Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.
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.

More code examples? We heard you.
To access an ever-growing list of hundreds of PSoC code examples, please visit our code examples web page. You can also explore the Cypress video training library here.

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.................................................................................1
2 PSoC Resources .......................................................................2
3 PSoC Creator ............................................................................2
   3.1 PSoC Creator Help .............................................................3
   3.2 Technical Support ...............................................................3
4 Code Examples .........................................................................4
5 PSoC 5LP Feature Set .............................................................5
   5.1 PSoC is More Than an MCU .............................................7
   5.2 The Concept of PSoC Creator Components .........7
6 My First PSoC 5LP Design ......................................................8
   6.1 Before You Begin ..............................................................8
   6.2 About The Design ............................................................9
6.3 Part 1: Create the Design ....................................................9
6.4 Part 2: Program the Device ...............................................16
7 Summary ...................................................................................16
8 Related Documents ...............................................................18
   Document History ..................................................................18
9 Worldwide Sales and Design Support .................................23
10 Products ...............................................................................23
   PSoC® Solutions ..................................................................23
   Cypress Developer Community ...........................................23
11 Technical Support .................................................................23

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 3:

- **Overview**: PSoC Portfolio, PSoC Roadmap
- **Product Selectors**: PSoC 1, PSoC 3, PSoC 4, PSoC 5LP, or PSoC 6 MCU. In addition, PSoC Creator includes a device selection tool.
- **Datasheets**: Describe and provide electrical specifications for the PSoC 3, PSoC 4, PSoC 5LP, and PSoC 6 MCU device families.
- **CapSense® Design Guides**: Learn how to design capacitive touch-sensing applications with the PSoC 3, PSoC 4, PSoC 5LP, and PSoC 6 MCU 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.

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
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 example projects**: Choose the menu item File > Code Examples. These example projects demonstrate how to configure and use PSoC Creator Components.
- **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 3 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 code examples are available from the PSoC Creator Start Page, as Figure 2 shows.

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

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

- Filter for examples based on architecture or device family, such as, 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.
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.

---

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, I^2S, 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>
<thead>
<tr>
<th>PSoC 5LP Kit</th>
<th>PSoC 5LP Device Part Number</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY8CKIT-059</td>
<td>CY8C5888LTI-LP097</td>
<td>Integrated programmer</td>
</tr>
<tr>
<td>CY8CKIT-050</td>
<td>CY8C5868AXI-LP035</td>
<td>Integrated programmer</td>
</tr>
<tr>
<td>CY8CKIT-001</td>
<td>CY8C5868AXI-LP035</td>
<td>MiniProg3 program and debug kit</td>
</tr>
</tbody>
</table>

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 pull-down menu, select **Kit: CY8CKIT-059 (PSoC 5LP)**.
   C. Click **Next**.

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

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).
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](image)

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](image)
7. Select the wire tool (Figure 13) to connect the logic (or press ‘w’ as a shortcut).

Wire the Components as Figure 7 shows.

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.

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.
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.

```
#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.
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](image1)

![Figure 18. Select and Acquire the Target for Programming](image2)
2. Connect to the PSoC on your target DVK. See Figure 19.
   A. Click 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

---

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.

---

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

<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN61290, AN86819</td>
<td>PSoC® 3 and PSoC 5LP Hardware Design Considerations, PSoC® 4 Hardware Design Considerations</td>
</tr>
<tr>
<td>AN81623</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP Digital Design Best Practices</td>
</tr>
<tr>
<td>AN77900, AN86233, AN90114</td>
<td>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</td>
</tr>
<tr>
<td>AN68403</td>
<td>PSoC® 3 and PSoC 5LP Analog Signal Chain Calibration</td>
</tr>
<tr>
<td>AN57821</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP Mixed-Signal Circuit Board Layout Considerations</td>
</tr>
<tr>
<td>AN58827</td>
<td>PSoC® 3 and PSoC 5LP Internal Analog Routing Considerations</td>
</tr>
<tr>
<td>AN73854</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders</td>
</tr>
<tr>
<td>AN60616</td>
<td>PSoC® 3 and PSoC 5LP Startup Procedure</td>
</tr>
<tr>
<td>AN60631</td>
<td>PSoC® 3 and PSoC 5LP Clocking Resources</td>
</tr>
<tr>
<td>AN77835</td>
<td>PSoC® 3 to PSoC 5LP Migration Guide</td>
</tr>
<tr>
<td>AN78175, AN89056</td>
<td>PSoC® 3 and PSoC 5LP IEC60730 Class B Safety Software Library, PSoC® 4 IEC60730 Class B Safety Software Library</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU and Interrupts</td>
<td></td>
</tr>
<tr>
<td>AN89610</td>
<td>PSoC® 4 and PSoC 5LP Arm Cortex Code Optimization</td>
</tr>
<tr>
<td>AN54460</td>
<td>PSoC® 3 and PSoC 5LP Interrupts</td>
</tr>
<tr>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>CE95313</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP Emulated EEPROM Memory</td>
</tr>
<tr>
<td>Direct Memory Access (DMA)</td>
<td></td>
</tr>
<tr>
<td>AN52705</td>
<td>PSoC® 3 and PSoC 5LP – Getting Started with DMA</td>
</tr>
<tr>
<td>AN84810</td>
<td>PSoC® 3 and PSoC 5LP Advanced DMA Topics</td>
</tr>
<tr>
<td>AN61102</td>
<td>PSoC® 3 and PSoC 5LP – ADC Data Buffering Using DMA</td>
</tr>
<tr>
<td>Document</td>
<td>Document Name</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CE95375</td>
<td>SPI Master and DMA with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95376</td>
<td>SPI Slave and DMA with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>Digital Filter Block (DFB)</td>
<td></td>
</tr>
<tr>
<td>CE95316</td>
<td>Filter From to ADC to VDAC Using DFB with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95317</td>
<td>Filter From to ADC to VDAC Using DFB in Polling Mode with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>I2C</td>
<td></td>
</tr>
<tr>
<td>CE95324</td>
<td>I2C LCD with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95314</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP EZI2C</td>
</tr>
<tr>
<td>USB</td>
<td></td>
</tr>
<tr>
<td>AN57294</td>
<td>USB 101: An Introduction to Universal Serial Bus 2.0</td>
</tr>
<tr>
<td>AN57473</td>
<td>USB HID Basics with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>AN58726</td>
<td>USB HID Intermediate with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>AN56377</td>
<td>PSoC® 3 and PSoC 5LP – Introduction to Implementing USB Data Transfers</td>
</tr>
<tr>
<td>AN82072</td>
<td>PSoC® 3 and PSoC 5LP USB General Data Transfer with Standard HID Drivers</td>
</tr>
<tr>
<td>AN73503</td>
<td>USB HID Bootloader for PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95390</td>
<td>USB Audio with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95395</td>
<td>USB MIDI with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95394</td>
<td>USB HID Mouse with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95393</td>
<td>USB Bulk Transfer with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95392</td>
<td>USB Bootloader with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95396</td>
<td>USB UART with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>Controller Area Network (CAN)</td>
<td></td>
</tr>
<tr>
<td>AN52701</td>
<td>PSoC® 3 and PSoC 5LP – Getting Started with Controller Area Network (CAN)</td>
</tr>
<tr>
<td>CE95282</td>
<td>CAN as Control Node with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95283</td>
<td>CAN as Remove Node with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA86565</td>
<td>Difference Between Full CAN and Basic CAN Mailbox</td>
</tr>
<tr>
<td>KBA86566</td>
<td>Acceptance Filter Implementation for CAN Receive Message</td>
</tr>
<tr>
<td>KBA86567</td>
<td>Modifying the Full CAN Mailbox’s Identifier in the Program</td>
</tr>
<tr>
<td>Universal Digital Blocks (UDB)</td>
<td></td>
</tr>
<tr>
<td>AN82250</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP – Implementing Programmable Logic Designs with Verilog</td>
</tr>
<tr>
<td>AN82156</td>
<td>PSoC® 3, PSoC 4, and PSoC 5LP – Designing PSoC Creator™ Components with UDB Datapaths</td>
</tr>
<tr>
<td>CE95295</td>
<td>8-Bit UDB Counter with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95384</td>
<td>16-Bit UDB-Based Timer with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95323</td>
<td>Hardware Fan Control with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA85325</td>
<td>Comparison of Resource Utilization Between PSoC® 3 and PSoC 5LP UDBs and Other Vendor CPLDs</td>
</tr>
<tr>
<td>KBA86336</td>
<td>Just Enough Verilog for PSoC®</td>
</tr>
<tr>
<td>Analog to Digital Converter (ADC)</td>
<td></td>
</tr>
<tr>
<td>AN84783</td>
<td>Accurate Measurement Using PSoC® 3 and PSoC 5LP Delta-Sigma ADC</td>
</tr>
<tr>
<td>Document</td>
<td>Document Name</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CE95277</td>
<td>Delta-Sigma ADC in Single-Ended Mode with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95271</td>
<td>Delta-Sigma ADC in Differential Mode with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95276</td>
<td>Sequencing SAR ADC with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA1866</td>
<td>Best Method of Amplification to Get Better Performance from PSoC® 3 and PSoC 5LP Delta-Sigma ADC</td>
</tr>
<tr>
<td>KBA4753</td>
<td>Choice of Reference Voltage for Accurate ADC Measurements in PSoC® 3, PSoC 4 and PSoC 5LP</td>
</tr>
</tbody>
</table>

### Digital to Analog Converter (DAC)
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN60305</td>
<td>Using PSoC® 3 and PSoC 5LP IDACs to build a better VDAC</td>
</tr>
<tr>
<td>AN64275</td>
<td>PSoC® 3 and PSoC 5LP: Getting More Resolution from 8-Bit DACs</td>
</tr>
<tr>
<td>AN69133</td>
<td>PSoC® 3 and PSoC 5LP Easy Waveform Generation with the WaveDAC8 Component</td>
</tr>
<tr>
<td>CE95397</td>
<td>Voltage DAC with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95309</td>
<td>Dithered Voltage DAC with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA4732</td>
<td>VDAC8 Output Voltage in PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA3238</td>
<td>Driving an External Load using VDAC in PSoC® 3 or PSoC 5LP</td>
</tr>
</tbody>
</table>

### Comparator
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN60220</td>
<td>PSoC® 3 and PSoC 5LP Multiplexed Comparator</td>
</tr>
<tr>
<td>CE95292</td>
<td>Analog Voltage Comparator with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95361</td>
<td>Scanning Comparator Using Internal VDAC with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95360</td>
<td>Scanning Comparator Using Common Mode with PSoC® 3, PSoC 4, and PSoC 5LP</td>
</tr>
</tbody>
</table>

### Operational Amplifier (Opamp)
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE95339</td>
<td>Operational Amplifier (Opamp) with PSoC® 3 and PSoC 5LP</td>
</tr>
</tbody>
</table>

### Programmable Analog Block (SC/CT)
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN60321</td>
<td>Peak Detection with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>AN62582</td>
<td>AM Modulation and Demodulation</td>
</tr>
<tr>
<td>CE95342</td>
<td>Programmable Gain Amplifier (PGA) with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95343</td>
<td>Inverting Programmable Gain Amplifier with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95383</td>
<td>Transimpedance Amplifier (TIA) with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95357</td>
<td>Sample and Hold with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95337</td>
<td>Analog Signal Mixer with PSoC® 3 and PSoC 5LP</td>
</tr>
</tbody>
</table>

### CapSense
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN75400</td>
<td>PSoC® 3 and PSoC 5LP CapSense® Design Guide</td>
</tr>
<tr>
<td>CE95287</td>
<td>CapSense® CSD Using Tuner with PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>CE95284</td>
<td>CapSense® CSD Design with PSoC® 3 and PSoC 5LP</td>
</tr>
</tbody>
</table>

### I/O
<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN72382</td>
<td>Using PSoC® 3 and PSoC 5LP GPIO Pins</td>
</tr>
<tr>
<td>AN60580</td>
<td>SIO Tips and Tricks in PSoC® 3 and PSoC 5LP</td>
</tr>
<tr>
<td>KBA82883</td>
<td>Controlling a PSoC® 3 and PSoC 5LP GPIO in Firmware</td>
</tr>
<tr>
<td>KBA91716</td>
<td>Differences Between SIO and GPIO Pins in PSoC® 3 and PSoC 5LP</td>
</tr>
</tbody>
</table>
### Document Name Table

<table>
<thead>
<tr>
<th>Document</th>
<th>Document Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment LCD</td>
<td></td>
</tr>
<tr>
<td>AN52927</td>
<td>PSoc® 3 and PSoc 5LP - Segment LCD Direct Drive</td>
</tr>
<tr>
<td>CE95368</td>
<td>Segment LCD with PSoc® 5LP</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Orig. of Change</th>
<th>Submission Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>3554384</td>
<td>ROSS</td>
<td>03/20/2012</td>
<td>New application note</td>
</tr>
<tr>
<td>*A</td>
<td>3819945</td>
<td>RNJT</td>
<td>11/23/2012</td>
<td>Updated for PSoC 5LP</td>
</tr>
<tr>
<td>*B</td>
<td>3866015</td>
<td>RNJT</td>
<td>01/11/2013</td>
<td>Fixed PSoC 5LP links</td>
</tr>
</tbody>
</table>
| *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 |
| *F       | 5834826 | AESATMP8        | 07/27/2017      | Updated logo and Copyright. |
|          |         |                 |                 | Updated "PSoC Resources" section to include PSoC 6.  
|          |         |                 |                 | Updated the template. |
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

- Arm® Cortex® Microcontrollers: cypress.com/arm
- Automotive: cypress.com/automotive
- Clocks & Buffers: cypress.com/clocks
- Interface: cypress.com/interface
- Internet of Things: cypress.com/iot
- Memory: cypress.com/memory
- Microcontrollers: cypress.com/mcu
- PSoC: cypress.com/psoc
- Power Management ICs: cypress.com/pmic
- Touch Sensing: cypress.com/touch
- USB Controllers: cypress.com/usb
- Wireless Connectivity: cypress.com/wireless

PSoc® Solutions

- PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

- Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

- cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

© Cypress Semiconductor Corporation, 2012-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.