**Objective**

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

**Overview**

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

**Requirements**

**Tool:** PSoC Creator™ 4.2  
**Programming Language:** C (Arm® GCC 5.4.1)  
**Associated Parts:** PSoC 4 family  
**Hardware:** CY8CKIT-041, CY8CKIT-042, CY8CKIT-042-BLE, CY8CKIT-044, CY8CKIT-046, CY8CKIT-048

**Hardware Setup**

This code example is set up for CY8CKIT-042. If you are using a different kit, see Reusing This Example.

For the CY8CKIT-042:
3. Other kits use different pins for the UART. Make sure that you select the pins that are right for your kit.

**Software Setup**

This design requires a serial terminal emulator such as PuTTY or Tera Term. The instructions in this document use Tera Term as the terminal emulator.

**Operation**

Follow these steps to communicate with the PC host:
1. Ensure that the right 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. 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.  
   For Windows 7, select Windows > Devices and Printers > Cypress KitProg > Properties > Hardware  
   For Windows 10, right-click Windows > Device Manager > Ports > Cypress KitProg
5. Configure the baud rate to 115200, data bits to 8, no parity bits, stop bit as 1, and no control flow.
6. Press the reset button on the kit and observe the following welcome message printed on the terminal program:
7. Start typing and observe that the terminal program is echoing the character typed.

**Design and Implementation**

In this example, the SCB Component is configured as a UART. The UART first transmits a welcome message through a terminal emulator. Then, the main program constantly checks if a new character is received (sent from keyboard inputs). If so, the program will echo back the received characters. If there are no keyboard inputs, the program waits for a new character.

The top-level design of the PSoC Creator project is shown in Figure 1.

![Figure 1. Top Design Schematic](image)

**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>
<thead>
<tr>
<th>Component</th>
<th>Instance Name</th>
<th>Purpose</th>
<th>Non-default Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>UART (SCB Mode)</td>
<td>UART</td>
<td>Enable device communication with the user</td>
<td>Default settings only</td>
</tr>
</tbody>
</table>

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

**Reusing This Example**

To port this design to a different PSoC 4 MCU device and/or kit, do the following:

1. In PSoC Creator, select Project > Device Selector to change the target device. Select your device as listed in Table 2.

<table>
<thead>
<tr>
<th>Development Kit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY8CKIT-041</td>
<td>CY8C4146AZI-S433</td>
</tr>
<tr>
<td>CY8CKIT-042</td>
<td>CY8C4245AXI-483</td>
</tr>
<tr>
<td>CY8CKIT-042-BLE</td>
<td>CY8C4247LQI-BL483</td>
</tr>
<tr>
<td>CY8CKIT-044</td>
<td>CY8C4247AZI-M485</td>
</tr>
<tr>
<td>CY8CKIT-046</td>
<td>CY8C4248BZI-L489</td>
</tr>
<tr>
<td>CY8CKIT-048</td>
<td>CY8C4445AZI-483</td>
</tr>
</tbody>
</table>

2. Ensure that the IMO frequency is set to 24 MHz after the device is changed.

3. In PSoC Creator Workspace Explorer, select the Clocks interface listed under Design Wide Resources.

4. Set IMO Desired Frequency to 24 MHz if it is not already.
The project is designed for CY8CKIT-042, and therefore, the pin assignments are made accordingly. For other kits, go to the PSoC Creator Workspace Explorer and select the Pins interface. Configure the pins as necessary for your kit following Table 3.

**Note:** Connect the UART:rx pin to P12[7] on header J8.

Table 3. Pin Assignments for Different Kits

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>CY8CKIT-041</th>
<th>CY8CKIT-042</th>
<th>CY8CKIT-042-BLE</th>
<th>CY8CKIT-044</th>
<th>CY8CKIT-046</th>
<th>CY8CKIT-048</th>
</tr>
</thead>
</table>

Related Documents

<table>
<thead>
<tr>
<th>Application Notes</th>
<th>Code Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN79953</td>
<td>Getting Started with PSoC 4</td>
</tr>
<tr>
<td></td>
<td>Describes PSoC 4 and shows how to build the attached code example</td>
</tr>
</tbody>
</table>

Code Examples

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE95363</td>
<td>I2C Master using a Serial Communication Block (SCB) with PSoC 4</td>
</tr>
<tr>
<td></td>
<td>This code example demonstrates the basic operation of the I2C Master (SCB mode) Component. I2C master sends packet with command to I2C slave to control RGB LED color. Packet with status is read back.</td>
</tr>
<tr>
<td>CE95364</td>
<td>I2C Slave using a Serial Communication Block (SCB) with PSoC 4</td>
</tr>
<tr>
<td></td>
<td>This code example demonstrates the basic operation of the I2C Slave (SCB mode) Component. I2C slave accepts packet with command from I2C master to control RGB LED color. I2C slave updates its buffer with status packet in response to accepted command.</td>
</tr>
<tr>
<td>CE95365</td>
<td>SPI Transmit and Receive using a Serial Communication Block (SCB) with PSoC 4</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
</tbody>
</table>

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:** CE195366 – PSoC 4 SCB UART  
**Document Number:** 001-95366

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Orig. of Change</th>
<th>Submission Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>5544081</td>
<td>WESL</td>
<td>12/09/2016</td>
<td>New code example</td>
</tr>
</tbody>
</table>
| *A       | 6144429| BFMC           | 02/23/2018      | Updated to new template  
|          |        |                |                 | Added Hardware Setup and Software Setup Sections  
|          |        |                |                 | Updated terminal emulator image  
|          |        |                |                 | Removed UART Configuration Images  
|          |        |                |                 | Added Instructions for locating COM port  
|          |        |                |                 | Changed heading “Kit Configuration and Pin Assignments” to “Reusing This 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 | PSoC 6 MCU

Cypress Developer Community
Community | Projects | Videos | Blogs | Training | Components

Technical Support
cypress.com/support

© Cypress Semiconductor Corporation, 2016-2018. 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, CYSPRESS 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.