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.

CE210558 - PSoC® 4 GPIO Interrupt

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.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C

Associated Parts: All PSoC 4 parts

Related Hardware: CY8CKIT-042, CY8CKIT-042-BLE, CY8CKIT-040, CY8CKIT-044, CY8CKIT-046

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.

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.

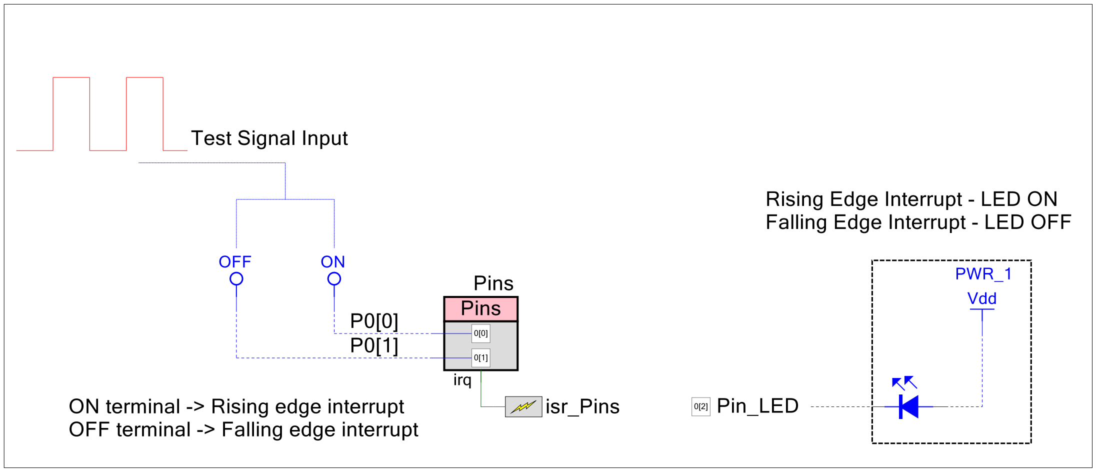
Design

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


Test Signal Input
OFF
ON
P0[0]
P0[1]
Pins
Pins
0[0]
0[1]
irq
isr_Pins
ON terminal -> Rising edge interrupt
OFF terminal -> Falling edge interrupt
Rising Edge Interrupt - LED ON
Falling Edge Interrupt - LED OFF
PWR_1
Vdd
0[2]
Pin_LED

Components

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

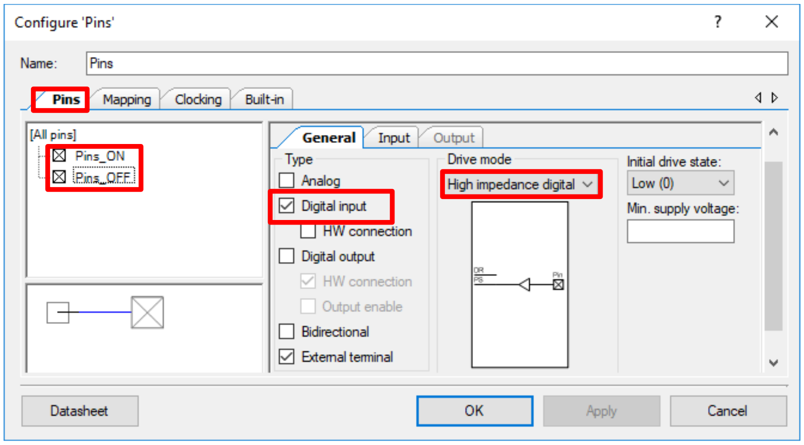
Table 1. List of PSoC Creator Components

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



Configure 'Pins'

Name: Pins

Tabs: Pins, Mapping, Clocking, Built-in

[All pins]
- ☒ Pins_ON
- ☒ Pins_OFF

General | Input | Output

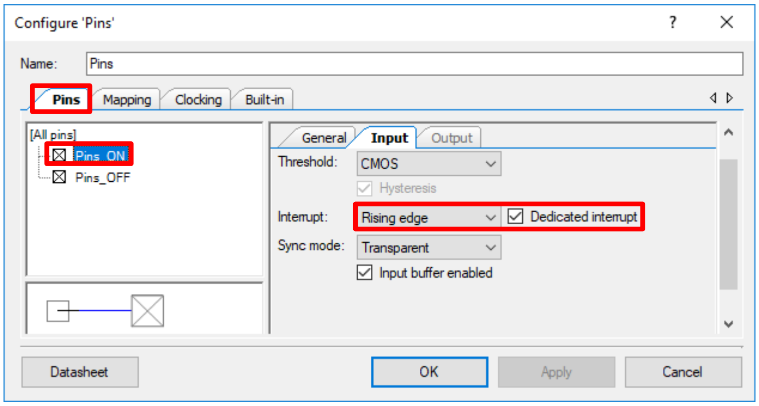
Type
- ☐ Analog
- ☑ Digital input
  - ☐ HW connection
- ☐ Digital output
  - ☑ HW connection
  - ☐ Output enable
- ☐ Bidirectional
- ☑ External terminal

Drive mode: High impedance digital

Initial drive state: Low (0)

Min. supply voltage:

Figure 3. Pins Component Configuration Tool – Input Tab for Pins_ON

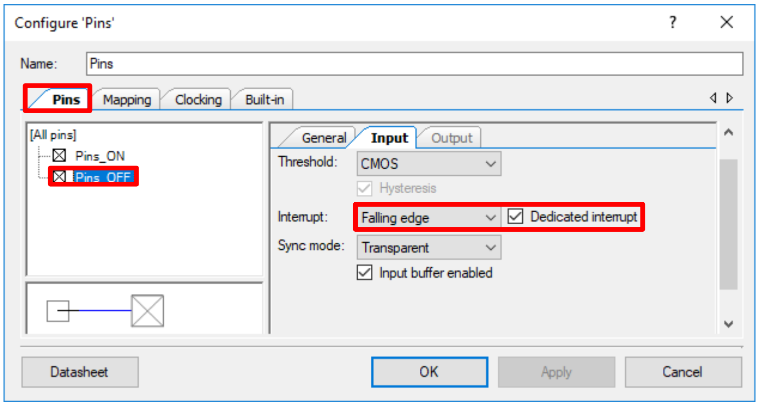

Configure 'Pins'

Name: Pins

[All pins]
☒ Pins_ON
☒ Pins_OFF

Threshold: CMOS
☑ Hysteresis
Interrupt: Rising edge ☑ Dedicated interrupt
Sync mode: Transparent
☑ Input buffer enabled

Figure 4. Pins Component Configuration Tool – Input Tab for Pins_OFF


Configure 'Pins'

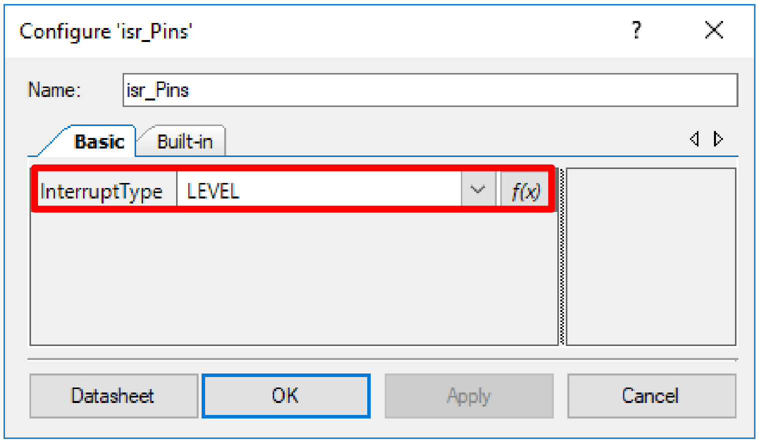
Name: Pins

[All pins]
☒ Pins_ON
☒ Pins_OFF

Threshold: CMOS
☑ Hysteresis
Interrupt: Falling edge ☑ Dedicated interrupt
Sync mode: Transparent
☑ Input buffer enabled

Figure 5 shows the changed settings for the isr_Pins Component.

Figure 5. isr_Pins Configuration

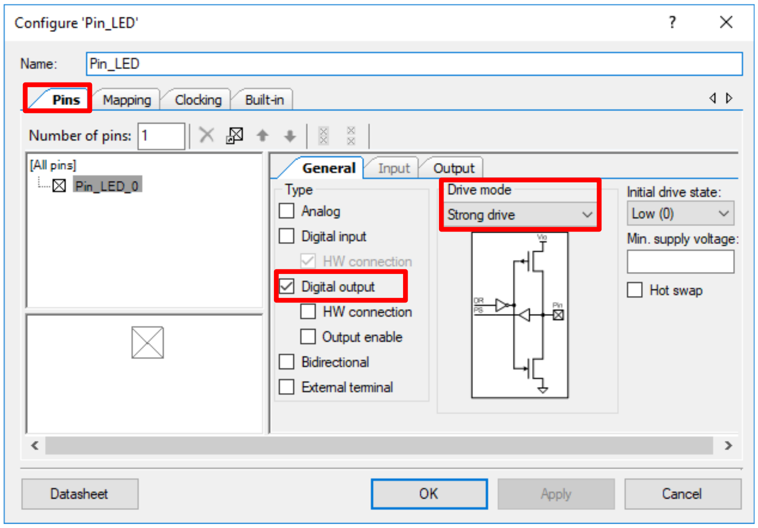

Configure 'isr_Pins'

Name: isr_Pins

InterruptType LEVEL

Figure 6 shows the changed settings for the Pin_LED Component.

Figure 6. Pin_LED Configuration



Name: Pin_LED

Number of pins: 1

[All pins]
☒ Pin_LED_0

Type
- ☐ Analog
- ☐ Digital input
  - ☑ HW connection
- ☑ Digital output
  - ☐ HW connection
  - ☐ Output enable
- ☐ Bidirectional
- ☐ External terminal

Drive mode: Strong drive

Initial drive state: Low (0)

Min. supply voltage:

☐ Hot swap

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]	CY8CKIT-042
Pin_LED	P0[2]	CY8CKIT-040
Pin_LED	P2[6]	CY8CKIT-042-BLE
Pin_LED	P2[6]	CY8CKIT-044
Pin_LED	P5[2]	CY8CKIT-046

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

Related Documents

Document	Comment
AN79953 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
AN90799 - PSoC® 4 Interrupts	Provides interrupts architecture and its configuration
AN86439 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.
PSoC Creator Component Datasheets	
Pins	Controls interface with physical I/O port pins
Interrupt	The Interrupt component defines hardware triggered interrupts. It also provides a software method to genrate interrupt.
Device Documentation	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals
Development Kit (DVK) Documentation	
PSoC 4 Kits	

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.
*B	6493390	AJYA	02/26/2019	Updated project to PSoC Creator 4.2

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	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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

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

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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. 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. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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.