

Interrupts

1.0

Features

- Enables GPIO interrupts
- Clears pending interrupts
- Explains use of interrupt priority and nested interrupts
- Sets up custom ISR

General Description

This example project demonstrates the basic operation of interrupts: setup and enable interrupts, generate a GPIO interrupt, set and change the interrupt priority. The project shows users how they can set up a custom interrupt and configure it as a nested interrupt, change the interrupts priority, clear pending interrupts, and generate a GPIO interrupt.

Development kit configuration

This example project is designed to run on the CY8CKIT-042 kit from Cypress Semiconductor. A description of the kit, along with more example programs and ordering information, can be found at <http://www.cypress.com/go/cy8ckit-042>.

The project requires configuration settings changes to run on other kits from Cypress Semiconductor. Table 1 is the list of the supported kits. To switch from CY8CKIT-042 to any other kit, change the project's device with the help of Device Selector called from the project's context menu.

Table 1. Development Kits vs Parts

Development Kit	Device
CY8CKIT-042	CY8C4245AXI-483
CY8CKIT-040	CY8C4014LQI-422
CY8CKIT-042-BLE	CY8C4247LQI-BL483
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L485
CY8CKIT-030	CY8C3866AXI-040
CY8CKIT-050	CY8C5868AXI-LP035
CY8CKIT-001	CY8C3866AXI-040 / CY8C5868AXI-LP035

The pin assignments for the supported kits are in Table 2.

Table 2. Pin Assignment

Pin Name	Development Kit							
	CY8CKIT-042	CY8CKIT-040	CY8CKIT-042 BLE	CY8CKIT-044	CY8CKIT-046	CY8CKIT-030	CY8CKIT-050	CY8CKIT-001**
LED_Isr	P0[2]	P1[1]	P3[6]	P2[6]	P5[3]	P6[2]	P6[2]	P1[7]
LED_Nested	P1[6]	P3[2]	P2[6]	P0[6]	P5[2]	P6[3]	P6[3]	P2[7]
Pin_Sw	P0[7]	P0[0]*	P2[7]	P0[7]	P0[7]	P6[1]	P6[1]	P0[0]

* CY8CKIT-040 kit does not have an integrated onboard switch, use external.

** For the CY8CKIT-001 kit: connect P1[7] (LED_Isr) to P14 (LED1); connect P2[7] (LED_Nested) to P14 (LED4); connect P0[0] (Pin_Sw) to P14 (SW1). In the Workspace Explorer window, double-click the project's design-wide resource file and assign the pins according to Table 2.

The following steps should be performed to observe the project operation:

1. Build the project and program the hex file into the target device.
2. Power cycle the device and observe the results on the LEDs.

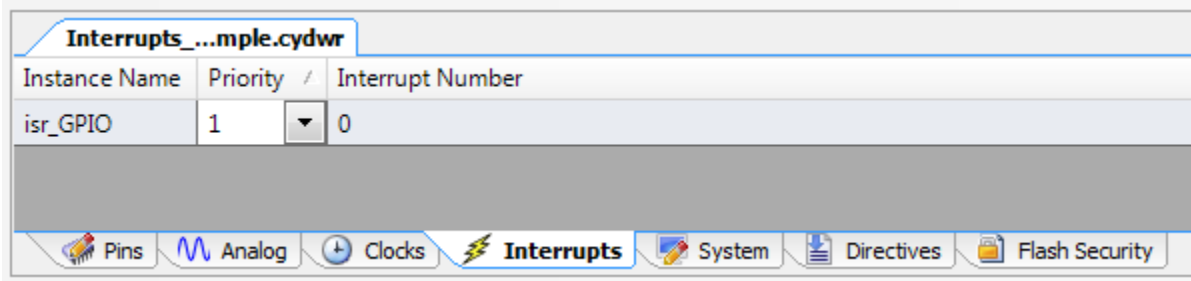
Project Configuration

The example project consists of the Interrupt and pin components. The custom interrupt is a part of the cy_boot component.

Pin_Sw is configured to generate an interrupt every time when the SW button is unpressed.

Isr_GPIO has changed the priority from the default value (3) to 1(Figure 1).

Figure 1. Priority of isr_GPIO Interrupt in Design-Wide Resource (Interrupts Tab)



Project Description

At the beginning of the main function, a GPIO interrupt is set up and enabled. After that, the initial priority for the GPIO interrupt is changed from 1 to 3 (DEFAULT_PRIORITY). The custom

interrupt (nested) is configured (sets up, sets priority) and enabled. In GPIOIsrHandler, the isr_GPIO pending interrupt clears, the pin interrupt clears, the green LED turns on, the nested software interrupt occurs after approximately a 1 second delay, the green LED turns off. In NestedIsrHandler, the software interrupt clears, the red LED turns on for approximately 1 second, the red LED turns off, a delay of approximately 1 second occurs.

Expected Results

Program the device with the project. Press SW, the green LED turns on. After 1 second delay, the red LED blinks for 1 second (the yellow color lights when green and red mix), the green LED turns off. After SW is pressed, the described above sequence repeats.



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

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

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges. PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

