AN77759

Getting Started with PSoC® 5LP

Author: Nidhin MS

Associated Part Family: All PSoC 5LP parts

Related Documents: For a complete list, click here.

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN77759.

AN77759 introduces you to PSoC® 5LP, an ARM® Cortex®-M3-based programmable system-on-chip. It describes the PSoC 5LP architecture and development environment, and shows you how to create a simple design using PSoC Creator™, the development tool for PSoC 5LP. This application note also guides you to more resources for in-depth learning about PSoC 5LP as well as PSoC in general.

Contents

1 Introduction

PSoC 5LP is a true programmable embedded system-on-chip, integrating custom analog and digital peripheral functions, memory, and an ARM Cortex-M3 CPU on a single chip.

PSoC 5LP provides a cost-effective alternative to the combination of MCU and external ICs. The PSoC 5LP architecture boosts performance through:

- 32-bit ARM Cortex-M3 core plus DMA controller and digital filter processor, at up to 80 MHz
- Ultra-low power with industry's widest voltage range
- Programmable digital and analog peripherals enable custom functions
- Flexible routing of any analog or digital peripheral function to any pin

A single PSoC device can integrate as many as 100 digital and analog peripheral functions, reducing design time, board space, power consumption, and system cost while improving system quality.

Using this Document

The next few pages describe the PSoC 5LP and the advantages of designing with PSoC and PSoC Creator. Or, you can jump right in and quickly build a simple design – go to My First PSoC 5LP Design. The design created in this section is also available in code example CE203303.

2 PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see KBA86521, How to Design with PSoC 3, PSoC 4, and PSoC 5LP. The following is an abbreviated list for PSoC 5LP:

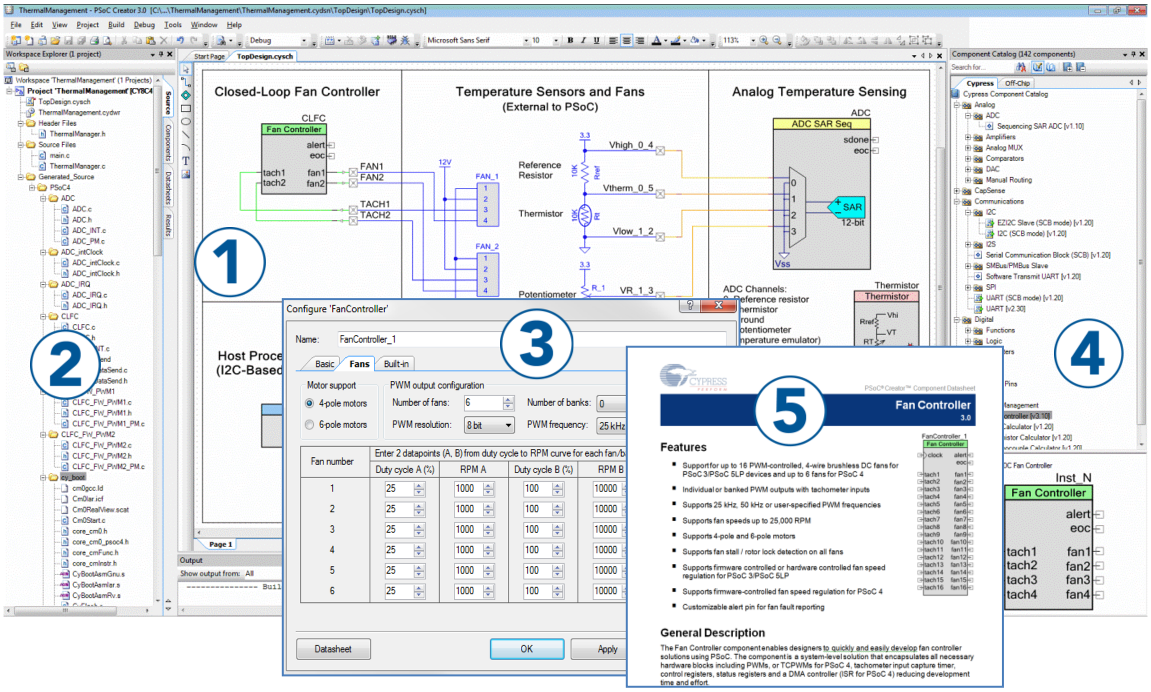
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **CapSense® Design Guides:** Learn how to design capacitive touch-sensing applications with the PSoC 3, PSoC 4, and PSoC 5LP families of devices.
- **Application Notes** and **Code Examples:** Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each of the PSoC 3, PSoC 4, and PSoC 5LP device families.
- **PSoC Training Videos:** These videos provide step-by-step instructions on how to get started building complex designs with PSoC.
- **Development Kits:**
  - CY8CKIT-059 is a low-cost prototyping platform for sampling PSoC 5LP devices.
  - CY8CKIT-050 is designed for analog performance. It enables you to develop and evaluate high-precision analog, low-power, and low-voltage applications.

3 PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, and PSoC 5LP. See Figure 1 – with PSoC Creator, you can:

1. Drag and drop Components for hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware
3. Configure Components using configuration tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 1. PSoC Creator Features


ThermalManagement - PSoC Creator 3.0
Workspace Explorer (1 project)
Component Catalog (142 components)
Closed-Loop Fan Controller
Temperature Sensors and Fans (External to PSoC)
Analog Temperature Sensing
CLFC
Fan Controller
ADC SAR Seq
Thermistor
Host Proce
(I2C-Based
1
2
3
4
5
Fan Controller

Configure 'FanController'

Name: FanController_1

Basic | **Fans** | Built-in

Motor support: ● 4-pole motors ○ 6-pole motors

PWM output configuration: Number of fans: 6; Number of banks: 0; PWM resolution: 8 bit; PWM frequency: 25 kHz

Fan number	Duty cycle A (%)	RPM A	Duty cycle B (%)	RPM B
1	25	1000	100	10000
2	25	1000	100	10000
3	25	1000	100	10000
4	25	1000	100	10000
5	25	1000	100	10000
6	25	1000	100	10000

Datasheet | OK | Apply

PSoC® Creator™ Component Datasheet

Fan Controller 3.0

Features

- Support for up to 16 PWM-controlled, 4-wire brushless DC fans for PSoC 3/PSoC 5LP devices and up to 6 fans for PSoC 4
- Individual or banked PWM outputs with tachometer inputs
- Supports 25 kHz, 50 kHz or user-specified PWM frequencies
- Supports fan speeds up to 25,000 RPM
- Supports 4-pole and 6-pole motors
- Supports fan stall / rotor lock detection on all fans
- Supports firmware controlled or hardware controlled fan speed regulation for PSoC 3/PSoC 5LP
- Supports firmware-controlled fan speed regulation for PSoC 4
- Customizable alert pin for fan fault reporting

General Description

The Fan Controller component enables designers to quickly and easily develop fan controller solutions using PSoC. The component is a system-level solution that encapsulates all necessary hardware blocks including PWMs, or TCPWMs for PSoC 4, tachometer input capture timer, control registers, status registers and a DMA controller (ISR for PSoC 4) reducing development time and effort.

3.1 PSoC Creator Help

Visit the PSoC Creator home page to download the latest version of PSoC Creator. Then, launch PSoC Creator and navigate to the following items:

- **Quick Start Guide:** Choose the menu item **Help** > **Documentation** > **Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple Component example projects:** Choose the menu item **File** > **Open** > **Example projects**. These example projects demonstrate how to configure and use PSoC Creator Components.
- **Starter designs:** Choose the menu item **File** > **New** > **Project** > **PSoC 5LP Starter Designs**. These starter designs demonstrate the unique features of PSoC 5LP.
- **System Reference Guide**: Choose the menu item **Help** > **System Reference** > **System Reference Guide**. This guide lists and describes the system functions provided by PSoC Creator.
- **Component datasheets:** Right-click a Component and select “Open Datasheet.” Visit the PSoC 5LP Component Datasheets page for a list of all PSoC 5LP Component datasheets.
- **Document Manager:** PSoC Creator provides a document manager to help you to easily find and review document resources. To open the document manager, choose the menu item **Help** > **Document Manager**.

3.2 Technical Support

If you have any questions, our technical support team is happy to assist you. You can create a support request on the Cypress Technical Support page.

If you are in the United States, you can talk to our technical support team by calling our toll-free number: +1-800-541-4736. Select option 8 at the prompt.

You can also use the following support resources if you need quick assistance.

- Self-help
- Local Sales Office Locations

4 Code Examples

PSoC Creator includes a large number of code example projects. These projects are available from the PSoC Creator Start Page, as Figure 2 shows.

Example projects can speed up your design process by starting you off with a complete design, instead of a blank page. The example projects also show how PSoC Creator Components are used in various applications. Code examples and datasheets are included, as Figure 3 shows.

In the Find Example Project dialog shown in Figure 3, you have several options:

- Filter for examples based on architecture or device family, i.e., PSoC 3, PSoC 4 or PSoC 5LP; category; or keyword
- Select from the menu of examples offered based on the Filter Options
- Review the datasheet for the selection (on the **Documentation** tab)
- Review the code example for the selection. You can copy and paste code from this window to your project, which can help speed up code development, or
- Create a new project (and a new workspace if needed) based on the selection. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 2. Code Examples in PSoC Creator

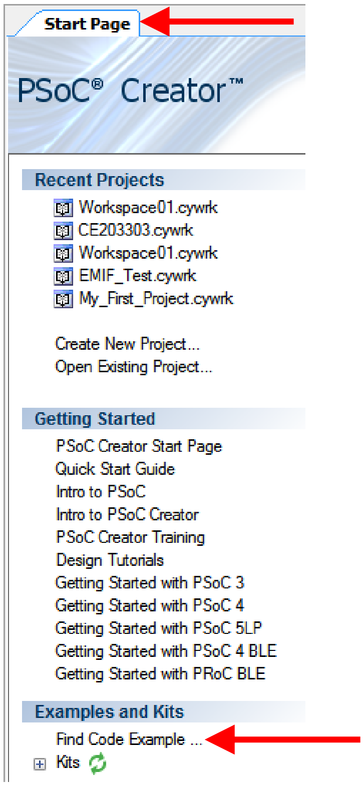
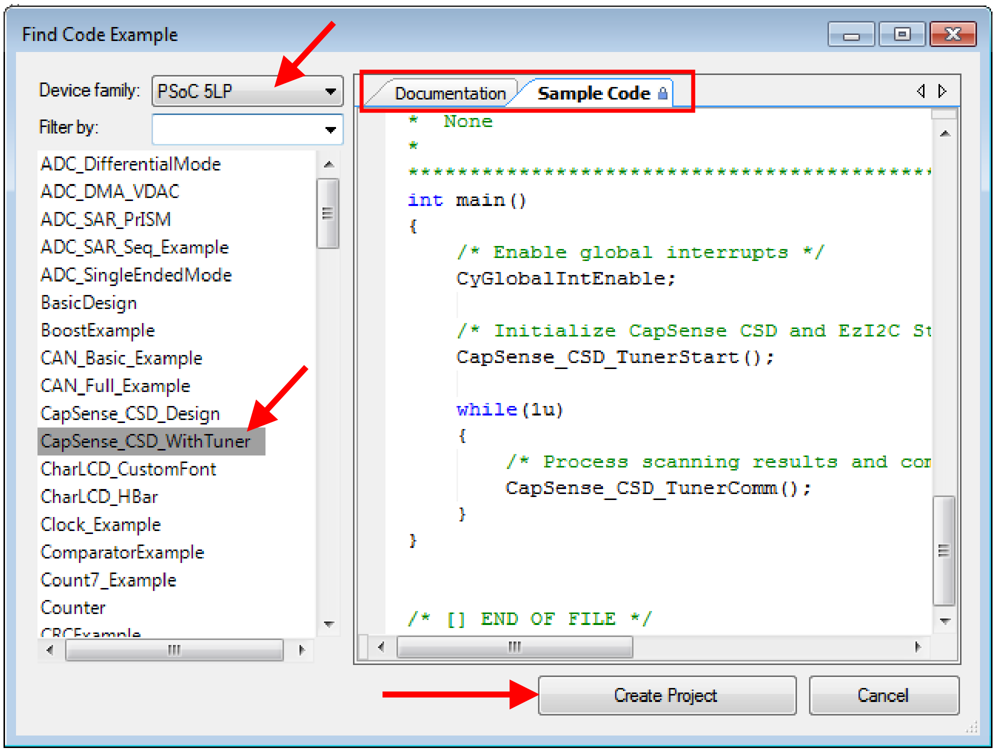


Figure 3. Code Example Projects, with Sample Code



Device family: PSoC 5LP

Filter by:

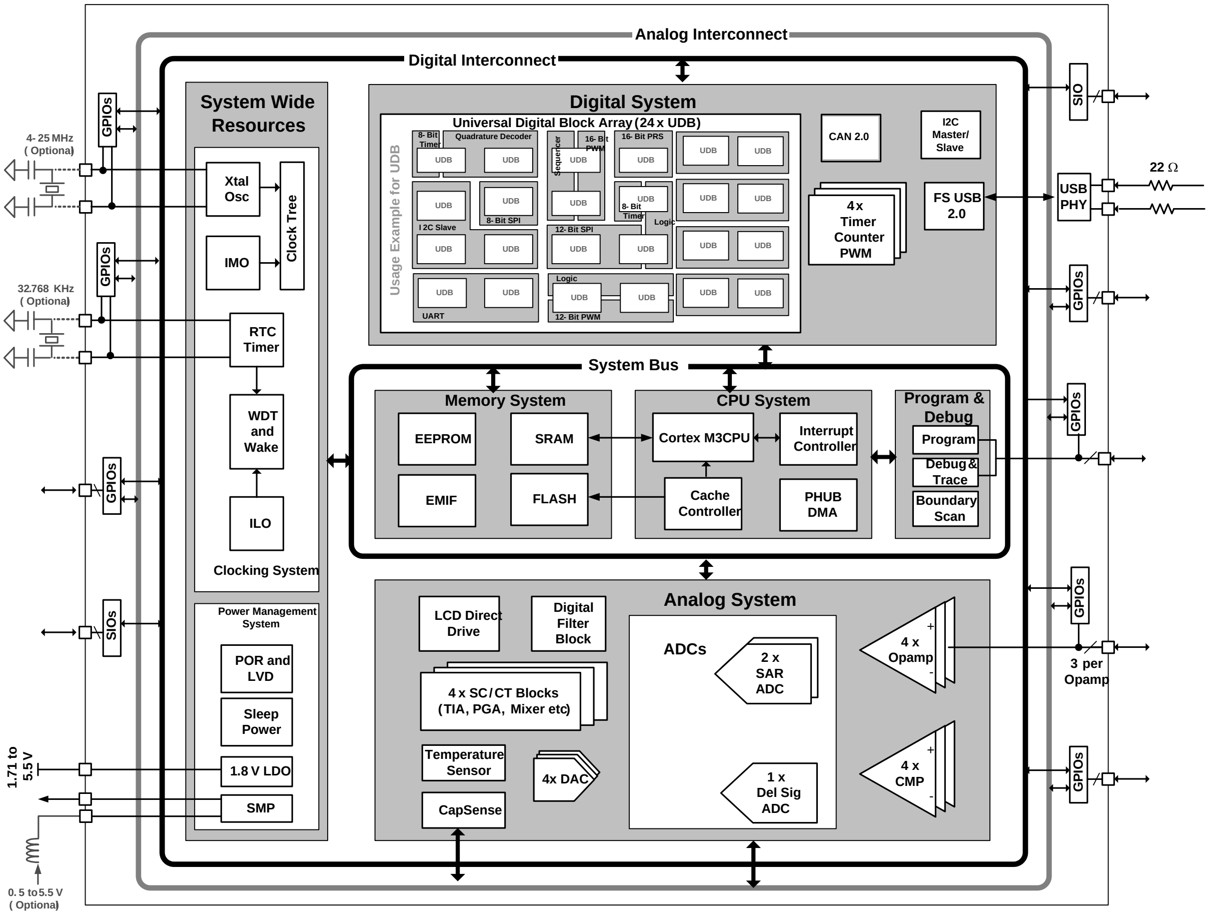
```
*   None
*
*****************************************************
int main()
{
    /* Enable global interrupts */
    CyGlobalIntEnable;

    /* Initialize CapSense CSD and EzI2C St
    CapSense_CSD_TunerStart();

    while(1u)
    {
        /* Process scanning results and cor
        CapSense_CSD_TunerComm();
    }
}


/* [] END OF FILE */
```

5 PSoC 5LP Feature Set

PSoC 5LP has an extensive set of features, which include a CPU and memory subsystem, a digital subsystem, an analog subsystem, and system resources, as Figure 4 shows (for the CY8C58xxLP device family).

For more information, see the PSoC 5LP family device datasheets, technical reference manuals (TRMs), and application notes listed previously.

Figure 4. PSoC 5LP Architecture (CY8C58xxLP)



Following is a list of major features of the PSoC 5LP. For details, see Related Documents, or see a PSoC 5LP device datasheet.

- Performance
  - DC to 80-MHz operation
  - 32-bit ARM Cortex-M3 CPU, 32 interrupts
  - 24-channel direct memory access (DMA) controller
  - 24-bit 64-tap digital filter processor (DFB)
- Memories
  - Up to 256 KB program flash
  - Up to 32 KB additional flash for error correcting code (ECC)
  - Up to 64 KB of SRAM
  - 2 KB EEPROM

- Digital peripherals
  - Four 16-bit timer, counter, and PWM (TCPWM)
  - I²C, 1-Mbps bus speed
  - USB 2.0-certified Full-Speed (FS) 12 Mbps
  - Full CAN 2.0b, 16 Rx, 8 Tx buffers
  - 20 to 24 universal digital blocks (UDB), programmable to create any number of functions:
    - 8-, 16-, 24-, and 32-bit timers, counters, and PWMs
    - I²C, UART, SPI, I2S, and LIN 2.0 interfaces
    - Cyclic redundancy check (CRC)
    - Pseudo random sequence (PRS) generators
    - Quadrature decoders
    - Gate-level logic functions
- Analog Subsystem
  - Configurable 8- to 20-bit delta-sigma ADC
  - Up to two 12-bit SAR ADCs
  - Four 8-bit DACs
  - Four comparators
  - Four operational amplifiers (opamps)
  - Four programmable analog blocks, to create:
    - Programmable gain amplifier (PGA)
    - Transimpedance amplifier (TIA)
    - Mixer
    - Sample and hold (S/H) circuit
  - CapSense® support, up to 62 sensors
  - 1.024 V ±0.1% internal voltage reference
- Versatile I/O system
  - 46 to 72 I/O pins; up to 62 general-purpose I/Os (GPIOs)
  - Up to eight performance I/O (SIO) pins
    - 25 mA current sink
    - Programmable input threshold and output high voltages
    - Can act as a general-purpose comparator
    - Hot swap capability and overvoltage tolerance
  - Two USBIO pins that can be used as GPIOs
  - Route any digital or analog peripheral to any GPIO
  - LCD direct drive from any GPIO, up to 46 × 16 segments
  - CapSense support from any GPIO
  - 1.2-V to 5.5-V interface voltages, up to four power domains
- Programmable clocking
  - 3- to 74-MHz internal oscillator, 1% accuracy at 3 MHz
  - 4- to 25-MHz external crystal oscillator
  - Internal PLL clock generation up to 80 MHz
  - Low-power internal oscillator at 1, 33, and 100 kHz
  - 32.768-kHz external watch crystal oscillator
  - 12 clock dividers routable to any peripheral or I/O

Refer to the datasheet for a full review of PSoC 5LP features.

5.1 PSoC is More Than an MCU

Figure 5 shows that a typical MCU contains a CPU and a set of peripheral functions such as ADC, DAC, UART, SPI, and general I/O, all linked to the CPU's register interface. Within the MCU, the CPU is the "heart" of the device – the CPU manages everything from setup to data movement to timing. Without the CPU the MCU cannot function.

Figure 6 shows that PSoC is quite different. The CPU, analog, digital, and I/O are equally important resources in a programmable system. *It is the system's interconnect and programmability that is the heart of PSoC – not the CPU.* The analog and digital peripherals are interconnected with a highly configurable routing matrix, which allows you to create custom designs to precisely meet your application requirements. *You can program PSoC to emulate an MCU, but you cannot program an MCU to emulate PSoC.*

Figure 5. Block Diagram of a Typical MCU

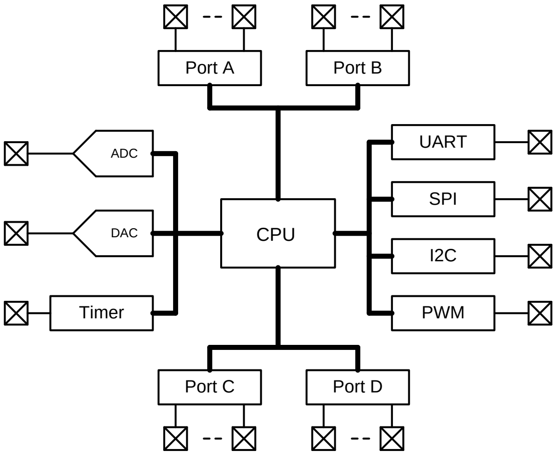
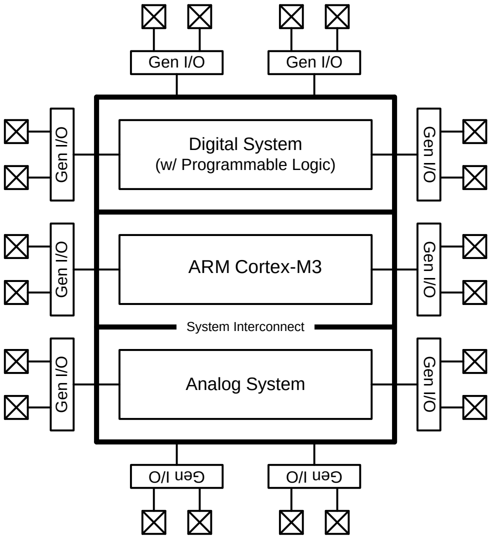


Figure 6. PSoC 5LP Block Diagram



A typical MCU requires CPU firmware to process state machines, use a timer for timing, and drive an output pin. Thus the functional path is almost always through the CPU. However with PSoC asynchronous parallel processing is possible. You can configure a PSoC to have elements that operate independently from the CPU.

For example, Figure 6 shows that PSoC 5LP has no UART. However, you can make as many UARTs as you need within the configurable digital logic, using the predesigned and pretested UART Component in PSoC Creator. You can configure each UART to have as few or as many features as you need.

5.2 The Concept of PSoC Creator Components

The key to successful PSoC designs is the PSoC Creator IDE. PSoC Creator encapsulates PSoC peripherals and other resources as graphical elements called Components. Components are dragged and dropped onto a schematic, and wired together, making the design process fast and easy. Design changes can be quickly made with just a few mouse clicks.

For example, in a traditional MCU, to blink an LED using a PWM peripheral you must:

1. Locate the registers corresponding to the PWM.
2. Calculate the values to be written to the PWM registers, based on the required PWM period and duty cycle.
3. Write many lines of code to configure the PWM registers, set the pin drive mode and to connect the PWM output to the pin.

To implement the same functionality in PSoC is a trivial exercise, as you will find in the next section.

Pin Component: Connect Any Function to Any Pin

PSoC 5LP includes an extensive routing fabric that allows you to route almost any function – digital or analog – to any pin. PSoC Creator makes this easy to do by providing a Pin Component, which with just a few mouse clicks you can configure, connect to a PSoC resource, and associate with a physical pin. You can also easily change Pin Component connections, which lets you rapidly handle board-level design changes.

Components Based on Programmable Digital Resources

PSoC 5LP has programmable digital blocks called Universal Digital Blocks (UDBs). PSoC Creator provides a number of Components made from the UDBs. These include UART, SPI, I^2C, I2S, Timer, PWM, Counter, CRC, quadrature decoder, digital gates (AND, OR, NOT, XOR, etc.), and many more. You can even create your own custom state machines and digital logic.

Components Based on Programmable Analog Resources

PSoC 5LP also has programmable analog blocks called switched capacitor continuous time (SC/CT) blocks. PSoC Creator provides analog Components, such as programmable gain amplifier (PGA) and transimpedance amplifier (TIA), that are made from the SC/CT blocks.

6 My First PSoC 5LP Design

This section does the following:

- Demonstrates how PSoC can be programmed to do more than a traditional MCU
- Shows how to build a simple PSoC design and install it in a development kit
- Provides detailed steps that make it easy to learn PSoC design techniques using the PSoC Creator IDE

6.1 Before You Begin

Have You Installed PSoC Creator?

Download and install PSoC Creator from the PSoC Creator home page. Note that the installation may take a long time – see the PSoC Creator Release Notes for more information.

Do You Have a Development Kit?

Table 1 lists all Cypress development kits for the PSoC 5LP. Kits are also available from other manufacturers.

Table 1. Cypress PSoC 5LP Kits

PSoC 5LP Kit	PSoC 5LP Device Part Number	Programming
CY8CKIT-059	CY8C5888LTI-LP097	Integrated programmer
CY8CKIT-050	CY8C5868AXI-LP035	Integrated programmer
CY8CKIT-001	CY8C5868AXI-LP035	MiniProg3 program and debug kit

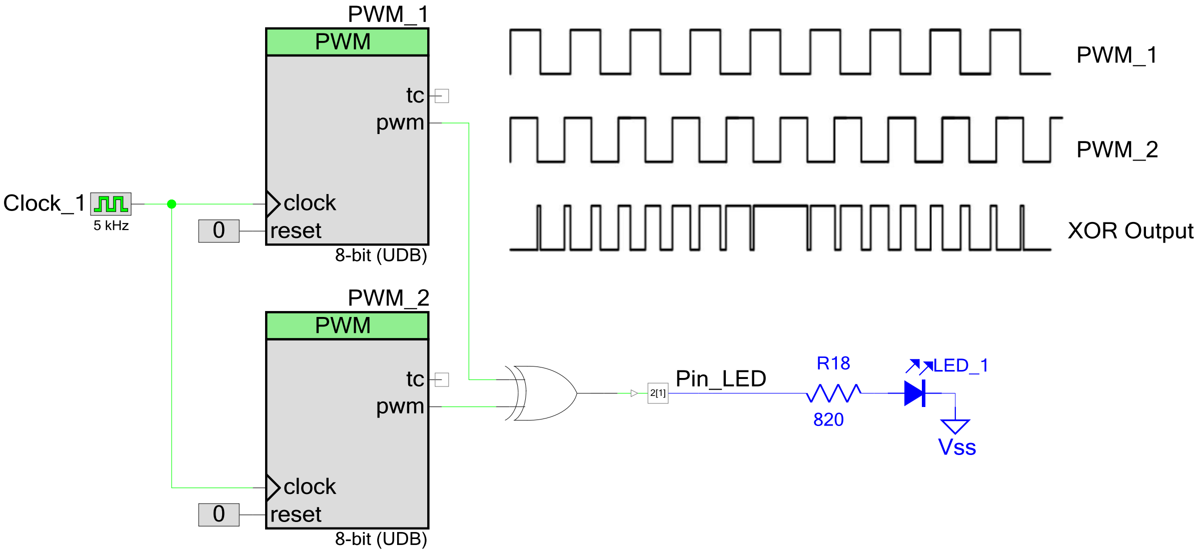
Want To See the Project In Action?

If you don't want to go through the development process shown in the next section, you can get the completed code example project at CE203303. You can then jump to the Build and Program steps. The code example is designed for the CY8CKIT-059; you can easily modify it for other kits.

6.2 About The Design

This design is described in detail in code example CE203303, PSoC 3 and PSoC 5LP Breathing LED. It implements a “breathing LED” effect exclusively in hardware, with no CPU usage beyond initialization. Figure 7 shows the PSoC Creator schematic.

Figure 7. Breathing LED Schematic (Pin and LED are selected for CY8CKIT-059)



6.3 Part 1: Create the Design

This section takes you through the design process, step by step. It guides you through both hardware and firmware design entry.

Note: These instructions assume that you are using PSoC Creator 3.3. The overall development process is the same for other versions of PSoC Creator, however some of the dialog boxes may be different.

1. Create a new PSoC Creator project.

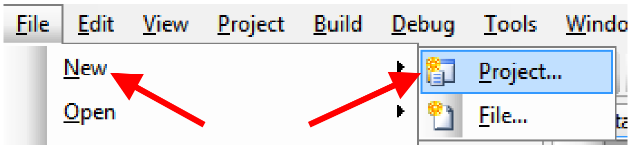
   A project contains all of the source code and other files required to create a single output module that can be downloaded to a target PSoC 5LP device.

   A. Start PSoC Creator.

   B. Select menu item
   File > **New** > **Project…**
   as Figure 8 shows.

   A Create Project window is displayed.

Figure 8. Create a New PSoC Creator Project



2. Select the project type for the CY8CKIT-059 kit. See Figure 9.

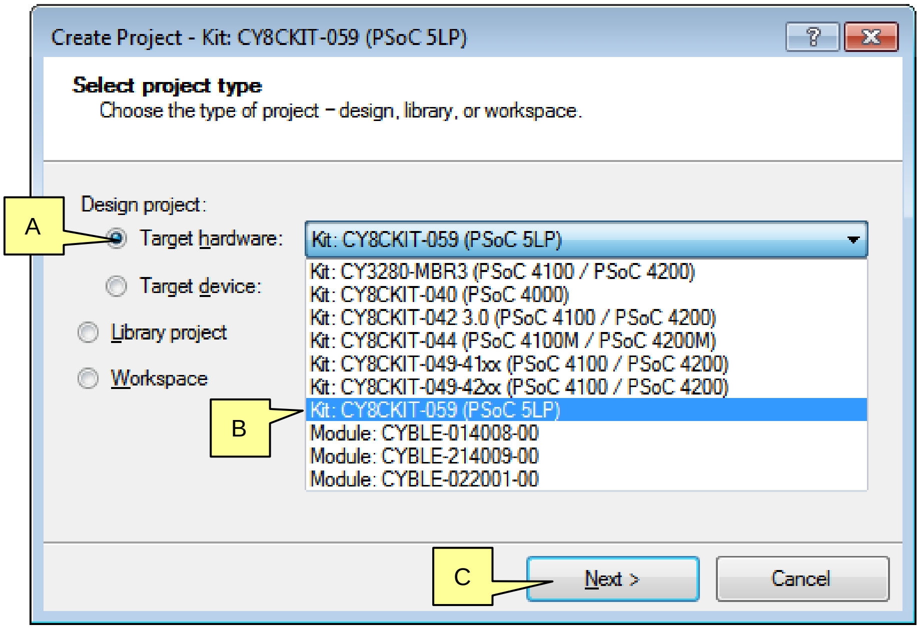
   PSoC Creator can speed up the development process by automatically setting various project options for specified development kits or target devices.

   A. Click **Target hardware**.

   B. In the pulldown menu, select **Kit: CY8CKIT-059 (PSoC 5LP)**.

   C. Click **Next**.

Figure 9. Create a New Project for the CY8CKIT-059



Create Project - Kit: CY8CKIT-059 (PSoC 5LP)

Select project type

Choose the type of project – design, library, or workspace.

Design project:

● Target hardware: Kit: CY8CKIT-059 (PSoC 5LP)

○ Target device:

○ Library project

○ Workspace

Kit: CY3280-MBR3 (PSoC 4100 / PSoC 4200)
Kit: CY8CKIT-040 (PSoC 4000)
Kit: CY8CKIT-042 3.0 (PSoC 4100 / PSoC 4200)
Kit: CY8CKIT-044 (PSoC 4100M / PSoC 4200M)
Kit: CY8CKIT-049-41xx (PSoC 4100 / PSoC 4200)
Kit: CY8CKIT-049-42xx (PSoC 4100 / PSoC 4200)
Kit: CY8CKIT-059 (PSoC 5LP)
Module: CYBLE-014008-00
Module: CYBLE-214009-00
Module: CYBLE-022001-00
Next >
Cancel
A
B
C

3. Select an empty schematic as a project template. See Figure 10.

   PSoC Creator can speed up the development process by basing a new design on an existing code example. For this exercise, we will start from an empty schematic.

   A. Click **Empty Schematic**.

   B. Click **Next**.

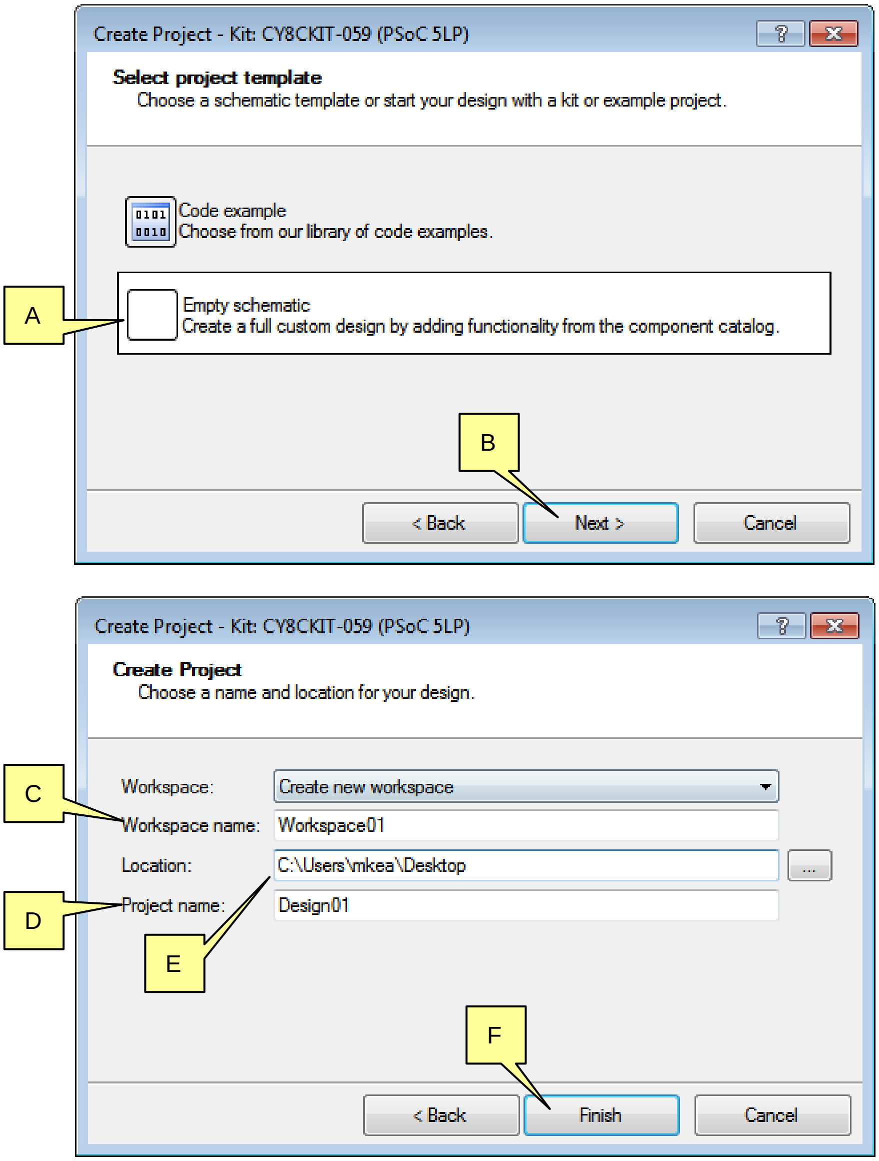
   C. In the next dialog, enter text for a **Workspace name**. A workspace is a container for one or more projects. A project is usually contained in a workspace.

   D. Enter text for a **Project name**. The project and workspace names can be the same or different.

   E. Specify the **Location** of your workspace and project.

   F. Click **Finish**.

Figure 10. Create a New Empty Project



Create Project - Kit: CY8CKIT-059 (PSoC 5LP)

Select project template

Choose a schematic template or start your design with a kit or example project.

Code example
Choose from our library of code examples.

Empty schematic
Create a full custom design by adding functionality from the component catalog.

Create Project - Kit: CY8CKIT-059 (PSoC 5LP)

Create Project

Choose a name and location for your design.

Workspace: Create new workspace
Workspace name: Workspace01
Location: C:\Users\mkea\Desktop
Project name: Design01

A project is created. Several new panes are displayed: **Workspace Explorer**, **Schematic** (*TopDesign.cysch*), and **Component Catalog**.

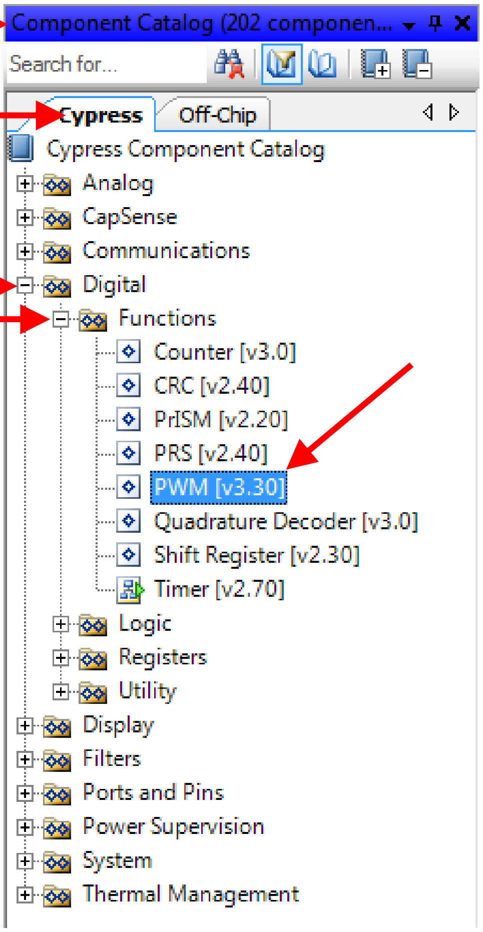
4. Build the hardware portion of the design.

   In this step, you drag Components from the Component Catalog onto the schematic. You then configure each Component, and wire them together.

   a. In the **Component Catalog** window, **Cypress** tab, find the **PWM** Component, as Figure 11 shows.

   b. Drag two instances of the PWM Component onto the schematic (see Figure 7).

Figure 11. Select PWM Component



Component Catalog (202 componen...

Search for...

Cypress | Off-Chip

- Cypress Component Catalog
  - Analog
  - CapSense
  - Communications
  - Digital
    - Functions
      - Counter [v3.0]
      - CRC [v2.40]
      - PrISM [v2.20]
      - PRS [v2.40]
      - PWM [v3.30]
      - Quadrature Decoder [v3.0]
      - Shift Register [v2.30]
      - Timer [v2.70]
    - Logic
    - Registers
    - Utility
  - Display
  - Filters
  - Ports and Pins
  - Power Supervision
  - System
  - Thermal Management

5. Configure the PWM Components, as Figure 12 shows.

   This creates square wave outputs from both PWMs; the square waves have slightly different frequencies. The difference in frequencies results in a beat frequency that is modulated on the LED.

   On the schematic, double-click each PWM Component to configure it.

   A. For PWM_1, change the **PWM Mode** to **One Output**.

   B. No other changes need be made to PWM_1. Click **OK** to close the dialog.

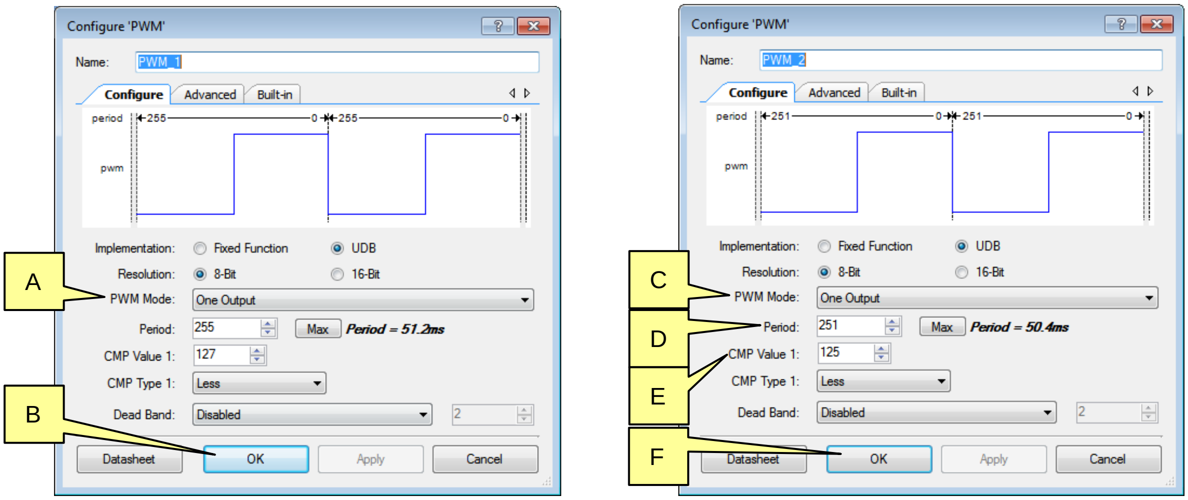
   C. For PWM_2, change the **PWM Mode** to **One Output**.

   D. Set the **Period** value of PWM_2 to be slightly different from the default.

   E. Set the **CMP Value 1** of PWM_2 to approximately half the period.

   F. Changes for PWM_2 are complete. Click **OK** to close the dialog.

Figure 12. Configure the PWM Components



Configure 'PWM'

Name: PWM_1

Implementation: ○ Fixed Function ● UDB

Resolution: ● 8-Bit ○ 16-Bit

PWM Mode: One Output

Period: 255 *Period = 51.2ms*

CMP Value 1: 127

CMP Type 1: Less

Dead Band: Disabled 2

Configure 'PWM'

Name: PWM_2

Implementation: ○ Fixed Function ● UDB

Resolution: ● 8-Bit ○ 16-Bit

PWM Mode: One Output

Period: 251 *Period = 50.4ms*

CMP Value 1: 125

CMP Type 1: Less

Dead Band: Disabled 2

6. Drag from the Component Catalog to the schematic, and configure, the additional Components listed in Table 2.

   The **Off-Chip** Components are not required, but help to show the overall purpose of the design.

   Note that in each configuration dialog the **Name** field is automatically populated; you can change the name to any valid text. Each Component name must be unique in the schematic.

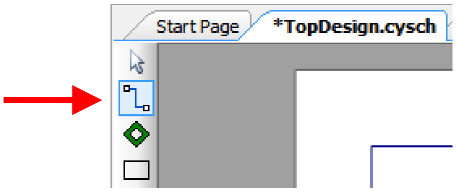
Table 2. Design Components

Component	Component Catalog		Change from Default Configuration
	Tab	**Group**	
Logic Low	Cypress	Digital > Logic	none
Xor	Cypress	Digital > Logic	none
Clock	Cypress	System	Set **Frequency** to 5 kHz
Digital Output Pin	Cypress	Ports and Pins	Check the **External terminal** box
Resistor	Off-Chip	Passive	none
LED	Off-Chip	Diodes	none
Ground	Off-Chip	Power	none

7. Select the wire tool (Figure 13) to connect the logic (or press 'w' as a shortcut).

   Wire the Components as Figure 7 shows.

Figure 13. Select the Wire Tool



8. At this point, the hardware design is complete, however the Pin Component must still be associated with a physical pin.

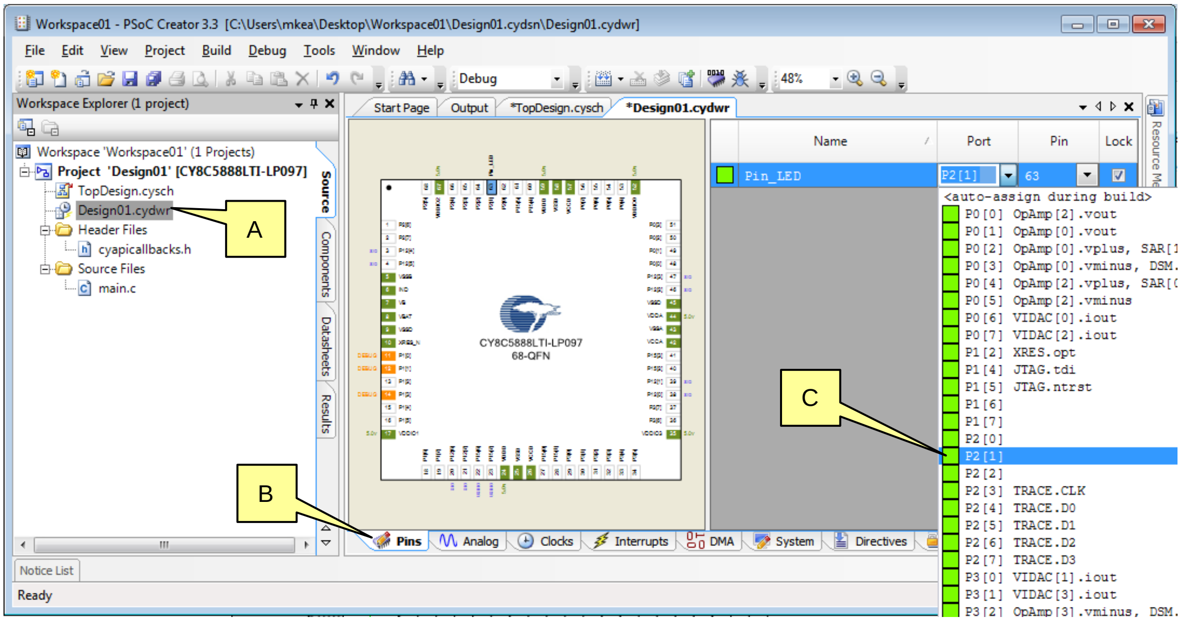
   Choose the physical pin for the LED on the development kit that you are using. (For the CY8CKIT-059, the pin used is port 2, pin 1, pr P2[1].)

   A. In the Workspace Explorer window, double-click the *.cydwr* file in your project, as Figure 14 shows. This opens the design-wide resources (DWR) window.

   B. Select the **Pins** tab. The Pin Components defined in the project are displayed, as well as a pin diagram of the target device.

   C. Associate the schematic Pin Component with the desired physical pin.

Figure 14. Associate the Pins



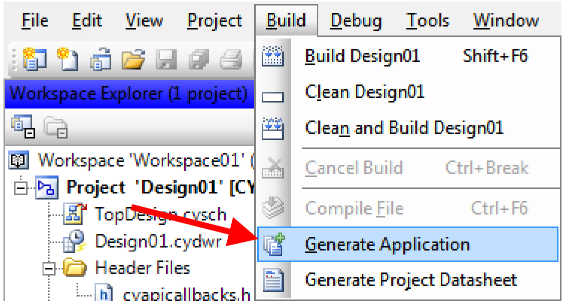
Name	Port	Pin	Lock
Pin_LED	P2[1]	63	☑

```
<auto-assign during build>
P0[0]  OpAmp[2].vout
P0[1]  OpAmp[0].vout
P0[2]  OpAmp[0].vplus, SAR[1
P0[3]  OpAmp[0].vminus, DSM.
P0[4]  OpAmp[2].vplus, SAR[(
P0[5]  OpAmp[2].vminus
P0[6]  VIDAC[0].iout
P0[7]  VIDAC[2].iout
P1[2]  XRES.opt
P1[4]  JTAG.tdi
P1[5]  JTAG.ntrst
P1[6]
P1[7]
P2[0]
P2[1]
P2[2]
P2[3]  TRACE.CLK
P2[4]  TRACE.D0
P2[5]  TRACE.D1
P2[6]  TRACE.D2
P2[7]  TRACE.D3
P3[0]  VIDAC[1].iout
P3[1]  VIDAC[3].iout
P3[2]  OpAmp[3].vminus, DSM.
```

9. You must now write a couple of lines of firmware. Before doing so, it is best to have PSoC Creator generate all of the code that is associated with the Components.

   Select the PSoC Creator menu item **Build** > **Generate Application**, as Figure 15 shows. If there are no errors, PSoC Creator generates several code files, under the folder *Generated_Source*.

Figure 15. Generate Application



10. Add code to the auto-generated file *main.c*. It has a framework for adding code; the code that you must add, to start the two PWM Components, is highlighted, as Code 1 shows. In the Workspace Explorer window, double-click the *main.c* file in your project to open it.

    Note: This code assumes that the PWM Components have the default names. If you renamed your PWM Components to something other than the default values, use those names in the _Start() function calls.

Code 1. Main Code for Breathing LED

```
#include <project.h>

int main()
{
    //CyGlobalIntEnable; /* Enable global interrupts. */

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    PWM_1_Start();
    PWM_2_Start();

    for(;;)
    {
        /* Place your application code here. */
    }
}
```

11. If you skipped to this step without going through the design process, do the following:

    A. Download the code example file *CE203303.zip* from CE203303, and extract it to a convenient location in your computer.

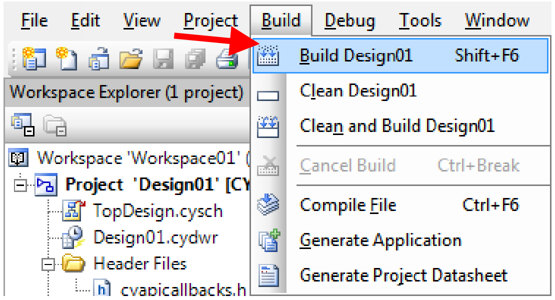
    B. Download and install PSoC Creator as described in step 1 on page 9.

    C. Open the file *CE203303.cywrk* in PSoC Creator.

    D. Confirm that the project pin assignments match your development kit (DVK), as described in step 8 on page 14.

    E. Select the PSoC Creator menu item **Build** > **Build <project name>**, as Figure 16 shows. If there are no errors, the project is built and ready to program to the target DVK.

Figure 16. Build Project



6.4 Part 2: Program the Device

The programming process is the same for all the development kit boards. To set up your DVK, follow the instructions in the Kit Guide document.

1. Confirm the connection between PSoC Creator and your DVK.

   Select the PSoC Creator menu item **Debug** > **Select Debug Target**., as Figure 17 shows.

   A. A "Select Debug Target" dialog is displayed, as Figure 18 shows. Click on your target DVK (PSoC Creator supports multiple DVK connections).

   B. Click **Port Acquire**.

Figure 17. Select Debug Target

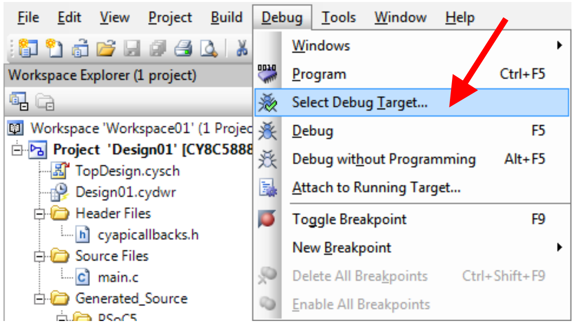
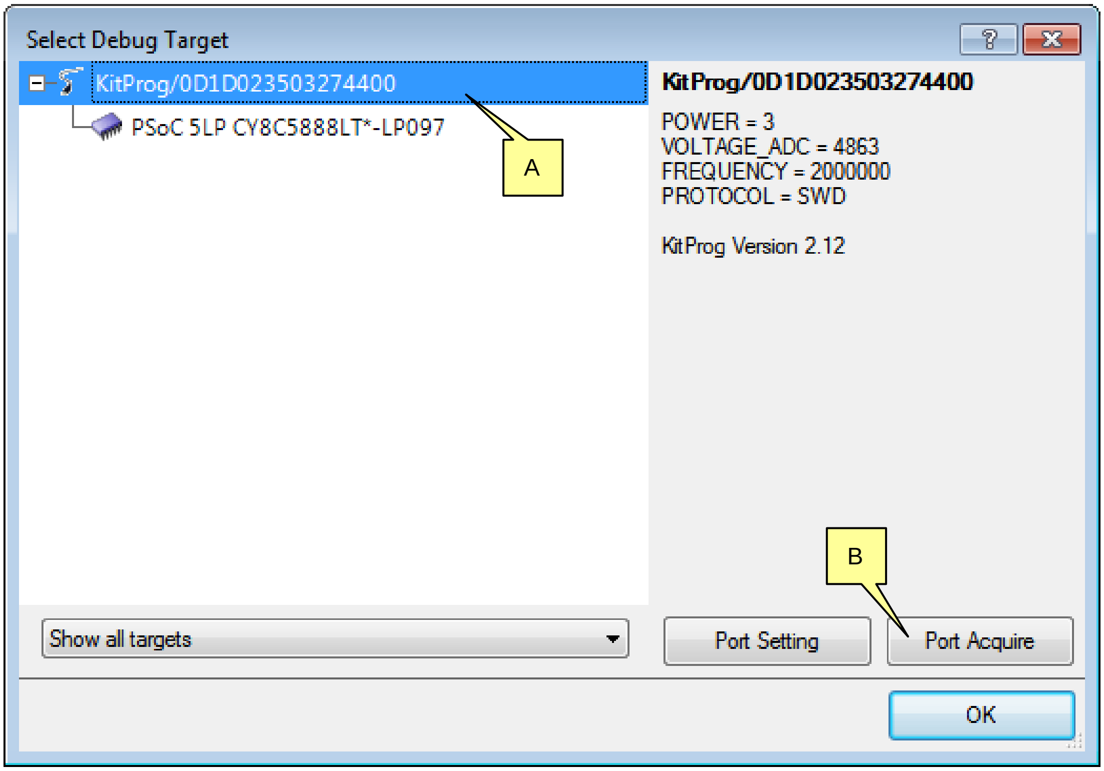


Figure 18. Select and Acquire the Target for Programming



KitProg/0D1D023503274400

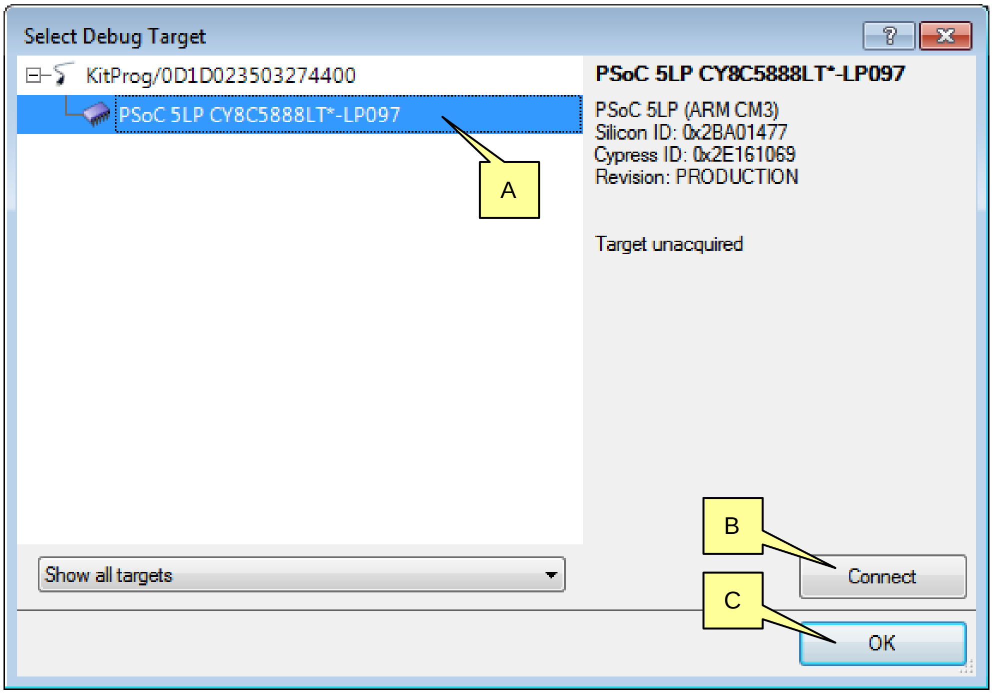
POWER = 3
VOLTAGE_ADC = 4863
FREQUENCY = 2000000
PROTOCOL = SWD

KitProg Version 2.12

2. Connect to the PSoC on your target DVK. See Figure 19.
   A. Click the PSoC 5LP.
   B. Click **Connect**. The “Target unacquired” message changes to “Target acquired”, and the button label changes to “Disconnect”.
   C. Click **OK** to close the dialog.

   PSoC Creator is now connected to the target DVK and PSoC, and you can now program the PSoC.

Figure 19. Connect to the Target PSoC 5LP



Select Debug Target

KitProg/0D1D023503274400

PSoC 5LP CY8C5888LT*-LP097

PSoC 5LP CY8C5888LT*-LP097

PSoC 5LP (ARM CM3)
Silicon ID: 0x2BA01477
Cypress ID: 0x2E161069
Revision: PRODUCTION

Target unacquired

3. To program the PSoC 5LP, select the PSoC Creator menu item **Debug** > **Program**, as Figure 20 shows.
4. Programming begins; programming status is displayed in the PSoC Creator status bar (the lower-left corner of the window, as Figure 21 shows).

   Note: You may see a warning message “This programmer is currently out of date”. Refer to the KitProg User Guide in your kit documentation for information on how to upgrade your programmer firmware.

Figure 20. Program Device

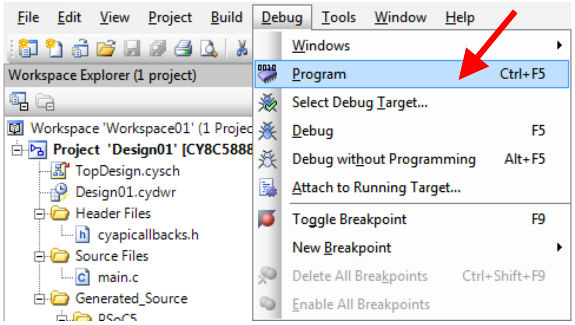
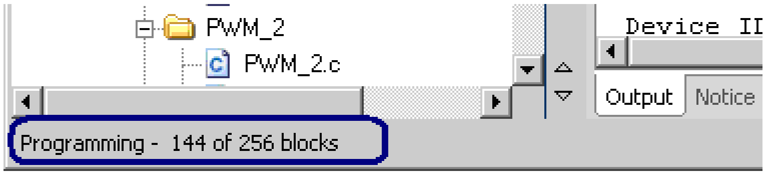


Figure 21. Programming Status



Programming - 144 of 256 blocks

On the CY8CKIT-059 DVK, the blue LED gradually transitions from full ON to full OFF over a few seconds.

7 Summary

This application note explored the PSoC 5LP architecture and development tools. The most important concept to be gained from this application note is that PSoC is more than an MCU. PSoC 5LP is a truly programmable embedded system-on-chip, integrating configurable analog and digital peripheral functions, memory, and a 32-bit Cortex-M3 CPU on a single chip.

Because of the integrated features and low-leakage power modes, PSoC 5LP is an ideal choice for low-power and cost- effective embedded systems.

8 Related Documents

Table 3 lists system-level and general application notes that are recommended for the next steps in learning about PSoC and PSoC Creator:

Table 3. General and System-Level Application Notes

Document	Document Name
AN61290, AN88619	PSoC® 3 and PSoC 5LP Hardware Design Considerations, PSoC® 4 Hardware Design Considerations
AN81623	PSoC® 3, PSoC 4, and PSoC 5LP Digital Design Best Practices
AN77900, AN86233, AN90114	PSoC® 3 and PSoC 5LP Low-power Modes and Power Reduction Techniques, PSoC® 4 Low-power Modes and Power Reduction Techniques, PSoC® 4000 Low-power Modes and Power Reduction Techniques
AN68403	PSoC® 3 and PSoC 5LP Analog Signal Chain Calibration
AN57821	PSoC® 3, PSoC 4, and PSoC 5LP Mixed-Signal Circuit Board Layout Considerations
AN58827	PSoC® 3 and PSoC 5LP Internal Analog Routing Considerations
AN73854	PSoC® 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders
AN60616	PSoC® 3 and PSoC 5LP Startup Procedure
AN60631	PSoC® 3 and PSoC 5LP Clocking Resources
AN77835	PSoC® 3 to PSoC 5LP Migration Guide
AN78175, AN89056	PSoC® 3 and PSoC 5LP IEC60730 Class B Safety Software Library, PSoC® 4 IEC60730 Class B Safety Software Library

Table 4 lists application notes (AN), code examples (CE), and knowledge base articles (KBA) that are linked to the device description in PSoC 5LP Feature Set.

Table 4. Documents Related to PSoC 5LP Features

Document	Document Name
CPU and Interrupts	
AN89610	PSoC® 4 and PSoC 5LP ARM Cortex Code Optimization
AN54460	PSoC® 3 and PSoC 5LP Interrupts
Memory	
CE95313	PSoC® 3, PSoC 4, and PSoC 5LP Emulated EEPROM Memory
Direct Memory Access (DMA)	
AN52705	PSoC® 3 and PSoC 5LP – Getting Started with DMA
AN84810	PSoC® 3 and PSoC 5LP Advanced DMA Topics
AN61102	PSoC® 3 and PSoC 5LP – ADC Data Buffering Using DMA

Document	Document Name
CE95375 CE95376	SPI Master and DMA with PSoC® 3 and PSoC 5LP SPI Slave and DMA with PSoC® 3 and PSoC 5LP
Digital Filter Block (DFB)	
CE95316	Filter From to ADC to VDAC Using DFB with PSoC® 3 and PSoC 5LP
CE95317	Filter From to ADC to VDAC Using DFB in Polling Mode with PSoC® 3 and PSoC 5LP
I2C	
CE95324	I^{2}C LCD with PSoC® 3 and PSoC 5LP
CE95314	PSoC® 3, PSoC 4, and PSoC 5LP EZI2C
USB	
AN57294	USB 101: An Introduction to Universal Serial Bus 2.0
AN57473	USB HID Basics with PSoC® 3 and PSoC 5LP
AN58726	USB HID Intermediate with PSoC® 3 and PSoC 5LP
AN56377	PSoC® 3 and PSoC 5LP – Introduction to Implementing USB Data Transfers
AN82072	PSoC® 3 and PSoC 5LP USB General Data Transfer with Standard HID Drivers
AN73503	USB HID Bootloader for PSoC® 3 and PSoC 5LP
CE95390	USB Audio with PSoC® 3 and PSoC 5LP
CE95395	USB MIDI with PSoC® 3 and PSoC 5LP
CE95394	USB HID Mouse with PSoC® 3 and PSoC 5LP
CE95393	USB Bulk Transfer with PSoC® 3 and PSoC 5LP
CE95392	USB Bootloader with PSoC® 3 and PSoC 5LP
CE95396	USB UART with PSoC® 3 and PSoC 5LP
Controller Area Network (CAN)	
AN52701	PSoC® 3 and PSoC 5LP – Getting Started with Controller Area Network (CAN)
CE95282	CAN as Control Node with PSoC® 3 and PSoC 5LP
CE95283	CAN as Remove Node with PSoC® 3 and PSoC 5LP
KBA86565	Difference Between Full CAN and Basic CAN Mailbox
KBA86566	Acceptance Filter Implementation for CAN Receive Message
KBA86567	Modifying the Full CAN Mailbox's Identifier in the Program
Universal Digital Blocks (UDB)	
AN82250	PSoC® 3, PSoC 4, and PSoC 5LP – Implementing Programmable Logic Designs with Verilog
AN82156	PSoC® 3, PSoC 4, and PSoC 5LP – Designing PSoC Creator™ Components with UDB Datapaths
CE95295	8-Bit UDB Counter with PSoC® 3 and PSoC 5LP
CE95384	16-Bit UDB-Based Timer with PSoC® 3 and PSoC 5LP
CE95323	Hardware Fan Control with PSoC® 3 and PSoC 5LP
KBA85325	Comparison of Resource Utilization Between PSoC® 3 and PSoC 5LP UDBs and Other Vendor CPLDs
KBA86336	Just Enough Verilog for PSoC®
Analog to Digital Converter (ADC)	
AN84783	Accurate Measurement Using PSoC® 3 and PSoC 5LP Delta-Sigma ADC

Document	Document Name
CE95277	Delta-Sigma ADC in Single-Ended Mode with PSoC® 3 and PSoC 5LP
CE95271	Delta-Sigma ADC in Differential Mode with PSoC® 3 and PSoC 5LP
CE95276	Sequencing SAR ADC with PSoC® 3 and PSoC 5LP
KBA81866	Best Method of Amplification to Get Better Performance from PSoC® 3 and PSoC 5LP Delta-Sigma ADC
KBA84753	Choice of Reference Voltage for Accurate ADC Measurements in PSoC® 3, PSoC 4 and PSoC 5LP
Digital to Analog Converter (DAC)	
AN60305	Using PSoC® 3 and PSoC 5LP IDACs to build a better VDAC
AN64275	PSoC® 3 and PSoC 5LP: Getting More Resolution from 8-Bit DACs
AN69133	PSoC® 3 and PSoC 5LP Easy Waveform Generation with the WaveDAC8 Component
CE95397	Voltage DAC with PSoC® 3 and PSoC 5LP
CE95309	Dithered Voltage DAC with PSoC® 3 and PSoC 5LP
KBA84732	VDAC8 Output Voltage in PSoC® 3 and PSoC 5LP
KBA83238	Driving an External Load using VDAC in PSoC® 3 or PSoC 5LP
Comparator	
AN60220	PSoC® 3 and PSoC 5LP Multiplexed Comparator
CE95292	Analog Voltage Comparator with PSoC® 3 and PSoC 5LP
CE95361	Scanning Comparator Using Internal VDAC with PSoC® 3 and PSoC 5LP
CE95360	Scanning Comparator Using Common Mode with PSoC® 3, PSoC 4, and PSoC 5LP
Operational Amplifier (Opamp)	
CE95339	Operational Amplifier (Opamp) with PSoC® 3 and PSoC 5LP
Programmable Analog Block (SC/CT)	
AN60321	Peak Detection with PSoC® 3 and PSoC 5LP
AN62582	AM Modulation and Demodulation
CE95342	Programmable Gain Amplifier (PGA) with PSoC® 3 and PSoC 5LP
CE95343	Inverting Programmable Gain Amplifier with PSoC® 3 and PSoC 5LP
CE95383	Transimpedance Amplifier (TIA) with PSoC® 3 and PSoC 5LP
CE95357	Sample and Hold with PSoC® 3 and PSoC 5LP
CE95337	Analog Signal Mixer with PSoC® 3 and PSoC 5LP
CapSense	
AN75400	PSoC® 3 and PSoC 5LP CapSense® Design Guide
CE95287	CapSense® CSD Using Tuner with PSoC® 3 and PSoC 5LP
CE95284	CapSense® CSD Design with PSoC® 3 and PSoC 5LP
I/O	
AN72382	Using PSoC® 3 and PSoC 5LP GPIO Pins
AN60580	SIO Tips and Tricks in PSoC® 3 and PSoC 5LP
KBA82883	Controlling a PSoC® 3 and PSoC 5LP GPIO in Firmware
KBA91716	Differences Between SIO and GPIO Pins in PSoC® 3 and PSoC 5LP

Document	Document Name
Segment LCD	
AN52927	PSoC® 3 and PSoC 5LP - Segment LCD Direct Drive
CE95368	Segment LCD with PSoC® 5LP

About the Author

Name:	Nidhin MS
Title:	Applications Engineer Sr.
Background:	Nidhin graduated from GEC Thrissur, with a Bachelor's degree in Electronics and Communication Engineering. His technical interests are analog signal processing, low-power design, and capacitive touch sensing.

Document History

Document Title: AN77759 - Getting Started with PSoC® 5LP

Document Number: 001-77759

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3554384	ROSS	03/20/2012	New application note
*A	3819945	RNJT	11/23/2012	Updated for PSoC 5LP
*B	3866015	RNJT	01/11/2013	Fixed PSoC 5LP links
*C	4592410	MKEA	12/10/2014	Added PSoC Resources section. Updated for PSoC Creator 3.0 SP2 Edits and rewrites throughout.
*D	4645196	NIDH	02/17/2015	Updated for PSoC Creator 3.1 Updated template Sunset review
*E	5013158	MKEA	11/25/2015	Deleted attached project; transferred it to code example CE203303. Added references to the code example. Updated for PSoC Creator 3.3 Expanded Related Documents section Miscellaneous minor edits, mainly to better align with AN79953, Getting Started with PSoC 4

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer’s representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

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

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, 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 products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

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

Use may be limited by and subject to the applicable Cypress software license agreement.