

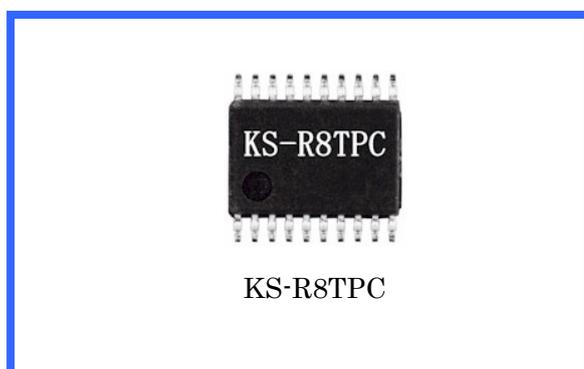
---

# タッチパネルコントローラ IC

## KS-R8TPC

### 補足マニュアル

2009年9月初版  
2009年10月改定



(注意) 上記写真のシルク印刷は合成です。事実とは多少異なる場合があります。

---

● はじめに ●

この度は、タッチパネルコントローラ IC(KS-R8TPC)をお買い上げ頂きまして誠にありがとうございます。本マニュアルは、各 LCD コントローラマニュアルの中で記述できていない内容をまとめさせていただきました。どうか LCD コントローラマニュアルと本マニュアルを熟読され、効率の良い開発にお役立て下さい。

● 重要なお知らせ ●

1. 本製品および本文書は、何らの通知無しに変更される場合があります。本製品をご使用になる前に、最新のカatalog、マニュアルなどを当方インターネット経由で取得して下さい。
2. 本製品は、直接に生命に関わる装置、原子力施設、航空機、交通機器、各種安全装置など製品の故障が直接に人の死亡、傷害、または重大な物理的もしくは環境上の損害を引き起こすようなシステム機器または装置に使用するために設計されたものではありません。本製品をこのようなシステム機器または装置に使用されることによる危険および損害は製品を使用されるお客様にご負担頂きます。
3. お客様が製品を誤った、または不当な方法で使用または操作された結果の損害につきましては、当方は一切責任を負いません。
4. 本文書に記載されている使用例は、単に本製品の機能を説明したものに過ぎません。当方は、本文書に記載されている例に基づいた使用により生じるかもしれない一切のクレーム、事故、その他一切の不利益に関して、何らの責任も負いません。

---

---

## 目次

---

1.	LCD コントローラ IC とタッチパネル	
	コントローラの構成	4 ページ
2.	製品の特長、概要	4 ページ
3.	タッチパネルコントローラ IC の機能	5 ページ
4.	LCD コントローラのレジスタについて	6 ページ
5.	A/D 変換値の活用例	7 ページ

---

## 1. LCD コントローラ IC とタッチパネルコントローラの構成

### 1) KS-Lxxx-2P,KS3224-LxxP シリーズ (LCD コントローラ IC)

ラティス社製 FPGA を採用しています。

LCD コントローラ IC は各シリーズによって VGA TFT 液晶、HVGA TFT 液晶や QVGA TFT 液晶などの表示をコントロールします。

詳しくは、各 LCD コントローラ IC のマニュアルをご参照ください。

### 2) KS-R8TPC (タッチパネルコントローラ)

2 チャンネル 8bitA/D 変換付き CPU デバイス (メーカー: ルネサス、型式: R5F21154SP-U00G) です。

(本 CPU のメーカー及び品番は、ピン互換品へ予告無く変更する場合があります。)

以上、2つのセットをお使い頂くことで、液晶の表示とタッチパネルのコントロールを実現します。

## 2. 製品の特長、概要

KS-R8TPC は、組込システム専用開発したタッチパネルコントローラ IC で、弊社 LCD コントローラとセットで使うことによって、タッチパネルが押された位置を 8 ビットの A/D 変換値で求めることができます。

また、専用ピン (XD,YD) の H/L を切り替えることにより A/D 変換値の増減方向を変更することができます。この設定によって、表示方向とタッチパネルのデータの増減方向を合わせる事が可能です。

その他、TOUCH (P1-2) 信号はタッチパネルが押されたとき、ロー信号を出力しますので、タッチパネルが押されたことを、割り込み等で検出することもできます。

---

### 3. タッチパネルコントローラICの機能

#### (1) A/D値の出力反転

ICのポートP3-4 (XD) , P3-5 (YD) は、プルアップ抵抗をお付けいただいておりますが、このポートをLにさせていただくと、LCDコントローラICのレジスタから読むことが出来るA/D値を反転して読むことが出来ます。

この設定によって、表示方向とタッチパネルのデータの方角を合わせる事が可能です。

#### (2) TOUCH (P1-2) 信号

TOUCH (P1-2) 信号はタッチパネルが押されたとき、ロー信号を出力します。LED等の点灯動作ができ、また、CPUのIRQ割り込みポートに接続すれば、タッチパネルが押されたことを、割り込みで検出することもできます。

## 4. LCDコントローラのレジスタについて

LCDコントローラICのコントロールレジスタは、そのICの種類によってレジスタアドレスが違いますので、各種LCDコントローラのマニュアルをご参照ください。

KS-R8TPCは8bitA/D変換機能を持っており、この変換結果をLCDコントローラICが受け取り、レジスタへ自動格納されます。サンプリングスピードは約5mS～8mSで常時行われており、コントロールレジスタ1, 2からいつでも読み出すことができます。

本機能によりアナログジョイスティックやアナログタッチパネル、その他のアナログセンサ類のインターフェースを余分なハードウェア無しに実現出来ます。

### (1) コントロールレジスタ1 (DCR1)

アドレス 1FFFCH, 3FFFCH, 7FFFCH (ADX) (読み出し側)

ビット	7	6	5	4	3	2	1	0
名前	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
R/W	R	R	R	R	R	R	R	R
初期値	-	-	-	-	-	-	-	-

KS-R8TPCは8bitA/D変換機能を持っており、この変換結果をLCDコントローラICが受け取り、レジスタへ自動格納されます。KS-R8TPCのP1-0ポートから取得されたA/D変換値を、上記レジスタからいつでも読み出すことができます。

### (2) コントロールレジスタ2 (DCR2)

アドレス 1FFFDH, 3FFFDH, 7FFFDH (ADY) (読み出し側)

ビット	7	6	5	4	3	2	1	0
名前	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
R/W	R	R	R	R	R	R	R	R
初期値	-	-	-	-	-	-	-	-

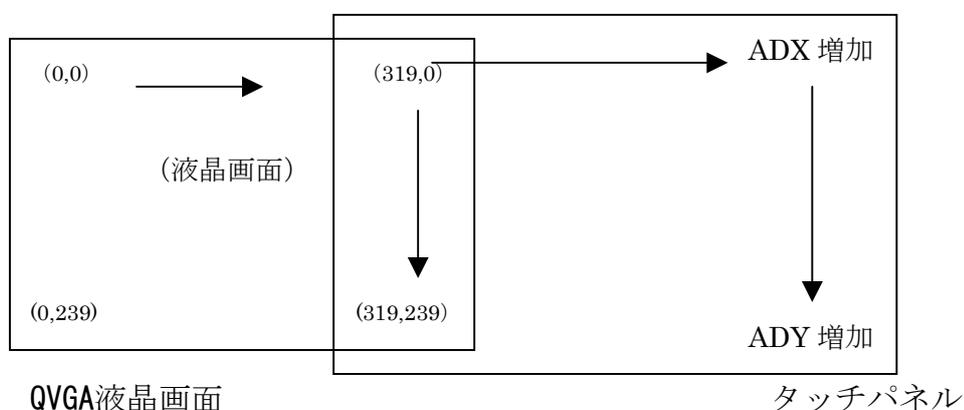
本レジスタの機能もコントロールレジスタ1と同じく、A/D変換結果を読み出せます。KS-R8TPCのP1-1ポートから取得されたA/D変換値を、上記レジスタからいつでも読み出すことができます。

## 5. A/D変換値の活用例

LCDのサイズをQVGAで説明しておりますが、お使いのLCDコントローラICやLCDによって異なりますので、ご注意ください。

### (1) タッチパネルとLCD画面のマッチング

LCDにも向きがあるようにタッチパネルにも上下左右の向きがあります。LCDによっては表示したい方向とタッチパネルの向きが合わない場合がありますので、初めにLCDの表示方向とタッチパネルコントローラICから読み出されるA/D変換値の向きを合わすようにします。

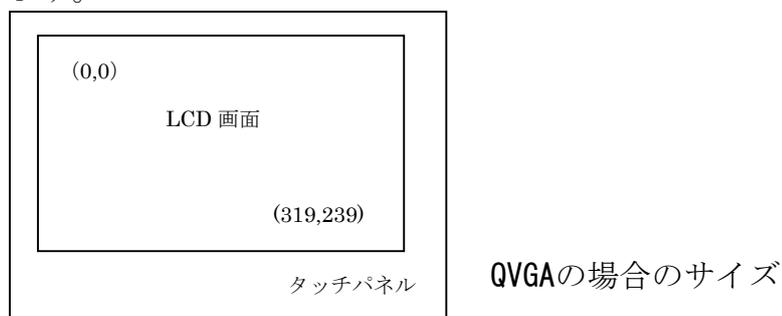


弊社のLCDコントローラICは、LCDの左上がフレームバッファの先頭番地でそこから順に右へ並んでおり、最後が右下になります。

よって、コントロールレジスタから読まれるA/D値もADXが右方向に増加、ADYが下方向に増加するようにXD, YDをハード的に反転させるか、コントロールレジスタから読んだA/D値を反転させて合わせます。

### (2) ドット座標変換

LCDの表示方向とタッチパネルからのA/D値の増加方向が一致したら、次にサイズを調整します。



---

元々、LCDが表示できるサイズとタッチパネルが反応できるサイズは異なっています。よって、タッチパネルが押される位置とLCDのドット位置が一致していれば、実際の活用には便利です。

まず (0, 0) 位置と (319, 239) 位置で、実際のA/D値がいくらになっているのかを調べます。

(0, 0) 位置のA/D値 (ADX\_MIN, ADY\_MIN)

(319, 239) 位置のA/D値 (ADX\_MAX, ADY\_MAX)

その値の範囲内が有効値になるので、逸脱した場合は、押していないものと見なせます。

取得範囲が決まったら座標変換を行ないましょう。

<計算例>

```
ucSet_AD_x /* ADXからのA/D値とします*/
```

```
ucSet_AD_y /* ADYからのA/D値とします*/
```

```
/* AD変換値の範囲をチェックし、範囲外なら押されていないと見なします*/
```

```
if(ucSet_AD_x < ADX_MIN) ucRet = TPB_STATUS_OPEN;
```

```
if(ADX_MAX < ucSet_AD_x) ucRet = TPB_STATUS_OPEN;
```

```
if(ucSet_AD_y < ADY_MIN) ucRet = TPB_STATUS_OPEN;
```

```
if(ADY_MAX < ucSet_AD_y) ucRet = TPB_STATUS_OPEN;
```

```
/* 開放状態だったら通常ありえない座標値を入れておきます*/
```

```
if(ucRet == TPB_STATUS_OPEN) {
```

```
    uiX = 0xffff;
```

```
    uiY = 0xffff;
```

```
}else{
```

```
/* 座標に変換*/
```

```
    uiX = (ucSet_AD_x - ADX_MIN) * 319 / (ADX_MAX - ADX_MIN);
```

```
    uiY = (ucSet_AD_y - ADY_MIN) * 239 / (ADY_MAX - ADY_MIN);
```

```
}
```

このようにして得られる値が押された時の座標 (uiX, uiY) です。