# PSoC® Creator™
# Project Datasheet for CapSense_CSD_P4_Sample

**Creation Time: 02/20/2016 16:37:36**
**User: AMIT\Amit**
**Project: CapSense_CSD_P4_Sample**
**Tool: PSoC Creator 3.3 SP1**

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
http://www.cypress.com

### Copyright



### Trademarks

PSoC and CapSense are registered trademarks of Cypress Semiconductor Corporation. PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

### Philips I2C Patent Rights

Purchase of I2C components from Cypress or one of its sublicensed Associated Companies conveys a license under the Philips I2C Patent Rights to use these components in an I2C system, provided that the system conforms to the I2C Standard Specification as defined by Philips. As from October 1st, 2006 Philips Semiconductors has a new trade name, NXP Semiconductors.

### Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of a Cypress product in a life support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

### Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Datasheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as 'unbreakable.'

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

# Table of Contents

# 1 Overview

The Cypress PSoC 4 is a family of 32-bit devices with the following characteristics:

- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals such as PWM, UART, SPI and I2C
- Analog subsystem that includes 12-bit SAR ADC, comparators, op amps, CapSense, LCD drive and more
- Several types of memory elements, including SRAM and flash
- Programming and debug system through Serial Wire Debug (SWD)
- High-performance 32-bit ARM Cortex-M0 core with a nested vectored interrupt controller (NVIC)
- Flexible routing to all pins

Figure 1 shows the major components of a typical PSoC 4200 BLE family member PSoC 4 device. For details on all the systems listed above, please refer to the PSoC 4 Technical Reference Manual .

Figure 1. PSoC 4200 BLE Device Family Block Diagram

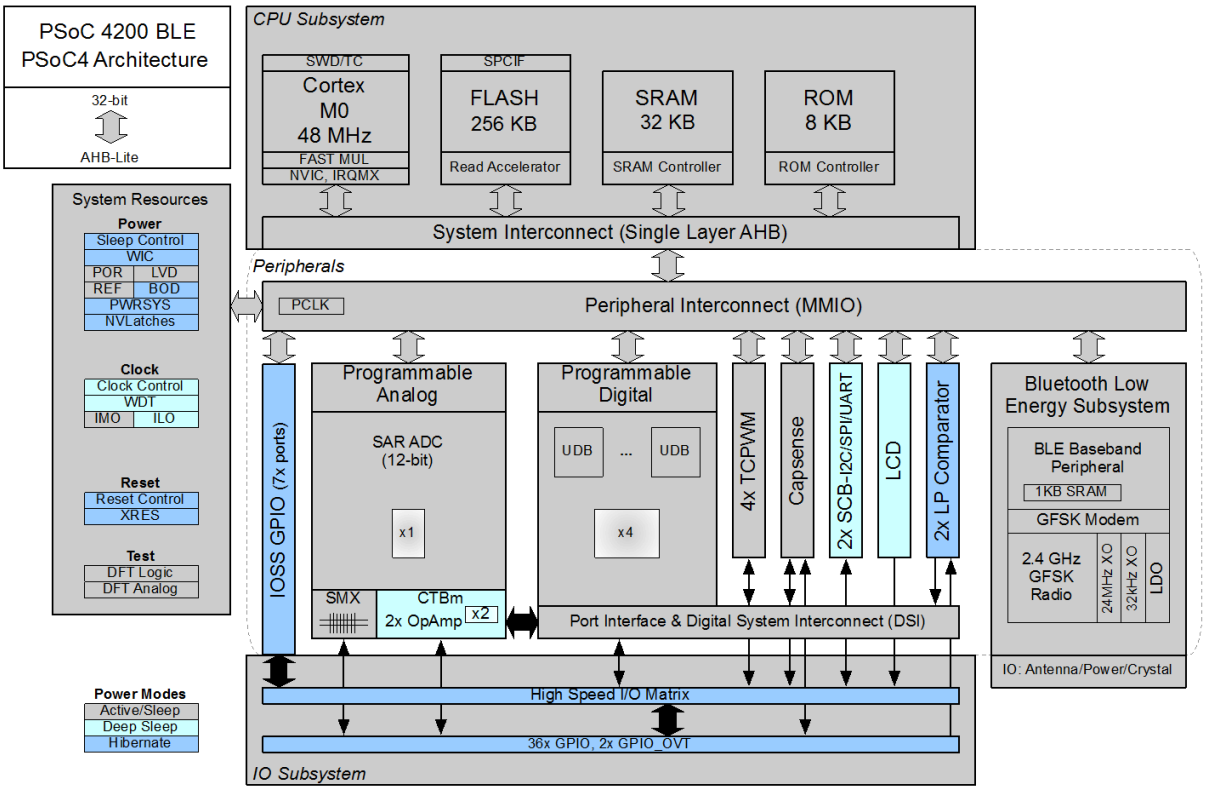


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

| Name | Value |
|---|---|
| Part Number | CY8C4247LQI-BL483 |
| Package Name | 56-QFN |
| Architecture | PSoC 4 |
| Family | PSoC 4200 BLE |
| CPU speed (MHz) | 48 |
| Flash size (kBytes) | 128 |
| SRAM size (kBytes) | 16 |
| Vdd range (V) | 1.9 to 5.5 |
| Automotive qualified | No (Industrial Grade Only) |
| Temp range (Celcius) | -40 to 85 |

NOTE: The CPU speed noted above is the maximum available speed. The CPU is clocked by HFCLK, listed in the System Clocks section below.

Table 2 lists the device resources that this design uses:

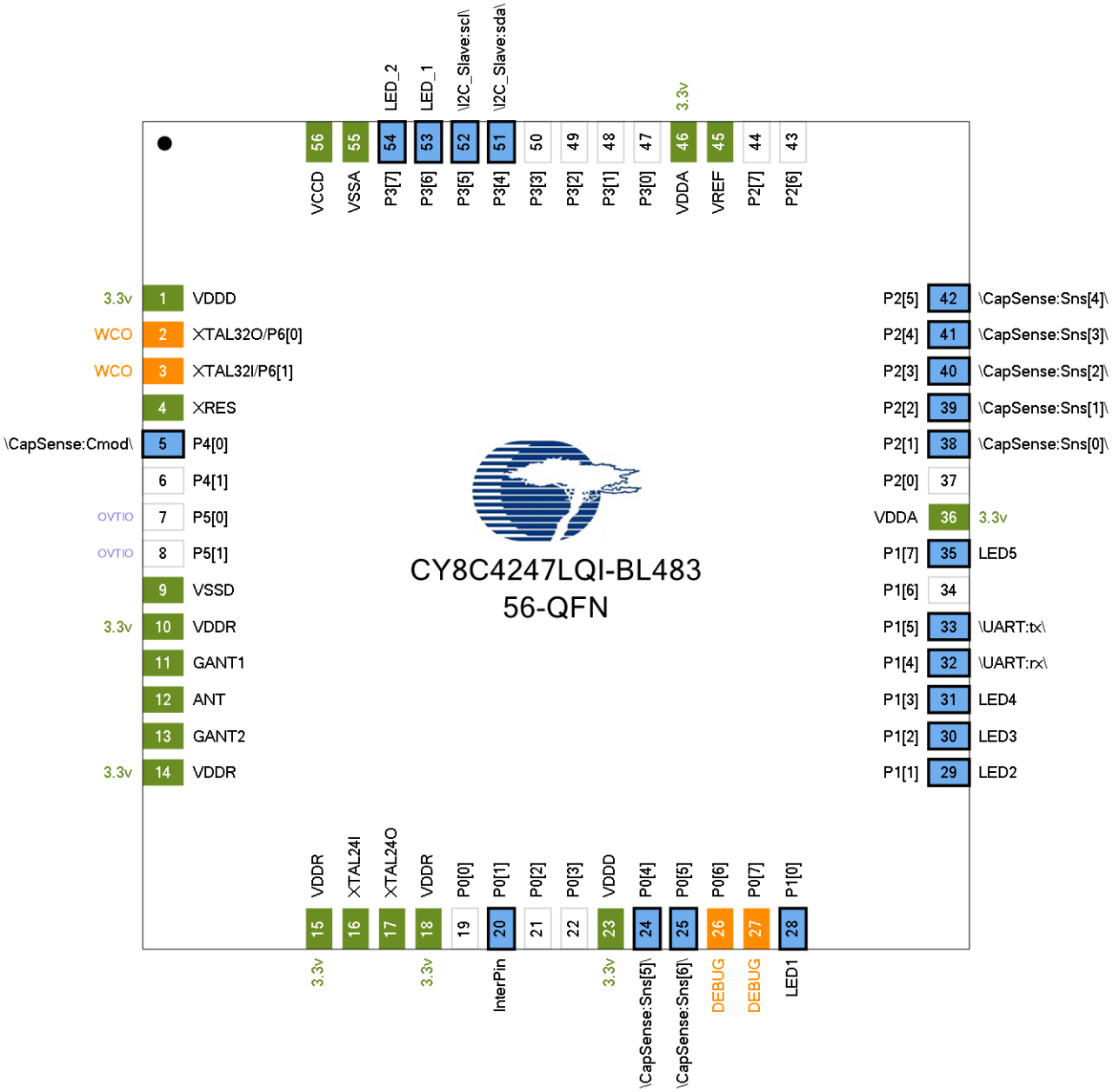
Table 2. Device Resources

| Resource Type | Used | Free | Max | % Used |
|---|---|---|---|---|
| Digital Clocks | 0 | 4 | 4 | 0.00 % |
| Interrupts | 2 | 30 | 32 | 6.25 % |
| IO | 24 | 14 | 38 | 63.16 % |
| Segment LCD | 0 | 1 | 1 | 0.00 % |
| CapSense | 1 | 0 | 1 | 100.00 % |
| Die Temp | 0 | 1 | 1 | 0.00 % |
| Serial Communication (SCB) | 2 | 0 | 2 | 100.00 % |
| BLE | 0 | 1 | 1 | 0.00 % |
| Timer/Counter/PWM | 1 | 3 | 4 | 25.00 % |
| UDB | | | | |
| Macrocells | 1 | 31 | 32 | 3.13 % |
| Unique P-terms | 0 | 64 | 64 | 0.00 % |
| Total P-terms | 0 | | | |
| Datapath Cells | 0 | 4 | 4 | 0.00 % |
| Status Cells | 0 | 4 | 4 | 0.00 % |
| Control Cells | 0 | 4 | 4 | 0.00 % |
| Comparator/Opamp | 0 | 4 | 4 | 0.00 % |
| LP Comparator | 0 | 2 | 2 | 0.00 % |
| SAR ADC | 0 | 1 | 1 | 0.00 % |
| DAC | | | | |
| 7-bit IDAC | 1 | 0 | 1 | 100.00 % |
| 8-bit IDAC | 1 | 0 | 1 | 100.00 % |

# 2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



## 2.1 Hardware Pins

Table 3 contains information about the pins on this device in device pin order. (No connection ["n/c"] pins have been omitted.)

Table 3. Device Pins

| Pin | Port | Name | Type | Drive Mode |
|---|---|---|---|---|
| 1 | VDDD | VDDD | Power | |
| 2 | XTAL32O/P6[0] | XTAL 32kHz:Xo | Reserved | |
| 3 | XTAL32I/P6[1] | XTAL 32kHz:Xi | Reserved | |
| 4 | XRES | XRES | Dedicated | |
| 5 | P4[0] | \CapSense:Cmod\ | Analog | HiZ analog |
| 6 | P4[1] | GPIO [unused] | | |
| 7 | P5[0] | OVT IO [unused] | | |
| 8 | P5[1] | OVT IO [unused] | | |
| 9 | VSSD | VSSD | Power | |
| 10 | VDDR | VDDR | Power | |
| 11 | GANT1 | GANT1 | Dedicated | |
| 12 | ANT | ANT | Dedicated | |
| 13 | GANT2 | GANT2 | Dedicated | |
| 14 | VDDR | VDDR | Power | |
| 15 | VDDR | VDDR | Power | |
| 16 | XTAL24I | XTAL24I | Dedicated | |
| 17 | XTAL24O | XTAL24O | Dedicated | |
| 18 | VDDR | VDDR | Power | |
| 19 | P0[0] | GPIO [unused] | | |
| 20 | P0[1] | InterPin | Software Output | Strong drive |
| 21 | P0[2] | GPIO [unused] | | |
| 22 | P0[3] | GPIO [unused] | | |
| 23 | VDDD | VDDD | Power | |
| 24 | P0[4] | \CapSense:Sns[5]\ | Analog | HiZ analog |
| 25 | P0[5] | \CapSense:Sns[6]\ | Analog | HiZ analog |
| 26 | P0[6] | Debug:SWD_IO | Reserved | |
| 27 | P0[7] | Debug:SWD_CK | Reserved | |
| 28 | P1[0] | LED1 | Software Output | Res pull down |
| 29 | P1[1] | LED2 | Software Output | Res pull down |
| 30 | P1[2] | LED3 | Software Output | Res pull down |
| 31 | P1[3] | LED4 | Software Output | Res pull down |
| 32 | P1[4] | \UART:rx\ | Dgtl In | HiZ digital |
| 33 | P1[5] | \UART:tx\ | Dgtl Out | Strong drive |
| 34 | P1[6] | GPIO [unused] | | |
| 35 | P1[7] | LED5 | Software Output | Res pull down |
| 36 | VDDA | VDDA | Power | |
| 37 | P2[0] | GPIO [unused] | | |
| 38 | P2[1] | \CapSense:Sns[0]\ | Analog | HiZ analog |
| 39 | P2[2] | \CapSense:Sns[1]\ | Analog | HiZ analog |
| 40 | P2[3] | \CapSense:Sns[2]\ | Analog | HiZ analog |

| Pin | Port | Name | Type | Drive Mode |
|---|---|---|---|---|
| 41 | P2[4] | \CapSense:Sns[3]\ | Analog | HiZ analog |
| 42 | P2[5] | \CapSense:Sns[4]\ | Analog | HiZ analog |
| 43 | P2[6] | GPIO [unused] | | |
| 44 | P2[7] | GPIO [unused] | | |
| 45 | VREF | VREF | Dedicated | |
| 46 | VDDA | VDDA | Power | |
| 47 | P3[0] | GPIO [unused] | | |
| 48 | P3[1] | GPIO [unused] | | |
| 49 | P3[2] | GPIO [unused] | | |
| 50 | P3[3] | GPIO [unused] | | |
| 51 | P3[4] | \I2C_Slave:sda\ | Dgtl In | OD, DL |
| 52 | P3[5] | \I2C_Slave:scl\ | Dgtl In | OD, DL |
| 53 | P3[6] | LED_1 | Dgtl Out | Strong drive |
| 54 | P3[7] | LED_2 | Dgtl Out | Strong drive |
| 55 | VSSA | VSSA | Power | |
| 56 | VCCD | VCCD | Power | |

Abbreviations used in Table 3 have the following meanings:

- HiZ analog = High impedance analog
- Res pull down = Resistive pull down
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Dgtl Out = Digital Output
- OD, DL = Open drain, drives low

## 2.2 Hardware Ports

Table 4 contains information about the pins on this device in device port order. (No connection ["n/c"], power and dedicated pins have been omitted.)

Table 4. Device Ports

| Port | Pin | Name | Type | Drive Mode |
|---|---|---|---|---|
| P0[0] | 19 | GPIO [unused] | | |
| P0[1] | 20 | InterPin | Software Output | Strong drive |
| P0[2] | 21 | GPIO [unused] | | |
| P0[3] | 22 | GPIO [unused] | | |
| P0[4] | 24 | \CapSense:Sns[5]\ | Analog | HiZ analog |
| P0[5] | 25 | \CapSense:Sns[6]\ | Analog | HiZ analog |
| P0[6] | 26 | Debug:SWD_IO | Reserved | |
| P0[7] | 27 | Debug:SWD_CK | Reserved | |
| P1[0] | 28 | LED1 | Software Output | Res pull down |
| P1[1] | 29 | LED2 | Software Output | Res pull down |
| P1[2] | 30 | LED3 | Software Output | Res pull down |
| P1[3] | 31 | LED4 | Software Output | Res pull down |
| P1[4] | 32 | \UART:rx\ | Dgtl In | HiZ digital |
| P1[5] | 33 | \UART:tx\ | Dgtl Out | Strong drive |
| P1[6] | 34 | GPIO [unused] | | |
| P1[7] | 35 | LED5 | Software Output | Res pull down |
| P2[0] | 37 | GPIO [unused] | | |
| P2[1] | 38 | \CapSense:Sns[0]\ | Analog | HiZ analog |
| P2[2] | 39 | \CapSense:Sns[1]\ | Analog | HiZ analog |
| P2[3] | 40 | \CapSense:Sns[2]\ | Analog | HiZ analog |
| P2[4] | 41 | \CapSense:Sns[3]\ | Analog | HiZ analog |
| P2[5] | 42 | \CapSense:Sns[4]\ | Analog | HiZ analog |
| P2[6] | 43 | GPIO [unused] | | |
| P2[7] | 44 | GPIO [unused] | | |
| P3[0] | 47 | GPIO [unused] | | |
| P3[1] | 48 | GPIO [unused] | | |
| P3[2] | 49 | GPIO [unused] | | |
| P3[3] | 50 | GPIO [unused] | | |
| P3[4] | 51 | \I2C_Slave:sda\ | Dgtl In | OD, DL |
| P3[5] | 52 | \I2C_Slave:scl\ | Dgtl In | OD, DL |
| P3[6] | 53 | LED_1 | Dgtl Out | Strong drive |
| P3[7] | 54 | LED_2 | Dgtl Out | Strong drive |
| P4[0] | 5 | \CapSense:Cmod\ | Analog | HiZ analog |
| P4[1] | 6 | GPIO [unused] | | |
| P5[0] | 7 | OVT IO [unused] | | |
| P5[1] | 8 | OVT IO [unused] | | |
| XTAL32I/P6[1] | 3 | XTAL 32kHz:Xi | Reserved | |
| XTAL32O/P6[0] | 2 | XTAL 32kHz:Xo | Reserved | |

Abbreviations used in Table 4 have the following meanings:

- HiZ analog = High impedance analog
- Res pull down = Resistive pull down
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Dgtl Out = Digital Output
- OD, DL = Open drain, drives low

## 2.3 Software Pins

Table 5 contains information about the software pins on this device in alphabetical order. (Only software-accessible pins are shown.)

Table 5. Software Pins

| Name | Port | Type |
|---|---|---|
| \CapSense:Cmod\ | P4[0] | Analog |
| \CapSense:Sns[0]\ | P2[1] | Analog |
| \CapSense:Sns[1]\ | P2[2] | Analog |
| \CapSense:Sns[2]\ | P2[3] | Analog |
| \CapSense:Sns[3]\ | P2[4] | Analog |
| \CapSense:Sns[4]\ | P2[5] | Analog |
| \CapSense:Sns[5]\ | P0[4] | Analog |
| \CapSense:Sns[6]\ | P0[5] | Analog |
| \I2C_Slave:scl\ | P3[5] | Dgtl In |
| \I2C_Slave:sda\ | P3[4] | Dgtl In |
| \UART:rx\ | P1[4] | Dgtl In |
| \UART:tx\ | P1[5] | Dgtl Out |
| Debug:SWD_CK | P0[7] | Reserved |
| Debug:SWD_IO | P0[6] | Reserved |
| GPIO [unused] | P2[7] | |
| GPIO [unused] | P1[6] | |
| GPIO [unused] | P2[6] | |
| GPIO [unused] | P3[0] | |
| GPIO [unused] | P2[0] | |
| GPIO [unused] | P3[1] | |
| GPIO [unused] | P0[0] | |
| GPIO [unused] | P0[2] | |
| GPIO [unused] | P4[1] | |
| GPIO [unused] | P3[3] | |
| GPIO [unused] | P0[3] | |
| GPIO [unused] | P3[2] | |
| InterPin | P0[1] | Software Output |
| LED_1 | P3[6] | Dgtl Out |
| LED_2 | P3[7] | Dgtl Out |
| LED1 | P1[0] | Software Output |
| LED2 | P1[1] | Software Output |
| LED3 | P1[2] | Software Output |
| LED4 | P1[3] | Software Output |
| LED5 | P1[7] | Software Output |
| OVT IO [unused] | P5[0] | |
| OVT IO [unused] | P5[1] | |
| XTAL 32kHz:Xi | XTAL32I/P6[1] | Reserved |
| XTAL 32kHz:Xo | XTAL32O/P6[0] | Reserved |

Abbreviations used in Table 5 have the following meanings:

- Dgtl In = Digital Input
- Dgtl Out = Digital Output

For more information on reading, writing and configuring pins, please refer to:

- Pins chapter in the System Reference Guide
  - CyPins API routines
- Programming Application Interface section in the cy_pins component datasheet

# 3 System Settings

## 3.1 System Configuration

Table 6. System Configuration Settings

| Name | Value |
|---|---|
| Device Configuration Mode | Compressed |
| Unused Bonded IO | Allow but warn |
| Heap Size (bytes) | 0x0100 |
| Stack Size (bytes) | 0x0400 |
| Include CMSIS Core Peripheral Library Files | True |

## 3.2 System Debug Settings

Table 7. System Debug Settings

| Name | Value |
|---|---|
| Chip Protection | Open |
| Debug Select | SWD (serial wire debug) |

## 3.3 System Operating Conditions

Table 8. System Operating Conditions

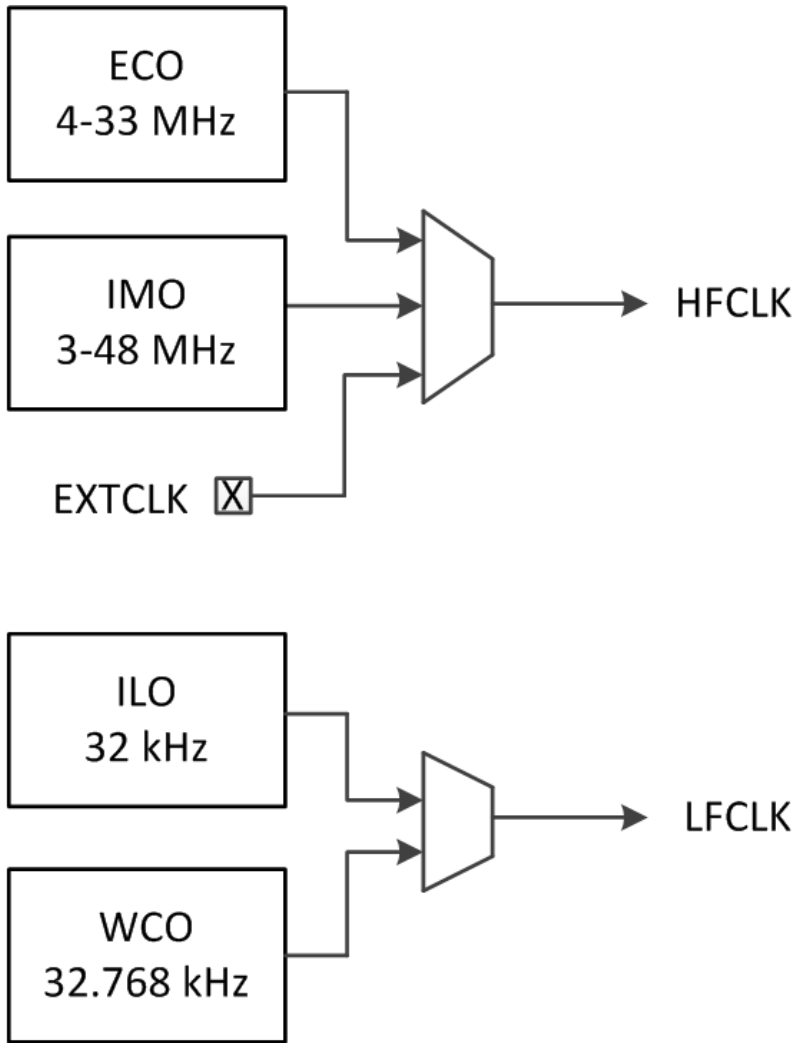
| Name | Value |
|---|---|
| Variable VDDA | True |
| VDDA (V) | 3.3 |
| VDDD (V) | 3.3 |
| VDDR (V) | 3.3 |

# 4 Clocks

The clock system includes these clock resources:

- Four internal clock sources:
  - 3 to 48 MHz Internal Main Oscillator (IMO) ±2% at 3 MHz
  - 4 to 33 MHz External Crystal Oscillator (ECO)
  - 32 kHz Internal Low Speed Oscillator (ILO) output
  - 32.768 kHz Watch Crystal Oscillator (ILO) output
- HFCLK can be generated using an external signal from EXTCLK pin
- Twelve clock dividers, each with 16-bit divide capability:
  - Eight can be used for fixed-function blocks
  - Four can be used for the UDBs

Figure 3. System Clock Configuration



## 4.1 System Clocks

Table 9 lists the system clocks used in this design.

Table 9. System Clocks

| Name | Domain | Source | Desired Freq | Nominal Freq | Accuracy (%) | Start at Reset | Enabled |
|---|---|---|---|---|---|---|---|
| Direct_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| SYSCLK | NONE | HFCLK | ? MHz | 24 MHz | ±2 | True | True |
| ECO | NONE | | 24 MHz | 24 MHz | ±0 | True | True |
| PLL0_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| HFCLK | NONE | Direct_Sel | 24 MHz | 24 MHz | ±2 | True | True |
| IMO | NONE | | 24 MHz | 24 MHz | ±2 | True | True |
| PLL1_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| WCO | NONE | | 32.768 kHz | 32.768 kHz | ±0.015 | True | True |
| LFCLK | NONE | ILO | ? MHz | 32 kHz | ±60 | True | True |
| ILO | NONE | | 32 kHz | 32 kHz | ±60 | True | True |
| RTC_Sel | NONE | None | ? MHz | ? MHz | ±0 | True | True |
| DigSig1 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| DigSig2 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| DigSig4 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| EXTCLK | NONE | | 24 MHz | ? MHz | ±0 | False | False |
| Timer0 (WDT0) | NONE | LFCLK | ? MHz | ? MHz | ±0 | False | False |
| DigSig3 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| Timer2 (WDT2) | NONE | LFCLK | ? MHz | ? MHz | ±0 | False | False |
| Timer1 (WDT1) | NONE | LFCLK | ? MHz | ? MHz | ±0 | False | False |

## 4.2 Local and Design Wide Clocks

Local clocks drive individual analog and digital blocks. Design wide clocks are a user-defined optimization, where two or more analog or digital blocks that share a common clock profile (frequency, etc) can be driven from the same clock divider output source.

Figure 4. Local and Design Wide Clock Configuration

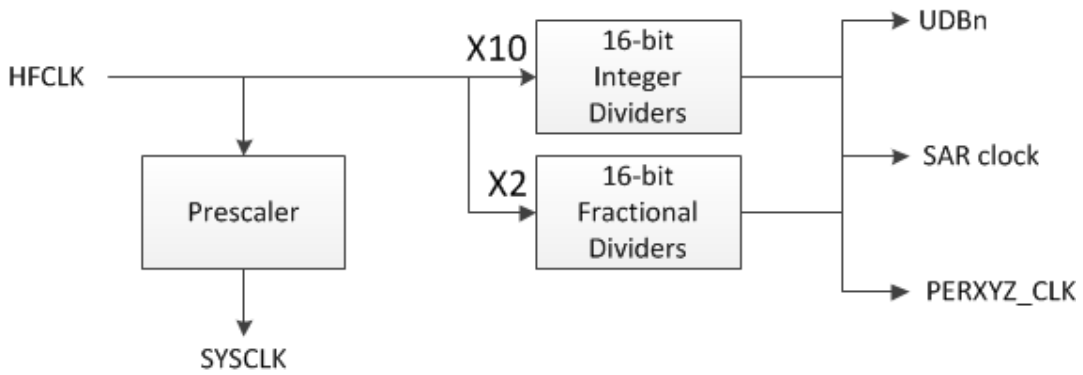



Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

| Name | Domain | Source | Desired Freq | Nominal Freq | Accuracy (%) | Start at Reset | Enabled |
|---|---|---|---|---|---|---|---|
| Clock_PWM | FIXED_-FUNCT-ION | HFCLK | 12 MHz | 12 MHz | ±2 | True | True |
| I2C_Slave_-SCBCLK | FIXED_-FUNCT-ION | HFCLK | 1.55 MHz | 1.6 MHz | ±2 | True | True |
| UART_SCBCLK | FIXED_-FUNCT-ION | HFCLK | 1.382 MHz | 1.412 MHz | ±2 | True | True |
| CapSense_-SampleClk | FIXED_-FUNCT-ION | HFCLK | ? MHz | 94.118 kHz | ±2 | True | True |
| CapSense_-SenseClk | FIXED_-FUNCT-ION | HFCLK | ? MHz | 94.118 kHz | ±2 | True | True |

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 4 Technical Reference Manual
- Clocking chapter in the System Reference Guide
  - CySysClkImo API routines
  - CySysClkIlo API routines
  - CySysClkEco API routines
  - CySysClkWco API routines
  - CySysClkWrite API routines

# 5 Interrupts

## 5.1 Interrupts

This design contains the following interrupt components: (0 is the highest priority)

Table 11. Interrupts

| Name | Priority | Vector |
|---|---|---|
| CapSense_ISR | 3 | 16 |
| I2C_Slave_SCB_IRQ | 3 | 10 |

For more information on interrupts, please refer to:

- Interrupt Controller chapter in the PSoC 4 Technical Reference Manual
- Interrupts chapter in the System Reference Guide
  - CyInt API routines and related registers
- Datasheet for cy_isr component

# 6 Flash Memory

PSoC 4 devices offer a host of Flash protection options and device security features that you can leverage to meet the security and protection requirements of an application. These requirements range from protecting configuration settings or Flash data to locking the entire device from external access.

Table 12 lists the Flash protection settings for your design.

Table 12. Flash Protection Settings

| Start Address | End Address | Protection Level |
|---|---|---|
| 0x0 | 0x1FFFF | U - Unprotected |

Flash memory is organized as rows with each row of flash having 128 bytes. Each flash row can be assigned one of four protection levels:

- U - Unprotected
- W - Full Protection

For more information on Flash memory and protection, please refer to:

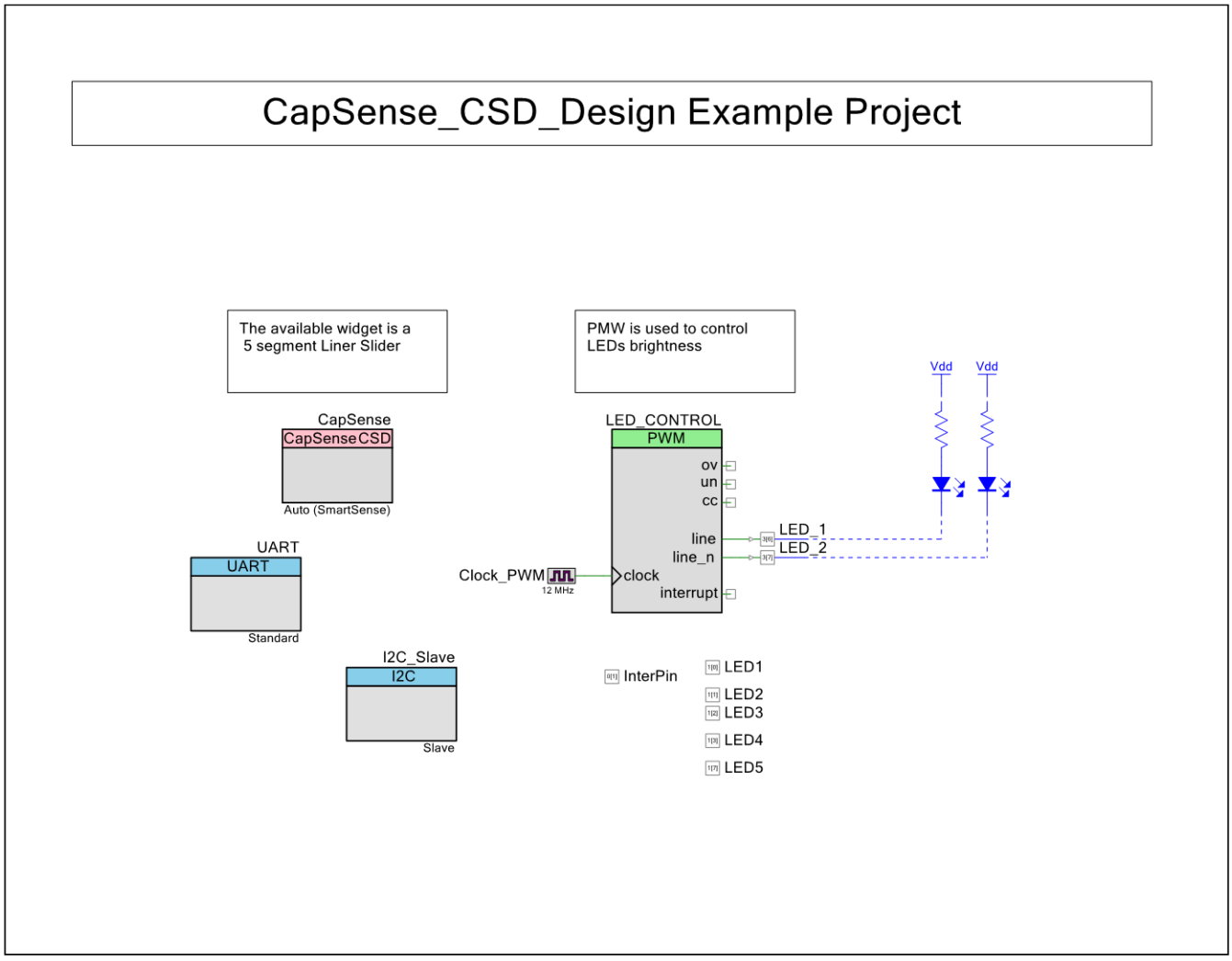
- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CySysFlash API routines

# 7 Design Contents

This design's schematic content consists of the following schematic sheet:

## 7.1 Schematic Sheet: Schematic

Figure 5. Schematic Sheet: Schematic




This schematic sheet contains the following component instances:

- Instance CapSense (type: CapSense_CSD_P4_v2_40)
- Instance I2C_Slave (type: SCB_P4_v3_10)
- Instance LED_CONTROL (type: TCPWM_P4_v2_10)
- Instance UART (type: SCB_P4_v3_10)

# 8 Components

## 8.1 Component type: CapSense_CSD_P4 [v2.40]

### *8.1.1 Instance CapSense*

**Description: The CapSense CSD component provides capacitive buttons, sliders, touch pads, matrix buttons and proximity sensors.**
**Instance type: CapSense_CSD_P4 [v2.40]**
**Datasheet: online component datasheet for CapSense_CSD_P4**

Table 13. Component Parameters for CapSense

| Parameter Name | Value | Description |
|---|---|---|
| AdvCrossCouplingThreshold | 5 | This value must be equal to the value of a sensor when a finger is near the sensor, but it is not touching the sensor. This can be determined by slowly dragging a finger across the panel and finding the inflection point of the difference counts at the base of the curve. The difference value at this point should be the Cross Coupling Threshold. The default value is 0x5. |
| AdvPenultimateThreshold | 100 | This value is the threshold for determining an arrival at the edge. This value may have to be increased for small diamonds, so that the edge handling is initiated sooner. If this number is too high, there is jumping at the edge with a smaller finger. If this number is too low, there is jumping at the edge with a larger finger. The default value is 0x5. |
| AdvVirtualSensorThreshold | 100 | This value must be set to the value of any sensor when a middle sized finger is placed directly over it. If this value is too low, the finger is followed by the resolved location. If this value is too high, the finger is led by the resolved location. The default value is 0x64. |
| AnalogSwitchDivider | 12 | Defines the clock divider for an analog switch source. |
| AvgSamplesNumber | 1 | Defines the number of samples for averaging during the calibration procedure. |
| BallisticMultiplierEnabled | false | A ballistic multiplier is used to provide better pointer movement experience for the user. |

| Parameter Name | Value | Description |
|---|---|---|
| CalibrationResolution | 7 | Defines the scanning resolution setting during the calibration procedure. |
| CmodPrecharge | Precharge byVref buffer | Defines the pre-charge option for the Cmod capacitor. |
| ConnectInactiveSensors | Ground | Defines the sensor inactive state. |
| CshTankPrecharge | Precharge byVref buffer | Defines the pre-charge option for the Csh_tank capacitor. |
| CurrentSource | IDAC Sourcing | Defines the IDAC mode. |
| CustomEzI2CInstanceName | SCB | Default instance name of the Tuner communication component. |
| DbPwmMode | OFF | This parameter sets the dead mand PWM modulator mode. |
| DbPwmPeriod | 0 | This parameter sets the period of the dead band PWM modulator. |
| DfbEnable | false | Enables the digital filter on the CSD comparator. |
| DynamicButtonNumber | 5 | The number of buttons that can be created from the slider segments. |
| EmcSensorNumber | 10 | Defines the number of channels that can support the HIGH immunity mode. |
| EnableAutoCalibration | true | Enables the Auto Calibration in the manual tuning mode. |
| EnableBIST | false | Indicates whether a Built-in self-test is enabled. |
| EnableTuneHelper | false | Allows generation of tuner APIs. |
| Gestures_-ActiveEdgeSwipeThreshold | 10 | This parameter sets the minimum active step distance (in pixels) from the point of a touch-down, near the edge, that has to be exceeded before the gesture is triggered. |
| Gestures_-BottomAngleThreshold | 45 | This parameter defines the maximum angle (in degrees) that the path of a finger can subtend on the point of a touch-down, near the edge. |
| Gestures_ClicksEnabled | true | Indicates whether click gestures enabled. |
| Gestures_ClickXRadiusPixels | 30 | This parameter sets the maximum X-Axis displacement for click gestures. |
| Gestures_ClickYRadiusPixels | 30 | This parameter sets the maximum Y-Axis displacement for click gestures. |
| Gestures_-DebounceTwoFingerSc-rollToZoomCount | 5 | This parameter sets the number of zoom gestures to be triggered for a valid zoom after a scroll gesture has been observed without removing the fingers from the trackpad. |

| Parameter Name | Value | Description |
|---|---|---|
| Gestures_-DebounceZoomCount | 3 | This parameter sets the number of sequential zoom gestures in a particular direction (in or out) that have to be observed before the zoom gesture is deemed valid. |
| Gestures_-DoubleClickMaxRadius | 20 | This parameter sets the maximum pixel radius that the second click in a double click sequence can extend. |
| Gestures_-DoubleClickMaxTimeout | 1000 | This parameter is the maximum time allowed between two sequential clicks so that a double click gesture is reported. |
| Gestures_-DoubleClickMinTimeout | 10 | This parameter sets the minimum duration between two sequential clicks before a double click operation is considered valid. |
| Gestures_-EdgeSwipeCompleteTimeout | 2000 | Defines time when Edge Swipe Gesture is complete. After that time the other gestures are allowed to detect. |
| Gestures_EdgeSwipesEnabled | true | Indicates whether the edge swipe gestures have been enabled. |
| Gestures_EdgeSwipeTime | 100 | For edge swipe a detection finger should exceed Edge Swipe Active Distance within Edge Swipe Timeout. |
| Gestures_FlickActiveDistance-ThresholdX | 30 | This parameter sets the minimum active step distance (in pixels) that has to be exceeded before motion is considered active. |
| Gestures_FlickActiveDistance-ThresholdY | 30 | This parameter sets the minimum active step distance (in pixels) that has to be exceeded before motion is considered active. |
| Gestures_FlickSampleTime | 3 | This parameter sets the number of similar, sequential flick gestures that should be performed before the flick motion is considered valid. |
| Gestures_FlicksEnabled | false | Indicates whether flick gestures have been enabled. |
| Gestures_OneFingerInertialSc-rollActiveDistanceThresholdX | 5 | This parameter sets the active distance in X direction that has to be exceeded before a lift-off event to trigger inertial scroll. |
| Gestures_OneFingerInertialSc-rollActiveDistanceThresholdY | 5 | This parameter sets the active distance in Y direction that has to be exceeded before a lift-off event to trigger inertial scroll. |
| Gestures_OneFingerInertialSc-rollCountLevel | Low | This use can select Low or High levels of inertial count. |

| Parameter Name | Value | Description |
|---|---|---|
| Gestures_-OneFingerScrollDebo-unceCount | 3 | This parameter sets the number of similar, sequential scroll gestures that should be performed before the scroll motion is considered valid. |
| Gestures_-OneFingerScrollsEnabled | true | Indicates whether one finger scroll gestures have been enabled. |
| Gestures_OneFingerScrollStep-1 | 1 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 1 X/Y. |
| Gestures_OneFingerScrollStep-2 | 3 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 2 X/Y. |
| Gestures_OneFingerScrollStep-3 | 5 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 3 X/Y. |
| Gestures_OneFingerScrollStep-4 | 7 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 4 X/Y. |
| Gestures_OneFingerScrollThre-shold1X | 5 | This parameter sets the active distance in X direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 1. |
| Gestures_OneFingerScrollThre-shold1Y | 5 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 1. |
| Gestures_OneFingerScrollThre-shold2X | 7 | This parameter sets the active distance in X direction that has to be exceeded to trigger second level scroll and update scroll step value parameter to scroll step 2. |
| Gestures_OneFingerScrollThre-shold2Y | 7 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 2. |
| Gestures_OneFingerScrollThre-shold3X | 8 | This parameter sets the active distance in X direction that has to be exceeded to trigger third level scroll and updates scroll step value parameter to scroll step 3. |

| Parameter Name | Value | Description |
|---|---|---|
| Gestures_OneFingerScrollThreshold3Y | 9 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 3. |
| Gestures_OneFingerScrollThreshold4X | 11 | This parameter sets the active distance in X direction that has to be exceeded to trigger fourth level scroll and updates scroll step value parameter to scroll step 4. |
| Gestures_OneFingerScrollThreshold4Y | 11 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 4. |
| Gestures_RotateActiveAngleThreshold | 90 | This parameter sets the minimum angle that should be made from the start of the gesture, before rotate gesture is reported. |
| Gestures_RotateDebounceLimit | 20 | This parameter sets the number of sequential pan gestures in a particular direction that have to be observed before the rotate gesture is deemed invalid. |
| Gestures_RotateEnabled | false | Indicates whether rotate gestures have been enabled. |
| Gestures_SingleClickMaxTimeout | 1000 | This parameter sets the maximum time during which a finger can be on the trackpad for a single click event to be considered valid. |
| Gestures_SingleClickMinTimeout | 20 | This parameter sets the minimum time during which a finger can be on the trackpad for a single click event to be considered valid. |
| Gestures_TopAngleThreshold | 45 | This parameter defines the maximum angle (in degrees) that the path of a finger can subtend on the point of touch-down, near the edge. |
| Gestures_TwoFingerClickMaxTimeout | 1000 | This parameter sets the maximum time during which two fingers can be placed on the trackpad before being disqualified as a two finger click event. |
| Gestures_TwoFingerClickMinTimeout | 100 | This parameter sets the minimum duration two fingers need to be on the trackpad before a two finger click event is registered. |

| Parameter Name | Value | Description |
|---|---|---|
| Gestures_TwoFingerInertialScrollActiveDistanceThresholdX | 5 | This parameter sets the active distance in X direction that has to be exceeded before a lift-off event to trigger inertial scroll. |
| Gestures_TwoFingerInertialScrollActiveDistanceThresholdY | 5 | This parameter sets the active distance in Y direction that has to be exceeded before a lift-off event to trigger inertial scroll. |
| Gestures_TwoFingerInertialScrollCountLevel | Low | This use can select Low or High levels of inertial count. |
| Gestures_-TwoFingerScrollDebounceCount | 3 | This parameter sets the number of similar, sequential scroll gestures that should be performed before the scroll motion is considered valid. |
| Gestures_-TwoFingerScrollsEnabled | true | Indicates whether two fingers scroll gestures have been enabled. |
| Gestures_TwoFingerScrollStep-1 | 1 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 1 X/Y. |
| Gestures_TwoFingerScrollStep-2 | 3 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 2 X/Y. |
| Gestures_TwoFingerScrollStep-3 | 5 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 3 X/Y. |
| Gestures_TwoFingerScrollStep-4 | 7 | This parameter sets number of scrolls to be reported when finger exceeds Scroll Threshold 4 X/Y. |
| Gestures_TwoFingerScrollThreshold1X | 5 | This parameter sets the active distance in X direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 1. |
| Gestures_TwoFingerScrollThreshold1Y | 5 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 1. |
| Gestures_TwoFingerScrollThreshold2X | 7 | This parameter sets the active distance in X direction that has to be exceeded to trigger second level scroll and update scroll step value parameter to scroll step 2. |
| Gestures_TwoFingerScrollThreshold2Y | 7 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 2. |

| Parameter Name | Value | Description |
|---|---|---|
| Gestures_TwoFingerScrollThreshold3X | 8 | This parameter sets the active distance in X direction that has to be exceeded to trigger third level scroll and updates scroll step value parameter to scroll step 3. |
| Gestures_TwoFingerScrollThreshold3Y | 9 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 3. |
| Gestures_TwoFingerScrollThreshold4X | 11 | This parameter sets the active distance in X direction that has to be exceeded to trigger fourth level scroll and updates scroll step value parameter to scroll step 4. |
| Gestures_TwoFingerScrollThreshold4Y | 11 | This parameter sets the active distance in Y direction that has to be exceeded to trigger first level scroll and updates scroll step value parameter to scroll step 4. |
| Gestures_TwoFingerSettlingCount | 3 | This parameter sets a delay threshold must be met before two finger gestures are computed. |
| Gestures_WidthOfDisambiguationRegion | 5 | This parameter sets the edge area for the edge swipe gestures. A valid edge swipe gesture should start within the width of disambiguation region. |
| Gestures_ZoomActiveDistanceThresholdX | 8 | This parameter sets the minimum active step distance (in pixels) that has to be cleared before a motion is considered an active zoom (in or out). |
| Gestures_ZoomActiveDistanceThresholdY | 8 | This parameter sets the minimum active step distance (in pixels) that has to be cleared before a motion is considered an active zoom (in or out). |
| Gestures_ZoomEnabled | true | Indicates whether zoom gestures have been enabled. |
| GlitchEliminateTimeout | 0 | Eliminates the glitch before scanning |
| GuardSensorEnable | false | Enables the Guard sensor. This type of sensor is typically required for water proof applications. |
| I2cCommunication | Internal I2C Communication | Defines whether the internal or external I2C communication component will be used. |
| IDACRange | 4x | Defines Idac Range for all sensors. |
| IDACsCount | 2 | Sets IDACs count. |
| ImmunityLevel | Low | Defines the noise immunity level. |

| Parameter Name | Value | Description |
|---|---|---|
| InputClkFreq | 3 | Defines the signal frequency used to drive the comparator latch and period counter. |
| IsStreetFighter | false | Defines whether the component is used for the StreeTFighter firmware. |
| IsTrackpadSupported | false | Defines whether the component supports the trackpad with gestures. |
| KValueScalingFactor | 8 | Defines the scaling factor of the K value for proximity sensors. |
| LowBaselineReset | 5 | Defines the number of samples with raw counts less than the baseline needed to make baseline snap down to the raw count level. |
| ModulatorClkDivider | 12 | Defines the modulator clock divider. |
| NegativeNoiseThreshold | 20 | Defines the negative difference between the raw count and baseline levels for baseline resting to the raw count level. |
| OversamplingEn | false | Enables the oversampling functionality. |
| PrechargeClkDivider | 12 | Defines the clock divider for an analog switch source. |
| PrechargeClkFreq | 3 | Defines the precharge signal frequency |
| PrsOptions | 12bits | Defines the source of the sensors switching controlling signal |
| RawDataFilterType | First Order IIR 1/4 | Defines the filter applied to raw data values. |
| SensorAutoReset | false | Enabling the auto reset causes baseline to always update regardless of whether the difference counts are above or below the noise threshold. When auto reset is disabled, Baseline only updates when difference counts are within the plus/minus noise threshold (the noise threshold is mirrored.) |
| SensorNumber | 7 | Total sensors count |
| SensorsFreqSettingsInd | true | Select individual frequency settings for each sensor or common for all sensors. |
| ShieldDelay | None | Defines a shield signal delay relative to the switches controlling signal. |
| ShieldEnable | false | Defines using the shield output. |
| ShieldTankEnable | false | Enables the external shield tank capacitor. |
| SliderThresholdsUpdateManual | false | This variable is intended to define the thresholds updating mods for sliders. |

| Parameter Name | Value | Description |
|---|---|---|
| SnsAlias | LinearSlider0_e0__LS, LinearSlider0_e1__LS, LinearSlider0_e2__LS, LinearSlider0_e3__LS, LinearSlider0_e4__LS, Button0_-_BTN, Button1__BTN | Contains all aliases for sensors. |
| ThresholdMode | false | Enables the Flexible Threshols in the Auto Tuning mode. |
| TrackpadApiResolutionCol | 100 | This parameter specifies the column API resolution of the trackpad with a gesture widget. |
| TrackpadApiResolutionRow | 100 | This parameter specifies the row API resolution of the trackpad with a gesture widget. |
| TrackpadExists | false | Indicates whether the trackpad with a gesture widget was added. |
| TunerIntfAddress | 8 | This parameter specifies the I2C 7-bits slave address (MSB ignored). |
| TunerIntfDataRate | 400 | This parameter specifies the I2C Data rate in kbps. The standard data rates are: 50, 100, 400 kbps. |
| TunerProperties | | Contains additional parameters required for the tuner. |
| TuningMethod | Auto | Defines the tuning method for the CapSense system. |
| WaterProofingEnabled | false | Enables special capsense system settings to use in water proof designs. |
| WidgetResolution | 8 | Defines the Signal resolution as uint8 or uint16. The valid values are 8 and 16. |

## 8.2 Component type: SCB_P4 [v3.10]

### *8.2.1 Instance I2C_Slave*

**Description: Serial Communication Block (SCB)**
**Instance type: SCB_P4 [v3.10]**
**Datasheet: online component datasheet for SCB_P4**

Table 14. Component Parameters for I2C_Slave

| Parameter Name | Value | Description |
|---|---|---|
| EzI2cBusVoltage | 3.3 | When the SCB mode is EZI2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus.<br><br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |

| Parameter Name | Value | Description |
|---|---|---|
| EzI2cByteModeEnable | false | When the SCB mode is EZI2C, this parameter specifies the number of bits per FIFO data element.<br>The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries.<br>The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries.<br><br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| EzI2cClockFromTerm | false | When the SCB mode is EZI2C, this parameter provides a clock terminal to connect a clock outside the component. |
| EzI2cClockStretching | true | When the SCB mode is EZI2C, this parameter specifies whether the SCL is stretched while in EZI2C operation. |
| EzI2cDataRate | 100 | When the SCB mode is EZI2C, this parameter defines EZI2C Data rate in kbps. The standard data rates are: 100, 400 and 1000 kbps. |
| EzI2cNumberOfAddresses | 1 | When the SCB mode is EZI2C, this parameter defines the number of I2C slave addresses that device respond to. |
| EzI2cPrimarySlaveAddress | 8 | When the SCB mode is EZI2C, this parameter specifies EZI2C primary 7-bits slave address (MSB ignored). |
| EzI2cSecondarySlaveAddress | 9 | When the SCB mode is EZI2C, this parameter specifies EZI2C secondary 7-bits slave address (MSB ignored).<br>Only applicable when EZI2C clock stretching option is set. |
| EzI2cSlewRate | Fast | When the SCB mode is EZI2C, this parameter specifies the slew rate settings of the I2C pins.<br>For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined.Refer to the Device Datasheet to determine which pins are GPIO_OVT capable. |

| Parameter Name | Value | Description |
|---|---|---|
| EzI2cSubAddressSize | 8 | When the SCB mode is EZI2C, this parameter specifies the maximum size of the slave buffer that is exposed to the master: 8bits – maximum buffer size is 256 bytes, 16 bits – maximum buffer size is 65535 bytes. |
| EzI2cWakeEnable | false | When the SCB mode is EZI2C, this parameter enables wakeup from Deep Sleep on I2C address match event. |
| I2cAcceptAddress | false | When the SCB mode is I2C, this parameter specifies whether to accept the match slave address in RX FIFO or not. All slave matched addresses are ACKed. The user has to register the callback function to handle accepted addresses. This feature has to be used when more than one address support is required. |
| I2cAcceptGeneralCall | false | When the SCB mode is I2C, this parameter specifies whether to accept the general call address. The general call address is ACKed when accepted and NAKed otherwise. The user has to register the callback function to handle the general call address. |
| I2cBusVoltage | 3.3 | When the SCB mode is I2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| I2cByteModeEnable | false | When the SCB mode is I2C, this parameter specifies the number of bits per FIFO data element. The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries.<br>The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| I2cClockFromTerm | false | When the SCB mode is I2C, this parameter provides a clock terminal to connect a clock outside the component. |

| Parameter Name | Value | Description |
|---|---|---|
| I2cDataRate | 100 | When the SCB mode is I2C, this parameter specifies the data rate in kbps. The standard data rates are: 100, 400 and 1000 kbps. |
| I2cExternIntrHandler | false | When the SCB mode is I2C, this parameter specifies whether the I2C interrupt handler is configured in SCB_I2CInit(). This parameter is intended to be used by the PM/SM bus component. The modification parameter default value causes I2C mode failures. |
| I2cManualOversampleControl | false | When the SCB mode is I2C, this parameter specifies the method of calculating the oversampling as manual or automatic. |
| I2cMode | Slave | When the SCB mode is I2C, this parameter defines the I2C operation mode as: Slave, Master, Multi-Master or Multi-Master-Slave. |
| I2cOvsFactor | 16 | When the SCB mode is I2C, this parameter defines the oversampling factor of SCBCLK. |
| I2cOvsFactorHigh | 8 | When the SCB mode is I2C, this parameter defines the high oversampling factor of SCBCLK. Only applicable for I2C Master modes. |
| I2cOvsFactorLow | 8 | When the SCB mode is I2C, this parameter defines the low oversampling factor of SCBCLK. Only applicable for I2C Master modes. |
| I2cSlaveAddress | 8 | When the SCB mode is I2C, this parameter specifies the I2C 7-bits slave address (MSB ignored). |
| I2cSlaveAddressMask | 254 | When the SCB mode is I2C, this parameter specifies the I2C Slave address mask. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the I2C slave address. |

| Parameter Name | Value | Description |
|---|---|---|
| I2cSlewRate | Fast | When the SCB mode is I2C, this parameter specifies the slew rate settings of the I2C pins. For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined. Refer to the Device Datasheet to determine which pins are GPIO_OVT capable. |
| I2cWakeEnable | false | When the SCB mode is I2C, this parameter enables wakeup from Deep Sleep on an I2C address match event. |
| ScbMisoSdaTxEnable | true | This parameter defines the availability of the spi_miso_i2c_-sda_uart_tx pin. |
| ScbMode | I2C | This parameter defines the mode of operation for the SCB component. |
| ScbMosiSclRxEnable | true | This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx pin. |
| ScbRxWakeIrqEnable | false | This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx_wake pin. |
| ScbSclkEnable | false | This parameter defines the availability of the sclk pin. |
| ScbSs0Enable | false | This parameter defines the availability of the ss0 pin. |
| ScbSs1Enable | false | This parameter defines the availability of the ss1 pin. |
| ScbSs2Enable | false | This parameter defines the availability of the ss2 pin. |
| ScbSs3Enable | false | This parameter defines the availability of the ss3 pin. |
| SpiBitRate | 1000 | When the SCB mode is SPI, this parameter specifies the Bit rate in kbps (up to 8000 kbps); the actual rate may differ based on available clock frequency and component settings. This parameter has no effect if the Clock from terminal parameter is enabled. |
| SpiBitsOrder | MSB First | When the SCB mode is SPI, this parameter defines the bit order as: MSB first or LSB first. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiByteModeEnable | false | When the SCB mode is SPI, this parameter specifies the number of bits per FIFO data element. The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiClockFromTerm | false | When the SCB mode is SPI, this parameter provides a clock terminal to connect a clock outside the component. |
| SpiFreeRunningSclk | false | When the SCB mode is SPI, this parameter specifies the SCLK generation by the master as: gated or free running (continuous). Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiInterruptMode | None | When the SCB mode is SPI, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside the component. |
| SpiIntrMasterSpiDone | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_M. SPI_DONE interrupt source. SCB.INTR_M. SPI_DONE: all data are sent into TX FIFO and the TX FIFO and the shifter register are emptied. Only applicable for SPI Master mode. |
| SpiIntrRxFull | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL trigger condition: RX FIFO is full. |
| SpiIntrRxNotEmpty | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY trigger condition: RX FIFO is not empty. There is at least one entry to get data from. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiIntrRxOverflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW trigger condition: attempt to write to a full RX FIFO. |
| SpiIntrRxTrigger | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER trigger condition: remains active until RX FIFO has more entries than the value specified by SpiRxTriggerLevel. |
| SpiIntrRxUnderflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW trigger condition: attempt to read from an empty RX FIFO. |
| SpiIntrSlaveBusError | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_SLAVE.BUS_-ERROR interrupt source. SCB.INTR_SLAVE.BUS_-ERROR trigger condition: slave select line is deselected at an unexpected time in the SPI transfer. Only applicable for SPI Slave mode. |
| SpiIntrTxEmpty | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY trigger condition: TX FIFO is empty. |
| SpiIntrTxNotFull | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL trigger condition: TX FIFO is not full. There is at least one entry to put data. |
| SpiIntrTxOverflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW trigger condition: attempt to write to a full TX FIFO. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiIntrTxTrigger | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER trigger condition: remains active until TX FIFO has fewer entries than the value specified by SpiTxTriggerLevel. |
| SpiIntrTxUnderflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW trigger condition: attempt to read from an empty TX FIFO. |
| SpiLateMisoSampleEnable | false | When the SCB mode is SPI, this parameter enables late sampling of the MISO line by the master. |
| SpiMedianFilterEnable | false | When the SCB mode is SPI, this parameter applies a digital 3 tap median filter to the SPI input line. |
| SpiMode | Slave | When the SCB mode is SPI, this parameter selects SPI mode of operation as: Slave or Master. |
| SpiNumberOfRxDataBits | 8 | When the SCB mode is SPI, this parameter specifies the number of data bits inside the SPI byte/word for RX direction. |
| SpiNumberOfSelectLines | 1 | When the SCB mode is SPI, this parameter defines the number of slave select lines. The SPI Slave has only one slave select line. The SPI Master has up to 4 lines. |
| SpiNumberOfTxDataBits | 8 | When the SCB mode is SPI, this parameter define the number of data bits inside the SPI byte/word for TX direction. |
| SpiOvsFactor | 16 | When the SCB mode is SPI, this parameter defines the oversampling factor of SCBCLK. |
| SpiRemoveMiso | false | When the SCB mode is SPI, this parameter removes the MISO pin. |
| SpiRemoveMosi | false | When the SCB mode is SPI, this parameter removes the MOSI pin. |
| SpiRemoveSclk | false | When the SCB mode is SPI, this parameter removes the SCLK pin. |
| SpiRxBufferSize | 8 | When the SCB mode is SPI, this parameter defines the size of the RX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiRxOutputEnable | false | When the SCB mode is SPI, this parameter enables the RX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| SpiRxTriggerLevel | 7 | When the SCB mode is SPI, this parameter defines the number of entries in the RX FIFO to control the SCB.INTR_-RX.TRIGGER interrupt event or RX DMA trigger output. |
| SpiSclkMode | CPHA = 0, CPOL = 0 | When the SCB mode is SPI, this parameter defines the serial clock phase (CPHA) and polarity (CPOL). |
| SpiSs0Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 0. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs1Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 1. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs2Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 2. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs3Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 3. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSubMode | Motorola | When the SCB mode is SPI, this parameter defines the sub mode of the SPI as: Motorola, TI(Start Coincides), TI(Start Precedes), or National Semiconductor. |
| SpiTransferSeparation | Continuous | When the SCB mode is SPI, this parameter defines the type of SPI transfers separation as: continuous or separated. |
| SpiTxBufferSize | 8 | When the SCB mode is SPI, this parameter defines the size of the TX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiTxOutputEnable | false | When the SCB mode is SPI, this parameter enables the TX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| SpiTxTriggerLevel | 0 | When the SCB mode is SPI, this parameter defines the number of entries in the TX FIFO to control the SCB.INTR_-TX.TRIGGER interrupt event or TX DMA trigger output. |
| SpiWakeEnable | false | When the SCB mode is SPI, this parameter enables wakeup from Deep Sleep on slave select event. |
| UartByteModeEnable | false | When the SCB mode is UART, this parameter specifies the number of bits per FIFO data element.<br>The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries.<br>The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartClockFromTerm | false | When the SCB mode is UART, this parameter provides a clock terminal to connect a clock outside the component. |
| UartCtsEnable | false | When the SCB mode is UART, this parameter enables the cts input.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartCtsPolarity | Active Low | When the SCB mode is UART, this parameter specifies active polarity of an input cts signal.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartDataRate | 115200 | When the SCB mode is UART, this parameter specifies the Baud rate in bps (up to 1000 kbps); the actual rate may differ based on available clock frequency and component settings. This parameter has no effect if the Clock from terminal parameter is enabled. |

| Parameter Name | Value | Description |
|---|---|---|
| UartDirection | TX + RX | When the SCB mode is UART, this parameter enables RX or TX direction or both simultaneously. |
| UartDropOnFrameErr | false | When the SCB mode is UART, this parameter defines whether the data is dropped from RX FIFO on a frame error event. |
| UartDropOnParityErr | false | When the SCB mode is UART, this parameter determines whether the data is dropped from RX FIFO on a parity error event. |
| UartInterruptMode | None | When the SCB mode is UART, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside component. |
| UartIntrRxFrameErr | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FRAME_-ERROR interrupt source. SCB.INTR_RX.FRAME_-ERROR trigger condition: frame error in received data frame. |
| UartIntrRxFull | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL trigger condition: RX FIFO is full. |
| UartIntrRxNotEmpty | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY trigger condition: RX FIFO is not empty. There is at least one entry to get data from. |
| UartIntrRxOverflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW trigger condition: attempt to write to a full RX FIFO. |
| UartIntrRxParityErr | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.PARITY_-ERROR interrupt source. SCB.INTR_RX.PARITY_-ERROR trigger condition: parity error in received data frame. |

| Parameter Name | Value | Description |
|---|---|---|
| UartIntrRxTrigger | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER trigger condition: remains active until RX FIFO has more entries than the value specified by UartRxTriggerLevel. |
| UartIntrRxUnderflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW trigger condition: attempt to read from an empty RX FIFO. |
| UartIntrTxEmpty | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY trigger condition: TX FIFO is empty. |
| UartIntrTxNotFull | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL trigger condition: TX FIFO is not full. There is at least one entry to put data. |
| UartIntrTxOverflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW trigger condition: attempt to write to a full TX FIFO. |
| UartIntrTxTrigger | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER trigger condition: remains active until TX FIFO has fewer entries than the value specified by UartTxTriggerLevel. |
| UartIntrTxUartDone | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_DONE interrupt source. SCB.INTR_TX.UART_DONE trigger condition: all data are sent in to TX FIFO and the transmit FIFO and the shifter register are emptied. |

| Parameter Name | Value | Description |
|---|---|---|
| UartIntrTxUartLostArb | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_ARB_-LOST interrupt source. SCB.INTR_TX.UART_ARB_-LOST trigger condition: UART lost arbitration, the value driven on the TX line is not the same as the value observed on the RX line. This event is useful when the transmitter and the receiver share a TX/RX line. Only applicable for UART SmartCard mode. |
| UartIntrTxUartNack | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_NACK interrupt source. SCB.INTR_TX.UART_NACK trigger condition: UART transmitter received a negative acknowledgement. Only applicable for UART SmartCard mode. |
| UartIntrTxUnderflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW trigger condition: attempt to read from an empty TX FIFO. |
| UartIrdaLowPower | false | When the SCB mode is UART, this parameter enables the low power receiver option. Only applicable for UART IrDA mode. |
| UartIrdaPolarity | Non-Inverting | When the SCB mode is UART, this parameter inverts the incoming RX line signal. Only applicable for UART IrDA mode. |
| UartMedianFilterEnable | false | When the SCB mode is UART, this parameter applies a digital 3 tap median filter to the UART input line. |
| UartMpEnable | false | When the SCB mode is UART, this parameter enables the UART multi-processor mode. Only applicable for UART Standard mode. |
| UartMpRxAcceptAddress | false | When the SCB mode is UART, this parameter define whether to put the matched UART address into RX FIFO. Only applicable for UART multi-processor mode. |

| Parameter Name | Value | Description |
|---|---|---|
| UartMpRxAddress | 2 | When the SCB mode is UART, this parameter defines the UART address. Only applicable for UART multi-processor mode. |
| UartMpRxAddressMask | 255 | When the SCB mode is UART, this parameter defines the address mask in multi-processor operation mode. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the UART address. Only applicable for UART multi-processor mode. |
| UartNumberOfDataBits | 8 bits | When the SCB mode is UART, this parameter defines the number of data bits inside the UART byte/word. |
| UartNumberOfStopBits | 1 bit | When the SCB mode is UART, this parameter defines the number of Stop bits. |
| UartOvsFactor | 12 | When the SCB mode is UART, this parameter defines the oversampling factor of SCBCLK. |
| UartParityType | None | When the SCB mode is UART, this parameter applies UART parity check as Odd or Even or discards the parity entirely. |
| UartRtsEnable | false | When the SCB mode is UART, this parameter enables the rts output. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRtsPolarity | Active Low | When the SCB mode is UART, this parameter specifies active polarity of the output rts signal. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRtsTriggerLevel | 4 | When the SCB mode is UART, this parameter specifies the number of entries in the RX FIFO to activate the rts output signal. When the receiver FIFO has fewer entries than the UartRtsTriggerLevel, an rts output signal is activated. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRxBufferSize | 8 | When the SCB mode is UART, this parameter defines the size of the RX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| UartRxOutputEnable | false | When the SCB mode is UART, this parameter enables the RX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| UartRxTriggerLevel | 7 | When the SCB mode is UART, this parameter defines the number of entries in the RX FIFO to trigger control the SCB.INTR_RX.TRIGGER interrupt event or RX DMA trigger output. |
| UartSmCardRetryOnNack | false | When the SCB mode is UART, this parameter defines whether to send a message again when a NACK response is received. Only applicable for UART SmartCard mode. |
| UartSubMode | Standard | When the SCB mode is UART, this parameter defines the sub mode of UART as: Standard, SmartCard or IrDA. |
| UartTxBufferSize | 8 | When the SCB mode is UART, this parameter defines the size of the TX buffer. |
| UartTxOutputEnable | false | When the SCB mode is UART, this parameter enables the TX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| UartTxTriggerLevel | 0 | When the SCB mode is UART, this parameter defines the number of entries in the TX FIFO to control the SCB.INTR_-TX.TRIGGER interrupt event or TX DMA trigger output. |
| UartWakeEnable | false | When the SCB mode is UART, this parameter enables the wakeup from Deep Sleep on start bit event. The actual wakeup source is RX GPIO. The skip start UART feature allows it to continue receiving bytes. |

### *8.2.2 Instance UART*

**Description: Serial Communication Block (SCB)**
**Instance type: SCB_P4 [v3.10]**
**Datasheet: online component datasheet for SCB_P4**

Table 15. Component Parameters for UART

| Parameter Name | Value | Description |
|---|---|---|
| EzI2cBusVoltage | 3.3 | When the SCB mode is EZI2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus.<br><br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| EzI2cByteModeEnable | false | When the SCB mode is EZI2C, this parameter specifies the number of bits per FIFO data element.<br>The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries.<br>The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries.<br><br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| EzI2cClockFromTerm | false | When the SCB mode is EZI2C, this parameter provides a clock terminal to connect a clock outside the component. |
| EzI2cClockStretching | true | When the SCB mode is EZI2C, this parameter specifies whether the SCL is stretched while in EZI2C operation. |
| EzI2cDataRate | 100 | When the SCB mode is EZI2C, this parameter defines EZI2C Data rate in kbps. The standard data rates are: 100, 400 and 1000 kbps. |
| EzI2cNumberOfAddresses | 1 | When the SCB mode is EZI2C, this parameter defines the number of I2C slave addresses that device respond to. |
| EzI2cPrimarySlaveAddress | 8 | When the SCB mode is EZI2C, this parameter specifies EZI2C primary 7-bits slave address (MSB ignored). |
| EzI2cSecondarySlaveAddress | 9 | When the SCB mode is EZI2C, this parameter specifies EZI2C secondary 7-bits slave address (MSB ignored).<br>Only applicable when EZI2C clock stretching option is set. |

| Parameter Name | Value | Description |
|---|---|---|
| EzI2cSlewRate | Fast | When the SCB mode is EZI2C, this parameter specifies the slew rate settings of the I2C pins. For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined.Refer to the Device Datasheet to determine which pins are GPIO_OVT capable. |
| EzI2cSubAddressSize | 8 | When the SCB mode is EZI2C, this parameter specifies the maximum size of the slave buffer that is exposed to the master: 8bits – maximum buffer size is 256 bytes, 16 bits – maximum buffer size is 65535 bytes. |
| EzI2cWakeEnable | false | When the SCB mode is EZI2C, this parameter enables wakeup from Deep Sleep on I2C address match event. |
| I2cAcceptAddress | false | When the SCB mode is I2C, this parameter specifies whether to accept the match slave address in RX FIFO or not. All slave matched addresses are ACKed. The user has to register the callback function to handle accepted addresses. This feature has to be used when more than one address support is required. |
| I2cAcceptGeneralCall | false | When the SCB mode is I2C, this parameter specifies whether to accept the general call address. The general call address is ACKed when accepted and NAKed otherwise. The user has to register the callback function to handle the general call address. |
| I2cBusVoltage | 3.3 | When the SCB mode is I2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |

| Parameter Name | Value | Description |
|---|---|---|
| I2cByteModeEnable | false | When the SCB mode is I2C, this parameter specifies the number of bits per FIFO data element. The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| I2cClockFromTerm | false | When the SCB mode is I2C, this parameter provides a clock terminal to connect a clock outside the component. |
| I2cDataRate | 100 | When the SCB mode is I2C, this parameter specifies the data rate in kbps. The standard data rates are: 100, 400 and 1000 kbps. |
| I2cExternIntrHandler | false | When the SCB mode is I2C, this parameter specifies whether the I2C interrupt handler is configured in SCB_I2CInit(). This parameter is intended to be used by the PM/SM bus component. The modification parameter default value causes I2C mode failures. |
| I2cManualOversampleControl | true | When the SCB mode is I2C, this parameter specifies the method of calculating the oversampling as manual or automatic. |
| I2cMode | Slave | When the SCB mode is I2C, this parameter defines the I2C operation mode as: Slave, Master, Multi-Master or Multi-Master-Slave. |
| I2cOvsFactor | 16 | When the SCB mode is I2C, this parameter defines the oversampling factor of SCBCLK. |
| I2cOvsFactorHigh | 8 | When the SCB mode is I2C, this parameter defines the high oversampling factor of SCBCLK. Only applicable for I2C Master modes. |
| I2cOvsFactorLow | 8 | When the SCB mode is I2C, this parameter defines the low oversampling factor of SCBCLK. Only applicable for I2C Master modes. |
| I2cSlaveAddress | 8 | When the SCB mode is I2C, this parameter specifies the I2C 7-bits slave address (MSB ignored). |

| Parameter Name | Value | Description |
|---|---|---|
| I2cSlaveAddressMask | 254 | When the SCB mode is I2C, this parameter specifies the I2C Slave address mask. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the I2C slave address. |
| I2cSlewRate | Fast | When the SCB mode is I2C, this parameter specifies the slew rate settings of the I2C pins. For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined. Refer to the Device Datasheet to determine which pins are GPIO_OVT capable. |
| I2cWakeEnable | false | When the SCB mode is I2C, this parameter enables wakeup from Deep Sleep on an I2C address match event. |
| ScbMisoSdaTxEnable | true | This parameter defines the availability of the spi_miso_i2c_-sda_uart_tx pin. |
| ScbMode | UART | This parameter defines the mode of operation for the SCB component. |
| ScbMosiSclRxEnable | true | This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx pin. |
| ScbRxWakeIrqEnable | false | This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx_wake pin. |
| ScbSclkEnable | false | This parameter defines the availability of the sclk pin. |
| ScbSs0Enable | false | This parameter defines the availability of the ss0 pin. |
| ScbSs1Enable | false | This parameter defines the availability of the ss1 pin. |
| ScbSs2Enable | false | This parameter defines the availability of the ss2 pin. |
| ScbSs3Enable | false | This parameter defines the availability of the ss3 pin. |
| SpiBitRate | 1000 | When the SCB mode is SPI, this parameter specifies the Bit rate in kbps (up to 8000 kbps); the actual rate may differ based on available clock frequency and component settings. This parameter has no effect if the Clock from terminal parameter is enabled. |
| SpiBitsOrder | MSB First | When the SCB mode is SPI, this parameter defines the bit order as: MSB first or LSB first. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiByteModeEnable | false | When the SCB mode is SPI, this parameter specifies the number of bits per FIFO data element. The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiClockFromTerm | false | When the SCB mode is SPI, this parameter provides a clock terminal to connect a clock outside the component. |
| SpiFreeRunningSclk | false | When the SCB mode is SPI, this parameter specifies the SCLK generation by the master as: gated or free running (continuous). Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiInterruptMode | None | When the SCB mode is SPI, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside the component. |
| SpiIntrMasterSpiDone | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_M. SPI_DONE interrupt source. SCB.INTR_M. SPI_DONE: all data are sent into TX FIFO and the TX FIFO and the shifter register are emptied. Only applicable for SPI Master mode. |
| SpiIntrRxFull | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL trigger condition: RX FIFO is full. |
| SpiIntrRxNotEmpty | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY trigger condition: RX FIFO is not empty. There is at least one entry to get data from. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiIntrRxOverflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW trigger condition: attempt to write to a full RX FIFO. |
| SpiIntrRxTrigger | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER trigger condition: remains active until RX FIFO has more entries than the value specified by SpiRxTriggerLevel. |
| SpiIntrRxUnderflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW trigger condition: attempt to read from an empty RX FIFO. |
| SpiIntrSlaveBusError | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_SLAVE.BUS_-ERROR interrupt source. SCB.INTR_SLAVE.BUS_-ERROR trigger condition: slave select line is deselected at an unexpected time in the SPI transfer. Only applicable for SPI Slave mode. |
| SpiIntrTxEmpty | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY trigger condition: TX FIFO is empty. |
| SpiIntrTxNotFull | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL trigger condition: TX FIFO is not full. There is at least one entry to put data. |
| SpiIntrTxOverflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW trigger condition: attempt to write to a full TX FIFO. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiIntrTxTrigger | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER trigger condition: remains active until TX FIFO has fewer entries than the value specified by SpiTxTriggerLevel. |
| SpiIntrTxUnderflow | false | When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW trigger condition: attempt to read from an empty TX FIFO. |
| SpiLateMisoSampleEnable | false | When the SCB mode is SPI, this parameter enables late sampling of the MISO line by the master. |
| SpiMedianFilterEnable | false | When the SCB mode is SPI, this parameter applies a digital 3 tap median filter to the SPI input line. |
| SpiMode | Slave | When the SCB mode is SPI, this parameter selects SPI mode of operation as: Slave or Master. |
| SpiNumberOfRxDataBits | 8 | When the SCB mode is SPI, this parameter specifies the number of data bits inside the SPI byte/word for RX direction. |
| SpiNumberOfSelectLines | 1 | When the SCB mode is SPI, this parameter defines the number of slave select lines. The SPI Slave has only one slave select line. The SPI Master has up to 4 lines. |
| SpiNumberOfTxDataBits | 8 | When the SCB mode is SPI, this parameter define the number of data bits inside the SPI byte/word for TX direction. |
| SpiOvsFactor | 16 | When the SCB mode is SPI, this parameter defines the oversampling factor of SCBCLK. |
| SpiRemoveMiso | false | When the SCB mode is SPI, this parameter removes the MISO pin. |
| SpiRemoveMosi | false | When the SCB mode is SPI, this parameter removes the MOSI pin. |
| SpiRemoveSclk | false | When the SCB mode is SPI, this parameter removes the SCLK pin. |
| SpiRxBufferSize | 8 | When the SCB mode is SPI, this parameter defines the size of the RX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiRxOutputEnable | false | When the SCB mode is SPI, this parameter enables the RX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| SpiRxTriggerLevel | 7 | When the SCB mode is SPI, this parameter defines the number of entries in the RX FIFO to control the SCB.INTR_-RX.TRIGGER interrupt event or RX DMA trigger output. |
| SpiSclkMode | CPHA = 0, CPOL = 0 | When the SCB mode is SPI, this parameter defines the serial clock phase (CPHA) and polarity (CPOL). |
| SpiSs0Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 0. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs1Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 1. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs2Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 2. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSs3Polarity | Active Low | When the SCB mode is SPI, this parameter specifies active polarity of slave select 3. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| SpiSubMode | Motorola | When the SCB mode is SPI, this parameter defines the sub mode of the SPI as: Motorola, TI(Start Coincides), TI(Start Precedes), or National Semiconductor. |
| SpiTransferSeparation | Continuous | When the SCB mode is SPI, this parameter defines the type of SPI transfers separation as: continuous or separated. |
| SpiTxBufferSize | 8 | When the SCB mode is SPI, this parameter defines the size of the TX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| SpiTxOutputEnable | false | When the SCB mode is SPI, this parameter enables the TX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| SpiTxTriggerLevel | 0 | When the SCB mode is SPI, this parameter defines the number of entries in the TX FIFO to control the SCB.INTR_-TX.TRIGGER interrupt event or TX DMA trigger output. |
| SpiWakeEnable | false | When the SCB mode is SPI, this parameter enables wakeup from Deep Sleep on slave select event. |
| UartByteModeEnable | false | When the SCB mode is UART, this parameter specifies the number of bits per FIFO data element.<br>The byte mode – false: a 16 bits FIFO data element. The FIFO depth is 8 entries.<br>The byte mode – true: an 8 bits FIFO data element. The FIFO depth is 16 entries.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartClockFromTerm | false | When the SCB mode is UART, this parameter provides a clock terminal to connect a clock outside the component. |
| UartCtsEnable | false | When the SCB mode is UART, this parameter enables the cts input.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartCtsPolarity | Active Low | When the SCB mode is UART, this parameter specifies active polarity of an input cts signal.<br>Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartDataRate | 115200 | When the SCB mode is UART, this parameter specifies the Baud rate in bps (up to 1000 kbps); the actual rate may differ based on available clock frequency and component settings. This parameter has no effect if the Clock from terminal parameter is enabled. |

| Parameter Name | Value | Description |
|---|---|---|
| UartDirection | TX + RX | When the SCB mode is UART, this parameter enables RX or TX direction or both simultaneously. |
| UartDropOnFrameErr | false | When the SCB mode is UART, this parameter defines whether the data is dropped from RX FIFO on a frame error event. |
| UartDropOnParityErr | false | When the SCB mode is UART, this parameter determines whether the data is dropped from RX FIFO on a parity error event. |
| UartInterruptMode | None | When the SCB mode is UART, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside component. |
| UartIntrRxFrameErr | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FRAME_-ERROR interrupt source. SCB.INTR_RX.FRAME_-ERROR trigger condition: frame error in received data frame. |
| UartIntrRxFull | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL trigger condition: RX FIFO is full. |
| UartIntrRxNotEmpty | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY trigger condition: RX FIFO is not empty. There is at least one entry to get data from. |
| UartIntrRxOverflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.OVERFLOW interrupt source. SCB.INTR_RX.OVERFLOW trigger condition: attempt to write to a full RX FIFO. |
| UartIntrRxParityErr | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.PARITY_-ERROR interrupt source. SCB.INTR_RX.PARITY_-ERROR trigger condition: parity error in received data frame. |

| Parameter Name | Value | Description |
|---|---|---|
| UartIntrRxTrigger | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.TRIGGER interrupt source. SCB.INTR_RX.TRIGGER trigger condition: remains active until RX FIFO has more entries than the value specified by UartRxTriggerLevel. |
| UartIntrRxUnderflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW trigger condition: attempt to read from an empty RX FIFO. |
| UartIntrTxEmpty | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.EMPTY interrupt source. SCB.INTR_TX.EMPTY trigger condition: TX FIFO is empty. |
| UartIntrTxNotFull | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL trigger condition: TX FIFO is not full. There is at least one entry to put data. |
| UartIntrTxOverflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW trigger condition: attempt to write to a full TX FIFO. |
| UartIntrTxTrigger | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.TRIGGER interrupt source. SCB.INTR_TX.TRIGGER trigger condition: remains active until TX FIFO has fewer entries than the value specified by UartTxTriggerLevel. |
| UartIntrTxUartDone | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_DONE interrupt source. SCB.INTR_TX.UART_DONE trigger condition: all data are sent in to TX FIFO and the transmit FIFO and the shifter register are emptied. |

| Parameter Name | Value | Description |
|---|---|---|
| UartIntrTxUartLostArb | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_ARB_-LOST interrupt source. SCB.INTR_TX.UART_ARB_-LOST trigger condition: UART lost arbitration, the value driven on the TX line is not the same as the value observed on the RX line. This event is useful when the transmitter and the receiver share a TX/RX line. Only applicable for UART SmartCard mode. |
| UartIntrTxUartNack | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UART_NACK interrupt source. SCB.INTR_TX.UART_NACK trigger condition: UART transmitter received a negative acknowledgement. Only applicable for UART SmartCard mode. |
| UartIntrTxUnderflow | false | When the SCB mode is UART, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW trigger condition: attempt to read from an empty TX FIFO. |
| UartIrdaLowPower | false | When the SCB mode is UART, this parameter enables the low power receiver option. Only applicable for UART IrDA mode. |
| UartIrdaPolarity | Non-Inverting | When the SCB mode is UART, this parameter inverts the incoming RX line signal. Only applicable for UART IrDA mode. |
| UartMedianFilterEnable | false | When the SCB mode is UART, this parameter applies a digital 3 tap median filter to the UART input line. |
| UartMpEnable | false | When the SCB mode is UART, this parameter enables the UART multi-processor mode. Only applicable for UART Standard mode. |
| UartMpRxAcceptAddress | false | When the SCB mode is UART, this parameter define whether to put the matched UART address into RX FIFO. Only applicable for UART multi-processor mode. |

| Parameter Name | Value | Description |
|---|---|---|
| UartMpRxAddress | 2 | When the SCB mode is UART, this parameter defines the UART address. Only applicable for UART multi-processor mode. |
| UartMpRxAddressMask | 255 | When the SCB mode is UART, this parameter defines the address mask in multi-processor operation mode. Bit value 0 – excludes bit from address comparison. Bit value 1 – the bit needs to match with the corresponding bit of the UART address. Only applicable for UART multi-processor mode. |
| UartNumberOfDataBits | 8 bits | When the SCB mode is UART, this parameter defines the number of data bits inside the UART byte/word. |
| UartNumberOfStopBits | 1 bit | When the SCB mode is UART, this parameter defines the number of Stop bits. |
| UartOvsFactor | 12 | When the SCB mode is UART, this parameter defines the oversampling factor of SCBCLK. |
| UartParityType | None | When the SCB mode is UART, this parameter applies UART parity check as Odd or Even or discards the parity entirely. |
| UartRtsEnable | false | When the SCB mode is UART, this parameter enables the rts output. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRtsPolarity | Active Low | When the SCB mode is UART, this parameter specifies active polarity of the output rts signal. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRtsTriggerLevel | 4 | When the SCB mode is UART, this parameter specifies the number of entries in the RX FIFO to activate the rts output signal. When the receiver FIFO has fewer entries than the UartRtsTriggerLevel, an rts output signal is activated. Only applicable for PSoC 4100 BLE/PSoC 4200 BLE/PSoC 4100 M/PSoC 4200 M devices. |
| UartRxBufferSize | 8 | When the SCB mode is UART, this parameter defines the size of the RX buffer. |

| Parameter Name | Value | Description |
|---|---|---|
| UartRxOutputEnable | false | When the SCB mode is UART, this parameter enables the RX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| UartRxTriggerLevel | 7 | When the SCB mode is UART, this parameter defines the number of entries in the RX FIFO to trigger control the SCB.INTR_RX.TRIGGER interrupt event or RX DMA trigger output. |
| UartSmCardRetryOnNack | false | When the SCB mode is UART, this parameter defines whether to send a message again when a NACK response is received. Only applicable for UART SmartCard mode. |
| UartSubMode | Standard | When the SCB mode is UART, this parameter defines the sub mode of UART as: Standard, SmartCard or IrDA. |
| UartTxBufferSize | 8 | When the SCB mode is UART, this parameter defines the size of the TX buffer. |
| UartTxOutputEnable | false | When the SCB mode is UART, this parameter enables the TX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller. |
| UartTxTriggerLevel | 0 | When the SCB mode is UART, this parameter defines the number of entries in the TX FIFO to control the SCB.INTR_-TX.TRIGGER interrupt event or TX DMA trigger output. |
| UartWakeEnable | false | When the SCB mode is UART, this parameter enables the wakeup from Deep Sleep on start bit event. The actual wakeup source is RX GPIO. The skip start UART feature allows it to continue receiving bytes. |

## 8.3 Component type: TCPWM_P4 [v2.10]

### *8.3.1 Instance LED_CONTROL*

**Description: 16-bit Timer Counter PWM (TCPWM)**
**Instance type: TCPWM_P4 [v2.10]**
**Datasheet: online component datasheet for TCPWM_P4**

Table 16. Component Parameters for LED_CONTROL

| Parameter Name | Value | Description |
|---|---|---|
| PWMCompare | 1 | The initial value for the comparison register when in the PWM mode |
| PWMCompareBuf | 65535 | The initial value for the second comparison register when in the PWM mode |
| PWMCompareSwap | Disable swap | Determines whether the PWM swap check box is enabled or disabled |
| PWMCountMode | Level | Determines whether the PWM counter counts at level detection or in various modes of edge detection |
| PWMCountPresent | false | Determines if the PWM count signal is present and controls the visibility of the count pin |
| PWMDeadTimeCycle | 0 | Sets the number of cycles of dead time insertion |
| PWMInterruptMask | None | The mask used for enabling the interrupt bit in the PWM mode |
| PWMKillEvent | Asynchronous | Selects whether a PWM kill event is synchronous or asynchronous to the input clock |
| PWMLinenSignal | Direct Output | Selects whether the PWM line_n signal is inverted or is directly output |
| PWMLineSignal | Direct Output | Selects whether the PWM line signal is inverted or is directly output |
| PWMMode | PWM | Selects one of the three PWM modes - PWM, PWM with dead time insertion, or Pseudo random PWM |
| PWMPeriod | 65535 | The initial value for the period register when in the PWM mode |
| PWMPeriodBuf | 65535 | The initial value for the second period register when in the PWM mode |
| PWMPeriodSwap | Disable swap | Enables swap between the PWM period and period_buf registers |
| PWMPrescaler | 0 | Defines the prescaler used to divide the TCPWM clock to create the counter clock |
| PWMReloadMode | Rising edge | Determines whether the PWM reload signal is accepted at level detection or in various modes of edge detection |
| PWMReloadPresent | false | Determines whether the PWM reload signal is present and controls its pin visibility |
| PWMRunMode | Continuous | Selects between continuous and one shot run mode for the PWM |

| Parameter Name | Value | Description |
|---|---|---|
| PWMSetAlign | Left align | Selects the alignment of the PWM waveform to be either left, right, center or asymmetrically aligned |
| PWMStartMode | Rising edge | Determines whether the PWM start signal is accepted at level detection or in various modes of edge detection |
| PWMStartPresent | false | Determines whether the PWM start signal is present and controls its pin visibility |
| PWMStopEvent | Don't stop on Kill | Selects whether to kill the PWM on a stop signal or not |
| PWMStopMode | Rising edge | Determines whether the PWM stop signal is accepted at level detection or in various modes of edge detection |
| PWMStopPresent | false | Determines whether the PWM stop signal is present and controls its pin visibility |
| PWMSwitchMode | Rising edge | Determines whether the PWM switch signal is accepted at level detection or in various modes of edge detection |
| PWMSwitchPresent | false | Determines whether the PWM switch signal is present and controls its pin visibility |
| QuadEncodingModes | x1 Encoding mode | Selects one of the three quadrature decoder modes – x1, x2, or x4 encoding mode |
| QuadIndexMode | Rising edge | Determines whether the Quadrature Decoder index signal is accepted at level detection or in various modes of edge detection |
| QuadIndexPresent | false | Determines whether the Quadrature Decoder index signal is present and controls its pin visibility |
| QuadInterruptMask | Terminal count mask | The mask used to configure which Quadrature Decoder event causes an interrupt |
| QuadPhiAMode | Level | Determines whether the Quadrature Decoder PhiA signal is accepted at level detection or in various modes of edge detection |
| QuadPhiBMode | Level | Determines whether the Quadrature Decoder PhiB signal is accepted at level detection or in various modes of edge detection |
| QuadStopMode | Rising edge | Determines whether the Quadrature Decoder stop signal is accepted at level detection or in various modes of edge detection |

| Parameter Name | Value | Description |
|---|---|---|
| QuadStopPresent | false | Determines whether the Quadrature Decoder stop signal is present and controls its pin visibility |
| TCCaptureMode | Rising edge | Determines whether the Timer/Counter capture signal is accepted at level detection or in various modes of edge detection |
| TCCapturePresent | false | Determines whether the Timer/Counter capture signal is present and controls its pin visibility |
| TCCompare | 65535 | The initial value for the comparison register when in the Timer/Counter mode |
| TCCompareBuf | 65535 | The initial value for the second comparison register when in the Timer/Counter mode |
| TCCompareSwap | Disable swap | Determines whether the Timer/Counter swap check box is enabled or disabled |
| TCCompCapMode | Capture Mode | Selects whether the Timer/Counter capture or the compare mode is enabled |
| TCCountingModes | Counts up | Selects the count direction of the counter |
| TCCountMode | Level | Determines whether the Timer/Counter count signal is accepted at a level detect or at various modes of edge detection |
| TCCountPresent | false | Determines whether the Timer/Counter count signal is present and controls its pin visibility |
| TCInterruptMask | Terminal count mask | The mask used to determine which Timer/Counter event causes an interrupt |
| TCPeriod | 65535 | The initial value for the Timer/Counter period register |
| TCPrescaler | 0 | Selects the prescaler value to apply to the Timer/Counter clock |
| TCPWMCapturePresent | false | Determines whether the Unconfigured capture signal is present and controls its pin visibility |
| TCPWMConfig | PWM | Selects the TCPWM mode - Unconfigured, Timer/Counter, PWM, or Quadrature Decoder |
| TCPWMCountPresent | false | Determines whether the Unconfigured count signal is present and controls its pin visibility |

| Parameter Name | Value | Description |
|---|---|---|
| TCPWMReloadPresent | false | Determines whether the Unconfigured reload signal is present and controls its pin visibility |
| TCPWMStartPresent | false | Determines whether the Unconfigured start signal is present and controls its pin visibility |
| TCPWMStopPresent | false | Determines whether the Unconfigured stop signal is present and controls its pin visibility |
| TCReloadMode | Rising edge | Determines whether the Timer/Counter reload signal is accepted at level detection or in various modes of edge detection |
| TCReloadPresent | false | Determines whether the Timer/Counter reload signal is present and controls its pin visibility |
| TCRunMode | Continuous | Selects whether the counter runs continuously or one shot |
| TCStartMode | Rising edge | Determines whether the start signal is accepted at level detection or in various modes of edge detection |
| TCStartPresent | false | Determines whether the Timer/Counter start signal is present and controls its pin visibility |
| TCStopMode | Rising edge | Determines whether the Timer/Counter stop signal is accepted at level detection or in various modes of edge detection |
| TCStopPresent | false | Determines whether the Timer/Counter stop signal is present and controls its pin visibility |

# 9 Other Resources

The following documents contain important information on Cypress software APIs that might be relevant to this design:

- Standard Types and Defines chapter in the System Reference Guide
  - Software base types
  - Hardware register types
  - Compiler defines
  - Cypress API return codes
  - Interrupt types and macros
- Registers
  - The full PSoC 4 register map is covered in the PSoC 4 Registers Technical Reference Manual
  - Register Access chapter in the System Reference Guide
    - CY_GET API routines
    - CY_SET API routines
- System Functions chapter in the System Reference Guide
  - General API routines
  - CyDelay API routines
  - CyVd Voltage Detect API routines
- Power Management
  - Power Supply and Monitoring chapter in the PSoC 4 Technical Reference Manual
  - Low Power Modes chapter in the PSoC 4 Technical Reference Manual
  - Power Management chapter in the System Reference Guide
    - CyPm API routines
- Watchdog Timer chapter in the System Reference Guide
  - CyWdt API routines