

## Objective

This example demonstrates how to set up PSoC 4 GPIO interrupts. It also shows how to create your own interrupt handler function (ISR) instead of using the default ISR generated by PSoC Creator.

## Overview

This example demonstrates PSoC 4 GPIO interrupts. It uses two GPIO pins to generate interrupts – one on the rising edge and one on the falling edge. The interrupt handler function (ISR) turns a GPIO pin on or off, depending on the input edge. This causes an LED connected to the pin to show which edge was received.

## Requirements

**Tool:** PSoC Creator™ 3.3 SP1 or higher.

**Programming Language:** C (GCC 4.9-2015-q1-update).

**Associated Parts:** All PSoC 4 parts

**Related Hardware:** All PSoC 4 kits; see [Table 1](#) and [Table 2](#) for pin assignments.

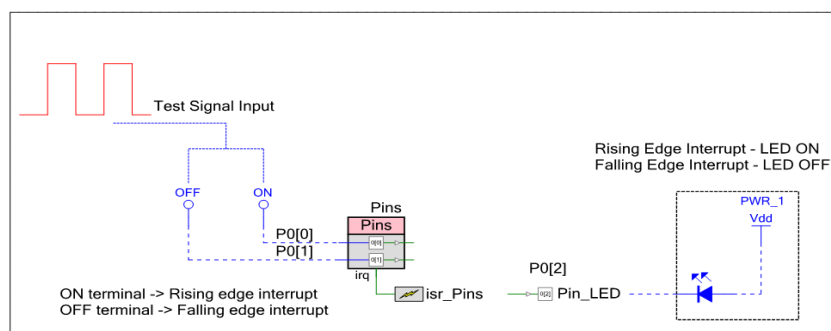
## Design

The hardware part of the design features two GPIO pins configured as inputs as [Figure 1](#) shows. They are also configured to generate interrupts – one on a rising edge and the other on a falling edge. The two pins selected are from the same port. So, there is only one interrupt request line (IRQ) signal in the schematic for that physical port. Port 0 is selected for this purpose in this code example.

In the firmware design, the ISR code distinguishes between these two interrupts and controls the state of an output pin depending on the interrupt source. The output pin drives an LED; therefore the LED shows which edge was received.

Two pins are configured to generate the interrupt. The “Pins” Component is configured such that ON terminal generates an interrupt on the rising-edge signal and the OFF terminal generate an interrupt on the falling-edge signal as [Figure 1](#) shows. The interrupt service routine has a code to distinguish between these two interrupts. On the rising-edge interrupt, the LED is turned ON; on the falling-edge interrupt, the LED is turned OFF.

Figure 1. GPIO Interrupt Schematic



## Components

Table 1 lists the PSoC Creator Components used in this example as well as the placement used by each.

Table 1. List of PSoC Creator Components

Component	Placement
Pins	2 pins
isr_Pins	One entry in interrupt vector memory
Pin_LED	1 pin

## Parameter Settings

Figure 2, Figure 3, and Figure 4 show the changed settings for the Pins Component.

Figure 2. Pins Configuration

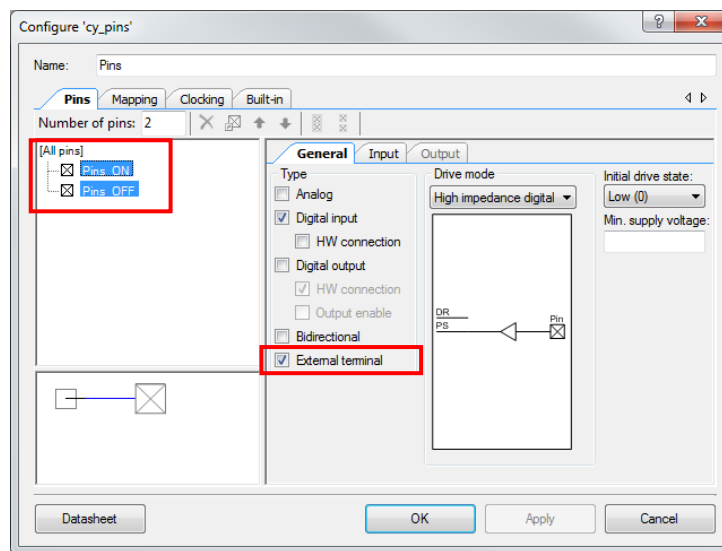


Figure 3. Pins Component Configuration Tool – Input Tab for Pins\_ON

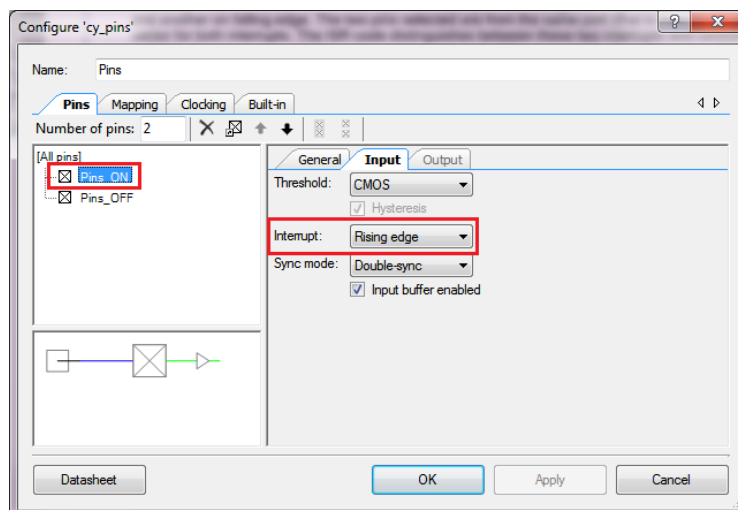


Figure 4. Pins Component Configuration Tool – Input Tab for Pins\_OFF

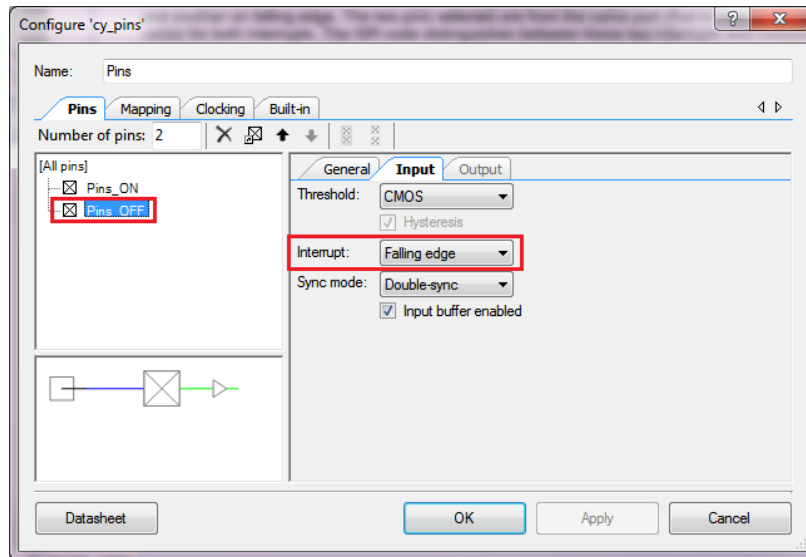


Figure 5 shows the changed settings for the isr\_Pins Component.

Figure 5. isr\_Pins Configuration

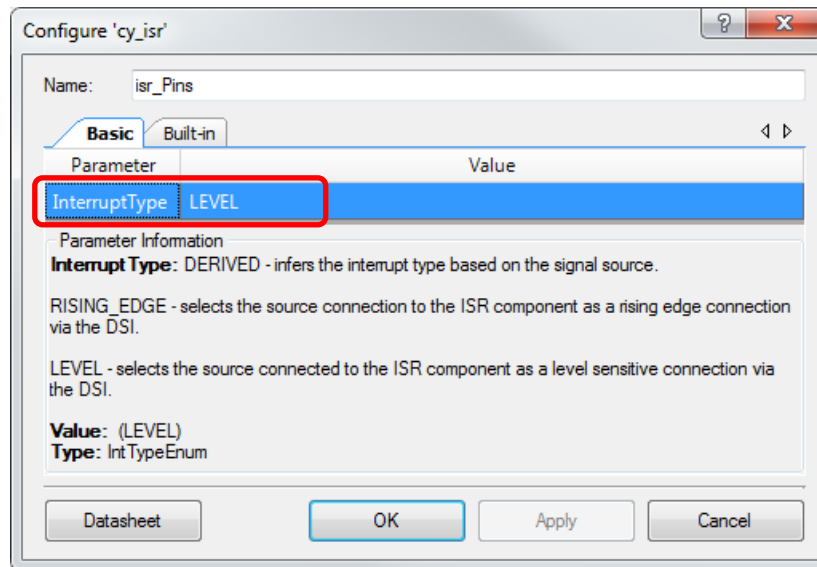
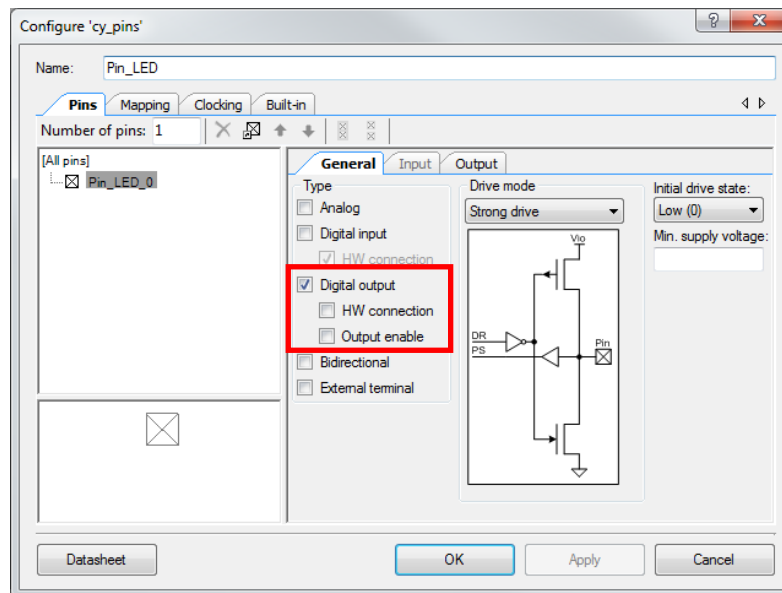


Figure 6 shows the changed settings for the Pin\_LED Component.

Figure 6. Pin\_LED Configuration



## Design-Wide Resources

By default, the code example is for development kit [CY8CKIT-042](#). If you are developing the project on other development kit, change the pin assignment for Pin\_LED per [Table 2](#).

Table 2. Pin assignment for Pin\_LED

Name of the Component	Port No.	Development Board
Pin_LED	P0[2]	<a href="#">CY8CKIT-042</a>
Pin_LED	P0[2]	<a href="#">CY8CKIT-040</a>
Pin_LED	P2[6]	<a href="#">CY8CKIT-042-BLE</a>
Pin_LED	P2[6]	<a href="#">CY8CKIT-044</a>
Pin_LED	P5[2]	<a href="#">CY8CKIT-046</a>

[Table 3](#) shows the pin assignment for component Pins.

Table 3. Pin assignment for Pins

Name of the component	Port No.	Development Board
Pins	P0[1:0]	All

## Operation

After the completing the schematic, follow the step given below to test the project:

1. Build the project to generate the hex file.
2. Connect the PC to the development board with the USB cable.
3. Download the hex file to the PSoC 4 chip.
4. Short P0[0] with P0[1].
5. Connect a signal generator output to P0[0] and turn ON the signal generator.
6. Notice that the LED on the development board toggles on every edge of the square-wave signal.

## Related Documents

Table 4 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 4. Related Documents

Document	Title	Comment
<a href="#">AN79953</a>	Getting Started with PSoC® 4	Introduces you to PSoC 4, an ARM® Cortex®-M0 MCU based programmable system-on-chip. It helps you explore the architecture and Creator development tools
<a href="#">AN90799</a>	PSoC® 4 Interrupts	Provides interrupts architecture and its configuration
<a href="#">AN86439</a>	PSoC® 4 - Using GPIO Pins	How to use PSoC 4 GPIO pins effectively and take full advantage of their features. Major topics include GPIO basics, configuration options, mixed-signal use, registers, interrupts, and low-power behavior.
<a href="#">AN89610</a>	PSoC® 4 and PSoC 5LP ARM Cortex Code Optimization	This application note shows how to optimize C and assembler code for the ARM Cortex CPUs in PSoC 4 and PSoC 5LP. GCC and Keil Microcontroller Development Kit (MDK) C compilers are supported.
<a href="#">001-94480</a>	System Reference Guide	This System Reference Guide describes functions supplied by the PSoC Creator cy_boot Component. The cy_boot Component provides the system functionality for a project to give better access to chip resources. The functions are not part of the Component libraries but may be used by them. You can use the function calls to reliably perform needed chip functions.
<a href="#">CE210557</a>	PSoC® 4 GPIO Interrupt	This example demonstrates how to generate a periodic interrupt using a fixed-function timer/counter/PWM (TCPWM).
<a href="#">CE95400</a>	Watchdog Timer Reset and Interrupt for PSoC® 4 Devices	This example demonstrates how to use the watchdog in PSoC 4 devices to both reset the system and wake up from the Deep Sleep low power mode.
PSoC Creator Component Datasheets		
<a href="#">Pins</a>	Controls interface with physical I/O port pins	
<a href="#">Interrupt</a>	The Interrupt component defines hardware triggered interrupts. It also provides a software method to genrate interrupt.	
Device Documentation		
<a href="#">PSoC 4 Datasheets</a>		<a href="#">PSoC 4 Technical Reference Manuals</a>
Development Kit (DVK) Documentation		
<a href="#">PSoC 4 Kits</a>		

## PSoC Resources

Cypress provides a wealth of data at [www.cypress.com](http://www.cypress.com) to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see [KBA86521, How to Design with PSoC 3, PSoC 4, and PSoC 5LP](#). The following is an abbreviated list for PSoC 4:

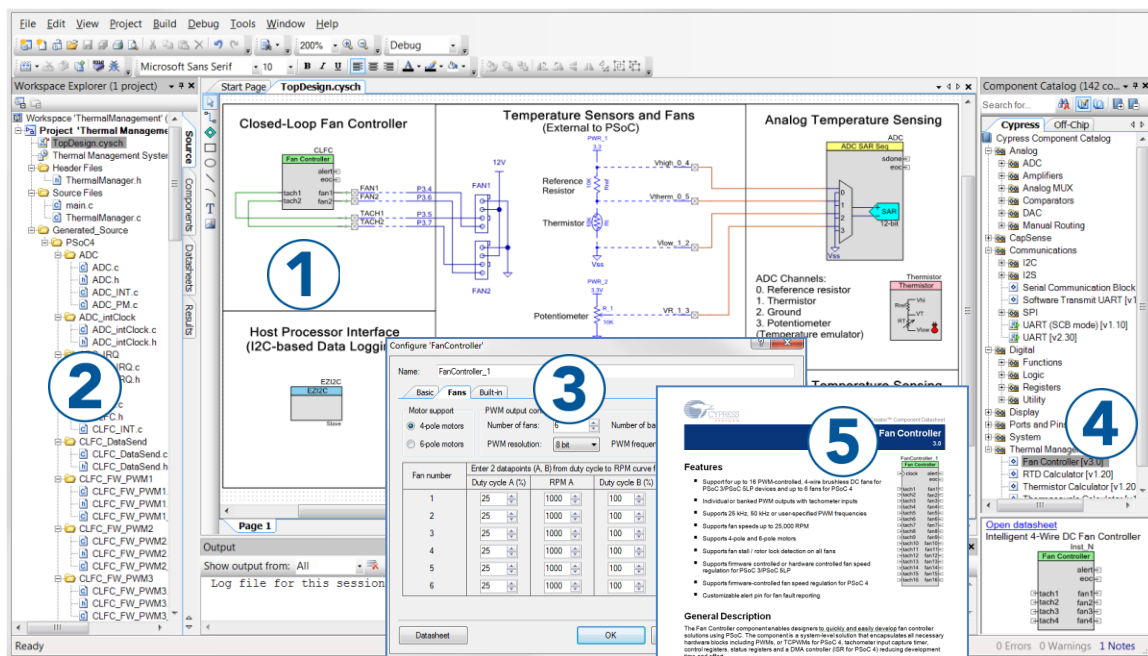
- **Overview:** [PSoC Portfolio](#), [PSoC Roadmap](#)
- **Product Selectors:** [PSoC 1](#), [PSoC 3](#), [PSoC 4](#), or [PSoC 5LP](#). In addition, [PSoC Creator](#) includes a device selection tool.
- **Datasheets** describe and provide electrical specifications for the [PSoC 4](#) device families
- **CapSense Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 4 family of devices.
- **Application Notes and Code Examples** cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM)** provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- **Development Kits:**
  - [CY8CKIT-042](#) PSoC 4 Kit, is an easy-to-use and inexpensive development platform. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
  - [CY8CKIT-044](#) PSoC 4 Kit, is platform enabling design and debug of PSoC 4 M-Series devices.
  - [CY8CKIT-046](#) is a low-cost hardware and software platform to enable design and debug of the PSoC 4200L device.
  - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
  - The [MiniProg3](#) device provides an interface for flash programming and debug.

## PSoC Creator

[PSoC Creator](#) is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, and PSoC 5LP. See [Figure 7](#) – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware
3. Configure Components using configuration tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 7. PSoC Creator Features



## Document History

Document Title: CE210558 - PSoC® 4 GPIO Interrupt

Document Number: 002-10558

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5096932	ASRI	01/21/2016	New code example
*A	5741063	AESATP12	05/26/2017	Updated logo and copyright.

## 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

ARM® Cortex® Microcontrollers	<a href="#">cypress.com/arm</a>
Automotive	<a href="#">cypress.com/automotive</a>
Clocks & Buffers	<a href="#">cypress.com/clocks</a>
Interface	<a href="#">cypress.com/interface</a>
Internet of Things	<a href="#">cypress.com/iot</a>
Memory	<a href="#">cypress.com/memory</a>
Microcontrollers	<a href="#">cypress.com/mcu</a>
PSoC	<a href="#">cypress.com/psoc</a>
Power Management ICs	<a href="#">cypress.com/pmic</a>
Touch Sensing	<a href="#">cypress.com/touch</a>
USB Controllers	<a href="#">cypress.com/usb</a>
Wireless Connectivity	<a href="#">cypress.com/wireless</a>

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

## PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

## Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

## Technical Support

[cypress.com/support](#)



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

© Cypress Semiconductor Corporation, 2016-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](#). Other names and brands may be claimed as property of their respective owners.