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.

AN79953

Getting Started with PSoC 4

Authors: Ranjith M, Nidhin M S

Associated Part Family: All PSoC 4 Parts

Related Application Notes: AN54181, AN77759

Software Version: PSoC Creator™4.3 SP2 or higher

To get the latest version of this application note, please visit http://www.cypress.com/AN79953

More code examples? We heard you.

To access an ever-growing list of hundreds of PSoC code examples, visit our code examples web page. You can also explore the PSoC 4 video library here.

AN79953 introduces you to PSoC® 4, an Arm® Cortex®-M0/M0+ based programmable system-on-chip. It helps you explore the PSoC 4 architecture and development tools and shows you how to create your first project using PSoC Creator™, the development tool for PSoC 4. This application note also guides you to more resources to accelerate in-depth learning about PSoC 4.

Contents

1 Introduction....1
2 PSoC Resources....2
2.1 PSoC Creator....2
2.2 Code Examples....4
2.3 PSoC Creator Help....5
2.4 Technical Support....5
3 PSoC 4 Feature Set....6
4 PSoC is More than an MCU....10
4.1 The Concept of PSoC Creator Components....11
5 My First PSoC 4 Design....12
5.1 Before You Begin....12
5.2 About the Design....13
5.3 Part 1: Create the Design....13
5.4 Part 2: Program the Device....22
5.5 Convert Project to Bootloadable for CY8CKIT-049....22
5.6 Bootload Your CY8CKIT-049....23
5.7 More PSoC 4 Code Examples....25
6 Summary....33

1 Introduction

PSoC 4 is a true programmable embedded system-on-chip, integrating custom analog and digital peripheral functions, memory, and an Arm Cortex-M0 or Cortex-M0+ microcontroller on a single chip. This type of system is different from most mixed-signal embedded systems, which use a combination of a microcontroller unit (MCU) and external analog and digital peripherals. Such systems typically require many integrated circuits in addition to the MCU, such as opamps, ADCs, and Application-specific Integrated Circuit (ASICs).

PSoC 4 provides a low-cost alternative to the combination of MCU and external ICs. In addition to reducing overall system cost, the programmable analog and digital subsystems allow great flexibility, in-field tuning of the design, and speedy time to market.

The capacitive touch-sensing feature in PSoC 4, known as CapSense®, offers unprecedented signal-to-noise ratio; best-in-class waterproofing; and a wide variety of sensor types such as buttons, sliders, track pads, and proximity sensors. PSoC 4 offers a best-in-class current consumption of 150 nA while retaining SRAM, programmable logic, and the ability to wake up from an interrupt. PSoC 4 consumes only 20 nA while maintaining wakeup capability in its non-retention power mode. The PSoC 4 family of devices also contains PSoC 4 BLE, which integrates a Bluetooth Low Energy (BLE) radio system. For more details on PSoC 4 BLE, see AN91267.

Using this Document

The next few pages describe PSoC 4 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 4 Design.

2 PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device, and to help you to 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 4:

- **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 4000, PSoC 4100, and PSoC 4200, PSoC 4xx7 BLE, PSoC 4100M, PSoC 4200M, PSoC 4200-L, PSoC 4000S, PSoC 4100S, PSoC 4100PS, and PSoC 4100S Plus devices.
- **CapSense Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 4 family of devices. Refer to AN64846 - Getting Started with CapSense® to learn more about CapSense technology.
- **Application Notes and Code Examples** cover a broad range of topics, from basic to advanced level. PSoC Creator provides more code examples – see Code Examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- **PSoC Training Videos:** These videos provide step-by-step instructions on how to get started building complex designs with PSoC.
- **Development Kits:**
  - CY8CKIT-040, CY8CKIT-041, CY8CKIT-042, CY8CKIT-044, CY8CKIT-046 and CY8CKIT-042-BLE PSoC 4 Pioneer Kits are easy-to-use and inexpensive development platforms. These include connectors for Arduino-compatible shields and Digilent® Pmod™ daughter cards.
  - CY8CKIT-043, CY8CKIT-049, CY8CKIT-145, CY8CKIT-147 and CY8CKIT-149 are very low-cost prototyping platforms for sampling PSoC 4 devices.

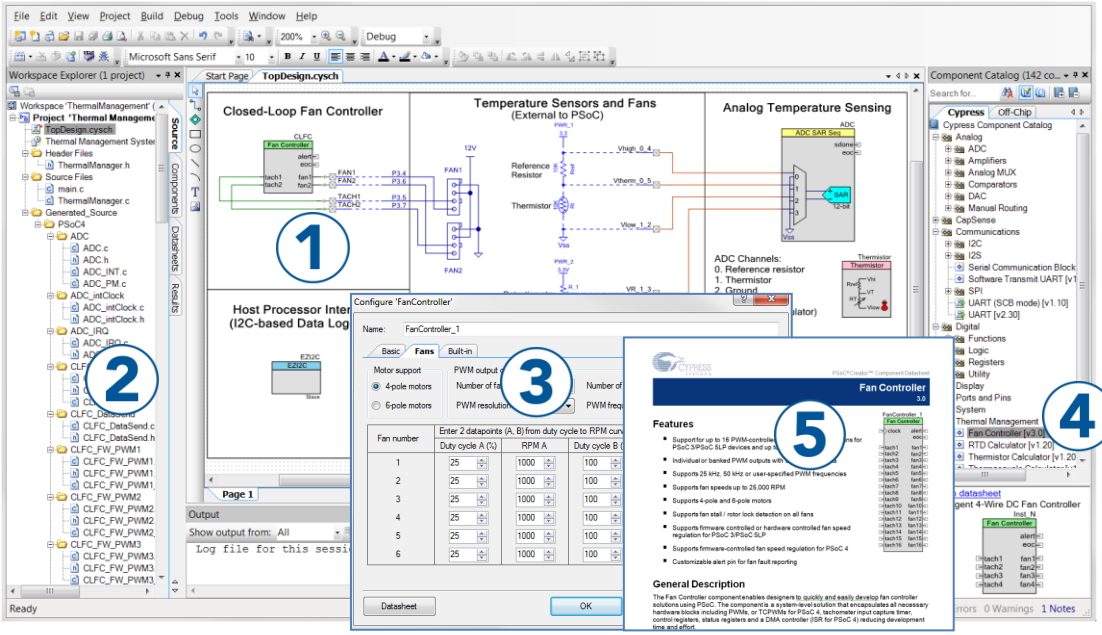
The MiniProg3 device provides an interface for flash programming and debug.

2.1 PSoC Creator

PSoC Creator is a free Windows-based Integrated Development Environment (IDE). It enables concurrent hardware and firmware design of PSoC 3, PSoC 4, PSoC 5LP, and PSoC 6 MCU systems. See Figure 1 – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design
2. Co-design your application firmware with the PSoC hardware
3. Configure Components with config tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 1. PSoC Creator Features



2.2 Code Examples

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

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 can be used for various applications. Code examples and datasheets are included, as shown in Figure 3.

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

- Filter for examples based on architecture or device family, that is, PSoC 3, PSoC 4, PSoC 5LP; or PSoC 6 MCU, category; or keyword
- Select from the menu of examples offered based on the Filter Options
- Review the datasheet for the selection (in 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

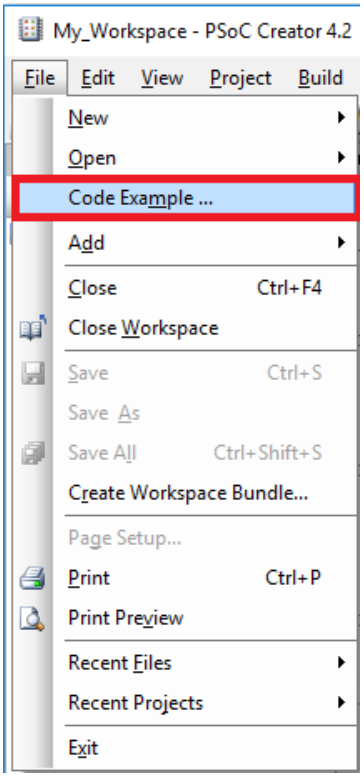
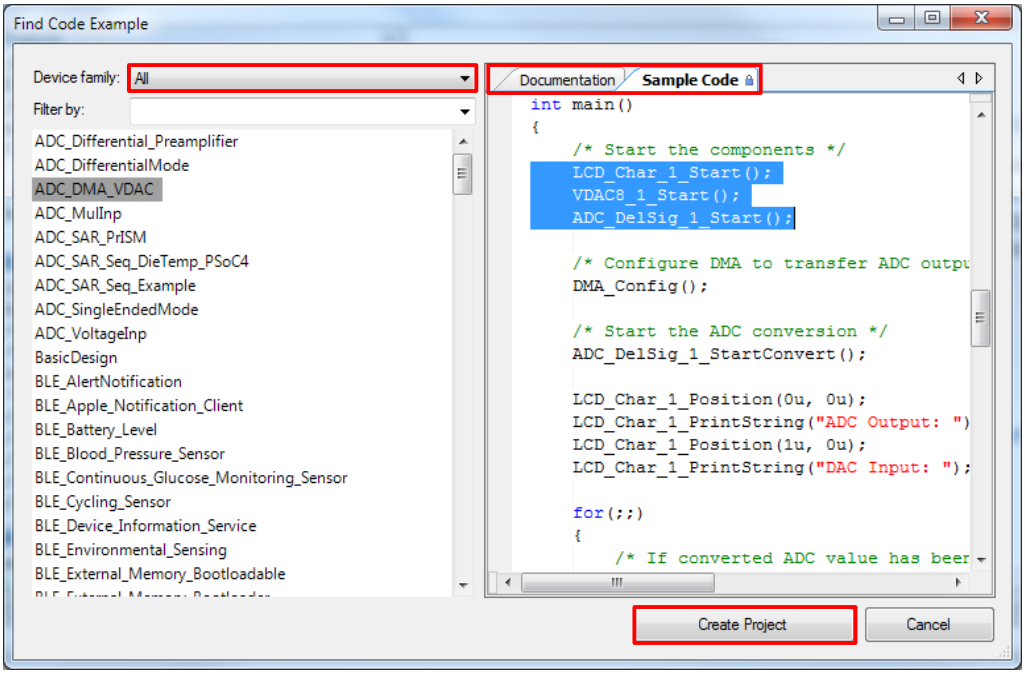


Figure 3. Code Example Projects with Sample Code



2.3 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 **Help** > **Documentation** > **Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple Component example projects**: Choose **File** > **Open** > **Example projects**. These example projects demonstrate how to configure and use PSoC Creator Components.
- **System Reference Guide**: Choose **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 4 Component Datasheets page for a list of all PSoC 4 Component datasheets.
- **PSoC Creator Training Videos**: These videos provide step-by-step instructions on how to get started with PSoC Creator.
- **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**.

2.4 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

3 PSoC 4 Feature Set

PSoC 4 has an extensive set of features, which include a CPU and memory subsystem, a digital subsystem, an analog subsystem, and system resources, as shown in Figure 4. The following sections give brief descriptions of each feature. For more information, see the PSoC 4 family device datasheets, technical reference manuals (TRMs), and application notes listed in PSoC Resources.

Figure 4. PSoC 4 Architecture (PSoC 4200L)

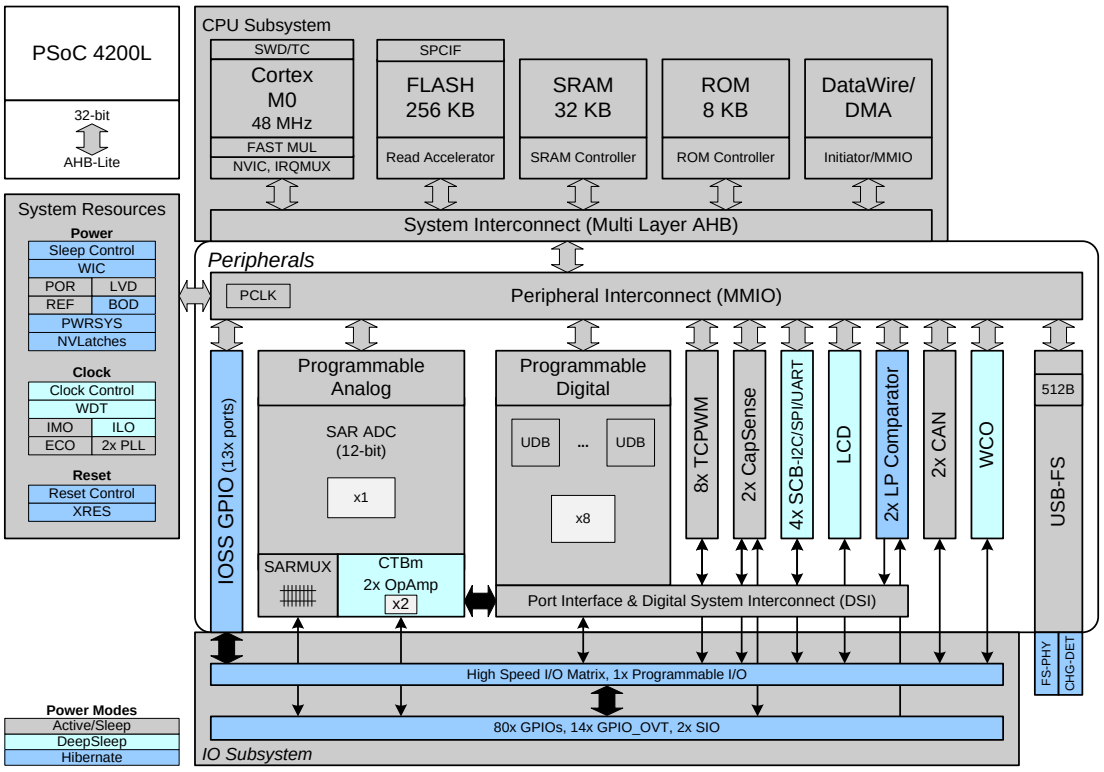


Table 1 shows the features available for different PSoC 4 device families. Depending on the device, all or a subset of these features may be available. See the PSoC 4 Product Selector Guide for details.

Table 1.PSoC 4 Device Families

Features	Device Family									
	CY8C4000	CY8C4000S	CY8C4100	PSoC 4100S	PSoC 4100S Plus	PSoC 4100PS	CY8C4200	CY8C4200M	CY8C4200L	PSoC 4500
CPU	16-MHz Cortex-M0	48-MHz Cortex-M0+ with single-cycle multiply	24-MHz Cortex-M0 with single-cycle multiply	48-MHz Cortex-M0+ with single-cycle multiply	48-MHz Cortex-M0+ with single-cycle multiply	48-MHz Cortex-M0+ with single-cycle multiply	48-MHz Cortex-M0 with single-cycle multiply	48-MHz Cortex-M0 with single-cycle multiply	48-MHz Cortex-M0 with single-cycle multiply	48-MHz Cortex-M0+ with single-cycle multiply
DMA	N/A	N/A	N/A	N/A	8 channels	8 channels	N/A	8 channels	32 channels	8 channels
Flash memory	16 KB	32 KB	32 KB	64 KB	128 KB	32 KB	32 KB	128 KB	256 KB	256 KB
SRAM	2 KB	4 KB	4 KB	8 KB	16 KB	4 KB	4 KB	16 KB	32 KB	32 KB
GPIOs	20	36	36	36	54	38	36	55	96	54
CapSense	16 sensors	35 sensors	35 sensors	35 sensors	53 sensors	33 sensors	35 sensors	54 sensors	94 sensors	N/A
ADC	None	10-bit 46-ksps Single-slope ADC	12-bit, 806-ksps SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer	12-bit, 1-MSPS Scanning SAR ADC	12-bit, 1-MSPS SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer	Two 12-bit, 1-MSPS SAR ADCs with sequencer
Opamps/programmable gain amplifiers (PGAs)	None	None	Two programmable opamps	Two programmable opamps	Two programmable opamps	Four programmable opamps/PGAs	Two programmable opamps	Two programmable opamps	Four programmable opamps	Three programmable opamps
Programmable Voltage Reference (PVref)	None	None	None	None	None	Four channels	None	None	None	None
Voltage DAC (VDAC)	None	None	None	None	None	13-bit VDAC	None	None	None	None
Comparators	1 CSD comparator with fixed threshold (1.2 V)	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature	Two low-power comparators with wakeup feature
IDACs[1]	One 7-bit and one 8-bit	Two 7-bit	One 7-bit and one 8-bit	Two 7-bit	Two 7-bit	Two 7-bit	One 7-bit and one 8-bit	Two 7-bit and two 8-bit	Two 7-bit and two 8-bit	None
Programmable logic blocks (UDBs)	None	None	None	None	None	None	Four UDBs, each with eight macrocells and one datapath	Four UDBs, each with eight macrocells and one datapath	Eight UDBs, each with eight macrocells and one datapath	None
Smart I/O ports	None	2	None	2	2	1	None	None	1	2

[1] IDACs are available only when CapSense is not in use. See the respective PSoC 4 Technical Reference Manual for more details.

Features	Device Family									
	CY8C4000	CY8C4000S	CY8C4100	PSoC 4100S	PSoC 4100S Plus	PSoC 4100PS	CY8C4200	CY8C4200M	CY8C4200L	PSoC 4500
Power supply range	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V	1.71 V to 5.5 V
Low-power modes	Deep-Sleep at 2.5 µA	Deep-Sleep at 2.5 µA	Deep-Sleep at 1.3 µA, Hibernate at 150 nA, Stop at 20 nA	Deep-Sleep at 2.5 µA	Deep-Sleep at 2.5 µA	Deep-Sleep at 2.5 µA	Deep-Sleep at 1.3 µA, Hibernate at 150 nA, Stop at 20 nA	Deep-Sleep at 1.3 µA, Hibernate at 150 nA, Stop at 20 nA	Deep-Sleep at 1.3 µA, Hibernate at 150 nA, Stop at 20 nA	Deep-Sleep at 1.3 µA, Hibernate at 150 nA, Stop at 20 nA
Segment LCD drive	None	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive	8 COM segment LCD drive	None
Serial communication	One I^2C	Two serial communication blocks (SCBs) with programmable I^2C, SPI, or UART	Two SCBs with programmable I^2C, SPI, or UART	Three SCBs with programmable I^2C, SPI, or UART	Five SCBs with programmable I^2C, SPI, or UART	Three SCBs with programmable I^2C, SPI, or UART	Two SCBs with programmable I^2C, SPI, or UART	Two SCBs with programmable I^2C, SPI, or UART	Four SCBs with programmable I^2C, SPI, or UART	Five SCBs with programmable I^2C, SPI, or UART
Timer Counter Pulse-Width Modulator (TCPWM)	1	5	4	5	8	8	4	8	8	8
Controller Area Network (CAN)	None	None	None	None	1	None	None	2	2	None
Universal Serial Bus (USB)	None	None	None	None	None	None	None	None	Full Speed USB Device with eight endpoints	None
Motor Control Acceleration (MCA)	None	None	None	None	None	None	None	None	None	2
Clocks	24-MHz / 32-MHz internal main oscillator (IMO) 32-kHz internal low-speed oscillator (ILO)	24-MHz to 48-MHz IMO 40-kHz ILO 32-kHZ watch crystal oscillator (WCO)	3-MHz to 24-MHz IMO 32-kHz ILO	24-MHz to 48-MHz IMO 40-kHz ILO 32-kHZ WCO	24-MHz to 48-MHz IMO 40-kHz ILO 32-kHZ WCO	24-MHz to 48-MHz IMO 40-kHz ILO 32-kHZ WCO	3-MHz to 48-MHz IMO 32-kHz ILO	3-MHz to 48-MHz IMO 32-kHz ILO 32-kHZ WCO	3-MHz to 48-MHz IMO 4-MHz to 33-MHz external crystal oscillator (ECO) 32-kHz ILO 32-kHZ WCO	3-MHz to 48-MHz IMO 32-kHz ILO 32-kHZ WCO
Power supply monitoring	Power-on reset (POR) Brown-out detection (BOD)	POR, BOD	POR, BOD, Low-voltage detection (LVD)	POR, BOD	POR, BOD	POR, BOD	POR, BOD, LVD	POR, BOD, LVD	POR, BOD, LVD	POR, BOD, LVD

Features	Device Family									
	CY8C4000	CY8C4000S	CY8C4100	PSoC 4100S	PSoC 4100S Plus	PSoC 4100PS	CY8C4200	CY8C4200M	CY8C4200L	PSoC 4500
Supported Kit	CY8CKIT-040 Pioneer kit	CY8CKIT-041 Pioneer kit	CY8CKIT-049 Prototyping Kit	CY8CKIT-041 Pioneer kit	CY8CKIT-149 Prototyping Kit	CY8CKIT-147 Prototyping Kit	CY8CKIT-042 Pioneer kit	CY8CKIT-044 Pioneer kit	CY8CKIT-046 Pioneer kit	None

4 PSoC is More than an MCU

Figure 5 shows that a typical MCU contains a CPU (such as 8051 or an Arm Cortex) with a set of peripheral functions such as ADCs, DACs, UARTs, SPIs, 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. With PSoC, 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 peripheral analog and digital are interconnected with a highly configurable matrix of signal and data bus meshing that allows you to create custom designs that 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

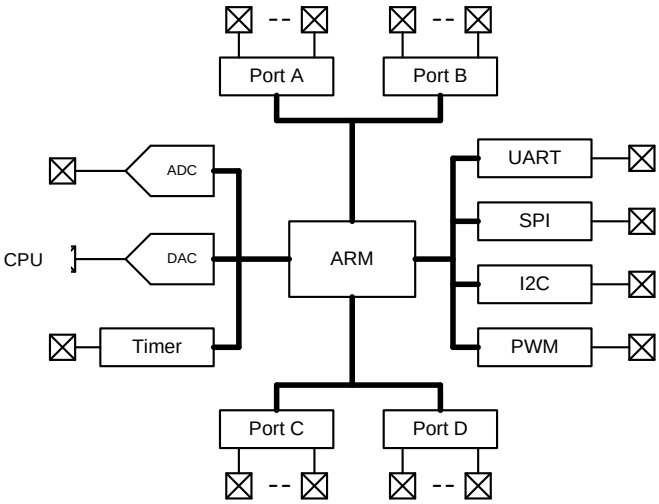
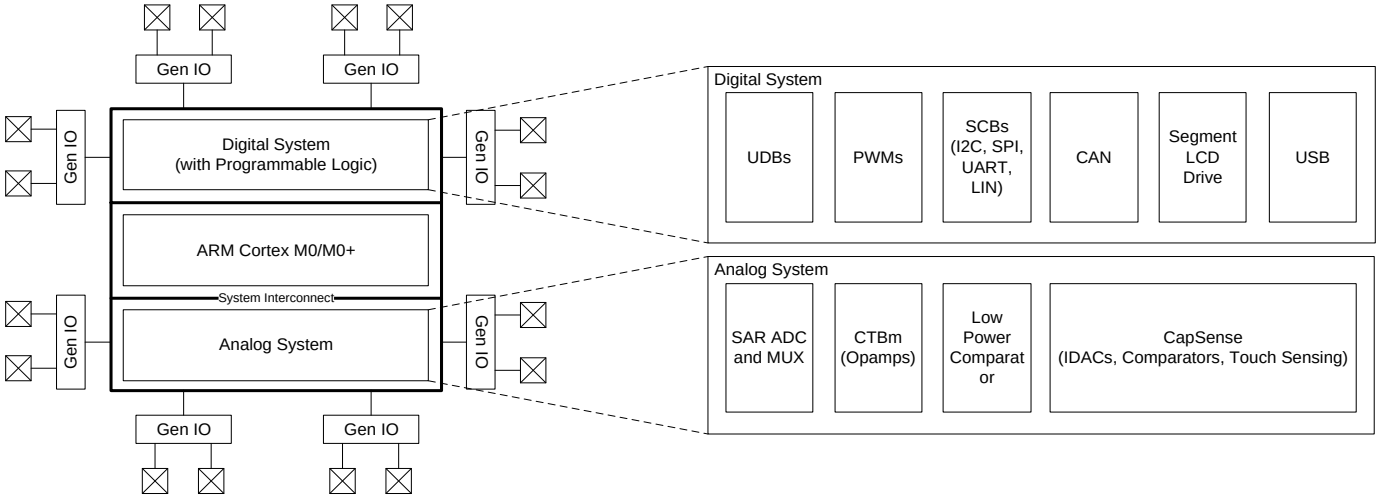


Figure 6. PSoC Block Diagram



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. The projects included with this application note demonstrate this concept. The PSoC is configured to make an LED blink without writing any code for the CPU.

4.1 The Concept of PSoC Creator Components

One other important thing about PSoC is the availability of PSoC Creator IDE. In PSoC Creator, different PSoC resources are organized as graphical elements called Components which can be dragged and dropped on to a schematic to quickly build designs. Every peripheral in PSoC is available as a pre-validated PSoC Creator Component – PWM Component, ADC Component, DAC Component, CapSense Component, UART Component and so on. The availability of pre-validated Components in the PSoC Creator significantly reduces the development time. It also allows you to quickly make changes in the design using graphical options.

For example, configuring a PWM to blink an LED in a typical microcontroller involves the following:

1. Locate the registers corresponding to the PWM block.
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. Many MCUs do not offer alternate pins to connect to the internal blocks.

To implement the same functionality in PSoC is a trivial exercise as you will find out later in this application note. Later, if you need to reconfigure the same PWM block to a Timer, you do not need anything more than a few mouse clicks in PSoC Creator!

The PSoC also has programmable digital blocks known as Universal Digital Blocks (UDBs). PSoC Creator also provides several Components made of UDBs such as UART, SPI, I2S, Timer, PWM, Counter, Digital Gates (AND, OR, NOT, XOR, and so on), and many more. You can even create your own custom state machines and digital logic using the UDBs in PSoC Creator. The method to create your own custom PSoC Creator Components is provided in the PSoC® Creator™ Component Author Guide.

5 My First PSoC 4 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 and how to use PSoC Creator

5.1 Before You Begin

5.1.1 Have you installed PSoC Creator?

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

5.1.2 Do you have a Development Kit or Prototyping Kit?

Testing this design requires one of the kits listed in Table 2, which has an integrated programmer.

Table 2. List of PSoC 4 Pioneer Kits, Prototyping Kits, and Supported Devices

Kit Name	Kit Type	Supported Device Family	Part Number
CY8CKIT-040	Pioneer Kit	PSoC 4000	CY8C4014LQI-422
CY8CKIT-041	Pioneer Kit	PSoC 4000S	CY8C4045AZI-S413
CY8CKIT-041	Pioneer Kit	PSoC 4100S	CY8C4146AZI-S433
CY8CKIT-042	Pioneer Kit	PSoC 4200	CY8C4245AXI-483
CY8CKIT-044	Pioneer Kit	PSoC 4200M	CY8C4247AZI-M485
CY8CKIT-046	Pioneer Kit	PSoC 4200L	CY8C4248BZI-L489
CY8CKIT-042-BLE	Pioneer Kit	PSoC 4200 BLE	CY8C4247LQI-BL483
CY8CKIT-043	Prototyping Kit	PSoC 4200M	CY8C4247AZI-M485
CY8CKIT-049	Prototyping Kit	PSoC 4200	CY8C4245AXI-483
		PSoC 4100	CY8C4125AXI-483
CY8CKIT-145	Prototyping Kit	PSoC 4000S	CY8C4045AZI-S413
CY8CKIT-147	Prototyping Kit	PSoC 4100PS	CY8C4145LQI-PS433
CY8CKIT-149	Prototyping Kit	PSoC 4100S Plus	CY8C4147AZI-S475

If you have any of the above kits, jump to the section Part 1: Create the Design.

If you are using CY8CKIT-049, which has a USB-serial bootloader instead of a programmer, use the *CY8CKIT_049_Example* provided along with this application note. You can download it from the AN79953 landing page as a part of AN79953.zip. To know how to bootload this example project to your CY8CKIT-049, navigate to the section Bootload Your CY8CKIT-049.

You can also evaluate the code examples provided with the kit instead of this design. See the “Code Examples” section in the kit guide for details. Go to the CY8CKIT-049 kit webpage to download kit guide and code examples.

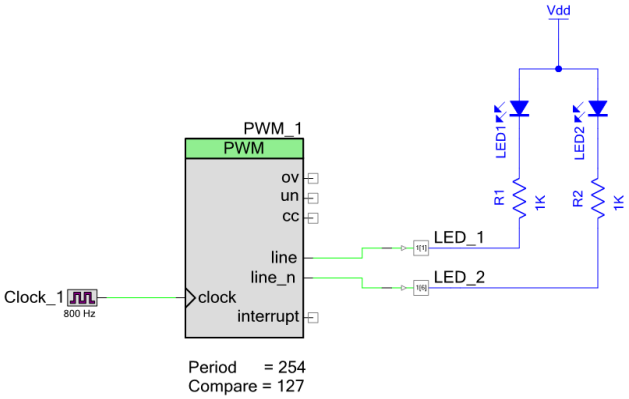
5.1.3 Want to see the project in action?

If you do not want to go through the design process, you can get the completed PSoC Creator project at http://www.cypress.com/go/AN79953. You can then jump to the Build and Program steps.

5.2 About the Design

This design simply blinks two LEDs using a TCPWM Component, as shown in Figure 7. The TCPWM is configured in PWM mode. The two complementary outputs of this PWM control the LEDs. The PWM operates at a very low frequency and 50 percent duty cycle so that the toggling of the LEDs is visible. If you use a dual-color LED instead of two separate LEDs, this project can toggle the color of the dual-color LED.

Figure 7. My First PSoC 4 Design

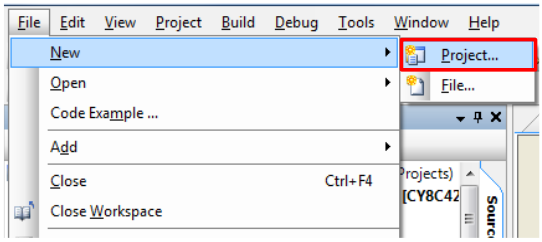


5.3 Part 1: Create the Design

This section takes you on a step-by-step guided tour of the design process. It starts with creating an empty project and guides you through hardware and firmware design entry.

1. Start PSoC Creator, and from the **File** menu choose **New** > **Project**, as shown in Figure 8.

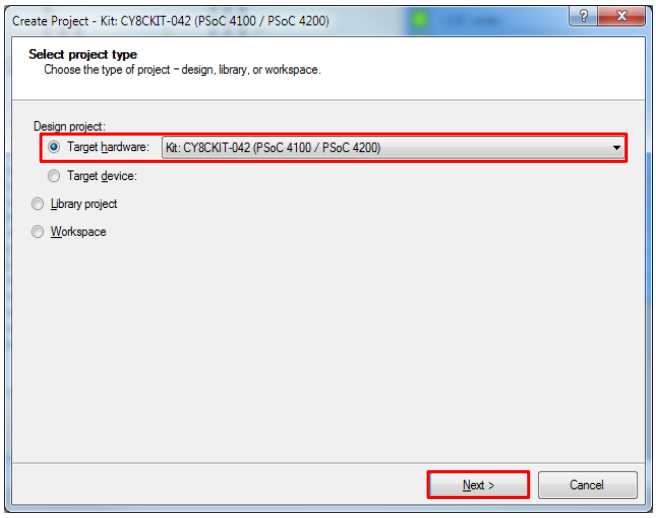
Figure 8. Creating a New Project



2. Select your development kit in the pop-up window. For example, if you have a CY8CKIT-042, select **Kit: CY8CKIT-042 (PSoC 4100 / PSoC 4200)** and click **Next**. If you do not see your PSoC 4 development kit listed in the menu, download and install the kit setup for your kit from the Cypress website.

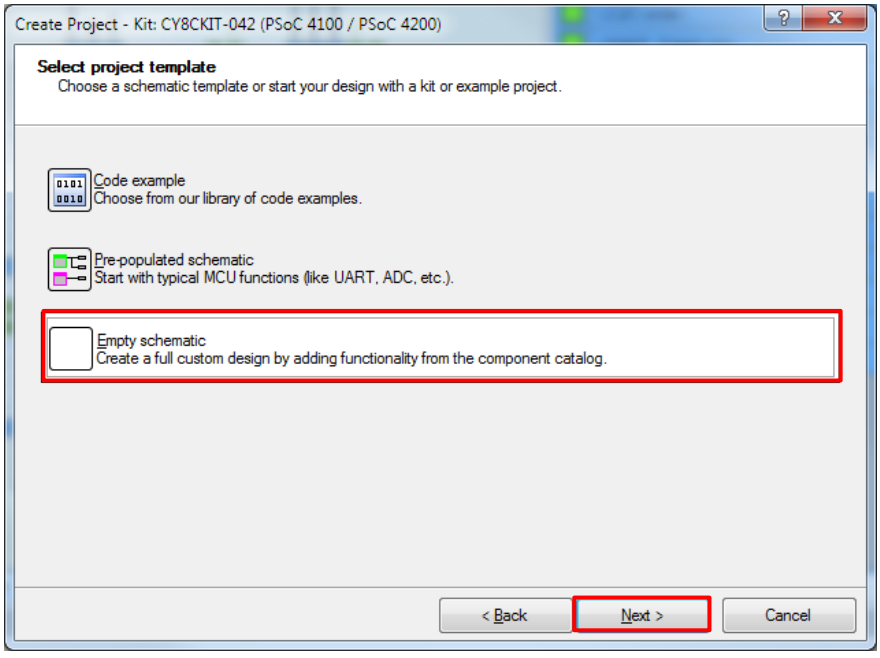
   Alternately, you can also select the target device radio button instead of the target hardware and select the appropriate device and click **Next**.

Figure 9. Create a New Empty PSoC 4 Project

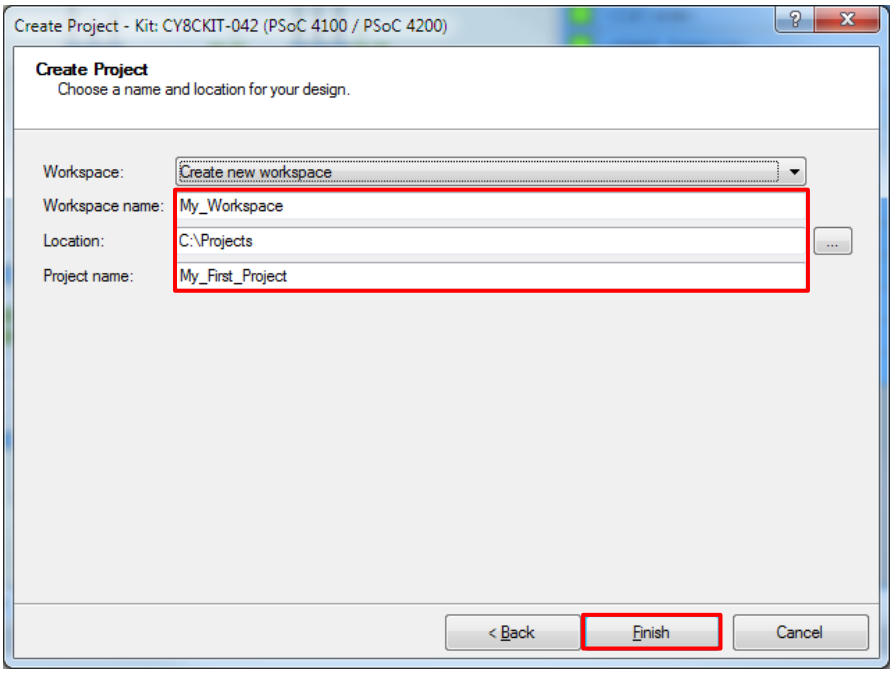


3. Select the option **Empty Schematic** from the next window and click **Next.**

Figure 10. Select Empty Schematic

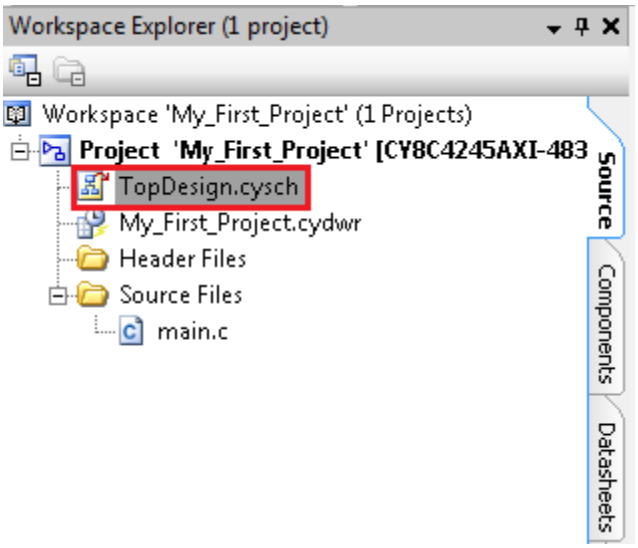


4. Provide a project name (for example, “My_First_Project”) and Workspace Name as shown in the figure below. Choose an appropriate location for your new project, and click **Finish**.



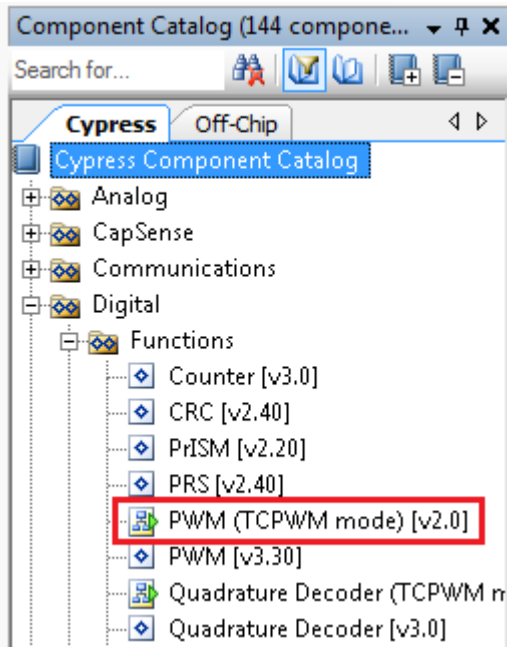
5. Creating a new project generates a project folder with a baseline set of files shown in the **Workspace Explorer** (see Figure 11). To open the project schematic file, double-click *TopDesign.cysch*.

Figure 11. Opening TopDesign Schematic



6. Drag one **PWM (TCPWM mode)** Component from the **Component Catalog** onto the schematic, as shown in Figure 12.

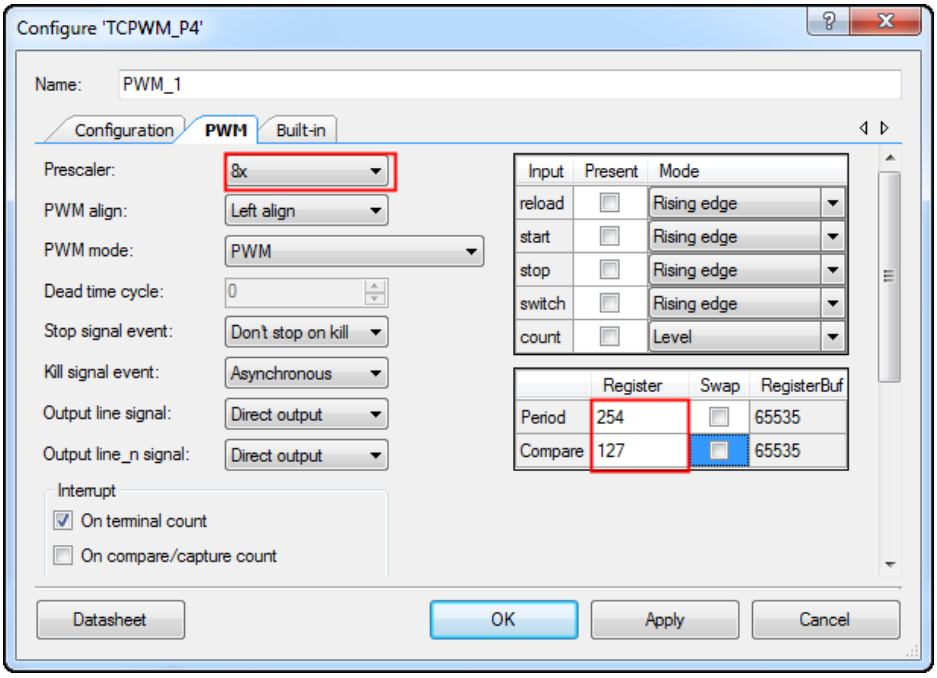
Figure 12. Location of the PWM Component



7. Double-click the PWM Component on the schematic to configure the Component properties, as shown in Figure 13. Click the **PWM** tab, and set the Period value to 254 and the Compare value to 127 to generate a PWM signal with a 50 percent duty cycle.

   Set the **Prescaler** to 8x, to divide the input clock frequency by 8.

Figure 13. Configuring the PWM Component



8. A PWM Component requires an input clock for its operation. Drag and drop a **Clock** Component onto the schematic, and configure the **Frequency** to 800 Hz by double-clicking on the Component, as shown in Figure 14 and Figure 15.

   Since the Prescaler value set in PWM Component is 8, the effective input clock of the PWM is only 100 Hz. Therefore, the PWM period of 254 results in a PWM output time period of 2.54 seconds.

Figure 14. Location of the Clock Component

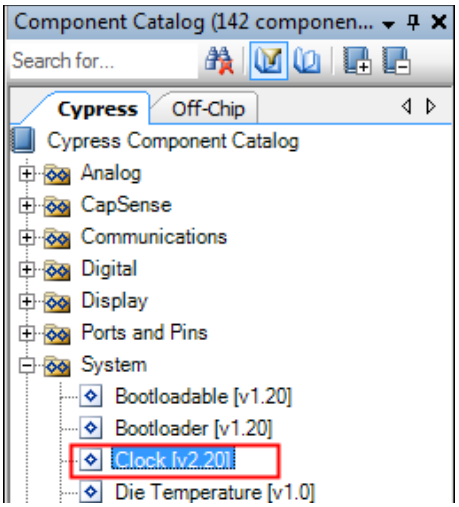
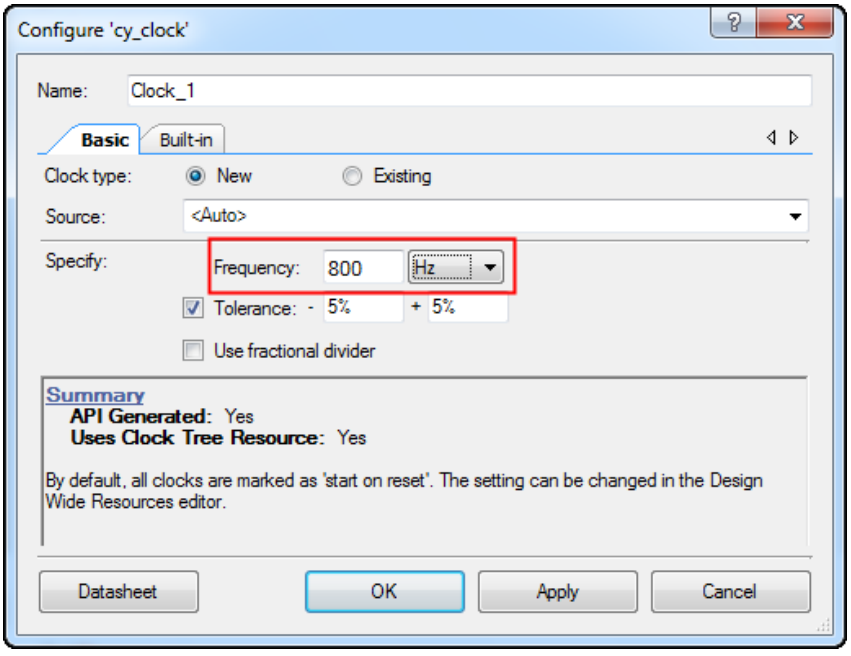


Figure 15. Configuring the Clock Component



9. Drag and drop a **Digital Output Pin** Component. Change the name to LED_1 as shown in Figure 16 and Figure 17. Add another Digital Output Pin Component and change its name to LED_2.

Figure 16. Location of the Digital Output Pin Component

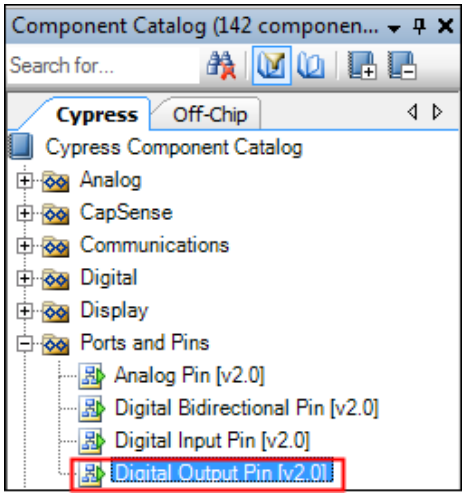
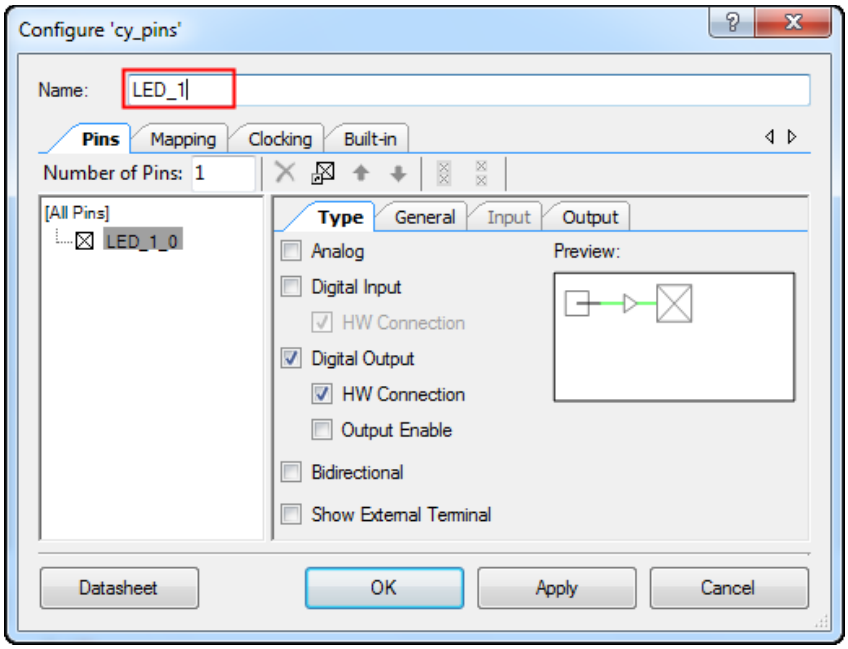
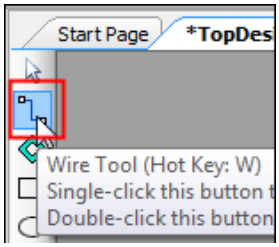


Figure 17. Renaming a Pin Component



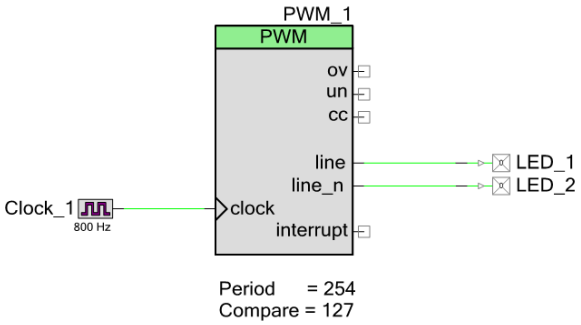
10. In the schematic window, select the wire tool as shown in Figure 18, or press **W**.

Figure 18. Selecting the Wire Tool



11. Wire the Components together, as shown in Figure 19.

Figure 19. Wiring the Schematic



12. Most Components are disabled at device reset (the major exception being the Clock Component, which is automatically started as a default), and you must add code to the project to enable them. Open *main.c* from **Workspace Explorer** and add code to the `main()` function, as provided in Code 1.

Code 1. Enabling the PWM Component

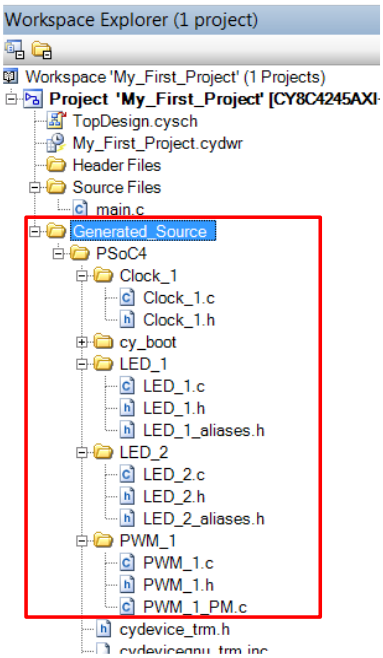
```
int main()
{
  /* Enable and start the PWM */
  PWM_1_Start();

  for(;;)
  {

  }
}
```

13. Select **Build My_First_Project** from the Build menu. Notice in the **Workspace Explorer** window that PSoC Creator automatically generates source code files for the PWM, Clock, and Digital Output Pin Components, as shown in Figure 20.

Figure 20. Generated Source Files



14. Open the file *My_First_Project.cydwr* (Design-Wide Resource file) from **Workspace Explorer** and click the **Pins** tab. You can use this tab to select the device pins for the outputs LED_1 and LED_2.

    Figure 21 shows the pin configuration to connect the LED_1 and LED_2 pins to the green and red LEDs in the CY8CKIT-042 PSoC 4 Pioneer Kit. Refer to

Table 3 if you are using a different PSoC 4 Pioneer Kit, or Table 4 if you are using a PSoC 4 prototyping kit.

Figure 21. Pin Selection

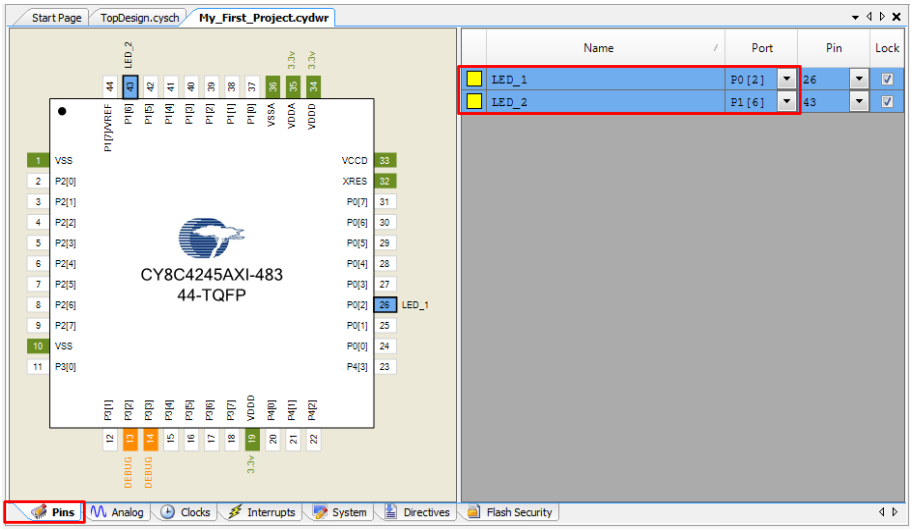


Table 3. Pin Mapping Table across Pioneer Kits

Function	CY8CKIT-040 (PSoC 4000)	CY8CKIT-041 (PSoC 4000S / PSoC 4100S)	CY8CKIT-042 (PSoC 4200)	CY8CKIT-042-BLE (PSoC 4200 BLE)	CY8CKIT-044 (PSoC 4200M)	CY8CKIT-046 (PSoC 4200L)
Green LED (Active LOW)	P1[1]	P2[6]	P0[2]	P3[6]	P2[6]	P5[3]
Red LED (Active LOW)	P3[2]*	P3[4]**	P1[6]	P2[6]	P0[6]	P5[2]

Table 4. Pin Mapping Table Across Prototyping Kits

Function	**CY8CKIT-145** (PSoC 4000S)	**CY8CKIT-149** (PSoC 4100S Plus)
Green LED - LED 1 (Active LOW)	P3[4]	P1[6]
Green LED - LED 2 (Active LOW)	P3[5]	P5[7]

*PSoC 4000 parts have fixed pins for complementary PWM outputs – P1[1] and P1[6]. You cannot use any other pins for PWM outputs. Refer to the device datasheet for more details. If you are using the CY8CKIT-040, you can use the green LED connected to P1[1], as LED1. To use the red LED as LED2, connect P3[2] from header J4 to P1[6] from header J3, using a wire. Alternately, you can connect an external LED to P1[6] as LED2.

**Similar to the note above, If you are using the CY8CKIT-041, you can use the green LED connected to P2[6], as LED1 and the complementary PWM output P2[7] for LED2. To use the red LED as LED2, connect P3[4] from header J2 to P2[7] from header J3, using a wire. Alternately, you can connect an external LED to P2[7] as LED2.

Note: CY8CKIT-043 and CY8CKIT-049 have only one LED connected to P1[6]. If you are using CY8CKIT-049, you can connect an external LED to pin P0[2]. CY8CKIT-147 has only one LED connected to P0[2]; you can connect an external LED to pin P0[3].

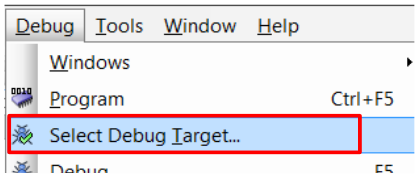
15. Finally, rebuild the project as Step 12 explains.
16. Continue to the next section if you are not using a CY8CKIT-049. If you are using a CY8CKIT-049, navigate to the section Convert Project to Bootloadable for CY8CKIT-049.

5.4 Part 2: Program the Device

This section shows how to program the device. If you are using CY8CKIT-040, CY8CKIT-041, CY8CKIT-042, CY8CKIT-044, CY8CKIT-046, CY8CKIT-043, CY8CKIT-049, CY8CKIT-145, CY8CKIT-147 or CY8CKIT-149, connect the kit board to your computer using the USB cable.

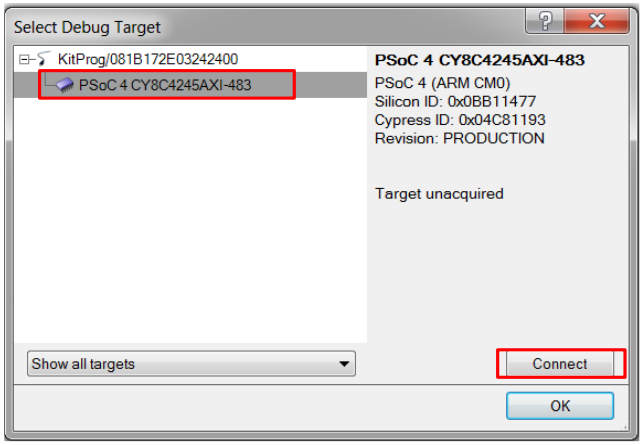
1. Select the PSoC Creator menu item **Debug** > **Select Debug Target**, as shown in Figure 22.

Figure 22. Selecting Debug Target



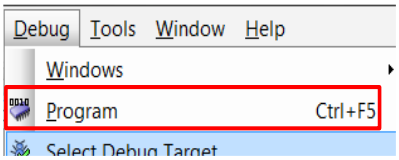
2. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect**, as shown in Figure 23. Click **OK** to close the dialog box.

Figure 23. Connecting to a Device



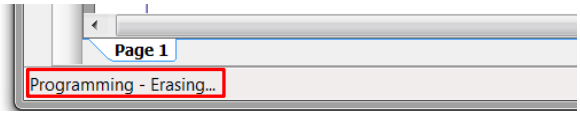
3. Choose the menu item **Debug** > **Program** to program the device with the project, as shown in Figure 24.

Figure 24. Programming the Device



4. You can view the programming status on the status bar (lower-left corner of the window), as shown in Figure 25.

Figure 25. Programming Status



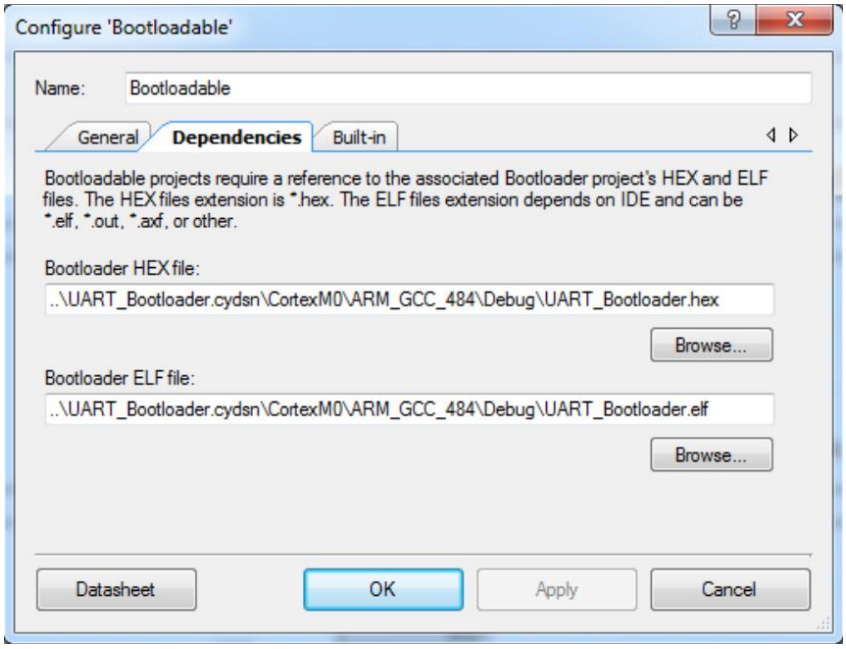
5. After the device is programmed, verify the operation of the project by viewing the toggling of the LEDs.

5.5 Convert Project to Bootloadable for CY8CKIT-049

1. Navigate to the TopDesign schematic by double-clicking the **TopDesign.cysch** in **Workspace Explorer**.
2. Drag and drop a **Bootloadable** Component from the **Component Catalog** to the TopDesign Schematic.

3. Double- click the Bootloadable Component and click the **Dependencies** tab to select the .hex and .elf files from the UART Bootloader project included with the kit (*\CY8CKIT-049-42xx\Firmware\SCB_Bootloader\UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug*). This is done to point the bootloadable project to the bootloader running in the kit. Click **Apply** and then **OK**.

Figure 26. Adding UART Bootloader Dependency



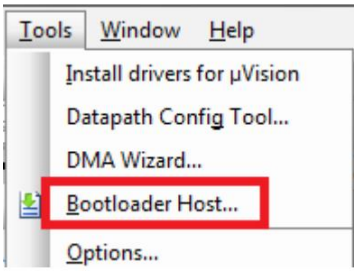
4. Finally, rebuild the project by selecting the option **Build My_First_Project** from the **Build** menu.

5.6 Bootload Your CY8CKIT-049

The CY8CKIT-049 is a little different from the other PSoC 4 development kits. The CY8CKIT-049 does not have an on-board programmer and needs to be bootloaded. See AN73854 for additional details on bootloading. To bootload the example project provided with this application note to the CY8CKIT-049, perform these steps:

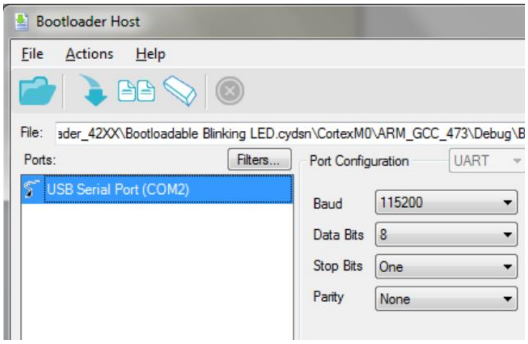
1. Connect the CY8CKIT-049-4xxx prototyping board to the PC. When connecting the kit to the port, depress the SW1 button as it is plugged in. You will notice that the blue LED begins to blink rapidly; this indicates that the PSoC 4 is in 'Bootloader Mode' and is ready to be loaded with the latest firmware. This must be done each time you bootload the PSoC 4.
2. Select **Tools** > **Bootloader Host** to open the Bootloader Host tool.

Figure 27. Launch Bootloader Host Tool



The Bootloader Host tool opens.

Figure 28. Bootloader Host Tool



3. Click **Filters** and select the **Show UART Devices** option from the Port Filters window, and then click **OK**. This lists all COM devices connected to the computer.

   Note: The PID of the Bootloader is F13B. You may enter this PID in the Port filters window to list only the Kit Bootloader.

Figure 29. Port Filters



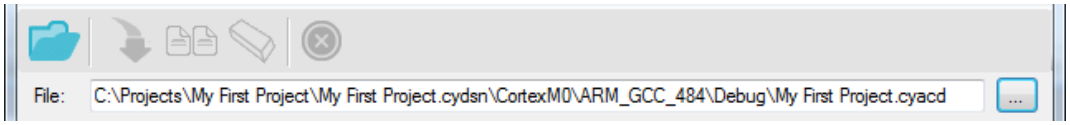
The Bootloader Host tool will now display all of the available UART based COM ports.

4. Click the COM port from the list of available ports and enter the UART configuration such as Baud Rate, Data Bits, Stop Bits, and Parity for the USB-UART configuration on the USB-Serial device.

   The values for the UART are: 115200 baud rate, 8 data bits, 1 stop bit, and no parity

5. Click **File** > **Open** and navigate to the `My_First_Project.cyacd` file generated in the CortexM0 folder in your project directory, and then click **Open**.

Figure 30. Opening the Generated File



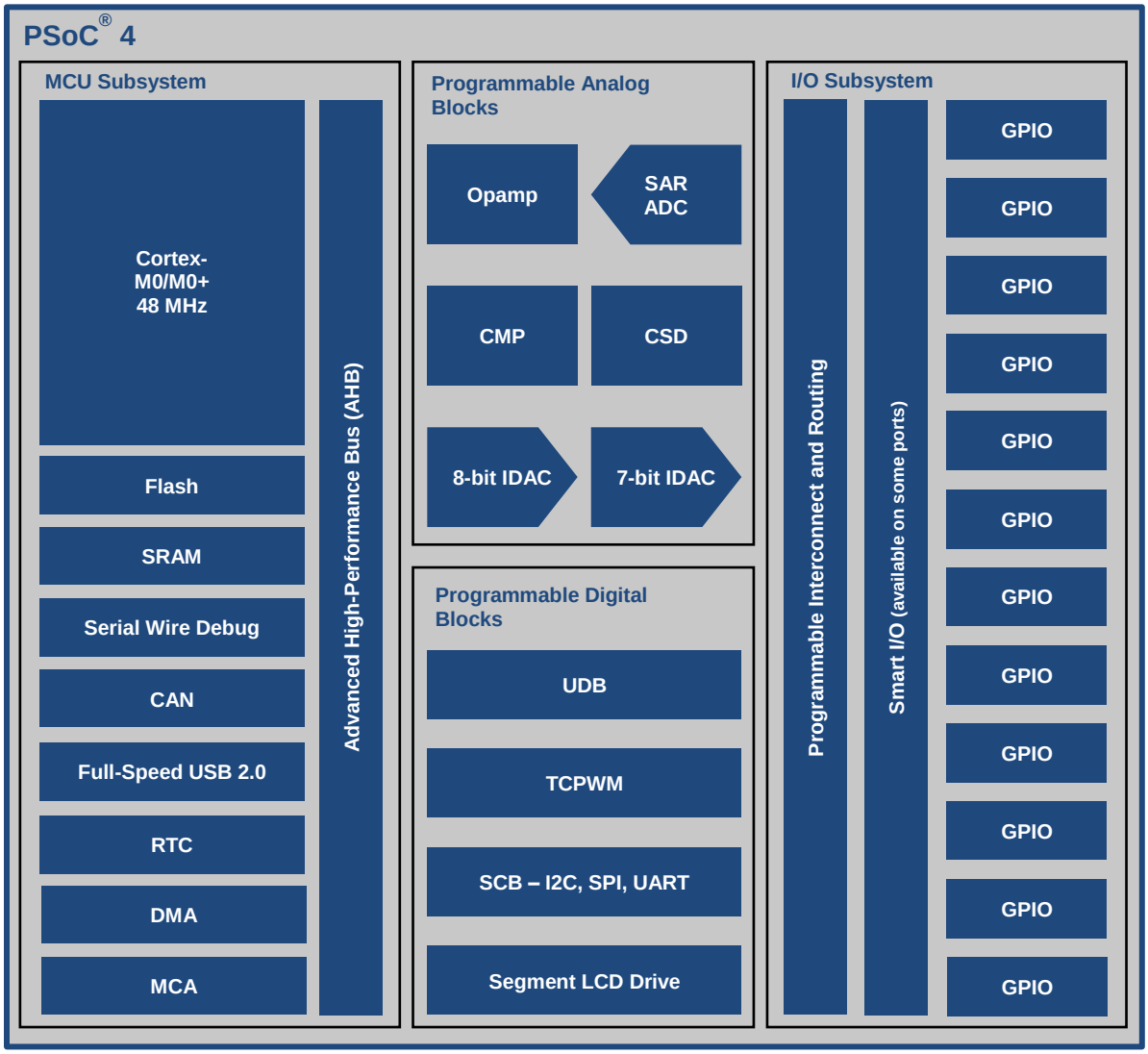
6. Click the **Program** button to flash the part with your new application code. The status window provides output message and a status bar indicates the programming progress. When bootloading is complete, your application executes with the latest version of the application code.

Figure 31. Program the Device With Application Code



5.7 More PSoC 4 Code Examples

You can find more PSoC 4 code examples in the Cypress website. The following figure shows a cumulative representation of different blocks available in devices of the PSoC 4 family. Examples corresponding to each block are categorized in the following section. Click on any of the PSoC 4 block to navigate to the corresponding code example link.



5.7.1 CPU – Arm Cortex M0 / Cortex M0+

The PSoC® 4 Arm Cortex-M0 core is a 32-bit CPU optimized for low-power operation. It has an efficient three-stage pipeline, a fixed 4-GB memory map, and supports the Armv6-M Thumb instruction set. Cortex-M0 also features a single-cycle 32- bit multiply instruction and low-latency interrupt handling. Other subsystems tightly linked to the CPU core include a nested vectored interrupt controller (NVIC), a SYSTICK timer, and debug. Arm Cortex-M0+ available on PSoC 4000S, PSoC 4100S, and PSoC 4100S Plus is an optimized superset of Cortex-M0 with a core pipeline of just two stages resulting in better power efficiency.

Table 5. CPU Subsystem Application Notes

Document Number	Document Name
AN90799	PSoC® 4 Interrupts
AN89610	PSoC® 4 and PSoC 5LP Arm Cortex Code Optimization

5.7.2 Memory - Flash and SRAM

The PSoC 4 has a flash module with a flash accelerator, tightly coupled to the CPU to improve average access times from the flash block. The flash accelerator delivers 85% of single-cycle SRAM access performance on average. Part of the flash module can be used to emulate EEPROM operation if required. SRAM memory is retained during Hibernate.

Table 6. Emulated EEPROM Code Examples

Document Number	Document Name
CE95313	Emulated EEPROM memory with PSoC 3/4/5LP

5.7.3 DMA

A DMA engine, with 32 channels, is provided on PSoC 4200L that can do 32-bit transfers between peripherals or memory.

Table 7. PSoC 4 DMA Component Datasheet

Document Number	Document Name
001-96043	PSoC 4 Direct Memory Access (DMA) Channel Component Datasheet

5.7.4 MCA

The Motor Control Acceleration (MCA) block supports 32-bit divide and square root operations used in motor control algorithms. MCA runs independent of the CPU, therefore reducing the execution times of Field Oriented Control (FOC) and Power Factor Correction (PFC) algorithms.

Table 8. Related Application Notes

Document Number	Document Name
001-93637	PSoC® 4 Sensorless Field-Oriented Control (FOC)
002-04344	FM3 CY9B100A/300A/400A/500A Microcontroller Power Factor Correction

5.7.5 System Resources – Clocks, RTC, Power System

The PSoC 4 system resources consist of the power system, clock system, reset, and voltage reference. The power system provides assurance that voltage levels are as required for each respective mode and either delay mode entry (on power-on reset (POR), for example) until voltage levels are as required for proper function or generate resets (brown-out detect (BOD)) or interrupts (low voltage detect (LVD)).

The PSoC 4 clock system is responsible for providing clocks to all subsystems that require clocks and for switching between different clock sources without glitching. In addition, the clock system ensures that no meta-stable conditions occur. The clock system for the PSoC 4200L consists of a Watch Crystal Oscillator (WCO) running at 32 kHz, the IMO (3 to 48 MHz) and the ILO (32-kHz nominal) internal oscillators, and provision for an external clock. Refer to the device datasheet of the respective PSoC 4 device for the exact clock resources available on the chip.

Table 9. System Resources Application Notes

Document Number	Document Name
AN86233	PSoC® 4 Low-Power Modes and Power Reduction Techniques
AN90114	PSoC® 4000 Family Low-Power System Design Techniques
AN96667	PSoC® Real-Time Clock Based on Power-Line Frequency

Table 10. RTC Code Examples

Document Number	Document Name
CE96926	PSoC® Real-Time Clock Based on Power Line Frequency
CE95915	Implementing an RTC with PSoC® 4100/PSoC 4200 Devices

Table 11. System Resources Code Examples

Document Number	Document Name
CE95322	Hibernate and Wake Up with PSoC 4
CE95321	Hibernate and Stop Power Modes with PSoC 4

Table 12. WDT Code Examples

Document Number	Document Name
CE95401	Watchdog timer example using LCD with PSoC 4
CE95400	Watchdog timer interrupts and device reset for CY8CKIT-040 with PSoC 4000

5.7.6 SAR ADC

The 12-bit 1 MSample/second SAR ADC can operate at a maximum clock rate of 18 MHz and requires a minimum of 18 clocks at that frequency to do a 12-bit conversion.

Table 13. SAR ADC Application Note

Document Number	Document Name
AN60590	PSoC® 3, PSoC 4, and PSoC 5LP – Temperature Measurement with a Diode

Table 14. SAR ADC Code Examples

Document Number	Document Name
CE95275	Sequencing SAR ADC and Die temperature sensor with PSoC 4
CE95272	SAR ADC in Differential Mode using Pre-Amplifier with PSoC 4

Table 15. PSoC Creator SAR ADC Component Datasheet

Document Number	Document Name
001-96792	PSoC 4 Sequencing Successive Approximation ADC (ADC_SAR_Seq)
002-16214	PSoC 4 Scanning SAR ADC (Scan_ADC)

5.7.7 CapSense and IDAC

The PSoC 4 has a CapSense block that allows you to detect finger touches. The CapSense block can be used to implement user interfaces like buttons, linear and radial sliders, track pads, and proximity sensors.

Each CapSense block has two IDACs which can be used for general purposes if CapSense is not being used.

Table 16. CapSense Application Notes

Document Number	Document Name
AN85951	PSoC® 4 CapSense® Design Guide
AN92239	Proximity Sensing with CapSense®

Table 17. CapSense Code Examples

Document Number	Document Name
CE95297	1.2 Volt Comparator (CSD Comp) using Analog Multiplexer with PSoC 4
CE95289	CapSense Proximity with PSoC 4
CE95288	CapSense Low Power with PSoC 4
CE95286	CapSense CSD using Tuner with PSoC 4
CE95285	CapSense CSD with PSoC 4
CE95327	8-Bit Current Digital to Analog Converter (IDAC8) with PSoC 4

Table 18. PSoC Creator CapSense and IDAC Component Datasheets

Document Number	Document Name
001-96069	PSoC 4 Capacitive Sensing (CapSense® CSD)
001-96067	PSoC 4 Capacitive Sensing (CapSense® Gesture)
001-96481	PSoC 4 Current Digital to Analog Converter

5.7.8 Programmable Analog – Opamp, Low Power Comparators, Programmable Reference and Voltage DAC

The PSoC 4 has up to four opamps with comparator modes, which allow most common analog functions to be performed on-chip eliminating external components; PGAs, voltage buffers, filters, trans-impedance amplifiers, and other functions can be realized with external passives saving power, cost, and space. The PSoC 4 also have up to 2 low-power comparators, which can also operate in the Deep Sleep and Hibernate modes. This allows the analog system blocks to be disabled while retaining the ability to monitor external voltage levels during low-power modes. PSoC 4100PS family of devices has programmable voltage reference (PVref), which can be used to generate up to four channels of programmable voltage references; it also has a 13-bit voltage DAC (VDAC).

Table 19. Opamp Code Examples

Document Number	Document Name
CE95341	Operational Amplifier (OpAmp) with PSoC 4
CE95340	Amplifier with Dynamic Gain Switching with PSoC 4

Table 20. Comparator Code Examples

Document Number	Document Name
CE95360	Scanning Comparator using common mode with PSoC 3/4/5LP
CE95293	Analog Voltage Comparator with PSoC 4
CE95338	Multiplexed Analog Comparator with PSoC 4
CE95333	Low Power Comparator with PSoC 4

Table 21. PSoC Creator Opamp, Low Power Comparator and Programmable Reference Component Datasheet

Document Number	Document Name
001-92541	PSoC 4 Operational Amplifier (Opamp)
001-92647	PSoC 4 Low Power Comparator
001-90677	1.2 Volt Comparator (CSD Comp)
002-10611	PSoC 4 Programmable Voltage Reference (PVref_P4)
002-16213	VDAC Component Datasheet

5.7.9 TCPWM

The Timer, Counter, and Pulse Width Modulator (TCPWM) block in PSoC® 4 implements the 16-bit timer, counter, pulse width modulator (PWM), and quadrature decoder functionality. The block can be used to measure the period and pulse width of an input signal (timer), find the number of times a particular event occurs (counter), generate PWM signals, or decode quadrature signals.

Table 22. TCPWM Application Notes

Document Number	Document Name
AN93637	PSoC® 4 Sensorless Field-Oriented Control (FOC)

Table 23. TCPWM Code Examples

Document Number	Document Name
CE95354	Quadrature Decoder with PSoC 4
CE95351	Fixed Function PWM with PSoC 4
CE95349	Pseudo Random Sequence (PRS) generator with PSoC ¾/5LP
CE95385	TCPWM Timer/Counter Operation with PSoC 4
CE95380	TCPWM configuration with PSoC 4

Table 24. PSoC Creator TCPWM Component Datasheet

Document Number	Document Name
001-92448	PSoC 4 Timer Counter Pulse Width Modulator (TCPWM)

5.7.10 SCB – I2C, SPI, UART

The PSoC 4 has four SCBs, which can each implement an I2C, UART, or SPI interface.

Table 25. SCB Code Examples

Document Number	Document Name
CE96999	Basic LIN Slave Implementation in PSoC® 4
CE95325	I2C LCD with PSoC 4
CE95366	UART Transmit and Receive using a Serial Communication Block (SCB) with PSoC 4
CE95365	SPI Transmit and Receive using a Serial Communication Block (SCB) with PSoC 4
CE95364	I2C Slave using a Serial Communication Block (SCB) with PSoC 4
CE95363	I2C Master using a Serial Communication Block (SCB) with PSoC 4
CE95362	Cypress EzI2C communication using a Serial Communication Block (SCB) with PSoC 4
CE95345	PMBus Slave in Thermal Management Application with PSoC 4
CE95389	UART Transmit with PSoC 3/4/5LP
CE95388	UART Receive with PSoC 3/4/5LP

Table 26. PSoC Creator SCB Component Datasheet

Document Number	Document Name
001-96075	PSoC 4 Serial Communication Block (SCB)

5.7.11 UDB

Universal Digital Blocks (UDBs) are programmable digital blocks that can be used to create custom digital logic apart from the resources that are already present in PSoC. PSoC Creator provides Components for PWM, Counter, UART, SPI, I2S and many other commonly used digital blocks which are made out of UDBs.

Apart from these, the UDBs can also be used to create a state machine or a custom communication protocol. They can also be used to add logic gates to your design like OR, AND, XOR etc.

The PSoC 4 has up to eight UDBs; the UDB array also provides a switched Digital System Interconnect (DSI) fabric that allows signals from peripherals and ports to be routed to and through the UDBs for communication and control.

Table 27. UDB Application Notes

Document Number	Document Name
AN82156	PSoC® 3, PSoC 4, and PSoC 5LP – Designing PSoC Creator™ Components with UDB Datapaths
AN82250	PSoC® 3, PSoC 4, and PSoC 5LP – Implementing Programmable Logic Designs with Verilog
AN60024	PSoC® 3, PSoC 4, PSoC 5LP Switch Debouncer and Glitch Filter
AN62510	Implementing State Machines with PSoC® 3, PSoC 4, and PSoC 5LP

Table 28. UDB Code Examples

Document Number	Document Name
CE95319	Hardware Glitch Filter with PSoC 3/4/5LP
CE95348	Precise Illumination Signal Modulation (PrISM) with PSoC 3/4/5LP
CE95298	Switch Debouncer with PSoC 3/4/5LP
CE95294	7-bit Down Counter with PSoC 3/4/5LP
CE95334	Lookup table (LUT) with PSoC 3/4/5LP
CE95386	TMP05 temperature sensor interface with PSoC 3/4/5LP

5.7.12 CAN

The PSoC 4 CAN peripheral is a fully functional Controller Area Network (CAN) supporting communication baud rates up to 1 Mbps. There are two independent CAN 2.0B blocks in PSoC 4200M and PSoC 4200L, which are certified CAN conformant.

Table 29. CAN Code Examples

Document Number	Document Name
CE97311	PSoC® 4 M: CAN Simplex Communication with CapSense®

Table 30. PSoC Creator CAN Component Datasheet

Document Number	Document Name
001-85034	Controller Area Network (CAN)

5.7.13 Universal Serial Bus (USB)

PSoC 4 has a Full-speed USB 2.0 compliant device interface. It has one control endpoint and eight other endpoints. The interface has a USB transceiver and can be operated form the IO obviating the need for a crystal oscillator.

5.7.14 Segment LCD Drive

PSoC 4 has an LCD controller, which can drive up to eight commons and up to 56 segments. Any pin can be either a common or a segment pin. It uses full digital methods to drive the LCD segments requiring no generation of internal LCD voltages.

Table 31. Segment LCD Application Note

Document Number	Document Name
AN87391	PSoC® 4 Segment LCD Direct Drive

Table 32. Segment LCD Code Examples

Document Number	Document Name
CE95369	Segment LCD with PSoC 4
001-88604	PSoC 4 Segment LCD (SegLCD)

5.7.15 GPIOs

PSoC 4 has upto 98 GPIOs in the 124-ball VFBGA package. The pins are organized in logical entities called ports, which are 8-bit in width.

Table 33. GPIO Application Notes

Document Number	Document Name
AN86439	PSoC® 4 – Using GPIO Pins

Table 34. PSoC Creator Pins Component Datasheet

Document Number	Document Name
001-92674	Pins

5.7.16 Smart I/O™

PSoC 4 has a Smart I/O Component associated with some of its ports and provides a programmable logic fabric interposed between a GPIO port and the connections to it from various peripherals and sources. Smart I/O is a fabric of switches and LUTs that allows Boolean functions to be performed on signals being routed to the pins of a GPIO port. The Smart I/O Component should be used whenever simple logic operations and routing are required to be performed on signals going to and coming from I/O pins. Each Smart I/O Component is associated with a particular GPIO port and consumes the port entirely. If the Component is not used, then the Smart I/O functionality for that port is bypassed.

Table 35. Smart I/O Code Examples

Document Number	Document Name
CE209974	PSoC® 4 Breathing LED with Smart I/O
CE209975	PSoC® 4 Clock Buffer with Smart I/O
CE209976	PSoC® 4 SPI Slave Select Inversion with Smart I/O

5.7.17 Program and Debug

The PSoC 4 Program and Debug interface provides a communication gateway for an external device to perform programming or debugging. The external device can be a Cypress-supplied programmer and debugger, or a third-party device that supports PSoC 4 programming and debugging. The serial wire debug (SWD) interface is used as the communication protocol between the external device and PSoC 4.

Table 36. Program and Debug Application Notes

Document Number	Document Name
AN84858	PSoC® 4 Programming Using an External Microcontroller (HSSP)
KBA93541	Using the CY8CKIT-049 to Program Another PSoC 4

5.7.18 System Level Application Notes and Code Examples

Table 37. System Design Application Notes

Document Number	Document Name
AN57821	PSoC® 3, PSoC 4, and PSoC 5LP Mixed-Signal Circuit Board Layout Considerations
AN81623	PSoC® 3, PSoC 4, and PSoC 5LP Digital Design Best Practices
AN88619	PSoC® 4 Hardware Design Considerations
AN2155	PSoC® EMI Design Considerations
AN89056	PSoC® 4 – IEC 60730 Class B and IEC 61508 SIL Safety Software Library
AN80994	PSoC® 3, PSoC 4, and PSoC 5LP EMC Best Practices and Recommendations

Table 38. Fan Controller Application Note

Document Number	Document Name
AN89346	PSoC® 4 Intelligent Fan Controller

Table 39. Bootloader Application Notes

Document Number	Document Name
AN68272	PSoC® 3, PSoC 4, and PSoC 5LP UART Bootloader
AN73854	PSoC® 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders
AN86526	PSoC® 4 I2C Bootloader

Table 40. Character LCD Code Examples

Document Number	Document Name
CE95291	Char LCD using Horizontal Bar Graph (Hbar) with PSoC 4
CE95290	Char LCD using Custom Font with PSoC 4

Table 41. Fan Controller Code Examples

Document Number	Document Name
CE95318	Firmware Fan Control with PSoC 3/4/5LP
CE95315	Auto Firmware Fan Control with Alert with PSoC 4

Table 42. Bootloader Code Examples

Document Number	Document Name
CE95281	Bootloader Project with PSoC 4
CE95280	Bootloadable Project with PSoC 4

Table 43. Miscellaneous Code Examples

Document Number	Document Name
CE95329	Compensation of ILO Trimming with PSoC 4
CE95296	Cyclic Redundancy Check with PSoC 3/4/5LP
CE95379	Software based UART transmit with PSoC 3/4/5LP

Table 44. PSoC Creator System Level Component Datasheets

Document Number	Document Name
001-92648	Bootloader and Bootloadable
001-85137	Interrupt
001-96071	PSoC® Creator™ System Reference Guide (CY_Boot Component)

6 Summary

This application note explored the PSoC 4 architecture and development tools. PSoC 4 is a truly programmable embedded system-on-chip, integrating configurable analog and digital peripheral functions, memory, and an Arm Cortex-M0/M0+ microcontroller on a single chip. Because of the integrated features and low-leakage power modes, PSoC 4 is an ideal choice for low-power and cost-effective embedded systems.

This application note also guided you to a comprehensive collection of resources to accelerate in-depth learning about PSoC 4.

About the Authors

Name: Nidhin M S

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.

Name: Ranjith M

Title: Applications Engineer Sr.

Background: Ranjith graduated from GEC Thrissur with a Bachelor's degree in Electronics and Communication Engineering. His technical interests are communication protocols, digital logic design, and device programming.

Document History

Document Title: AN79953 – Getting Started with PSoC 4

Document Number: 001-79953

Revision	ECN	Submission Date	Description of Change
**	3881879	01/24/2013	New Application Note
*A	3968932	04/11/2013	Demo project changed to leverage Pioneer kit. Added architecture introduction.
*B	3996226	05/09/2013	Reformatted graphics. Updated links.
*C	4219723	12/19/2013	Updated attached Associated Project files. Updated content across the entire document. Updated in new template.
*G	4339565	04/10/2014	Updated the projects and the respective section in the AN to support PSoC. Creator 3.0 SP1 and PSoC 4000 device.
*H	4514729	09/25/2014	Added Code Examples section. Minor edits and format changes throughout the document.
*I	4679544	03/17/2015	Added More Information section. Removed detailed feature descriptions. Updated for PSoC 4200M family of devices.
*J	4826009	09/10/2015	Updated PSoC Resources and PSoC is More than an MCU. Added the following sections: Convert Project to Bootloadable for CY8CKIT-049, Bootload Your CY8CKIT-049 and More PSoC 4 Code Examples. Updated Figure 4.
*K	4922589	09/16/2015	Updated for PSoC 4200L. Updated the example projects to PSoC Creator 3.3.
*L	5068160	12/30/2015	Updated the example projects to PSoC Creator 3.3 SP1.
*M	5127180	02/05/2016	Updated for PSoC 4000S and PSoC 4100S. Updated the example projects to PSoC Creator 3.3 SP2. Updated Table 2, Table 3 to add PSoC 4200 BLE.
*N	5687926	04/19/2017	Updated logo and copyright
*O	5898247	10/06/2017	Added references to PSoC 4100S Plus throughout the document. Updated Table 1 to add PSoC 4100S Plus. Updated PSoC Resources with the reference of AN64846 - Getting Started with CapSense®.
*P	5943836	11/03/2017	Updated the example projects to PSoC Creator 4.2 Updated Table 1 to add supported kit Updated Table 2 to add CY8CKIT-49, CY8CKIT-145 and CY8CKIT-149 Added Table 4 to include CY8CKIT-145 and CY8CKIT-149 pin mapping Updated Figure 2 Added example project of CY8CKIT-145 and CY8CKIT-149 as a part of the AN79953.zip file in this application note landing page.
*Q	6079643	03/06/2018	Updated template Updated for PSoC 4100PS Minor edits and format changes throughout the document Added example project of CY8CKIT-147 as a part of the AN79953.zip file
*R	6165575	05/04/2018	Updated template Corrected the link to PSoC 4100S Plus in PSoC Resources
*S	6669181	09/06/2019	Updated Table 1 to add PSoC 4500 and update the specifications of PSoC 4100S Plus device family. Added Motor Control Accelerator feature.

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

Community | 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
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2013-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("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 shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. 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. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.
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.