PSoC® Creator™
Project Datasheet for CAN_Full_Extruder

Creation Time: 08/01/2017 15:33:14
User: JAMESWISE9DC6\jwise
Project: CAN_Full_Extruder
Tool: PSoC Creator 4.1

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

Copyright

Trademarks

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

Philips I2C Patent Rights

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

Disclaimer

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

Flash Code Protection

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

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

Table of Contents

1 Overview

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

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

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

Figure 1. CY8C58LP Device Series Block Diagram

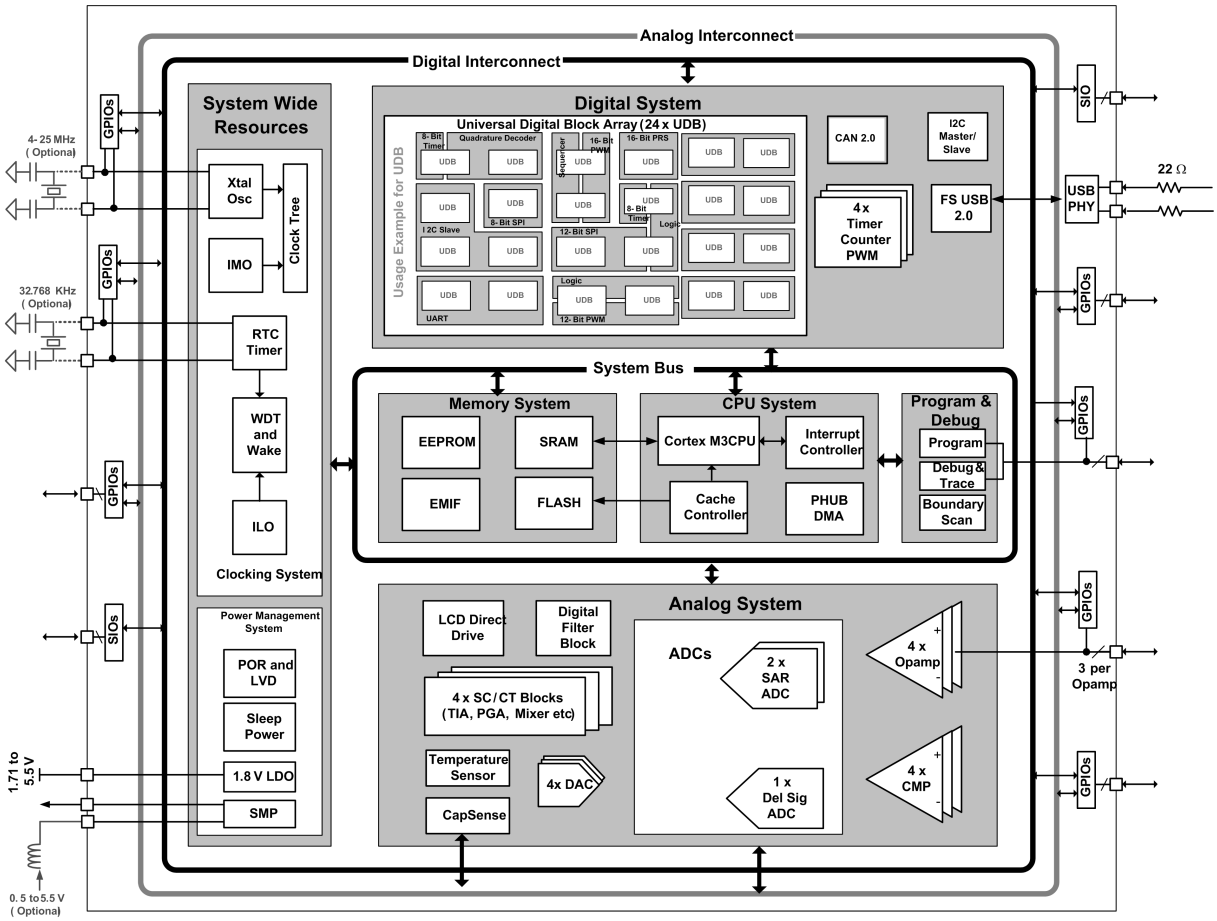


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

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

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

Table 2 lists the device resources that this design uses:

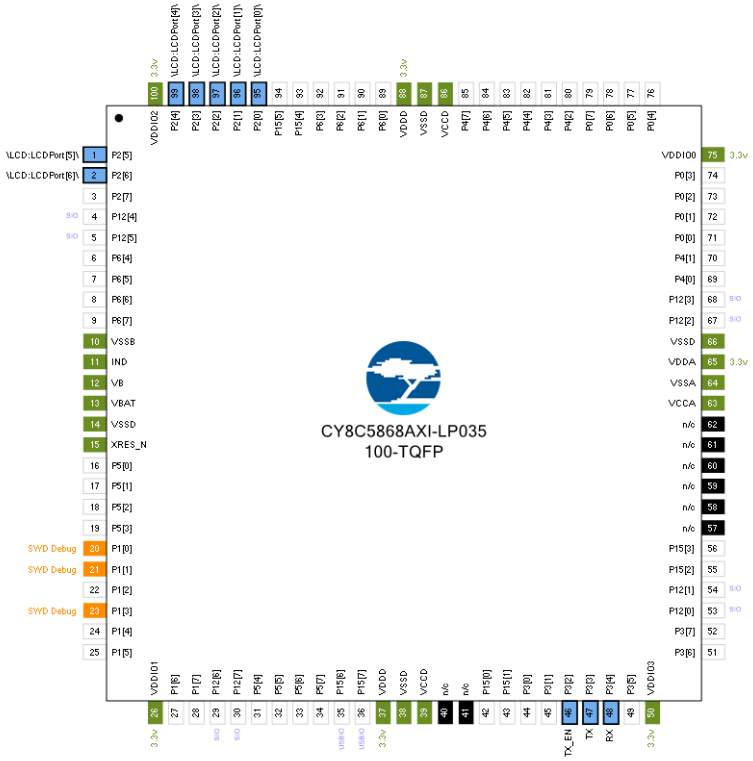
Table 2. Device Resources

Resource Type	Used	Free	Max	% Used
Digital Clocks	0	8	8	0.00 %
Analog Clocks	0	4	4	0.00 %
CapSense Buffers	0	2	2	0.00 %
Digital Filter Block	0	1	1	0.00 %
Interrupts	1	31	32	3.13 %
IO	13	59	72	18.06 %
Segment LCD	0	1	1	0.00 %
CAN 2.0b	1	0	1	100.00 %
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	0	192	192	0.00 %
Unique P-terms	0	384	384	0.00 %
Total P-terms	0			
Datapath Cells	0	24	24	0.00 %
Status Cells	0	24	24	0.00 %
Control Cells	0	24	24	0.00 %
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				
VIDAC	0	4	4	0.00 %

2 Pins

Figure 2 shows the pin layout of this device.

Figure 2. Device Pin Layout



2.1 Hardware Pins

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

Table 3. Device Pins

Pin	Port	Name	Type	Drive Mode	Reset State
1	P2[5]	\LCD:LCDPort[5]\	Software In/Out	Strong drive	HiZ Analog Unb
2	P2[6]	\LCD:LCDPort[6]\	Software In/Out	Strong drive	HiZ Analog Unb
3	P2[7]	GPIO [unused]			HiZ Analog Unb
4	P12[4]	SIO [unused]			HiZ Analog Unb
5	P12[5]	SIO [unused]			HiZ Analog Unb
6	P6[4]	GPIO [unused]			HiZ Analog Unb
7	P6[5]	GPIO [unused]			HiZ Analog Unb
8	P6[6]	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]	GPIO [unused]			HiZ Analog Unb
17	P5[1]	GPIO [unused]			HiZ Analog Unb
18	P5[2]	GPIO [unused]			HiZ Analog Unb
19	P5[3]	GPIO [unused]			HiZ Analog Unb
20	P1[0]	Debug:SWD_IO	Reserved		
21	P1[1]	Debug:SWD_CK	Reserved		
22	P1[2]	GPIO [unused]			HiZ Analog Unb
23	P1[3]	Debug:SWV	Reserved		
24	P1[4]	GPIO [unused]			HiZ Analog Unb
25	P1[5]	GPIO [unused]			HiZ Analog Unb
26	VDDIO1	VDDIO1	Power		
27	P1[6]	GPIO [unused]			HiZ Analog Unb
28	P1[7]	GPIO [unused]			HiZ Analog Unb
29	P12[6]	SIO [unused]			HiZ Analog Unb
30	P12[7]	SIO [unused]			HiZ Analog Unb
31	P5[4]	GPIO [unused]			HiZ Analog Unb
32	P5[5]	GPIO [unused]			HiZ Analog Unb
33	P5[6]	GPIO [unused]			HiZ Analog Unb
34	P5[7]	GPIO [unused]			HiZ Analog Unb
35	P15[6]	USB IO [unused]			HiZ Analog Unb
36	P15[7]	USB IO [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]	TX_EN	Dgtl Out	Strong drive	HiZ Analog Unb

Pin	Port	Name	Type	Drive Mode	Reset State
47	P3[3]	TX	Dgtl Out	Strong drive	HiZ Analog Unb
48	P3[4]	RX	Dgtl In	HiZ digital	HiZ Analog Unb
49	P3[5]	GPIO [unused]			HiZ Analog Unb
50	VDDIO3	VDDIO3	Power		
51	P3[6]	GPIO [unused]			HiZ Analog Unb
52	P3[7]	GPIO [unused]			HiZ Analog Unb
53	P12[0]	SIO [unused]			HiZ Analog Unb
54	P12[1]	SIO [unused]			HiZ Analog Unb
55	P15[2]	GPIO [unused]			HiZ Analog Unb
56	P15[3]	GPIO [unused]			HiZ Analog Unb
63	VCCA	VCCA	Power		
64	VSSA	VSSA	Power		
65	VDDA	VDDA	Power		
66	VSSD	VSSD	Power		
67	P12[2]	SIO [unused]			HiZ Analog Unb
68	P12[3]	SIO [unused]			HiZ Analog Unb
69	P4[0]	GPIO [unused]			HiZ Analog Unb
70	P4[1]	GPIO [unused]			HiZ Analog Unb
71	P0[0]	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	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]	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]	GPIO [unused]			HiZ Analog Unb
90	P6[1]	GPIO [unused]			HiZ Analog Unb
91	P6[2]	GPIO [unused]			HiZ Analog Unb
92	P6[3]	GPIO [unused]			HiZ Analog Unb
93	P15[4]	GPIO [unused]			HiZ Analog Unb
94	P15[5]	GPIO [unused]			HiZ Analog Unb
95	P2[0]	\LCD:LCDPort[0]\	Software In/Out	Strong drive	HiZ Analog Unb
96	P2[1]	\LCD:LCDPort[1]\	Software In/Out	Strong drive	HiZ Analog Unb
97	P2[2]	\LCD:LCDPort[2]\	Software In/Out	Strong drive	HiZ Analog Unb
98	P2[3]	\LCD:LCDPort[3]\	Software In/Out	Strong drive	HiZ Analog Unb
99	P2[4]	\LCD:LCDPort[4]\	Software In/Out	Strong drive	HiZ Analog Unb
100	VDDIO2	VDDIO2	Power		

Abbreviations used in Table 3 have the following meanings:

- HiZ Analog Unb = Hi-Z Analog Unbuffered
- 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]	71	GPIO [unused]			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	GPIO [unused]			HiZ Analog Unb
P1[7]	28	GPIO [unused]			HiZ Analog Unb
P12[0]	53	SIO [unused]			HiZ Analog Unb
P12[1]	54	SIO [unused]			HiZ Analog Unb
P12[2]	67	SIO [unused]			HiZ Analog Unb
P12[3]	68	SIO [unused]			HiZ Analog Unb
P12[4]	4	SIO [unused]			HiZ Analog Unb
P12[5]	5	SIO [unused]			HiZ Analog Unb
P12[6]	29	SIO [unused]			HiZ Analog Unb
P12[7]	30	SIO [unused]			HiZ Analog Unb
P15[0]	42	GPIO [unused]			HiZ Analog Unb
P15[1]	43	GPIO [unused]			HiZ Analog Unb
P15[2]	55	GPIO [unused]			HiZ Analog Unb
P15[3]	56	GPIO [unused]			HiZ Analog Unb
P15[4]	93	GPIO [unused]			HiZ Analog Unb
P15[5]	94	GPIO [unused]			HiZ Analog Unb
P15[6]	35	USB IO [unused]			HiZ Analog Unb
P15[7]	36	USB IO [unused]			HiZ Analog Unb
P2[0]	95	\LCD:LCDPort[0]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[1]	96	\LCD:LCDPort[1]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[2]	97	\LCD:LCDPort[2]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[3]	98	\LCD:LCDPort[3]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[4]	99	\LCD:LCDPort[4]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[5]	1	\LCD:LCDPort[5]\	Software In/Out	Strong drive	HiZ Analog Unb
P2[6]	2	\LCD:LCDPort[6]\	Software In/Out	Strong drive	HiZ Analog Unb

Port	Pin	Name	Type	Drive Mode	Reset State
P2[7]	3	GPIO [unused]			HiZ Analog Unb
P3[0]	44	GPIO [unused]			HiZ Analog Unb
P3[1]	45	GPIO [unused]			HiZ Analog Unb
P3[2]	46	TX_EN	Dgtl Out	Strong drive	HiZ Analog Unb
P3[3]	47	TX	Dgtl Out	Strong drive	HiZ Analog Unb
P3[4]	48	RX	Dgtl In	HiZ digital	HiZ Analog Unb
P3[5]	49	GPIO [unused]			HiZ Analog Unb
P3[6]	51	GPIO [unused]			HiZ Analog Unb
P3[7]	52	GPIO [unused]			HiZ Analog Unb
P4[0]	69	GPIO [unused]			HiZ Analog Unb
P4[1]	70	GPIO [unused]			HiZ Analog Unb
P4[2]	80	GPIO [unused]			HiZ Analog Unb
P4[3]	81	GPIO [unused]			HiZ Analog Unb
P4[4]	82	GPIO [unused]			HiZ Analog Unb
P4[5]	83	GPIO [unused]			HiZ Analog Unb
P4[6]	84	GPIO [unused]			HiZ Analog Unb
P4[7]	85	GPIO [unused]			HiZ Analog Unb
P5[0]	16	GPIO [unused]			HiZ Analog Unb
P5[1]	17	GPIO [unused]			HiZ Analog Unb
P5[2]	18	GPIO [unused]			HiZ Analog Unb
P5[3]	19	GPIO [unused]			HiZ Analog Unb
P5[4]	31	GPIO [unused]			HiZ Analog Unb
P5[5]	32	GPIO [unused]			HiZ Analog Unb
P5[6]	33	GPIO [unused]			HiZ Analog Unb
P5[7]	34	GPIO [unused]			HiZ Analog Unb
P6[0]	89	GPIO [unused]			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 Unb = Hi-Z Analog Unbuffered
- Dgtl Out = Digital Output
- Dgtl In = Digital Input
- HiZ digital = High impedance digital

2.3 Software Pins

Table 5 contains information about the software pins on this device in alphabetical order. (Only software-accessible pins are shown.)

Table 5. Software Pins

Name	Port	Type	Reset State
\LCD:LCDPort[0]\	P2[0]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[1]\	P2[1]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[2]\	P2[2]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[3]\	P2[3]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[4]\	P2[4]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[5]\	P2[5]	Software In/Out	HiZ Analog Unb
\LCD:LCDPort[6]\	P2[6]	Software In/Out	HiZ Analog Unb
Debug:SWD_CK	P1[1]	Reserved	
Debug:SWD_IO	P1[0]	Reserved	
Debug:SWV	P1[3]	Reserved	
GPIO [unused]	P15[2]		HiZ Analog Unb
GPIO [unused]	P15[3]		HiZ Analog Unb
GPIO [unused]	P0[3]		HiZ Analog Unb
GPIO [unused]	P4[6]		HiZ Analog Unb
GPIO [unused]	P3[7]		HiZ Analog Unb
GPIO [unused]	P3[6]		HiZ Analog Unb
GPIO [unused]	P3[5]		HiZ Analog Unb
GPIO [unused]	P0[4]		HiZ Analog Unb
GPIO [unused]	P6[0]		HiZ Analog Unb
GPIO [unused]	P4[7]		HiZ Analog Unb
GPIO [unused]	P0[5]		HiZ Analog Unb
GPIO [unused]	P4[2]		HiZ Analog Unb
GPIO [unused]	P4[1]		HiZ Analog Unb
GPIO [unused]	P4[0]		HiZ Analog Unb
GPIO [unused]	P0[7]		HiZ Analog Unb
GPIO [unused]	P0[6]		HiZ Analog Unb
GPIO [unused]	P4[3]		HiZ Analog Unb
GPIO [unused]	P0[1]		HiZ Analog Unb
GPIO [unused]	P0[2]		HiZ Analog Unb
GPIO [unused]	P0[0]		HiZ Analog Unb
GPIO [unused]	P4[4]		HiZ Analog Unb
GPIO [unused]	P4[5]		HiZ Analog Unb
GPIO [unused]	P1[4]		HiZ Analog Unb
GPIO [unused]	P1[2]		HiZ Analog Unb
GPIO [unused]	P5[3]		HiZ Analog Unb
GPIO [unused]	P1[5]		HiZ Analog Unb
GPIO [unused]	P5[4]		HiZ Analog Unb
GPIO [unused]	P1[7]		HiZ Analog Unb
GPIO [unused]	P1[6]		HiZ Analog Unb

Name	Port	Type	Reset State
GPIO [unused]	P5[2]		HiZ Analog Unb
GPIO [unused]	P6[5]		HiZ Analog Unb
GPIO [unused]	P6[4]		HiZ Analog Unb
GPIO [unused]	P2[7]		HiZ Analog Unb
GPIO [unused]	P6[6]		HiZ Analog Unb
GPIO [unused]	P5[1]		HiZ Analog Unb
GPIO [unused]	P5[0]		HiZ Analog Unb
GPIO [unused]	P6[7]		HiZ Analog Unb
GPIO [unused]	P15[0]		HiZ Analog Unb
GPIO [unused]	P6[1]		HiZ Analog Unb
GPIO [unused]	P6[2]		HiZ Analog Unb
GPIO [unused]	P3[1]		HiZ Analog Unb
GPIO [unused]	P3[0]		HiZ Analog Unb
GPIO [unused]	P15[1]		HiZ Analog Unb
GPIO [unused]	P5[7]		HiZ Analog Unb
GPIO [unused]	P5[5]		HiZ Analog Unb
GPIO [unused]	P5[6]		HiZ Analog Unb
GPIO [unused]	P6[3]		HiZ Analog Unb
GPIO [unused]	P15[4]		HiZ Analog Unb
GPIO [unused]	P15[5]		HiZ Analog Unb
RX	P3[4]	Dgtl In	HiZ Analog Unb
SIO [unused]	P12[6]		HiZ Analog Unb
SIO [unused]	P12[5]		HiZ Analog Unb
SIO [unused]	P12[7]		HiZ Analog Unb
SIO [unused]	P12[4]		HiZ Analog Unb
SIO [unused]	P12[1]		HiZ Analog Unb
SIO [unused]	P12[2]		HiZ Analog Unb
SIO [unused]	P12[0]		HiZ Analog Unb
SIO [unused]	P12[3]		HiZ Analog Unb
TX	P3[3]	Dgtl Out	HiZ Analog Unb
TX_EN	P3[2]	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:

- 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)	0x80
Stack Size (bytes)	0x0800
Include CMSIS Core Peripheral Library Files	True

3.2 System Debug Settings

Table 7. System Debug Settings

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

3.3 System Operating Conditions

Table 8. System Operating Conditions

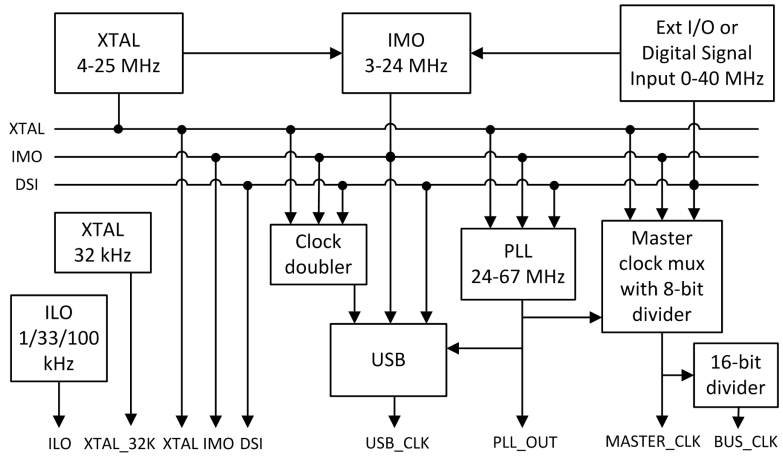
Name	Value
VDDA (V)	3.3
VDDD (V)	3.3
VDDIO0 (V)	3.3
VDDIO1 (V)	3.3
VDDIO2 (V)	3.3
VDDIO3 (V)	3.3
Variable VDDA	False
Temperature Range	-40C - 85/125C

4 Clocks

The clock system includes these clock resources:

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

Figure 3. System Clock Configuration



4.1 System Clocks

Table 9 lists the system clocks used in this design.

Table 9. System Clocks

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

4.2 Local and Design Wide Clocks

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

Figure 4. Local and Design Wide Clock Configuration

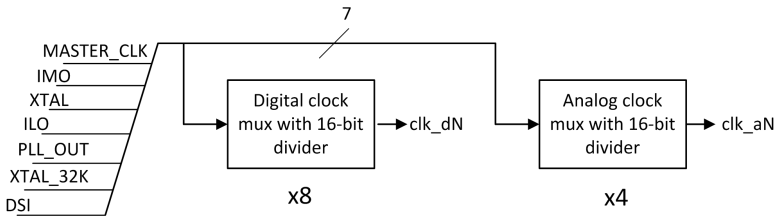


Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

Name	Domain	Source	Desired Freq	Nominal Freq	Accuracy (%)	Start at Reset	Enabled
CAN_Clock	DIGITAL	BUS_CLK	? MHz	24 MHz	±1	True	True

For more information on clocking resources, please refer to:

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

5 Interrupts and DMAs

5.1 Interrupts

This design contains the following interrupt components: (0 is the highest priority)

Table 11. Interrupts

Name	Intr Num	Vector	Priority
CAN_isr	16	16	7

For more information on interrupts, please refer to:

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

5.2 DMAs

This design contains no DMA components.

6 Flash Memory

PSoC 5LP devices offer a host of Flash protection options and device security features that you can leverage to meet the security and protection requirements of an application. These requirements range from protecting configuration settings or Flash data to locking the entire device from external access.

Table 12 lists the Flash protection settings for your design.

Table 12. Flash Protection Settings

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

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

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

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

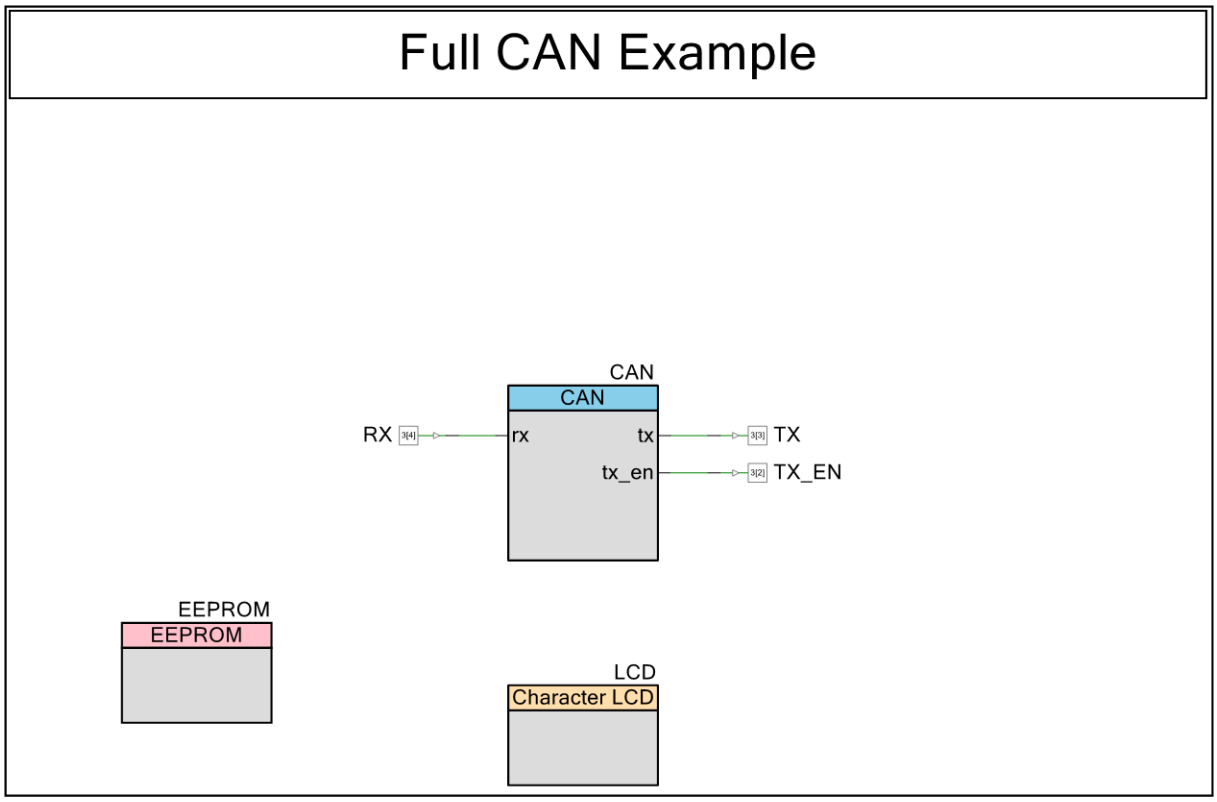
- Flash Protection chapter in the PSoC 5LP Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyWrite API routines
  - CyFlash API routines

7 Design Contents

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

7.1 Schematic Sheet: Page 1

Figure 5. Schematic Sheet: Page 1



This schematic sheet contains the following component instances:

- Instance CAN (type: CAN_v3_0)
- Instance EEPROM (type: EEPROM_v3_0)
- Instance LCD (type: CharLCD_v2_20)

8 Components

8.1 Component type: CAN [v3.0]

8.1.1 Instance CAN

Description: Controller Area Network (ISO-11898-1)
Instance type: CAN [v3.0]
Datasheet: online component datasheet for CAN

Table 13. Component Parameters for CAN

Parameter Name	Value	Description
AckError	false	Message acknowledge error detected interrupt
AckErrorUseHelper	true	Use ISR helper for Message acknowledge error detected interrupt
AdvancedInterruptConfig	true	Advanced interrupt configuration
Arbiter	0	Transmit buffer arbitration
ArbLost	false	Arbitration lost detected interrupt
ArbLostUseHelper	true	Use ISR helper for Arbitration lost detected interrupt
BaudRate	1000	The desired baud rate
BitError	false	Bit error detected interrupt
BitErrorUseHelper	true	Use ISR helper for Bit error detected interrupt
Bitrate	2	Bit rate prescaler (BRP)
BussOff	true	Bus off state interrupt
BussOffUseHelper	true	Use ISR helper for Bus off state interrupt
ClkFrequency	24	The system clock frequency equal to BUS_CLK (PSoC 3/5LP) or SYSCLK (PSoC 4)
ConnectExtInterruptLine	false	Use External interrupt line as the third output
ConnectTxEn	true	Use external transceiver enable signal as the second output
CrcError	false	CRC error detected interrupt
CrcErrorUseHelper	true	Use ISR Helper for CRC error detected interrupt
EdgeMode	0	CAN bus synchronization logic
FormError	false	Message format error detected interrupt
FormErrorUseHelper	true	Use ISR helper for Message format error detected interrupt
FullCustomIntISR	true	Enable the internal ISR with fully custom code
IntEnable	true	Global Interrupt enable flag
IntISRDisable	false	Disable/bypass the internal ISR component
Overload	false	Overload frame received interrupt

Parameter Name	Value	Description
OverloadUseHelper	true	Use ISR helper for Overload frame received interrupt
Reset	1	Bus-off restart
RTRAutomaticReply	false	RTR automatic reply sent interrupt
RTRAutomaticReplyUseHelper	true	Use ISR helper for RTR automatic reply sent interrupt
RxMsg	true	Message received interrupt
RxMsgLost	true	Receive buffer full interrupt
RxMsgLostUseHelper	true	Use ISR helper for Receive buffer full interrupt
RxMsgUseHelper	true	Use ISR helper for Message received interrupt
Sjw	3	Synchronization Jump Width
Sm	0	CAN bus Bit sampling
SSTError	false	Single shot transmission failure interrupt
SSTErrorUseHelper	true	Use ISR helper for Single shot transmission failure interrupt
StuckAtZero	false	Stuck at zero interrupt
StuckAtZeroUseHelper	true	Use ISR helper for Stuck at zero interrupt
StuffError	false	Bit stuffing error detected interrupt
StuffErrorUseHelper	true	Use ISR helper for Bit stuffing error detected interrupt
SwapDataByteEndianness	false	The byte position of the CAN receive and transmit data field endianness
Tseg1	4	The length of time segment1
Tseg2	3	The length of time segment2
TxMsg	true	Message transmitted interrupt
TxMsgUseHelper	true	Use ISR helper for Message transmitted interrupt
User Comments		Instance-specific comments.

8.2 Component type: CharLCD [v2.20]

8.2.1 Instance LCD

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

Table 14. 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.

Parameter Name	Value	Description
User Comments		Instance-specific comments.

8.3 Component type: EEPROM [v3.0]

8.3.1 Instance EEPROM

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

Table 15. Component Parameters for EEPROM

Parameter Name	Value	Description
User Comments		Instance-specific comments.

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