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.

CE223001 - PSoC 6 MCU UART printf

Objective

This example demonstrates how to use the `printf` function with a Serial Communication Block (SCB) based UART in PSoC® 6 MCU.

Overview

This example is designed to redirect the `printf` function to use the UART API. Project is designed to print number of times, the kit button SW2 is pressed, on the terminal.

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: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Hardware Setup

This example uses the kit's default configuration. See the kit guide to ensure the kit is configured correctly. If the settings are different from the default values, see the 'Selection Switches' table in the kit guide to reset to the default settings.

Table 2 lists the PSoC Creator pin connection settings required on the CY8CKIT-062-BLE Kit.

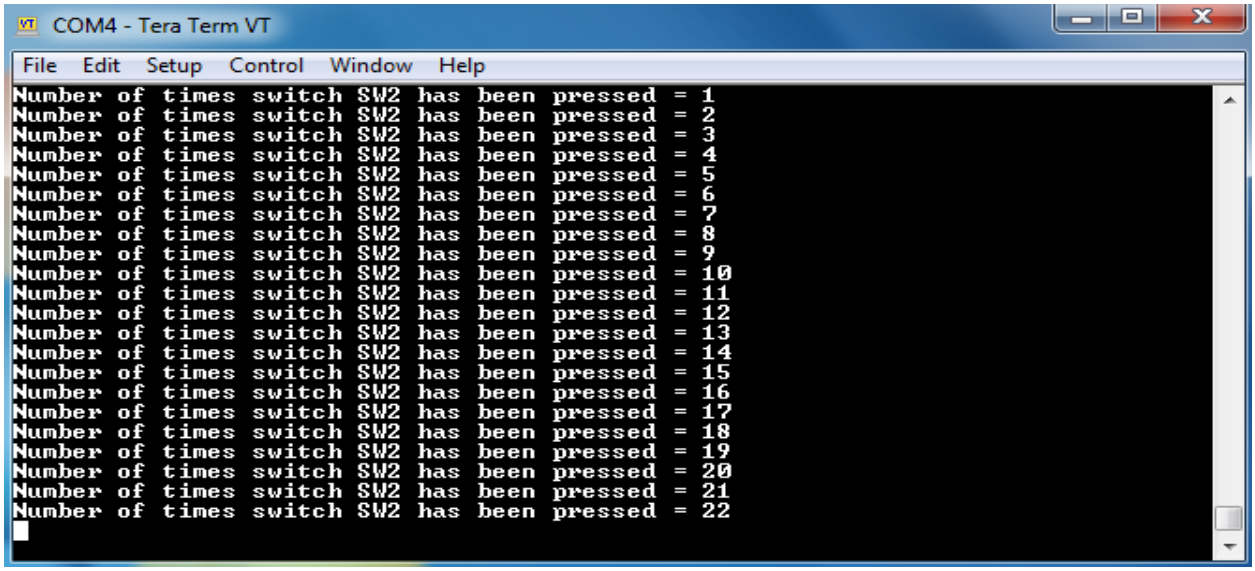
Software Setup

Set up a terminal emulator like Tera Term or PuTTY on your personal computer with baud rate and other settings shown in Figure 2.

Operation

1. Plug the CY8CKIT-062-BLE kit board into your computer's USB port.
2. Build the UART_printf 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.
3. Open a serial port communication program such as Tera Term and select the corresponding COM port. Configure the terminal with baud rate of 115200, data bits of 8, stop bits of 1, and with parity and flow control set to none.
4. Press the SW2 switch on the kit. The terminal will print the number of times the switch SW2 is pressed as shown in Figure 1.

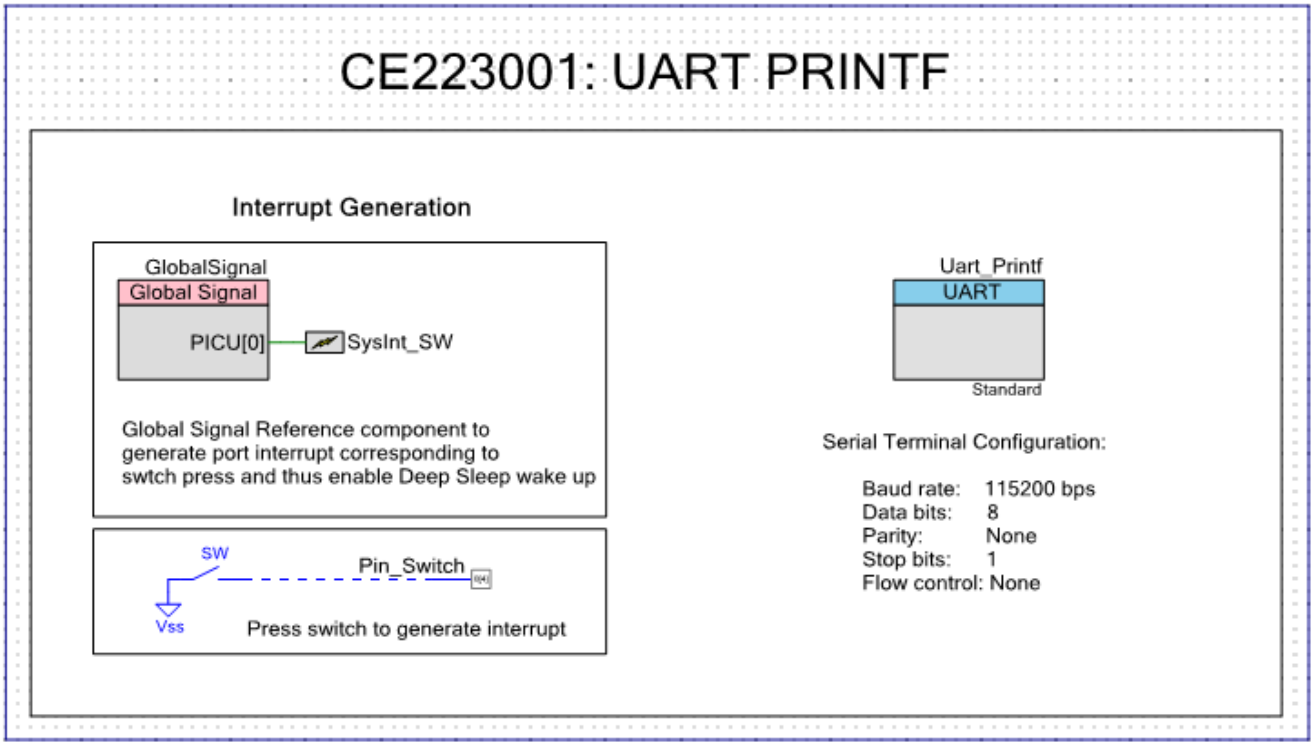
Figure 1. Terminal Prints



Design and Implementation

The design shown in Figure 2 has a UART (Uart_Printf) Component, Global Signal Reference (GlobalSignal) Component, System Interrupt Component (SysInt_Sw), and a GPIO (pin_switch) Component. UART is configured in TX mode to transmit data at baud rate of 115200. GlobalSignal is configured to connect a GPIO (pin_switch) interrupt signal to SysInt_Sw. SysInt_Sw increments the count when pin_switch (SW2 on kit) is pressed. Number of times the kit button SW2 is pressed is printed on the terminal using `printf` function. Count is reset when its value reaches 65535.

Figure 2. UART `printf` Example Schematic



Components and Settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the hardware resources used by each Component. Non-default settings for each Component are listed in Table 1.

Table 1. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Parameter Settings
UART(SCB_UART_PDL)	Uart_Printf	To handle UART serial communication	See Figure 3.
Global Signal Reference (GSRef)	GlobalSignal	To generate Interrupt	Tab Basic- Global Signal Name: Port Interrupt 0 (PICU[0]).
System Interrupt (SysInt)	SysInt_SW	To process Interrupt	Tab Basic- Check Box: Deep Sleep Capable.
General Purpose Input / Output (GPIO)	Pin_Switch	To Connect to switch	See Figure 4.

Figure 3. UART Parameter Settings

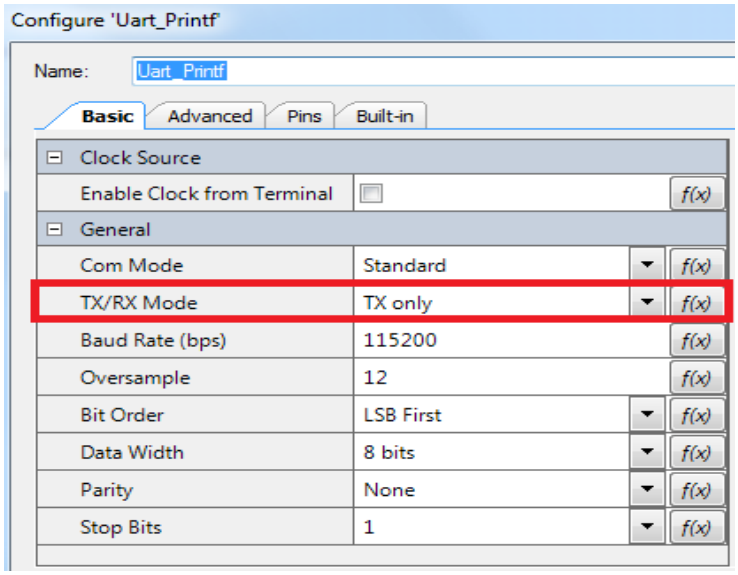
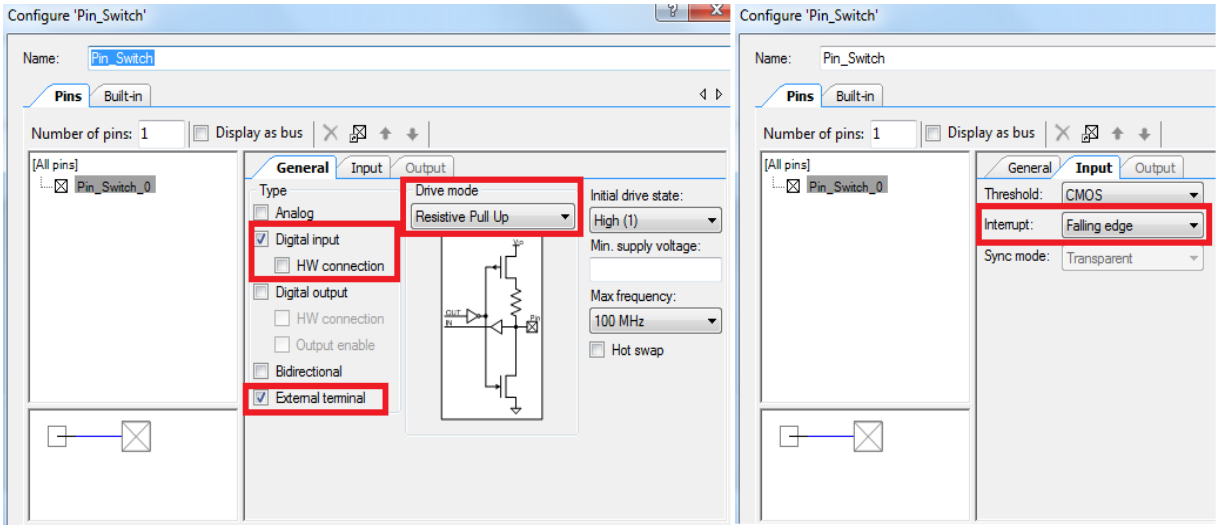


Figure 4.Pin_Switch GPIO Parameter Settings



Design-Wide Resources

Table 2 shows the pin assignment for the code example.

Table 2. Pin Names and Location

Pin Name	Location
Uart_Printf:tx	P5[1]
Pin_Switch	P0[4]

Reusing This Example

This example is designed for the CY8CKIT-062-BLE pioneer kit. To port the design to a different PSoC 6 MCU device, kit, or both, 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) may not be supported on another device. In those cases, 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 supported devices.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 63 with Bluetooth Low Energy (BLE) Connectivity and how to build your first PSoC Creator project
PSoC Creator Component Datasheets	
Global Signal Reference	Connections to device global signals
System Interrupt	Interrupt vectoring and control
General Purpose Input / Output	Supports Analog, Digital I/O and Bidirectional signal types
UART	Supports UART communication
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE223001 – PSoC 6 MCU UART printf

Document Number: 002-23001

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6104587	VJYA	03/23/2018	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

Forums | WICED IOT Forums | 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. 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.