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.

CE224891 – PSoC 4 Down Counter 7-bit

Objective

This example shows how to use the Count7 Component as a down counter on a PSoC® 4 device.

Requirements

Tool: PSoC Creator 4.2

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: PSoC 4 devices with UDBs

Related Hardware: CY8CKIT-042 PSoC 4 Pioneer Kit

Overview

This example demonstrates the count-down and enable option features of the Count7 Component. A Control Register Component is used to drive the Count7 enable signal. The Count7 output is read both directly and through a Status Register Component. Both values are displayed on the terminal via UART. A kit button is used to control whether the Count7 Component is enabled.

Hardware Setup

This code example is set up for CY8CKIT-042. If you are using a different kit, see Reusing this Example. In this kit, the USB-UART bridge in the 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 correct pins for your kit.

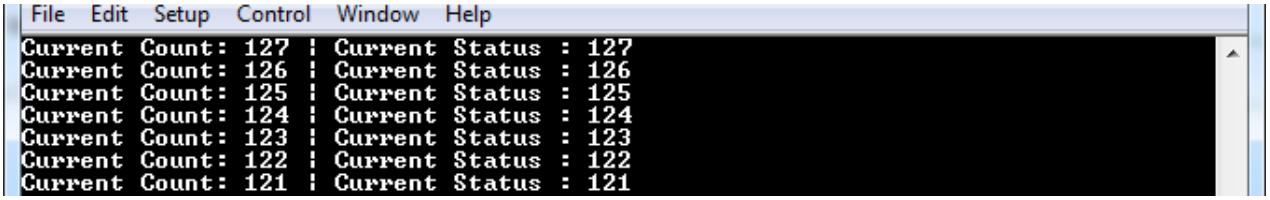
Software Setup

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

Operation

1. Ensure that the correct pins are connected for your kit, as noted in the Hardware Setup section.
2. Connect the USB cable between the PC and the PSoC 4 Pioneer Kit.
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, data bits to 8, no parity bits, stop bit as 1, and no control flow.
5. Confirm that the terminal display shows the Component counting down, and that the count and status register values are the same (Figure 1). Press kit button SW2 and confirm that counting stops and restarts.

Figure 1. Project Terminal Output



Design and Implementation

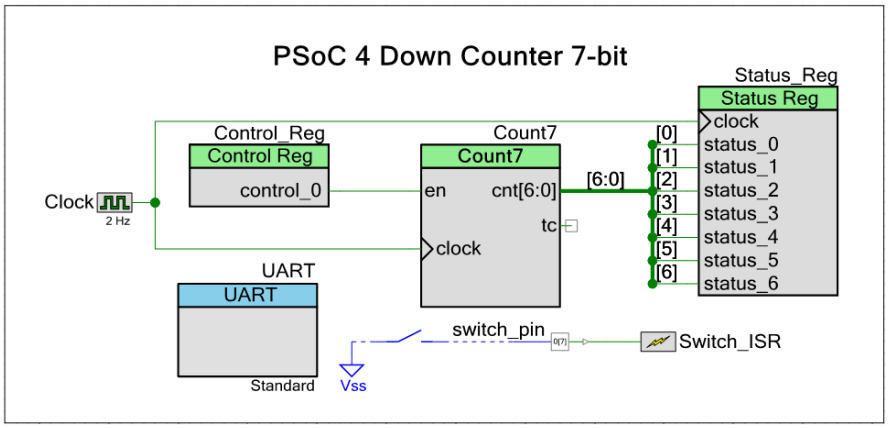
The Count7 timer period can be varied. In this example, Count7 is configured with a period of 128, so the first count is 127 and Count7 resets when it reaches 0. A kit button and a control register are used to enable the counter, as Figure 2 shows. The status register allows for reading of the counter output.

Firmware does the following:

1. Starts the UART and Count7 operation.
2. Configures the `Switch_InterruptHandler` interrupt handler.
3. Before obtaining the current value of the counter, the counter is stopped. The counter is restarted after getting a copy of the count.
4. The counter is read directly, and its digital outputs are read via the status register. If either value has changed, the UART displays both values. The two values should always be the same.

When the button is pressed, Switch_ISR interrupt occurs, `Switch_InterruptHandler` is called, and the Control register output is toggled.

Figure 2. PSoC Creator Project Schematic



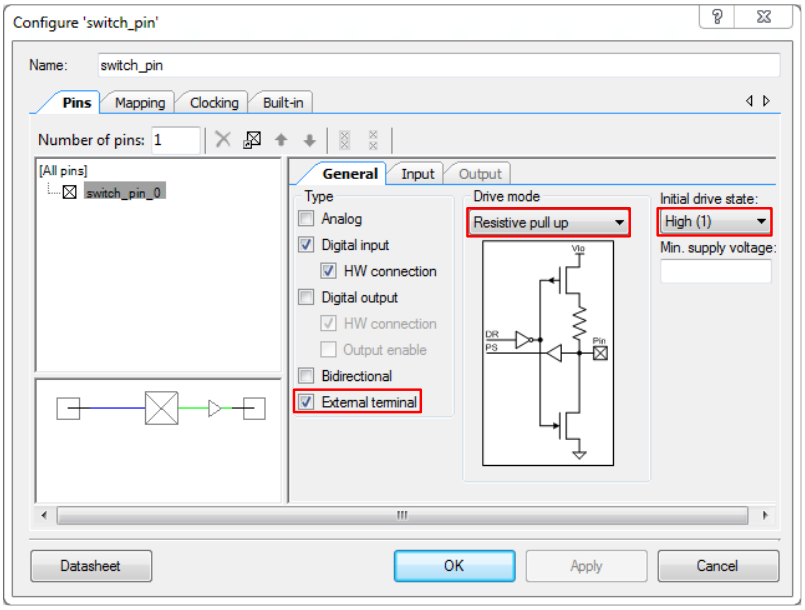
Components and Settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
Down Counter (7-bit) [v1.0]	Count7	Implements the down counter functionality	Set **EnableSignal** to Enabled
Control Register [v1.80]	Control_Reg	Allows for digital signal outputs	Set **Outputs** to 1
Status Register [v1.90]	Status_Reg	Allows for digital signals to be read	Set **Inputs** to 7
Digital Input Pin [v2.20]	Switch_pin	Handles the SW2 connection on the device	See Figure 3
UART (SCB mode) [v4.0]	UART	Handles UART communication	None
Interrupt [v1.70]	Switch_ISR	Handles Interrupt	None

Figure 3. switch_pin Parameter Settings



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 the design to a different PSoC 4 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. Table 2 shows pin assignments required for UART operation on other PSoC 4 devices.

Note: This project cannot be built for PSoC 4 devices with no UDBs.

Table 2. Pin Assignments for Different Kits

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

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 and shows how to build the attached code example
PSoC Creator Component Datasheets	
Down Counter 7-bit (Count7)	The Count7 Component is a 7-bit down counter available as hardware signals.
SCB	A multifunction hardware block that implements these communication components: I2C, SPI, UART, and EZI2C.
Control Register	The Control Register allows the firmware to output digital signals.
Status Register	The Status Register allows the firmware to read digital signals.
General Purpose Input/Output (GPIO)	A multifunctional Component that allows hardware resources to connect to a physical port pin and provides access to external signals through an appropriately configured physical I/O pin.
Interrupt	The interrupt Component defines hardware-triggered interrupts. There are three types of system interrupt waveforms that can be processed by the interrupt controller: Level, Pulse, and Edge.
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: CE224891 – PSoC 4 Down Counter 7-bit

Document Number: 002-24891

Revision	ECN	Submission Date	Description of Change
**	6302288	09/14/2018	New code example
*A	6802444	02/24/2020	Fixed the broken links.

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 | 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, 2018-2020. 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.