PSoC® Creator™
Project Datasheet for HydroponicAutomationCalibration

Creation Time: 04/20/2019 11:44:55
User: LAPTOP-U4EUI2U9\leryj
Project: HydroponicAutomationCalibration
Tool: PSoC Creator 4.2

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

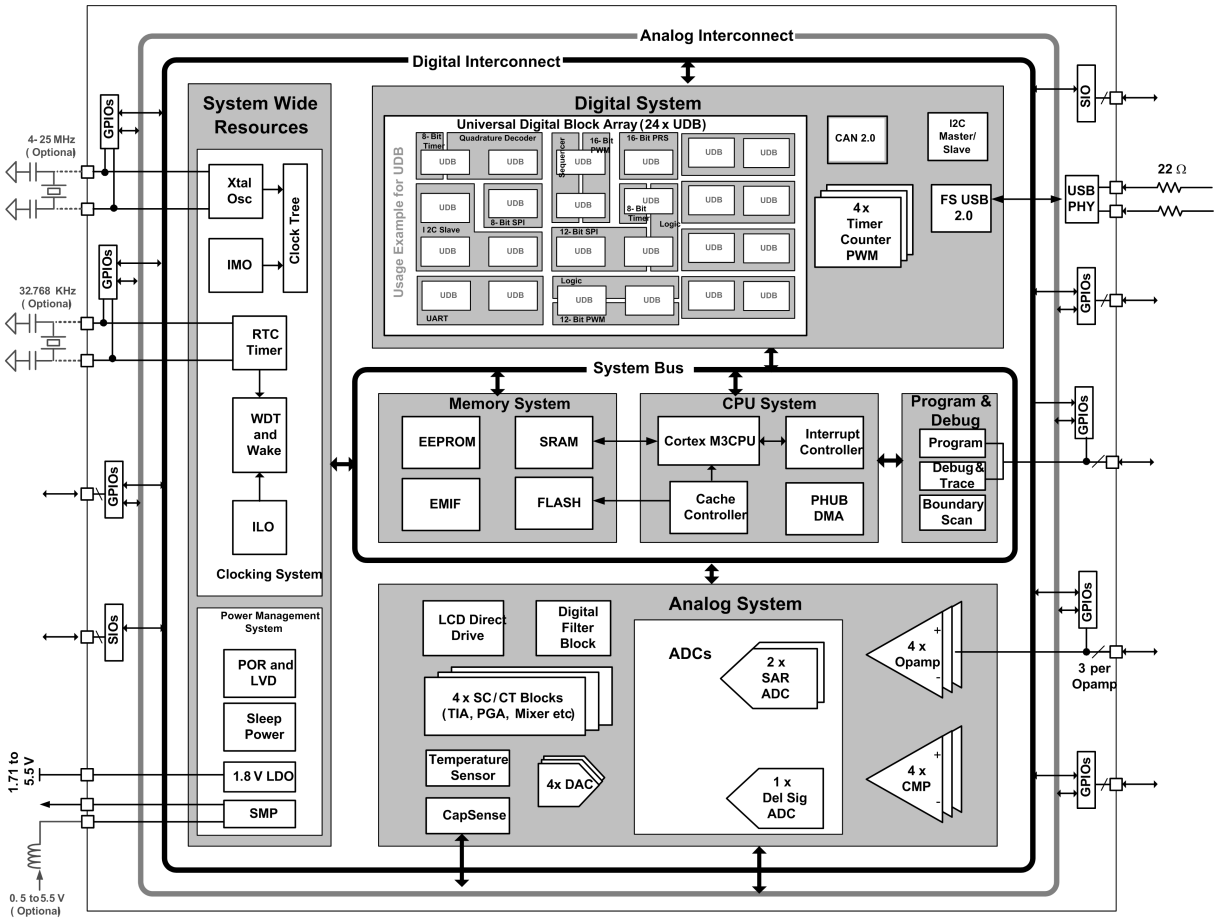


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Part Number	CY8C5888LTI-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 85
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

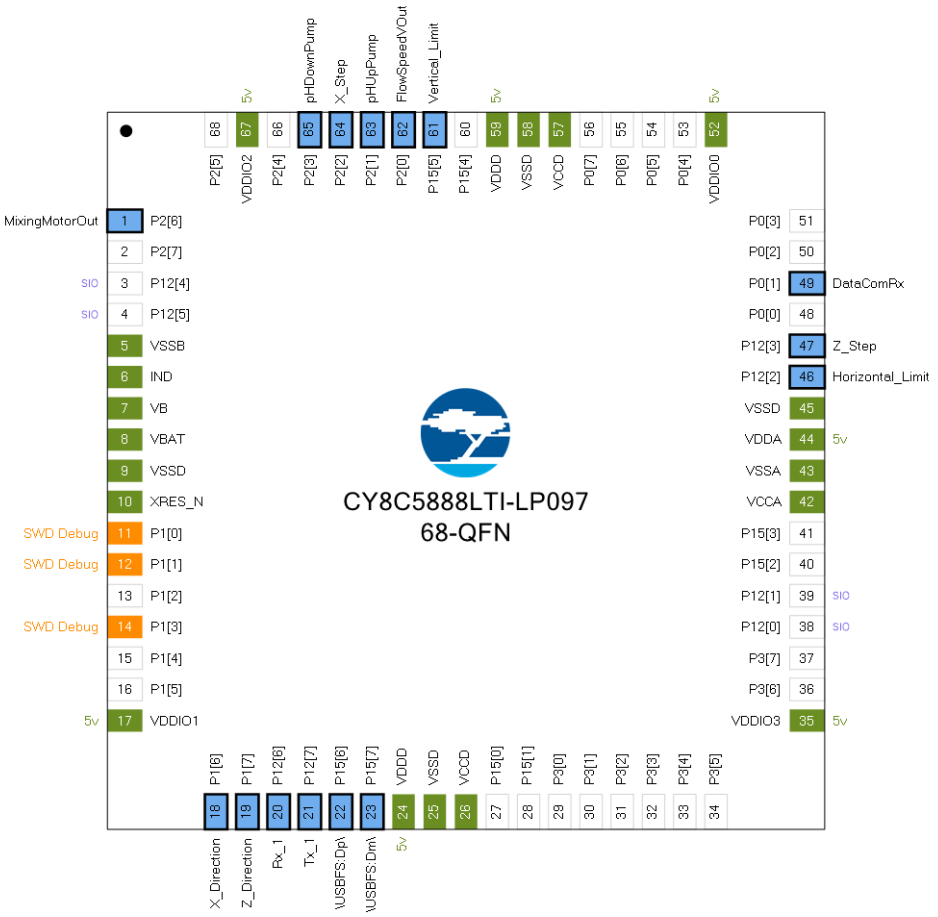
Resource Type	Used	Free	Max	% Used
Digital Clocks	8	0	8	100.00 %
Analog Clocks	1	3	4	25.00 %
CapSense Buffers	0	2	2	0.00 %
Digital Filter Block	0	1	1	0.00 %
Interrupts	14	18	32	43.75 %
IO	18	30	48	37.50 %
Segment LCD	0	1	1	0.00 %
CAN 2.0b	0	1	1	0.00 %
I2C	0	1	1	0.00 %
USB	1	0	1	100.00 %
DMA Channels	2	22	24	8.33 %
Timer	3	1	4	75.00 %
UDB				
Macrocells	65	127	192	33.85 %
Unique P-terms	96	288	384	25.00 %
Total P-terms	112			
Datapath Cells	12	12	24	50.00 %
Status Cells	9	15	24	37.50 %
StatusI Registers	7			
Routed Count7 Load/Enable	2			
Control Cells	10	14	24	41.67 %
Control Registers	8			
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	0	4	4	0.00 %
DAC				

Resource Type	Used	Free	Max	% Used
VIDAC	2	2	4	50.00 %

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]	MixingMotorOut	Analog	HiZ analog	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]	X_Direction	Dgtl Out	Strong drive	HiZ Analog Unb
19	P1[7]	Z_Direction	Dgtl Out	Strong drive	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]	\USBFS:Dp\	Analog	HiZ analog	HiZ Analog Unb
23	P15[7]	\USBFS:Dm\	Analog	HiZ analog	HiZ Analog Unb
24	VDDD	VDDD	Power		
25	VSSD	VSSD	Power		
26	VCCD	VCCD	Power		
27	P15[0]	GPIO [unused]			HiZ Analog Unb
28	P15[1]	GPIO [unused]			HiZ Analog Unb
29	P3[0]	GPIO [unused]			HiZ Analog Unb
30	P3[1]	GPIO [unused]			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]	GPIO [unused]			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]	Horizontal_Limit	Dgtl In	Strong drive	HiZ Analog Unb
47	P12[3]	Z_Step	Dgtl Out	Strong drive	HiZ Analog Unb
48	P0[0]	GPIO [unused]			HiZ Analog Unb
49	P0[1]	DataComRx	Dgtl In	HiZ digital	HiZ Analog Unb
50	P0[2]	GPIO [unused]			HiZ Analog Unb
51	P0[3]	GPIO [unused]			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]	Vertical_Limit	Dgtl In	Strong drive	HiZ Analog Unb
62	P2[0]	FlowSpeedVOut	Analog	HiZ analog	HiZ Analog Unb
63	P2[1]	pHUpPump	Dgtl Out	Strong drive	HiZ Analog Unb
64	P2[2]	X_Step	Dgtl Out	Strong drive	HiZ Analog Unb
65	P2[3]	pHDownPump	Dgtl Out	Strong drive	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 = High impedance analog
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl Out = Digital Output
- Dgtl In = Digital Input
- HiZ digital = High impedance digital

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	GPIO [unused]			HiZ Analog Unb
P0[1]	49	DataComRx	Dgtl In	HiZ digital	HiZ Analog Unb
P0[2]	50	GPIO [unused]			HiZ Analog Unb
P0[3]	51	GPIO [unused]			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	X_Direction	Dgtl Out	Strong drive	HiZ Analog Unb
P1[7]	19	Z_Direction	Dgtl Out	Strong drive	HiZ Analog Unb
P12[0]	38	SIO [unused]			HiZ Analog Unb
P12[1]	39	SIO [unused]			HiZ Analog Unb
P12[2]	46	Horizontal_Limit	Dgtl In	Strong drive	HiZ Analog Unb
P12[3]	47	Z_Step	Dgtl Out	Strong drive	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	GPIO [unused]			HiZ Analog Unb
P15[1]	28	GPIO [unused]			HiZ Analog Unb
P15[2]	40	GPIO [unused]			HiZ Analog Unb
P15[3]	41	GPIO [unused]			HiZ Analog Unb
P15[4]	60	GPIO [unused]			HiZ Analog Unb
P15[5]	61	Vertical_Limit	Dgtl In	Strong drive	HiZ Analog Unb
P15[6]	22	\USBFS:Dp\	Analog	HiZ analog	HiZ Analog Unb
P15[7]	23	\USBFS:Dm\	Analog	HiZ analog	HiZ Analog Unb
P2[0]	62	FlowSpeedVOut	Analog	HiZ analog	HiZ Analog Unb
P2[1]	63	pHUpPump	Dgtl Out	Strong drive	HiZ Analog Unb
P2[2]	64	X_Step	Dgtl Out	Strong drive	HiZ Analog Unb
P2[3]	65	pHDownPump	Dgtl Out	Strong drive	HiZ Analog Unb
P2[4]	66	GPIO [unused]			HiZ Analog Unb
P2[5]	68	GPIO [unused]			HiZ Analog Unb
P2[6]	1	MixingMotorOut	Analog	HiZ analog	HiZ Analog Unb
P2[7]	2	GPIO [unused]			HiZ Analog Unb
P3[0]	29	GPIO [unused]			HiZ Analog Unb
P3[1]	30	GPIO [unused]			HiZ Analog Unb
P3[2]	31	GPIO [unused]			HiZ Analog Unb
P3[3]	32	GPIO [unused]			HiZ Analog Unb
P3[4]	33	GPIO [unused]			HiZ Analog Unb

Port	Pin	Name	Type	Drive Mode	Reset State
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:

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

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
\USBFS:Dm\	P15[7]	Analog	HiZ Analog Unb
\USBFS:Dp\	P15[6]	Analog	HiZ Analog Unb
DataComRx	P0[1]	Dgtl In	HiZ Analog Unb
Debug:SWD_CK	P1[1]	Reserved	
Debug:SWD_IO	P1[0]	Reserved	
Debug:SWV	P1[3]	Reserved	
FlowSpeedVOut	P2[0]	Analog	HiZ Analog Unb
GPIO [unused]	P15[2]		HiZ Analog Unb
GPIO [unused]	P3[7]		HiZ Analog Unb
GPIO [unused]	P3[4]		HiZ Analog Unb
GPIO [unused]	P3[3]		HiZ Analog Unb
GPIO [unused]	P2[5]		HiZ Analog Unb
GPIO [unused]	P3[6]		HiZ Analog Unb
GPIO [unused]	P3[2]		HiZ Analog Unb
GPIO [unused]	P15[3]		HiZ Analog Unb
GPIO [unused]	P0[7]		HiZ Analog Unb
GPIO [unused]	P0[6]		HiZ Analog Unb
GPIO [unused]	P2[4]		HiZ Analog Unb
GPIO [unused]	P15[4]		HiZ Analog Unb
GPIO [unused]	P0[5]		HiZ Analog Unb
GPIO [unused]	P0[2]		HiZ Analog Unb
GPIO [unused]	P0[0]		HiZ Analog Unb
GPIO [unused]	P0[4]		HiZ Analog Unb
GPIO [unused]	P0[3]		HiZ Analog Unb
GPIO [unused]	P3[1]		HiZ Analog Unb
GPIO [unused]	P1[2]		HiZ Analog Unb
GPIO [unused]	P2[7]		HiZ Analog Unb
GPIO [unused]	P1[5]		HiZ Analog Unb
GPIO [unused]	P1[4]		HiZ Analog Unb
GPIO [unused]	P15[1]		HiZ Analog Unb
GPIO [unused]	P3[0]		HiZ Analog Unb
GPIO [unused]	P3[5]		HiZ Analog Unb
GPIO [unused]	P15[0]		HiZ Analog Unb
Horizontal_Limit	P12[2]	Dgtl In	HiZ Analog Unb
MixingMotorOut	P2[6]	Analog	HiZ Analog Unb
pHDownPump	P2[3]	Dgtl Out	HiZ Analog Unb
pHUpPump	P2[1]	Dgtl Out	HiZ Analog Unb
Rx_1	P12[6]	Dgtl In	HiZ Analog Unb
SIO [unused]	P12[0]		HiZ Analog Unb
SIO [unused]	P12[1]		HiZ Analog Unb
SIO [unused]	P12[5]		HiZ Analog Unb
SIO [unused]	P12[4]		HiZ Analog Unb
Tx_1	P12[7]	Dgtl Out	HiZ Analog Unb
Vertical_Limit	P15[5]	Dgtl In	HiZ Analog Unb
X_Direction	P1[6]	Dgtl Out	HiZ Analog Unb

Name	Port	Type	Reset State
X_Step	P2[2]	Dgtl Out	HiZ Analog Unb
Z_Direction	P1[7]	Dgtl Out	HiZ Analog Unb
Z_Step	P12[3]	Dgtl Out	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)	0x300
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

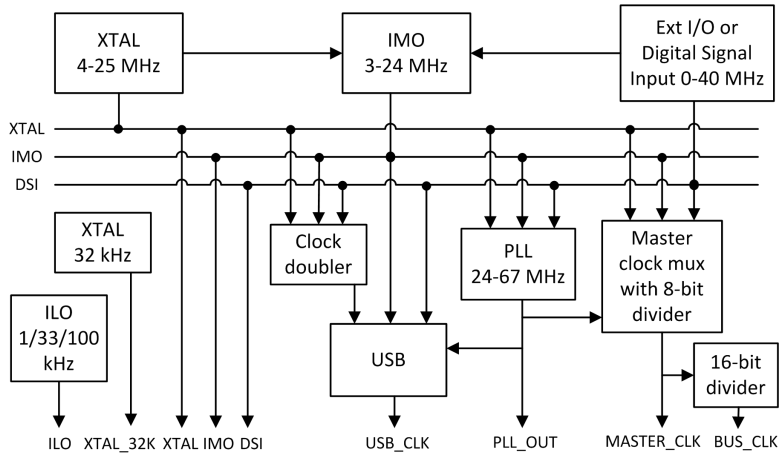
Name	Value
VDDA (V)	5
VDDD (V)	5
VDDIO0 (V)	5
VDDIO1 (V)	5
VDDIO2 (V)	5
VDDIO3 (V)	5
Variable VDDA	True
Temperature Range	-40C - 85/125C

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
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

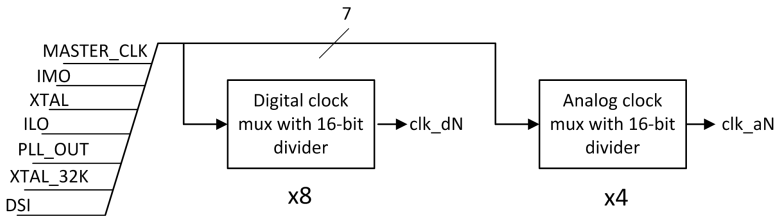


Table 10 lists the design wide clocks used in this design.

Table 10. Design Wide Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
ScBoostClk	ANALOG	IMO	10 MHz	12 MHz	±0.25	False	True

Table 11 lists the local clocks used in this design.

Table 11. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
timer_clock_1	DIGITAL	BUS_CLK	? MHz	24 MHz	±0.25	True	True
timer_clock_2	DIGITAL	BUS_CLK	? MHz	24 MHz	±0.25	True	True
timer_clock	DIGITAL	MASTER_CLK	1 MHz	1 MHz	±0.25	True	True
Clock_1	DIGITAL	MASTER_CLK	1 MHz	1 MHz	±0.25	True	True

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
UART_IntClock	DIGITAL	MASTER_CLK	921.6 kHz	923.077 kHz	±0.25	True	True
Clock_2	DIGITAL	MASTER_CLK	200 kHz	200 kHz	±0.25	True	True
Clock_4	DIGITAL	MASTER_CLK	100 kHz	100 kHz	±0.25	True	True
SensorRxCom-UART_IntClock	DIGITAL	MASTER_CLK	76.8 kHz	76.677 kHz	±0.25	True	True
Clock_3	DIGITAL	MASTER_CLK	1 kHz	1 kHz	±0.25	True	True
timer_clock_3	DIGITAL	MASTER_CLK	500 Hz	500 Hz	±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
  - 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 12. Interrupts

Name	Intr Num	Vector	Priority
DataComRxISR	0	0	7
FlowCounterTimerISR	1	1	7
X_Limit_ISR	2	2	7
Z_Limit_ISR	3	3	7
USBFS_ep_1	4	4	7
USBFS_ep_2	5	5	7
isr_rx	6	6	7
pHControlISR	7	7	7
USBFS_dp_int	12	12	7
X_Step_ISR	17	17	7
Z_Step_ISR	18	18	7
USBFS_arb_int	22	22	7
USBFS_bus_reset	23	23	7
USBFS_ep_0	24	24	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 the following DMA components: (0 is the highest priority)

Table 13. DMAs

Name	Priority	Channel Number
USBFS_ep1	2	0
USBFS_ep2	2	1

For more information on DMAs, please refer to:

- PHUB and DMAC chapter in the PSoC 5LP Technical Reference Manual
- DMA chapter in the System Reference Guide
  - DMA API routines and related registers
- Datasheet for cy_dma component

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 14 lists the Flash protection settings for your design.

Table 14. 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:

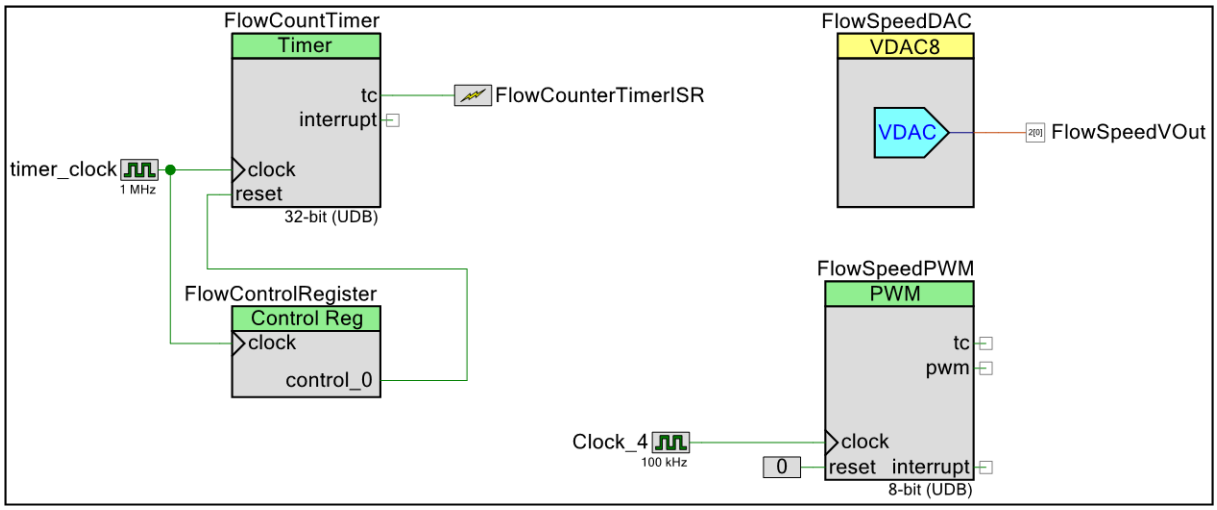
- Flash Protection chapter in the PSoC 5LP Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyWrite API routines
  - CyFlash API routines

7 Design Contents

This design's schematic content consists of the following 5 schematic sheets:

7.1 Schematic Sheet: FlowController

Figure 5. Schematic Sheet: FlowController

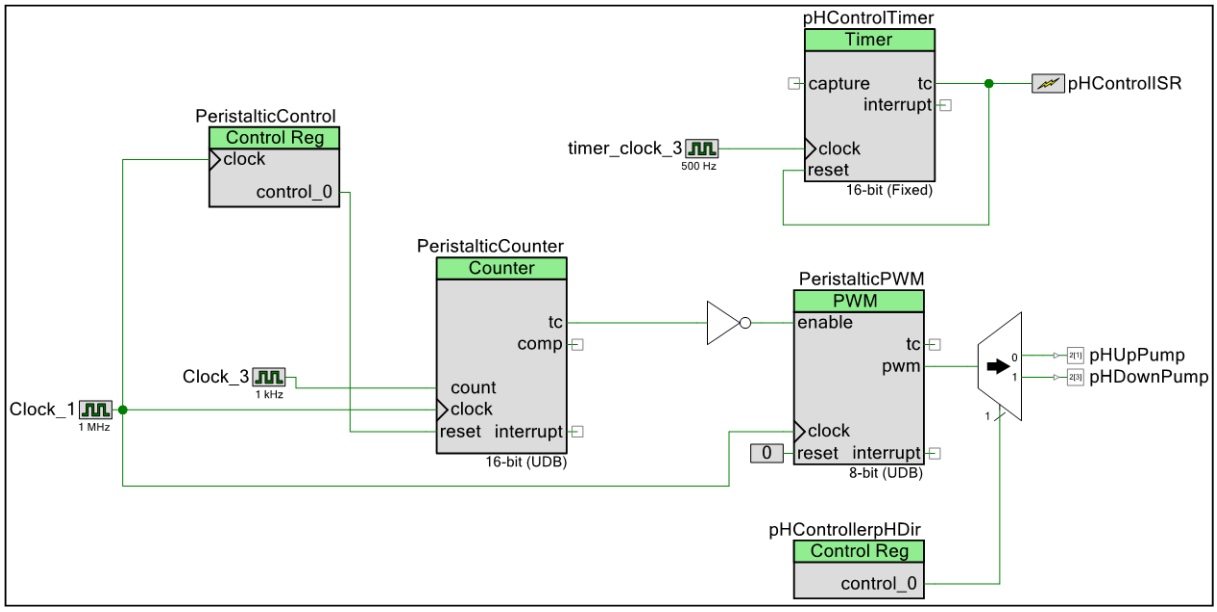


This schematic sheet contains the following component instances:

- Instance FlowControlRegister (type: CyControlReg_v1_80)
- Instance FlowCountTimer (type: Timer_v2_70)
- Instance FlowSpeedDAC (type: VDAC8_v1_90)
- Instance FlowSpeedPWM (type: PWM_v3_30)

7.2 Schematic Sheet: pHController

Figure 6. Schematic Sheet: pHController

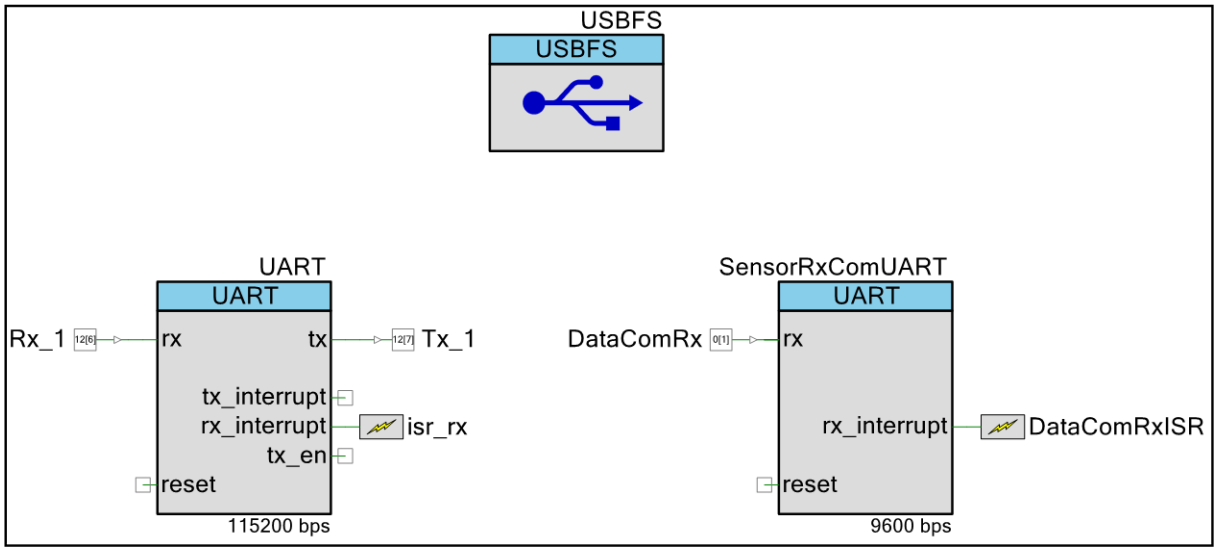


This schematic sheet contains the following component instances:

- Instance demux_1 (type: demux_v1_10)
- Instance PeristalticControl (type: CyControlReg_v1_80)
- Instance PeristalticCounter (type: Counter_v3_0)
- Instance PeristalticPWM (type: PWM_v3_30)
- Instance pHControllerpHDir (type: CyControlReg_v1_80)
- Instance pHControlTimer (type: Timer_v2_80)

7.3 Schematic Sheet: SerialCom

Figure 7. Schematic Sheet: SerialCom

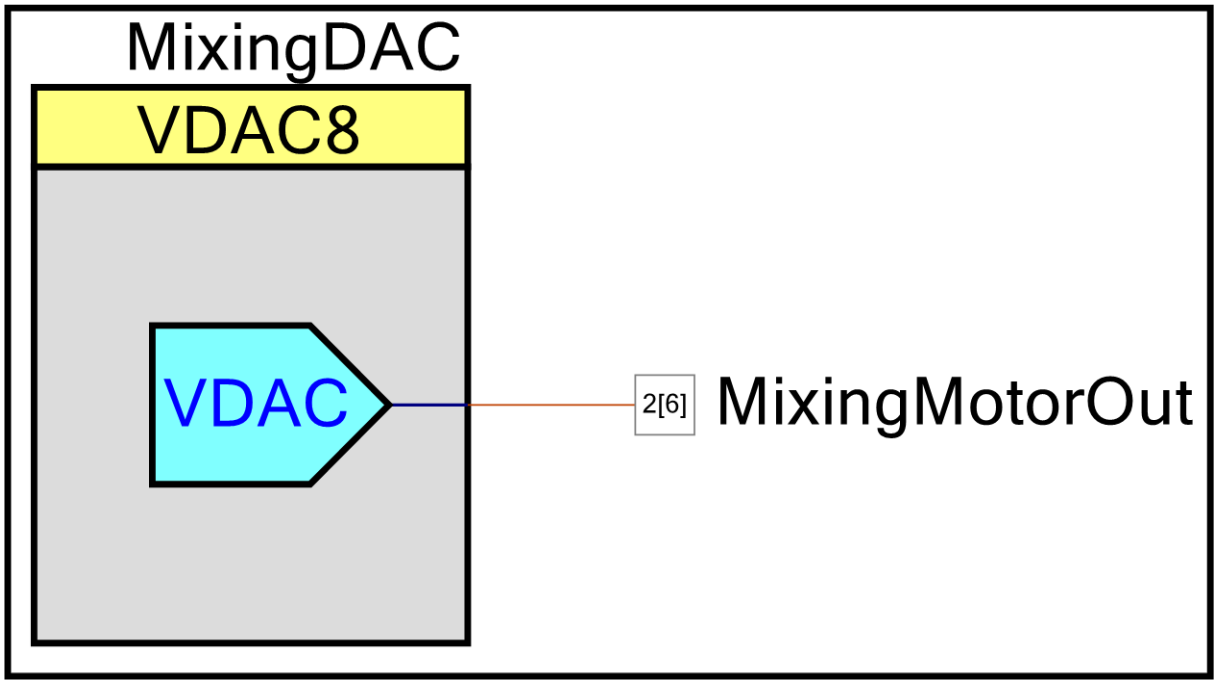


This schematic sheet contains the following component instances:

- Instance SensorRxComUART (type: UART_v2_50)
- Instance UART (type: UART_v2_50)
- Instance USBFS (type: USBFS_v3_20)

7.4 Schematic Sheet: Mixing

Figure 8. Schematic Sheet: Mixing

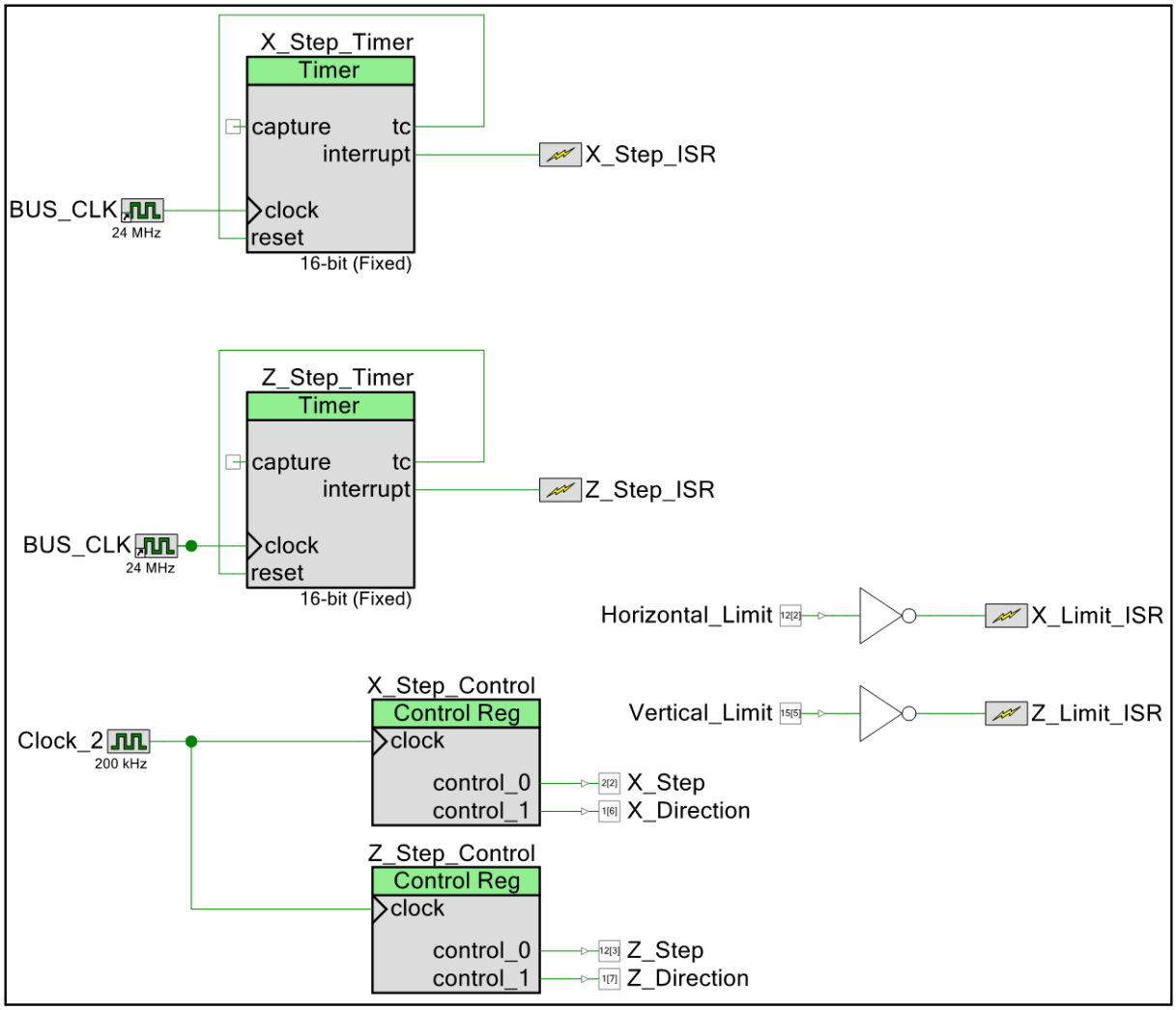


This schematic sheet contains the following component instances:

- Instance MixingDAC (type: VDAC8_v1_90)

7.5 Schematic Sheet: Calibration

Figure 9. Schematic Sheet: Calibration



This schematic sheet contains the following component instances:

- Instance X_Step_Control (type: CyControlReg_v1_80)
- Instance X_Step_Timer (type: Timer_v2_80)
- Instance Z_Step_Control (type: CyControlReg_v1_80)
- Instance Z_Step_Timer (type: Timer_v2_80)

8 Components

8.1 Component type: Counter [v3.0]

8.1.1 Instance PeristalticCounter

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

Table 15. Component Parameters for PeristalticCounter

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	Down 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	false	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	false	Enables the counter status register to produce an interrupt output signal on terminal count.
Period	200	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	false	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	One Shot	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: CyControlReg [v1.80]

8.2.1 Instance FlowControlRegister

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 16. Component Parameters for FlowControlRegister

Parameter Name	Value	Description
Bit0Mode	PulseMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode

Parameter Name	Value	Description
BitValue	1	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.2.2 Instance PeristalticControl

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 17. Component Parameters for PeristalticControl

Parameter Name	Value	Description
Bit0Mode	PulseMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.2.3 Instance pHControllerpHDir

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 18. Component Parameters for pHControllerpHDir

Parameter Name	Value	Description
Bit0Mode	DirectMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus

Parameter Name	Value	Description
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.2.4 Instance X_Step_Control

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 19. Component Parameters for X_Step_Control

Parameter Name	Value	Description
Bit0Mode	PulseMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	2	Defines the number of outputs needed (1-8)
User Comments		Instance-specific comments.

8.2.5 Instance Z_Step_Control

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.80]
Datasheet: online component datasheet for CyControlReg

Table 20. Component Parameters for Z_Step_Control

Parameter Name	Value	Description
Bit0Mode	PulseMode	Defines bit 0 mode
Bit1Mode	DirectMode	Defines bit 1 mode
Bit2Mode	DirectMode	Defines bit 2 mode
Bit3Mode	DirectMode	Defines bit 3 mode
Bit4Mode	DirectMode	Defines bit 4 mode
Bit5Mode	DirectMode	Defines bit 5 mode
Bit6Mode	DirectMode	Defines bit 6 mode
Bit7Mode	DirectMode	Defines bit 7 mode
BitValue	0	Defines bit value
BusDisplay	false	Displays the output terminals as bus
ExternalReset	false	Shows the reset terminal
NumOutputs	2	Defines the number of outputs needed (1-8)

Parameter Name	Value	Description
User Comments		Instance-specific comments.

8.3 Component type: demux [v1.10]

8.3.1 Instance demux_1

Description: De-Multiplexer with configurable number of output terminals and terminal width.
Instance type: demux [v1.10]
Datasheet: online component datasheet for demux

Table 21. Component Parameters for demux_1

Parameter Name	Value	Description
NumOutputTerminals	2	Number of output terminals of the De-Multiplexer. Acceptable values are 2, 4, 8 and 16.
TerminalWidth	1	Width of each terminal
User Comments		Instance-specific comments.

8.4 Component type: PWM [v3.30]

8.4.1 Instance FlowSpeedPWM

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

Table 22. Component Parameters for FlowSpeedPWM

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	Sets the compare value comparison type setting for the compare 1 output
CompareType2	Less	Sets the compare value comparison type setting for the compare 2 output
CompareValue1	127	Compares Output 1 to value
CompareValue2	63	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.

Parameter Name	Value	Description
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
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	255	Defines the PWM period value
PWMMode	One Output	Defines the overall mode of the PWM
Resolution	8	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	true	Enables the placement and usage of the status register
User Comments		Instance-specific comments.

8.4.2 Instance PeristalticPWM

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

Table 23. Component Parameters for PeristalticPWM

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	Sets the compare value comparison type setting for the compare 1 output
CompareType2	Less	Sets the compare value comparison type setting for the compare 2 output

Parameter Name	Value	Description
CompareValue1	127	Compares Output 1 to value
CompareValue2	63	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	Hardware 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
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	255	Defines the PWM period value
PWMMode	One Output	Defines the overall mode of the PWM
Resolution	8	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	true	Enables the placement and usage of the status register
User Comments		Instance-specific comments.

8.5 Component type: Timer [v2.70]

8.5.1 Instance FlowCountTimer

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

Table 24. Component Parameters for FlowCountTimer

Parameter Name	Value	Description
CaptureAlternatingFall	false	Enables data capture on either edge but not until a valid falling edge is detected first.
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	false	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	999999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	32	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	One Shot	Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered.

Parameter Name	Value	Description
TriggerMode	None	Defines the required trigger input signal to cause a valid trigger enable of the timer
User Comments		Instance-specific comments.

8.6 Component type: Timer [v2.80]

8.6.1 Instance pHControlTimer

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.80]
Datasheet: online component datasheet for Timer

Table 25. Component Parameters for pHControlTimer

Parameter Name	Value	Description
CaptureAlternatingFall	false	Enables data capture on either edge but not until a valid falling edge is detected first.
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	Rising Edge	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	true	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	false	Parameter to check whether interrupt on a TC is enabled or disabled.

Parameter Name	Value	Description
NumberOfCaptures	1	Number of captures allowed until the counter is cleared or disabled.
Period	14999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	16	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	One Shot	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
User Comments		Instance-specific comments.

8.6.2 Instance X_Step_Timer

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.80]
Datasheet: online component datasheet for Timer

Table 26. Component Parameters for X_Step_Timer

Parameter Name	Value	Description
CaptureAlternatingFall	false	Enables data capture on either edge but not until a valid falling edge is detected first.
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	Rising Edge	This parameter defines the capture input signal requirements to trigger a valid capture event

Parameter Name	Value	Description
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	true	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	31999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	16	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
User Comments		Instance-specific comments.

8.6.3 Instance Z_Step_Timer

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.80]
Datasheet: online component datasheet for Timer

Table 27. Component Parameters for Z_Step_Timer

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

Parameter Name	Value	Description
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	Rising Edge	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	true	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	31999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	16	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
User Comments		Instance-specific comments.

8.7 Component type: UART [v2.50]

8.7.1 Instance SensorRxComUART

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

Table 28. Component Parameters for SensorRxComUART

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	9600	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
IntOnByteRcvd	true	Enables the interrupt on RX byte received event by default

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

8.7.2 Instance UART

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

Table 29. Component Parameters for UART

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.

Parameter Name	Value	Description
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	true	Enables the interrupt on break signal detected event by default
IntOnByteRcvd	true	Enables the interrupt on RX byte received event by default
IntOnOverrunError	true	Enables the interrupt on overrun error event by default
IntOnParityError	true	Enables the interrupt on parity error event by default
IntOnStopError	true	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

Parameter Name	Value	Description
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.
User Comments		Instance-specific comments.

8.8 Component type: USBFS [v3.20]

8.8.1 Instance USBFS

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

Table 30. Component Parameters for USBFS

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_DMAmanual	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.

Parameter Name	Value	Description
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	true	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	false	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.
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.

Parameter Name	Value	Description
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	1	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.
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

8.9 Component type: VDAC8 [v1.90]

8.9.1 Instance FlowSpeedDAC

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

Table 31. Component Parameters for FlowSpeedDAC

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	100	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	1600	This parameter sets the voltage value.

8.9.2 Instance MixingDAC

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

Table 32. Component Parameters for MixingDAC

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	100	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	1600	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