CE216259 – FM0+ Software Watchdog Reset

A watchdog reset is indicated by a permanently lit LED. The watchdog can be "fed", to delay the reset event, by pressing the user switch.

Overview

The application exercises the RESET and SWWDG (Software Watchdog) components, and uses a green LED to indicate the state of the application. If the application resets as a result of a watchdog underflow, the LED is lit permanently. Otherwise the watchdog is started with a 5 second period, which will result in a 10 second time period before it resets (the first timer underflow generates an IRQ (unhandled) and the second causes the reset). The reset can be delayed by pressing the user switch, which feeds the watchdog and blinks the LED.

Requirements

Tool: PSoC Creator 4.0 and Peripheral Driver Library (PDL) 2.1

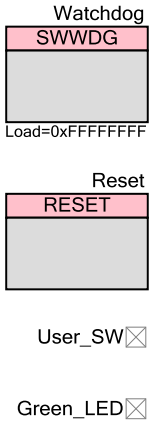
Programming Language: C (GCC 4.9.3)

Associated Parts: All S6E1 parts

Related Hardware: FM0-V48-S6E1A1 and FM0-64L-S6E1C3

Design

The schematic file includes the SWWDG, RESET and two GPIO Components, renamed as shown below.



The firmware performs following functions:

1. Initialize the LED (off) and user switch (up)
2. Read the reset cause
3. If reset was because of the watchdog, turn on the LED and loop forever (until user reset)
4. Initialize and start the watchdog
5. Feed the watchdog when the user presses the switch (and blink the LED)

Design Considerations

Pin Selection

The project includes control files to automatically place the user switch and LED IO onto the appropriate pins for the supported kit hardware. To change the pin selections, delete the control file or over-ride the control file selections in the Design Wide Resources Pin Editor.

PDL Installation

The project assumes that you have installed the PDL in the location specified in the **Project Management** panel of the **Tools** > **Options** dialog. If that location is incorrect you will see the build error "The given PDL path is invalid. Unable to find required PDSC file." To correct this problem in a newly-created project, open the **Project** > **Properties** dialog and enter the correct path to the PDL. To avoid the problem in projects you create in the future, make sure you put the correct path in the **Tools** > **Options** dialog.

Hardware Setup

The LED GPIO is connected to a green LED. The switch GPIO is connected to SW3 on FM0-V48-S6E1A1 and SW2 on both FM0-100L-S6E1B8 and FM0-64L-S6E1C3.

Table 1 lists the pin connections required to use this code example on FM0+ kits.

Table 1. List of Pins

Pin	FM0-V48-S6E1A1	FM0-64L-S6E1C3
User_SW:GPIO	P04	P30
Green_LED:GPIO	P61	P3E

Components

Table 2 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 2. List of PSoC Creator Components

Component	Version	Hardware Resources
PDL_SWWDG	1.0	None (enables driver)
PDL_RESET	1.0	None (enables driver)
PDL_GPIO	1.0	GPIO pin

Parameter Settings

All Components, except the PDL_SWWDG, use their default parameter settings. Only the Component instance names have been changed for readability.

The PDL_SWWDG Component uses default parameter settings, with these exceptions.

Table 3: Component Settings

Tab	Setting	Value
None	Name	Watchdog
Basic	bResetEnable	true
	pfnSwwdgIrqCb	<blank>

Note that the pointer to the watchdog IRQ callback function (pfnSwwdgIrqCb) is cleared. Without this change to the default setting, the user code is required to provide a callback handler function, which is not used in this example.

Operation

After programming, reset the board and wait 10 seconds for the watchdog to expire and reset the board. This is indicated by the LED turning on permanently.

Reset the board again and press the user switch to feed the watchdog. The LED is lit temporarily to indicate that the watchdog is being fed. If the watchdog is not fed for 10 seconds the device resets and the LED turns, and remains, on.

Related Documents

Table 4 lists relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 4. Related Documents

PSoC Creator Component Datasheets	
PDL_SWWDG	Supports watchdog interrupts, device reset, and window mode (right-click on the Component to access)
PDL_RESET	Enables determination of the reset cause (right-click on the Component to access)
PDL_GPIO	Supports firmware access to physical pins (right-click on the Component to access)
Device Documentation	
S6E1A	FM0+ S6E1A-Series 5 V Robust ARM® Cortex®-M0+ Microcontroller (MCU) Family
S6E1C	FM0+ S6E1C-Series Ultra Low Power ARM® Cortex®-M0+ Microcontroller (MCU) Family
Development Kit (DVK) Documentation	
FM0-V48-S6E1A1	ARM® Cortex®-M0+ FM0+ MCU Evaluation Board
FM0-64L-S6E1C3	ARM® Cortex®-M0+ MCU Starter Kit with USB and Digital Audio Interface

Document History

Document Title: CE216259 – FM0+ Software Watchdog Reset

Document Number: 002-16259

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5420957	YFS	08/30/16	New Code Example.
*A	5448732	YFS	9/29/16	Added workspace file.
*B	5776568	YFS	6/16/17	Added search keyword so that user can quickly find Code Examples from the component instance popup menu. Updated logo and copyright date.
*C	5988559	YFS	12/8/17	Removing S6E1B support.

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
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2016-2017. 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.