PSoC® Creator™
Project Datasheet for esWiFi

Creation Time: 08/24/2016 12:18:12
User: LAPTOP-86R56EE2\steve
Project: esWiFi
Tool: PSoC Creator 3.3 CP3

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 4200L 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 4200L Device Family Block Diagram

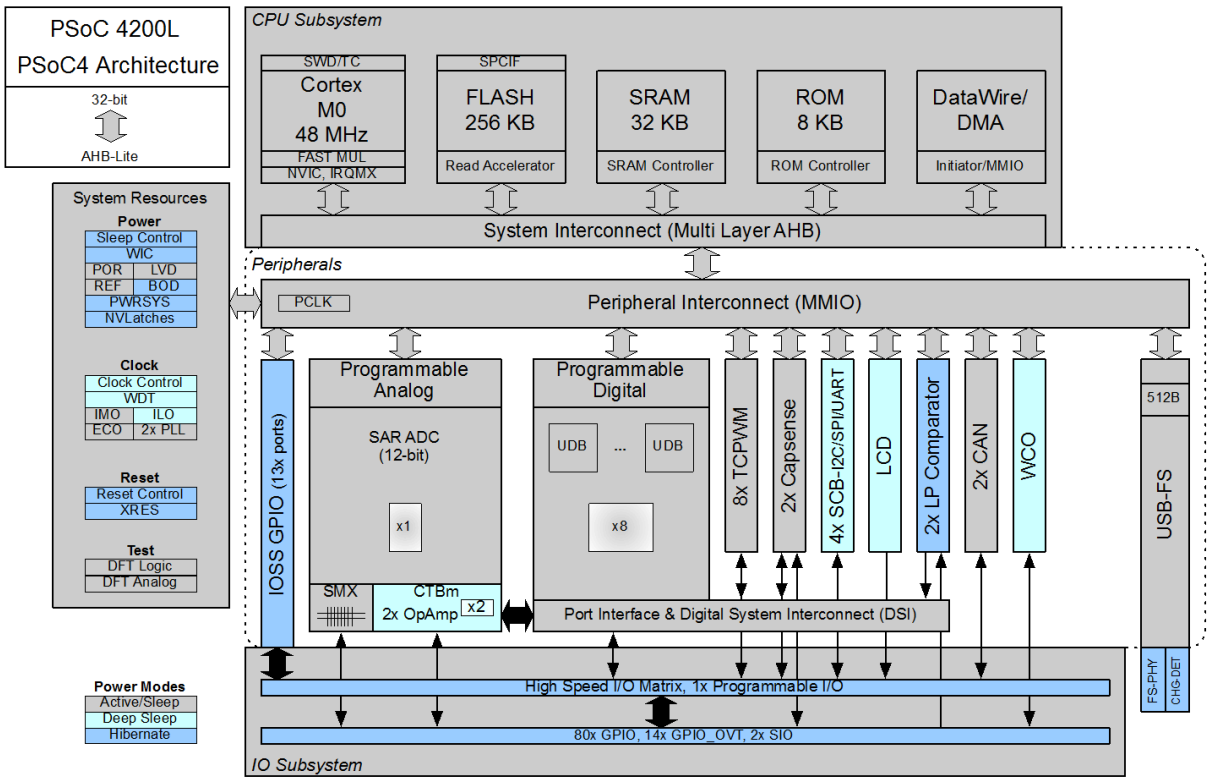


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C4248BZI-L489
Package Name	124-VFBGA
Architecture	PSoC 4
Family	PSoC 4200L
CPU speed (MHz)	48
Flash size (kBytes)	256
SRAM size (kBytes)	32
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:

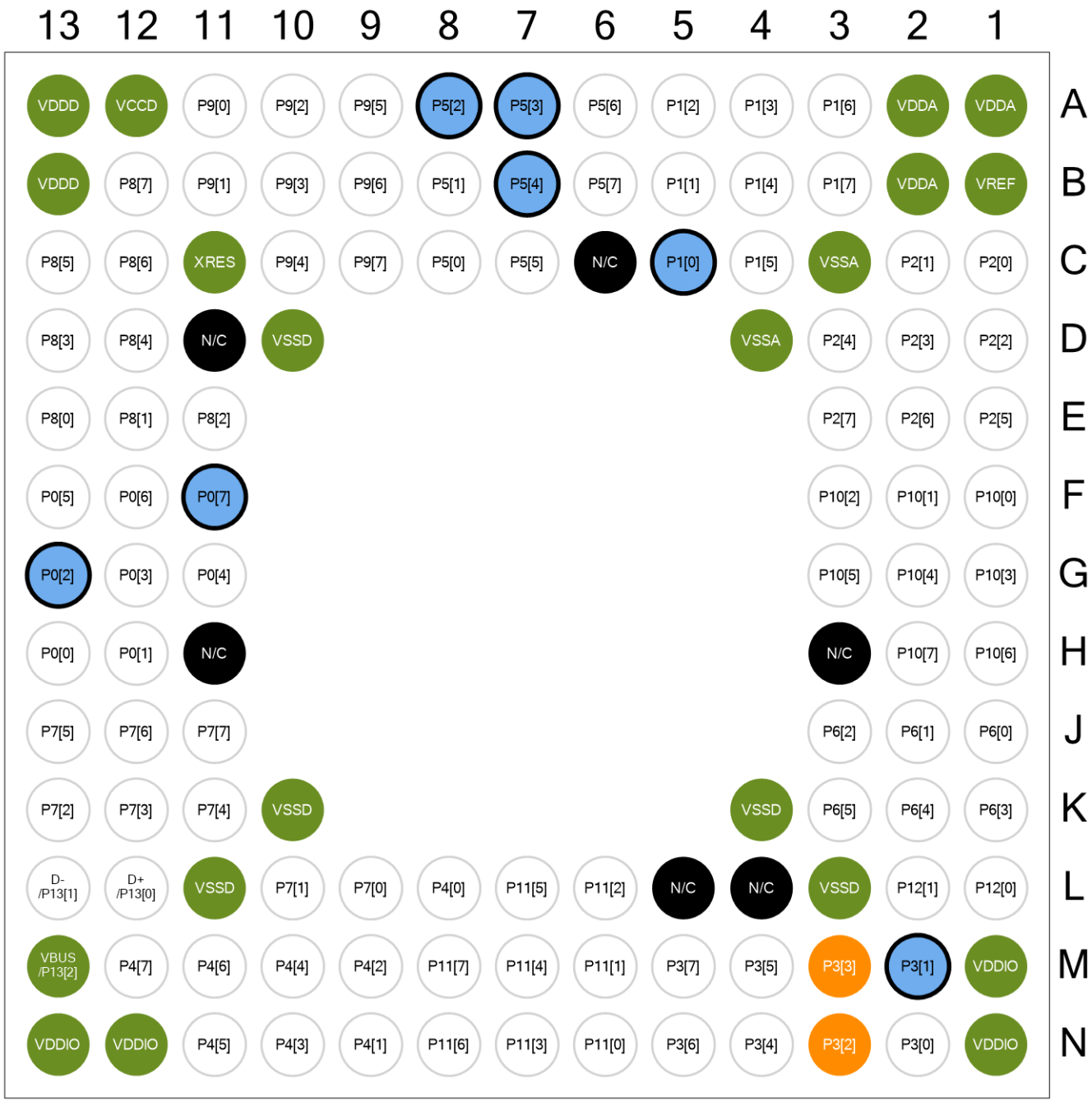
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Digital Clocks	1	3	4	25.00 %
Interrupts	2	30	32	6.25 %
IO	9	89	98	9.18 %
Segment LCD	0	1	1	0.00 %
CapSense	0	2	2	0.00 %
Die Temp	0	1	1	0.00 %
CAN 2.0b	0	2	2	0.00 %
Serial Communication (SCB)	1	3	4	25.00 %
USB	0	1	1	0.00 %
DMA Channels	0	32	32	0.00 %
Timer/Counter/PWM	3	5	8	37.50 %
Smart IO Ports	0	1	1	0.00 %
UDB				
Macrocells	25	39	64	39.06 %
Unique P-terms	45	83	128	35.16 %
Total P-terms	54			
Datapath Cells	4	4	8	50.00 %
Status Cells	4	4	8	50.00 %
StatusI Registers	3			
Routed Count7 Load/Enable	1			
Control Cells	2	6	8	25.00 %
Control Registers	1			
Count7 Cells	1			
Comparator/Opamp	0	4	4	0.00 %
LP Comparator	0	2	2	0.00 %
SAR ADC	0	1	1	0.00 %
DAC				
7-bit IDAC	0	2	2	0.00 %
8-bit IDAC	0	2	2	0.00 %

2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



CY8C4248BZI-L489
124-VFBGA
(bottom view)

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	VDDA	VDDA	Power	
2	VDDA	VDDA	Power	
3	P1[6]	GPIO [unused]		
4	P1[3]	GPIO [unused]		
5	P1[2]	GPIO [unused]		
6	P5[6]	GPIO [unused]		
7	P5[3]	Pin_GreenLED	Dgtl Out	HiZ digital
8	P5[2]	Pin_RedLED	Dgtl Out	HiZ digital
9	P9[5]	OVT IO [unused]		
10	P9[2]	OVT IO [unused]		
11	P9[0]	OVT IO [unused]		
12	VCCD	VCCD	Power	
13	VDDD	VDDD	Power	
14	VREF	VREF	Dedicated	
15	VDDA	VDDA	Power	
16	P1[7]	GPIO [unused]		
17	P1[4]	GPIO [unused]		
18	P1[1]	GPIO [unused]		
19	P5[7]	GPIO [unused]		
20	P5[4]	Pin_BlueLED	Dgtl Out	HiZ digital
21	P5[1]	GPIO [unused]		
22	P9[6]	OVT IO [unused]		
23	P9[3]	OVT IO [unused]		
24	P9[1]	OVT IO [unused]		
25	P8[7]	GPIO [unused]		
26	VDDD	VDDD	Power	
27	P2[0]	GPIO [unused]		
28	P2[1]	GPIO [unused]		
29	VSSA	VSSA	Power	
30	P1[5]	GPIO [unused]		
31	P1[0]	Rx_1	Dgtl In	HiZ digital
33	P5[5]	GPIO [unused]		
34	P5[0]	GPIO [unused]		
35	P9[7]	OVT IO [unused]		
36	P9[4]	OVT IO [unused]		
37	XRES	XRES	Dedicated	
38	P8[6]	GPIO [unused]		
39	P8[5]	GPIO [unused]		
40	P2[2]	GPIO [unused]		
41	P2[3]	GPIO [unused]		
42	P2[4]	GPIO [unused]		
43	VSSA	VSSA	Power	
49	VSSD	VSSD	Power	
51	P8[4]	GPIO [unused]		
52	P8[3]	GPIO [unused]		

Pin	Port	Name	Type	Drive Mode
53	P2[5]	GPIO [unused]		
54	P2[6]	GPIO [unused]		
55	P2[7]	GPIO [unused]		
63	P8[2]	GPIO [unused]		
64	P8[1]	GPIO [unused]		
65	P8[0]	GPIO [unused]		
66	P10[0]	GPIO [unused]		
67	P10[1]	GPIO [unused]		
68	P10[2]	GPIO [unused]		
76	P0[7]	SW2_P0_7	Software Input	Res pull up
77	P0[6]	GPIO [unused]		
78	P0[5]	GPIO [unused]		
79	P10[3]	GPIO [unused]		
80	P10[4]	GPIO [unused]		
81	P10[5]	GPIO [unused]		
89	P0[4]	GPIO [unused]		
90	P0[3]	GPIO [unused]		
91	P0[2]	Tx_1	Dgtl Out	Strong drive
92	P10[6]	GPIO [unused]		
93	P10[7]	GPIO [unused]		
103	P0[1]	GPIO [unused]		
104	P0[0]	GPIO [unused]		
105	P6[0]	OVT IO [unused]		
106	P6[1]	OVT IO [unused]		
107	P6[2]	OVT IO [unused]		
115	P7[7]	GPIO [unused]		
116	P7[6]	GPIO [unused]		
117	P7[5]	GPIO [unused]		
118	P6[3]	OVT IO [unused]		
119	P6[4]	OVT IO [unused]		
120	P6[5]	OVT IO [unused]		
121	VSSD	VSSD	Power	
127	VSSD	VSSD	Power	
128	P7[4]	GPIO [unused]		
129	P7[3]	GPIO [unused]		
130	P7[2]	GPIO [unused]		
131	P12[0]	SIO [unused]		
132	P12[1]	SIO [unused]		
133	VSSD	VSSD	Power	
136	P11[2]	GPIO [unused]		
137	P11[5]	GPIO [unused]		
138	P4[0]	GPIO [unused]		
139	P7[0]	GPIO [unused]		
140	P7[1]	GPIO [unused]		
141	VSSD	VSSD	Power	
142	D+/P13[0]	USB IO [unused]		
143	D-/P13[1]	USB IO [unused]		
144	VDDIO	VDDIO	Power	
145	P3[1]	\UART_DBG:tx\	Dgtl Out	Strong drive
146	P3[3]	Debug:SWD_CK	Reserved	
147	P3[5]	GPIO [unused]		
148	P3[7]	GPIO [unused]		
149	P11[1]	GPIO [unused]		

Pin	Port	Name	Type	Drive Mode
150	P11[4]	GPIO [unused]		
151	P11[7]	GPIO [unused]		
152	P4[2]	GPIO [unused]		
153	P4[4]	GPIO [unused]		
154	P4[6]	GPIO [unused]		
155	P4[7]	GPIO [unused]		
156	VBUS/P13[2]	VBUS/P13[2]	Power	
157	VDDIO	VDDIO	Power	
158	P3[0]	GPIO [unused]		
159	P3[2]	Debug:SWD_IO	Reserved	
160	P3[4]	GPIO [unused]		
161	P3[6]	GPIO [unused]		
162	P11[0]	GPIO [unused]		
163	P11[3]	GPIO [unused]		
164	P11[6]	GPIO [unused]		
165	P4[1]	GPIO [unused]		
166	P4[3]	GPIO [unused]		
167	P4[5]	GPIO [unused]		
168	VDDIO	VDDIO	Power	
169	VDDIO	VDDIO	Power	

Abbreviations used in Table 3 have the following meanings:

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

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
D-/P13[1]	143	USB IO [unused]		
D+/P13[0]	142	USB IO [unused]		
P0[0]	104	GPIO [unused]		
P0[1]	103	GPIO [unused]		
P0[2]	91	Tx_1	Dgtl Out	Strong drive
P0[3]	90	GPIO [unused]		
P0[4]	89	GPIO [unused]		
P0[5]	78	GPIO [unused]		
P0[6]	77	GPIO [unused]		
P0[7]	76	SW2_P0_7	Software Input	Res pull up
P1[0]	31	Rx_1	Dgtl In	HiZ digital
P1[1]	18	GPIO [unused]		
P1[2]	5	GPIO [unused]		
P1[3]	4	GPIO [unused]		
P1[4]	17	GPIO [unused]		
P1[5]	30	GPIO [unused]		
P1[6]	3	GPIO [unused]		
P1[7]	16	GPIO [unused]		
P10[0]	66	GPIO [unused]		
P10[1]	67	GPIO [unused]		
P10[2]	68	GPIO [unused]		
P10[3]	79	GPIO [unused]		
P10[4]	80	GPIO [unused]		
P10[5]	81	GPIO [unused]		
P10[6]	92	GPIO [unused]		
P10[7]	93	GPIO [unused]		
P11[0]	162	GPIO [unused]		
P11[1]	149	GPIO [unused]		
P11[2]	136	GPIO [unused]		
P11[3]	163	GPIO [unused]		
P11[4]	150	GPIO [unused]		
P11[5]	137	GPIO [unused]		
P11[6]	164	GPIO [unused]		
P11[7]	151	GPIO [unused]		
P12[0]	131	SIO [unused]		
P12[1]	132	SIO [unused]		
P2[0]	27	GPIO [unused]		
P2[1]	28	GPIO [unused]		
P2[2]	40	GPIO [unused]		
P2[3]	41	GPIO [unused]		
P2[4]	42	GPIO [unused]		
P2[5]	53	GPIO [unused]		
P2[6]	54	GPIO [unused]		
P2[7]	55	GPIO [unused]		
P3[0]	158	GPIO [unused]		

Port	Pin	Name	Type	Drive Mode
P3[1]	145	\UART_DBG:tx\	Dgtl Out	Strong drive
P3[2]	159	Debug:SWD_IO	Reserved	
P3[3]	146	Debug:SWD_CK	Reserved	
P3[4]	160	GPIO [unused]		
P3[5]	147	GPIO [unused]		
P3[6]	161	GPIO [unused]		
P3[7]	148	GPIO [unused]		
P4[0]	138	GPIO [unused]		
P4[1]	165	GPIO [unused]		
P4[2]	152	GPIO [unused]		
P4[3]	166	GPIO [unused]		
P4[4]	153	GPIO [unused]		
P4[5]	167	GPIO [unused]		
P4[6]	154	GPIO [unused]		
P4[7]	155	GPIO [unused]		
P5[0]	34	GPIO [unused]		
P5[1]	21	GPIO [unused]		
P5[2]	8	Pin_RedLED	Dgtl Out	HiZ digital
P5[3]	7	Pin_GreenLED	Dgtl Out	HiZ digital
P5[4]	20	Pin_BlueLED	Dgtl Out	HiZ digital
P5[5]	33	GPIO [unused]		
P5[6]	6	GPIO [unused]		
P5[7]	19	GPIO [unused]		
P6[0]	105	OVT IO [unused]		
P6[1]	106	OVT IO [unused]		
P6[2]	107	OVT IO [unused]		
P6[3]	118	OVT IO [unused]		
P6[4]	119	OVT IO [unused]		
P6[5]	120	OVT IO [unused]		
P7[0]	139	GPIO [unused]		
P7[1]	140	GPIO [unused]		
P7[2]	130	GPIO [unused]		
P7[3]	129	GPIO [unused]		
P7[4]	128	GPIO [unused]		
P7[5]	117	GPIO [unused]		
P7[6]	116	GPIO [unused]		
P7[7]	115	GPIO [unused]		
P8[0]	65	GPIO [unused]		
P8[1]	64	GPIO [unused]		
P8[2]	63	GPIO [unused]		
P8[3]	52	GPIO [unused]		
P8[4]	51	GPIO [unused]		
P8[5]	39	GPIO [unused]		
P8[6]	38	GPIO [unused]		
P8[7]	25	GPIO [unused]		
P9[0]	11	OVT IO [unused]		
P9[1]	24	OVT IO [unused]		
P9[2]	10	OVT IO [unused]		
P9[3]	23	OVT IO [unused]		
P9[4]	36	OVT IO [unused]		
P9[5]	9	OVT IO [unused]		
P9[6]	22	OVT IO [unused]		
P9[7]	35	OVT IO [unused]		

Port	Pin	Name	Type	Drive Mode
VBUS/P13[2]	156	VBUS/P13[2]	Power	

Abbreviations used in Table 4 have the following meanings:

- Dgtl Out = Digital Output
- Res pull up = Resistive pull up
- 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
\UART_DBG:tx\	P3[1]	Dgtl Out
Debug:SWD_CK	P3[3]	Reserved
Debug:SWD_IO	P3[2]	Reserved
GPIO [unused]	P3[0]	
GPIO [unused]	P8[0]	
GPIO [unused]	P0[4]	
GPIO [unused]	P0[3]	
GPIO [unused]	P8[1]	
GPIO [unused]	P8[2]	
GPIO [unused]	P10[6]	
GPIO [unused]	P10[7]	
GPIO [unused]	P0[6]	
GPIO [unused]	P3[4]	
GPIO [unused]	P0[5]	
GPIO [unused]	P10[3]	
GPIO [unused]	P10[1]	
GPIO [unused]	P10[0]	
GPIO [unused]	P10[2]	
GPIO [unused]	P10[4]	
GPIO [unused]	P7[0]	
GPIO [unused]	P7[1]	
GPIO [unused]	P4[0]	
GPIO [unused]	P11[2]	
GPIO [unused]	P11[5]	
GPIO [unused]	P4[2]	
GPIO [unused]	P3[5]	
GPIO [unused]	P3[7]	
GPIO [unused]	P11[1]	
GPIO [unused]	P11[7]	
GPIO [unused]	P11[4]	
GPIO [unused]	P7[6]	
GPIO [unused]	P7[5]	
GPIO [unused]	P7[7]	
GPIO [unused]	P0[1]	
GPIO [unused]	P0[0]	
GPIO [unused]	P7[4]	
GPIO [unused]	P4[6]	
GPIO [unused]	P4[4]	
GPIO [unused]	P4[7]	
GPIO [unused]	P7[3]	
GPIO [unused]	P7[2]	
GPIO [unused]	P2[7]	
GPIO [unused]	P10[5]	
GPIO [unused]	P8[7]	
GPIO [unused]	P5[7]	

Name	Port	Type
GPIO [unused]	P5[1]	
GPIO [unused]	P2[1]	
GPIO [unused]	P11[6]	
GPIO [unused]	P4[1]	
GPIO [unused]	P2[0]	
GPIO [unused]	P1[1]	
GPIO [unused]	P1[2]	
GPIO [unused]	P5[6]	
GPIO [unused]	P1[6]	
GPIO [unused]	P1[3]	
GPIO [unused]	P1[7]	
GPIO [unused]	P1[4]	
GPIO [unused]	P4[5]	
GPIO [unused]	P4[3]	
GPIO [unused]	P11[0]	
GPIO [unused]	P3[6]	
GPIO [unused]	P8[5]	
GPIO [unused]	P2[4]	
GPIO [unused]	P8[6]	
GPIO [unused]	P2[2]	
GPIO [unused]	P2[3]	
GPIO [unused]	P8[4]	
GPIO [unused]	P5[5]	
GPIO [unused]	P5[0]	
GPIO [unused]	P1[5]	
GPIO [unused]	P11[3]	
GPIO [unused]	P2[6]	
GPIO [unused]	P8[3]	
GPIO [unused]	P2[5]	
OVT IO [unused]	P9[5]	
OVT IO [unused]	P9[0]	
OVT IO [unused]	P9[2]	
OVT IO [unused]	P9[1]	
OVT IO [unused]	P9[4]	
OVT IO [unused]	P6[3]	
OVT IO [unused]	P9[7]	
OVT IO [unused]	P6[0]	
OVT IO [unused]	P6[1]	
OVT IO [unused]	P6[2]	
OVT IO [unused]	P6[5]	
OVT IO [unused]	P9[6]	
OVT IO [unused]	P9[3]	
OVT IO [unused]	P6[4]	
Pin_BlueLED	P5[4]	Dgtl Out
Pin_GreenLED	P5[3]	Dgtl Out
Pin_RedLED	P5[2]	Dgtl Out
Power	VBUS/P13[2]	
Rx_1	P1[0]	Dgtl In
SIO [unused]	P12[0]	
SIO [unused]	P12[1]	
SW2_P0_7	P0[7]	Software Input
Tx_1	P0[2]	Dgtl Out
USB IO [unused]	D+/P13[0]	

Name	Port	Type
USB IO [unused]	D-/P13[1]	

Abbreviations used in Table 5 have the following meanings:

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

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)	0x0800
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 7. System Debug Settings

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

3.3 System Operating Conditions

Table 8. System Operating Conditions

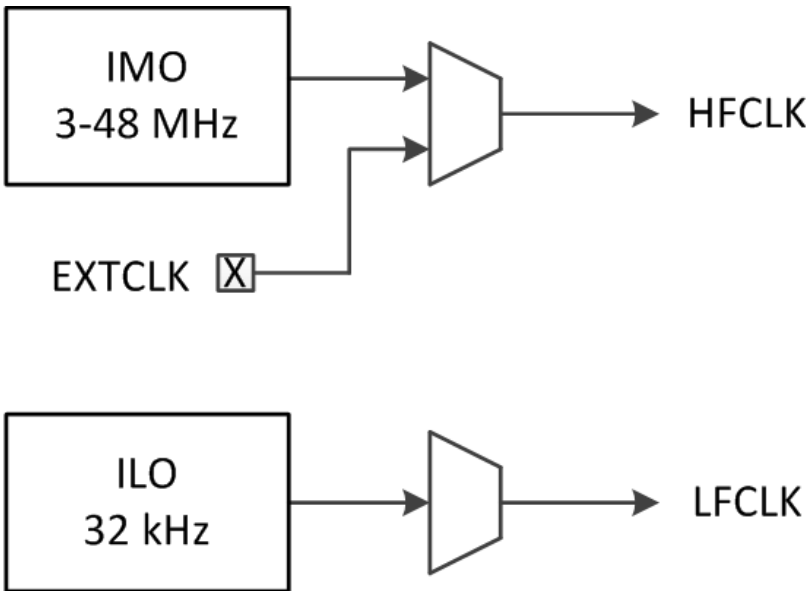
Name	Value
Variable VDDA	True
VDDA (V)	3.3
VDDD (V)	3.3
VDDIO (V)	3.3
VBUS (V)	5.0

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	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
DPLL_Sel	NONE	IMO	48 MHz	48 MHz	±2	True	True
SYSCLK	NONE	HFCLK	? MHz	48 MHz	±2	True	True
Direct_Sel	NONE	IMO	48 MHz	48 MHz	±2	True	True
PLL1_Sel	NONE	IMO	48 MHz	48 MHz	±2	True	True
PLL0_Sel	NONE	IMO	48 MHz	48 MHz	±2	True	True
HFCLK	NONE	Direct_Sel	48 MHz	48 MHz	±2	True	True
IMO	NONE		48 MHz	48 MHz	±2	True	True
LFCLK	NONE	ILO	? MHz	32 kHz	±60	True	True
ILO	NONE		32 kHz	32 kHz	±60	True	True
RTC_Sel	NONE	None	? MHz	? MHz	±0	True	True
Timer1 (WDT1)	NONE	LFCLK	? MHz	? MHz	±0	False	False
EXTCLK	NONE		24 MHz	? MHz	±0	False	False
PLL1	NONE	PLL1_Sel	96 MHz	? MHz	±0.25	False	False
Timer2 (WDT2)	NONE	LFCLK	? MHz	? MHz	±0	False	False
Timer0 (WDT0)	NONE	LFCLK	? MHz	? MHz	±0.025	False	False
DigSig2	NONE		? MHz	? MHz	±0	False	False
DigSig4	NONE		? MHz	? MHz	±0	False	False
DigSig3	NONE		? MHz	? MHz	±0	False	False
ECO	NONE		24 MHz	? MHz	±0	False	False
WCO	NONE		32.768 kHz	? MHz	±0.025	False	False
DigSig1	NONE		? MHz	? MHz	±0	False	False
PLL0	NONE	PLL0_Sel	48 MHz	? MHz	±0.25	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

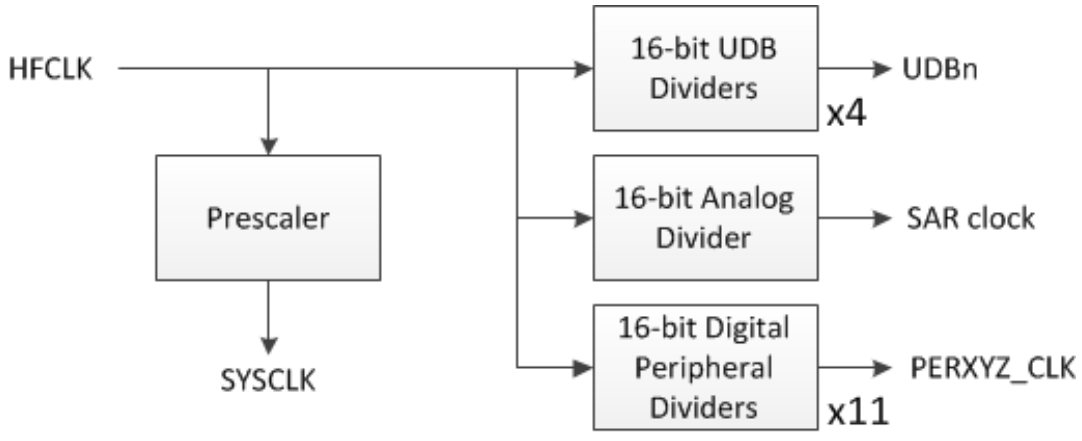


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
clockTimer	NONE	HFCLK	48 MHz	48 MHz	±2	True	True
Clock_PWM	FIXED_-FUNCT-ION	HFCLK	? MHz	48 MHz	±2	True	True
UART_DBG_-SCBCLK	FIXED_-FUNCT-ION	HFCLK	1.382 MHz	1.371 MHz	±2	True	True
UART_-ESWIFI_-IntClock	DIGITAL	HFCLK	921.6 kHz	923.077 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
  - o CySysClkImo API routines
  - o CySysClkIlo API routines
  - o CySysClkPll1 API routines
  - o CySysClkPll0 API routines
  - o CySysClkEco API routines
  - o CySysClkWco API routines
  - o CySysClkWrite API routines

5 Interrupts and DMAs

5.1 Interrupts

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

Table 11. Interrupts

Name	Priority	Vector
ESWIFI_RX_ISR	3	0
TimerISR	3	1

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

5.2 DMAs

This design contains no DMA components.

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	0x3FFFF	U - Unprotected

Flash memory is organized as rows with each row of flash having 256 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:

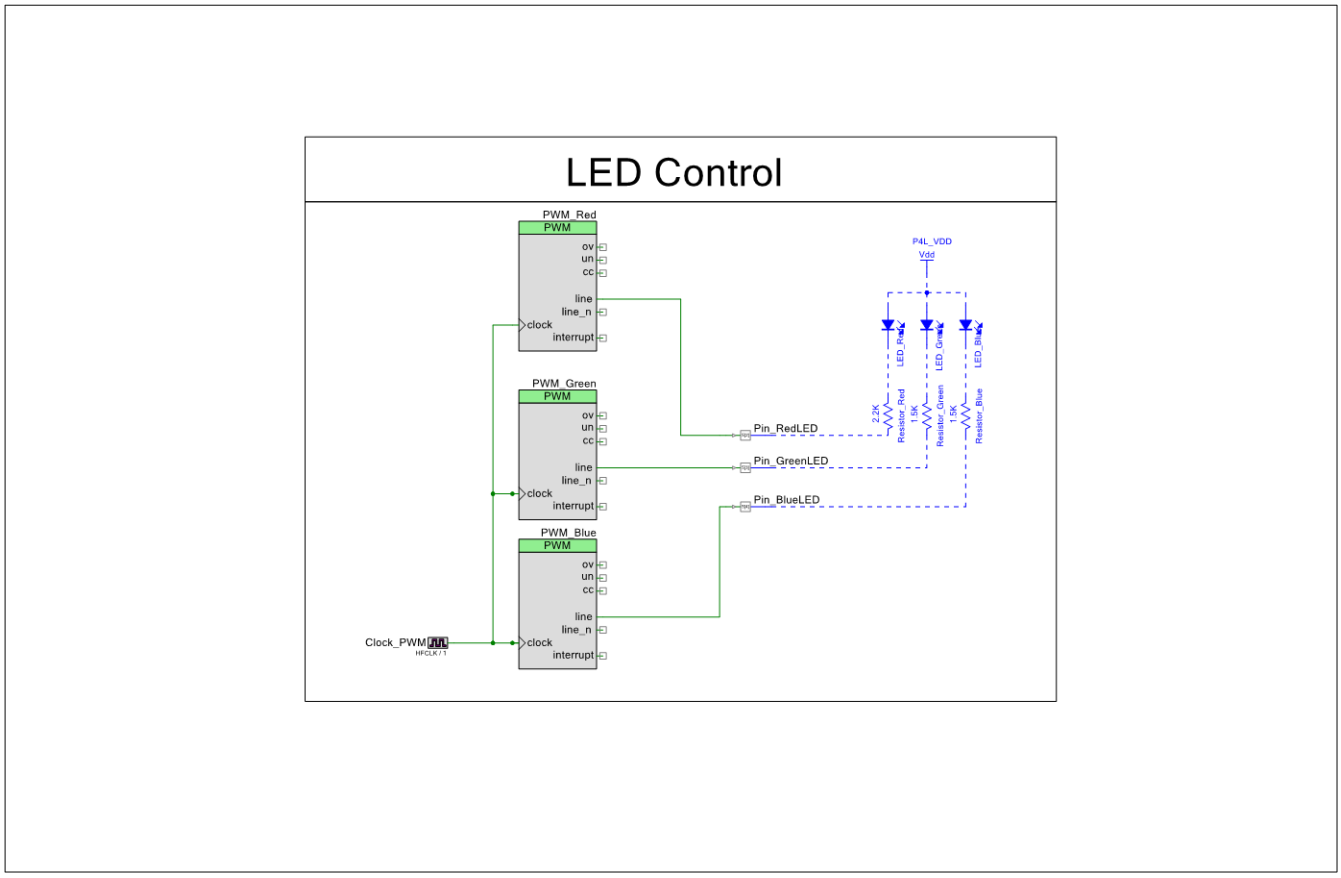
- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CySysFlash API routines

7 Design Contents

This design's schematic content consists of the following 4 schematic sheets:

7.1 Schematic Sheet: LED Control

Figure 5. Schematic Sheet: LED Control

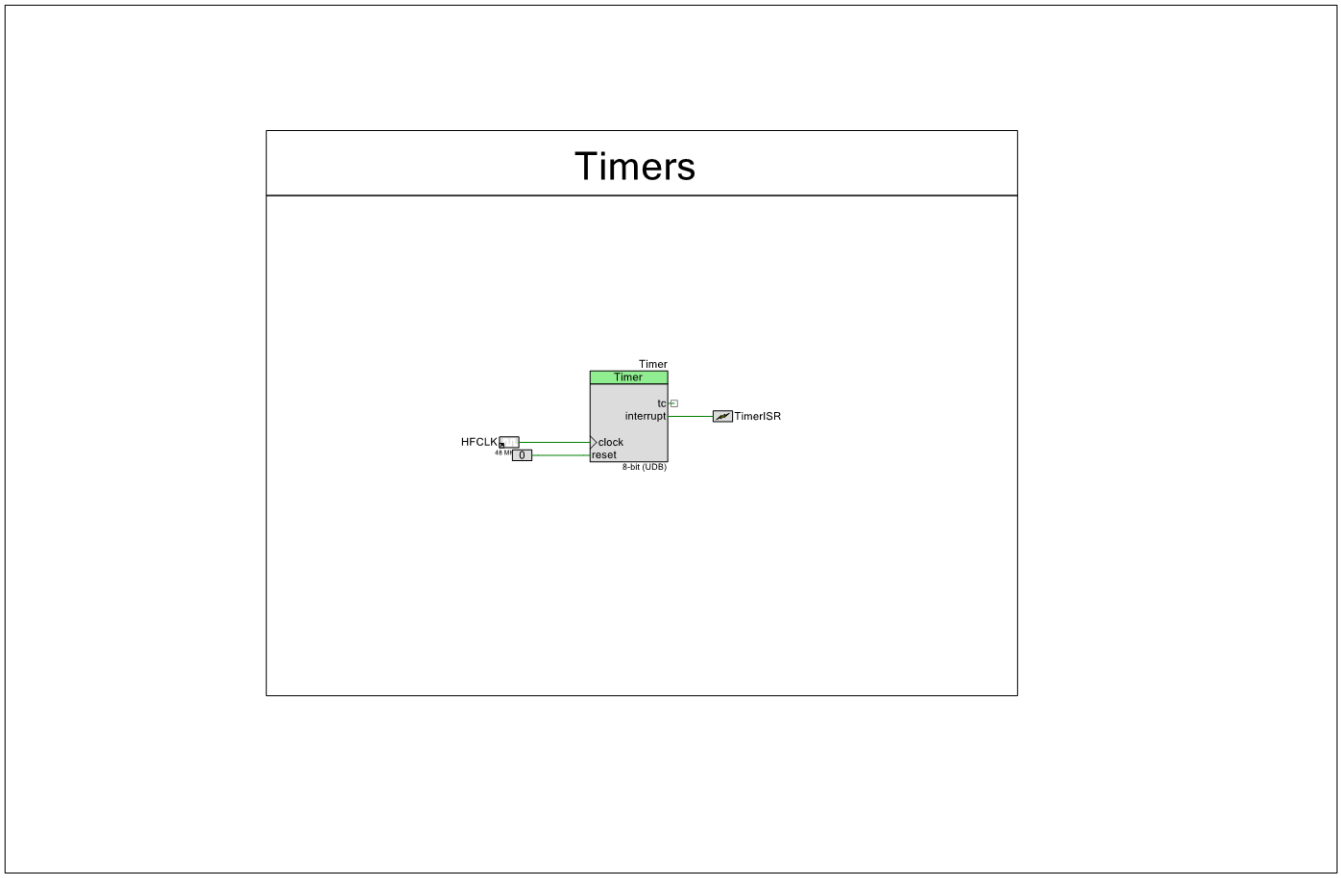


This schematic sheet contains the following component instances:

- Instance PWM_Blue (type: TCPWM_P4_v2_10)
- Instance PWM_Green (type: TCPWM_P4_v2_10)
- Instance PWM_Red (type: TCPWM_P4_v2_10)

7.2 Schematic Sheet: Timer

Figure 6. Schematic Sheet: Timer

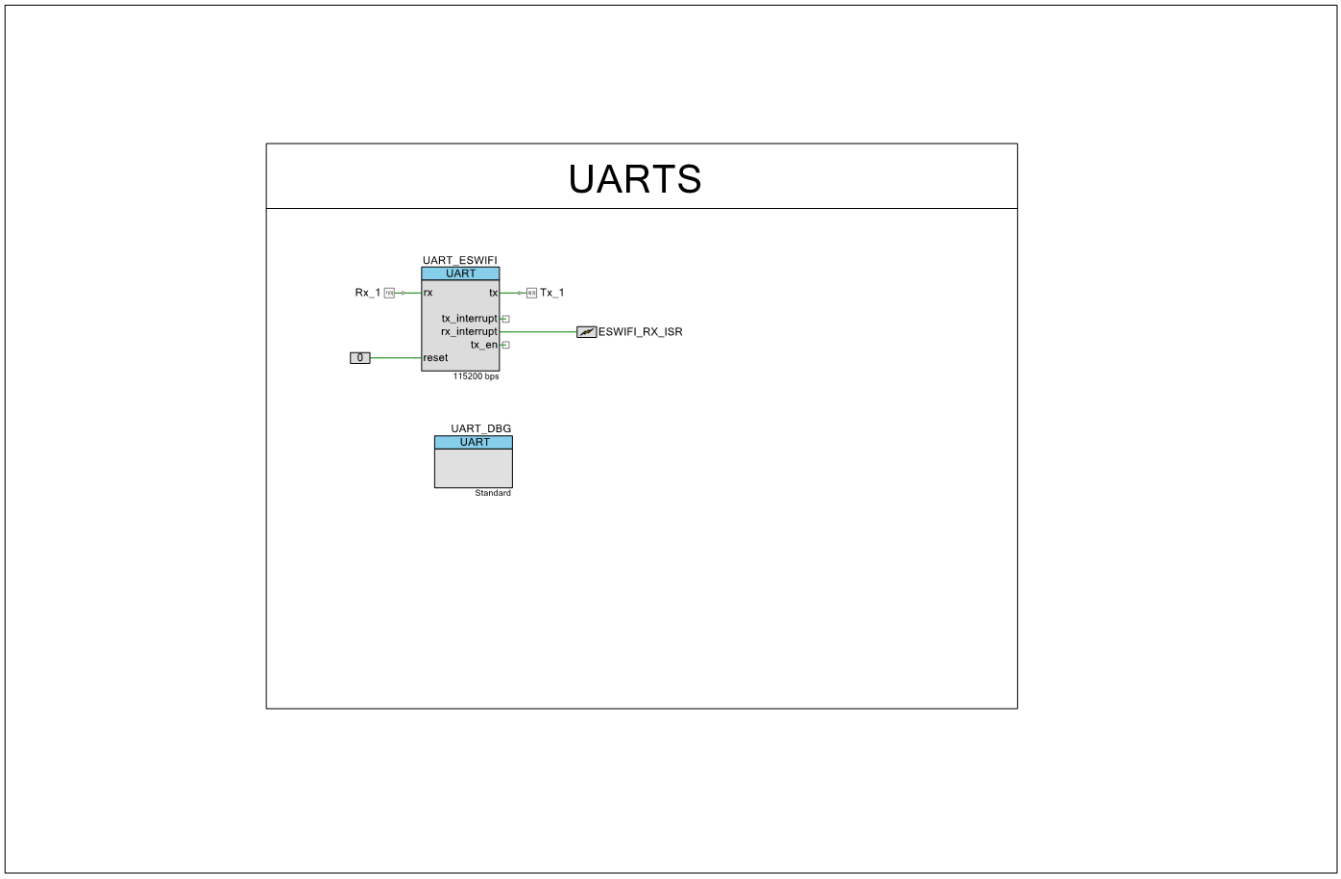


This schematic sheet contains the following component instances:

- Instance Timer (type: Timer_v2_70)

7.3 Schematic Sheet: UARTS

Figure 7. Schematic Sheet: UARTS



This schematic sheet contains the following component instances:

- Instance UART_DBG (type: SCB_P4_v3_20)
- Instance UART_ESWIFI (type: UART_v2_50)

7.4 Schematic Sheet: Page 1

Figure 8. Schematic Sheet: Page 1

8 Components

8.1 Component type: SCB_P4 [v3.20]

8.1.1 Instance UART_DBG

Description: Serial Communication Block (SCB)
Instance type: SCB_P4 [v3.20]
Datasheet: online component datasheet for SCB_P4

Table 13. Component Parameters for UART_DBG

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. Only applicable for devices other than PSoC 4000/PSoC 4100/PSoC 4200.
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.
EzI2cSlewRate	Fast	When the SCB mode is EZI2C, this parameter specifies the slew rate settings of the I2C pins. For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined.Refer to the Device Datasheet to determine which pins are GPIO_OVT capable.
EzI2cSubAddressSize	8	When the SCB mode is EZI2C, this parameter specifies the maximum size of the slave buffer that is exposed to the master: 8bits – maximum buffer size is 256 bytes, 16 bits – maximum buffer size is 65535 bytes.
EzI2cWakeEnable	false	When the SCB mode is EZI2C, this parameter enables wakeup from Deep Sleep on I2C address match event.
I2cAcceptAddress	false	When the SCB mode is I2C, this parameter specifies whether to accept the match slave address in RX FIFO or not. All slave matched addresses are ACKed. The user has to register the callback function to handle accepted addresses. This feature has to be used when more than one address support is required.
I2cAcceptGeneralCall	false	When the SCB mode is I2C, this parameter specifies whether to accept the general call address. The general call address is ACKed when accepted and NAKed otherwise. The user has to register the callback function to handle the general call address.
I2cBusVoltage	3.3	When the SCB mode is I2C, this parameter specifies the voltage applied to the pull-up resistors on the I2C bus. Only applicable for devices other than PSoC 4000/PSoC 4100/PSoC 4200.

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.
I2cSlewRate	Fast	When the SCB mode is I2C, this parameter specifies the slew rate settings of the I2C pins. For devices supporting GPIO Over-Voltage Tolerance (GPIO_OVT) pins, I2C FM+ options should be used when I2C data rate is greater than 400 kbps. This option also requires the I2C bus voltage to be defined. Refer to the Device Datasheet to determine which pins are GPIO_OVT capable.
I2cWakeEnable	false	When the SCB mode is I2C, this parameter enables wakeup from Deep Sleep on an I2C address match event.
ScbMisoSdaTxEnable	true	This parameter defines the availability of the spi_miso_i2c_-sda_uart_tx pin.
ScbMode	UART	This parameter defines the mode of operation for the SCB component.
ScbMosiSclRxEnable	true	This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx pin.
ScbRxWakeIrqEnable	false	This parameter defines the availability of the spi_mosi_i2c_-scl_uart_rx_wake pin.
ScbSclkEnable	false	This parameter defines the availability of the sclk pin.
ScbSs0Enable	false	This parameter defines the availability of the ss0 pin.
ScbSs1Enable	false	This parameter defines the availability of the ss1 pin.
ScbSs2Enable	false	This parameter defines the availability of the ss2 pin.
ScbSs3Enable	false	This parameter defines the availability of the ss3 pin.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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).
SpiSmartioEnable	false	When the SCB mode is SPI, this parameter enables the SmartIO support.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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 only	When the SCB mode is UART, this parameter enables RX or TX direction or both simultaneously.
UartDropOnFrameErr	false	When the SCB mode is UART, this parameter defines whether the data is dropped from RX FIFO on a frame error event.
UartDropOnParityErr	false	When the SCB mode is UART, this parameter determines whether the data is dropped from RX FIFO on a parity error event.
UartInterruptMode	None	When the SCB mode is UART, this parameter specifies the interrupt mode. None: Removes all interrupt support. Internal: Leaves the interrupt SCBIRQ inside the component - the interrupt terminal becomes invisible. External: Provides an interrupt terminal to connect an interrupt outside component.
UartIntrRxFrameErr	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FRAME_-ERROR interrupt source. SCB.INTR_RX.FRAME_-ERROR trigger condition: frame error in received data frame.
UartIntrRxFull	false	When the SCB mode is UART, this parameter enables the SCB.INTR_RX.FULL interrupt source. SCB.INTR_RX.FULL trigger condition: RX FIFO is full.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.

Parameter Name	Value	Description
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.
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.
UartSmartioEnable	false	When the SCB mode is UART, this parameter enables the SmartIO support.
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.

Parameter Name	Value	Description
UartTxOutputEnable	false	When the SCB mode is UART, this parameter enables the TX trigger output terminal of the component. This terminal must be connected to the DMA input trigger or left unconnected. Only applicable for devices which have a DMA controller.
UartTxTriggerLevel	0	When the SCB mode is UART, this parameter defines the number of entries in the TX FIFO to control the SCB.INTR_-TX.TRIGGER interrupt event or TX DMA trigger output.
UartWakeEnable	false	When the SCB mode is UART, this parameter enables the wakeup from Deep Sleep on start bit event. The actual wakeup source is RX GPIO. The skip start UART feature allows it to continue receiving bytes.

8.2 Component type: TCPWM_P4 [v2.10]

8.2.1 Instance PWM_Blue

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

Table 14. Component Parameters for PWM_Blue

Parameter Name	Value	Description
PWMCompare	32767	The initial value for the comparison register when in the PWM mode
PWMCompareBuf	65535	The initial value for the second comparison register when in the PWM mode
PWMCompareSwap	Disable swap	Determines whether the PWM swap check box is enabled or disabled
PWMCountMode	Level	Determines whether the PWM counter counts at level detection or in various modes of edge detection
PWMCountPresent	false	Determines if the PWM count signal is present and controls the visibility of the count pin
PWMDeadTimeCycle	0	Sets the number of cycles of dead time insertion
PWMInterruptMask	None	The mask used for enabling the interrupt bit in the PWM mode
PWMKillEvent	Asynchronous	Selects whether a PWM kill event is synchronous or asynchronous to the input clock

Parameter Name	Value	Description
PWMLinenSignal	Direct Output	Selects whether the PWM line_n signal is inverted or is directly output
PWMLineSignal	Inverse 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
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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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

8.2.2 Instance PWM_Green

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 PWM_Green

Parameter Name	Value	Description
PWMCompare	32767	The initial value for the comparison register when in the PWM mode
PWMCompareBuf	65535	The initial value for the second comparison register when in the PWM mode
PWMCompareSwap	Disable swap	Determines whether the PWM swap check box is enabled or disabled
PWMCountMode	Level	Determines whether the PWM counter counts at level detection or in various modes of edge detection
PWMCountPresent	false	Determines if the PWM count signal is present and controls the visibility of the count pin
PWMDeadTimeCycle	0	Sets the number of cycles of dead time insertion
PWMInterruptMask	None	The mask used for enabling the interrupt bit in the PWM mode
PWMKillEvent	Asynchronous	Selects whether a PWM kill event is synchronous or asynchronous to the input clock
PWMLinenSignal	Direct Output	Selects whether the PWM line_n signal is inverted or is directly output
PWMLineSignal	Inverse 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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
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
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

8.2.3 Instance PWM_Red

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 PWM_Red

Parameter Name	Value	Description
PWMCompare	32767	The initial value for the comparison register when in the PWM mode
PWMCompareBuf	65535	The initial value for the second comparison register when in the PWM mode
PWMCompareSwap	Disable swap	Determines whether the PWM swap check box is enabled or disabled
PWMCountMode	Level	Determines whether the PWM counter counts at level detection or in various modes of edge detection
PWMCountPresent	false	Determines if the PWM count signal is present and controls the visibility of the count pin
PWMDeadTimeCycle	0	Sets the number of cycles of dead time insertion
PWMInterruptMask	None	The mask used for enabling the interrupt bit in the PWM mode
PWMKillEvent	Asynchronous	Selects whether a PWM kill event is synchronous or asynchronous to the input clock
PWMLinenSignal	Direct Output	Selects whether the PWM line_n signal is inverted or is directly output
PWMLineSignal	Inverse 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

Parameter Name	Value	Description
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
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

8.3 Component type: Timer [v2.70]

8.3.1 Instance Timer

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.70]
Datasheet: online component datasheet for Timer

Table 17. Component Parameters for Timer

Parameter Name	Value	Description
CaptureAlternatingFall	false	Enables data capture on either edge but not until a valid falling edge is detected first.

Parameter Name	Value	Description
CaptureAlternatingRise	false	Enables data capture on either edge but not until a valid rising edge is detected first.
CaptureCount	2	The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed.
CaptureCounterEnabled	false	Enables the capture counter to count capture events (up to 127) before a capture is triggered.
CaptureMode	None	This parameter defines the capture input signal requirements to trigger a valid capture event
EnableMode	Software Only	This parameter specifies the methods in enabling the component. Hardware mode makes the enable input pin visible. Software mode may reduce the resource usage if not enabled.
FixedFunction	false	Configures the component to use fixed function HW block instead of the UDB implementation.
InterruptOnCapture	false	Parameter to check whether interrupt on a capture event is enabled or disabled.
InterruptOnFIFOFull	false	Parameter to check whether interrupt on a FIFO Full event is enabled disabled.
InterruptOnTC	true	Parameter to check whether interrupt on a TC is enabled or disabled.
NumberOfCaptures	1	Number of captures allowed until the counter is cleared or disabled.
Period	191	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	8	Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals.
RunMode	Continuous	Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered.
TriggerMode	None	Defines the required trigger input signal to cause a valid trigger enable of the timer

8.4 Component type: UART [v2.50]

8.4.1 Instance UART_ESWIFI

Description: Universal Asynchronous Receiver Transmitter
Instance type: UART [v2.50]
Datasheet: online component datasheet for UART

Table 18. Component Parameters for UART_ESWIFI

Parameter Name	Value	Description
Address1	0	This parameter specifies the RX Hardware Address #1.
Address2	0	This parameter specifies the RX Hardware Address #2.
BaudRate	115200	Sets the target baud rate.
BreakBitsRX	13	Specifies the break signal length for the RX (detection) channel.
BreakBitsTX	13	Specifies the break signal length for the TX channel.
BreakDetect	false	Enables the break detect hardware.
CRCoutputsEn	false	Enables the CRC outputs.
EnIntRXInterrupt	false	Enables the internal RX interrupt configuration and the ISR.
EnIntTXInterrupt	false	Enables the internal TX interrupt configuration and the ISR.
FlowControl	None	Enable the flow control signals.
HalfDuplexEn	false	Enables half duplex mode on the RX Half of the UART module.
HwTXEnSignal	true	Enables the external TX enable signal output.
InternalClock	true	Enables the internal clock. This parameter removes the clock input pin.
InterruptOnTXComplete	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX complete' event.
InterruptOnTXFifoEmpty	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO empty' event.
InterruptOnTXFifoFull	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO full' event.
InterruptOnTXFifoNotFull	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO not full' event.
IntOnAddressDetect	false	Enables the interrupt on hardware address detected event by default
IntOnAddressMatch	false	Enables the interrupt on hardware address match detected event by default
IntOnBreak	false	Enables the interrupt on break signal detected event by default

Parameter Name	Value	Description
IntOnByteRcvd	true	Enables the interrupt on RX byte received event by default
IntOnOverrunError	false	Enables the interrupt on overrun error event by default
IntOnParityError	false	Enables the interrupt on parity error event by default
IntOnStopError	false	Enables the interrupt on stop error event by default
NumDataBits	8	Defines the number of data bits. Values can be 5, 6, 7 or 8 bits.
NumStopBits	1	Defines the number of stop bits. Values can be 1 or 2 bits.
OverSamplingRate	8	This parameter defines the over sampling rate.
ParityType	None	Sets the parity type as Odd, Even or Mark/Space
ParityTypeSw	false	This parameter allows the parity type to be changed through software by using the WriteControlRegister API
RXAddressMode	None	Configures the RX hardware address detection mode
RXBufferSize	4	The size of the RAM space allocated for the RX input buffer.
RXEnable	true	Enables the RX in the UART
TXBitClkGenDP	true	When enabled, this parameter enables the TX clock generation on DataPath resource. When disabled, TX clock is generated from Clock7.
TXBufferSize	4	The size of the RAM space allocated for the TX output buffer.
TXEnable	true	Enables the TX in the UART
Use23Polling	true	Allows the use of 2 out of 3 polling resources on the RX UART sampler.

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