

Application Note

MSPM0 に基づく抵抗性タッチスクリーン制御

Eason Zhou, Wiky Liao, Yashraj Motwani, Divyansh Mittal

概要

このアプリケーション ノートでは、MSPM0L1306 マイコンを使った 4 線式抵抗膜方式タッチスクリーンの実装について説明します。GPIO と ADC のインターフェイスを使って、抵抗膜方式タッチスクリーンでリアルタイムにタッチを検出します。この実装には二つのサンプルがあります。一つはタッチの有無を確認して座標データを送信するだけのタッチ検出サンプル、もう一つはクリック イベントとドラッグ / 長押しイベントを区別し、それに応じて座標データを送信するクリック / ドラッグ検出サンプルです。このドキュメントでは、ハードウェア セットアップ、ピン設定の方針、ファームウェアの実装について詳しく説明します。また、MSPM0L1306 プラットフォームを使って組込みシステムに抵抗膜方式タッチ機能を組み込む開発者向けに、詳しい手順も紹介します。

目次

1 概要.....	3
2 LaunchPad のセットアップ.....	4
3 ハードウェア接続.....	5
4 設計上の考慮事項.....	7
4.1 IO 選択の検討事項.....	7
4.2 測定方式.....	9
5 ソフトウェアの構造.....	11
5.1 タッチ検出の例.....	11
5.2 抵抗性クリック アンド ドラッグの検出.....	11
5.3 ピン モード スイッチング.....	12
6 実行手順.....	12
7 結果.....	13
7.1 タッチ検出.....	13
7.2 クリックおよびドラッグの検出.....	14
8 まとめ.....	15
9 参考資料.....	15
10 改訂履歴.....	16

図の一覧

図 2-1. Launchpad のジャンパの取り外し.....	4
図 3-1. システム ブロック図.....	5
図 3-2. ハードウェア設定.....	6
図 4-1. タッチ検出セットアップ.....	9
図 4-2. X 軸 ADC の測定.....	10
図 4-3. Y 軸 ADC の測定.....	10
図 5-1. ソフトウェアプロジェクト.....	11
図 7-1. タッチ検出: 12 ビット分解能.....	13
図 7-2. クリックの出力.....	14
図 7-3. 長押し用の出力.....	14

表の一覧

表 2-1. LaunchPad ピンの接続.....	4
表 4-1. 絶対最大定格.....	7

商標

Code Composer Studio (CCS)[™] is a trademark of Texas Instruments.

すべての商標は、それぞれの所有者に帰属します。

1 概要

抵抗膜方式タッチスクリーンは、堅牢で低コストなうえ、手袋をはめた指を含むあらゆる入力手段で操作できることから、産業用、医療用、民生用のアプリケーションで広く採用されています。静電容量方式タッチスクリーンと異なり、抵抗膜方式は特殊な導電性材料を必要とせず、過酷な環境でも優れた性能を発揮します。

MSPM0L1306 マイコンは、抵抗膜方式タッチスクリーン コントローラー式を実装するのに必要なアナログ インターフェイスとデジタル インターフェイスを備えています。このアプリケーション ノートでは、**GPIO** ピンで抵抗膜層を駆動し、内蔵 **ADC** で座標を測定する、タッチ検出と解析のための実用的なファームウェア実装を紹介します。二つのサンプル プロジェクトを用意しています。一つは基本的なタッチ検出システム、もう一つは **TimerG** ペリフェラルを使った時間計測により、クリック イベントとドラッグ / 長押しジェスチャを区別できる応用システムです。

抵抗性タッチスクリーンは抵抗ネットワーク全体に電圧を印加し、入力スタイラス、ペン、または指でスクリーンに触れた、マトリックス上の特定のポイントの抵抗変化を測定することで機能します。このデバイスは、**ITO (酸化インジウム スズ)** でできた二枚の透明な抵抗膜層で構成されており、二枚の層は薄い空気層で隔てられています。ユーザーがスタイラス、ペン、または指で押すと、上側の層が下側の層に接触し、その位置で回路が閉じます。マイコンはこの方法を使用してタッチの位置を取得します。

関連するソフトウェアは [MSPM0L1306-RESISTIVE-TOUCH-APPSW](#) で入手可能です。

2 LaunchPad のセットアップ

MSPM0L1306 LaunchPad については、表 2-1 はピンおよびピン機能を示しています。これは図 3-1 にも示されています。詳細については、ソフトウェアプロジェクトの SysConfig ファイルを参照してください。

表 2-1. LaunchPad ピンの接続

使用ピン	ピン機能	スクリーン制御
PA16	GPIO、ADC チャンネル 8	Y 軸正方向
PA24	GPIO	
PA0	GPIO	
PA17	GPIO	X 軸正方向
PA25	GPIO	Y 軸負方向
PA21	GPIO	
PA22	GPIO	
PA15	GPIO、ADC チャンネル 9	X 軸負方向

このデモで使用するピンは、LaunchPad 上の他のペリフェラルにも接続されています。図 2-1 に示すように、テストする前にこれらのジャンパを取り外す必要があります。これらのジャンパを取り外すと、ピンがデフォルトのペリフェラルの割り当てから解放され、GPIO および ADC ピンとして機能できるようになります。これはタッチスクリーンの実装に不可欠です。ファームウェアがこれらのピンに対して直接的な GPIO 制御と ADC サンプル機能が必要とするためです。ジャンパは、PA25 については J4、PA22 については J6、PA15 については J1、PA0 については J2 を取り外します。

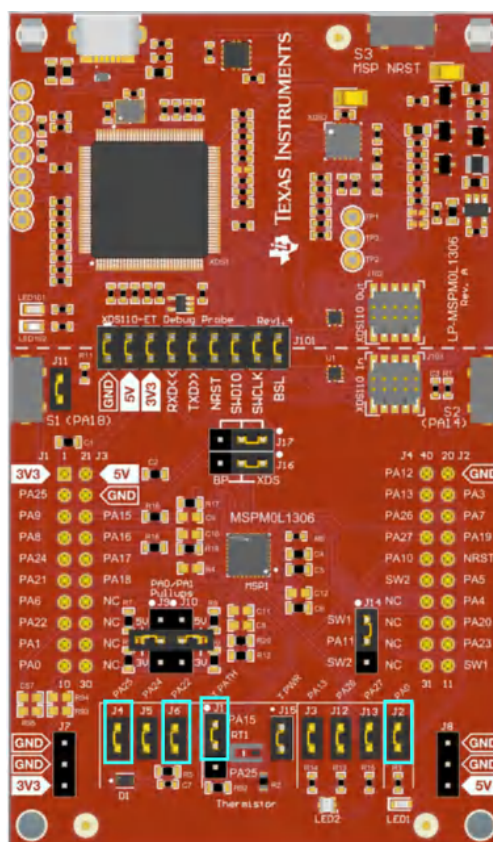


図 2-1. Launchpad のジャンパの取り外し

3 ハードウェア接続

図 3-1 に、Launchpad ピンの接続表の後に続くシステム ブロック図を示します。

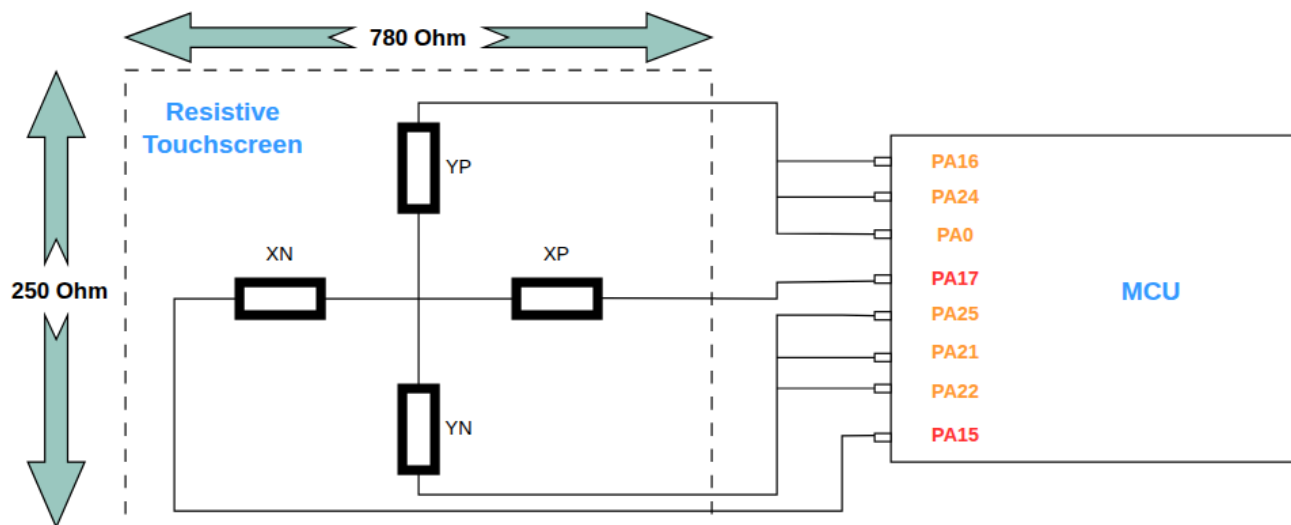


図 3-1. システム ブロック図

図 3-2 に、タッチスクリーンで詳細を確認できるハードウェア設定を示します。使用した抵抗膜方式タッチスクリーンは、X 軸方向が 780Ω、Y 軸方向が 250Ω の抵抗値です。使用する抵抗性タッチスクリーンのピン構成情報と、抵抗に基づいて行う必要のある計算については、「[設計上の考慮事項](#)」セクションを参照します。

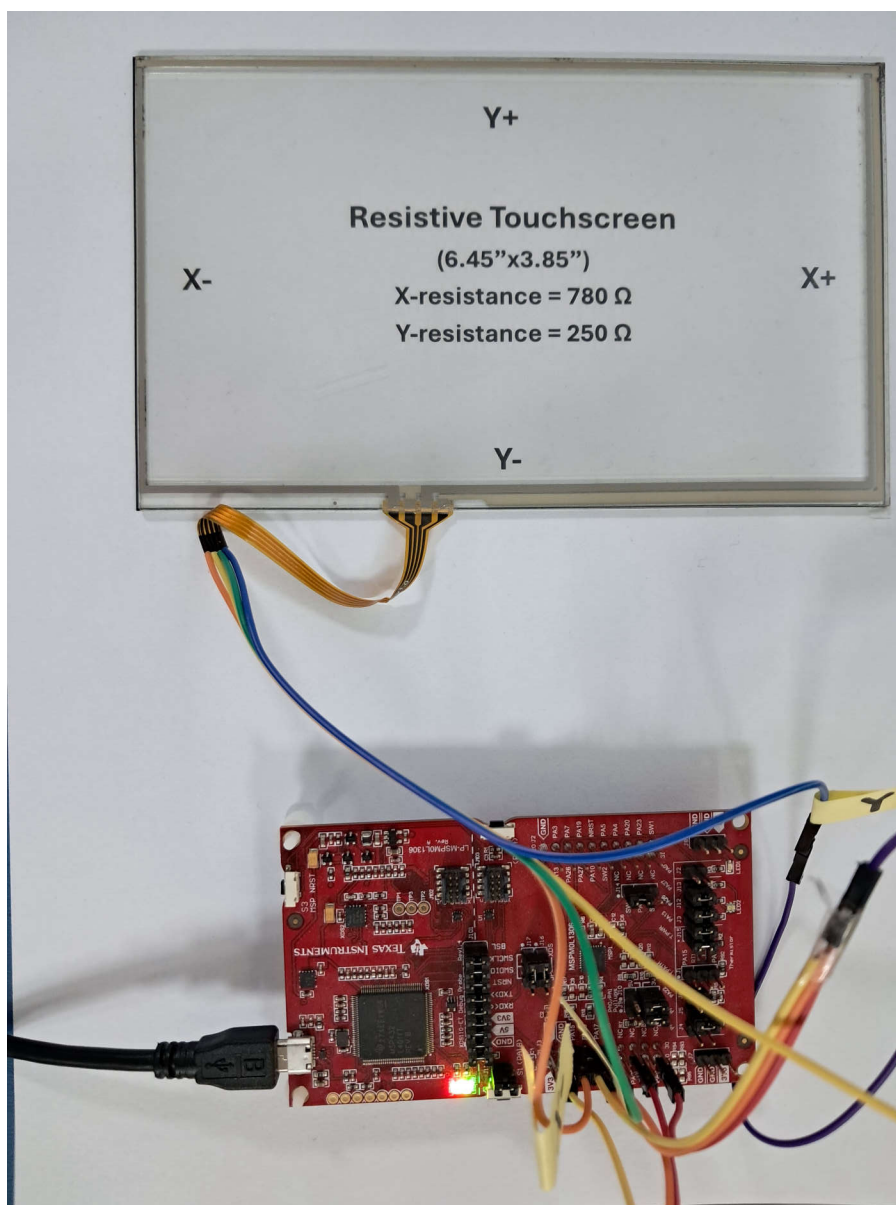


図 3-2. ハードウェア設定

4 設計上の考慮事項

4.1 IO 選択の検討事項

図 3-1 では、3 つの IO を Y 軸方向の駆動に使用し、一つの IO を X 軸の駆動に使用しています。注意すべき 2 つの検討事項があります：

1. MSPM0 IO の電流出力と入力制限
2. MSPM0 IO の内部抵抗

抵抗式タッチスクリーンには X 軸があります：780Ω および Y 軸：250Ω。3.3V をシステム電源として使用します。

$$I = V/R \quad (1)$$

そのため、ユーザーが X 軸の抵抗を駆動するには 4.2mA が必要で、Y 軸の抵抗を駆動するには 13.2mA が必要です。

表 4-1 に示すように、MSPM0 L1306 の電流制限は 6mA で、MSPM0 データシートに記載されています。この仕様を超えると、MSPM0 に恒久的な損傷が発生する可能性があります。そのため TI は、この抵抗膜方式タッチスクリーンにおいて、Y 軸の駆動に 3 つの GPIO、X 軸の駆動に一つの GPIO を使用することを推奨しています。ただし、必要なピン数は使用するタッチスクリーンによって変わる場合があります。実装に必要な理想的なピン数を求めるには、上記の方法に従ってください。

注

上記のパネル抵抗値に基づいてピンを追加または削除する必要がある場合は、そのピンが必要どおりに機能するよう、MSPM0 側でジャンパが取り外されていることも確認する必要があります。

表 4-1. 絶対最大定格

パラメータ		テスト条件	最小値	最大値	単位
VDD	電源電圧	VDD ピンで	-0.3	4.1	V
V _I	入力電圧	5V 対応のオープンドレイン ピンに印加	-0.3	5.5	V
V _I	入力電圧	任意の通常耐圧ピンに印加	-0.3	V _{DD} + 0.3 (最大 4.1)	V
I _{VDD}	VDD ピンに流れ込む電流 (ソース)	-40°C ≤ T _j ≤ 130°C		80	mA
	VDD ピンに流れ込む電流 (ソース)	-40°C ≤ T _j ≤ 85°C		100	mA
I _{VSS}	VSS ピンから流れ出す電流 (シンク)	-40°C ≤ T _j ≤ 130°C		80	mA
	VSS ピンから流れ出す電流 (シンク)	-40°C ≤ T _j ≤ 85°C		100	mA
I _{IO}	SDIO ピンの電流	SDIO ピンによってシンクまたはソースされる電流		6	mA
	HSIO ピンの電流	HSIO ピンによってシンクまたはソースされる電流		6	mA
	ODIO ピンの電流	ODIO ピンによってシンクされる電流		20	mA
I _D	サポートされているダイオード電流	すべてのデバイス ピンのダイオード電流		±2 (1)	mA

- (1) 絶対最大定格を上回るストレスが加わった場合、デバイスに永続的な損傷が発生する可能性があります。これはストレスの定格のみについて言及して、絶対最大定格において、またはこのデータシートの「推奨動作条件」に示された値を超える他のいかなる条件でも、本製品が正しく動作することを暗に示すものではありません。絶対最大定格の状態に長時間置くと、デバイスの信頼性に影響を及ぼす場合があります。

MSPM0 の内部 IO 抵抗により、抵抗性スクリーン検出にオフセットが発生します。一部の高分解能アプリケーションでは、これが許容されません。影響を軽減したい場合は、パフォーマンスをさらに向上させる二つの方法があります：

1. より多くの IO を追加してスクリーンを駆動し、負荷電流をより多くの IO ピンに分散します。

2. 正の軸と負の軸の両方の IO に ADC を追加します。スクリーンの駆動時に、IO の内部抵抗に割り当てられている電圧を直接測定します。

4.2 測定方式

4.2.1 タッチ検出方式

抵抗レイヤが接続されていない場合、ADC はフローティングになります。タッチがない場合、結果は IO スイッチからの結合ノイズの影響が大きくなります。つまり、ADC は常に固定座標を出力します。タッチされると 2 つの抵抗レイヤの間に抵抗が含まれ、タッチが非常に弱いときに抵抗が非常に高くなります。これにより ADC 出力にも外乱が発生します。

セクション 2 に MSPM0 セットアップを示します。検出を行うときは、Y 軸を正の IO に、すべての出力の Y 軸を負の IO に設定します。このデモでは、測定に ADC を使用しています。ローパワー要件の場合、代わりに GPIO を使用し、ウェイクアップ機能と組み合わせて使用できます。

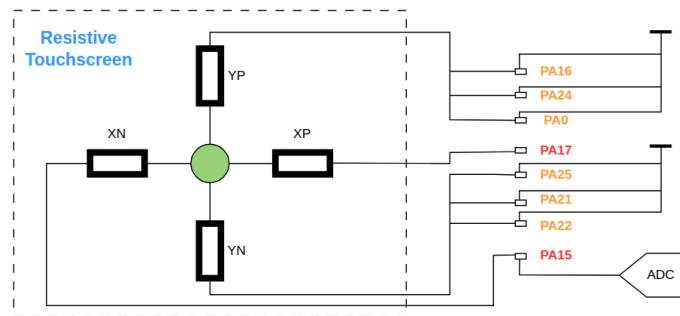


図 4-1. タッチ検出セットアップ

ADC を使用して検出を実行し、タッチ動作の低圧力状態をフィルタリングします。MSPM0 の ADC は 12 ビットモードに設定されており、読み取られる ADC 値の最大カウントは 4095 となります。resistive_detection.h ファイルで定義されている TOUCH_DETECTION_THD マクロを 4000 に設定して、抵抗性パネルのタッチ検出を検出します。読み取る ADC のノイズに応じて、試行錯誤により検出信号を解除するタイミングを確認しながら、このパラメータ値を調整できます。

4.2.2 X 軸と Y 軸の測定

ユーザーが押すと、上側の層が下側の層に接触し、その位置で回路が閉じます。一方の軸 (例えば Y 軸) に電圧を印加し、それと直交する軸 (X 軸) で電圧降下を測定することで、マイコンはタッチ位置の X 座標と Y 座標を正確に求めることができます。

セクション 4.2.1 にタッチ検出方法を示します。タッチがあると、この検出方式では抵抗膜層の間に閉じた回路ができます。一方の軸 (Y 軸) に既知の電圧 (3.3V) を印加し、それと直交する軸 (X 軸) で生じる電圧降下を測定することで、マイコンはタッチ位置を正確に計算します。

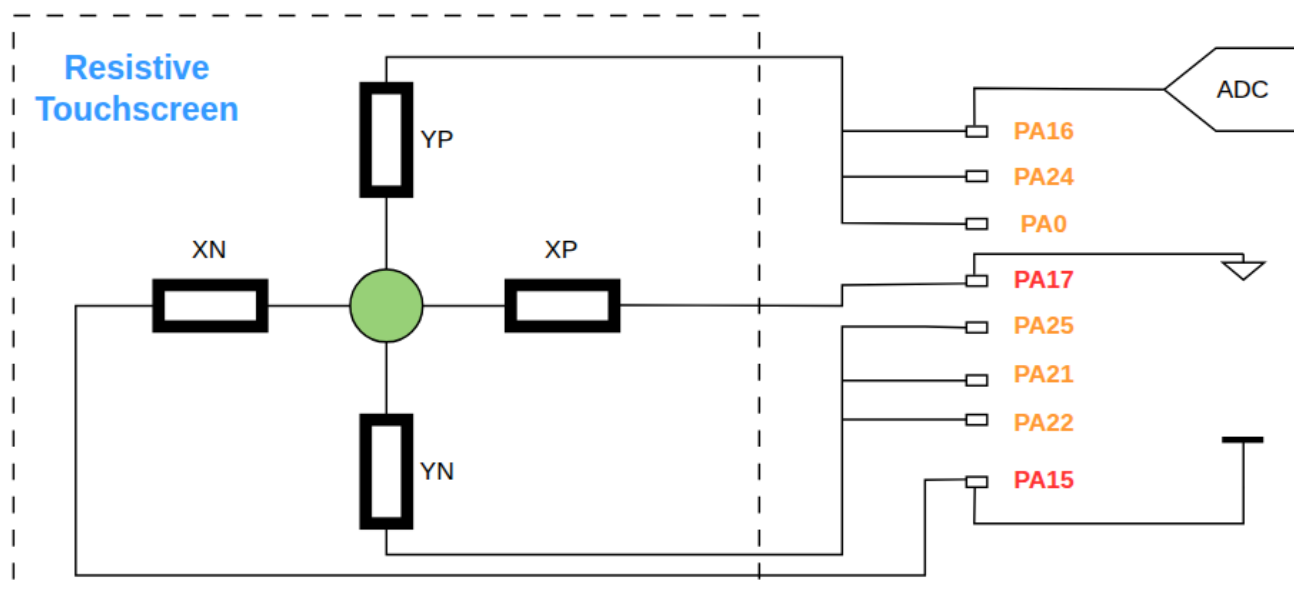


図 4-2. X 軸 ADC の測定

X 軸検出の場合は、PA17 を出力 High に、PA15 を出力 Low に設定します。他のピンをアナログ機能に設定します (ハイインピーダンスモード)。次に、PA16 の ADC チャンネルを有効にして測定を実行します。

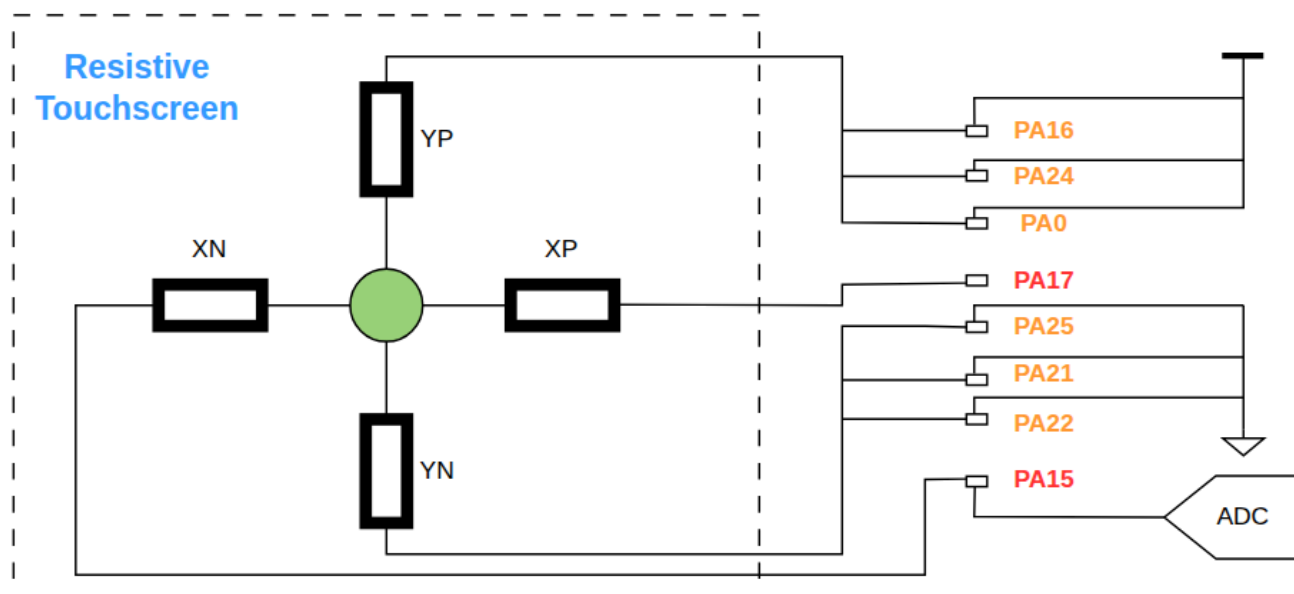


図 4-3. Y 軸 ADC の測定

Y 軸を検出するには、PA16、PA24、PA0 を High 出力に、PA25、PA22、PA21 を Low 出力に設定します。他のピンをアナログ機能に設定します (ハイインピーダンスモード)。次に、PA15 の ADC チャンネルを有効にして測定を実行します。

電圧を取得した後、ADC はノイズを減らすためにオーバーサンプリングを行い、その ADC の結果をそのまま座標として UART コンソールに出力します。

5 ソフトウェアの構造

CCS で開発されたソフトウェアプロジェクトを図 5-1 に示します。ソフトウェア プロジェクトは主に 3 つの部分で構成されています:

1. **resistive_detection** 部には、Y 軸と X 軸を取得するための IO と ADC 制御が含まれます。
2. **UART** 部には主に M0 の UART プロトコルに基づく書き込み機能が含まれており、バックチャネル UART 経由で測定結果を PC にアップロードします。
3. 主要部分には、最高のシステム機能コードが含まれています。

マイコンの電源投入後、MSPM0 は抵抗膜方式のタッチ検出を開始します。タッチが検出されると、抵抗膜方式スクリーン検出は Y 軸と X 軸の電圧を測定し、座標を UART コンソールにアップロードします。

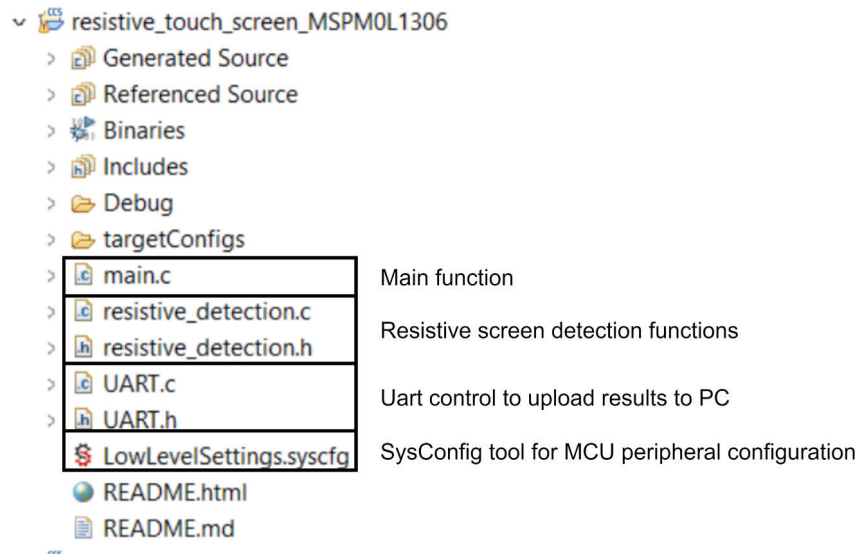


図 5-1. ソフトウェアプロジェクト

5.1 タッチ検出の例

タッチ検出の例では、タッチが検出された場合、座標値はサンプリング レート 200ms (**INACTIVE_SAMPLE_DELAY**) で印刷されます。この例では、クリックと長押しやドラッグの区別はありません。

タッチが検出されると、ADC 値が読み取られて X 座標と Y 座標が取得され、UART 経由で PC のコンソールに送信されます。

注

使用する抵抗膜方式タッチスクリーンパネルの構成に応じて、ADC の最小値と最大値を更新します。

5.2 抵抗性クリック アンド ドラッグの検出

この例では、抵抗性タッチスクリーンでタッチが検出されない場合のサンプリング レート (ms) が 200ms (**INACTIVE_SAMPLE_DELAY**) です。スレッショルドは 80ms (**LONG_TOUCH_THRESHOLD_MS**) に設定されており、クリック / 短押しとドラッグ / 長押しを区別するために使用されます。タッチの継続時間がしきい値未満の場合、MSPM0 はこれをクリックと判定します。一方、タッチの継続時間がスレッショルドを超えた場合、MSPM0 はこれを長押しと判定します。

長押しの場合、タッチが解除されたことを検出するために使用される 4ms (**TOUCH_RELEASE_SAMPLING_RATE**) で連続サンプリングが行われます。各サンプリングによって、一連の X 座標と Y 座標が生成され、UART を使用して PC コンソールに送信されます。

分類は、タッチの時間の長さに基づいて行われます。この時間は **TimerG** ペリフェラルで計測し、SysConfig ツールで有効にします。これにより、タッチの開始時刻と終了時刻を取得できます。

注

現在のセットアップの最大タッチ時間は 100 秒です。タッチがその時間を超えると、タイマがリセットされて誤った時間が読み取られ、クリックとドラッグの判定にも支障が出る可能性があります。

注

ユーザーは、使用する抵抗膜方式タッチスクリーン パネルに合わせて、ADC の最小値と最大値を変更する必要があります。そうでない場合、これは送信される座標を妨げ、ソフトウェアは意図したとおりに動作しない可能性があります。

5.3 ピン モード スイッチング

このセクションでは、ADC ピンおよび GPIO ピンとして使用された PA15 および PA16 のピン設定について説明します。これは、二つのピンのピン モードを切り替えることで行います。マイコンは、同じ物理ピン上で読み取りと駆動を同時に行うことはできません。そのため、二相で測定する方法が必要になります。

X 座標を読み取るときは、Y 軸のピンを GPIO 出力として駆動して電位差を作り、X 軸のピン (PA15) を ADC 入力に設定して、その電圧を ADC のチャンネル 8 で測定します。これは、`resistorical_detection.c` ファイル内の `readTouchX()` API に実装されています。Y 座標の読み取りも同様に、PA16 ピンを使って行います。そのため、PA15 と PA16 のピンを GPIO モードと ADC モードの間で切り替える必要があります。

`touchIOReset()` 関数は、`DL_GPIO_initPeripheralAnalogFunction()` を使って、すべてのタッチ ピンをペリフェラルのアナログ モードに戻します。次に、測定関数は `DL_GPIO_initDigitalOutput()` を呼び出して、特定のピンを GPIO モードに一時的に切り替えます。測定が終わると、次のサイクルに備えてピンはアナログ モードに戻ります。ピン設定を変更した後は、ADC で測定する前に電圧が安定するよう、セトリング遅延を入れます。

6 実行手順

このセクションでは、MSPM0L1306 Launchpad で抵抗膜方式タッチ検出のサンプルを実行する手順を説明します。

1. セクション 4.1 の手順に従って、必要な IO (抵抗とピン) を計算してください。
2. セクション 3 に従って、MSPM0L1306 LaunchPad と抵抗膜方式タッチスクリーンの間のハードウェアをセットアップしてください。
3. Code Composer Studio (CCS)™ をコンピュータにインストールします。
4. 「はじめに」セクションのリンクに示されている例をインポートします。`resistive_touch_detection` または `resistive_click_drag_detection` の例をロードします。
5. ユーザーは、使用するタッチスクリーンの X 軸と Y 軸について、最小値と最大値を追加する必要があります。これにより、ノイズが座標として送信されないことが保証されます。
6. また、パネルの感度とノイズに応じて、`TOUCH_DETECTION_THD` と検出周波数も調整する必要があります。
7. サンプルをビルドします。
8. CCS、`MSPM0_XDS110.ccxml` で MSPM0 のターゲット構成を作成し、XDS110 デバッグ プローブと MSPM0L1306 デバイス タイプを選択します。
9. 「Target Configurations」(ターゲット構成) ウィンドウで、「`MSPM0_XDS110.ccxml`」を右クリックします。
10. 選択した構成を起動するを選択します
11. 「Run」(実行) -> 「Connect Target」(ターゲットの接続) の順に選択します
12. 「Run」(実行) -> 「Load」(ロード) -> 「Load Program」(プログラムのロード) の順に選択します
13. MSPM0 プロジェクトのバイナリを参照し、「OK」をクリックします。
14. これにより、バイナリでフラッシュに書き込まれます。
15. 抵抗性タッチスクリーンにタッチした後で、X 軸と Y 軸の ADC 値が検出され、結果が UART ペリフェラルを経由してコマンドラインに送信されます (「結果」セクションを参照)。座標は、パネルでタッチが検出された場合にのみ更新されます。

7 結果

7.1 タッチ検出

図 7-1 に、12 ビット分解能に構成された ADC の出力を示します。

```
DEBUG: Touch detection started
TOUCH DETECTED! X=2183 Y=1199
TOUCH DETECTED! X=1721 Y=1235
TOUCH DETECTED! X=1713 Y=1243
TOUCH DETECTED! X=1711 Y=1238
TOUCH DETECTED! X=3246 Y=2658
TOUCH DETECTED! X=1555 Y=2702
TOUCH DETECTED! X=2065 Y=2707
TOUCH DETECTED! X=2965 Y=571.
TOUCH DETECTED! X=3424 Y=498.
TOUCH DETECTED! X=2300 Y=477.
TOUCH DETECTED! X=1689 Y=470.
TOUCH DETECTED! X=1453 Y=477.
TOUCH DETECTED! X=3518 Y=487.
TOUCH DETECTED! X=1316 Y=462.
TOUCH DETECTED! X=1019 Y=447.
TOUCH DETECTED! X=980. Y=452.
TOUCH DETECTED! X=932. Y=465.
```

図 7-1. タッチ検出: 12 ビット分解能:

7.2 クリックおよびドラッグの検出

図 7-2 および図 7-3 は、抵抗膜方式タッチスクリーン上でユーザーが行える異なる種類の押下 (クリックと長押し (またはドラッグ)) に対する出力を示しています。

```
TOUCH DETECTED! X=2160.0 Y=1902.0 TIME=54.000ms - CLICK
TOUCH DETECTED! X=3053.0 Y=1656.0 TIME=46.000ms - CLICK
TOUCH DETECTED! X=3143.0 Y=1661.0 TIME=69.000ms - CLICK
TOUCH DETECTED! X=3893.0 Y=1577.0 TIME=38.000ms - CLICK
TOUCH DETECTED! X=3088.0 Y=1448.0 TIME=69.000ms - CLICK
TOUCH DETECTED! X=3265.0 Y=1478.0 TIME=54.000ms - CLICK
TOUCH DETECTED! X=3616.0 Y=1414.0 TIME=54.000ms - CLICK
```

図 7-2. クリックの出力

```
TOUCH DETECTED! X=2335.0 Y=1066.0 (2364.0,1092.0) (2723.0,1061.0) (3992.0,531.00) TIME=171.00ms - LONG TOUCH
TOUCH DETECTED! X=2374.0 Y=1173.0 (2358.0,1209.0) (2418.0,1198.0) (2412.0,1193.0) (2417.0,1199.0) (2387.0,1207.0) (2336.0,1221.0) (2331.0,1228.0) (2332.0,1229.0) (2352.0,1221.0) (2436.0,1185.0) (3032.0,1158.0) TIME=396.00ms - LONG TOUCH
TOUCH DETECTED! X=4056.0 Y=583.00 (2864.0,1392.0) (2881.0,1435.0) (2945.0,1485.0) (2929.0,1533.0) (3042.0,1630.0) (2557.0,1675.0) (2688.0,1732.0) (2719.0,1760.0) (2776.0,1805.0) (3055.0,1783.0) TIME=365.00ms - LONG TOUCH
TOUCH DETECTED! X=2980.0 Y=1405.0 (2552.0,1451.0) (2301.0,1466.0) (2933.0,1476.0) (2786.0,1538.0) (2595.0,1622.0) (2790.0,1659.0) (2518.0,1688.0) (2515.0,1723.0) (2526.0,1763.0) (2603.0,1798.0) (2761.0,1870.0) (2719.0,1922.0) (2790.0,1945.0) (3135.0,1958.0) TIME=489.00ms - LONG TOUCH
TOUCH DETECTED! X=2201.0 Y=830.00 (2086.0,961.00) (2144.0,1000.0) (2346.0,1051.0) (2444.0,1100.0) (2540.0,1183.0) (2863.0,1264.0) (2777.0,1345.0) (3221.0,1527.0) (3080.0,1481.0) (3841.0,1632.0) (3942.0,1576.0) TIME=388.00ms - LONG TOUCH
TOUCH DETECTED! X=2154.0 Y=1024.0 (1932.0,1062.0) (2235.0,1031.0) TIME=139.00ms - LONG TOUCH
TOUCH DETECTED! X=2305.0 Y=1320.0 (2859.0,1383.0) TIME=108.00ms - LONG TOUCH
```

図 7-3. 長押し用の出力

8 まとめ

このアプリケーションノートでは、GPIO および ADC インターフェイスを使用して MSPM0L1306 マイコン上に抵抗膜方式タッチスクリーン制御を実装するための包括的な手順を説明します。制御手法、実装上の重要な考慮事項を文書化し、単純なタッチ検出ならびにクリックおよびドラッグ検出の動作するソフトウェア例を提供することで、このガイドは開発者が最小限の変更で抵抗膜方式タッチスクリーンの実装を移植できるようにします。

9 参考資料

- テキサス インスツルメンツ、[MSPM0L1306-RESISTIVE-TOUCH-APPSW](#)、ウェブページ。
- テキサス インスツルメンツ、『[MSPM0L130x ミックスド シグナル マイコン](#)』、データシート。

10 改訂履歴

Changes from Revision * (August 2025) to Revision A (September 2026)	Page
• クリック アンド ドラッグの例の詳細を追加。.....	1
• 「はじめに」に、抵抗膜方式タッチスクリーンと MSPM0 マイコンの詳細を追加。クリックとドラッグの例に関する詳細を追加.....	3
• ドキュメント全体にわたって表、図、相互参照の採番方法を更新.....	3
• YP および YN の 3 番目のピンを追加。選んだピンの関係でジャンパをもう 1 つ外すことになったため、Launchpad のジャンパの画像を更新し、その理由を追記しました。.....	4
• ソフトウェアの実装に使用したタッチスクリーンとセットアップに基づいて画像を更新。.....	5
• ピン構成の計算に関する情報と、ピンを追加または削除する際に考慮すべき注意事項を追加。.....	7
• パネル抵抗値の違いにより、Y プラスおよび Y マイナスを 3 ピンとした図に更新.....	9
• クリックおよびドラッグ検出の例と、ピンモード切り替えに関する詳細を追加.....	11
• サンプルを実行するための詳しい手順を追記.....	12
• 12 ビット ADC 分解能でのタッチ検出結果と、クリックおよびドラッグ検出の例を追加。.....	13

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含みいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](https://www.ti.com) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月