# PSoC® Creator™
# Project Datasheet for MainBoard

**Creation Time: 09/30/2024 13:01:11**
**User: MIDNIGHT-VA5S7H\JusticeV**
**Project: MainBoard**
**Tool: PSoC Creator 3.3 CP3**

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

### Copyright



### Trademarks

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

### Philips I2C Patent Rights

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

### Disclaimer

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

### Flash Code Protection

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

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

# Table of Contents

# 1 Overview

The Cypress PSoC 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 family member PSoC 5 device. For details on all the systems listed above, please refer to the PSoC 5 Technical Reference Manual .

Figure 1. CY8C58LP Device Family Block Diagram

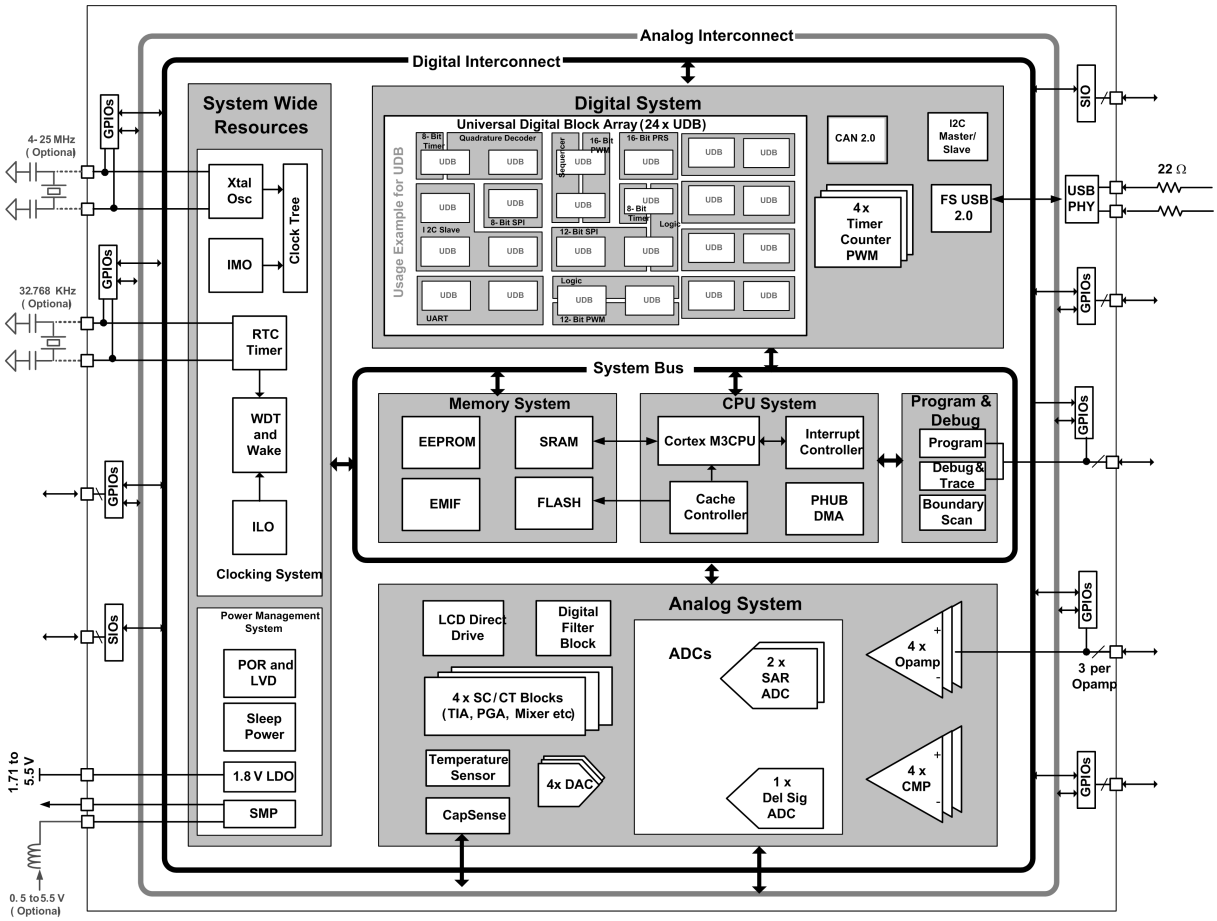


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

| Name | Value |
|---|---|
| Part Number | CY8C5868AXI-LP035 |
| Package Name | 100-TQFP |
| Architecture | PSoC 5 |
| Family | CY8C58LP |
| CPU speed (MHz) | 67 |
| Flash size (kBytes) | 256 |
| SRAM size (kBytes) | 64 |
| EEPROM size (Bytes) | 2048 |
| Vdd range (V) | 1.71 to 5.5 |
| Automotive qualified | No (Industrial Grade Only) |
| Temp range (Celcius) | -40 to 85 |
| JTAG ID | 0x2E123069 |

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

Table 2 lists the device resources that this design uses:

Table 2. Device Resources

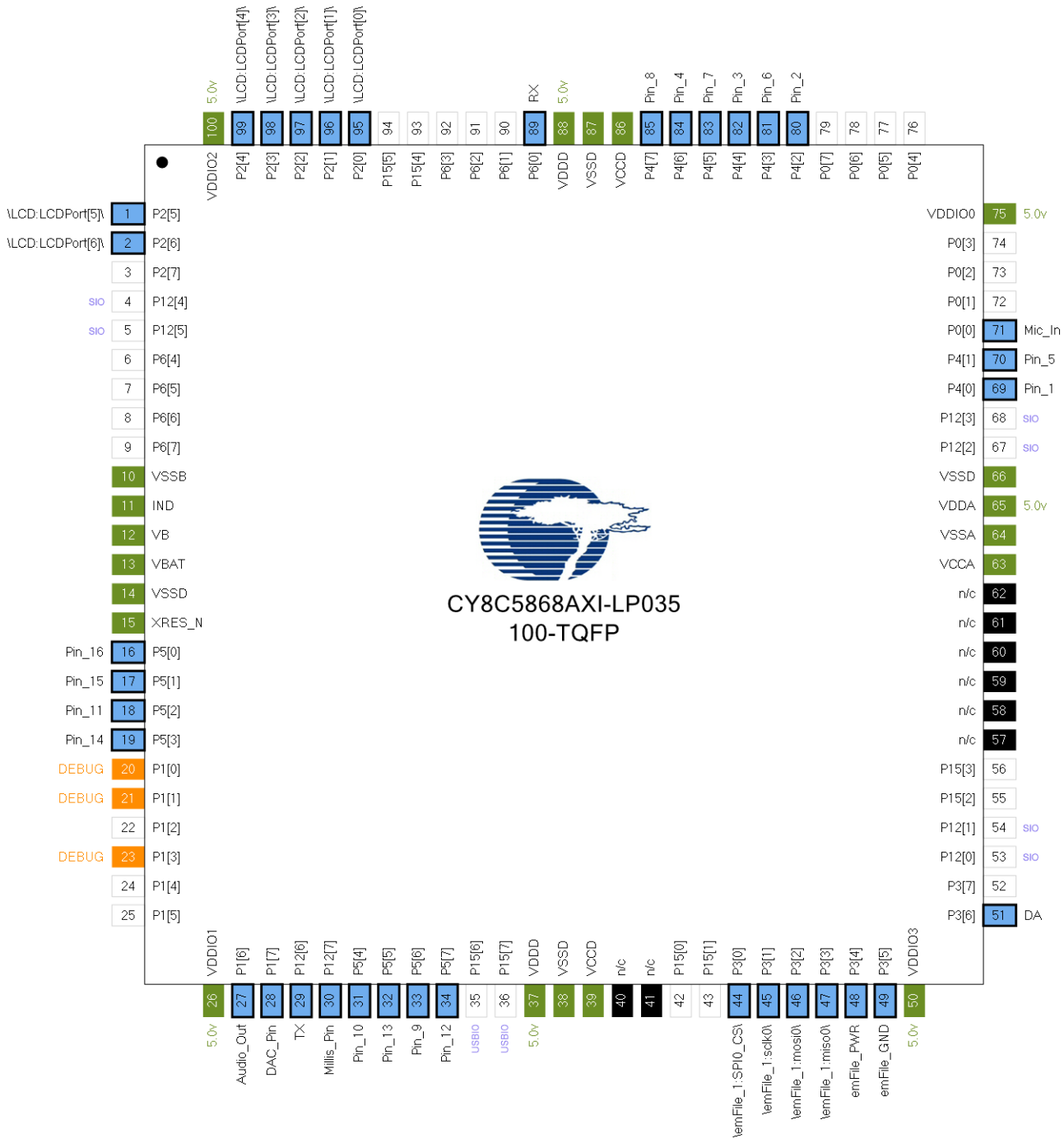
| Resource Type | Used | Free | Max | % Used |
|---|---|---|---|---|
| Digital Clocks | 5 | 3 | 8 | 62.50 % |
| Analog Clocks | 1 | 3 | 4 | 25.00 % |
| CapSense Buffers | 0 | 2 | 2 | 0.00 % |
| Digital Filter Block | 0 | 1 | 1 | 0.00 % |
| Interrupts | 4 | 28 | 32 | 12.50 % |
| IO | 39 | 33 | 72 | 54.17 % |
| Segment LCD | 0 | 1 | 1 | 0.00 % |
| CAN 2.0b | 0 | 1 | 1 | 0.00 % |
| I2C | 0 | 1 | 1 | 0.00 % |
| USB | 0 | 1 | 1 | 0.00 % |
| DMA Channels | 0 | 24 | 24 | 0.00 % |
| Timer | 0 | 4 | 4 | 0.00 % |
| UDB | | | | |
| Macrocells | 48 | 144 | 192 | 25.00 % |
| Unique P-terms | 96 | 288 | 384 | 25.00 % |
| Total P-terms | 111 | | | |
| Datapath Cells | 6 | 18 | 24 | 25.00 % |
| Status Cells | 7 | 17 | 24 | 29.17 % |
| StatusI Registers | 5 | | | |
| Routed Count7 Load/Enable | 2 | | | |
| Control Cells | 3 | 21 | 24 | 12.50 % |
| Control Registers | 1 | | | |
| Count7 Cells | 2 | | | |
| Opamp | 0 | 4 | 4 | 0.00 % |
| Comparator | 0 | 4 | 4 | 0.00 % |
| Delta-Sigma ADC | 1 | 0 | 1 | 100.00 % |
| LPF | 0 | 2 | 2 | 0.00 % |
| SAR ADC | 0 | 2 | 2 | 0.00 % |
| Analog (SC/CT) Blocks | 0 | 4 | 4 | 0.00 % |
| DAC | | | | |

| Resource Type | Used | Free | Max | % Used |
|---|---|---|---|---|
| VIDAC | 1 | 3 | 4 | 25.00 % |

# 2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



## 2.1 Hardware Pins

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

Table 3. Device Pins

| Pin | Port | Name | Type | Drive Mode | Reset State |
|---|---|---|---|---|---|
| 1 | P2[5] | \LCD:LCDPort[5]\ | Software Output | Strong drive | HiZ Analog Unb |
| 2 | P2[6] | \LCD:LCDPort[6]\ | Software Output | Strong drive | HiZ Analog Unb |
| 3 | P2[7] | GPIO [unused] | | | HiZ Analog Unb |
| 4 | P12[4] | SIO [unused] | | | HiZ Analog Unb |
| 5 | P12[5] | SIO [unused] | | | HiZ Analog Unb |
| 6 | P6[4] | GPIO [unused] | | | HiZ Analog Unb |
| 7 | P6[5] | GPIO [unused] | | | HiZ Analog Unb |
| 8 | P6[6] | GPIO [unused] | | | HiZ Analog Unb |
| 9 | P6[7] | GPIO [unused] | | | HiZ Analog Unb |
| 10 | VSSB | VSSB | Dedicated | | |
| 11 | IND | IND | Dedicated | | |
| 12 | VB | VB | Dedicated | | |
| 13 | VBAT | VBAT | Dedicated | | |
| 14 | VSSD | VSSD | Power | | |
| 15 | XRES_N | XRES_N | Dedicated | | |
| 16 | P5[0] | Pin_16 | Software Output | Strong drive | HiZ Analog Unb |
| 17 | P5[1] | Pin_15 | Software Output | Strong drive | HiZ Analog Unb |
| 18 | P5[2] | Pin_11 | Software Output | Strong drive | HiZ Analog Unb |
| 19 | P5[3] | Pin_14 | Software Output | Strong drive | HiZ Analog Unb |
| 20 | P1[0] | Debug:SWD_IO | Reserved | | |
| 21 | P1[1] | Debug:SWD_CK | Reserved | | |
| 22 | P1[2] | GPIO [unused] | | | HiZ Analog Unb |
| 23 | P1[3] | Debug:SWV | Reserved | | |
| 24 | P1[4] | GPIO [unused] | | | HiZ Analog Unb |
| 25 | P1[5] | GPIO [unused] | | | HiZ Analog Unb |
| 26 | VDDIO1 | VDDIO1 | Power | | |
| 27 | P1[6] | Audio_Out | Analog | HiZ analog | HiZ Analog Unb |
| 28 | P1[7] | DAC_Pin | A/D Out | Strong drive | HiZ Analog Unb |
| 29 | P12[6] | TX | Dgtl Out | Strong drive | HiZ Analog Unb |
| 30 | P12[7] | Millis_Pin | Dgtl Out | Strong drive | HiZ Analog Unb |
| 31 | P5[4] | Pin_10 | Software Output | Strong drive | HiZ Analog Unb |
| 32 | P5[5] | Pin_13 | Software Output | Strong drive | HiZ Analog Unb |
| 33 | P5[6] | Pin_9 | Software Output | Strong drive | HiZ Analog Unb |
| 34 | P5[7] | Pin_12 | Software Output | Strong drive | HiZ Analog Unb |
| 35 | P15[6] | USB IO [unused] | | | HiZ Analog Unb |
| 36 | P15[7] | USB IO [unused] | | | HiZ Analog Unb |
| 37 | VDDD | VDDD | Power | | |

| Pin | Port | Name | Type | Drive Mode | Reset State |
|---|---|---|---|---|---|
| 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] | \emFile_1:SPI0_CS\ | Software Output | Strong drive | HiZ Analog Unb |
| 45 | P3[1] | \emFile_1:sclk0\ | Dgtl Out | Strong drive | HiZ Analog Unb |
| 46 | P3[2] | \emFile_1:mosi0\ | Dgtl Out | Strong drive | HiZ Analog Unb |
| 47 | P3[3] | \emFile_1:miso0\ | Dgtl In | HiZ digital | HiZ Analog Unb |
| 48 | P3[4] | emFile_PWR | Software Input | Res pull up | HiZ Analog Unb |
| 49 | P3[5] | emFile_GND | Software Input | Res pull up | HiZ Analog Unb |
| 50 | VDDIO3 | VDDIO3 | Power | | |
| 51 | P3[6] | DA | Software Input | Res pull down | HiZ Analog Unb |
| 52 | P3[7] | GPIO [unused] | | | HiZ Analog Unb |
| 53 | P12[0] | SIO [unused] | | | HiZ Analog Unb |
| 54 | P12[1] | SIO [unused] | | | HiZ Analog Unb |
| 55 | P15[2] | GPIO [unused] | | | HiZ Analog Unb |
| 56 | P15[3] | GPIO [unused] | | | HiZ Analog Unb |
| 63 | VCCA | VCCA | Power | | |
| 64 | VSSA | VSSA | Power | | |
| 65 | VDDA | VDDA | Power | | |
| 66 | VSSD | VSSD | Power | | |
| 67 | P12[2] | SIO [unused] | | | HiZ Analog Unb |
| 68 | P12[3] | SIO [unused] | | | HiZ Analog Unb |
| 69 | P4[0] | Pin_1 | Software Input | Res pull down | HiZ Analog Unb |
| 70 | P4[1] | Pin_5 | Software Input | Res pull down | HiZ Analog Unb |
| 71 | P0[0] | Mic_In | Analog | HiZ analog | HiZ Analog Unb |
| 72 | P0[1] | GPIO [unused] | | | HiZ Analog Unb |
| 73 | P0[2] | GPIO [unused] | | | HiZ Analog Unb |
| 74 | P0[3] | GPIO [unused] | | | HiZ Analog Unb |
| 75 | VDDIO0 | VDDIO0 | Power | | |
| 76 | P0[4] | GPIO [unused] | | | HiZ Analog Unb |
| 77 | P0[5] | GPIO [unused] | | | HiZ Analog Unb |
| 78 | P0[6] | GPIO [unused] | | | HiZ Analog Unb |
| 79 | P0[7] | GPIO [unused] | | | HiZ Analog Unb |
| 80 | P4[2] | Pin_2 | Software Input | Res pull down | HiZ Analog Unb |
| 81 | P4[3] | Pin_6 | Software Input | Res pull down | HiZ Analog Unb |
| 82 | P4[4] | Pin_3 | Software Input | Res pull down | HiZ Analog Unb |
| 83 | P4[5] | Pin_7 | Software Input | Res pull down | HiZ Analog Unb |
| 84 | P4[6] | Pin_4 | Software Input | Res pull down | HiZ Analog Unb |
| 85 | P4[7] | Pin_8 | Software Input | Res pull down | HiZ Analog Unb |
| 86 | VCCD | VCCD | Power | | |
| 87 | VSSD | VSSD | Power | | |
| 88 | VDDD | VDDD | Power | | |

| Pin | Port | Name | Type | Drive Mode | Reset State |
|---|---|---|---|---|---|
| 89 | P6[0] | RX | Dgtl In | HiZ digital | HiZ Analog Unb |
| 90 | P6[1] | GPIO [unused] | | | HiZ Analog Unb |
| 91 | P6[2] | GPIO [unused] | | | HiZ Analog Unb |
| 92 | P6[3] | GPIO [unused] | | | HiZ Analog Unb |
| 93 | P15[4] | GPIO [unused] | | | HiZ Analog Unb |
| 94 | P15[5] | GPIO [unused] | | | HiZ Analog Unb |
| 95 | P2[0] | \LCD:LCDPort[0]\ | Software Output | Strong drive | HiZ Analog Unb |
| 96 | P2[1] | \LCD:LCDPort[1]\ | Software Output | Strong drive | HiZ Analog Unb |
| 97 | P2[2] | \LCD:LCDPort[2]\ | Software Output | Strong drive | HiZ Analog Unb |
| 98 | P2[3] | \LCD:LCDPort[3]\ | Software Output | Strong drive | HiZ Analog Unb |
| 99 | P2[4] | \LCD:LCDPort[4]\ | Software Output | Strong drive | HiZ Analog Unb |
| 100 | VDDIO2 | VDDIO2 | Power | | |

Abbreviations used in Table 3 have the following meanings:

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

## 2.2 Hardware Ports

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

Table 4. Device Ports

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

| Port | Pin | Name | Type | Drive Mode | Reset State |
|---|---|---|---|---|---|
| P2[7] | 3 | GPIO [unused] | | | HiZ Analog Unb |
| P3[0] | 44 | \emFile_1:SPI0_CS\ | Software Output | Strong drive | HiZ Analog Unb |
| P3[1] | 45 | \emFile_1:sclk0\ | Dgtl Out | Strong drive | HiZ Analog Unb |
| P3[2] | 46 | \emFile_1:mosi0\ | Dgtl Out | Strong drive | HiZ Analog Unb |
| P3[3] | 47 | \emFile_1:miso0\ | Dgtl In | HiZ digital | HiZ Analog Unb |
| P3[4] | 48 | emFile_PWR | Software Input | Res pull up | HiZ Analog Unb |
| P3[5] | 49 | emFile_GND | Software Input | Res pull up | HiZ Analog Unb |
| P3[6] | 51 | DA | Software Input | Res pull down | HiZ Analog Unb |
| P3[7] | 52 | GPIO [unused] | | | HiZ Analog Unb |
| P4[0] | 69 | Pin_1 | Software Input | Res pull down | HiZ Analog Unb |
| P4[1] | 70 | Pin_5 | Software Input | Res pull down | HiZ Analog Unb |
| P4[2] | 80 | Pin_2 | Software Input | Res pull down | HiZ Analog Unb |
| P4[3] | 81 | Pin_6 | Software Input | Res pull down | HiZ Analog Unb |
| P4[4] | 82 | Pin_3 | Software Input | Res pull down | HiZ Analog Unb |
| P4[5] | 83 | Pin_7 | Software Input | Res pull down | HiZ Analog Unb |
| P4[6] | 84 | Pin_4 | Software Input | Res pull down | HiZ Analog Unb |
| P4[7] | 85 | Pin_8 | Software Input | Res pull down | HiZ Analog Unb |
| P5[0] | 16 | Pin_16 | Software Output | Strong drive | HiZ Analog Unb |
| P5[1] | 17 | Pin_15 | Software Output | Strong drive | HiZ Analog Unb |
| P5[2] | 18 | Pin_11 | Software Output | Strong drive | HiZ Analog Unb |
| P5[3] | 19 | Pin_14 | Software Output | Strong drive | HiZ Analog Unb |
| P5[4] | 31 | Pin_10 | Software Output | Strong drive | HiZ Analog Unb |
| P5[5] | 32 | Pin_13 | Software Output | Strong drive | HiZ Analog Unb |
| P5[6] | 33 | Pin_9 | Software Output | Strong drive | HiZ Analog Unb |
| P5[7] | 34 | Pin_12 | Software Output | Strong drive | HiZ Analog Unb |
| P6[0] | 89 | RX | Dgtl In | HiZ digital | HiZ Analog Unb |
| P6[1] | 90 | GPIO [unused] | | | HiZ Analog Unb |
| P6[2] | 91 | GPIO [unused] | | | HiZ Analog Unb |
| P6[3] | 92 | GPIO [unused] | | | HiZ Analog Unb |
| P6[4] | 6 | GPIO [unused] | | | HiZ Analog Unb |
| P6[5] | 7 | GPIO [unused] | | | HiZ Analog Unb |
| P6[6] | 8 | GPIO [unused] | | | HiZ Analog Unb |
| P6[7] | 9 | GPIO [unused] | | | HiZ Analog Unb |

Abbreviations used in Table 4 have the following meanings:

- HiZ analog = High impedance analog

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

## 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 |
|---|---|---|---|
| \emFile_1:miso0\ | P3[3] | Dgtl In | HiZ Analog Unb |
| \emFile_1:mosi0\ | P3[2] | Dgtl Out | HiZ Analog Unb |
| \emFile_1:sclk0\ | P3[1] | Dgtl Out | HiZ Analog Unb |
| \emFile_1:SPI0_CS\ | P3[0] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[0]\ | P2[0] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[1]\ | P2[1] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[2]\ | P2[2] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[3]\ | P2[3] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[4]\ | P2[4] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[5]\ | P2[5] | Software Output | HiZ Analog Unb |
| \LCD:LCDPort[6]\ | P2[6] | Software Output | HiZ Analog Unb |
| Audio_Out | P1[6] | Analog | HiZ Analog Unb |
| DA | P3[6] | Software Input | HiZ Analog Unb |
| DAC_Pin | P1[7] | A/D Out | HiZ Analog Unb |
| Debug:SWD_CK | P1[1] | Reserved | |
| Debug:SWD_IO | P1[0] | Reserved | |
| Debug:SWV | P1[3] | Reserved | |
| emFile_GND | P3[5] | Software Input | HiZ Analog Unb |
| emFile_PWR | P3[4] | Software Input | HiZ Analog Unb |
| GPIO [unused] | P0[7] | | HiZ Analog Unb |
| GPIO [unused] | P6[1] | | HiZ Analog Unb |
| GPIO [unused] | P0[3] | | HiZ Analog Unb |
| GPIO [unused] | P0[4] | | HiZ Analog Unb |
| GPIO [unused] | P6[4] | | HiZ Analog Unb |
| GPIO [unused] | P0[6] | | HiZ Analog Unb |
| GPIO [unused] | P0[5] | | HiZ Analog Unb |
| GPIO [unused] | P0[2] | | HiZ Analog Unb |
| GPIO [unused] | P6[6] | | HiZ Analog Unb |
| GPIO [unused] | P6[5] | | HiZ Analog Unb |
| GPIO [unused] | P2[7] | | HiZ Analog Unb |
| GPIO [unused] | P15[3] | | HiZ Analog Unb |
| GPIO [unused] | P0[1] | | HiZ Analog Unb |
| GPIO [unused] | P3[7] | | HiZ Analog Unb |
| GPIO [unused] | P15[2] | | HiZ Analog Unb |
| GPIO [unused] | P1[4] | | HiZ Analog Unb |
| GPIO [unused] | P1[2] | | HiZ Analog Unb |

| Name | Port | Type | Reset State |
|---|---|---|---|
| GPIO [unused] | P15[4] | | HiZ Analog Unb |
| GPIO [unused] | P1[5] | | HiZ Analog Unb |
| GPIO [unused] | P15[5] | | HiZ Analog Unb |
| GPIO [unused] | P15[1] | | HiZ Analog Unb |
| GPIO [unused] | P6[7] | | HiZ Analog Unb |
| GPIO [unused] | P15[0] | | HiZ Analog Unb |
| GPIO [unused] | P6[3] | | HiZ Analog Unb |
| GPIO [unused] | P6[2] | | HiZ Analog Unb |
| Mic_In | P0[0] | Analog | HiZ Analog Unb |
| Millis_Pin | P12[7] | Dgtl Out | HiZ Analog Unb |
| Pin_1 | P4[0] | Software Input | HiZ Analog Unb |
| Pin_10 | P5[4] | Software Output | HiZ Analog Unb |
| Pin_11 | P5[2] | Software Output | HiZ Analog Unb |
| Pin_12 | P5[7] | Software Output | HiZ Analog Unb |
| Pin_13 | P5[5] | Software Output | HiZ Analog Unb |
| Pin_14 | P5[3] | Software Output | HiZ Analog Unb |
| Pin_15 | P5[1] | Software Output | HiZ Analog Unb |
| Pin_16 | P5[0] | Software Output | HiZ Analog Unb |
| Pin_2 | P4[2] | Software Input | HiZ Analog Unb |
| Pin_3 | P4[4] | Software Input | HiZ Analog Unb |
| Pin_4 | P4[6] | Software Input | HiZ Analog Unb |
| Pin_5 | P4[1] | Software Input | HiZ Analog Unb |
| Pin_6 | P4[3] | Software Input | HiZ Analog Unb |
| Pin_7 | P4[5] | Software Input | HiZ Analog Unb |
| Pin_8 | P4[7] | Software Input | HiZ Analog Unb |
| Pin_9 | P5[6] | Software Output | HiZ Analog Unb |
| RX | P6[0] | Dgtl In | HiZ Analog Unb |
| SIO [unused] | P12[3] | | HiZ Analog Unb |
| SIO [unused] | P12[5] | | HiZ Analog Unb |
| SIO [unused] | P12[4] | | HiZ Analog Unb |
| SIO [unused] | P12[0] | | HiZ Analog Unb |
| SIO [unused] | P12[2] | | HiZ Analog Unb |
| SIO [unused] | P12[1] | | HiZ Analog Unb |
| TX | P12[6] | Dgtl Out | HiZ Analog Unb |
| USB IO [unused] | P15[6] | | HiZ Analog Unb |
| USB IO [unused] | P15[7] | | HiZ Analog Unb |

Abbreviations used in Table 5 have the following meanings:

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

- Dgtl Out = Digital Output
- A/D Out = Analog / 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) | 0x1000 |
| Stack Size (bytes) | 0x4000 |
| 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

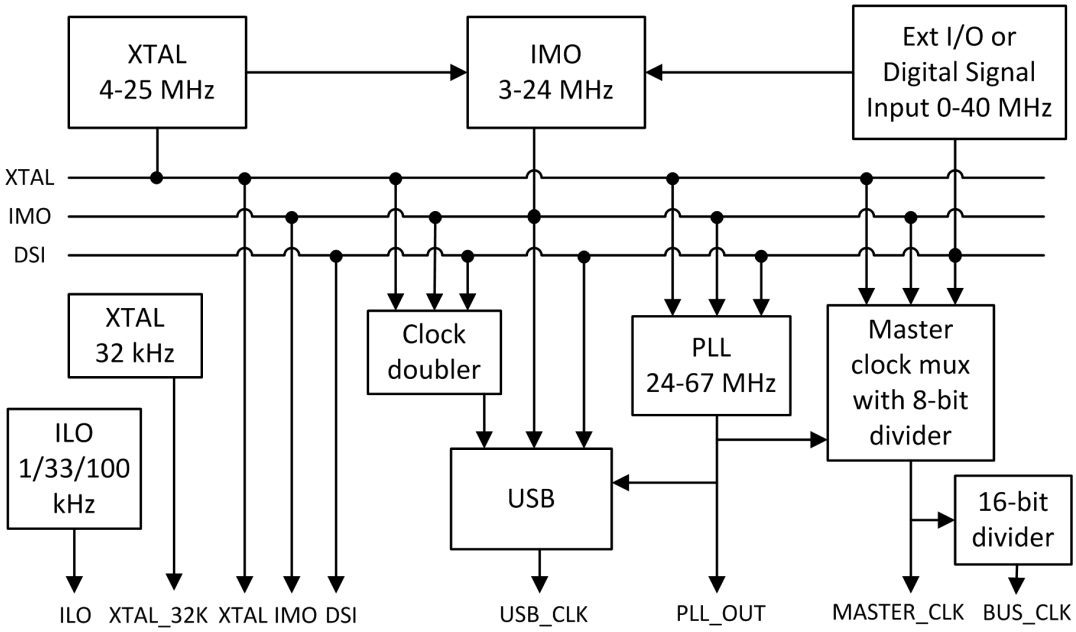
| Name | Value |
|---|---|
| Variable VDDA | False |
| VDDA (V) | 5.0 |
| VDDD (V) | 5.0 |
| VDDIO0 (V) | 5.0 |
| VDDIO1 (V) | 5.0 |
| VDDIO2 (V) | 5.0 |
| VDDIO3 (V) | 5.0 |
| Temperature Range | -40C - 85/125C |

# 4 Clocks

The clock system includes these clock resources:

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

Figure 3. System Clock Configuration



## 4.1 System Clocks

Table 9 lists the system clocks used in this design.

Table 9. System Clocks

| Name | Domain | Source | Desired Freq | Nominal Freq | Accuracy (%) | Start at Reset | Enabled |
|---|---|---|---|---|---|---|---|
| BUS_CLK | DIGITAL | MASTER_CLK | ? MHz | 24 MHz | ±1 | True | True |
| PLL_OUT | DIGITAL | IMO | 24 MHz | 24 MHz | ±1 | True | True |
| MASTER_CLK | DIGITAL | PLL_OUT | ? MHz | 24 MHz | ±1 | True | True |
| IMO | DIGITAL | | 3 MHz | 3 MHz | ±1 | True | True |
| ILO | DIGITAL | | ? MHz | 1 kHz | -50,+100 | True | True |
| USB_CLK | DIGITAL | IMO | 48 MHz | ? MHz | ±0 | False | False |
| XTAL | DIGITAL | | 25 MHz | ? MHz | ±0 | False | False |
| XTAL 32kHz | DIGITAL | | 32.768 kHz | ? MHz | ±0 | False | False |
| Digital Signal | DIGITAL | | ? MHz | ? MHz | ±0 | False | False |

## 4.2 Local and Design Wide Clocks

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

Figure 4. Local and Design Wide Clock Configuration

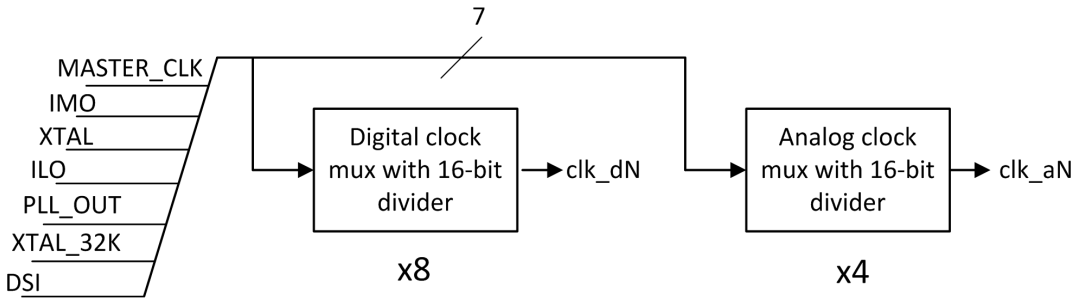



Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

| Name | Domain | Source | Desired Freq | Nominal Freq | Accuracy (%) | Start at Reset | Enabled |
|---|---|---|---|---|---|---|---|
| emFile_1_-Clock_1 | DIGITAL | MASTER_CLK | ? MHz | 24 MHz | ±1 | True | True |
| Mic_ADC_Ext_-CP_Clk | DIGITAL | MASTER_CLK | ? MHz | 24 MHz | ±1 | True | True |
| Clock_1 | DIGITAL | MASTER_CLK | 12 MHz | 12 MHz | ±1 | True | True |
| Mic_ADC_-theACLK | ANALOG | MASTER_CLK | 176 kHz | 176.471 kHz | ±1 | True | True |
| UART_IntClock | DIGITAL | MASTER_CLK | 76.8 kHz | 76.677 kHz | ±1 | True | True |
| Millis_Clock | DIGITAL | ILO | 1 kHz | 1 kHz | -50,+100 | 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 11. Interrupts

| Name | Priority | Vector |
|---|---|---|
| Mic_ADC_IRQ | 7 | 29 |
| rx_int | 7 | 0 |
| sample_int | 7 | 5 |
| tx_int | 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
- Datasheet for cy_isr component

## 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 12 lists the Flash protection settings for your design.

Table 12. Flash Protection Settings

| Start Address | End Address | Protection Level |
|---|---|---|
| 0x0 | 0x3FFFF | U - Unprotected |

Flash memory is organized as rows with each row of flash having 256 bytes. Each flash row can be assigned one of four protection levels:

- U - Unprotected
- F - Factory Upgrade
- R - Field Upgrade
- W - Full Protection

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

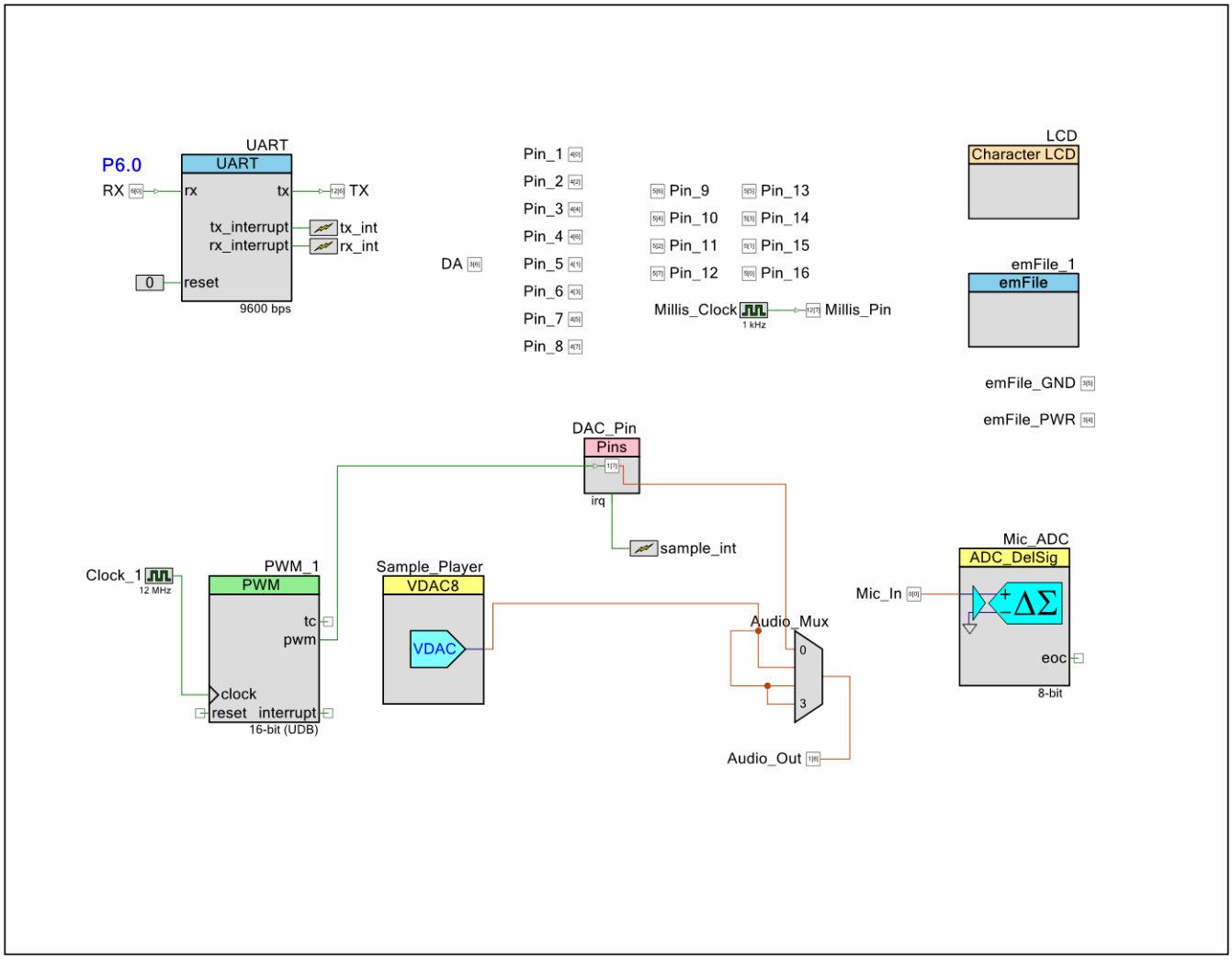
- Flash Protection chapter in the PSoC 5 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyWrite API routines
  - CyFlash API routines

# 7 Design Contents

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

## 7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1




This schematic sheet contains the following component instances:

- Instance Audio_Mux (type: AMux_v1_80)
- Instance emFile_1 (type: emFile_v1_20)
- Instance LCD (type: CharLCD_v2_20)
- Instance Mic_ADC (type: ADC_DelSig_v3_20)
- Instance PWM_1 (type: PWM_v3_30)
- Instance Sample_Player (type: VDAC8_v1_90)
- Instance UART (type: UART_v2_50)

# 8 Components

## 8.1 Component type: ADC_DelSig [v3.20]

### *8.1.1 Instance Mic_ADC*

**Description: Delta-Sigma ADC**
**Instance type: ADC_DelSig [v3.20]**
**Datasheet: online component datasheet for ADC_DelSig**

Table 13. Component Parameters for Mic_ADC

| Parameter Name | Value | Description |
|---|---|---|
| ADC_Alignment | Right | This parameter determines how the result is aligned in the 24 bit result word. |
| ADC_Alignment_Config2 | Right | This parameter determines how the result is aligned in the 24 bit result word. |
| ADC_Alignment_Config3 | Right | This parameter determines how the result is aligned in the 24 bit result word. |
| ADC_Alignment_Config4 | Right | This parameter determines how the result is aligned in the 24 bit result word. |
| ADC_Charge_Pump_Clock | true | Low power charge pump clock selection |
| ADC_Clock | Internal | Parameter for selecting the ADC clock type. |
| ADC_Input_Mode | Single | Differential or Single ended input mode |
| ADC_Input_Range | Vssa to Vdda | Choose input operating mode that best supports the range of the signals being measured. |
| ADC_Input_Range_Config2 | 0.0 to Vref | Choose input operating mode that best supports the range of the signals being measured. |
| ADC_Input_Range_Config3 | 0.0 to Vref | Choose input operating mode that best supports the range of the signals being measured. |
| ADC_Input_Range_Config4 | 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 Vdda/4 | 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 | 8 | 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 |

| Parameter Name | Value | Description |
|---|---|---|
| Clock_Frequency | 64000 | Determines the ADC clock frequency. |
| Comment_Config1 | Default Config | Parameter which holds the user comment for the config1. |
| Comment_Config2 | Second Config | Parameter which holds the user comment for the config2. |
| Comment_Config3 | Third Config | Parameter which holds the user comment for the config3. |
| Comment_Config4 | Fourth Config | Parameter which holds the user comment for the config4. |
| Config1_Name | CFG1 | This parameter is used to create constants in the header file for config 1. |
| Config2_Name | CFG2 | This parameter is used to create constants in the header file for config 2. |
| Config3_Name | CFG3 | This parameter is used to create constants in the header file for config 3. |
| Config4_Name | CFG4 | This parameter is used to create constants in the header file for config 4. |
| Configs | 1 | Number of active configurations |
| Conversion_Mode | 2 - Continuous | ADC conversion mode |
| Conversion_Mode_Config2 | 2 - Continuous | ADC conversion mode |
| Conversion_Mode_Config3 | 2 - Continuous | ADC conversion mode |
| Conversion_Mode_Config4 | 2 - Continuous | ADC conversion mode |
| Enable_Vref_Vss | false | Determines whether or not to connect ADC's reference Vssa to AGL[6]. |
| EnableModulatorInput | false | When this parameter is enabled, the modulator input terminal will be enabled on the symbol. |
| Input_Buffer_Gain | 1 | Gain of input amplifier |
| Input_Buffer_Gain_Config2 | 1 | Gain of input amplifier |
| Input_Buffer_Gain_Config3 | 1 | Gain of input amplifier |
| Input_Buffer_Gain_Config4 | 1 | Gain of input amplifier |
| Input_Buffer_Mode | 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 | 1.25 | Set reference voltage |
| Ref_Voltage_Config2 | 1.024 | Set reference voltage |
| Ref_Voltage_Config3 | 1.024 | Set reference voltage |
| Ref_Voltage_Config4 | 1.024 | Set reference voltage |
| rm_int | false | Removes internal interrupt (IRQ) |
| Sample_Rate | 11025 | 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 |
| Start_of_Conversion | Software | Continuous conversions or hardware controlled |

## 8.2 Component type: AMux [v1.80]

### *8.2.1 Instance Audio_Mux*

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

Table 14. Component Parameters for Audio_Mux

| Parameter Name | Value | Description |
|---|---|---|
| AtMostOneActive | false | Limit to at most one active channel. |
| Channels | 4 | 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: CharLCD [v2.20]

### *8.3.1 Instance LCD*

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

Table 15. Component Parameters for LCD

| Parameter Name | Value | Description |
|---|---|---|
| ConversionRoutines | true | Defines if the conversion routines will be included in the project. |
| CustomCharacterSet | None | Defines the type of custom character set (User defined, Vertical or Horizontal bargraph). Based on the selection a look-up table with proper characters representation will be generated in the source code. |

## 8.4 Component type: emFile [v1.20]

### *8.4.1 Instance emFile_1*

**Description: emFile file system for SD card in SPI mode**
**Instance type: emFile [v1.20]**
**Datasheet: online component datasheet for emFile**

Table 16. Component Parameters for emFile_1

| Parameter Name | Value | Description |
|---|---|---|
| Max_SPI_Frequency | 4000 | Maximum frequency (in kHz) of the SPI Master serial clock (sclk). See the SPI Master data sheet for details. |
| NumberSDCards | 1 | The number of SD cards in the system. The maximum is four (4). |

| Parameter Name | Value | Description |
|---|---|---|
| WP0_En | false | Enable write protect signal for SD card #1. If disabled the SD card isn't write protected. |
| WP1_En | false | Enable write protect signal for SD card #2. If disabled the SD card is not write protected. |
| WP2_En | false | Enable write protect signal for SD card #3. If disabled the SD card is not write protected. |
| WP3_En | false | Enable write protect signal for SD card #4. If disabled the SD card is not write protected. |

## 8.5 Component type: PWM [v3.30]

### *8.5.1 Instance PWM_1*

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

Table 17. Component Parameters for PWM_1

| 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 | 65535 | Compares Output 2 to value |
| DeadBand | Disabled | Defines whether dead band outputs are desired or not. |
| DeadTime | 1 | Defines the number of required dead band clock cycles |
| DitherOffset | 0.00 | Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM. |
| EnableMode | Software Only | Specifies the method of enabling the PWM. This can be either hardware or software. |
| FixedFunction | false | Determines whether the fixed function counter timer is used or the UDB implementation is used. |
| InterruptOnCMP1 | false | Enables the interrupt on compare1 true event |

| Parameter Name | Value | Description |
|---|---|---|
| InterruptOnCMP2 | false | Enables the interrupt on compare2 true event |
| InterruptOnKill | false | Enables the interrupt on a kill event |
| InterruptOnTC | false | Enables the interrupt on terminal count event |
| KillMode | Disabled | Parameter to select the kill mode for build time. |
| MinimumKillTime | 1 | Sets the minimum number of clock cycles that a kill must be active on the outputs when KillMode is set to Minimum Kill Time mode |
| Period | 255 | Defines the PWM period value |
| PWMMode | One Output | Defines the overall mode of the PWM |
| Resolution | 16 | Defines the bit width of the PWM (8 or 16 bits) |
| RunMode | Continuous | Defines the run mode options to be either continuous or one shot |
| TriggerMode | None | Determines the mode of starting the PWM, i.e. triggering the PWM counter to start |
| UseInterrupt | true | Enables the placement and usage of the status register |

## 8.6 Component type: UART [v2.50]

### *8.6.1 Instance UART*

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

Table 18. 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 | 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. |

| Parameter Name | Value | Description |
|---|---|---|
| HalfDuplexEn | false | Enables half duplex mode on the RX Half of the UART module. |
| 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. |

| Parameter Name | Value | Description |
|---|---|---|
| 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. |

## 8.7 Component type: VDAC8 [v1.90]

### *8.7.1 Instance Sample_Player*

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

Table 19. Component Parameters for Sample_Player

| 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. |
| VDAC_Range | 0 - 1.020V (4mV/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 | 400 | 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 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