# PSoC® Creator™
# Project Datasheet for Rugemaskine

**Creation Time: 05/26/2014 21:38:40**
**User: WIN-C8TFSBJA6FM\Simon Mouridsen**
**Project: Rugemaskine**

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

## 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 3 is a family of 8-bit devices with the following characteristics:

- An 8-bit single cycle pipelined 8051 processor, running up to 67 MHz, with a nested vectored interrupt controller (NVIC) and a high-performance DMA controller. The single cycle 8051 CPU runs ten times faster than a standard 8051 processor.
- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals, such as USB, CAN and I2C
- Analog subsystem that includes configurable switched (SC) and continuous time (CT) blocks, up to 20-bit Delta Sigma converters (ADC), SAR ADCs, 8-bit DACs that can be configured for 12-bit operation, op amps, comparators, PGAs, 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 CY8C38 family member PSoC 3 device. For details on all the systems listed above, please refer to the PSoC 3 Technical Reference Manual .

Figure 1. CY8C38 Device Family Block Diagram

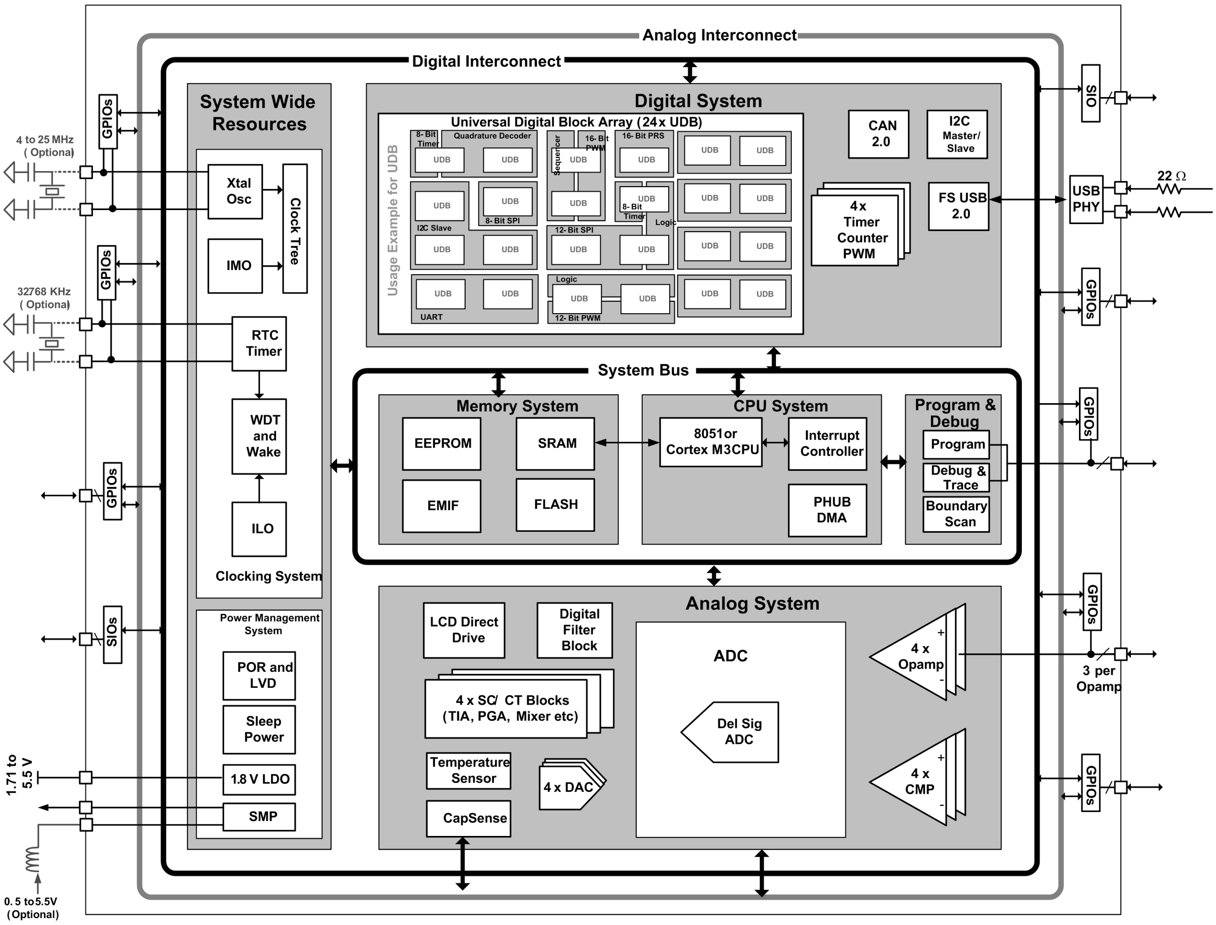


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

| Name | Value |
|---|---|
| Architecture | PSoC 3 |
| Family | CY8C38 |
| CPU speed (MHz) | 67 |
| Flash size (kBytes) | 64 |
| SRAM size (kBytes) | 8 |
| EEPROM size (Bytes) | 2048 |
| Trace Buffer (kBytes) | 4 |
| Vdd range (V) | 1.7 to 5.5 |
| Automotive qualified | No (Industrial Grade Only) |
| Temp range (Celcius) | -40 to 85 |
| JTAG ID | 0x1E028069 |

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:

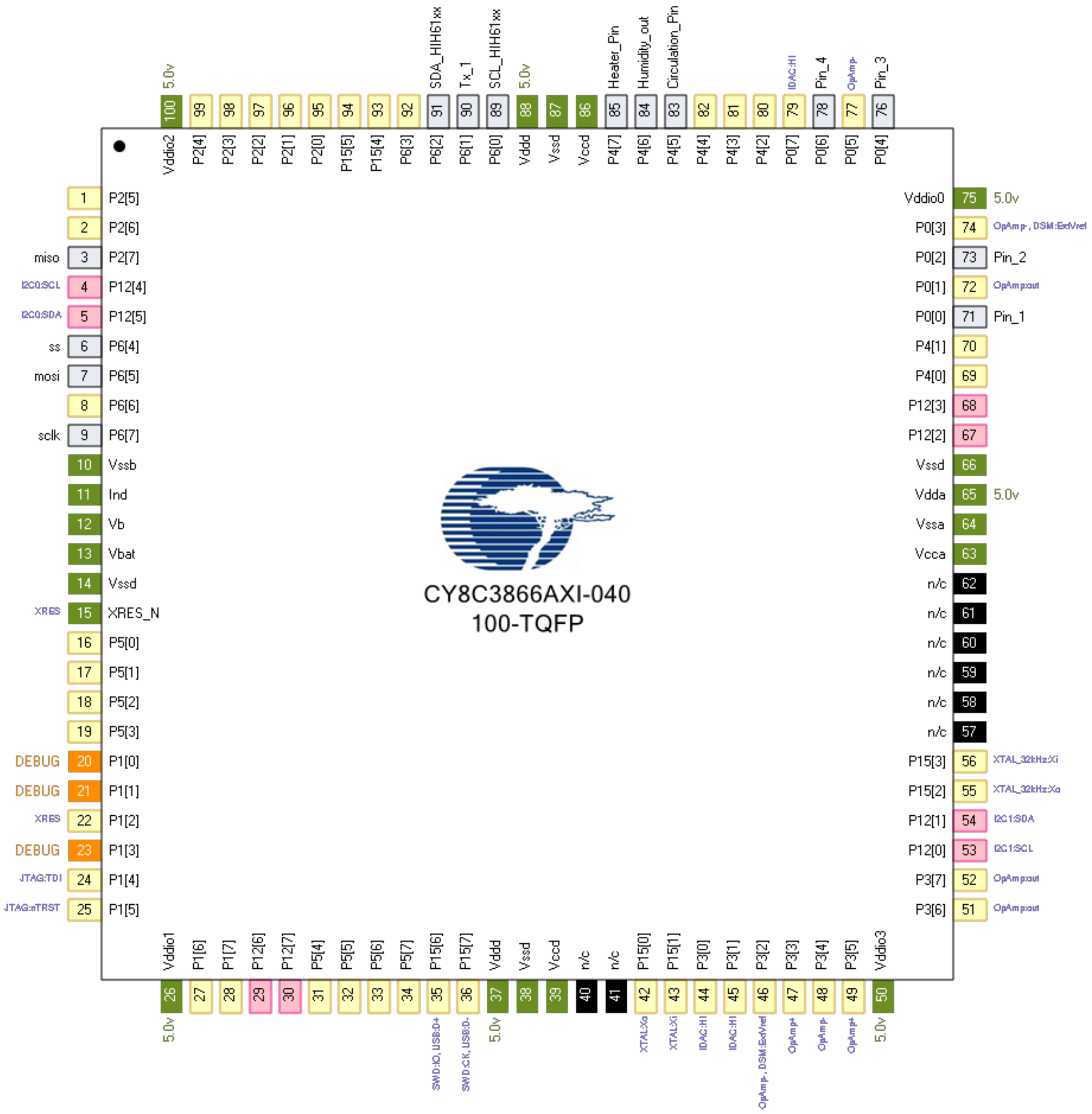
Table 2. Device Resources

| Name | Resources in Use | Total Resources Available |
|---|---|---|
| Digital clock dividers | 5 (62.5%) | 8 |
| Analog clock dividers | 0 (0.0%) | 4 |
| Pins | 17 (23.6%) | 72 |
| UDB Macrocells | 36 (18.8%) | 192 |
| UDB Unique Pterms | 49 (12.8%) | 384 |
| UDB Datapath Cells | 10 (41.7%) | 24 |
| UDB Status Cells | 7 (29.2%) | 24 |
| UDB Control Cells | 5 (20.8%) | 24 |
| DMA Channels | 0 (0.0%) | 24 |
| Interrupts | 6 (18.8%) | 32 |
| DSM Fixed Blocks | 0 (0.0%) | 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 | 0 (0.0%) | 1 |
| I2C Fixed Blocks | 1 (100.0%) | 1 |
| Timer Fixed Blocks | 2 (50.0%) | 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 |

# 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] | GPIO [unused] | | | HiZ Analog Unb |
| 2 | P2[6] | GPIO [unused] | | | HiZ Analog Unb |
| 3 | P2[7] | miso | Dgtl Out | Strong drive | HiZ Analog Unb |
| 4 | P12[4] | SIO [unused] | | | HiZ Analog Unb |
| 5 | P12[5] | SIO [unused] | | | HiZ Analog Unb |
| 6 | P6[4] | ss | Dgtl In | HiZ digital | HiZ Analog Unb |
| 7 | P6[5] | mosi | Dgtl In | HiZ digital | HiZ Analog Unb |
| 8 | P6[6] | GPIO [unused] | | | HiZ Analog Unb |
| 9 | P6[7] | sclk | Dgtl In | HiZ digital | HiZ Analog Unb |
| 10 | Vssb | Vssb | Power | | |
| 11 | Ind | Power | | | |
| 12 | Vb | Vb | Power | | |
| 13 | Vbat | Vbat | Power | | |
| 14 | Vssd | Vssd | Power | | |
| 15 | XRES_N | 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] | GPIO [unused] | | | HiZ Analog Unb |
| 23 | P1[3] | GPIO [unused] | | | HiZ Analog Unb |
| 24 | P1[4] | GPIO [unused] | | | HiZ Analog Unb |
| 25 | P1[5] | GPIO [unused] | | | 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] | 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 [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 |

| Pin | Port | Name | Type | Drive Mode | Reset State |
|---|---|---|---|---|---|
| 48 | P3[4] | GPIO [unused] | | | HiZ Analog Unb |
| 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] | 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] | Pin_1 | Dgtl Out | Strong drive | HiZ Analog Unb |
| 72 | P0[1] | GPIO [unused] | | | HiZ Analog Unb |
| 73 | P0[2] | Pin_2 | Dgtl Out | Strong drive | HiZ Analog Unb |
| 74 | P0[3] | GPIO [unused] | | | HiZ Analog Unb |
| 75 | Vio0 | Vio0 | Power | | |
| 76 | P0[4] | Pin_3 | Dgtl Out | Strong drive | HiZ Analog Unb |
| 77 | P0[5] | GPIO [unused] | | | HiZ Analog Unb |
| 78 | P0[6] | Pin_4 | Dgtl Out | Strong drive | 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] | Circulation_Pin | Dgtl Out | Strong drive | HiZ Analog Unb |
| 84 | P4[6] | Humidity_out | | Strong drive | HiZ Analog Unb |
| 85 | P4[7] | Heater_Pin | Dgtl Out | Strong drive | HiZ Analog Unb |
| 86 | Vccd | Vccd | Power | | |
| 87 | Vssd | Vssd | Power | | |
| 88 | Vddd | Vddd | Power | | |
| 89 | P6[0] | SCL_HIH61xx | Dgtl I/O | OD, DL | HiZ Analog Unb |
| 90 | P6[1] | Tx_1 | Dgtl Out | Strong drive | HiZ Analog Unb |
| 91 | P6[2] | SDA_HIH61xx | Dgtl I/O | OD, DL | 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] | GPIO [unused] | | | HiZ Analog Unb |
| 96 | P2[1] | GPIO [unused] | | | HiZ Analog Unb |
| 97 | P2[2] | GPIO [unused] | | | HiZ Analog Unb |
| 98 | P2[3] | GPIO [unused] | | | HiZ Analog Unb |
| 99 | P2[4] | GPIO [unused] | | | HiZ Analog Unb |
| 100 | Vio2 | Vio2 | 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
- Dgtl I/O = Digital In/Out

- OD, DL = Open drain, drives low

## 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 |
|---|---|---|---|
| Circulation_Pin | P4[5] | Dgtl Out | HiZ Analog Unb |
| Heater_Pin | P4[7] | Dgtl Out | HiZ Analog Unb |
| Humidity_out | P4[6] | | HiZ Analog Unb |
| miso | P2[7] | Dgtl Out | HiZ Analog Unb |
| mosi | P6[5] | Dgtl In | HiZ Analog Unb |
| Pin_1 | P0[0] | Dgtl Out | HiZ Analog Unb |
| Pin_2 | P0[2] | Dgtl Out | HiZ Analog Unb |
| Pin_3 | P0[4] | Dgtl Out | HiZ Analog Unb |
| Pin_4 | P0[6] | Dgtl Out | HiZ Analog Unb |
| Power | Ind | | |
| SCL_HIH61xx | P6[0] | Dgtl I/O | HiZ Analog Unb |
| sclk | P6[7] | Dgtl In | HiZ Analog Unb |
| SDA_HIH61xx | P6[2] | Dgtl I/O | HiZ Analog Unb |
| ss | P6[4] | Dgtl In | HiZ Analog Unb |
| Tx_1 | P6[1] | Dgtl Out | HiZ Analog Unb |

Abbreviations used in Table 4 have the following meanings:

- Dgtl Out = Digital Output
- HiZ Analog Unb = Hi-Z Analog Unbuffered
- Dgtl In = Digital Input
- Dgtl I/O = Digital In/Out

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 | Compressed |
| Enable Error Correcting Code (ECC) | False |
| Store Configuration Data in ECC Memory | True |
| Instruction Cache Enabled | True |
| Enable Fast IMO During Startup | True |
| Clear SRAM During Startup | True |
| Unused Bonded IO | Disallowed |

## 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 |
| Use Optional XRES | False |

## 3.3 System Operating Conditions

Table 7. System Operating Conditions

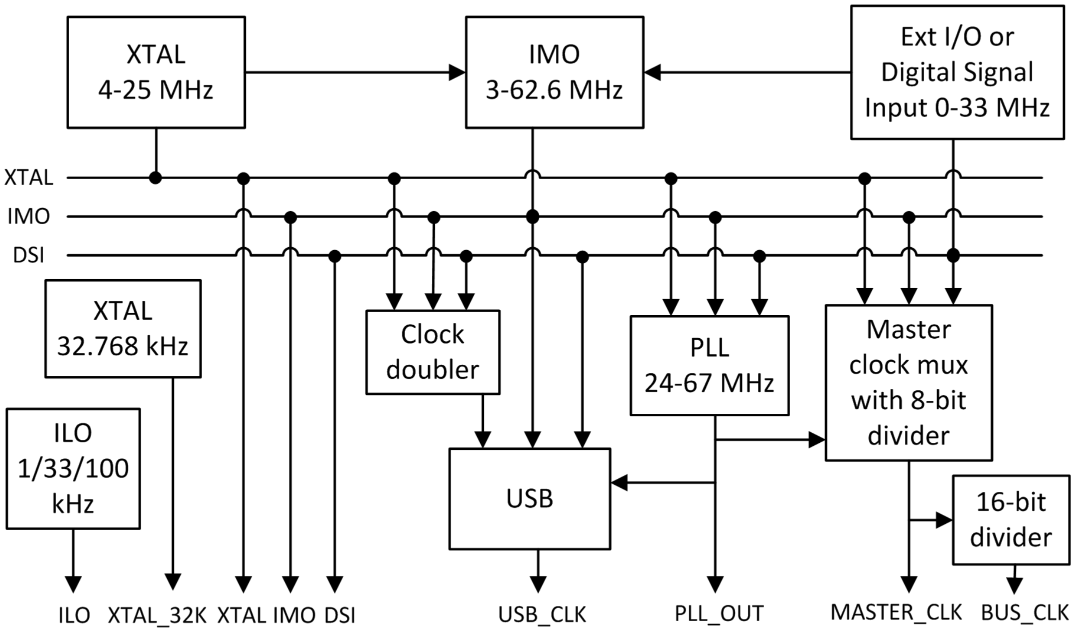
| Name | Value |
|---|---|
| Vddd (V) | 5.0 |
| Vdda (V) | 5.0 |
| Variable Vdda | False |
| Vddio0 (V) | 5.0 |
| Vddio1 (V) | 5.0 |
| Vddio2 (V) | 5.0 |
| Vddio3 (V) | 5.0 |
| Temperature Range | -40C - 85/125C |

# 4 Clocks

The clock system includes these clock resources:

- Four internal clock sources increase system integration:
  - 3 to 62.6 MHz Internal Main Oscillator (IMO) ±1% at 3 MHz
  - 1 kHz, 33 kHz, 100 kHz Internal Low Speed Oscillator (ILO) outputs
  - 12 to 67 MHz clock doubler output, 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 | 0 | 24 | ±1 | True | True |
| MASTER_CLK | DIGITAL | PLL_OUT | 0 | 24 | ±1 | True | True |
| Digital Signal | DIGITAL |  | 0 | 0 | ±0 | False | False |
| XTAL 32kHz | DIGITAL |  | 0.0328 | 0 | ±0 | False | False |
| XTAL | DIGITAL |  | 24 | 0 | ±0 | False | False |
| ILO | DIGITAL |  | 0 | 0.001 | -50,+100 | True | True |
| PLL_OUT | DIGITAL | IMO | 24 | 24 | ±1 | True | True |
| IMO | DIGITAL |  | 3 | 3 | ±1 | 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

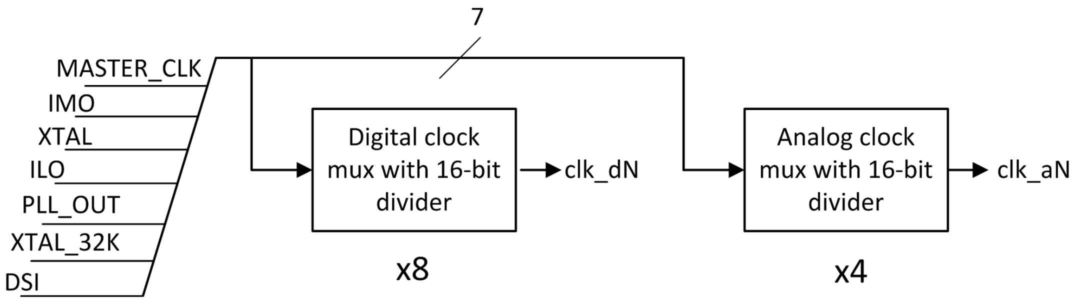

7
MASTER_CLK
IMO
XTAL
ILO
PLL_OUT
XTAL_32K
DSI
Digital clock mux with 16-bit divider
clk_dN
x8
Analog clock mux with 16-bit divider
clk_aN
x4

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 |
|---|---|---|---|---|---|---|---|
| Clock_1 | DIGITAL | MASTER_CLK | 12 | 12 | ±1 | True | True |
| clock_2 | DIGITAL | IMO | 0.01 | 0.01 | ±1 | True | True |
| UART_1_IntClock | DIGITAL | MASTER_CLK | 0.9216 | 0.9231 | ±1 | True | True |
| PWM_Clock | DIGITAL | IMO | 0.5 | 0.5 | ±1 | True | True |
| Timer_clock | DIGITAL | IMO | 0.0001 | 0.0001 | ±1 | True | True |
| I2C_HIH61xx_BusClock | DIGITAL | BUS_CLK | 0 | 24 | ±1 | True | True |
| timer_clock_1 | DIGITAL | BUS_CLK | 0 | 24 | ±1 | True | True |

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 3 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 |
|---|---|---|
| Humidity_ISR | 7 | 0 |
| I2C_HIH61xx_I2C_IRQ | 7 | 15 |
| MotorISR | 7 | 18 |
| PID_ISR | 7 | 1 |
| rx_isr | 7 | 3 |
| TimerISR | 7 | 2 |

For more information on interrupts, please refer to:

- Interrupt Controller chapter in the PSoC 3 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 3 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 | 0xFFFF | 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:

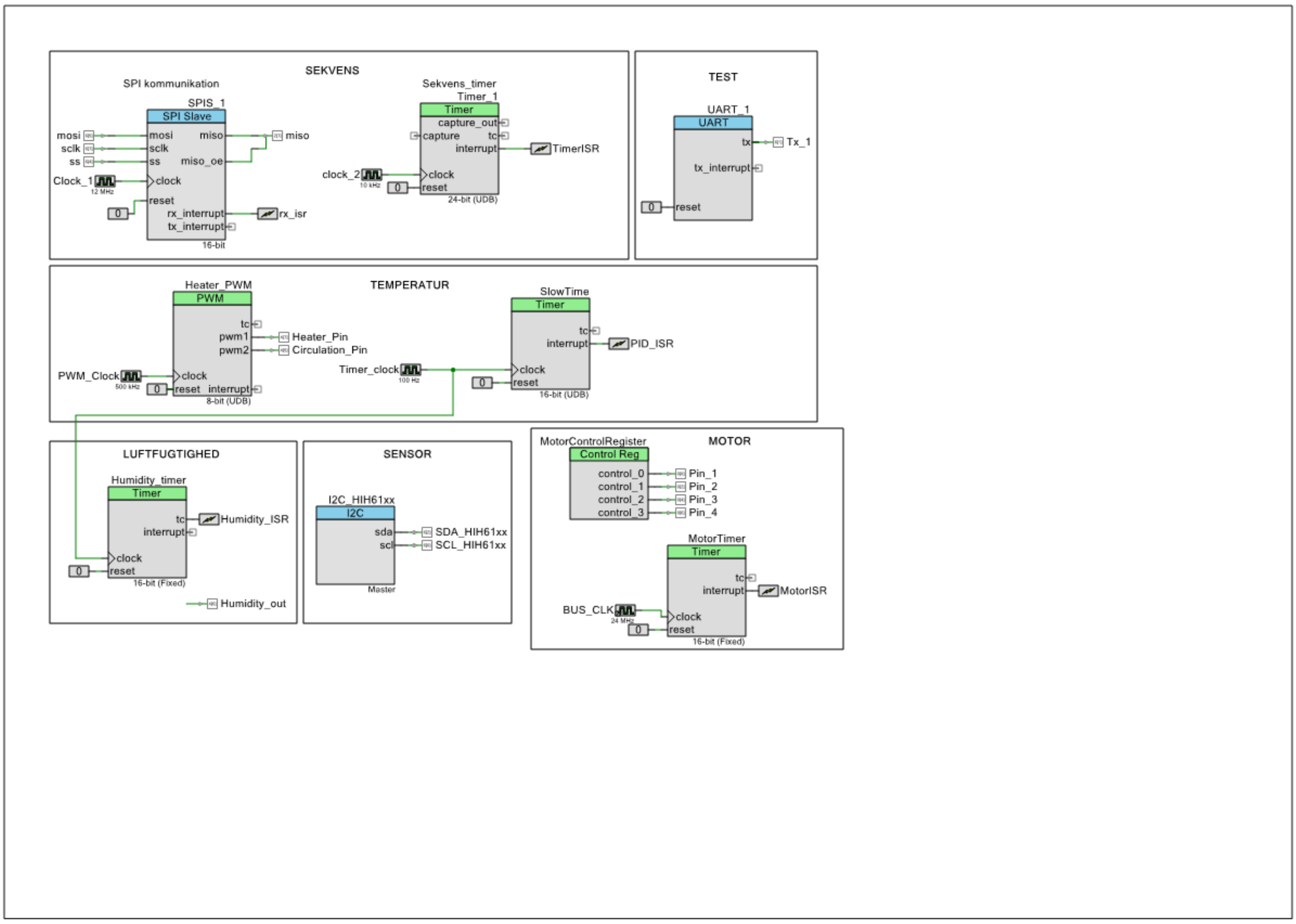
- Flash Protection chapter in the PSoC 3 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CyFlash API routines
  - CyWrite API routines

# 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:

- Instance Heater_PWM (type: PWM_v3_0)
- Instance Humidity_timer (type: Timer_v2_50)
- Instance I2C_HIH61xx (type: I2C_v3_30)
- Instance MotorControlRegister (type: CyControlReg_v1_70)
- Instance MotorTimer (type: Timer_v2_50)
- Instance SlowTime (type: Timer_v2_50)
- Instance SPIS_1 (type: SPI_Slave_v2_60)
- Instance Timer_1 (type: Timer_v2_50)
- Instance UART_1 (type: UART_v2_30)

# 8 Components

## 8.1 Component type: CyControlReg [v1.70]

### *8.1.1 Instance MotorControlRegister*

**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 12. Component Parameters for MotorControlRegister

| 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 | 3 | Defines bit value |
| BusDisplay | false | Displays the output terminals as bus |
| ExternalReset | false | Shows the reset terminal |
| NumOutputs | 4 | Defines the number of outputs needed (1-8) |

## 8.2 Component type: I2C [v3.30]

### *8.2.1 Instance I2C_HIH61xx*

**Description: Standard I2C communication interface**
**Instance type: I2C [v3.30]**
**Datasheet: online component datasheet for I2C**

Table 13. Component Parameters for I2C_HIH61xx

| Parameter Name | Value | Description |
|---|---|---|
| Address_Decode | Hardware | Determines either hardware or software address match logic. |
| BusSpeed_kHz | 100 | I2C Data Rate in kbps. Standard settings are 50, 100, 400 or 1000. The value must be between 1 and 1000. |
| EnableWakeup | false | Determines if I2C is selected as wakeup source. |
| ExternalBuffer | false | Exposes scl and sda in and out terminals outside the component. |
| Externi2cIntrHandler | false | Allows I2C interrupt handler to be set outside the I2C component. This feature intended only for PM/SM bus usage. |

| Parameter Name | Value | Description |
|---|---|---|
| ExternTmoutIntrHandler | false | Allows I2C timeout interrupt handler to be set outside the I2C component. This feature intended only for PM/SM bus usage. |
| Hex | false | Indicates that address has been input in hexadecimal format. |
| I2C_Mode | Master | Determines I2C mode - (Slave/Master/Multi--Master/Multi-Master-Slave). |
| I2cBusPort | Any | Determines which I2C pins have been selected. Select I2C0/I2C1 and connect to corresponding pins to be able use I2C as wakeup source. |
| Implementation | FixedFunction | Determines either I2C implementation Fixed Function or UDB. |
| NotSlaveClockMinusTolerance | 25 | Internal component clock negative tolerance value in Master, Multi-Master or Multi-Master-Slave mode. |
| NotSlaveClockPlusTolerance | 5 | Internal component clock positive tolerance value in Master, Multi-Master or Multi-Master-Slave mode. |
| PrescalerEnabled | false | Enables prescaler (7-bit counter) to expand timeout timer range. |
| PrescalerPeriod | 3 | Prescaler period of timeout timer. |
| SclTimeoutEnabled | false | Enables low time monitoring of scl line. |
| SdaTimeoutEnabled | false | Enables low time monitoring of sda line. |
| Slave_Address | 8 | 7-bits I2C slave address. |
| SlaveClockMinusTolerance | 5 | Internal component clock negative tolerance value in Slave mode. |
| SlaveClockPlusTolerance | 50 | Internal component clock positive tolerance value in Slave mode. |
| TimeoutImplementation | UDB | Determines either timeout timer feature implementation as UDB or Fixed Function. The Fixed Function implementation only available for PSoC5LP. |
| TimeOutms | 25 | Determines maximum time allowed for scl or sda to be low state (in mS). The timeout timer generates interrupt after timeout expires. |
| TimeoutPeriodff | 39999 | Period of timeout timer (Fixed Function). |
| TimeoutPeriodUdb | 39999 | Period of timeout timer (UDB). |
| UdbInternalClock | false | Determines either internal or external clock source for I2C UDB. |

| Parameter Name | Value | Description |
|---|---|---|
| UdbSlaveFixedPlacementEnable | false | Enables fixed placement for I2C UDB. Only available in slave mode. |

## 8.3 Component type: PWM [v3.0]

### *8.3.1 Instance Heater_PWM*

**Description: 8 or 16-bit Pulse Width Modulator**
**Instance type: PWM [v3.0]**
**Datasheet: online component datasheet for PWM**

Table 14. Component Parameters for Heater_PWM

| 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 | 0 | Compares Output 1 to value |
| CompareValue2 | 127 | 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 |
| 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. |

| Parameter Name | Value | Description |
|---|---|---|
| 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 | Two Outputs | Defines the overall mode of the PWM |
| Resolution | 8 | Defines the bit width of the PWM (8 or 16 bits) |
| RunMode | Continuous | Defines the run mode options to be either continuous or one shot |
| TriggerMode | None | Determines the mode of starting the PWM, i.e. triggering the PWM counter to start |
| UseInterrupt | true | Enables the placement and usage of the status register |

## 8.4 Component type: SPI_Slave [v2.60]

### *8.4.1 Instance SPIS_1*

**Description: Serial Peripheral Interface Slave**
**Instance type: SPI_Slave [v2.60]**
**Datasheet: online component datasheet for SPI_Slave**

Table 15. Component Parameters for SPIS_1

| Parameter Name | Value | Description |
|---|---|---|
| BidirectMode | false | Bidirectional mode setting |
| ClockInternal | false | Defines whether internal clock is used or not |
| DesiredBitRate | 1000000 | Desired Bit Rate in Hz |
| FixedPlacementEnabled | false | |
| InterruptOnByteComplete | true | Set Initial Interrupt Source to Enable Interrupt on Byte Transfer Complete |
| InterruptOnDone | false | Set Initial Interrupt Source to Enable Interrupt on SPI Done |
| InterruptOnRXEmpty | false | Set Initial Interrupt Source to Enable Interrupt on RX FIFO Empty |
| InterruptOnRXFull | false | Set Initial Interrupt Source to Enable Interrupt on RX FIFO full |
| InterruptOnRXNotEmpty | true | Set Initial Interrupt Source to Enable Interrupt on RX Not Empty |
| InterruptOnRXOverrun | false | Set Initial Interrupt Source to Enable Interrupt on RX FIFO overrun |
| InterruptOnTXEmpty | false | Set Initial Interrupt Source to Enable Interrupt on TX FIFO Empty |
| InterruptOnTXFull | false | Set Initial Interrupt Source to Enable Interrupt on TX FIFO full |
| InterruptOnTXNotFull | false | Set Initial Interrupt Source to Enable Interrupt on TX FIFO not full |

| Parameter Name | Value | Description |
|---|---|---|
| Mode | CPHA = 1, CPOL = 1 | Allows for setting the SPI Clock Polarity and Clock Phase from one of the four well known modes |
| MultiSlaveEnable | true | Allows using of the SPI MISO output enable terminal for multislave mode support |
| NumberOfDataBits | 16 | Data Width (3-16 bits) |
| RxBufferSize | 4 | RAM size used to store RX Data |
| ShiftDir | MSB First | Data Shift Direction (MSB First or LSB First) |
| TxBufferSize | 4 | RAM size used to store TX Data |
| UseInternalInterrupt | false | Defines whether internal interrupt is used or not |
| UseRxInternalInterrupt | false | Defines whether Rx internal interrupt is used or not |
| UseTxInternalInterrupt | false | Defines whether Tx internal interrupt is used or not |

## 8.5 Component type: Timer [v2.50]

### *8.5.1 Instance Humidity_timer*

**Description: 8, 16, 24 or 32-bit Timer**
**Instance type: Timer [v2.50]**
**Datasheet: online component datasheet for Timer**

Table 16. Component Parameters for Humidity_timer

| Parameter Name | Value | Description |
|---|---|---|
| CaptureAlternatingFall | false | Enables data capture on either edge but not until a valid falling edge is detected first. |
| CaptureAlternatingRise | false | Enables data capture on either edge but not until a valid rising edge is detected first. |
| CaptureCount | 2 | The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed. |
| CaptureCounterEnabled | false | Enables the capture counter to count capture events (up to 127) before a capture is triggered. |
| CaptureMode | None | This parameter defines the capture input signal requirements to trigger a valid capture event |

| Parameter Name | Value | Description |
|---|---|---|
| EnableMode | Software Only | This parameter specifies the methods in enabling the component. Hardware mode makes the enable input pin visible. Software mode may reduce the resource usage if not enabled. |
| FixedFunction | true | Configures the component to use fixed function HW block instead of the UDB implementation. |
| InterruptOnCapture | false | Parameter to check whether interrupt on a capture event is enabled or disabled. |
| InterruptOnFIFOFull | false | Parameter to check whether interrupt on a FIFO Full event is enabled disabled. |
| InterruptOnTC | true | Parameter to check whether interrupt on a TC is enabled or disabled. |
| NumberOfCaptures | 1 | Number of captures allowed until the counter is cleared or disabled. |
| Period | 2999 | Defines the timer period (This is also the reload value when terminal count is reached) |
| Resolution | 16 | Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals. |
| RunMode | Continuous | Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered. |
| TriggerMode | None | Defines the required trigger input signal to cause a valid trigger enable of the timer |

### *8.5.2 Instance MotorTimer*

**Description: 8, 16, 24 or 32-bit Timer**
**Instance type: Timer [v2.50]**
**Datasheet: online component datasheet for Timer**

Table 17. Component Parameters for MotorTimer

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

| Parameter Name | Value | Description |
|---|---|---|
| CaptureCount | 2 | The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed. |
| CaptureCounterEnabled | false | Enables the capture counter to count capture events (up to 127) before a capture is triggered. |
| CaptureMode | 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 | true | Configures the component to use fixed function HW block instead of the UDB implementation. |
| InterruptOnCapture | false | Parameter to check whether interrupt on a capture event is enabled or disabled. |
| InterruptOnFIFOFull | false | Parameter to check whether interrupt on a FIFO Full event is enabled disabled. |
| InterruptOnTC | true | Parameter to check whether interrupt on a TC is enabled or disabled. |
| NumberOfCaptures | 1 | Number of captures allowed until the counter is cleared or disabled. |
| Period | 65535 | Defines the timer period (This is also the reload value when terminal count is reached) |
| Resolution | 16 | Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals. |
| RunMode | Continuous | Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered. |
| TriggerMode | None | Defines the required trigger input signal to cause a valid trigger enable of the timer |

### *8.5.3 Instance SlowTime*

**Description: 8, 16, 24 or 32-bit Timer**
**Instance type: Timer [v2.50]**
**Datasheet: online component datasheet for Timer**

Table 18. Component Parameters for SlowTime

| Parameter Name | Value | Description |
|---|---|---|
| CaptureAlternatingFall | false | Enables data capture on either edge but not until a valid falling edge is detected first. |
| CaptureAlternatingRise | false | Enables data capture on either edge but not until a valid rising edge is detected first. |
| CaptureCount | 2 | The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed. |
| CaptureCounterEnabled | false | Enables the capture counter to count capture events (up to 127) before a capture is triggered. |
| CaptureMode | None | This parameter defines the capture input signal requirements to trigger a valid capture event |
| EnableMode | Software Only | This parameter specifies the methods in enabling the component. Hardware mode makes the enable input pin visible. Software mode may reduce the resource usage if not enabled. |
| FixedFunction | false | Configures the component to use fixed function HW block instead of the UDB implementation. |
| InterruptOnCapture | false | Parameter to check whether interrupt on a capture event is enabled or disabled. |
| InterruptOnFIFOFull | false | Parameter to check whether interrupt on a FIFO Full event is enabled disabled. |
| InterruptOnTC | true | Parameter to check whether interrupt on a TC is enabled or disabled. |
| NumberOfCaptures | 1 | Number of captures allowed until the counter is cleared or disabled. |
| Period | 2999 | Defines the timer period (This is also the reload value when terminal count is reached) |

| Parameter Name | Value | Description |
|---|---|---|
| Resolution | 16 | Defines the resolution of the hardware. This parameter affects how many bits are used in the Period counter and defines the maximum resolution of the internal component signals. |
| RunMode | Continuous | Defines the hardware to run continuously, run until a terminal count is reached or run until an interrupt event is triggered. |
| TriggerMode | None | Defines the required trigger input signal to cause a valid trigger enable of the timer |

### *8.5.4 Instance Timer_1*

**Description: 8, 16, 24 or 32-bit Timer**
**Instance type: Timer [v2.50]**
**Datasheet: online component datasheet for Timer**

Table 19. Component Parameters for Timer_1

| Parameter Name | Value | Description |
|---|---|---|
| CaptureAlternatingFall | false | Enables data capture on either edge but not until a valid falling edge is detected first. |
| CaptureAlternatingRise | false | Enables data capture on either edge but not until a valid rising edge is detected first. |
| CaptureCount | 2 | The CaptureCount parameter works as a divider on the hardware input "capture". A CaptureCount value of 2 would result in an actual capture taking place every other time the input "capture" is changed. |
| CaptureCounterEnabled | false | Enables the capture counter to count capture events (up to 127) before a capture is triggered. |
| CaptureMode | Rising Edge | This parameter defines the capture input signal requirements to trigger a valid capture event |
| EnableMode | Software Only | This parameter specifies the methods in enabling the component. Hardware mode makes the enable input pin visible. Software mode may reduce the resource usage if not enabled. |
| FixedFunction | 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 | true | Parameter to check whether interrupt on a TC is enabled or disabled. |
| NumberOfCaptures | 1 | Number of captures allowed until the counter is cleared or disabled. |
| Period | 999 | 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_1*

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

Table 20. Component Parameters for UART_1

| 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. |

| Parameter Name | Value | Description |
|---|---|---|
| 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 | 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 | false | 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 | Even | 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 | false | Enables the RX in the UART |

| Parameter Name | Value | Description |
|---|---|---|
| TXBitClkGenDP | true | When enabled, this parameter enables the TX clock generation on DataPath resource. When disabled, TX clock is generated from Clock7. |
| TXBufferSize | 4 | The size of the RAM space allocated for the TX output buffer. |
| TXEnable | true | Enables the TX in the UART |
| Use23Polling | true | Allows the use of 2 out of 3 polling resources on the RX UART sampler. |

# 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 3 register map is covered in the PSoC 3 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 3 Technical Reference Manual
    - Low Power Modes chapter in the PSoC 3 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 3 Technical Reference Manual
    - Cache chapter in the System Reference Guide