MICROCHIP

AN1478

mTouch™ センシング ソリューションのアクイジション手法：静電容量式分圧器

著者： Burke Davison
Microchip Technology Inc.

はじめに

静電容量式センサとは、一定面積の導体材料を (場合によっては直列抵抗を介して)PIC® MCU のピンに接続したものです。センサ周辺の環境が変化すると、グランドに対して導体材料の静電容量が変化します。ピン静電容量の計測には多数の方法がありますが、そのほとんどはノイズの少ない信号を得るための専用ハードウェアまたは高度なデジタル フィルタ処理システムを必要とします。

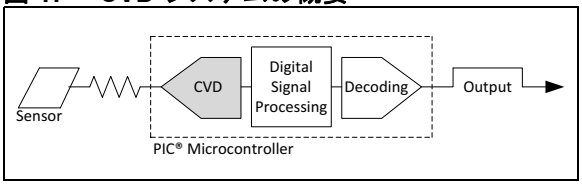
マイクロチップ社が開発した差動静電容量方式分圧回路 (CVD) は、ADC (A/D コンバータ) と最低限のデジタル処理しか必要としないアクイジション手法です。CVD は非常に多くのデバイスに実装できます。

本書は、mTouch™ センシング ソリューション CVD センシング手法を実装する方法を解説し、環境の変化に対する信号のふるまいを分析し、感度を向上し出力ノイズを低減する方法を取り上げます。

本書に掲載したサンプルコードは説明のみを目的としたものです。信頼性の高いノイズ除去を必要とする実用アプリケーションでは、マイクロチップ アプリケーション ライブラリ (MLA、http://www.microchip.com/mla) で提供している mTouch センシング フレームワークおよびライブラリの使用を強く推奨します。

PIC MCU で静電容量計測を行う方法は CVD だけではありません。マイクロチップ社のウェブサイト www.microchip.com/mTouch では、CTMU(充電時間計測ユニット)によるセンシング手法に関するアプリケーション ノート (AN1250 - マイクロチップ社製 CTMU の静電容量式タッチ アプリケーションへの適用) とハードウェアとソフトウェアの設計ガイドライン (AN1334 - 堅牢なタッチセンシングの設計手法) を掲載しています。

図 1: CVD システムの概要



静電容量式タッチセンシングの基本概念

通常、静電容量式センサはプリント基板に金属を貼って作成しますが、アルミ箔で簡単に作る事もできます。この導電性パッドをトレースで PIC MCU に接続します。また、必要な場合には直列抵抗も接続します。図 1 に示すように、PIC MCU はこのパッドの静電容量を常時ポーリングし、大きな変化がないか監視します。センサはスキャンの計測段階では高インピーダンスで、その他の状態では常に低インピーダンスです。

「大きな変化」の定義はノイズレベルによって異なります。最悪条件のノイズレベルとはっきり区別できる大きさの変化が必要です。この「大きな変化」は、周囲環境が短時間で変化する場合、その環境を原因とする変化量を大きく上回る必要があります。従って、総ノイズ量は考えられる各種ノイズ源からもたらされる高周波と低周波の外乱を組み合わせたものです。

このため、静電容量式センサの信号品質は式 1 で定義されるように常に信号 / ノイズ比 (SNR) によって定義する必要があり、単にアクティブ時の信号変化に基づいて判断する事はできません。

式 1: 信号 / ノイズ比

$$SNR = \frac{\mu}{\sigma}$$

μ：アクティブ時の変化量
σ：ノイズの標準偏差

本書では、CVD 計測手法により、センサの静電容量をデジタル処理できるように整数値に変換する方法を解説します。**ただし、マイクロチップ社は手作業による CVD の実装を推奨していません。**マイクロチップ社がアプリケーション ライブラリで提供している mTouch センシング フレームワークと mTouch センシング ライブラリは、スキャンを自動的に実装しています。また、高いノイズ耐性を持つ事も実証済みです。ゼロからスキャンを実装するよりも、これらのリソースを使う事を強く推奨します。

静電容量式分圧器

CVD は、ADC (A/D コンバータ) モジュールだけを使ってピンの相対静電容量を計測する、電荷 / 電圧に基づいた手法です。ほとんど全ての PIC MCU は ADC を内蔵しており、CVD はそれら全てに実装できます。

この手法では、内部 ADC サンプルホールド静電容量の大きさから相対静電容量を計測します。このコンデンサの静電容量の代表値は PIC MCU の電気的仕様で定義されています。しかし、製造公差により最大 20% のばらつきがあります。そのため、校正を実行し、環境条件が変化しない事を保証できない限り、CVD によって絶対静電容量を計測する事を推奨しません。タッチおよび近接アプリケーションで必要なのは相対計測のみです。そのため、環境変化による変動分をフィルタ処理で取り除けば校正が不要です。

CVD によるセンシングの利点

CVD が実用アプリケーションで良好に動作する理由は複数あります。以下の特性は、タッチの最終判断の信頼性を高め、静電容量式タッチを組み込むコストを低減します。

- 温度依存性が小さい

  -20 ～ +60 ℃での信号変化量は 1 ～ 3% (typ.) です。通常この変化はソフトウェアで取り除きます。

- VDD の影響が小さい

  CVD の波形は VDD に大きく影響されません。センサの充電にもADCの正参照電圧にもVDDを使っているためです。従って、VDD の低周波変動による影響は極小です。しかし、VDD の高周波外乱は有害な信号ノイズを発生する場合があります。

- ハードウェア要件が最小限である

  必要に応じて直列抵抗を挿入し、信号の高周波ノイズを低減する事を推奨します。ノイズの問題がない場合、外付け部品は不要です。

- 低周波ノイズが除去される

  低周波ノイズによる影響は、CVD 波形の 2 つの ADC サンプルに対して同じ方向に生じます。一方、静電容量変化の影響は 2 つのサンプルで逆方向に生じます。従って、2 つのサンプルを減算すれば信号を 2 倍にすると同時にノイズの影響を除去できます。

Note:	除去できる周波数レンジは、2 つのサンプル間の時間差で決まります。時間差が小さいほど、ノイズ除去できる帯域幅が大きくなります。

全ての静電容量式センシング手法がこれらの利点を備えているわけではないため、CVD はタッチ アプリケーションにおける優れた選択肢として抜きん出た存在です。

動作原理

センサの SNR にとってタイミングが重要であるため、CVD を手作業で実装する場合の言語として推奨できるのはアセンブリだけです。マイクロチップ社はマイクロチップ アプリケーション ライブラリでスキャンの実装用ライブラリを提供しています。スキャンをゼロから実装するのではなく、このライブラリを使う事を強く推奨します。このライブラリは、弊社のウェブサイト www.microchip.com/mla より入手できます。本書でも、複数の波形構成に対するアセンブリのサンプルコードを掲載しています。

静電容量式センサは PIC MCU のアナログピンに接続します。必要に応じて回路内に直列抵抗を挿入してローパスフィルタを構成し、信号の高周波ノイズを減衰できます。続いて入出力ポートと ADC のみを使ってサンプリングを行います。

ステップ 1: コンデンサのプリチャージ

2 つのコンデンサを逆の電圧に充電します。充電を 1 回目に実行する方を「サンプル A」とします。2 回目 (ステップ 4 ～ 6 で説明) に実行する方を「サンプル B」とします。図 4 にこれを示します。

サンプル A:

- 外部センサを VSS に放電
- 内部センサを VDD に充電

サンプル B:

- 外部センサを VDD に充電
- 内部センサを VSS に放電

ステップ 2: コンデンサの接続とセトリング

2 つのコンデンサを並列に接続して、電荷を定常状態に移行 (セトリング) させます。外部静電容量が増えれば、初期電荷も増加します (式 2)。内部静電容量は変化しないため、その電荷は一定です。図 5 にこのステップを示します。

ステップ 3: A/D 変換

C_{hold} の最終電圧は、内部静電容量と外部静電容量の比で決まります (式 3)。

ステップ 4 ～ 6: プリチャージ電圧の反転と反復

上記の操作を再度実行します。ただし、今回はプリチャージ電圧を反転します。

2 つの結果の差を現在のセンサ読み値として使います。このスキャン手法が一般的に「差動 CVD」と呼ばれるのは、この理由からです。図 2 に、差動 CVD センシング手法の全波形を示します。

ユーザが静電容量式センサに接近すると、内部静電容量に対して外部静電容量の大きさが増加します。これによって 2 つのサンプルのセトリング電圧が変化します。

サンプル A については、外部センサの静電容量が増加して VSS に放電されると最終セトリング電圧は低下します。サンプル B については、外部センサの静電容量が増加して VDD に充電されると最終セトリング電圧は上昇します。従って、外部静電容量が増加すると CVD 波形の 2 つのセトリング電圧の差が広がり、センサ読み値が変化します。

ノイズは、その信号の位相に応じてセトリング電圧を変動させます。低周波ノイズの場合、位相は両方のサンプルでほぼ同じです。ノイズの影響が両サンプルでほぼ同じであるため、2 つのセトリング電圧間の差分を取り、ノイズの影響を相殺する事で信号に乗ったノイズを大方取り除く事ができます。ノイズの周波数が高くなるほど、この方法ではノイズを取り除く事ができないため、別の方法を使う必要があります。2 つのサンプル間の時間差を小さくすると、高周波の除去能力が改善します。これが、ノイズ耐性という点で C よりアセンブリが優れている理由の 1 つです。

図 2: 差動 CVD の波形

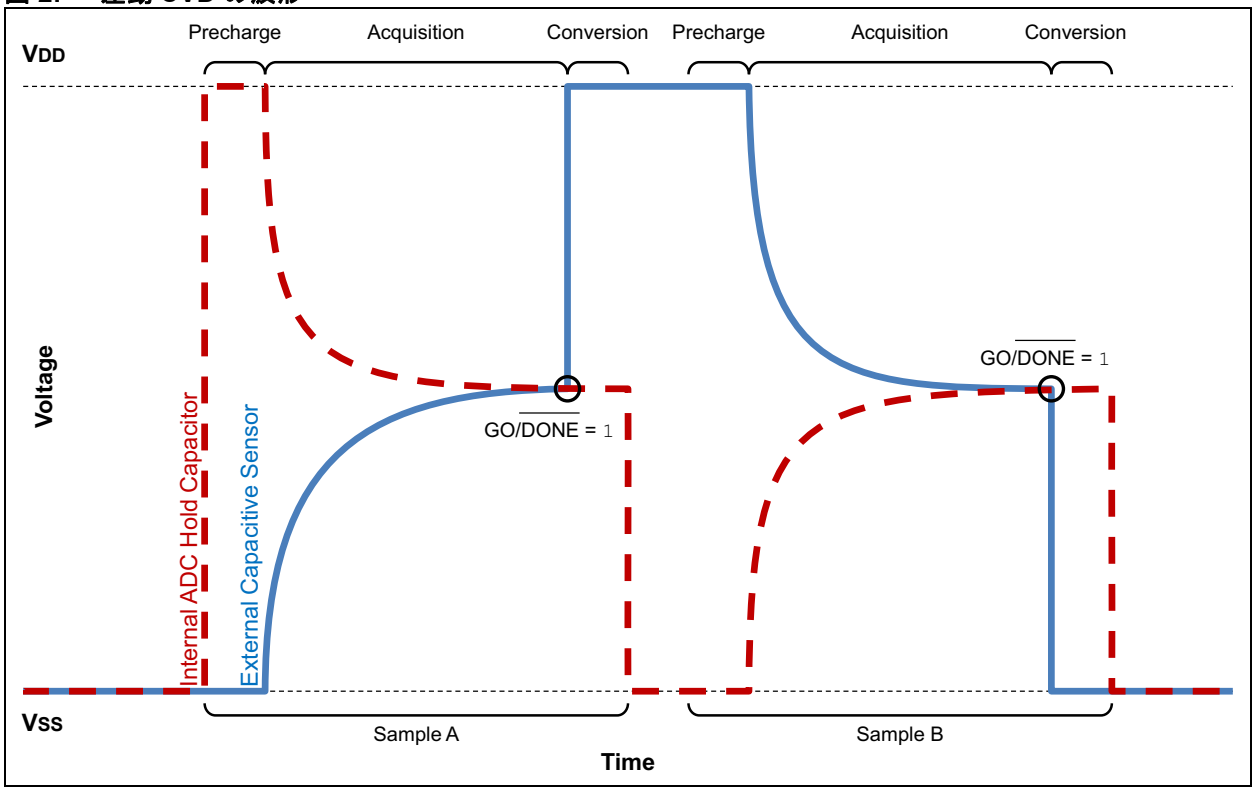
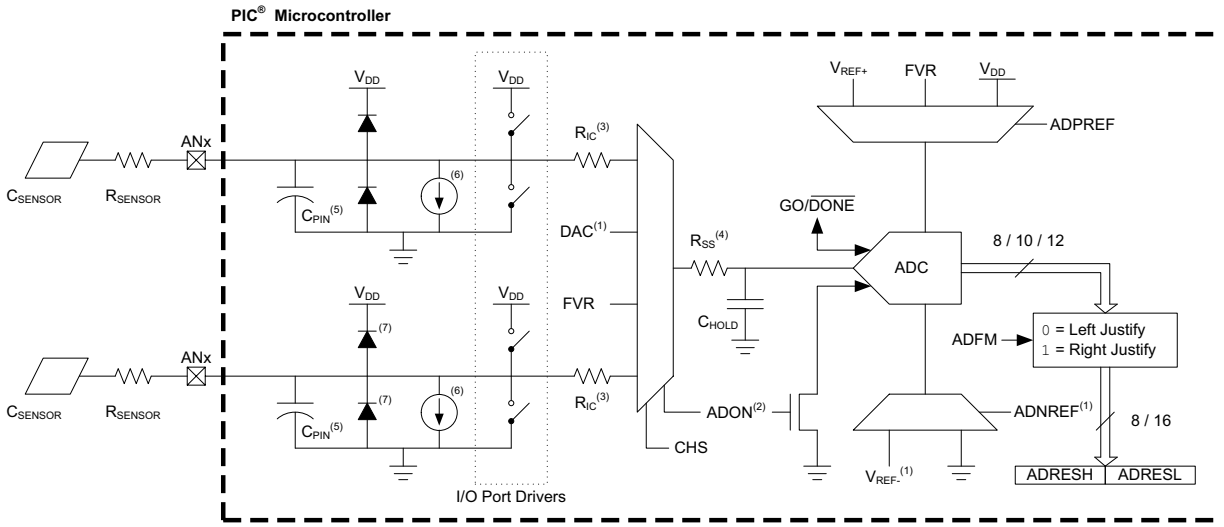


図 3 に、PIC MCU を使った静電容量式タッチ設計の一般的なシステム図を示します。

図 3:　PIC® MCU を使った静電容量式タッチ設計の一般的なシステム図



Note 1: 一部の PIC® マイクロコントローラでは使えません。
2: ADON = 0 の場合、マルチプレクサの入力は全て切り離されます。
3: ピンの相互接続抵抗です。RIC ≤ 1kΩ
4: サンプリング スイッチ抵抗です。
5: 入力静電容量です。 CPIN ≈ 5pF
6: ILEAKAGE:: 各マイクロコントローラのデータシート内の「電気的特性」を 参照してください。
7: VT ≈ 0.6V

ステップごとの分析

プリチャージ段階

式 2 で定義されるように、2 つのコンデンサが逆方向に充電されます。内部 ADC の静電容量は、未使用アナログピンのドライバ、他のセンサピン、DAC (D/A コンバータ、ADC チャンネルとして選択できる場合) のどれかによって充電できます。

式 2:　プリチャージ段階

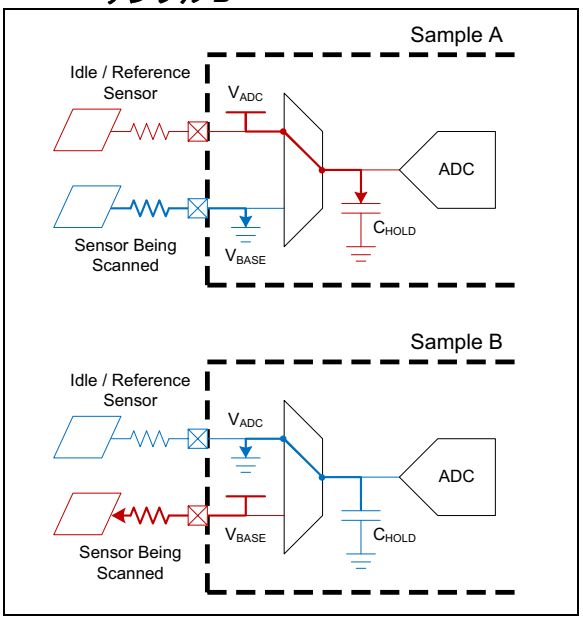
$$Q_{base} = C_{base}V_{base}$$
$$Q_{hold} = C_{hold}V_{hold}$$

Q_{base} : 外部の総電荷
Q_{hold} : 内部の総電荷
V_{base} : プリチャージ段階で外部センサに供給される電圧
V_{hold} : プリチャージ段階で内部 ADC ホールド静電容量に供給される電圧
C_{base} : 「解放」状態におけるベース外部静電容量
C_{hold} : センサの内部 ADC 静電容量

図 4 に示すように、V_{base} と VADC の値は波形のサンプル A とサンプル B で互いに入れ代わります。どちらの場合も、一方の電圧値は VSS、もう一方の電圧は VDD です。

Note: 通常は VDD が一定と見なされるため、以下に示す計算の大部分は VDD で割る事によってこの項を消去しています。これは、VDD =1 V であると見なす事と等価です。

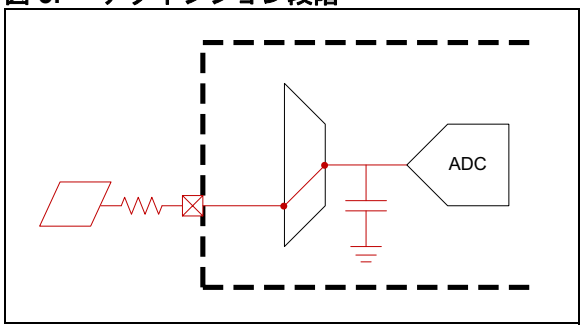
図 4:　プリチャージ段階、サンプル A とサンプル B



アクイジション段階

2 つのコンデンサを逆の電圧状態に充電した上で、図 5 に示すように両者を並列に接続します。これによって、式 3 に定義される通り、両方のコンデンサ間で電荷が合計されます。コンデンサ両端の電圧は、外部静電容量に対する C_{hold} の関係で決まる中間値にセトリングして等しくなります。

図 5:　アクイジション段階



コンデンサが並列に接続されたため、両者の値を組み合わせて回路の総静電容量を求める事ができます。

$$C_{total} = C_{hold} // C_{base} = C_{hold} + C_{base}$$

コンデンサ間の総電荷は個々の電荷の合計です。

$$Q_{total} = C_{hold}V_{hold} + C_{base}V_{base}$$

式 3:　アクイジション段階

$$V_{settle} = \frac{C_{hold}V_{hold} + C_{base}V_{base}}{C_{hold} + C_{base}}$$

Q_{total} : アクイジション段階で接続された場合の両コンデンサの総電荷

C_{total} : アクイジション段階で内部および外部コンデンサの両方を接続し電荷を共有した場合の回路の総静電容量

V_{settle} : 通常の CVD 波形のアクイジション段階における最終セトリング電圧

差動結果

セトリング電圧を表す上の式は、1 回目の差動サンプルと 2 回目の差動サンプルの両方にあてはまる汎用式です。1 回目のサンプル (A) に対する実際のセトリング電圧は、V_{base} に 0、V_{ADC} に V_{DD} を代入する事で求められます。2 回目のサンプル (B) に対する実際のセトリング電圧は、V_{base} に V_{DD}、V_{ADC} に 0 を代入する事で求められます。

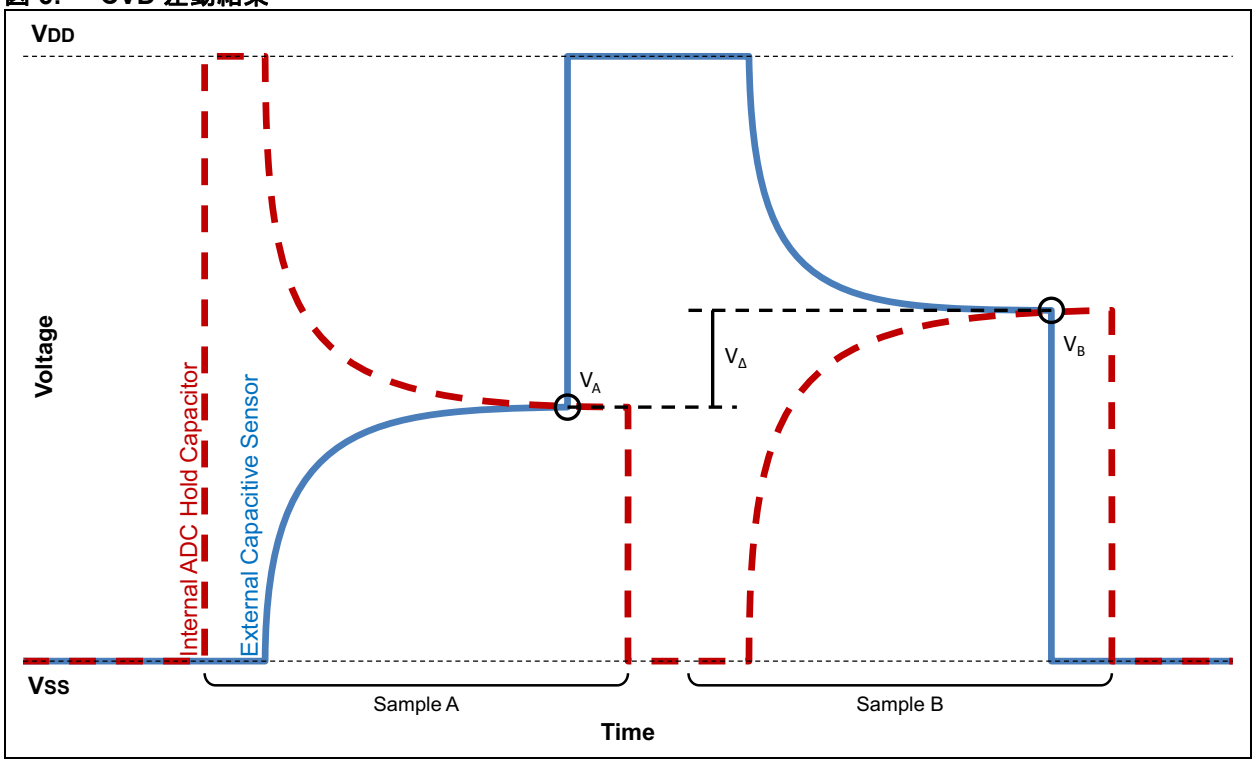
$$V_A = V_{settle}\{V_{hold} = V_{DD}, V_{base} = 0\}$$

$$V_B = V_{settle}\{V_{hold} = 0, V_{base} = V_{DD}\}$$

次に、2 つの電圧の差を求める事でセンサの読み値を計算します。実際には、V_A よりも V_B の値の方が大きい事が多いため、減算の順序を変えて正の値が得られるようにします。

$$V_\Delta = V_B - V_A$$

図 6:　CVD 差動結果



Note: 計算結果を確実に正の値とするため、ソフトウェアによって V_B の値を 2^N 分オフセットする事を推奨します (N は ADC のビット数)。計算を簡略化するため、このオフセットは無視しています。

$$V_{\Delta released} = V_{settle}(V_{hold} = 0, V_{base} = V_{DD}) - V_{settle}(V_{hold} = V_{DD}, V_{base} = 0)$$

$$\frac{V_{\Delta released}}{V_{DD}} = \frac{C_{base}}{C_{hold} + C_{base}} + \frac{C_{hold}}{C_{hold} + C_{base}}$$

式 4:　差動結果

2 つの CVD セトリング電圧間の差

$$\frac{V_{\Delta released}}{V_{DD}} = \frac{C_{base} - C_{hold}}{C_{hold} + C_{base}}$$

指の静電容量の追加

今回も同じ分析を繰り返しますが、回路に新たなコンデンサが加わります。ユーザの指です。ユーザの指の静電容量は、外部静電容量と総静電容量を変化させます。

$$C_{external,pressed} = C_{base} + C_{finger}$$

$$C_{total,pressed} = C_{hold} + C_{base} + C_{finger}$$

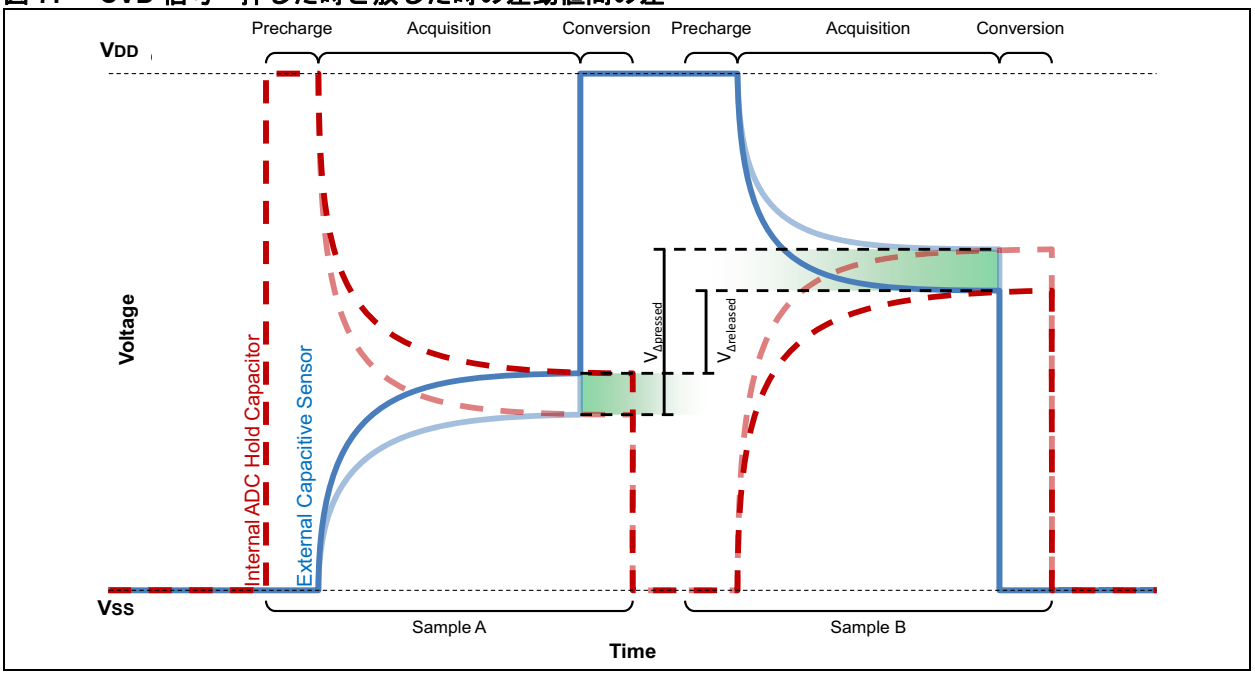
センサ上に指がある時の CVD セトリング電圧を以下の汎用式で求めます。

$$V_{settle,pressed} = \frac{(C_{base} + C_{finger})V_{base} + C_{hold}V_{hold}}{C_{hold} + C_{base} + C_{finger}}$$

V_Δ の式を使い、センサ上に指がある時の CVD 波形のセトリング電圧の差を求めます。

$$\frac{V_{\Delta pressed}}{V_{DD}} = \frac{(C_{base} + C_{finger}) - C_{hold}}{C_{hold} + C_{base} + C_{finger}}$$

図 7:　CVD 信号 - 押した時と放した時の差動値間の差



最後に、指で押した時の差から押していない時の差を引く事で、CVD の総信号を求めます。式 5 は、回路に指が加わった事によるセンサ読み値の変化量です。この値ができるだけ大きくなるように設計します。

$$CVD = V_{\Delta pressed} - V_{\Delta released}$$

$$\frac{CVD}{V_{DD}} = \frac{(C_{base} + C_{finger}) - C_{ADC}}{C_{ADC} + C_{base} + C_{finger}} - \frac{C_{base} - C_{ADC}}{C_{ADC} + C_{base}}$$

式 5:　CVD 信号

$$\frac{CVD}{V_{DD}} = \frac{2C_{ADC}C_{finger}}{(C_{ADC} + C_{base})(C_{ADC} + C_{base} + C_{finger})}$$

タイミングに関する注意事項

プリチャージの遅延

定義 : 内部および外部コンデンサの充電に要する時間

この遅延は、システムのノイズ性能に大きな影響を与えません。ただし、CVD の実装時に、内部のホールドコンデンサの参照電圧源として他のセンサを使い、かつセンサまたはその参照電圧源のどちらかが大きな時定数を持つ場合、既定値の遅延では、プリチャージ段階が終了するまでに、外部参照電圧源に V_{DD} まで完全に充電するための十分な時間が与えられない可能性があります。

両方のコンデンサが完全に充電されない場合、センサの信号は不正確です。アクイジション段階に移行する前に、回路に既知の電荷量が与えられず、電荷は静電容量によって変化します。これでは CVD スキャンを実行できません。回避する必要があります。全サンプルで完全に充電される事が確実となるよう、プリチャージ遅延を延ばす必要があります。

オシロスコープで見ると、非常に激しくタッチしているような波形が観察されます。アクイジション段階に移行する前にセンサが完全に充電されない状況がオシロスコープで確認された場合、プリチャージ時間を延長します。

本書に掲載されているサンプルコードでプリチャージ時間を延ばすには、コメント文「Optional additional and/or variable delay」より後ろ、アクイジション段階より前の行に `NOP` を追加します。

その他のコードライブラリまたはフレームワークでプリチャージ時間を延ばすには、波形の詳細設定部分を探し、その中からプリチャージ設定を見つけます (C_{hold} - 充電遅延とも呼ぶ)。

アクイジション / セトリングの遅延

定義: コンデンサが逆状態にプリチャージされた後に、中間電圧に等化されるまでに要する時間

A/D 変換が始まるまでの CVD 波形のアクイジション段階の全期間で、センサは入力に設定されます(TRIS = `1`)。これはセンサが高インピーダンスである事を意味します。センサの近くにある低インピーダンス源はセンサをさらに放電または充電して、セトリング済みの電荷に影響を与える可能性があります。つまり、CVD 手法がノイズの影響を受けやすい期間であるという事です。従って、この段階にとどまる時間はできるだけ短くする必要があります。

このセトリング遅延時間を決定する際は、トレードオフを検討する必要があります。セトリング遅延が短すぎると、2 つのコンデンサ間で等化される値まで電荷を完全にセトリングできません。その場合、回路に外部静電容量が追加された時の感度が低下します。しかし、セトリング遅延が長すぎると、センサへのノイズカップリングが生じ、最終電圧を不正確にします。通常は、完全にセトリングした場合の感度の 90 ～ 95% 以上が得られる範囲で最小の値に設定します。アプリケーションによって、より長いセトリング遅延 (例 : 大きな外部静電容量 / 抵抗が存在する) または短いセトリング遅延 (例 : 回路内のノイズが既知である) が必要な場合があります。

差動遅延

定義 : サンプル A とサンプル B の時間差

波形の低周波ノイズを最大限除去するために、差動遅延はできるだけ小さくする必要があります。

2 つのサンプルの時間差をわずかに無作為化すると、低 kHz レンジでノイズ減衰の効果が得られます (これは必須ではありません)。

サンプリング遅延

定義 : サンプル A と次のサンプル A の間の時間、つまり CVD 波形間の時間

センサのサンプリング レートの高調波に相当する周波数のノイズを減衰するにはサンプリング遅延を無作為化します。

2 つのアクイジション段階を用いた CVD

通常の CVD 波形に対する理想的なセトリング電圧は、サンプル A とサンプル B のどちらも 1/2*V_{DD} です。この時セトリング電圧と電圧レール間の分離が最大であり、ノイズによるクリッピングに対する耐性が最大となります。セトリング電圧を 1/2*V_{DD} に近づけるには、内部と外部のコンデンサの値をほぼ等しくします。

内部コンデンサが外部コンデンサよりもはるかに大きいまたはその逆の場合、セトリング電圧は大きい方のコンデンサの開始値に比例して影響を受けます。この場合感度は低くなります。

この問題を解決するために、アクイジション段階を 2 回連続して実行し、最終電圧を強制的に 1/2*V_{DD} に近づけます。

外部コンデンサが内部コンデンサよりもはるかに大きい場合 :

- 通常の CVD プリチャージおよびアクイジション段階を実行します。(式 3)
- 外部コンデンサの電圧を維持し、同時に内部コンデンサを再充電します。
- 再度、アクイジション段階を実行します。(式 6)

図 8 に示す通り、センサの波形をオシロスコープで観測すると、1 回目のアクイジション段階で得られる初期セトリング電圧が上記のプロセスによって倍増している事が分かります。ここから、この波形は「ダブル CVD 波形」と呼ばれます。

$$V_{total}^{double} = C_{base}V_{settle}^{normal} + C_{hold}V_{hold}$$

$$V_{total}^{double} = \frac{C_{base}V_{settle}^{normal} + C_{hold}V_{hold}}{C_{base} + C_{hold}}$$

図 8:　ダブル CVD 波形

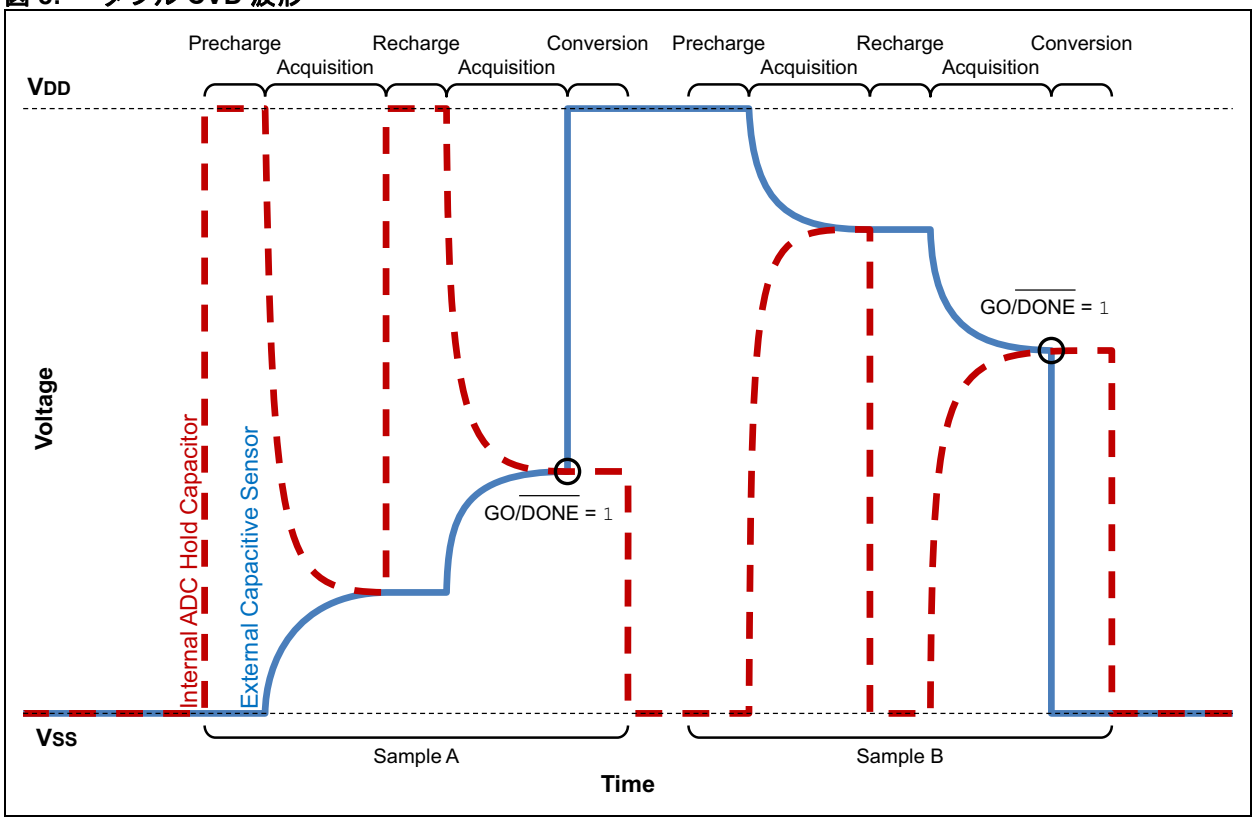
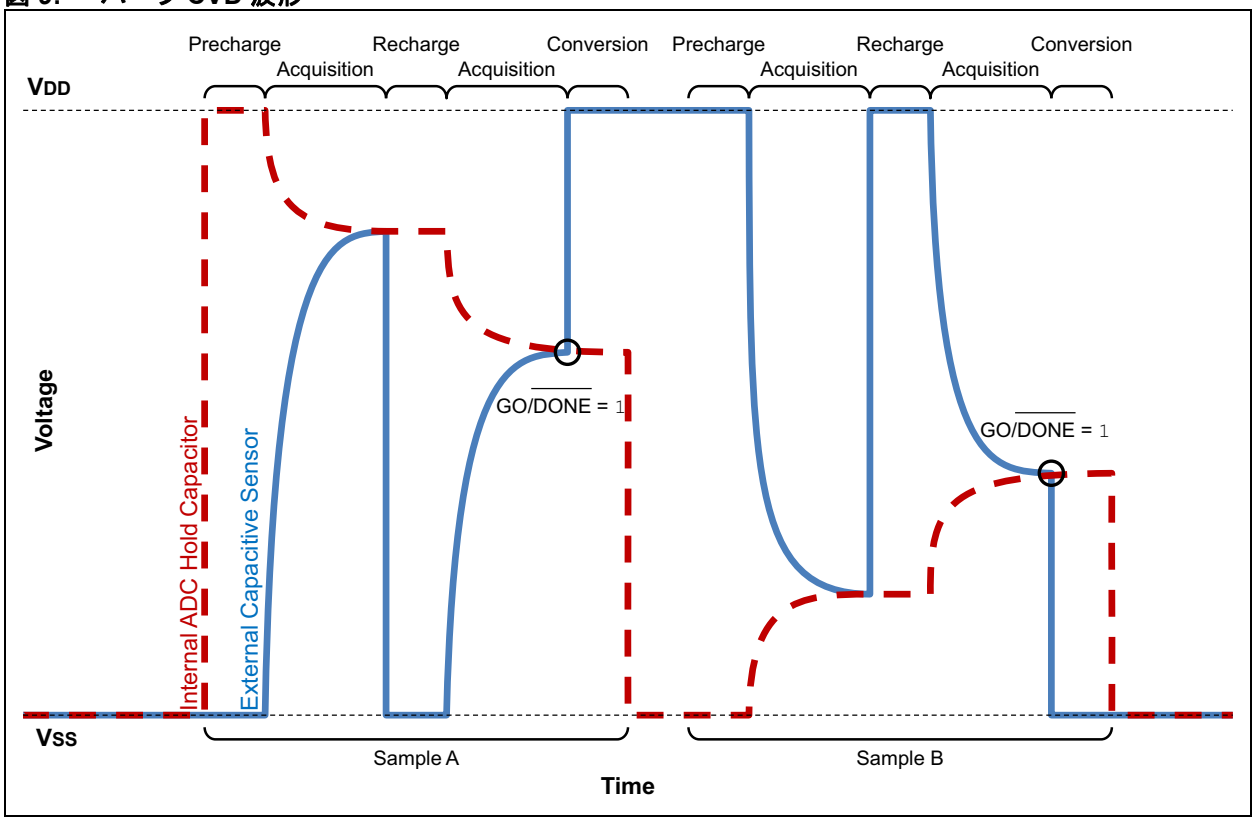


図 9:　ハーフ CVD 波形



内部コンデンサが外部コンデンサよりもはるかに大きい場合 :

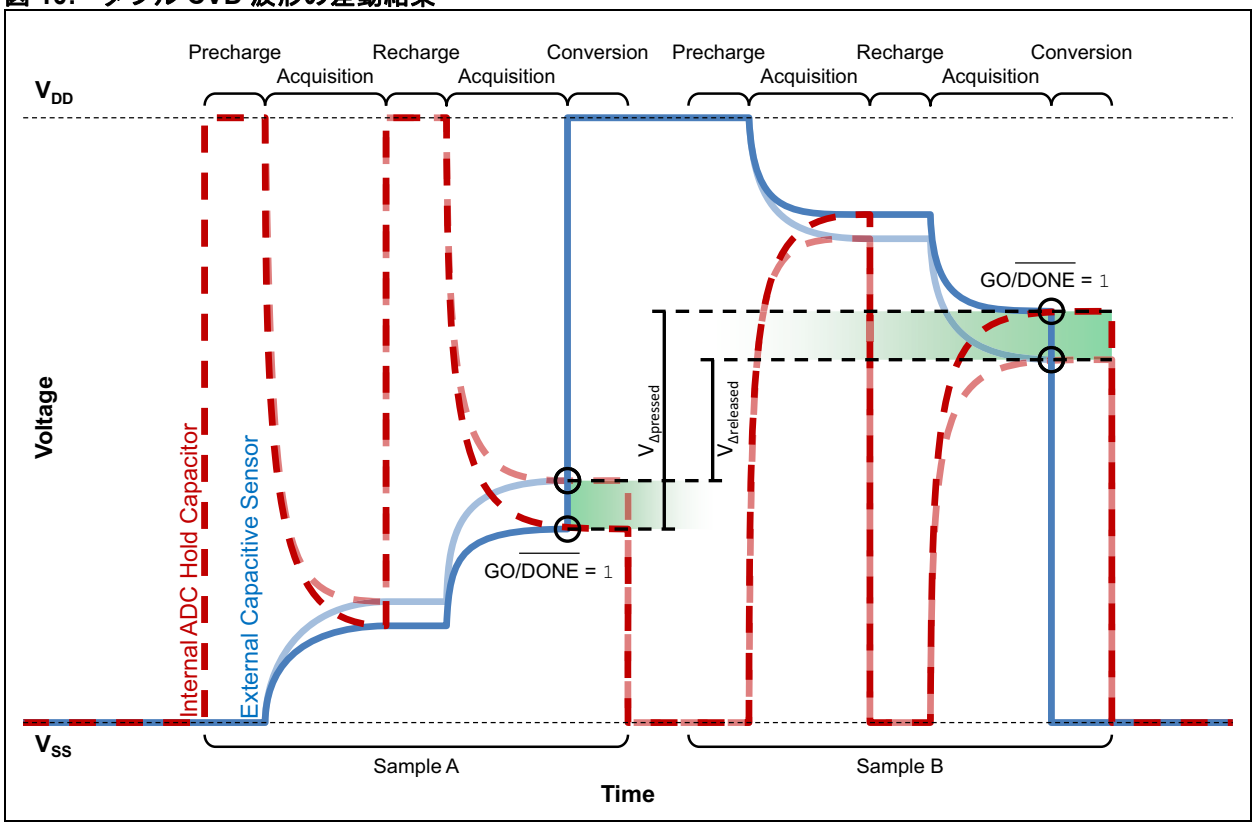
- 通常の CVD プリチャージおよびアクイジション段階を実行します。(式 3)
- 内部コンデンサの電圧を保持し、同時に外部コンデンサを再充電します。
- 再度、アクイジション段階を実行します。(式 7)

図 9 に示す通り、センサの波形をオシロスコープで観測すると、1 回目のアクイジション段階で得られる初期セトリング電圧が上記のプロセスによって半減している事が分かります。ここから、この波形は「ハーフ CVD 波形」と呼ばれます。

$$Q_{total}^{half} = C_{base}V_{base} + C_{hold}V_{settle}^{normal}$$

$$V_{total}^{half} = \frac{C_{base}V_{sensor} + C_{hold}V_{settle}^{normal}}{C_{base} + C_{hold}}$$

図 10:　ダブル CVD 波形の差動結果



「通常」のセトリング電圧に式 3 を代入して単純化すると、ダブルおよびハーフ CVD 波形の最終セトリング電圧は次式で求められます。

式 6:　ダブル CVD 波形のセトリング電圧

$$V_{total}^{double} = \frac{C_{hold}^2 (V_{base} - V_{hold})}{(C_{base} + C_{hold})^2} + \frac{2C_{hold}(V_{hold} - V_{base})}{C_{base} + C_{hold}} + V_{base}$$

V_{total}^{double} : ダブル CVD 波形の最終セトリング電圧

式 7:　ハーフ CVD 波形のセトリング電圧

$$V_{total}^{half} = \frac{C_{hold}^2 (V_{hold} - V_{base})}{(C_{base} + C_{hold})^2} + V_{base}$$

V_{settle}^{half} : ハーフ CVD 波形の最終セトリング電圧

2 段階アクイジションの差動結果

セトリング電圧を表す上の式は、1 回目の差動サンプルと 2 回目の差動サンプルの両方にあてはまる汎用式です。1 回目のサンプル（「A」）に対する実際のセトリング電圧は、V_{base} に 0、VADC に VDD を代入する事で求められます。2 回目のサンプル（「B」）に対する実際のセトリング電圧は、V_{base} に VDD、VADC に 0 を代入する事で求められます。

$$V_A = V_{settle}\{V_{hold} = VDD, V_{base} = 0\}$$

$$V_B = V_{settle}\{V_{hold} = 0, V_{base} = VDD\}$$

次に、2 つの電圧の差を求める事で、センサの読み値を計算します。実際には、多くの場合 V_A よりも V_B の値の方が大きいため、減算の順序を変えて正の値が得られるようにします。図 10 にこの計算を示します。

$$V_\Delta = V_B - V_A$$

式 8:　2つのダブルCVDセトリング電圧間の差

$$\frac{V_{\Delta released}^{double}}{VDD} = \frac{C_{base}^2 - 2C_{hold}C_{base} - C_{hold}^2}{(C_{base} + C_{hold})^2}$$

式 9:　2つのハーフCVDセトリング電圧間の差

$$\frac{V_{\Delta released}^{half}}{VDD} = \frac{C_{base}^2 + 2C_{hold}C_{base} - C_{hold}^2}{(C_{base} + C_{hold})^2}$$

指の静電容量の追加

今回も同じ分析を繰り返しますが、回路に新たなコンデンサが加わります。ユーザの指です。ユーザの指の静電容量は、外部静電容量と総静電容量を変化させます。

$$C_{external,pressed} = C_{base} + C_{finger}$$

$$C_{total,pressed} = C_{hold} + C_{base} + C_{finger}$$

押した時の 2 つのダブル CVD セトリング電圧間の差：

$$\frac{V_{\Delta pressed}^{double}}{VDD} = \frac{(C_{base} + C_{finger})^2 - 2C_{hold}(C_{base} + C_{finger}) - C_{hold}^2}{(C_{hold} + C_{base} + C_{finger})^2}$$

押した時の 2 つのハーフ CVD セトリング電圧間の差：

$$\frac{V_{\Delta pressed}^{half}}{VDD} = \frac{(C_{base} + C_{finger})^2 + 2C_{hold}(C_{base} + C_{finger}) - C_{hold}^2}{(C_{hold} + C_{base} + C_{finger})^2}$$

最後に、両タイプについて、押した時の差動値から押していない時の差動値を引く事で、CVD の総信号を求めます。これが、回路に指が加わった事によるセンサ読み値の変化量です。式 10 と式 11 は、できるだけ大きくなるように設計すべき値を含んでいます。

式 10:　押した時の差動値から放した時の差動値を引いた差（差動値はどちらもダブル CVD スキャン手法による）

$$\frac{CVD_{double}}{VDD} = \frac{2C_{hold}C_{finger}[2C_{base}(C_{hold} + C_{base} + C_{finger}) + C_{hold}C_{finger}]}{(C_{base} + C_{hold})^2(C_{finger} + C_{base} + C_{hold})^2}$$

式 11:　押した時の差動値から放した時の差動値を引いた差（差動値はどちらもハーフ CVD スキャン手法による）

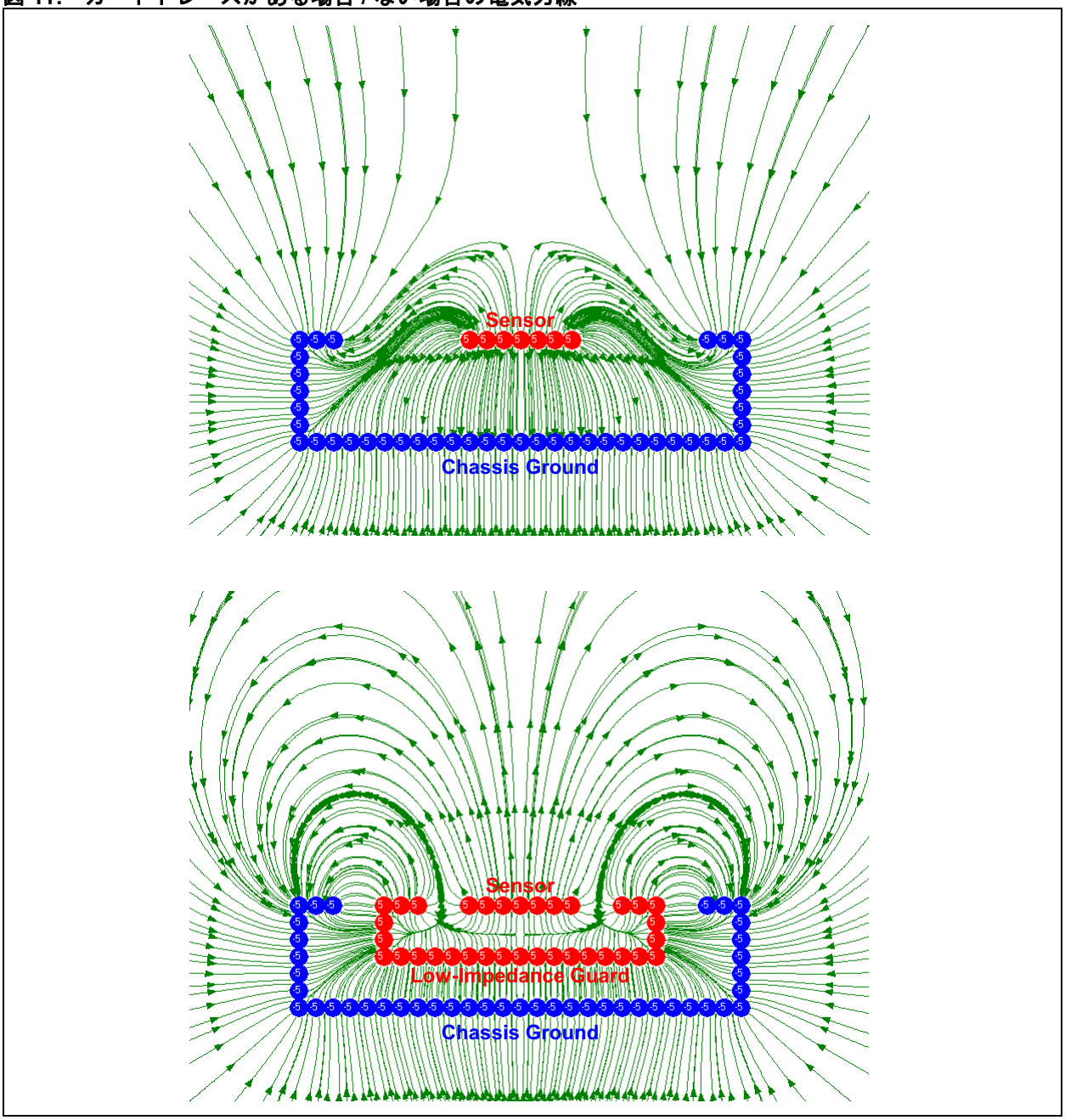
$$\frac{CVD_{half}}{VDD} = \frac{2C_{hold}^2C_{finger}(2C_{base} + 2C_{hold} + C_{finger})}{(C_{base} + C_{hold})^2(C_{finger} + C_{base} + C_{hold})^2}$$

アクティブガード付き CVD スキャン

CVD スキャンは総静電容量を計測します。従って、センサのベース静電容量が減少すると、ユーザの指による信号変化は増加します。アクティブガードは、センサと周囲の間の電位差を低減する事でベース静電容量を小さくする方法です。

最適な方法は、センサとそのトレースの周囲を完全に囲う事です。それが不可能な場合、センサの近くに低インピーダンス源がある全ての場所で細心の注意を払います。通信線、モータ駆動線、グランドプレーン、電源プレーンは、静電容量式センサから遠ざける必要があるトレースの例です。ガードトレースは、センサとこれらのトレースの間に配置します。ただし、ユーザとセンサの間にはガードを配置できません。ユーザの指による静電容量の変化をガードが遮断してしまうためです。

図 11:　ガードトレースがある場合 / ない場合の電気力線



ガードトレースの設計ガイドライン

以下にガードを設計する際に注意すべきガイドラインを示します。

1 つのガードトレースを全てのセンサに使えます。センサは 1 つずつスキャンされるため、他のセンサには影響を与えずに、現在スキャン中のセンサに対するガードのみをアクティブに駆動する事ができます。

電源プレーンまたは低インピーダンスのトレースは全てセンサからガードします。

センサパッド周囲のガードトレースは厚さを 1 mm とし、センサから 2 ～ 3 mm 離します。

PIC MCU ピンに戻るセンサのトレースに沿ったガードトレースは、センサのトレースと同じ厚さ (0.1 ～ 0.3 mm) にできます。センサトレースからガードトレースまでの距離は 0.5 mm であれば十分です。

ガードの標準的手法として、センサ波形をユニティゲイン アンプでバッファし、周囲のトレースに信号を乗せる方法があります。ユニティゲイン アンプによる方法はコストがかかります。しかし、ガード信号の駆動には外付け部品を必要としない方法が 2 つあります。

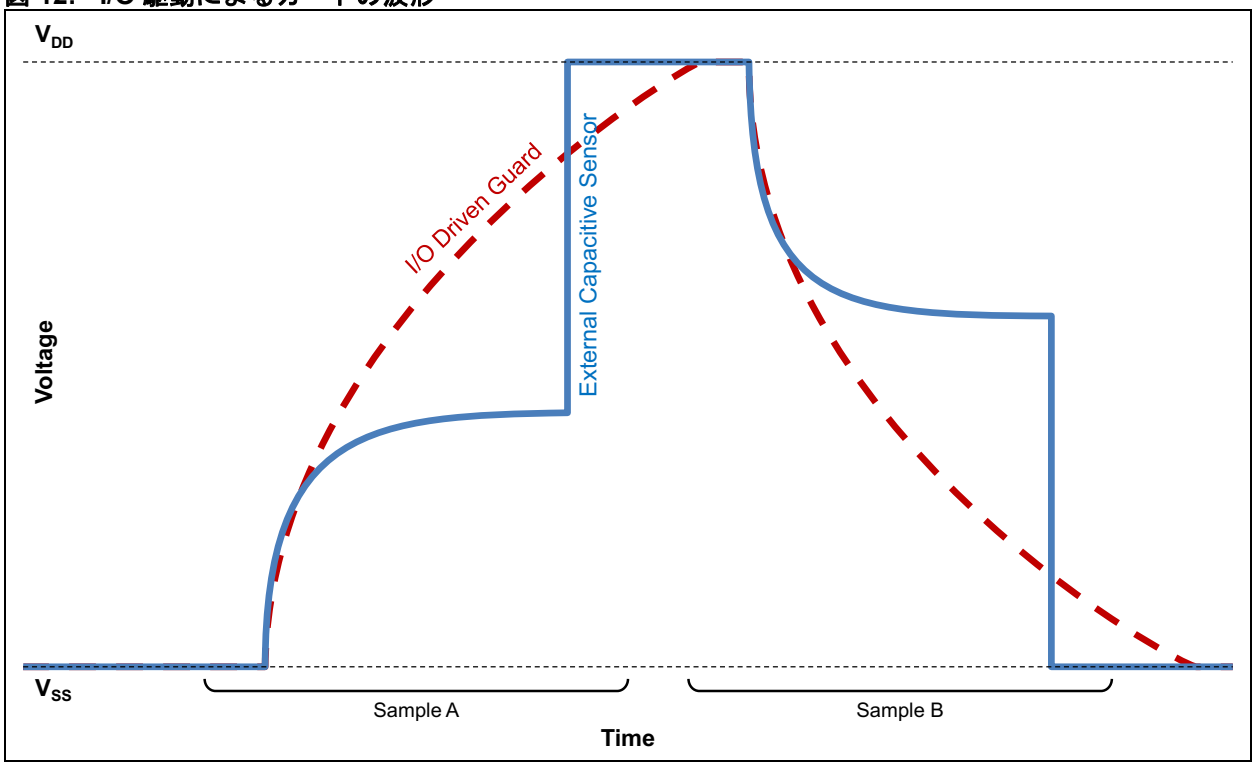
任意の I/O ピンによるガード

図 12 に示すように、CVD ガード信号の駆動に任意の I/O ピンを使えます。波形は完全には一致しないため、ガードの効率は低下します。しかしこの方法を検証した結果、完全に一致した波形による効果の 50 ～ 70% は達成できる事が分かりました。ピン 1 本を使う価値があるのは確かです。

ガードに直列抵抗を追加すれば充放電時の時定数を大きくする事ができます。ガード波形とセンサ波形の一致度が高い設計ほど、センサが感じる電位は小さくなり感度が高まります。

この実装方法は、ガードの出力と 2 つのコンデンサの接続の間に 1 命令サイクルの時間差があるため若干効率が下がります (補遺のサンプルコードを参照してください)。この時間差は、ハードウェア タイマと PWM 出力を使い、ガードの電位変化を 2 つのコンデンサの接続に同期させる事で解消できます。この方法は FOSC に応じて調整する必要があるため、サンプルコードを提供していません。

図 12: I/O 駆動によるガードの波形

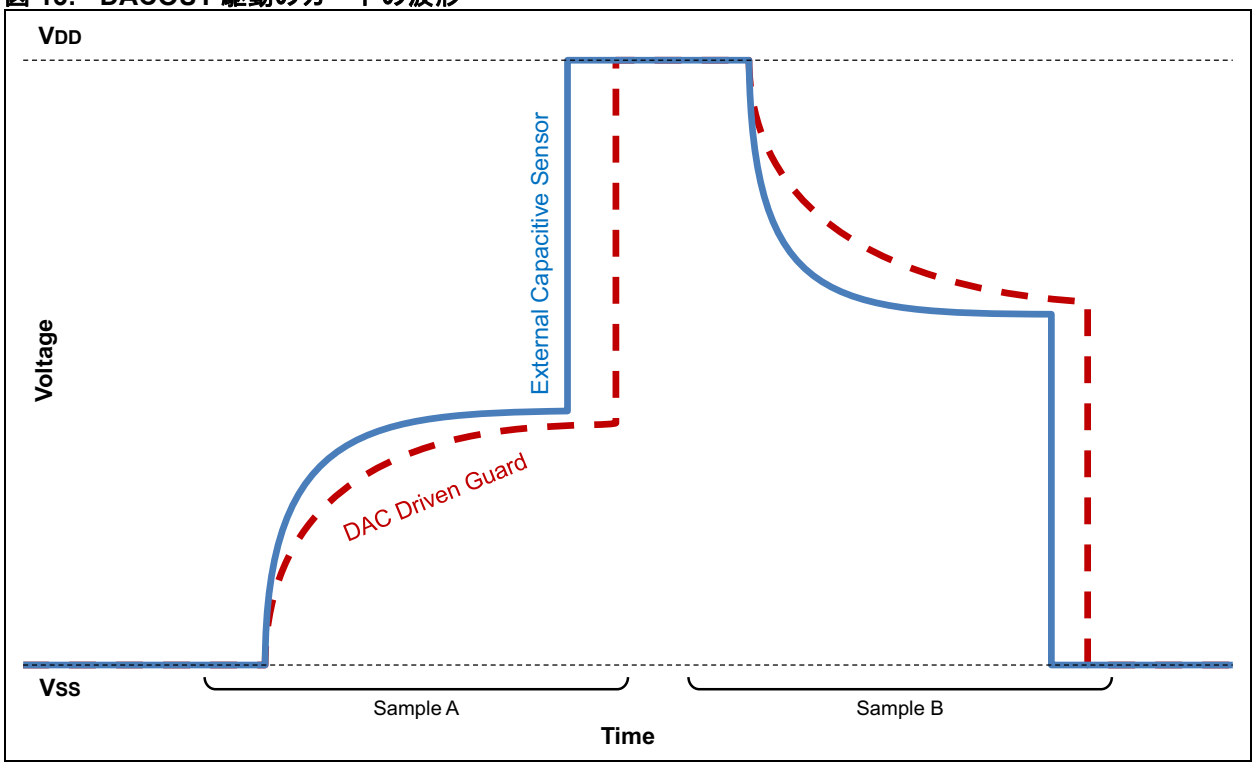


DACOUT ピンによるガード

図 13 に示すように DACOUT ピンをガード信号の駆動に使い、2 つのアクイジション段階における DAC のセトリング値を選択する事で、センサ波形との一致度を大きく向上できます。この方法では、完全に一致した波形の 70 ～ 90% の効果が得られます。DACOUT ピンを使える場合、この手法を推奨します。DACOUT ピンを使えない場合の次善策が、I/O 駆動によるガードを使う方法です。

サンプル A とサンプル B のそれぞれについて個別の CVD セトリング値を使い、各出力電圧に最も一致するように常時 DAC を調整する事で、ガードの効果をさらに向上できます。10 ビット ADC と 5 ビット DAC の場合、ADC 結果を 5 ビット右にシフトした値を次のサンプルの DACOUT セトリング電圧として使うだけで、この動作を実現できます。

図 13: DACOUT 駆動のガードの波形



相互ドライバによる CVD スキャン

CVD センシング手法は静電容量の相対変化を計測するために設計されました。しかし、静電容量式センサのような高インピーダンスのトレースは、グランド / 電源プレーン、アンテナ、高周波デジタルトレース等の近くの低インピーダンス源の影響も受けます。

アクティブ ガード トレースの目的は、周囲環境の電位がセンサに及ぼす影響を抑える事です。本書のガードに関するセクションで説明した通り、この手法は静電容量変化に対するセンサの感度を高めます。

一方、相互駆動トレースの目的は、高インピーダンスの静電容量式センサと低インピーダンスの相関信号の間のカップリングの変化を検出する事です。

相互駆動適用のシナリオ

相互駆動信号が設計上有利になる状況には主に以下の 2 つがあります。

- 金属片がセンサに接触する可能性がある場合にその金属を相関信号によって駆動すると、短絡発生時に生じるセンサの読み値のグリッチを抑止できます。

  (例えば Metal-Over-Capacitive システムの金属層です。これは必須ではありませんが、金属層がセンサと短絡する可能性がある場合には有効です。)
- 検出対象がセンサのグランド基準から絶縁されている場合、センサの近くに相互駆動を配置すると、センサと相互駆動の間の誘電率変化を検出できます。

Note:	マイクロチップ社の Metal-Over-Capacitive デザインの詳細は、アプリケーション ノート AN1325、『mTouch™ Metal-Over-Cap Technology』を参照してください。このアプリケーション ノートはマイクロチップ社のウェブサイト www.microchip.com/mTouch に掲載しています。

動作原理

相互駆動信号はサンプル A の全期間で Low、サンプル B の全期間で High に駆動される矩形波です (従って、それぞれのサンプルで外部センサの開始電圧と一致しています)。図 14 にこれを示します。

全ての CVD 波形の実行時に、静電容量式センサによって以下の 2 つの効果を計測します。

1. 標準的な CVD 波形の物理に従い、センサの静電容量が通常通り計測されます。

   サンプル A の値は、静電容量が増加すると小さくなります。

   サンプル B の値は、静電容量が増加すると大きくなります。
2. センサの相互駆動に対するカップリングも計測します。

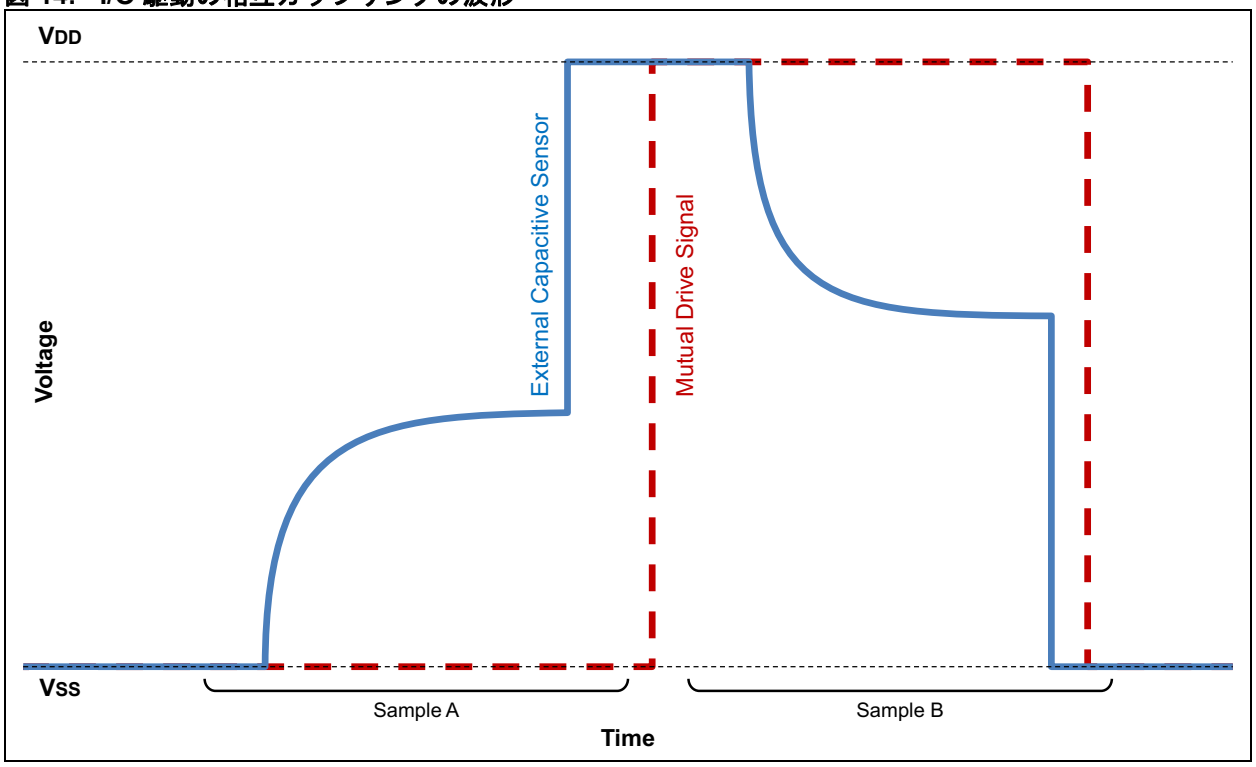
   サンプル A の値は、カップリングが増加すると小さくなります。

   サンプル B の値は、カップリングが増加すると大きくなります。

相互駆動がない場合、トレースとセンサ間のカップリングが増加すると、発生中の 2 つの効果の間に競合が発生します。

例えば、トレースが全期間 V_{SS} に駆動されている場合、カップリングの増加はサンプル B のセトリング電圧を低下させます。同時に、静電容量の増加はサンプル B のセトリング電圧を上昇させます。これら 2 つの効果は同時に発生する傾向があるため、相反する方向への変化により感度が低下します。さらに、一方の効果がもう一方の効果を上回ると、センサ信号が不自然な挙動を示します。

図 14:　I/O 駆動の相互カップリングの波形



相互駆動トレースと静電容量式センサの間のカップリングが増加すると、カップリング増による変化の方向と、静電容量増による変化の方向が一致します。これによりセンサの感度が高まり、競合の影響が避けられます。

ハードウェア CVD

一部の PIC® デバイスはハードウェアによって CVD 波形を生成できる高度な ADC モジュールを搭載しています。

例えば、以下のようなデバイスです。

- PIC16(L)F1512/3
  - アナログ入力 17 チャンネル
- PIC12LF1552
  - アナログ入力 4 チャンネル

このモジュールを使うと、各センサの専用スキャンルーチンを実装する必要がなくなるため、プログラムサイズを節約できます。また、mTouch センシングサービス ルーチンに必要な実行時間が短縮され、アプリケーションで実行可能な他の機能の数を増やす事ができます。ハードウェアがスキャン手法を管理して実行するため、CPU は他のアルゴリズムを自由に実行できます。

特長

ハードウェア CVD モジュールは、PIC16 拡張コアデバイスに搭載された通常の 10 ビット ADC に機能を追加しています。

- 詳細設定が可能な差動 CVD 波形を自動生成し、任意のアナログ チャンネルに出力
- サンプル A とサンプル B の結果を格納する ADC 結果レジスタを 2 セット搭載
- ソフトウェアによって内部 ADC 静電容量を +0 ～ +28 pF の範囲で 4 pF 刻みの調整が可能
- タイマと CCP モジュールに基づき変換を自動トリガ
- 内部 ADC バスにピンを接続すれば外部からの観測も可能
- プリチャージとアクイジションの波形タイミングを設定可能
- 1 ～ 2 本のピンに同期ガードリング駆動を出力

モジュールの設定

AADCON0 では ADC モジュールを有効にします。

AADCON1 では現在の FOSC の値に応じた変換クロックを設定します。個々のアプリケーション向け設定方法の詳細はデバイスのデータ シートを参照してください。推奨正参照電圧は、常に VDD です。

AADCON2 は、ADC の GO/$\overline{\text{DONE}}$ ビットを自動的にセットするトリガが存在するかどうかを決定します。存在する場合、GIE、PEIE、ADIE を有効にして割り込みサービスルーチン内で ADIF フラグを使い、結果を処理する事を推奨します。ADIF フラグは手動でクリアする必要があります。

AADCON3 は CVD 波形を操作します。プリチャージ段階が適切に機能するように、ADEPPOL と ADIPPOL には逆の値を設定する必要があります。本書における波形は ADEPPOL = 0、ADIPPOL = 1 ですが、逆でも動作します。2 つ目のサンプルを有効にするために、ADDSEN を High に設定します。2 つ目のサンプルを反転させて、サンプル A と同じではなく差動波形となるように ADIPEN を High に設定します。

AADPRE には「0」より大きい値を格納する必要があります。外部および内部コンデンサを完全に充電するのに十分な時間を確保する値に設定してください。詳細は、本書のタイミングに関する注意事項のセクションを参照してください。

AADACQ には「0」より大きい値を格納する必要があります。両方のコンデンサが完全にセトリングできる範囲で最小の値を設定します。詳細は、本書のタイミングに関する注意事項のセクションを参照してください。

必要に応じて AADGRD を使う事で、1 ～ 2 本のピンの同期ガード駆動を有効にできます。個々のアプリケーション向け設定方法の詳細はデバイスのデータシートを参照してください。

AADCAP を使うと、内部 ADC ホールド静電容量を増加させる事ができます。これは、外部センサが内部コンデンサよりも大きな静電容量を持つ場合にのみ効果があります。(外部コンデンサの静電容量が内部コンデンサよりも大きいのは、サンプル A のセトリング電圧が VDD/2 よりも低くなる場合です。) セトリング電圧が VDD/2 に一致するまで AADCAP の値を調整するのが理想的です。

動作原理

プロセスが自動的に実行されるという特性を持つため、このモジュールを使う場合は波形の駆動に C コードを使えます。CVD を手作業で実装する際のプログラミングについては、アセンブリのみを使う事を推奨します。

モジュールの設定を完了し、GO/$\overline{\text{DONE}}$ ビットをセットすると、ADC モジュールは CVD ステートマシンを実行して 2 回の変換を行います。GO/$\overline{\text{DONE}}$ ビットがクリアされ ADIF ビットがセットされるのは、両方の変換が終了した後です。

2 つの変換結果は、対応する結果レジスタセット (AADRES0 と AADRES1) に格納されます。(結果が格納される 8 ビットレジスタは、AADRES0L、AADRES0H、AADRES1L、AADRES1H の合計 4 つあります。)

例 1: ハードウェア CVD モジュールの初期化と使用

```
void InitHardwareADC(void)
{

    AADCON0bits.ADON    = 1;

    // Right aligned, Vref = Vdd
    AADCON1bits.ADFM    = 1;
    AADCON1bits.ADCS    = 0b101;
    AADCON1bits.ADPREF  = 0b00;

    // No trigger selected
    ADCON2bits.TRIGSEL  = 0b000;

    // Sample A: External Vss, Internal Vdd
    AADCON3bits.ADEPPOL = 0;
    AADCON3bits.ADIPPOL = 1;
    AADCON3bits.ADDSEN  = 1;
    AADCON3bits.ADIPEN  = 1;
    AADCON3bits.ADOLEN  = 0;
    AADCON3bits.ADOEN   = 0;
    AADCON3bits.ADOOEN  = 0;

    AADPRE   = PRECHARGE_DELAY;
    AADACQ   = SETTLING_DELAY;

    // Single-Guard-Pin enabled
    AADGRDbits.GRDAOE   = 1;
    AADGRDbits.GRDBOE   = 0;
    AADGRDbits.GRDPOL   = 0;

    // 4pf additional ADC capacitance
    ADCAPbits.ADDCAP    = 0b00001;

}

void ServiceHardwareADC(void)
{

    AADCON0bits.CHS = 0b00000;
    AADCON0bits.GO  = 1;

    while(AADCON0bits.GO);
    differentialResult  = AADRES1 | 0x0400;
    differentialResult -= AADRES0;

}
```

まとめ

静電容量式分圧 (CVD) センシングは、静電容量の相対変化を効果的に、低コストで検出できる手法です。

基本的なスキャン設定で大半のアプリケーションが正しく動作し、市場のあらゆるソリューションの中で最も少ない部品数しか必要としません。設計上の判断が正しく行われる限り、ガードトレース駆動や相互カップリング駆動等の高度な機能によって、あらゆるシステムで高水準の信号 / ノイズ比を実現できます。

マイクロチップ社の mTouch™ センシングの詳細と耐ノイズ性を高める推奨ハードウェア設計ガイドラインについては、弊社のウェブサイト www.microchip.com/mTouch をご覧ください。

補遺 A - サンプルコード

例 : DAC を参照電圧とする場合

このサンプルコードでは、内部ホールド コンデンサの参照電圧源に、独立したアナログ チャンネルではなく DAC を使って CVD 波形を実装しています。シングルセンサのアプリケーションでこの方法を使えば、スキャンの参照電圧用にピンを 1 本占有せずに済みます。

サンプル A とサンプル B で異なる部分を赤の太字で示しています。

サンプル A:

```
#define    PIC_DACCON0_VDD          0xC0
#define    PIC_DACCON1_VDD          0x1F
#define    PIC_ADCON0_SELECT_DAC    0x79
#define    PIC_ADCON0_SELECT_AN0    0x01

#define    SENSOR_LAT               LATA
#define    SENSOR_TRIS              TRISA
#define    SENSOR_PIN               0

; Initialize the DAC for 'VDD' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VDD
movwf       DACCON0
movlw       PIC_DACCON1_VDD
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
bsf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    PIC_DACCON0_VSS          0x00
#define    PIC_DACCON1_VSS          0x00
#define    PIC_ADCON0_SELECT_DAC    0x79
#define    PIC_ADCON0_SELECT_AN0    0x01

#define    SENSOR_LAT               LATA
#define    SENSOR_TRIS              TRISA
#define    SENSOR_PIN               0

; Initialize the DAC for 'VSS' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VSS
movwf       DACCON0
movlw       PIC_DACCON1_VSS
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for next Sample A
bcf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

例：センサを参照電圧とする場合

このサンプルコードでは、内部ホールド コンデンサの参照電圧源に、アナログ チャンネルをもう 1 つ使って CVD 波形を実装しています。DAC が不要なため、電力消費が最も少なくて済むスキャン方法です。

サンプル A とサンプル B で異なる部分を赤の太字で示しています。

サンプル A:

```
#define   SENSOR_LAT              LATA
#define   SENSOR_TRIS             TRISA
#define   SENSOR_PIN              0
#define   REFERENCE_LAT           LATA
#define   REFERENCE_PIN           1

#define   PIC_ADCON0_SELECT_AN0   0x01
#define   PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bsf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
bsf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define   SENSOR_LAT              LATA
#define   SENSOR_TRIS             TRISA
#define   SENSOR_PIN              0
#define   REFERENCE_LAT           LATA
#define   REFERENCE_PIN           1

#define   PIC_ADCON0_SELECT_AN0   0x01
#define   PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bcf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample A
bcf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

例：参照電圧に DAC を使ったダブル CVD 波形

このサンプルコードでは、内部 ADC ホールド静電容量の参照電圧源に DAC を使ってダブル CVD 波形を実装しています。

サンプル A:

```
#define    PIC_DACCON0_VDD          0xC0
#define    PIC_DACCON1_VDD          0x1F
#define    PIC_ADCON0_SELECT_DAC    0x79
#define    PIC_ADCON0_SELECT_AN0    0x01
#define    SENSOR_LAT               LATA
#define    SENSOR_TRIS              TRISA
#define    SENSOR_PIN               0

; Initialize the DAC for 'VDD' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VDD
movwf       DACCON0
movlw       PIC_DACCON1_VDD
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; First Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP
; Recharge the internal ADC capacitance
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0
NOP         ; Optional, variable delay
; Second Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
bsf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    PIC_DACCON0_VSS          0x00
#define    PIC_DACCON1_VSS          0x00
#define    PIC_ADCON0_SELECT_DAC    0x79
#define    PIC_ADCON0_SELECT_AN0    0x01
#define    SENSOR_LAT               LATA
#define    SENSOR_TRIS              TRISA
#define    SENSOR_PIN               0

; Initialize the DAC for 'VSS' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VSS
movwf       DACCON0
movlw       PIC_DACCON1_VSS
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; First Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP
; Recharge the internal ADC capacitance
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0
NOP         ; Optional, variable delay
; Second Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample A
bcf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

例：参照電圧に DAC を使ったハーフ CVD 波形

この方法では、未使用の ADC チャンネルで ADC の電荷を保存する必要があります。

このサンプルコードでは、内部 ADC ホールド静電容量の参照電圧源に DAC を使い、ハーフ CVD 波形を実装しています。

サンプル A:

```
#define    PIC_DACCON0_VDD         0xC0
#define    PIC_DACCON1_VDD         0x1F
#define    PIC_ADCON0_SELECT_DAC   0x79
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_UNIMP_CH     0xF1
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0

; Initialize the DAC for 'VDD' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VDD
movwf       DACCON0
movlw       PIC_DACCON1_VDD
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; First Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP
; Recharge the external capacitance
movlw       PIC_ADCON0_UNIMP_CH
movwf       ADCON0
bcf         INDF0, SENSOR_PIN   ; (TRIS)
NOP         ; Optional, variable delay
; Second Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1
NOP         ; (Recommended) 0.5 TAD Delay

; Precharge Sensor for Sample B
bsf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1
; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    PIC_DACCON0_VSS         0x00
#define    PIC_DACCON1_VSS         0x00
#define    PIC_ADCON0_SELECT_DAC   0x79
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_UNIMP_CH     0xF1
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0

; Initialize the DAC for 'VSS' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VSS
movwf       DACCON0
movlw       PIC_DACCON1_VSS
movwf       DACCON1

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_DAC
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; First Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP
; Recharge the external capacitance
movlw       PIC_ADCON0_UNIMP_CH
movwf       ADCON0
bcf         INDF0, SENSOR_PIN   ; (TRIS)
NOP         ; Optional, variable delay
; Second Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0
NOP
NOP         ; Acquisition / Settling Delay
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0, 1
NOP         ; (Recommended) 0.5 TAD Delay

; Precharge Sensor for Sample B
bcf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

btfsc       ADCON0, 1
goto        $-1
; Result stored in ADRESL and ADRESH
```

例：I/O 駆動のガード

このサンプルコードでは、内部ホールド コンデンサの参照電圧源に別のセンサを使い、ガード信号用に専用 I/O ピンを使った CVD 波形を実装しています。

サンプル A とサンプル B で異なる部分を赤の太字で示しています。参照電圧の初期化とガード駆動を除き、各ステップは同じである事に注意してください。

サンプル A:

```
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0
#define    REFERENCE_LAT           LATA
#define    REFERENCE_PIN           1
#define    GUARD_LAT               LATA
#define    GUARD_PIN               2
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bsf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   GUARD_LAT
movwf       FSR1L
movlw HIGH  GUARD_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF1, GUARD_PIN    ; Guard
bsf         INDF0, SENSOR_PIN   ; TRIS
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
BANKSEL     SENSOR_LAT
bsf         SENSOR_LAT, SENSOR_PIN
bcf         INDF0, SENSOR_PIN   ; (TRIS)

BANKSEL     ADCON0
btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0
#define    REFERENCE_LAT           LATA
#define    REFERENCE_PIN           1
#define    GUARD_LAT               LATA
#define    GUARD_PIN               2
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bcf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   GUARD_LAT
movwf       FSR1L
movlw HIGH  GUARD_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bcf         INDF1, GUARD_PIN    ; Guard
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample A
BANKSEL     SENSOR_LAT
bcf         SENSOR_LAT, SENSOR_PIN
bcf         INDF0, SENSOR_PIN   ; (TRIS)

BANKSEL     ADCON0
btfsc       ADCON0, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

例 : DACOUT 駆動のガード

このサンプルコードでは、内部ホールド コンデンサの参照電圧源に DAC を使うと同時に、DACOUT によるガードドライバとしても DAC を使い、CVD 波形を実装しています。

サンプル A:

```
#define    PIC_DACCON0_VDD         0xC0
#define    PIC_DACCON1_VDD         0x1F
#define    PIC_DACCON1_SETTLE_A    0x0E
#define    PIC_ADCON0_SELECT_DAC   0x79
#define    PIC_ADCON0_SELECT_AN1   0x05
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              1
#define    DACOUT_LAT              LATA
#define    DACOUT_PIN              0

; Initialize the DAC for 'VDD' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VDD
movwf       DACCON0
movlw       PIC_DACCON1_VDD
movwf       DACCON1

; Precharge ADC Capacitor
movlw LOW   ADCON0
movwf       FSR1L
movlw HIGH  ADCON0
movwf       FSR1H

movlw       PIC_ADCON0_SELECT_DAC
movwf       INDF1

movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

NOP         ; Optional delay

; Acquisition Stage
BANKSEL     DACCON0
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       INDF1
movlw       PIC_DACCON1_SETTLE_A
movwf       DACCON1          ; Set value
bsf         DACCON0, DACOE   ; Enable DACOUT

NOP         ; Acquisition / Settling Delay
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         INDF1, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
BANKSEL     SENSOR_LAT
bsf         SENSOR_LAT, SENSOR_PIN
bcf         INDF0, SENSOR_PIN   ; (TRIS)
BANKSEL     DACCON0
bcf         DACCON0, DACOE ; Disable DACOUT
BANKSEL     DACOUT_LAT
bsf         DACOUT_LAT, DACOUT_PIN

btfsc       INDF1, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    PIC_DACCON0_VSS         0x00
#define    PIC_DACCON1_VSS         0x00
#define    PIC_DACCON1_SETTLE_B    0x10

; Initialize the DAC for 'VSS' Reference
BANKSEL     DACCON0
movlw       PIC_DACCON0_VSS
movwf       DACCON0
movlw       PIC_DACCON1_VSS
movwf       DACCON1

; Precharge ADC Capacitor
movlw LOW   ADCON0
movwf       FSR1L
movlw HIGH  ADCON0
movwf       FSR1H

movlw       PIC_ADCON0_SELECT_DAC
movwf       INDF1

movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

NOP         ; Optional delay

; Acquisition Stage
BANKSEL     DACCON0
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       INDF1
movlw       PIC_DACCON1_SETTLE_B
movwf       DACCON1          ; Set value
bsf         DACCON0, DACOE   ; Enable DACOUT

NOP         ; Acquisition / Settling Delay
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         INDF1, 1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample A
BANKSEL     SENSOR_LAT
bcf         SENSOR_LAT, SENSOR_PIN
bcf         INDF0, SENSOR_PIN   ; (TRIS)
BANKSEL     DACCON0
bcf         DACCON0, DACOE ; Disable DACOUT
BANKSEL     DACOUT_LAT
bcf         DACOUT_LAT, DACOUT_PIN

btfsc       INDF1, 1
goto        $-1

; Result stored in ADRESL and ADRESH
```

例：参照電圧と相互駆動としてセンサを使った CVD サンプリング

サンプル A とサンプル B で異なる部分を赤の太字で示しています。参照電圧の初期化と相互駆動の方向を除き、各ステップは同じである事に注意してください。

サンプル A:

```
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0
#define    REFERENCE_LAT           LATA
#define    REFERENCE_PIN           1
#define    MUTUAL_LAT              LATA
#define    MUTUAL_PIN              2
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bsf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample B
bsf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

; Prepare Mutual Drive for Sample B
BANKSEL     MUTUAL_LAT
bsf         MUTUAL_LAT, MUTUAL_PIN

BANKSEL     ADCON0
btfsc       ADCON0, 1
goto        $-1
; Result stored in ADRESL and ADRESH
```

サンプル B:

```
#define    SENSOR_LAT              LATA
#define    SENSOR_TRIS             TRISA
#define    SENSOR_PIN              0
#define    REFERENCE_LAT           LATA
#define    REFERENCE_PIN           1
#define    MUTUAL_LAT              LATA
#define    MUTUAL_PIN              2
#define    PIC_ADCON0_SELECT_AN0   0x01
#define    PIC_ADCON0_SELECT_AN1   0x05

; Initialize AN1 as the Reference Source
BANKSEL     REFERENCE_LAT
bcf         REFERENCE_LAT, REFERENCE_PIN

; Precharge ADC Capacitor
BANKSEL     ADCON0
movlw       PIC_ADCON0_SELECT_AN1
movwf       ADCON0

movlw LOW   SENSOR_LAT
movwf       FSR1L
movlw HIGH  SENSOR_LAT
movwf       FSR1H
movlw LOW   SENSOR_TRIS
movwf       FSR0L
movlw HIGH  SENSOR_TRIS
movwf       FSR0H

; Optional additional and/or variable delay
NOP

; Acquisition Stage
movlw       PIC_ADCON0_SELECT_AN0
bsf         INDF0, SENSOR_PIN   ; (TRIS)
movwf       ADCON0

; Acquisition / Settling Delay
NOP
NOP
NOP

; Perform the ADC Conversion (GO/nDONE = 1)
bsf         ADCON0 ,1

; (Recommended) 0.5 TAD Delay
NOP

; Precharge Sensor for Sample A
bcf         INDF1, SENSOR_PIN   ; (LAT)
bcf         INDF0, SENSOR_PIN   ; (TRIS)

; Prepare Mutual Drive for Sample A
BANKSEL     MUTUAL_LAT
bcf         MUTUAL_LAT, MUTUAL_PIN

BANKSEL     ADCON0
btfsc       ADCON0, 1
goto        $-1
; Result stored in ADRESL and ADRESH
```

NOTES:

マイクロチップ社製デバイスのコード保護機能に関して以下の点にご注意ください。

- マイクロチップ社製品は、該当するマイクロチップ社データシートに記載の仕様を満たしています。
- マイクロチップ社では、通常の条件ならびに仕様に従って使用した場合、マイクロチップ社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解では、こうした手法はマイクロチップ社データシートにある動作仕様書以外の方法でマイクロチップ社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- マイクロチップ社は、コードの保全性に懸念を抱いているお客様と連携し、対応策に取り組んでいきます。
- マイクロチップ社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、マイクロチップ社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。マイクロチップ社では、常に製品のコード保護機能の改善に取り組んでいます。マイクロチップ社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

商標

ISBN: 978-1-62077-102-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

マイクロチップ社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州) の本部、設計部およびウェハー製造工場そしてカリフォルニア州とインドのデザインセンターが ISO/TS-16949:2009 認証を取得しています。マイクロチップ社の品質システム プロセスおよび手順は、PIC® MCU および dsPIC® DSC、KEELOQ® コード ホッピング デバイス、シリアル EEPROM、マイクロペリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関するマイクロチップ社の品質システムは ISO 9001:2000 認証を取得しています。

各国の営業所とサービス

北米

本社
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
技術サポート :
http://www.microchip.com/support
URL:
www.microchip.com

アトランタ
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

ボストン
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

シカゴ
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

クリーブランド
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

ダラス
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

デトロイト
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

インディアナポリス
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

ロサンゼルス
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

サンタクララ
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

トロント
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

アジア / 太平洋

アジア太平洋支社
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

オーストラリア - シドニー
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重慶
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港 SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青島
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 瀋陽
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武漢
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦門
Tel: 86-592-2388138
Fax: 86-592-2388130

中国 - 珠海
Tel: 86-756-3210040
Fax: 86-756-3210049

アジア / 太平洋

インド - バンガロール
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

インド - ニューデリー
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

インド - プネ
Tel: 91-20-3019-1500

日本 - 大阪
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 - 東京
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韓国 - 大邱
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韓国 - ソウル
Tel: 82-2-554-7200
Fax: 82-2-558-5932 または
82-2-558-5934

マレーシア - クアラルンプール
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

マレーシア - ペナン
Tel: 60-4-227-8870
Fax: 60-4-227-4068

フィリピン - マニラ
Tel: 63-2-634-9065
Fax: 63-2-634-9069

シンガポール
Tel: 65-6334-8870
Fax: 65-6334-8850

台湾 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

台湾 - 高雄
Tel: 886-7-213-7828
Fax: 886-7-330-9305

台湾 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

タイ - バンコク
Tel: 66-2-694-1351
Fax: 66-2-694-1350

ヨーロッパ

オーストリア - ヴェルス
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

デンマーク - コペンハーゲン
Tel: 45-4450-2828
Fax: 45-4485-2829

フランス - パリ
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

ドイツ - ミュンヘン
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

イタリア - ミラノ
Tel: 39-0331-742611
Fax: 39-0331-466781

オランダ - ドリューネン
Tel: 31-416-690399
Fax: 31-416-690340

スペイン - マドリッド
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

イギリス - ウォーキンガム
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/20/13