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.

CE219521 – PSoC 6 MCU - GPIO Interrupt

Objective

This example demonstrates how to configure a GPIO to generate an interrupt in PSoC® 6 MCU using PSoC Creator™.

Requirements

Tool: PSoC Creator™ 4.2; Peripheral Driver Library (PDL) 3.0.4

Programming Language: C (Arm® GCC 5.4.1 and Arm MDK 5.22)

Associated Parts: All PSoC 6 MCU parts

Related Hardware: PSoC 6 BLE Pioneer Kit, PSoC 6 WiFi-BT Pioneer Kit

Overview

This code example demonstrates the use of GPIO configured as an input pin to generate interrupts on the CM4 CPU in PSoC 6 MCU. The GPIO signal interrupts the CPU and executes a user-defined Interrupt Service Routine (ISR). The GPIO interrupt acts as a wakeup source to wake the CPU from Deep Sleep.

Hardware Setup

This example uses the kit's default configuration. Refer to the kit guide to ensure the kit is configured correctly.

Software Setup

None.

Operation

1. Plug the kit into your computer's USB port.
2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help. Flash for both CPUs is programmed in a single program operation.
3. Confirm that the LED blinks four times and then turns OFF, indicating that the CPU has entered Deep Sleep.
4. Press the user switch connected to P0[4] to trigger an interrupt. This should cause the device to wake up, causing the LED to resume blinking at a new interval (faster blink rate to indicate that the ISR has executed). The LED blinks for four times and the device enters Deep Sleep again.
5. Pressing the switch again repeats the wakeup cycle and the LED resumes blinking with the original interval of 1 second. With every interrupt and execution of ISR, the interval of blinking is alternated between 1 second and 500 milliseconds.

Design and Implementation

PSoC 6 MCU is a dual-CPU architecture MCU with Arm® Cortex® M0+ (CM0+) and Arm Cortex M4 (CM4) CPUs. The CM0+ CPU enables the CM4 CPU on device reset. This code example uses a GPIO interrupt to wake the CM4 CPU from Deep Sleep. An LED is connected to an output pin and it is used for indicating the current state of the CPU. A blinking LED indicates that the CPU is active. After four successive blinks, the CPU is instructed to enter Deep Sleep. Because the GPIO state is retained during Deep Sleep, the LED stops blinking and stays OFF to indicate that the CPU is in Deep Sleep.

An input pin, externally connected to a switch, is configured to generate an interrupt when the switch is pressed. The interrupt triggers following two actions:

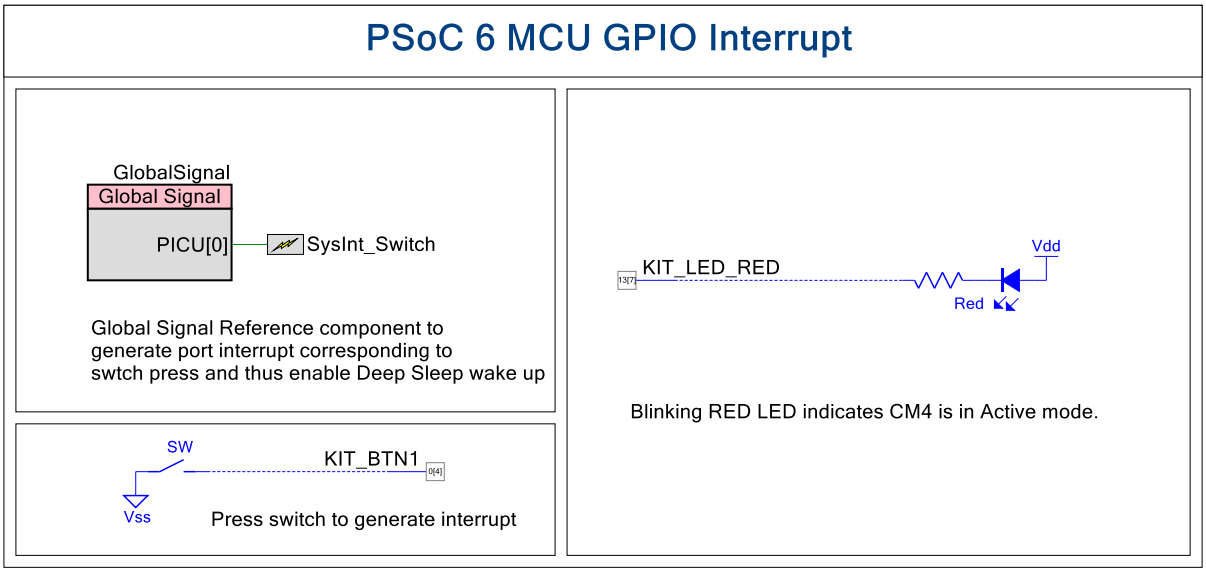
1. Generates a signal that wakes the CPU from Deep Sleep.

2. Executes an ISR.

When the ISR is executed, a flag is updated, which is used to change the rate of blinking the LED. With every press of the switch, the LED alternates blinking in intervals of 500 milliseconds and 1 second.

Figure 1 shows the PSoC Creator schematic of this code example. The project uses GPIO, Global Signal Reference, and SysInt Components.

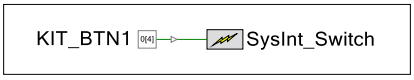
Figure 1. PSoC Creator Project Schematic for GPIO Interrupt



Design Considerations

The Global Signal Reference Component is used to route a Port Interrupt from the input pin. Port Interrupts are available in the Deep Sleep power mode and can thus wake up a CPU from Deep Sleep. The SysInt Component can also be connected to the pin directly as shown in Figure 2. This results in pin being used as a UDB Interrupt source. However, this interrupt signal is routed through Digital System Interconnect (DSI) and is not available during Deep Sleep and cannot wake up the CPU. For a list of Deep Sleep-compatible interrupt sources, see the "Interrupts" chapter in PSoC 6 MCU Architecture TRM.

Figure 2. Direct Connection of Input Pin to SysInt Component



Components and Settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
Digital Output Pin	KIT_LED_RED	Provides visual feedback.	See Figure 3
Digital Input Pin	KIT_BTN1	Provides a user interface.	See Figure 4
Global Signal Reference	GlobalSignal	Provides connections to device-level global signals and shared resource Interrupts.	See Figure 5
SysInt	SysInt_Switch	Provides an interface to connect hardware signals to a CPU interrupt request line.	Default

For information on the hardware resources used by a Component, see the Component datasheet.

Figure 3 to Figure 5 highlight the non-default settings for each Component in this example.

Figure 3. GPIO Pin Configuration for LED

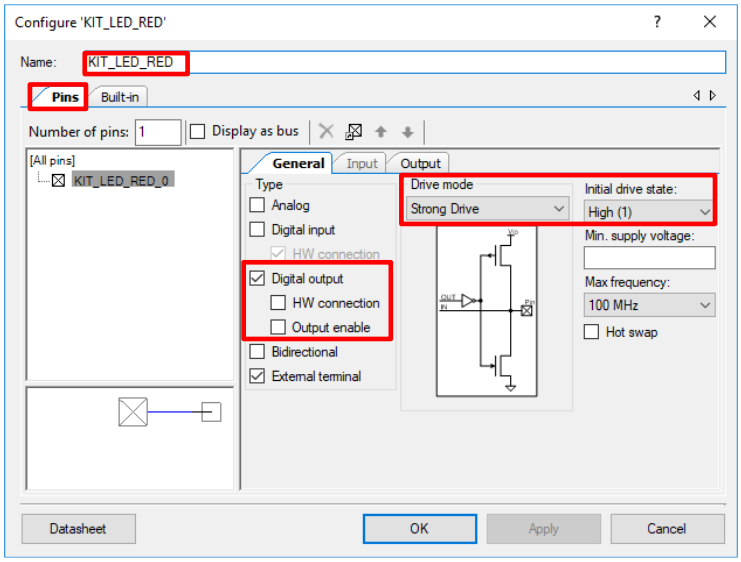


Figure 4. GPIO Pin configuration for Switch Input

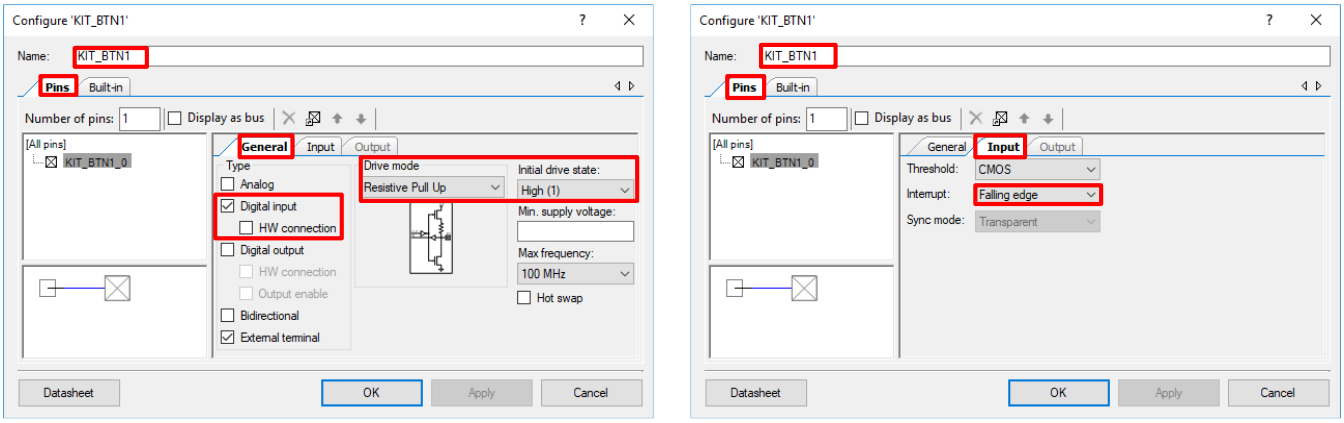
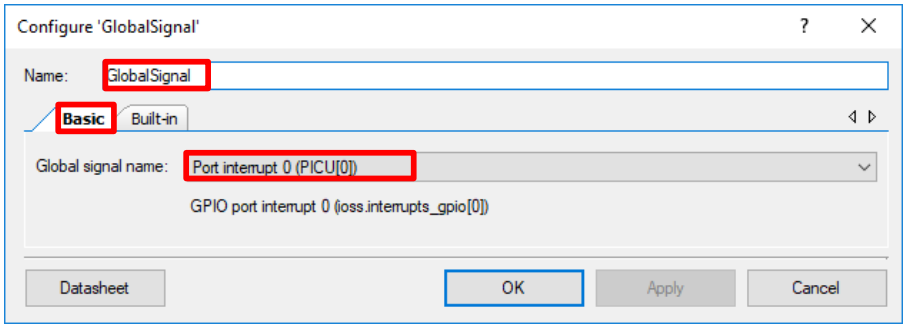


Figure 5. Global Signal Reference Configuration



Design-Wide / Global Resources

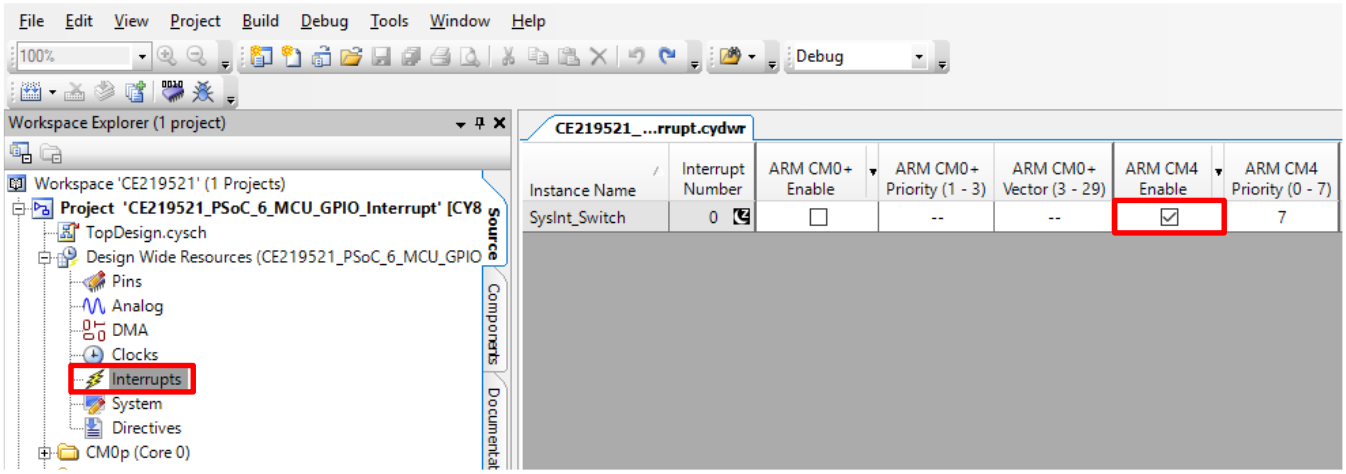
Table 2 shows the pin assignments for the switch and LEDs of CY8CKIT-062 BLE.

Table 2. DWR Pin Assignment Table

Component	Instance Name	Pin
Digital Input Pin	KIT_BTN1	P0[4]
Digital Output Pin	KIT_LED_RED	P13[7]

Figure 6 shows the interrupt configuration for the project. Note that the SysInt Component is assigned to CM4 through the checkbox. Interrupt priority is left to its default value because there are no other interrupts. For more information on configuring interrupts using the DWR, see AN217666 – PSoC 6 MCU Interrupts.

Figure 6. System Interrupt Configuration



Related Documents

For a comprehensive list of PSoC 6 MCU resources, see KBA223067 in the Cypress community.

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-CPU System Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
SysInt	Supports generating interrupts from hardware signals
Global Signal Reference	Provides connections to device global signal and shared resource Interrupts.
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
Tool Documentation	
PSoC Creator	Look in the downloads tab for Quick Start and User Guides
Peripheral Driver Library (PDL)	Installed by PSoC Creator 4.2. Look in the *<PDL install folder>/doc* folder for the User Guide and the API Reference.

Document History

Document Title: CE219521 - PSoC 6 MCU - GPIO Interrupt

Document Number: 002-19521

Revision	ECN	Orig. of Change	Submission Date	Description of Change
*A	5856495	JSLN	08/17/2017	Initial public release
*B	6405636	AJYA	12/10/2018	Moved project from CM0+ to CM4

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 Forums | 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

© Cypress Semiconductor Corporation, 2017-2018. 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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. 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. 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.