

Objective

This example demonstrates how to generate a periodic interrupt using the timer/counter pulse-width modulation (TCPWM) Component in Timer/Counter mode for PSoC® 6 MCU devices.

Overview

This example uses the TCPWM Component in the Timer/Counter mode to generate a periodic interrupt. An LED toggles whenever the interrupt occurs.

Requirements

Tool: PSoC Creator™ 4.2 with Peripheral Driver Library (PDL) 3.0.1

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

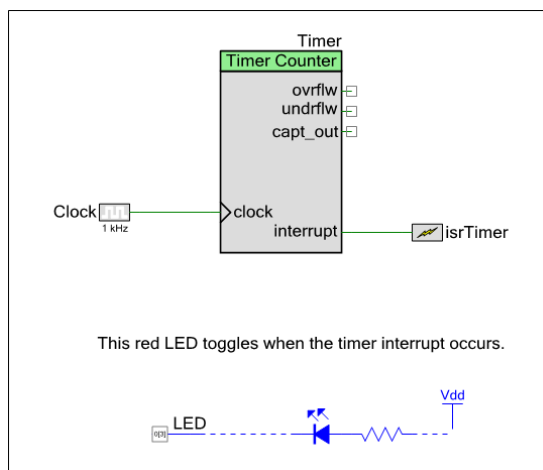
Associated Parts: All PSoC 6 MCU parts

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

This example configures an instance of the TCPWM Component in continuous up counter mode to generate a periodic interrupt. The CPU enters sleep mode. It wakes up whenever the interrupt occurs and reenters sleep mode after servicing the interrupt. The interrupt handler simply toggles an LED. You can change the timer period by changing the `TIMER_PERIOD_MSEC` macro in the `main_cm4.c` file. Figure 1 shows the PSoC Creator schematic for this code example.

Figure 1. TopDesign Schematic



Design Considerations

This example runs on CY8CKIT-062-BLE. To port the design to other PSoC 6 MCU family devices and kits, change the target device in the PSoC Creator Device Selector and the pin assignments in the `.cydwr` file.

Hardware Setup

This example uses the CY8CKIT-062-BLE kit's default configuration. Refer to the kit guide to ensure the kit is configured correctly.

Operation

1. Plug the CY8CKIT-062-BLE kit board 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. The red LED toggles at one second interval.
4. Change the timer period by modifying the `TIMER_PERIOD_MSEC` macro in `main_cm4.c` file, program the device, and observe that the LED now blinks at a different rate.

Components

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

Table 1. PSoC Creator Components

Component	Instance Name	Hardware Resources
Digital Output Pin	LED	1 GPIO pin
Timer Counter (TCPWM)	Timer	1 counter instance of a TCPWM block
Clock	Clock	1 peripheral clock divider
Interrupt	isrTimer	None. It only generates the configuration for the TCPWM interrupt

Parameter Settings

Figure 2, Figure 3, and Figure 4 show the Component configuration with the changed settings highlighted. See the corresponding Component datasheet for information on these settings. To access the datasheet, right-click the Component and click **Open Datasheet**.

Figure 2. Digital Output Pin Component Configuration

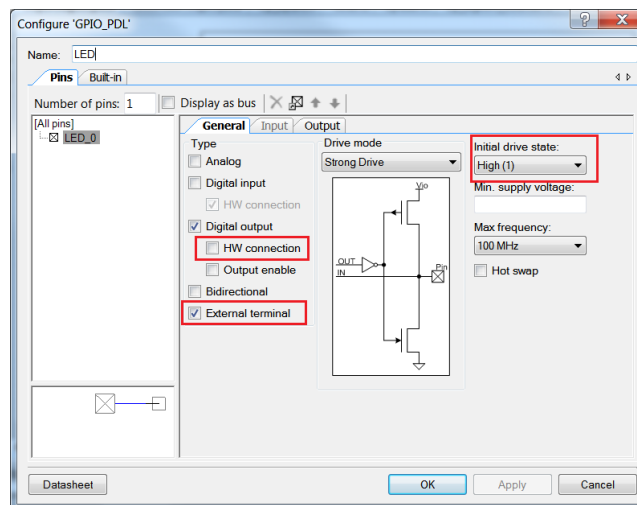


Figure 3. Timer Counter (TCPWM) Component Configuration

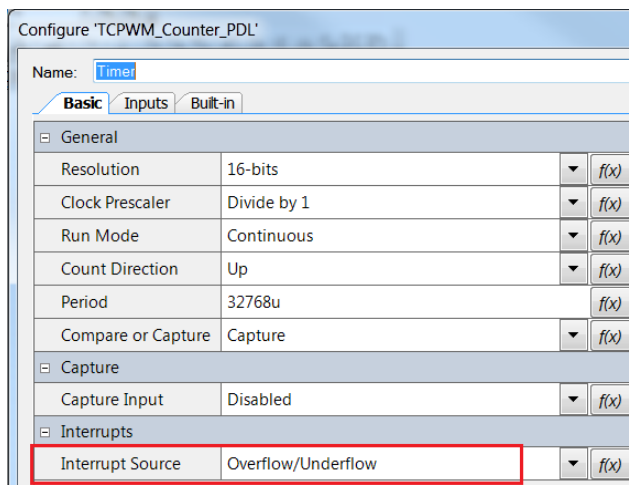
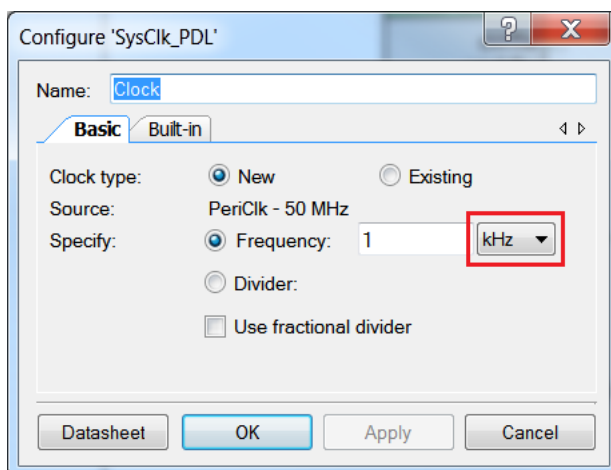


Figure 4. Clock Component Configuration



Design-Wide Resources

Figure 5 shows the pin assignment for the project done through the **Pins** tab in the **Design Wide Resources** window. These assignments are compatible with CY8CKIT-062-BLE.

Figure 5. Pin Assignments


	Name	Port	Pin	Lock
	LED	P0 [3]	E3	<input checked="" type="checkbox"/>

Figure 6 shows the interrupt assignment for the project done through the **Interrupts** tab in the **Design Wide Resources** window. PSoC Creator assigns the interrupt to CM4 by default, which works for this project because CM4 services the periodic interrupt generated by the TCPWM Component.

Figure 6. Interrupt Assignments

CE22016...pt.cydwr					
Instance Name	Interrupt Number	ARM CM0+ Enable	ARM CM0+ Priority (1 - 3)	ARM CM0+ Vector (3 - 29)	ARM CM4 Enable
isrTimer	90	<input type="checkbox"/>	--	--	<input checked="" type="checkbox"/>
					ARM CM4 Priority (0 - 7)
					7

Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with BLE Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 PSoC 6 MCU Dual-Core CPU system Design	Describes the dual-core CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-core 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
Timer Counter (TCPWM)	Supports fixed-function Timer/Counter implementation
Clock	Supports local clock generation
Interrupt	Supports generating interrupts from hardware signals
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE 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: CE220169 - PSoC 6 MCU: Periodic Interrupt using TCPWM

Document Number: 002-20169

Revision	ECN	Orig. of Change	Submission Date	Description of Change
*B	5891665	VAIR	09/21/2017	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

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

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

© Cypress Semiconductor Corporation, 2017. 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. 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.