CE219521 – PSoC 6 MCU - GPIO Interrupt

Objective

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

Requirements

Tool: ModusToolbox™ IDE 1.0

Programming Language: C

Associated Parts: All PSoC 6 MCU parts

Related Hardware: PSoC 6 BLE Pioneer Kit, PSoC 6 WiFi-BT Pioneer Kit, PSoC6 WiFi-Prototyping Kit

Overview

This code example demonstrates the use of GPIO configured as an input pin to generate interrupts on 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 PSoC 6 BLE Pioneer Kit's default configuration. Refer to the kit guide to ensure the kit is configured correctly.

Note: The PSoC 6 BLE Pioneer kit and the PSoC 6 WiFi-BT Pioneer kit ship with KitProg2. ModusToolbox only works with KitProg3. Before using this code example, make sure that the kit is upgraded to KitProg3. See ModusToolbox Help > ModusToolbox IDE Documentation > User Guide; section PSoC 6 MCU KitProg Firmware Loader. If you do not upgrade, you will see an error like "unable to find CMSIS-DAP device" or "KitProg firmware is out of date".

Software Setup

None.

Operation

1. Connect the kit to your PC using the provided USB cable.
2. Import the code example into a new workspace. See KBA225201.
3. Program the PSoC 6 MCU device. In the project explorer, select the **mainapp** project. In the Quick Panel, scroll to the **Launches** section and click the **Program (KitProg3)** configuration.
4. Confirm that the LED blinks four times and then turns OFF, indicating that the CPU has entered Deep Sleep.
5. 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 (the faster blink rate to indicate that the ISR has executed). The LED blinks for four times and the device enters Deep Sleep again.
6. 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.

Debugging

You can debug the example to step through the code. Use the **Debug (KitProg3)** configuration. If you are unfamiliar with how to start a debug session with ModusToolbox IDE, see KBA224621.

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. In this example. 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 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 the intervals of 500 milliseconds and 1 second.

Resources and Settings

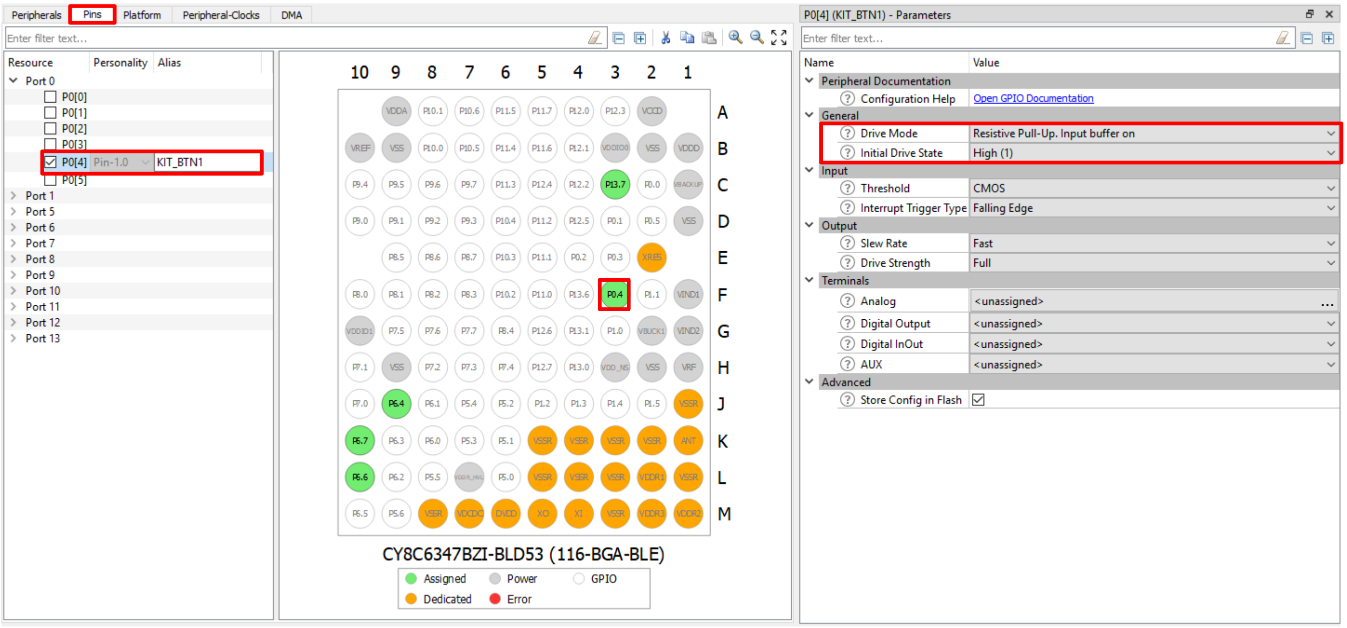
Table 1 lists some of the ModusToolbox resources used in the example, and how they are used in the design. The *design.modus* file contains all the configuration settings. For example, for pin usage and configuration, open the **Pins** tab of the design file.

Table 1. ModusToolbox Resources

Resource	Alias	Purpose	Non-default Settings
Digital Input Pin	KIT_BTN1	Provide user interface	See Figure 1
Digital Output Pin	KIT_LED2	Provide visual feedback	See Figure 2

Figure 1 and Figure 2 highlight the non-default settings for each resource in this example.

Figure 1. GPIO Configuration for Switch Input

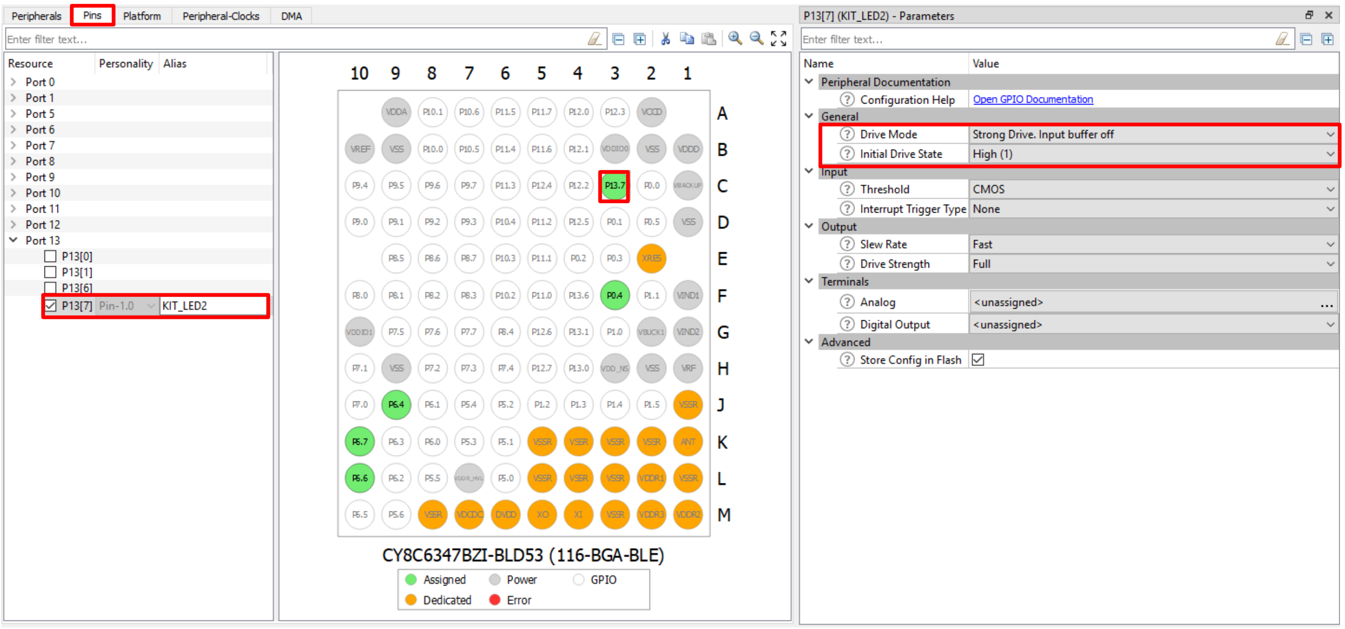


Resource	Personality	Alias
Port 0		
□ P0[0]		
□ P0[1]		
□ P0[2]		
□ P0[3]		
☑ P0[4]	Pin-1.0	KIT_BTN1
□ P0[5]		
Port 1		
Port 5		
Port 6		
Port 7		
Port 8		
Port 9		
Port 10		
Port 11		
Port 12		
Port 13		

P0[4] (KIT_BTN1) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
General	
Drive Mode	Resistive Pull-Up. Input buffer on
Initial Drive State	High (1)
Input	
Threshold	CMOS
Interrupt Trigger Type	Falling Edge
Output	
Slew Rate	Fast
Drive Strength	Full
Terminals	
Analog	<unassigned>
Digital Output	<unassigned>
Digital InOut	<unassigned>
AUX	<unassigned>
Advanced	
Store Config in Flash	☑

Figure 2. GPIO Pin Configuration for LED



Resource	Personality	Alias
Port 0		
Port 1		
Port 5		
Port 6		
Port 7		
Port 8		
Port 9		
Port 10		
Port 11		
Port 12		
Port 13		
□ P13[0]		
□ P13[1]		
□ P13[6]		
☑ P13[7]	Pin-1.0	KIT_LED2

10	9	8	7	6	5	4	3	2	1	
	VDDA	P10.1	P10.6	P11.5	P11.7	P12.0	P12.3	VCCD		A
VREF	VSS	P10.0	P10.5	P11.4	P11.6	P12.1	VDDIO0	VSS	VDDD	B
P9.4	P9.5	P9.6	P9.7	P11.3	P12.4	P12.2	P13.7	P0.0	VBACKUP	C
P9.0	P9.1	P9.2	P9.3	P10.4	P11.2	P12.5	P0.1	P0.5	VSS	D
	P8.5	P8.6	P8.7	P10.3	P11.1	P0.2	P0.3	XRES		E
P8.0	P8.1	P8.2	P8.3	P10.2	P11.0	P13.6	P0.4	P1.1	VIND1	F
VDDIO1	P7.5	P7.6	P7.7	P8.4	P12.6	P13.1	P1.0	VBUCK1	VIND2	G
P7.1	VSS	P7.2	P7.3	P7.4	P12.7	P13.0	VDD_NS	VSS	VRF	H
P7.0	P6.4	P6.1	P5.4	P5.2	P1.2	P1.3	P1.4	P1.5	VSSR	J
P6.7	P6.3	P6.0	P5.3	P5.1	VSSR	VSSR	VSSR	VSSR	ANT	K
P6.6	P6.2	P5.5	VDDR_HVL	P5.0	VSSR	VSSR	VSSR	VDDR1	VSSR	L
P6.5	P5.6	VSSR	VDCDC	DVDD	XO	XI	VSSR	VDDR3	VDDR2	M

CY8C6347BZI-BLD53 (116-BGA-BLE)

● Assigned ● Power ○ GPIO
● Dedicated ● Error

P13[7] (KIT_LED2) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
General	
Drive Mode	Strong Drive. Input buffer off
Initial Drive State	High (1)
Input	
Threshold	CMOS
Interrupt Trigger Type	None
Output	
Slew Rate	Fast
Drive Strength	Full
Terminals	
Analog	<unassigned>
Digital Output	<unassigned>
Advanced	
Store Config in Flash	☑

Reusing This Example

This example is configured for the supported kit(s). To port the design to a different PSoC 6 MCU device, right-click an application project and choose **Change Device**. If changing to a different kit, you may need to reassign pins.

Table 2. Device and Pin Mapping across PSoC 6 MCU Kits

Kit Name	Device Used	KIT_LED2	KIT_BTN1
CY8CKIT-062-WiFi-BT	CY8C6247BZI-D54	P13[7]	P0[4]
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	P13[7]	P0[4]
CY8CPROTO-062-4343W	CY8C624ABZI-D44	P13[7]	P0[4]

In some cases, a resource used by a code example (for example, an IP block) is not supported on another device. In that case the example will not work. If you build the code targeted at such a device, you will get errors. See the device datasheet for information on which resources a device supports.

Related Documents

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 and ModusToolbox application
AN221774 – Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first PSoC Creator project ModusToolbox application
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
Code Examples	
Visit the Cypress GitHub site for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kits	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	
CY8CPROTO-062-4343W PSoC 6 Wi-Fi BT Prototyping Kit	
Tool Documentation	
ModusToolbox IDE	The Cypress IDE for IoT designers

Cypress Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right device, and quickly and effectively integrate the device into your design.

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

Document History

Document Title: CE219521 - PSoC 6 MCU - GPIO Interrupt

Document Number: 002-25556

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6376846	AJYA	11/16/2018	New code example

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 | Code Examples | 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 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.