Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN2376

PSoC® 1 – Interface to Four-Wire Resistive Touchscreen

Author: Svyatoslav Paliy, Andrij Bilynskyy
Associated Project: Yes
Associated Part Family: CY8C24x94
Software Version: PSoC® Designer™ 5.4 CP1
Related Application Notes: None

Abstract

AN2376 shows how PSoC 1 can be used to control a resistive touchscreen, and read (x,y) positions of single touches as well as touch pressure. It describes the essential mathematics in detail, and includes a method for calibrating a touchscreen to a display. A project is included that passes touchscreen readings to a PC through a USB HID interface.

Contents

Introduction

Touchscreen interfaces are effective in many information appliances, in personal digital assistants (PDAs), and as generic pointing devices for instrumentation and control applications. This application note describes resistive types of touchscreens. Their construction is simple, their cost is low, and their operation is well understood by users. The only concern is that the resistive layers can be damaged by very sharp objects. This document considers the basic principles of how resistive touchscreens work and how to best convert these analog inputs into usable digital data using a PSoC.

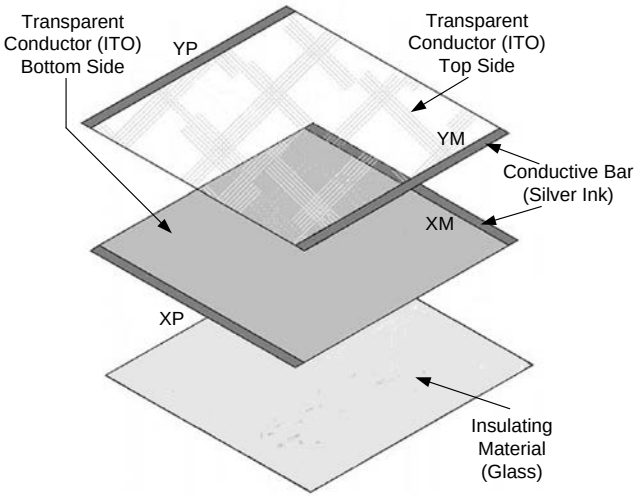
When developing more complex touch-assisted projects; there is often a need for additional peripheral units, such as operational and instrument amplifiers, filters, timers, digital logic circuits, AD and DA convertors. As a general rule, implementation of these extra peripherals brings in additional difficulties: space for new components, additional attention during production of a printed circuit board, power consumption increase. All of these factors can significantly affect the project price and development cycle.

PSoC has made many engineers' dreams come true; that is, to have all their project needs covered in one chip.

Construction of Resistive Touchscreens

Resistive touchscreens are constructed as shown in Figure 1. They consist of two top transparent glass or acrylic panels and a bottom transparent insulating panel. Both top panels are coated with electrically conductive layers. These layers have uniform resistance made from indium tin oxide (ITO). The panels are separated by invisible non-conducting spacers.

Figure 1. Resistive Touchscreen Construction



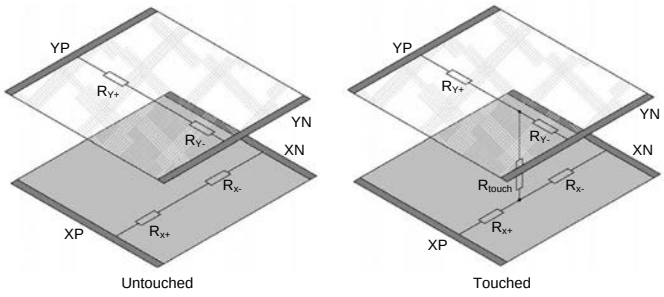
The top layer panel is flexible and the bottom layer panel is rigid. Pressing the flexible top sheet creates an electrical contact between the resistive layers, essentially closing a switch in the circuit. The four electrical wires are connected to conductive bars.

Resistive Touchscreen Measurement Method

A touch can be defined by three parameters. The first and second parameters are X-position and Y-position. The third parameter relates to “touch pressure” and allows the touchscreen to differentiate between finger and stylus contacts.

The conductive bars are located on the opposite edges of the panel. The layer has uniform resistance; that is, the voltage applied to the layer produces a linear gradient across this layer. Both layers are orthogonal to each other, and voltage gradients in layers are orthogonal, too. In an equivalent circuit of a resistive touchscreen, we can offer conductive layers as resistors between the conductive bars in the corresponding layers. When we press a touchscreen, we get a connection between the resistive layers. In a circuit, we offer this connection as a resistor, too. Equivalent circuits for touched and untouched touchscreens are shown in Figure 2.

Figure 2. Touchscreen Equivalent Circuits

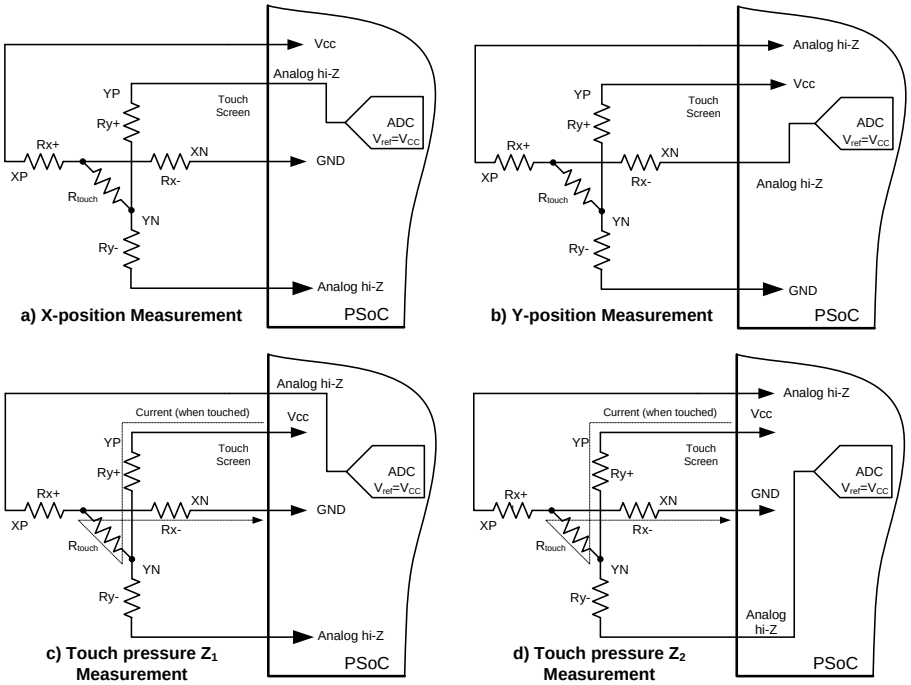


To measure a 4-wire touch sensor, a V_{CC} voltage is applied to a conductive bar on one of the layers while the other conductive bar on the same layer is grounded, see Figure 3a. In this powered layer we have a linear voltage gradient (0.. V_{CC}). One of the conductive bars in the other layer is connected to an ADC through large impedance. The ADC reference is set to V_{CC}, which makes the ADC range (0..max ADC value). When the screen is touched, the ADC reading corresponds to the position on one of the axes.

To obtain the second coordinate, the other layer must be powered and read by the ADC. V_{CC}, GND, Analog hi-Z, and ADC input are switched between the two layers, as shown in the Y-position measurement in Figure 3. The second ADC reading corresponds to the position on the other axis.

Finally, to obtain the touch pressure, we take two measurements of the cross-layer resistance. V_{CC} is applied to a conductive bar on one of the layers while a conductive bar on the other layer is grounded. We measure the voltages on the unconnected bars, shown in depictions c and d in Figure 3.

Figure 3. Various Parameter Measurements



Now it's possible to develop a mathematical equation to calculate the cross-layer resistance, from our ADC results X, Y, Z_1, Z_2 (see Figure 3).

From Figure 3a, you can write:

$$\frac{X}{AD_{\max}} = \frac{V_{in}}{V_{ref}} = \frac{V_{in}}{V_{cc}} = \frac{iR_{x-}}{i(R_{x-} + R_{x+})} = \frac{R_{x-}}{R_{x-} + R_{x+}} = \frac{R_{x-}}{R_{x\_plate}}$$

Equation 1

Where,

X = ADC value when ADC input voltage is equal V_{in}

$AD_{max}=2^{ADC\_resolution}$ –- Maximum ADC value + 1

V_{ref} – Reference ADC voltage, for measurements using $V_{ref} = V_{cc}$

$R_{x\_plate} = R_{x-}+R_{x+}$ – X-plate resistance

By analogy from Figure 3b:

$$\frac{Y}{AD_{\max}} = \frac{R_{y-}}{R_{y-} + R_{y+}} = \frac{R_{y-}}{R_{y\_plate}}$$

Equation 2

Where,

$R_{y\_plate} = R_{y-}+R_{y+}$ – Y-plate resistant

From Figure 3c:

$$\frac{Z_1}{AD_{\max}} = \frac{R_{x-}}{R_{x-} + R_{touch} + R_{y+}}$$

Equation 3

And from Figure 3d:

$$\frac{Z_2}{AD_{\max}} = \frac{R_{x-} + R_{touch}}{R_{x-} + R_{touch} + R_{y+}}$$

Equation 4

You can then obtain the cross-layer resistance that represents touch pressure, from Equations 1, 3, and 4:

$$R_{touch} = R_{x\_plate} \cdot \frac{X}{2^{ADC\_resolution}} \left(\frac{Z_2}{Z_1} - 1 \right)$$

Equation 5

And the other cross-layer resistance, which also represents touch pressure, from Equations 1, 2, and 3:

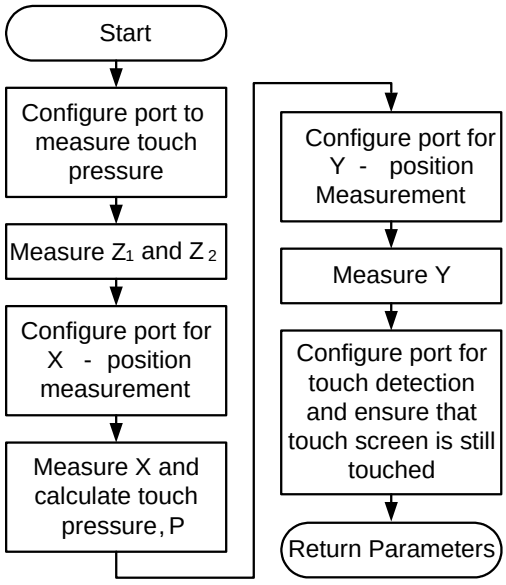
$$R_{touch} = R_{x\_plate} \cdot \frac{X}{2^{ADC\_resolution}} \left(\frac{2^{ADC\_resolution}}{Z_1} - 1 \right) - R_{y\_plate} \left(1 - \frac{Y}{2^{ADC\_resolution}} \right)$$

Equation 6

Equation 5 requires a known X-plate resistance, measurements of X-position (X), and two additional cross-panel measurements (Z_1 and Z_2) of the touchscreen Equation 6 requires known X- and Y-plate resistance values but only allows measurement of Z_1. For calculation touch pressure, you can use one of those equations. In the real projects, you must have three touchscreen parameters such as X-position, Y-position, and pressure. To calculate touch pressure, you can use either of these equations. Equation 6 requires three ADC measurements while Equation 5 requires four. Using Equation 6 is faster, but its arithmetic is more complicated.

A flowchart for the touchscreen parameter measurement routine is shown in Figure 4.

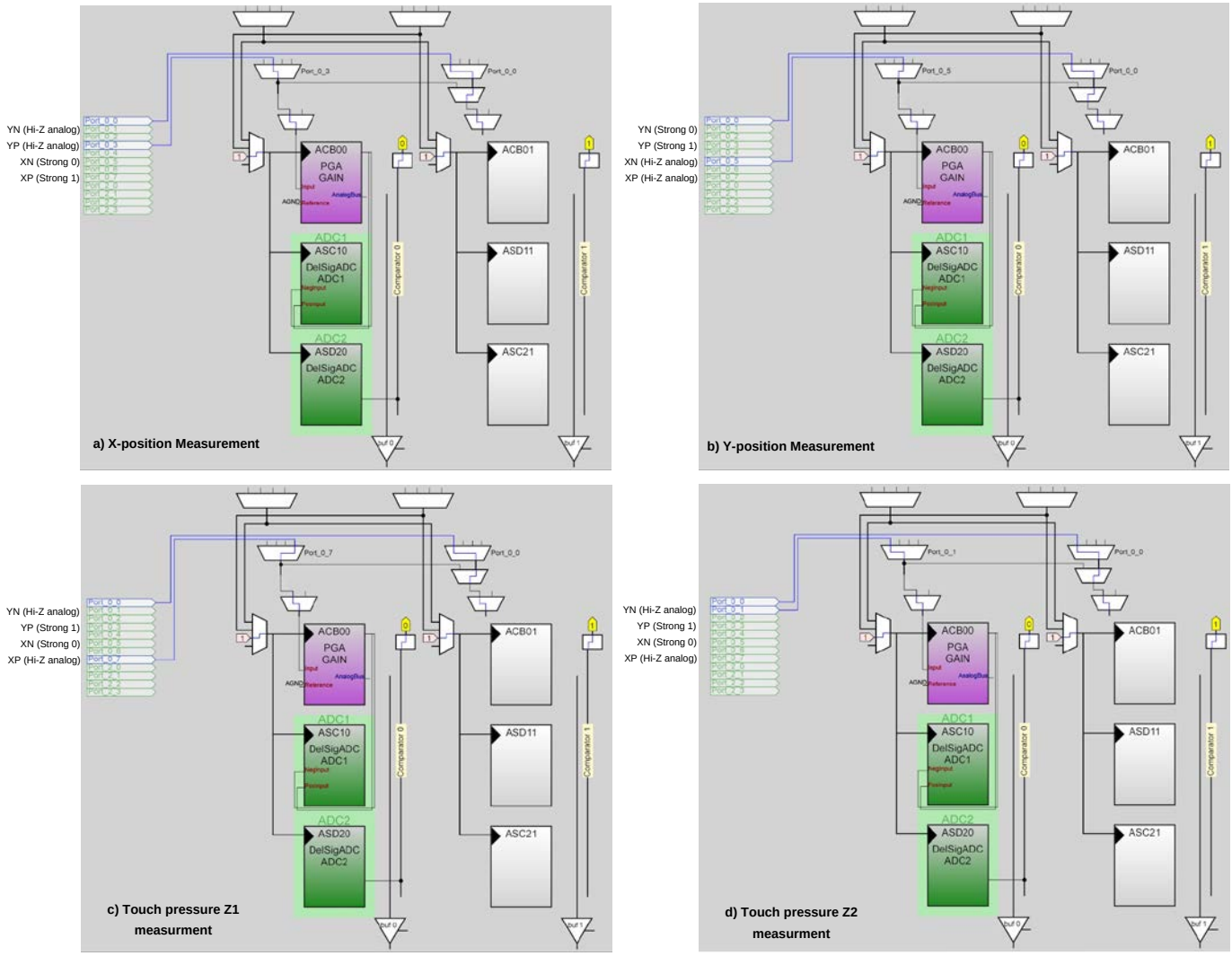
Figure 4. General Touchscreen Parameter Measurement Routine Flowchart



PSoC Implementation

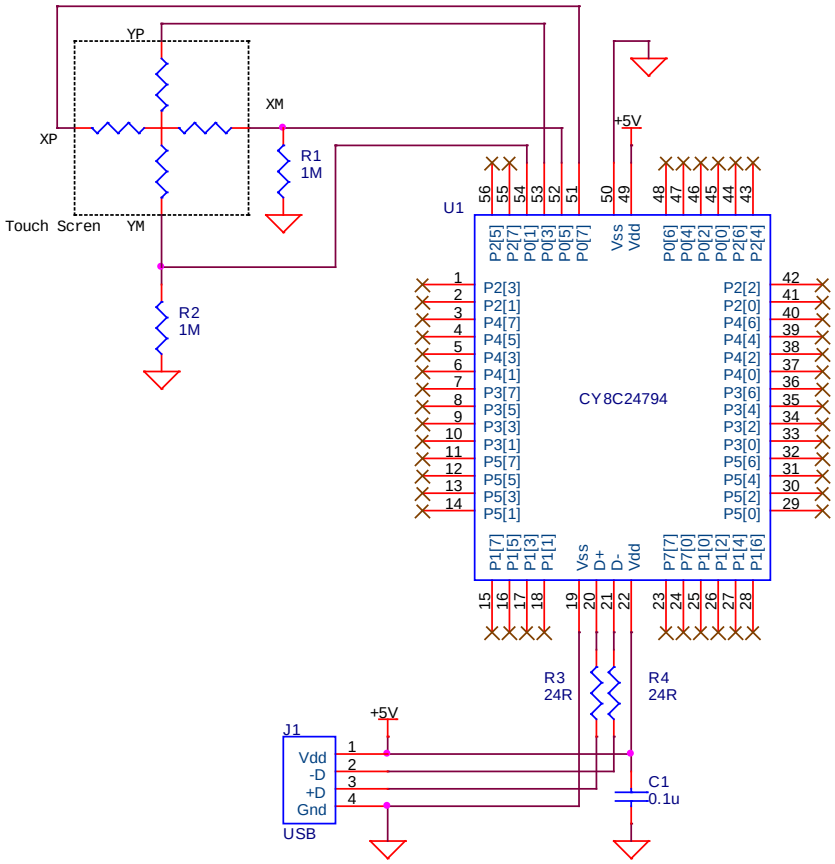
The PSoC internal structure is very simple (see Figure 5). Pins P0[1], P0[3], P0[5], and P0[7] are used for the touch panel interface. Signals from the input pins are switched through the analog column input multiplexer to the PGA, and then to the delta-sigma ADC.

Figure 5. PSoC Configurations for each of the Measurement Diagrams in Figure 3



The PSoC-based schematic is also very simple (see Figure 6). The touchscreen is directly connected to the PSoC device ports. Resistors R1 and R2 are used as pull downs to terminate the PSoC inputs when the plates are not driven.

Figure 6. Device Schematic



To debug this project, measured data is transferred to the PC through the USB interface. The USB HID device class is used to simplify data transfer to the PC and to avoid writing a separate USB driver for the PC. The PSoC device sends a data packet to the PC that contains four 16-bit parameters. Table 1 shows the data packet structure. A simple PC application has been developed for evaluation (Appendix C).

Table 1. Data Packet Structure

Number	Offset in Bytes	Length in Bytes	Parameter Name
1	0	2	X touch coordinate
2	2	2	Y touch coordinate
3	4	2	Z_1
4	6	2	Z_2

Power Consumption (Pen Interrupt)

PSoC devices have very little power consumption in sleep mode. If a touchscreen has not been touched for a long period of time, there is no need for operation or measurement. The touchscreen is allowed into sleep mode and waits for a pen interrupt, at which time the controller wakes up and measures the touch parameters.

To use this feature (see Figure 7), apply a voltage to the touchscreen XP connector through resistors with the resistance exceeding the maximal cross-panel touchscreen resistance in the touched mode by four or more times. In the proposed design, cross-panel touchscreen resistance values are about 1K. An internal pull-up resistor $R_{pull\ up}$ with a resistance of 5.6 K can be used. The port pin that is connected to the touchscreen XP connector is configured to pull-up mode and set to a logic state of high. The port pin connected to the XN connector is configured as a digital input and the pin-interrupt is enabled by a falling edge. If the touchscreen is still untouched, XN holds the logic in a high state, while if the touchscreen is touched; the voltage of XN falls to a low logic level and initiates an interrupt that wakes up the PSoC. A flowchart for the touchscreen controller performance using sleep mode is shown in Figure 8.

Figure 7. Touchscreen Configured for a Waiting Pen Interrupt

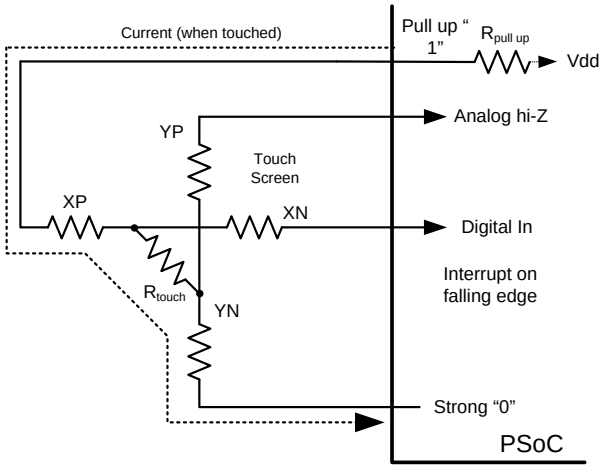
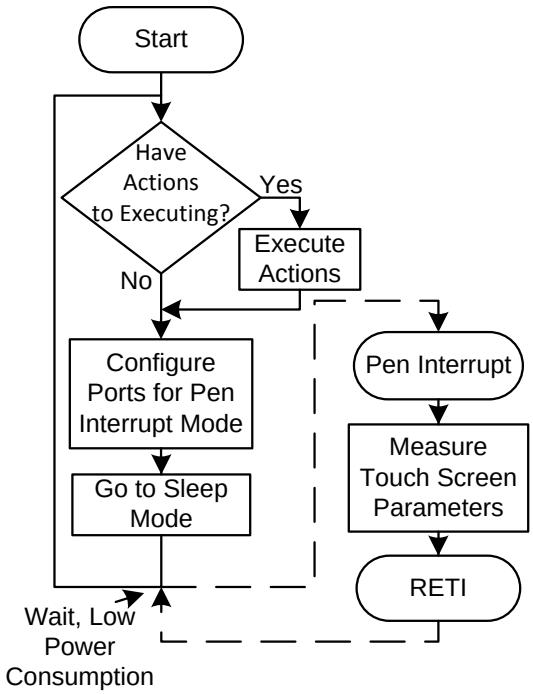


Figure 8. General Touchscreen Controller Routine Flowchart Using Sleep Mode



Touchscreen Calibration

In most cases, the touchscreen is mounted over an LCD or another display. If this is the case, the data measured by the touchscreen must be translated into true screen coordinates.

The calibration algorithm proposed in this application note allows eliminating scaling factors and mechanical misalignment of the touchscreen. The challenge for the calibration algorithm is to translate the set of coordinates reported by the touchscreen into a set of coordinates that accurately represents a point on the display.

$$[X_D, Y_D] = \vec{f}([X_T, Y_T])$$

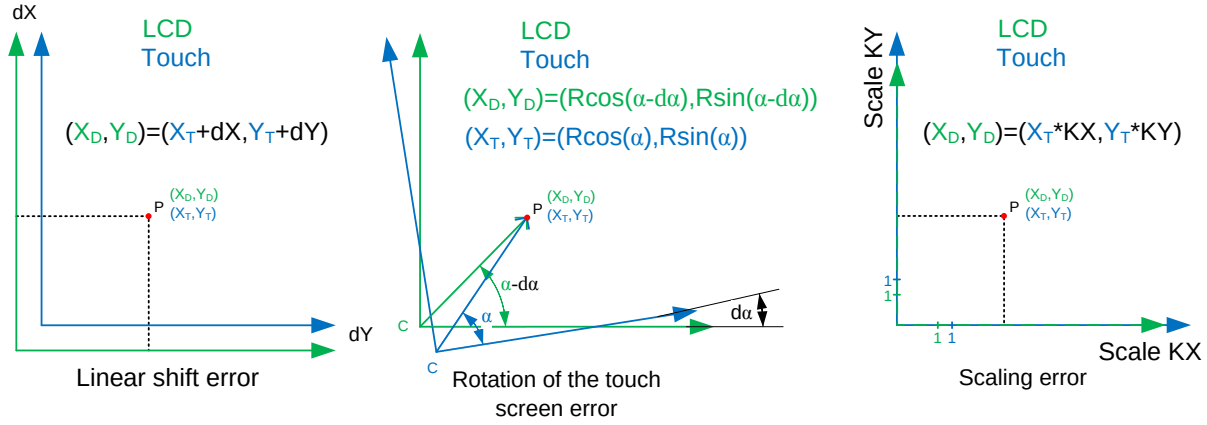
Equation 7

Figure 9 shows some misaligned touchscreen and display coordinates. This figure also shows that it is possible to describe each point on the display as *PD* = [*XD*,*YD*] and each point on the touchscreen as *PT* = [*XT*,*YT*].

Three error factors are presented:

- Rotation of the touchscreen coordinates relative to the display coordinates.
- Linear shift of the coordinates.
- A scaling factor.

Figure 9. Error Factors



The LCD coordinate X*D* can be expressed as:

$$\begin{aligned} X_D &= K_x R\cos(\alpha - d\alpha) + dX \\ &= K_x R\cos\alpha\cos d\alpha + K_x \sin\alpha \sin d\alpha + dX \\ &= K_x X_T \cos d\alpha + K_x Y_T \sin d\alpha + dX \\ &= \alpha_x X_T + \beta_x Y_T + dX \end{aligned}$$

Equation 8

$$\begin{aligned} Y_D &= K_y R\sin(\alpha - d\alpha) + dY \\ &= K_y R\sin\alpha\cos d\alpha - K_y \cos\alpha \sin d\alpha + dY \\ &= K_y Y_T \cos d\alpha - K_y X_T \sin d\alpha + dY \\ &= \alpha_y X_T + \beta_y Y_T + dY \end{aligned}$$

Equation 9

Where,
$\alpha_x = K_x\cos(d\alpha)$, $\beta_x = K_x\sin(d\alpha)$, $\alpha_y = -K_y\sin(d\alpha)$, and $\beta y = K_y\cos(d\alpha)$.

From Equations 8 and 9, when there are three independent points (not on one line), you can get the coefficients α_x, α_y, β_x, β_y, dX, and dY. Assuming that (X_{D1}, Y_{D1}), (X_{D2}, Y_{D2}), and (X_{D3}, Y_{D3}) are three independent points selected on the LCD, and (X_{T1}, Y_{T1}), (X_{T2}, Y_{T2}), and (X_{T3}, Y_{T3}) are the corresponding points on the touchscreen, Equations 8 and 9 can be used to write Equation 10:

$$
\begin{aligned}
X_{D1} &= \alpha_x X_{T1} + \beta_x Y_{T1} + dX \\
X_{D2} &= \alpha_x X_{T2} + \beta_x Y_{T2} + dX \\
X_{D3} &= \alpha_x X_{T3} + \beta_x Y_{T3} + dX \\
Y_{D1} &= \alpha_y X_{T1} + \beta_y Y_{T1} + dY \\
Y_{D2} &= \alpha_y X_{T2} + \beta_y Y_{T2} + dY \\
Y_{D3} &= \alpha_y X_{T3} + \beta_y Y_{T3} + dY
\end{aligned}
$$

Equation 10

Or in matrix form:

$$
\begin{pmatrix} X_{D1} \\ X_{D2} \\ X_{D3} \end{pmatrix} = A \times \begin{pmatrix} \alpha_x \\ \beta_x \\ dX \end{pmatrix} \quad and \quad \begin{pmatrix} Y_{D1} \\ Y_{D2} \\ Y_{D3} \end{pmatrix} = A \times \begin{pmatrix} \alpha_y \\ \beta_y \\ dY \end{pmatrix} \qquad \text{Where, } A = \begin{pmatrix} X_{T1} & Y_{T1} & 1 \\ X_{T2} & Y_{T2} & 1 \\ X_{T3} & Y_{T3} & 1 \end{pmatrix}
$$

From Equation 10:

$$
\begin{pmatrix} \alpha_x \\ \beta_x \\ dX \end{pmatrix} = A^{-1} \times \begin{pmatrix} X_{D1} \\ X_{D2} \\ X_{D3} \end{pmatrix} \quad and \quad \begin{pmatrix} \alpha_y \\ \beta_y \\ dY \end{pmatrix} = A^{-1} \times \begin{pmatrix} Y_{D1} \\ Y_{D2} \\ Y_{D3} \end{pmatrix}
$$

Equation 11

Where,

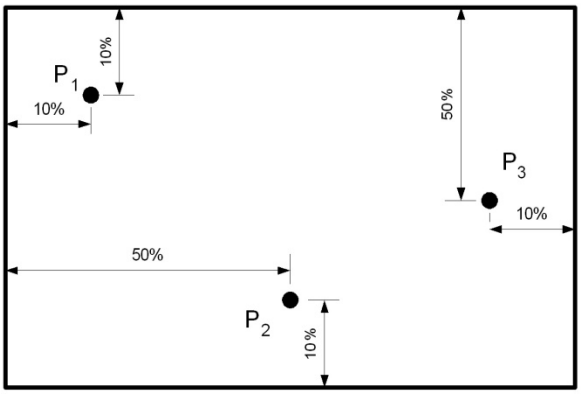
A^{-1} is the inverse of matrix A. The three points—(X_{D1}, Y_{D1}), (X_{D2}, Y_{D2}), and (X_{D3}, Y_{D3})—are selected on the display surface, and the elements in matrix A are measured from the touchscreen during calibration.

The 3-points calibration's equations, in a non-matrix form, and source code for it, optimized for 8-bit cores, are described in Appendix D.

For best results, the first sample point must be located at the distance of approximately 10% of the screen size in the upper-left corner. The second and third points must be at a distance of approximately 10% of the screen size from the screen edges in the center of each side (see Figure 10). For the calibration process, points 1, 2, and 3 must be touched in an ascending order.

Calibration must be performed before use for each device that contains a touchscreen. There is no need to perform calibration every time the device is powered on. However, it is prudent to save the calibration parameters in the energy-independent memory. Every time the device is used after calibration, it reads these coefficients and uses them to calculate the true screen coordinates.

Figure 10. Calibration Point's Placement on Screen



In many applications, users may use more than three points in their calibration routines to average or filter the noisy readings from the touchscreen controller. For calibration with more than three points:

$$\begin{pmatrix} X_{D1} \\ X_{D2} \\ \cdot \\ \cdot \\ \cdot \\ X_{DN} \end{pmatrix} = A \times \begin{pmatrix} \alpha_x \\ \beta_x \\ dX \end{pmatrix} \quad and \quad \begin{pmatrix} Y_{D1} \\ Y_{D2} \\ \cdot \\ \cdot \\ \cdot \\ Y_{DN} \end{pmatrix} = A \times \begin{pmatrix} \alpha_y \\ \beta_y \\ dY \end{pmatrix} \quad where \quad A = \begin{pmatrix} X_{T1} & Y_{T1} & 1 \\ X_{T2} & Y_{T2} & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ X_{TN} & Y_{TN} & 1 \end{pmatrix}$$

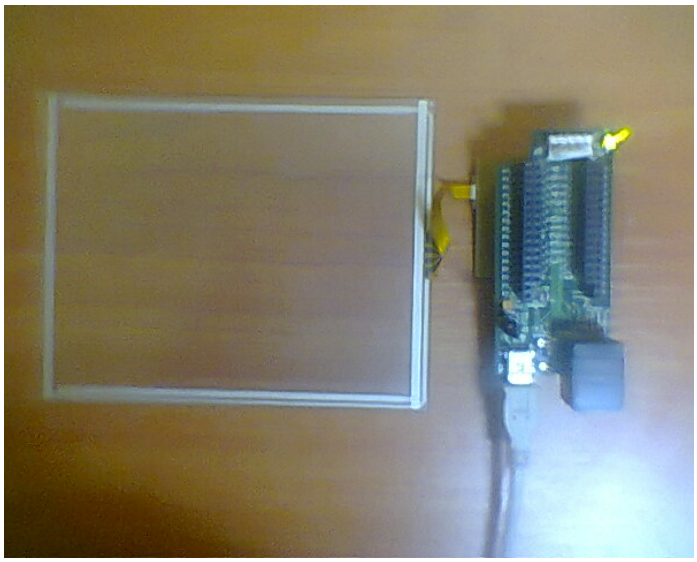
Equation 12

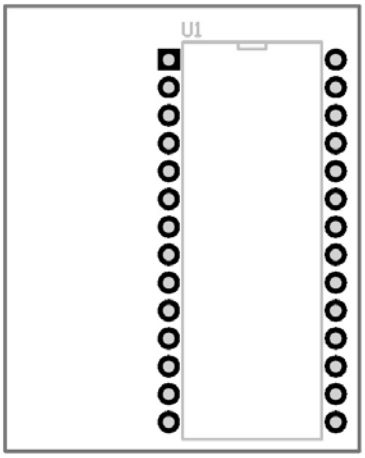
To resolve Equation 12, both sides can be multiplied by A's pseudo-inverse matrix, $(A^T \times A)^{-1} \times A^T$, where A^T is A's transpose matrix.

$$\begin{pmatrix} \alpha_x \\ \beta_x \\ dX \end{pmatrix} = \left(A^T \times A\right)^{-1} \times A^T \times \begin{pmatrix} X_{D1} \\ X_{D1} \\ \cdot \\ \cdot \\ \cdot \\ X_{DN} \end{pmatrix} \quad and \quad \begin{pmatrix} \alpha_y \\ \beta_y \\ dY \end{pmatrix} = \left(A^T \times A\right)^{-1} \times A^T \times \begin{pmatrix} Y_{D1} \\ Y_{D1} \\ \cdot \\ \cdot \\ \cdot \\ Y_{DN} \end{pmatrix}$$

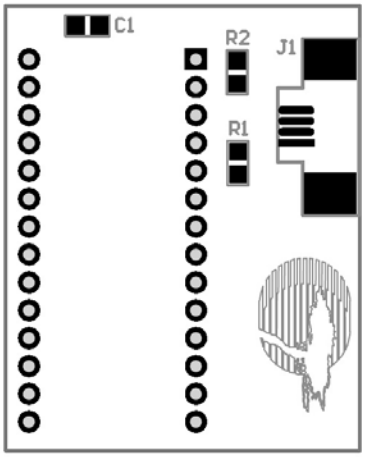
Equation 13

Appendix A: Hardware Setup (Not Actual Size)

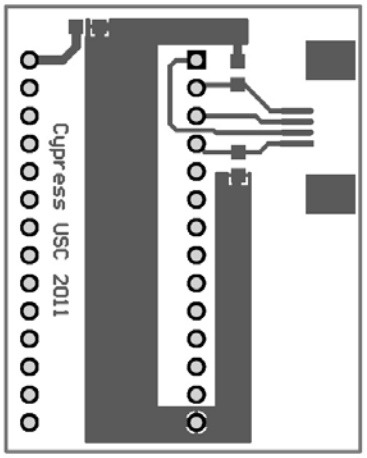




Top Layer Components



Bottom Layer Components

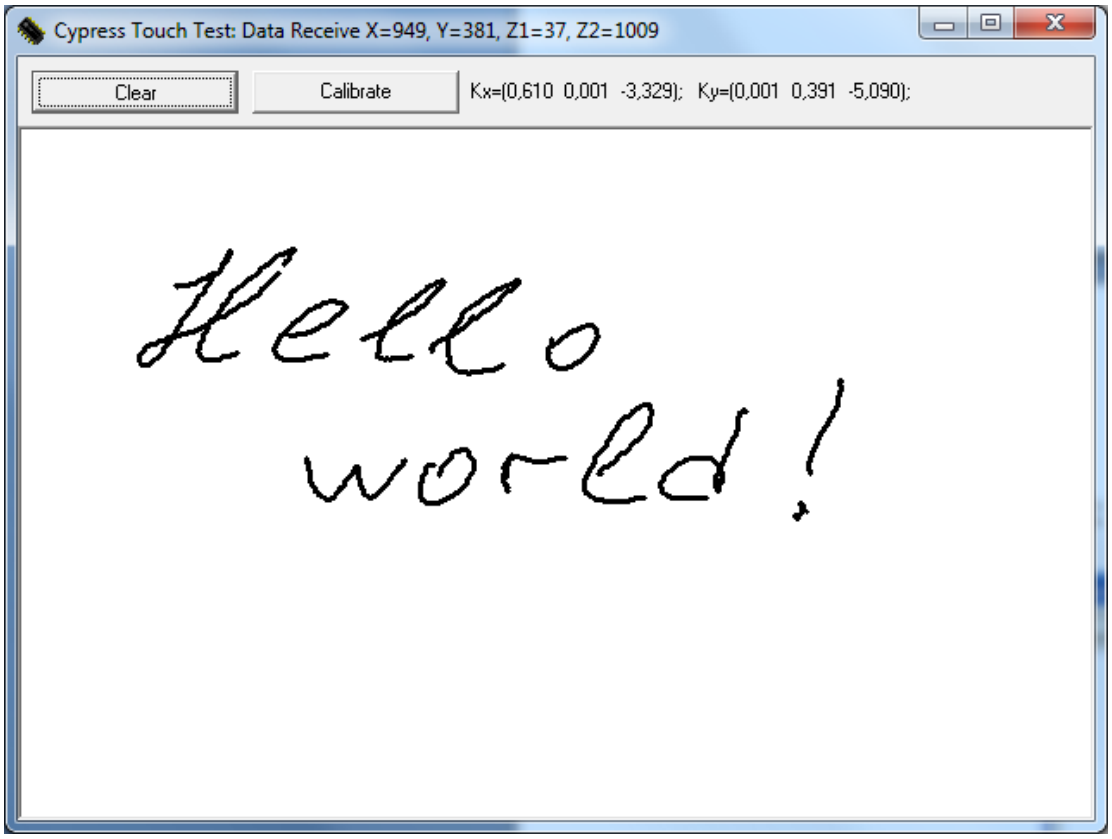


Bottom Layer Wires

Appendix B: Board Bill of Materials

Description	Count	Designator	Manufacturer	Manufacturer Part Number	Digi-Key Part Number
EVALUATION POD FOR CY8C24X94	1	U1	Cypress	CY3210-24X94	428-1998-ND
Touchscreen	1	-	Bergquist	400426	BER275-ND
Connector	1	J1	Molex Connector Corporation	52271-0479	WM7954TR-ND
Connector	2	U1 sockets	FCI	95200-301LF	95200-301LF-ND
Resistors (R1, R2)	2	R1, R2	Panasonic - ECG	ERJ-6GEYJ105V	P1.0MATR-ND
Capacitor	1	C1	Kemet	C0805C104M5RACTU	399-1169-2-ND

Appendix C: Touchscreen Controller PC Test Application



Appendix D: Three-Points Calibration Procedure

The calculation of inverse matrix is very difficult for 8-bit cores. Equation 11 in a non-matrix form:

$$\alpha_x = \frac{(X_{D1} - X_{D3})(Y_{T2} - Y_{T3}) - (X_{D2} - X_{D3})(Y_{T1} - Y_{T3})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

$$\beta_x = \frac{(X_{T1} - X_{T3})(Y_{D2} - Y_{D3}) - (X_{D2} - X_{D3})(Y_{T1} - Y_{T3})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

$$dX = \frac{Y_{T1}(X_{T3}X_{D2} - X_{T2}X_{D3}) + Y_{T2}(X_{T1}X_{D3} - X_{T3}X_{D1}) + Y_{T3}(X_{T2}X_{D1} - X_{T1}X_{D2})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

$$\alpha_y = \frac{(Y_{D1} - Y_{D3})(Y_{T2} - Y_{T3}) - (Y_{D2} - Y_{D3})(Y_{T1} - Y_{T3})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

$$\beta_y = \frac{(X_{T1} - X_{T3})(Y_{D2} - Y_{D3}) - (Y_{D1} - Y_{D3})(X_{T2} - X_{T3})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

$$dY = \frac{Y_{T1}(X_{T3}Y_{D2} - X_{T2}Y_{D3}) + Y_{T2}(X_{T1}Y_{D3} - X_{T3}Y_{D1}) + Y_{T3}(X_{T2}Y_{D1} - X_{T1}Y_{D2})}{(X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})}$$

The right sides of the all expressions contain the same denominator. This can be very useful for integer arithmetic, isolating the denominator and marking it as a C vector:

$$C[0] = (X_{T1} - X_{T3})(Y_{T2} - Y_{T3}) - (X_{T2} - X_{T3})(Y_{T1} - Y_{T3})$$

$$C[1] = (X_{D1} - X_{D3})(Y_{T2} - Y_{T3}) - (X_{D2} - X_{D3})(Y_{T1} - Y_{T3})$$

$$C[2] = (X_{T1} - X_{T3})(Y_{D2} - Y_{D3}) - (X_{D2} - X_{D3})(Y_{T1} - Y_{T3})$$

$$C[3] = Y_{T1}(X_{T3}X_{D2} - X_{T2}X_{D3}) + Y_{T2}(X_{T1}X_{D3} - X_{T3}X_{D1}) + Y_{T3}(X_{T2}X_{D1} - X_{T1}X_{D2})$$

$$C[4] = (Y_{D1} - Y_{D3})(Y_{T2} - Y_{T3}) - (Y_{D2} - Y_{D3})(Y_{T1} - Y_{T3})$$

$$C[5] = (X_{T1} - X_{T3})(Y_{D2} - Y_{D3}) - (Y_{D1} - Y_{D3})(X_{T2} - X_{T3})$$

$$C[6] = Y_{T1}(X_{T3}Y_{D2} - X_{T2}Y_{D3}) + Y_{T2}(X_{T1}Y_{D3} - X_{T3}Y_{D1}) + Y_{T3}(X_{T2}Y_{D1} - X_{T1}Y_{D2})$$

Now Equations 8 and 9 look like:

$$X_D = \frac{C[1]X_T + C[2]Y_T + C[3]}{C[0]}$$

$$Y_D = \frac{C[4]X_T + C[5]Y_T + C[6]}{C[0]}$$

This method is optimal for 8-bit cores. The following code demonstrates it.

```
//  PSoC designer sample code for 3-points calibration procedure.
//  Define point type X,Y
typedef struct tPoint{
    signed int x;    //  signed value by calibration
    signed int y;    //  rotation or shift can return little neg value
}tPoint;

//  Calibration vector C, init it default values
static signed long CIndex[7];

//  Global points on touch and display
//  Tp – received point on touch
//  Dp – calculated point on display
static tPoint Dp, Tp;

//  Refresh the display point from the received touch point
void DisplayPointGet(void)
{
    Dp.x = (CIndex[1]*Tp.x + CIndex[2]*Tp.y + CIndex[3])/CIndex[0];
    Dp.y = (CIndex[4]*Tp.x + CIndex[5]*Tp.y + CIndex[6])/CIndex[0];
}

//  3-point calculate calibration indexes
//  Input:
//  Tp - pointer to array of 3 points on touchscreen
//  Dp - pointer to array of 3 points on display

void CalculateCalibrationConstants(tPoint *Dp, tPoint *Tp)
{
    CIndex[0] = (signed long)(Tp->x-(Tp+2)->x)*((Tp+1)->y-(Tp+2)->y)
              - (signed long)((Tp+1)->x-(Tp+2)->x)*(Tp->y-(Tp+2)->y);

    CIndex[1] = (signed long)(Dp->x-(Dp+2)->x)*((Tp+1)->y-(Tp+2)->y)
              - (signed long)((Dp+1)->x-(Dp+2)->x)*(Tp->y-(Tp+2)->y);

    CIndex[2] = (signed long)(Tp->x-(Tp+2)->x)*((Dp+1)->x-(Dp+2)->x)
              - (signed long)(Dp->x-(Dp+2)->x)*((Tp+1)->x-(Tp+2)->x);

    CIndex[3] = (signed long)Tp->y*((signed long)(Tp+2)->x*(Dp+1)->x
                                  -(signed long)(Tp+1)->x*(Dp+2)->x)
              + (signed long)(Tp+1)->y*((signed long)Tp->x*(Dp+2)->x
                                  -(signed long)(Tp+2)->x*Dp->x)
              + (signed long)(Tp+2)->y*((signed long)(Tp+1)->x*Dp->x
                                  -(signed long)Tp->x*(Dp+1)->x);

    CIndex[4] = (signed long)(Dp->y-(Dp+2)->y)*((Tp+1)->y-(Tp+2)->y)              - (signed
long)((Dp+1)->y-(Dp+2)->y)*(Tp->y-(Tp+2)->y);

    CIndex[5] = (signed long)(Tp->x-(Tp+2)->x)*((Dp+1)->y-(Dp+2)->y)
              - (signed long)(Dp->y-(Dp+2)->y)*((Tp+1)->x-(Tp+2)->x);

    CIndex[6] = (signed long)Tp->y*((signed long)(Tp+2)->x*(Dp+1)->y
                                  -(signed long)(Tp+1)->x*(Dp+2)->y)
              + (signed long)(Tp+1)->y*((signed long)Tp->x*(Dp+2)->y
                                     -(signed long)(Tp+2)->x*Dp->y)
              + (signed long)(Tp+2)->y*((signed long)(Tp+1)->x*Dp->y
                                     -(signed long)Tp->x*(Dp+1)->y);
}
```

Appendix E: Capacitive Touchscreens as Alternative to Resistive Touchscreens

A capacitive touchscreen panel consists of an insulator such as glass coated with a transparent conductor such as indium tin oxide (ITO). Because a human body is also an electrical conductor, touching the surface of the screen results in a distortion of the screen's electrostatic field measurable as a change in its capacitance. Different technologies can be used to determine the location of a touch. The location is then sent to the controller for processing. Unlike a resistive touchscreen, you cannot use a capacitive touchscreen through most types of electrically insulating materials, such as gloves. You need a special capacitive stylus, or a special-application glove with finger tips that generate static electricity. This disadvantage especially affects capacitive touchscreens' usability in consumer electronics, such as touch tablet PCs and capacitive smart phones in cold weather.

In surface capacitance basic technology, only one side of the insulator is coated with a conductive layer. A small voltage is applied to the layer, resulting in a uniform electrostatic field. When a conductor, such as a human finger, touches an uncoated surface, a capacitor is dynamically formed. The sensor's controller can determine the location of a touch indirectly, basing on the change in the capacitance as measured from the four corners of the panel. Because it has no moving parts, the controller is moderately durable but has a limited resolution. It is prone to false signals from parasitic capacitive couplings, and needs calibration during manufacturing. It is therefore most often used in simple applications such as industrial controls and kiosks.

Projected capacitive touch (PCT) technology is a capacitive technology that permits more accurate and flexible operation by etching the conductive layer. An X-Y grid is formed either by etching a single layer to form a grid pattern of electrodes, or by etching two separate, perpendicular layers of a conductive material with parallel lines or tracks to form a grid (comparable to the pixel grid found in many LCD displays).

A greater resolution of PCT allows operation without a direct contact. The conducting layers can be coated with further protective insulating layers, and operate even under screen protectors, or behind weather and vandal-proof glass. PCT is a more robust solution compared to the resistive touch technology because the PCT layers are made from glass. Depending on the implementation, an active or passive stylus can be used instead of or in addition to a finger. This is common with point-of-sale devices that require a signature capture. Gloved fingers may or may not be sensed, depending on the implementation and gain settings. Conductive smudges and similar interference on the panel surface can interfere with the performance. Such conductive smudges come mostly from sticky or sweaty fingertips, especially in high humidity environments. Collected dust, which adheres to the screen due to the moisture from fingertips, can also be a problem. There are two types of PCT: Self Capacitance and Mutual Capacitance.

A PCT screen consists of an insulator such as glass or foil, coated with a transparent conductor (Copper, ATO, Nanocarbon, or ITO). If a finger (which is also a conductor) touches the surface of the screen, the local electrostatic field distorts. This distortion can be measured to obtain the finger coordinates. Nowadays, mutual capacitive technology is more common than PCT technology.

In mutual capacitive sensors, there is a capacitor at every intersection of each row and each column. A 16-by-14 array, for example, has 224 independent capacitors. A voltage is applied to the rows or columns. Bringing a finger or conductive stylus close to the surface of the sensor changes the local electrostatic field, which reduces the mutual capacitance. The capacitance change at every individual point on the grid can be measured to accurately determine the touch location by measuring the voltage in the other axis. Mutual capacitance allows multi-touch operation where multiple fingers, palms, or styli can be accurately tracked at the same time.

Self-capacitance sensors can have the same X-Y grid as mutual capacitance sensors, but the columns and rows operate independently. With self-capacitance, the capacitive load of a finger is measured on each column or row electrode by a current meter. This method produces a stronger signal than the mutual capacitive method, but it is unable to detect accurately more than one finger, which results in "ghosting", or misplaced location sensing.

About the Author

Name:	Svyatoslav Paliy
Title:	Engineer
Background:	Svyatoslav Paliy graduated from Lviv Polytechnic National University (Ukraine) in 2000. His interests include various aspects of embedded systems design in addition to Windows and Linux programming.
Contact:	svyp@cypress.com

Name:	Andrij Bilynskyy
Title:	Application Engineer
Background:	Andrij Bilynskyy graduated from Ivan Franko National University of Lviv in 2003. His interests include SW/HW electronic design and Embedded OS.
Contact:	anbi_ukr@cypress.com

Document History

Document Title: PSoC® 1 – Interface to Four-wire Resistive Touchscreen - AN2376

Document Number: 001-15228

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1063380	XSG	05/11/2007	New application note.
*A	1551303	XSG	10/05/2007	Changed title
*B	3347438	ANBI_UKR	08/18/2011	Project support the latest version PSoC designer
*C	3479341	ANJR_UKR	01/12/2011	Title Updated Major content update.
*D	4528452	RJVB	10/08/2014	Updated Software Version as “PSoC® Designer™ 5.4 CP1” in page 1. Updated attached example project to PD5.4 CP1. Updated to new template. Completing Sunset Review.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
Optical Navigation Sensors	cypress.com/go/ons
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com