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.

CY8CKIT-049-4xxx

PSoC® 4 Prototyping Kit Guide

Doc. #: 001-90711 Rev. *J

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

Contents

Safety Information

Regulatory Compliance

The CY8CKIT-049-4xxx Prototyping Kit is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. This may cause interference to other electrical or electronic devices in close proximity. In a domestic environment, this product may cause radio interference. In such cases, you may be required to take adequate preventive measures. In addition, this board should not be used near any medical equipment or RF devices.

Attaching additional wiring to this product or modifying the product operation from the factory default may affect its performance and cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

The CY8CKIT-049-4xxx Prototyping Kit, as shipped from the factory, has been verified to meet with requirements of CE as a Class A product.

The CY8CKIT-049-4xxx contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-049-4xxx boards in the protective shipping package.

End-of-Life/Product Recycling

This kit has an end-of life five years from the date of manufacture mentioned on the back of the box. Contact your nearest recycler for discarding the kit.

General Safety Instructions

ESD Protection

ESD can damage boards and associated components. Cypress recommends that you perform procedures only at an ESD workstation. If such a workstation is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to the chassis ground (any unpainted metal surface) on your board when handling parts.

Handling Boards

CY8CKIT-049-4xxx boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad if available. Do not slide board over any surface.

1. Introduction

Thank you for your interest in the PSoC® 4 CY8CKIT-049-4xxx family of prototyping kits. The prototyping kit is designed as an easy-to-use and inexpensive prototyping platform for users wishing to rapidly develop products using the PSoC 4 families and use the unique flexibility of the PSoC 4 architecture. Designed for flexibility, these kits offer an open footprint breakout board to maximize the end utility of the PSoC 4 device. These kits provide a low-cost alternative to device samples while providing a platform to easily develop and integrate the PSoC 4 device into your end system. In addition, the board includes the following features:

- Onboard CMOD capacitors to enable CapSense® development
- A bypass capacitor to ensure the high quality ADC conversions
- An LED to provide feedback
- A push button to provide a simple user input and trigger the bootloader programming mode
- The CY8CKIT-049-41xx PSoC 4 Prototyping kit has CY8C4125AXI-483 device on-board and CY8CKIT-049-42xx PSoC 4 Prototyping kit has CY8C4245AXI-483 device on-board

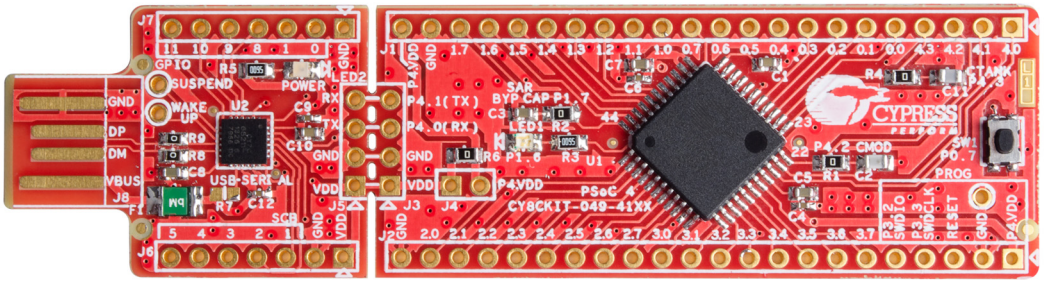
The CY8CKIT-049-4xxx development kit also supports the Cypress USB-Serial CY7C65211 Full-Speed USB controller that enables PC connectivity and serial interfaces, such as USB-UART, USB-I2C, USB-SPI, and USB-GPIO. The development kit includes a Cypress USB-Serial controller used to bootload the target PSoC 4 device. The PSoC 4 prototyping board is breakable, allowing you to separate the USB-Serial board from the PSoC 4 board.

This kit supports either 5 V or 3.3 V power supply voltages. The device can be programmed using the bootloader or the Cypress MiniProg3 programmer. The PSoC 4 Prototyping Kit supports boards based on the 4100 and 4200 device families, delivering a programmable platform for a wide range of embedded applications at a very low cost. The PSoC 4 is a scalable and reconfigurable platform architecture for a family of mixed-signal programmable embedded system controllers with an ARM® Cortex™-M0 CPU. It combines programmable and reconfigurable analog and digital blocks with flexible automatic routing.

1.1 Kit Contents

This kit contains only the PSoC 4 Prototyping Kit, either the 4100 or 4200 series device.

Figure 1-1. PSoC 4 CY8CKIT-049-4xxx Prototyping Kit



1.2 Getting Started

This user guide helps you to get acquainted with the PSoC 4 Prototyping Kit. The Software Installation chapter on page 13 describes the installation of the PSoC Creator software. The Kit Operation chapter on page 16 explains how to program the kit using a bootloader or a MiniProg3. The Hardware chapter on page 26 details the hardware operation of the kit. The Code Examples chapter on page 35 and USB-Serial Configuration chapter on page 53 walk you through making projects and configuring the USB-Serial device on the kit. The Appendix on page 62 provides the schematics, pin assignment, and bill of materials (BOM).

1.3 Additional Learning 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:

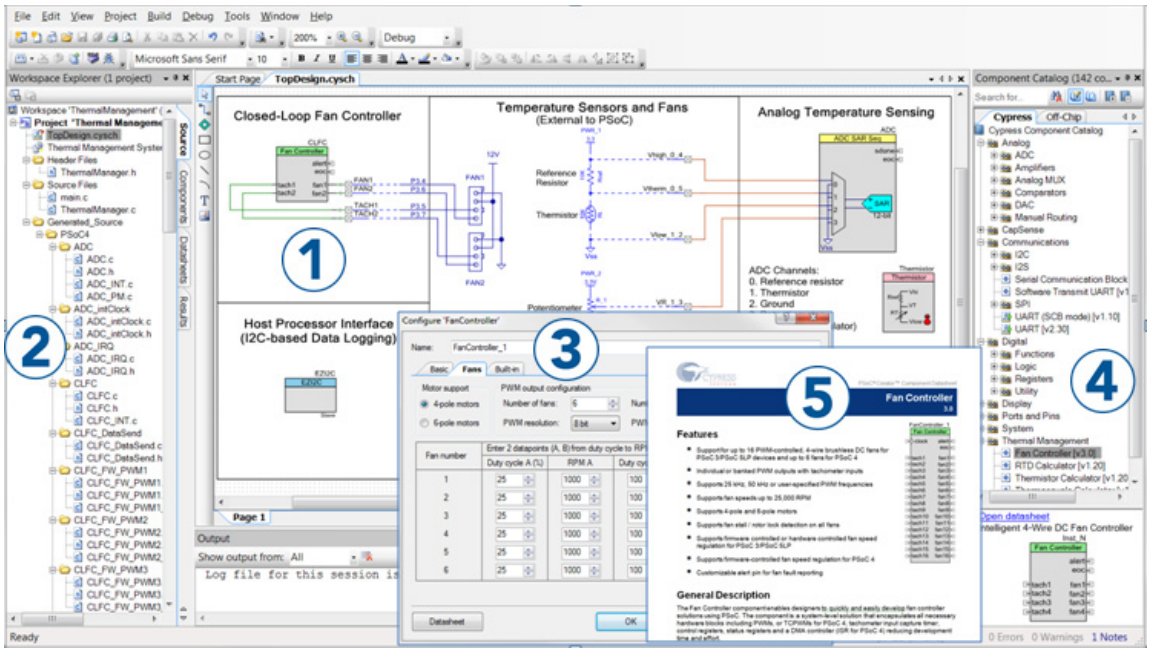
- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 4000, PSoC 4100, and PSoC 4200 device families.
- **CapSense 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. Visit the PSoC 3/4/5 Code Examples webpage for a list of all available PSoC Creator code examples. For accessing code examples from within PSoC Creator – see PSoC Creator Code Examples on page 10. These code examples available across PSoC Creator, Application notes and kits for the most part will NOT be bootloadable. However, with a very small modification you can port them to a bootloadable project and use them with the CY8CKIT-049. Refer to Converting a Non-bootloadable Project to a Bootloadable Project on page 40 for details.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- **Development Kits:**
  - CY8CKIT-042 and CY8CKIT-040, 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 4 devices.
  - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg3 device provides an interface for flash programming and debug.
- **Knowledge Base Articles (KBA):** Provide design and application tips from experts on the devices/kits. For instance, KBA93541, explains how to use CY8CKIT-049 to program another PSoC 4.

1.3.1 PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, and PSoC 5LP. See Figure 1-2 – with PSoC Creator, you can:

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

Figure 1-2. PSoC Creator Features



Visit PSoC Creator training page for video tutorials on learning and using PSoC Creator.

1.3.2 PSoC Creator Code Examples

PSoC Creator includes a large number of code example projects. These projects are available from the PSoC Creator Start Page, as Figure 1-3 shows.

Example projects can speed up your design process by starting you off with a complete design, instead of a blank page. The example projects also show how PSoC Creator Components can be used for various applications. Code examples and datasheets are included, as Figure 1-4 on page 11 shows.

In the Find Example Project dialog shown in Figure 1-4 on page 11, 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 1-3. Code Examples in PSoC Creator

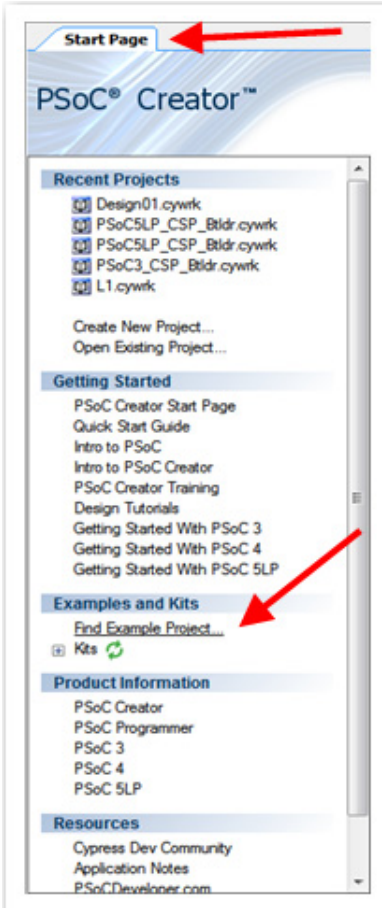
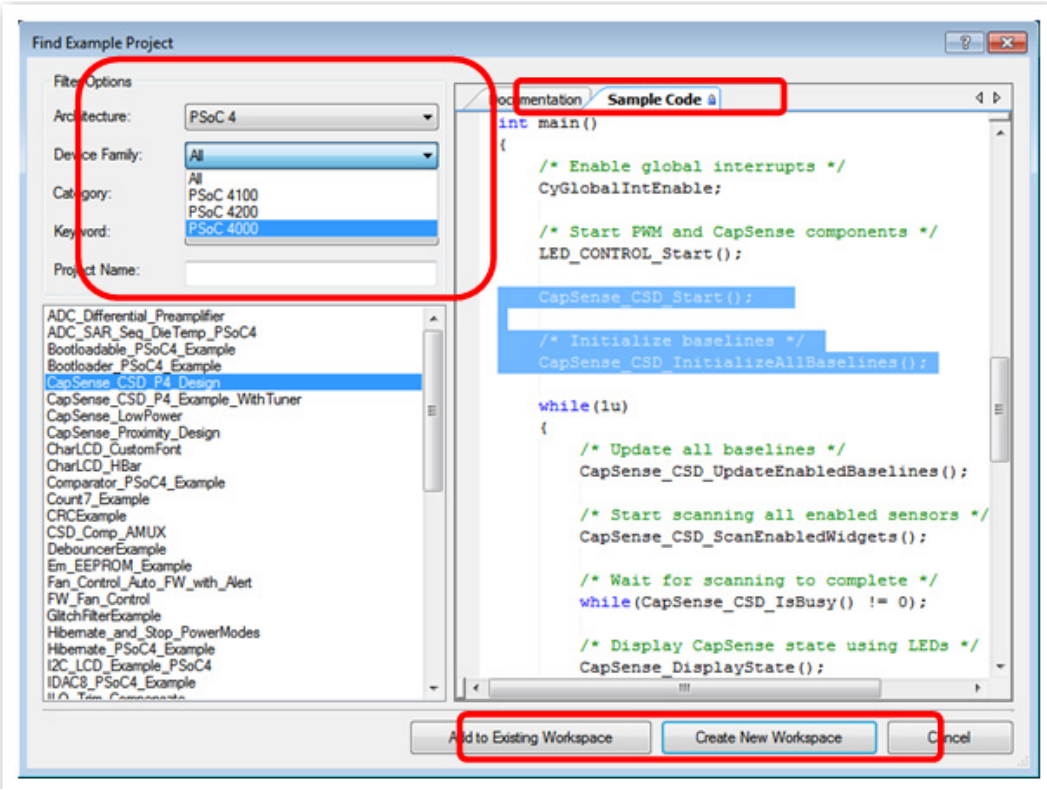


Figure 1-4. Code Example Projects, with Sample Code



1.3.3 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.
- **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**.

1.3.4 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 2 at the prompt.

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

- Self-help
- Local Sales Office Locations

1.4 Document Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: `C:\...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Creator User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes Cautions or unique functionality of the product.

2. Software Installation

2.1 Before You Begin

All Cypress software installations require administrator privileges, but these are not required to run the software after it is installed. Close any other Cypress software that is currently running before installing the kit software.

Note: The kit contents are installed in the C:\Program Files\Cypress folder by default. If the

Code examples are being run from the default install location, administrator privileges are required. If you do not have administrator privileges, copy the Firmware folder from the default install location to any other location on your PC and access the files

2.2 CY8CKIT-049-41xx/CY8CKIT-049-42xx Software

The kit requires Cypress' proprietary software, such as PSoC Creator, PSoC Programmer and the USB-Serial Configuration Utility, and generic software such as .NET Framework, Windows Installer, and Internet Explorer. The kit software is available on the kit web page in three formats:

Table 2-1. Kit Software Formats

Install Package	File Format	Usage
CY8CKIT-049-41xx_Kit ISO/ CY8CKIT-049-42xx_Kit ISO	ISO	This package can be used if the PC does not have any Cypress or non-Cypress prerequisite software installed. It first installs the prerequisites and then the kit content (firmware, hardware, and documentation files) in the specified location.
CY8CKIT-049-41xx_Kit Setup/ CY8CKIT-049-42xx_Kit Setup	EXE	This package can be used if the PC does not have any Cypress prerequisite software installed. If any non-Cypress prerequisites are found to be missing during installation, the installer provides links to download and install them and then installs the kit content (firmware, hardware, and documentation files) in the specified location.
CY8CKIT-049-41xx_Kit Only/ CY8CKIT-049-42xx_Kit Only	EXE	This package can be used if the PC has all the Cypress and non-Cypress prerequisites installed. It installs only the kit content (firmware, hardware, and documentation files) in the specified location. If any of the prerequisites are found missing during the installation process, the installer prompts you to install all the required software before attempting to install the kit. The installer redirects to the kit web page to download and install any missing Cypress software. Similarly, it provides links to download and install the missing non-Cypress prerequisites.

Note: The USB-Serial Configuration Utility requires Visual C++. If Visual C++ is not installed in the PC, the installer provides link to install the Visual C++ Redistributable package in the PC.

Note: Adobe Reader is required to view kit documents. If Adobe Reader is not installed on your PC, the installer provides the link to download and install it.

Note: PSoC Creator is provided with a free Keil C licence that has to be registered within 30 days of installing PSoC Creator. To register your Keil license, you will require an internet connection. Please read http://www.cypress.com/?id=4&rID=38519 for more details. Please read http://www.cypress.com/?id=4&rID=44355 if PSoC Creator needs to be used on a PC that does not have internet connection. The product ID for the Keil compiler is IKA1P-M6Q0E-8W7ST.

2.3 Install Software

1. Run *cyautorun.exe* in the kit ISO to start the installation process.
2. Click **Install CY8CKIT-049-41xx** to start CY8CKIT-049-41xx PSoC 4 Prototyping kit installation. Click **Install CY8CKIT-049-42xx** to start CY8CKIT-049-42xx PSoC 4 Prototyping kit installation.
3. Select the folder to install the kit files. Choose the directory and click **Next**. The installation directory is referred to as <*Install_Directory*> in this document.
4. When you click **Next**, the kit installer automatically installs the required software, if it is not present on your computer.
5. Select the installation type in the **Product Installation Overview** window. The drop-down menu contains three options: **Typical** (installs all the required features), **Custom** (lets you choose the features to be installed), and **Complete** (installs all the contents). Click **Next** after you select the installation type.

   Note: It is recommended that you choose the **Complete** installation type.
6. Read and Accept the End-User License Agreement and click **Next** to proceed with the installation.
7. When the installation begins, a list of packages appears on the installation page. A green check mark appears adjacent to every package after successful installation.
8. After installing all the packages, the CyInstaller finish page opens up. Please provide the requested contact information or uncheck the “Continue Without Contact Information” checkbox and click on **Finish** button to complete the kit installation.

After the installation is complete, the kit contents are available at the following location:

1. For CY8CKIT-049-41xx: <Install_Directory>\ CY8CKIT-049-41xx \<version>
2. For CY8CKIT-049-42xx: <Install_Directory>\ CY8CKIT-049-42xx \<version>

2.4 Install Hardware

There is no additional hardware installation required for this kit.

2.5 Uninstall Software

To uninstall the software, do one of the following:

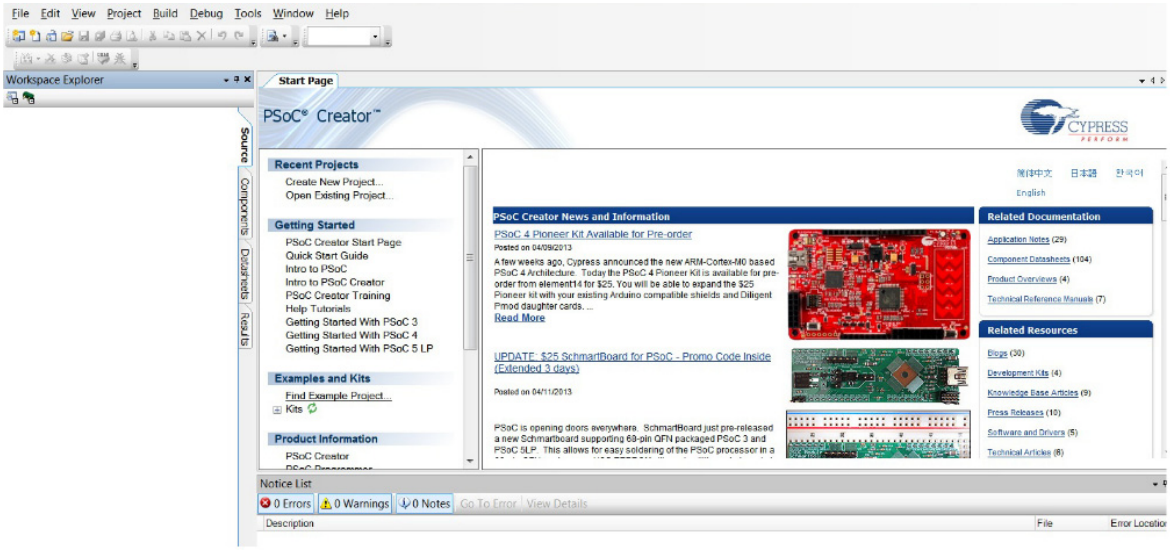
- Go to **Start** > **All Programs** > **Cypress** > **Cypress Update Manager** > **Cypress Update Manager**, and then select the **Uninstall** button corresponding to the kit software.
- Go to **Start** > **Control Panel** > **Programs and Features**, and then select the **Uninstall/Change** button corresponding to the kit software.

2.6 Open the “PSoC 4 Code” Code Example in PSoC Creator

Note: The code examples require administrator privileges if they are run directly from the default install location (C:\Program Files\Cypress). If you do not have administrator privileges, copy the *Firmware* folder from the default install location to any other location on your PC and use the files.

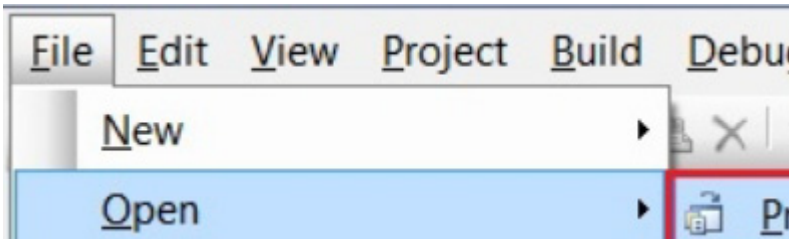
1. Launch the PSoC Creator software from the Start menu.

Figure 2-1. PSoC Creator Start Page



2. Open the *SCB_Bootloader.cywrk* workspace by choosing **File** > **Open** > **Project/Workspace** and navigating to the directory in which your project is present.

Figure 2-2. Open Project/Workspace

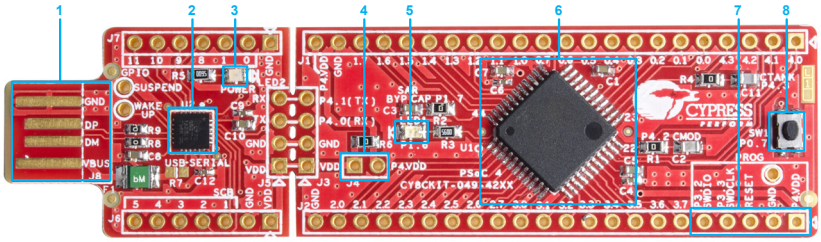


The workspace includes two sample projects linked in the **Workspace Explorer**. Subsequent chapters of this user guide show how to build, program, and understand the code examples supplied with this kit.

3. Kit Operation

The PSoC 4 Prototyping Kit is simplistic in design and focuses on providing you with complete access to develop applications using the PSoC 4 device family. The development kit supports a number of onboard functions such as an LED, push button, through-hole connections, USB-Serial connectivity to the PC, and a breakable board design to separate the two target boards.

Figure 3-1. PSoC 4 Prototyping Kit



1. PCB USB Connector (J8)
2. USB-Serial Bridge Controller (U2)
3. Power LED (LED2)
4. Current Measurement Jumper (J4)
5. User LED (LED1)
6. PSoC 4 (U1)
7. Programming Header
8. Push button (SW1)

3.1 Connecting the PSoC 4 Prototyping Kit to a Computer

To use the PSoC 4 Prototyping Kit, you need to connect the kit to a target PC. The kit is designed to be connected to the computer through USB. The USB connector will provide power to the target boards and enable serial communication. CY8CKIT-049-4xxx implements a PCB-based USB connector that makes connections to the USB port when plugged in. The amber LED turns on when the board is plugged into the port to indicate power.

Figure 3-2. Connecting the PSoC 4 Prototyping Kit to a Computer

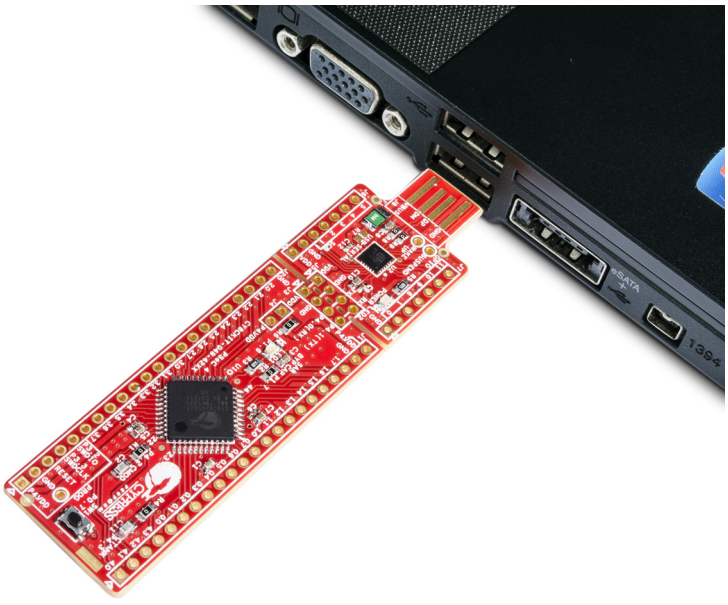
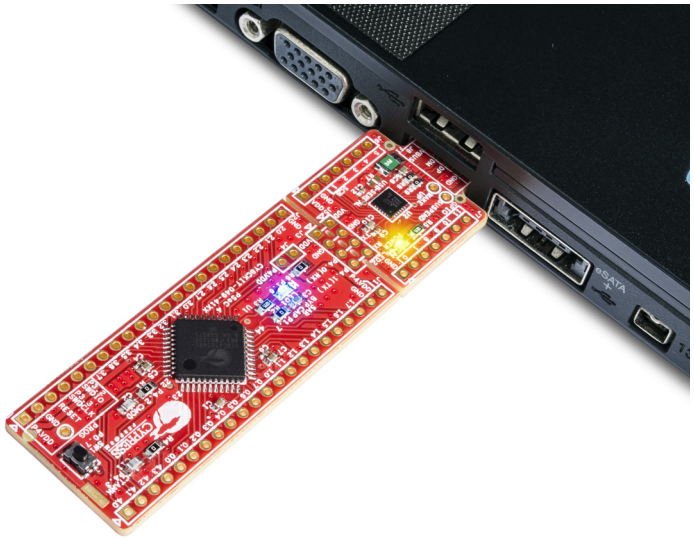


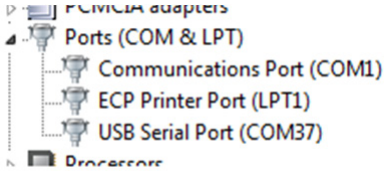
Figure 3-3. PSoC 4 Prototyping Kit Connected to the Computer



3.2 CY8CKIT-049-4xxx USB COM Port

When you connect the CY8CKIT-049-4xxx to the PC over a USB interface, it enumerates as a COM port device under the Device Manager window on the Windows OS. Often, the COM port number will be higher than any existing COM port value. For example, in the following image the CY8CKIT-049-4xxx enumerates as **COM37**.

Figure 3-4. CY8CKIT-049-4xxx USB COM Port in Device Manager



When connecting your CY8CKIT-049-4xxx to a computer for the first time or to any new USB port, it may take a moment to enumerate because the computer will complete an online check for the latest drivers.

Figure 3-5. Automatic Driver Software Installation for CY8CKIT-049-4xxx

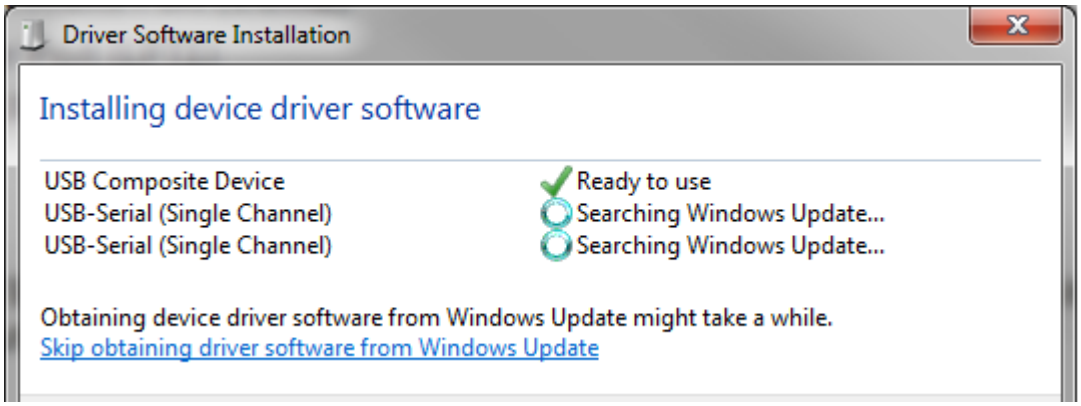
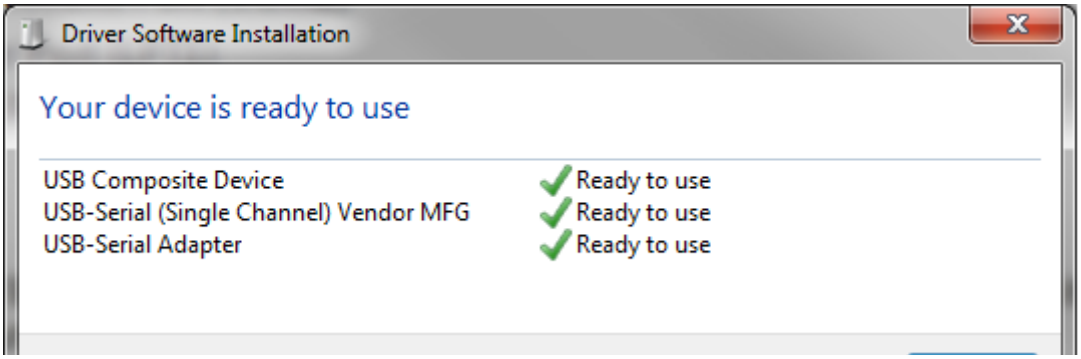


Figure 3-6. Driver Software Installation Complete



Note: The baud rate settings do not apply to the virtual COM port as the data transmission is taking place using the Full-Speed USB bus at 12 Mbps. However, because of the virtual COM port driver that sits between the PC and the USB-Serial device, the operating system will see the device as a normal serial port and you will be able to set the baud rate. However, these settings may be ignored.

3.3 Programming a CY8CKIT-049-4xxx Project Using the Bootloader

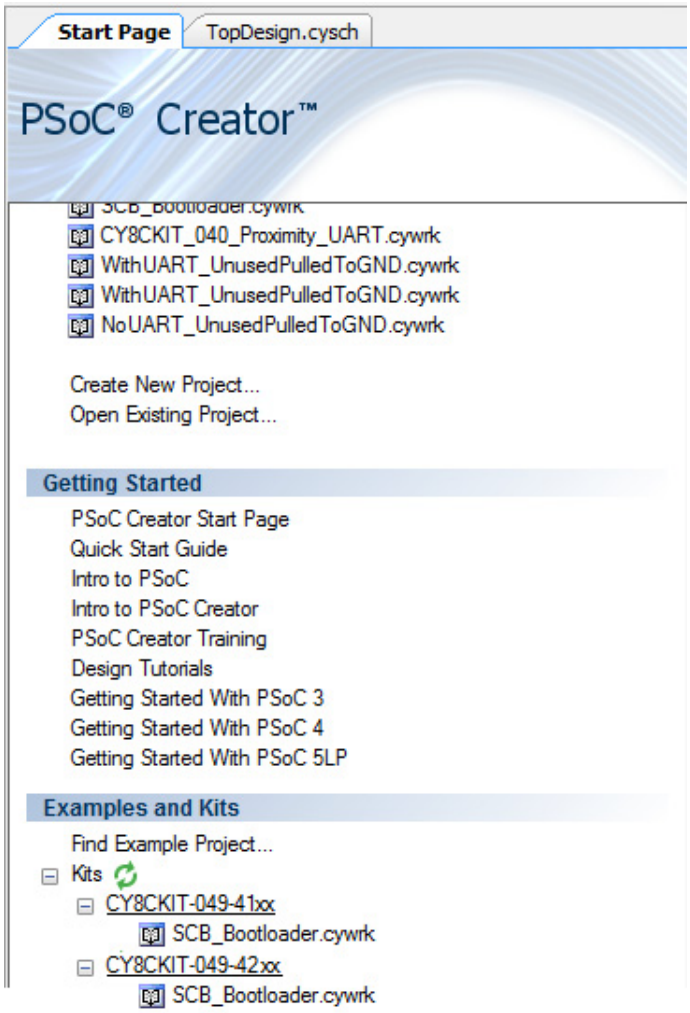
The following example shows how to bootload (program) a project into the PSoC 4 with the USB-Serial device, using the Bootloader Host. To use this method, the PSoC 4 device must contain the bootloader and the project must be configured as bootloadable. This is the default programming method for new users.

The following steps use the code example included in the kit installer.

1. Launch PSoC Creator from the Start menu.

2. Open the *SCB_Bootloader.cywrk* workspace from **Examples and Kits** > **Kits**. Select CY8CKIT-049-41xx folder for CY8CKIT-049-41xx PSoC 4 Prototyping Kit and CY8CKIT-049-42xx folder for CY8CKIT-049-42xx PSoC 4 Prototyping Kit.
3. Select the folder where you want to save the project and click **OK**.

Figure 3-7. Open the Project

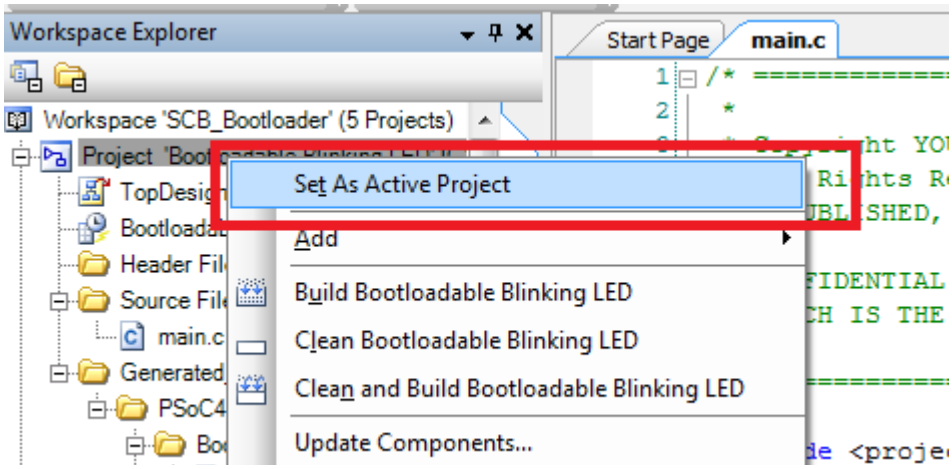


4. Click on **Build** > **Build All Projects**.

   Note: The UART_Bootloader project is a dependency for the Bootloadable Blinking LED project. Hence, both example projects must be built by selecting **Build** > **Build All Projects**.

5. In the Workspace Explorer, right-click the *Bootloadable Blinking LED* project and select **Set As Active Project**.

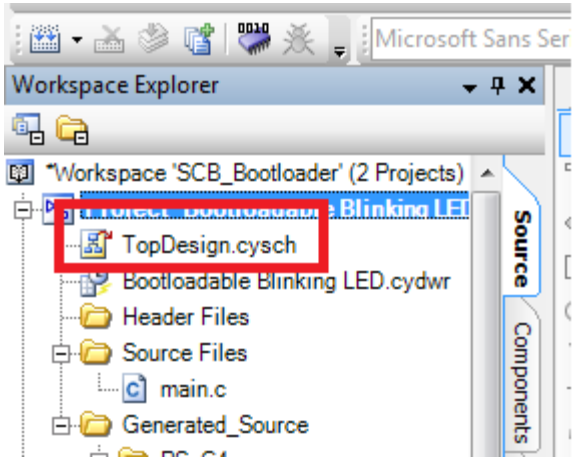
Figure 3-8. Set the Code Example as Active Project



   The bootloadable project must be associated with the bootloader project's HEX and ELF files. This will ensure that the firmware code mapping aligns with the code on the target device.

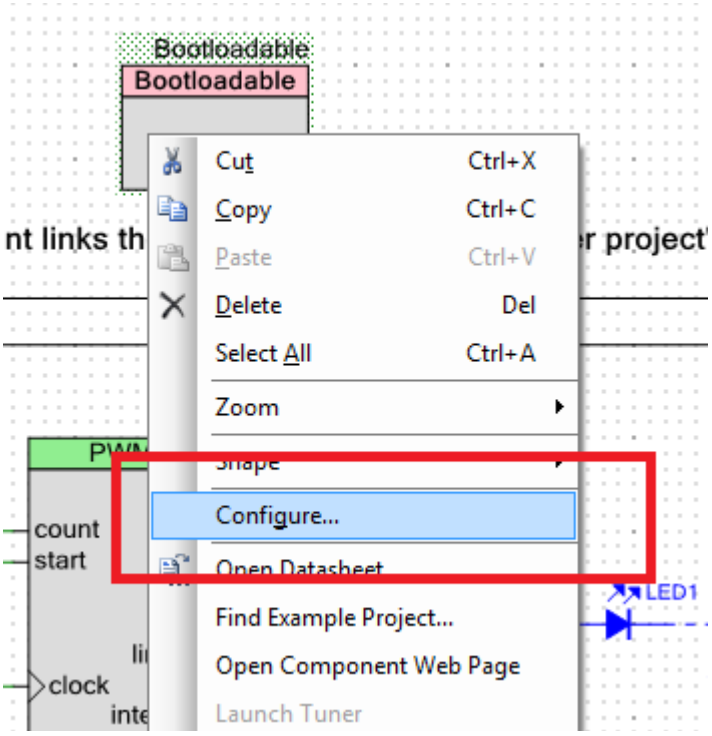
6. Under the 'Bootloadable Blinking LED' code example, double-click the *'TopDesign.cysch'* file to open the schematic view. Select the tab for the Bootloadable Project schematic page if it is not already selected.

Figure 3-9. Open Schematic View



7. In the schematic view, right-click the **Bootloadable** component and select **Configure**.

Figure 3-10. Configure the Bootloader Component

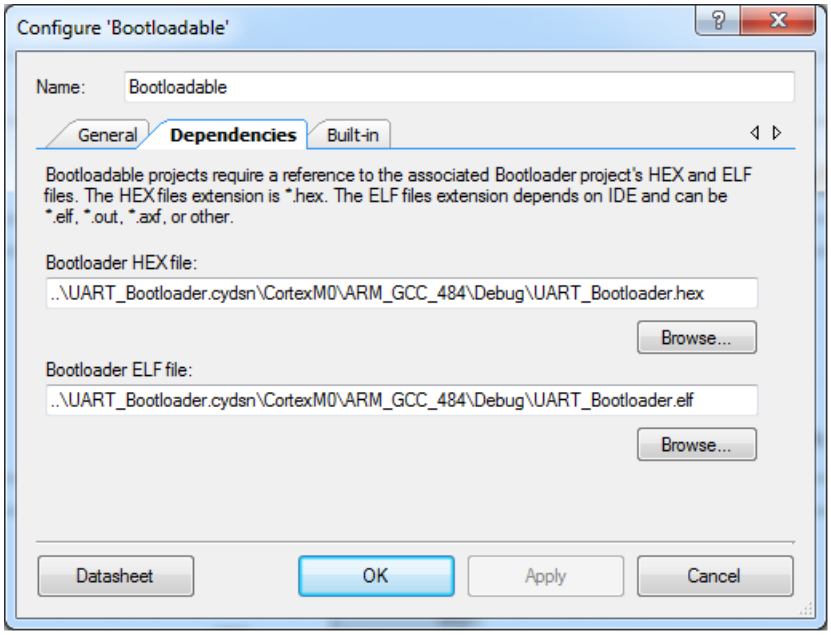


8. In the configuration window, select the **Dependencies** tab and click the **Browse** button to point to the HEX and ELF files present in the 'Dependencies' folder under project directory; click **OK**. The file paths (assuming the CY8CKIT-040-42xx) will appear as follows:

   `<Project_Directory>\SCB_Bootloader\UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug\UART_Bootloader.hex`

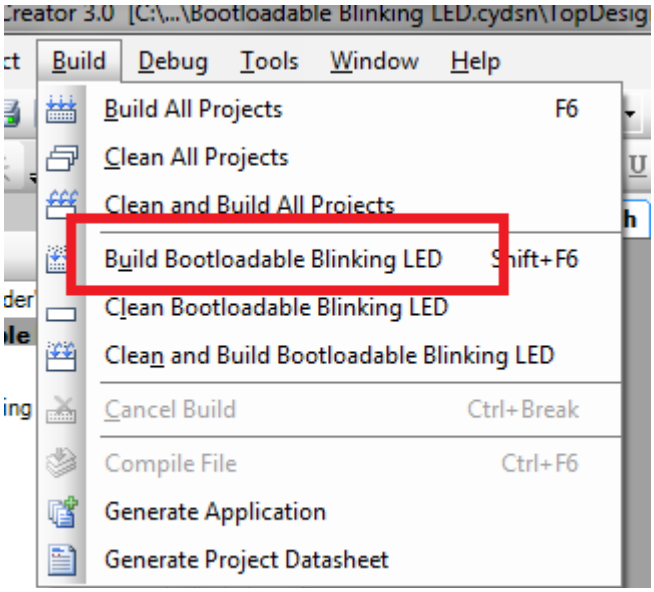
   `<Project_Directory>\SCB_Bootloader\UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug\UART_Bootloader.elf`

Figure 3-11. Configure Dependencies for Bootloader Component



9. Select **Build > Build Bootloadable Blinking LED**.

Figure 3-12. Build the Project

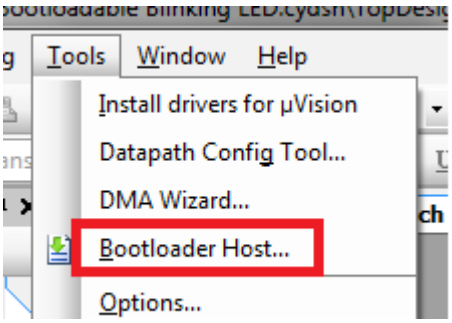


10. Connect the CY8CKIT-049-4xxx prototyping board to the PC. When connecting the kit to the port, depress the **SW1** button as it is plugged in.

    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.

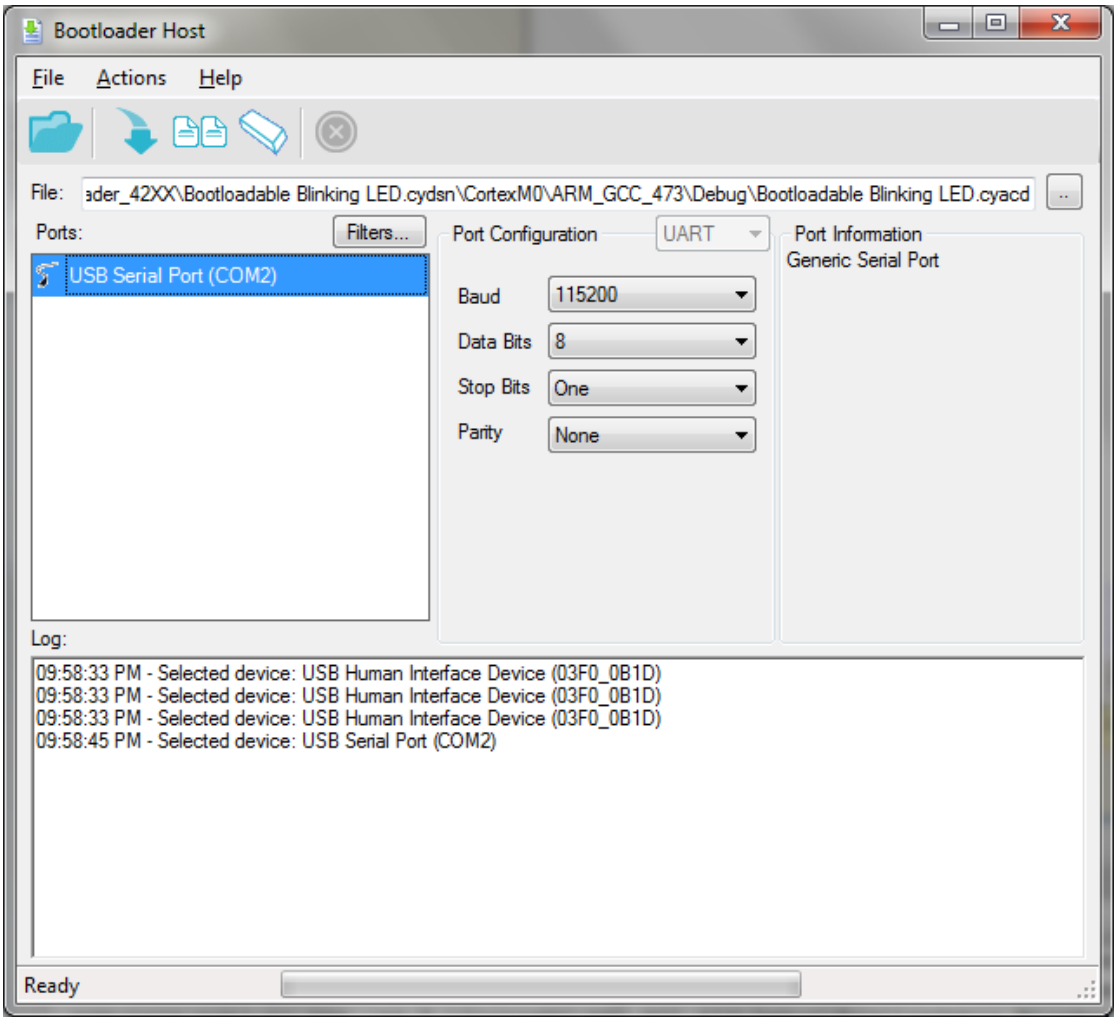
11. Select **Tools > Bootloader Host** to open the Bootloader Host tool.

Figure 3-13. Launch Bootloader Host Tool



The Bootloader Host tool opens.

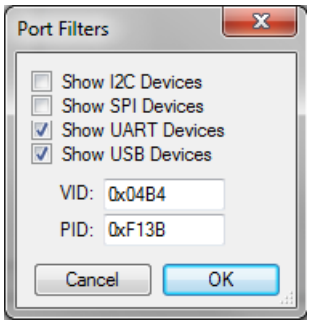
Figure 3-14. Bootloader Host Tool



12. Click **Filters** and select the **Show UART Devices** option from the Port Filters window and 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 3-15. Port Filters



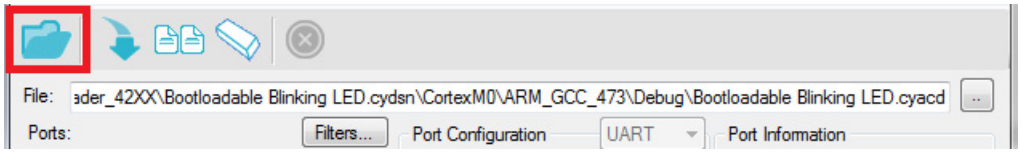
The Bootloader Host tool will now display all of the available UART based COM ports.

13. 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.

14. Click **File > Open** and navigate to the *Bootloadable_Blinking_LED.cyacd* file generated in the CortexM0 folder in your project directory, and click **Open**.

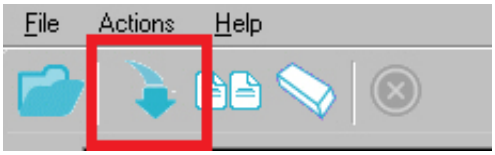
Figure 3-16. Opening the Generated File



15. 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 3-17. Program the Device With Application Code



See application note AN73854 for additional details on bootloading.

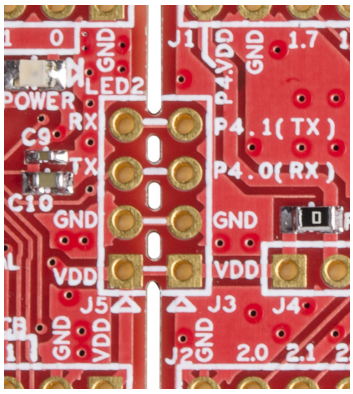
3.4 USB-UART Default Settings

The default configuration of the USB-Serial device on the CY8CKIT-049-4xxx prototyping kit is the USB-UART mode. The CY8CKIT-049-4xxx also enables a default UART connection between the USB-Serial device and the PSoC 4. This connection is indicated by two parallel 4-pin headers in the middle of the board. The default pin connections are shown in Table 3-2.

Table 3-1. Pin Mapping for USB-Serial Port and PSoC 4 UART

USB-Serial UART	PSoC 4 UART
TX	P4.0 (RX)
RX	P4.1 (TX)
GND	GND
VDD	VDD

Figure 3-18. UART Pin Connections on CY8CKIT-049-4xxx



The USB-Serial device provides the PSoC 4 device with an interface to a PC. The USB-Serial device enumerates as a COM port to allow any terminal software to be used to communicate with the PSoC 4. To use the USB-UART functionality in the COM terminal software, select the corresponding COM port. Note that if the USB-Serial device board is separated from the PSoC 4 board, you will still be able to use the USB-Serial device to communicate with any UART device using the 4-pin header.

The USB-Serial device is by default configured as a USB-UART device with the following specifications. The USB-UART settings can also be configured from the Cypress USB-Serial Configuration Utility. The default setting are as shown in bold in the table below.

Table 3-2. USB-UART settings as seen in Cypress USB-Serial Configuration Utility

Parameter	Supported Values
Baud Rate	100, 200, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, **115200**, 230400, 460800, 921600, 1000000, 3000000
Type	**2 pin**, 4 pin, 6 pin
Data Bits	7 bits, **8 bits**
Stop Bits	**1 bit**, 2 bits
Parity	**None**, Odd, Even, Mark, Space
Drop Packets on RX error	**Disabled,** Enabled
Disable CTS and DSR pull-up during suspend	Disabled, **Enabled**

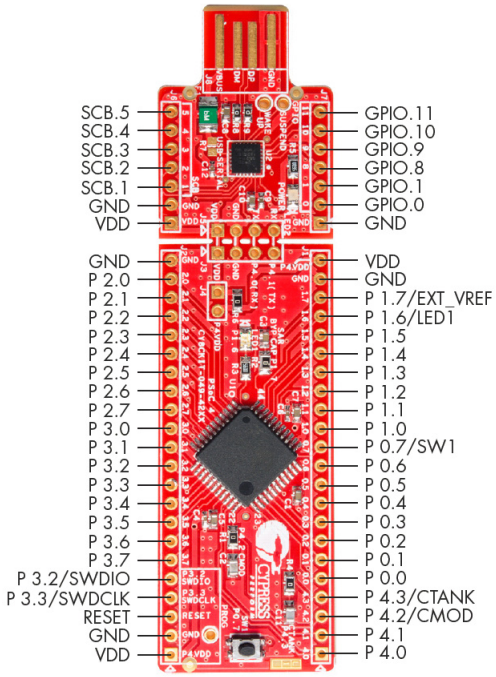
4. Hardware

4.1 Board Details

The PSoC 4 Prototyping Kit consists of the following blocks:

- PSoC 4 device
- PSoC 4 header ports J1 and J2
- USB-Serial device
- USB-Serial header ports J5 and J6 (GPIO)
- UART connection J3 and J4 (SCB and GPIO)
- PCB USB connector
- One amber LED (Power)
- One blue LED (User)
- Push button
- External reference capacitor (ADC Bypass)
- CapSense and shield capacitors (CMOD and CTANK)
- Programming connector
- Perforated 'snappable' board design

Figure 4-1. CY8CKIT-049-4xxx Pin Details



4.2 Theory of Operation

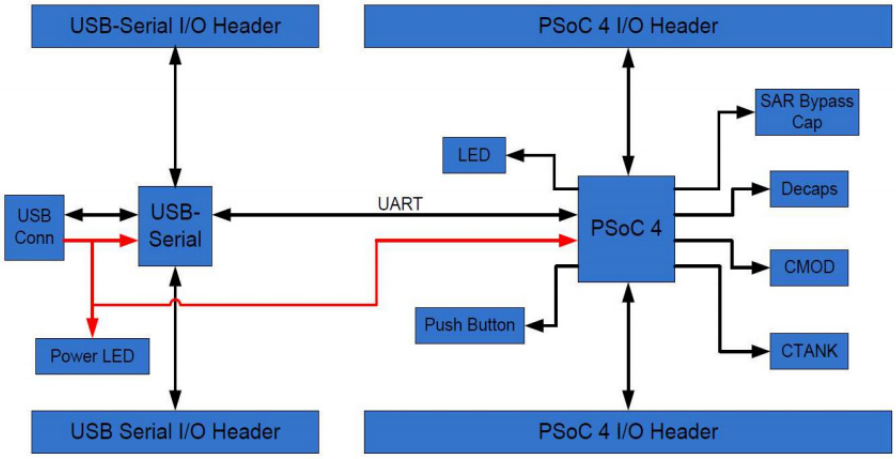
PSoC 4 is a new generation of programmable system-on-chip device from Cypress for embedded applications. It combines programmable analog, programmable digital logic, programmable I/O, and a high-performance ARM Cortex-M0 core. With PSoC 4, you can create the combination of peripherals required to meet your application's specifications.

The PSoC 4 Prototyping Kit features an onboard USB-Serial device, which communicates to a PC through USB to provide serial communication support and serial port debugging.

The PSoC 4 Prototyping Kit has a user LED and a power status LED. This kit includes a button that connects to the PSoC 4 device, which can be used to develop applications. This button is also used to enable the onboard bootloader. The PSoC 4 pins are brought out onto headers J1 to J2 on the kit and support 100-mil breadboard spacing.

The PSoC 4 Prototyping Kit can be powered from USB or an external power supply. The input voltage is either 5 V from USB or a variable supply from an external source.

Figure 4-2. CY8CKIT-049-4xxx Functional Block Diagram



4.3 Functional Description

4.3.1 Power Supply System

The power supply system on this board is dependent on the source of the power. For most applications, you can use the 5 V supply from the USB connection to power the system. You can also connect an external power supply to the board for low-voltage applications. The kit supports the following connections:

- 5 V from the PCB USB
- 1.8-5 V from a regulated supply connected to VDD

It is important to understand that this prototyping kit does not have any onboard ESD protection circuitry. Therefore, the power source for the CY8CKIT-049-4xxx must be of a high quality to ensure that the board is protected from any over-current conditions and swapped-power connections.

4.3.1.1 *Measure PSoC 4 Current Consumption*

You can measure the current consumption of the PSoC 4 device by using one of these methods:

Method 1:

1. Separate the USB-Serial board by 'snapping' the perforated edge between the two boards.
2. Power the remaining prototyping board via any of the VDD terminals.
3. Place an ammeter in series with the VDD connection to measure the current consumption.

Method 2:

1. Remove the resistor R6 and install a 2-pin jumper in the supplied holes of J4.
2. Connect an ammeter across the 2-pin jumper to measure the current to the PSoC 4 device.

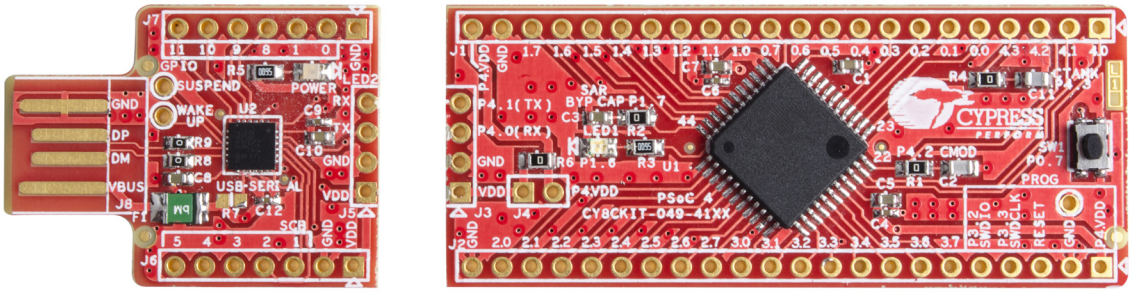
This method can be used either with USB power or with power supplied to one of the VDD pins.

4.3.2 Board Separation (Snapping)

CY8CKIT-049-4xxx supports both the PSoC 4 and USB-Serial boards. To separate the two boards for testing or development, break the two boards apart at the built-in perforated edge.

The easiest method of separating the two boards is to place the kit on the edge of a table, where the edge of the table is directly below the perforated edge and the smaller USB-Serial device is off the table edge. Press gently on the USB-Serial board and snap the two boards apart. If any material is removed from the edge of the boards, use sheers to clean up the edge of the kit.

Figure 4-3. CY8CKIT-049-4xxx Broken as Two Parts



4.3.3 Header Connections

The CY8CKIT-049-4xxx Prototyping Kit supports a number of unpopulated headers on both the USB-Serial and the PSoC 4 boards.

4.3.3.1 *Functionality of the J1 and J2 Headers (PSoC 4)*

The main PSoC 4 board contains two dual-inline headers (J1 and J2). These headers are both 1×22-pin headers and include all of the I/O available on the PSoC 4 devices. These headers support all of the available ports, GND, VDD, and connections to passive elements and user-input devices.

The J1 and J2 headers support 100-mil spacing, so you can solder the male connectors to connect the CY8CKIT-049-4xxx to any development breadboard.

Figure 4-4. J1 and J2 Headers

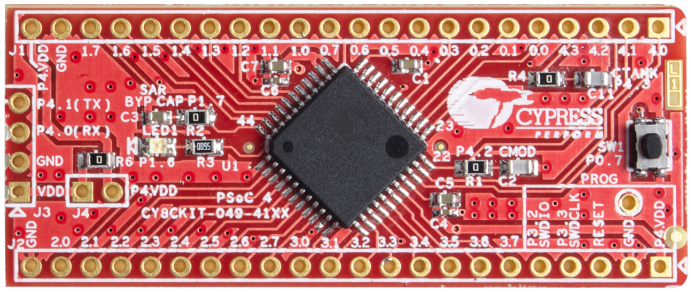


Table 4-1. J1 Header Pin Details

PSoC 4 GPIO Header (J1)		
Pin	**Signal**	**Description**
J1_01	P4.0	GPIO
J1_02	P4.1	GPIO
J1_03	P4.2	GPIO/CMOD
J1_04	P4.3	GPIO/CTANK
J1_05	P0.0	GPIO
J1_06	P0.1	GPIO
J1_07	P0.2	GPIO
J1_08	P0.3	GPIO
J1_09	P0.4	GPIO
J1_10	P0.5	GPIO
J1_11	P0.6	GPIO
J1_12	P0.7	GPIO/SW1
J1_13	P1.0	GPIO
J1_14	P1.1	GPIO
J1_15	P1.2	GPIO
J1_16	P1.3	GPIO
J1_17	P1.4	GPIO
J1_18	P1.5	GPIO
J1_19	P1.6	GPIO/LED1
J1_20	P1.7	GPIO/SAR Bypass (EXT_VREF)
J1_21	GND	Ground
J1_22	VDD	Power

Table 4-2. J2 Header Pin Details

PSoC 4 GPIO Header (J2)		
Pin	**Signal**	**Description**
J2_01	VDD	Power
J2_02	GND	Ground
J2_03	RESET	Reset
J2_04	P3.3	GPIO/SWDCLK
J2_05	P3.2	GPIO/SWDIO
J2_06	P3.7	GPIO
J2_07	P3.6	GPIO
J2_08	P3.5	GPIO
J2_09	P3.4	GPIO
J2_10	P3.3	GPIO/SWDCLK
J2_11	P3.2	GPIO/SWDIO
J2_12	P3.1	GPIO
J2_13	P3.0	GPIO
J2_14	P2.7	GPIO
J2_15	P2.6	GPIO
J2_16	P2.5	GPIO
J2_17	P2.4	GPIO
J2_18	P2.3	GPIO
J2_19	P2.2	GPIO
J2_20	P2.1	GPIO
J2_21	P2.0	GPIO
J2_22	GND	Ground

4.3.3.2 *Functionality of J3 and J5 Headers (PSoC 4 to USB-Serial)*

Both the USB-Serial and the PSoC 4 prototyping boards each contain a 1×4-pin header. This header provides a physical connection between the two devices. Specifically, the connection includes the UART (RX and TX), VDD, and GND connections between the two devices. When the boards are separated, this physical connection is broken.

Figure 4-5. J3 and J5 Headers

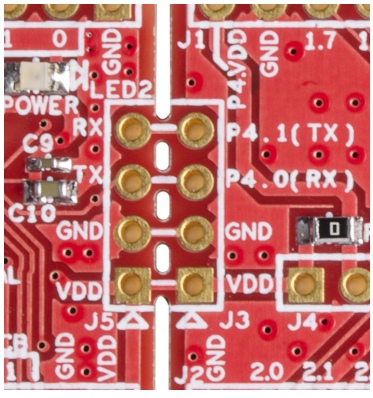


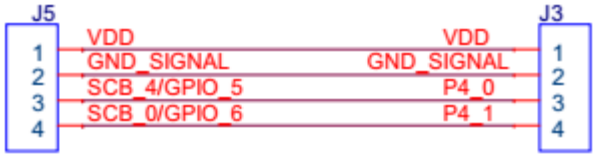
Table 4-3. Pin Details of J3 Header

PSoC 4 to USB-Serial Header (J3)		
Pin	**Signal**	**Description**
J3_01	VDD	Power
J3_02	GND	Ground
J3_03	P4.0	UART RX
J3_04	P4.1	UART TX

Table 4-4. Pin Details of J5 Header

USB-Serial to PSoC 4 Header (J5)		
Pin	**Signal**	**Description**
J5_01	VDD	Power
J5_02	GND	Ground
J5_03	SCB.0/GPIO_6	UART TX
J5_04	SCB.4/GPIO_5	UART RX

Figure 4-6. UART Connection to PSoC 4



4.3.3.3 *Functionality of J6 and J7 Headers (USB-Serial)*

The USB-Serial board contains two dual-inline headers (J6 and J7). These headers are both 1x7-pin headers and include all of the GPIO and SCB connections. These headers support all of the available ports, GND, VDD, and connections to passive elements and user-input devices.

The J6 and J7 headers support 100-mil spacing, so you can solder the male connectors to connect the USB-Serial board to any development breadboard.

Figure 4-7. J6 and J7 Connectors

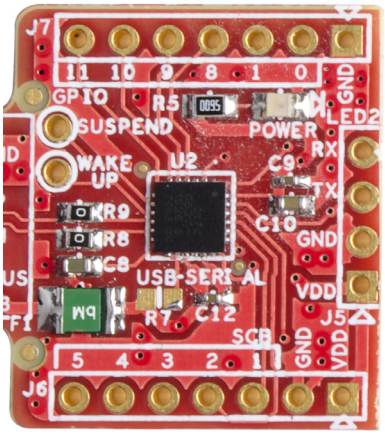


Table 4-5. Pin Details of J6

USB-Serial Comm/GPIO Header (J6)		
Pin	**Signal**	**Description**
J6_01	VDD	Power
J6_02	GND	Ground
J6_03	S SEL	Mode 0-6
J6_04	MISO/SCL	Mode 0-6
J6_05	MOSI/SDA	Mode 0-6
J6_06	SCLK	Mode 0-6
J6_07	SCB.5/GPIO_7	Mode 0-6

Table 4-6. Pin Details of J7

USB-Serial GPIO Header (J7)		
Pin	**Signal**	**Description**
J7_01	GND	Power
J7_02	GPIO.0	Ground
J7_03	GPIO.1	Reset
J7_04	GPIO.8	GPIO
J7_05	GPIO.9	GPIO
J7_06	GPIO.10	GPIO
J7_07	GPIO.11	GPIO

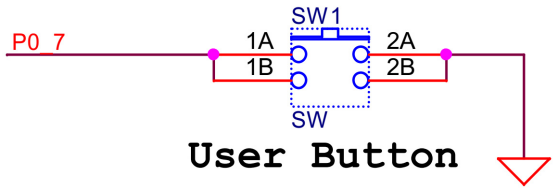
4.3.4 User and Passive Inputs

4.3.4.1 Push Button

The main PSoC 4 board contains a single push button connected to the P0.7 pin on the PSoC 4 device. This button can be used for general user inputs and for triggering the bootloader for programming.

Figure 4-8. Push Button on the Board

Figure 4-9. Push Button Schematic



4.3.4.2 *CY8CKIT-049-4xxx LEDs*

CY8CKIT-049-4xxx contains two LEDs: the amber LED, which indicates the board is power applied and the blue LED that is directly connected to the PSoC 4 device through the pin P1.6. The blue LED is also used to indicate the bootloader mode by rapidly blinking. The power LED is on the USB-Serial board; if the boards are separated, the PSoC 4 section does not consume current through the power LED.

Figure 4-10. Power LED

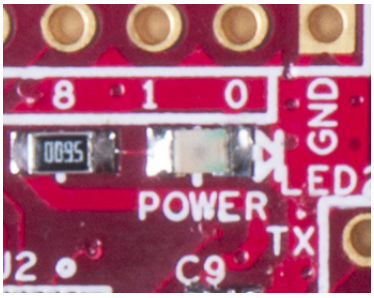


Figure 4-11. User LED

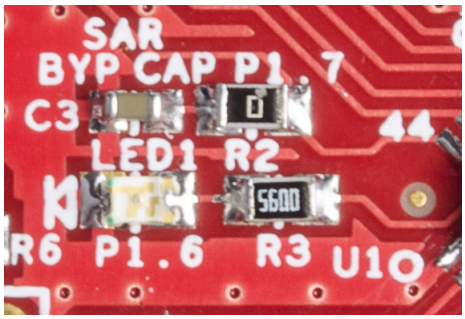


Figure 4-12. Power LED Connection

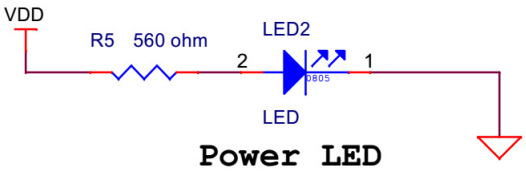
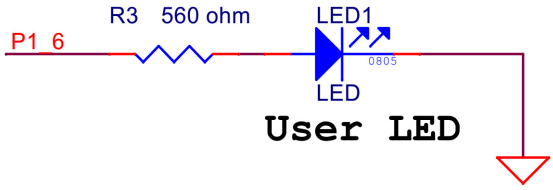


Figure 4-13. User LED Connection

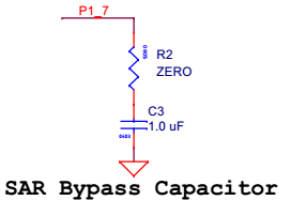


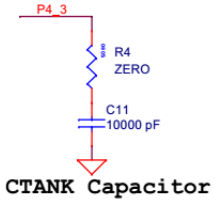
4.3.4.3 *System Capacitors*

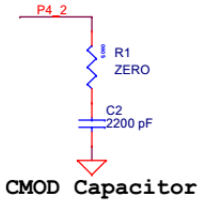
The three capacitors on the CY8CKIT-049-4xxx prototyping kit enable proper development of ADC and CapSense code examples. These capacitors are the following:

- A SAR ADC bypass capacitor: Required for proper sampling at high frequencies,
- Two CapSense capacitors (CMOD and CTANK): Required for proper CapSense functionality.

Figure 4-14. System Capacitors Circuit Diagram







5. Code Examples

This section describes how to use the code example included with the kit and how to develop custom bootloadable code examples for new applications.

For a list of all code examples available with PSoC Creator, visit the PSoC 3/4/5 Code Examples page. This page lists all the PSoC Creator based code examples available across PSoC Creator, Application notes and kits. Most of these examples CANNOT be directly used with the kit. Refer to Converting a Non-bootloadable Project to a Bootloadable Project on page 40 for details on how to port these projects to use with CY8CKIT-049.

For PSoC Creator related training and video tutorials, visit PSoC Creator training page.

5.1 Bootloader Base Code Example

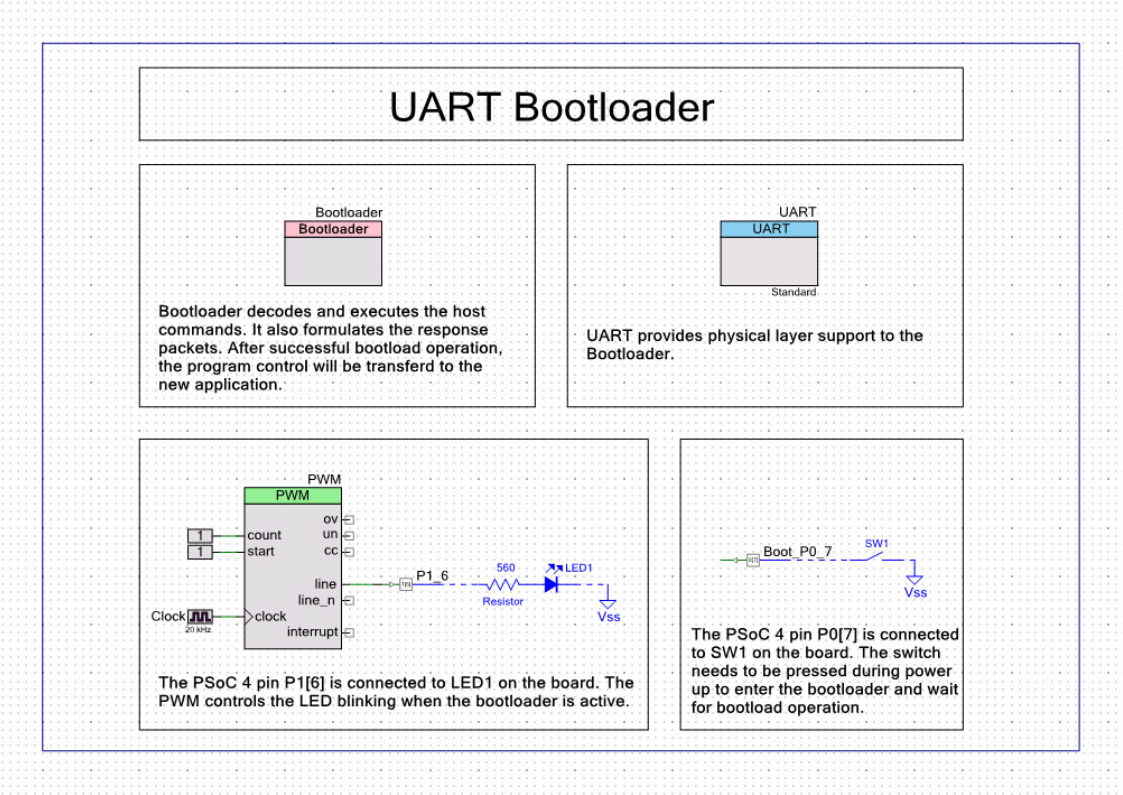
The CY8CKIT-049-4xxx prototyping board is pre-programmed with a simple blinking LED code example. This code example uses a PWM to slowly blink an LED. Included in the application project is the Bootloader Base project. The Bootloader code is detailed in the UART_Bootloader project available on the kit web page.

In the bootloader example, the device rapidly blinks an LED when the bootloader is active, provides UART communication support for bootloading, and reads the state of the switch (SW1). You can observe the state of the board by noticing the rate at which the LED blinks. The bootloader is activated when you plug in the CY8CKIT-049-4xxx while pressing the SW1 button. The bootloader reads the state of this button during power-up. If the button is not pressed, the bootloader jumps to the user's application code. If the button is pressed, then the bootloader waits for a new application to be transferred. While the bootloader waits for the new application, it rapidly blinks the onboard blue LED.

Note that the bootloader project is fully customizable, so you can use different methods for entering the bootloader mode, and different feedback mechanisms. For example, you can change the project so that it waits at power-up for a specified amount of time for a new application to be loaded rather than through the use of a button press. All resources of the PSoC are available for use by the bootloader project.

For details on UART bootloader in PSoC 4, please refer AN68272 - PSoC® 3, PSoC 4 and PSoC 5LP UART Bootloader. The landing page of the application note also includes a video providing an overview of a UART bootloader.

Figure 5-1. UART Bootloader



The Bootloader Base Project includes the source code in the *main.c* and the *UART_Btld.c* files, which support bootloading the PSoC 4 device. The source code is available for reference, but is not necessary to create bootloadable applications.

5.2 Bootloadable Code Example

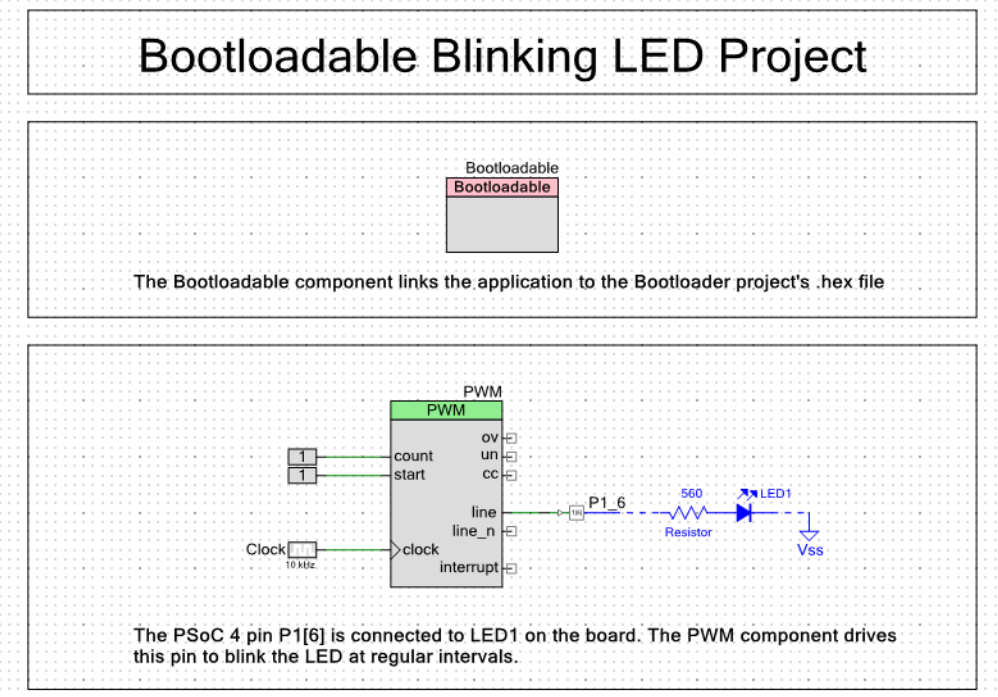
The example in Programming a CY8CKIT-049-4xxx Project Using the Bootloader on page 18 showed how to bootload the application code for a blinking LED project into the device using the USB-Serial controller.

In the bootloadable code example, the following components are used:

- Bootloadable
- PWM
- Clock
- Digital Output Pin
- Digital Constants (logic HIGH/LOW)
- Off-Chip Components (external resistor, LED, and Vss)

In this code example, the PWM component is used to drive an output on a pin connected to the user LED. The bootloadable component is placed to ensure that your application code is correctly mapped to the target PSoC 4 bootloader flash-space mapping.

Figure 5-2. Bootloadable Blinking LED Project



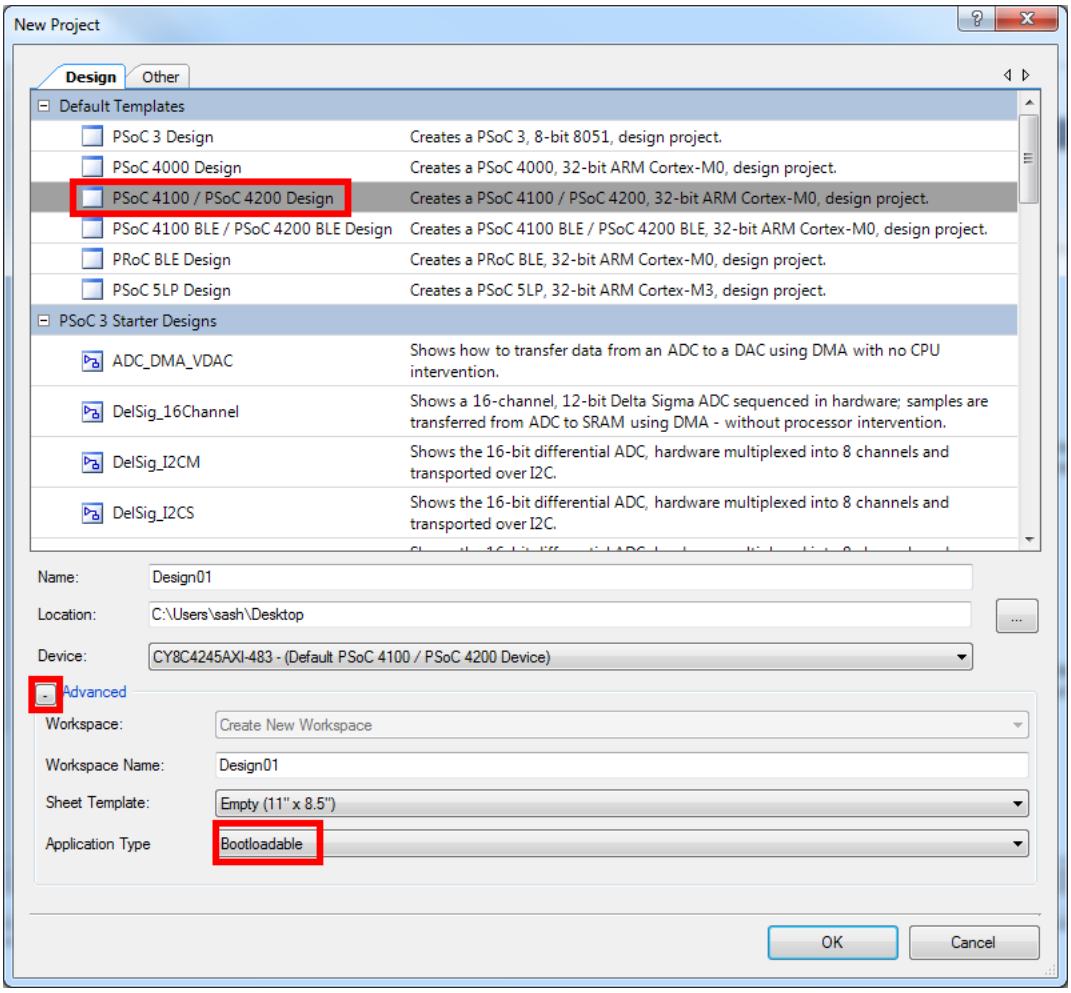
5.3 Creating a New Bootloadable Project

To create a new bootloadable project, do the following:

1. On the **Start Page** of PSoC Creator, click **Create New Project**.
2. On the New Project window, select an **PSoC 4100/PSoC 4200 Design**.
3. Enter a name for the project and select a workspace. You can either add the new project to an existing workspace or create a new one.
4. Set the device to the PSoC 4 device on your CY8CKIT-049-4xxx. For example, the CY8CKIT-049-42XX kit requires the CY8C4245AXI-483 device.
5. Click the + button next to **Advanced**, set the **Application Type** as **Bootloadable**.

6. Click **OK**.

Figure 5-3. Creating a New Bootloadable Project

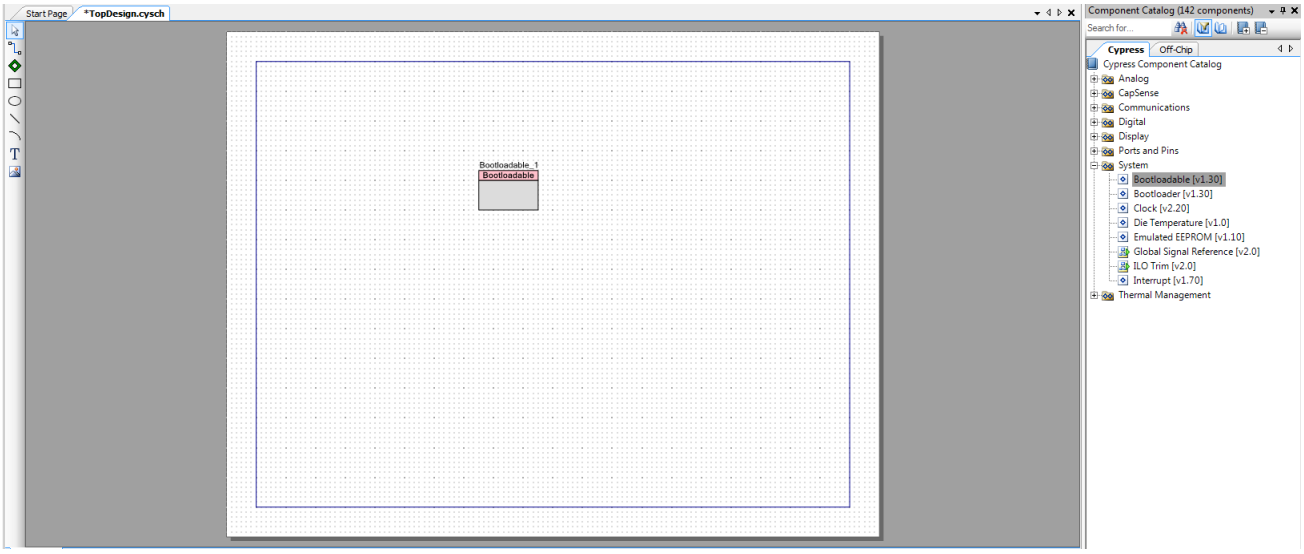


PSoC Creator generates a new project.

7. Navigate to the schematic view to place your components (double-click on the *.cysch* file from the project in Workspace Explorer).

   Select the **Page 1** tab in the schematic if it is not already selected. The key component that must be added is the Bootloadable component, which is used to generate the bootloadable application code.

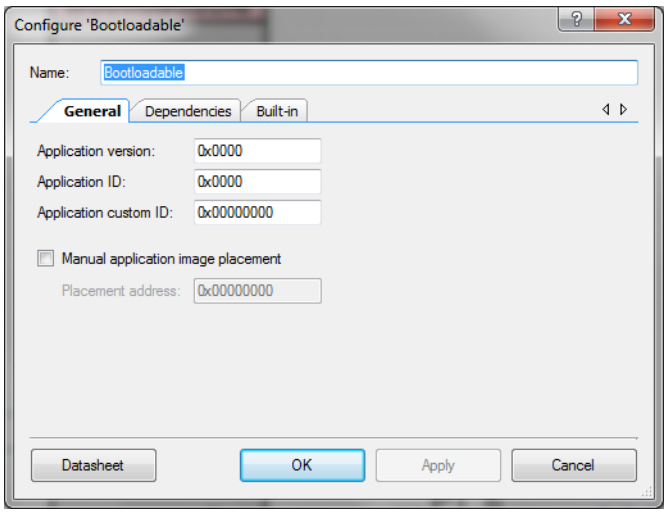
Figure 5-4. Bootloadable Project Schematic



8. Double-click the **Bootloadable** component to configure the selections.

   The selections must be the same as those in the code example. Refer to Programming a CY8CKIT-049-4xxx Project Using the Bootloader on page 18.

Figure 5-5. Configure Bootloadable Component

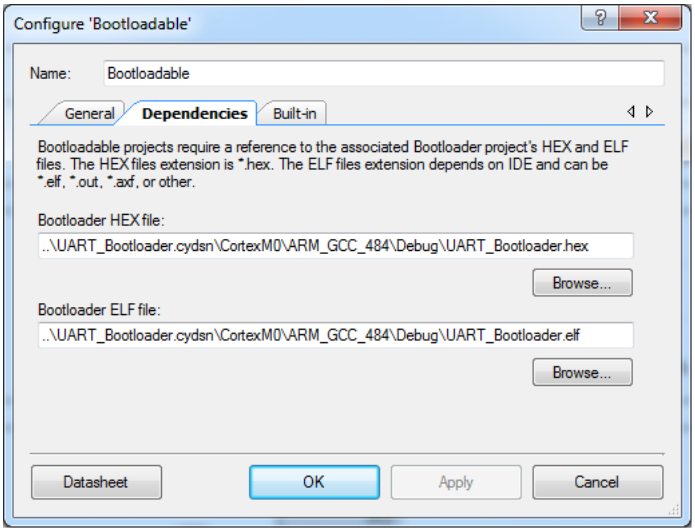


9. Click the **Dependencies** tab to select the HEX and ELF files from the *UART Bootloader* project included with the kit.

   You must always point your bootloadable project to a base bootloader project. The bootloader project can be in the same workspace as your bootloadable project, but this is not necessary. This code example uses the default application shipped with CY8CKIT-049-4xxx.

10. Click **Apply** and then click **OK**.

Figure 5-6. Specifying the References



11. After the project builds without errors, follow the steps shown in Programming a CY8CKIT-049-4xxx Project Using the Bootloader on page 18 to bootload the new code into the target using the Bootloader Host application.

This example does not have any source code, but is a base code example. Follow the steps in the next examples to add functionality to this base project.

5.4 Converting a Non-bootloadable Project to a Bootloadable Project

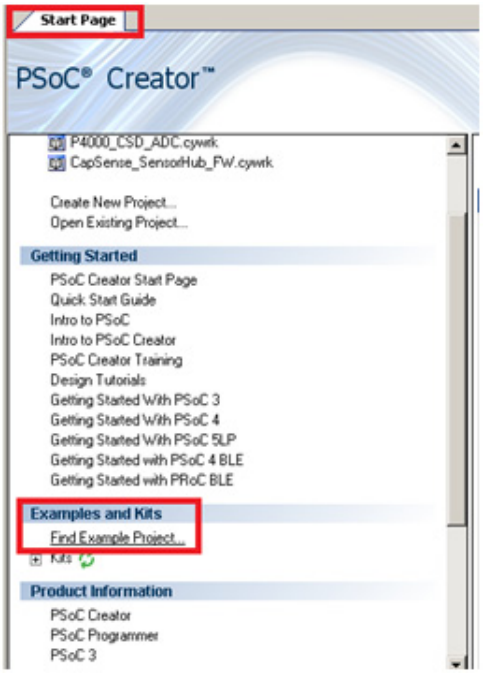
As part of PSoC Creator application notes and kits available at www.cypress.com, Cypress provides many code examples that you can work with. Most of these code examples do not include a bootloader in the project and are not directly usable with the CY8CKIT-049 kit. However, with a very small modification to the project, they can be easily made to work with the CY8CKIT-049 kit. The following section describes how to port a code example that is not bootloadable into a bootloadable example. We will take a simple PSoC Creator code example, **PWMExample**, for this purpose. You can follow the same steps to port any non-bootloadable project to a bootloadable project.

5.4.1 PWMExample

This example explains how to get the **PWMExample** project from the PSoC Creator code examples list into your workspace and target the CY8CKIT-049-4xxx development kits.

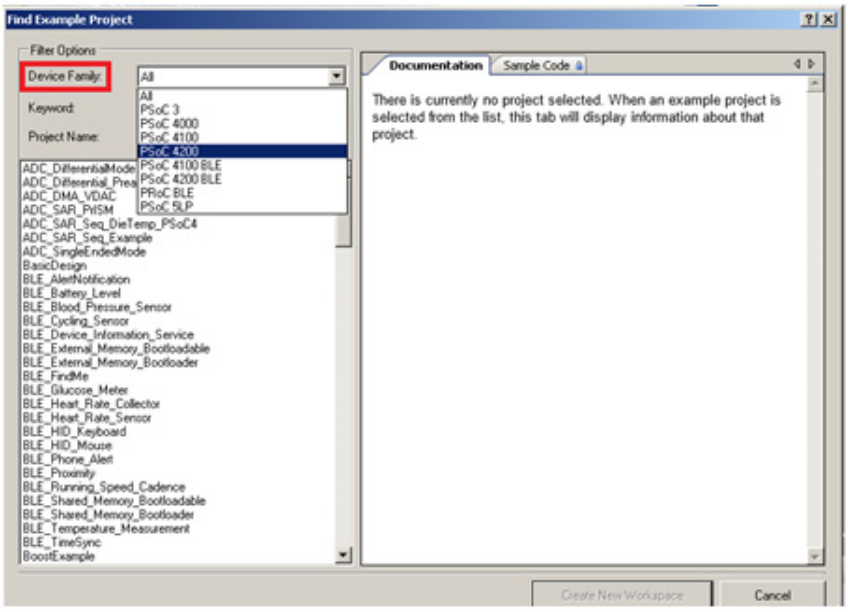
1. In PSoC Creator, click **Find Example Project**… under **Start Page** > **Examples and Kits**.

Figure 5-7. Opening “Find Example Project” Window



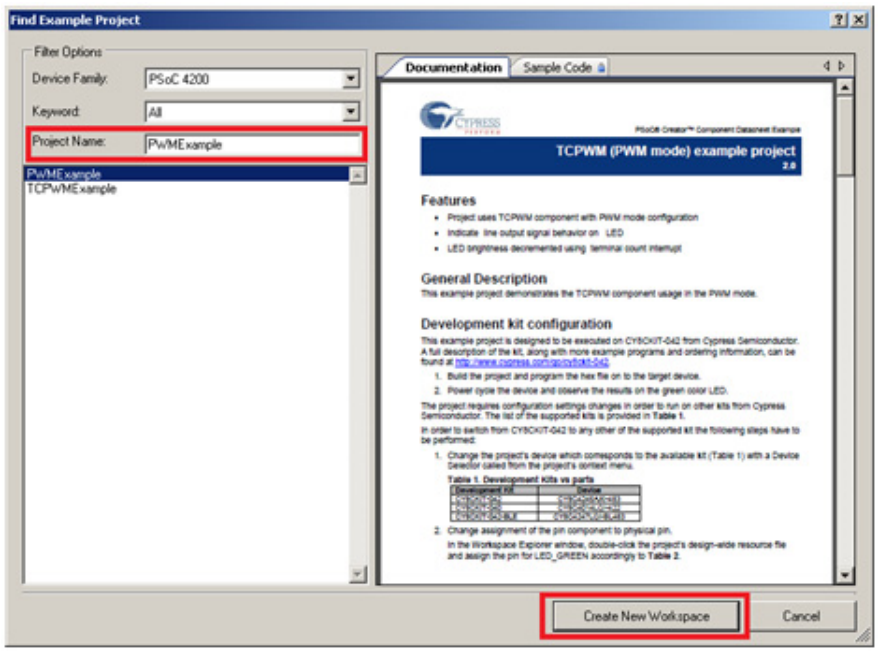
2. Set the **Device Family** option to the kit family; select **PSoC 4200** for the CY8CKIT-049-42xx kit and **PSoC 4100** for the CY8CKIT-049-41xx kit. The PSoC 4200 device family and CY8CKIT-049-42xx are used in this example.

Figure 5-8. Selecting Device Family



3. You should now see all the examples supported by PSoC Creator for the selected family. In the **Project Name** field, enter ‘PWMExample’. Select the **PWMExample** project from the list displayed and click **Create New Workspace**.

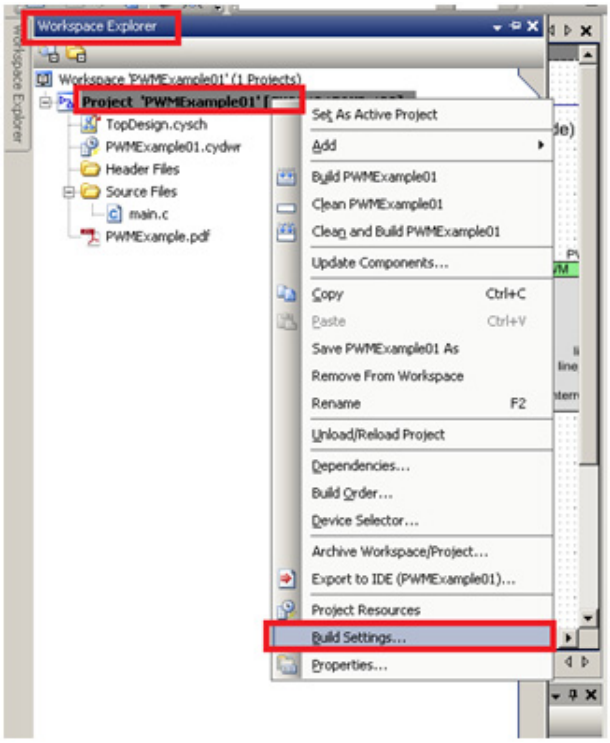
Figure 5-9. PWMExample Project Lookup



4. In the dialog box that appears, navigate to the folder location where you want to create the project.
5. After the workspace is created, PSoC Creator opens the help document detailing the project features and content. You can close this document and open it from the Workspace Explorer when required.

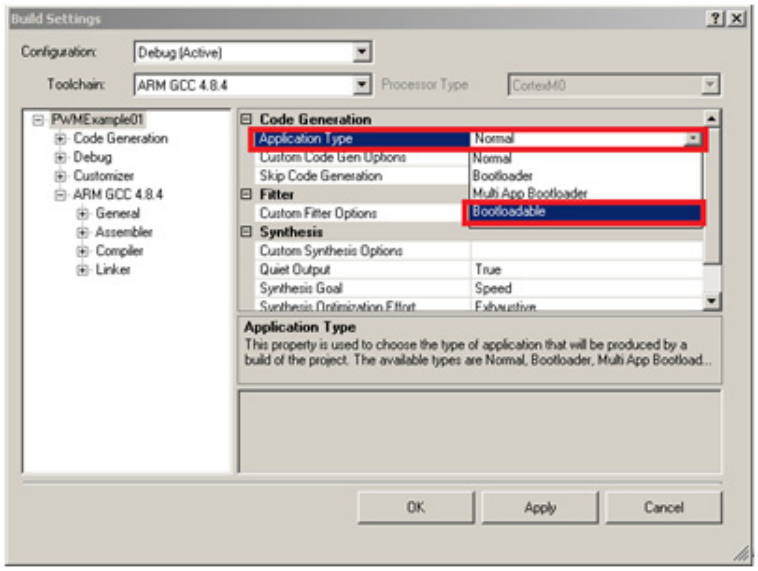
6. When the project opens in PSoC Creator, navigate to the Workspace Explorer window, right-click on the project, and select **Build Settings**.

Figure 5-10. Accessing Build Settings



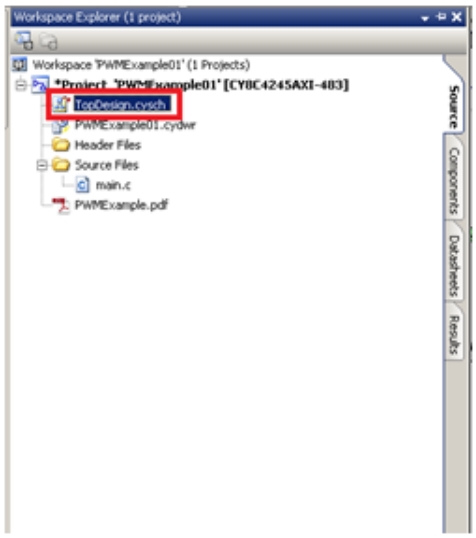
7. Because this project was originally set as a *Normal* **Application Type**, we need to change it to *Bootloadable*. In the **Application Type** drop-down menu, select *Bootloadable*. Click **Apply** and then **OK**.

Figure 5-11. Project Build Settings



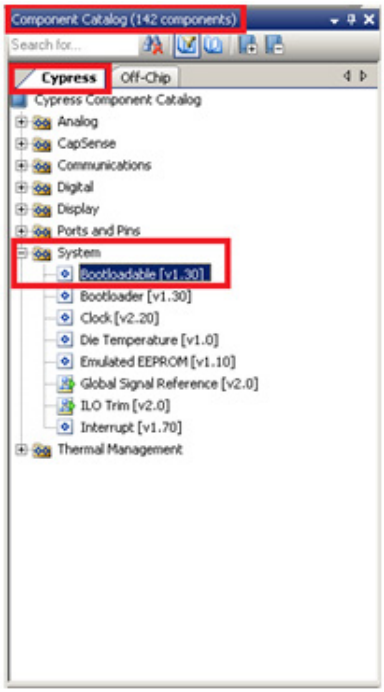
8. Double-click the **TopDesign.cysch** file from the Workspace Explorer to open the schematic view.

Figure 5-12. Opening “TopDesign.cysch”



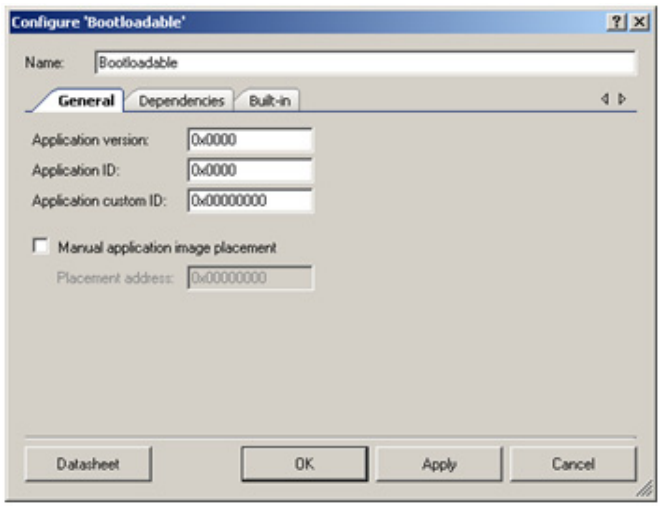
9. Drag and drop the **Bootloadable** component into the schematic window from the **Component Catalog** under **System**.

Figure 5-13. Bootloadable Component in the Component Catalog



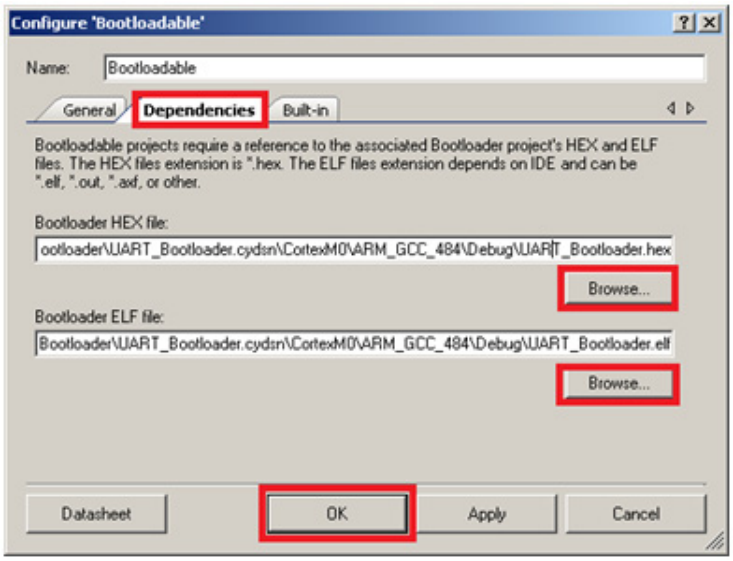
10. Double-click on the placed bootloadable component to configure the selections.

Figure 5-14. General Tab Parameters



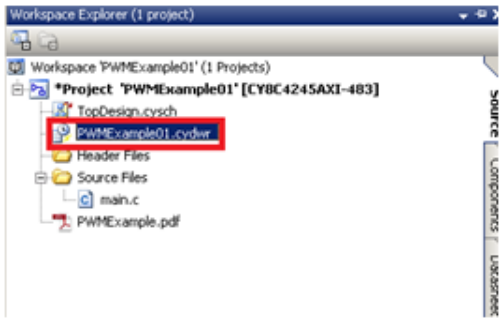
11. Click the **Dependencies** tab to select the .hex and .elf files from the UART Bootloader project included with the kit (<Install_Directory>\CY8CKIT-049-42xx\<version>\Firmware\SCB_Bootloader\ UART_Bootloader.cydsn\CortexM0\ARM_GCC_484\Debug\). This is done to point the bootloadable project to the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

Figure 5-15. Dependencies Tab Parameters



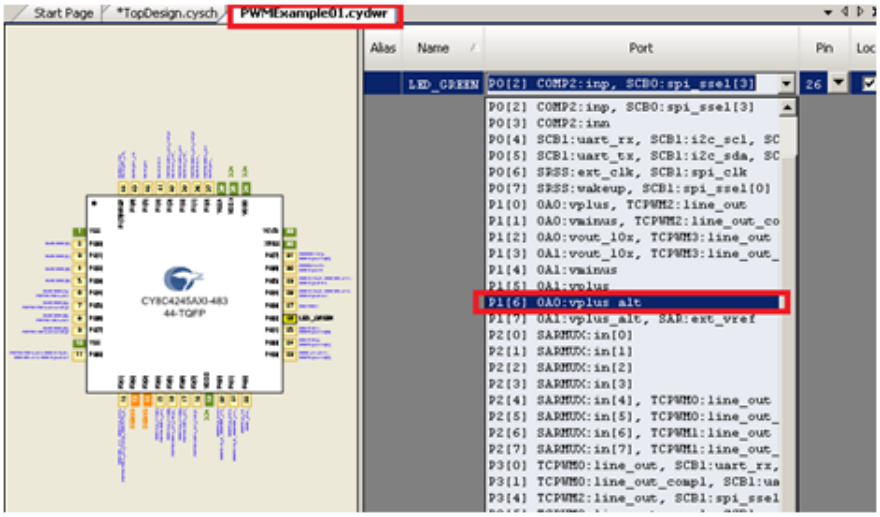
12. Because the project is not designed for the CY8CKIT-049 kit, you need to change the PWM output pin to the LED pin in CY8CKIT-049 (P1[6]). To do this, open **PWMExample.cydwr** from the Workspace Explorer.

Figure 5-16. Opening ".cydwr" File



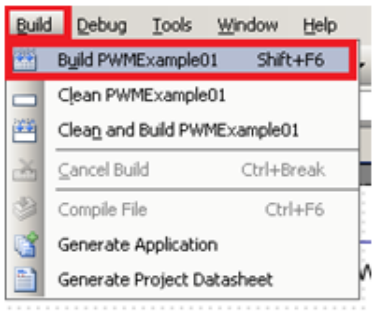
13. Select the **LED_GREEN** pin to P1[6] in the CY8CKIT-049 kit.

Figure 5-17. 'cydwr' Pin Settings



14. Select **Build** > **Build PWMExample**.

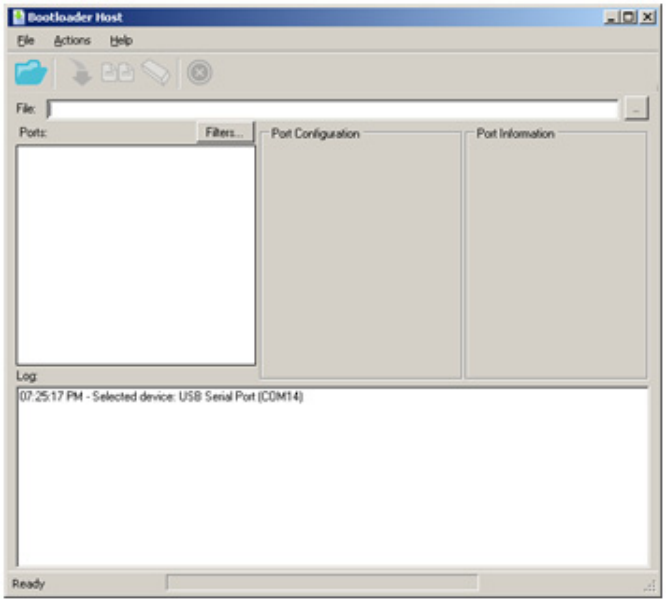
Figure 5-18. Building the Project



15. Connect the CY8CKIT-049-4xxx kit to the PC while pressing the SW1 button; this will put the device into Bootloader mode. Entry into Bootloader mode is shown by the blinking LED.

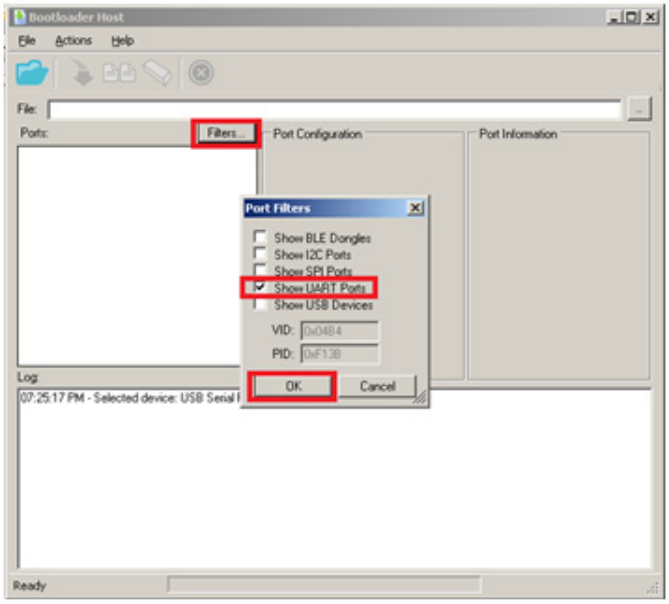
16. Open the Bootloader Host utility by selecting **Tools** > **Bootloader Host** from the PSoC Creator menu. Connect to the COM port and make your port configurations.

Figure 5-19. Bootloader Host Tool



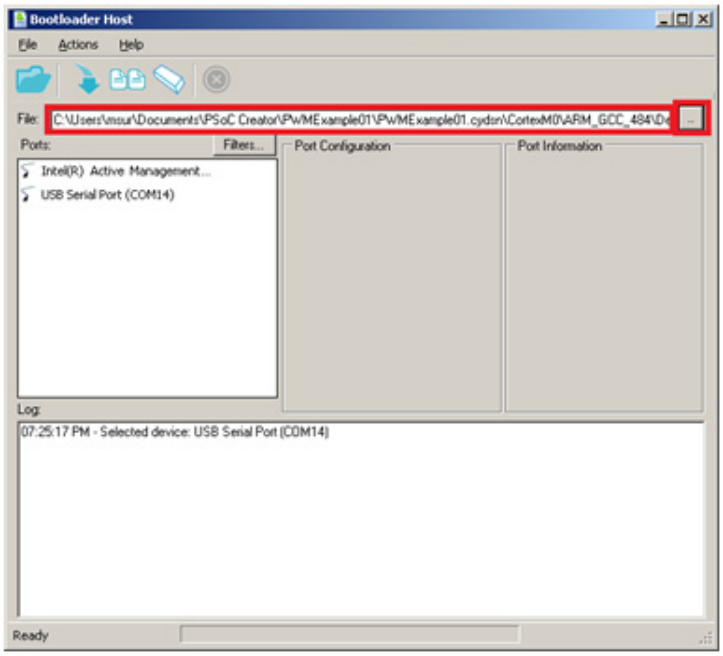
17. Click **Filters…**; configure it as shown in Figure 14 and then click **OK**.

Figure 5-20. Bootloader Host “Ports Filters” Settings



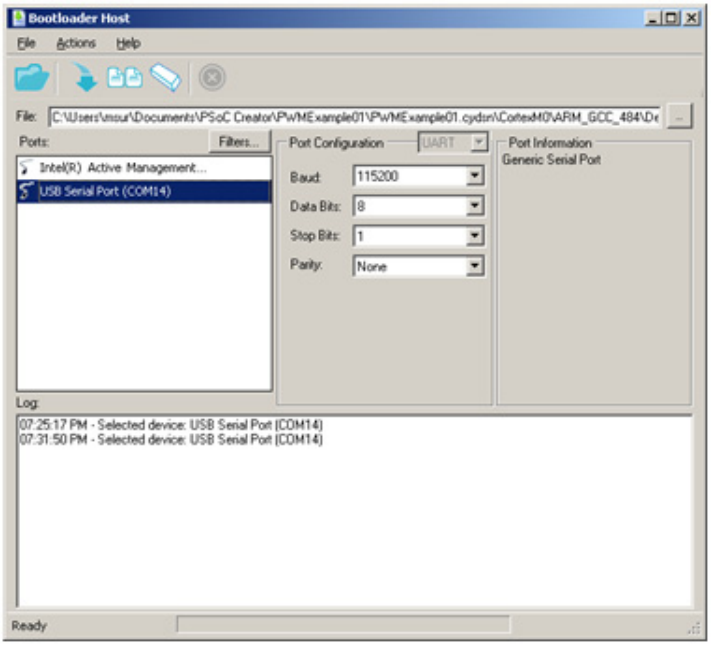
18. Click the **Open File** button and navigate to the project's .cyacd file available at the following path: <Project Directory>\PWMExample01\PWMExample01.cydsn\CortexM0\ARM_GCC_484\Debug

Figure 5-21. Bootloader Host "cyacd" File Load Setting



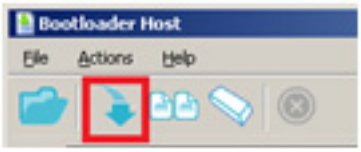
19. Select the **USB Serial Port (COMxx)** corresponding to your kit and configure the settings, as shown in Figure 5-22.

Figure 5-22. USB Serial Port Selection



20. Click the **Program** button. When the firmware is programmed, observe that the LED moves from low intensity to higher intensity in steps of 100. You can change the BRIGHTNESS_DECREASE macro to modify the step size.

Figure 5-23. Program Button in Bootloader Host



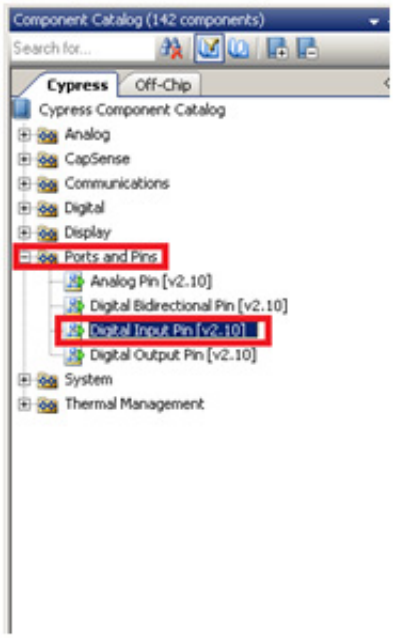
5.4.2 Entering Bootloader Mode from the Bootloadable Application

To program the PSoC 4 device on the CY8CKIT-049 kit, you need to enter the bootloader mode through which you can load the desired application via the Bootloader Host utility and the USB-Serial bridge. The default bootloader in the onboard PSoC 4 enters the bootloader mode when the SW1 button is pressed during board/device power up. This method of entering the bootloader can be time consuming and you may not prefer removing the board from the PC each time you want to program it. You can also enter the bootloader mode using the Bootloadable component present in the application project. When you want to enter bootloader mode, say when a button is pressed for a predefined duration or a command is sent via UART, all you need to do is detect the condition to enter bootloader mode from the bootloadable application and call the “Bootloadable_Load()” API when the condition is met. Let us use the example in PWMExample on page 41 and modify the project to enter bootloader mode from the bootloadable application when SW1 is pressed for more than 2 seconds.

Follow these steps to modify the project in PWMExample on page 41 to enter bootloader mode from the application.

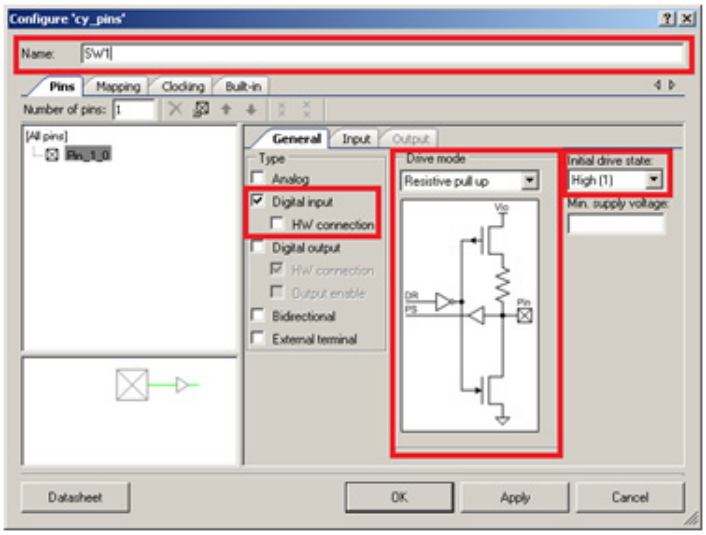
1. Open **TopDesign.cysch** of the project; drag and drop a **Digital Input Pin** component into the schematic view.

Figure 5-24. Digital Input Pin Component in Component Catalog



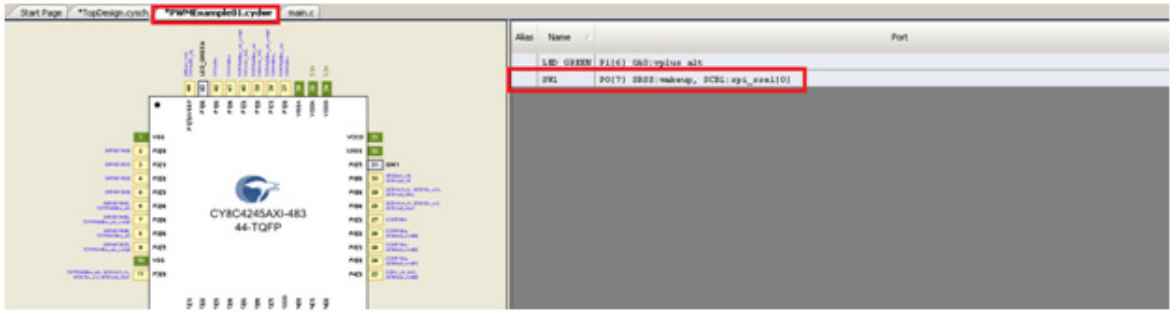
2. Double-click the placed component and configure the parameters as shown in Figure 5-25 and click **OK**.

Figure 5-25. Digital Input Pin Parameter Configuration



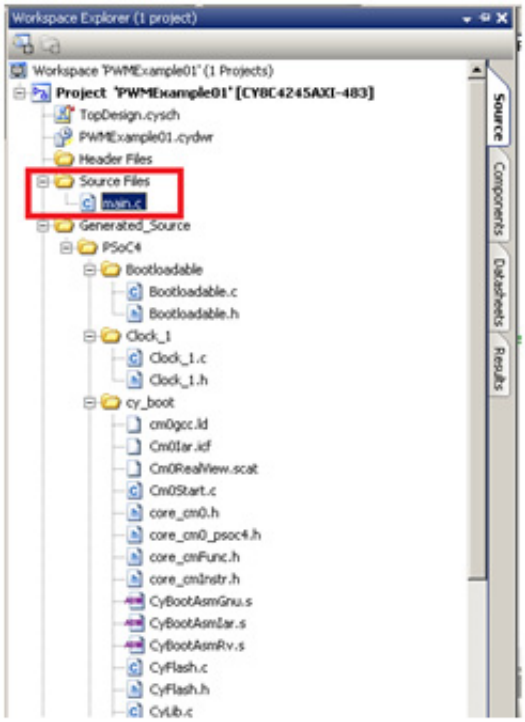
3. In the *PWMExample01.cydwr* file, select **P0[7]** as the pin connection for SW1.

Figure 5-26. 'cydwr' Pin Settings



4. Open main.c file from the Workspace Explorer.

Figure 5-27. Opening main.c



5. Add the following code inside the infinite “for” loop. Note that you need to define a variable ‘uint8 swCounter’ in the beginning of the main loop.

```
for(;;)
    {
        /* Reset the software counter if SW1 is not pressed (Pulled high)
*/
if(SW1_Read())
{
swCounter = 0;
}
else
{
/* Increment the counter if SW1 is pressed (Shorted to GND) */
swCounter++;
}

/* Provide a 10 ms delay to make the counter approximately periodic every
10 ms */
CyDelay(10);

/* Check if the software counter has passed 2 s (200 * 10 ms)
==> Since the software counter is incremented only when SW1 is
==> pressed, SW1 is pressed for more than 2 seconds
==> Enter bootloader mode */
if(swCounter > 200)
{
Bootloadable_Load();
}
    }
```

6. Build the project using **Build** > **Build PWMExample01**.
7. Program the PSoC 4 device as described from Step 15 in PWMExample on page 41.
8. When the new application is loaded, the output of the project will remain the same as in PWMExample on page 41. If you press and hold SW1 for 2 seconds without unplugging the device from the PC, the device will enter bootloader mode (the LED will blink). To exit the bootloader mode, unplug and re-plug the board from the USB port.

6. USB-Serial Configuration

The CY8CKIT-049-4xxx Prototyping Kits support the CY7C6521x family of USB controller products. The CY7C6521x devices are a family of full-speed USB-Serial bridge controllers. These bridge controllers offer configurable serial channels for UART, I2C, SPI, or GPIO interfaces, with the industry's lowest power consumption in the stand-by mode (5 µA).

USB-Serial bridge controllers integrate the CapSense capacitive-touch sensing technology and USB-IF Battery Charging specification version 1.2. These controllers are ideal for applications such as portable medical devices (such as blood-glucose meters), point-of-sales terminals, serial cables (including USB-to-UART and RS-232 cables) and other applications requiring USB connectivity.

CY8CKIT-049-4xxx development kits use the USB-Serial device to provide connectivity to a PC and to perform USB-UART bootload programming. The following sections provide instructions on how to use and configure the USB-Serial device on the CY8CKIT-049-4xxx kits.

6.1 USB-Serial Resources

Use the following links to access a wide variety of content that includes custom software, utilities, datasheets, and knowledge base articles.

- USB-Serial landing page: www.cypress.com/go/usbserial
- Software and drivers: USB-Serial Software Development Kit (SDK)
- Datasheets: CY7C6521x Device Families
- Additional information: Knowledge Base Articles

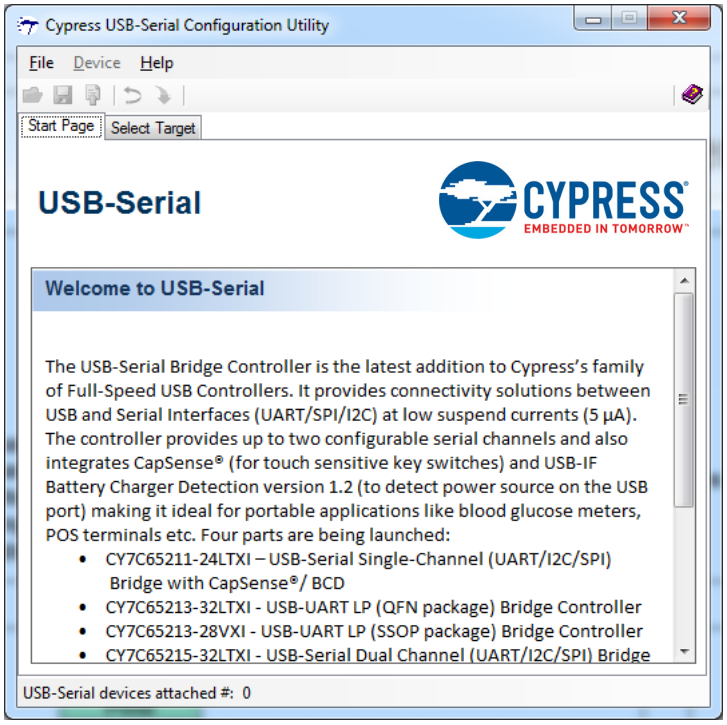
6.2 USB-Serial Configuration Utility

Cypress USB-Serial Configuration Utility is an application included in the USB-Serial software development kit (SDK) installation. This utility is used to configure the USB-Serial device configuration, and helps use additional capabilities of USB-Serial device such as USB-UART configurations, USB-GPIO controls, and custom development using the USB-I2C and USB-SPI protocols.

After you install the USB-Serial SDK, click **Start** > **All Programs** > **Cypress** > **Cypress USB Serial** > **Cypress USB-Serial Configuration Utility** to launch the USB-Serial Configuration Utility.

Note: For Windows 8 or higher OS, launch Cypress USB-Serial Configuration Utility by double-clicking on <Install_Directory>\USB-Serial SDK\<version>\bin\USB Serial Configuration Utility.exe.

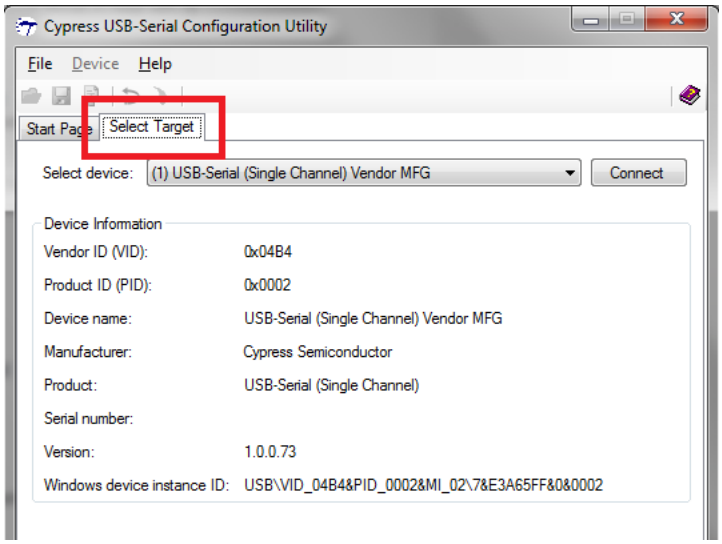
Figure 6-1. USB-Serial Configuration Utility



6.2.1 Connecting to a USB-Serial Device

1. Connect the CY8CKIT-049-4xxx prototyping kit to the PC.
2. Open the USB-Serial Configuration Utility.
3. Select the **Select Target** tab.

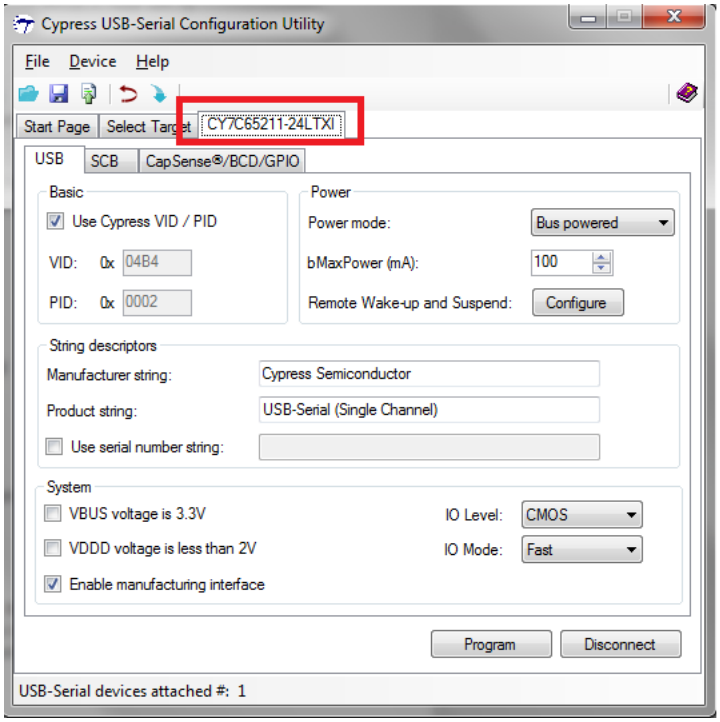
Figure 6-2. Selecting the Target in USB-Serial Configuration Utility



The USB-Serial Configuration Utility will automatically detect that the USB-Serial Device has been connected to the PC and will display the device in the **Select Device** drop-down menu.

4. Click **Connect**.
   After connecting to the device, a new tab opens that displays the device marketing part number.

Figure 6-3. Selecting the Connected Device



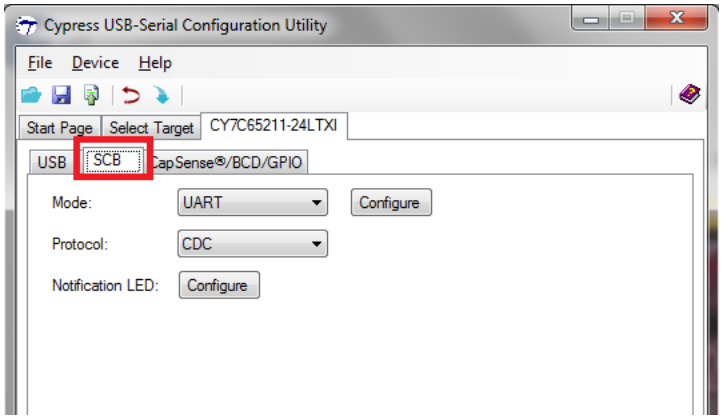
5. Select the new tab and begin configuring the device.

6.2.2 Configuring a Serial Port

The USB-Serial device acts as a USB-UART bridge for the CY8CKIT-049-4xxx development kit. You can use the Configuration Utility to read the default settings and configure new UART settings.

1. After connecting to the USB-Serial device, click the **CY7C65211-24LTXI** tab.
2. Select the **SCB** tab under the **CY7C65211-24LTXI** tab to see the default UART settings on the USB-Serial device.

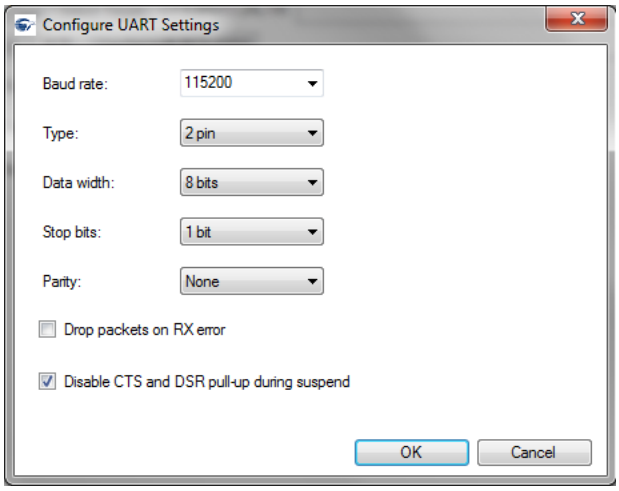
Figure 6-4. Configuring the Serial Port



3. Click **Configure** next to the UART mode select.

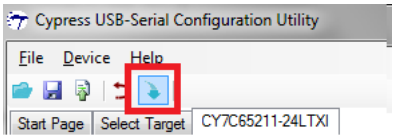
   The Configure UART Settings window appears, which displays the default settings for your UART-Serial device.

Figure 6-5. Configuring UART Settings



4. Change the UART settings such as **Baud Rate** or **Type** by selecting the new values from the respective drop-down lists, and click **OK**.
5. Click the **Program Device** button from the menu options at the top of the Configuration Utility to program the device with the new settings.

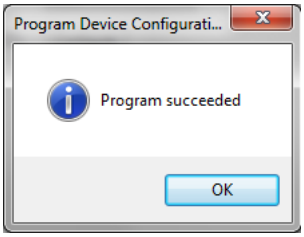
Figure 6-6. Programming the Device With UART Settings



When the configuration has been programmed to the target device, a popup window will be shown letting you know that programming was successful.

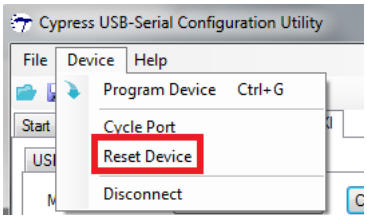
Note: Configurations will not be set immediately. You will need to reset the COM port for the settings to be applied.

Figure 6-7. Confirmation for Device Programming



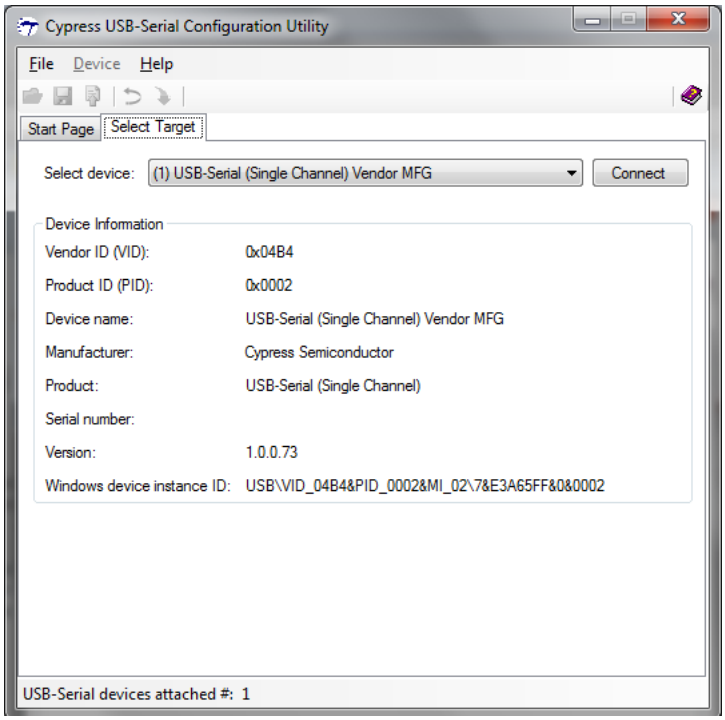
6. To reset the device from the Configuration Utility, navigate to the **Device** menu and select **Reset Device**. This initiates a reset to the device.

Figure 6-8. Resetting the Device



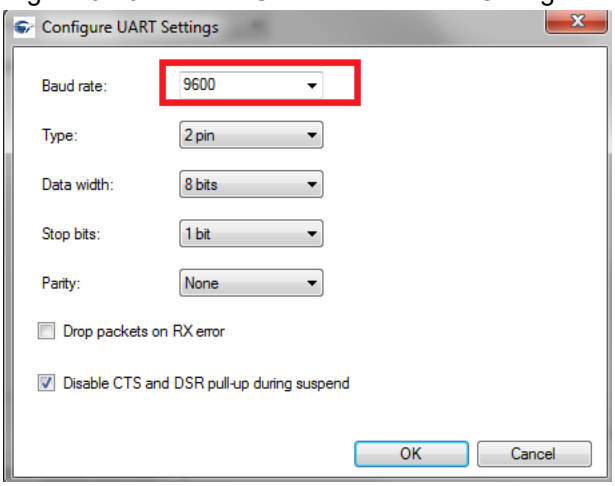
The utility will immediately detect that the device has been reconnected and display the **Select Target** tab.

Figure 6-9. Device After Reconnection



7. Connect to the device and navigate to the UART configuration window to see that the new configuration has been set. In this example, the **Baud Rate** is changed from 115200 to 9600.

Figure 6-10. Device UART Parameters Changed



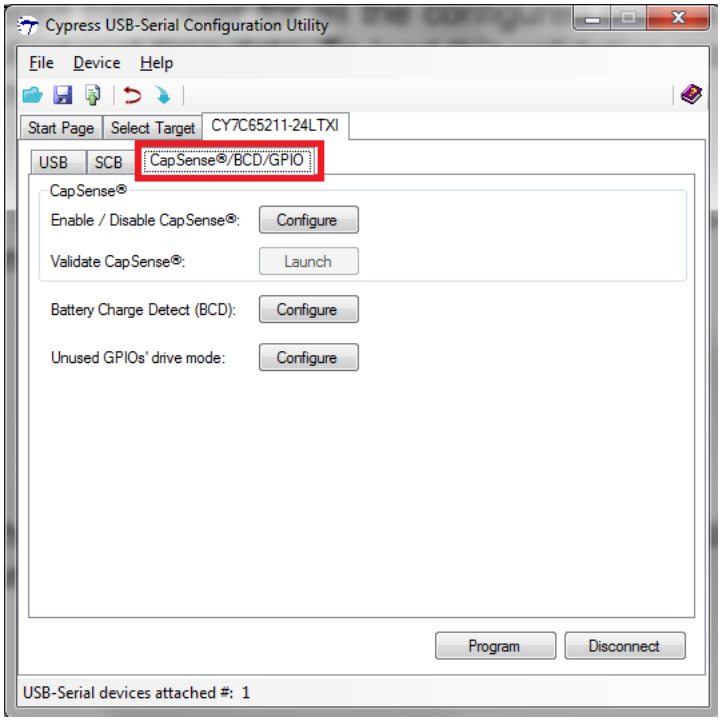
6.2.3 Configuring GPIOs

The USB-Serial device included in the CY8CKIT-049-4xxx Prototyping Kit also supports GPIO controls through the J6 header. Each of the serial protocols requires a different number of GPIO pins. Based on your serial configuration, the number of available GPIOs will change. The Configuration Utility will only display the available GPIOs based on your serial configuration. For more information on the serial configuration and the respective GPIO consumption, refer to the USB-Serial device datasheet.

1. Plug CY8CKIT-049-4xxx into a USB port on the PC, and connect to the USB-Serial device using the Configuration Utility.

2. Navigate to the **CapSense/BCD/GPIO** tab.

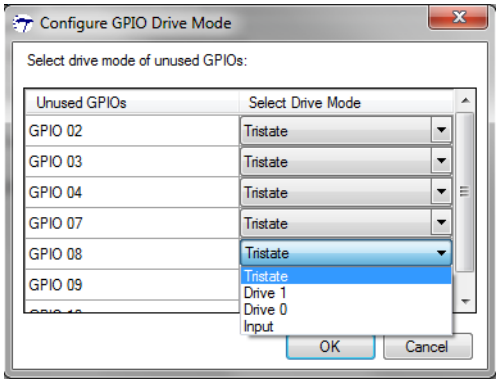
Figure 6-11. Device GPIO



3. Click **Configure** on the **Unused GPIOs** drive mode. This launches the GPIO configuration window.

   This example shows how to change the output mode of the GPIO 08 pin to drive an output. You can connect the pin to the PSoC 4, an LED, or any external circuitry.
4. Click the **Select Drive Mode** drop-down menu for the GPIO 08 pin.

Figure 6-12. Configuring GPIO Drive Mode



5. Select **Drive 1** from the available options and click **OK**. This example makes the pin HIGH.
6. Program the new configuration into the device and cycle the port to see the new configuration applied. For example, if the GPIO 08 pin is connected to an LED, you will see that the LED is on.

6.2.4 Additional Features of the USB-Serial Device

Apart from the UART and the GPIO features described in earlier sections, the USB-Serial device included in the CY8CKIT-049-4xxx provides several other features, such as the following:

- USB-I2C (Master/Slave)
- USB-SPI (Master/Slave)
- Cypress CapSense (up to eight buttons)
- Battery Charging Detect (BCD)

For more information on these features, refer to the device datasheet and the USB-Serial Configuration Utility user guide. (Select **Help > Help Topics**).

Note: USB-UART works in the USB Communication Device Class (CDC), while all other configuration controls such as GPIO, SPI, and I2C use the Cypress vendor driver on the PC. Therefore, COM port tools such as PuTTY or HyperTerminal will only work for the UART bridge. You can use the C++ APIs to create scripts and tools included with the USB-Serial SDK to evaluate and control the other bridge options.

A. Appendix

A.1 CY8CKIT-049-4xxx Schematics

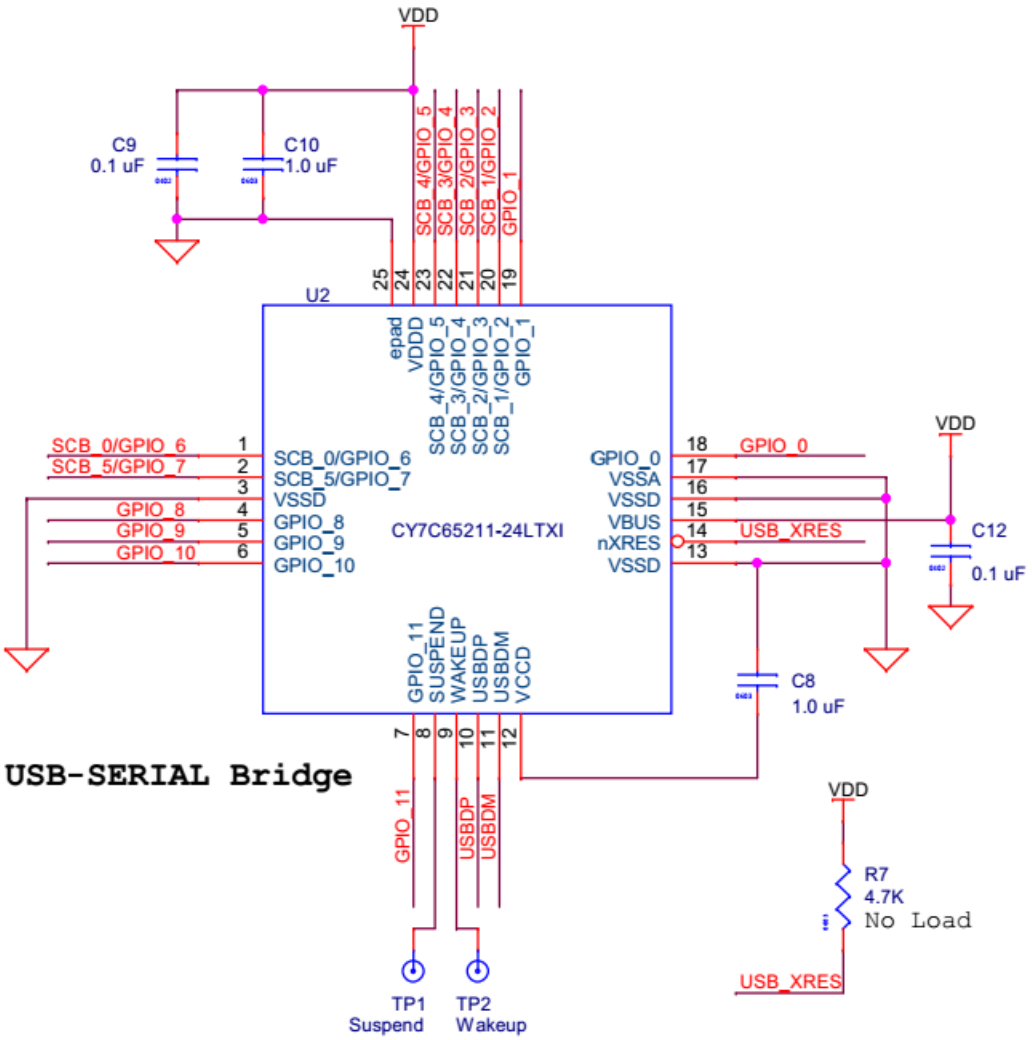


Figure A-1. CY8C4245AXI-483 (CY8CKIT-049-42XX only)

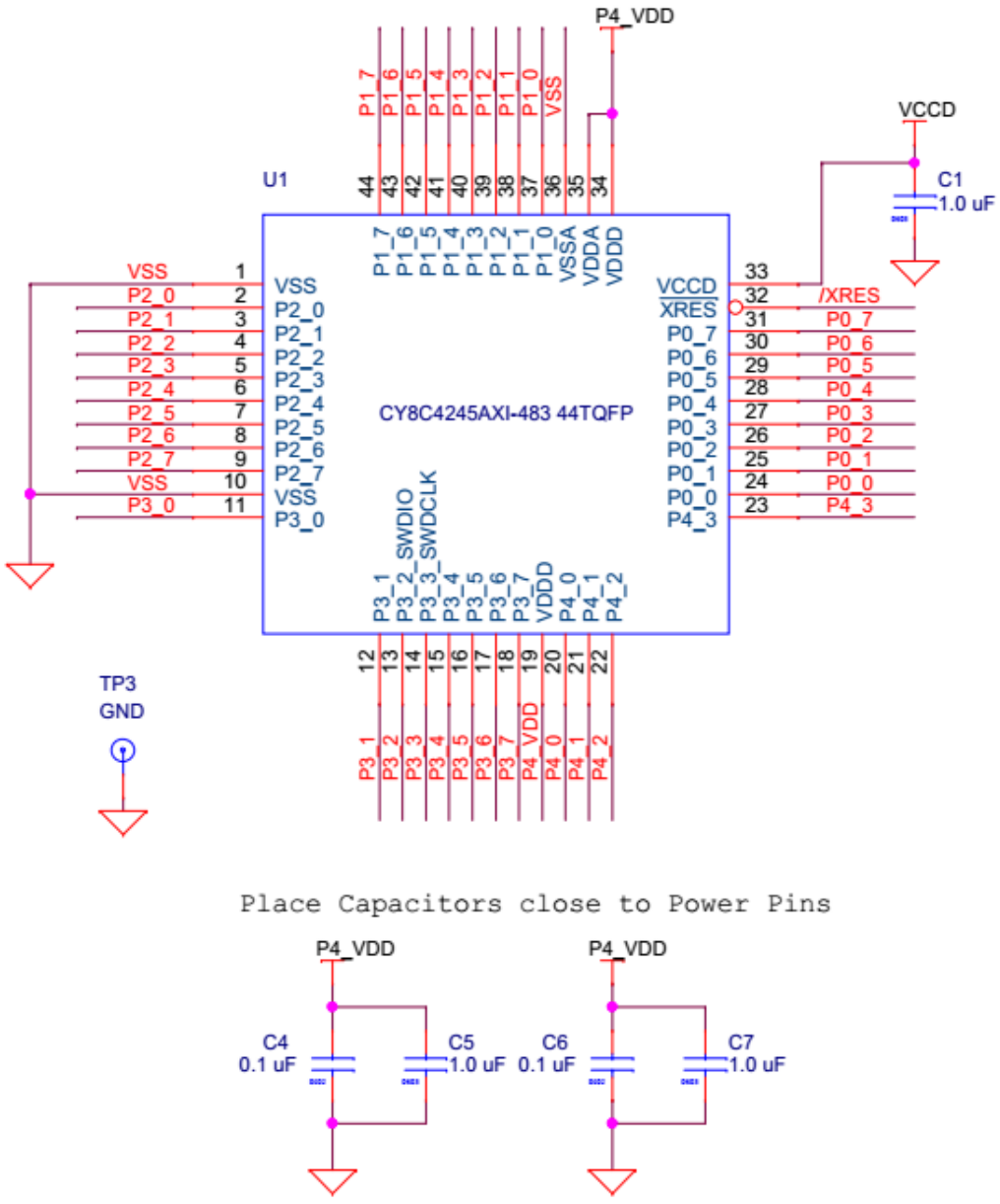
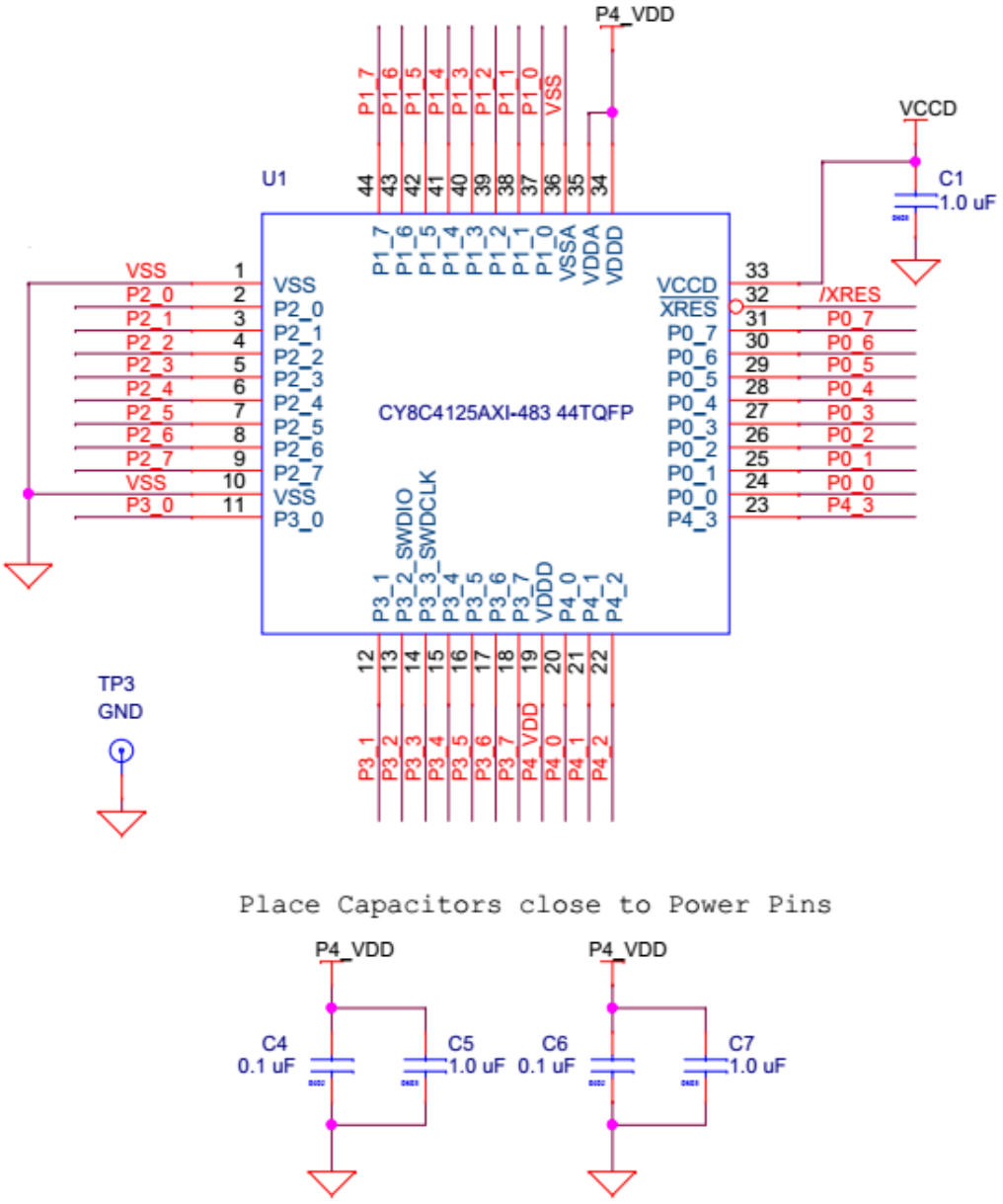
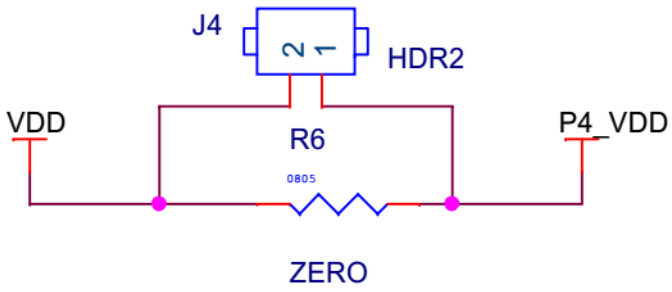
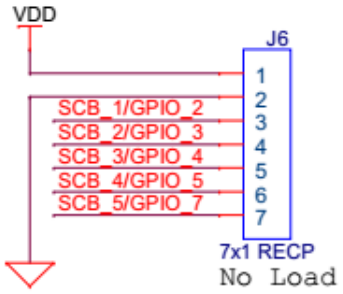
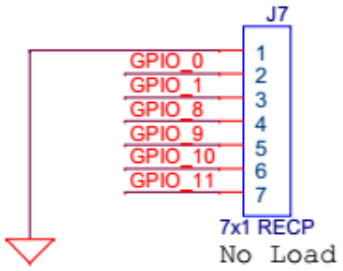


Figure A-2. CY8C4125AXI-483 (CY8CKIT-049-41XX only)

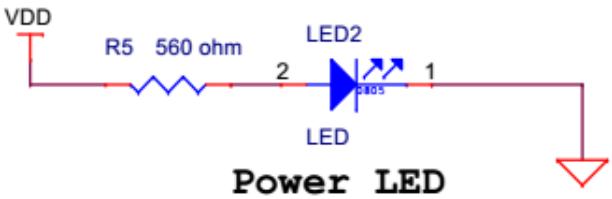




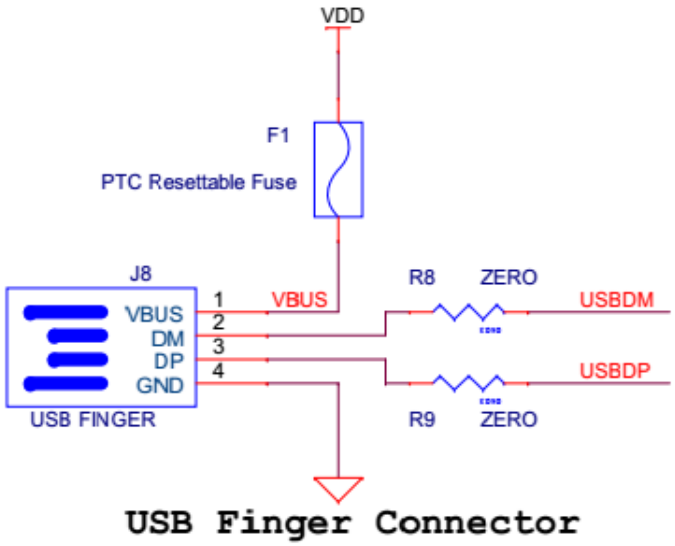




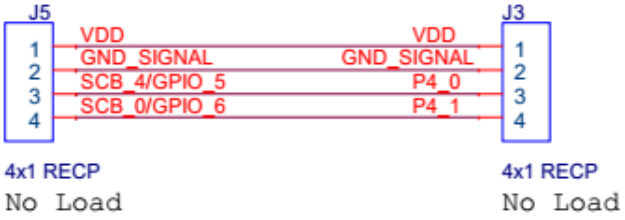
USB-SERIAL Headers



Power LED

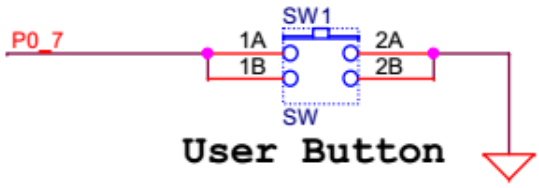


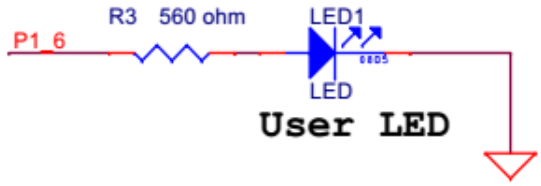
USB Finger Connector

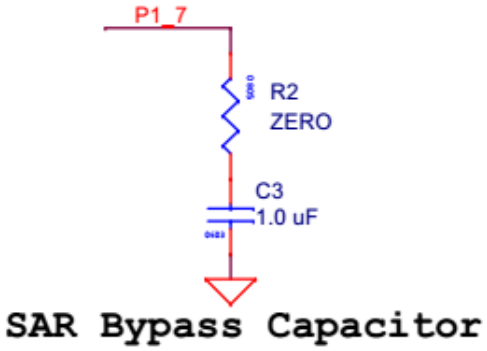


UART connection to PSoC 4

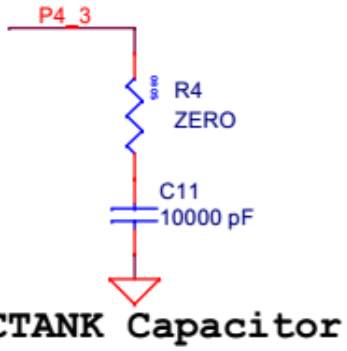

J1
1 P4_0
2 P4_1
3 P4_2
4 P4_3
5 P0_0
6 P0_1
7 P0_2
8 P0_3
9 P0_4
10 P0_5
11 P0_6
12 P0_7
13 P1_0
14 P1_1
15 P1_2
16 P1_3
17 P1_4
18 P1_5
19 P1_6
20 P1_7
21 VSS
22 P4_VDD
CONN 22
No Load
J2
1 P4_VDD
2 VSS
3 /XRES
4 P3_3
5 P3_2
6 P3_7
7 P3_6
8 P3_5
9 P3_4
10 P3_3
11 P3_2
12 P3_1
13 P3_0
14 P2_7
15 P2_6
16 P2_5
17 P2_4
18 P2_3
19 P2_2
20 P2_1
21 P2_0
22 VSS
CONN 22
No Load
PSoC 4 I/O Headers


P0_7
SW1
1A
1B
2A
2B
SW
User Button

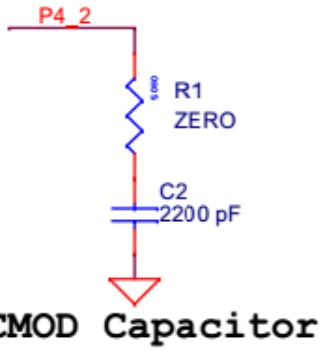

R3 560 ohm
LED1
P1_6
LED
User LED



SAR Bypass Capacitor



CTANK Capacitor



CMOD Capacitor

A.2 Programming a CY8CKIT-049-4xxx Project Using MiniProg3

To use MiniProg3 for programming, connect wires or a 5-pin 100-mil spaced header to the programming header on the CY8CKIT-049-4xxx board. The programming header is a 5-pin header indicated on the silkscreen and is labeled '**PROG**'. The suggested part numbers are
Molex 22-05-3051 (right angled male header) and Molex 22-23-2051 (straight header). The suggested part numbers for 5 pin headers are 5-pin Molex-KK right-angle male header
(Part number# 22-05-3051) or 5-pin Molex-KK vertical male header (Part number# 22-23-3051). Please see images below for orientation of the header. The CY8CKIT-049 supports both power cycle and reset programming modes.

Figure A-3. Connecting CY8CKIT-049-4xxx to MiniProg3 using right angled header

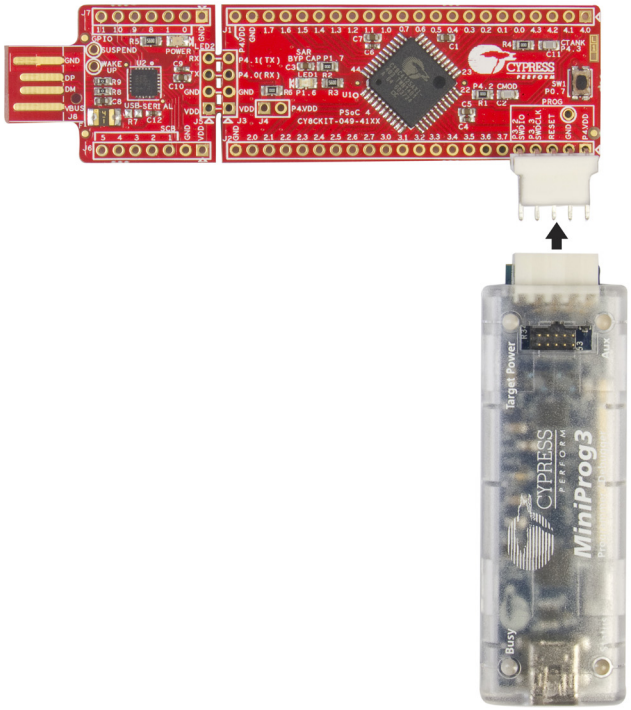
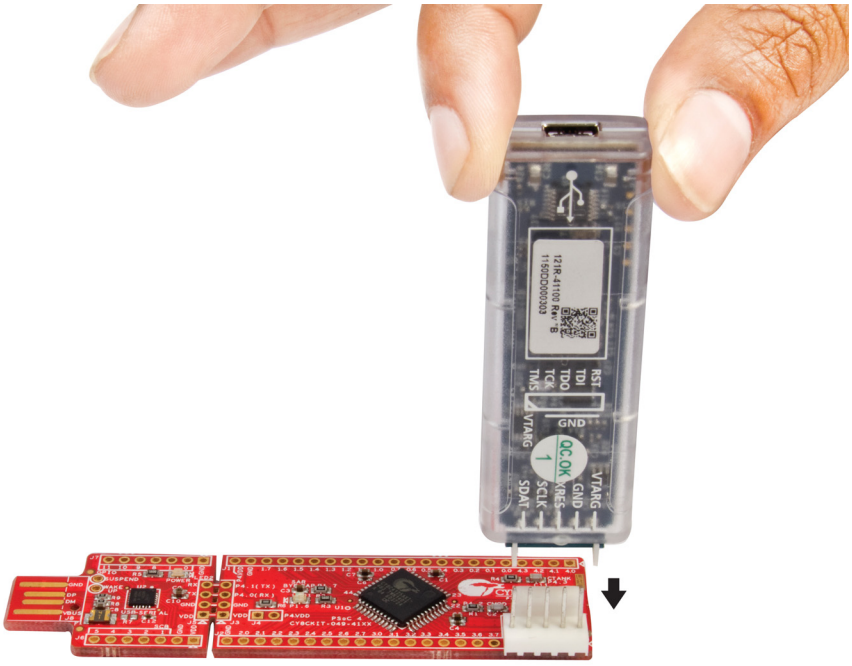


Figure A-4. Connecting CY8CKIT-049-4xxx to MiniProg3 using straight header



Note: CY8CKIT-002 MiniProg3 is not part of the PSoC 4 Prototyping Kit contents and can be purchased from the Cypress Online Store.

The following images show the pinout for MiniProg3 and the connections on CY8CKIT-049-4xxx.

Figure A-5. Pinout for MiniProg3

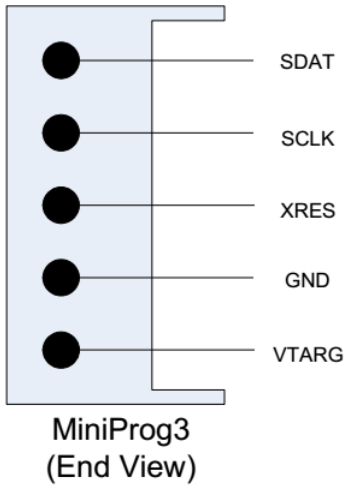
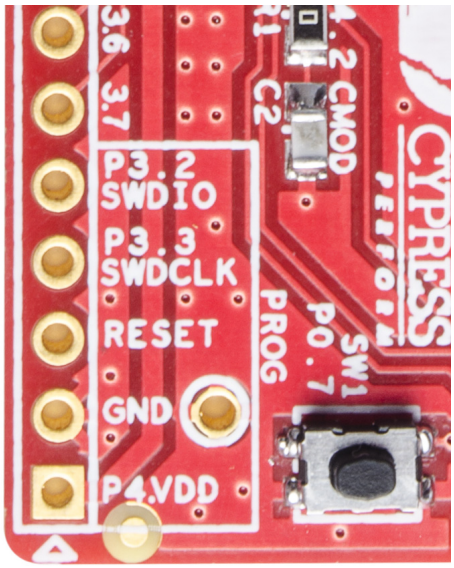


Figure A-6. CY8CKIT-049-4xxx Connections for MiniProg3

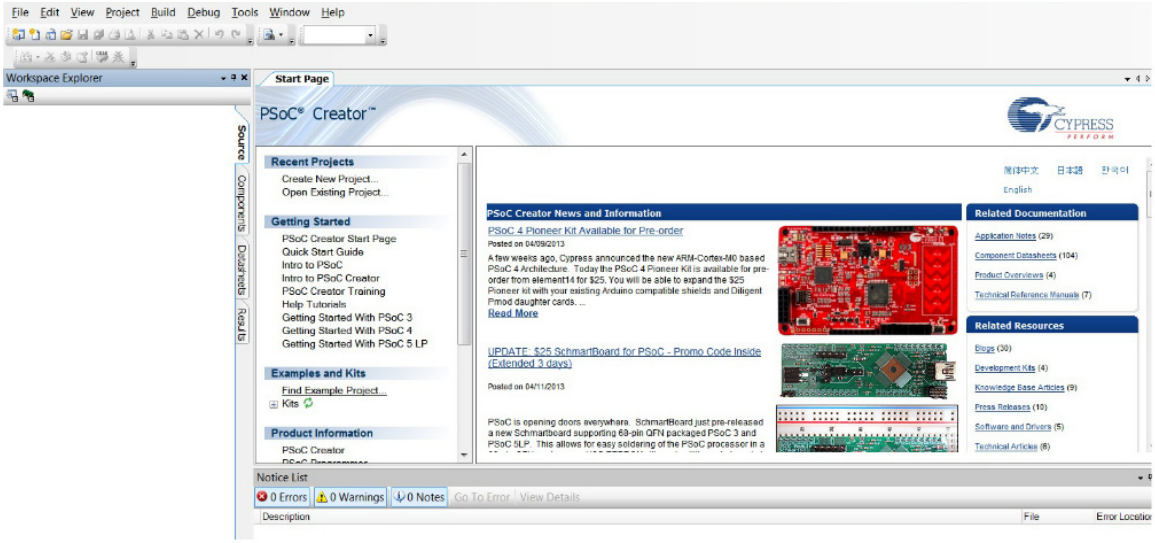


The code examples described in this section show how to bootload new projects into PSoC 4 and create bootloader and bootloadable projects. To access the kit examples, download the examples from the kit web page.

The initial example shows how to program the kit with just a bootloader using MiniProg3. Note that the kit is pre-programmed with a project containing a bootloader, so this step is not necessary to change the application firmware.

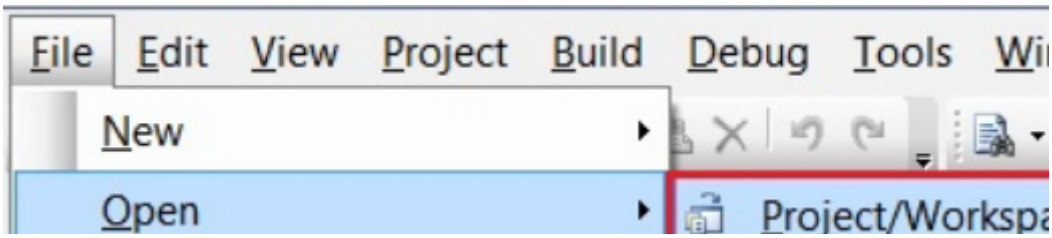
1. Launch PSoC Creator from the Start menu.

Figure A-7. PSoC Creator Start Page



2. Open the *SCB_Bootloader.cywrk* workspace by choosing **File** > **Open** > **Project/Workspace** and navigating to the directory in which your project is present.

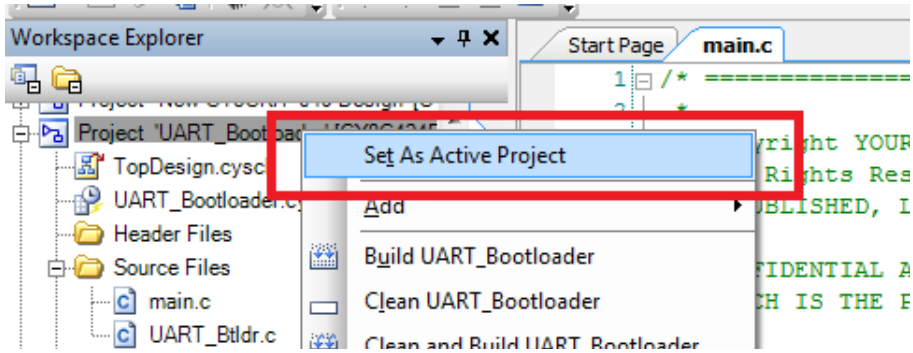
Figure A-8. Opening the Project in PSoC Creator



The workspace includes two sample projects linked in the Workspace Explorer - a bootloader project and a bootloadable project.

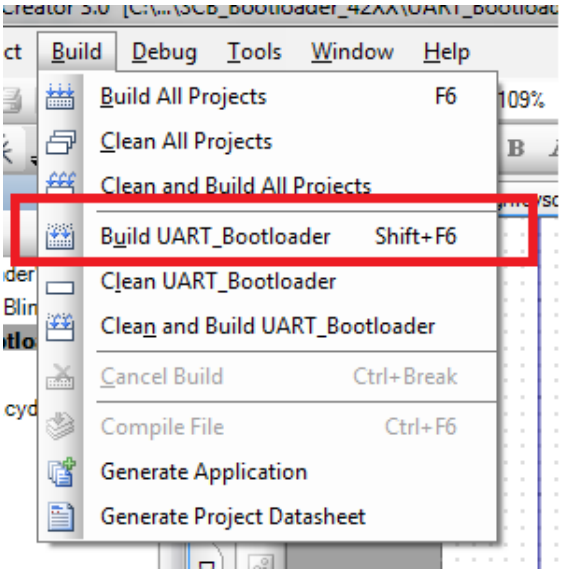
3. Right-click the *UART_Bootloader* project and select **Set As Active Project**.

Figure A-9. Setting Code Example as Active Project in Workspace Explorer



4. Select **Build > Build UART_Bootloader**.

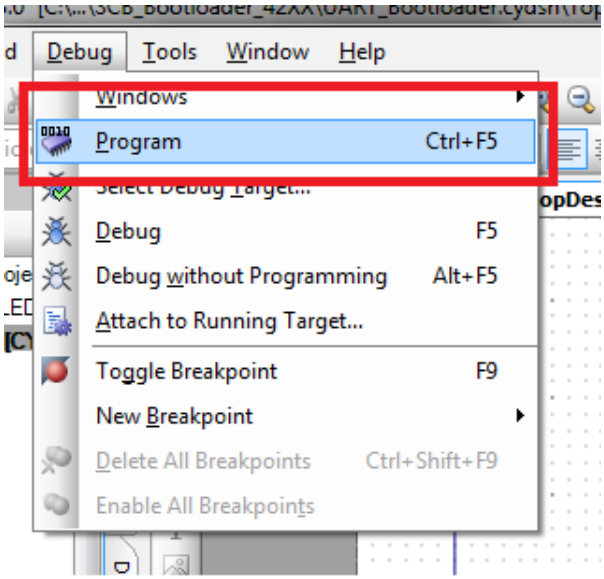
Figure A-10. Building the Project



5. Connect MiniProg3 to the CY8CKIT-049-4xxx prototyping board.

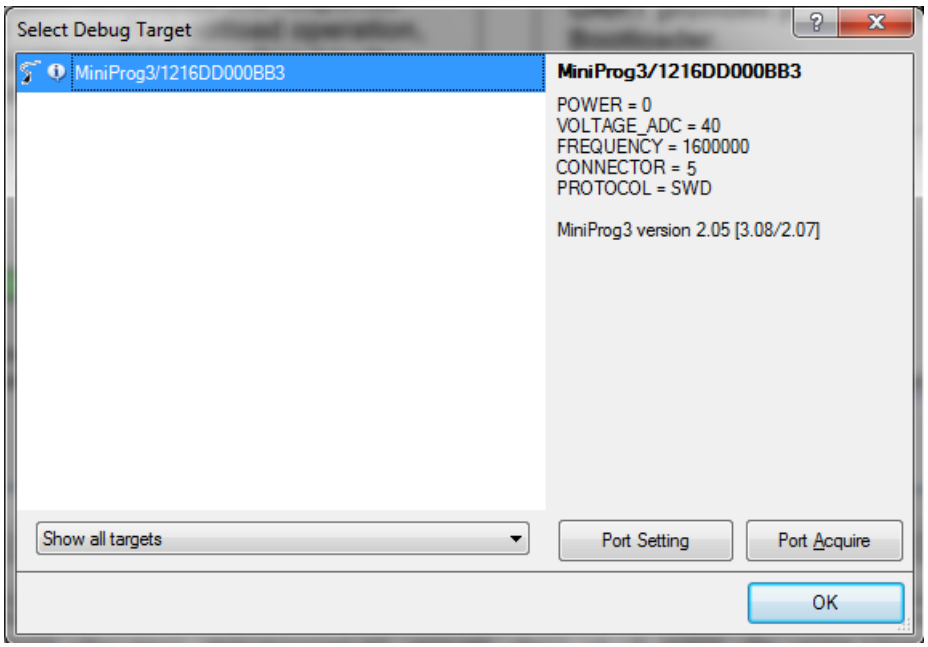
6. Select **Debug > Program**.

Figure A-11. Programming the CY8CKIT-049-4xxx



The Select Debug Target window opens.

Figure A-12. Debug Target Window



7. Click **Port Settings**, set the connector to **5** pin, and then click **OK**.
8. Click **Port Acquire** to detect the target device, click **Connect**, and then click **OK**.

PSoC Creator programs the target. When programming is complete, the PSoC 4 will contain the bootloader but not any application code.

To bootload any application code, refer:
"Programming a CY8CKIT-049-4xxx Project Using the Bootloader" on page 18.

A.3 Bill of Materials

Table A-1. Bill of Materials

Item	Qty	Reference	Value	Description	Mfr Name	Mfr Part Number
1	1	N/A	N/A	PCB, 92.13mm x 24.13mm, High Tg, ENIG finish, 2 layer, Color = RED, Silk = WHITE	Cypress Semiconductors	600-60178-01
2	6	C1,C3,C5,C7,C8, C10	1.0 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
3	1	C2	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata	GRM2165C1H222JA01D
4	4	C4,C6,C9,C12	0.1 uF	CAP .1UF 16V CERAMIC Y5V 0402	AVX Corporation	0402YG104ZAT2A
5	1	C11	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata	GRM2195C1H103JA01D
6	1	F1	FUSE	PTC Resettable Fuses 15Volts 100Amps	Bourns	MF-MSMF050-2
7	1	LED1	Blue User LED	LED BLUE CLEAR 0805 SMD	Lite-On Inc	LTST-C170TBKT
8	1	LED2	Power LED Amber	LED AMBER 591NM DIFF LENS 2012	Sharp Microelectronics	LT1ZV40A
9	4	R1, R2, R4, R6	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic	ERJ-6GEY0R00V
10	2	R3, R5	560	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic	ERJ-6GEYJ561V
11	2	R8, R9	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V
12	1	SW1	Switch	SWITCH TACTILE SPST-NO 0.05 A 32 V	C&K Components	KMR221GLFS
13	1	U1	PSoC 4	44TQFP PSoC4A target chip	Cypress Semiconductor	CY8C4245AXI-483 (4200 family) CY8C4125AXI-483 (4100 family)
14	1	U2	CY7C65211-24LTXI	24QFN USB-Serial Bridge	Cypress Semiconductor	CY7C65211-24LTXI
No Load Components						
15	1	R7	4.7 K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic-ECG	ERJ-3GEYJ472V
Label						
16	1	N/A	N/A	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60140-01 REV 01 (YYWWVVXXXXX)	Cypress Semiconductor	
17	1	N/A	N/A	LBL, PCBA Anti-Static Warning, 10mm X 10mm	Cypress Semiconductor	

Revision History

Document Revision History

Document Title: CY8CKIT-049-4xxx PSoC® 4 Prototyping Kit Guide Document Number: 001-90711				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
**	4270301	02/03/2014	RKAD	New kit guide.
*A	4312949	03/20/2014	RKAD	Updated Kit Operation chapter on page 16: Updated "Programming a CY8CKIT-049-4xxx Project Using the Bootloader" on page 18: Replaced "Programming the CY8CKIT-049-4xxx Kit" with "Programming a CY8CKIT-049-4xxx Project Using the Bootloader" in heading. Updated description. Removed "Programming a CY8CKIT-049-4xxx Project Using MiniProg3". Removed sub-heading "Programming a CY8CKIT-049-4xxx Project Using the Bootloader" only (as contents are coming under main heading). Updated Appendix chapter on page 62: Added "Programming a CY8CKIT-049-4xxx Project Using MiniProg3" on page 69.
*B	4416482	06/23/2014	RKAD	Updated "Title" in the file properties.
*C	4473817	10/09/2014	SASH	Replaced "USB Serial" with "USB-Serial" in all instances across the document. Updated Introduction chapter on page 7: Updated "Additional Learning Resources" on page 8: Updated description. Updated "Document Conventions" on page 12: Replaced "PSoC Designer User Guide" with "PSoC Creator User Guide".

Document Revision History *(continued)*

Document Title: CY8CKIT-049-4xxx PSoC® 4 Prototyping Kit Guide Document Number: 001-90711				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
*C (cont.)	4473817	10/09/2014	SASH	Updated Software Installation chapter on page 13: Renamed Chapter “Kit Installation” as “Software Installation”. Removed “Kit Software”. Added “Before You Begin” on page 13. Added “CY8CKIT-049-41xx/CY8CKIT-049-42xx Software” on page 13. Added “Install Software” on page 14. Updated “Uninstall Software” on page 14: Updated description. Added “Uninstall USB-Serial Drivers” on page 15. Updated “Open the “PSoC 4 Code” Code Example in PSoC Creator” on page 15: Updated description. Updated Kit Operation chapter on page 16: Updated “CY8CKIT-049-4xxx USB COM Port” on page 17: Updated description. Updated “Programming a CY8CKIT-049-4xxx Project Using the Boot-loader” on page 18: Updated description. Updated Figure 3-7. Updated Figure 3-11. Updated Appendix chapter on page 62: Updated “Programming a CY8CKIT-049-4xxx Project Using MiniProg3” on page 69: Updated description.
*D	4652397	02/05/2015	SASH	Updated Introduction chapter on page 7: Updated “Additional Learning Resources” on page 8: Updated description. Added “PSoC Creator” on page 9. Added “PSoC Creator Code Examples” on page 10. Added “PSoC Creator Help” on page 11. Added “Technical Support” on page 12. Updated Software Installation chapter on page 13: Updated “CY8CKIT-049-41xx/CY8CKIT-049-42xx Software” on page 13: Updated description. Updated “Install Software” on page 14: Updated description. Updated “Open the “PSoC 4 Code” Code Example in PSoC Creator” on page 15: Updated description.

Document Revision History *(continued)*

Document Title: CY8CKIT-049-4xxx PSoC® 4 Prototyping Kit Guide Document Number: 001-90711				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
*D (cont.)	4652397	02/05/2015	SASH	Updated Kit Operation chapter on page 16: Updated "CY8CKIT-049-4xxx USB COM Port" on page 17: Updated description. Updated "Programming a CY8CKIT-049-4xxx Project Using the Bootloader" on page 18: Updated description. Updated Figure 3-11. Updated "USB-UART Default Settings" on page 25: Updated description. Updated Table 3-2 (Updated caption only). Updated Code Examples chapter on page 35: Updated "Creating a New Bootloadable Project" on page 37: Updated description. Updated Figure 5-3. Updated Figure 5-4. Updated Figure 5-6. Removed "Adapting Projects from 100 Projects in 100 Days". Updated USB-Serial Configuration chapter on page 53: Updated "USB-Serial Resources" on page 53: Updated description. Updated "USB-Serial Configuration Utility" on page 54: Updated Figure 6-1. Updated Appendix chapter on page 62: Updated "Programming a CY8CKIT-049-4xxx Project Using MiniProg3" on page 69: Updated description. Updated Figure A-3. Added Figure A-4.
*E	4675832	03/02/2015	SASH	Updated Hardware chapter on page 26: Updated "Functional Description" on page 27: Updated "Header Connections" on page 28: Updated "Functionality of J3 and J5 Headers (PSoC 4 to USB-Serial)" on page 30: Updated Table 4-4: Updated details in "Pin" column. Updated Figure 4-6. Updated "Functionality of J6 and J7 Headers (USB-Serial)" on page 31: Updated Table 4-5: Updated details in "Pin" column. Updated Table 4-6: Updated details in "Pin" column.

Document Revision History *(continued)*

Document Title: CY8CKIT-049-4xxx PSoC® 4 Prototyping Kit Guide Document Number: 001-90711				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
*F	4690296	03/19/2015	SASH	Updated Introduction chapter on page 7: Updated “Additional Learning Resources” on page 8: Updated description. Updated “Technical Support” on page 12: Updated description. Updated Software Installation chapter on page 13: Updated “Install Software” on page 14: Updated description. Removed figure “The 'Cypress' icon”. Removed figure “USB-Serial Driver Installation window”. Removed figure “USB-Serial Driver installation - Finish page”. Removed “Uninstall USB-Serial Drivers” on page 15. Updated Code Examples chapter on page 35: Updated description. Updated “Bootloader Base Code Example” on page 35: Updated description. Updated “Bootloadable Code Example” on page 36: Updated description. Updated USB-Serial Configuration chapter on page 53: Updated “USB-Serial Configuration Utility” on page 54: Updated description.
*G	4716987	04/08/2015	MSUR	Updated Introduction chapter on page 7: Updated “Additional Learning Resources” on page 8: Updated description. Updated Code Examples chapter on page 35: Updated description. Added “Converting a Non-bootloadable Project to a Bootloadable Project” on page 40.
*H	5612049	01/31/2017	RKAD	Updated to new template. Completing Sunset Review.
*I	5739971	05/17/2017	AESATMP8	Updated logo and Copyright.
*J	6201368	06/08/2018	SRDS	Updated Kit Operation chapter on page 16: Updated Figure 3-1.