PSoC® Creator™
Project Datasheet for Configuracion_General

Creation Time: 08/22/2023 23:44:06
User: JAVIER\Usuario
Project: Configuracion_General
Tool: PSoC Creator 4.4

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 1
2 Pins 3
    2.1 Hardware Pins 4
    2.2 Hardware Ports 6
    2.3 Software Pins 8
3 System Settings 10
    3.1 System Configuration 10
    3.2 System Debug Settings 10
    3.3 System Operating Conditions 10
4 Clocks 11
    4.1 System Clocks 12
    4.2 Local and Design Wide Clocks 12
5 Interrupts 14
    5.1 Interrupts 14
6 Flash Memory 15
7 Design Contents 16
    7.1 Schematic Sheet: Page 1 16
8 Components 17
    8.1 Component type: CapSense_P4 [v7.10] 17
        8.1.1 Instance CapSense 17
    8.2 Component type: SCB_P4 [v4.0] 30
        8.2.1 Instance UART 30
    8.3 Component type: TCPWM_P4 [v2.10] 44
        8.3.1 Instance PWM1 44
        8.3.2 Instance PWM2 48
        8.3.3 Instance PWM3 52
9 Other Resources 57

1 Overview

CY8C4000S1 family is one of the smallest members of the PSoC 4 family of devices and is upward compatible with larger members of PSoC 4.

- High-performance, 32-bit single cycle Cortex-M0 CPU core
- Capacitive touch sensing (CapSense®)
- Configurable Timer/Counter/PWM block
- Two current sourcing/sinking DACs (IDACs)
- Comparator with 1.2 V reference
- Configurable I2C block with master, slave, and multi-master operating modes
- Low-power operating modes including Sleep and Deep-Sleep

Figure 1 shows the major components of a typical PSoC 4000S series member PSoC 4 device. For details on all the systems listed above, please refer to the PSoC 4 Technical Reference Manual .

Figure 1. PSoC 4000S Device Series Block Diagram

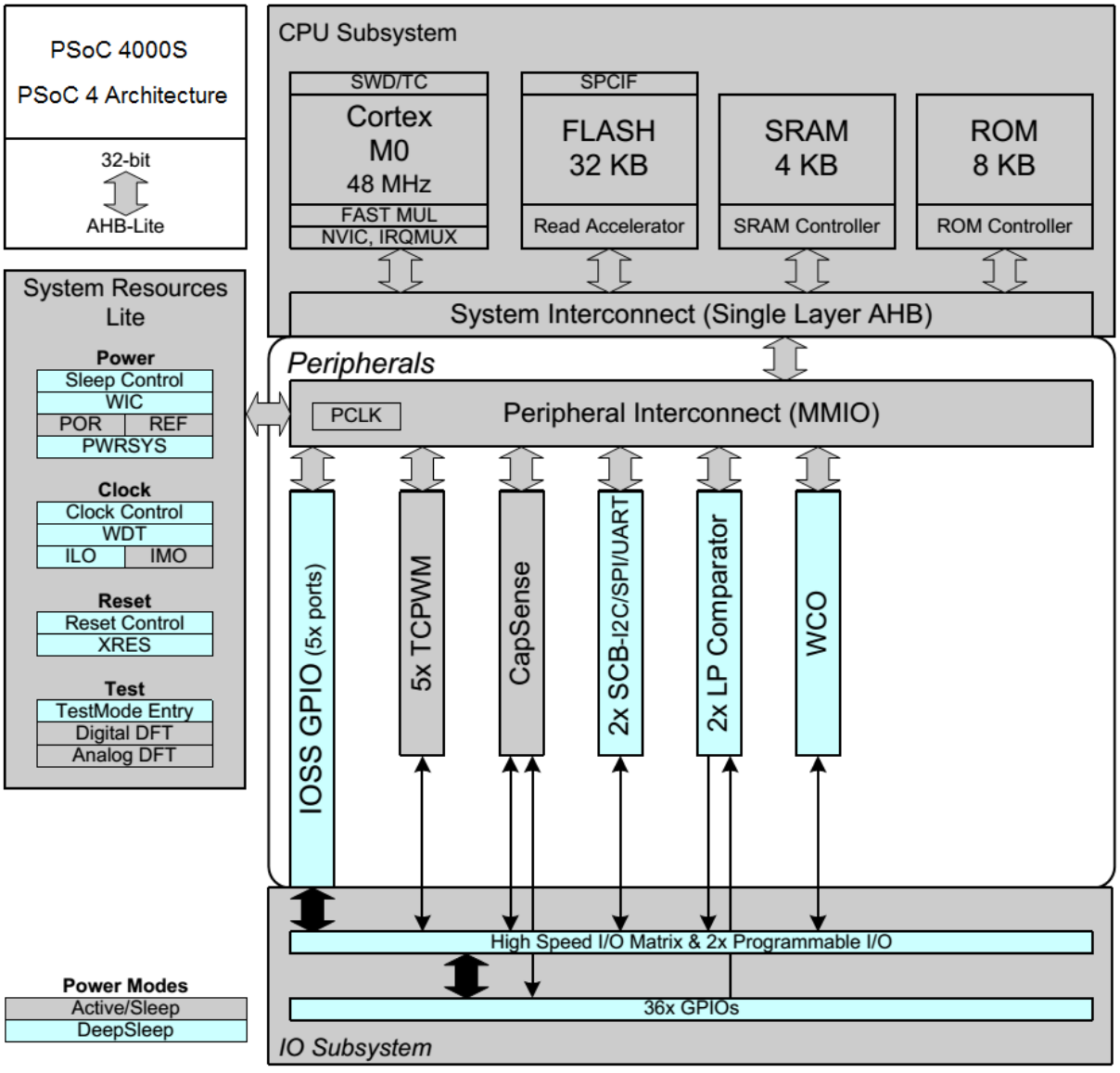


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C4045AZI-S413
Package Name	48-TQFP
Family	PSoC 4
Series	PSoC 4000S
Max CPU speed (MHz)	48
Flash size (kB)	32
SRAM size (kB)	4
Vdd range (V)	1.71 to 5.5
Automotive qualified	No (Industrial Grade Only)
Temp range (Celsius)	-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:

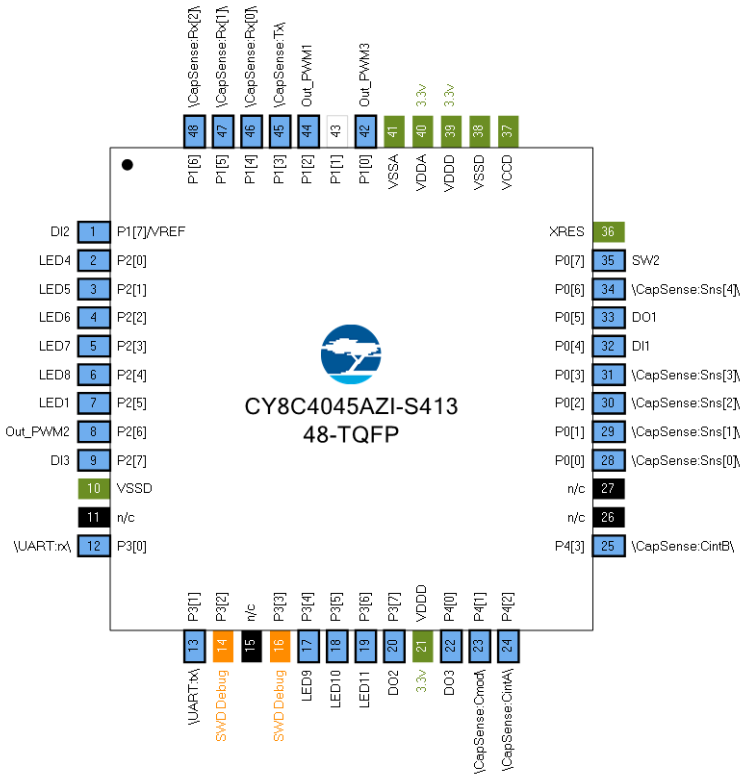
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Interrupts	1	15	16	6.25 %
IO	35	1	36	97.22 %
Segment LCD	0	1	1	0.00 %
CapSense	1	0	1	100.00 %
Serial Communication (SCB)	1	1	2	50.00 %
Timer/Counter/PWM	3	2	5	60.00 %
Smart IO Ports	0	2	2	0.00 %
Comparator	1	0	1	100.00 %
LP Comparator	0	2	2	0.00 %
DAC				
7-bit IDAC	2	0	2	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	P1[7]/VREF	DI2	Software In/Out	Res pull up
2	P2[0]	LED4	Software In/Out	Strong drive
3	P2[1]	LED5	Software In/Out	Strong drive
4	P2[2]	LED6	Software In/Out	Strong drive
5	P2[3]	LED7	Software In/Out	Strong drive
6	P2[4]	LED8	Software In/Out	Strong drive
7	P2[5]	LED1	Software In/Out	Strong drive
8	P2[6]	Out_PWM2	Dgtl Out	Strong drive
9	P2[7]	DI3	Software In/Out	Res pull up
10	VSSD	VSSD	Power	
12	P3[0]	\UART:rx\	Dgtl In	HiZ digital
13	P3[1]	\UART:tx\	Dgtl Out	Strong drive
14	P3[2]	Debug:SWD_IO	Reserved	
16	P3[3]	Debug:SWD_CK	Reserved	
17	P3[4]	LED9	Software In/Out	Strong drive
18	P3[5]	LED10	Software In/Out	Strong drive
19	P3[6]	LED11	Software In/Out	Strong drive
20	P3[7]	DO2	Software In/Out	Strong drive
21	VDDD	VDDD	Power	
22	P4[0]	DO3	Software In/Out	Strong drive
23	P4[1]	\CapSense:Cmod\	Analog	HiZ analog
24	P4[2]	\CapSense:CintA\	Analog	HiZ analog
25	P4[3]	\CapSense:CintB\	Analog	HiZ analog
28	P0[0]	\CapSense:Sns[0]\	Analog	HiZ analog
29	P0[1]	\CapSense:Sns[1]\	Analog	HiZ analog
30	P0[2]	\CapSense:Sns[2]\	Analog	HiZ analog
31	P0[3]	\CapSense:Sns[3]\	Analog	HiZ analog
32	P0[4]	DI1	Software In/Out	Res pull up
33	P0[5]	DO1	Software In/Out	Strong drive
34	P0[6]	\CapSense:Sns[4]\	Analog	HiZ analog
35	P0[7]	SW2	Software In/Out	Res pull up

Pin	Port	Name	Type	Drive Mode
36	XRES	XRES	Dedicated	
37	VCCD	VCCD	Power	
38	VSSD	VSSD	Power	
39	VDDD	VDDD	Power	
40	VDDA	VDDA	Power	
41	VSSA	VSSA	Power	
42	P1[0]	Out_PWM3	Dgtl Out	Strong drive
43	P1[1]	GPIO [unused]		
44	P1[2]	Out_PWM1	Dgtl Out	Strong drive
45	P1[3]	\CapSense:Tx\	Analog	HiZ analog
46	P1[4]	\CapSense:Rx[0]\	Analog	HiZ analog
47	P1[5]	\CapSense:Rx[1]\	Analog	HiZ analog
48	P1[6]	\CapSense:Rx[2]\	Analog	HiZ analog

Abbreviations used in Table 3 have the following meanings:

- Res pull up = Resistive pull up
- Dgtl Out = Digital Output
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- HiZ analog = High impedance analog

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]	28	\CapSense:Sns[0]\	Analog	HiZ analog
P0[1]	29	\CapSense:Sns[1]\	Analog	HiZ analog
P0[2]	30	\CapSense:Sns[2]\	Analog	HiZ analog
P0[3]	31	\CapSense:Sns[3]\	Analog	HiZ analog
P0[4]	32	DI1	Software In/Out	Res pull up
P0[5]	33	DO1	Software In/Out	Strong drive
P0[6]	34	\CapSense:Sns[4]\	Analog	HiZ analog
P0[7]	35	SW2	Software In/Out	Res pull up
P1[0]	42	Out_PWM3	Dgtl Out	Strong drive
P1[1]	43	GPIO [unused]		
P1[2]	44	Out_PWM1	Dgtl Out	Strong drive
P1[3]	45	\CapSense:Tx\	Analog	HiZ analog
P1[4]	46	\CapSense:Rx[0]\	Analog	HiZ analog
P1[5]	47	\CapSense:Rx[1]\	Analog	HiZ analog
P1[6]	48	\CapSense:Rx[2]\	Analog	HiZ analog
P1[7]/VREF	1	DI2	Software In/Out	Res pull up
P2[0]	2	LED4	Software In/Out	Strong drive
P2[1]	3	LED5	Software In/Out	Strong drive
P2[2]	4	LED6	Software In/Out	Strong drive
P2[3]	5	LED7	Software In/Out	Strong drive
P2[4]	6	LED8	Software In/Out	Strong drive
P2[5]	7	LED1	Software In/Out	Strong drive
P2[6]	8	Out_PWM2	Dgtl Out	Strong drive
P2[7]	9	DI3	Software In/Out	Res pull up
P3[0]	12	\UART:rx\	Dgtl In	HiZ digital
P3[1]	13	\UART:tx\	Dgtl Out	Strong drive
P3[2]	14	Debug:SWD_IO	Reserved	
P3[3]	16	Debug:SWD_CK	Reserved	
P3[4]	17	LED9	Software In/Out	Strong drive
P3[5]	18	LED10	Software In/Out	Strong drive
P3[6]	19	LED11	Software In/Out	Strong drive
P3[7]	20	DO2	Software In/Out	Strong drive

Port	Pin	Name	Type	Drive Mode
P4[0]	22	DO3	Software In/Out	Strong drive
P4[1]	23	\CapSense:Cmod\	Analog	HiZ analog
P4[2]	24	\CapSense:CintA\	Analog	HiZ analog
P4[3]	25	\CapSense:CintB\	Analog	HiZ analog

Abbreviations used in Table 4 have the following meanings:

- HiZ analog = High impedance analog
- Res pull up = Resistive pull up
- Dgtl Out = Digital Output
- Dgtl In = Digital Input
- HiZ digital = High impedance digital

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:CintA\	P4[2]	Analog
\CapSense:CintB\	P4[3]	Analog
\CapSense:Cmod\	P4[1]	Analog
\CapSense:Rx[0]\	P1[4]	Analog
\CapSense:Rx[1]\	P1[5]	Analog
\CapSense:Rx[2]\	P1[6]	Analog
\CapSense:Sns[0]\	P0[0]	Analog
\CapSense:Sns[1]\	P0[1]	Analog
\CapSense:Sns[2]\	P0[2]	Analog
\CapSense:Sns[3]\	P0[3]	Analog
\CapSense:Sns[4]\	P0[6]	Analog
\CapSense:Tx\	P1[3]	Analog
\UART:rx\	P3[0]	Dgtl In
\UART:tx\	P3[1]	Dgtl Out
Debug:SWD_CK	P3[3]	Reserved
Debug:SWD_IO	P3[2]	Reserved
DI1	P0[4]	Software In/Out
DI2	P1[7]/VREF	Software In/Out
DI3	P2[7]	Software In/Out
DO1	P0[5]	Software In/Out
DO2	P3[7]	Software In/Out
DO3	P4[0]	Software In/Out
GPIO [unused]	P1[1]	
LED1	P2[5]	Software In/Out
LED10	P3[5]	Software In/Out
LED11	P3[6]	Software In/Out
LED4	P2[0]	Software In/Out
LED5	P2[1]	Software In/Out
LED6	P2[2]	Software In/Out
LED7	P2[3]	Software In/Out
LED8	P2[4]	Software In/Out
LED9	P3[4]	Software In/Out

Name	Port	Type
Out_PWM1	P1[2]	Dgtl Out
Out_PWM2	P2[6]	Dgtl Out
Out_PWM3	P1[0]	Dgtl Out
SW2	P0[7]	Software In/Out

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	Disallowed
Heap Size (bytes)	0x80
Stack Size (bytes)	0x0400
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 7. System Debug Settings

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

3.3 System Operating Conditions

Table 8. System Operating Conditions

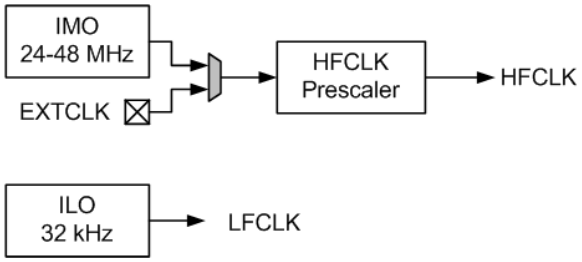
Name	Value
VDDA (V)	3.3
VDDD (V)	3.3
Variable VDDA	True

4 Clocks

The clock system includes these clock resources:

- Two internal clock sources:
  - 24 to 48 MHz Internal Main Oscillator (IMO) ±2% at all frequencies with trim
  - 32 kHz Internal Low Speed Oscillator (ILO)
- External clock (EXTCLK) generated using a signal from an I/O pin
- High-frequency clock (HFCLK) of up to 48 MHz selected from IMO or external clock
- Dedicated prescaler for HFCLK
- Low-frequency clock (LFCLK sourced by ILO
  - Dedicated prescaler for system clock (SYSCLK) of up to 48 MHz sourced by HFCLK
- 24 to 48 MHz Internal Main Oscillator (IMO) ±2%

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
SysClk	NONE	HFClk	? MHz	24 MHz	±2	True	True
IMO	NONE		24 MHz	24 MHz	±2	True	True
HFClk	NONE	IMO	24 MHz	24 MHz	±2	True	True
LFClk	NONE	ILO	? MHz	40 kHz	-50,+100	True	True
Timer_Sel	NONE	ILO	40 kHz	40 kHz	-50,+100	True	True
ILO	NONE		40 kHz	40 kHz	-50,+100	True	True
ExtClk	NONE		16 MHz	? MHz	±0	False	False
Timer (WDT)	NONE	LFClk	? MHz	? MHz	±0	False	False
WCO	NONE		32.768 kHz	? MHz	±0.015	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

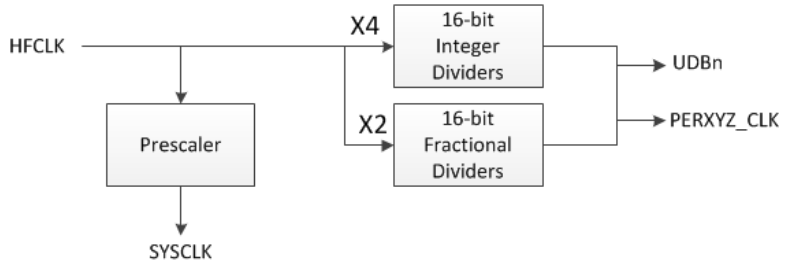


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
Clock3	FIXED_-FUNCT-ION	HFClk	12 MHz	12 MHz	±2	True	True
Clock1	FIXED_-FUNCT-ION	HFClk	12 MHz	12 MHz	±2	True	True
Clock2	FIXED_-FUNCT-ION	HFClk	12 MHz	12 MHz	±2	True	True
UART_SCBCLK	FIXED_-FUNCT-ION	HFClk	1.382 MHz	1.412 MHz	±2	True	True

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
CapSense_-ModClk	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
  - 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	Intr Num	Vector	Priority
CapSense_ISR	10	10	3

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	0x7FFF	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:

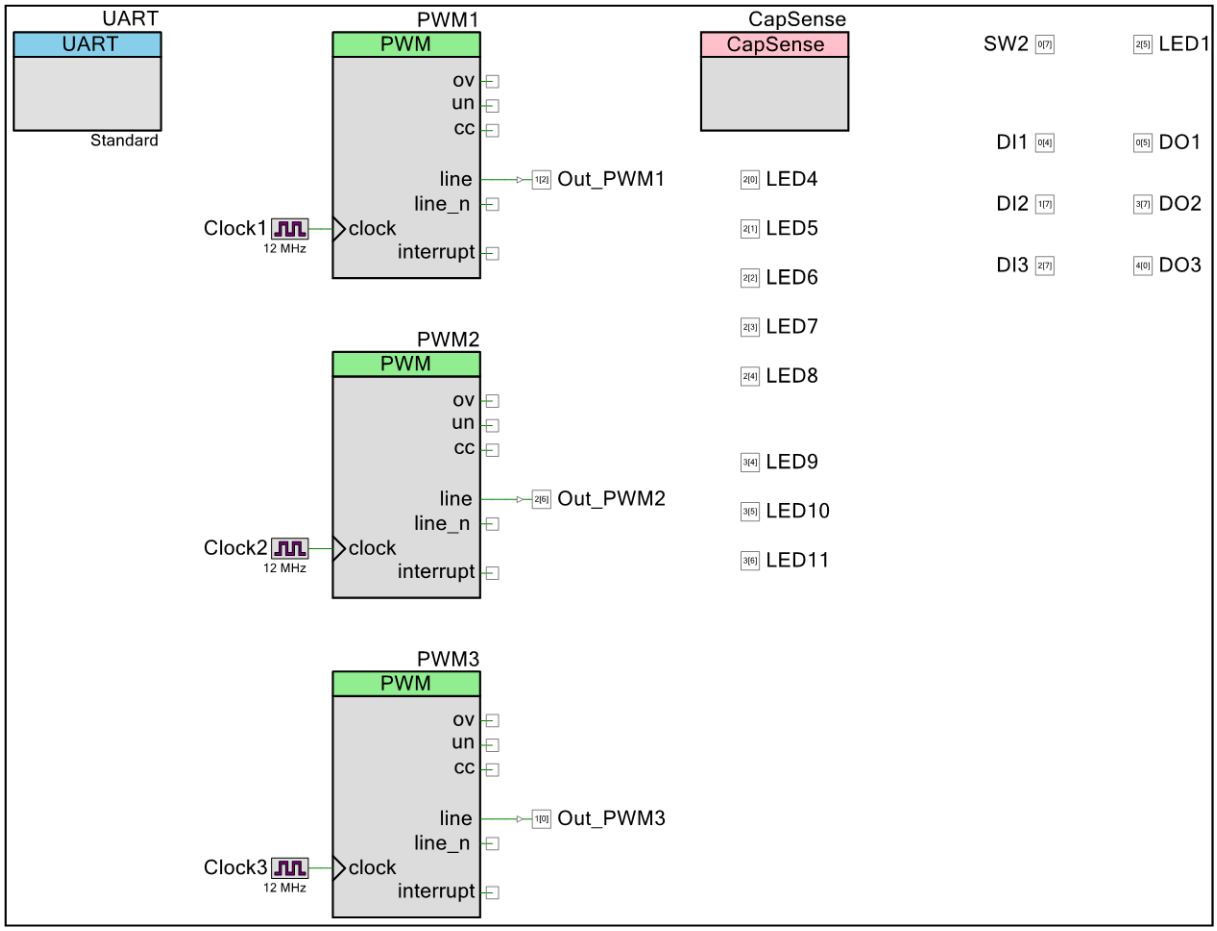
- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CySysFlash API routines

7 Design Contents

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

7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1



This schematic sheet contains the following component instances:

- Instance CapSense (type: CapSense_P4_v7_10)
- Instance PWM1 (type: TCPWM_P4_v2_10)
- Instance PWM2 (type: TCPWM_P4_v2_10)
- Instance PWM3 (type: TCPWM_P4_v2_10)
- Instance UART (type: SCB_P4_v4_0)

8 Components

8.1 Component type: CapSense_P4 [v7.10]

8.1.1 Instance CapSense

Description: (custom component)
Instance type: CapSense_P4 [v7.10]
Datasheet: online component datasheet for CapSense_P4

Table 13. Component Parameters for CapSense

Parameter Name	Value	Description
Ballistic Enable	false	Enables the Ballistic filter for the component.
BaselineType	IIR	Selects the type of baseline needed for design. IIR (default) - Selects the IIR filter based baseline algorithm. CY (Bucket) Baseline - Selects Cypress' "bucket" method for the baseline algorithm.
BlockOffAfterScanEnable	false	Enable the turning-off block after a scan to save additional power. Disabled (default) - The CSD block will be always turned ON. This allows the other components (IDAC) work along with CapSense component in a project. Enabled - The CSD block will be turned ON only during a scan.
Centroid4PtsEnable	false	Enables the 4-point method of a maxima finding for single dimension sliders.
Csd0IdacGainV2	High (2400 nA/bit)	Selects the IDAC gain setting for CSD sensing on CSD block 0. Applicable only for CSDv2 IP.
Csd0PinAlias	LinearSlider_Sns0, LinearSlider_Sns1, LinearSlider_Sns2, LinearSlider_Sns3, LinearSlider_Sns4	Contains a comma-separated list of the electrode aliases for CSD widgets on CSD block 0. Used by the Sns/Sns0 pins on the component schematic.
Csd0PinCount	5	Contains the total count of the CSD electrodes on CSD block 0. Used by the Sns/Sns0 pins on the component schematic.

Parameter Name	Value	Description
Csd0ShieldEnable	false	A shield electrode is used to reduce the sensor parasitic capacitance, enable water-tolerant CapSense designs, and enhance the detection range for Proximity sensors. When the shield electrode is disabled, all configurable parameters associated with the shield electrode are hidden. Applicable to the first CSD block if CSD2x is enabled.
Csd2xEnable	false	Enabling this parameter allows two sensors to be scanned in two, simultaneously used, CSD hardware blocks. This option is available only on devices supporting two CSD blocks.
CsdAnalogStartupDelayUs	10	Defines delay prior to start of the scan, that is intended to ensure proper initialization of the CSDV2 analog part.
CsdAutoZeroEnable	false	Enables auto-zero prior to fine initialization for the CSD sensing method. Applicable only for CSDv2 IP.
CsdCalibrationError	10	Defines acceptable rawcount range around calibration target in percentage. If target is 85% and calibration error is 10% then the rawcount range is 85% to 95%. The range of valid values is from 1 to 99.
CsdColRowIdacAlignmentEn	true	When enabled, the modulator IDACs for rows and columns are aligned to produce the same sensitivity. Applicable for CSD touchpads and CSD matrix buttons.
CsdCommonSenseClockEnable	false	When selected, all CSD widgets share the same sense clock with the frequency specified in the Sense clock frequency (kHz) parameter. Otherwise, a sense clock frequency can be entered separately for each CSD widget in the Widget Details tab.
CsdDedicatedIdacCompEnable	true	Selects the compensation IDAC implementation method when using CSDv2. Enabled (default) - Use IDACB as the compensation IDAC Disabled - Use LEG2 of IDACA as compensation IDAC Applicable only for CSDv2 IP.

Parameter Name	Value	Description
CsdDualIdacLevel	50	Represents the percentage of contribution by the compensation IDAC when using the Dual IDAC sensing.
CsdFineInitCycles	10	Sets a fine-init time period
CsdIdacAutoCalibrateEnable	true	When enabled, the values of IDACs of CSD widgets are automatically set by the component. It is recommended to select Enable IDAC auto-calibration for robust operation. SmartSense Auto-tuning can be enabled only when Enable IDAC auto-calibration is selected.
CsdIdacAutoGainEnable	true	IDAC calibration algorithm finds the optimal IDAC gain when enabled. When disabled the IDAC gain is fixed and defined by CsdIdacGainIndexDefault parameter.
CsdIdacCompEnable	true	The compensation IDAC is used to compensate for the sensor parasitic capacitance to improve the performance. Enabling the compensation IDAC is recommended unless one IDAC is required for other purpose use in the project.
CsdIdacConfig	IDAC Sourcing	Selects the sensing Config needed in CSD mode. IDAC Sourcing (default) - Select IDAC sourcing sensing configuration (-ve charge transfer) IDAC Sinking - Select the IDAC sinking sensing configuration (+ve charge transfer).

Parameter Name	Value	Description
CsdIdacGainIndexDefault	-1	Defines IDAC gain when IDAC auto-calibration or SmartSense enabled. When set to -1 (default), the gain is chosen as follow: - third generation CapSense: Index = 0 (1200nA/lsb), - forth generation CapSense: Index = 4 (2400nA/lsb), - forth generation (rev2) CapSense: Index = 5 (2400nA/lsb). You can specify IDAC gain index defined by table CapSense_idacGainTable[]. When CsdIdacAutoGainEnable parameter is false the current parameter defines used IDAC gain. When CsdIdacAutoGainEnable parameter is true the current parameter defines the upper IDAC gain and auto-calibration may decrease the gain for better performance.
CsdIdacMin	-1	Defines the minimum IDAC at which IDAC gain is switched to the lower IDAC range by IDAC auto-calibration algorithm. When set to -1 CsdIdacMin is defined as follow: - Compensation IDAC is disabled: CsdIdacMin = RoundUp(CsdRawCount-CalibrationLevel / CsdCalibrationError + 1) - Compensation IDAC is enabled: CsdIdacMin = RoundUp((CsdRawCoun-tCalibrationLevel + 100%)/ CsdCalibrationError + 1) The range of valid values: -1 or from 1 to maximum IDAC.

Parameter Name	Value	Description
CsdInactiveSensorConnection	Ground	Selects the state of the sensor when not being scanned. - Ground (default) - All inactive sensors are connected to Ground. - High-Z - All inactive sensors are floating (not connected to GND or Shield). - Shield - All inactive sensors are connected to Shield. The Shield option is available only if an Enable shield electrode check-box is set. Ground is the recommended option for this parameter when the water tolerance is not required for the design. Selects Shield when the design needs water tolerance or sensor parasitic capacitance reduction in a design.
CsdMFSDividerOffsetF1	1	
CsdMFSDividerOffsetF2	2	
CsdModClockFreq	24000	Selects the modulator clock frequency used for the CSD sensing method. Enters any value between the min and max limits, based on the availability of the clock divider. The higher modulator clock frequency reduces sensor scan time. Therefore, results in lower power and reduces the noise in the raw counts, so recommended to use the highest possible frequency.
CsdNoiseMetricEnable	false	Enables noise metric evaluation for CSD scan. Applicable only for CSDv2 IP.
CsdPrescanSettlingTime	5	Represents the settling time delay (in uS) prior to scan is started.
CsdRawCountCalibrationLevel	85	Represents the rawcount calibration level (percentage) to be used when auto-calibration of CSD widgets is enabled.

Parameter Name	Value	Description
CsdSenseClockSource	Auto	The sense clock is used to sample the input sensor. The Spread Spectrum Clock (SSC) provides a dithering clock source with a center frequency equal to the frequency set in the sense clock frequency parameter. The PRS clock source spreads the clock using pseudo-random sequencer. Direct source disables both SSC and PRS sources and uses a fixed-frequency clock.
CsdSensingMethod	Legacy	
CsdSnsClockConstantR	1000	Represents a series resistance value to be considered while decising on the widget/sensor clock value when auto-tuning is enabled.
CsdTuningMode	Manual	Select the tuning mode for CSD widgets. Three options are available: SmartSense (Full Auto-Tune) - This is the quickest way to tune a design. Mostly all widget parameters are automatically tuned by the component. SmartSense (Hardware parameters only) - Hardware parameters are automatically set by the component, all threshold parameters can be manually set by the user Manual - SmartSense auto-tuning is disabled, all widget parameters must be manually tuned. This setting is applicable only to CSD widgets (CSX widgets always use Manual tuning).
CsdV2AnalogWakeupDelayUs	0	Defines delay in the CapSense_Wakeup() API that is intended to ensure proper initialization of the CSDV2 analog part.

Parameter Name	Value	Description
CsdVrefV2	-1	The reference voltage used for CSDv2 operation, in Volts. The range of valid values is from 1.2V to VDDA - 0.6V, where VDDA is the input voltage of the VDDA pin as set in the Design-Wide Resources (*.cydwr) System editor. When set to -1 (default), the reference voltage value (VREF) depends on VDDA: VDDA < 2.6V: VREF = 1.2V 2.6V <= VDDA < 3.2V: VREF = 1.4769V 3.2V <= VDDA < 4.7V: VREF = 2.0211V VDDA >= 4.7V: VREF = 2.7429V The macro generated by the API customizer reflects the VREFGEN gain register value. Applicable only for CSDv2 IP.
Csx0IdacGainV2	Medium (300 nA/bit)	Selects the IDAC gain setting for CSX sensing on CSD block 0. Applicable only for CSDv2 IP.
Csx1IdacGainV2	Medium (300 nA/bit)	Selects the IDAC gain setting for CSX sensing on CSD block 1. Applicable only for CSDv2 IP. Applicable only if CSX2x is enabled.
CsxAnalogStartupDelayUs	10	Defines delay prior to start of the scan, that is intended to ensure proper initialization of the CSDV2 analog part.
CsxAutoZeroEnable	false	Enables auto-zero prior to fine initialization for the CSX sensing method.
CsxCalibrationError	20	Defines acceptable rawcount range around calibration target in percentage. If target is 40% and calibration error is 10% then the rawcount range is 30% to 50%. Valid value is 1 to 99.
CsxCommonTxClockEnable	false	When selected, all CSX widgets share the same Tx clock with the frequency specified in the Tx clock frequency (kHz) parameter. Otherwise, a Tx clock frequency can be entered separately for each CSX widget in the Widget Details tab.
CsxFineInitCycles	4	Sets a fine-init time period

Parameter Name	Value	Description
CsxIdacAutoCalibrateEnable	true	When enabled, IDACs values for the CSX widgets are automatically set by the component. It is recommended to select Enable IDAC auto-calibration for robust operation.
CsxIdacBitsUsedV2	7	Controls how many of the IDAC bits should be considered for auto-calibration of the CSX widgets. Less bits leads to faster calibration time. Applicable only for CSDv2 IP.
CsxInactiveSensorConnection	Ground	Selects the state of the sensor when not being scanned. - Ground (default) - All inactive sensors are connected to Ground. - High-Z - All inactive sensors are floating.
CsxInitShieldSwitchRes	High	Selects the resistance of switches used to drive the shield electrode when an internal shield drive is used.
CsxInitSwitchRes	Medium	Selects the resistance of switches used for Cint1 and Cint2 initialization.
CsxMaxFingers	1	Indicates the maximum number of reported fingers. If the number of fingers on the touchpad exceeds this number, no finger will be reported.
CsxMaxLocalPeaks	5	The maximum possible number of local maxima for CSX touchpad.
CsxMFSDividerOffsetF1	1	
CsxMFSDividerOffsetF2	2	
CsxModClockFreq	24000	Selects the modulator clock frequency used for the CSX sensing method. Enter any value between the min and max limits based on the availability of the clock divider.
CsxMultiphaseTxEnable	false	Enable/disable the multi-phase scan for CSX.
CsxNoiseMetricEnable	false	Enables the noise metric evaluation for the CSX scan.
CsxPinAliasRx	BTN0_Rx0, BTN1_Rx0, BTN2_-Rx0	Contains a comma-separated list of the Rx electrode aliases for the CSX widgets. Used by the Rx pin on the component schematic. Applicable only if CSX2x is disabled.

Parameter Name	Value	Description
CsxPinAliasTx	BTN0_Tx	Contains a comma-separated list of the Tx electrode aliases for the CSX widgets. Used by the Tx/Tx2x pins on the component schematic.
CsxPinCountRx	3	Contains the total count of the Rx electrodes for the CSX widgets. Used by the Rx pin on the component schematic. Applicable only if CSX2x is disabled.
CsxPinCountTx	1	Contains the total count of the Rx electrodes for the CSX widgets. Used by the Tx/Tx2x pin on the component schematic.
CsxRawCountCalibrationLevel	40	Represents the rawcount calibration level (percentage) to be used when auto-calibration of the CSX widgets is enabled.
CsxScanShieldSwitchRes	Low	Selects the resistance of switches used to drive the shield electrode when an internal shield drive is used.
CsxScanSwitchRes	Low	Selects the resistance of switches used for Cint1 and Cint2 initialization.
CsxSkipAndOversampleNodes	false	Enable/Disable over sampling and scan skip on specific nodes.
CsxTxClockSource	Auto	A Tx clock is used to sample the input sensor. The Spread Spectrum Clock (SSC) provides a dithering clock source with the center frequency equal to the frequency set in the Tx Clock frequency parameter. Direct source disables the SSC source and uses a fixed-frequency clock. Auto is the recommended clock source selection.
CustomDataStructSize	0	0 - indicates no custom parameters are added to "CapSense_dsRam" data structure. Non-zero value adds uint8 array (with size specified by value of this parameter) to global parameters of "CapSense_-dsRam" data structure.
Gesture Enable	false	Defines if the gestures are enabled on the Gestures tab.
Gesture Global Enable	false	Enables the Gesture library for the component.

Parameter Name	Value	Description
ImoFreqOffsetF1	20	Sets the trim offset to define the IMO frequency for the first channel. Valid range [0..63] LSB of this parameter shifts the IMO frequency by 0.25%. The first-channel frequency will be reduced by (0.25 * ImoFreqOffsetF1) percent in relative to the zero-channel frequency in case if the value of the ImoFreqOffsetF1 is greater that or equal to the zero-channel trim (CapSense_-immunity[0u]). Otherwise the first-channel frequency will be increased by (0.25 * ImoFreqOffsetF1) percent in relative to the zero-channel frequency.
ImoFreqOffsetF2	20	Sets the trim offset to define the IMO frequency for the second channel. Valid range [0..63] LSB of this parameter shifts the IMO frequency by 0.25%. The second-channel frequency will be increased by (0.25 * (ImoFreqOffsetF1 + ImoFreqOffsetF2)) percent in relative to the zero-channel frequency in case if the value of the ImoFreqOffsetF1 is greater than zero-channel trim (CapSense_immunity[0u]). The second-channel frequency will be decreased by (0.25 * (ImoFreqOffsetF1 + ImoFreqOffsetF2)) percent in relative to the zero-channel frequency in case if the value of the (255 - ImoFreqOffsetF2) is less than zero-channel trim (CapSense_immunity[0u]). Otherwise the second-channel frequency will be increased by (0.25 * ImoFreqOffsetF2) percent in relative to the zero-channel frequency.
LowBaselineResetSize	8 bits	Represents a low baseline reset size for sensors.
MultiFreqScanEnable	false	Indicates whether multi-frequency scanning is enabled.
MultiFreqScanMethod	Change IMO	

Parameter Name	Value	Description
NumCentroids	1 (Legacy)	Selects a number of centroid supported on sliders. The available options are 1, 2 or 3. The default is 1 (Legacy). Applicable only to Radial and Linear slider widgets. Not supported on diplexed sliders.
OffDebounceEnable	false	Indicates whether the debounce for ON to OFF transition is enabled.
PosIirFilterCoeff	128	The centroid Position IIR filter coefficient for sliders and touchpads. The range of valid values is 1-255.
ProxAverageFilterEnable	false	The finite impulse response filter (no feedback) with equally weighted coefficients. It takes four of most recent samples and computes their average. Eliminates the periodic noise (e.g. noise from AC remains). Applicable only to proximity widgets.
ProxCustomFilterEnable	false	Enables the custom filter. Applicable only to proximity widgets.
ProxIirFilterBaselineN	1	Baseline IIR filter coefficient selection for sensors in proximity widgets. The range of valid values is 1-255.
ProxIirFilterBaselineType	Performance	Applicable only to proximity widgets.
ProxIirFilterEnable	false	Enables the infinite-impulse response filter with a step response similar to an RC low-pass filter thereby passing the low frequency signals (finger touch responses). Applicable only to proximity widgets.
ProxMedianFilterEnable	false	Enables a non-linear filter that takes three of most recent samples and computes the median value. This filter eliminates spikes noise typically caused by motors and switching power supplies. Applicable only to proximity widgets.

Parameter Name	Value	Description
RadialSliderPosIirResetThr	35	Configures reset threshold of position IIR filter for Radial slider widget. When difference between between input position and filter history is bigger than the threshold then the filter history is reset with input position. Valid range [25..50] in terms of maximum position percentage.
RegularAverageFilterEnable	false	The finite-impulse response filter (no feedback) with equally weighted coefficients. It takes four of most recent samples and computes their average. Eliminates the periodic noise (e.g. noise from AC remains). Applicable only to regular (non-proximity) widgets.
RegularCustomFilterEnable	false	Enables the custom filter. Applicable only to regular (non-proximity) widgets.
RegularIirFilterBaselineN	1	Baseline IIR filter coefficient selection for sensors in non-proximity widgets. The range of valid values is 1-255.
RegularIirFilterBaselineType	Performance	
RegularIirFilterEnable	false	Enables the infinite-impulse response filter with a step response similar to an RC low-pass filter thereby passing low frequency signals (finger touch responses). Applicable only to regular (non-proximity) widgets.
RegularMedianFilterEnable	false	Enables a non-linear filter that takes three of most recent samples and computes the median value. This filter eliminates spikes noise typically caused by motors and switching power supplies. Applicable only to regular (non-proximity) widgets.
SecondFinger5x5FilterEnable	false	Enables position filtering of the second touch. Applicable only to CSD touchpad widgets with 5x5 centroid and two finger detection enabled.

Parameter Name	Value	Description
SelfTestEnable	false	The BIST/Class-B library supports the following: the sensor short test, test baseline and raw count limits, CRC for widget-specific register map data, measuring external cap (Cmod, Csh_tank, CintA and CintB) and sensor's and shield's cap values and test baseline data consistency. Additionally, measuring of VDDA and two internal reference caps are supported for CSDv2.
SensorAutoResetEnable	false	When enabled, the baseline is always updated and when disabled, the baseline is updated only when the difference between the baseline and raw count is less than the noise threshold. The sensor auto-reset prevents the sensor from permanently turning on when the raw count accidentally rises because of a large power-supply voltage fluctuation or due to other spurious conditions.
SliderMultiplierMethod	MaxPos / (SnsNum -1)	Defines the method of multiplier calculation for linear slider widget
ThresholdSize	16 bits	Selects a data size for widgets in the component. This applies to Finger Threshold (all widgets) and Proximity Touch Threshold (proximity widgets). In SmartSense (Full Auto-tune) mode, parameter value is ignored and threshold register size is always 16-bit.
Timestamp Interval	1	Defines the increment value for the timestamp register.
TouchpadDisplaySettings		
TouchpadMultiplierMethod	MaxPos / (SnsNum -1)	Defines the method of multiplier calculation for touchpad widget (is not applicable for CSD 5x5 touchpad)
TouchProxThresholdCoeff	300	Sets coefficient to define touch threshold for proximity sensors
TunerLayoutPreferences		
TunerOptionsPreferences		
TunerViewPreferences		
TunerWidgetData		

Parameter Name	Value	Description
Two-finger Settling time (ms)	3	This parameter defines a delay threshold that must be met before two finger gestures are computed. This parameter helps to avoid instances where a two-finger gesture is reported when two fingers are placed on the panel one after the other.
User Comments		Instance-specific comments.
WidgetBaselineCoeffEnable	false	Enables setting of baseline coefficient separately for each widget.

8.2 Component type: SCB_P4 [v4.0]

8.2.1 Instance UART

Description: Serial Communication Block (SCB)
Instance type: SCB_P4 [v4.0]
Datasheet: online component datasheet for SCB_P4

Table 14. Component Parameters for UART

Parameter Name	Value	Description
EzI2cByteModeEnable	false	When the SCB mode is EZI2C, this parameter specifies the number of bits per FIFO data element. The byte mode – false: a 16-bit FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8-bit FIFO data element. The FIFO depth is 16 entries. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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).

Parameter Name	Value	Description
EzI2cSecondarySlaveAddress	9	When the SCB mode is EZI2C, this parameter specifies EZI2C secondary 7-bits slave address (MSB ignored). Only applicable when EZI2C clock stretching option is set.
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.
I2C Bus Voltage	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 devices other than PSoC 4000/PSoC 4100/PSoC 4200.
I2C Bus Voltage	3.3	When the SCB mode is EZI2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus. Only applicable for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.

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-bit FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8-bit FIFO data element. The FIFO depth is 16 entries. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.

Parameter Name	Value	Description
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.
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.
Show EZI2C Terminals	false	When the SCB mode is EZI2C, this parameter removes internal pins and expose signals to terminals. The exposed terminals must be connected to the pins.
Show I2C Terminals	false	When the SCB mode is I2C, this parameter removes internal pins and expose signals to terminals. The exposed terminals must be connected to the pins.
Show SPI Terminals	false	When the SCB mode is SPI, this parameter removes internal pins and expose signals to terminals. The exposed terminals must be connected to the pins or SmartIO component.

Parameter Name	Value	Description
Show UART Terminals	false	When the SCB mode is UART, this parameter removes internal pins and expose signals to terminals. The exposed terminals must be connected to the pins or SmartIO component.
Slew Rate	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.
Slew Rate	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.
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.
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-bit FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8-bit FIFO data element. The FIFO depth is 16 entries. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.

Parameter Name	Value	Description
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). Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
SpiSs1Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 1. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
SpiSs2Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 2. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
SpiSs3Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 3. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.
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.

Parameter Name	Value	Description
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. The byte mode – false: a 16-bit FIFO data element. The FIFO depth is 8 entries. The byte mode – true: an 8-bit FIFO data element. The FIFO depth is 16 entries. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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. Only applicable for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
UartCtsPolarity	Active Low	When the SCB mode is UART, this parameter specifies active polarity of an input cts signal. Only applicable for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.
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.

Parameter Name	Value	Description
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.
UartIntrRxBreakDetected	false	This parameter enables the RX break detection interrupt source to trigger the interrupt output.
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. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
UartRtsPolarity	Active Low	When the SCB mode is UART, this parameter specifies active polarity of the output rts signal. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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. Applicable only for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
UartRxBreakWidth	11	This parameter specifies the break width in bits.

Parameter Name	Value	Description
UartRxBufferSize	8	When the SCB mode is UART, this parameter defines the size of the RX buffer.
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.
User Comments		Instance-specific comments.

8.3 Component type: TCPWM_P4 [v2.10]

8.3.1 Instance PWM1

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

Table 15. Component Parameters for PWM1

Parameter Name	Value	Description
PWMCompare	65535	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	Terminal count mask	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

Parameter Name	Value	Description
PWMRunMode	Continuous	Selects between continuous and one shot run mode for the PWM
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

Parameter Name	Value	Description
QuadStopMode	Rising edge	Determines whether the Quadrature Decoder stop signal is accepted at level detection or in various modes of edge detection
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

Parameter Name	Value	Description
TCPWMCountPresent	false	Determines whether the Unconfigured count signal is present and controls its pin visibility
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
User Comments		Instance-specific comments.

8.3.2 Instance PWM2

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 PWM2

Parameter Name	Value	Description
PWMCompare	65535	The initial value for the comparison register when in the PWM mode

Parameter Name	Value	Description
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	Terminal count mask	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
PWMSetAlign	Left align	Selects the alignment of the PWM waveform to be either left, right, center or asymmetrically aligned

Parameter Name	Value	Description
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
QuadStopPresent	false	Determines whether the Quadrature Decoder stop signal is present and controls its pin visibility

Parameter Name	Value	Description
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
TCPWMReloadPresent	false	Determines whether the Unconfigured reload signal is present and controls its pin visibility

Parameter Name	Value	Description
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
User Comments		Instance-specific comments.

8.3.3 Instance PWM3

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

Table 17. Component Parameters for PWM3

Parameter Name	Value	Description
PWMCompare	65535	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

Parameter Name	Value	Description
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	Terminal count mask	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
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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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
User Comments		Instance-specific comments.

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