PSoC® Creator™
Project Datasheet for Microcontroller_coolingsystem

Creation Time: 12/14/2021 11:20:32
User: LAPTOP-DSVHTDDH\nico_
Project: Microcontroller_coolingsystem
Tool: PSoC Creator 4.4

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

Copyright

Trademarks

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

Philips I2C Patent Rights

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

Disclaimer

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

Flash Code Protection

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

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

Table of Contents

1 Overview

The Cypress PSoC 5 is a family of 32-bit devices with the following characteristics:

- High-performance 32-bit ARM Cortex-M3 core with a nested vectored interrupt controller (NVIC) and a high-performance DMA controller
- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals, such as USB, I2C and SPI
- Analog subsystem that includes 20-bit Delta Sigma converters (ADC), SAR ADCs, 8-bit DACs that can be configured for 12-bit operation, comparators, op amps and configurable switched capacitor (SC) and continuous time (CT) blocks to create PGAs, TIAs, mixers, and more
- Several types of memory elements, including SRAM, flash, and EEPROM
- Programming and debug system through JTAG, serial wire debug (SWD), and single wire viewer (SWV)
- Flexible routing to all pins

Figure 1 shows the major components of a typical CY8C58LP series member PSoC 5LP device. For details on all the systems listed above, please refer to the PSoC 5LP Technical Reference Manual.

Figure 1. CY8C58LP Device Series Block Diagram

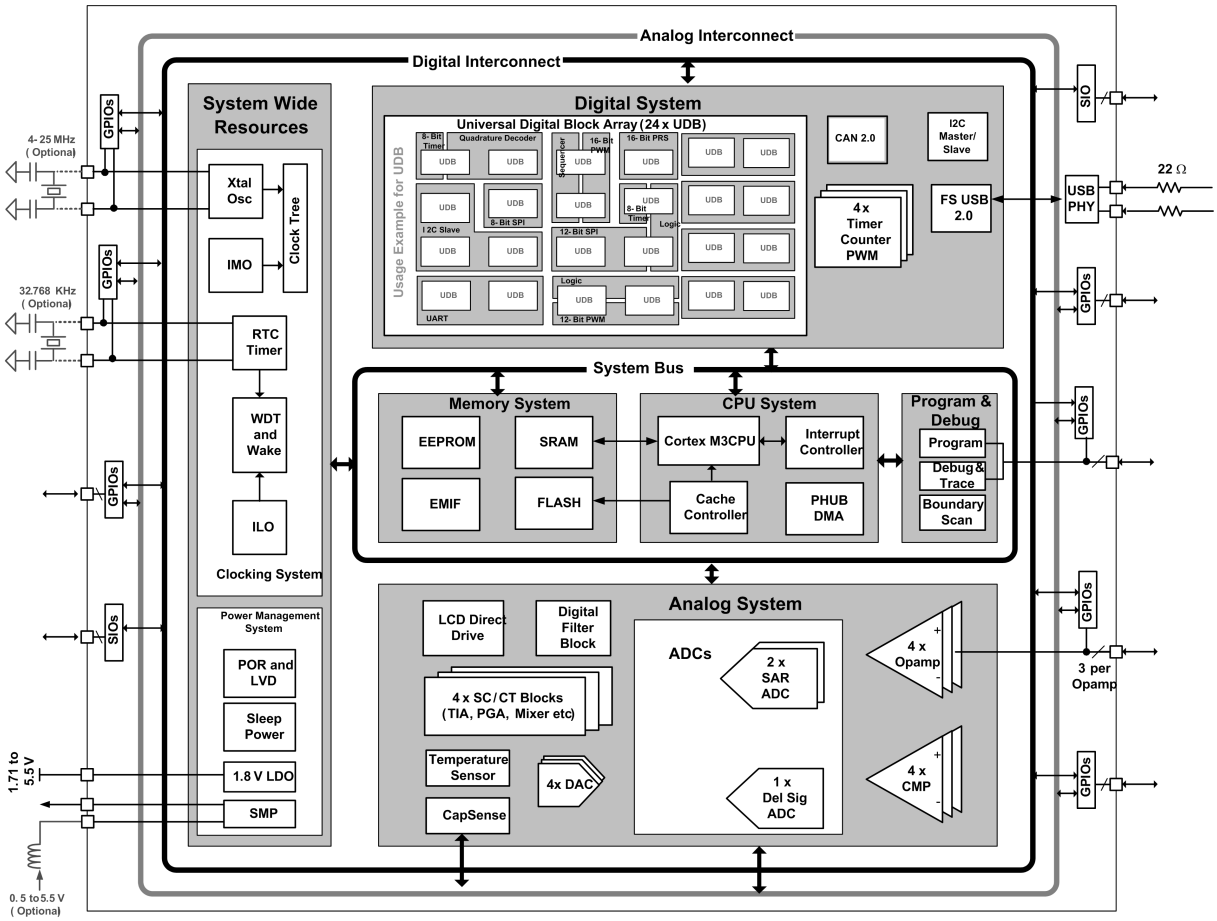


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C5888LTQ-LP097
Package Name	68-QFN
Family	PSoC 5LP
Series	CY8C58LP
Max CPU speed (MHz)	0
Flash size (kB)	256
SRAM size (kB)	64
EEPROM size (bytes)	2048
Vdd range (V)	1.71 to 5.5
Automotive qualified	No (Industrial Grade Only)
Temp range (Celsius)	-40 to 105
JTAG ID	0x2E161069

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

Table 2 lists the device resources that this design uses:

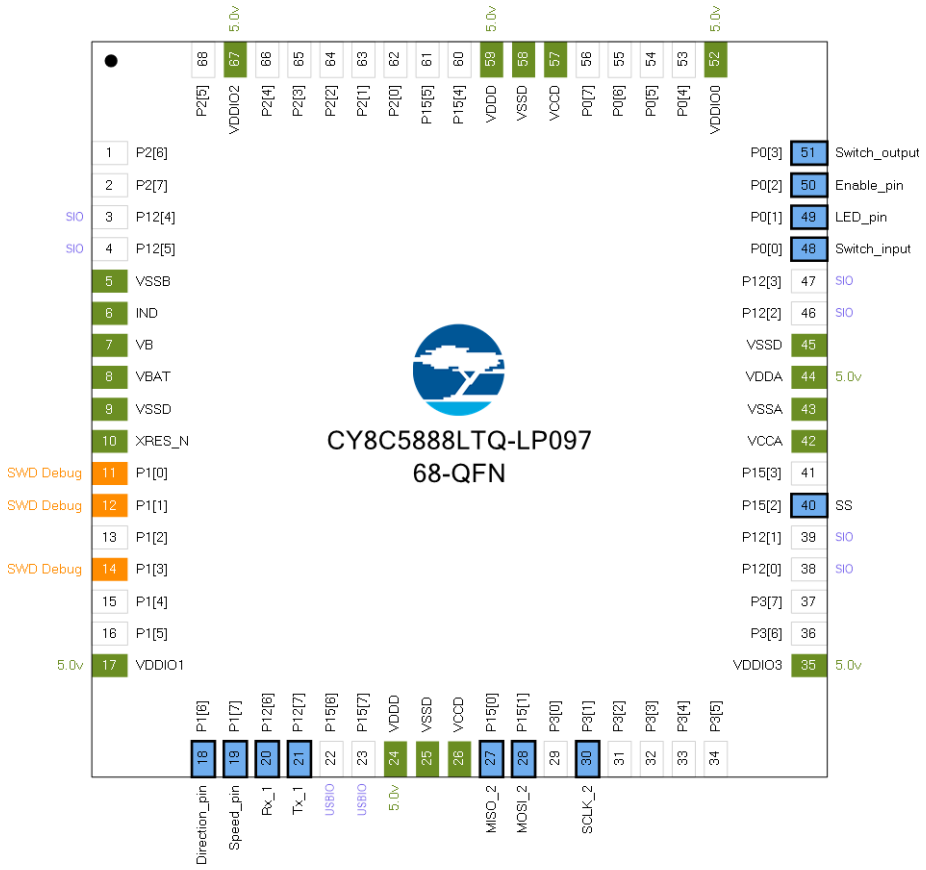
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Digital Clocks	3	5	8	37.50 %
Analog Clocks	0	4	4	0.00 %
CapSense Buffers	0	2	2	0.00 %
Digital Filter Block	0	1	1	0.00 %
Interrupts	3	29	32	9.38 %
IO	15	33	48	31.25 %
Segment LCD	0	1	1	0.00 %
CAN 2.0b	0	1	1	0.00 %
I2C	0	1	1	0.00 %
USB	0	1	1	0.00 %
DMA Channels	0	24	24	0.00 %
Timer	1	3	4	25.00 %
UDB				
Macrocells	43	149	192	22.40 %
Unique P-terms	89	295	384	23.18 %
Total P-terms	101			
Datapath Cells	4	20	24	16.67 %
Status Cells	6	18	24	25.00 %
StatusI Registers	4			
Routed Count7 Load/Enable	2			
Control Cells	2	22	24	8.33 %
Count7 Cells	2			
Opamp	0	4	4	0.00 %
Comparator	0	4	4	0.00 %
Delta-Sigma ADC	0	1	1	0.00 %
LPF	0	2	2	0.00 %
SAR ADC	0	2	2	0.00 %
Analog (SC/CT) Blocks	1	3	4	25.00 %
DAC				
VIDAC	3	1	4	75.00 %

2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



2.1 Hardware Pins

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

Table 3. Device Pins

Pin	Port	Name	Type	Drive Mode	Reset State
1	P2[6]	GPIO [unused]			HiZ Analog Unb
2	P2[7]	GPIO [unused]			HiZ Analog Unb
3	P12[4]	SIO [unused]			HiZ Analog Unb
4	P12[5]	SIO [unused]			HiZ Analog Unb
5	VSSB	VSSB	Dedicated		
6	IND	IND	Dedicated		
7	VB	VB	Dedicated		
8	VBAT	VBAT	Dedicated		
9	VSSD	VSSD	Power		
10	XRES_N	XRES_N	Dedicated		
11	P1[0]	Debug:SWD_IO	Reserved		
12	P1[1]	Debug:SWD_CK	Reserved		
13	P1[2]	GPIO [unused]			HiZ Analog Unb
14	P1[3]	Debug:SWV	Reserved		
15	P1[4]	GPIO [unused]			HiZ Analog Unb
16	P1[5]	GPIO [unused]			HiZ Analog Unb
17	VDDIO1	VDDIO1	Power		
18	P1[6]	Direction_pin	Analog	HiZ analog	HiZ Analog Unb
19	P1[7]	Speed_pin	Analog	HiZ analog	HiZ Analog Unb
20	P12[6]	Rx_1	Dgtl In	HiZ digital	HiZ Analog Unb
21	P12[7]	Tx_1	Dgtl Out	Strong drive	HiZ Analog Unb
22	P15[6]	USB IO [unused]			HiZ Analog Unb
23	P15[7]	USB IO [unused]			HiZ Analog Unb
24	VDDD	VDDD	Power		
25	VSSD	VSSD	Power		
26	VCCD	VCCD	Power		
27	P15[0]	MISO_2	Dgtl In	HiZ digital	HiZ Analog Unb
28	P15[1]	MOSI_2	Dgtl Out	Strong drive	HiZ Analog Unb
29	P3[0]	GPIO [unused]			HiZ Analog Unb
30	P3[1]	SCLK_2	Dgtl Out	Strong drive	HiZ Analog Unb
31	P3[2]	GPIO [unused]			HiZ Analog Unb
32	P3[3]	GPIO [unused]			HiZ Analog Unb
33	P3[4]	GPIO [unused]			HiZ Analog Unb
34	P3[5]	GPIO [unused]			HiZ Analog Unb
35	VDDIO3	VDDIO3	Power		
36	P3[6]	GPIO [unused]			HiZ Analog Unb
37	P3[7]	GPIO [unused]			HiZ Analog Unb
38	P12[0]	SIO [unused]			HiZ Analog Unb
39	P12[1]	SIO [unused]			HiZ Analog Unb
40	P15[2]	SS	Software In/Out	Strong drive	HiZ Analog Unb
41	P15[3]	GPIO [unused]			HiZ Analog Unb
42	VCCA	VCCA	Power		
43	VSSA	VSSA	Power		
44	VDDA	VDDA	Power		
45	VSSD	VSSD	Power		

Pin	Port	Name	Type	Drive Mode	Reset State
46	P12[2]	SIO [unused]			HiZ Analog Unb
47	P12[3]	SIO [unused]			HiZ Analog Unb
48	P0[0]	Switch_input	Software In/Out	Res pull down	HiZ Analog Unb
49	P0[1]	LED_pin	Software In/Out	Res pull down	HiZ Analog Unb
50	P0[2]	Enable_pin	Analog	HiZ analog	HiZ Analog Unb
51	P0[3]	Switch_output	Software In/Out	Res pull down	HiZ Analog Unb
52	VDDIO0	VDDIO0	Power		
53	P0[4]	GPIO [unused]			HiZ Analog Unb
54	P0[5]	GPIO [unused]			HiZ Analog Unb
55	P0[6]	GPIO [unused]			HiZ Analog Unb
56	P0[7]	GPIO [unused]			HiZ Analog Unb
57	VCCD	VCCD	Power		
58	VSSD	VSSD	Power		
59	VDDD	VDDD	Power		
60	P15[4]	GPIO [unused]			HiZ Analog Unb
61	P15[5]	GPIO [unused]			HiZ Analog Unb
62	P2[0]	GPIO [unused]			HiZ Analog Unb
63	P2[1]	GPIO [unused]			HiZ Analog Unb
64	P2[2]	GPIO [unused]			HiZ Analog Unb
65	P2[3]	GPIO [unused]			HiZ Analog Unb
66	P2[4]	GPIO [unused]			HiZ Analog Unb
67	VDDIO2	VDDIO2	Power		
68	P2[5]	GPIO [unused]			HiZ Analog Unb

Abbreviations used in Table 3 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- HiZ analog = High impedance analog
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Dgtl Out = Digital Output
- Res pull down = Resistive pull down

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	Reset State
P0[0]	48	Switch_input	Software In/Out	Res pull down	HiZ Analog Unb
P0[1]	49	LED_pin	Software In/Out	Res pull down	HiZ Analog Unb
P0[2]	50	Enable_pin	Analog	HiZ analog	HiZ Analog Unb
P0[3]	51	Switch_output	Software In/Out	Res pull down	HiZ Analog Unb
P0[4]	53	GPIO [unused]			HiZ Analog Unb
P0[5]	54	GPIO [unused]			HiZ Analog Unb
P0[6]	55	GPIO [unused]			HiZ Analog Unb
P0[7]	56	GPIO [unused]			HiZ Analog Unb
P1[0]	11	Debug:SWD_IO	Reserved		
P1[1]	12	Debug:SWD_CK	Reserved		
P1[2]	13	GPIO [unused]			HiZ Analog Unb
P1[3]	14	Debug:SWV	Reserved		
P1[4]	15	GPIO [unused]			HiZ Analog Unb
P1[5]	16	GPIO [unused]			HiZ Analog Unb
P1[6]	18	Direction_pin	Analog	HiZ analog	HiZ Analog Unb
P1[7]	19	Speed_pin	Analog	HiZ analog	HiZ Analog Unb
P12[0]	38	SIO [unused]			HiZ Analog Unb
P12[1]	39	SIO [unused]			HiZ Analog Unb
P12[2]	46	SIO [unused]			HiZ Analog Unb
P12[3]	47	SIO [unused]			HiZ Analog Unb
P12[4]	3	SIO [unused]			HiZ Analog Unb
P12[5]	4	SIO [unused]			HiZ Analog Unb
P12[6]	20	Rx_1	Dgtl In	HiZ digital	HiZ Analog Unb
P12[7]	21	Tx_1	Dgtl Out	Strong drive	HiZ Analog Unb
P15[0]	27	MISO_2	Dgtl In	HiZ digital	HiZ Analog Unb
P15[1]	28	MOSI_2	Dgtl Out	Strong drive	HiZ Analog Unb
P15[2]	40	SS	Software In/Out	Strong drive	HiZ Analog Unb
P15[3]	41	GPIO [unused]			HiZ Analog Unb
P15[4]	60	GPIO [unused]			HiZ Analog Unb
P15[5]	61	GPIO [unused]			HiZ Analog Unb
P15[6]	22	USB IO [unused]			HiZ Analog Unb
P15[7]	23	USB IO [unused]			HiZ Analog Unb
P2[0]	62	GPIO [unused]			HiZ Analog Unb
P2[1]	63	GPIO [unused]			HiZ Analog Unb
P2[2]	64	GPIO [unused]			HiZ Analog Unb
P2[3]	65	GPIO [unused]			HiZ Analog Unb
P2[4]	66	GPIO [unused]			HiZ Analog Unb
P2[5]	68	GPIO [unused]			HiZ Analog Unb
P2[6]	1	GPIO [unused]			HiZ Analog Unb
P2[7]	2	GPIO [unused]			HiZ Analog Unb
P3[0]	29	GPIO [unused]			HiZ Analog Unb
P3[1]	30	SCLK_2	Dgtl Out	Strong drive	HiZ Analog Unb

Port	Pin	Name	Type	Drive Mode	Reset State
P3[2]	31	GPIO [unused]			HiZ Analog Unb
P3[3]	32	GPIO [unused]			HiZ Analog Unb
P3[4]	33	GPIO [unused]			HiZ Analog Unb
P3[5]	34	GPIO [unused]			HiZ Analog Unb
P3[6]	36	GPIO [unused]			HiZ Analog Unb
P3[7]	37	GPIO [unused]			HiZ Analog Unb

Abbreviations used in Table 4 have the following meanings:

- Res pull down = Resistive pull down
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- HiZ analog = High impedance analog
- Dgtl In = Digital Input
- HiZ digital = High impedance digital
- Dgtl Out = Digital Output

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	Reset State
Debug:SWD_CK	P1[1]	Reserved	
Debug:SWD_IO	P1[0]	Reserved	
Debug:SWV	P1[3]	Reserved	
Direction_pin	P1[6]	Analog	HiZ Analog Unb
Enable_pin	P0[2]	Analog	HiZ Analog Unb
GPIO [unused]	P0[5]		HiZ Analog Unb
GPIO [unused]	P15[3]		HiZ Analog Unb
GPIO [unused]	P0[4]		HiZ Analog Unb
GPIO [unused]	P3[5]		HiZ Analog Unb
GPIO [unused]	P3[4]		HiZ Analog Unb
GPIO [unused]	P3[3]		HiZ Analog Unb
GPIO [unused]	P3[7]		HiZ Analog Unb
GPIO [unused]	P3[2]		HiZ Analog Unb
GPIO [unused]	P3[6]		HiZ Analog Unb
GPIO [unused]	P2[2]		HiZ Analog Unb
GPIO [unused]	P2[1]		HiZ Analog Unb
GPIO [unused]	P2[3]		HiZ Analog Unb
GPIO [unused]	P2[5]		HiZ Analog Unb
GPIO [unused]	P2[4]		HiZ Analog Unb
GPIO [unused]	P0[7]		HiZ Analog Unb
GPIO [unused]	P0[6]		HiZ Analog Unb
GPIO [unused]	P15[4]		HiZ Analog Unb
GPIO [unused]	P2[0]		HiZ Analog Unb
GPIO [unused]	P15[5]		HiZ Analog Unb
GPIO [unused]	P1[2]		HiZ Analog Unb
GPIO [unused]	P1[4]		HiZ Analog Unb
GPIO [unused]	P1[5]		HiZ Analog Unb
GPIO [unused]	P2[6]		HiZ Analog Unb
GPIO [unused]	P3[0]		HiZ Analog Unb
GPIO [unused]	P2[7]		HiZ Analog Unb
LED_pin	P0[1]	Software In/Out	HiZ Analog Unb
MISO_2	P15[0]	Dgtl In	HiZ Analog Unb
MOSI_2	P15[1]	Dgtl Out	HiZ Analog Unb
Rx_1	P12[6]	Dgtl In	HiZ Analog Unb
SCLK_2	P3[1]	Dgtl Out	HiZ Analog Unb
SIO [unused]	P12[4]		HiZ Analog Unb
SIO [unused]	P12[5]		HiZ Analog Unb
SIO [unused]	P12[1]		HiZ Analog Unb
SIO [unused]	P12[3]		HiZ Analog Unb
SIO [unused]	P12[2]		HiZ Analog Unb
SIO [unused]	P12[0]		HiZ Analog Unb
Speed_pin	P1[7]	Analog	HiZ Analog Unb
SS	P15[2]	Software In/Out	HiZ Analog Unb

Name	Port	Type	Reset State
Switch_input	P0[0]	Software In/Out	HiZ Analog Unb
Switch_output	P0[3]	Software In/Out	HiZ Analog Unb
Tx_1	P12[7]	Dgtl Out	HiZ Analog Unb
USB IO [unused]	P15[7]		HiZ Analog Unb
USB IO [unused]	P15[6]		HiZ Analog Unb

Abbreviations used in Table 5 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl In = Digital Input
- Dgtl Out = Digital Output

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

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

3 System Settings

3.1 System Configuration

Table 6. System Configuration Settings

Name	Value
Device Configuration Mode	Compressed
Enable Error Correcting Code (ECC)	False
Store Configuration Data in ECC Memory	True
Instruction Cache Enabled	True
Enable Fast IMO During Startup	True
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x200
Stack Size (bytes)	0x0800
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 7. System Debug Settings

Name	Value
Debug Select	SWD+SWV (serial wire debug and viewer)
Enable Device Protection	False
Embedded Trace (ETM)	False
Use Optional XRES	False

3.3 System Operating Conditions

Table 8. System Operating Conditions

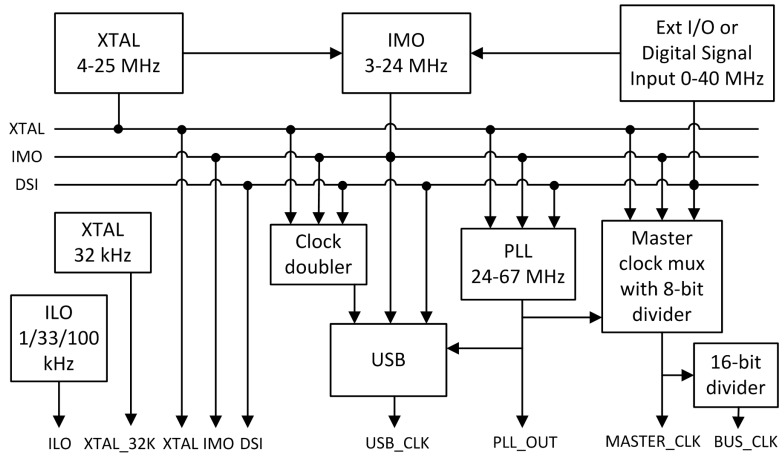
Name	Value
VDDA (V)	5.0
VDDD (V)	5.0
VDDIO0 (V)	5.0
VDDIO1 (V)	5.0
VDDIO2 (V)	5.0
VDDIO3 (V)	5.0
Variable VDDA	False
Temperature Range	-40C - 85/125C

4 Clocks

The clock system includes these clock resources:

- Four internal clock sources increase system integration:
  - 3 to 74.7 MHz Internal Main Oscillator (IMO) ±1% at 3 MHz
  - 1 kHz, 33 kHz, and 100 kHz Internal Low Speed Oscillator (ILO) outputs
  - 12 to 80 MHz clock doubler output, sourced from IMO, MHz External Crystal Oscillator (MHzECO), and Digital System Interconnect (DSI)
  - 24 to 80 MHz fractional Phase-Locked Loop (PLL) sourced from IMO, MHzECO, and DSI
- Clock generated using a DSI signal from an external I/O pin or other logic
- Two external clock sources provide high precision clocks:
  - 4 to 25 MHz External Crystal Oscillator (MHzECO)
  - 32.768 kHz External Crystal Oscillator (kHzECO) for Real Time Clock (RTC)
- Dedicated 16-bit divider for bus clock
- Eight individually sourced 16-bit clock dividers for the digital system peripherals
- Four individually sourced 16-bit clock dividers with skew for the analog system peripherals
- IMO has a USB mode that synchronizes to USB host traffic, requiring no external crystal for USB. (USB equipped parts only)

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
BUS_CLK	DIGITAL	MASTER_CLK	? MHz	24 MHz	±1	True	True
PLL_OUT	DIGITAL	IMO	24 MHz	24 MHz	±1	True	True
MASTER_CLK	DIGITAL	PLL_OUT	? MHz	24 MHz	±1	True	True
IMO	DIGITAL		3 MHz	3 MHz	±1	True	True
ILO	DIGITAL		? MHz	1 kHz	-50,+100	True	True
USB_CLK	DIGITAL	IMO	48 MHz	? MHz	±0	False	False
XTAL	DIGITAL		24 MHz	? MHz	±0	False	False
XTAL 32kHz	DIGITAL		32.768 kHz	? MHz	±0	False	False
Digital Signal	DIGITAL		? MHz	? MHz	±0	False	False

4.2 Local and Design Wide Clocks

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

Figure 4. Local and Design Wide Clock Configuration

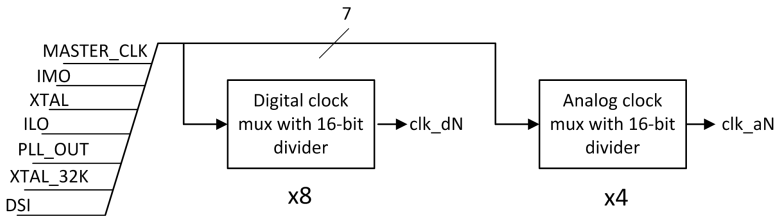


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
SPIM_1_-IntClock	DIGITAL	MASTER_CLK	2 MHz	2 MHz	±1	True	True
UART_1_-IntClock	DIGITAL	MASTER_CLK	460.8 kHz	461.538 kHz	±1	True	True
Clock_-TempSampling	DIGITAL	MASTER_CLK	1 kHz	1 kHz	±1	True	True

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 5LP Technical Reference Manual
- Clocking chapter in the System Reference Guide
  - CyPLL API routines
  - CyIMO API routines
  - CyILO API routines
  - CyMaster API routines

- CyXTAL 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	Intr Num	Vector	Priority
isr_uart_rx	0	0	7
switch_input_interrupt	4	4	7
isr_clock	17	17	7

For more information on interrupts, please refer to:

- Interrupt Controller chapter in the PSoC 5LP 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 5LP 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
- F - Factory Upgrade
- R - Field Upgrade
- W - Full Protection

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

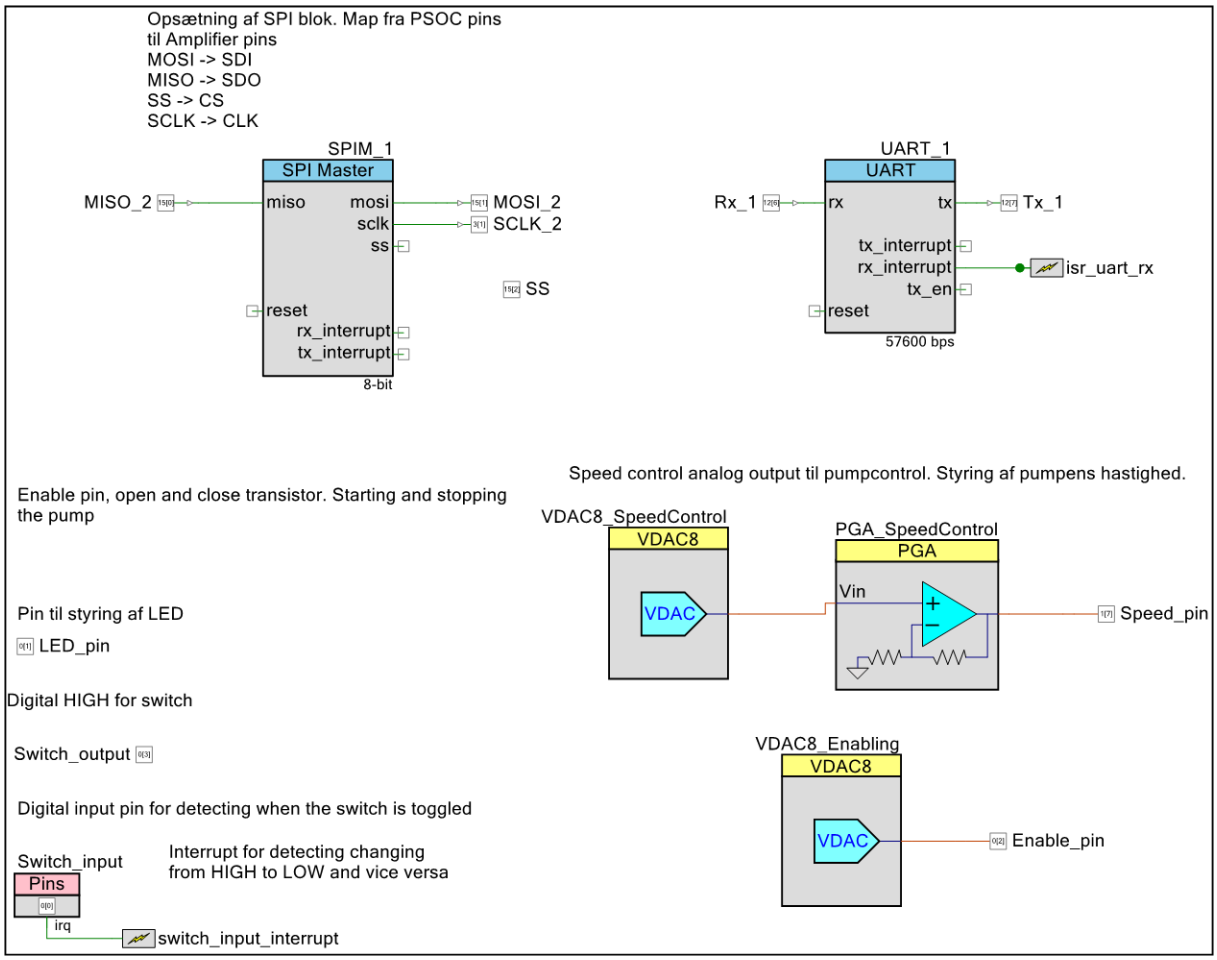
- Flash Protection chapter in the PSoC 5LP Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyWrite API routines
  - CyFlash API routines

7 Design Contents

This design's schematic content consists of the following 2 schematic sheets:

7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1

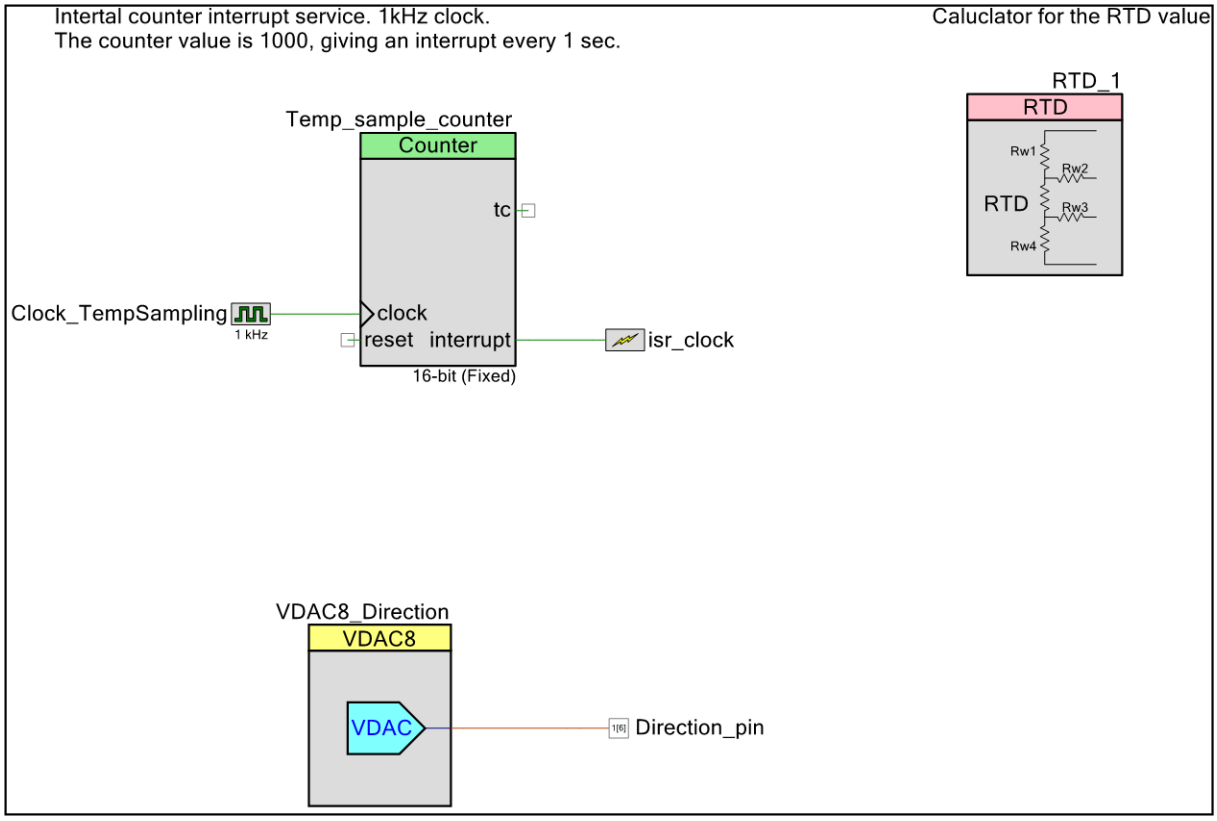


This schematic sheet contains the following component instances:

- Instance PGA_SpeedControl (type: PGA_v2_0)
- Instance SPIM_1 (type: SPI_Master_v2_50)
- Instance UART_1 (type: UART_v2_50)
- Instance VDAC8_Enabling (type: VDAC8_v1_90)
- Instance VDAC8_SpeedControl (type: VDAC8_v1_90)

7.2 Schematic Sheet: Page 2

Figure 6. Schematic Sheet: Page 2



This schematic sheet contains the following component instances:

- Instance RTD_1 (type: RTDCalc_v1_20)
- Instance Temp_sample_counter (type: Counter_v3_0)
- Instance VDAC8_Direction (type: VDAC8_v1_90)

8 Components

8.1 Component type: Counter [v3.0]

8.1.1 Instance Temp_sample_counter

Description: 8, 16, 24 or 32-bit Counter
Instance type: Counter [v3.0]
Datasheet: online component datasheet for Counter

Table 13. Component Parameters for Temp_sample_counter

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the capture input. Default is None which does not have a capture input pin
ClockMode	Up Counter	Defines the operation of the counter. \nBasic: Count is incremented on the rising edge of the clock input. \n Clock_-And_Direction: Clock is incremented or decremented on the rising edge of the clock input based on the direction of the input. \nClock_And_UpCnt_-DwnCnt: Clock is an oversampling clock. On the rising edge of UpCnt, the counter is incremented and on the rising edge of DwnCnt, the counter is decremented.
CompareMode	Less Than	Specifies the compare output mode.
CompareStatusEdgeSense	true	Specifies whether rising edge sense for interrupt generation with the Compare output will be used. May be disabled to reduce resource usage.
CompareValue	1	Defines the compare value. Valid vales are from 0 to the period value.
EnableMode	Software Only	Choose which enable controls the enable of the counter. This can be either through software with the control register, through hardware with the input pin or a combination of both where both must be active for the counter to be enabled.
FixedFunction	true	Defines whether Fixed Function Block usage is required.
InterruptOnCapture	false	Enables the counter status register to produce an interrupt output signal on a capture event.

Parameter Name	Value	Description
InterruptOnCompare	false	Enables the counter status register to produce an interrupt output signal on compare true.
InterruptOnOverUnderFlow	false	Enables the counter status register to produce an interrupt output signal on over flow or under flow.
InterruptOnTC	true	Enables the counter status register to produce an interrupt output signal on terminal count.
Period	1000	Defines the counter period value in clock counts from 1 to 2^Width-1.
ReloadOnCapture	false	Reloads the counter value to a set value on a capture input event.
ReloadOnCompare	false	Reloads the counter value to a set value on a compare equal event.
ReloadOnOverUnder	true	Reloads the counter value to a set value when overflow or underflow is detected.
ReloadOnReset	true	Reloads the counter value to a set value when reset input is high.
Resolution	16	Defines the width of the counter. It can be 8, 16, 24 or 32 (24 or 32 cannot use Fixed Function block).
RunMode	Continuous	Define the hardware operation to run continuously or run till a terminal count.
UseInterrupt	true	Allows for complete optimization of resource usage down to removing the status register if not required by the user.
User Comments		Instance-specific comments.

8.2 Component type: PGA [v2.0]

8.2.1 Instance PGA_SpeedControl

Description: Programmable Gain Amplifier
Instance type: PGA [v2.0]
Datasheet: online component datasheet for PGA

Table 14. Component Parameters for PGA_SpeedControl

Parameter Name	Value	Description
Gain	2	Selects supported gain value.
Power	Low Power	Selects the device power.
User Comments		Instance-specific comments.
Vref_Input	Internal Vss	Enables direct connection from the Analog ground (Agnd) to the inverting input.

8.3 Component type: RTDCalc [v1.20]

8.3.1 Instance RTD_1

Description: Resistance Temperature Detector Calculator
Instance type: RTDCalc [v1.20]
Datasheet: online component datasheet for RTDCalc

Table 15. Component Parameters for RTD_1

Parameter Name	Value	Description
CalcErrBudget	0.01	Choose Calculation Error budget.
Coeff_A	0.0039083	Coefficient A
Coeff_B	-5.775E-07	Coefficient B
Coeff_C	-4.183E-12	Coefficient C
Coefficients	false	Select IEC 60751 or Custom coefficients.
MaxTemp	50	Max Temperature
MinTemp	-50	Min Temperature
RTDType	PT1000	Select RTD Type
User Comments		Instance-specific comments.

8.4 Component type: SPI_Master [v2.50]

8.4.1 Instance SPIM_1

Description: Serial Peripheral Interface Master
Instance type: SPI_Master [v2.50]
Datasheet: online component datasheet for SPI_Master

Table 16. Component Parameters for SPIM_1

Parameter Name	Value	Description
BidirectMode	false	Bidirectional mode setting
ClockInternal	true	Allow use of the internal clock and desired bit rate or an external clock source
DesiredBitRate	1000000	Desired Bit Rate in bps
HighSpeedMode	true	Enables using of the High Speed Mode
InterruptOnByteComplete	false	Set Initial Interrupt Source to Enable Interrupt on Byte Transfer Complete
InterruptOnRXFull	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Full
InterruptOnRXNotEmpty	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Not Empty
InterruptOnRXOverrun	false	Set Initial Interrupt Source to Enable Interrupt on RX FIFO Overrun
InterruptOnSPIDone	false	Set Initial Interrupt Source to Enable Interrupt on SPI Done
InterruptOnSPIIdle	false	Set Initial Interrupt Source to Enable Interrupt on SPI Idle
InterruptOnTXEmpty	false	Set Initial Interrupt Source to Enable Interrupt on TX FIFO Empty

Parameter Name	Value	Description
InterruptOnTXNotFull	false	Set Initial Interrupt Source to Enable Interrupt on TX FIFO Not Full
Mode	CPHA = 1, CPOL = 0	SPI mode defines the Clock Phase and Clock Polarity desired
NumberOfDataBits	8	Set the Number of Data bits 3-16
RxBufferSize	4	Defines the amount of RAM Set asside for the RX Buffer
ShiftDir	MSB First	Set the Shift Out Direction
TxBufferSize	4	Defines the amount of RAM Set asside for the TX Buffer
User Comments		Instance-specific comments.
UseRxInternalInterrupt	false	Defines whether Rx internal interrupt is used or not
UseTxInternalInterrupt	false	Defines whether Tx internal interrupt is used or not

8.5 Component type: UART [v2.50]

8.5.1 Instance UART_1

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

Table 17. Component Parameters for UART_1

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	57600	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.

Parameter Name	Value	Description
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
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

Parameter Name	Value	Description
Use23Polling	true	Allows the use of 2 out of 3 polling resources on the RX UART sampler.
User Comments		Instance-specific comments.

8.6 Component type: VDAC8 [v1.90]

8.6.1 Instance VDAC8_Direction

Description: 8-Bit Voltage DAC
Instance type: VDAC8 [v1.90]
Datasheet: online component datasheet for VDAC8

Table 18. Component Parameters for VDAC8_Direction

Parameter Name	Value	Description
Data_Source	CPU or DMA (Data Bus)	Selects the method in which the data is written to the vDAC.
Initial_Value	0	Configures the initial vDAC output voltage. The output uses the following relation: Initial output voltage = value*(FullRange/255). This calculated output voltage value is invalid if DAC Bus is used.
Strobe_Mode	Register Write	Selects how the data is strobed into the DAC. For a register write, the data is strobed into the DAC on each CPU or DMA write. If operating in External mode, an external data strobe signal is required.
User Comments		Instance-specific comments.
VDAC_Range	0 - 4.080V (16mV/bit)	Specifies the full voltage scale range of the vDAC
VDAC_Speed	Low Speed	Specifies the vDAC settling speed. Note that the 'Slow Speed' selection consumes less power.
Voltage	0	This parameter sets the voltage value.

8.6.2 Instance VDAC8_Enabling

Description: 8-Bit Voltage DAC
Instance type: VDAC8 [v1.90]
Datasheet: online component datasheet for VDAC8

Table 19. Component Parameters for VDAC8_Enabling

Parameter Name	Value	Description
Data_Source	CPU or DMA (Data Bus)	Selects the method in which the data is written to the vDAC.

Parameter Name	Value	Description
Initial_Value	0	Configures the initial vDAC output voltage. The output uses the following relation: Initial output voltage = value*(FullRange/255). This calculated output voltage value is invalid if DAC Bus is used.
Strobe_Mode	Register Write	Selects how the data is strobed into the DAC. For a register write, the data is strobed into the DAC on each CPU or DMA write. If operating in External mode, an external data strobe signal is required.
User Comments		Instance-specific comments.
VDAC_Range	0 - 4.080V (16mV/bit)	Specifies the full voltage scale range of the vDAC
VDAC_Speed	Low Speed	Specifies the vDAC settling speed. Note that the 'Slow Speed' selection consumes less power.
Voltage	0	This parameter sets the voltage value.

8.6.3 Instance VDAC8_SpeedControl

Description: 8-Bit Voltage DAC
Instance type: VDAC8 [v1.90]
Datasheet: online component datasheet for VDAC8

Table 20. Component Parameters for VDAC8_SpeedControl

Parameter Name	Value	Description
Data_Source	CPU or DMA (Data Bus)	Selects the method in which the data is written to the vDAC.
Initial_Value	0	Configures the initial vDAC output voltage. The output uses the following relation: Initial output voltage = value*(FullRange/255). This calculated output voltage value is invalid if DAC Bus is used.
Strobe_Mode	Register Write	Selects how the data is strobed into the DAC. For a register write, the data is strobed into the DAC on each CPU or DMA write. If operating in External mode, an external data strobe signal is required.
User Comments		Instance-specific comments.
VDAC_Range	0 - 4.080V (16mV/bit)	Specifies the full voltage scale range of the vDAC
VDAC_Speed	Low Speed	Specifies the vDAC settling speed. Note that the 'Slow Speed' selection consumes less power.
Voltage	0	This parameter sets the voltage value.

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 5LP register map is covered in the PSoC 5LP 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 5LP Technical Reference Manual
  - Low Power Modes chapter in the PSoC 5LP 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
- Cache Management
  - Cache Controller chapter in the PSoC 5LP Technical Reference Manual
  - Cache chapter in the System Reference Guide
    - CyFlushCache() API routine