

Objective

This example shows how to use the Serial Communication Block (SCB) Component as UART in PSoC® 4.

Overview

This example shows how the [Serial Communication Block](#) (SCB) Component is used as Universal Asynchronous Receiver Transmitter (UART) to transmit and receive data. It also demonstrates the basic Application Programming Interface (API) of the SCB component in the UART mode.

Requirements

Tool: [PSoC Creator™](#) 4.0 or higher

Programming Language: C (ARM® GCC 4.9.3)

Associated Parts: [PSoC 4](#) family

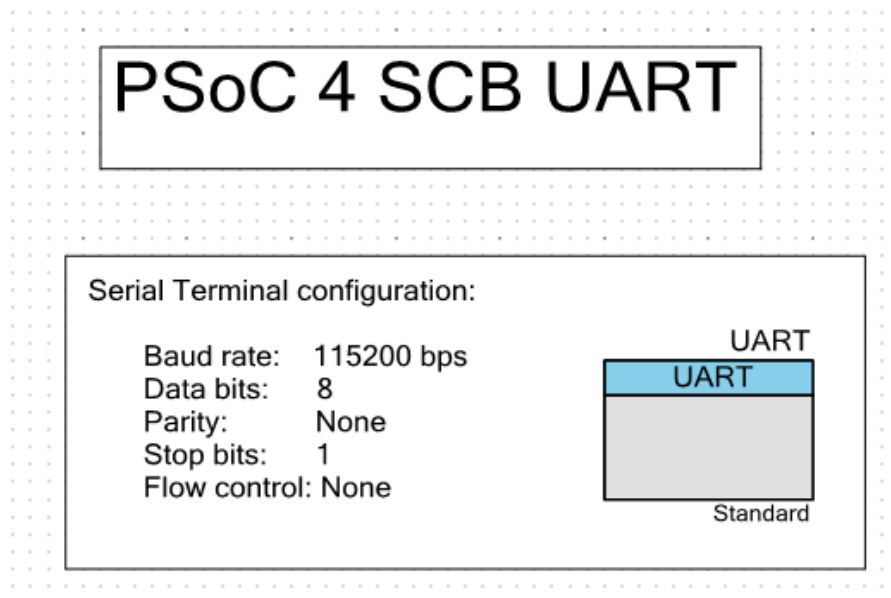
Hardware: [CY8CKIT-041](#), [CY8CKIT-042](#), [CY8CKIT-042-BLE](#), [CY8CKIT-044](#), [CY8CKIT-046](#), [CY8CKIT-048](#)

Design

In this example, the SCB Component is configured as a UART interface to interact with the user. It is used to transmit a welcome message through a terminal emulator. The main program constantly checks if a new character is received (sent from keyboard strokes). If so, the program will echo back the received characters. If there are no keyboard strokes, the program waits for a user input.

The top-level design of the PSoC Creator project is shown in [Figure 1](#).

Figure 1. Top Design Schematic



Kit Configuration and Pin Assignments

1. Select the appropriate device in the project's Design-Wide Resource files (.cydwr file) according to [Table 1](#).

Table 1. 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	CY8C4445AZI-483

2. Ensure that the IMO frequency is set to 24 MHz after the device is changed (go to the clocks section of .cydwr file and double-click any clock to verify that the IMO frequency is correct.)
3. The project is designed for the CY8CKIT-042, and therefore, the pin assignments are made accordingly. Edit the project's .cydwr file to modify the physical pins for UART according to [Table 2](#).

Table 2. 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]

4. **CY8CKIT-042 ONLY:** Connect P0[4] (\\UART:rx\\) to P12[7] on header J8, P0[5] (\\UART:tx\\) to the Pin P12[6] on header J8.

Components

[Table 3](#) lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 3. PSoC Creator Components

Component	Version	Hardware Resources
SCB	3.2	SCB

The SCB Component is configured for UART mode and most settings are left at their default values as shown in [Figure 2](#), [Figure 3](#), and [Figure 4](#).

Figure 2. SCB Configuration

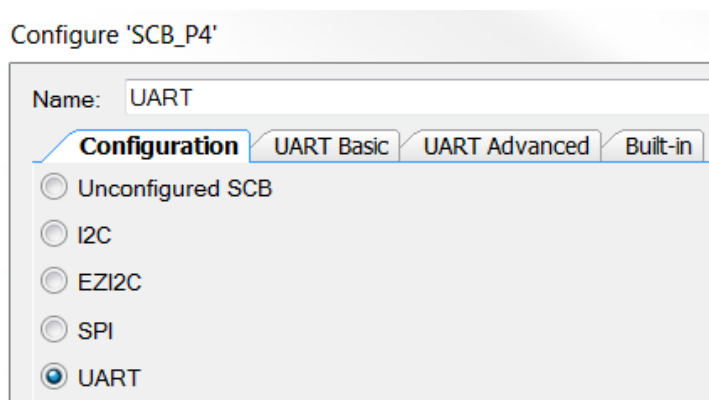


Figure 3. UART Basic Tab Configuration

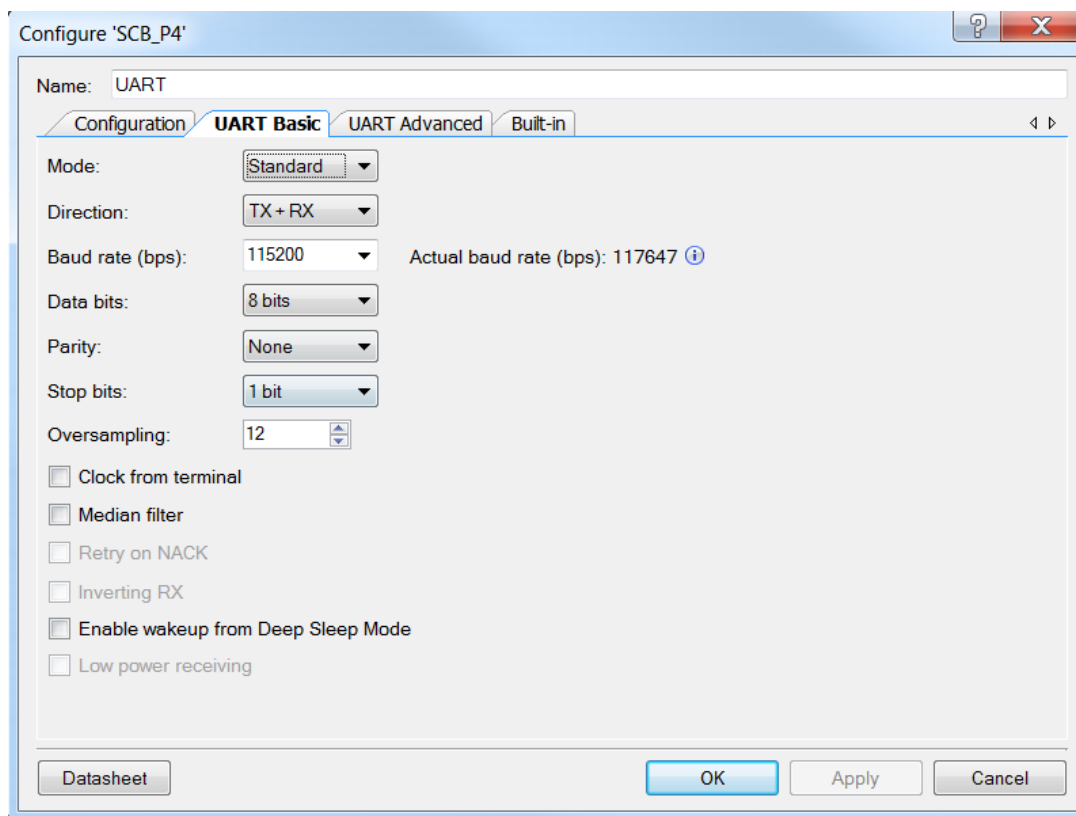
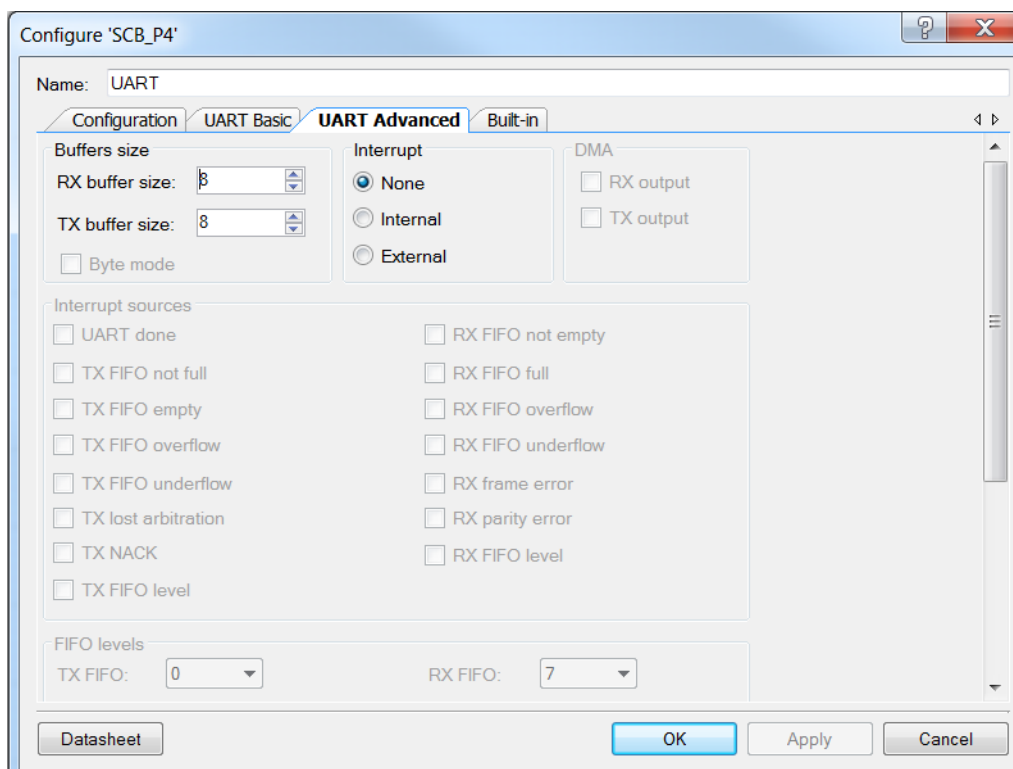


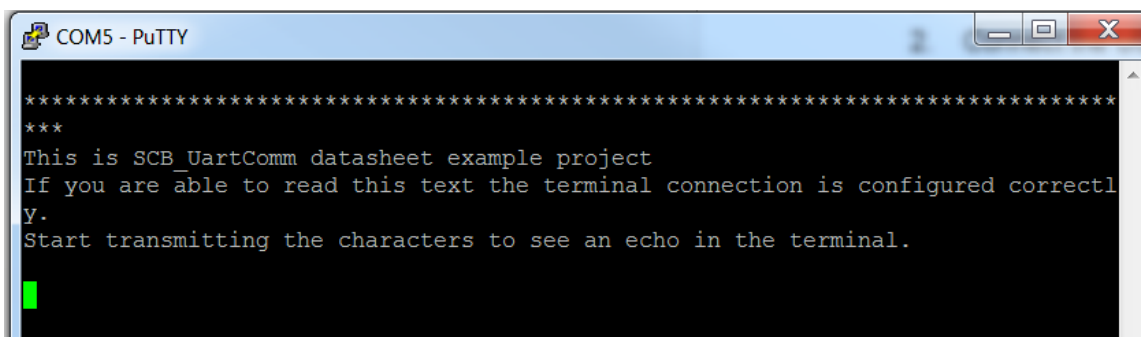
Figure 4. UART Advanced Tab Configuration



Operation

Follow these steps to communicate with the PC host.

1. Ensure that the kit has been configured as instructed in the [Kit Configuration and Pin Assignment](#) section.
2. Connect the USB cable between the PC and the PSoC 4 Pioneer Kit.
3. Program the example project from PSoC Creator in the Pioneer Kit.
4. Open a terminal emulator and configure the program to the appropriate COM port as listed in your computer. Also configure the baud rate to 115200, data bits to 8, no parity bits, stop bit as 1 and no control flow.
5. Press the reset button on the kit and observe the following welcome message printed on the terminal program:



6. Start typing and observe that the terminal program is echoing the character typed.

Related Documents

Table 4 lists the relevant application notes, code examples, Component datasheets, and device and DVK documentation.

Table 4. Related Documents

Application Notes		
AN79953	Getting Started with PSoC 4	Describes PSoC 4 and shows how to build the attached code example
Code Examples		
CE95363	I2C Master using a Serial Communication Block (SCB) with PSoC 4	This code example demonstrates the basic operation of the I2C Master (SCB mode) Component. I ² C master sends packet with command to I ² C slave to control RGB LED color. Packet with status is read back.
CE95364	I2C Slave using a Serial Communication Block (SCB) with PSoC 4	This code example demonstrates the basic operation of the I2C Slave (SCB mode) Component. I ² C slave accepts packet with command from I ² C master to control RGB LED color. I ² C slave updates its buffer with status packet in response to accepted command.
CE95365	SPI Transmit and Receive using a Serial Communication Block (SCB) with PSoC 4	This datasheet code example demonstrates operation of the SCB Component configured in SPI. The first instance of SCB is configured as SPI master and the second as SPI slave mode. SPI master communicates with slave (bit rate 1 Mbps).
CE95366	UART Transmit and Receive using a Serial Communication Block (SCB) with PSoC 4	This datasheet code example demonstrates operation of the SCB Component configured in UART Tx+Rx mode. Data typed on the HyperTerminal is sent through serial port to DVK and displayed on LCD. Afterward, this data is sent back and displayed on HyperTerminal.
PSoC Creator Component Datasheets		
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		
PSoC 4 Kits		

Document History

Document Title: CE95366 – PSoC® 4 SCB UART

Document Number: 001-95366

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5544081	WESL	12/09/2016	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

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

Technical Support

cypress.com/support



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

© Cypress Semiconductor Corporation, 2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spanion 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, Spanion, the Spanion 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.