Getting Started with PSoC® 4

Author: Nidhin M S and Ranjith M

Associated Part Family: All PSoC 4 Parts

Related Application Notes: AN54181, AN77759

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

AN79953 introduces you to PSoC® 4, an ARM® Cortex®-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

PSoC 4 is a true programmable embedded system-on-chip, integrating custom analog and digital peripheral functions, memory, and an ARM 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 circuits (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.

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 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:

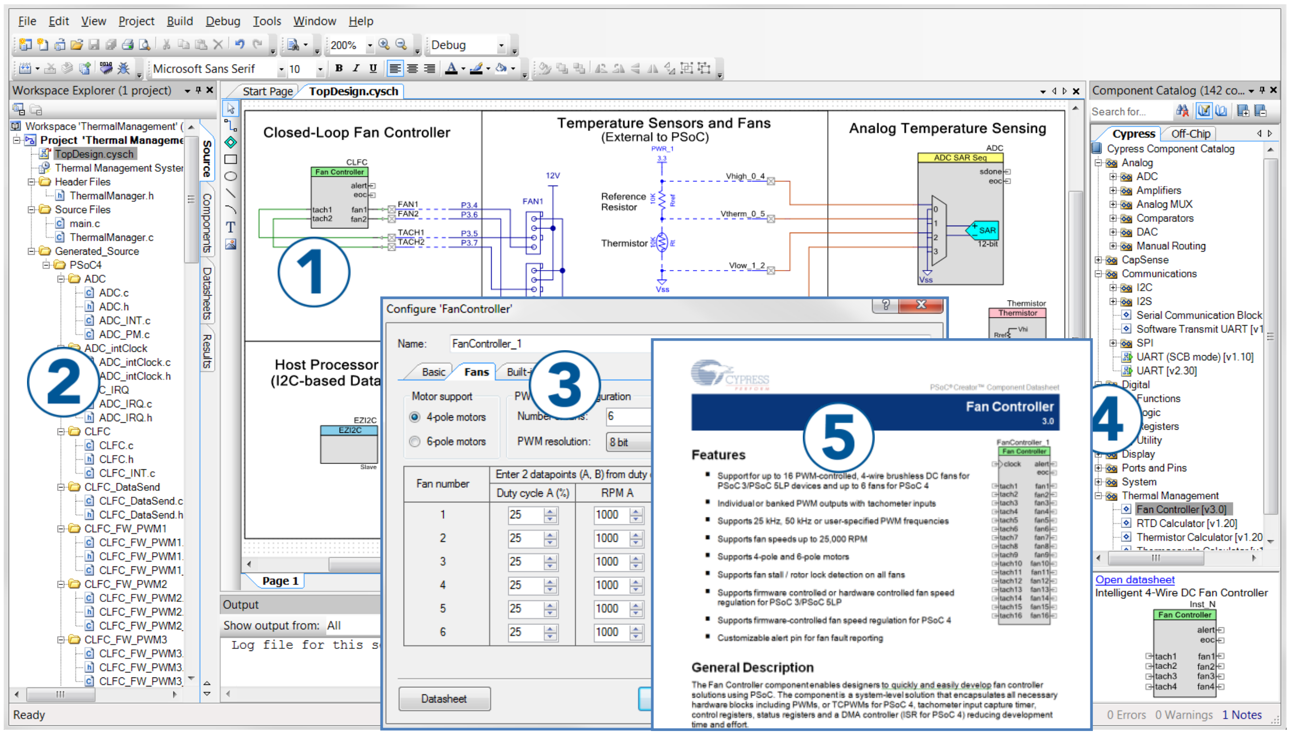
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets**: Describe and provide electrical specifications for the PSoC 4000, PSoC 4100, and PSoC 4200, PSoC 4xx7 BLE, PSoC 4200M device families
- **CapSense Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 4 family 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. PSoC Creator provides additional 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 build complex designs with PSoC.
- **Development Kits:**
- CY8CKIT-040, CY8CKIT-042, and CY8CKIT-044 PSoC 4 Pioneer Kits are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
- CY8CKIT-049 is a very low-cost prototyping platform for sampling PSoC 4200 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 systems based on PSoC 3, PSoC 4, and PSoC 5LP. See Figure 1 – with PSoC Creator, you can:

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

Figure 1. PSoC Creator Features


File Edit View Project Build Debug Tools Window Help
Workspace Explorer (1 project)
Start Page
TopDesign.cysch
Closed-Loop Fan Controller
Temperature Sensors and Fans (External to PSoC)
Analog Temperature Sensing
Host Processor (I2C-based Data
Configure 'FanController'
Name: FanController_1
Basic
Fans
Fan Controller
Features
General Description
Component Catalog
Output
1
2
3
4
5
Ready

2.2 Code Examples

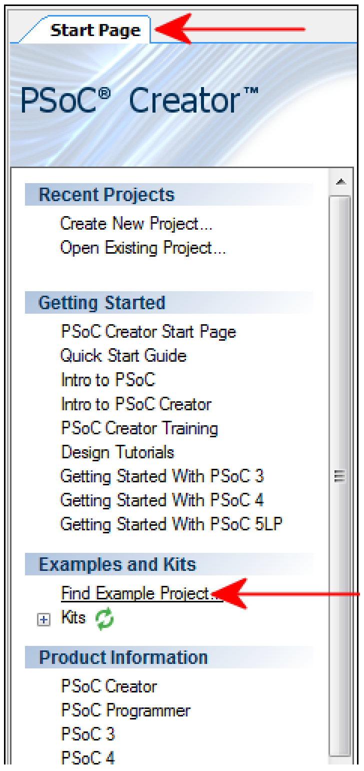
PSoC Creator includes a large number of code example projects. These projects are available from the PSoC Creator Start Page, as shown in Figure 2.

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

Figure 2. Code Examples in PSoC Creator


Start Page

PSoC® Creator™

Recent Projects

Create New Project...
Open Existing Project...

Getting Started

PSoC Creator Start Page
Quick Start Guide
Intro to PSoC
Intro to PSoC Creator
PSoC Creator Training
Design Tutorials
Getting Started With PSoC 3
Getting Started With PSoC 4
Getting Started With PSoC 5LP

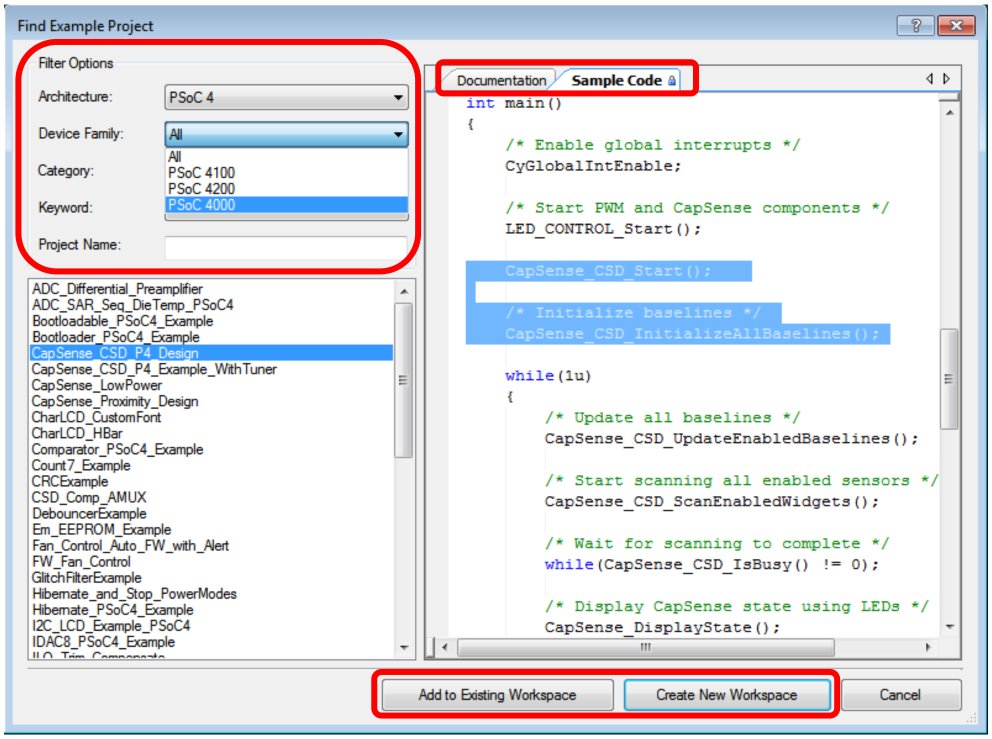
Examples and Kits

Find Example Project...
Kits

Product Information

PSoC Creator
PSoC Programmer
PSoC 3
PSoC 4

Figure 3. Code Example Projects with Sample Code



Find Example Project

Filter Options

Architecture: PSoC 4

Device Family: All

All
PSoC 4100
PSoC 4200
PSoC 4000

Category:

Keyword:

Project Name:

ADC_Differential_Preamplifier
ADC_SAR_Seq_DieTemp_PSoC4
Bootloadable_PSoC4_Example
Bootloader_PSoC4_Example
CapSense_CSD_P4_Design
CapSense_CSD_P4_Example_WithTuner
CapSense_LowPower
CapSense_Proximity_Design
CharLCD_CustomFont
CharLCD_HBar
Comparator_PSoC4_Example
Count7_Example
CRCExample
CSD_Comp_AMUX
DebouncerExample
Em_EEPROM_Example
Fan_Control_Auto_FW_with_Alert
FW_Fan_Control
GlitchFilterExample
Hibernate_and_Stop_PowerModes
Hibernate_PSoC4_Example
I2C_LCD_Example_PSoC4
IDAC8_PSoC4_Example

Documentation | Sample Code

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

    /* Start PWM and CapSense components */
    LED_CONTROL_Start();

    CapSense_CSD_Start();

    /* Initialize baselines */
    CapSense_CSD_InitializeAllBaselines();

    while(1u)
    {
        /* Update all baselines */
        CapSense_CSD_UpdateEnabledBaselines();

        /* Start scanning all enabled sensors */
        CapSense_CSD_ScanEnabledWidgets();

        /* Wait for scanning to complete */
        while(CapSense_CSD_IsBusy() != 0);

        /* Display CapSense state using LEDs */
        CapSense_DisplayState();
```

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.
- **Starter designs**: Choose **File > New** > **Project** > **PSoC 4 Starter Designs**. These starter designs demonstrate the unique features of PSoC 4.
- **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 4200-M)

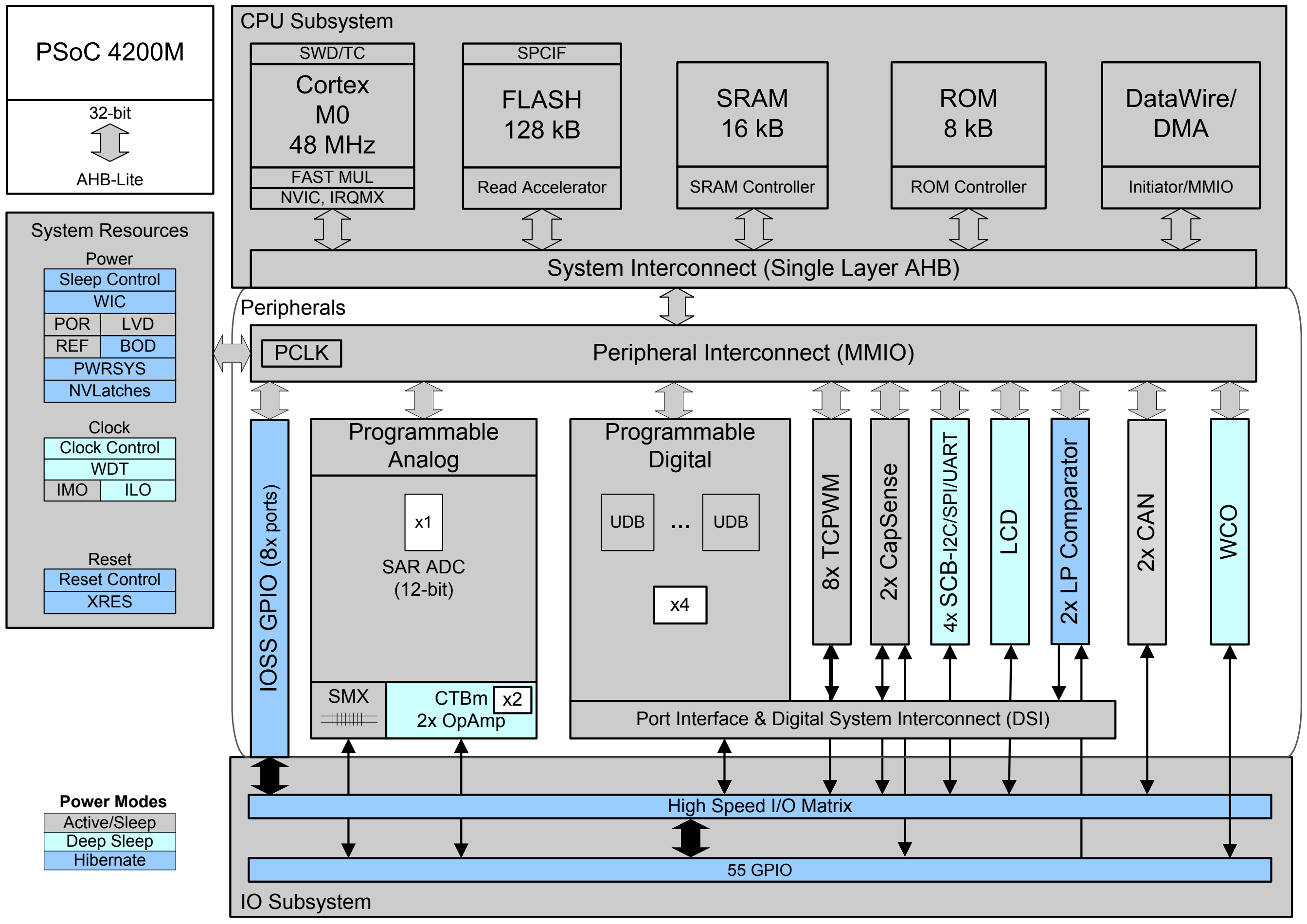


Table 1 shows the features available for the biggest device in different families. Depending on the device, all or a subset of these features may be available. Refer to the PSoC 4 Product Selector Guide for details.

Table 1.PSoC 4 Device Families

Features	Device Family			
	CY8C4000	**CY8C4100**	**CY8C4200**	**CY8C4200M**
CPU	16-MHz Cortex-M0	24-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	Eight channels
Flash memory	16 KB	32 KB	32 KB	128 KB
SRAM	2 KB	4 KB	4 KB	16 KB
GPIOs	20	36	36	55
CapSense	16 sensors	35 sensors	35 sensors	54 sensors
ADC	None	12-bit, 806-ksps SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer	12-bit, 1-MSPS SAR ADC with sequencer
Opamps	None	Two programmable opamps	Two programmable opamps	Two programmable opamps
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
IDACs*	One 7-bit and one 8-bit	One 7-bit and one 8-bit	One 7-bit and one 8-bit	Two 7-bit and two 8-bit
Programmable logic blocks (UDBs)	None	None	Four UDBs, each with eight macrocells and one datapath	Four UDBs, each with eight macrocells and one datapath
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
Low-power modes	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
Segment LCD drive	None	4 COM segment LCD drive	4 COM segment LCD drive	4 COM segment LCD drive
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	Two SCBs with programmable I^2C, SPI, or UART
Timer Counter Pulse-Width Modulator (TCPWM)	1	4	4	8
Controller Area Network (CAN)	None	None	None	2
Clocks	24-MHz / 32-MHz internal main oscillator (IMO) 32-kHz internal low-speed oscillator (ILO)	3-MHz to 24-MHz IMO 32-kHz ILO	3-MHz to 48-MHz IMO 32-kHz ILO	3-MHz to 48-MHz IMO 32-kHz ILO 32-kHZ watch crystal oscillator (WCO)
Power supply monitoring	Power-on reset (POR) Brown-out detection (BOD)	POR, BOD, Low-voltage detection (LVD)	POR, BOD, LVD	POR, BOD, LVD

*IDACs are available only when CapSense is not in use. Refer to the respective PSoC 4 Technical Reference Manual for more details.

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

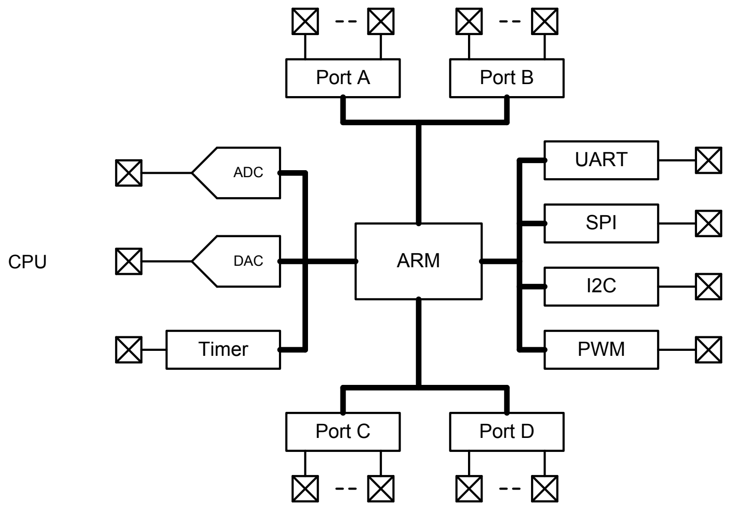


Figure 6. PSoC Block Diagram

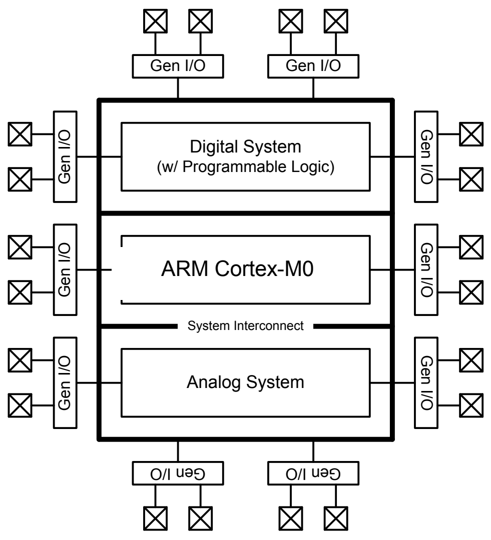


Figure 7. Digital System of PSoC 4

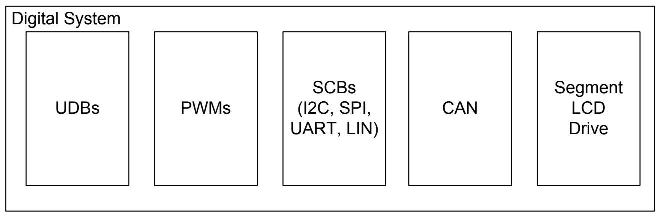
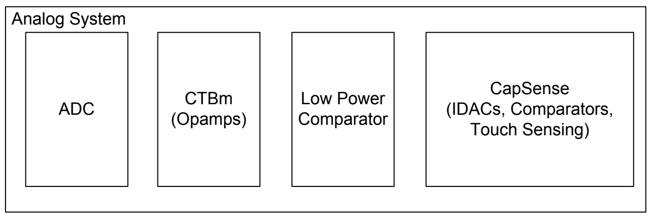


Figure 8. Analog System of PSoC 4



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 out of UDBs like UART, SPI, I2S, Timer, PWM, Counter, Digital Gates (AND, OR, NOT, XOR etc), 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 the PSoC Creator IDE

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?

Testing this design requires CY8CKIT-040, CY8CKIT-042, or CY8CKIT-044, which has an integrated programmer.

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

Pioneer Kit	Supported Device Family	Part Number
CY8CKIT-040	PSoC 4000	CY8C4014LQI-422
CY8CKIT-042	PSoC 4200	CY8C4245AXI-483
CY8CKIT-044	PSoC 4200M	CY8C4247AZI-M485

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.

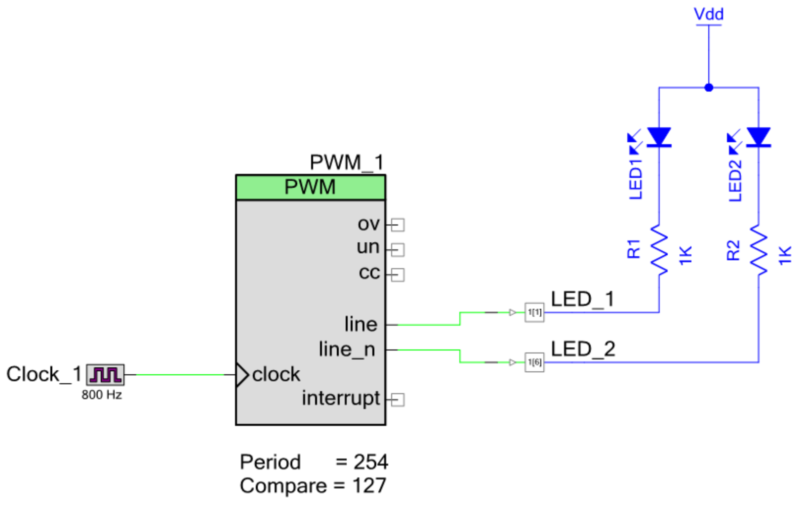
5.1.3 Want to see the project in action?

If you don’t 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 9. 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 9. My First PSoC 4 Design

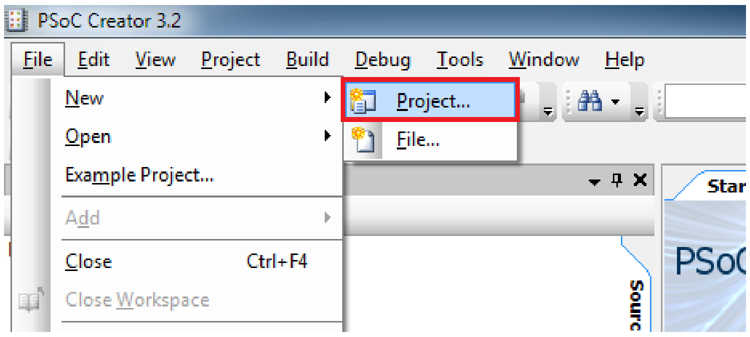


5.3 Part 1: Create the Design

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.

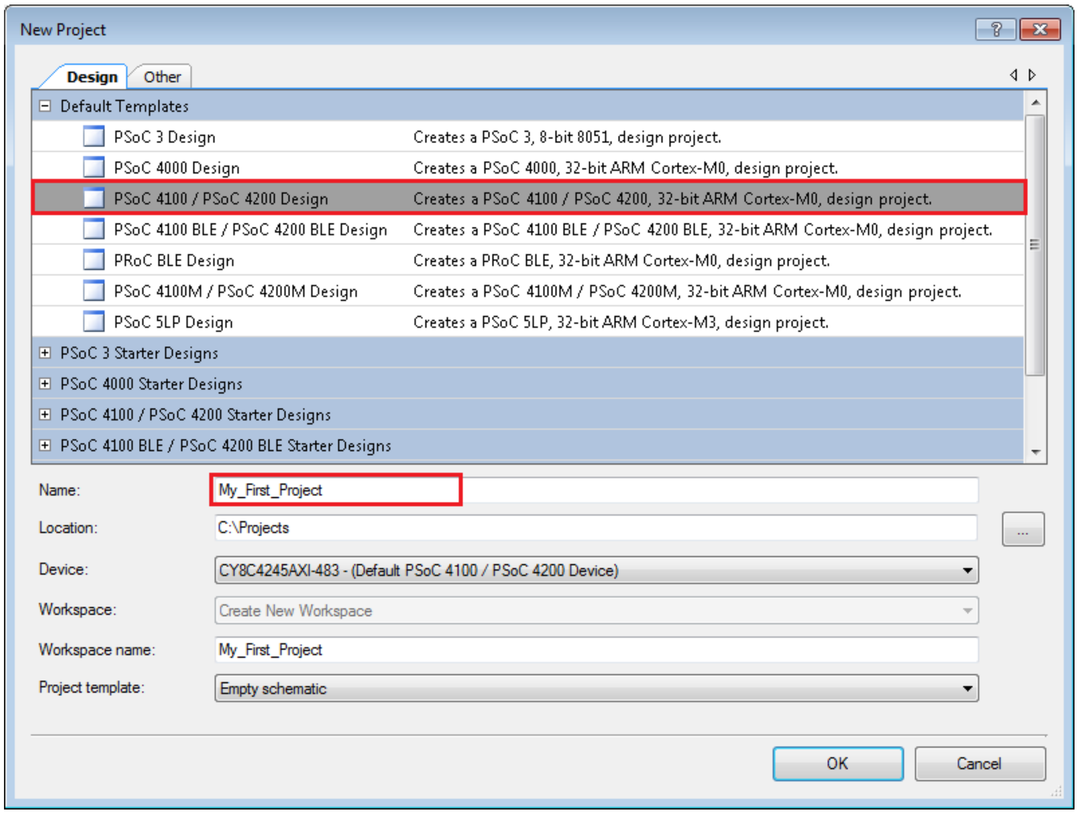
1. Start PSoC Creator, and from the **File** menu choose **New** > **Project**, as shown in Figure 10.

Figure 10. Creating a New Project



2. Select the PSoC 4 design according to your development kit. For example, if you have a CY8CKIT-042, select **PSoC 4100 / PSoC 4200 Design**, and provide a project name (for example, “My_First_Project”) as shown in Figure 11. See Table 2 for a list of PSoC 4 kits and associated devices. Choose an appropriate location for your new project, and click **OK**.

Figure 11. Create a New Empty PSoC 4 Project

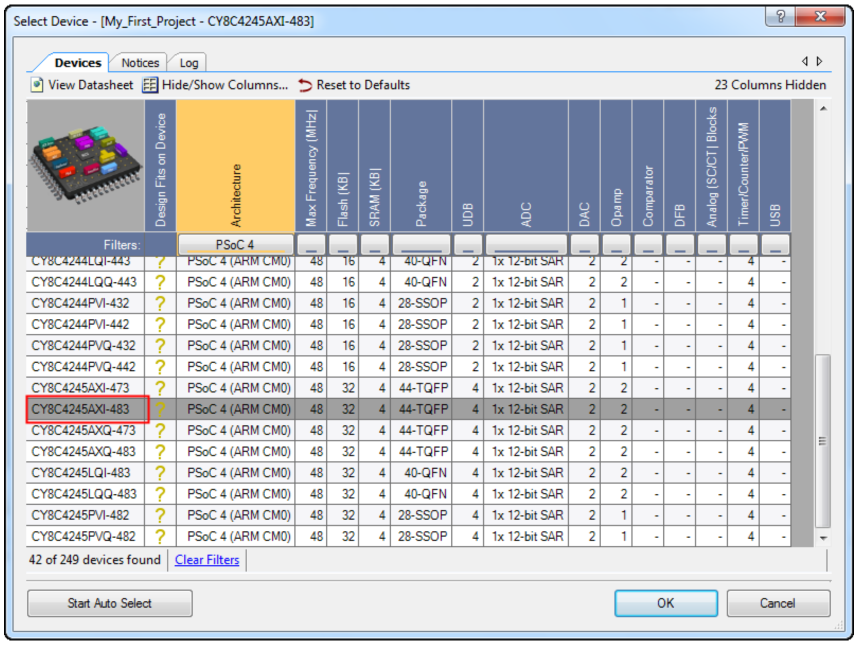


Default Templates	
PSoC 3 Design	Creates a PSoC 3, 8-bit 8051, design project.
PSoC 4000 Design	Creates a PSoC 4000, 32-bit ARM Cortex-M0, design project.
PSoC 4100 / PSoC 4200 Design	Creates a PSoC 4100 / PSoC 4200, 32-bit ARM Cortex-M0, design project.
PSoC 4100 BLE / PSoC 4200 BLE Design	Creates a PSoC 4100 BLE / PSoC 4200 BLE, 32-bit ARM Cortex-M0, design project.
PRoC BLE Design	Creates a PRoC BLE, 32-bit ARM Cortex-M0, design project.
PSoC 4100M / PSoC 4200M Design	Creates a PSoC 4100M / PSoC 4200M, 32-bit ARM Cortex-M0, design project.
PSoC 5LP Design	Creates a PSoC 5LP, 32-bit ARM Cortex-M3, design project.
PSoC 3 Starter Designs	
PSoC 4000 Starter Designs	
PSoC 4100 / PSoC 4200 Starter Designs	
PSoC 4100 BLE / PSoC 4200 BLE Starter Designs	

Name:	My_First_Project
Location:	C:\Projects
Device:	CY8C4245AXI-483 - (Default PSoC 4100 / PSoC 4200 Device)
Workspace:	Create New Workspace
Workspace name:	My_First_Project
Project template:	Empty schematic

3. Select the PSoC 4 device that you want to use. Go to **Project** > **Device Selector** and select the device. If you are using a development kit, read the part number from the kit or refer to the kit user guide for the part number. Figure 12 shows an example selection for the CY8CKIT-042 PSoC 4 Pioneer Kit.

Figure 12. Device Selection

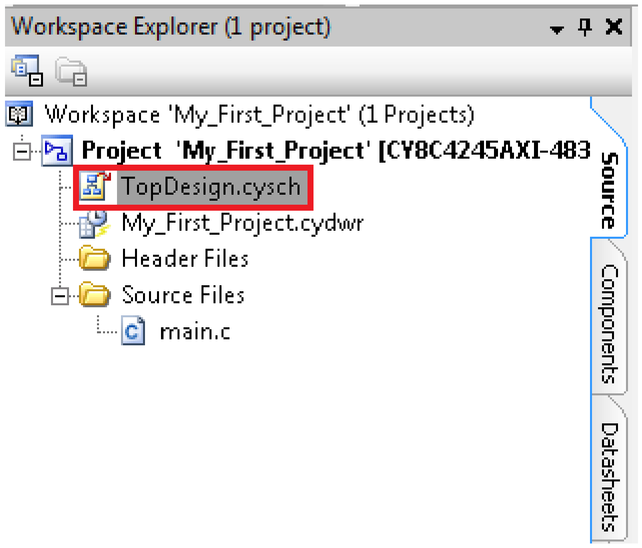


	Design Fits on Device	Architecture	Max Frequency (MHz)	Flash (KB)	SRAM (KB)	Package	UDB	ADC	DAC	Opamp	Comparator	DFB	Analog (SC/CT) Blocks	Timer/Counter/PWM	USB
Filters:		PSoC 4													
CY8C4244LQI-443	?	PSoC 4 (ARM CM0)	48	16	4	40-QFN	2	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4244LQQ-443	?	PSoC 4 (ARM CM0)	48	16	4	40-QFN	2	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4244PVI-432	?	PSoC 4 (ARM CM0)	48	16	4	28-SSOP	2	1x 12-bit SAR	2	1	-	-	-	4	-
CY8C4244PVI-442	?	PSoC 4 (ARM CM0)	48	16	4	28-SSOP	2	1x 12-bit SAR	2	1	-	-	-	4	-
CY8C4244PVQ-432	?	PSoC 4 (ARM CM0)	48	16	4	28-SSOP	2	1x 12-bit SAR	2	1	-	-	-	4	-
CY8C4244PVQ-442	?	PSoC 4 (ARM CM0)	48	16	4	28-SSOP	2	1x 12-bit SAR	2	1	-	-	-	4	-
CY8C4245AXI-473	?	PSoC 4 (ARM CM0)	48	32	4	44-TQFP	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245AXI-483	?	PSoC 4 (ARM CM0)	48	32	4	44-TQFP	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245AXQ-473	?	PSoC 4 (ARM CM0)	48	32	4	44-TQFP	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245AXQ-483	?	PSoC 4 (ARM CM0)	48	32	4	44-TQFP	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245LQI-483	?	PSoC 4 (ARM CM0)	48	32	4	40-QFN	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245LQQ-483	?	PSoC 4 (ARM CM0)	48	32	4	40-QFN	4	1x 12-bit SAR	2	2	-	-	-	4	-
CY8C4245PVI-482	?	PSoC 4 (ARM CM0)	48	32	4	28-SSOP	4	1x 12-bit SAR	2	1	-	-	-	4	-
CY8C4245PVQ-482	?	PSoC 4 (ARM CM0)	48	32	4	28-SSOP	4	1x 12-bit SAR	2	1	-	-	-	4	-

42 of 249 devices found | Clear Filters

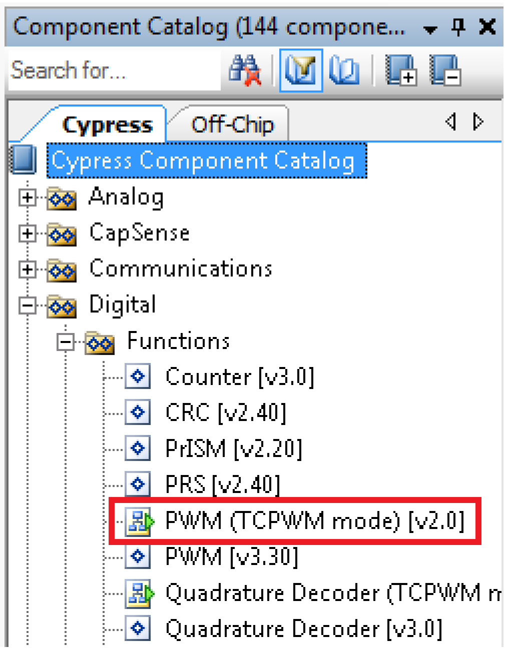
4. Creating a new project generates a project folder with a baseline set of files shown in the **Workspace Explorer** (see Figure 13). To open the project schematic file, double-click *TopDesign.cysch*.

Figure 13. Opening TopDesign Schematic


Workspace Explorer (1 project)
Workspace 'My_First_Project' (1 Projects)
Project 'My_First_Project' [CY8C4245AXI-483
TopDesign.cysch
My_First_Project.cydwr
Header Files
Source Files
main.c
Source
Components
Datasheets

5. Drag one **PWM (TCPWM mode)** Component from the **Component Catalog** onto the schematic, as shown in Figure 14.

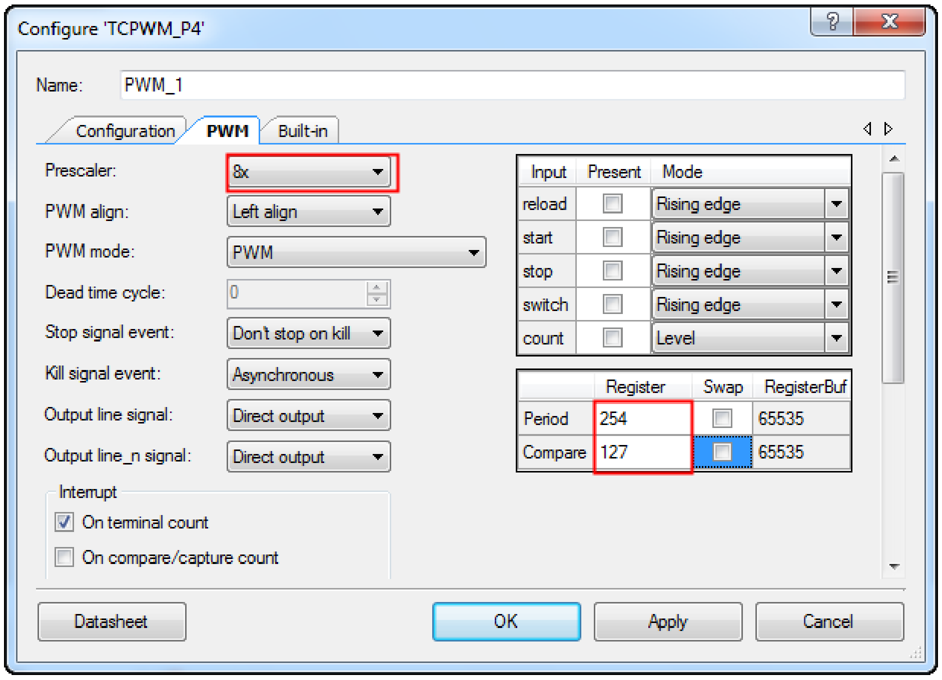
Figure 14. Location of the PWM Component



6. Double-click the PWM Component on the schematic to configure the Component properties, as shown in Figure 15. 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 15. Configuring the PWM Component



Configure 'TCPWM_P4'

Name: PWM_1

Configuration | **PWM** | Built-in

Prescaler: 8x
PWM align: Left align
PWM mode: PWM
Dead time cycle: 0
Stop signal event: Don't stop on kill
Kill signal event: Asynchronous
Output line signal: Direct output
Output line_n signal: Direct output

Interrupt
[✓] On terminal count
[ ] On compare/capture count

Input	Present	Mode
reload	☐	Rising edge
start	☐	Rising edge
stop	☐	Rising edge
switch	☐	Rising edge
count	☐	Level

	Register	Swap	RegisterBuf
Period	254	☐	65535
Compare	127	☐	65535

7. 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 16 and Figure 17.

   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 16. Location of the Clock Component

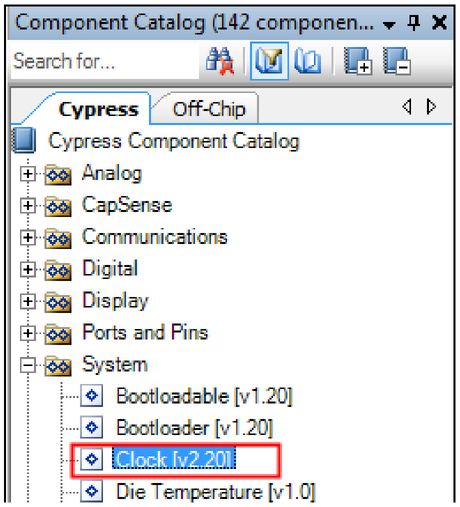
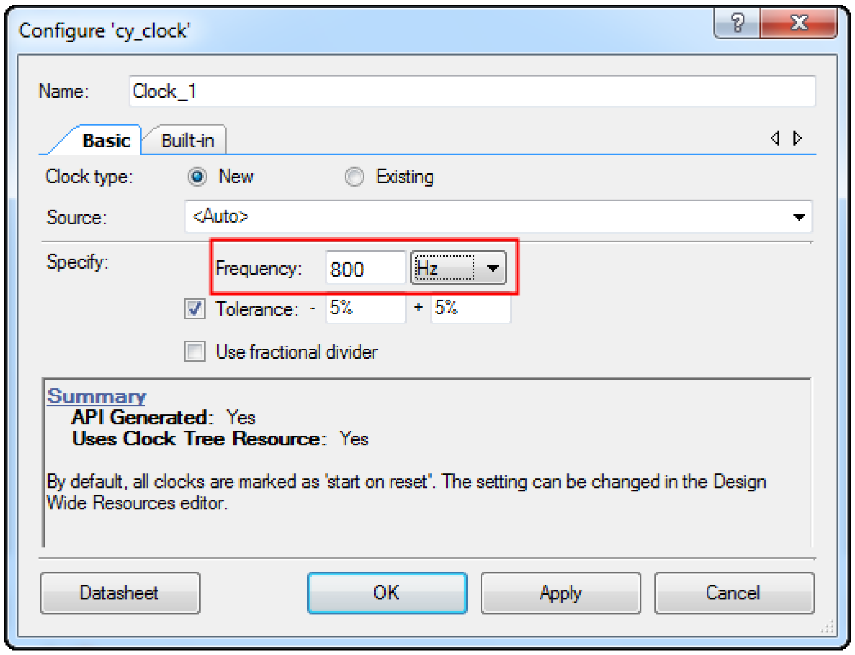


Figure 17. Configuring the Clock Component


Configure 'cy_clock'

Name: Clock_1

Basic | Built-in

Clock type: [✓ New] Existing

Source: <Auto>

Specify: Frequency: 800 Hz

[✓] Tolerance: - 5% + 5%

[ ] Use fractional divider

Summary
API Generated: Yes
Uses Clock Tree Resource: Yes

By default, all clocks are marked as 'start on reset'. The setting can be changed in the Design Wide Resources editor.

8. Drag and drop a **Digital Output Pin** Component. Change the name to LED_1 as shown in Figure 18 and Figure 19. Add another Digital Output Pin Component and change its name to LED_2.

Figure 18. Location of the Digital Output Pin Component

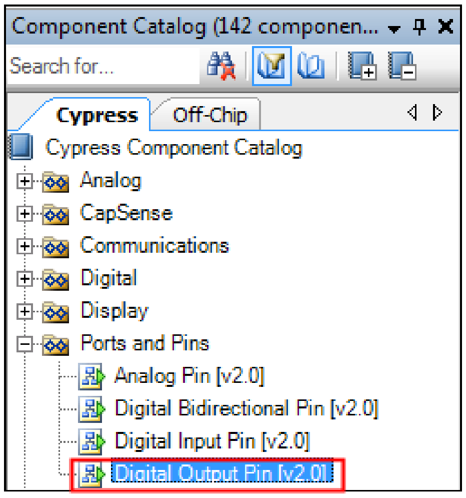
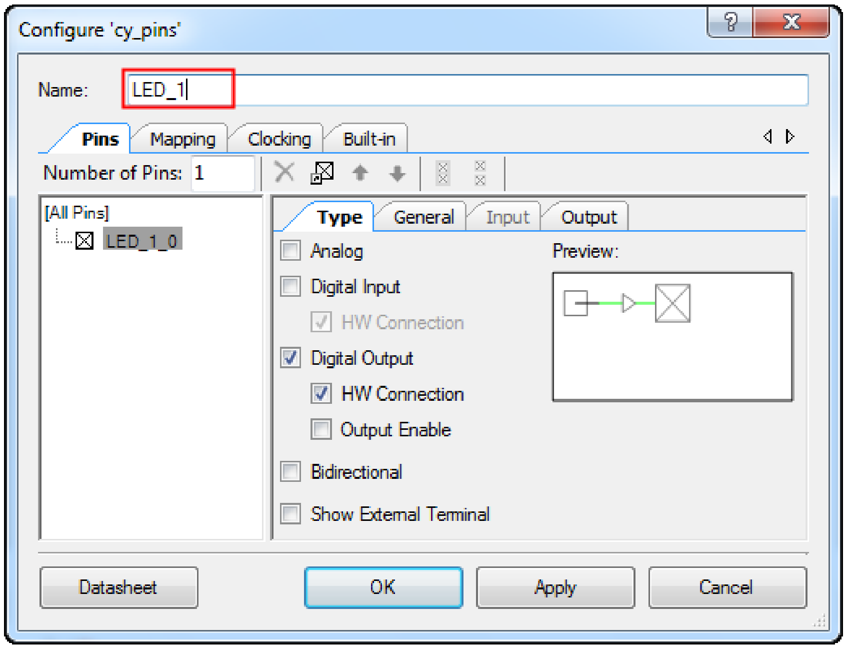
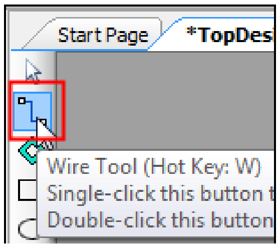


Figure 19. Renaming a Pin Component



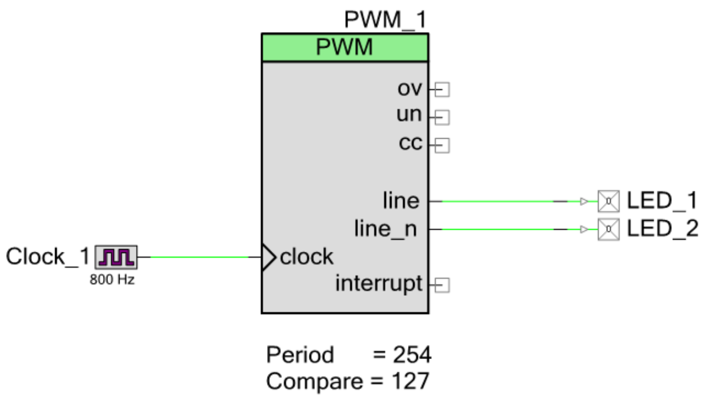
9. In the schematic window, select the wire tool as shown in Figure 20, or press **W**.

Figure 20. Selecting the Wire Tool



10. Wire the Components together, as shown in Figure 21.

Figure 21. Wiring the Schematic



11. 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

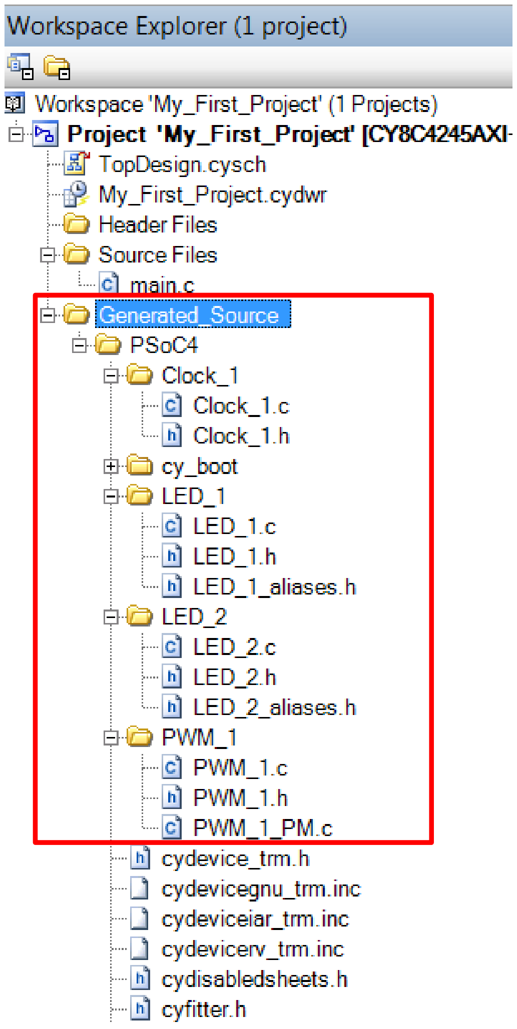
```
int main()
{
  /* Enable and start the PWM */
  PWM_1_Start();

  for(;;)
  {

  }
}
```

12. 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 22.

Figure 22. Generated Source Files

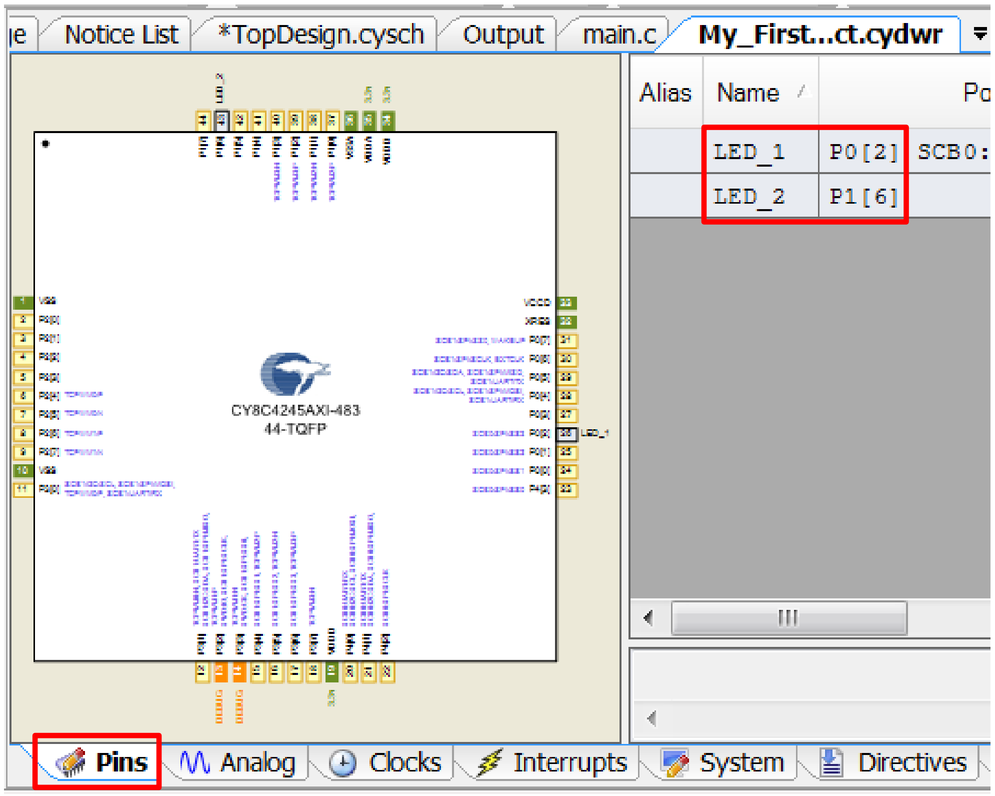


```
Workspace Explorer (1 project)
Workspace 'My_First_Project' (1 Projects)
Project 'My_First_Project' [CY8C4245AXI-
  TopDesign.cysch
  My_First_Project.cydwr
  Header Files
  Source Files
    main.c
  Generated_Source
    PSoC4
      Clock_1
        Clock_1.c
        Clock_1.h
      cy_boot
      LED_1
        LED_1.c
        LED_1.h
        LED_1_aliases.h
      LED_2
        LED_2.c
        LED_2.h
        LED_2_aliases.h
      PWM_1
        PWM_1.c
        PWM_1.h
        PWM_1_PM.c
      cydevice_trm.h
      cydevicegnu_trm.inc
      cydeviceiar_trm.inc
      cydevicerv_trm.inc
      cydisabledsheets.h
      cyfitter.h
```

13. 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 23 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 kit.

Figure 23. Pin Selection



Alias	Name	Po
	LED_1	P0[2] SCB0:
	LED_2	P1[6]

Table 3. Pin Mapping Table Across Pioneer Kits

Function	CY8CKIT-040 (PSoC 4000)	CY8CKIT-042 (PSoC 4200)	CY8CKIT-044 (PSoC 4200M)
Green LED (Active LOW)	P1[1]	P0[2]	P2[6]
Red LED (Active LOW)	P3[2]*	P1[6]	P0[6]

*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. You can also connect an external LED to P1[6] as LED2.

Note: CY8CKIT-049 has only one LED. If you are using CY8CKIT-049, you can connect an external LED to pin P0[2].

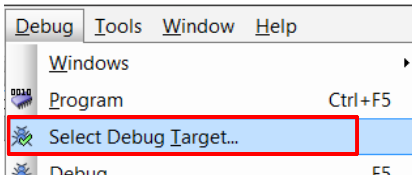
14. Finally, rebuild the project as Step 12 explains.
15. Continue to the next section if you are not using a CY8CKIT-049. If you are using a CY8CKIT-049, navigate to the section Convert Project to Bootloadable for CY8CKIT-049.

5.4 Part 2: Program the Device

This section shows how to program the device. If you are using CY8CKIT-040, CY8CKIT-042 or CY8CKIT-044, connect the kit board to your computer using the USB cable.

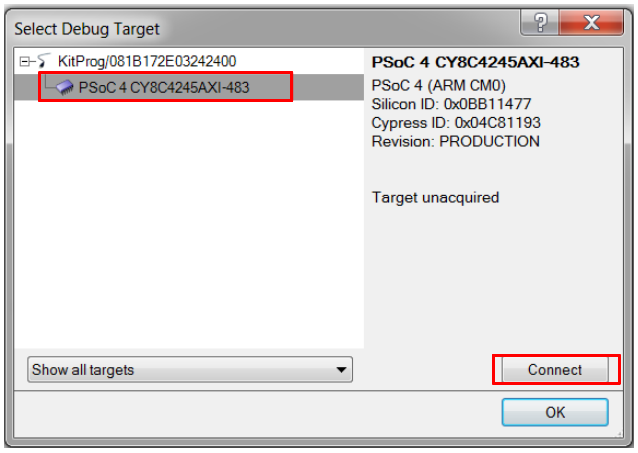
1. Select the PSoC Creator menu item **Debug** > **Select Debug Target**, as shown in Figure 24.

Figure 24. Selecting Debug Target



2. In the **Select Debug Target** dialog box, click **Port Acquire**, and then click **Connect**, as shown in Figure 25. Click **OK** to close the dialog box.

Figure 25. Connecting to a Device

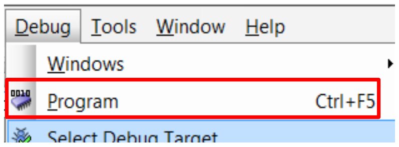


PSoC 4 CY8C4245AXI-483
PSoC 4 (ARM CM0)
Silicon ID: 0x0BB11477
Cypress ID: 0x04C81193
Revision: PRODUCTION

Target unacquired

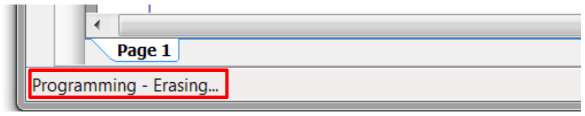
3. Choose the menu item **Debug** > **Program** to program the device with the project, as shown in Figure 26.

Figure 26. Programming the Device



4. You can view the programming status on the status bar (lower-left corner of the window), as shown in Figure 27.

Figure 27. Programming Status

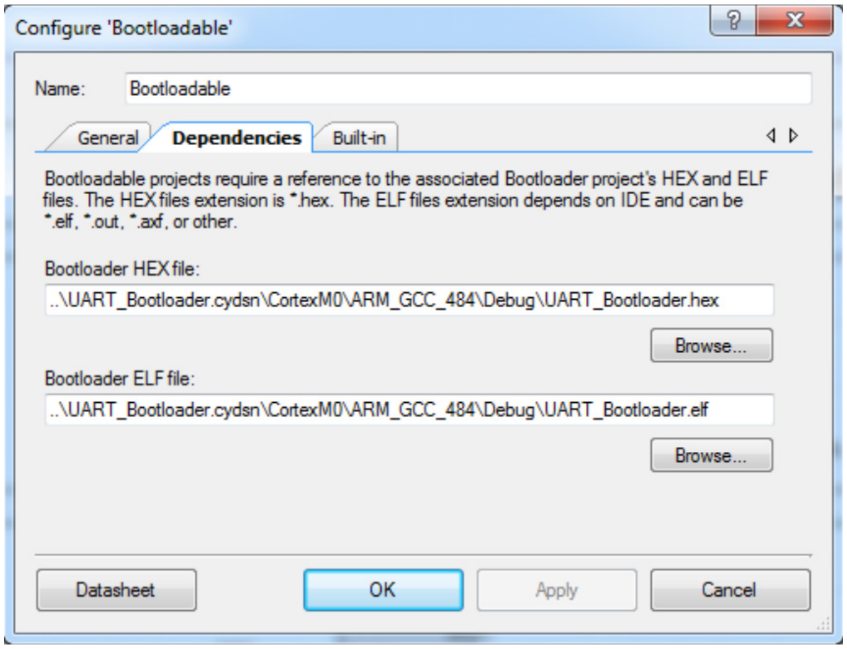


5. After the device is programmed, verify the operation of the project by viewing the toggling of the LEDs.

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 28. Adding UART Bootloader Dependency



Bootloadable projects require a reference to the associated Bootloader project's HEX and ELF files. The HEX files extension is *.hex. The ELF files extension depends on IDE and can be *.elf, *.out, *.axf, or other.

Bootloader HEX file:
..\UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug\UART_Bootloader.hex

Bootloader ELF file:
..\UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug\UART_Bootloader.elf

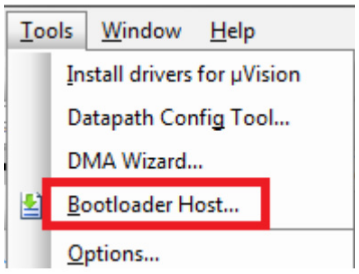
4. Finally, rebuild the project by selecting the option **Build My_First_Project** from the **Build** menu.

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. To bootload the example project provided with this application note to the CY8CKIT-049, perform these steps:

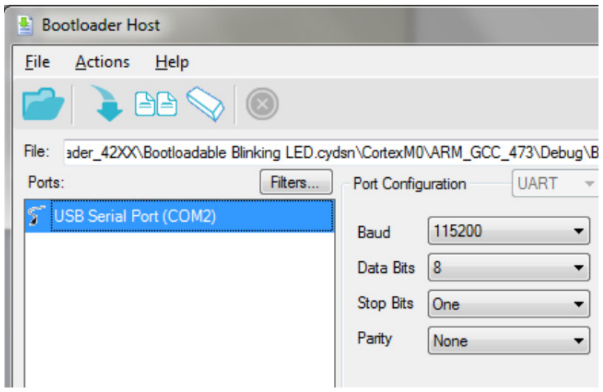
1. Connect the CY8CKIT-049-4xxx prototyping board to the PC. When connecting the kit to the port, depress the SW1 button as it is plugged in. You will notice that the blue LED begins to blink rapidly; this indicates that the PSoC 4 is in 'Bootloader Mode' and is ready to be loaded with the latest firmware.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

Figure 29. Launch Bootloader Host Tool



The Bootloader Host tool opens.

Figure 30. Bootloader Host Tool

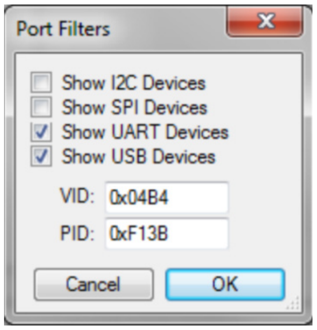


File: ader_42XX\Bootloadable Blinking LED.cydsn\CortexM0\ARM_GCC_473\Debug\B
Ports: USB Serial Port (COM2)
Port Configuration
Baud 115200
Data Bits 8
Stop Bits One
Parity None

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 31. Port Filters

   Show I2C Devices ☐
   Show SPI Devices ☐
   Show UART Devices ✓
   Show USB Devices ✓
   VID: 0x04B4
   PID: 0xF13B

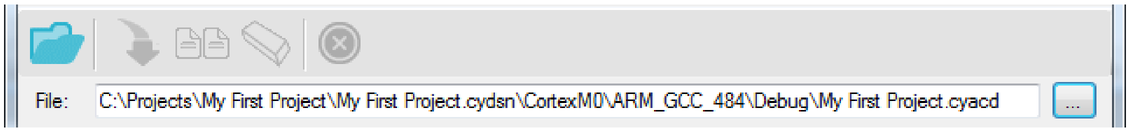
   The Bootloader Host tool will now display all of the available UART based COM ports.

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**.

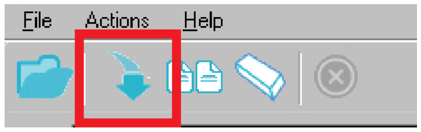
   Figure 32. Opening the Generated File

   File: C:\Projects\My First Project\My First Project.cydsn\CortexM0\ARM_GCC_484\Debug\My First Project.cyacd

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.

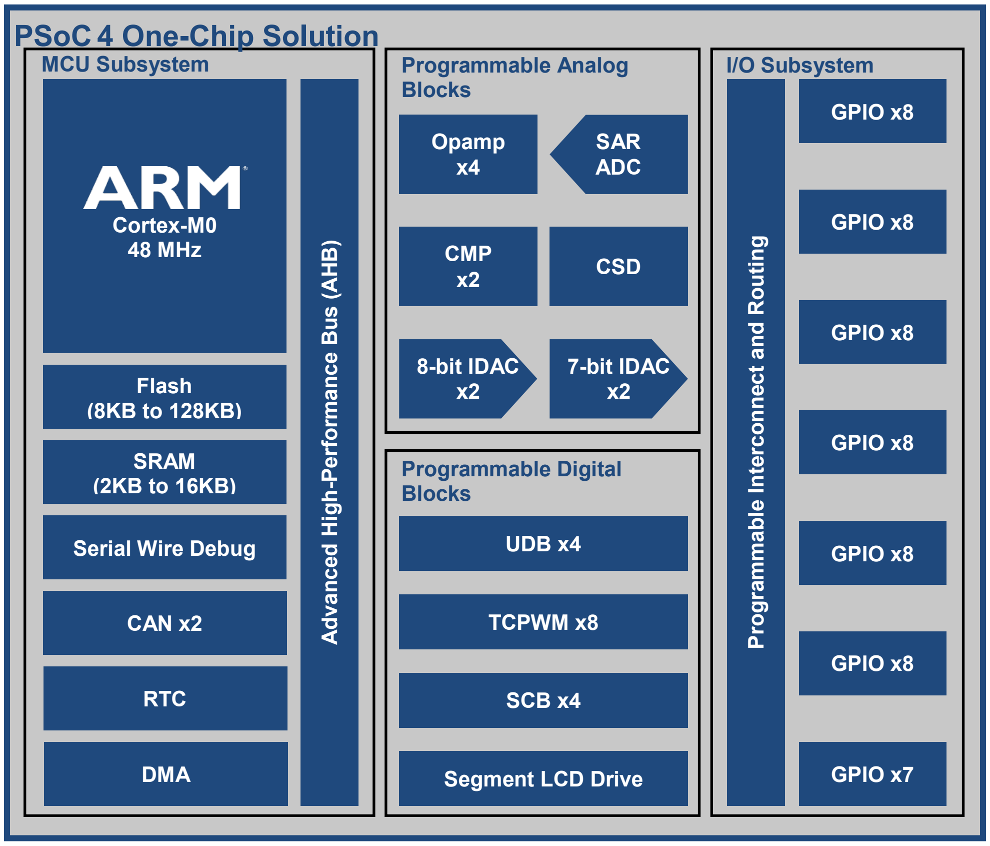
   Figure 33. Program the Device With Application Code

   See AN73854 for additional details on bootloading.

5.7 More PSoC 4 Code Examples

You can find more PSoC 4 code examples in the Cypress website. The following figure shows the different blocks in the largest member of 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

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. The 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.

Table 4. CPU Subsytem 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 5. 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 eight channels, is provided on PSoC 4200M that can do 32-bit transfers between peripherals or memory.

5.7.4 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 4200M 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 6. 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 7. 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 8. 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 9. 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

5.7.5 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 10. SAR ADC Application Note

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

Table 11. 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

5.7.6 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 12. CapSense Application Notes

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

Table 13. 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

5.7.7 Programmable Analog – Opamp and Low Power Comparators

The PSoC 4 has upto 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.

Table 14. Opamp Code Examples

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

Table 15. 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

5.7.8 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 16. TCPWM Application Notes

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

Table 17. 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 3/4/5LP
CE95385	TCPWM Timer/Counter Operation with PSoC 4
CE95380	TCPWM configuration with PSoC 4

5.7.9 SCB – I2C, SPI, UART

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

Table 18. 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

5.7.10 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 upto four 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 19. 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 20. 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

5.7.11 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, which are certified CAN conformant.

Table 21. CAN Code Examples

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

5.7.12 Segment LCD Drive

The PSoC 4200M has an LCD controller, which can drive up to four commons and up to 51 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 22. Segment LCD Application Note

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

Table 23. Segment LCD Code Examples

Document Number	Document Name
CE95369	Segment LCD with PSoC 4

5.7.13 GPIOs

The PSoC 4200M has 55 GPIOs in the 68-pin QFN package. The pins are organized in logical entities called ports, which are 8-bit in width.

Table 24. GPIO Application Notes

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

5.7.14 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 25. 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

5.7.15 System Level Application Notes and Code Examples

Table 26. 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 27. Fan Controller Application Note

Document Number	Document Name
AN89346	PSoC® 4 Intelligent Fan Controller

Table 28. 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 29. 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 30. 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 31. Bootloader Code Examples

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

Table 32. 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

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 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	Orig. of Change	Submission Date	Description of Change
**	3881879	RLIU	01/24/2013	New Application Note
*A	3968932	RLIU	04/11/2013	Demo project changed to leverage Pioneer kit. Added architecture introduction.
*B	3996226	MKEA	05/09/2013	Reformatted graphics. Updated links.
*C	4219723	NIDH	12/19/2013	Updated attached Associated Project files. Updated content across the entire document. Updated in new template.
*G	4339565	NIDH	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	MKEA	09/25/2014	Added Code Examples section. Minor edits and format changes throughout the document.
*I	4679544	NIDH	03/17/2015	Added More Information section. Removed detailed feature descriptions. Updated for PSoC 4200M family of devices.
*J	4826009	RNJT	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.

Worldwide Sales and Design Support

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

Products

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

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

Cypress Semiconductor	Phone	: 408-943-2600
198 Champion Court	Fax	: 408-943-4730
San Jose, CA 95134-1709	Website	: www.cypress.com