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.

CE224703 – PSoC™ 4 Watchdog Timer

Objective

This example shows how to use a watchdog timer (WDT) to initiate system reset in a PSoC™ 4 device.

Overview

This example demonstrates the use of a WDT to keep track of count until device reset and identifying the source of a reset. The status of the WDT and reset sources is printed using the USB-UART bridge in a Cypress PSoC™ 4 development kit.

Requirements

Tool: PSoC Creator™ 4.4

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: PSoC™ 4

Related Hardware: CY8CKIT-042 PSoC 4 Pioneer Kit

Hardware Setup

This code example is set up for CY8CKIT-042. If you are using a different kit, see Reusing This Example.

For CY8CKIT-042, the USB-UART bridge in KitProg2 module is used:

1. Connect the \UART:rx\ pin P0[4] to P12[7] on header J8.
2. Connect the \UART:tx\ pin P0[5] to P12[6] on header J8.

Other kits use different pins for the UART. Make sure that you select the pins that are right for your kit.

Software Setup

This design requires a terminal emulator such as PuTTY or Tera Term running on your computer.

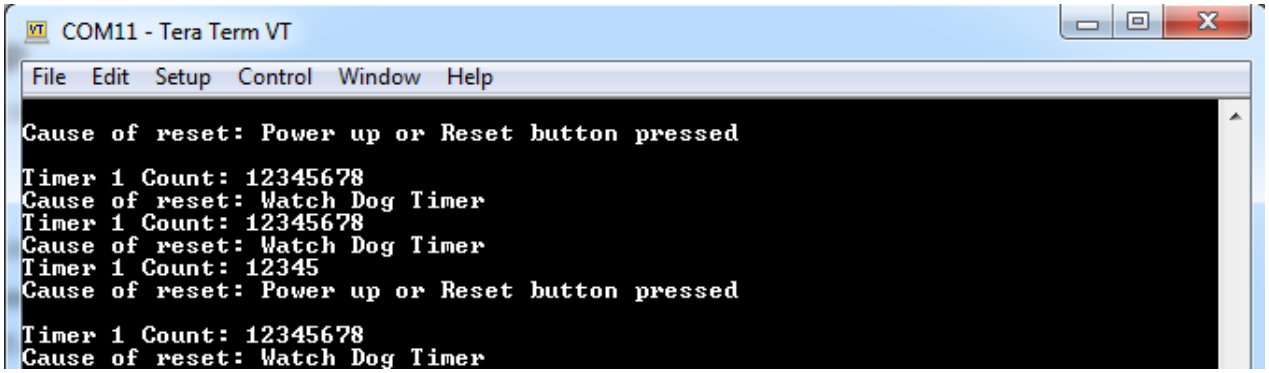
Operation

Do the following:

1. Confirm the correct kit USB-UART bridge connections, as noted in Hardware Setup.
2. Connect CY8CKIT-042 to your computer using a USB cable.
3. Build the project and program it into the PSoC 4 device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
4. Open a terminal emulator on your computer and configure the program to the appropriate COM port. Configure the baud rate to 115200, 8 data bits, no parity bits, 1 stop bit, and no control flow.

5. Confirm that the count displayed in the terminal window matches what Figure 1 shows, followed by an indication of a WDT reset. Push the reset button on the kit and confirm a different cause of reset.

Figure 1 Message Printed on the Terminal



Design and Implementation

This example demonstrates the setup and use of the WDT to reset the PSoC 4 device. The WDT and the kit reset button both cause device reset. The cause of reset is displayed at startup. The WDT counters can be configured to determine the WDT timeout period as well as periodic interrupt rate.

In this example, the following functions are performed:

1. The cause/source of reset is printed.
2. Interrupt number is set with ISR_Watchdog as the interrupt handler.
3. WDT counter 0 is set to generate an interrupt on match.
4. WDT counter 0 and 1 cascade is enabled.
5. WDT counter 1 is set to generate reset on match.
6. Both counter 0 and 1 are enabled.
7. Count is monitored through interrupt and printed via UART if changed.

The ISR_Watchdog function does the following:

1. Sets the flag to print the current count status of WDT.
2. Clears the interrupt state.

Note: The time that the WDT waits before reset depends on the parameters of the API function `CySysWdtWriteMatch()`. You can change the code to determine both speed of count and the count value at WDT reset.

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.

Table 1. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
UART (SCB Mode) (Polling)	UART	Handle UART serial communication	None

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

Reusing This Example

This example is designed for the CY8CKIT-042 pioneer kit. To port this design to a different PSoC 4 device, kit, or, both, do the following:

1. In PSoC Creator IDE, select **Project** > **Device Selector** to change the target device. Select your device as listed in Table 2.
2. Make sure that the **SysClk Desired frequency** is set to 24 MHz after the device is changed.
3. In the PSoC Creator Workspace Explorer, select the **Clocks** interface listed under **Design Wide Resources**.
4. Set the **SysClk Desired Frequency** to 24 MHz, if it is not already.

Table 2. Development Kits and Associated Devices

Development Kit	Device
CY8CKIT-042	CY8C4245AXI-483
CY8CKIT-042-BLE-A	CY8C4248LQI-BL583
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L489

5. In CYDWR file, route the \UART:tx\ and \UART:rx\ to the pins based on the selected kit. See Table 3 for pin details.

Table 3. Pin Assignments for Different Kits

Pin Name	Development Kit			
	CY8CKIT-042	CY8CKIT-042-BLE-A	CY8CKIT-044	CY8CKIT-046
\UART:rx\	P0[4]	P1[4]	P7[0]	P3[0]
\UART:tx\	P0[5]	P1[5]	P7[1]	P3[1]

Additionally, for CY8CKIT-042, the PSoC™ 4 UART signals need to be physically connected to KitProg2 (PSoC™ 5LP) UART signals using jumper wires. Connect P12[7] and P12[6] on header J8 to P0[4] and P0[5] on header J4. For other kits, the pins are already connected.

6. Change the value of the macro `interruptNumWDT` in main.c as listed in Table 4.

Table 4 Interrupt number for different kits

Macro name	Development Kit			
	CY8CKIT-042	CY8CKIT-042-BLE-A	CY8CKIT-044	CY8CKIT-046
WDT interrupt number	9	8	7	8

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 what a device supports.

Related Documents

Application Notes	
AN79953 Getting Started with PSoC® 4	Describes PSoC 4 devices and shows how to build the attached code example
AN90799 PSoC® 4 Interrupts	Explains the interrupt architecture in PSoC 4 and its configuration in the PSoC Creator™ IDE with the help of three example projects.
PSoC Creator Component Datasheets	
PSoC 4 Serial Communication Block (SCB)	A multifunction hardware block that implements the following communication components: I2C, SPI, UART, and EZI2C
Device Documentation	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals
Development Kit (DVK) Documentation	
CY8CKIT-042 PSoC® 4 Pioneer Kit	
PSoC 4 Kits	
Tool Documentation	
PSoC™ Creator user guide	
PSoC™ Creator quick start guide	

Document History

Document Title: CE224703– PSoC™ 4 Watchdog Timer

Document Number: 002-24703

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6296219	SYAO	09/07/2018	New code example.
*A	8044661	SASH	07/30/2024	Updated setup details for all supported dev kits.

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

Technical Support

cypress.com/support

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

© Cypress Semiconductor Corporation, 2016-2024. 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.