PSoC® Creator™
Project Datasheet for PSoC_PyunPyun

Creation Time: 03/12/2015 17:20:38
User: ATCS\gizmo
Project: PSoC_PyunPyun
Tool: PSoC Creator 3.1 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 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 Device Family Block Diagram

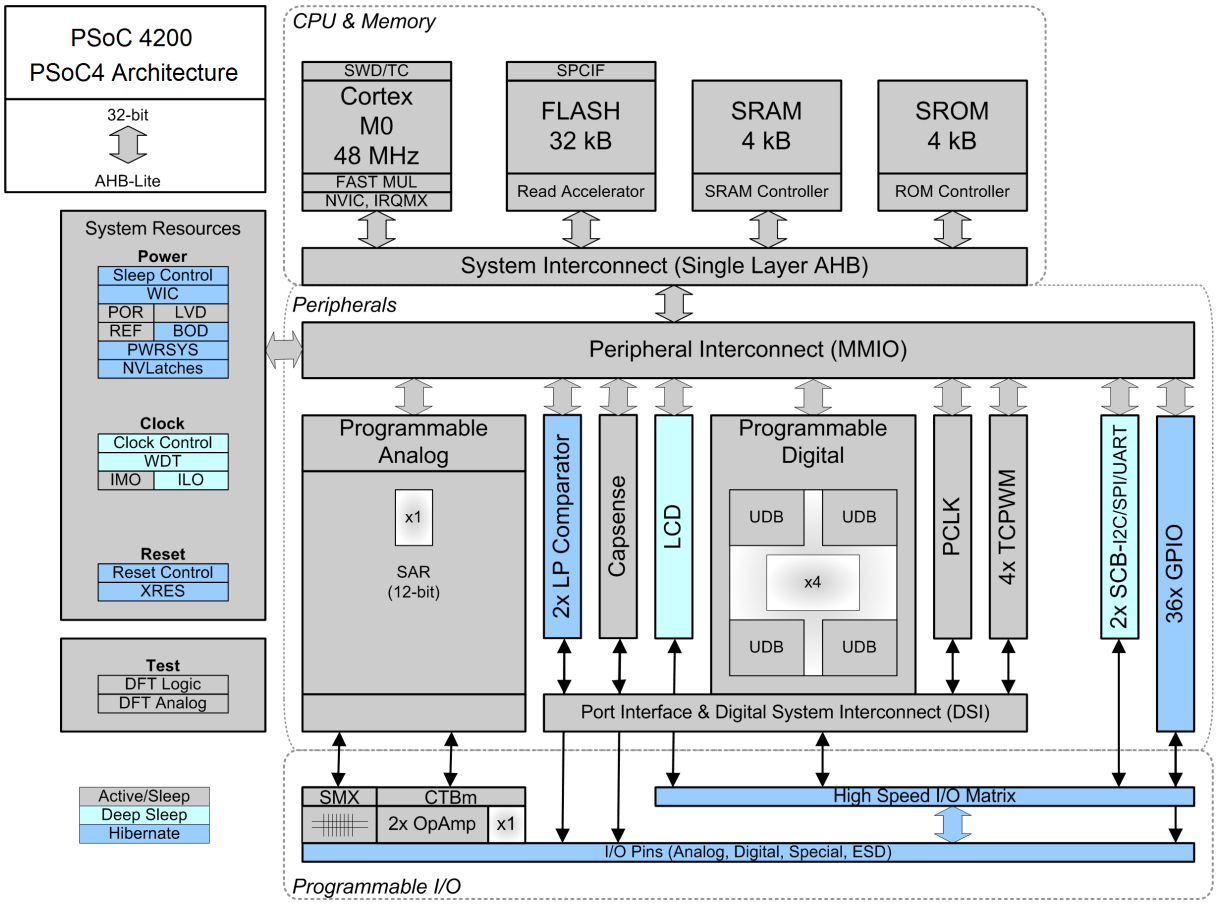


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C4245AXI-483
Package Name	44-TQFP
Architecture	PSoC 4
Family	PSoC 4200
CPU speed (MHz)	48
Flash size (kBytes)	32
SRAM size (kBytes)	4
Vdd range (V)	1.71 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:

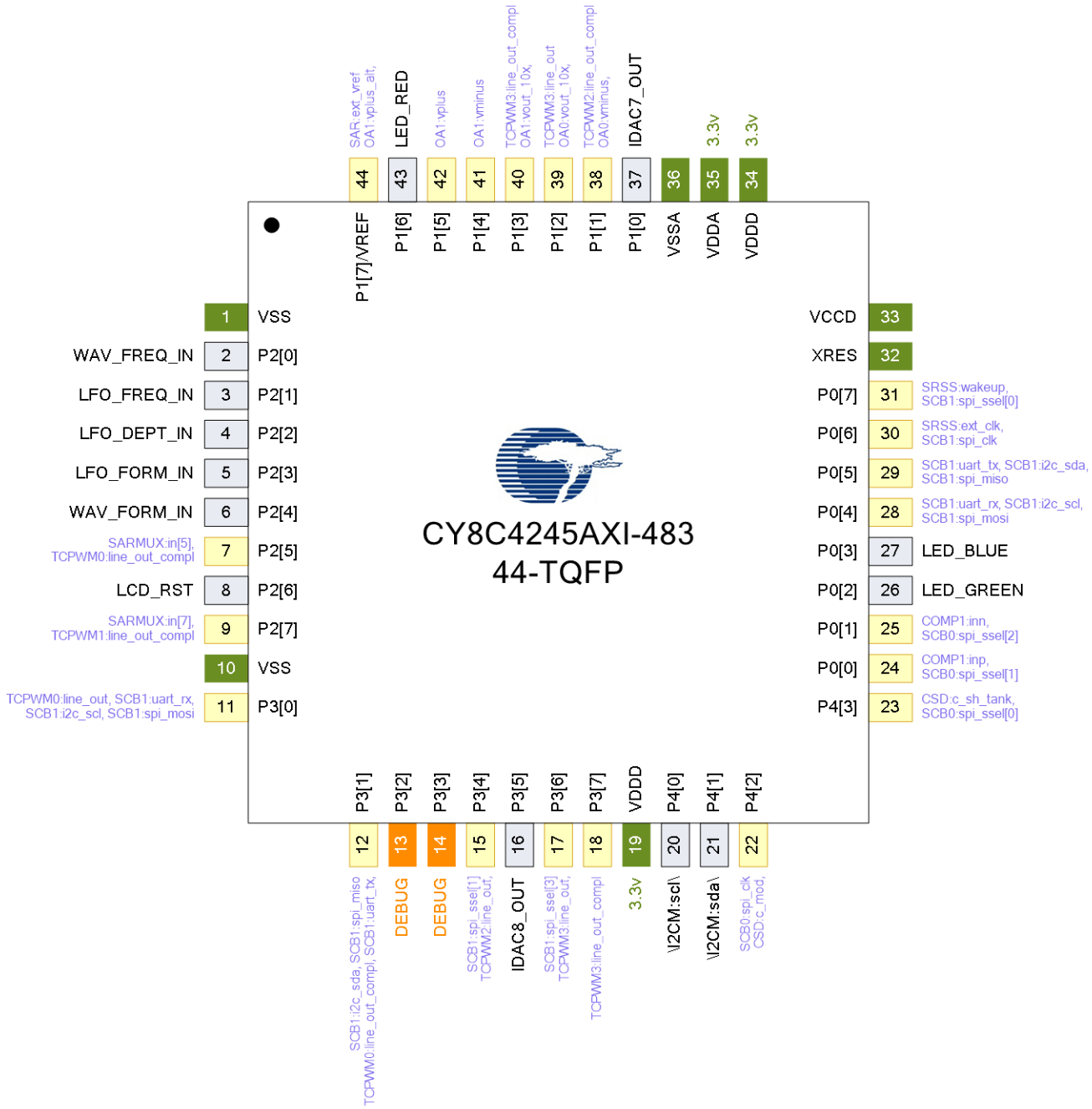
Table 2. Device Resources

Name	In Use	Free	Total Resources Available	% in Use
Digital clock dividers	2	2	4	50.0%
Pins	15	21	36	41.7%
UDB Macrocells	7	25	32	21.9%
UDB Unique Pterms	6	58	64	9.4%
UDB Datapath Cells	0	4	4	0.0%
UDB Status Cells	0	4	4	0.0%
UDB Control Cells	0	4	4	0.0%
Interrupts	5	27	32	15.6%
Comparator/Opamp Fixed Blocks	0	2	2	0.0%
SAR Fixed Blocks	1	0	1	100.0%
CSD Fixed Blocks	0	1	1	0.0%
CapSense Blocks	0	1	1	0.0%
8-bit CapSense IDACs	1	0	1	100.0%
7-bit CapSense IDACs	1	0	1	100.0%
Temperature Sensors	0	1	1	0.0%
Low Power Comparators	0	2	2	0.0%
TCPWM Blocks	1	3	4	25.0%
Serial Communication Blocks	1	1	2	50.0%
Segment LCD Blocks	0	1	1	0.0%

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	VSS	VSS	Power	
2	P2[0]	WAV_FREQ_IN	Analog	HiZ analog
3	P2[1]	LFO_FREQ_IN	Analog	HiZ analog
4	P2[2]	LFO_DEPT_IN	Analog	HiZ analog
5	P2[3]	LFO_FORM_IN	Dgtl In	Res pull up
6	P2[4]	WAV_FORM_IN	Dgtl In	Res pull up
7	P2[5]	GPIO [unused]		
8	P2[6]	LCD_RST	Software Output	Strong drive
9	P2[7]	GPIO [unused]		
10	VSS	VSS	Power	
11	P3[0]	GPIO [unused]		
12	P3[1]	GPIO [unused]		
13	P3[2]	Debug:SWD_IO	Reserved	
14	P3[3]	Debug:SWD_CK	Reserved	
15	P3[4]	GPIO [unused]		
16	P3[5]	IDAC8_OUT	A/D Out	HiZ analog
17	P3[6]	GPIO [unused]		
18	P3[7]	GPIO [unused]		
19	VDDD	VDDD	Power	
20	P4[0]	\I2CM:scl\	Dgtl In	OD, DL
21	P4[1]	\I2CM:sda\	Dgtl In	OD, DL
22	P4[2]	GPIO [unused]		
23	P4[3]	GPIO [unused]		
24	P0[0]	GPIO [unused]		
25	P0[1]	GPIO [unused]		
26	P0[2]	LED_GREEN	Software Output	Strong drive
27	P0[3]	LED_BLUE	Software Output	Strong drive
28	P0[4]	GPIO [unused]		
29	P0[5]	GPIO [unused]		
30	P0[6]	GPIO [unused]		
31	P0[7]	GPIO [unused]		
32	XRES	XRES	Dedicated	
33	VCCD	VCCD	Power	
34	VDDD	VDDD	Power	
35	VDDA	VDDA	Power	
36	VSSA	VSSA	Power	
37	P1[0]	IDAC7_OUT	A/D Out	HiZ analog
38	P1[1]	GPIO [unused]		
39	P1[2]	GPIO [unused]		
40	P1[3]	GPIO [unused]		
41	P1[4]	GPIO [unused]		
42	P1[5]	GPIO [unused]		

Pin	Port	Name	Type	Drive Mode
43	P1[6]	LED_RED	Software Output	Strong drive
44	P1[7]/VREF	GPIO [unused]		

Abbreviations used in Table 3 have the following meanings:

- HiZ analog = High impedance analog
- Dgtl In = Digital Input
- Res pull up = Resistive pull up
- A/D Out = Analog / 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]	24	GPIO [unused]		
P0[1]	25	GPIO [unused]		
P0[2]	26	LED_GREEN	Software Output	Strong drive
P0[3]	27	LED_BLUE	Software Output	Strong drive
P0[4]	28	GPIO [unused]		
P0[5]	29	GPIO [unused]		
P0[6]	30	GPIO [unused]		
P0[7]	31	GPIO [unused]		
P1[0]	37	IDAC7_OUT	A/D Out	HiZ analog
P1[1]	38	GPIO [unused]		
P1[2]	39	GPIO [unused]		
P1[3]	40	GPIO [unused]		
P1[4]	41	GPIO [unused]		
P1[5]	42	GPIO [unused]		
P1[6]	43	LED_RED	Software Output	Strong drive
P1[7]/VREF	44	GPIO [unused]		
P2[0]	2	WAV_FREQ_IN	Analog	HiZ analog
P2[1]	3	LFO_FREQ_IN	Analog	HiZ analog
P2[2]	4	LFO_DEPT_IN	Analog	HiZ analog
P2[3]	5	LFO_FORM_IN	Dgtl In	Res pull up
P2[4]	6	WAV_FORM_IN	Dgtl In	Res pull up
P2[5]	7	GPIO [unused]		
P2[6]	8	LCD_RST	Software Output	Strong drive
P2[7]	9	GPIO [unused]		
P3[0]	11	GPIO [unused]		
P3[1]	12	GPIO [unused]		
P3[2]	13	Debug:SWD_IO	Reserved	
P3[3]	14	Debug:SWD_CK	Reserved	
P3[4]	15	GPIO [unused]		
P3[5]	16	IDAC8_OUT	A/D Out	HiZ analog
P3[6]	17	GPIO [unused]		
P3[7]	18	GPIO [unused]		
P4[0]	20	\I2CM:scl\	Dgtl In	OD, DL
P4[1]	21	\I2CM:sda\	Dgtl In	OD, DL
P4[2]	22	GPIO [unused]		
P4[3]	23	GPIO [unused]		

Abbreviations used in Table 4 have the following meanings:

- A/D Out = Analog / Digital Output
- HiZ analog = High impedance analog
- Dgtl In = Digital Input
- Res pull up = Resistive pull up

- 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
\I2CM:scl\	P4[0]	Dgtl In
\I2CM:sda\	P4[1]	Dgtl In
Debug:SWD_CK	P3[3]	Reserved
Debug:SWD_IO	P3[2]	Reserved
IDAC7_OUT	P1[0]	A/D Out
IDAC8_OUT	P3[5]	A/D Out
LCD_RST	P2[6]	Software Output
LED_BLUE	P0[3]	Software Output
LED_GREEN	P0[2]	Software Output
LED_RED	P1[6]	Software Output
LFO_DEPT_IN	P2[2]	Analog
LFO_FORM_IN	P2[3]	Dgtl In
LFO_FREQ_IN	P2[1]	Analog
WAV_FORM_IN	P2[4]	Dgtl In
WAV_FREQ_IN	P2[0]	Analog

Abbreviations used in Table 5 have the following meanings:

- Dgtl In = Digital Input
- A/D Out = Analog / 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)	0x80
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

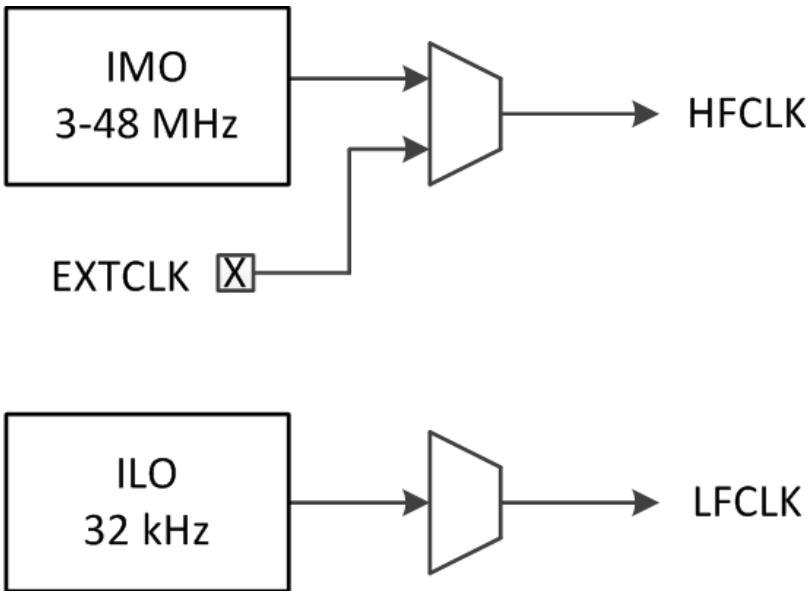
Name	Value
VDDA (V)	3.3
Variable VDDA	True
VDDD (V)	3.3

4 Clocks

The clock system includes these clock resources:

- Two internal clock sources:
  - 3 to 48 MHz Internal Main Oscillator (IMO) ±2% at 3 MHz
  - 32 kHz Internal Low Speed 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 (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
SYSCLK	NONE	HFCLK	0	48	±2	True	True
IMO	NONE		48	48	±2	True	True
HFCLK	NONE	Direct_Sel	48	48	±2	True	True
Direct_Sel	NONE	IMO	48	48	±2	True	True
PLL_Sel	NONE	IMO	48	48	±2	True	True
DBL_Sel	NONE	IMO	48	48	±2	True	True
DPLL_Sel	NONE	IMO	48	48	±2	True	True
LFCLK	NONE	ILO	0	0.032	±60	True	True
ILO	NONE		0.032	0.032	±60	True	True
EXTCLK	NONE		24	0	±0	False	False
DigSig3	NONE		0	0	±0	False	False
DigSig4	NONE		0	0	±0	False	False
DigSig2	NONE		0	0	±0	False	False
DigSig1	NONE		0	0	±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

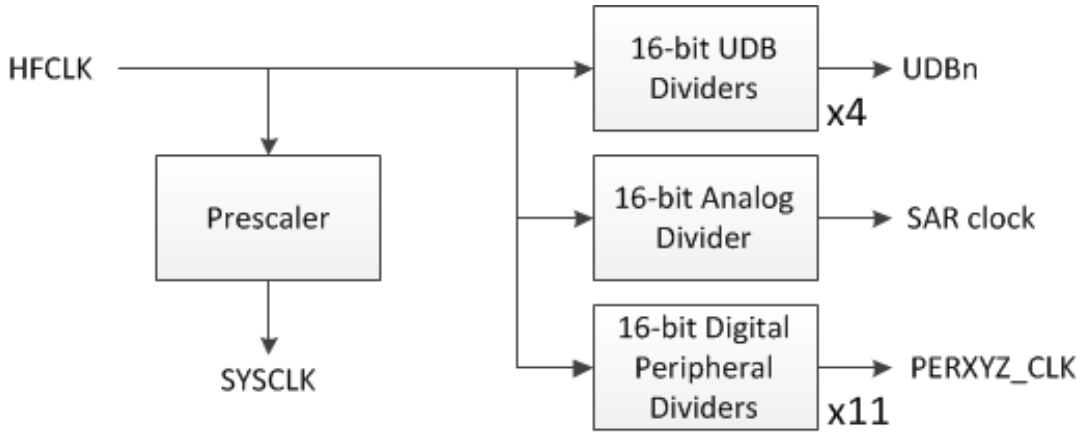


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
SamplingClock	FIXED_-FUNCT-ION	HFCLK	12	12	±2	True	True

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
I2CM_SCBCLK	FIXED_-FUNCT-ION	HFCLK	1.55	1.6	±2	True	True
ADC_SAR_S-EQ_intClock	FIXED_-FUNCT-ION	HFCLK	1	1	±2	True	True
LFO_FORM_-CLK	DIGITAL	HFCLK	0.0002	0.0002	±2	True	True
WAV_FORM_-CLK	DIGITAL	HFCLK	0.0002	0.0002	±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
  - o CySysClkImo API routines
  - o CySysClkIlo API routines
  - o 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
ADC_SAR_SEQ_IRQ	3	14
I2CM_SCB_IRQ	3	10
LFO_FORM_ISR	3	0
TimerISR	3	16
WAV_FORM_ISR	3	2

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:

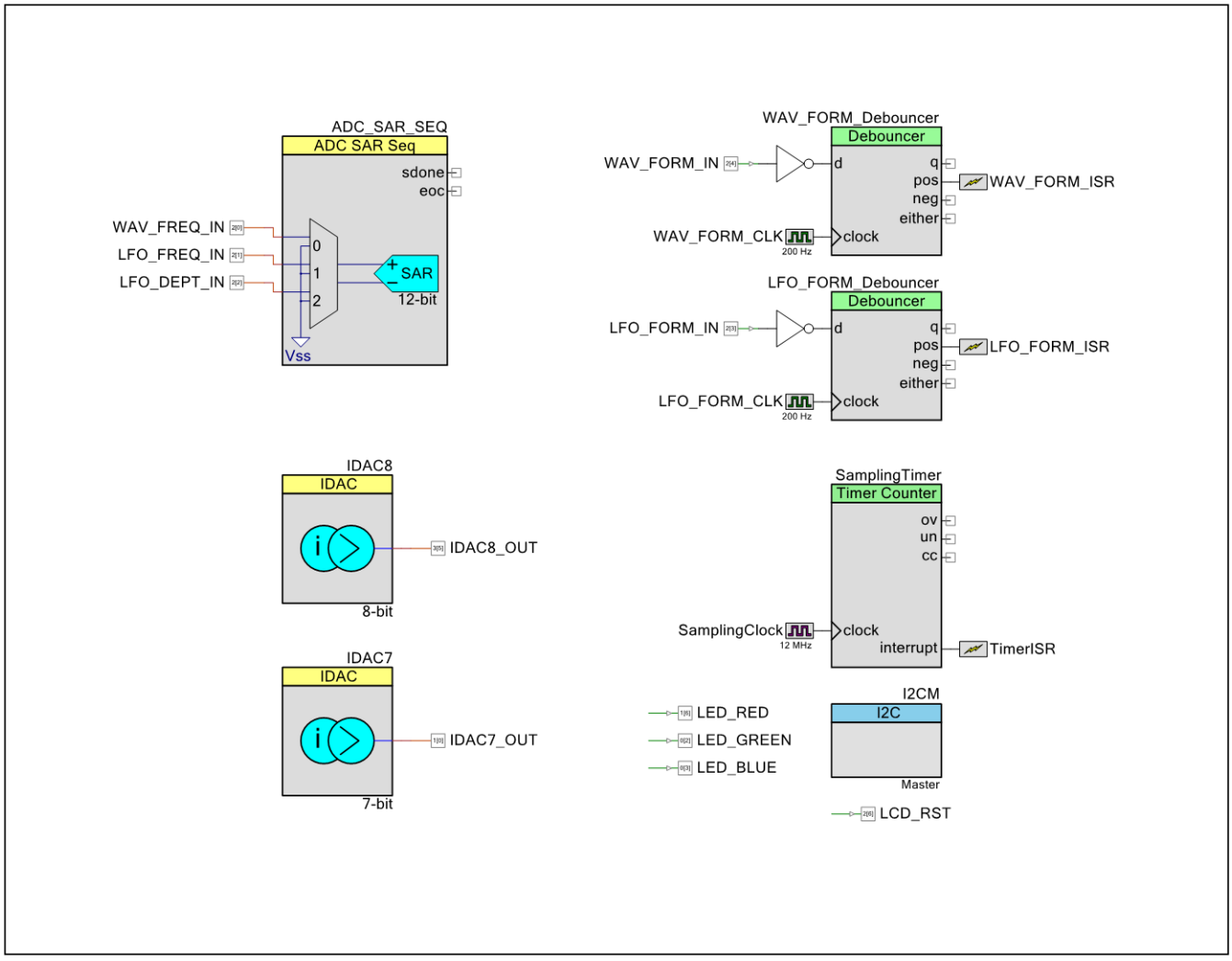
- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CySysFlash API routines

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 ADC_SAR_SEQ (type: ADC_SAR_SEQ_P4_v2_0)
- Instance I2CM (type: SCB_P4_v2_0)
- Instance IDAC7 (type: IDAC_P4_v1_0)
- Instance IDAC8 (type: IDAC_P4_v1_0)
- Instance LFO_FORM_Debouncer (type: Debouncer_v1_0)
- Instance SamplingTimer (type: TCPWM_P4_v2_0)
- Instance WAV_FORM_Debouncer (type: Debouncer_v1_0)

8 Components

8.1 Component type: ADC_SAR_SEQ_P4 [v2.0]

8.1.1 Instance ADC_SAR_SEQ

Description: PSoC 4 Sequencing Successive Approximation ADC
Instance type: ADC_SAR_SEQ_P4 [v2.0]
Datasheet: online component datasheet for ADC_SAR_SEQ_P4

Table 13. Component Parameters for ADC_SAR_SEQ

Parameter Name	Value	Description
AdcAClock	2	Acquisition time in clock cycles for configuration A.
AdcAdjust	ClockFreq	Timing parameter adjustable by the user.
AdcAlternateResolution	8	This parameter sets the alternate ADC resolution to either 8 or 10 bits.
AdcAvgMode	Fixed Resolution	This parameter sets how the averaging mode operates.
AdcAvgSamplesNum	8	This parameter sets the averaging rate for any channel that has its averaging option enabled.
AdcBClock	4	Acquisition time in clock cycles for configuration B.
AdcCClock	128	Acquisition time in clock cycles for configuration C.
AdcChannelsEnConf	7	This bitmask is intended to enable the channels for scanning during runtime.
AdcChannelsModeConf	0	Mode configuration for the channels (0 - Single, 1 - Differential)
AdcClock	Internal	Clock source type.
AdcClockFrequency	1000000	Specifies the internal clock frequency in Hz.
AdcCompareMode	Low_Limit <= Result < High_Limit	This parameter sets the condition in which the limit condition will occur.
AdcDataFormatJustification	Right	This parameter sets whether the output data is left or right justified for a 16-bit word. For signed values, the result will be sign extended when configured in right justification mode.
AdcDClock	4	Acquisition time in clock cycles for configuration D.
AdcDifferentialResultFormat	Signed	This parameter sets the whether the result from a differential measurement is Signed or Unsigned.
AdcHighLimit	4095	This parameter sets the high limit for a limit compare.

Parameter Name	Value	Description
AdcInjChannelEnabled	false	Determines whether the symbol will display the injection channel.
AdcInputBufGain	Disable	Sets the input buffer gain or disables it.
AdcLowLimit	0	This parameter sets the low limit for a limit compare.
AdcMaxResolution	12	Sets the maximum resolution of the ADC in bits.
AdcSampleMode	FreeRunning	Sampling mode.
AdcSarMuxChannelConfig	000	Channels mode configuration for the multiplexer (0 - Single, 1 - Differential)
AdcSequencedChannels	3	Number of input signals that will be scanned. This excludes the injection channel.
AdcSingleEndedNegativeInput	Vss	Negative input source for single ended operation.
AdcSingleResultFormat	Signed	This parameter sets whether the result from a single ended measurement is Signed or Unsigned.
AdcSymbolHasSingleEn-dedInputChannel	false	Determines whether the configuration contains an external negative input.
AdcVrefSelect	VDDA	The reference voltage that is used for the SAR ADC.
AdcVrefVoltage_mV	1024	The reference voltage value.
rm_int	false	Removes the internal interrupt

8.2 Component type: Debouncer [v1.0]

8.2.1 Instance LFO_FORM_Debouncer

Description: Debounces the input digital signal from most types of switches
Instance type: Debouncer [v1.0]
Datasheet: online component datasheet for Debouncer

Table 14. Component Parameters for LFO_FORM_Debouncer

Parameter Name	Value	Description
EitherEdgeDetect	true	Specifies whether the positive or negative edge detection is enabled for the component.
NegEdgeDetect	true	Specifies whether the negative edge detection is enabled for the component.
PosEdgeDetect	true	Specifies whether the positive edge detection is enabled for the component.
SignalWidth	1	Determines the bus width of input and output terminals.

8.2.2 Instance WAV_FORM_Debouncer

Description: Debounces the input digital signal from most types of switches

Instance type: Debouncer [v1.0]
Datasheet: online component datasheet for Debouncer

Table 15. Component Parameters for WAV_FORM_Debouncer

Parameter Name	Value	Description
EitherEdgeDetect	true	Specifies whether the positive or negative edge detection is enabled for the component.
NegEdgeDetect	true	Specifies whether the negative edge detection is enabled for the component.
PosEdgeDetect	true	Specifies whether the positive edge detection is enabled for the component.
SignalWidth	1	Determines the bus width of input and output terminals.

8.3 Component type: IDAC_P4 [v1.0]

8.3.1 Instance IDAC7

Description: 7 or 8-bit Current DAC
Instance type: IDAC_P4 [v1.0]
Datasheet: online component datasheet for IDAC_P4

Table 16. Component Parameters for IDAC7

Parameter Name	Value	Description
IDACRange	High Range	Dynamic range of the IDAC
IDACValue	120	IDAC value
Polarity	Positive (Source)	Selects the Polarity of the output
Resolution	7	Resolution of the IDAC

8.3.2 Instance IDAC8

Description: 7 or 8-bit Current DAC
Instance type: IDAC_P4 [v1.0]
Datasheet: online component datasheet for IDAC_P4

Table 17. Component Parameters for IDAC8

Parameter Name	Value	Description
IDACRange	Low Range	Dynamic range of the IDAC
IDACValue	120	IDAC value
Polarity	Positive (Source)	Selects the Polarity of the output
Resolution	8	Resolution of the IDAC

8.4 Component type: SCB_P4 [v2.0]

8.4.1 Instance I2CM

Description: Serial Communication Block (SCB)
Instance type: SCB_P4 [v2.0]

Datasheet: [online component datasheet for SCB_P4]

Table 18. Component Parameters for I2CM

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.
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 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 available for PSoC 4200-BL 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). 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 I2C pins depends on placement for PSoC 4200-BL devices.
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.

Parameter Name	Value	Description
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 a match I2C slave address in the RX FIFO or not. This option could be used for software address matching.
I2cBusVoltage	3.3	When the SCB mode is I2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus.
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 available for PSoC 4200-BL devices.
I2cClockFromTerm	false	When the SCB mode is I2C, this parameter allows the provision of a clock terminal to connect a clock from 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	false	When the SCB mode is I2C, this parameter specifies the method of calculating the oversampling as manual or automatic.
I2cMode	Master	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.

Parameter Name	Value	Description
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.
I2cSlewRate	Fast	When the SCB mode is I2C, this parameter specifies the slew rate settings of the I2C pins.
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 SPI Bit rate in kbps. The standard bit rates are: 500, 1000-8000 kbps.

Parameter Name	Value	Description
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 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 available for PSoC 4200-BL devices.
SpiClockFromTerm	false	When the SCB mode is SPI, this parameter provides a clock terminal to connect a clock outside the component in SPI mode.
SpiFreeRunningSclk	false	When the SCB mode is SPI, this parameter specifies the SCLK generation by the master as: gated or free running (continuous). Only available for pSoC 4200-BL 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: RX FIFO is full.

Parameter Name	Value	Description
SpiIntrRxNotEmpty	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_RX.NOT_EMPTY interrupt source. SCB.INTR_RX.NOT_EMPTY: 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: 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: 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: 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: 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: 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: TX FIFO is not full. There is at least one entry to put data.

Parameter Name	Value	Description
SpiIntrTxOverflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.OVERFLOW interrupt source. SCB.INTR_TX.OVERFLOW: attempt to write to a full TX FIFO.
SpiIntrTxTrigger	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.TRIGGER interr-upt source. SCB.INTR_TX.TRIGGER: TX FIFO has fewer entries than the value specified by SpiTxTrigge-rLevel.
SpiIntrTxUnderflow	false	When the SCB mode is SPI, this parameter enables the SCB.INTR_TX.UNDERFLOW interrupt source. SCB.INTR_TX.UNDERFLOW: 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.
SpiManualOversampleControl	true	OBSOLETE: this parameter is left for compatibility and it is not used any more.
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.
SpiRxBufferSize	8	When the SCB mode is SPI, this parameter defines the size of the RX buffer.

Parameter Name	Value	Description
SpiRxTriggerLevel	7	When the SCB mode is SPI, this parameter defines the number of entries in the RX FIFO to trigger the SCB.INTR_-RX.TRIGGER interrupt event.
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 available for PSoC 4200-BL devices.
SpiSs1Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 1. Only available for PSoC 4200-BL devices.
SpiSs2Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 2. Only available for PSoC 4200-BL devices.
SpiSs3Polarity	Active Low	When the SCB mode is SPI, this parameter specifies active polarity of slave select 3. Only available for PSoC 4200-BL 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.
SpiTxTriggerLevel	0	When the SCB mode is SPI, this parameter defines the number of entries in TX FIFO to trigger the INTR_TX.TRIGGER interrupt event.
SpiWakeEnable	false	When the SCB mode is SPI, this parameter enables wakeup from Deep Sleep on slave select event.

Parameter Name	Value	Description
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 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 available for PSoC 4200-BL 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. Only available for PSoC 4200-BL devices.
UartCtsPolarity	Active Low	When the SCB mode is UART, this parameter specifies active polarity of an input cts signal. Only available for PSoC 4200-BL devices.
UartDataRate	115200	When the SCB mode is UART, this parameter defines the UART baud rate in kbps. The standard baud rates are provided.
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.

Parameter Name	Value	Description
UartIntrRxFrameErr	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FRAME_-ERROR interrupt source. SCB.INTR_RX.FRAME_E-RROR: 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: 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: 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: at-tempt 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: parity error in received data frame.
UartIntrRxTrigger	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.TRIGGER inter-rupt source. SCB.INTR_RX.TRIGGER: RX FIFO has more entries than the value specified by UartRxTrigg-erLevel.
UartIntrRxUnderflow	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.UNDERFLOW interrupt source. SCB.INTR_RX.UNDERFLOW: -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: TX FIFO is empty.

Parameter Name	Value	Description
UartIntrTxNotFull	false	When the SCB mode is UART, this parameter enables the SCB.INTR_TX.NOT_FULL interrupt source. SCB.INTR_TX.NOT_FULL: 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: 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: 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: all data are sent in to TX FIFO and the transmit FIFO and the shifter register are emptied.
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: 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: 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: attempt to read from an empty TX FIFO.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
UartRtsEnable	false	When the SCB mode is UART, this parameter enables the rts output. Only available for PSoC 4200-BL devices.
UartRtsPolarity	Active Low	When the SCB mode is UART, this parameter specifies active polarity of the output rts signal. Only available for PSoC 4200-BL 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 available for PSoC 4200--BL devices.
UartRxBufferSize	8	When the SCB mode is UART, this parameter defines the size of the RX buffer.
UartRxTriggerLevel	7	When the SCB mode is UART, this parameter defines the number of entries in the RX FIFO to trigger the SCB.INTR_-RX.TRIGGER interrupt event.
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.
UartTxTriggerLevel	0	When the SCB mode is UART, this parameter defines the number of entries in the TX FIFO to trigger the SCB.INTR_-TX.TRIGGER interrupt event.
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.

8.5 Component type: TCPWM_P4 [v2.0]

8.5.1 Instance SamplingTimer

Description: 16-bit Timer Counter PWM (TCPWM)
Instance type: TCPWM_P4 [v2.0]
Datasheet: online component datasheet for TCPWM_P4

Table 19. Component Parameters for SamplingTimer

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	250	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	Timer Counter	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

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