PSoC® Creator™
Project Datasheet for Biosensor

Creation Time: 08/24/2024 22:19:16
User: DESKTOP-L29G79U\User
Project: Biosensor
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

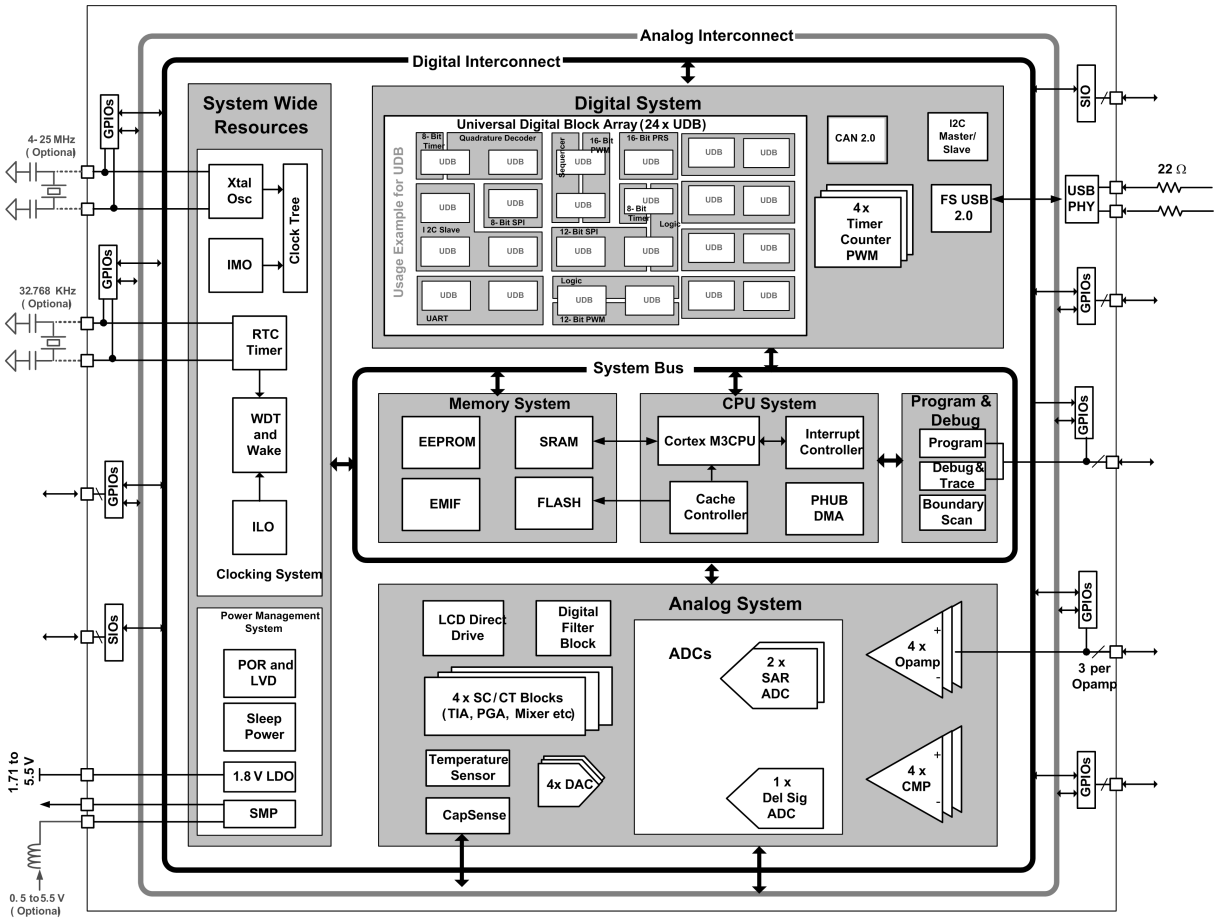


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C5868AXI-LP035
Package Name	100-TQFP
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 85
JTAG ID	0x2E123069

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:

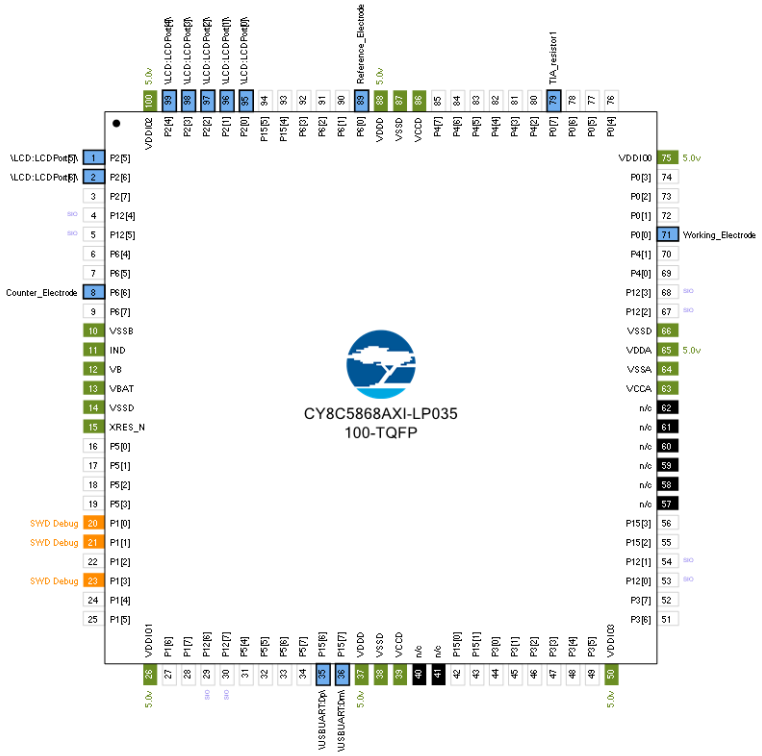
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Digital Clocks	2	6	8	25.00 %
Analog Clocks	1	3	4	25.00 %
CapSense Buffers	0	2	2	0.00 %
Digital Filter Block	0	1	1	0.00 %
Interrupts	12	20	32	37.50 %
IO	19	53	72	26.39 %
Segment LCD	0	1	1	0.00 %
CAN 2.0b	0	1	1	0.00 %
I2C	0	1	1	0.00 %
USB	1	0	1	100.00 %
DMA Channels	0	24	24	0.00 %
Timer	0	4	4	0.00 %
UDB				
Macrocells	4	188	192	2.08 %
Unique P-terms	4	380	384	1.04 %
Total P-terms	4			
Datapath Cells	2	22	24	8.33 %
Status Cells	0	24	24	0.00 %
Control Cells	1	23	24	4.17 %
Control Registers	1			
Opamp	1	3	4	25.00 %
Comparator	0	4	4	0.00 %
Delta-Sigma ADC	1	0	1	100.00 %
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[5]	\LCD:LCDPort[5]\	Software In/Out	Strong drive	HiZ Analog Unb
2	P2[6]	\LCD:LCDPort[6]\	Software In/Out	Strong drive	HiZ Analog Unb
3	P2[7]	GPIO [unused]			HiZ Analog Unb
4	P12[4]	SIO [unused]			HiZ Analog Unb
5	P12[5]	SIO [unused]			HiZ Analog Unb
6	P6[4]	GPIO [unused]			HiZ Analog Unb
7	P6[5]	GPIO [unused]			HiZ Analog Unb
8	P6[6]	Counter_Electrode	Analog	HiZ analog	HiZ Analog Unb
9	P6[7]	GPIO [unused]			HiZ Analog Unb
10	VSSB	VSSB	Dedicated		
11	IND	IND	Dedicated		
12	VB	VB	Dedicated		
13	VBAT	VBAT	Dedicated		
14	VSSD	VSSD	Power		
15	XRES_N	XRES_N	Dedicated		
16	P5[0]	GPIO [unused]			HiZ Analog Unb
17	P5[1]	GPIO [unused]			HiZ Analog Unb
18	P5[2]	GPIO [unused]			HiZ Analog Unb
19	P5[3]	GPIO [unused]			HiZ Analog Unb
20	P1[0]	Debug:SWD_IO	Reserved		
21	P1[1]	Debug:SWD_CK	Reserved		
22	P1[2]	GPIO [unused]			HiZ Analog Unb
23	P1[3]	Debug:SWV	Reserved		
24	P1[4]	GPIO [unused]			HiZ Analog Unb
25	P1[5]	GPIO [unused]			HiZ Analog Unb
26	VDDIO1	VDDIO1	Power		
27	P1[6]	GPIO [unused]	Reserved		
28	P1[7]	GPIO [unused]			HiZ Analog Unb
29	P12[6]	SIO [unused]			HiZ Analog Unb
30	P12[7]	SIO [unused]			HiZ Analog Unb
31	P5[4]	GPIO [unused]			HiZ Analog Unb
32	P5[5]	GPIO [unused]			HiZ Analog Unb
33	P5[6]	GPIO [unused]			HiZ Analog Unb
34	P5[7]	GPIO [unused]			HiZ Analog Unb
35	P15[6]	\USBUART:Dp\	Analog	HiZ analog	HiZ Analog Unb
36	P15[7]	\USBUART:Dm\	Analog	HiZ analog	HiZ Analog Unb
37	VDDD	VDDD	Power		
38	VSSD	VSSD	Power		
39	VCCD	VCCD	Power		
42	P15[0]	GPIO [unused]			HiZ Analog Unb
43	P15[1]	GPIO [unused]			HiZ Analog Unb
44	P3[0]	GPIO [unused]			HiZ Analog Unb
45	P3[1]	GPIO [unused]	Reserved		
46	P3[2]	GPIO [unused]			HiZ Analog Unb

Pin	Port	Name	Type	Drive Mode	Reset State
47	P3[3]	GPIO [unused]			HiZ Analog Unb
48	P3[4]	GPIO [unused]			HiZ Analog Unb
49	P3[5]	GPIO [unused]			HiZ Analog Unb
50	VDDIO3	VDDIO3	Power		
51	P3[6]	GPIO [unused]			HiZ Analog Unb
52	P3[7]	GPIO [unused]	Analog	HiZ analog	HiZ Analog Unb
53	P12[0]	SIO [unused]			HiZ Analog Unb
54	P12[1]	SIO [unused]			HiZ Analog Unb
55	P15[2]	GPIO [unused]			HiZ Analog Unb
56	P15[3]	GPIO [unused]			HiZ Analog Unb
63	VCCA	VCCA	Power		
64	VSSA	VSSA	Power		
65	VDDA	VDDA	Power		
66	VSSD	VSSD	Power		
67	P12[2]	SIO [unused]			HiZ Analog Unb
68	P12[3]	SIO [unused]			HiZ Analog Unb
69	P4[0]	GPIO [unused]			HiZ Analog Unb
70	P4[1]	GPIO [unused]			HiZ Analog Unb
71	P0[0]	Working_Electrode	Analog	HiZ analog	HiZ Analog Unb
72	P0[1]	GPIO [unused]			HiZ Analog Unb
73	P0[2]	GPIO [unused]			HiZ Analog Unb
74	P0[3]	GPIO [unused]			HiZ Analog Unb
75	VDDIO0	VDDIO0	Power		
76	P0[4]	GPIO [unused]			HiZ Analog Unb
77	P0[5]	GPIO [unused]			HiZ Analog Unb
78	P0[6]	GPIO [unused]			HiZ Analog Unb
79	P0[7]	TIA_resistor1	Analog	HiZ analog	HiZ Analog Unb
80	P4[2]	GPIO [unused]			HiZ Analog Unb
81	P4[3]	GPIO [unused]			HiZ Analog Unb
82	P4[4]	GPIO [unused]			HiZ Analog Unb
83	P4[5]	GPIO [unused]			HiZ Analog Unb
84	P4[6]	GPIO [unused]			HiZ Analog Unb
85	P4[7]	GPIO [unused]			HiZ Analog Unb
86	VCCD	VCCD	Power		
87	VSSD	VSSD	Power		
88	VDDD	VDDD	Power		
89	P6[0]	Reference_Electrode	Analog	HiZ analog	HiZ Analog Unb
90	P6[1]	GPIO [unused]			HiZ Analog Unb
91	P6[2]	GPIO [unused]			HiZ Analog Unb
92	P6[3]	GPIO [unused]			HiZ Analog Unb
93	P15[4]	GPIO [unused]			HiZ Analog Unb
94	P15[5]	GPIO [unused]			HiZ Analog Unb
95	P2[0]	\LCD:LCDPort[0]\	Software In/Out	Strong drive	HiZ Analog Unb
96	P2[1]	\LCD:LCDPort[1]\	Software In/Out	Strong drive	HiZ Analog Unb
97	P2[2]	\LCD:LCDPort[2]\	Software In/Out	Strong drive	HiZ Analog Unb
98	P2[3]	\LCD:LCDPort[3]\	Software In/Out	Strong drive	HiZ Analog Unb
99	P2[4]	\LCD:LCDPort[4]\	Software In/Out	Strong drive	HiZ Analog Unb
100	VDDIO2	VDDIO2	Power		

Abbreviations used in Table 3 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- HiZ analog = High impedance analog

2.2 Hardware Ports

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

Table 4. Device Ports

Port	Pin	Name	Type	Drive Mode	Reset State
P0[0]	71	Working_Electrode	Analog	HiZ analog	HiZ Analog Unb
P0[1]	72	GPIO [unused]			HiZ Analog Unb
P0[2]	73	GPIO [unused]			HiZ Analog Unb
P0[3]	74	GPIO [unused]			HiZ Analog Unb
P0[4]	76	GPIO [unused]			HiZ Analog Unb
P0[5]	77	GPIO [unused]			HiZ Analog Unb
P0[6]	78	GPIO [unused]			HiZ Analog Unb
P0[7]	79	TIA_resistor1	Analog	HiZ analog	HiZ Analog Unb
P1[0]	20	Debug:SWD_IO	Reserved		
P1[1]	21	Debug:SWD_CK	Reserved		
P1[2]	22	GPIO [unused]			HiZ Analog Unb
P1[3]	23	Debug:SWV	Reserved		
P1[4]	24	GPIO [unused]			HiZ Analog Unb
P1[5]	25	GPIO [unused]			HiZ Analog Unb
P1[6]	27	GPIO [unused]	Reserved		
P1[7]	28	GPIO [unused]			HiZ Analog Unb
P12[0]	53	SIO [unused]			HiZ Analog Unb
P12[1]	54	SIO [unused]			HiZ Analog Unb
P12[2]	67	SIO [unused]			HiZ Analog Unb
P12[3]	68	SIO [unused]			HiZ Analog Unb
P12[4]	4	SIO [unused]			HiZ Analog Unb
P12[5]	5	SIO [unused]			HiZ Analog Unb
P12[6]	29	SIO [unused]			HiZ Analog Unb
P12[7]	30	SIO [unused]			HiZ Analog Unb
P15[0]	42	GPIO [unused]			HiZ Analog Unb
P15[1]	43	GPIO [unused]			HiZ Analog Unb
P15[2]	55	GPIO [unused]			HiZ Analog Unb
P15[3]	56	GPIO [unused]			HiZ Analog Unb
P15[4]	93	GPIO [unused]			HiZ Analog Unb
P15[5]	94	GPIO [unused]			HiZ Analog Unb
P15[6]	35	\USBUART:Dp\	Analog	HiZ analog	HiZ Analog Unb
P15[7]	36	\USBUART:Dm\	Analog	HiZ analog	HiZ Analog Unb
P2[0]	95	\LCD:LCDPort[0]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[1]	96	\LCD:LCDPort[1]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[2]	97	\LCD:LCDPort[2]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[3]	98	\LCD:LCDPort[3]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[4]	99	\LCD:LCDPort[4]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[5]	1	\LCD:LCDPort[5]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[6]	2	\LCD:LCDPort[6]\	Software In/Out	Strong drive	HiZ Analog Unb

Port	Pin	Name	Type	Drive Mode	Reset State
P2[7]	3	GPIO [unused]			HiZ Analog Unb
P3[0]	44	GPIO [unused]			HiZ Analog Unb
P3[1]	45	GPIO [unused]	Reserved		
P3[2]	46	GPIO [unused]			HiZ Analog Unb
P3[3]	47	GPIO [unused]			HiZ Analog Unb
P3[4]	48	GPIO [unused]			HiZ Analog Unb
P3[5]	49	GPIO [unused]			HiZ Analog Unb
P3[6]	51	GPIO [unused]			HiZ Analog Unb
P3[7]	52	GPIO [unused]	Analog	HiZ analog	HiZ Analog Unb
P4[0]	69	GPIO [unused]			HiZ Analog Unb
P4[1]	70	GPIO [unused]			HiZ Analog Unb
P4[2]	80	GPIO [unused]			HiZ Analog Unb
P4[3]	81	GPIO [unused]			HiZ Analog Unb
P4[4]	82	GPIO [unused]			HiZ Analog Unb
P4[5]	83	GPIO [unused]			HiZ Analog Unb
P4[6]	84	GPIO [unused]			HiZ Analog Unb
P4[7]	85	GPIO [unused]			HiZ Analog Unb
P5[0]	16	GPIO [unused]			HiZ Analog Unb
P5[1]	17	GPIO [unused]			HiZ Analog Unb
P5[2]	18	GPIO [unused]			HiZ Analog Unb
P5[3]	19	GPIO [unused]			HiZ Analog Unb
P5[4]	31	GPIO [unused]			HiZ Analog Unb
P5[5]	32	GPIO [unused]			HiZ Analog Unb
P5[6]	33	GPIO [unused]			HiZ Analog Unb
P5[7]	34	GPIO [unused]			HiZ Analog Unb
P6[0]	89	Reference_Electrode	Analog	HiZ analog	HiZ Analog Unb
P6[1]	90	GPIO [unused]			HiZ Analog Unb
P6[2]	91	GPIO [unused]			HiZ Analog Unb
P6[3]	92	GPIO [unused]			HiZ Analog Unb
P6[4]	6	GPIO [unused]			HiZ Analog Unb
P6[5]	7	GPIO [unused]			HiZ Analog Unb
P6[6]	8	Counter_Electrode	Analog	HiZ analog	HiZ Analog Unb
P6[7]	9	GPIO [unused]			HiZ Analog Unb

Abbreviations used in Table 4 have the following meanings:

- HiZ analog = High impedance analog
- HiZ Analog Unb = Hi-Z Analog Unbuffered

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
\LCD:LCDPort[0]\	P2[0]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[1]\	P2[1]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[2]\	P2[2]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[3]\	P2[3]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[4]\	P2[4]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[5]\	P2[5]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[6]\	P2[6]	Software In/Out	HiZ Analog Unb
\USBUART:Dm\	P15[7]	Analog	HiZ Analog Unb
\USBUART:Dp\	P15[6]	Analog	HiZ Analog Unb
Counter_Electrode	P6[6]	Analog	HiZ Analog Unb
Debug:SWD_CK	P1[1]	Reserved	
Debug:SWD_IO	P1[0]	Reserved	
Debug:SWV	P1[3]	Reserved	
GPIO [unused]	P3[6]		HiZ Analog Unb
GPIO [unused]	P6[1]		HiZ Analog Unb
GPIO [unused]	P15[3]		HiZ Analog Unb
GPIO [unused]	P3[7]	Analog	HiZ Analog Unb
GPIO [unused]	P15[2]		HiZ Analog Unb
GPIO [unused]	P3[5]		HiZ Analog Unb
GPIO [unused]	P6[2]		HiZ Analog Unb
GPIO [unused]	P0[3]		HiZ Analog Unb
GPIO [unused]	P0[2]		HiZ Analog Unb
GPIO [unused]	P0[1]		HiZ Analog Unb
GPIO [unused]	P4[2]		HiZ Analog Unb
GPIO [unused]	P0[6]		HiZ Analog Unb
GPIO [unused]	P0[5]		HiZ Analog Unb
GPIO [unused]	P0[4]		HiZ Analog Unb
GPIO [unused]	P4[5]		HiZ Analog Unb
GPIO [unused]	P4[6]		HiZ Analog Unb
GPIO [unused]	P4[7]		HiZ Analog Unb
GPIO [unused]	P4[4]		HiZ Analog Unb
GPIO [unused]	P4[3]		HiZ Analog Unb
GPIO [unused]	P4[1]		HiZ Analog Unb
GPIO [unused]	P4[0]		HiZ Analog Unb
GPIO [unused]	P3[4]		HiZ Analog Unb
GPIO [unused]	P1[5]		HiZ Analog Unb
GPIO [unused]	P1[4]		HiZ Analog Unb
GPIO [unused]	P1[2]		HiZ Analog Unb
GPIO [unused]	P2[7]		HiZ Analog Unb

Name	Port	Type	Reset State
GPIO [unused]	P1[7]		HiZ Analog Unb
GPIO [unused]	P1[6]	Reserved	
GPIO [unused]	P5[3]		HiZ Analog Unb
GPIO [unused]	P6[7]		HiZ Analog Unb
GPIO [unused]	P6[5]		HiZ Analog Unb
GPIO [unused]	P6[4]		HiZ Analog Unb
GPIO [unused]	P5[2]		HiZ Analog Unb
GPIO [unused]	P5[1]		HiZ Analog Unb
GPIO [unused]	P5[0]		HiZ Analog Unb
GPIO [unused]	P5[4]		HiZ Analog Unb
GPIO [unused]	P3[0]		HiZ Analog Unb
GPIO [unused]	P15[1]		HiZ Analog Unb
GPIO [unused]	P15[0]		HiZ Analog Unb
GPIO [unused]	P3[3]		HiZ Analog Unb
GPIO [unused]	P3[2]		HiZ Analog Unb
GPIO [unused]	P3[1]	Reserved	
GPIO [unused]	P6[3]		HiZ Analog Unb
GPIO [unused]	P5[7]		HiZ Analog Unb
GPIO [unused]	P5[6]		HiZ Analog Unb
GPIO [unused]	P5[5]		HiZ Analog Unb
GPIO [unused]	P15[4]		HiZ Analog Unb
GPIO [unused]	P15[5]		HiZ Analog Unb
Reference_Electrode	P6[0]	Analog	HiZ Analog Unb
SIO [unused]	P12[7]		HiZ Analog Unb
SIO [unused]	P12[6]		HiZ Analog Unb
SIO [unused]	P12[5]		HiZ Analog Unb
SIO [unused]	P12[2]		HiZ Analog Unb
SIO [unused]	P12[3]		HiZ Analog Unb
SIO [unused]	P12[1]		HiZ Analog Unb
SIO [unused]	P12[0]		HiZ Analog Unb
SIO [unused]	P12[4]		HiZ Analog Unb
TIA_resistor1	P0[7]	Analog	HiZ Analog Unb
Working_Electrode	P0[0]	Analog	HiZ Analog Unb

Abbreviations used in Table 5 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered

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	Uncompressed
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)	0x80
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

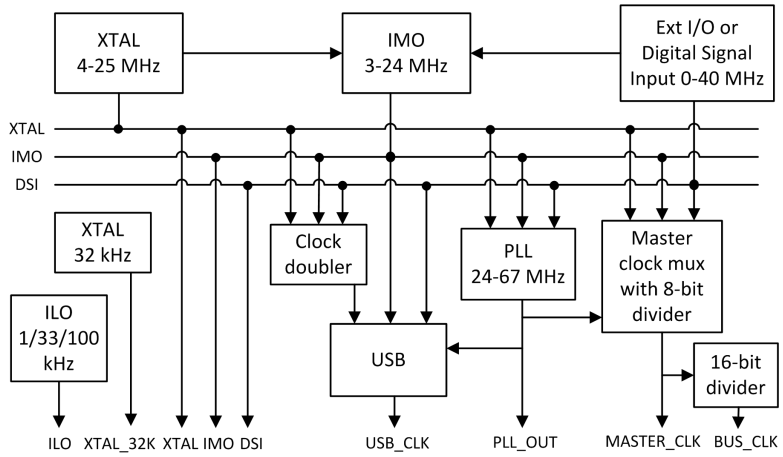
Name	Value
VDDA (V)	5.0
VDDD (V)	5.0
VDDIO0 (V)	5.0
VDDIO1 (V)	5.0
VDDIO2 (V)	5.0
VDDIO3 (V)	5.0
Variable VDDA	False
Temperature Range	0C - 85/125C

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
USB_CLK	DIGITAL	IMO	48 MHz	48 MHz	±0.25	False	True
IMO	DIGITAL		24 MHz	24 MHz	±0.25	True	True
MASTER_CLK	DIGITAL	PLL_OUT	? MHz	24 MHz	±0.25	True	True
BUS_CLK	DIGITAL	MASTER_CLK	? MHz	24 MHz	±0.25	True	True
PLL_OUT	DIGITAL	IMO	24 MHz	24 MHz	±0.25	True	True
ILO	DIGITAL		? MHz	100 kHz	-55,+100	True	True
XTAL 32kHz	DIGITAL		32.768 kHz	? MHz	±0	False	False
Digital Signal	DIGITAL		? MHz	? MHz	±0	False	False
XTAL	DIGITAL		24 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

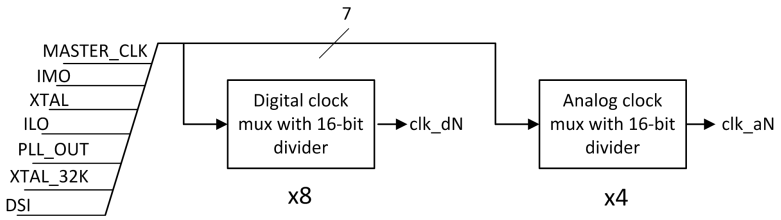


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
ADC_SigDel_-Ext_CP_Clk	DIGITAL	MASTER_CLK	? MHz	24 MHz	±0.25	True	True
ADC_SigDel_-theACLK	ANALOG	MASTER_CLK	1.6 MHz	1.6 MHz	±0.25	True	True
Clock_PWM	DIGITAL	MASTER_CLK	? MHz	240 kHz	±0.25	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
  - o CyPLL API routines
  - o CyIMO API routines
  - o CyILO API routines
  - o CyMaster API routines
  - o 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
USBUART_ep_1	0	0	7
USBUART_ep_2	1	1	7
USBUART_ep_3	2	2	7
isr_adc	3	3	2
isr_adcAmp	4	4	7
isr_dac	5	5	1
USBUART_dp_int	12	12	7
USBUART_sof_int	21	21	7
USBUART_arb_int	22	22	7
USBUART_bus_reset	23	23	7
USBUART_ep_0	24	24	7
ADC_SigDel_IRQ	29	29	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:

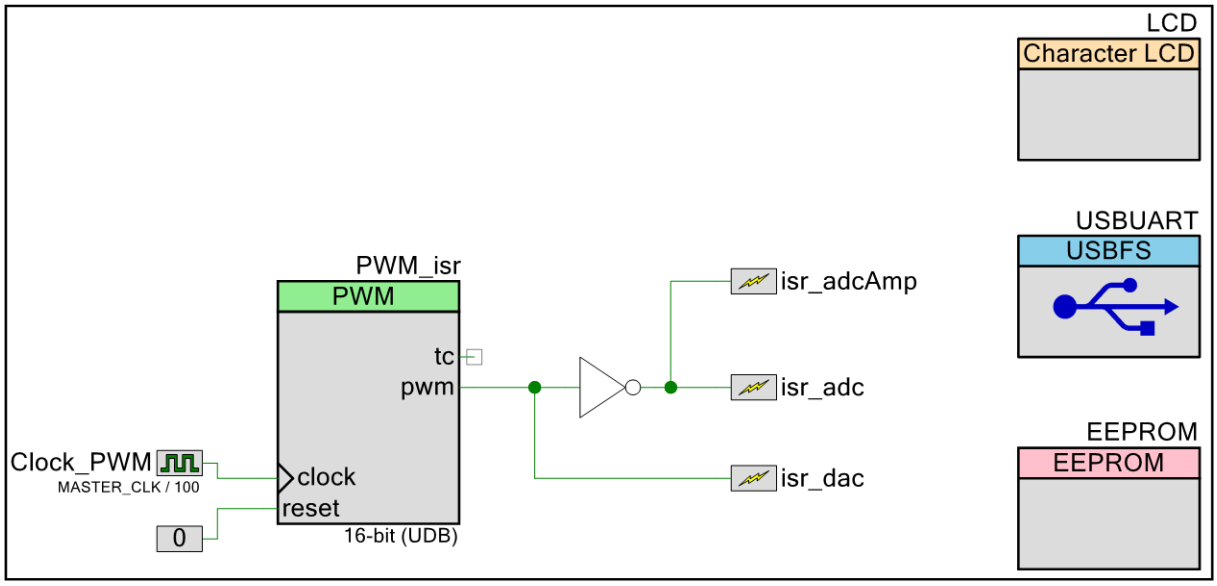
- Flash Protection chapter in the PSoC 5LP Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyWrite API routines
  - CyFlash API routines

7 Design Contents

This design's schematic content consists of the following 3 schematic sheets:

7.1 Schematic Sheet: Timing / Communication

Figure 5. Schematic Sheet: Timing / Communication

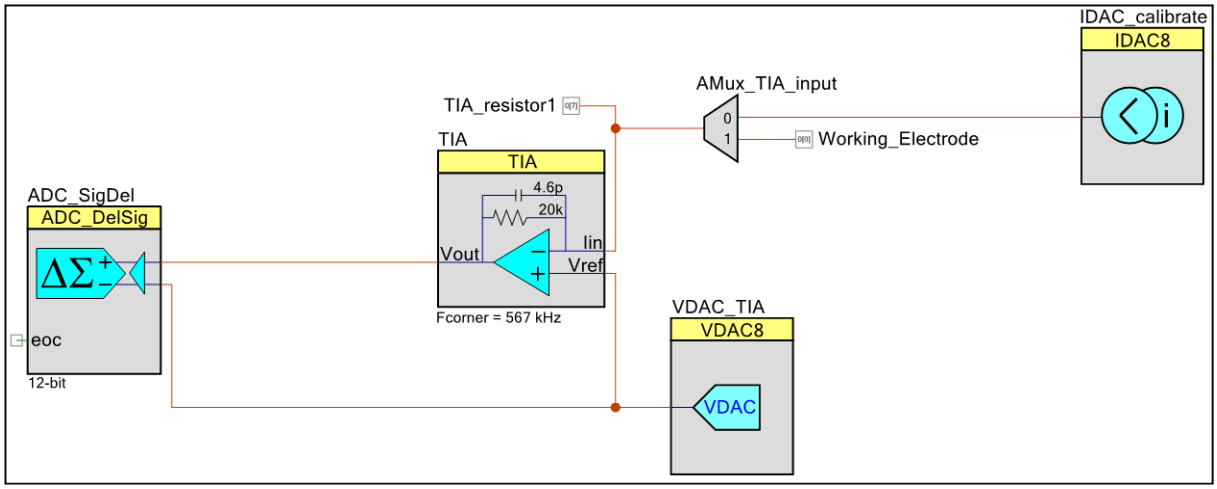


This schematic sheet contains the following component instances:

- Instance EEPROM (type: EEPROM_v3_0)
- Instance LCD (type: CharLCD_v2_20)
- Instance PWM_isr (type: PWM_v3_30)
- Instance USBUART (type: USBFS_v3_20)

7.2 Schematic Sheet: Current Measuring Circuit

Figure 6. Schematic Sheet: Current Measuring Circuit

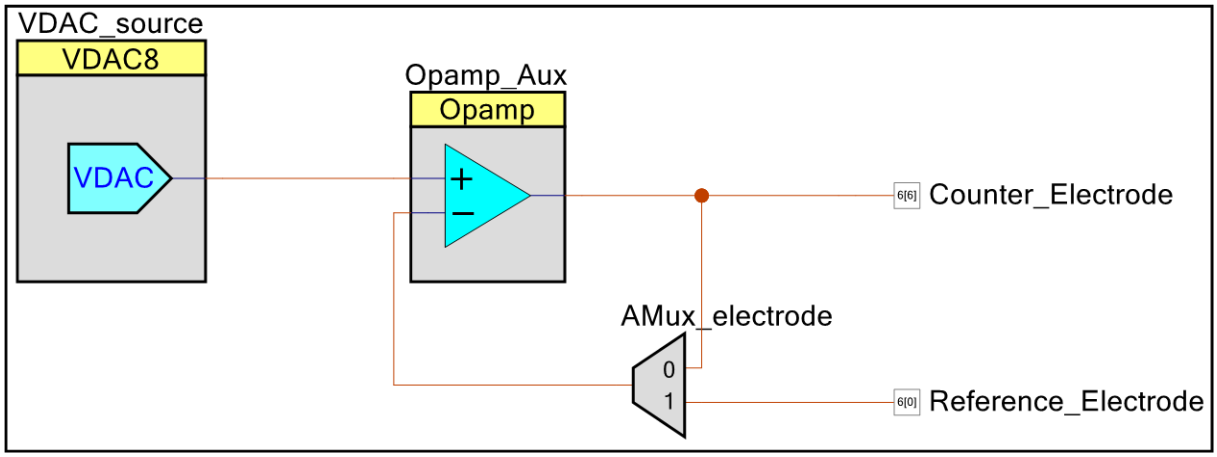


This schematic sheet contains the following component instances:

- Instance ADC_SigDel (type: ADC_DelSig_v3_30)
- Instance AMux_TIA_input (type: AMux_v1_80)
- Instance IDAC_calibrate (type: IDAC8_v2_0)
- Instance TIA (type: TIA_v2_0)
- Instance VDAC_TIA (type: VDAC8_v1_90)

7.3 Schematic Sheet: Voltage Control Circuit

Figure 7. Schematic Sheet: Voltage Control Circuit



This schematic sheet contains the following component instances:

- Instance AMux_electrode (type: AMux_v1_80)
- Instance Opamp_Aux (type: OpAmp_v1_90)
- Instance VDAC_source (type: VDAC8_v1_90)

8 Components

8.1 Component type: ADC_DelSig [v3.30]

8.1.1 Instance ADC_SigDel

Description: Delta-Sigma ADC
Instance type: ADC_DelSig [v3.30]
Datasheet: online component datasheet for ADC_DelSig

Table 13. Component Parameters for ADC_SigDel

Parameter Name	Value	Description
ADC_Alignment	Right	This parameter determines how the result is aligned in the 24 bit result word.
ADC_Alignment_Config2	Right	This parameter determines how the result is aligned in the 24 bit result word.
ADC_Alignment_Config3	Right	This parameter determines how the result is aligned in the 24 bit result word.
ADC_Alignment_Config4	Right	This parameter determines how the result is aligned in the 24 bit result word.
ADC_Charge_Pump_Clock	true	Low power charge pump clock selection
ADC_Clock	Internal	Parameter for selecting the ADC clock type.
ADC_Input_Mode	Differential	Differential or Single ended input mode
ADC_Input_Range	-Input +/- 2*Vref	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config2	-Input +/- Vref	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config3	-Input +/- Vref	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config4	-Input +/- Vref	Choose input operating mode that best supports the range of the signals being measured.
ADC_Power	Medium Power	Sets power level of ADC.
ADC_Reference	Internal 1.024 Volts	Selects voltage reference source and configuration.
ADC_Reference_Config2	Internal 1.024 Volts	Selects voltage reference source and configuration.
ADC_Reference_Config3	Internal 1.024 Volts	Selects voltage reference source and configuration.
ADC_Reference_Config4	Internal 1.024 Volts	Selects voltage reference source and configuration.
ADC_Resolution	12	ADC Resolution in bits
ADC_Resolution_Config2	12	ADC Resolution in bits
ADC_Resolution_Config3	16	ADC Resolution in bits
ADC_Resolution_Config4	16	ADC Resolution in bits

Parameter Name	Value	Description
Clock_Frequency	64000	Determines the ADC clock frequency.
Comment_Config1	Default Config	Parameter which holds the user comment for the config1.
Comment_Config2	Second Config	Parameter which holds the user comment for the config2.
Comment_Config3	Third Config	Parameter which holds the user comment for the config3.
Comment_Config4	Fourth Config	Parameter which holds the user comment for the config4.
Config1_Name	CFG1	This parameter is used to create constants in the header file for config 1.
Config2_Name	CFG2	This parameter is used to create constants in the header file for config 2.
Config3_Name	CFG3	This parameter is used to create constants in the header file for config 3.
Config4_Name	CFG4	This parameter is used to create constants in the header file for config 4.
Configs	2	Number of active configurations
Conversion_Mode	2 - Continuous	ADC conversion mode
Conversion_Mode_Config2	2 - Continuous	ADC conversion mode
Conversion_Mode_Config3	2 - Continuous	ADC conversion mode
Conversion_Mode_Config4	2 - Continuous	ADC conversion mode
Enable_Vref_Vss	false	Determines whether or not to connect ADC's reference Vssa to AGL[6].
EnableModulatorInput	false	When this parameter is enabled, the modulator input terminal will be enabled on the symbol.
Input_Buffer_Gain	1	Gain of input amplifier
Input_Buffer_Gain_Config2	1	Gain of input amplifier
Input_Buffer_Gain_Config3	1	Gain of input amplifier
Input_Buffer_Gain_Config4	1	Gain of input amplifier
Input_Buffer_Mode	Level Shift	Buffer Mode type selection
Input_Buffer_Mode_Config2	Rail to Rail	Buffer Mode type selection
Input_Buffer_Mode_Config3	Rail to Rail	Buffer Mode type selection
Input_Buffer_Mode_Config4	Rail to Rail	Buffer Mode type selection
Ref_Voltage	1.024	Set reference voltage
Ref_Voltage_Config2	1.024	Set reference voltage
Ref_Voltage_Config3	1.024	Set reference voltage
Ref_Voltage_Config4	1.024	Set reference voltage
rm_int	false	Removes internal interrupt (IRQ)
Sample_Rate	50000	Sample Rate in Hz
Sample_Rate_Config2	50000	Sample Rate in Hz
Sample_Rate_Config3	10000	Sample Rate in Hz
Sample_Rate_Config4	10000	Sample Rate in Hz
Start_of_Conversion	Software	Continuous conversions or hardware controlled
User Comments		Instance-specific comments.

8.2 Component type: AMux [v1.80]

8.2.1 Instance AMux_electrode

Description: Multiplexer used to route analog signals.
Instance type: AMux [v1.80]
Datasheet: online component datasheet for AMux

Table 14. Component Parameters for AMux_electrode

Parameter Name	Value	Description
AtMostOneActive	false	Limit to at most one active channel.
Channels	2	Channel count.
Isolation	Minimum	Specify minimum, medium, or maximum switch control; affects channel isolation and switching time.
MuxType	Single	Select between single or differential inputs.
User Comments		Instance-specific comments.

8.2.2 Instance AMux_TIA_input

Description: Multiplexer used to route analog signals.
Instance type: AMux [v1.80]
Datasheet: online component datasheet for AMux

Table 15. Component Parameters for AMux_TIA_input

Parameter Name	Value	Description
AtMostOneActive	false	Limit to at most one active channel.
Channels	2	Channel count.
Isolation	Medium	Specify minimum, medium, or maximum switch control; affects channel isolation and switching time.
MuxType	Single	Select between single or differential inputs.
User Comments		Instance-specific comments.

8.3 Component type: CharLCD [v2.20]

8.3.1 Instance LCD

Description: Character LCD Component
Instance type: CharLCD [v2.20]
Datasheet: online component datasheet for CharLCD

Table 16. Component Parameters for LCD

Parameter Name	Value	Description
ConversionRoutines	true	Defines if the conversion routines will be included in the project.

Parameter Name	Value	Description
CustomCharacterSet	None	Defines the type of custom character set (User defined, Vertical or Horizontal bargraph). Based on the selection a look-up table with proper characters representation will be generated in the source code.
User Comments		Instance-specific comments.

8.4 Component type: EEPROM [v3.0]

8.4.1 Instance EEPROM

Description: Provides an API to Erase and Write EEPROM.
Instance type: EEPROM [v3.0]
Datasheet: online component datasheet for EEPROM

Table 17. Component Parameters for EEPROM

Parameter Name	Value	Description
User Comments		Instance-specific comments.

8.5 Component type: IDAC8 [v2.0]

8.5.1 Instance IDAC_calibrate

Description: 8-Bit current DAC
Instance type: IDAC8 [v2.0]
Datasheet: online component datasheet for IDAC8

Table 18. Component Parameters for IDAC_calibrate

Parameter Name	Value	Description
Current	15	To set the current value
Data_Source	CPU or DMA (Data Bus)	Selects how the data will be written to the iDAC.
Hardware_Enable	false	UDB control for current flow
IDAC_Range	0 - 32uA (125nA/bit)	Selects full scale range of the iDAC.
IDAC_Speed	High Speed	Selects the iDAC settling speed. The Slow Speed selection consumes less power.
Initial_Value	120	Configures the initial iDAC output value. The initial current = value*(FullRange/255). This value is invalid if DAC Bus is used.
Polarity	Current Source	Selects the Sink or Source mode for the iDAC.
Strobe_Mode	Register Write	Selects how the data is strobed into the DAC. An external data strobe signal is required for External. For Register Write the the data is strobed into the DAC on each CPU or DMA write.
User Comments		Instance-specific comments.

8.6 Component type: OpAmp [v1.90]

8.6.1 Instance Opamp_Aux

Description: Opamp
Instance type: OpAmp [v1.90]
Datasheet: online component datasheet for OpAmp

Table 19. Component Parameters for Opamp_Aux

Parameter Name	Value	Description
Mode	OpAmp	Selects between uncommitted op-amp or follower mode.
Power	High Power	Selects the device power level.
User Comments		Instance-specific comments.

8.7 Component type: PWM [v3.30]

8.7.1 Instance PWM_isr

Description: 8 or 16-bit Pulse Width Modulator
Instance type: PWM [v3.30]
Datasheet: online component datasheet for PWM

Table 20. Component Parameters for PWM_isr

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the capture Input. The parameter determines which signal on the capture input is required to capture the current count value to the FIFO.
CompareStatusEdgeSense	true	Enables edge sense detection on compare outputs for use in edge sensitive interrupts
CompareType1	Less or Equal	Sets the compare value comparison type setting for the compare 1 output
CompareType2	Greater or Equal	Sets the compare value comparison type setting for the compare 2 output
CompareValue1	5999	Compares Output 1 to value
CompareValue2	10000	Compares Output 2 to value
DeadBand	Disabled	Defines whether dead band outputs are desired or not.
DeadTime	1	Defines the number of required dead band clock cycles
DitherOffset	0.00	Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM.
EnableMode	Software Only	Specifies the method of enabling the PWM. This can be either hardware or software.
FixedFunction	false	Determines whether the fixed function counter timer is used or the UDB implementation is used.
InterruptOnCMP1	false	Enables the interrupt on compare1 true event

Parameter Name	Value	Description
InterruptOnCMP2	false	Enables the interrupt on compare2 true event
InterruptOnKill	false	Enables the interrupt on a kill event
InterruptOnTC	false	Enables the interrupt on terminal count event
KillMode	Disabled	Parameter to select the kill mode for build time.
MinimumKillTime	1	Sets the minimum number of clock cycles that a kill must be active on the outputs when KillMode is set to Minimum Kill Time mode
Period	23999	Defines the PWM period value
PWMMode	One Output	Defines the overall mode of the PWM
Resolution	16	Defines the bit width of the PWM (8 or 16 bits)
RunMode	Continuous	Defines the run mode options to be either continuous or one shot
TriggerMode	None	Determines the mode of starting the PWM, i.e. triggering the PWM counter to start
UseInterrupt	false	Enables the placement and usage of the status register
User Comments		Instance-specific comments.

8.8 Component type: TIA [v2.0]

8.8.1 Instance TIA

Description: Trans-Impedance Amplifier
Instance type: TIA [v2.0]
Datasheet: online component datasheet for TIA

Table 21. Component Parameters for TIA

Parameter Name	Value	Description
Capacitive_Feedback	4.6 pF	Capacitive feedback for the TIA
Fcorner	567 kHz	Calculated -3dB frequency for the given feedback settings.
Power	Medium Power	Power setting for TIA
Resistive_Feedback	20k ohms	Nominal resistive feedback for the TIA
User Comments		Instance-specific comments.

8.9 Component type: USBFS [v3.20]

8.9.1 Instance USBUART

Description: USB 2.0 Full Speed Device Framework
Instance type: USBFS [v3.20]
Datasheet: online component datasheet for USBFS

Table 22. Component Parameters for USBUART

Parameter Name	Value	Description
EnableBatteryChargDetect	false	This parameter allows to detect a charging supported USB host port using the API function USBFS_DetectPortType().
EnableCDCApi	true	Enables additional high level API's that allow the CDC device to be used similar to a UART device.
EnableMidiApi	true	Enables additional high level MIDI API's.
endpointMA	MA_Static	Endpoint memory allocation
endpointMM	EP_Manual	Endpoint memory management
epDMAautoOptimization	false	This parameter enables resource optimization for DMA with Automatic Memory Management mode. Set this parameter value to true only when a single IN endpoint is present in the device. Enabling this parameter in a multi IN endpoint device configuration causes undesired effects.
extern_cls	false	This parameter allows for user or other component to implement his own handler for Class requests. USBFS_-DispatchClassRqst() function should be implemented if this parameter enabled.
extern_vbus	false	This parameter enables external VBUSDET input.
extern_vnd	false	This parameter allows for user or other component to implement his own handler for Vendor specific requests. USBFS_HandleVendorRqst() function should be implemented if this parameter enabled.
extJackCount	0	Max number of External MIDI IN Jack or OUT Jack descriptors
Gen16bitEpAccessApi	true	This parameter defines whether to generate APIs for the 16-bits endpoint access.
HandleMscRequests	true	This parameter is used to enable handling MSC requests and generate MSC APIs.
isrGroupArbiter	High	This parameter defines the interrupt group of the Arbiter Interrupt.
isrGroupBusReset	Low	This parameter defines the interrupt group of the Bus Reset Interrupt.
isrGroupEp0	Medium	This parameter defines the interrupt group of the Control Endpoint Interrupt (EP0).
isrGroupEp1	Medium	This parameter defines the interrupt group of the Data Endpoint 1 Interrupt.

Parameter Name	Value	Description
isrGroupEp2	Medium	This parameter defines the interrupt group of the Data Endpoint 2 Interrupt.
isrGroupEp3	Medium	This parameter defines the interrupt group of the Data Endpoint 3 Interrupt.
isrGroupEp4	Medium	This parameter defines the interrupt group of the Data Endpoint 4 Interrupt.
isrGroupEp5	Medium	This parameter defines the interrupt group of the Data Endpoint 5 Interrupt.
isrGroupEp6	Medium	This parameter defines the interrupt group of the Data Endpoint 6 Interrupt.
isrGroupEp7	Medium	This parameter defines the interrupt group of the Data Endpoint 7 Interrupt.
isrGroupEp8	Medium	This parameter defines the interrupt group of the Data Endpoint 8 Interrupt.
isrGroupLpm	High	This parameter defines the interrupt group of the LPM Interrupt.
isrGroupSof	Low	This parameter defines the interrupt group of the Start of Frame Interrupt.
max_interfaces_num	2	Defines maximum interfaces number
Mode	false	Specifies whether the implementation will create API for interfacing to UART component(s) for a corresponding set of external MIDI connections.
mon_vbus	false	The mon_vbus parameter adds a single VBUS monitor pin to the design. This pin must be connected to VBUS and must be assigned in the pin editor.
MscDescriptors		Mass Storage Class Descriptors
MscLogicalUnitsNum	1	This parameter allows to specify the number of logical units that should be supported by the Mass Storage device.
out_sof	false	The out_sof parameter enables Start-of-Frame output.
Pid	F232	Product ID
powerpad_vbus	false	This parameter enables VBUS power pad
ProdactName		This string is displayed by the Operating System when it is installing the mass storage device as the Product Name.
ProdactRevision		This string is displayed by the Operating System when it is installing the mass storage device as the Product Revision.

Parameter Name	Value	Description
rm_lpm_int	true	Removes LPM ISR
User Comments		Instance-specific comments.
VendorName		This string is displayed by the Operating System when it is installing the mass storage device as the Vendor Name.
Vid	04B4	Vendor ID

8.10 Component type: VDAC8 [v1.90]

8.10.1 Instance VDAC_source

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

Table 23. Component Parameters for VDAC_source

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	128	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	2048	This parameter sets the voltage value.

8.10.2 Instance VDAC_TIA

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

Table 24. Component Parameters for VDAC_TIA

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	128	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	2048	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