**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.

# CE220060 – PSoC 6 MCU Watchdog Timer

## Objective

This example demonstrates the Watchdog Timer (WDT) on PSoC® 6 MCU, using PSoC Creator™.

## Requirements

**Tool:** PSoC Creator™ 4.2; 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:** PSoC 6 BLE Pioneer Kit, PSoC 6 WiFi-BT Pioneer Kit

## Overview

This example explains the two use cases of WDT – as a watchdog that causes a device reset in the case of a malfunction, and as a periodic interrupt source.

A macro definition determines which mode to use. Out of the box, this example demonstrates the periodic interrupt. The blue LED toggles on every interrupt at an interval of ~1 s.

For reset mode, you change the macro definition and enable an infinite loop in the `main()` function, to block execution. The device resets in ~6 s. The green LED blinks after the device comes out of reset. If you use reset mode without blocking execution, the device does not reset. The red LED toggles every 1 s in the main loop to indicate that the CPU is in action.

In addition, the red LED blinks once after power cycling or an external reset event.

## Hardware Setup

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

## Software Setup

See the Operation section for information on how to modify the code for each demo.

## Operation

1. Connect the Pioneer kit to your PC using the provided USB cable.
2. Open the PSoC Creator project
3. Modify the source code if you want to use RESET mode. By default, the example is in INTERRUPT mode.
   a. Open *main.c* from CE220060 in the workspace.
   b. Set `WDT_DEMO to WDT_RESET_DEMO` for reset demonstration
   c. In the innermost "for" loop, uncomment the line of code //while(1); to cause the reset to happen.
4. When `WDT_INTERRUPT_DEMO` is selected, you can optionally put the device in Deep Sleep mode by uncommenting the function call `Cy_SysPm_DeepSleep()` in the main loop. The device wakes up from Deep Sleep mode on a WDT interrupt.
5. 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.
6. Observe the status of LEDs based on different events summarized in Table 1.

Table 1. LED Status

| Project Setting | LED Status |
|---|---|
| `WDT_DEMO` set to `WDT_INTERRUPT_DEMO` | Blue LED toggles on every WDT interrupt (interval of 1 s) |
| `WDT_DEMO` set to `WDT_RESET_DEMO` with blocking function | After approximately 6 s, the device resets and green LED blinks to indicate WDT reset. |
| `WDT_DEMO` set to `WDT_RESET_DEMO` without blocking function | Red LED toggles every 1 s to indicate that the CPU is in action. |

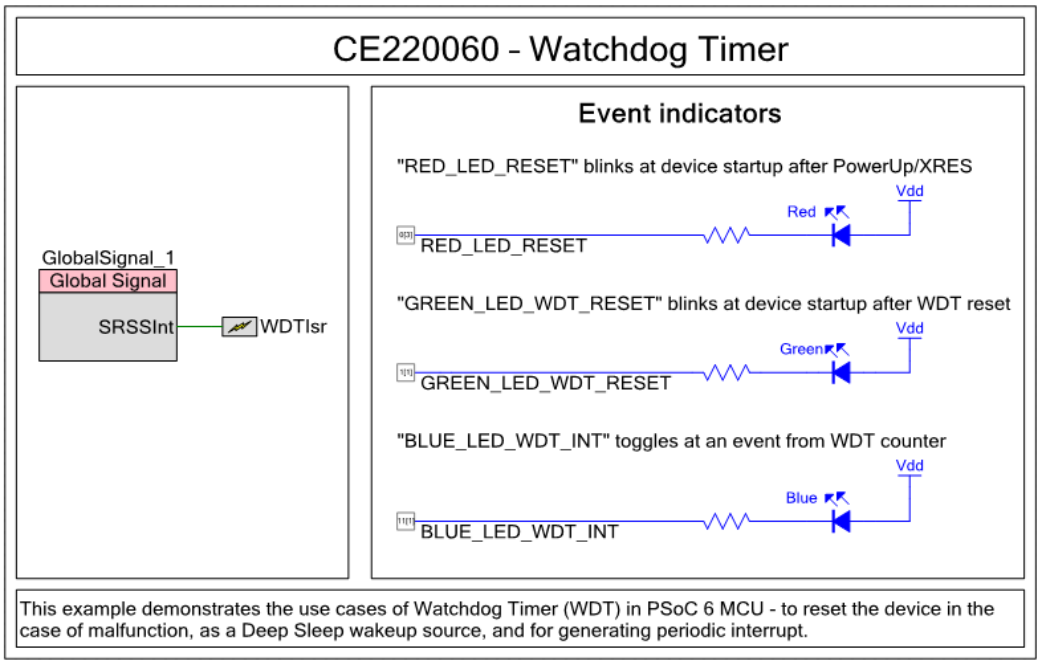
Note that red LED blinks once after power cycle or an external reset event.

## Design and Implementation

The WDT in PSoC 6 MCU is a 16-bit timer and uses the Internal Low-Speed Oscillator (ILO) clock of 32 kHz. It is primarily used as a watchdog against any system malfunction, but can also be used for generating periodic interrupt.

In PSoC Creator, the Global Signal Component and Interrupt Component are used. The Global Signal Component allows access to the interrupt signal generated by the WDT; this signal is routed to the Interrupt Component for triggering the CPU.

Figure 1. Top Design Schematic




In firmware, WDT is configured in eight steps in the project as follows:

1. Unlock the WDT to enable configuration.
2. Set the ignore bits for the match resolution. In the project, it is set to 0; that means full 16-bit resolution for the match count which gives an interval of 2.048 s ($2^{16} \div 32$ kHz) for the match event.
3. Write the match value. The WDT can generate an interrupt (if enabled) when the WDT counter reaches the match count. The project configures the match count using the macro `WDT_MATCH_COUNT`. Note that the interrupt handler modifies the match count when `WDT_INTERRUPT_DEMO` is selected to generate a periodic interrupt[1] every 1 s. The match count, however, is set to zero in `WDT_RESET_DEMO` mode to generate match events at equal intervals of time[2]. The device resets after three WDT match events, that is, 2.048 s x 3 = 6.144 s, if the match event is not cleared. You can reduce the duration of the match event by introducing ignore bits in Step 2.
4. Clear the pending WDT interrupt.

5. Enable the ILO, which is the source for the WDT.
6. Enable the interrupt generation if `WDT_INTERRUPT_DEMO` mode is selected and assign the handler. Interrupt should be disabled when using the `WDT_RESET_DEMO` mode[2].
7. Enable the WDT.
8. Lock the WDT to prevent inadvertent changes.

**Note:**

1. The WDT generates an interrupt on match. However, the counter is not reset on match. It continues to count across the full 16-bit resolution. For this reason, the match count is updated on every WDT interrupt, when `WDT_INTERRUPT_DEMO` is selected.
2. Interrupt should not be enabled when reset mode of the WDT is being used. If interrupt is enabled, upon an interrupt, it needs to be cleared to avoid repeated execution of the WDT interrupt handler. This removes the actual purpose of the WDT. Thus, interrupt generation should never be enabled and the WDT match event should always be cleared in the main loop.

### Components and Settings

Table 2 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

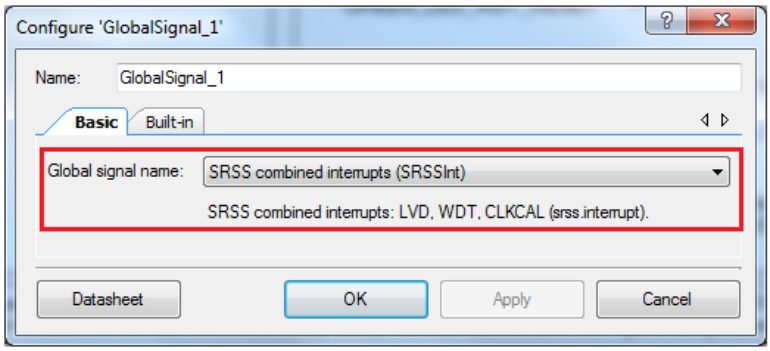
Table 2. PSoC Creator Components

| Component | Instance Name | Purpose | Non-default Settings |
|---|---|---|---|
| Global Signal Reference (GSRef) | GlobalSignal_1 | Used to access WDT interrupt signal | Global signal name set to **SRSS combined interrupts (SRSSInt)** |
| System Interrupt (SysInt) | WDTIsr | Used for generating CPU interrupt | None |

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

Figure 2 highlights the non-default settings for the GlobalSignal Component.

Figure 2. GlobalSignal Component Configuration




## Reusing This Example

This example is designed for the PSoC 6 BLE pioneer kit. To port the design to a different PSoC 6 MCU device and/or kit, change the target device using the Device Selector and update the pin assignments in the Design Wide Resources Pins settings as needed. For single-core PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c*.

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 particular device supports.

## Related Documents

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

For a comprehensive list of PSoC 3, PSoC 4, and PSoC 5LP resources, see KBA86521 in the Cypress community.

| **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 |
| 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 |
| 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 |
| Global Signal Reference | Supports connection to device global signals |
| 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 |
| PSoC 6 MCU: PSoC 62 Datasheet | PSoC 6 MCU: PSoC 62 Architecture Technical Reference Manual (TRM) |
| **Development Kit Documentation** | |
| CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit | |
| CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit | |
| **Tool Documentation** | |
| PSoC Creator | Look in the downloads tab for Quick Start and User Guides |
| WICED SDK | Installed with the SDK |
| Peripheral Driver Library (PDL) | Installed by PSoC Creator 4.2. Look in the <PDL install folder>/doc for the User Guide and the API Reference |

## 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.

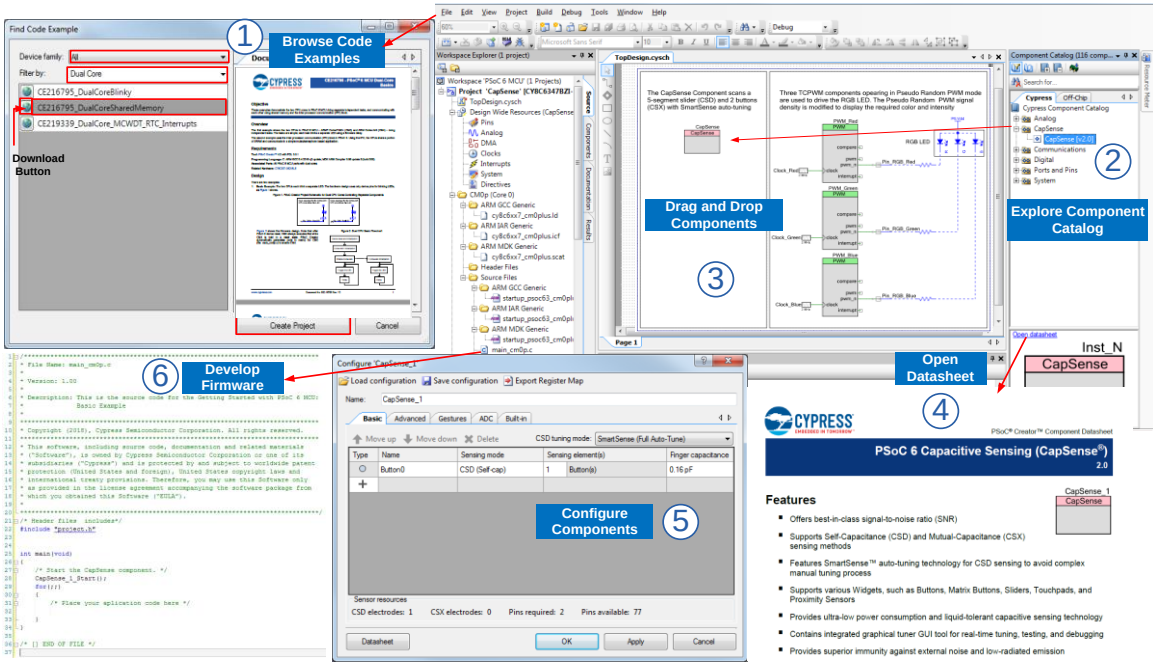
The following is an abbreviated list of resources related to this code example:

- **Overview:** MCU Portfolio, PSoC & MCU Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, PSoC 5LP, or PSoC 6. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for MCU and PSoC device families.
- **Application Notes:** Cover a broad range of topics, from basic to advanced level.
- **Code Examples**: for PSoC 3, PSoC 4, and PSoC 5LP; or for PSoC 6.
- PSoC **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers for a PSoC device family.
- **Training Videos**: These videos provide guidance on getting started with various Cypress product families and tools.
- **PSoC 6 MCU Training Videos**: Provide guidance on getting started.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications.
- **Development Kits:** Some examples include**:**
  - PSoC 6 BLE Pioneer Kit is a low-cost hardware platform that enables design and debug of the PSoC 63 series. It comes with an E-Ink display shield board.
  - PSoC 6 WiFi-BT Pioneer Kit supports the PSoC 62 series MCU along with Wi-Fi and BT connectivity
  - CY8CKIT-042 and CY8CKIT-040, Pioneer kits, are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
  - CY8CKIT-049 is a series of very low-cost prototyping platform for sampling PSoC 4 devices.
  - CY8CKIT-030 and CY8CKIT-050 are designed for analog performance. They enable you to evaluate, develop, and prototype high-precision analog, low-power, and low-voltage applications powered by PSoC 3 and PSoC 5LP, respectively.
  - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg3 device provides an interface for flash programming and debug.

## PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, PSoC 5LP, and PSoC 6 MCU. With PSoC Creator, you can:

1. Browse the collection of code examples from the **File** > **Code Example** menu.
2. Explore the library of 100+ Components
3. Drag and drop Components to build your hardware system design in the main design workspace
4. Review Component datasheets
5. Configure Components using configuration tools
6. Codesign your application firmware with the PSoC hardware



# Document History

Document Title: CE220060 – PSoC 6 MCU Watchdog Timer

Document Number: 002-20060

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | 5788931 | VJYA | 06/29/2017 | New code example |
| *A | 5856603 | VJYA | 08/23/2017 | Initial public release |
| *B | 5918159 | VJYA | 11/03/2017 | Updated project name |
| *C | 6372405 | RJVB | 02/01/2019 | Updated template<br>Updated PSoC Creator Project |

## 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

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

© Cypress Semiconductor Corporation, 2017-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("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 shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. 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. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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.