PSoC® Creator™
Project Datasheet for IA

Creation Time: 03/15/2013 22:57:35
User: Jérémy-PC\jerem
Project: IA

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

Copyrights

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

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.

The information in this document is subject to change without notice and should not be construed as a commitment by Cypress. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. No part of this document may be copied, or reproduced for commercial use, in any form or by any means without the prior written consent of Cypress.

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. 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 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 Serial Wire Debug (SWD), and Single Wire Viewer (SWV)
- Flexible routing to all pins

Figure 1 shows the major components of a typical CY8C55 family member PSoC 5 device. For details on all the systems listed above, please refer to the PSoC 5 Technical Reference Manual .

Figure 1. CY8C55 Device Family Block Diagram

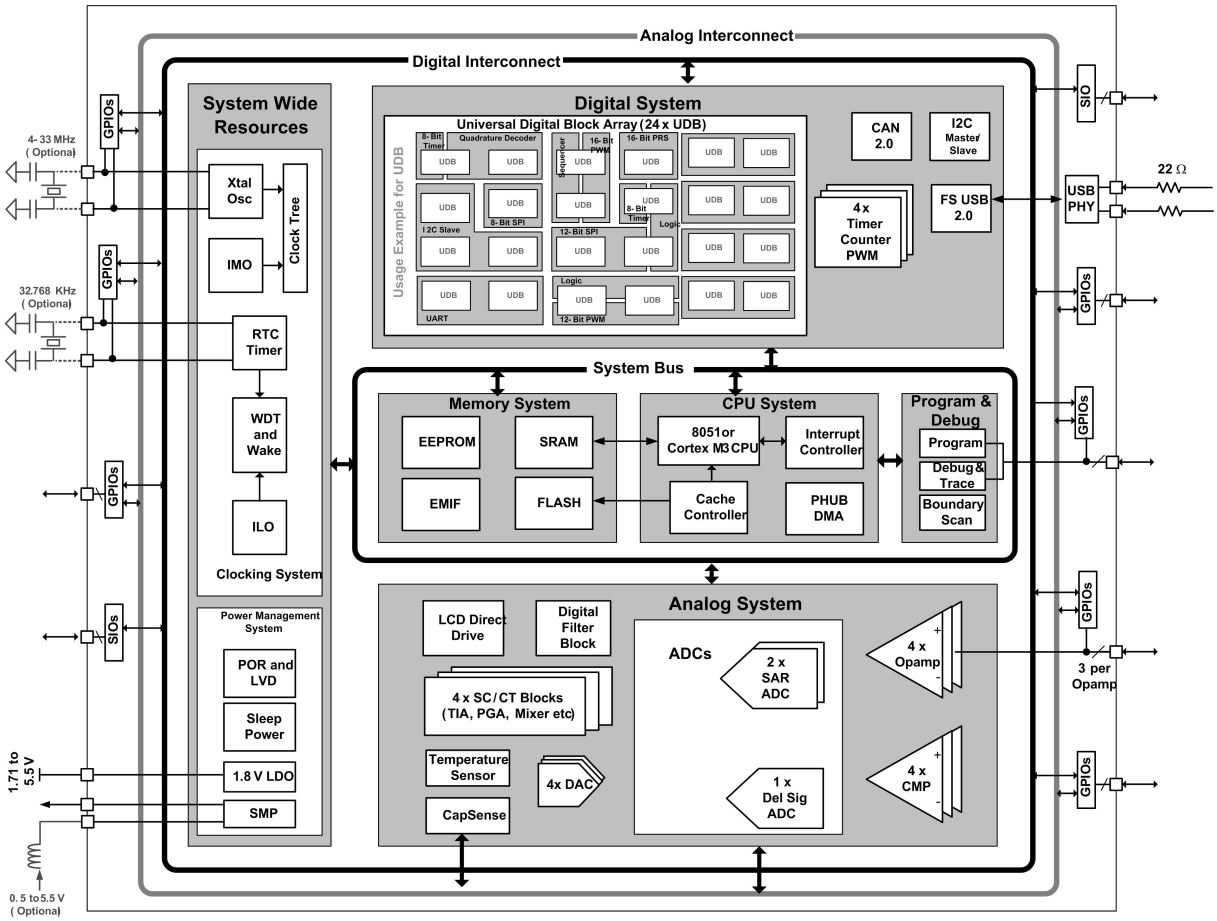


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

Name	Value
Architecture	PSoC 5
Family	CY8C55
CPU speed (MHz)	80
Flash size (kBytes)	256
SRAM size (kBytes)	64
EEPROM size (Bytes)	2048
Trace Buffer (kBytes)	0
Vdd range (V)	1.7 to 5.5
Automotive qualified	No (Industrial Grade Only)
Temp range (Celcius)	-40 to 85
JTAG ID	0x0E13C069

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

Table 2 lists the device resources that this design uses:

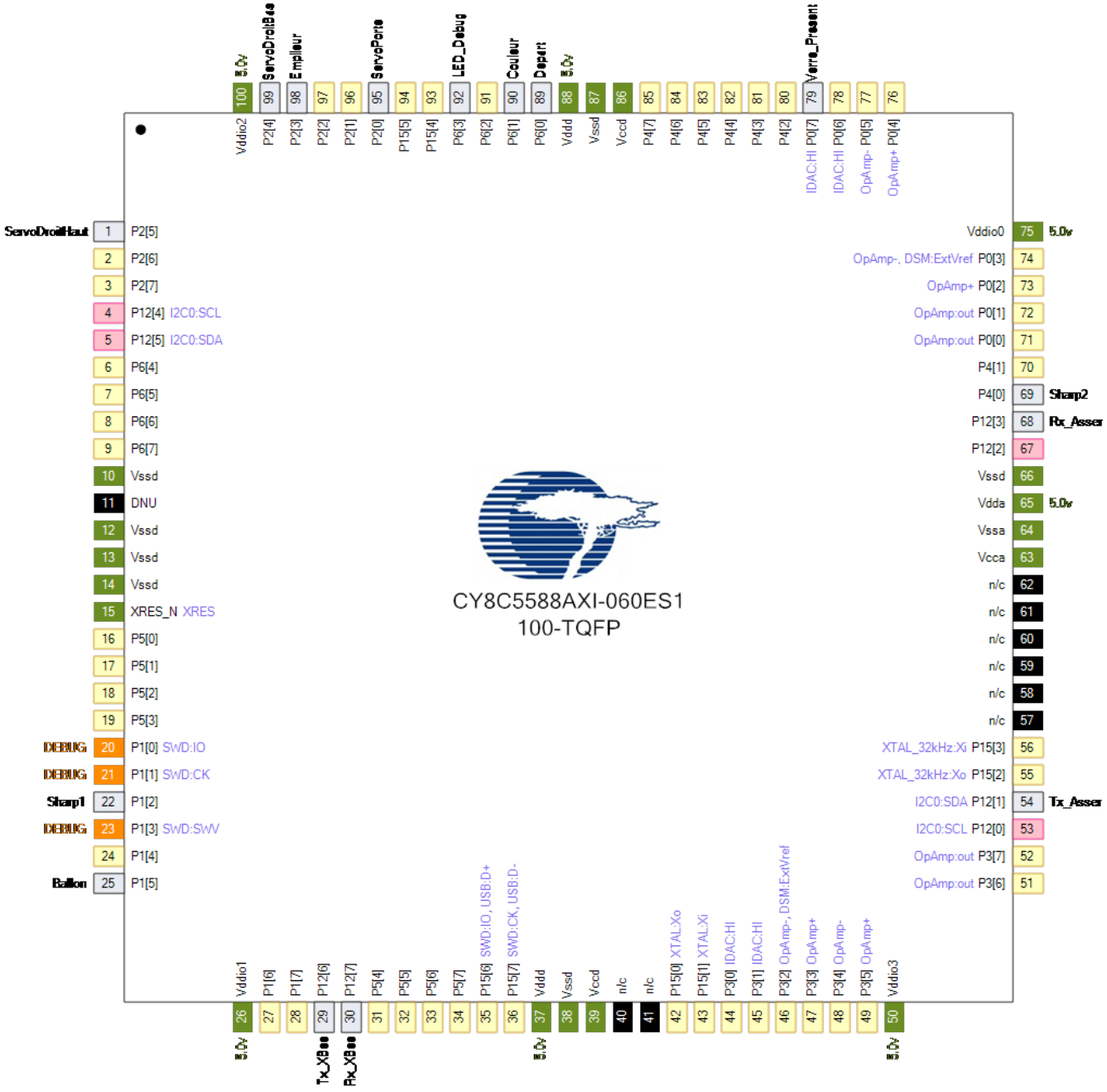
Table 2. Device Resources

Name	Resources in Use	Total Resources Available
Digital domain clock dividers	5 (62.5%)	8
Analog domain clock dividers	1 (25.0%)	4
Pins	18 (25.0%)	72
UDB Macrocells	72 (37.5%)	192
UDB Unique Pterms	106 (27.6%)	384
UDB Datapath Cells	18 (75.0%)	24
UDB Status Cells	10 (41.7%)	24
UDB Control Cells	8 (33.3%)	24
DMA Channels	0 (0.0%)	24
Interrupts	5 (15.6%)	32
DSM Fixed Blocks	1 (100.0%)	1
VIDAC Fixed Blocks	0 (0.0%)	4
SC Fixed Blocks	0 (0.0%)	4
Comparator Fixed Blocks	0 (0.0%)	4
Opamp Fixed Blocks	0 (0.0%)	4
CapSense Buffers	0 (0.0%)	2
CAN Fixed Blocks	0 (0.0%)	1
Decimator Fixed Blocks	1 (100.0%)	1
I2C Fixed Blocks	0 (0.0%)	1
Timer Fixed Blocks	0 (0.0%)	4
DFB Fixed Blocks	0 (0.0%)	1
USB Fixed Blocks	0 (0.0%)	1
LCD Fixed Blocks	0 (0.0%)	1
EMIF Fixed Blocks	0 (0.0%)	1
LPF Fixed Blocks	0 (0.0%)	2
SAR Fixed Blocks	0 (0.0%)	2

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]	ServoDroitHaut	Dgtl Out	Strong drive	HiZ Analog Unb
2	P2[6]	GPIO [unused]			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]	GPIO [unused]			HiZ Analog Unb
9	P6[7]	GPIO [unused]			HiZ Analog Unb
10	Vssd	Vssd	Power		
12	Vssd	Vssd	Power		
13	Vssd	Vssd	Power		
14	Vssd	Vssd	Power		
15	XRES_N	Power			
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]	GPIO [unused]			HiZ Analog Unb
21	P1[1]	GPIO [unused]			HiZ Analog Unb
22	P1[2]	Sharp1	Analog	HiZ analog	HiZ Analog Unb
23	P1[3]	GPIO [unused]			HiZ Analog Unb
24	P1[4]	GPIO [unused]			HiZ Analog Unb
25	P1[5]	Ballon		Strong drive	HiZ Analog Unb
26	Vio1	Vio1	Power		
27	P1[6]	GPIO [unused]			HiZ Analog Unb
28	P1[7]	GPIO [unused]			HiZ Analog Unb
29	P12[6]	Tx_XBee	Dgtl Out	Strong drive	HiZ Analog Unb
30	P12[7]	Rx_XBee	Dgtl In	HiZ digital	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]	USB [unused]			HiZ Analog Unb
36	P15[7]	USB [unused]			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]			HiZ Analog Unb
46	P3[2]	GPIO [unused]			HiZ Analog Unb
47	P3[3]	GPIO [unused]			HiZ Analog Unb
48	P3[4]	GPIO [unused]			HiZ Analog Unb

Pin	Port	Name	Type	Drive Mode	Reset State
49	P3[5]	GPIO [unused]			HiZ Analog Unb
50	Vio3	Vio3	Power		
51	P3[6]	GPIO [unused]			HiZ Analog Unb
52	P3[7]	GPIO [unused]			HiZ Analog Unb
53	P12[0]	SIO [unused]			HiZ Analog Unb
54	P12[1]	Tx_Asser	Dgtl Out	Strong drive	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]	Rx_Asser	Dgtl In	HiZ digital	HiZ Analog Unb
69	P4[0]	Sharp2	Analog	HiZ analog	HiZ Analog Unb
70	P4[1]	GPIO [unused]			HiZ Analog Unb
71	P0[0]	GPIO [unused]			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	Vio0	Vio0	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]	Verre_Present		HiZ digital	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]	Depart		HiZ digital	HiZ Analog Unb
90	P6[1]	Couleur		HiZ digital	HiZ Analog Unb
91	P6[2]	GPIO [unused]			HiZ Analog Unb
92	P6[3]	LED_Debug		Strong drive	HiZ Analog Unb
93	P15[4]	GPIO [unused]			HiZ Analog Unb
94	P15[5]	GPIO [unused]			HiZ Analog Unb
95	P2[0]	ServoPorte	Dgtl Out	Strong drive	HiZ Analog Unb
96	P2[1]	GPIO [unused]			HiZ Analog Unb
97	P2[2]	GPIO [unused]			HiZ Analog Unb
98	P2[3]	Empileur		Strong drive	HiZ Analog Unb
99	P2[4]	ServoDroitBas	Dgtl Out	Strong drive	HiZ Analog Unb
100	Vio2	Vio2	Power		

Abbreviations used in Table 3 have the following meanings:

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

2.2 Software Pins

Table 4 contains information about the software pins on this device in alphabetical order. (Only software-accessible pins are shown.)

Table 4. Software Pins

Name	Port	Type	Reset State
Ballon	P1[5]		HiZ Analog Unb
Couleur	P6[1]		HiZ Analog Unb
Depart	P6[0]		HiZ Analog Unb
Empileur	P2[3]		HiZ Analog Unb
LED_Debug	P6[3]		HiZ Analog Unb
Power	XRES_N		
Rx_Asser	P12[3]	Dgtl In	HiZ Analog Unb
Rx_XBee	P12[7]	Dgtl In	HiZ Analog Unb
ServoDroitBas	P2[4]	Dgtl Out	HiZ Analog Unb
ServoDroitHaut	P2[5]	Dgtl Out	HiZ Analog Unb
ServoPorte	P2[0]	Dgtl Out	HiZ Analog Unb
Sharp1	P1[2]	Analog	HiZ Analog Unb
Sharp2	P4[0]	Analog	HiZ Analog Unb
Tx_Asser	P12[1]	Dgtl Out	HiZ Analog Unb
Tx_XBee	P12[6]	Dgtl Out	HiZ Analog Unb
Verre_Present	P0[7]		HiZ Analog Unb

Abbreviations used in Table 4 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 5. System Configuration Settings

Name	Value
Device Configuration Mode	DMA
Use Dedicated Configuration Data Memory	True
Instruction Cache Enabled	True
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x1000
Stack Size (bytes)	0x4000
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 6. System Debug Settings

Name	Value
Debug Select	SWD+SWV (serial wire debug and viewer)
Enable Device Protection	False

3.3 System Operating Conditions

Table 7. System Operating Conditions

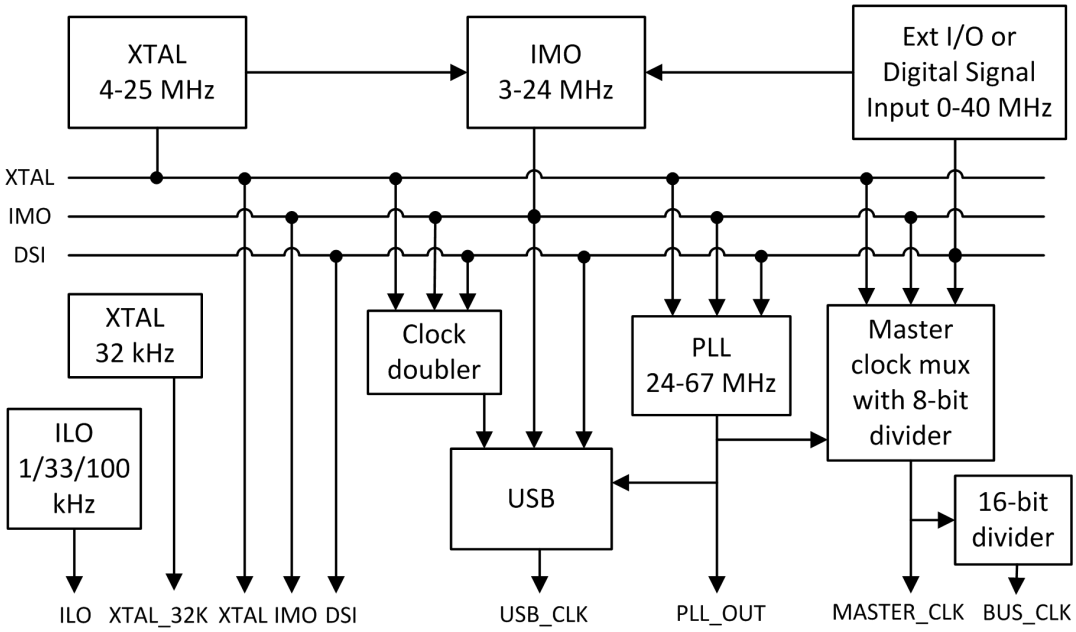
Name	Value
Vddd (V)	5.0
Vdda (V)	5.0
Vddio0 (V)	5.0
Vddio1 (V)	5.0
Vddio2 (V)	5.0
Vddio3 (V)	5.0
Temperature Range	-40C - 85/125C

4 Clocks

The clock system includes these clock resources:

- Four internal clock sources increase system integration:
  - 3 to 48 MHz Internal Main Oscillator (IMO) ±5% at 3 MHz
  - 1 kHz, 33 kHz, 100 kHz Internal Low Speed Oscillator (ILO) outputs
  - USB Clock Domain, sourced from IMO, MHz External Crystal Oscillator (MHzECO), and Digital System Interconnect (DSI)
  - 24 to 67 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 8 lists the system clocks used in this design.

Table 8. System Clocks

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
USB_CLK	DIGITAL	IMO	48	0	±0	False	False
BUS_CLK	DIGITAL	MASTER_CLK	76	76	±5	True	True
MASTER_CLK	DIGITAL	PLL_OUT	76	76	±5	True	True
Digital Signal	DIGITAL		0	0	±0	False	False
XTAL 32kHz	DIGITAL		0.0328	0	±0	False	False
XTAL	DIGITAL		33	0	±0	False	False
ILO	DIGITAL		0	0.001	-50,+100	True	True
PLL_OUT	DIGITAL	IMO	76	76	±5	True	True
IMO	DIGITAL		3	3	±5	True	True

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

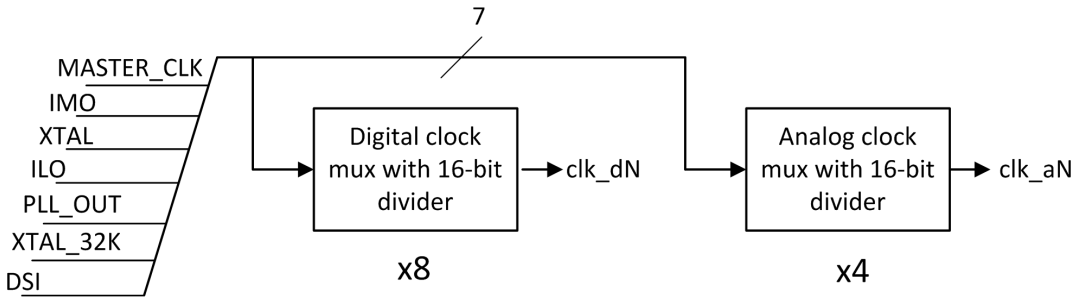


Table 9 lists the local clocks used in this design.

Table 9. Local Clocks

Name	Domain	Source	Desired Freq (MHz)	Nominal Freq (MHz)	Accuracy (%)	Start at Reset	Enabled
UART_XBee_-IntClock	DIGITAL	MASTER_CLK	0.9216	0.9268	±5	True	True
Clock_1	DIGITAL	MASTER_CLK	0.9195	0.9157	±5	True	True
Clock_4	DIGITAL	IMO	0.001	0.001	±5	True	True
ADCSharp_t-heACLK	ANALOG	MASTER_CLK	0.64	0.6387	±5	True	True
ADCSharp_E-xt_CP_Clk	DIGITAL	MASTER_CLK	2.56	2.5333	±5	True	True
Clock_2	DIGITAL	IMO	0.1	0.1	±5	True	True

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 5 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 10. Interrupts

Name	Priority	Vector
ADCSharp_IRQ	7	29
isr_Fin	7	2
isr_Fin_Emp	7	3
UART_Asser_RX_isr	7	0
UART_XBee_RX_isr	7	1

For more information on interrupts, please refer to:

- Interrupt Controller chapter in the PSoC 5 Technical Reference Manual
- Interrupts chapter in the System Reference Guide
  - CyInt API routines and related registers
- cy_isr component datasheet

5.2 DMAs

This design contains no DMA components.

6 Flash Memory

PSoC 5 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 11 lists the Flash protection settings for your design.

Table 11. 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 - External read protect (Factory upgrade)
- R - External write protect (Field upgrade)
- W - Full Protection

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

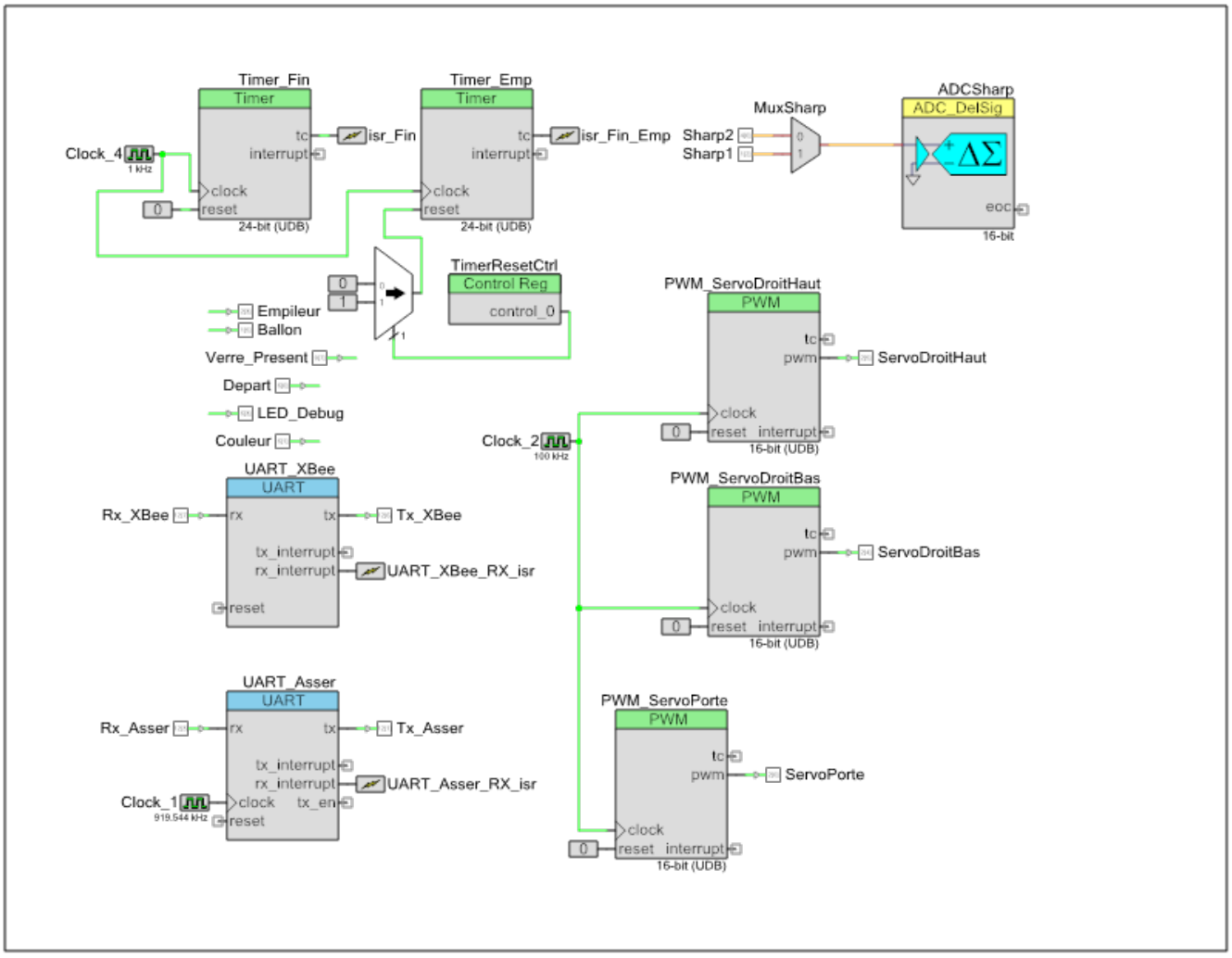
- Flash Protection chapter in the PSoC 5 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyFlash API routines
  - CyWrite API routines

7 Design Contents

This design's schematic content consists of the following schematic sheet:

7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1



This schematic sheet contains the following component instances:

- ADCSharp (type: ADC_DelSig_v2_30)
- MuxSharp (type: AMux_v1_70)
- PWM_ServoDroitBas (type: PWM_v2_10)
- PWM_ServoDroitHaut (type: PWM_v2_10)
- PWM_ServoPorte (type: PWM_v2_10)
- Timer_Emp (type: Timer_v2_40)
- Timer_Fin (type: Timer_v2_40)
- TimerResetCtrl (type: CyControlReg_v1_70)
- UART_Asser (type: UART_v2_30)
- UART_XBee (type: UART_v2_30)

8 Components

8.1 Component type: ADC_DelSig [v2.30]

8.1.1 Instance ADCSharp

Description: Delta-Sigma ADC
Instance type: ADC_DelSig [v2.30]
Datasheet: online component datasheet for ADC_DelSig

Table 12. Component Parameters for ADCSharp

Parameter Name	Value	Description
ADC_Charge_Pump_Clock	true	Low power charge pump clock selection
ADC_Clock	Internal	Parameter for selecting the ADC clock type.
ADC_Input_Mode	Single	Differential or Single ended input mode
ADC_Input_Range	Vssa to 6.144V (0.0 to 6*Vref)	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config2	Vssa to 1.024V (0.0 to Vref)	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config3	Vssa to 1.024V (0.0 to Vref)	Choose input operating mode that best supports the range of the signals being measured.
ADC_Input_Range_Config4	Vssa to 1.024V (0.0 to 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	16	ADC Resolution in bits
ADC_Resolution_Config2	16	ADC Resolution in bits
ADC_Resolution_Config3	16	ADC Resolution in bits
ADC_Resolution_Config4	16	ADC Resolution in bits
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.

Parameter Name	Value	Description
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	4	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].
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	Rail to Rail	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
Sample_Rate	10000	Sample Rate in Hz
Sample_Rate_Config2	10000	Sample Rate in Hz
Sample_Rate_Config3	10000	Sample Rate in Hz
Sample_Rate_Config4	10000	Sample Rate in Hz
sRate_Err	false	Parameter to hold the Conversion rate error status of ADC configuration.
Start_of_Conversion	Software	Continuous conversions or hardware controlled

8.2 Component type: AMux [v1.70]

8.2.1 Instance MuxSharp

Description: Multiplexer used to route analog signals.
Instance type: AMux [v1.70]
Datasheet: online component datasheet for AMux

Table 13. Component Parameters for MuxSharp

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.

8.3 Component type: CyControlReg [v1.70]

8.3.1 Instance TimerResetCtrl

Description: The Control Register allows the firmware to set values for to use for digital signals.
Instance type: CyControlReg [v1.70]
Datasheet: online component datasheet for CyControlReg

Table 14. Component Parameters for TimerResetCtrl

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
ExternalReset	false	Shows the reset terminal
NumOutputs	1	Defines the number of outputs needed (1-8)

8.4 Component type: PWM [v2.10]

8.4.1 Instance PWM_ServoDroitBas

Description: 8 or 16-bit Pulse Width Modulator
Instance type: PWM [v2.10]
Datasheet: online component datasheet for PWM

Table 15. Component Parameters for PWM_ServoDroitBas

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the Capture Input. What signal on the capture input is required to capture the current count value to the FIFO.

Parameter Name	Value	Description
CompareStatusEdgeSense	true	Enable Edge Sense detection on Compare ouptuts to the status register for use in Edge sensitive interrupts
CompareType1	Less	Compare Value Compare type setting for Compare 1 Output
CompareType2	Less	Compare Value Compare type setting for Compare 2 Output
CompareValue1	90	Compare Output 1 Compare to value
CompareValue2	101	Compare Output 2 Compare to value
DeadBand	Disabled	Defines whether Dead Band outputs are desired or not.
DeadTime	49	The number of DeadBand Clock cycles required
DitherOffset	0.00	Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM.
EnableMode	Software Only	Mode of Enabling the PWM
FixedFunction	false	Determines whether the Fixed Function Counter Timer is used or the UDB implementation is used.
InterruptOnCMP1	false	Enable an interrupt on Compare1 true event
InterruptOnCMP2	false	Enable an interrupt on Compare2 true event
InterruptOnKill	false	Enable Interrupt on a Kill Event
InterruptOnTC	false	Enable an interrupt on Terminal Count event
KillMode	Disabled	Kill mode selected for build time from one of the enumerated types
MinimumKillTime	1	Minimum number of clock cycles that kill must be active on the outputs when KillMode is set to Minimum Kill Time mode
Period	1999	PWM Period value
PWMMode	One Output	Overall Mode of the PWM selected from the enumerated types
Resolution	16	Bit Width of the PWM (8 or 16 bits)
RunMode	Continuous	Run Mode Options, Continuous or One Shot
TriggerMode	None	Mode of starting the PWM, i.e. triggering the PWM counter to start
UseInterrupt	true	Enable placement and usage of the status register

8.4.2 Instance PWM_ServoDroitHaut

Description: 8 or 16-bit Pulse Width Modulator
Instance type: PWM [v2.10]

Datasheet: [online component datasheet for PWM]

Table 16. Component Parameters for PWM_ServoDroitHaut

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the Capture Input. What signal on the capture input is required to capture the current count value to the FIFO.
CompareStatusEdgeSense	true	Enable Edge Sense detection on Compare ouptuts to the status register for use in Edge sensitive interrupts
CompareType1	Less	Compare Value Compare type setting for Compare 1 Output
CompareType2	Less	Compare Value Compare type setting for Compare 2 Output
CompareValue1	115	Compare Output 1 Compare to value
CompareValue2	101	Compare Output 2 Compare to value
DeadBand	Disabled	Defines whether Dead Band outputs are desired or not.
DeadTime	49	The number of DeadBand Clock cycles required
DitherOffset	0.00	Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM.
EnableMode	Software Only	Mode of Enabling the PWM
FixedFunction	false	Determines whether the Fixed Function Counter Timer is used or the UDB implementation is used.
InterruptOnCMP1	false	Enable an interrupt on Compare1 true event
InterruptOnCMP2	false	Enable an interrupt on Compare2 true event
InterruptOnKill	false	Enable Interrupt on a Kill Event
InterruptOnTC	false	Enable an interrupt on Terminal Count event
KillMode	Disabled	Kill mode selected for build time from one of the enumerated types
MinimumKillTime	1	Minimum number of clock cycles that kill must be active on the outputs when KillMode is set to Minimum Kill Time mode
Period	1999	PWM Period value
PWMMode	One Output	Overall Mode of the PWM selected from the enumerated types
Resolution	16	Bit Width of the PWM (8 or 16 bits)
RunMode	Continuous	Run Mode Options, Continuous or One Shot
TriggerMode	None	Mode of starting the PWM, i.e. triggering the PWM counter to start

Parameter Name	Value	Description
UseInterrupt	true	Enable placement and usage of the status register

8.4.3 Instance PWM_ServoPorte

Description: 8 or 16-bit Pulse Width Modulator
Instance type: PWM [v2.10]
Datasheet: online component datasheet for PWM

Table 17. Component Parameters for PWM_ServoPorte

Parameter Name	Value	Description
CaptureMode	None	Defines the functionality of the Capture Input. What signal on the capture input is required to capture the current count value to the FIFO.
CompareStatusEdgeSense	true	Enable Edge Sense detection on Compare ouptuts to the status register for use in Edge sensitive interrupts
CompareType1	Less	Compare Value Compare type setting for Compare 1 Output
CompareType2	Less	Compare Value Compare type setting for Compare 2 Output
CompareValue1	73	Compare Output 1 Compare to value
CompareValue2	101	Compare Output 2 Compare to value
DeadBand	Disabled	Defines whether Dead Band outputs are desired or not.
DeadTime	49	The number of DeadBand Clock cycles required
DitherOffset	0.00	Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM.
EnableMode	Software Only	Mode of Enabling the PWM
FixedFunction	false	Determines whether the Fixed Function Counter Timer is used or the UDB implementation is used.
InterruptOnCMP1	false	Enable an interrupt on Compare1 true event
InterruptOnCMP2	false	Enable an interrupt on Compare2 true event
InterruptOnKill	false	Enable Interrupt on a Kill Event
InterruptOnTC	false	Enable an interrupt on Terminal Count event
KillMode	Disabled	Kill mode selected for build time from one of the enumerated types
MinimumKillTime	1	Minimum number of clock cycles that kill must be active on the outputs when KillMode is set to Minimum Kill Time mode
Period	1999	PWM Period value

Parameter Name	Value	Description
PWMMode	One Output	Overall Mode of the PWM selected from the enumerated types
Resolution	16	Bit Width of the PWM (8 or 16 bits)
RunMode	Continuous	Run Mode Options, Continuous or One Shot
TriggerMode	None	Mode of starting the PWM, i.e. triggering the PWM counter to start
UseInterrupt	true	Enable placement and usage of the status register

8.5 Component type: Timer [v2.40]

8.5.1 Instance Timer_Emp

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.40]
Datasheet: online component datasheet for Timer

Table 18. Component Parameters for Timer_Emp

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.

Parameter Name	Value	Description
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	4999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	24	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 (Halt On Interrupt)	Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered.
TriggerMode	None	Defines the required trigger input signal to cause a valid trigger enable of the timer

8.5.2 Instance Timer_Fin

Description: 8, 16, 24 or 32-bit Timer
Instance type: Timer [v2.40]
Datasheet: online component datasheet for Timer

Table 19. Component Parameters for Timer_Fin

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.

Parameter Name	Value	Description
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	86999	Defines the timer period (This is also the reload value when terminal count is reached)
Resolution	24	Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals.
RunMode	Continuous	Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered.
TriggerMode	None	Defines the required trigger input signal to cause a valid trigger enable of the timer

8.6 Component type: UART [v2.30]

8.6.1 Instance UART_Asser

Description: Universal Asynchronous Receiver Transmitter
Instance type: UART [v2.30]
Datasheet: online component datasheet for UART

Table 20. Component Parameters for UART_Asser

Parameter Name	Value	Description
Address1	0	This parameter specifies the RX Hardware Address #1.

Parameter Name	Value	Description
Address2	0	This parameter specifies the RX Hardware Address #2.
BaudRate	115200	Sets the target baud rate.
BreakBitsRX	13	Specifies the break signal length for the RX (detection) channel.
BreakBitsTX	13	Specifies the break signal length for the TX channel.
BreakDetect	false	Enables the break detect hardware.
CRCoutputsEn	false	Enables the CRC outputs.
EnIntRXInterrupt	false	Enables the internal RX interrupt configuration and the ISR.
EnIntTXInterrupt	false	Enables the internal TX interrupt configuration and the ISR.
FlowControl	None	Enable the flow control signals.
HalfDuplexEn	false	Enables half duplex mode on the RX Half of the UART module.
HwTXEnSignal	true	Enables the external TX enable signal output.
InternalClock	false	Enables the internal clock. This parameter removes the clock input pin.
InterruptOnTXComplete	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX complete' event.
InterruptOnTXFifoEmpty	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO empty' event.
InterruptOnTXFifoFull	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO full' event.
InterruptOnTXFifoNotFull	false	This is an Interrupt mask used to enable/disable the interrupt on 'TX FIFO not full' event.
IntOnAddressDetect	false	Enables the interrupt on hardware address detected event by default
IntOnAddressMatch	false	Enables the interrupt on hardware address match detected event by default
IntOnBreak	false	Enables the interrupt on break signal detected event by default
IntOnByteRcvd	true	Enables the interrupt on RX byte received event by default
IntOnOverrunError	false	Enables the interrupt on overrun error event by default
IntOnParityError	false	Enables the interrupt on parity error event by default
IntOnStopError	false	Enables the interrupt on stop error event by default
NumDataBits	8	Defines the number of data bits. Values can be 5, 6, 7 or 8 bits.
NumStopBits	1	Defines the number of stop bits. Values can be 1 or 2 bits.

Parameter Name	Value	Description
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	1	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	1	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.

8.6.2 Instance UART_XBee

Description: Universal Asynchronous Receiver Transmitter
Instance type: UART [v2.30]
Datasheet: online component datasheet for UART

Table 21. Component Parameters for UART_XBee

Parameter Name	Value	Description
Address1	0	This parameter specifies the RX Hardware Address #1.
Address2	0	This parameter specifies the RX Hardware Address #2.
BaudRate	115200	Sets the target baud rate.
BreakBitsRX	13	Specifies the break signal length for the RX (detection) channel.
BreakBitsTX	13	Specifies the break signal length for the TX channel.
BreakDetect	false	Enables the break detect hardware.
CRCoutputsEn	false	Enables the CRC outputs.
EnIntRXInterrupt	false	Enables the internal RX interrupt configuration and the ISR.
EnIntTXInterrupt	false	Enables the internal TX interrupt configuration and the ISR.
FlowControl	None	Enable the flow control signals.
HalfDuplexEn	false	Enables half duplex mode on the RX Half of the UART module.

Parameter Name	Value	Description
HwTXEnSignal	false	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
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.

Parameter Name	Value	Description
TXEnable	true	Enables the TX in the UART
Use23Polling	true	Allows the use of 2 out of 3 polling resources on the RX UART sampler.

9 Other Resources

The following documents contain important information on Cypress software APIs that might be relevant to this design:

- Standard Types and Defines chapter in the System Reference Guide
  - Software base types
  - Hardware register types
  - Compiler defines
  - Cypress API return codes
  - Interrupt types and macros
- Registers
  - The full PSoC 5 register map is covered in the PSoC 5 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 5 Technical Reference Manual
  - Low Power Modes chapter in the PSoC 5 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 5 Technical Reference Manual
  - Cache chapter in the System Reference Guide
    - CyFlushCache() API routine