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™ 4.2

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: PSoC 4 family

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 Hardware Setup.

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.

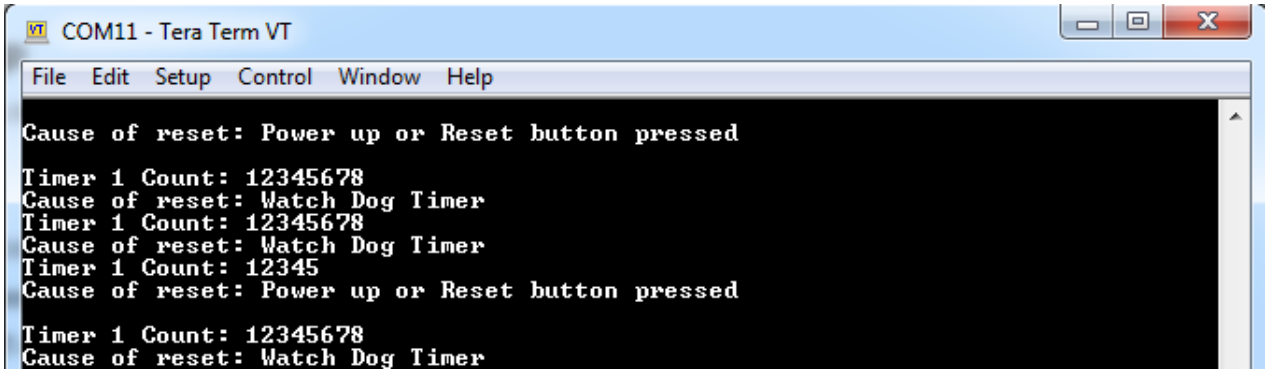
Operation

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-041	CY8C4146AZI-S433
CY8CKIT-042	CY8C4245AXI-483
CY8CKIT-042-BLE	CY8C4247LQI-BL483
CY8CKIT-044	CY8C4247AZI-M485
CY8CKIT-046	CY8C4248BZI-L489
CY8CKIT-048	CY8C4A45AZI-483

5. Route \UART:tx\ and \UART:rx\ to the pins listed in Table 3. For the CY8CKIT-048, install jumper wires for \UART:tx\ and \UART:rx\ to P12[6] and P12[7] on header J16, respectively.

Table 3. Pin Assignments for Different Kits

Pin Name	Development Kit					
	CY8CKIT-041	CY8CKIT-042	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046	CY8CKIT-048
\UART:rx\	P0[4]	P0[4]	P1[4]	P7[0]	P3[0]	P0[4]
\UART:tx\	P0[5]	P0[5]	P1[5]	P7[1]	P3[1]	P0[5]

For the CY8CKIT-048, connect \UART:tx\ to P12[6] and \UART:rx\ to P12[7] on header J16.

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	Go to the **Downloads** tab for Quick Start and User Guides

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

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 | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709