# PSoC® Creator™
# Project Datasheet for PredlogZabavnaKocka_01_KIT049_2018_001

**Creation Time: 02/02/2018 09:21:20**
**User: Vojko-HP\vojko_new**
**Project: PredlogZabavnaKocka_01_KIT049_2018_001**
**Tool: PSoC Creator 4.1 Update 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 4 is a family of 32-bit devices with the following characteristics:

- Digital system that includes configurable Universal Digital Blocks (UDBs) and specific function peripherals such as PWM, UART, SPI and I2C
- Analog subsystem that includes 12-bit SAR ADC, comparators, op amps, CapSense, LCD drive and more
- Several types of memory elements, including SRAM and flash
- Programming and debug system through Serial Wire Debug (SWD)
- High-performance 32-bit ARM Cortex-M0 core with a nested vectored interrupt controller (NVIC)
- Flexible routing to all pins

Figure 1 shows the major components of a typical PSoC 4200 series member PSoC 4 device. For details on all the systems listed above, please refer to the PSoC 4 Technical Reference Manual .

Figure 1. PSoC 4200 Device Series Block Diagram

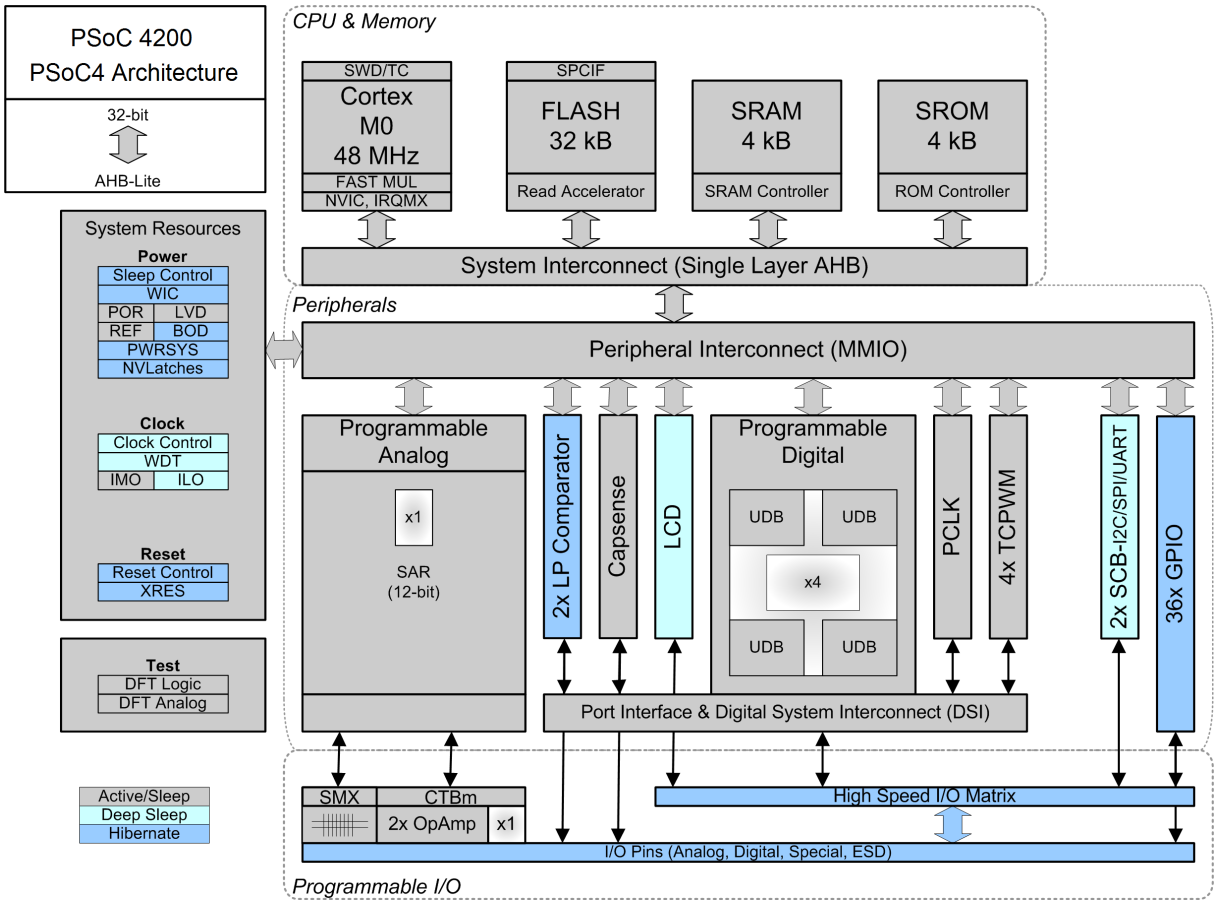


Table 1 lists the key characteristics of this device.

Table 1. Device Characteristics

| Name | Value |
|---|---|
| Part Number | CY8C4245AXI-483 |
| Package Name | 44-TQFP |
| Family | PSoC 4 |
| Series | PSoC 4200 |
| Max CPU speed (MHz) | 48 |
| Flash size (kB) | 32 |
| SRAM size (kB) | 4 |
| Vdd range (V) | 1.71 to 5.5 |
| Automotive qualified | No (Industrial Grade Only) |
| Temp range (Celsius) | -40 to 85 |

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

Table 2 lists the device resources that this design uses:

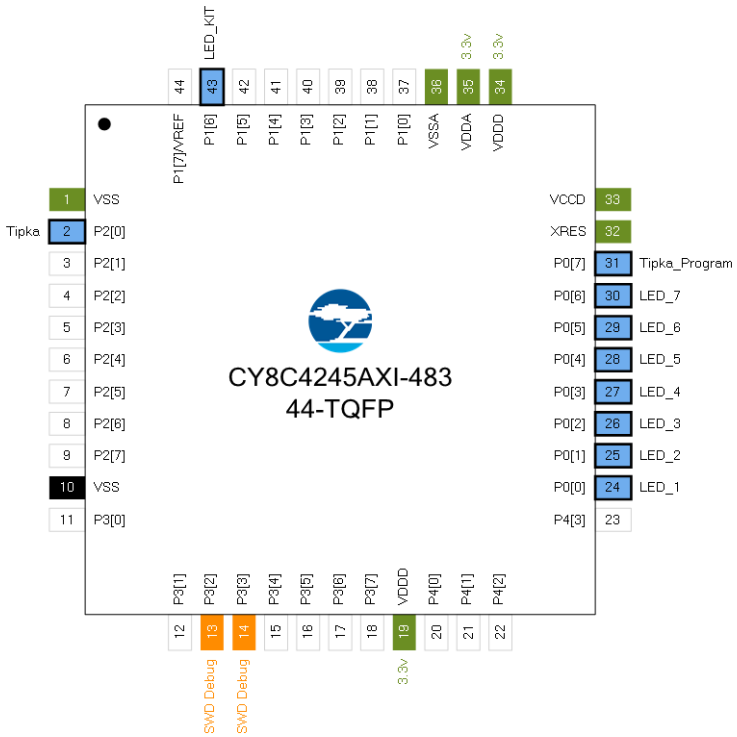
Table 2. Device Resources

| Resource Type | Used | Free | Max | % Used |
|---|---|---|---|---|
| Digital Clocks | 1 | 3 | 4 | 25.00 % |
| Interrupts | 0 | 32 | 32 | 0.00 % |
| IO | 12 | 24 | 36 | 33.33 % |
| Segment LCD | 0 | 1 | 1 | 0.00 % |
| CapSense | 0 | 1 | 1 | 0.00 % |
| Die Temp | 0 | 1 | 1 | 0.00 % |
| Serial Communication (SCB) | 0 | 2 | 2 | 0.00 % |
| Timer/Counter/PWM | 0 | 4 | 4 | 0.00 % |
| UDB | | | | |
| Macrocells | 5 | 27 | 32 | 15.63 % |
| Unique P-terms | 5 | 59 | 64 | 7.81 % |
| Total P-terms | 5 | | | |
| Datapath Cells | 2 | 2 | 4 | 50.00 % |
| Status Cells | 1 | 3 | 4 | 25.00 % |
| StatusI Registers | 1 | | | |
| Control Cells | 1 | 3 | 4 | 25.00 % |
| Control Registers | 1 | | | |
| Comparator/Opamp | 0 | 2 | 2 | 0.00 % |
| LP Comparator | 0 | 2 | 2 | 0.00 % |
| SAR ADC | 0 | 1 | 1 | 0.00 % |
| DAC | | | | |
| 7-bit IDAC | 0 | 1 | 1 | 0.00 % |
| 8-bit IDAC | 0 | 1 | 1 | 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 |
|---|---|---|---|---|
| 1 | VSS | VSS | Power | |
| 2 | P2[0] | Tipka | Software In/Out | Res pull up |
| 3 | P2[1] | GPIO [unused] | | |
| 4 | P2[2] | GPIO [unused] | | |
| 5 | P2[3] | GPIO [unused] | | |
| 6 | P2[4] | GPIO [unused] | | |
| 7 | P2[5] | GPIO [unused] | | |
| 8 | P2[6] | GPIO [unused] | | |
| 9 | P2[7] | GPIO [unused] | | |
| 11 | P3[0] | GPIO [unused] | | |
| 12 | P3[1] | GPIO [unused] | | |
| 13 | P3[2] | Debug:SWD_IO | Reserved | |
| 14 | P3[3] | Debug:SWD_CK | Reserved | |
| 15 | P3[4] | GPIO [unused] | | |
| 16 | P3[5] | GPIO [unused] | | |
| 17 | P3[6] | GPIO [unused] | | |
| 18 | P3[7] | GPIO [unused] | | |
| 19 | VDDD | VDDD | Power | |
| 20 | P4[0] | GPIO [unused] | | |
| 21 | P4[1] | GPIO [unused] | | |
| 22 | P4[2] | GPIO [unused] | | |
| 23 | P4[3] | GPIO [unused] | | |
| 24 | P0[0] | LED_1 | Software In/Out | Strong drive |
| 25 | P0[1] | LED_2 | Software In/Out | Strong drive |
| 26 | P0[2] | LED_3 | Software In/Out | Strong drive |
| 27 | P0[3] | LED_4 | Software In/Out | Strong drive |
| 28 | P0[4] | LED_5 | Software In/Out | Strong drive |
| 29 | P0[5] | LED_6 | Software In/Out | Strong drive |
| 30 | P0[6] | LED_7 | Software In/Out | Strong drive |
| 31 | P0[7] | Tipka_Program | Software In/Out | Res pull up |
| 32 | XRES | XRES | Dedicated | |
| 33 | VCCD | VCCD | Power | |
| 34 | VDDD | VDDD | Power | |
| 35 | VDDA | VDDA | Power | |
| 36 | VSSA | VSSA | Power | |
| 37 | P1[0] | GPIO [unused] | | |
| 38 | P1[1] | GPIO [unused] | | |

| Pin | Port | Name | Type | Drive Mode |
|---|---|---|---|---|
| 39 | P1[2] | GPIO [unused] | | |
| 40 | P1[3] | GPIO [unused] | | |
| 41 | P1[4] | GPIO [unused] | | |
| 42 | P1[5] | GPIO [unused] | | |
| 43 | P1[6] | LED_KIT | Dgtl Out | Strong drive |
| 44 | P1[7]/VREF | GPIO [unused] | | |

Abbreviations used in Table 3 have the following meanings:

- Res pull up = Resistive pull up
- Dgtl Out = Digital Output

## 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 |
|---|---|---|---|---|
| P0[0] | 24 | LED_1 | Software In/Out | Strong drive |
| P0[1] | 25 | LED_2 | Software In/Out | Strong drive |
| P0[2] | 26 | LED_3 | Software In/Out | Strong drive |
| P0[3] | 27 | LED_4 | Software In/Out | Strong drive |
| P0[4] | 28 | LED_5 | Software In/Out | Strong drive |
| P0[5] | 29 | LED_6 | Software In/Out | Strong drive |
| P0[6] | 30 | LED_7 | Software In/Out | Strong drive |
| P0[7] | 31 | Tipka_Program | Software In/Out | Res pull up |
| P1[0] | 37 | GPIO [unused] | | |
| P1[1] | 38 | GPIO [unused] | | |
| P1[2] | 39 | GPIO [unused] | | |
| P1[3] | 40 | GPIO [unused] | | |
| P1[4] | 41 | GPIO [unused] | | |
| P1[5] | 42 | GPIO [unused] | | |
| P1[6] | 43 | LED_KIT | Dgtl Out | Strong drive |
| P1[7]/VREF | 44 | GPIO [unused] | | |
| P2[0] | 2 | Tipka | Software In/Out | Res pull up |
| P2[1] | 3 | GPIO [unused] | | |
| P2[2] | 4 | GPIO [unused] | | |
| P2[3] | 5 | GPIO [unused] | | |
| P2[4] | 6 | GPIO [unused] | | |
| P2[5] | 7 | GPIO [unused] | | |
| P2[6] | 8 | GPIO [unused] | | |
| P2[7] | 9 | GPIO [unused] | | |
| P3[0] | 11 | GPIO [unused] | | |
| P3[1] | 12 | GPIO [unused] | | |
| P3[2] | 13 | Debug:SWD_IO | Reserved | |
| P3[3] | 14 | Debug:SWD_CK | Reserved | |
| P3[4] | 15 | GPIO [unused] | | |
| P3[5] | 16 | GPIO [unused] | | |
| P3[6] | 17 | GPIO [unused] | | |
| P3[7] | 18 | GPIO [unused] | | |
| P4[0] | 20 | GPIO [unused] | | |
| P4[1] | 21 | GPIO [unused] | | |
| P4[2] | 22 | GPIO [unused] | | |
| P4[3] | 23 | GPIO [unused] | | |

Abbreviations used in Table 4 have the following meanings:

- Res pull up = Resistive pull up
- Dgtl Out = Digital Output

## 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 |
|---|---|---|
| Debug:SWD_CK | P3[3] | Reserved |
| Debug:SWD_IO | P3[2] | Reserved |
| GPIO [unused] | P3[6] | |
| GPIO [unused] | P4[1] | |
| GPIO [unused] | P3[7] | |
| GPIO [unused] | P4[0] | |
| GPIO [unused] | P1[7]/VREF | |
| GPIO [unused] | P1[3] | |
| GPIO [unused] | P1[4] | |
| GPIO [unused] | P1[5] | |
| GPIO [unused] | P1[2] | |
| GPIO [unused] | P4[3] | |
| GPIO [unused] | P1[0] | |
| GPIO [unused] | P1[1] | |
| GPIO [unused] | P2[1] | |
| GPIO [unused] | P2[4] | |
| GPIO [unused] | P2[6] | |
| GPIO [unused] | P2[3] | |
| GPIO [unused] | P2[5] | |
| GPIO [unused] | P2[2] | |
| GPIO [unused] | P3[4] | |
| GPIO [unused] | P4[2] | |
| GPIO [unused] | P3[1] | |
| GPIO [unused] | P3[5] | |
| GPIO [unused] | P2[7] | |
| GPIO [unused] | P3[0] | |
| LED_1 | P0[0] | Software In/Out |
| LED_2 | P0[1] | Software In/Out |
| LED_3 | P0[2] | Software In/Out |
| LED_4 | P0[3] | Software In/Out |
| LED_5 | P0[4] | Software In/Out |
| LED_6 | P0[5] | Software In/Out |
| LED_7 | P0[6] | Software In/Out |
| LED_KIT | P1[6] | Dgtl Out |
| Tipka | P2[0] | Software In/Out |
| Tipka_Program | P0[7] | Software In/Out |

Abbreviations used in Table 5 have the following meanings:

- 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 |
| Unused Bonded IO | Disallowed |
| Heap Size (bytes) | 0x80 |
| Stack Size (bytes) | 0x0400 |
| Include CMSIS Core Peripheral Library Files | True |

## 3.2 System Debug Settings

Table 7. System Debug Settings

| Name | Value |
|---|---|
| Debug Select | SWD (serial wire debug) |
| Chip Protection | Open |

## 3.3 System Operating Conditions

Table 8. System Operating Conditions

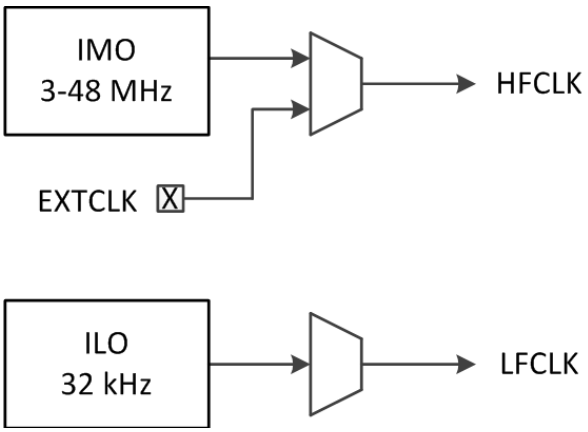
| Name | Value |
|---|---|
| VDDA (V) | 3.3 |
| VDDD (V) | 3.3 |
| Variable VDDA | True |

# 4 Clocks

The clock system includes these clock resources:

- Two internal clock sources:
  - 3 to 48 MHz Internal Main Oscillator (IMO) ±2% at 3 MHz
  - 32 kHz Internal Low Speed Oscillator (ILO) output
- HFCLK can be generated using an external signal from EXTCLK pin
- Twelve clock dividers, each with 16-bit divide capability:
  - Eight can be used for fixed-function blocks
  - Four can be used for the UDBs

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 |
|---|---|---|---|---|---|---|---|
| DPLL_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| SysClk | NONE | HFClk | ? MHz | 24 MHz | ±2 | True | True |
| Direct_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| PLL1_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| PLL0_Sel | NONE | IMO | 24 MHz | 24 MHz | ±2 | True | True |
| HFClk | NONE | Direct_Sel | 24 MHz | 24 MHz | ±2 | True | True |
| IMO | NONE | | 24 MHz | 24 MHz | ±2 | True | True |
| LFClk | NONE | ILO | ? MHz | 32 kHz | ±60 | True | True |
| ILO | NONE | | 32 kHz | 32 kHz | ±60 | True | True |
| Timer2 (WDT2) | NONE | LFClk | ? MHz | ? MHz | ±0 | False | False |
| ExtClk | NONE | | 24 MHz | ? MHz | ±0 | False | False |
| DigSig3 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| DigSig2 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| DigSig4 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| DigSig1 | NONE | | ? MHz | ? MHz | ±0 | False | False |
| Timer1 (WDT1) | NONE | LFClk | ? MHz | ? MHz | ±0 | False | False |
| Timer0 (WDT0) | NONE | LFClk | ? 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

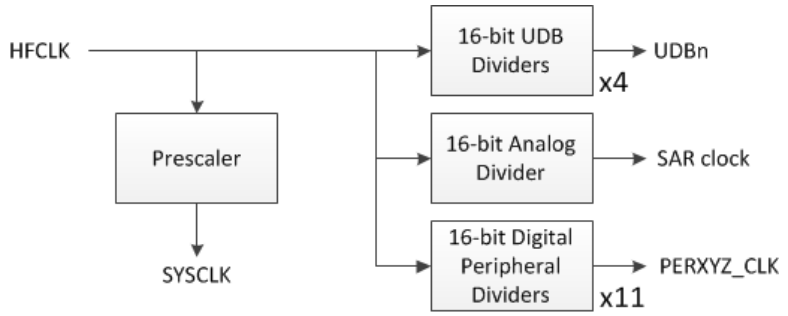



Table 10 lists the local clocks used in this design.

Table 10. Local Clocks

| Name | Domain | Source | Desired Freq | Nominal Freq | Accuracy (%) | Start at Reset | Enabled |
|---|---|---|---|---|---|---|---|
| Clock | DIGITAL | HFClk | 1 kHz | 1 kHz | ±2 | True | True |

For more information on clocking resources, please refer to:

- Clocking System chapter in the PSoC 4 Technical Reference Manual
- Clocking chapter in the System Reference Guide
  - CySysClkImo API routines
  - CySysClkIlo API routines
  - CySysClkWrite API routines

# 5 Interrupts

## 5.1 Interrupts

This design contains no interrupt components.

# 6 Flash Memory

PSoC 4 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 | 0xFFF | W - Full Protection |
| 0x1000 | 0x7FFF | U - Unprotected |

Flash memory is organized as rows with each row of flash having 128 bytes. Each flash row can be assigned one of four protection levels:

- U - Unprotected
- W - Full Protection

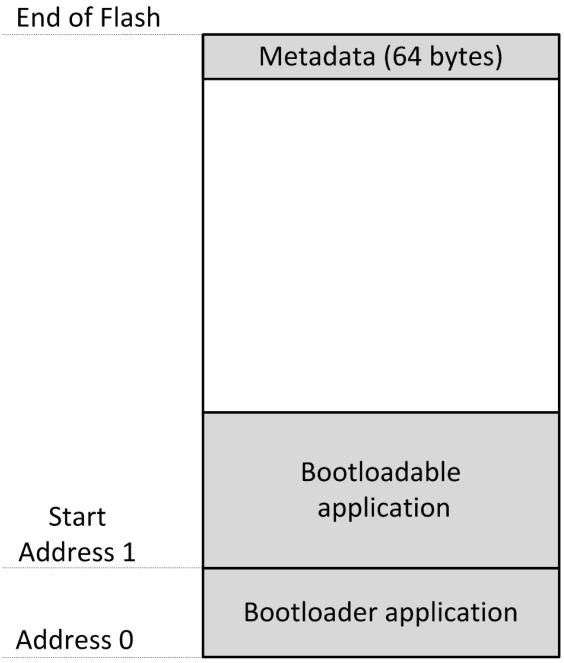
For more information on Flash memory and protection, please refer to:

- Flash Protection chapter in the PSoC 4 Technical Reference Manual
- Flash and EEPROM chapter in the System Reference Guide
  - CySysFlash API routines

# 7 Bootloader and Bootloadable

Figure 5 details the Flash memory map for the bootloader and/or bootloadable application(s) included in this design.

Figure 5. Bootloader Memory Map




## 7.1 Bootloadable Application

Table 12. Bootloadable Settings

| Name | Value |
|---|---|
| Application Version | 0x0001 |
| Application ID | 0x0000 |
| Application Custom ID | 0x0 |
| Application Image 1 Start Address | 0x1100 |
| Application Image 1 End Address | 0x7FFF |
| Manual Application Image Placement | False |

## 7.2 Bootloader Application

Table 13. Bootloader Settings

| Name | Value |
|---|---|
| Checksum Type | BasicChecksum |
| Supports Multiple Application Images | False |
| Application Version | 0x0001 |
| Bootloader Start Address | 0x0 |
| Bootloader End Address | 0x10A8 |

For more information on the bootloader and startup please refer to:

- Startup and Linking chapter in the System Reference Guide

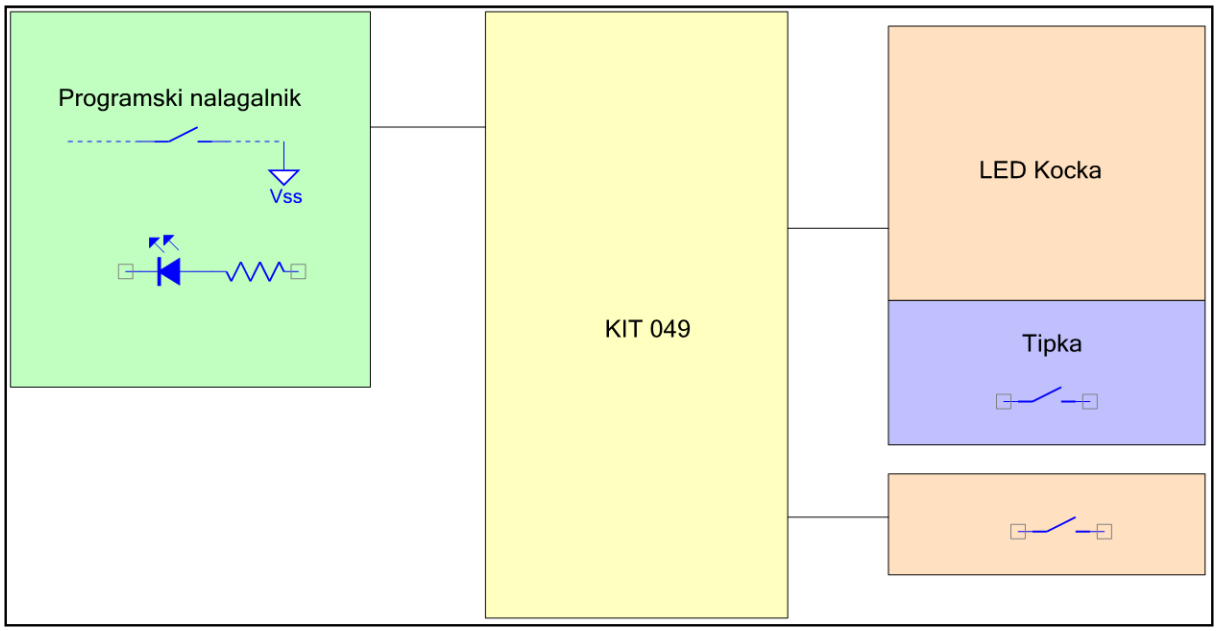
- Datasheet for Bootloader and Bootloadable component

# 8 Design Contents

This design's schematic content consists of the following 4 schematic sheets:

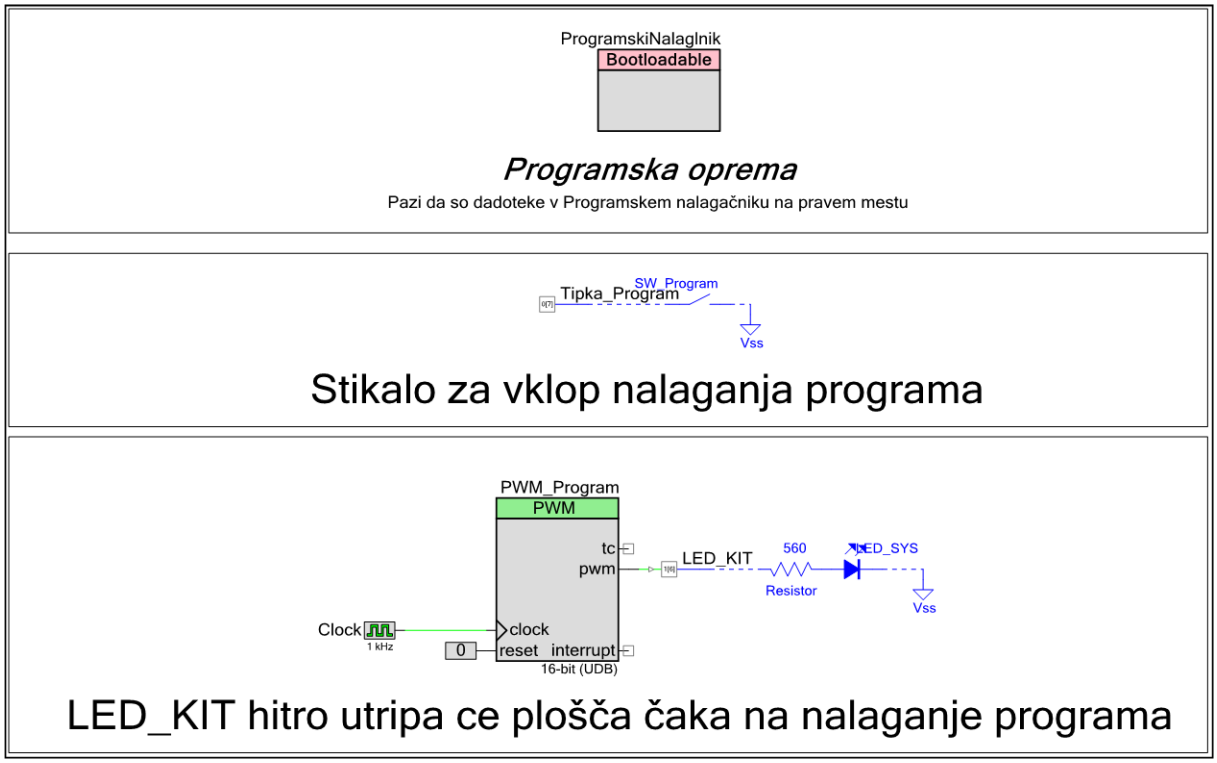
## 8.1 Schematic Sheet: BlokShema

Figure 6. Schematic Sheet: BlokShema



## 8.2 Schematic Sheet: Osnovno

Figure 7. Schematic Sheet: Osnovno

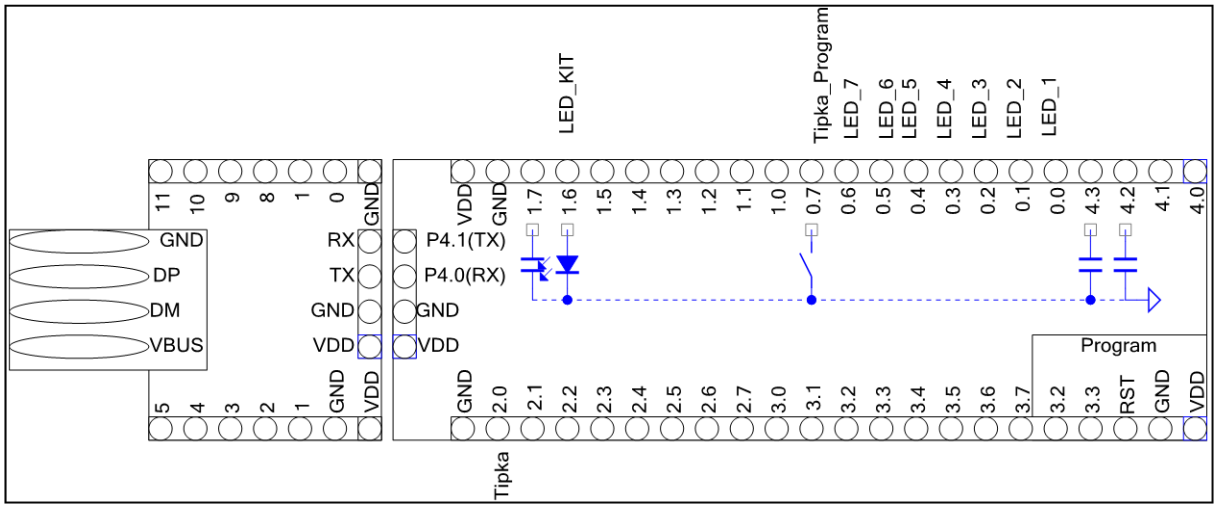



This schematic sheet contains the following component instances:

- Instance ProgramskiNalaglnik (type: Bootloadable_v1_50)
- Instance PWM_Program (type: PWM_v3_30)

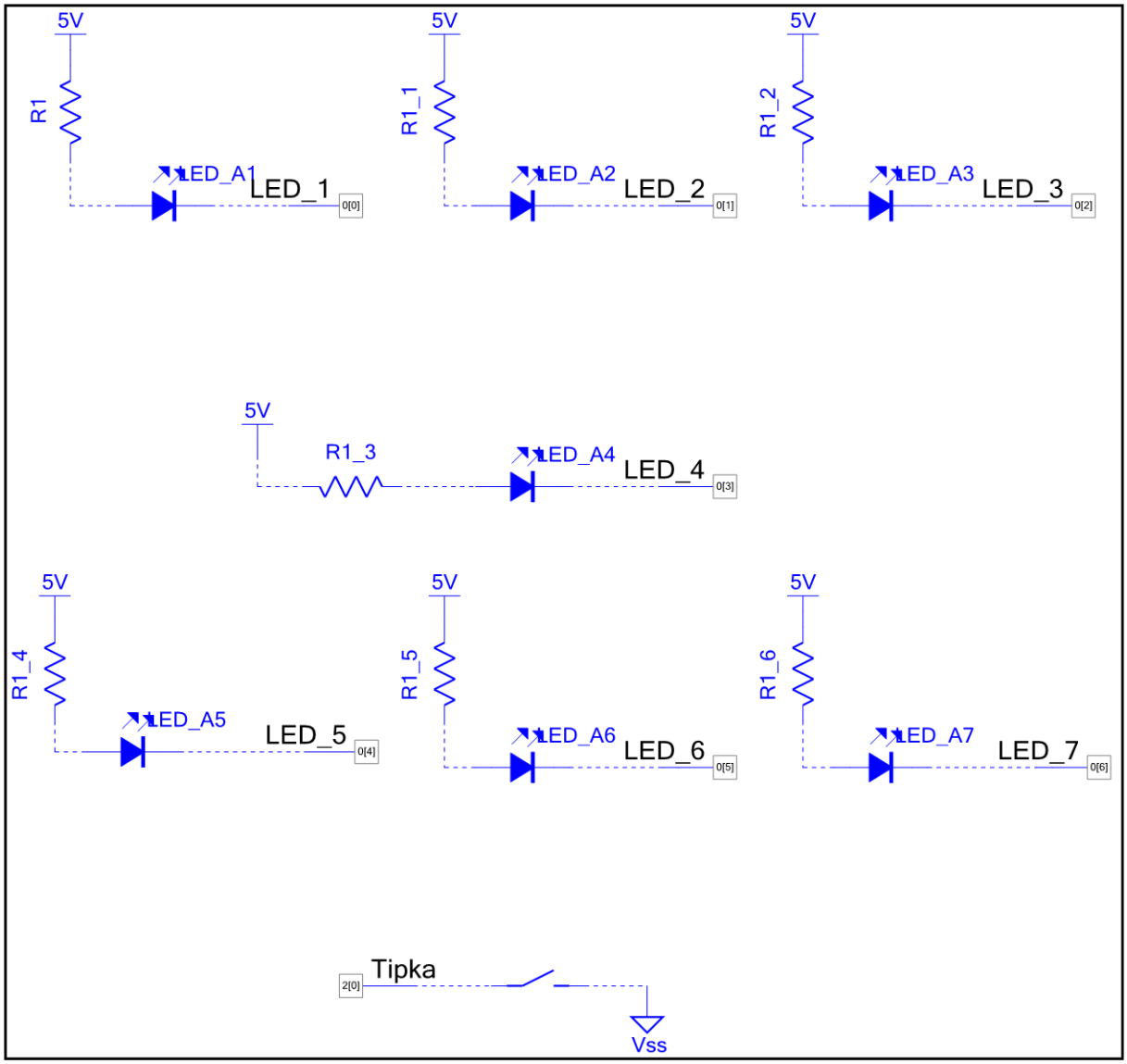
## 8.3 Schematic Sheet: CY8Ckit049

Figure 8. Schematic Sheet: CY8Ckit049



## 8.4 Schematic Sheet: LEDKocka

Figure 9. Schematic Sheet: LEDKocka



# 9 Components

## 9.1 Component type: Bootloadable [v1.50]

### *9.1.1 Instance ProgramskiNalaglnik*

**Description: Provides bootloadable application functionality.**
**Instance type: Bootloadable [v1.50]**
**Datasheet: online component datasheet for Bootloadable**

Table 14. Component Parameters for ProgramskiNalaglnik

| Parameter Name | Value | Description |
|---|---|---|
| appCustomID | 0 | Provides a 4 byte custom ID number to represent anything in the Bootloadable application. |
| appID | 0 | Provides a 2 byte number to represent the ID of the bootloadable application. |
| appVersion | 1 | Provides a 2 byte number to represent the version of the bootloadable application. |
| autoPlacement | true | Provides a method for PSoC Creator to place a Bootloadable application image at a specified location. If true, the image will be placed automatically. If false, the image will be placed at an address specified by the Placement Address option. |
| checksumExcludeSize | 0 | Provides a size in bytes of checksum exclude section |
| elfFilePath | ..\..\ProgramskiNala-galniki\ProgramskiNalagalnik_-CY8C42_001_-6junij2017\UART_-ProgramskiNalagalnik_42_-001.elf | Provides a reference to the Bootloader application (.elf) that is associated with this Bootloadable application. |
| hexFilePath | ..\..\ProgramskiNala-galniki\ProgramskiNalagalnik_-CY8C42_001_-6junij2017\UART_-ProgramskiNalagalnik_42_-001.hex | Provides a reference to the Bootloader application (.hex) that is associated with this Bootloadable application. |
| placementAddress | 0 | Allows specifying an address where the bootloadable application will be placed in the memory. Available only if the Automatic Application Image Placement option is true. |
| User Comments | | Instance-specific comments. |

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

### *9.2.1 Instance PWM_Program*

**Description: 8 or 16-bit Pulse Width Modulator**
**Instance type: PWM [v3.30]**

**Datasheet:** [online component datasheet for PWM]

Table 15. Component Parameters for PWM_Program

| 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 | 750 | Compares Output 1 to value |
| CompareValue2 | 63 | Compares Output 2 to value |
| DeadBand | Disabled | Defines whether dead band outputs are desired or not. |
| DeadTime | 1 | Defines the number of required dead band clock cycles |
| DitherOffset | 0.00 | Allows the user to implement dither to get more bits out of a 8 or 16 bit PWM. |
| EnableMode | Software Only | Specifies the method of enabling the PWM. This can be either hardware or software. |
| FixedFunction | false | Determines whether the fixed function counter timer is used or the UDB implementation is used. |
| InterruptOnCMP1 | false | Enables the interrupt on compare1 true event |
| InterruptOnCMP2 | false | Enables the interrupt on compare2 true event |
| InterruptOnKill | false | Enables the interrupt on a kill event |
| InterruptOnTC | false | Enables the interrupt on terminal count event |
| KillMode | Disabled | Parameter to select the kill mode for build time. |
| MinimumKillTime | 1 | Sets the minimum number of clock cycles that a kill must be active on the outputs when KillMode is set to Minimum Kill Time mode |
| Period | 1500 | 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 |

| Parameter Name | Value | Description |
|---|---|---|
| TriggerMode | None | Determines the mode of starting the PWM, i.e. triggering the PWM counter to start |
| UseInterrupt | true | Enables the placement and usage of the status register |
| User Comments | | Instance-specific comments. |

# 10 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 4 register map is covered in the PSoC 4 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 4 Technical Reference Manual
  - Low Power Modes chapter in the PSoC 4 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