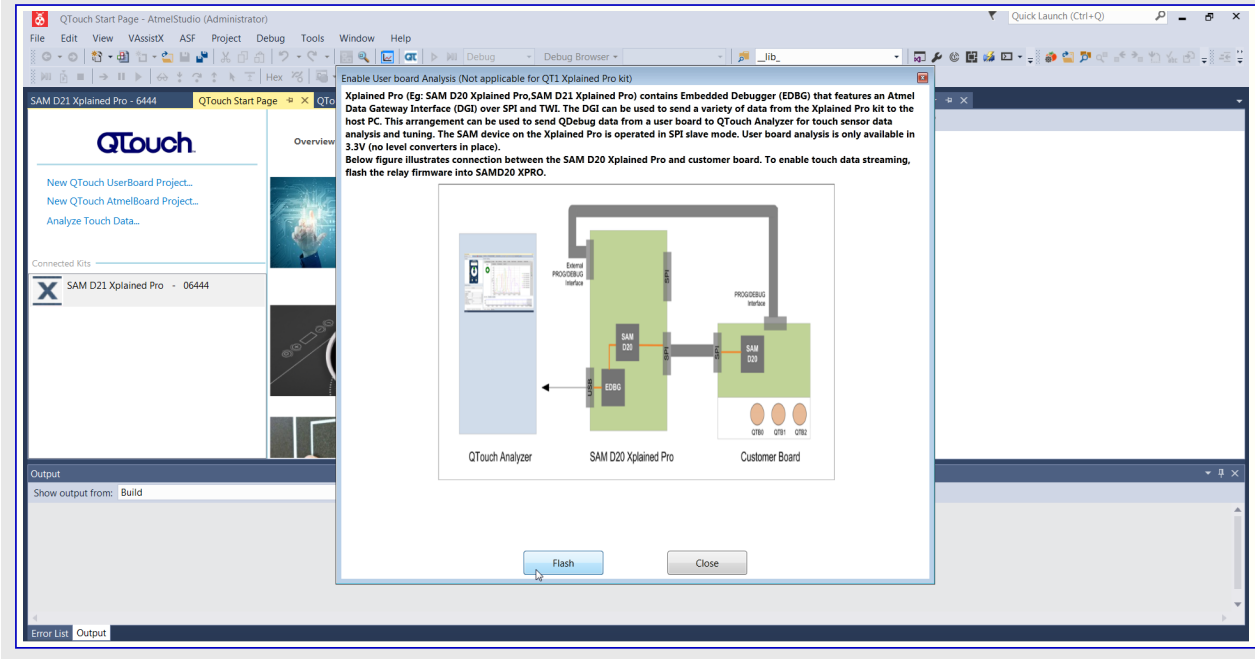
8.6. 使用者基板からQDebugデータの流にXplained Proキットの使用

SAM D20 Xplained ProはSPIとTWI上のAtmelデータ交換機インターフェース(DGI:Data Gateway Interface)を特徴とする組み込みデバッグ(EDBG)を含みます。DGIはXplained ProキットからホストPCへ色々なデータを送るのに使うことができます。この取り決めは接触感知部データを分析して調節するために使用者基板からAtmel Studio QTouch分析器(Analyzer)にQDebugデータを送るのに使うことができます。



QTouch構成器(Composer)を用いて生成された例プロジェクトはデータ転送用にSPIを使います。使用者基板からQDebugデータを流すには、取次ファームウェアがXplained Proキット上のSAM D20/D21マイクロ コントローラに書かれるべきです。SAM D20/D21 Xplained ProをPCに接続後、QTouch開始ページの接続キットでデバイス名が現れます。デバイス名を右クリックして取次ファームウェアを書くためにEnable User Board Analysisを選んでください。

図8-10. 取次ファームウェア書き込み



下表はSAM D20 Xplained Proキットと使用者基板間のSPI接続を示します。

表8-1. SPI接続情報		
SAM D20 Xplained Pro拡張ヘッダEXT3		使用者基板ピン
EXT3でのピン番号	機能	
16	SPI_MOSI (PB22)	MOSI
17	SPI_MISO (PB16)	MISO
18	SPI_SCK (PB23)	SCK
—	—	SS : GNDへ接続
19	GND	GND

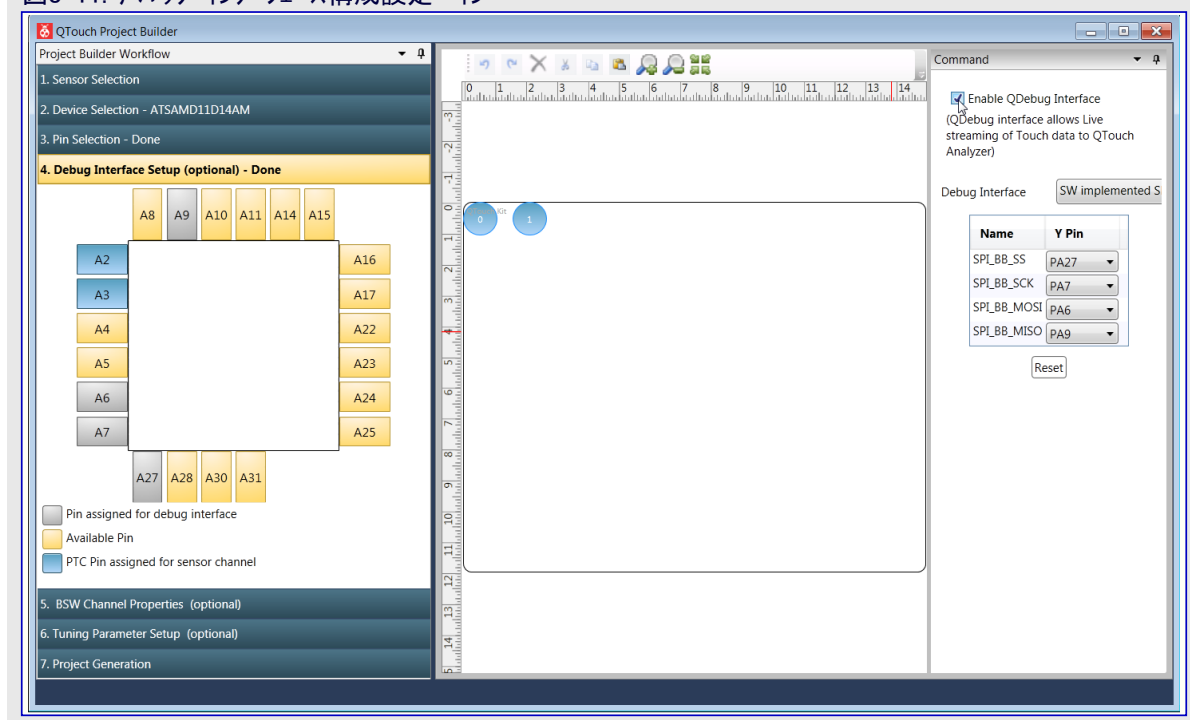
8.7. 使用者基板からQDebugデータの流にAtmel ICEの使用

Atmel ICEは使用者基板からのデータの流に使うことができます。
下表を参照してAtmel ICEのAVRヘッダから使用者基板へミニ バラ線ケーブルを接続してください。

QTouch構成器(Composer)のプロジェクト構築部(Builder)ウィザードを用いてプロジェクトを作成すると同時に、図で示されるようにデバッグ インターフェース構成設定ペインからSCK,MISO,SS,MOSIのピンを選ぶことができます。

Atmel ICE AVRポートピン	目的対象ピン	ミニ バラ線ピン
1 (TCK)	SCK	1
2 (GND)	GND	2
3 (TDO)	MISO	3
4 (VTG)	VCC	4
5 (TMS)	SS	5
6 (nSRST)	—	6
7 (未接続)	—	7
8 (nTRST)	—	8
9 (TDI)	MOSI	9
10 (GND)	—	0

図8-11. デバッグ インターフェース構成設定ペイン



接続が正しく行われてプロジェクトでデバッグ インターフェース構成設定も行われた時に、プロジェクトを使用者基板に書き込んでください。データが流されてQTouch分析器(Analyzer)経由で可視化することができます。

注: Atmel ICEはQDEBUG_DGIとしてQTouch Analyzerを一覧にするでしょう。

9. 既知の問題

1. 自己容量動作でのPTC

以下の障害はSAM D20(改訂B)に適用されます。

説明:

2つの最下位側利得設定が選択不可で、QTouchライブラリによるこれらの許可設定の試みは感知部に対して最適よりも高い感度に帰着するかもしれません。PTCは全ての接触を検出しないでしょう。この障害は規定のように動作する相互容量動作に影響を及ぼしません。

修正/対策:

自己容量接触感知に対してはSAM D20改訂Cまたはそれ以降を使ってください。

2. 接触採取が失敗して動作停止するかもしれません。

以下の障害は5.0.7版までのQTouchライブラリに適用されます。

説明:

QTouch応用に於いて、単一割り込みまたは入れ子にされた非PTC割り込みの連鎖のどちらかが総接触測定時間よりも長い持続時間を持つ場合、接触採取は失敗して動作を停止するかもしれません。この問題は少しの接触チャネル(2,3チャネル)と低水準の雑音処理(濾波水準16またはそれ以下と周波数飛び跳ねなし)を持つ応用に於いて最も高い可能性で発生します。

単一接触チャネルと濾波水準16を持つ応用では、総測定時間が約350 μ sです。2つの接触チャネルに対して総測定時間は2倍、3つの接触チャネルに対して3倍です。10個の接触チャネルで最大10倍または3.5msまで増加します。

修正/対策:

- 推奨される対策:
 - QTouchライブラリ5.0.8版またはそれ以降を使ってください。
- 他の代替策:
 1. 単一割り込みまたは入れ子にされた非PTC割り込みの連鎖の持続時間が総測定時間を超えないことを常に確実にしてください。

または、

2. 以下のコード断片で示されるように`touch_XXXXcap_sensors_measure()`関数に対して割り込みを禁止することによって重要な領域を追加してください。

```
Disable_global_interrupt();  
touch_ret = touch_XXXXcap_sensors_measure( touch_time.current_time_ms, NORMAL_ACQ_MODE, &  
touch_XXXXcap_measure_complete_callback);  
Enable_global_interrupt();
```

様々なCPU周波数に対して`touch_XXXXcap_sensors_measure` APIを実行する間の割り込み防止時間は次のとおりです。

割り込み防止時間はPTC_GCLK周波数、CPU周波数、ライブラリ版に基づいて変化します。実際の防止時間は`touch_XXXXcap_sensors_measure`関数の前後でGPIOピンを交互切り替えることによって測定することができます。

IARコンパイラが使われる時は、全体割り込みを各々、許可と禁止をするのに`system_interrupt_enable_global()`と`system_interrupt_disable_global()`の関数を使ってください。

10. PTC QTouchでの良くある質問

表10-1. 良くある質問

質問	答え
採取、感知部構成設定、感知部全体パラメータを変更できる時は？ 採取パラメータ変更後、感知部とPTCを再校正または再初期化が必要か？	(<code>touch_measure_data_t</code> 構造体の一部の) <code>measurement_done_touch</code> フラグが真の時にこれらのパラメータを更新するのにヘルパAPIを呼ぶことが最良で、これはライブラリが(以前開始した)未完了の採取の最中でないことを意味します。利得と濾波器の設定変更は信号値に影響を及ぼし得て、故に <code>touch_sensors_calibrate()</code> APIを呼び出すことによる再校正が必須です。
感知器は走行時に禁止して再許可することができますか？ 例えば、休止中に2つの感知器を走査し、その後にシステムが起きた時に全ての感知器を走査します。	はい、これは <code>touch_XXXXcap_sensor_disable()</code> と <code>touch_XXXXcap_sensor_reenable()</code> のAPIを使って可能です。
16採取試料の前に低振幅パルスが、16採取試料の後に大振幅のパルスがあります。	これらのパルスは感知線の初期状態を準備する部分です。
検出積分はPTC内またはQTouchライブラリによってのどちらで計算されますか？	これはQTouchライブラリによって行われます。
自動過採取が許可される時に接触タイミングはどう決めることができますか？	絶対最大周回は自動過採取が全チャネルに適用される場合で、 $(\text{標準採取時間}) \times (1 + \text{auto_os})$ です。 雑音が多い状況下で接触されたキーを除いて <code>AUTO_OS</code> が起きることを防ぐために <code>FILTER_LEVEL</code> が充分であるべきなので、これは不完全に調節されたシステムでだけ起き得ます。
感知部採取に使用されない時にLEDなどを駆動するために感知部信号線(YまたはX線)を使用することができますか？	いいえ。これは推奨されません。

11. 追補

11.1. マクロ

11.1.1. 接触ライブラリ採取状態ビット領域

キーワード	型	説明
TOUCH_NO_ACTIVITY	0x0000u	接触活動なし
TOUCH_IN_DETECT	0x0001u	最低1つの接触チャネルが検出
TOUCH_STATUS_CHANGE	0x0002u	最低1つの接触チャネルの接触状態が変更
TOUCH_ROTOR_SLIDER_POS_CHANGE	0x0004u	最低1つの回転子/摺動子の回転子/摺動子位置が変更
TOUCH_CHANNEL_REF_CHANGE	0x0008u	最低1つの接触チャネルの参照基準値が変更
TOUCH_BURST_AGAIN	0x0100u	濾波または校正状態を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_CAL	0x0200u	校正を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_FILTERIN	0x0400u	濾波を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_DI	0x0800u	検出積分を解決するのに再群発が必要なことを示します。
TOUCH_RESOLVE_POS_RECAL	0x1000u	再校正を解決するのに再群発が必要なことを示します。
TOUCH_CC_CALIB_ERROR	0x2000u	最低1つのチャネルでCC校正失敗を示します。
TOUCH_AUTO_OS_IN_PROGRESS	0x4000u	安定なチャネル信号を得るための自動OS進行中を示します。

11.1.2. 感知部状態構成設定

GET_SENSOR_STATE (SENSOR_NUMBER)

SENSOR_NUMBERを用いて指定した感知部に対応するパラメータに対する感知部状態(検出か否か)を得ます。

このマクロは0か1のどちらかを返します。ビット値が0なら、感知部は検出ではありません。ビット値が1なら、感知部は検出です。

```
#define GET_XXXXCAP_SENSOR_STATE (SENSOR_NUMBER) p_XXXXcap_measure_data->p_sensor_states ¥  
[(SENSOR_NUMBER / 8)] & (1 << (SENSOR_NUMBER % 8))) >> (SENSOR_NUMBER % 8)
```

GET_XXXXCAP_SENSOR_MOIS_STATUS (SNSR_NUM)

特定感知部の水分状態を得ます。戻り値は感知部が水分によって影響を及ぼされる場合に1で、感知部が水分によって影響を及ぼされない場合に0が返ります。

```
#define GET_XXXXCAP_SENSOR_MOIS_STATUS (SNSR_NUM) ((p_XXXXcap_measure_data->p_sensor_mois_status ¥  
[(SNSR_NUM) / 8] & (1 << ((SNSR_NUM) % 8))) >> (SNSR_NUM % 8))
```

GET_XXXXCAP_MOIS_GRP_SUM_DELTA (GRP_ID)

XXXXcap水分群合計差を得ます。戻り値は水分群の合計差を示す32ビット整数です。

```
#define GET_XXXXCAP_MOIS_GRP_SUM_DELTA (GRP_ID) (mois_XXXX_grp_delta_arr[(GRP_ID)-1])
```

GET_XXXXCAP_MOIS_GRP_ADJ_DELTA (GRP_ID)

XXXXcap水分群隣接差を得ます。戻り値は水分群の隣接差を示す32ビット整数です。

```
#define GET_MUTLCAP_MOIS_GRP_ADJ_DELTA (GRP_ID) (mois_mutl_grp_adj_delta_arr[(GRP_ID)-1])
```

GET_XXXXCAP_GLOB_LOCK_STATE

XXXXcap水分群の水分閉鎖状態を得ます。戻り値はどれかの水分群が水分全体閉鎖の場合に1、水分全体閉鎖の水分群がない場合に0です。

```
#define GET_MOIS_MUT_GLOB_LOCK_STATE (mois_lock_global_mutl)
```

GET_XXXXCAP_SENSOR_NOISE_STATUS (SENSOR_NUMBER)

特定感知部の雑音状況を得ます。戻り値は感知部が雑音多の場合に1で、感知部が雑音性なしの場合に0を返します。

```
#define GET_XXXXCAP_SENSOR_NOISE_STATUS (SENSOR_NUMBER) (p_XXXXcap_measure_data-> ¥  
p_sensor_noise_status [(SENSOR_NUMBER / 8)] & (1 << (SENSOR_NUMBER % 8))) >> (SENSOR_NUMBER % 8)
```

GET_ROTOR_SLIDER_POSITIONS (ROTOR_SLIDER_NUMBER)

回転子角度または摺動子位置を得ます。これらの値は回転子または摺動子の状態に対応する感知部の状態が検出の時にだけ有効です。

ROTOR_SLIDER_NUMBERはどれの位置が得られるかのためのパラメータです。

このマクロは回転子角度または感知部位置を返します。

```
#define GET_XXXXCAP_ROTOR_SLIDER_POSITION(ROTOR_SLIDER_NUMBER) p_XXXXcap_measure_data->  ¥
    p_rotor_slider_values [ROTOR_SLIDER_NUMBER]
```

DEF_TOUCH_MUTLFCAPは相互容量接触採取法を許可するために応用で1に設定されなければなりません。

DEF_TOUCH_SELFCAPは自己容量接触採取法を許可するために応用で1に設定されなければなりません。

11.2. 型定義

領域	単位	説明
threshold_t	uint8_t	感知部検出閾値を設定する符号なし8ビット数値
sensor_id_t	uint8_t	感知部番号型
touch_current_time_t	uint16_t	現在時型
touch_delta_t	int16_t	接触感知部差値型
touch_acq_status_t	uint16_t	接触測定の状態
mois_snsr_threshold_t	int32_t	個別感知部用水分閾値
mois_system_threshold_t	int32_t	システム全体用水分閾値

11.3. 列挙

11.3.1. 利得設定 (tag_gain_t)

接触チャネル毎の利得

利得は接触での接触感度の尺度増加を許すために基本チャネル毎に適用されます。

範囲：GAIN_1(増幅なし)～GAIN_32(32倍増幅)

データ領域

- GAIN_1
- GAIN_2
- GAIN_4
- GAIN_8
- GAIN_16
- GAIN_32

11.3.2. 濾波水準設定 (tag_filter_level_t)

接触ライブラリFILTER LEVEL設定

濾波水準設定は各採取を解決するために採取される採取数を制御します。より高い濾波水準設定は雑音が多い状況下で改善された信号対雑音比を提供し、同時に電力消費増加に帰着する測定に対する総時間を増します。touch_api_ptc.hでfilter_level_tを参照してください。

範囲：FILTER_LEVEL_1(採取なし)～FILTER_LEVEL_64(64採取)

データ領域

- FILTER_LEVEL_1
- FILTER_LEVEL_2
- FILTER_LEVEL_4
- FILTER_LEVEL_8
- FILTER_LEVEL_16
- FILTER_LEVEL_32
- FILTER_LEVEL_64

11.3.3. 自動過採取設定 (tag_auto_os_t)

自動過採取は'濾波水準'の既定設定で不安定な信号が検出された時に感知部チャネルの自動過採取を制御します。自動過採取許可は不安定な信号が観測された時に対応する感知部チャネルで取られる'濾波水準'×'自動過採取'の採取数に帰着します。'濾波水準'がFILTER_LEVEL_4に設定され、'自動過採取'がAUTO_OS_4に設定された場合、安定した信号で4過採取が取られ、不安定な信号が検出された時に4+16過採取が取られます。

範囲：AUTO_OS_DISABLE(過採取禁止)～AUTO_OS_128(128過採取)

データ領域

- AUTO_OS_DISABLE
- AUTO_OS_2
- AUTO_OS_4
- AUTO_OS_8
- AUTO_OS_16
- AUTO_OS_32
- AUTO_OS_64
- AUTO_OS_128

11.3.4. 低電力感知部走査速度 (tag_lowpower_scan_int_t)

CPUが活性からスタンバイ動作へ戻ると、低電力感知部として構成設定された感知値がこの間隔で走査されます。このパラメータに対する大きな値は電力消費を減らしますが、低電力感知部に対する応答時間を増します。

注: この列挙はATmegaデバイスにだけ適用できます。

範囲 : LOWPOWER_PER0_SCAN_3_P_9_MS ~ LOWPOWER_PER0_SCAN_250_MS

データ領域

- LOWPOWER_PER0_SCAN_3_P_9_MS
- LOWPOWER_PER0_SCAN_7_P_8_MS
- LOWPOWER_PER0_SCAN_15_P_625_MS
- LOWPOWER_PER0_SCAN_31_P_25_MS
- LOWPOWER_PER0_SCAN_62_P_5_MS
- LOWPOWER_PER0_SCAN_125_MS
- LOWPOWER_PER0_SCAN_250_MS

11.3.5. ライブライ異常符号 (tag_touch_ret_t)

接触ライブライ異常符号

データ領域

- TOUCH_SUCCESS - 接触操作の成功裏の完了
- TOUCH_ACQ_INCOMPLETE - 保留中の前の接触測定でライブライが多忙
- TOUCH_INVALID_INUT_PARAM - 無効な入力パラメータ
- TOUCH_INVALID_LIB_STATE - 操作が現在の接触ライブライの状態です許されません。
- TOUCH_INVALID_SELF_CAP_CONFIG_PARAM - 無効な自己容量構成設定入力パラメータ
- TOUCH_INVALID_MUTL_CAP_CONFIG_PARAM - 無効な相互容量構成設定入力パラメータ
- TOUCH_INVALID_RECAL_THRESHOLD - 無効な再校正閾値入力値
- TOUCH_INVALID_CHANNEL_NUM - 構成設定したチャンネル総数を越えたチャンネル数パラメータ
- TOUCH_INVALID_SENSOR_TYPE - 無効な感知器型。感知器型はSENSOR_TYPE_UNASSIGNEDであってはなりません。
- TOUCH_INVALID_SENSOR_ID - 無効な感知部番号パラメータ
- TOUCH_INVALID_RS_NUM - 回転子/摺動子を構成設定しようとする時に回転子/摺動子の番号を0として設定

11.3.6. 応用異常符号 (tag_touch_app_err_t)

応用異常符号は下で一覧にされます。

データ領域

- TOUCH_INIT_CONFIG_ERR - touch_XXXXcap_sensors_initが非対応/不完全なパラメータで与えられた。
- TOUCH_SENSOR_CONFIG_ERR - touch_XXXXcap_sensor_configが非対応パラメータで与えられた。接触ライブライがTOUCH_STATE_INITでない。
- TOUCH_MEASURE_INCOMPLETE - 無効入力パラメータのためにtouch_measure APIが異常を持つ。無効な接触ライブライ状態だった。
- TOUCH_MEASURE_CC_CAL_FAILED - ハードウェア校正異常。ハードウェアを調べてそれが正しいことを確実にしてください。異常が持続する場合、感知器設計指針に対する使用者手引書を調べてください。

11.3.7. 接触チャンネル (tag_channel_t)

感知部の感知部開始と終了のチャンネル型。チャンネル番号は0の値で始まります。

データ領域

CHANNEL_0 ~ CHANNEL_255

11.3.8. 接触ライブライ状態 (tag_touch_lib_state_t)

接触ライブライ状態

データ領域

- TOUCH_STATE_NULL - 接触ライブライは未初期化です。全ての感知器が禁止されます。
- TOUCH_STATE_INIT - 接触ライブライは初期化されています。
- TOUCH_STATE_READY - 接触ライブライは許可された感知器で新しい容量測定を開始する準備が整っています。
- TOUCH_STATE_CALIBRATE - 接触ライブライは全ての感知器で校正を実行中です。
- TOUCH_STATE_BUSY - 接触ライブライは進行中の容量測定で多忙です。

11.3.9. 感知器型 (tag_touch_lib_state_t)

利用可能な感知器型

データ領域

- `SENSOR_TYPE_UNASSIGNED` – 感知部が未だ構成設定されていません。
- `SENSOR_TYPE_KEY` – キー感知器型
- `SENSOR_TYPE_ROTATOR` – 回転子感知器型
- `SENSOR_TYPE_LUMP` – 一括感知器型
- `SENSOR_TYPE_SLIDER` – 摺動子感知器型
- `MAX_SENSOR_TYPE` – 試験用列挙の最大値

11.3.10. 接触感知型 (tag_touch_acq_t)

2つの電荷移動技術型に基づいて、容量接触感知は相互容量感知または自己容量感知のどちらかで有り得ます。

データ領域

- `TOUCH_MUTUAL` – 相互容量感知
- `TOUCH_SELF` – 自己容量感知
- `MAX_TOUCH_ACQ` – 列挙の最大値

11.3.11. 接触ライブラリ採取動作形態 (tag_touch_acq_mode_t)

接触ライブラリ採取動作形態

データ領域

RAW_ACQ_MODE

生採取動作が使われると、一旦最新の信号の値が利用可能になると、`measure_complete_callback`関数が直ちに呼ばれます。この動作では、接触ライブラリがどの後処理も実行しません。故に、この動作で参照基準、感知部の状態や回転子/摺動子の位置は更新されません。

NOMAL_ACQ_MODE

標準採取動作が使われると、接触ライブラリが得られた信号値の処理を完了した後でだけ`measure_complete_callback`関数が呼ばれます。この動作で参照基準、感知部の状態と回転子/摺動子の位置は更新されます。

11.3.12. 校正自動調節設定 (tag_auto_tune_type_t)

接触ライブラリPTC前置分周器クロックと直列抵抗の自動調節設定

データ領域

- `AUTO_TUNE_NONE` – 自動調節動作禁止。この動作は使用者定義のPTC前置分周器と直列抵抗値を使います。
- `AUTO_TUNE_PRSC` – 最良の雑音性能のためにPTC前置分周器を自動調節。この動作は使用者定義直列抵抗値を使います。
- `AUTO_TUNE_RSEL` – 最低電力消費のために直列抵抗を自動調節。この動作は使用者定義PTC前置分周器値を使います。

11.3.13. PTC採取周波数動作構成設定 (tag_freq_mode_sel_t)

周波数動作構成設定任意選択は以下の動作に構成設定されることをPTC採取に許します。

- 周波数と飛び跳ねとスペクトラム拡散を禁止
- 中央値濾波器と共に許可された周波数飛び跳ね
- 中央値濾波器なしで許可された周波数スペクトラム拡散
- 中央値濾波器と共に許可された周波数スペクトラム拡散

範囲: `FREQ_MODE_NONE`(周波数飛び跳ねとスペクトラム拡散なし)～`FREQ_MODE_SPREAD_MIDIAN`(中央値濾波器付きスペクトラム拡散)

データ領域

- `FREQ_MODE_NONE` – 0u
- `FREQ_MODE_HOP` – 1u
- `FREQ_MODE_SPREAD` – 2u
- `FREQ_MODE_SPREAD_MIDIAN` – 3u

11.3.14. PTCクロック前置分周器設定 (tag_prsc_div_sel_t)

`touch.c`で`touch_configure_ptc_clock()` APIを参照してください。

例: PTCへの標準クロック入力=4MHzの場合

- `PRSC_DIV_SEL_1`はPTCクロックを4MHzに設定します。
- `PRSC_DIV_SEL_2`はPTCクロックを2MHzに設定します。
- `PRSC_DIV_SEL_4`はPTCクロックを1MHzに設定します。
- `PRSC_DIV_SEL_8`はPTCクロックを500kHzに設定します。

データ領域

- `PRSC_DIV_SEL_1`
- `PRSC_DIV_SEL_2`
- `PRSC_DIV_SEL_4`
- `PRSC_DIV_SEL_8`

11.3.15. PTC直列抵抗設定 (tag_rsel_val_t)

相互容量動作に対して、この直列抵抗はYピンで内部的に切り替えられます。自己容量動作に対して、直列抵抗は感知部ピンで内部的に切り替えられます。

例:

- RSEL_VAL_0は内部直列抵抗を0Ωに設定します。
- RSEL_VAL_20は内部直列抵抗を20kΩに設定します。
- RSEL_VAL_50は内部直列抵抗を50kΩに設定します。
- RSEL_VAL_100は内部直列抵抗を100kΩに設定します。

データ領域

- RSEL_VAL_0
- RSEL_VAL_20
- RSEL_VAL_50
- RSEL_VAL_100

11.3.16. PTC採取周波数遅延設定 (tag_freq_hop_sel_t)

PTC採取周波数はPTCへの標準クロック入力とPTCクロック前置分周器設定に依存します。この遅延設定は与えられた感知部で連続的な測定間にn個のPTCクロック周期を挿入し、それによってPTC採取周波数を変えます。FREQ_HOP_SEL_1設定は連続的な測定間に0個のPTCクロック周期を挿入します。FREQ_HOP_SEL_16設定は15個のPTCクロック周期を挿入します。故に、より大きな遅延設定はより小さな遅延設定に比べて与えられた感知部での容量測定に対して必要とされる総時間を増やします。

望まれる設定は採取周波数と同じ周波数での雑音を避けます。

データ領域

- FREQ_HOP_SEL_1
- FREQ_HOP_SEL_2
- FREQ_HOP_SEL_3
- FREQ_HOP_SEL_4
- FREQ_HOP_SEL_5
- FREQ_HOP_SEL_6
- FREQ_HOP_SEL_7
- FREQ_HOP_SEL_8
- FREQ_HOP_SEL_9
- FREQ_HOP_SEL_10
- FREQ_HOP_SEL_11
- FREQ_HOP_SEL_12
- FREQ_HOP_SEL_13
- FREQ_HOP_SEL_14
- FREQ_HOP_SEL_15
- FREQ_HOP_SEL_16

11.3.17. AKS群 (tag_aks_group_t)

特定AKS群に属する感知部についての情報を提供します。NO_AKS_GROUPはその感知部がどのAKS群にも属せずに消去され得ないことを示します。AKS_GROUP_xはその感知器がAKS群xに属することを示します。

データ領域

- NO_AKS_GROUP
- AKS_GROUP_1
- AKS_GROUP_2
- AKS_GROUP_3
- AKS_GROUP_4
- AKS_GROUP_5
- AKS_GROUP_6
- AKS_GROUP_7
- MAX_AKS_GROUP – 試験用列举型の最大値

11.3.18. 感知部ヒステリシス設定 (tag_hysteresis_t)

感知部検出ヒステリシス値。これは感知部検出閾値の%として表されます。HYST_x=ヒステリシス値は検出閾値のx%(切り捨て丸め)です。

注: 最小値は2が使われます。

例: 検出閾値=20の場合

- 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、しかし値は2にハード制限されます。)

データ領域

- HYST_50
- HYST_25
- HYST_12_5
- HYST_6_25
- MAX_HYST – 試験用列举型の最大値

11.3.19. 感知部再校正閾値 (tag_recal_threshold_t)

これは感知部検出閾値の%として表されます。RECAL_x=再校正閾値は検出閾値のx%(切り捨て丸め)です。

注: 最小値は4が使われます。

例: 検出閾値=40の場合

- 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、しかし値は4にハード制限されます。)

データ領域

- RECAL_100
- RECAL_50
- RECAL_25
- RECAL_12_5
- RECAL_6_25
- MAX_RECAL – 試験用列举型の最大値

11.3.20. 回転子/摺動子分解能 (tag_resolution_t)

回転子と摺動子に対して報告される角度や位置の分解能

- RES_x_BIT = 回転子/摺動子が報告するxビット値

例: 摺動子の分解能がRES_7_BITなら、報告される位置は0～127の範囲です。

データ領域

- RES_1_BIT • RES_2_BIT • RES_3_BIT • RES_4_BIT • RES_5_BIT • RES_6_BIT • RES_7_BIT
- RES_8_BIT • MAX_RES - 試験用列挙型の最大値

11.3.21. PTC感知部雑音閉鎖設定 (nm_sensor_lockout_t)

感知部閉鎖設定任意選択は以下の動作で構成設定されることをシステムに許します。

- SINGLE_SENSOR_LOCKOUT - 単一感知部を閉鎖することができます。
- GLOBAL_SENSOR_LOCKOUT - 全ての感知部を接触検出に対して閉鎖することができます。
- NO_LOCK_OUT - 全ての感知部が接触検出に利用可能です。

範囲: SINGLE_SENSOR_LOCKOUT～NO_LOCK_OUT

データ領域

- SINGLE_SENSOR_LOCKOUT - 0u
- GLOBAL_SENSOR_LOCKOUT - 1u
- NO_LOCK_OUT - 2u

11.3.22. PTC汎用入出力状態 (ptc_gpio_state_t)

測定されない状況のPTC線状態はこの列挙を用いて設定することができます。

- PULLHIGH_WHEN_NOT_MEASURED - PTC線の既定状態がVCCであることを示します。
- GND_WHEN_NOT_MEASURED - PTC線の既定状態が接地されることを示します。

範囲: PULLHIGH_WHEN_NOT_MEASURED=0とGND_WHEN_NOT_MEASURED

データ領域

- PULLHIGH_WHEN_NOT_MEASURED • GND_WHEN_NOT_MEASURED

11.3.23. 水分群設定 (moisture_grp_t)

感知部はこの型を用いて水分群で構成設定することができます。

- MOIS_DISABLED - 感知部はどの水分群にも属さないことを示します。
- MOIS_GROUP_x - 感知部が水分群xに属することを示します。

範囲: MOIS_DISABLED=0～MOIS_GRP_7

データ領域

- MOIS_DISABLED=0 • MOIS_GROUP_0 • MOIS_GROUP_1 • MOIS_GROUP_2 • MOIS_GROUP_3
- MOIS_GROUP_4 • MOIS_GROUP_5 • MOIS_GROUP_6 • MOIS_GROUP_7 • MOIS_GROUPN

11.3.24. 複数接触群設定 (mltch_grp_t)

感知部はこの型を用いて複数接触群で構成設定することができます。

- MLTCH_NONE - 感知部はどの複数接触群にも属さないことを示します。
- MLTCH_GROUP_x - 感知部が複数接触群xに属することを示します。

範囲: MLTCH_NONE=0～MLTCH_GRP_7

データ領域

- MLTCH_NONE=0 • MLTCH_GROUP_0 • MLTCH_GROUP_1 • MLTCH_GROUP_2 • MLTCH_GROUP_3
- MLTCH_GROUP_4 • MLTCH_GROUP_5 • MLTCH_GROUP_6 • MLTCH_GROUP_7 • MLTCH_GROUPN

11.3.25. 接触動作構成設定 (tag_tch_mode)

接触動作は構成設定することができます。

注: これはATmegaデバイスにだけ適用できます。

データ領域

- TCH_MODE_POLLED - ホールグ動作
- TCH_MODE_ISR - 割り込み動作
- TCH_MODE_NONE - 接触動作なし

11.3.26. 起動動作 (tag_trigger_mode)

連続ハードウェアPTC採取用起動元。これは内部128kHzクロックのnクロック周期です。

注: これはATmegaデバイスにだけ適用できます。

データ領域

- TCH_TRIGGER_128KHZ_4MS

TCH_TRIGGER_128KHZ_32MS

TCH_TRIGGER_128KHZ_256MS
- TCH_TRIGGER_128KHZ_8MS

TCH_TRIGGER_128KHZ_64MS
- TCH_TRIGGER_128KHZ_16MS

TCH_TRIGGER_128KHZ_128MS

11.4. データ構造体

11.4.1. 接触ライブラリ タイミング情報 (tag_touch_time_t)

接触ライブラリ時間パラメータ

データ領域

領域	単位	説明
measurement_period_ms	uint16_t	msでの接触測定周期。この変数はどの位の頻度で新しい接触測定が行われなければならないかを決めます。
current_time_ms	volatile uint16_t	計時器ISRによって設定される現在時間
time_to_measure_touch	volatileuint8_t	接触を測定するための時間の時に計時器ISRによって設定されるフラグ

11.4.2. 感知器情報 (tag_sensor_t)

感知器関連情報を格納するための感知器構造体

データ領域

キーワード	型	説明
state	uint8_t	感知部状態(校正、ON、OFF、濾波IN、濾波OUT、禁止、位置再校正)
general_counter	uint8_t	校正、変動などに使われる汎用計数器
ndil_counter	uint8_t	検出積分に使われる計数器
type_aks_pos_hyst	uint8_t	ビット7,6: 感知部型{00:キー,01:回転子,10:摺動子,11:予約}、ビット5〜3: AKS群0〜7{0=AKS群なし}、ビット2: 正再校正フラグ、ビット1,0: ヒステリシス
threshold	uint8_t	感知部検出閾値
from_channel	uint8_t	キー:チャンネルから=チャンネルへ、回転子:先頭チャンネル、摺動子:最左端チャンネル、に対するチャンネルからの感知部。注:回転子/摺動子に対してだけto_channelが必要です。
to_channel	uint8_t	キー:不使用、回転子:左下チャンネル、摺動子:中央チャンネル。
index	uint8_t	回転子/摺動子の値の配列に対する指標

11.4.3. 全体感知器構成設定情報 (tag_touch_global_param_t)

接触ライブラリ全体パラメータ

データ領域

領域	単位	説明
di	uint8_t	検出積分(DI)限度
atch_drift_rate	uint8_t	接触から感知部離脱変動速度
tch_drift_rate	uint8_t	接触へ向かう感知部変動速度
max_on_duration	uint8_t	最大ON持続時間
drift_hold_time	uint8_t	感知部変動保持時間
atch_recal_delay	uint8_t	接触から感知部離脱再校正遅延
cal_seq_1_count	uint8_t	感知部校正偽装群発計数
cal_seq_2_count	uint8_t	感知部校正設定群発計数
recal_threshold	recal_threshold_t	接触から離脱感知部再校正閾値
touch_postprocess_mode	uint16_t	感知部後処理動作形態
auto_os_sig_stability_limit	uint8_t	自動過採取機能用安定限度
auto_tune_sig_stability_limit	uint16_t	周波数自動調節機能用安定限度

[次頁](#)へ続く

前頁からの続き

領域	単位	説明
auto_freq_tune_in_cnt	uint8_t	周波数自動調節IN計数器
nm_sig_stability_limit	uint16_t	雑音測定用安定限度
nm_noise_limit	uint8_t	雑音限度
nm_enable_sensor_lock_out	nm_sensor_lockout_t	感知部閉鎖機能変数
nm_lockout_countdown	uint8_t	雑音測定用閉鎖減少計数
Charge_share_delay	uint8_t	充電共用遅延値: SAM C20, SAM C21, SAM L22, ATmegaデバイスにだけ適用

11.4.4. 濾波器呼び戻しデータ型 (tag_touch_filter_data_t)

接触ライブラリ濾波器呼び戻しデータ型

データ領域

領域	単位	説明
num_channel_signals	uint16_t	測定した信号値一覧の長さ
p_channel_signals	uint16_t	各チャネルに対して測定した信号値へのポインタ

11.4.5. 測定データ型 (tag_touch_measure_data_t)

接触ライブラリ測定データ型

データ領域

領域	単位	説明
measurement_done_touch	volatile uint8_t	最新接触状態が利用可能な時にtouch_XXXXcap_measure_complete_callback()関数によって設定されるフラグ
acq_status	touch_acq_status_t	接触測定の状況
num_channel_signals	uint16_t	測定した信号値一覧の長さ
*p_channel_signals	uint16_t	各チャネルに対して測定した信号値へのポインタ
num_channel_references	uint16_t	測定した参照基準値一覧の長さ
*p_channel_references	uint16_t	各チャネルに対して測定した参照基準値へのポインタ
num_sensor_states	uint8_t	感知部状態バイト数
*p_sensor_states	uint8_t	各感知部の接触状態へのポインタ
num_rotor_slider_values	uint8_t	回転子と摺動子の位置値一覧の長さ
*p_rotor_slider_values	uint8_t	回転子と摺動子の位置値へのポインタ
num_sensors	uint16_t	感知部データ一覧の長さ
*p_cc_calibration_vals	uint16_t	与えられた感知部チャネルに対して校正された補償値へのポインタ
*p_sensors	sensor_t	感知部データへのポインタ
*p_sensor_noise_status	uint8_t	感知部の雑音状態へのポインタ
*p_nm_ch_noise_val	uint16_t	各チャネルの雑音水準値へのポインタ
*p_sensor_mois_status	uint8_t	水分状態へのポインタ
*p_auto_os_status	uint8_t	自動過採取状態へのポインタ
cc_calib_status_flag	uint8_t	このフラグはCC校正進行中の時に設定されます。

11.4.6. 感知部構成設定パラメータ (tag_touch_XXXXcap_param_t)

接触ライブラリの自己容量と相互容量の感知部パラメータ

データ領域

領域	単位	説明
aks_group	aks_group_t	感知部が属すAKS群
detect_threshold	threshold_t	感知部検出閾値を設定する符号なし8ビット数値

次頁へ続く

前頁からの続き

領域	単位	説明
detect_hysteresis	hysteresis_t	感知部検出ヒステリシス値。これは感知部検出閾値の%として表されます。HYST_x=ヒステリシス値は検出閾値のx%(切り捨て丸め)です。最小値は2が使われます。例: 検出閾値=20なら、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、値は2にハード制限されます。)
position_resolution	resolution_t	回転子と摺動子に対して報告される角度または位置の分解能。RES_x_BIT=回転子/摺動子はxビット値を報告します。例: 摺動子分解能がRES_7_BITなら、報告される位置は0~127の範囲です。
position_hysteresis	uint8_t	感知部位置ヒステリシス。これは回転子や摺動子でだけ有効です。ビット1.0:ヒステリシス。 注: このパラメータは相互容量法でだけ有効です。

11.4.7. 感知部採取パラメータ (tag_touch_XXXXcap_acq_param_t)

感知部採取パラメータ

データ領域

領域	単位	説明
*p_XXXXcap_gain_per_node	gain_t	節点毎の利得へのポインタ
touch_XXXXcap_freq_mode	Freq_mode_sel_t	採取周波数動作形態構成設定
*XXXXcap_ptc_prsc	prsc_div_sel_t	PTCクロック前置分周器値へのポインタ
*XXXXcap_resistor_value	r_sel_val_t	PTC直列抵抗値へのポインタ
p_XXXXcap_hop_freqs	*freq_hop_sel_t	採取周波数設定へのポインタ
*p_XXXXcap_filter_level	filter_level_t	濾波水準へのポインタ
*p_XXXXcap_auto_os	auto_os_t	自動過採取へのポインタ
*XXXXcap_ptc_prsc_cc_cal	prsc_div_sel_t	CC校正中のPTCクロック前置分周器値へのポインタ
*XXXXcap_resistor_value_cc_cal	r_sel_val_t	CC校正中のPTC感知抵抗値へのポインタ

11.4.8. 自己容量感知部構成設定 (tag_selfcap_config_t)

接触ライブラリ自己容量構成設定入力型

データ領域

領域	単位	説明
num_channels	uint16_t	チャンネル数
num_sensors	uint16_t	感知部数
num_rotors_and_sliders	uint8_t	回転子/摺動子の数
global_param	touch_global_param_t	全体感知部構成設定情報
touch_selfcap_acq_param	touch_selfcap_acq_param_t	感知部採取パラメータ情報
*p_data_blk	uint8_t	データ塊緩衝部へのポインタ
buffer_size	uint16_t	データ塊緩衝部の大きさ
*p_selfcap_y_nodes	uint16_t	自己容量節点へのポインタ
self_quick_reburst_enable	uint8_t	迅速再群発許可
(touch_filter_data_t *p_filter_data)	void(*filter_callback)	自己容量濾波器呼び戻し
enable_freq_auto_tune	uint8_t	周波数自動調節許可
enable_noise_measurement	uint8_t	雑音測定許可
nm_buffer_cnt	uint8_t	メモリ割り当て緩衝部
self_moistlrnce_enable	uint8_t	自己容量耐水性許可フラグ
self_moist_groups	uint8_t	自己容量水分群数
self_moist_quick_reburst_enable	uint8_t	水分迅速再群発許可
self_ptc_gpio_state	ptc_gpio_state_t	自己容量PTCピンに対するGPIO状態
tlb_feature_list	tlb_init_fn_ptr	ライブラリ機能一覧

11.4.9. 相互容量感知部構成設定 (tag_mutlcap_config_t)

接触ライブラリ相互容量構成設定入力型

データ領域

領域	単位	説明
num_channels	uint16_t	チャンネル数
num_sensors	uint16_t	感知部数
num_rotors_and_sliders	uint8_t	回転子/摺動子の数
global_param	touch_global_param_t	全体感知部構成設定情報
touch_mutlcap_acq_param	touch_mutlcap_acq_param_t	感知部採取パラメータ情報
*p_data_blk	uint8_t	データ塊緩衝部へのポインタ
buffer_size	uint16_t	データ塊緩衝部の大きさ
*p_mutlcap_xy_nodes	uint16_t	相互容量XY節点へのポインタ
mutl_quick_reburst_enable	uint8_t	迅速再群発許可
(touch_filter_data_t *p_filter_data)	void(*filter_callback)	相互容量濾波器呼び戻し
enable_freq_auto_tune	uint8_t	周波数自動調節許可
enable_noise_measurement	uint8_t	雑音測定許可
nm_buffer_cnt	uint8_t	メモリ割り当て緩衝部
mutl_mois_tlrnce_enable	uint8_t	相互容量耐水性許可フラグ
mutl_mois_groups	uint8_t	相互容量水分群数
mutl_mois_quick_reburst_enable	uint8_t	水分迅速再群発許可
mutl_ptc_gpio_state	ptc_gpio_state_t	相互容量PTCピンに対するGPIO状態
tlib_feature_list	tlib_init_fn_ptr	ライブラリ機能一覧

11.4.10. 水分構造体 (tag_snsr_mois_t)

水分と複数接触群の情報を格納するための構造体

データ領域

領域	単位	説明
mois_grp	uint8_t	水分群番号
multch_grp	uint8_t	複数接触群番号

11.4.11. 接触ライブラリ入力構成設定 (touch_config_t)

接触ライブラリ入力構成設定構造体

データ領域

領域	単位	説明
p_mutlcap_config	touch_mutlcap_config_t	相互容量構成設定構造体へのポインタ
p_selfcap_config	touch_selfcap_config_t	自己容量構成設定構造体へのポインタ
ptc_isr_lvl	uint8_t	PTC ISR優先レベル。 注: これはSAMデバイスにだけ適用
tch_mode	tch_mode_t	接触動作構成設定。 注: これはATmegaデバイスにだけ適用

11.4.12. ライブラリ機能一覧 (tag_tlib_init_fn_ptr_t)

接触ライブラリ支援機能初期化部

データ領域

領域	単位	説明
auto_tune_init	void(*auto_tune_init)	自動調節機能初期化部
auto_os_init	uint32_t (*auto_os_init)	自動過採取機能初期化部
lk_chk	void(*lk_chk)	感知部閉鎖機能初期化部
enable_aks	void enable_aks(void)	AKS機能初期化部

11.4.13. 接触ライブラリ情報 (tag_touch_info_t)

接触ライブラリ情報構造体

データ領域

領域	単位	説明
tlib_state	touch_tlib_state_t	接触ライブラリ状態が指定されます。
num_channels_in_use	uint16_t	対応する感知部が禁止または許可に関わらず、使用中のチャネル数
num_sensors_in_use	uint16_t	感知部が禁止または許可に関わらず、使用中の感知部数
num_rotors_sliders_in_use	uint8_t	回転子/摺動子が禁止または許可に関わらず、使用中の回転子/摺動子数
max_channels_per_rotor_slider	uint8_t	回転子または摺動子毎に可能な最大チャネル数

11.4.14. 接触ライブラリ版情報 (touch_libver_info_t)

接触ライブラリ版情報構造体

データ領域

領域	単位	説明
chip_id	uint32_t	チップID
product_id	uint16_t	製品ID
fw_version	uint16_t	接触ライブラリ版 ビット15～12 : (予約) ビット11～8 : TLIB_MAJOR_VERSION ビット7～4 : TLIB_MINOR_VERSION ビット3～0 : TLIB_PATCH_VERSION

11.5. 全体変数

領域	単位	説明
touch_time	touch_time_t	これはライブラリ タイミング情報を保持します。
touch_acq_status	touch_acq_status_t	これは接触ライブラリ採取状況を保持します。
cc_cal_max_signal_limit	uint16_t	CC校正最大信号限度変数
cc_cal_min_signal_limit	uint16_t	CC校正最小信号限度変数
*p_selfcap_measure_data	touch_measure_data_t	これは自己容量法測定データ ポインタを保持します。
*p_mutlcap_measure_data	touch_measure_data_t	これは相互容量法測定データ ポインタを保持します。
wake_up_touch	uint8_t	ライブラリから応用への接触起き上がり状態
low_power_mode	uint8_t	ライブラリから応用への低電力動作状態
mois_lock_global_mutl	uint8_t	相互容量法用水分全体施錠変数
mois_lock_global_self	uint8_t	自己容量法用水分全体施錠変数

11.6. API

11.6.1. 感知部初期化と初期化解除

```
touch_ret_t touch_mutlcap_sensors_init (touch_config_t * p_touch_config)
```

```
touch_ret_t touch_selfcap_sensors_init (touch_config_t * p_touch_config)
```

このAPIは使用者によって提供される相互容量法または自己容量法のピン、レジスタ、感知部構成設定で接触ライブラリを初期化するのに使われます。

パラメータ : p_touch_config - 接触構成設定構造体へのポインタ

戻り : touch_ret_t - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_sensors_deinit(void)
```

```
touch_ret_t touch_selfcap_sensors_deinit(void);
```

このAPIは特定感知群に対して感知部を初期化解除するのに使うことができます。

パラメータ : void

戻り : touch_ret_t - 接触ライブラリ異常状態

11.6.2. 感知部構成設定と構成設定

```
touch_ret_t touch_mutlcap_sensor_config (sensor_type_t sensor_type, 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, sensor_id_t * p_sensor_id)

touch_ret_t touch_selfcap_sensor_config (sensor_type_t sensor_type, 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, sensor_id_t * p_sensor_id)
```

このAPIはキー、回転子、摺動子の型の感知部を構成設定するのに使うことができます。

データ領域

領域	説明
sensor_type	キー、一括、回転子、摺動子の型にすることができます。
from_channel	摺動子感知器での最初のチャネル
to_channel	摺動子感知器での最後のチャネル
aks_group	(存在すれば)感知部がある(属す)AKS群
detect_threshold	感知部検出閾値
detect_hysteresis	感知部検出ヒステリシス値
position_resolution	報告される位置値の分機能
position_hysteresis	(相互容量動作に対してだけ利用可能な)位置値に対するヒステリシス
p_sensor_id	構成設定された感知部の感知部ID値は接触ライブラリによって更新されます。

戻り: touch_ret_t - 接触ライブラリ異常状態

11.6.3. 感知部校正

```
touch_ret_t touch_mutlcap_sensors_calibrate (auto_tune_type_t )

touch_ret_t touch_selfcap_sensors_calibrate (auto_tune_type_t )
```

このAPIは接触測定を開始するのに先立って初回に感知部を校正するのに使われます。このAPIは走行中に接触感知部パラメータのどれかが変更された時に感知部の校正を強制するのに利用することもできます。

戻り: touch_ret_t - 接触ライブラリ異常状態

11.6.4. 感知部測定

```
touch_ret_t touch_mutlcap_sensors_measure (touch_current_time_t current_time_ms, ¥
touch_acq_mode_t mutlcap_acq_mode, void(*) (void) measure_complete_callback)

touch_ret_t touch_selfcap_sensors_measure (touch_current_time_t current_time_ms, ¥
touch_acq_mode_t selfcap_acq_mode, void(*) (void) measure_complete_callback)
```

このAPIは接触測定を開始するのに使うことができます。

パラメータ: current_time_ms - msでの現在時間

measure_complete_callback - 測定完了を示すための割り込み呼び出し

戻り: touch_ret_t - 接触ライブラリ異常状態

11.6.5. 感知部一時停止と再開

```
touch_ret_t touch_suspend_ptc(void)

touch_ret_t touch_resume_ptc(void)
```

touch_suspend_ptc関数はPTCライブラリの現在の測定周回を一時停止します。操作の完了は応用によって初期化されなければならない呼び戻しポイントを通して示されます。「[感知部全体パラメータ](#)」を参照してください。

touch_resume_ptc関数はtouch_suspend_ptcを用いて一時停止されたPTCライブラリの現在の測定を再開します。touch_resume_ptc関数が応用によって呼ばれた後、touch_xxxxcap_sensors_measure APIは測定完了呼び戻し関数を受け取った後でだけ呼ばれるべきです。

パラメータ: void

戻り: touch_ret_t - 接触ライブラリ異常状態

11.6.6. 感知部禁止と再許可

```
touch_ret_t touch_mutlcap_sensor_disable (sensor_id_t sensor_id)
```

```
touch_ret_t touch_selfcap_sensor_disable (sensor_id_t sensor_id)
```

このAPIはどれかの感知部を禁止するのに使うことができます。

パラメータ: `sensor_id` - 禁止されることが必要な感知部番号

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_sensor_reenable (sensor_id_t sensor_id, uint8_t no_calib)
```

```
touch_ret_t touch_selfcap_sensor_reenable (sensor_id_t sensor_id, uint8_t no_calib)
```

このAPIは禁止された感知部を再許可するのに使うことができます。

パラメータ: `sensor_id` - 再許可が必要な感知部番号

`no_calib` - 値が1に設定されると、強制校正が適用できません。値が0に設定されると、強制校正が適用されます。

戻り: `touch_ret_t` - 接触ライブラリ異常状態

11.6.7. 感知部構成設定読み戻し

```
touch_ret_t touch_mutlcap_sensor_get_acq_config ¥  
(touch_mutlcap_acq_param_t * p_touch_mutlcap_acq_param)
```

```
touch_ret_t touch_selfcap_sensor_get_acq_config ¥  
(touch_selfcap_acq_param_t * p_touch_selfcap_acq_param)
```

このAPIは感知部採取パラメータを読み戻すのに使うことができます。

パラメータ: `p_touch_mutlcap_acq_param` - 相互容量用採取パラメータ

`p_touch_selfcap_acq_param` - 自己容量用採取パラメータ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_sensor_get_config ¥  
(sensor_id_t sensor_id, touch_mutlcap_param_t *p_touch_sensor_param)
```

```
touch_ret_t touch_selfcap_sensor_get_config ¥  
(sensor_id_t sensor_id, touch_selfcap_param_t *p_touch_sensor_param)
```

このAPIは感知部構成設定パラメータを読み戻すのに使うことができます。

パラメータ: `sensor_id` - パラメータが読み戻されなければならない感知部ID

`p_touch_sensor_param` - 相互または自己の容量に対する感知部パラメータ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_sensor_get_delta (sensor_id_t sensor_id, touch_delta_t * p_delta)
```

```
touch_ret_t touch_selfcap_sensor_get_delta (sensor_id_t sensor_id, touch_delta_t * p_delta)
```

このAPIは与えられた感知部に対応する差値を取得するのに使うことができます。

パラメータ: `sensor_id` - 差値が採られる感知部ID

`p_delta` - 接触ライブラリによって更新される差変数へのポインタ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_get_global_param (touch_global_param_t * p_global_param)
```

```
touch_ret_t touch_selfcap_get_global_param (touch_global_param_t * p_global_param)
```

このAPIは全体パラメータを読み戻すのに使うことができます。

パラメータ: `p_global_param` - 全体感知部構成設定へのポインタ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

11.6.8. 感知部構成設定更新

```
touch_ret_t touch_mutlcap_sensor_update_acq_config ¥  
(touch_mutlcap_acq_param_t *p_touch_mutlcap_acq_param)
```

```
touch_ret_t touch_selfcap_sensor_update_acq_config ¥  
(touch_selfcap_acq_param_t * p_touch_selfcap_acq_param)
```

このAPIは感知部採取パラメータを更新するのに使うことができます。

パラメータ: `p_touch_mutlcap_acq_param` - 相互容量用採取パラメータ
`p_touch_selfcap_acq_param` - 自己容量用採取パラメータ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_sensor_update_config ¥  
(sensor_id_t sensor_id, touch_mutlcap_param_t *p_touch_sensor_param)
```

```
touch_ret_t touch_selfcap_sensor_update_config ¥  
(sensor_id_t sensor_id, touch_selfcap_param_t *p_touch_sensor_param)
```

このAPIは感知部構成設定パラメータを更新するのに使うことができます。

パラメータ: `sensor_id` - 構成設定パラメータが更新されなければならない感知部ID
`p_touch_sensor_param` - 更新のために接触ライブラリによって使われる接触感知部パラメータ構造体

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_update_global_param (touch_global_param_t * p_global_param)
```

```
touch_ret_t touch_selfcap_update_global_param (touch_global_param_t * p_global_param)
```

このAPIは全体パラメータを更新するのに使うことができます。

パラメータ: `p_global_param` - 全体感知部構成設定へのポインタ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

11.6.9. ライブラリの情報と版を取得

```
touch_ret_t touch_mutlcap_get_libinfo (touch_info_t * p_touch_info)
```

```
touch_ret_t touch_selfcap_get_libinfo (touch_info_t * p_touch_info)
```

このAPIは接触ライブラリ情報を取得するのに使うことができます。

パラメータ: `p_touch_info` - 接触ライブラリによって更新される接触情報データ構造体へのポインタ

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_library_get_version_info (touch_libver_info_t * p_touch_libver_info)
```

このAPIは接触ライブラリ版情報を取得するのに使うことができます。

パラメータ: `p_touch_libver_info` - 接触ライブラリによって更新される接触ライブラリ版情報データ構造体へのポインタ

11.6.10. 耐水性API

```
touch_ret_t touch_mutlcap_cnfg_mois_mltchgrp ¥  
(sensor_id_t snsr_id, moisture_grp_t mois_grpid, mltch_grp_t mltch_grpid)
```

```
touch_ret_t touch_selfcap_cnfg_mois_mltchgrp ¥  
(sensor_id_t snsr_id, moisture_grp_t mois_grpid, mltch_grp_t mltch_grpid)
```

このAPIは感知部に対して水分群と複数接触群を割り当てるのに使うことができます。

パラメータ: `snsr_id` - 感知部ID
`mois_grpid` - 水分群ID
`mltch_grp_t` - 複数接触群

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_cnfg_mois_threshold (moisture_grp_t mois_grpid, ¥  
mois_snsr_threshold_t snsr_threshold, mois_system_threshold_t system_threshold)
```

```
touch_ret_t touch_selfcap_cnfg_mois_threshold (moisture_grp_t mois_grpid, ¥  
mois_snsr_threshold_t snsr_threshold, mois_system_threshold_t system_threshold)
```

このAPIは水分群IDに水分感知部閾値と水分システム閾値を割り当てるのに使うことができます。

パラメータ: `mois_grpid` - 水分群ID
`snsr_threshold` - 水分感知部閾値
`system_threshold` - 水分システム閾値

戻り: `touch_ret_t` - 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_mois_tolrnance_enable(void)
```

```
touch_ret_t touch_selfcap_mois_tolrnance_enable(void)
```

このAPIは走行時の間に耐水性調査を許可するのに使われます。

戻り: touch_ret_t – 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_mois_tolrnce_quick_reburst_enable(void)
```

```
touch_ret_t touch_selfcap_mois_tolrnce_quick_reburst_enable(void)
```

このAPIは走行時の間に耐水性迅速再群発機能を許可するのに使われます。

戻り: touch_ret_t – 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_mois_tolrnce_disable(void)
```

```
touch_ret_t touch_selfcap_mois_tolrnce_disable(void)
```

このAPIは走行時の間に耐水性調査を禁止するのに使われます。

戻り: touch_ret_t – 接触ライブラリ異常状態

```
touch_ret_t touch_mutlcap_mois_tolrnce_quick_reburst_disable(void)
```

```
touch_ret_t touch_selfcap_mois_tolrnce_quick_reburst_disable(void)
```

このAPIは走行時の間に耐水性迅速再群発機能を禁止するのに使われます。

戻り: touch_ret_t – 接触ライブラリ異常状態

12. 改訂履歴

文書改訂	日付	注釈
改訂A	2013年10月	初版文書公開
改訂B	2013年9月	4.章の既知の問題で障害情報を更新
改訂C	2013年10月	3.3.4. QDebug接触データデバッグ通信項を包含 3.3.3.項にIAR例プロジェクト用割り込み処理部での注意を包含
改訂D	2014年2月	QTouchライブラリとQTouch構成器(Composer)5.3に関連する文章に渡って全体更新
改訂E	2014年11月	雑音計数器測定に関して5.2.6.と5.2.7.の項を包含 容量接触技術の概要に関して3.章を包含
改訂F	2015年2月	低電力と一括動作の支援に関する関連情報を包含
改訂G	2015年4月	2.章を更新 – 支援されるデバイス変種とデバイス多重化任意選択での情報を包含
改訂H	2015年6月	2.章を改訂 – 支援されるデバイス変種とデバイス多重化任意選択での情報を包含 7.2.項を更新 – コードとデータのメモリ考察 5.2.1.項を更新 – ピン、チャネル、感知部のパラメータ
改訂I	2015年9月	電荷共用遅延を包含 5.2.8.と5.2.10.項を更新 – 迅速再群発用ライブラリ パラメータと水分パラメータを追加 11.6.8.項を更新 – 水分APIを追加 8.章を更新 – 例プロジェクトを更新
改訂J	2016年1月	以下の新しい項を包含 ・ 補償回路 ・ QDebugデータの流にATmel ICEを使用 ・ megaAVR用応用の流れ 最新拡張発売に対する参照で1.、5.、8.の章を更新
改訂K	2016年2月	ATmega324PB支援を追加 最新拡張発売に対する参照で1.、5.、8.の章を更新
改訂L	2016年4月	最新拡張発売に対する参照で1.、5.、8.の章を更新
改訂M	2016年7月	1.章で最新ソフトウェア版番号を更新 9.章で新しい障害を追加



Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA TEL:(+1)(408) 441-0311 FAX: (+1)(408) 436-4200 | www.atmel.com

© 2016 Atmel Corporation. / 改訂:Atmel-42195M-Peripheral-Touch-Controller_User Guide-07/2016

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, QTouch®, AKS®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。他の用語と製品名は一般的に他の商標です。ARM®とCortex®はARM Ltdの登録商標です。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。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2021.

本応用記述はAtmelのQTouchライブラリ周辺機能接触制御器使用者の手引き(Rev.42195M-07/2016)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。