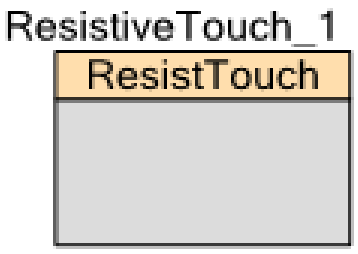
Resistive Touch (ResistiveTouch)

1.0

Features



- Supports 4-wire resistive touchscreen interface
- Supports the Delta Sigma Converter for both the PSoC 3 and PSoC 5 devices
- Supports the ADC Successive Approximation Register for PSoC 5 devices

General Description

This resistive touchscreen component is used to interface with a 4-wire resistive touch screen. The component provides a method to integrate and configure the resistive touch elements of a touchscreen with the emWin Graphics library. It integrates hardware-dependent functions that are called by the touchscreen driver supplied with emWin when polling the touch panel.

When to Use a ResistiveTouch

Use a ResistiveTouch component where low cost and simple interface electronics are required.

Input/Output Connections

This section describes the various input and output connections for the ResistiveTouch.

xm – Digital Input / Output

Signal x– from axis X of the resistive touch panel (active low).

xp – Analog / Digital Output

Signal x+ from axis X of the resistive touch panel (active high).

ym – Digital Input / Output

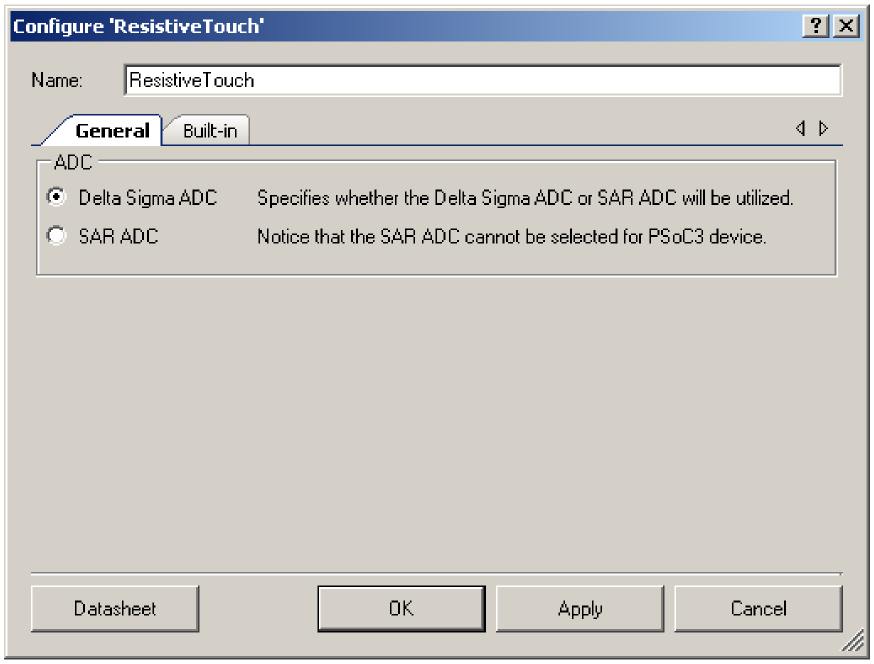
Signal y– from axis Y of the resistive touch panel (active low).

yp – Analog / Digital Output

Signal y+ from axis Y of the resistive touch panel (active high).

Component Parameters

Drag a ResistiveTouch component onto your design and double click it to open the **Configure** dialog.



Name: ResistiveTouch

ADC

[✓ Delta Sigma ADC] Specifies whether the Delta Sigma ADC or SAR ADC will be utilized.

SAR ADC Notice that the SAR ADC cannot be selected for PSoC3 device.

The ResistiveTouch component provides the following parameter.

ADC

The **ADC** parameter determines which ADC to use. For PSoC 3 devices, select the **Delta Sigma ADC** option.

Placement

All placement information for the ResistiveTouch is provided to the API through the *cyfitter.h* file.

Resources

The ResistiveTouch component is a dedicated piece of analog/digital hardware in the PSoC 3/PSoC 5 family that is used for connection to external resistive touch-panel devices.

Resources	Resource Type					API Memory (Bytes)		Pins (per External I/O)
	Decimator	SAR	DSM	PLDs	Interrupts	Flash	RAM	
ResistiveTouch_DelSig	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
ResistiveTouch_SAR	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name “ResistiveTouch_1” to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is “ResistiveTouch”.

Function	Description
ResistiveTouch_Start()	Calls the ResistiveTouch_Init() and ResistiveTouch_Enable() APIs.
ResistiveTouch_Stop()	Stops the DelSig ADC or SAR ADC and stops the AMux component.
ResistiveTouch_Init()	Calls the Init functions of the DelSig ADC or SAR ADC and AMux components
ResistiveTouch_Enable()	Enables the DelSig ADC or SAR ADC and enables the AMux component.
ResistiveTouch_ActivateY()	Configures the pins for measurement of Y-axis.
ResistiveTouch_ActivateX()	Configures the pins for measurement of X-axis.
ResistiveTouch_TouchDetect()	Detects a touch on the screen.
ResistiveTouch_Measure()	Returns the result of the A/D converter.
ResistiveTouch_SaveConfig()	Saves the configuration of the DelSig ADC or SAR ADC.
ResistiveTouch_Sleep()	Prepares the DelSig ADC or SAR ADC for low-power modes by calling SaveConfig and Stop functions.
ResistiveTouch_RestoreConfig()	Restores the configuration of the DelSig ADC or SAR ADC.
ResistiveTouch_Wakeup()	Restores the DelSig ADC or SAR ADC after waking up from a low-power mode.

Global Variables

Variable	Description
ResistiveTouch_initVar	Indicates whether the ResistiveTouch has been initialized. The variable is initialized to 0 and set to 1 the first time ResistiveTouch_Start() is called. This allows the component to restart without reinitialization after the first call to the ResistiveTouch_Start() routine. If reinitialization of the component is required, then the ResistiveTouch_Init() function can be called before the ResistiveTouch_Start() or ResistiveTouch_Enable() function.
ResistiveTouch_enableVar	This variable is used to indicate enable/disable component state.

void ResistiveTouch_Start(void)

Description:	Calls ResistiveTouch_Init() and ResistiveTouch_Enable() APIs.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_Init(void)

Description:	Calls the Init functions of the DelSig ADC or SAR ADC and AMux components.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_Enable(void)

Description:	Enables the DelSig ADC or SAR ADC and enables the AMux component.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_Stop(void)

Description:	Stops the DelSig ADC or SAR ADC and stops the AMux component.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_ActivateX(void)

Description:	Configures the pins for measurement of X-axis.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_ActivateY(void)

Description:	Configures the pins for measurement of Y-axis.
Parameters:	None
Return Value:	None
Side Effects:	None

int16 ResistiveTouch_Measure(void)

Description:	Returns the result of the A/D converter.
Parameters:	None
Return Value:	int16: The result of the ADC conversion
Side Effects:	None

uint8 ResistiveTouch_TouchDetect(void)

Description:	Detects a touch on the screen.
Parameters:	None
Return Value:	uint8: The touch state 0 – untouched 1 – touched
Side Effects:	None

void ResistiveTouch_SaveConfig(void)

Description:	Saves the configuration of the DelSig ADC or SAR ADC.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_RestoreConfig(void)

Description:	Restores the configuration of the DelSig ADC or SAR ADC.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch_Sleep(void)

Description:	Prepares the DelSig ADC or SAR ADC for low-power modes by calling SaveConfig and Stop functions.
Parameters:	None
Return Value:	None
Side Effects:	None

void ResistiveTouch Wakeup(void)

Description:	Restores the DelSig ADC or SAR ADC after waking up from a low-power mode.
Parameters:	None
Return Value:	None
Side Effects:	None

Sample Firmware Source Code

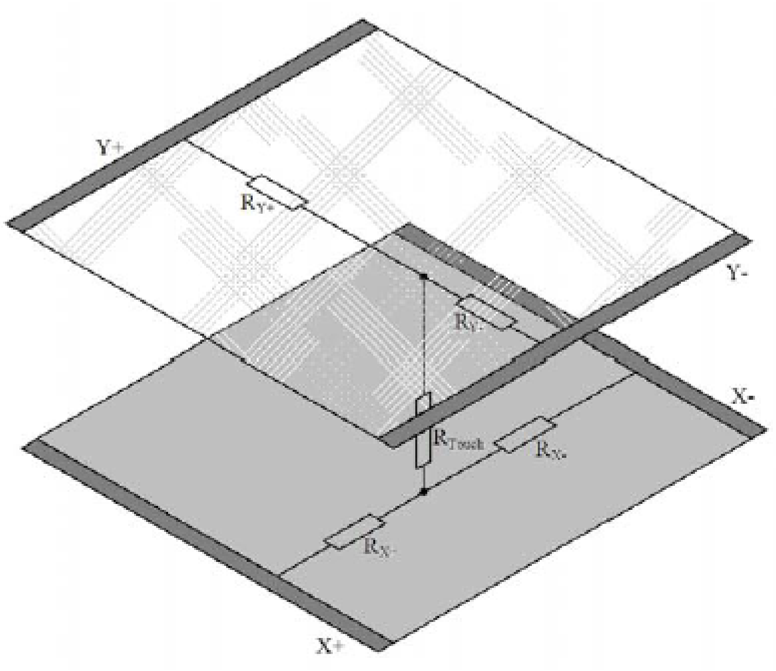
PSoC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Functional Description

This component provides a 4-wire resistive touch screen interface to read touchscreen coordinates and measure screen resistance. This component provides access to the touchscreen functionality of the SEGGER emWin graphics library for translation of resistance to screen coordinates.

The diagram that follows shows the schematic of a 4-wire touchscreen when pressure is applied.



The point of contact “divides” each layer in a series resistor network with two resistors (see the diagram), and a connecting resistor between the two layers. By measuring the voltage at this point, you get information about the position of the contact point orthogonal to the voltage gradient. To get a complete set of coordinates, you must apply the voltage gradient once in the vertical and then in the horizontal direction. First, you must apply a supply voltage applied to one layer and perform a measurement of the voltage across the other layer; next connect the supply to the other layer and measure the opposite layer voltage. When in touch mode, one of the lines is connected to detect touch activity. The following table defines the configuration of the pins while measuring the coordinates or touch.

	XP	XM	YP	YM
touch	Res Pullup	Digital Hi-Z	Analog Hi-Z	Strong Drive
X-Coordinate	Strong Drive	Strong Drive	Analog Hi-Z	Analog Hi-Z
Y-Coordinate	Analog Hi-Z	Analog Hi-Z	Strong Drive	Strong Drive

When a ResistiveTouch component is placed onto the project schematic, it is important that I/O port designations be set for xm, xp, ym, yp pins. Designation of these pins is not done on the symbol or schematic; instead, it is done in the Pins tab of the Design-Wide Resources window.

DC and AC Electrical Characteristics

TBD

Component Changes

Version 1.0 is the first release of the ResistiveTouch component.