

**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.

## Objective

This code example demonstrates how to interface Cypress SPI F-RAM devices with PSoC® 4 using dedicated serial NVRAM Component in PSoC Creator™.

## Overview

This code example provides details on Serial NVRAM Component configuration to access SPI F-RAM using PSoC 4. This document also discusses the hardware and software setup and its configuration to execute the associated example project and validate results.

## Requirements

**Tool:** [PSoC Creator 4.1](#) or above

**Programming Language:** C (GCC 5.4.3), Arm® Cortex®-M3 assembler

### Associated Parts:

- Cypress Serial NVRAMs: All SPI F-RAM™, SPI F-RAM processor companions, SPI nvSRAM, and SPI nvSRAM RTC parts. This code example is validated with 4-Mbit SPI F-RAM, CY15B104Q.
- Cypress PSoC: PSoC 4 family parts excluding PSoC 4000 devices which do not support SCB-SPI interface

### Related Hardware:

- Serial NVRAM Kits: [CY15FRAMKIT-001](#) is an Arduino-compatible F-RAM development board on PSoC 4 development kits.
- See [Table 1](#) for details on supported PSoC 4 kits.
- This code example is validated with CY8CKIT-042 PSoC 4 Pioneer Kit.

## Design

The key design features include configuring the following:

- UART to display result on PC
- SPI (SCB) as SPI master
- Serial NVRAM Component to enable SPI NVRAM access
- Control Register to drive Status LEDs

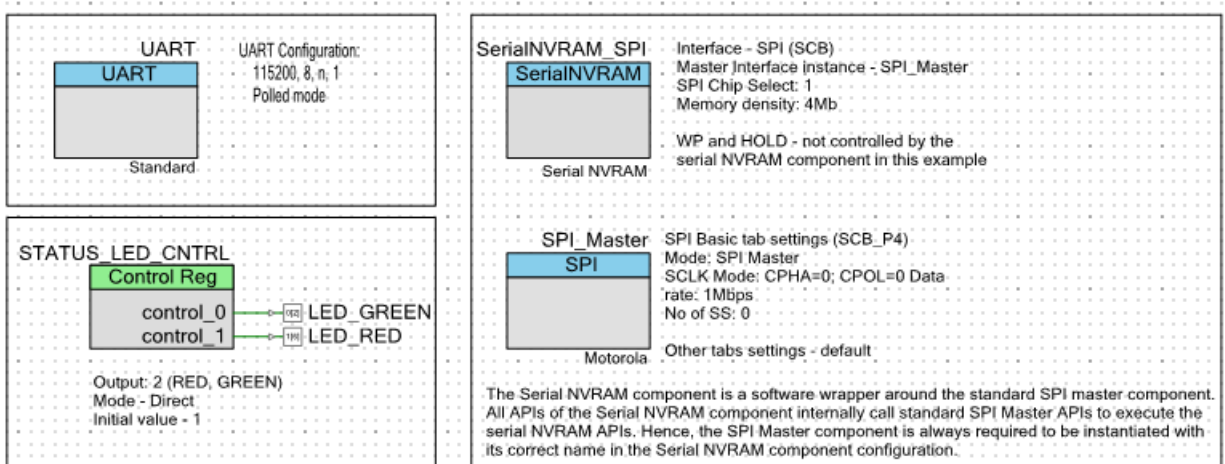
The code example demonstrates the use of the following features:

1. Write to serial F-RAM status register
2. Read from serial F-RAM status register
3. Write 1 byte to an address of serial F-RAM
4. Read 1 byte from an address of serial F-RAM
5. Write 16 bytes into serial F-RAM; burst write from a start address
6. Read 16 bytes from serial F-RAM; burst read from a start address

## Design Considerations

The SPI NVRAM Component uses PSoC 4 serial communication block (SCB) resource to enable the SPI host; therefore, the SPI host speed and data throughput are primarily determined by the PSoC 4 device. Cypress SPI NVRAMs can support SPI access up to 40 MHz. However, the full speed cannot be demonstrated on any PSoC 4 platform due to SPI speed limitations. See the [PSoC 4 datasheet](#) for more details on PSoC 4 SPI master capability and setup requirement to achieve the best performance. [Figure 1](#) shows PSoC Creator design schematic of the code example.

Figure 1. SPI NVRAM Access – PSoC Creator Design Schematic

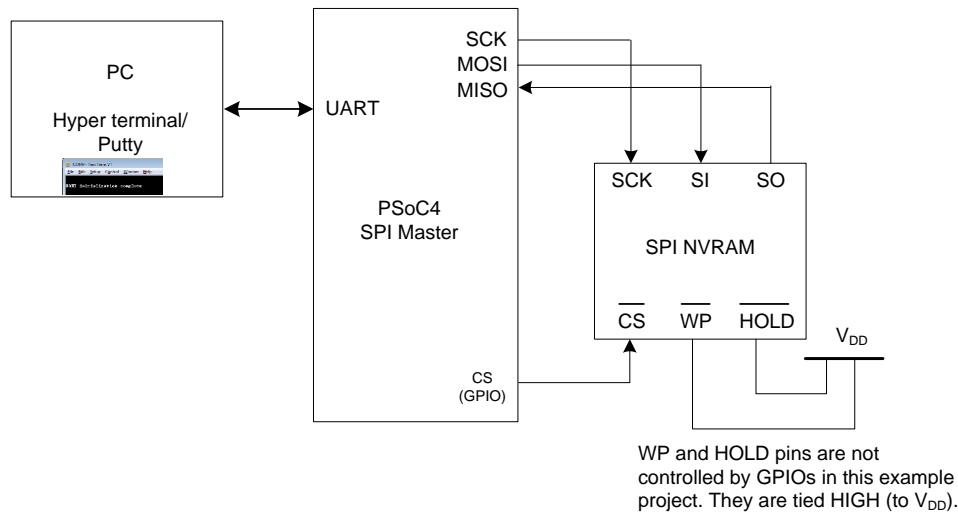


## Hardware Setup

The hardware setup involves connecting the F-RAM or nvSRAM SPI interface pins with the PSoC 4 SPI master pins. You can use either dedicated hardware as described in the [Requirements](#) section or connect through jumper wires.

This example project uses the default configuration of Cypress' PSoC 4 Pioneer Kit (CY8CKIT-042) and an internal daughter board with 4-Mbit SPI F-RAM (CY15B104Q), connected to the PSoC 4 Pioneer Kit using jumper wires, as shown in [Figure 2](#). See [CY8CKIT-042 PSoC 4 Pioneer Kit](#) for hardware connection details. If you use any readily available hardware, as described in the [Requirements](#) section, to run this code example, you may need to tweak the code to define the correct density and comment any non-supported opcodes, per SPI NVRAM device on the board.

Figure 2. Hardware Setup Block Diagram

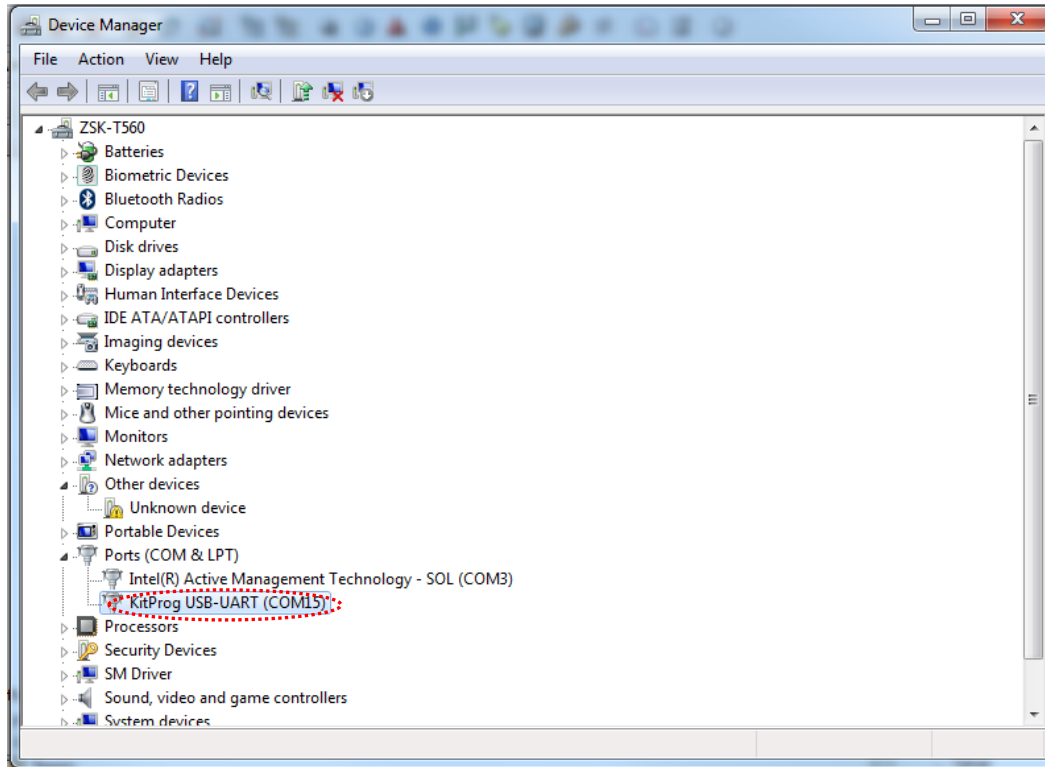


## Software Setup

This section demonstrates the procedure to set up the serial (UART) connection using HyperTerminal or PuTTY on a PC to communicate with the PSoC 4 Pioneer Kit. This code example is created using PSoC Creator software as described in the [Requirements](#) section. PuTTY is a free SSH and Telnet client for Windows. You can download PuTTY from [www.putty.org](http://www.putty.org). Follow these instructions to determine the COM port number and set up PuTTY to monitor the code example outputs on a PC.

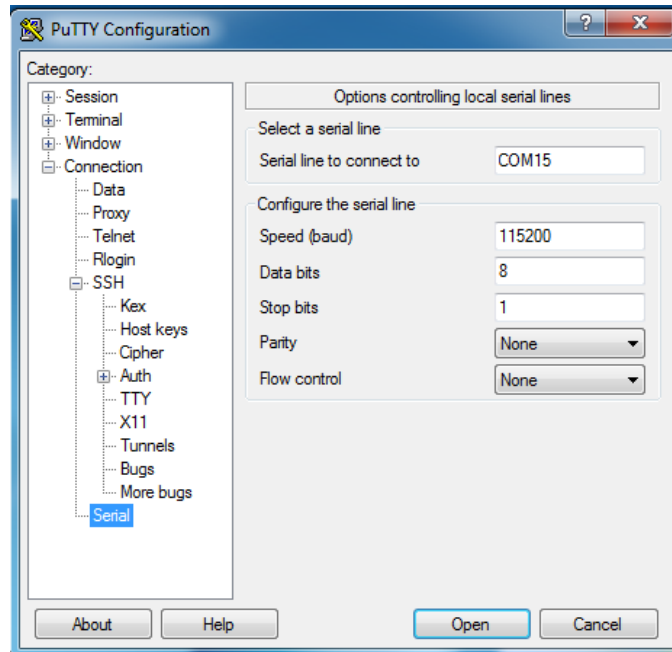
1. Connect PSoC 4 Pioneer Kit to the PC using USB Mini-B cable. The kit enumerates as KitProg USB-UART and is available under **Device Manager > Ports (COM & LPT)**. A communication port (COMx) is assigned to KitProg USB-UART, for example, COM15 is assigned to PSoC 4 Pioneer Kit on this sample setup as shown in [Figure 3](#).

Figure 3. KitProg USB-UART in Device Manager



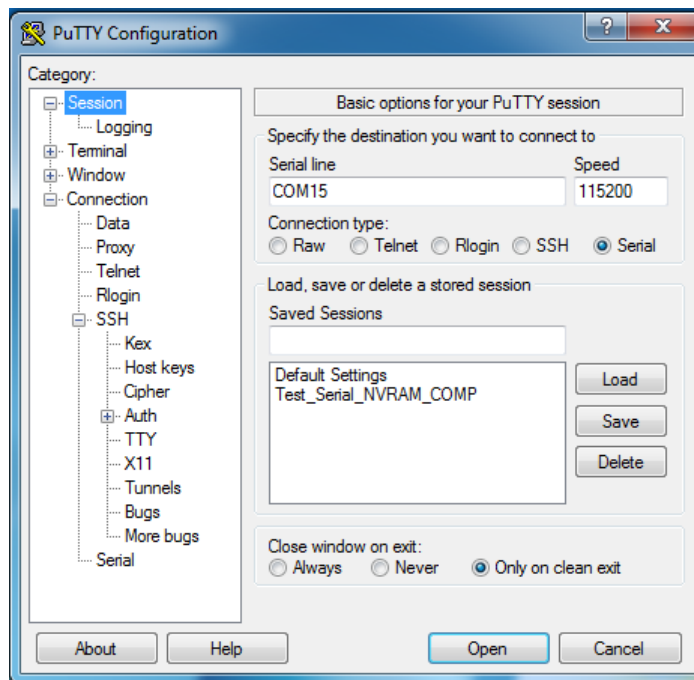
2. After you download and install PuTTY, double-click the PuTTY icon. In the PuTTY Configuration window, select **Connection > Serial** under **Category**.
3. A new window opens where the communication port can be selected. In the **Options controlling local serial lines** section:
  - Enter the PSoC 4 Port (COM & LPT), COMx, in **Serial line to connect to**. This code example uses **COM15**. Verify the COM setting for your setup and select the appropriate COMx.
  - Enter **Speed (baud)**, **Data bits**, and **Stop bits**.
  - Select the **Parity** and **Flow control**.

Figure 4. Open New Connection



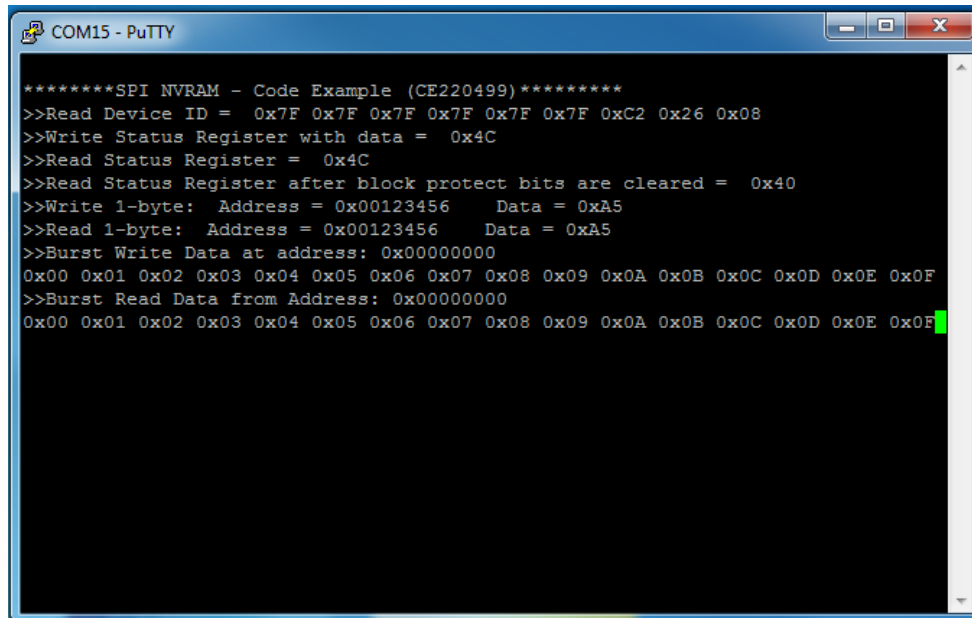
4. Select **Session** under **Category**. Select **Serial** as the **Connection type** as shown in Figure 5. You can save this current session and **load** the settings when required. Enter a name in Saved Sessions and click **Save**. Click **Open** to proceed.

Figure 5. Select Communication Type in PuTTY



5. The COM terminal window displays the code example results as shown in Figure 6. You may have to reprogram PSoC 4 with the code example hex file or reset PSoC 4 (already programmed) to restart the code execution and monitor the result.

Figure 6. Result Displayed on PuTTY



```

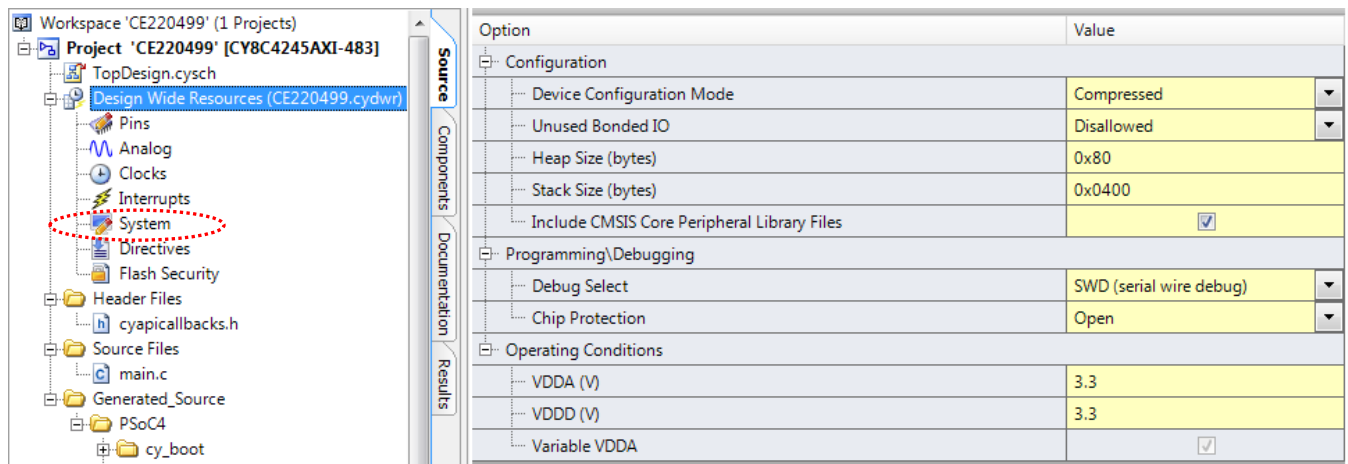
*****SPI NVRAM - Code Example (CE220499)*****
>>Read Device ID = 0x7F 0x7F 0x7F 0x7F 0x7F 0x7F 0xC2 0x26 0x08
>>Write Status Register with data = 0x4C
>>Read Status Register = 0x4C
>>Read Status Register after block protect bits are cleared = 0x40
>>Write 1-byte: Address = 0x00123456 Data = 0xA5
>>Read 1-byte: Address = 0x00123456 Data = 0xA5
>>Burst Write Data at address: 0x00000000
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
>>Burst Read Data from Address: 0x00000000
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
  
```

Alternatively, you can run the HyperTerminal, if supported on your PC, to monitor the above result.

## Design-Wide Resources

Make sure that  $V_{DD}$  (**PSoC Creator > Design Wide Resources > System** tab) is set to 2.7 V or above, as shown in [Figure 7](#), to drive the STATUS\_LED. Also, make sure that PSoC 4 I/O voltage is set correctly to match the SPI NVRAM operating range ( $V_{DD}/V_{CC}$ ).









 Figure 7.  $V_{DD}$  Setting using Design Wide Resources



| Option                                      | Value                               |
|---|-------------------------------------|
| <b>Configuration</b>                        |                                     |
| Device Configuration Mode                   | Compressed                          |
| Unused Bonded IO                            | Disallowed                          |
| Heap Size (bytes)                           | 0x80                                |
| Stack Size (bytes)                          | 0x0400                              |
| Include CMSIS Core Peripheral Library Files | <input checked="" type="checkbox"/> |
| <b>Programming/Debugging</b>                |                                     |
| Debug Select                                | SWD (serial wire debug)             |
| Chip Protection                             | Open                                |
| <b>Operating Conditions</b>                 |                                     |
| VDDA (V)                                    | 3.3                                 |
| VDDD (V)                                    | 3.3                                 |
| Variable VDDA                               | <input checked="" type="checkbox"/> |

[Figure 8](#) shows the pin assignments for the code example.

Figure 8. PSoC 4 Pin Assignments for Code Example

|   | Name                   | Port  | Pin | Lock                                |
|---|------------------------|-------|-----|-------------------------------------|
|  | \SerialNVRAM_SPI:CS_0\ | P3[4] | 15  | <input checked="" type="checkbox"/> |
|  | \SPI_Master:miso_m\    | P3[1] | 12  | <input checked="" type="checkbox"/> |
|  | \SPI_Master:mosi_m\    | P3[0] | 11  | <input checked="" type="checkbox"/> |
|  | \SPI_Master:sclk_m\    | P0[6] | 30  | <input checked="" type="checkbox"/> |
|  | \UART:rx\              | P4[0] | 20  | <input checked="" type="checkbox"/> |
|  | \UART:tx\              | P4[1] | 21  | <input checked="" type="checkbox"/> |
|  | LED_GREEN              | P0[2] | 26  | <input checked="" type="checkbox"/> |
|  | LED_RED                | P1[6] | 43  | <input checked="" type="checkbox"/> |

To switch the code example from CY8CKIT-042 to any other PSoC 4 kit, change the project's device with the help of Device Selector called from the project's context menu as shown in [Table 1](#).

Table 1. Supported PSoC 4 Development Kits Versus PSoC 4 Parts

| Development Kit | Device            |
|-----------------|-------------------|
| CY8CKIT-042     | CY8C4245AXI-483   |
| CY8CKIT-042-BLE | CY8C4248LQI-BL583 |
| CY8CKIT-044     | CY8C4247AZI-M485  |

The port (pin) assignments for the supported kits are shown in [Table 2](#).

Table 2. Pin Assignment on Supported Development Kits

| Code Example - Pin Name | PSoC 4 Development Kit - Port Name |                 |             |
|-------------------------|------------------------------------|-----------------|-------------|
|                         | CY8CKIT-042                        | CY8CKIT-042-BLE | CY8CKIT-044 |
| \SerialNVRAM_SPI:CS_0\  | P3[4]                              | P0[2]           | P2[7]       |
| \SPI_Master:miso_m\     | P3[1]                              | P0[1]           | P6[1]       |
| \SPI_Master:mosi_m\     | P3[0]                              | P0[0]           | P6[0]       |
| \SPI_Master:sclk_m\     | P0[6]                              | P0[3]           | P6[2]       |
| \UART:rx\               | P4[0]                              | P3[0]           | P7[0]       |
| \UART:tx\               | P4[1]                              | P3[1]           | P7[1]       |
| LED_GREEN               | P0[2]                              | P3[6]           | P2[6]       |
| LED_RED                 | P1[6]                              | P2[6]           | P0[6]       |

## Components

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

Table 3. PSoC Creator Components

| Component   | Instance Name    | Hardware Resources  |
|-------------|------------------|---|
| SerialNVRAM | SerialNVRAM_SPI  | Serial NVRAM Block in <b>System &gt; External Memory Interface</b> Component category |
| SPI         | SPI_Master       | SPI (SCB Mode) in <b>Communication &gt; SPI</b> Component category                    |
| UART        | UART             | UART (SCB Mode) in <b>Communication</b> Component category                            |
| Control Reg | STATUS_LED_CNTRL | Control Register in <b>Digital &gt; Registers</b> Component category                  |

Figure 9 to Figure 12 show the Component configuration for the code example project.

Figure 9. Serial NVRAM Component Configuration and Parameter Settings

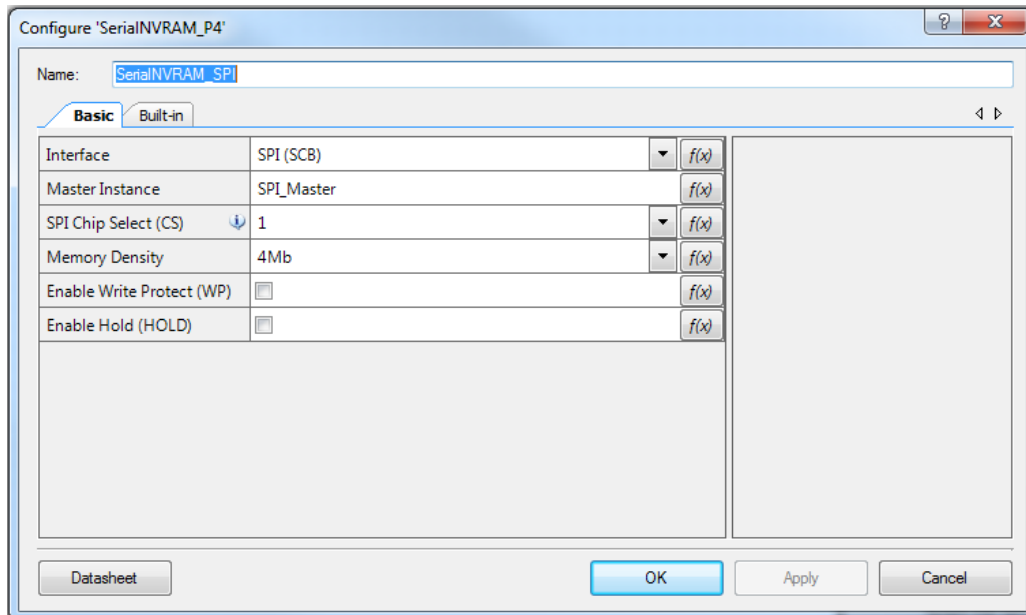


Figure 10. SPI (SCB mode) Configuration and Parameter Settings

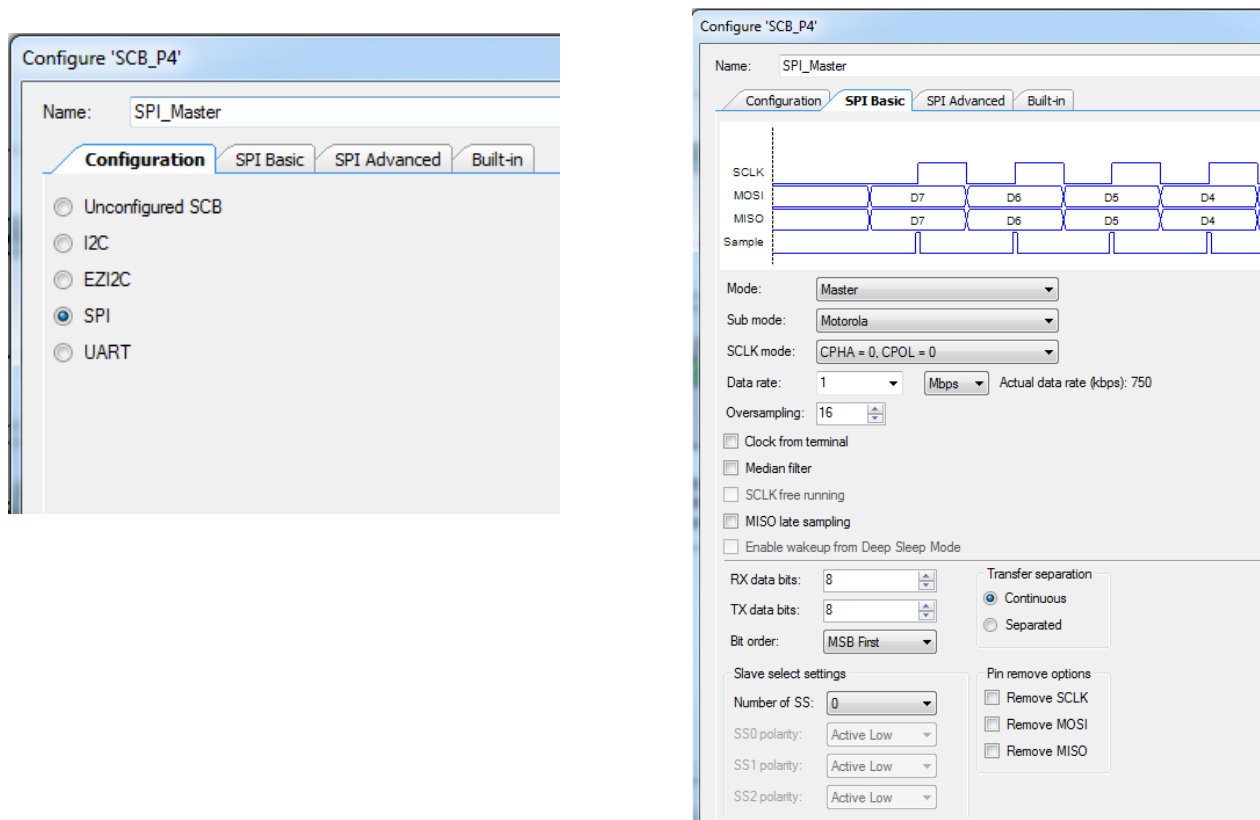
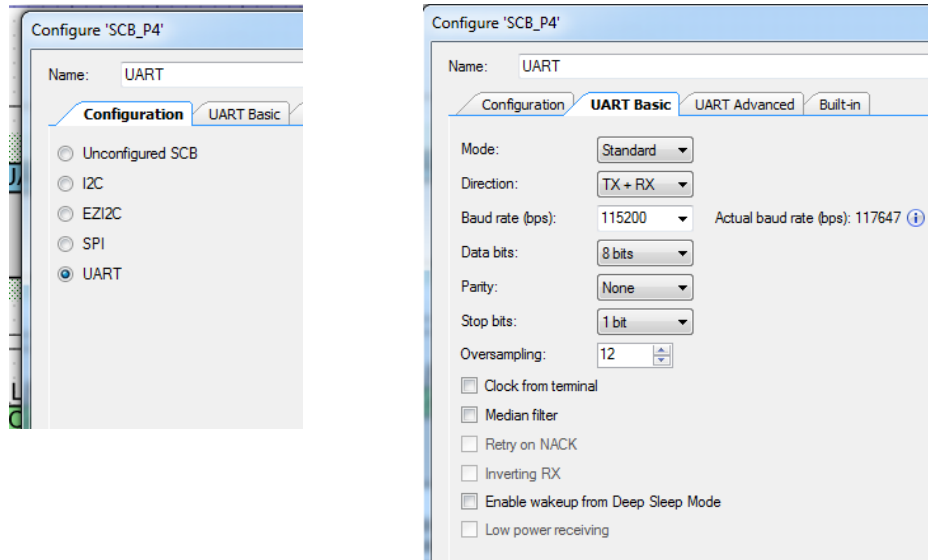




Figure 11. UART (SCB mode) Configuration and Parameter Settings



**Configure 'SCB\_P4'**

Name:

**Configuration** | UART Basic

☐ Unconfigured SCB  
☐ I2C  
☐ EZI2C  
☐ SPI  
☒ UART

---

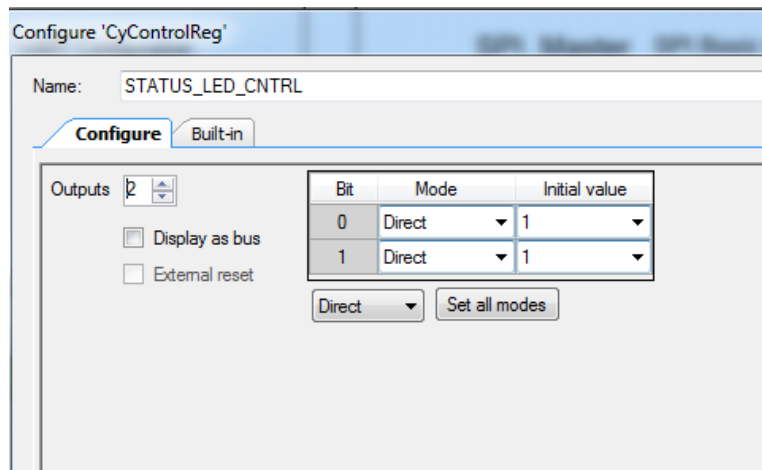
**Configure 'SCB\_P4'**

Name:

**Configuration** | **UART Basic** | UART Advanced | Built-in

Mode:   
 Direction:   
 Baud rate (bps):  Actual baud rate (bps): 117647 ⓘ  
 Data bits:   
 Parity:   
 Stop bits:   
 Oversampling:   
☐ Clock from terminal  
☐ Median filter  
☐ Retry on NACK  
☐ Inverting RX  
☒ Enable wakeup from Deep Sleep Mode  
☐ Low power receiving

Figure 12. Control Register Configuration and Parameter Settings



**Configure 'CyControlReg'**

Name:

**Configure** | Built-in

Outputs:

☐ Display as bus  
☐ External reset

| Bit | Mode   | Initial value |
|-----|--------|---------------|
| 0   | Direct | 1             |
| 1   | Direct | 1             |

## Related Documents

| Application Notes  |   |
|--|---|
| <a href="#">AN304 – SPI Guide for F-RAM™</a>   | Describes SPI F-RAM design in a system  |
| <a href="#">AN64574 – Designing with Serial Peripheral Interface (SPI) nvSRAM</a>                                  | Describes SPI nvSRAM design in a system   |
| Component Datasheets<br>(Right click on the component in PSoC Creator schematic to access the component datasheet) |   |
| Serial NVRAM   | Serial NVRAM component configuration for different serial interfaces (SPI and I <sup>2</sup> C) and API details                           |
| SPI  | SPI interface SCB mode  |
| UART   | UART configuration SCB mode   |
| Control Reg  | Allows firmware to set values to use for digital signals  |
| Device Documentation   |   |
| <a href="#">CY15B104Q</a>  | 4-Mbit, SPI F-RAM datasheet   |
| <a href="#">All Serial NVRAM Datasheets</a>  | Serial (SPI and I2C) NVRAM (F-RAM and nvSRAM)   |
| Development Kit Documentation  |   |
| <a href="#">CY8CKIT-042 PSoC 4 Pioneer Kit</a>   | Development kit for the PSoC 4200 device family   |
| <a href="#">CY15FRAMKIT-001 Serial F-RAM Development Kit</a>   | Development kit for Cypress's serial F-RAMs. This kit has a wide operating range (2.7V to 5.5V), 256-Kbit SPI F-RAM in 8-pin SOIC package |

## Document History

Document Title: CE220499 - Interfacing the SPI F-RAM with PSoC 4

Document Number: 002-20499

| Revision | ECN     | Orig. of Change | Submission Date | Description of Change |
|----------|---------|-----------------|-----------------|-----------------------|
| **       | 5955510 | ZSK             | 11/07/2017      | 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 | <a href="http://cypress.com/arm">cypress.com/arm</a>               |
| Automotive                    | <a href="http://cypress.com/automotive">cypress.com/automotive</a> |
| Clocks & Buffers              | <a href="http://cypress.com/clocks">cypress.com/clocks</a>         |
| Interface                     | <a href="http://cypress.com/interface">cypress.com/interface</a>   |
| Internet of Things            | <a href="http://cypress.com/iot">cypress.com/iot</a>               |
| Memory                        | <a href="http://cypress.com/memory">cypress.com/memory</a>         |
| Microcontrollers              | <a href="http://cypress.com/mcu">cypress.com/mcu</a>               |
| PSoC                          | <a href="http://cypress.com/psoc">cypress.com/psoc</a>             |
| Power Management ICs          | <a href="http://cypress.com/pmic">cypress.com/pmic</a>             |
| Touch Sensing                 | <a href="http://cypress.com/touch">cypress.com/touch</a>           |
| USB Controllers               | <a href="http://cypress.com/usb">cypress.com/usb</a>               |
| Wireless Connectivity         | <a href="http://cypress.com/wireless">cypress.com/wireless</a>     |

## PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

## Cypress Developer Community

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

## Technical Support

[cypress.com/support](http://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, 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](http://cypress.com). Other names and brands may be claimed as property of their respective owners.