

抵抗膜式タッチパネル 変換基板キット 《取扱説明書》

■ 概要

本製品は、抵抗膜式タッチパネルの汎用キットです。タッチパネル本体と、端子形状変換用の基板とコネクタやケーブルなどをセットにしました。Arduinoをはじめとする各種 CPU ボードに接続するための汎用製品となっています。

■ 使用上の注意

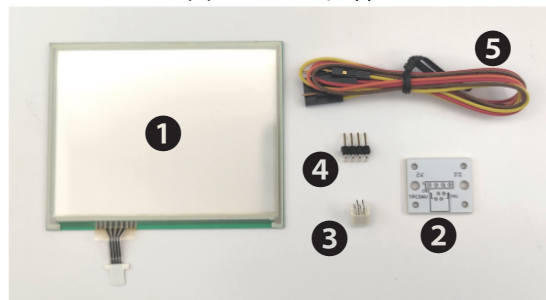
- **タッチパネルは非常に繊細な部品のため、無理な力を加えると破損する場合があります。** 曲げやねじりの力を加えたり、鋭い物体で突いたりしないでください。万一破損した場合はすみやかに使用を中止のうえ廃棄してください。破損したタッチパネルの修理を行うことはできません。
- **本製品の活用（タッチ位置の読み取り）には、基板への部品のハンダ付け作業、別途 CPU ボード（マイコンボード）の用意、制御用プログラムの作成が必要となります。** 本製品は汎用の接続形式となっており、特定の CPU ボードを対象を定めておりませんが、使用可否につきましてはユーザー様での検証・判断をお願いします。
- **タッチパネルの仕様上、2点以上の同時検出（マルチタッチ）には対応しません。** 検出面内の1点のみをタッチした場合に限り検出可能です。

1 セット内容

本製品のセット内容は下記の通りです（図1）。

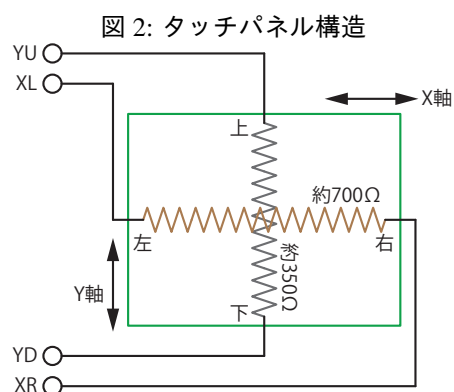
1	抵抗膜式タッチパネル サイズ：87.5 × 69.6 mm（外寸・実測）	1 枚
2	中継基板 [TPCONV] サイズ：24 × 20 mm （コネクタ要ハンダ付け）	1 枚
3	タッチパネル端子用 FPC コネクタ [CN1]	1 個
4	L型ヘッダーピン 4P [CN2]	1 個
5	QI ケーブル 4S-1Px4（型番：311-214） ケーブル長：約 30cm	1 本

図1: セット内容



2 タッチパネルの構造

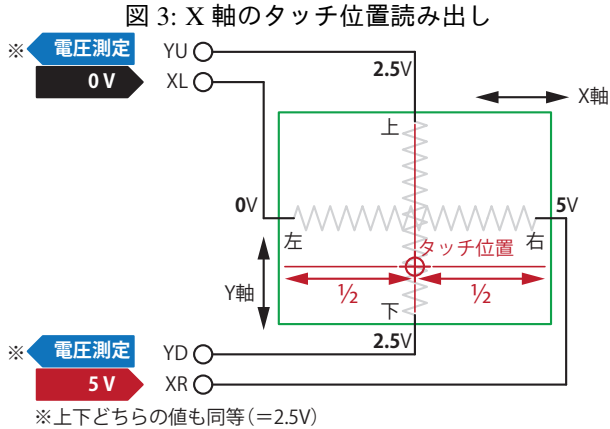
タッチパネルの構造は、横方向（X 軸）を検出する抵抗膜と縦方向（Y 軸）を検出する抵抗膜を重ねて、お互いが接触しない形で保持されています。この構造により、タッチするまでは縦検出用と横検出用の抵抗膜間は導通していません。横用の抵抗膜の左右端に電極が形成されていて、外部端子に接続されています。縦方向も同じ構造で上下端に電極があり、外部端子に接続されています。外部に引き出されている端子は合計4本です。横方向（X 軸）の抵抗値は 700 Ω 程度、縦方向（Y 軸）は 350 Ω 程度（いずれも実測値）です（図2）。



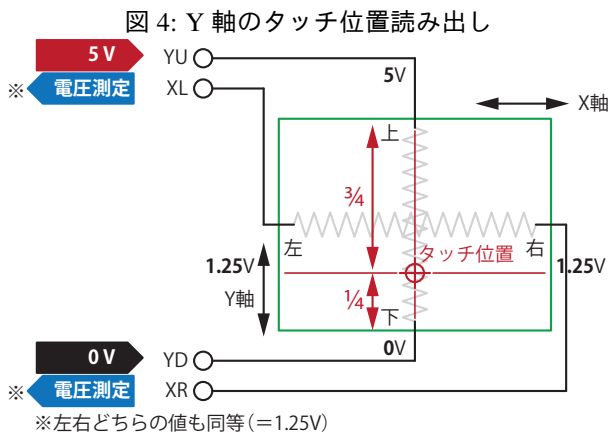
指などでパネルを押す事で、横用の抵抗膜と縦用の抵抗膜が接触します。この接触する位置に従って、左右または上下に加えた電圧が分割されます。ボリューム（可変抵抗）の両端に電圧を加え、可動接点から読み出している状態と同じです。ただし、タッチパネルの場合は可動接点にあたる端子は存在しないので、そのかわりに検出

中とは異なる方向の抵抗膜の端子を通して読み出します。例えば、タッチパネルの「横方向：中央、縦方向：下から1/4の場所」を押した場合を考えます。

横方向（X軸）の検出を行う場合は、左端子を0V、右端子を5Vとして横の抵抗膜に5Vの電圧をかけることで、上端子と下端子には、左端子（0V）と右端子（5V）の中間値である2.5Vが出力されます（図3）。



縦方向（Y軸）の検出を行う場合は、下端子を0V、右端子を5Vとして縦の抵抗膜に5Vの電圧をかけることで、左端子と右端子には、5Vの1/4である1.25Vが出力されます（図4）。



このように、横方向と縦方向を交互に切り替えながら端子電圧を読み込む事で、タッチ位置を取得することができます。上記の理由より、制御に使用するCPUボードにはデジタル入出力とアナログ入力の機能を備えたピンが必要となります。

このタッチパネルは、タッチ点が1箇所の場合のみ正しく位置を読み出す事ができます。2箇所以上の検出はできません（マルチタッチ不可）。

3 タッチパネルの方向と端子

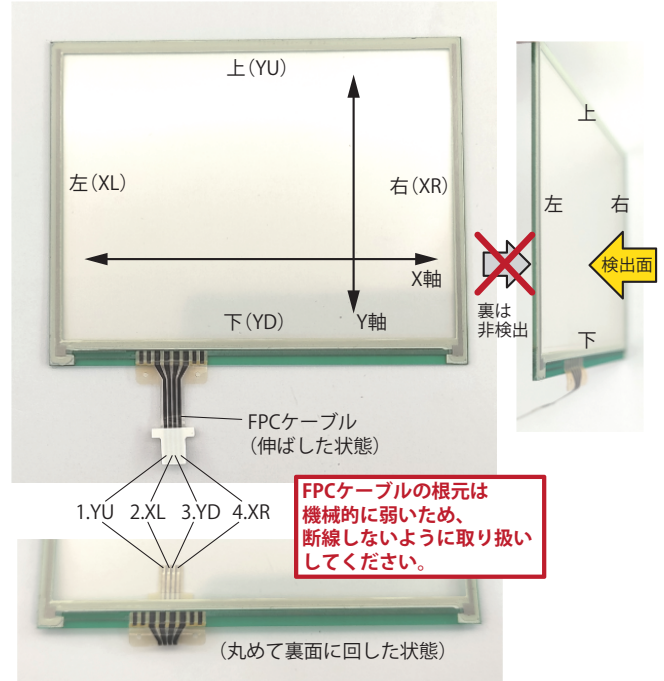
タッチパネルの方向と端子配列を（図5）に示します。

パネルの上下左右は、端子が左下に来るように向けた場合で表します。それ以外の方向で使用する場合は、上下左右を適宜読み替えてお使いください。

この状態でのパネルの手前側が検出面となります。裏面は検出面ではないため反応しません。

端子のFPCケーブルの電極は、片面（金属箔側）のみ導通があります。ケーブルを真っすぐ伸ばした場合、電極は裏側（非検出面）となります。

図5: タッチパネルの方向と端子



タッチパネルのFPCケーブルは非常に薄いため、引っ張りやねじれの力が加わると破損する場合があります。使用の際は必ず中継基板を使用し、FPCケーブル部に力が掛からないようにタッチパネル本体と中継基板の両方をしっかりと固定してください。《中継基板の組み立て（要ハンダ付け作業）》の項目もご参照ください。

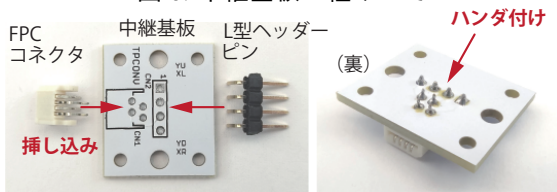
4 中継基板の組み立て（要ハンダ付け作業）

タッチパネルの端子（FPCケーブル）は、中継基板を使用してCPUボードに接続します。

中継基板は付属のコネクタ2種類をハンダ付けする必要があります。基板に黒い印刷がされている面にFPCコネクタ [CN1] とヘッダーピン [CN2] を挿し込み、裏側でハンダ付けしてください（図6）。

タッチパネルのFPCケーブルは耐熱性がないため、電線をFPCケーブルに直接ハンダ付けしないでください。中継基板を使わない場合は、FPCケーブルの電極に電線をテープで接着するなど、熱を加えない配線方法を検討し

図 6: 中継基板の組み立て

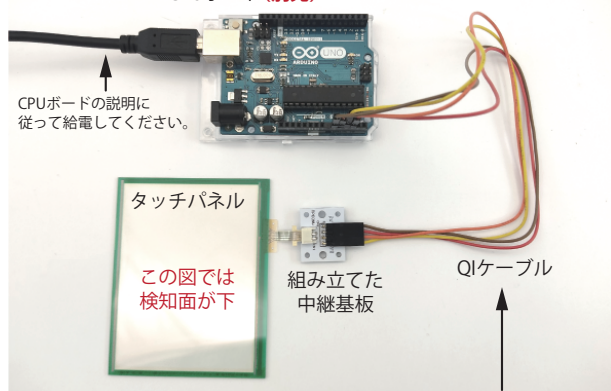


てください。

5 接続

タッチパネルー中継基板ー QI ケーブルー CPU ボードの順番に接続します (図 7)。

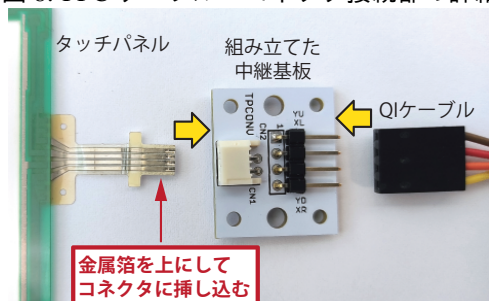
図 7: 接続例 (Arduino UNO の場合)
CPUボード (別売)



- ピンの接続先は、CPUボードの仕様に合わせて決定してください。
- 本文書内のサンプルスケッチを使用する場合の割り当ての詳細は、該当項目 ➡《Arduino用サンプルスケッチ》を参照してください。

FPC コネクタ [CN1] にタッチパネルの FPC ケーブルを挿し込む際の表裏にご注意ください (図 8)。FPC ケーブルの端子 (金属箔) が上を向くように挿し込みます。従って、(図 7) のタッチパネルは検知面が下向きになります。破損を防ぐため、FPC ケーブルは真っすぐ挿し込んでください。

図 8: FPC ケーブル・コネクタ接続部の詳細



ヘッダーピン [CN2] から CPU ボード側への接続については、タッチパネル側の信号線 4 本 (YU,XL,YD,XR) を、CPU ボードの入出力ピンに接続します。

使用する CPU ボード側の入出力ピンは「デジタル出力」「デジタル入力 (内蔵プルアップの有効無効選択)」「アナログ入力」の全てのモードに対応し、プログラム中で随時切り替えできる事が条件です (ほとんどのマイクロコントローラでは対応しています)。

この条件を満たすピンを 4 本割り当てて、それぞれにタッチパネルの信号線を 1 本ずつ配線します。QI ケーブルの CPU 側は先端が 1 ピンずつ分かれていますので、CPU ボード上のピンに自由に接続できます。

別項 ➡《Arduino 用サンプルスケッチ》の例をお試しになる場合は、項目内の接続に従ってタッチパネルと Arduino を接続してください。

6 Arduino 用サンプルスケッチ

▷ ダウンロード

Arduino UNO 用のサンプルスケッチは、下記ページにて配布しております。

【製品ページ】

<https://prod.kyohritsu.com/KP-TPCONV.html>



製品ページ内のリンク「参考用 Arduino スケッチ」をクリックして、ダウンロードしてください。動作確認済み環境は下記の通りです。

- CPU ボード : Arduino UNO
- 開発環境 : Arduino IDE (バージョン 1.8.19)

【プログラムに関するご注意】

上記ページ掲載のプログラムは無保証にて配布いたします。プログラムの内容に関するご質問への回答・サポート等を行っておりません。ご理解の上、参考目的としてご利用ください。

▷ 接続

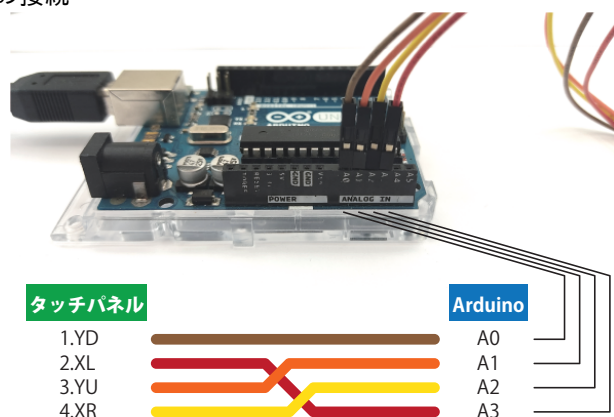
Arduino で最もスタンダードな「Arduino UNO」には全部で 20 本の入出力ピンがありますが、その内、アナログ入力対応でタッチパネルとの接続に使用できるピンは A0~A5 の 6 本となります。サンプルスケッチでは、A0, A1, A2, A3 に割り当てて振り分けています。スケッチの記述を変更することで使用するピンの配列を変更できま

す。ただし、Arduino UNO では A4 と A5 は I2C 通信に使用するため、I2C 接続デバイスも同時に使用する場合は A4, A5 を空ける必要があります。

各ピンの割り当ては下表および (図 9) の通りです。

タッチパネル端子	Arduino	QI ケーブル
YD (下)	A0	1 番ピン (茶色線)
YU (上)	A1	3 番ピン (橙色線)
XR (右)	A2	4 番ピン (黄色線)
XL (左)	A3	2 番ピン (赤色線)

図 9: サンプルスケッチにおけるタッチパネルと Arduino の接続



書き込み用の PC と Arduino を USB ケーブルで接続し、スケッチを Arduino に書き込んでください。

▷ スケッチの動作

読み取りルーチンの流れを説明します。まず最初に 4 本のピンを下記の状態に設定します。

- YU と YD … デジタル出力に設定し、LOW を出力
- XL … デジタル入力に設定 (プルアップ有効)
- XR … デジタル入力に設定 (プルアップ無効)

続いて、読み取りを実行します。まずはプルアップを有効化した XL ピンの状態を読み取り、タッチの有無を判定します。

- XL = LOW の場合、タッチ有り
- XL = HIGH の場合、タッチ無し

タッチ無しの場合は読み取り処理は終了となります。タッチ有りであった場合は、続けて具体的なタッチ位置を確認します。

- X 軸の値を取得

1. YU と YD をデジタル入力に設定 (プルアップ無効)、XL と XR をデジタル出力に設定し、XL は LOW を出力、XR は HIGH を出力。
2. 安定待ちの為、一定時間待機 (待機時間は、スケッチでは 5 マイクロ秒)
3. YD ピンのアナログ値を読み取り、結果を valX として保持
4. XL と XR の出力を反対に設定。XL は HIGH を出力、XR は LOW を出力。
5. 安定待ちの為、一定時間待機 (待機時間は、スケッチでは 5 マイクロ秒)
6. YD ピンのアナログ値を読み取り、アナログ入力の最大値である 1023 から結果の値を引いた反転値を valXa として保持。*

- 同様に、Y 軸の値を取得

1. XL と XR をデジタル入力に設定 (プルアップ無効)、YD と YU をデジタル出力に設定し、YD は LOW を出力、YU は HIGH を出力。
2. 安定待ちの為、一定時間待機 (待機時間は、スケッチでは 5 マイクロ秒)
3. XL ピンのアナログ値を読み取り、結果を valY として保持
4. YD と YU の出力を反対に設定。YD は HIGH を出力、YU は LOW を出力。
5. 安定待ちの為、一定時間待機 (待機時間は、スケッチでは 5 マイクロ秒)
6. XL ピンのアナログ値を読み取り、アナログ入力の最大値である 1023 から結果の値を引いた反転値を valYa として保持。*

- 結果を求めます。

- タッチ有りの場合、X 軸の結果を valX と valXa の平均値 (足して 2 で割った値) を求める。Y 軸についても同様に valY と valYa の平均値 (足して 2 で割った値) を求める。
- タッチ無しの場合、X 軸と Y 軸はそれぞれ 0,0 として返す。

- 今回のタッチ有無の判定結果を関数内部に保存しておきます。

* タッチ位置読み取りの際、出力電圧の HIGH と LOW を反対にして各軸 2 回ずつアナログ入力を行うのは、安定した読み出しを行うためです。どちらか一方だけで結果を取得しても問題ありません。

サンプルスケッチ内、readPanel 関数がタッチ位置読み取りのルーチンです。

```
bool readPanel(int *X, int *Y)
```

- 関数の戻り値は bool 型で、true = タッチ有り、false = タッチ無しを表します。
- 引数 X, Y は、読み取った交点座標の格納変数 (ポインタ) です。
 - タッチ無し (戻り値 = false) の場合は、X = 0, Y = 0 が格納されます。
 - タッチ有り (戻り値 = true) の場合は、交点座標を表す測定結果値が代入されます。Arduino UNO のアナログ入力は理論的に 0 ~ 1023 の範囲の値を取りますが、実際のパネルのタッチ時は、X = 約 60~960、Y = 約 90~930 の有効範囲になります。数値には個体差がありますので、利用する際は実物の値の範囲を必ず確認してください。

補足と注意点を下記に示します。

- X 軸・Y 軸とも、機械的な中央位置を表す値は、厳密に 512 (アナログ最大値 1023 の約半分) となりません。タッチパネルに形成された上下左右の電極と引き出しパターンに若干の抵抗値があるため、実際の位置とずれます。また CPU の電気的特性 (出力時の駆動能力や入力インピーダンス等) の影響も受けます。
- X または Y の値で 1023 が出力される場合は、配線間違いや接触不良、端子の断線が発生している可能性があります。
- Arduino UNO の場合、読み出しルーチンの実行時間は、タッチ有りの場合約 50 マイクロ秒かかります。タッチ無しの場合はアナログの測定を行わないため短時間で終了します。ただし、直前の呼び出し時にタッチ無し、かつ今回タッチ有り判定である場合、チャタリングを防ぐため約 10 ミリ秒の待機時間が追加されます。

サンプルスケッチの loop 内では、連続して readPanel を呼び出し、タッチ有りの場合に得られた結果の値をシリアルポート経由でプリントしています。スケッチ実行中に、PC と Arduino UNO を USB ケーブルで接続した状態で、Arduino IDE から「シリアルモニタ」を開き、パネル上をタッチすると、X 座標と Y 座標の読み取り値が随時表示されます (図 10)。

同じく Arduino IDE の「シリアルプロッタ」では、X 軸と Y 軸の時間ごとの経過をグラフ形式で表示できます (図 11)。

図 10: シリアルモニタ出力例 (Arduino IDE)

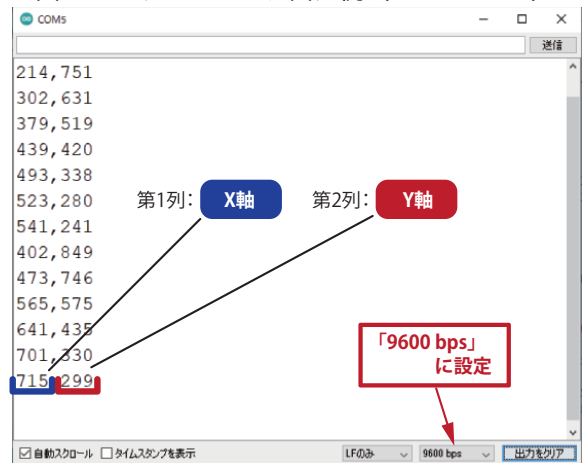
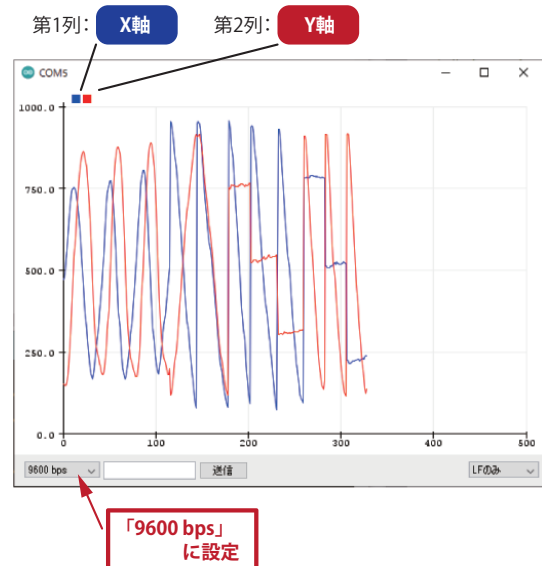
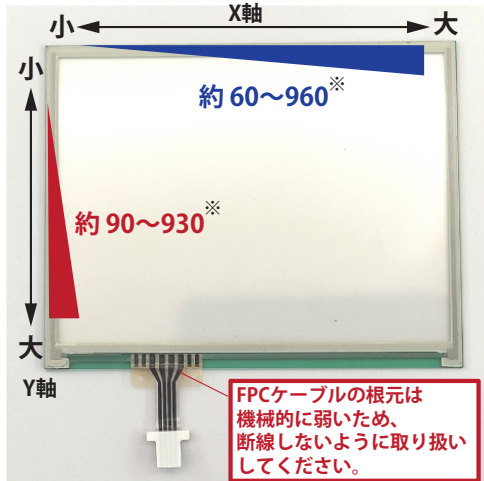


図 11: シリアルプロッタ出力例 (Arduino IDE)



サンプルスケッチの場合、X、Y 各軸の出力値の大小関係と実際のパネルとの対応は（図 12）の通りになります。

図 12: サンプルスケッチにおける出力値



※値の範囲には個体差があります。

※「1023」が出力される場合は配線間違い、接触不良、断線の可能性があります。

- ・本製品およびそれらを構成するパーツ類は、改良・性能向上のため予告なく仕様・外観等を変更する場合があります。あらかじめご了承ください。
- ・本製品は組立キットまたは半完成品です。製作作業中の安全確保のため説明書をよくお読みになり、正しい工具の使用・手順を守ってください。
- ・完成品でない商品の性格上、組み立て後の完璧な性能・品質・安全運用等の保証はできません。完成後はお客様（組立業者）ご自身の責任のもとでご使用ください。
- ・本製品は機器への組み込み他、工業製品としての使用を想定した設計は行っていません。また、本製品に起因する直接、間接の損害につきましては当社修理サポートの規定範囲を超えての補償には応じられません。

Electronic Devices, Parts, Kits & Robots

KYORITSU

共立電子産業株式会社 共立プロダクツ事業所

〒556-0005 大阪市浪速区日本橋5-8-26

TEL:06-6644-4447 FAX:06-6644-4448

【“共立プロダクツ”ブランドとは】

当ブランドの製品はユーザーニーズを捉えた製品をリーズナブルな価格でのご提供を目指しています。

そのためユーザーサポートはメールに限定しておりますことをご理解、ご了承ください。

✉ Email:wonderkit@keic.jp

Twitterやblogで応用例や製品紹介を更新中です。ぜひご覧になってください。

共立プロダクツ

検索