CE219521 – PSoC 6 MCU - GPIO Interrupt

Objective

This example demonstrates how to configure a GPIO to generate an interrupt using PSoC 6 MCU. The example also shows how the GPIO interrupt can be used to wake the CPU from Deep Sleep low-power mode.

Overview

This code example demonstrates the use of GPIO configured as an input pin to generate interrupts 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.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (ARM® GCC 5.4-2016-q2-update), ARM MDK Generic

Associated Parts: PSoC 6 MCU

Related Hardware: CY8CKIT-062 BLE Pioneer Kit

Design

This code example uses GPIO interrupt to wake the ARM Cortex®-M0+ CPU from Deep Sleep.

A green LED is connected to an output pin and it blinks at an interval of 1 second. A blinking green LED indicates that the CPU is active. After four successive blinks, the CPU is instructed to enter Deep Sleep. Just before entering Deep Sleep, another output pin is used to turn a red LED ON. Because the GPIO state is retained during Deep Sleep, the red LED stays on 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

Upon wakeup, the CPU turns the red LED OFF. When the ISR is executed, a flag is updated, which is used to change the rate of blinking the green LED. With every press of the switch, the green LED alternates blinking in the intervals of 500 ms and 1 s.

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 on CM0+.

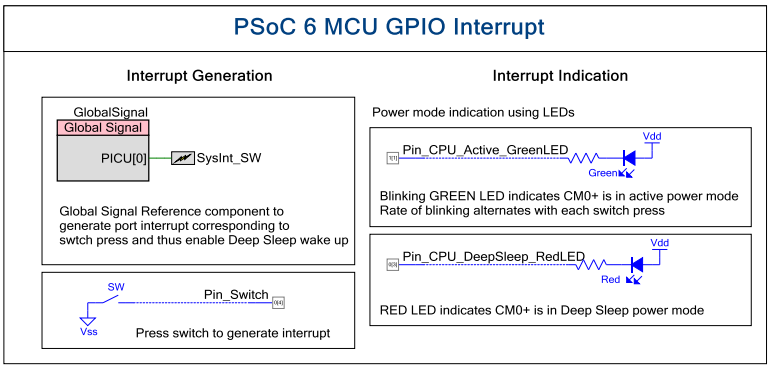
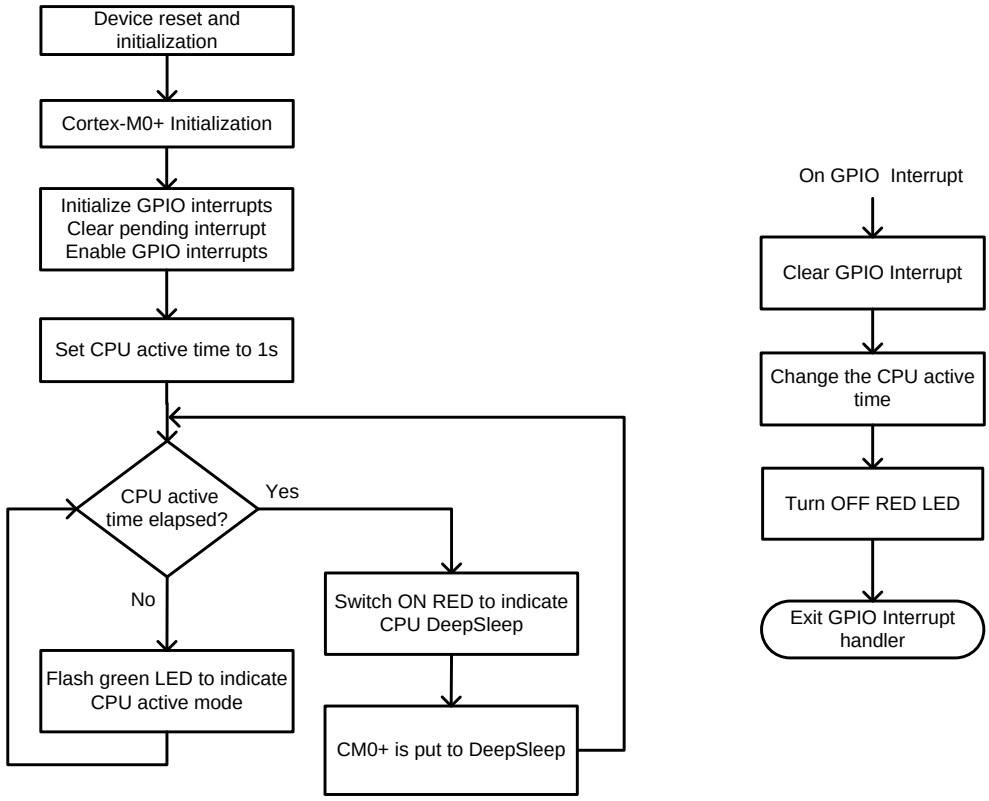


Figure 2 shows the firmware flowchart.

Note that after a PSoC 6 MCU device reset, CM0+ always executes first while CM4 is held in a reset state. In this example, only CM0+ is used to execute the code.

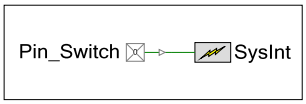
Figure 2. GPIO_Interrupt Example Firmware Flow



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_Switch pin directly as shown in Figure 3. This results in Pin_Switch 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 3. Direct Connection of Input Pin to SysInt Component



CY8CKIT-062 BLE has one RGB LED module. Red and green LEDs of the RGB LED module are used in this project. The time between interrupts and the rate of blinking of LEDs is kept sufficiently high to observe the transition in the LED state; see Software Setup.

Hardware Setup

No special hardware setup is required for CY8CKIT-062-BLE. Connect the kit to your computer using the USB cable provided. The on-board RGB LED module (LED5) contains the green and red LEDs required to view the project's outputs.

Leave the switches and jumpers in their default positions. Default positions of relevant switches are as shown in Table 1.

Table 1. Switch Selection

Switch / Jumper	Position
SW5	3.3V
SW6	PSoC 6 BLE
SW7	VDDD / KitProg2

Software Setup

Macro values can be altered to change the blinking frequency of the green LED and the number of seconds the LED blinks according to user preference.

```
#define DELAY_SHORT          (250)  /*delay of 250ms between toggling corresponding to a blinking  frequency of 2Hz*/
#define DELAY_LONG           (500)  /*delay of 500ms between toggling corresponding to a blinking frequency of 1Hz*/
#define NUM_BLINK_SECONDS    (4)  /*Time in seconds for which CPU is active  */
```

Operation

Plug the CY8CKIT-062 BLE Pioneer Kit into your computer's USB port.

- Build the project and program it into the PSoC 6 MCU device. For more information on device programming, see PSoC Creator Help. Flash for both cores is programmed in a single program operation.
- Confirm that the green LED blinks at an interval of 1 second and turns OFF after four seconds. The red LED then turns ON, indicating that CPU has entered Deep Sleep.
- Press the switch connected to P0.4 to trigger an interrupt. This should cause the device to wake up and the red LED to switch OFF. The ISR executes and updates a flag, causing the green LED to resume blinking at a new interval of 500 ms (the faster blink rate to indicate that the ISR has executed). The green LED blinks for four seconds and the device enters Deep Sleep again (green LED stops blinking, red LED turns ON).
- Pressing the switch again repeats the wakeup cycle and the green 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 ms.

Components

Table 2 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

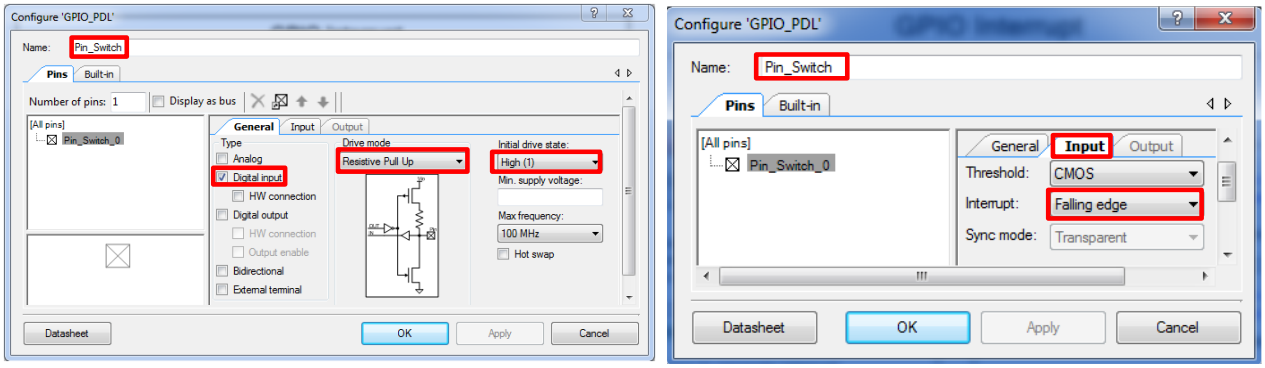
Table 2. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
Global Signal Reference	GlobalSignal	v1.0	1 Multi Counter Watch Dog Timer
GPIO	Pin_Switch	v1.0	1 Digital input pin
	Pin_CPU_Active_GreenLED	v1.0	1 Digital Output Pin
	Pin_CPU_DeepSleep_RedLED	v1.0	1 Digital Output Pin
SysInt	SysInt_SW	v1.0	1 Interrupt Component

Parameter Settings

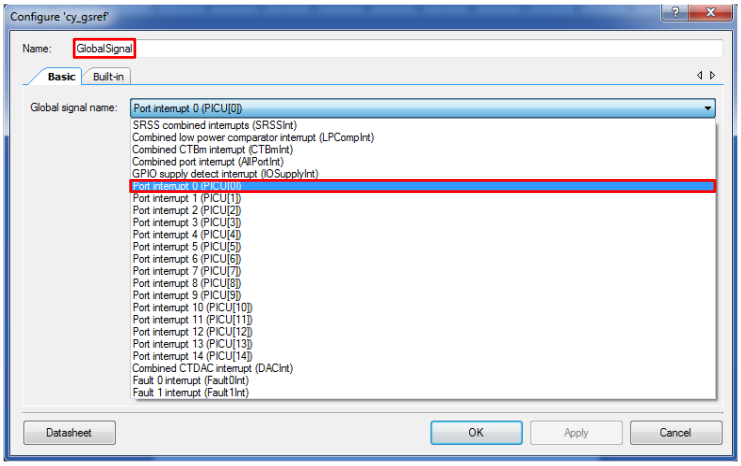
Figure 4 to Figure 7 shows the configuration for the Components used in the PSoC Creator project. For more information on Component configuration options, refer to respective component datasheets.

Figure 4. GPIO_SW Component Configuration



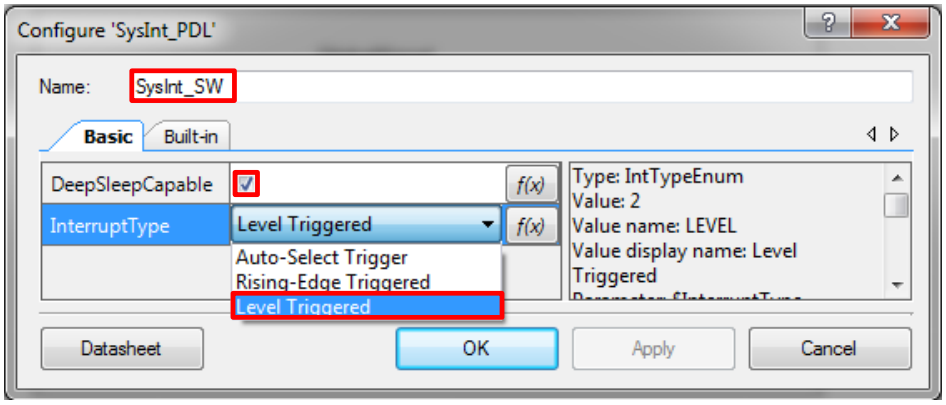
Pin_Switch is connected to the active LOW switch SW2 on CY8CKIT-062 BLE Pioneer Kit and therefore configured with a resistive pull-up drive mode. The switch when pressed results in logic 0 and provides a falling-edge signal.

Figure 5. Global Signal Reference Component Configuration



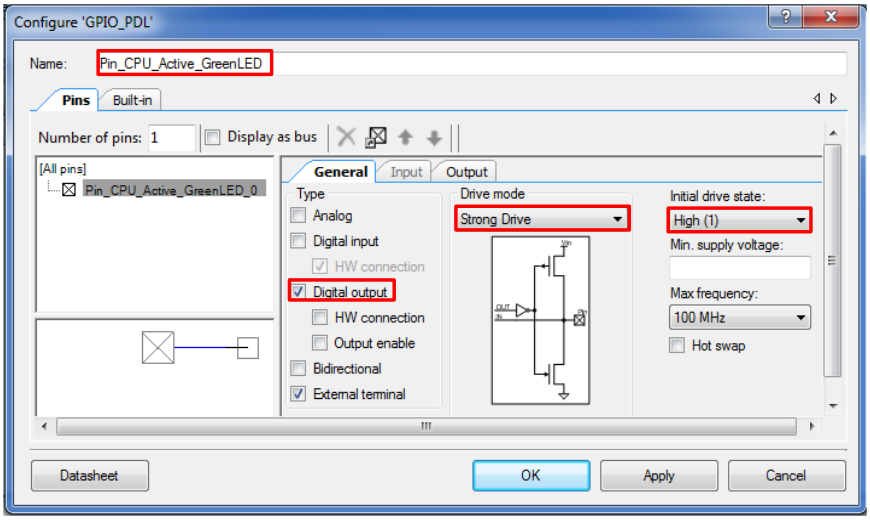
The switch is connected to P0.4 (Port 0, Pin 4). Port Interrupt 0 is selected from the GlobalSignal Component to route this signal to the SysInt Interrupt Component.

Figure 6. SysInt Component Configuration



The Port Interrupt 0 signal is capable of waking up the CPU from Deep Sleep, so the DeepSleepCapable option is selected in the Sysint Component configuration. InterruptType is selected as 'Level Triggered' because the SysInt Component is connected to a dedicated interrupt source (Port Interrupt).

Figure 7. GPIO Output Pin Configuration



Output pins are connected to active LOW-driven LEDs. The output pins are driven with logic 1 to ensure that the LEDs are OFF initially at boot up. A similar configuration is made for Pin_CPU_DeepSleep_RedLED.

Design-Wide / Global Resources

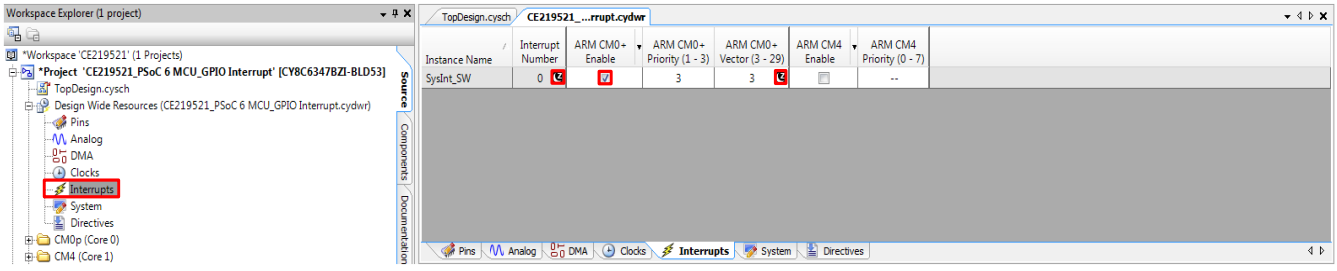
Table 3 shows the pin assignments for the switch and LEDs of CY8CKIT-062 BLE.

Table 3. DWR Pin Assignment Table

Component	Instance Name	Pin
Digital Input Pin	Pin_Switch	P0[4]
Digital Output Pin	Pin_CPU_Active_GreenLED	P1[1]
Digital Output Pin	Pin_CPU_DeepSleep_RedLED	P0[3]

Figure 8 shows the interrupt configuration for the project. It can be seen that the SysInt Component is assigned to CM0+ through the checkbox. CM0+ vector number 3 is chosen to support Deep Sleep wakeup (Vectors 0 – 7 are Deep-Sleep-capable). Interrupt priority is left to its default value because there are no other interrupts. For more information on configuring interrupts using the DWR, refer to AN217666 – PSoC 6 MCU Interrupts.

Figure 8. System Interrupt Configuration



Related Documents

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

Table 4. Related Documents

Application Notes		
AN210781	Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes the PSoC 6 MCU with BLE Connectivity and how to build the code examples.
AN217666	PSoC 6 MCU Interrupts	Presents theory and configuration details related to interrupts in PSoC 6 MCU
PSoC Creator Component Datasheets		
Global Signal Reference	Provides connections to device global signal and shared resource Interrupts.	
GPIO	Supports connection of hardware resources to physical pins	
SysInt	Provides SysInt component settings	
Device Documentation		
PSoC 6 MCU: PSoC 63 with BLE Datasheet (PRELIMINARY) PSoC 6 MCU: PSoC 62 Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual	
Development Kit (DVK) Documentation		
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit		

.

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

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

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