CY8CKIT-042

PSoC® 4 Pioneer Kit Guide

Doc. # 001-86371 Rev. *B

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl): +1.408.943.2600
http://www.cypress.com

Copyrights

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress Datasheets. Cypress believes that its family of products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Contents

Safety Information

Regulatory Compliance

The PSoC 4 Pioneer kit, CY8CKIT-042 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. Due to this reason the board 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, the user may be required to take adequate preventive measures. Also, 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-042 as shipped from the factory has been verified to meet with requirements of CE as a Class A product.

The CY8CKIT-042 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-042 boards in the protective shipping package.

End-of-Life / Product Recycling

This Kit has an end-of-life cycle after five years from the date of manufacturing mentioned on the back side of the box. Please contact your nearest recycler for discarding the kit.

General Safety Instructions

ESD Protection

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

Handling Boards

CY8CKIT-042 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 Pioneer Kit. This Pioneer kit is designed as an easy-to-use and inexpensive development kit, showcasing the unique flexibility of the PSoC 4 architecture. Designed for flexibility, this kit offers footprint-compatibility with several third party Arduino™ shields. This kit has a provision to populate an extra header to support Digilent® Pmod™ Peripheral modules. In addition, the board features a CapSense® slider, an RGB LED, a pushbutton switch, an integrated USB programmer, a program/debug header and USB-UART/I2C bridges. This kit supports either 5 V or 3.3 V as power supply voltages.

The PSoC 4 Pioneer Kit is based on the PSoC 4200 device family, delivering a programmable platform for a wide range of embedded applications. 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 re-configurable analog and digital blocks with flexible automatic routing.

1.1 Kit Contents

The PSoC 4 Pioneer kit contains:

1. PSoC 4 Pioneer board
2. Quick Start Guide
3. USB Standard Type A to USB Mini B cable
4. Jumper Wires

Figure 1-1. Kit Contents

Inspect the contents of the kit; if you find any part missing, contact your nearest Cypress sales office for help: www.cypress.com/go/support.

1.2 PSoC® Creator™

PSoC Creator is a state-of-the-art, easy-to-use integrated design environment (IDE). It introduces revolutionary hardware and software co-design, powered by a library of pre-verified and pre-characterized PSoC Components™.

With PSoC Creator, the user can:

- Drag-and-Drop PSoC Components to build a schematic of your custom design
- Automatically place and route Components and configure GPIOs
- Develop and debug firmware using the included Component APIs

PSoC Creator also enables the user to tap into an entire tools ecosystem with integrated compiler chains and production programmers for PSoC devices.

For more information, please visit www.cypress.com/Creator.

1.3 Getting Started

This user guide helps the user to get acquainted with the PSoC 4 Pioneer kit. The Software Installation chapter on page 11 describes the installation of the PSoC 4 Pioneer kit software. The Kit Operation chapter on page 17 describes the kit operation. It explains how to program the PSoC 4 with a programmer and debugger – either the onboard PSoC 5LP or external MiniProg3 (CY8CKIT-002). The Hardware chapter on page 27 details the hardware operation. The Example Projects chapter on page 43 details the operation of code examples. The Advanced Section chapter on page 57 deals with advanced topics such as building projects for the PSoC 5LP, USB-UART functionality, and USB-I2C functionality of the PSoC 5LP. The Appendix section provides the schematics, pin assignment, use of zero-ohm resistors and NOLOAD, troubleshooting, and the bill of materials (BOM).

1.4 Additional Learning Resources

Visit the PSoC 4 webpage: www.cypress.com/PSoC4 for additional learning resources in the form of datasheets, technical reference manual, and application notes.

- Beginner Resources: PSoC Creator Training: www.cypress.com/training
- Engineers Looking for More: Click here: www.cypress.com/appnotes to view a growing list of application notes for PSoC 3, PSoC 4, and PSoC 5 LP.
- Learning from Peers: Cypress Developer Community Forums: www.cypress.com/forums

1.5 Technical Support

For assistance, go to our support: www.cypress.com/support web page, or contact our customer support at +1 (800) 541-4736 Ext. 8 (in the USA), or +1 (408) 943-2600 Ext. 8 (International).

1.6 Document Revision History

Table 1-1. Revision History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	04/23/2013	ANCY	Initial version of kit guide
*A	04/25/2013	ANCY	Minor Changes across the guide
*B	05/23/2013	RKAD	Updated Figure 1-1 and minor changes across the guide. Added PSoC 5LP Factory Program Restore Instructions on page 85

1.7 Documentation Conventions

Table 1-2. Document Conventions for Guides

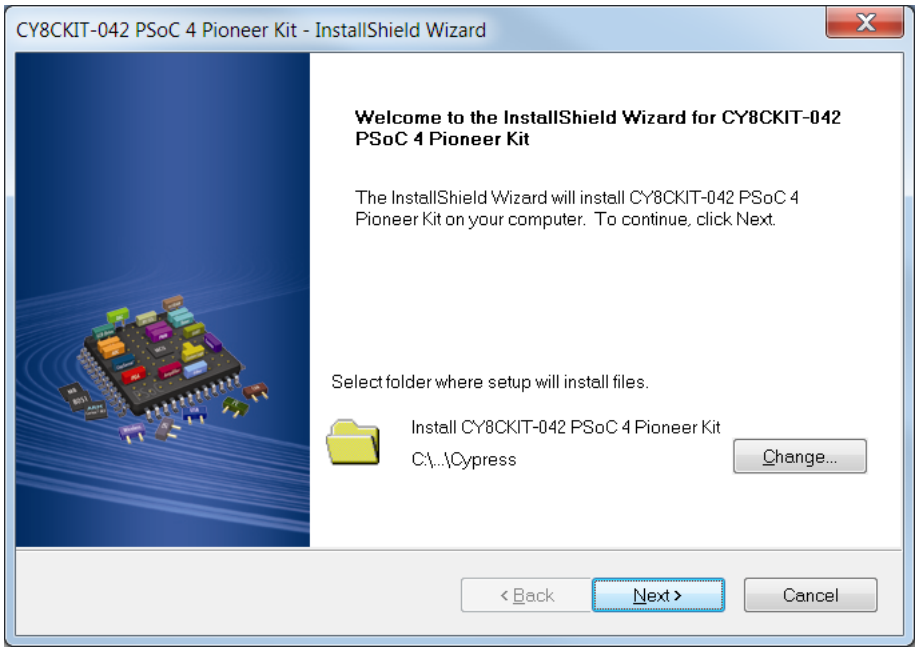
Convention	Usage
`Courier New`	Displays file locations, user entered text, and source code: `C:\ ...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Designer User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**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

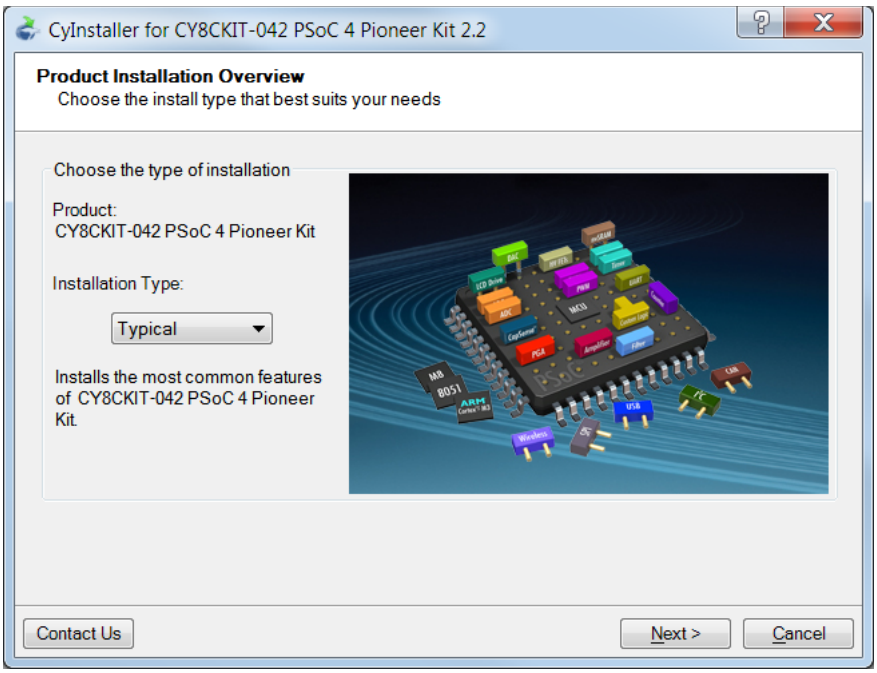
2.1 PSoC 4 Pioneer Kit Software

Follow the below steps to install the PSoC 4 Pioneer kit software.

- Download and install the PSoC 4 Pioneer kit software from the following web page: www.cypress.com/go/CY8CKIT-042.
- Select the folder to install the CY8CKIT-042 (PSoC 4 Pioneer Kit) related files. Choose the directory and click **Next**.



- Select the installation type and click **Next**.



- After the installation is complete, the kit contents are available at the following location:
  `<InstallDirectory>:\CY8CKIT-042 PSoC 4 Pioneer Kit\<version>`

Note: For Windows 7 users the installed files and the folder is **Read-only**. Change the property of the folder by right clicking the folder, **Properties > Attributes** and disable the radio button for **Read-only**. Click **Apply** and **OK** to close the window.

2.2 Install Hardware

There is no additional hardware installation required for this kit.

2.3 Install Software

When installing the PSoC 4 Pioneer Kit, the installer checks if the required software is installed in the system. If the required applications are not installed, then the installer prompts the user to download and install them.

The following software is required:

1. PSoC Creator 2.2 SP1 and later versions support the PSoC 4 family. Download the latest software from www.cypress.com/go/Creator.
2. PSoC Programmer 3.18 or later: Download the latest software from www.cypress.com/go/Programmer.
3. Code examples: After the kit installation is complete the code examples are available in the kit firmware folder. The user can also download the code examples from www.cypress.com/go/CY8CKIT-042

2.4 Uninstall Software

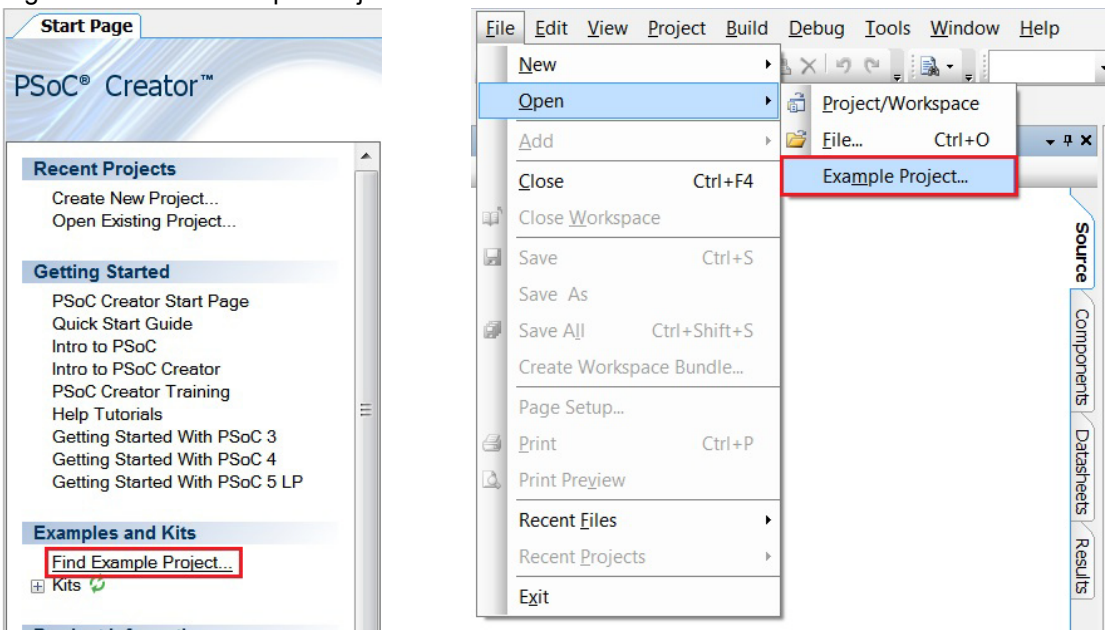
The software can be uninstalled using one of the following methods:

- Go to **Start** > **All Programs** > **Cypress** > **Cypress Update Manager** > **Cypress Update Manager**; select the **Uninstall** button.
- Go to **Start** > **Control Panel** > **Programs and Features**; select the **Uninstall/Change** button.

2.5 Develop Code Fast and Easy with Code Examples

PSoC Creator provides several example projects that make code development fast and easy. To access these example projects, click on **Find Example Project…** under the **Example and Kits** section in the **Start Page** of PSoC Creator or navigate to the Creator tool bar menu and select **File > Open > Example Project**.

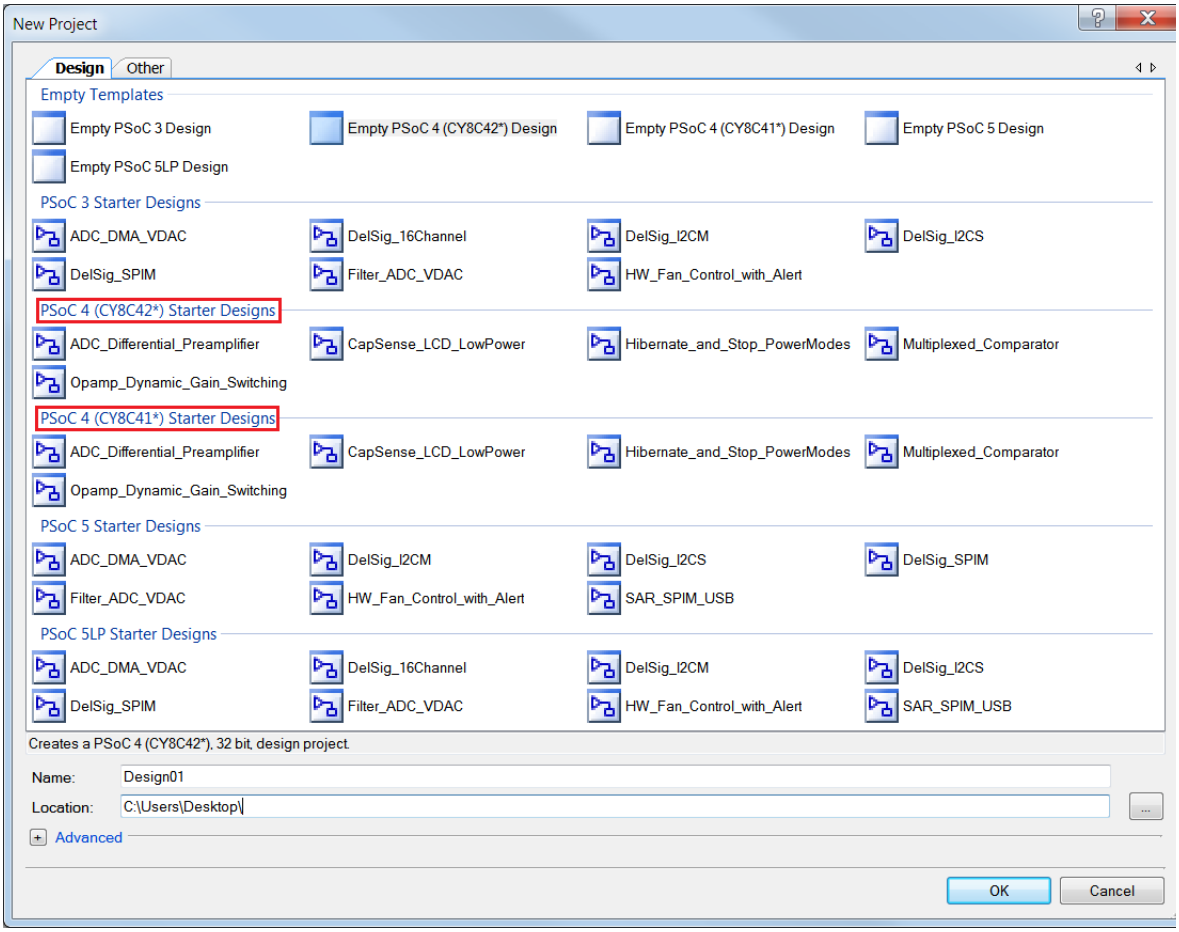
Figure 2-1. Find Example Project



The **Find Example Project** section has various filters that help the user locate the most relevant project.

PSoC Creator also provides several starter designs for each device family. These designs highlight features that are unique to each PSoC device family. They provide users with a starting place instead of creating a new empty design. These starter projects come loaded with various components pre-selected. To use a starter design, navigate to **File > New > Project** and select the design required.

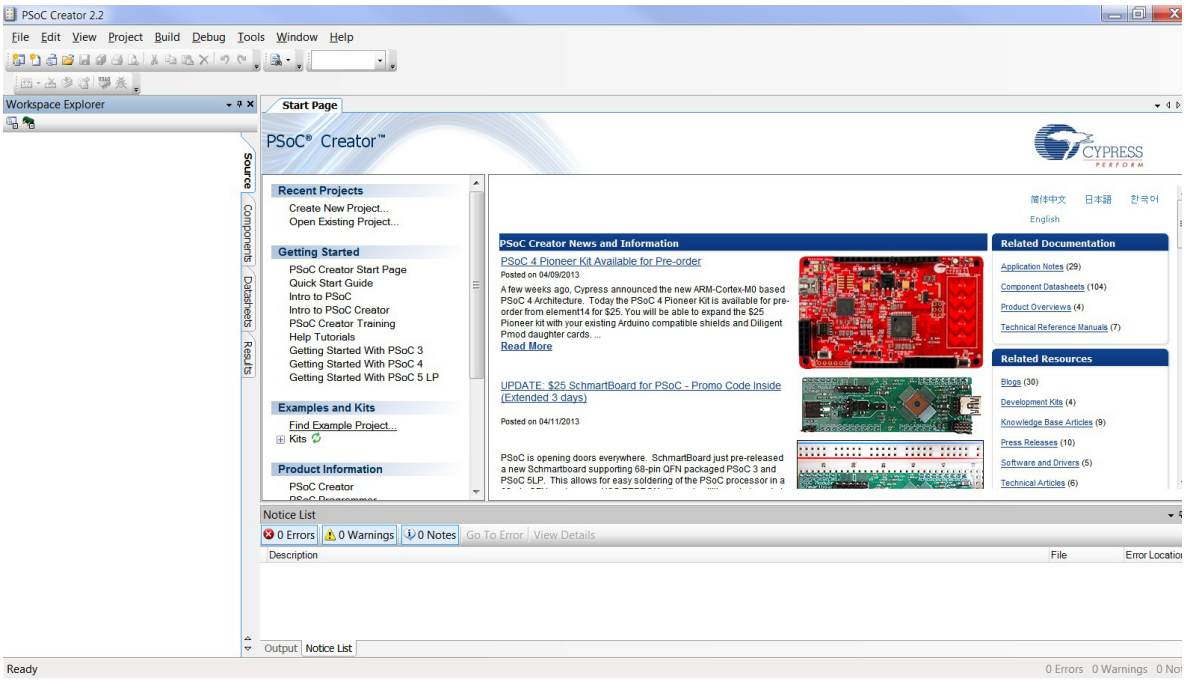
Figure 2-2. Starter Designs



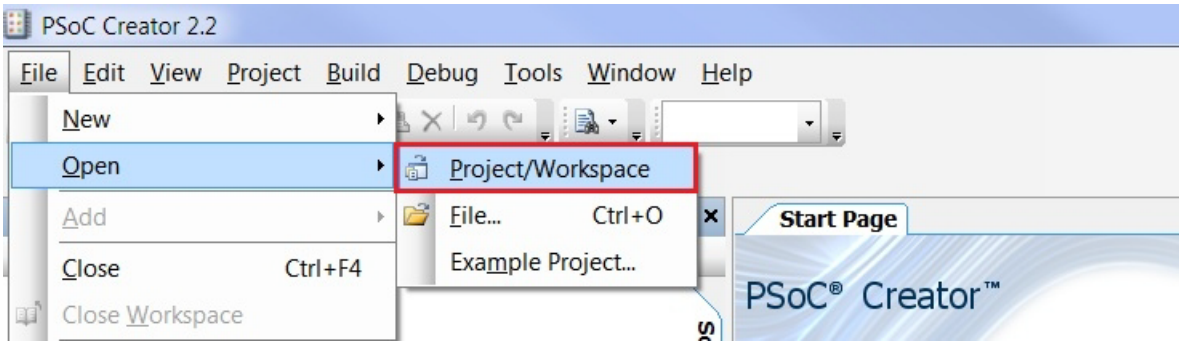
In addition to the example projects and starter designs that are available within PSoC Creator, Cypress continuously strives to provide the best support. Click here to view a growing list of application notes for PSoC 3, PSoC 4, and PSoC 5 LP.

2.6 Open a Code Example Project in PSoC Creator

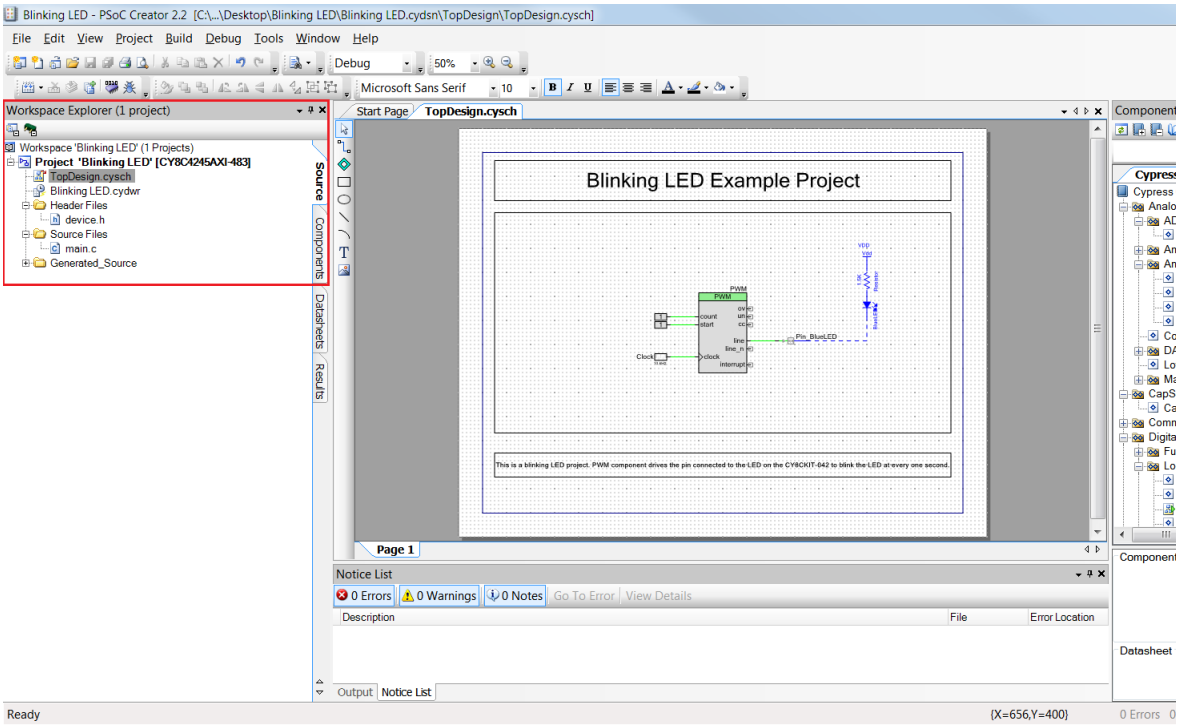
1. Launch the PSoC Creator from the Start Menu.



2. Open the example projects from the **Start Page** by clicking the <Project.cywrk> present below the **Examples and Kits > Kits > CY8CKIT-042**.



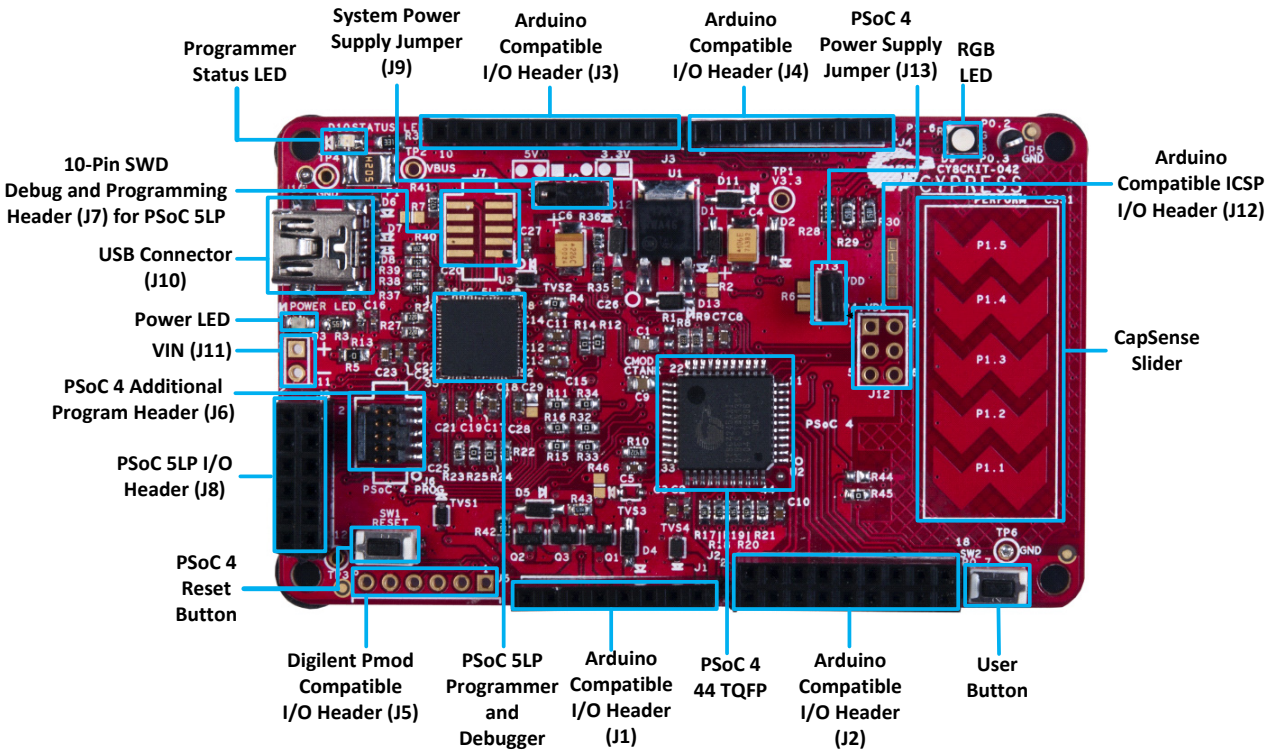
3. The example project opens and display the project files in the Workspace Explorer. Subsequent sections of this user guide shows how to build, program, and understand the example projects supported in this kit.



3. Kit Operation

The PSoC 4 Pioneer kit enables the user to develop applications using the PSoC 4 family of devices. The user can use this kit to develop various applications using Arduino™ shields and Digilent® Pmod™ daughter cards. Figure 3-1 is an image of the PSoC 4 Pioneer board with references to the onboard components.

Figure 3-1. PSoC 4 Pioneer Board



3.1 Pioneer Kit USB Connection

The PSoC 4 Pioneer kit connects to the PC over a USB interface. The kit enumerates as a composite device and three separate devices appear under the device manager window in a Windows operating system.

Table 3-1. PSoC 4 Pioneer Kit in Device Manager after Enumeration

Port	Description
USB Input Device	USB-I2C Bridge
KitProg	Programmer and Debugger
KitProg USB-UART	USB-UART Bridge will appear as COM# port

Figure 3-2. KitProg Driver Installation

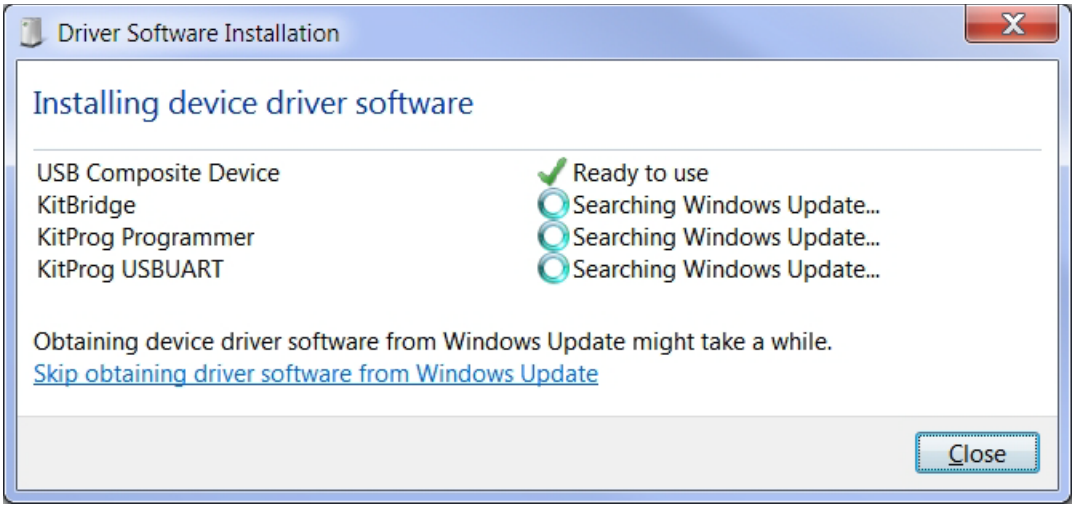
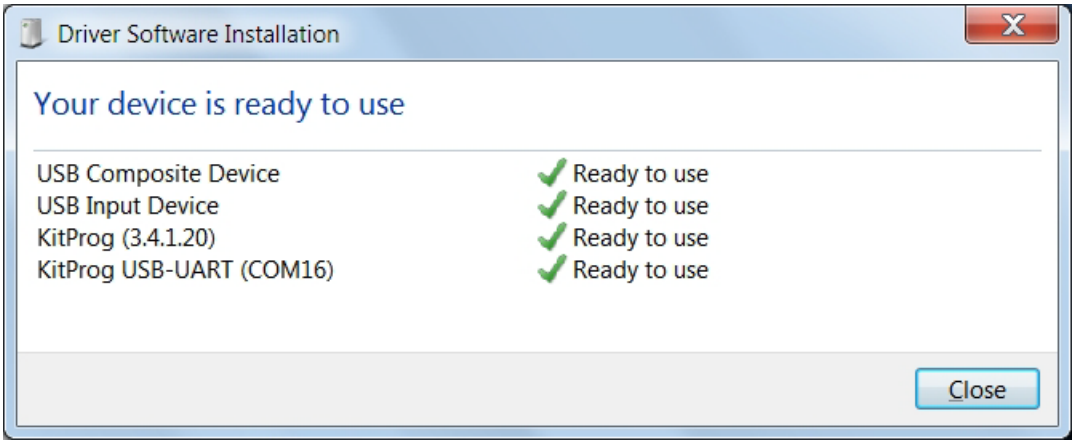


Figure 3-3. KitProg Driver Installation



3.2 Programming and Debugging the PSoC 4 Device

The kit allows programming and debugging of the PSoC 4 in two modes:

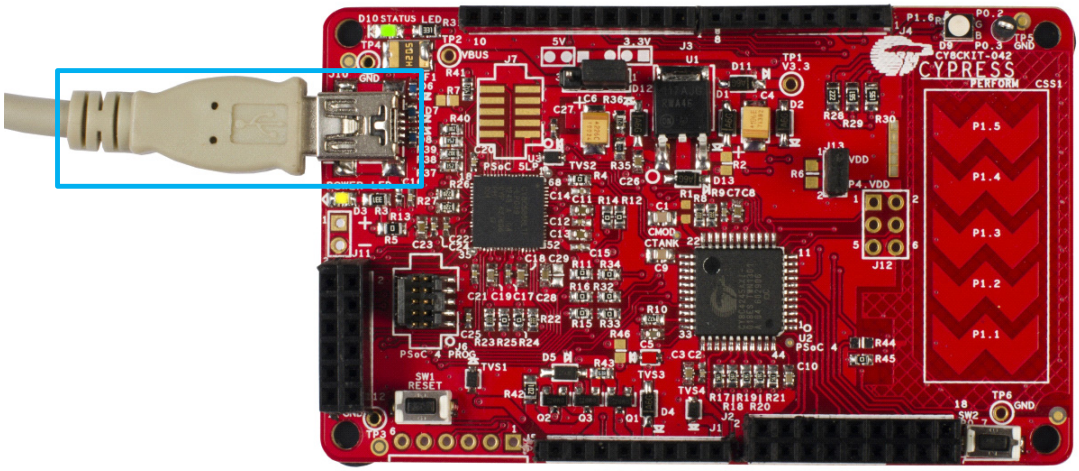
1. Using the onboard PSoC 5LP programmer and debugger.
2. Using a CY8CKIT-002 MiniProg3 programmer and debugger.

3.2.1 Using the Onboard PSoC 5LP Programmer and Debugger

The default programming interface for the kit is a USB-based, onboard programming interface. Before trying to program the device, PSoC Creator and PSoC Programmer must be installed. Refer to the Install Software chapter on page 12 to install software for the PSoC 4 Pioneer Kit.

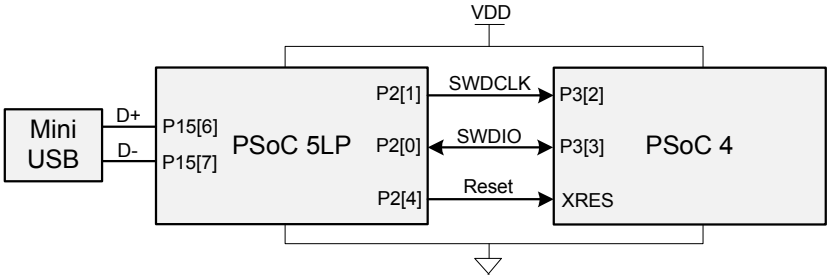
1. To program the device, plug the USB cable into the programming USB connector J10, as shown in Figure 3-4. The kit will enumerate as a composite device. Refer to the Pioneer Kit USB Connection on page 18 for details.

Figure 3-4. Connect USB Cable to J10



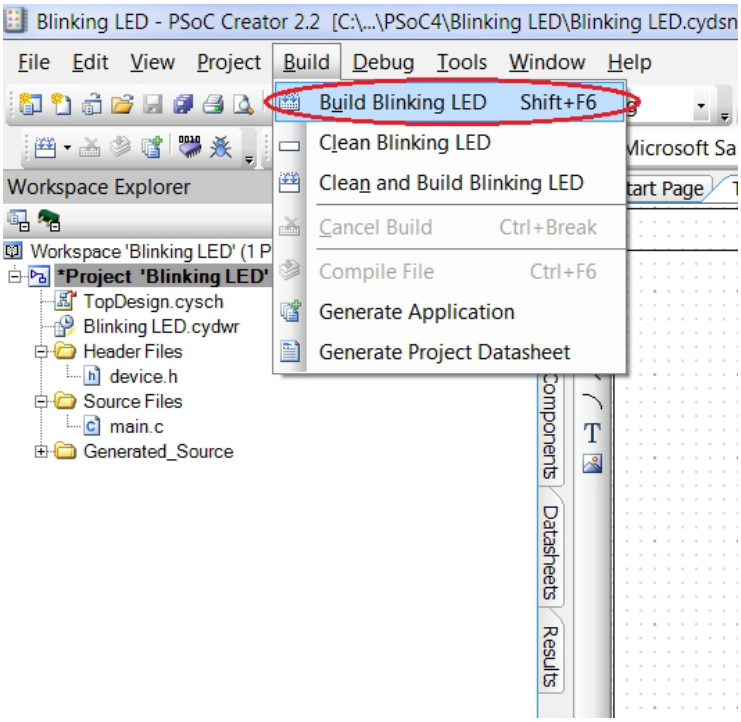
2. The onboard PSoC 5 LP uses SWD to program the PSoC 4 device. Refer to Figure 3-5 for this implementation.

Figure 3-5. SWD Programming PSoC 4 using the PSoC 5LP



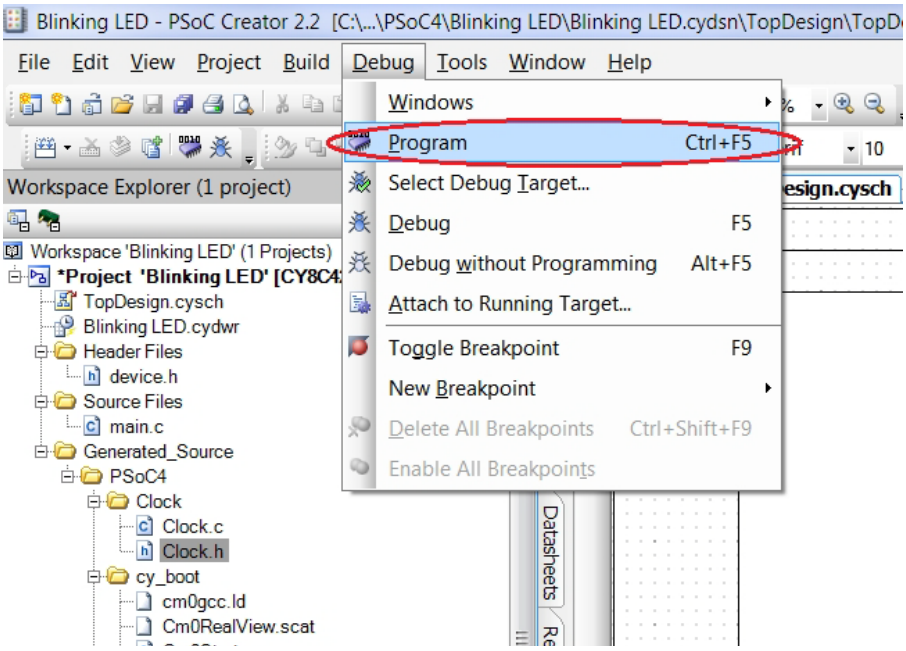
3. The Pioneer kit's onboard programmer will enumerate on the PC and in the software tools as **KitProg**. Load an example project in PSoC Creator (such as the project described in Section 2.3 Install Software), and initiate the build in PSoC Creator by clicking **Build > Build Project** or [Shift]+[F6].

Figure 3-6. Build Project in PSoC Creator



4. After the project is built without errors and warnings select **Debug > Program** or [Ctrl]+[F5] to program the device.

Figure 3-7. Program the Device from PSoC Creator



The onboard programmer supports only the **RESET** programming mode. When using the onboard programmer, the board can either be powered by the USB (VBUS) or by an external source like an Arduino shield. If the board is already powered from another source, plugging in the USB programmer does not damage the board.

3.2.2 Using a CY8CKIT-002 MiniProg3 Programmer and Debugger

The PSoC 4 on the Pioneer kit can also be programmed using a MiniProg3 (CY8CKIT-002). To use a MiniProg3 for programming, use the connector J6 on the board, as shown in Figure 3-8. With the MiniProg3, programming is similar to the onboard programmer; however, the setup enumerates as a **MiniProg3**. Only the **RESET** mode of programming is available.

The board can also be powered from the MiniProg3. To power the board from the MiniProg3, open **Tool > Options**. In the **Options** expand **Program and Debug**, expand **Port Configuration**, select MiniProg3 and make the settings shown in Figure 3-9.

Note: The MiniProg3 (CY8CKIT-002) is not part of the PSoC 4 Pioneer kit contents. It can be purchased from the Cypress Online Store.

Figure 3-8. PSoC 4 Programming/Debug using MiniProg3

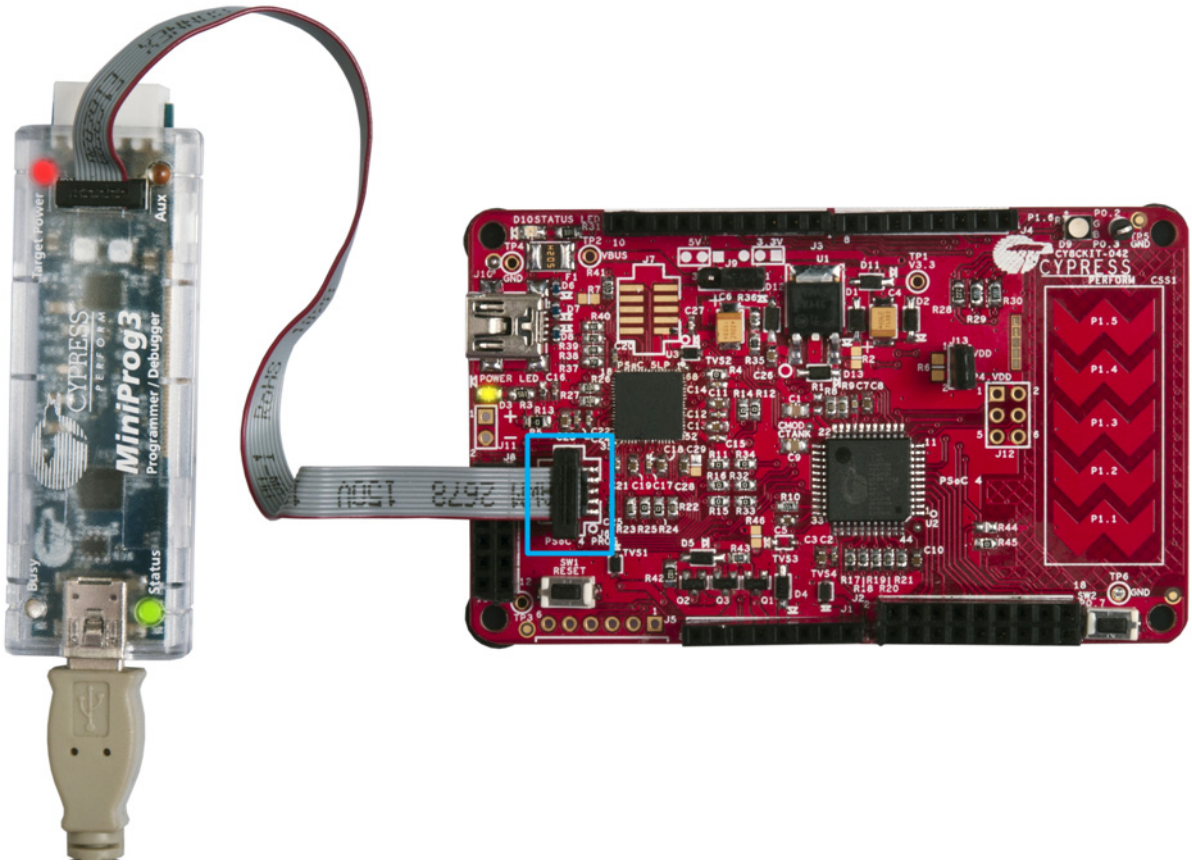
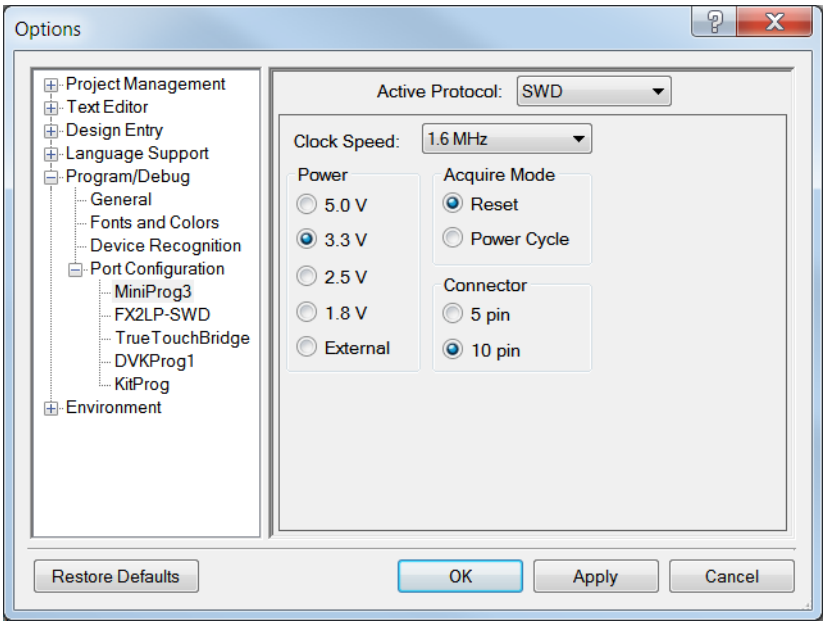


Figure 3-9. MiniProg3 Configuration



3.3 USB-UART Bridge

The onboard PSoC 5LP can also act as a USB-UART Bridge to transfer and receive data from the PSoC 4 device to the PC via COM terminal software. When the USB Mini B is connected to J10 of the PSoC 4 Pioneer Kit, a device named **KitProg USBUART** is available under the Ports (COM and LPT) in the device manager. This user guide provides in-depth information about the USB-UART functionality in Using PSoC 5LP as USB-UART Bridge on page 57.

To use the USB-UART functionality in the COM terminal software select the corresponding COM port as the communication port to transfer data to and from the COM terminal software.

The UART lines from the PSoC 5LP are brought to the pins P12[6] (J8_9) and P12[7] (J8_10) of the header J8. This interface can be used to send or receive data from any PSoC 4 design that has a UART by connecting the pins on header J8 to the RX and TX pins assigned in PSoC 4. The UART can be used as an additional interface to debug designs. This bridge can also be used to interface with other external UART based devices. Figure 3-10 shows the connection between the RX and TX lines of the PSoC 5LP and that of PSoC 4. In this example, the PSoC 4 UART has been routed to the J3 header and the user has to connect wires between the PSoC 5LP's RX and TX lines available on J8.

Figure 3-10. Example Connection between the RX and TX Lines of the PSoC 5LP and the PSoC 4

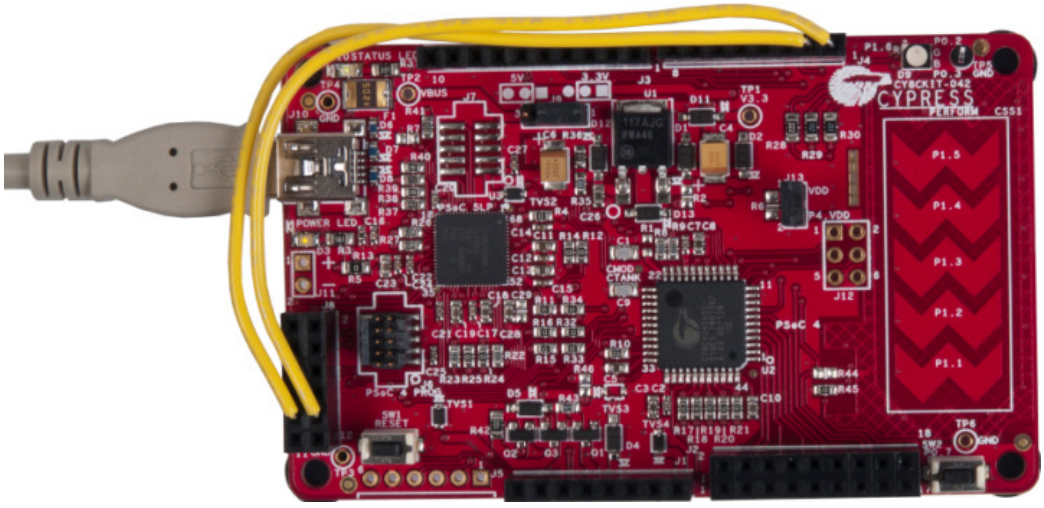


Table 3-2 lists the specifications supported by USB-UART bridge.

Table 3-2. Specifications Supported by USB-UART Bridge

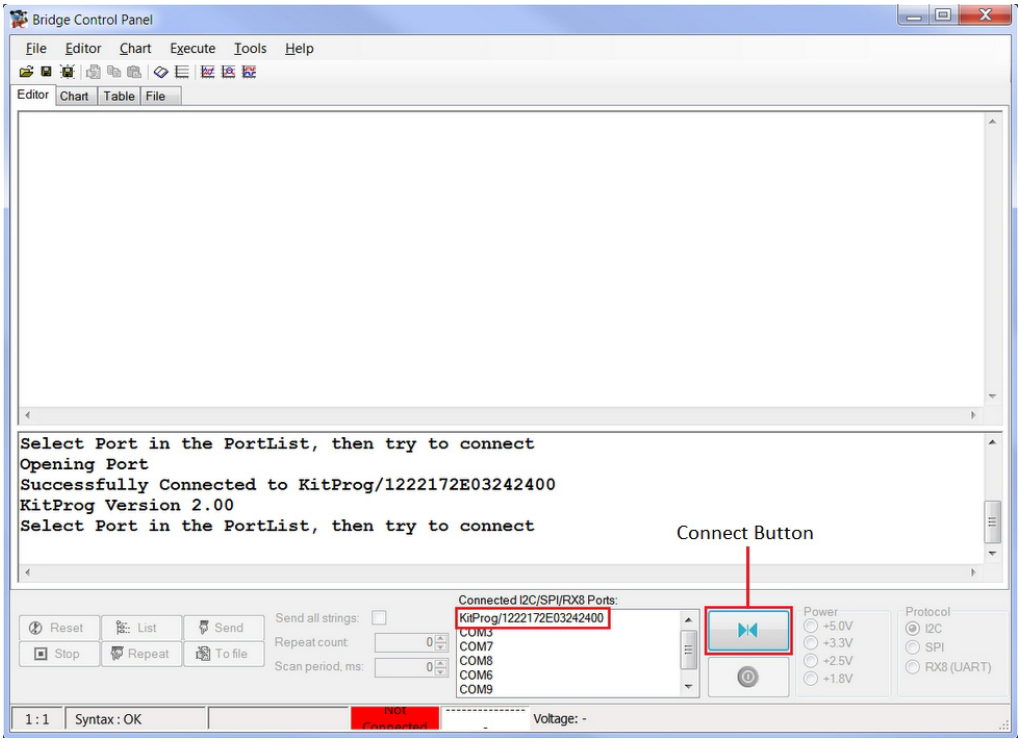
Parameter	Supported values
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None
File transfer protocols supported	Xmodem, 1K Xmodem, Ymodem, Kermit and Zmodem (only speeds greater than 2400 baud).

3.4 USB-I2C Bridge

The onboard PSoC 5 LP also functions as a USB-I2C Bridge. The PSoC 4 communicates with the PSoC 5LP using an I2C interface and the PSoC 5LP transfers the data over the USB to the USB-I2C software utility, called the Bridge Control Panel (BCP), on the PC.

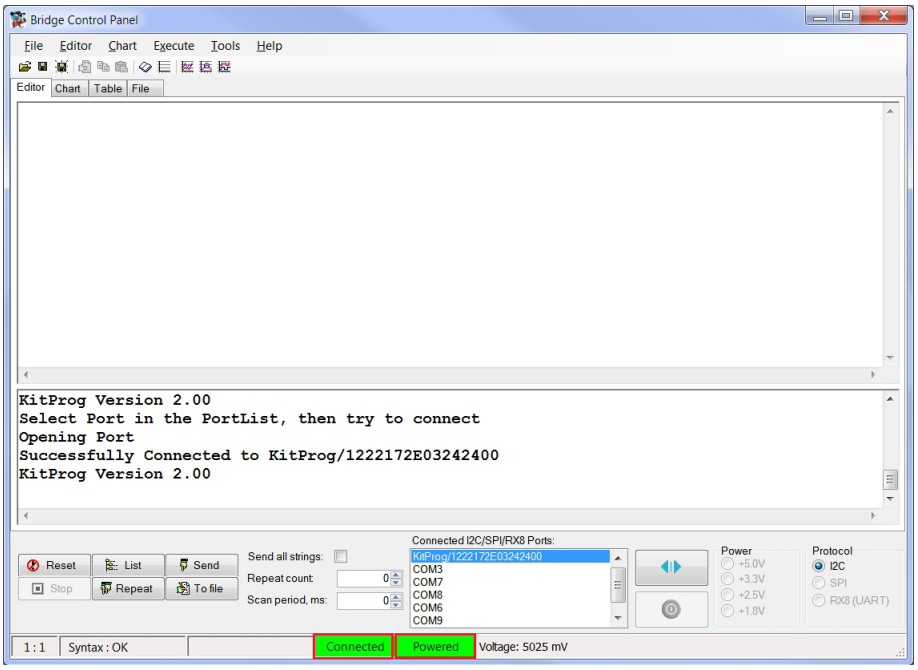
The BCP is available as part of the PSoC Programmer installation. This software can be used to send and receive USB-I2C data from the PSoC 5LP. When the USB Mini B is connected to the header J10 on the Pioneer Kit, the **KitProg USB-I2C** is available under the **Connected I2C/SPI/RX8 Ports** in the BCP.

Figure 3-11. Bridge Control Panel



To use the USB_I2C functionality, select the **KitProg USB-I2C** in the BCP. On successful connection, the **Connected** and **Powered** tabs turn green.

Figure 3-12. KitProg USB-I2C Connected in Bridge Control Panel



USB-I2C is implemented using the USB and I2C components of the PSoC 5LP. The SCL (P12_0) and SDA (P12_1) lines from the PSoC 5LP are connected to SCL (P3_0) and SDA (P3_1) lines of I2C of the PSoC 4.

The USB-I2C Bridge currently supports I2C speed of 50 kHz, 100 kHz, 400 kHz, and 1 MHz.

Refer to the section on page 56 for building a project, which uses USB-I2C Bridge functionality.

3.5 Updating the Firmware of Onboard Programmer PSoC 5LP

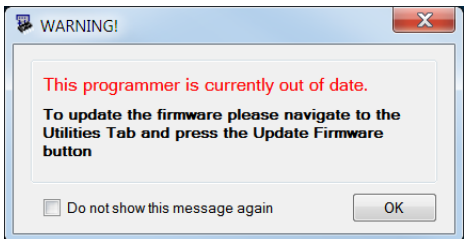
The firmware of the onboard programmer and debugger - PSoC 5LP can be updated from PSoC Programmer.

When a new firmware is available or when the KitProg firmware is corrupt (refer to the section Error in Firmware / Status Indication in Status LED on page 100), PSoC Programmer displays a Warning window indicating that the new firmware is available.

The user can open the PSoC Programmer from **Start > All Programs > Cypress > PSoC Programmer<version>**.

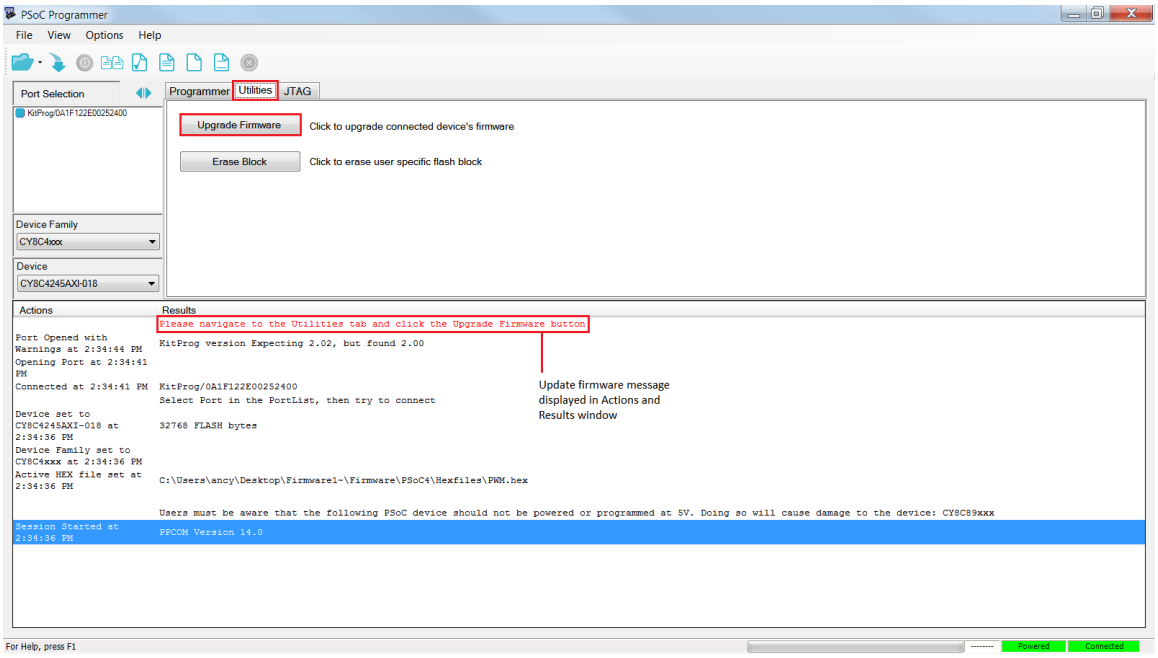
When the PSoC Programmer is opened, a **WARNING** window pops up saying that the programmer is currently out of date.

Figure 3-13. Warning Window to Update the Firmware



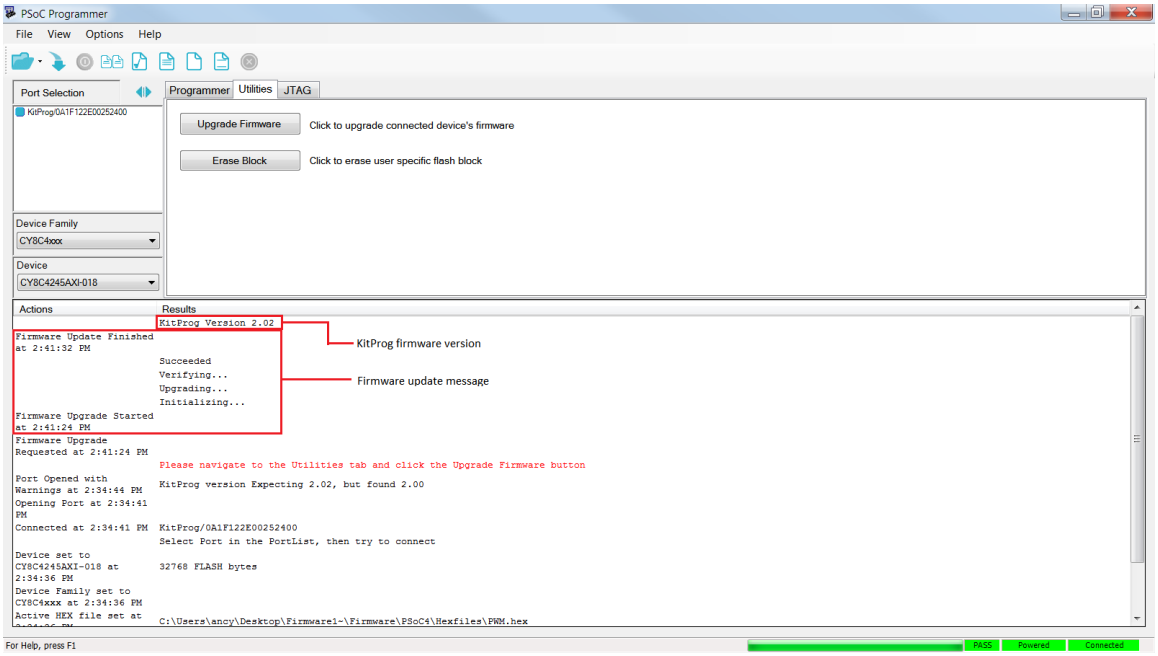
Click **OK** to close the window. On closing the warning window the **Action and Results** window displays “Please navigate to the Utilities tab and click the Upgrade Firmware button”

Figure 3-14. Update Firmware Message Displayed in PSoC Programmer



Click the **Utilities** tab and click the **Upgarde** Firmware button to upgrade the firmware. On successful upgrade, the **Action and Results** window displays the success message with the KitProg version.

Figure 3-15. Firmware Updated in PSoC Programmer



4. Hardware

4.1 Board Details

The PSoC 4 Pioneer kit consists of following sections:

- PSoC 4
- PSoC 5LP
- Power supply system
- Programming interfaces (J6, J7 - unpopulated, J10)
- Arduino compatible headers (J1, J2, J3, J4, and J12 - unpopulated)
- Digilent Pmod compatible header (J5 - unpopulated)
- PSoC 5LP GPIO header (J8)
- CapSense® slider
- Pioneer Board LEDs
- Push buttons (Reset Button and User Button)

Figure 4-1. PSoC 4 Pioneer Kit Details

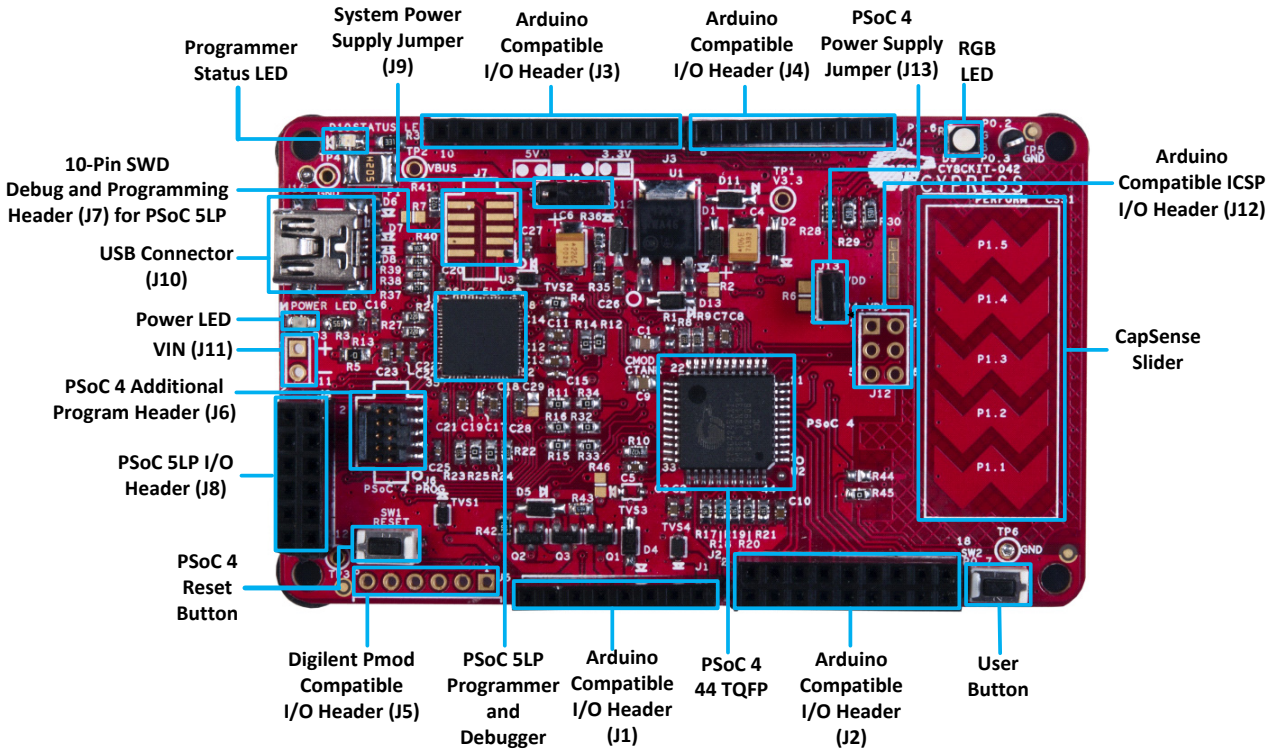
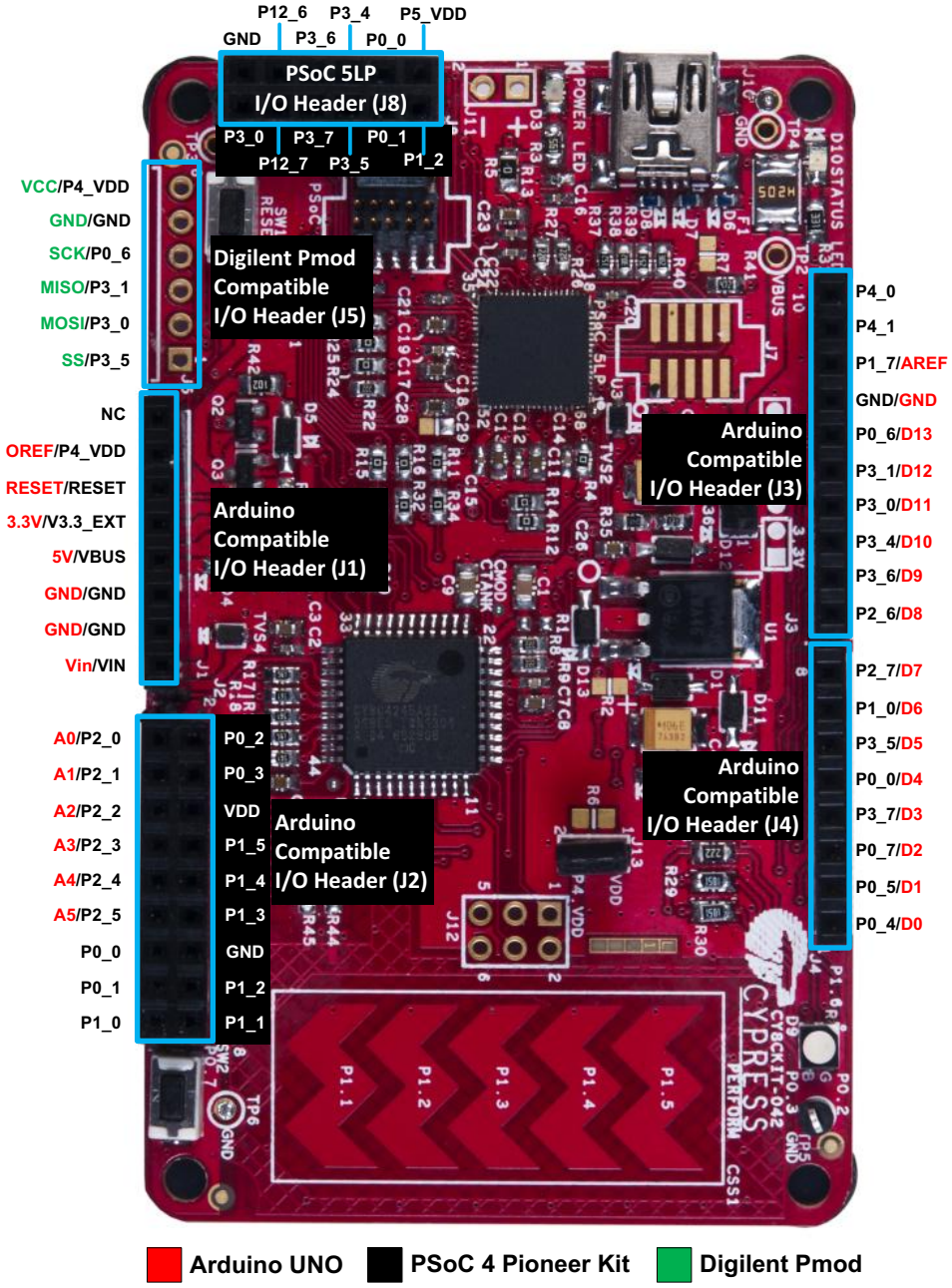


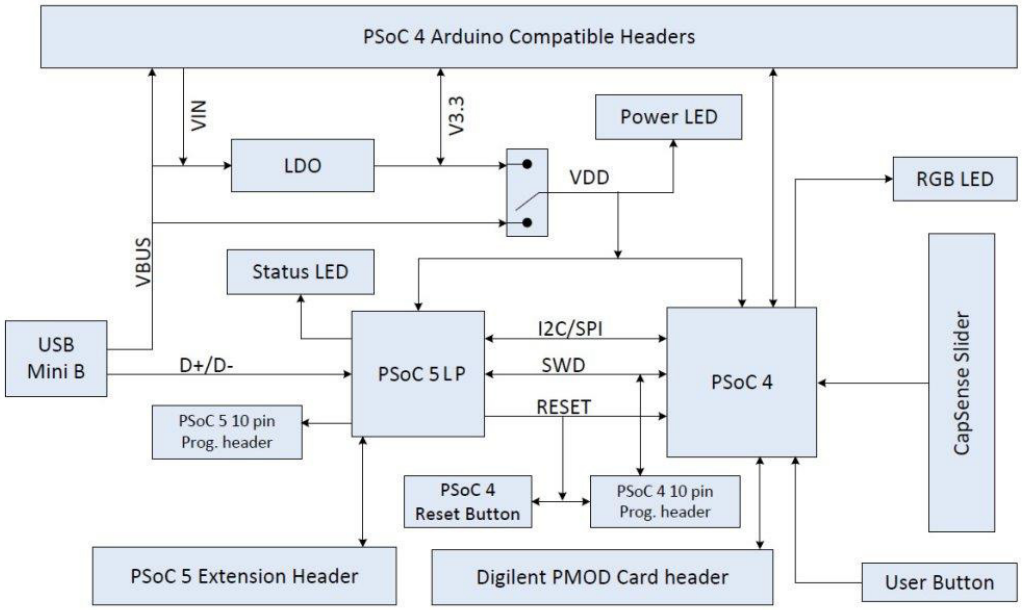
Figure 4-2. PSoC 4 Pioneer Kit Pin Mapping



4.2 Theory of Operation

This section provides the block-level description of the PSoC 4 Pioneer Kit.

Figure 4-3. Block Diagram



The PSoC 4 is a new generation of Programmable System-on-Chip from Cypress for embedded applications. It combines programmable analog, programmable digital logic, programmable I/O, and a high-performance ARM Cortex-M0 subsystem. With the PSoC 4, the user can create the exact combination of peripherals to meet the application requirements.

The PSoC 4 Pioneer Kit features an onboard PSoC 5LP which communicates through the USB to program and debug the PSoC 4 using Serial Wire Debug (SWD). The PSoC 5LP also functions as a USB-I2C bridge and a USB-UART bridge.

The PSoC 4 Pioneer Kit has an RGB LED, a status LED, and a power LED. The RGB LED is connected to the PSoC 4 and the status LED is connected to the PSoC 5LP.

For more information on the status LED, refer to section A.5 Error in Firmware / Status Indication in Status LED on page 100. The PSoC 4 Pioneer Kit also includes a PSoC 4 reset button, which connects to the XRES of the PSoC 4, a user button, and a 5-segment CapSense® slider that can be used to develop touch-based applications. The PSoC 4 pins are brought out onto the headers (J1-J4) on the kit to support Arduino shields. The PSoC 5LP pins are brought out to a header (J8) to allow the user to utilize the onboard PSoC 5LP to develop custom applications.

The PSoC 4 Pioneer Kit can be powered from the USB Mini B, the Arduino compatible header or an external power supply. The input voltage is regulated by an LDO regulator to 3.3 V. The user has an option to select between VBUS (5 V) and 3.3 V by suitably plugging the jumper onto the voltage selection header VDD.

4.3 Functional Description

4.3.1 PSoC 4

This kit uses the PSoC 4200 family device. The PSoC 4200 devices are a combination of a microcontroller with programmable logic, high-performance analog-to-digital conversion, two Op Amps with Comparator mode and commonly used fixed-function peripherals. For more information, refer to the PSoC 4 web page and the PSoC 4200 family datasheet.

Features

- 32-bit MCU Sub-system
  - 48 MHz ARM Cortex-M0 CPU with single cycle multiply
  - Up to 32 kB of flash with Read Accelerator
  - Up to 4 kB of SRAM
- Programmable Analog
  - Two op amps with reconfigurable high-drive external and high-bandwidth internal drive and Comparator modes and ADC input buffering capability
  - 12-bit 1 Msps SAR ADC with differential and single-ended modes and Channel Sequencer with signal averaging
  - Two current DACs (IDACs) for general-purpose or capacitive sensing applications on any pin
  - Two low-power comparators that operate in Deep Sleep
- Programmable Digital
  - Four programmable logic blocks, each with 8 Macrocells and data path (called universal digital blocks, UDBs)
  - Cypress provided peripheral component library, user-defined state machines, and Verilog input
- Low Power 1.71 to 5.5 V operation
  - 20 nA Stop Mode with GPIO pin wakeup
  - Hibernate and Deep Sleep modes allow wakeup-time versus power trade-offs
- Capacitive Sensing
  - Cypress Capacitive Sigma-Delta (CSD) provides best-in-class SNR (>5:1) and water tolerance
  - Cypress supplied software component makes capacitive sensing design easy
  - Automatic hardware tuning (SmartSense™)
- Segment LCD Drive
  - LCD drive supported on all pins (common or segment)
  - Operates in Deep Sleep mode with 4 bits per pin memory
- Serial Communication
  - Two independent run-time reconfigurable Serial Communication Blocks (SCBs) with re-configurable I2C, SPI, or UART functionality
- Timing and Pulse-Width Modulation
  - Four 16-bit Timer/Counter Pulse-Width Modulator (TCPWM) blocks
  - Center-aligned, Edge, and Pseudo-random modes
  - Comparator-based triggering of Kill signals for motor drive and other high reliability digital logic applications
- Up to 36 Programmable GPIO
  - 44-pin TQFP, 40-pin QFN, and 28-pin SSOP packages.
  - Any GPIO Pin can be Capsense, LCD, Analog, or Digital
  - Drive modes, strengths, and slew rates are programmable
- PSoC Creator Design Environment
  - Integrated Development Environment provides schematic design entry and build (with analog and digital automatic routing)

- Applications Programming Interface (API Component) for all fixed-function and programmable peripherals
- Industry Standard Tool Compatibility
  - After schematic entry, development can be done with ARM-based industry-standard development tools

For more information see CY8C42 family datasheet.

4.3.2 PSoC 5LP

An onboard PSoC 5LP is used to program and debug the PSoC 4. The PSoC 5LP connects to the USB port of the PC through a USB Mini B connector and to the SWD interface of the PSoC 4 device.

PSoC 5LP is a true system level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C58LPxx family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals. For more information, refer to the PSoC 5LP web page.

Features

- 32-bit ARM Cortex-M3 CPU core
  - DC to 67 MHz operation
  - Flash program memory, up to 256 KB, 100,000 write cycles, 20-year retention, and multiple security features
  - Up to 32-KB flash error correcting code (ECC) or configuration storage
  - Up to 64 KB SRAM
  - 2-KB electrically erasable programmable read-only memory (EEPROM) memory, 1 M cycles, and 20 years retention
  - 24-channel direct memory access (DMA) with multilayer AHB bus access
    a.Programmable chained descriptors and priorities
    b.High bandwidth 32-bit transfer support
- Low voltage, ultra low power
  - Wide operating voltage range: 0.5 V to 5.5 V
  - High-efficiency boost regulator from 0.5 V input to 1.8 V to 5.0 V output
  - 3.1 mA at 6 MHz
  - Low power modes including:
    a.2-µA sleep mode with real time clock (RTC) and low-voltage detect (LVD) interrupt
    b.300-nA hibernate mode with RAM retention
- Versatile I/O system
  - 28 to 72 I/Os (62 GPIOs, 8 SIOs, 2 USBIOs[2])
  - Any GPIO to any digital or analog peripheral routability
  - LCD direct drive from any GPIO, up to 46×16 segments
  - CapSense support from any GPIO[3]
  - 1.2 V to 5.5 V I/O interface voltages, up to 4 domains
  - Maskable, independent IRQ on any pin or port
  - Schmitt-trigger transistor-transistor logic (TTL) inputs
  - All GPIOs configurable as open drain high/low, pull-up/pull-down, High-Z, or strong output
  - Configurable GPIO pin state at power-on reset (POR)
  - 25 mA sink on SIO
- Digital peripherals
  - 20 to 24 programmable logic device (PLD) based universal digital blocks (UDBs)
  - Full CAN 2.0b 16 RX, 8 TX buffers[2]
  - Full-Speed (FS) USB 2.0 12 Mbps using internal oscillator[2]

- Four 16-bit configurable timers, counters, and PWM blocks
- 67-MHz, 24-bit fixed point digital filter block (DFB) to implement finite impulse response (FIR) and infinite impulse response (IIR) filters
- Library of standard peripherals
  a.8-, 16-, 24-, and 32-bit timers, counters, and PWMs
  b.Serial peripheral interface (SPI), universal asynchronous transmitter receiver (UART), and I2C
  c.Many others available in catalog
- Library of advanced peripherals
  a.Cyclic redundancy check (CRC)
  b.Pseudo random sequence (PRS) generator
  c.Local interconnect network (LIN) bus 2.0
  d.Quadrature decoder
- Analog peripherals (1.71 V ≤ VDDA ≤ 5.5 V)
- 1.024 V ±0.1% internal voltage reference across –40 °C to +85 °C
- Configurable delta-sigma ADC with 8- to 20-bit resolution
- Sample rates up to 192 ksps
- Programmable gain stage: ×0.25 to ×16
- 12-bit mode, 192 ksps, 66-dB signal to noise and distortion ratio (SINAD), ±1-bit INL/DNL
- 16-bit mode, 48 ksps, 84-dB SINAD, ±2-bit INL, ±1-bit DNL
- Up to two SAR ADCs, each 12-bit at 1 Msps
- Four 8-bit 8 Msps current IDACs or 1-Msps voltage VDACs
- Four comparators with 95-ns response time
- Four uncommitted opamps with 25-mA drive capability
- Four configurable multifunction analog blocks. Example configurations are programmable gain amplifier (PGA), transimpedance amplifier (TIA), mixer, and sample and hold
- CapSense support

- Programming, debug, and trace
  - JTAG (4 wire), serial wire debug (SWD) (2 wire), single wire viewer (SWV), and TRACEPORT interfaces
  - Cortex-M3 flash patch and breakpoint (FPB) block
  - Cortex-M3 Embedded Trace Macrocell™ (ETM™) generates an instruction trace stream.
  - Cortex-M3 data watchpoint and trace (DWT) generates data trace information
  - Cortex-M3 Instrumentation Trace Macrocell (ITM) can be used for printf-style debugging
  - DWT, ETM, and ITM blocks communicate with off-chip debug and trace systems via the SWV or TRACEPORT
  - Bootloader programming supportable through I2C, SPI, UART, USB, and other interfaces
- Precision, programmable clocking
  - 3- to 62-MHz internal oscillator over full temperature and voltage range
  - 4- to 25-MHz crystal oscillator for crystal PPM accuracy
  - Internal PLL clock generation up to 67 MHz
  - 32.768-kHz watch crystal oscillator
  - Low power internal oscillator at 1, 33, and 100 kHz

For more, see CY8C58LPxx family datasheet.

4.3.3 Power Supply System

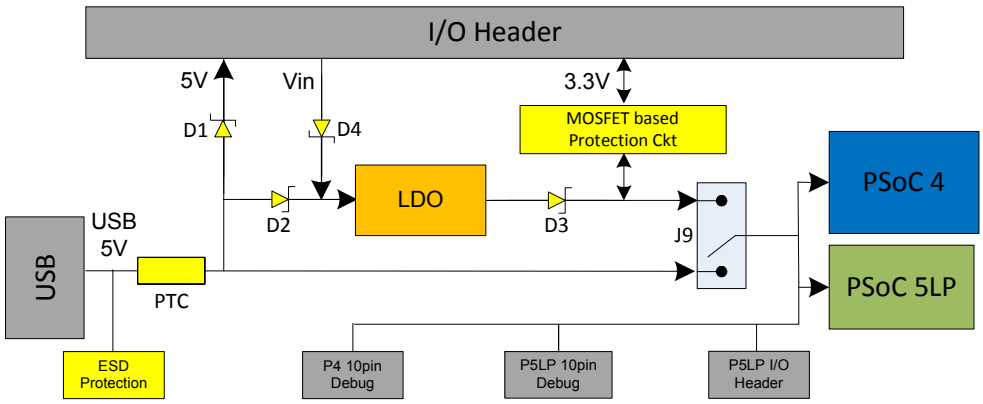
The power supply system on this board is versatile, allowing the input supply to come from the following sources:

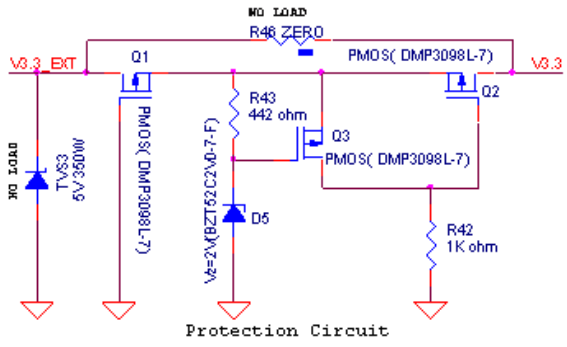
- 5 V power from onboard USB programming header J10
- 5-12 V power from Arduino shield using J1_01 header
- VTARG - power from the on-board SWD programming using J6 or J7
- VIN - J11

The PSoC 4 and the PSoC 5LP are powered by either a 3.3 V or a 5 V. The selection between the 3.3 V and the 5 V is made through the J9 jumper. The board can supply 3.3 V and 5 V to the I/O headers, and it can receive a 3.3 V from the I/O headers. The board can also be powered with an external power supply through the VIN (J11) header and the allowed voltage range for the VIN is 5 V-12 V. There is a Low Drop Out regulator that takes the VIN and regulates it down to 3.3 V. Figure 4-4 shows the power supply block diagram and the protection circuitry.

Note: The 5 V domain is directly powered by the USB (VBUS). For this reason this domain is unregulated.

Figure 4-4. Power Supply Block Diagram with Protection Circuits





4.3.3.1 Protection Circuit

The power supply rail has reverse-voltage, over-voltage, short circuits, and excess current protection features as seen in Figure 4-4.

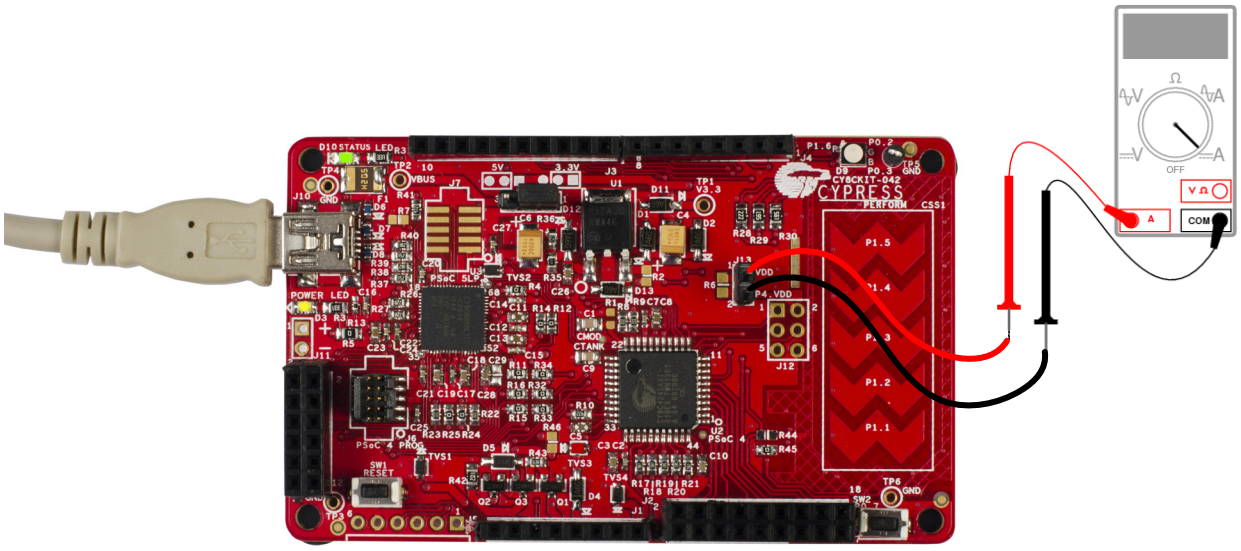
- The Schottky diode (D1) ensures power cannot be supplied to the 5 V domain of the board from the I/O header.
- The series protection diode (D2) ensures VIN (power supply from the I/O header) does not back power the USB.
- The Schottky diode (D3) ensures 3.3 V from I/O header doesn't back power the LDO.
- The series protection diode (D4) ensures that the reverse-voltage cannot be supplied from the VIN to the input of the regulator.
- A PTC resettable fuse is connected to protect the computer's USB ports from shorts and over current.
- The MOSFET-based protection circuit provides over-voltage and reverse-voltage protection to the 3.3 V rail. The PMOS Q1 protects the components on the board from a reverse-voltage condition. The PMOS Q2 protects the PSoC from an over-voltage condition. The PMOS Q2 will turn off when a voltage greater than 4.2 V is applied, protecting the PSoC 4.
- The output voltage of the LDO is adjusted such that it takes into account the voltage drop across the Schottky diode and provides 3.3 V.

4.3.3.2 Procedure to Measure the PSoC 4 Current Consumption

The following three methods are supported for measuring current consumption of the PSoC 4 device.

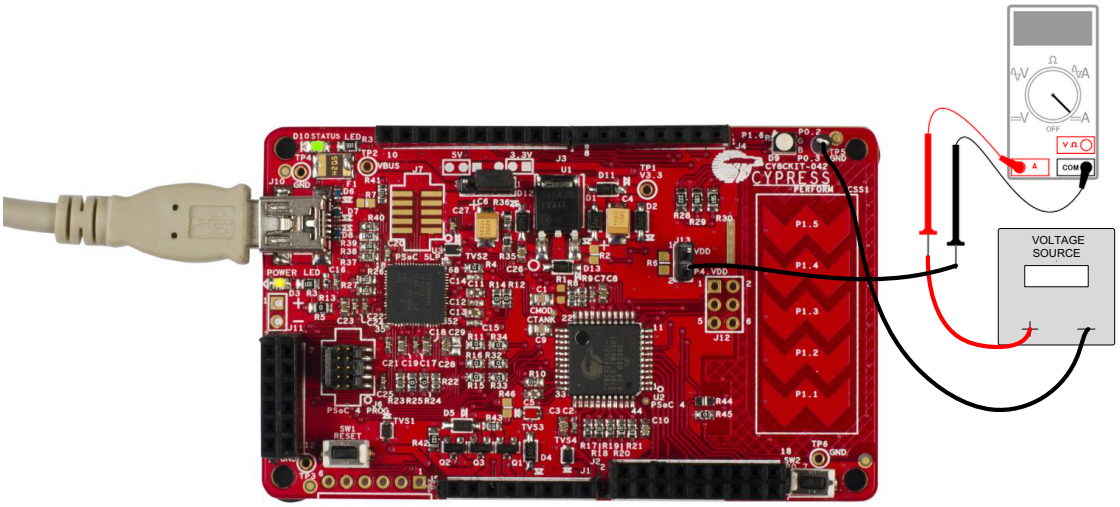
1. When the board is powered through the USB port (J10), remove the jumper J13 and connect ammeter as shown in Figure 4-5.

Figure 4-5. PSoC 4 Current Measurement when Powered from the USB port



2. When using a separate power supply for the PSoC 4 with USB powering the board (regulator output on the USB supply must be within 0.5 V of the separate power supply).
   Remove the jumper J13. Connect the positive terminal of voltage supply to positive terminal of ammeter and the negative terminal of ammeter to lower pin of J13. Figure 4-6 shows the necessary connections to be made.

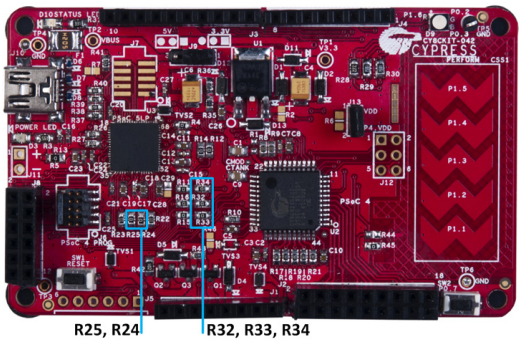
Figure 4-6. PSoC 4 Current Measurement when Powered Separately



3. When the PSoC 4 is powered separately and the PSoC 5LP is not powered, the user has to make these changes to avoid leakage currents while performing current measurement:

- Remove the zero-ohm resistors R24 and R25. Removing these resistors will affect the USB-I2C functionality.
- Remove R32, R33, and R34, which are meant for programming the PSoC 4. Removing these resistors disables the PSoC 5LP capability for programming.
- Connect an ammeter between the pins 1 and 2 of header J13 to measure current.

Figure 4-7. Zero-ohm Resistor Position on the Board



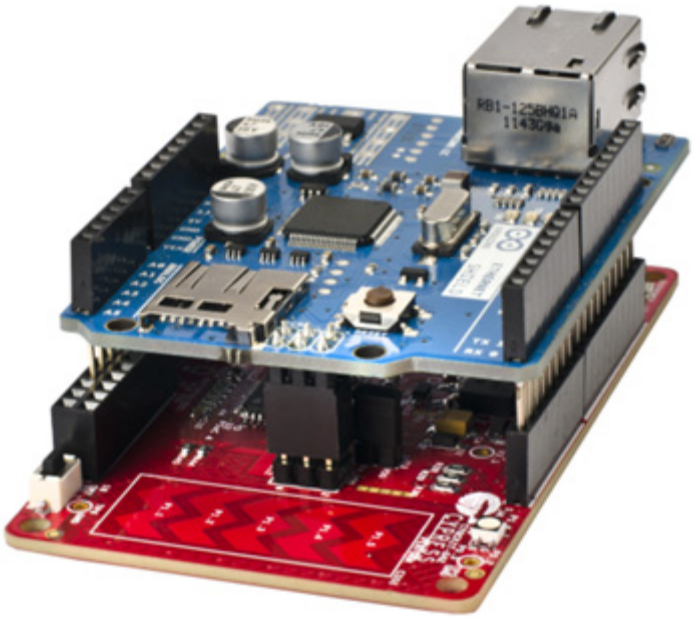
4.3.4 Programming Interface

The kit allows programming and debugging of the PSoC 4 in two modes:

- Using the Onboard PSoC 5LP Programmer and Debugger
- Using a CY8CKIT-002 MiniProg3 Programmer and Debugger

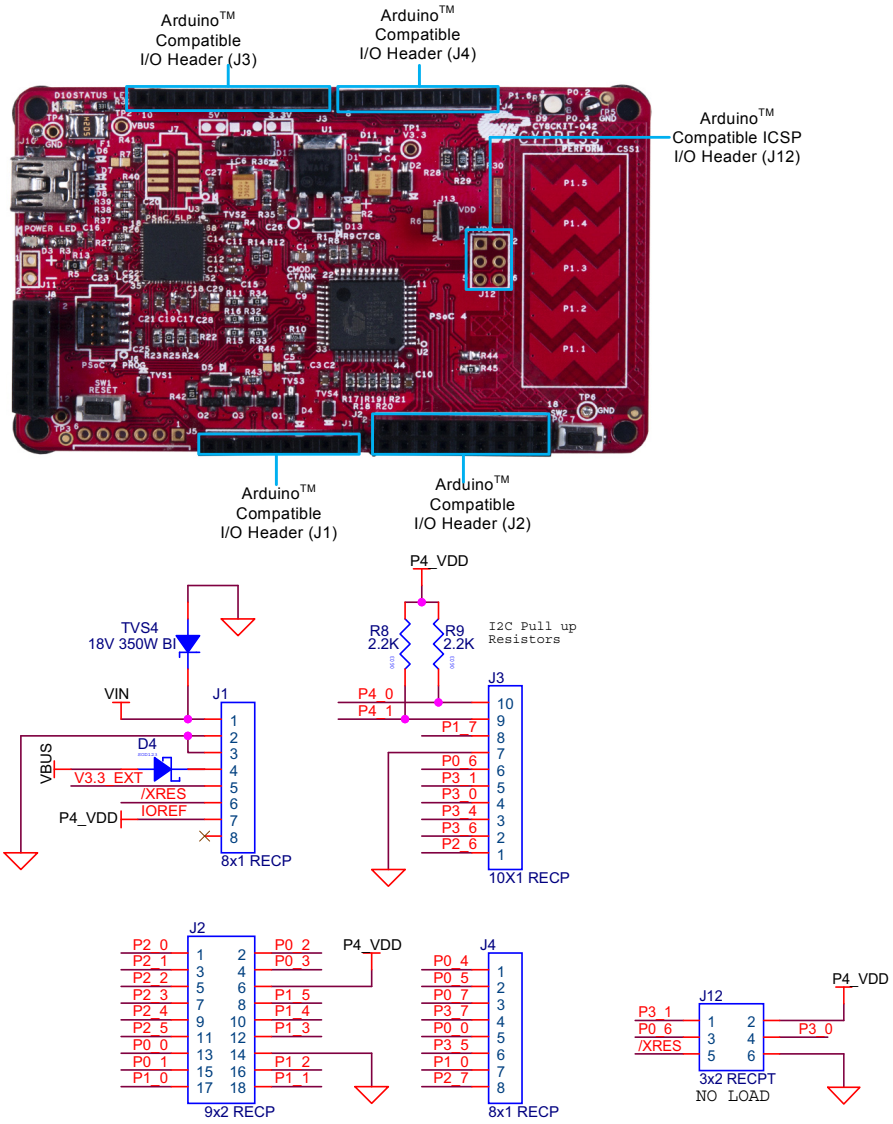
4.3.5 Arduino Compatible Headers (J1, J2, J3, J4, and J12 - unpopulated)

This kit has five Arduino compatible headers; J1, J2, J3, J4 and J12. The user can develop applications based on the Arduino shield's hardware.



The J1 header contains I/O pins for reset, internal reference voltage (IOREF) and power supply line. The J2 header is an analog port. It contains I/O pins for SAR ADC, Comparator and OpAmp. The J3 header is primarily a digital port. It contains I/O pins for PWM, I2C, SPI, and analog reference. The J4 header is also a digital port. It contains I/O pins for UART and PWM. The J12 header is an Arduino ICSP compatible header for SPI interface. This header is not populated. Refer to the section "No load components" of Bill of Materials (BOM) chapter on page 100 in the Appendix for the part number of the header.

Figure 4-8. Arduino Compatible Headers



(J1-J4) Arduino Compatible Headers

4.3.5.1 *Additional Functionality of Header J2*

The J2 header is a 9x2 header that supports Arduino shields. The Port 0, Port 1, and Port 2 pins of PSoC 4 are brought to this header. The Port 1 pins additionally connect to the onboard CapSense slider through 560-ohm resistors. When the CapSense feature is not used, remove these resistors to ensure a better performance with these pins.

4.3.5.2 *Functionality of Unpopulated Header J12*

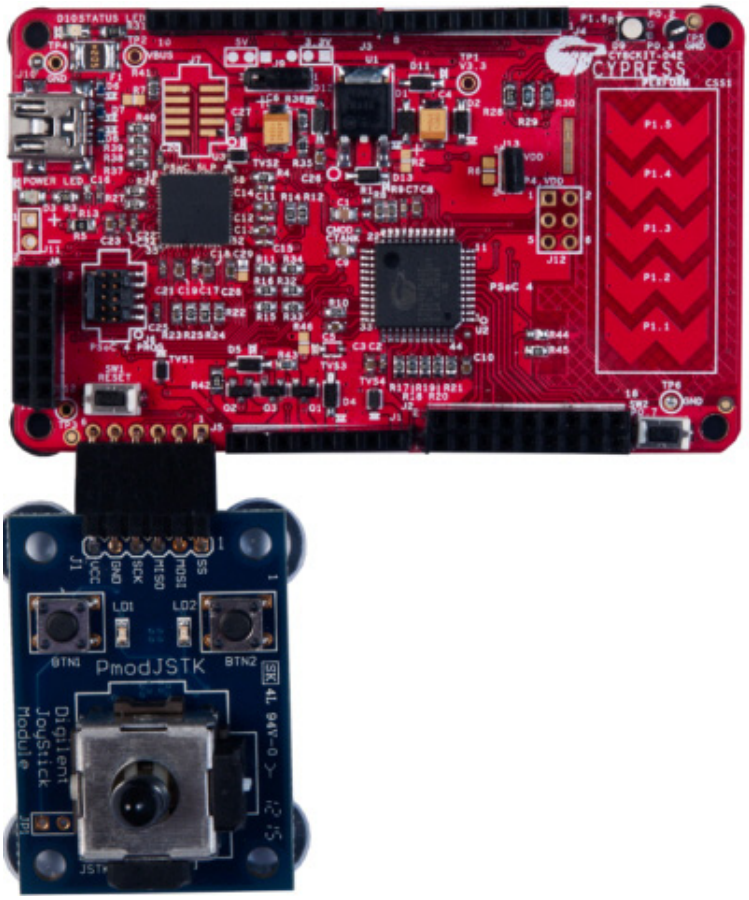
The J12 header is a 2 x 3 header that supports Arduino shields. This header is used on a small subset of shields and is unpopulated on the PSoC 4 Pioneer Kit. Note that the J12 header only functions in 5.0 V mode. To ensure proper shield functionality, the user must ensure the power jumper is connected in 5.0 V mode.

4.3.6 Digilent Pmod Compatible Header (J5 - unpopulated)

This port supports Digilent Pmod peripheral modules. Pmods are small I/O interface. Pmods interface with the embedded control boards through either 6- or 12-pin connectors. The PSoC Pioneer kit supports the 6-pin Pmod type 2 (SPI) interface. For Digilent Pmod cards, go to www.digilentinc.com.

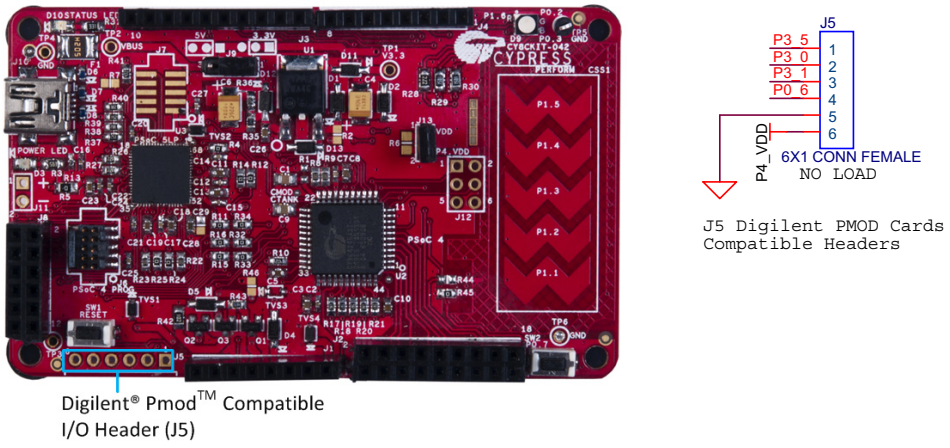
This header is not populated on the PSoC 4 Pioneer kit. The user is expected to populate this header before interfacing PMod daughter cards.

Figure 4-9. Pmod Connection



This header is not populated on the Pioneer kit. Refer to the section “No load components” of Bill of Materials (BOM) chapter on page 100 in Appendix for the part number of the header.

Figure 4-10. Digilent PMOD Interface

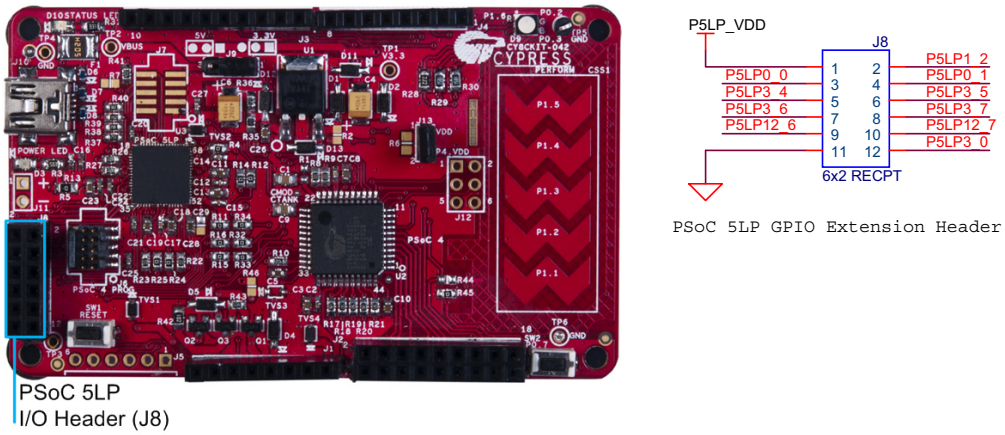


See Pin Assignment Table chapter on page 96 for details on the pin descriptions for the J5 header.

4.3.7 PSoC 5LP GPIO Header (J8)

A limited set of the PSoC 5LP pins are brought to this header. Refer to the section, Developing Applications for the PSoC 5LP chapter on page 77 to develop custom applications. Refer to the section, Pin Assignment Table chapter on page 96 of the Appendix for pin details.

Figure 4-11. PSoC 5LP GPIO Header (J8)



4.3.8 CapSense Slider

The kit has a five-segment linear capacitive touch slider on the board, which is connected to pins P1[1]–P1[5] of the PSoC 4 device.

The Cmod (modulation capacitor) is connected to pin P4[2] and an optional bleeder resistor (R1) can be connected across the Cmod. This board supports CapSense designs that enable water proofing.

The Water proofing design uses a concept called shield which is a conductor placed around the sensors. This shield has to be connected to a designated shield pin on the device to function. The shield must be connected to the ground when not used. On the PSoC 4 Pioneer kit, the connection of the shield to the pin or to the ground is made by resistors R44 and R45 respectively. By default, R45 is mounted on the board which connects the shield to the ground. Populate R44 when evaluating water proofing designs, which will connect the shield to the designated pin, P0[1]. This shield is different from the “Arduino shields,” which are boards that connect over the Arduino header. Refer to the CapSense Design guide for further details related to CapSense.

Figure 4-12. CapSense Slider

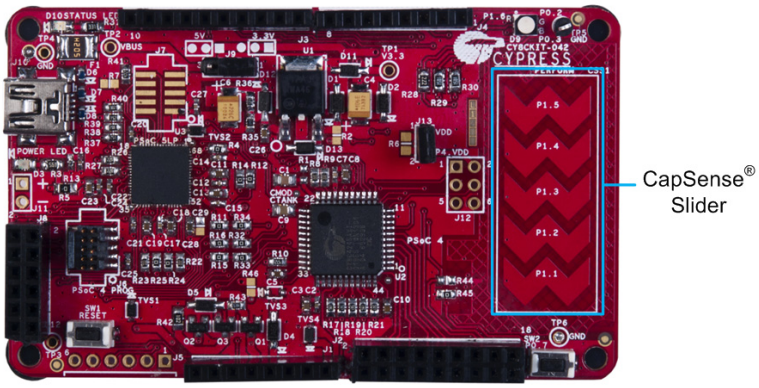
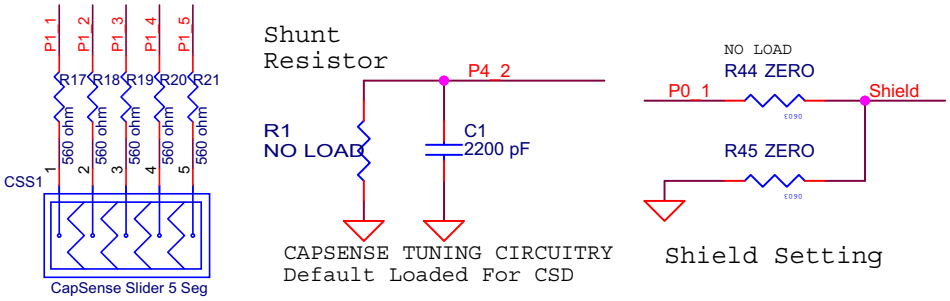


Figure 4-13. CapSense Slider Connection



4.3.9 Pioneer Board LEDs

The PSoC 4 Pioneer board has three LEDs. A green LED (D10) that indicates the status of the programmer. Refer to section A.5 Error in Firmware / Status Indication in Status LED for a detailed list of LED indications. An amber LED (D3) indicates status of power supplied to the board. The kit also has a general purpose tricolor LED (D9) for user applications that connect to specific pins of PSoC 4.

Figure 4-14 shows the indication of all the above mentioned LEDs on board. Figure 4-15 and Figure 4-16 detail the schematic of the LEDs.

Figure 4-14. Pioneer Kit LEDs

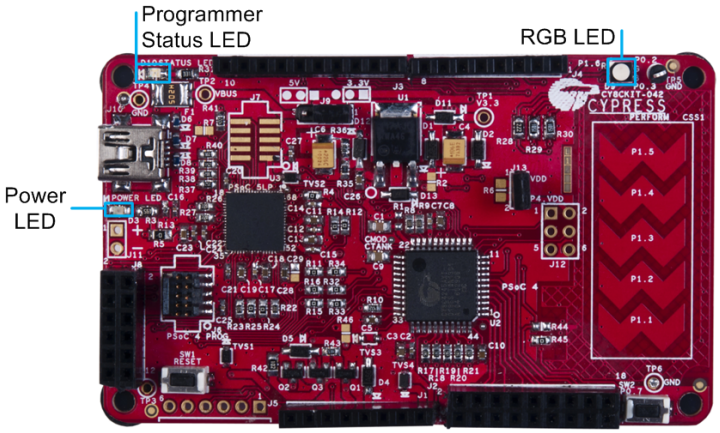


Figure 4-15. Status LED and Power LED

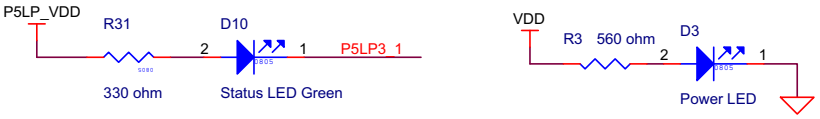
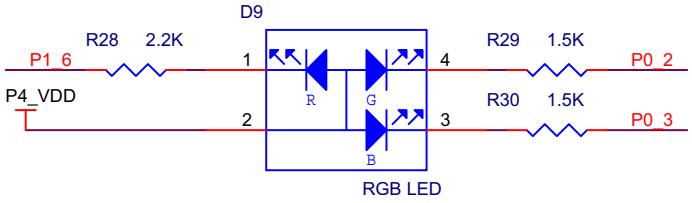


Figure 4-16. RGB LED

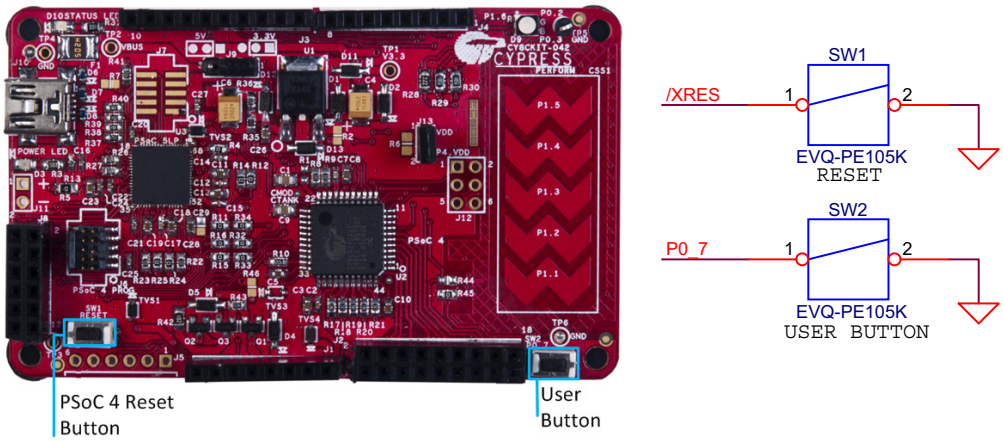


4.3.10 Push Buttons

The kit contains a reset push button and a user push button as shown in Figure 4-17.

The reset push button is connected to the XRES pin of the PSoC 4 and is used to reset the PSoC 4 device onboard. The user push button is connected to P0[7] of PSoC 4 device. Both the push buttons connect to ground on activation, i.e., active low.

Figure 4-17. Push Buttons



5. Example Projects

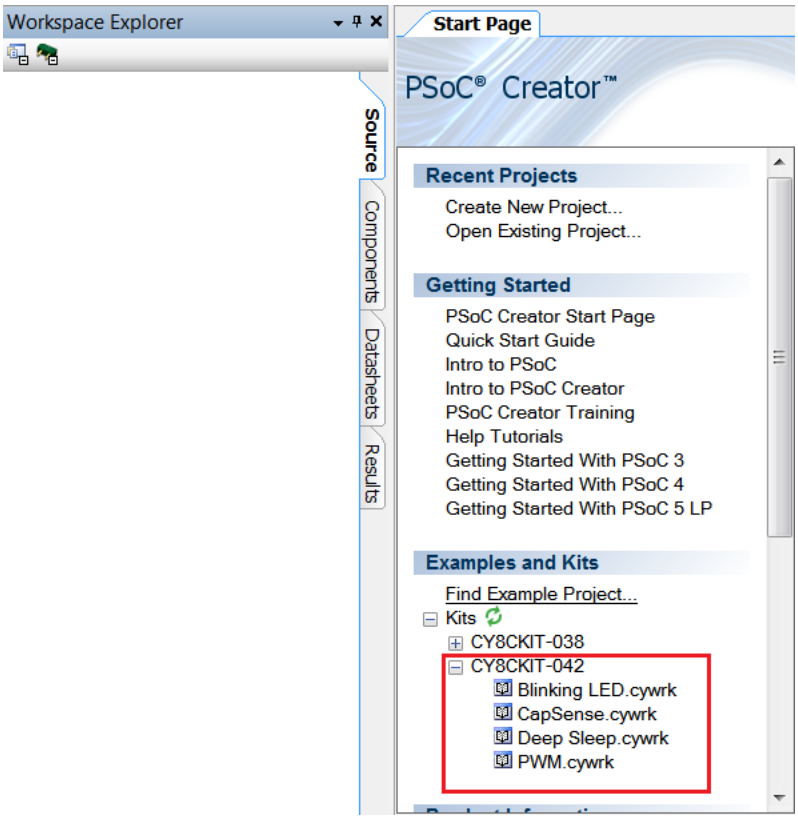
The example projects described in the following sections introduce the functionality of the PSoC 4 device and the onboard components to the user.

To access the code examples described in this section, download them from the kit web page.

Follow these steps to open and program code examples:

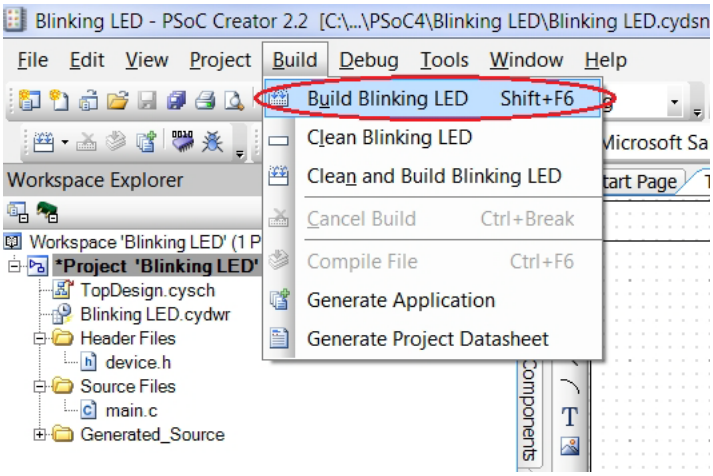
1. Launch the PSoC Creator from the Start menu.
2. Open the example project by clicking the **<Project.cywrk>** present below the **Examples and Kits > Find Example Project > Kits > CY8CKIT-042**.

Figure 5-1. Open Example Project from PSoC Creator



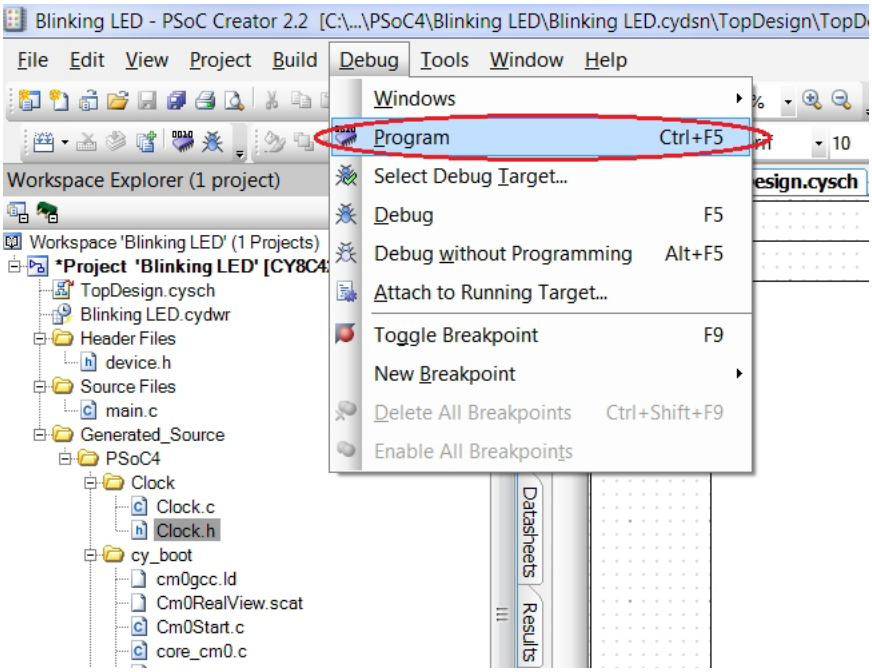
3. Build the code example by clicking **Build** > **Build <Project name>** to generate the hex file.

Figure 5-2. Building the Project from PSoC Creator



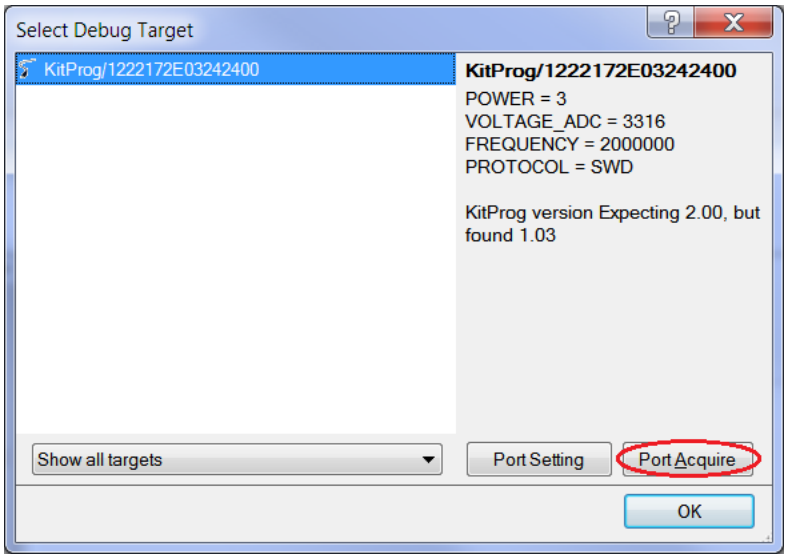
4. To program, connect the board to a computer using the USB cable connected to port J10, as described in section 3.1 Pioneer Kit USB Connection. The board is detected as **KitProg**.
5. Click **Debug > Program** from PSoC Creator.

Figure 5-3. Programming the Device from PSoC Creator



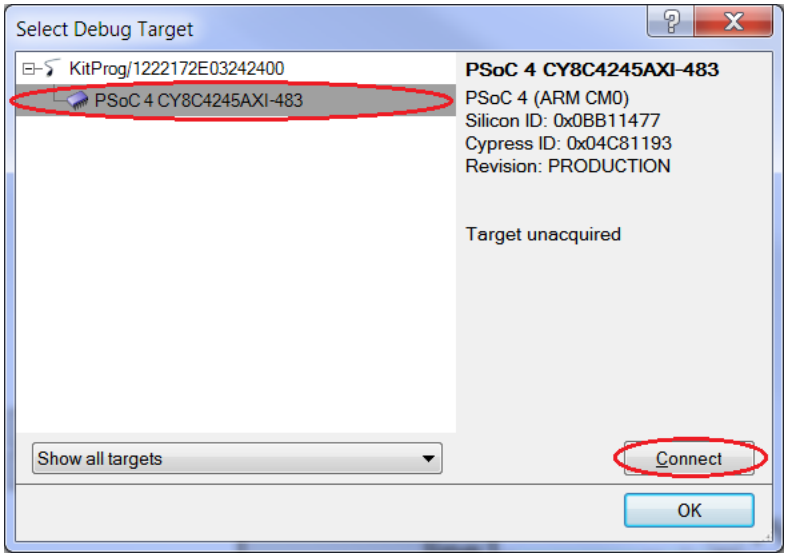
6. If the device is not yet acquired PSoC Creator will open the Programming window. Select **KitProg/** and click on the **Port Acquire** button.

Figure 5-4. Acquiring the Device from PSoC Creator



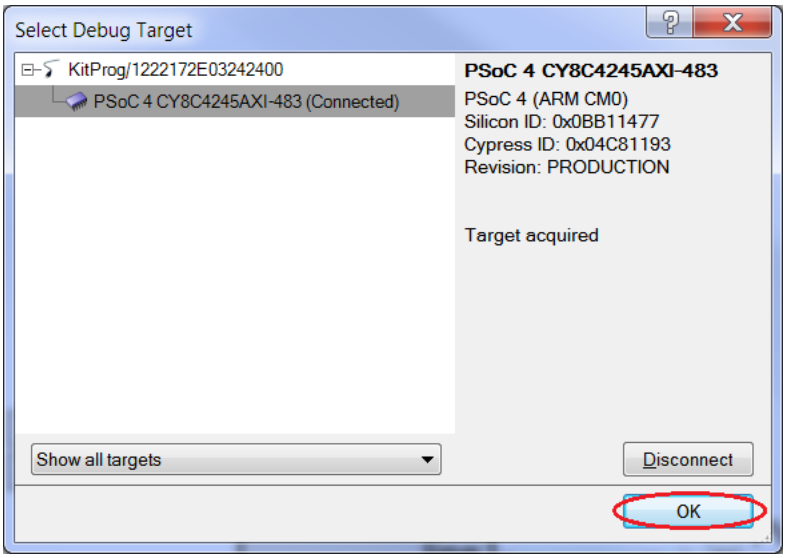
7. After the device is acquired, it is shown in a tree structure below the **KitProg**. Now, click on the **Connect** button.

Figure 5-5. Connecting the Device from PSoC Creator



8. Click **OK** to exit the window and start programming.

Figure 5-6. Programming the device from PSoC Creator



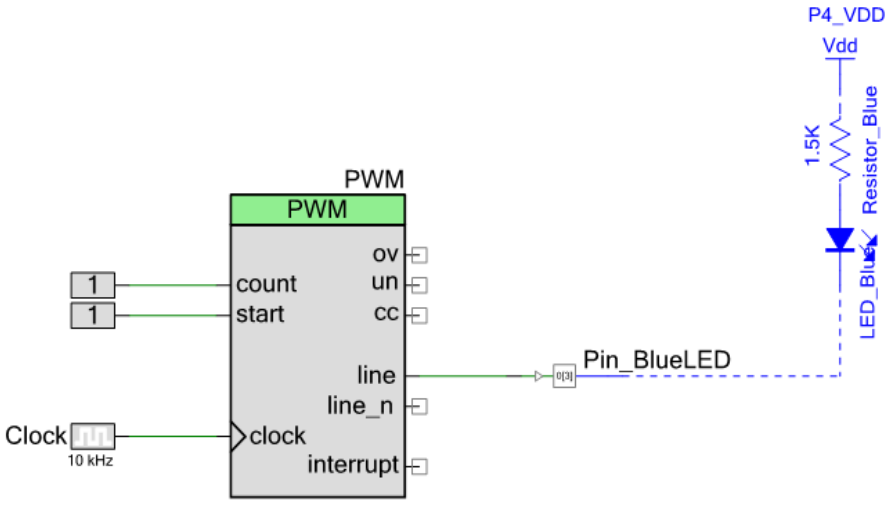
5.1 Project: Blinking LED

5.1.1 Project Description

This example code uses a pulse-width modulator (PWM) to illuminate the RGB LED. The output of the PWM is connected to pin P0_3 (Blue) of the RGB LED. The frequency of blinking is set to 1 Hz with a duty cycle of 50%. The frequency of blinking and the duty cycle can be varied by varying the period and compare value respectively.

Note: The PSoC 4 Pioneer Kit is factory programmed with this example.

Figure 5-7. PsoC Creator Schematic Design of Blinking LED Project



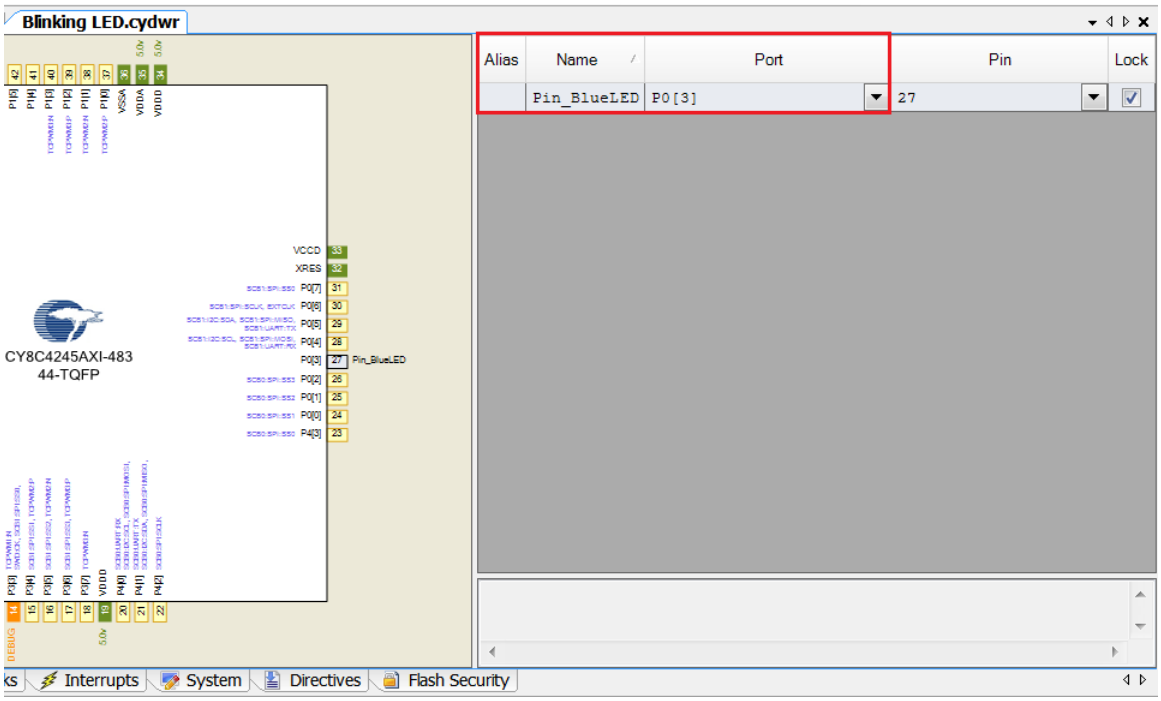
5.1.2 Hardware Connections

There are no specific hardware connections required for this project because all connections are hard wired on the board. Open the Blinking LED.cydwr in the Workspace explorer and select the suitable pin.

Table 5-1. Pin Connection

Pin Name	Port Name
PWM	P0_3 (Blue)

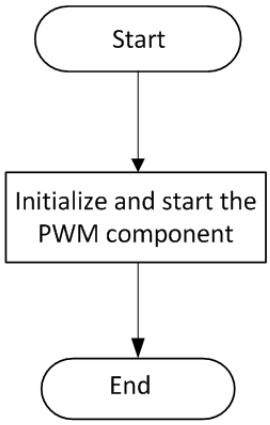
Figure 5-8. Pin Selection for Blinking LED Project



5.1.3 Flowchart

Figure 5-7 shows the flowchart of code implemented in main.c

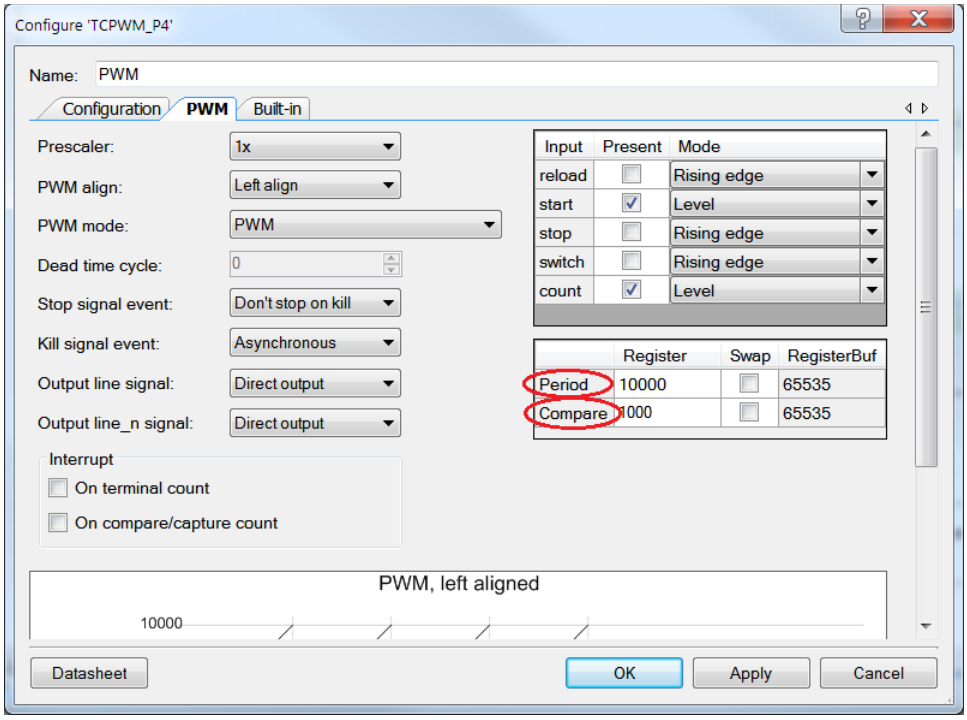
Figure 5-9. Flowchart of Blinking LED Project



5.1.4 Verify Output

Build and program the code example on to the device. Observe the frequency and the duty cycle of the blinking LED. Change the period and compare value in the PWM component as shown in the Figure 5-10. Rebuild and reprogram the device to vary the frequency and duty cycle.

Figure 5-10. Configuration window of PWM Component

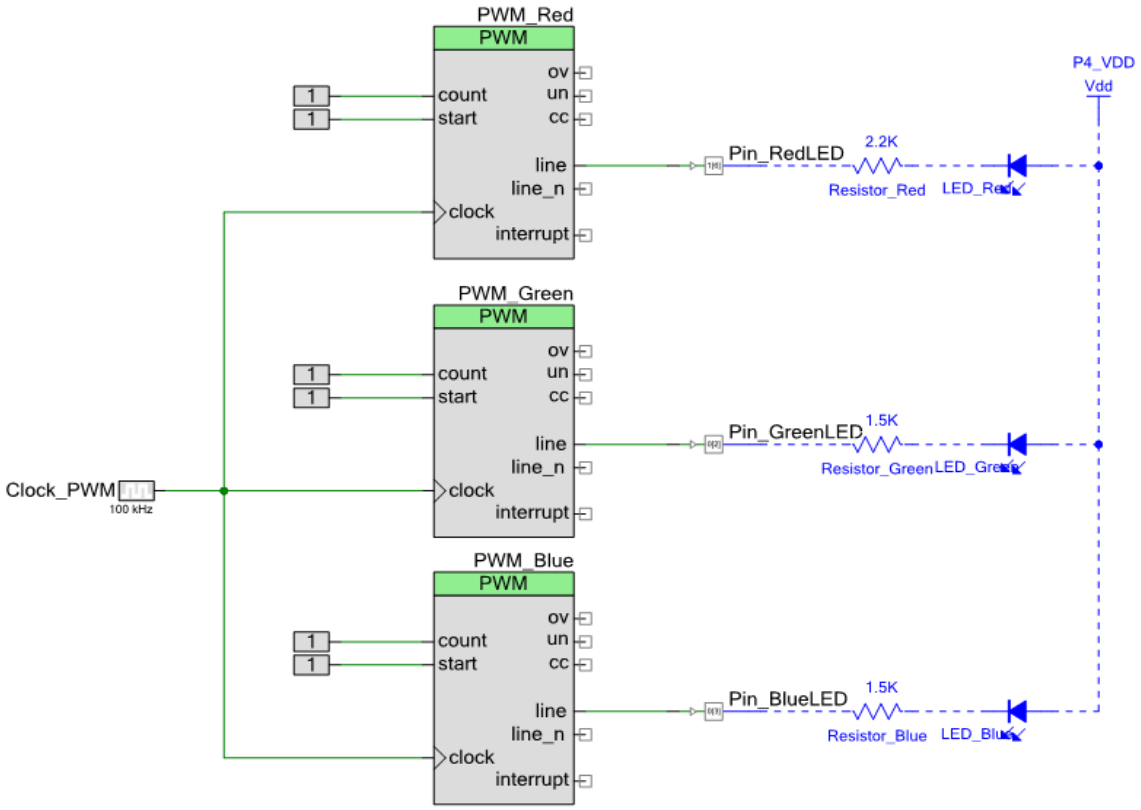


5.2 Project: PWM

5.2.1 Project Description

This example project demonstrates the use of the PWM component. The project uses three PWM outputs to set the color of RGB LED on the Pioneer kit. The LED cycles through seven colors – violet > indigo > blue > green > yellow > orange > red (VIBGYOR). Each color is maintained for a duration of 1 second. The different colors are achieved by changing the pulse width of the PWMs.

Figure 5-11. PSoC Creator Schematic Design of PWM Project



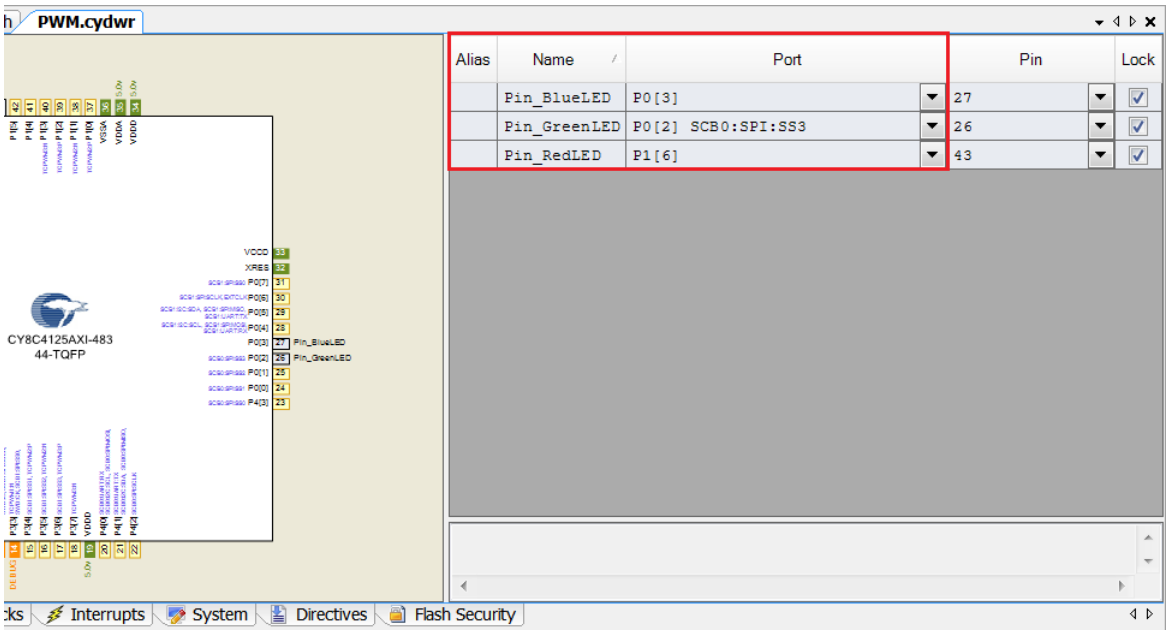
5.2.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hard wired on the board. Open PWM.cydwr in the Workspace explorer and select the suitable pins.

Table 5-2. Pin Connections

Pin Name	Port Name
PWM1	P1_6 (Red)
PWM2	P0_2 (Green)
PWM3	P0_3 (Blue)

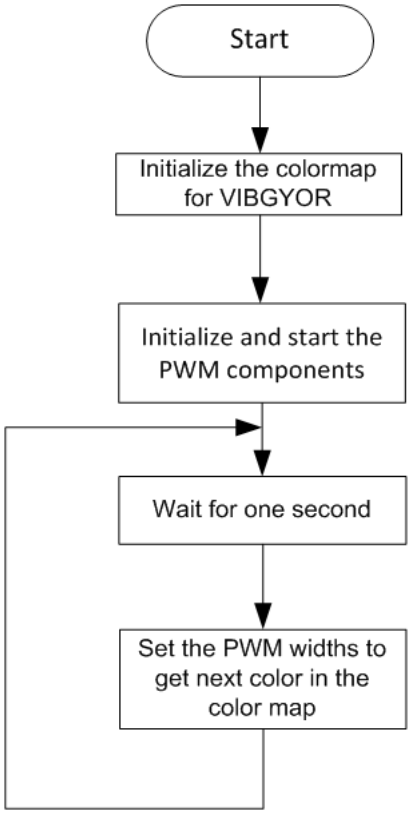
Figure 5-12. Pin Selection for PWM Project



5.2.3 Flowchart

Figure 5-13 shows the flowchart of code implemented in C.

Figure 5-13. Flowchart of PWM project



5.2.4 Verify Output

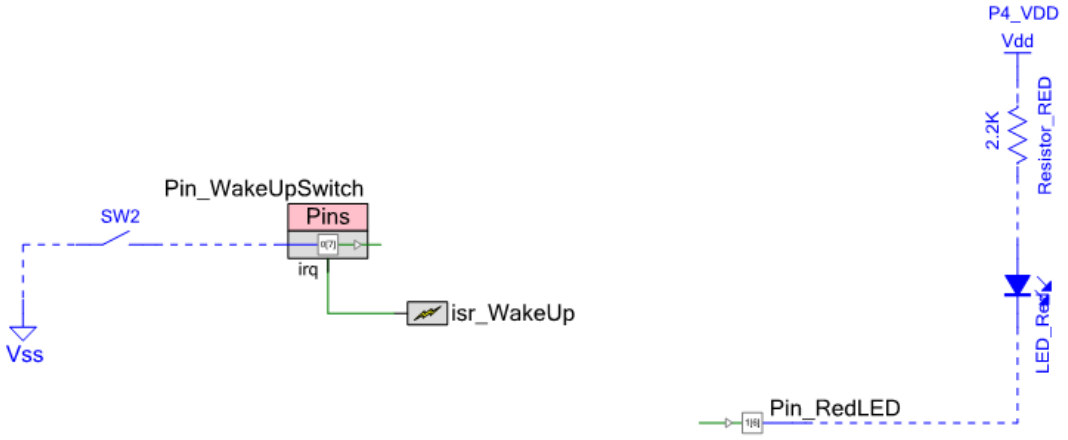
Build and program the code example, and reset the device. Observe the RGB LED cycles through the color pattern.

5.3 Project: Deep Sleep

5.3.1 Project Description

This project demonstrates the low power functionality of the PSoC 4. The LED is turned on for one second to indicate active mode and then the device enters into deep sleep mode. Whenever switch SW2 is pressed, the device wakes up and the LED is turned on for one second and then goes back into Deep Sleep mode.

Figure 5-14. PSoC Creator Schematic Design of Deep Sleep Project



5.3.2 Hardware Connections

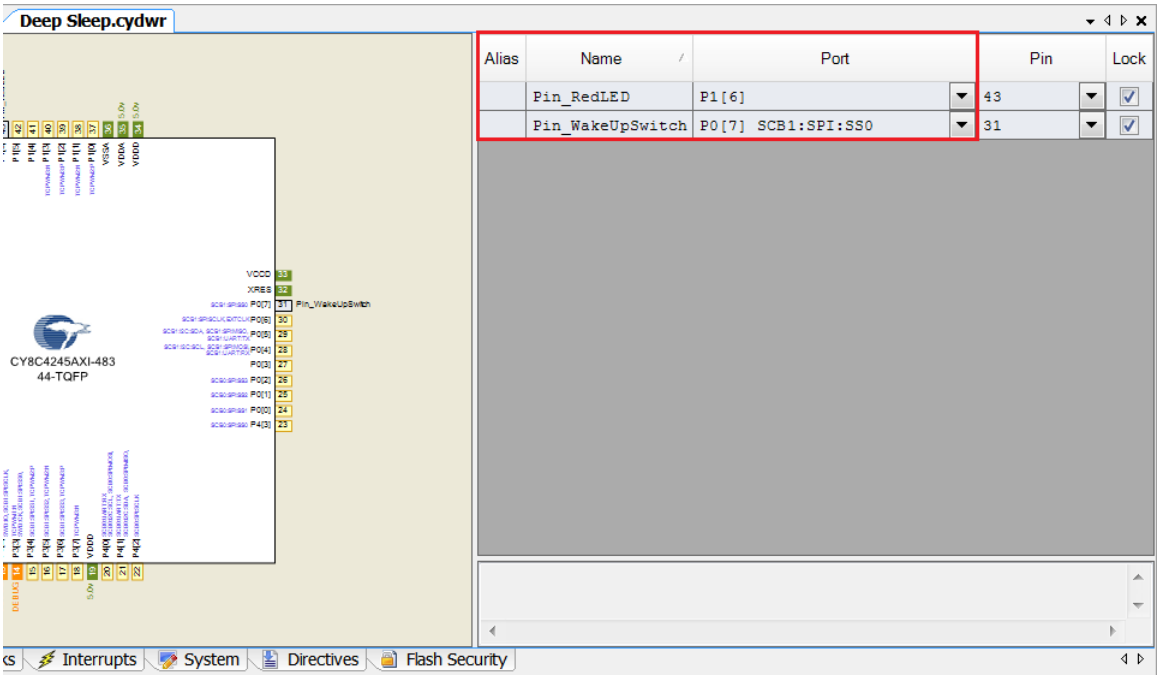
No extra connections are required for the project functionality as the connections are hardwired onto the board. To make low-power measurements using this project, refer to the use case detailed in section 4.3.3.2 Procedure to Measure the PSoC 4 Current Consumption on page 34.

Open Deep Sleep.cydwr in the Workspace explorer and select suitable pin.

Table 5-3. Pin Connection

Pin Name	Port Name
LED	P1_6 (Red)
Switch	P0_7

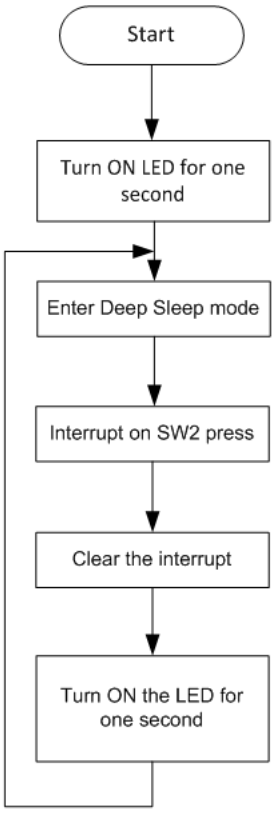
Figure 5-15. Pin Selection for Deep Sleep Project



5.3.3 Flowchart

Figure 5-16 shows the flowchart of code implemented in C.

Figure 5-16. Flowchart of Deep Sleep project



5.3.4 Verify Output

Build and program the code example, and reset the device. LED is on for one second and turns off, which indicates that the device has entered Deep sleep mode. Press SW2 switch to wake up the device from Deep Sleep mode and enter Active mode. The device goes back to sleep after one second.

Note: When the device is in deep sleep mode, the programmer will have to reacquire the device before programming can start.

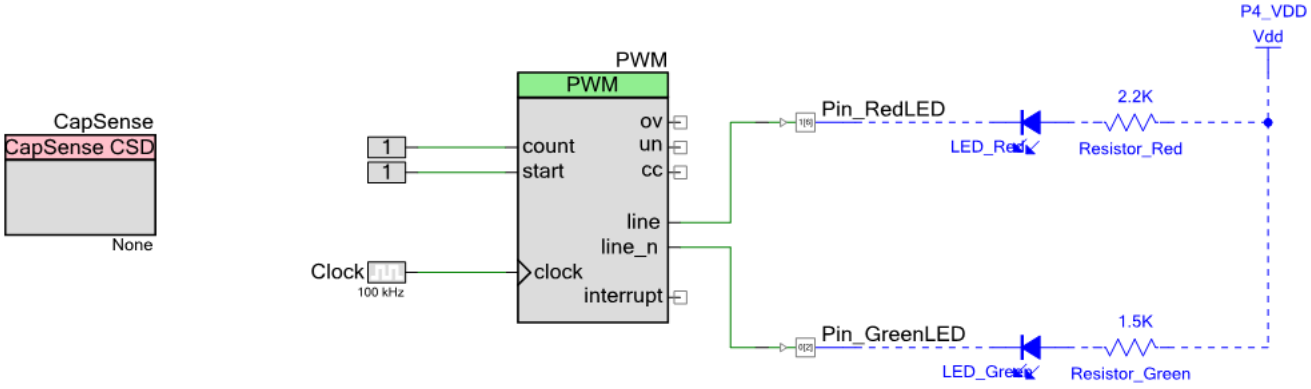
5.4 Project: CapSense

5.4.1 Project Description

This example project demonstrates CapSense on PSoC 4. The example project uses the 5-segment CapSense slider on the board. Each capacitive sensor on the slider is scanned using the Cypress's CapSense Sigma Delta (CSD) algorithm implemented in the CapSense component. This project is pre-tuned to take care of the board parasitics. For more information on the CapSense component and CapSense tuning please refer the CapSense component datasheet in PSoC Creator.

In this example project the brightness of the green and red LEDs are varied, based on the position of the user's finger on the CapSense slider.

Figure 5-17. PSoC Creator Schematic Design of CapSense Project



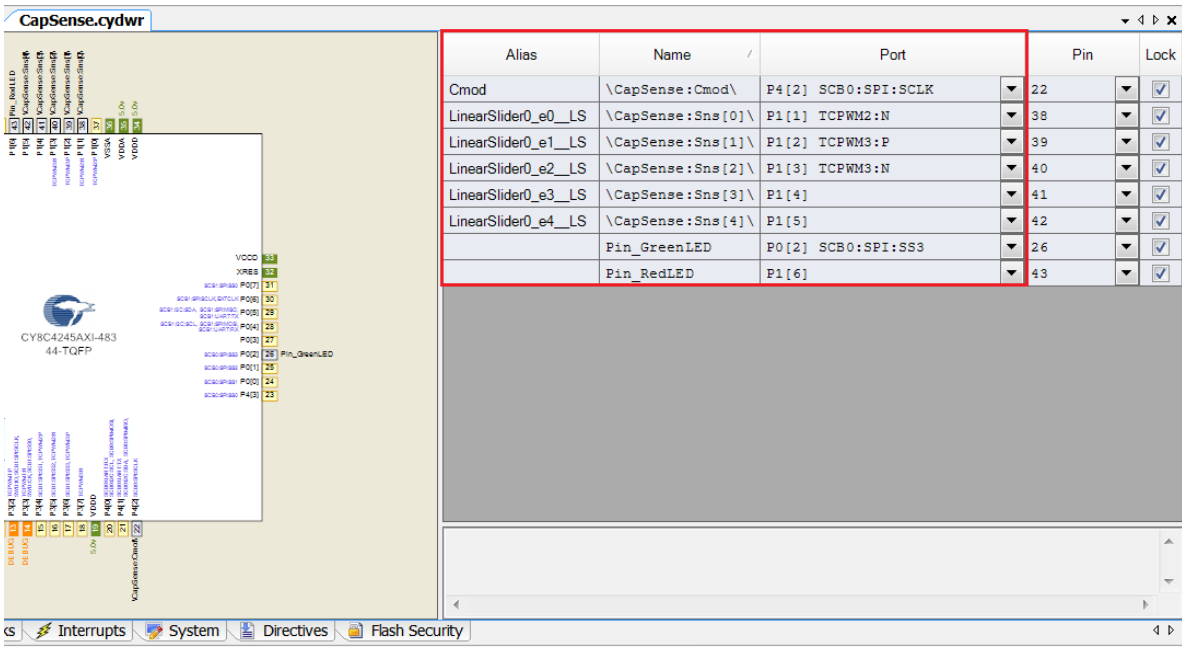
5.4.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hard wired on the board. Open CapSense.cydwr in Workspace explorer and select the suitable pins.

Table 5-4. Pin Connection

Pin Name	Port Name
CapSense Linear Slider	P1_1 Segment1 P1_2 Segment2 P1_3 Segment3 P1_4 Segment4 P1_5 Segment5
LEDs	P1_6(Red) and P0_2(Green)

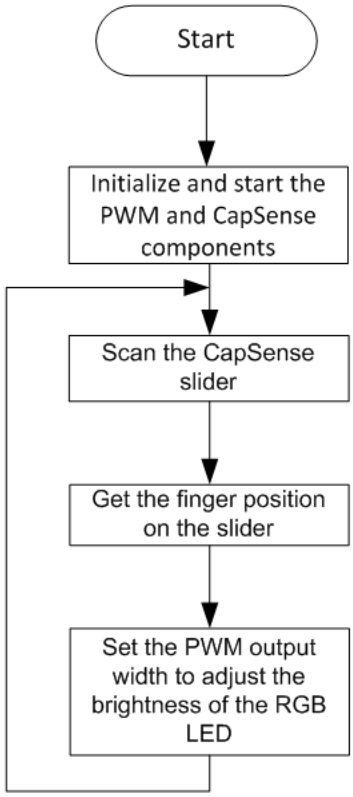
Figure 5-18. Pin Selection for CapSense Project



5.4.3 Flowchart

Figure 5-19 shows the flowchart of code implemented in C.

Figure 5-19. Flowchart of CapSense project



5.4.4 Verify Output

The brightness of the green and the red LEDs are varied based on the position of user's finger on the CapSense slider. When the finger is on segment 5 (P1[5]) of the slider, brightness of the green LED is more compared to the red LED and when the finger is on segment 1 (P1[1]) of slider, brightness of the red LED is more than the green LED.

6. Advanced Section

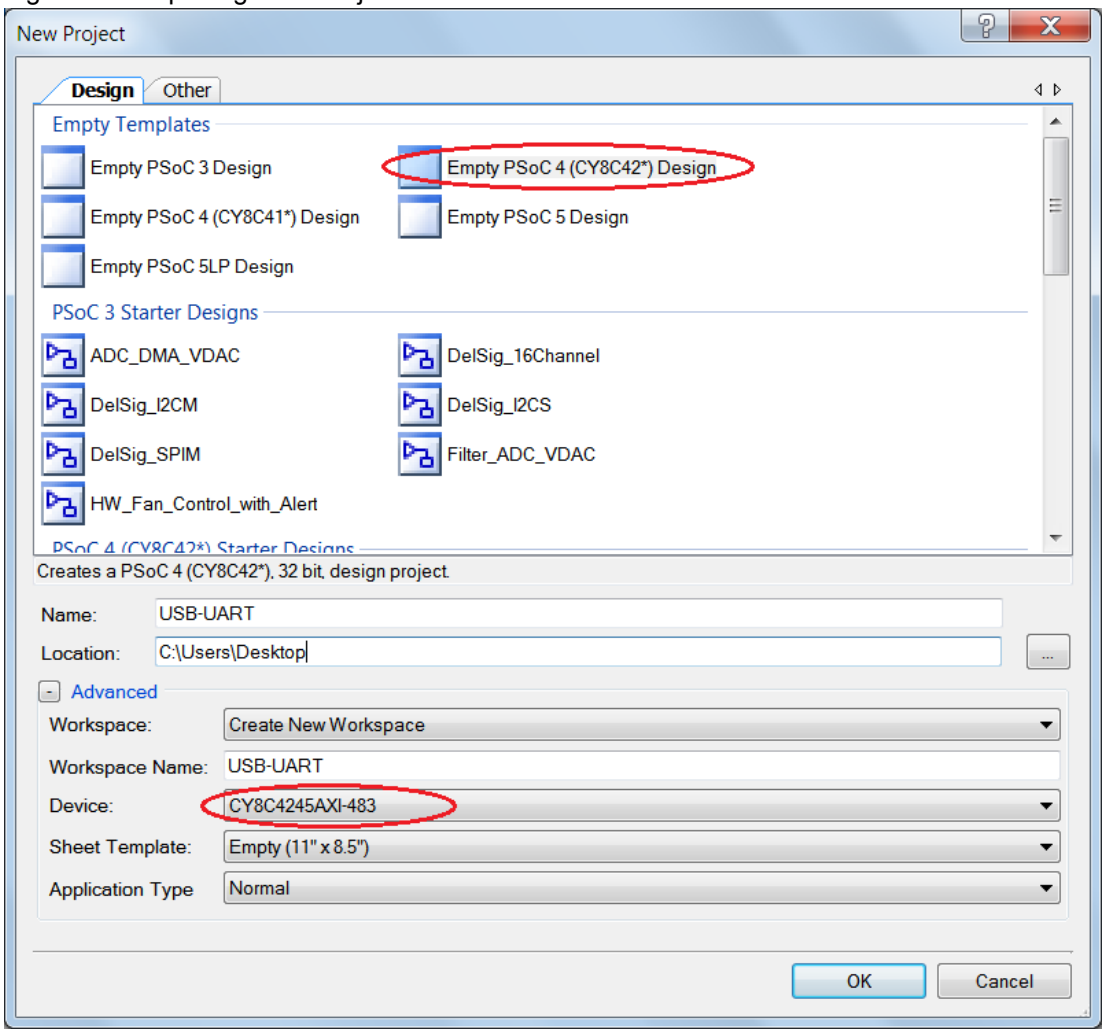
6.1 Using PSoC 5LP as USB-UART Bridge

The PSoC 5LP serves as a USB-UART Bridge, which can be used to communicate with the COM terminl software. This section explains how to create a PSoC 4 example project to communicate with the COM terminal software. This project is available with other example projects for the PSoC 4 Pioneer kit at the element14 partner web page, 100 Projects in 100 days.

Users who have a Windows operating system that does not have a hyperterminal can use an alternate terminal software such as PuTTY.

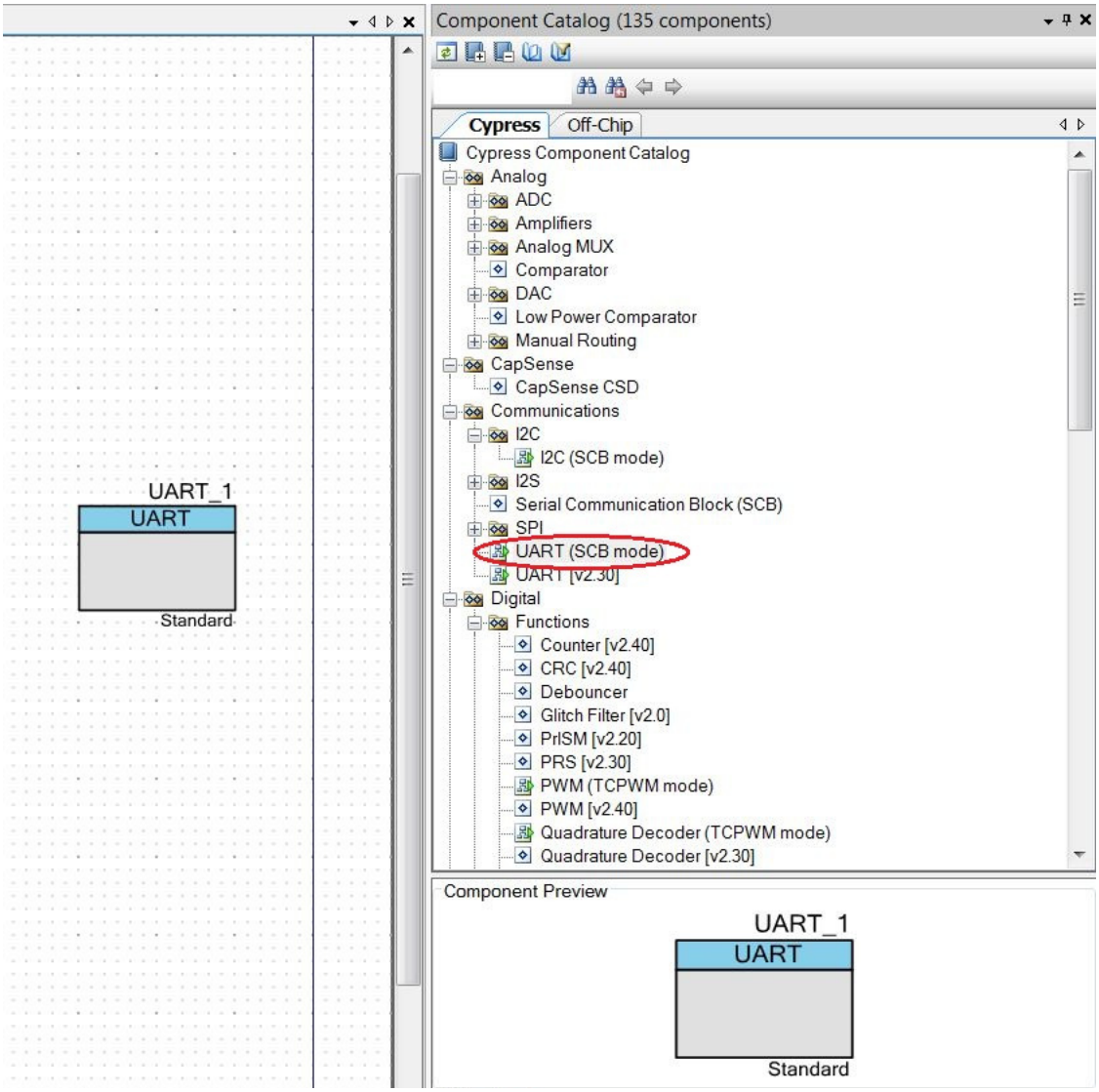
1. Open a new PSoC 4 project in the PSoC Creator.

Figure 6-1. Opening New Project from the PSoC Creator



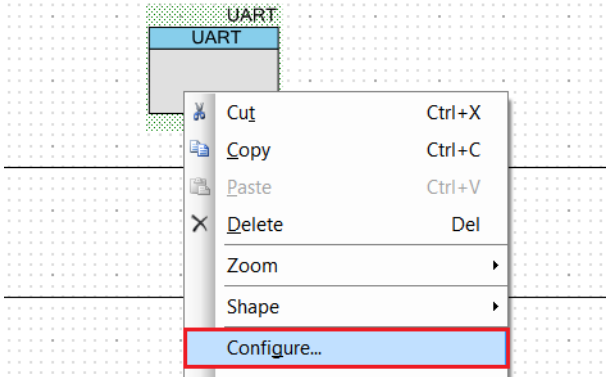
2. Drag and drop a UART (SCB) component to the TopDesign.

Figure 6-2. USB Component Under Component Catalog



3. To configure the UART, double-click or right-click on the UART component and select **Configure**.

Figure 6-3. Open the UART Configuration Window



4. Configure the UART as shown in the below figures.

Figure 6-4. UART Basic Configuration Window

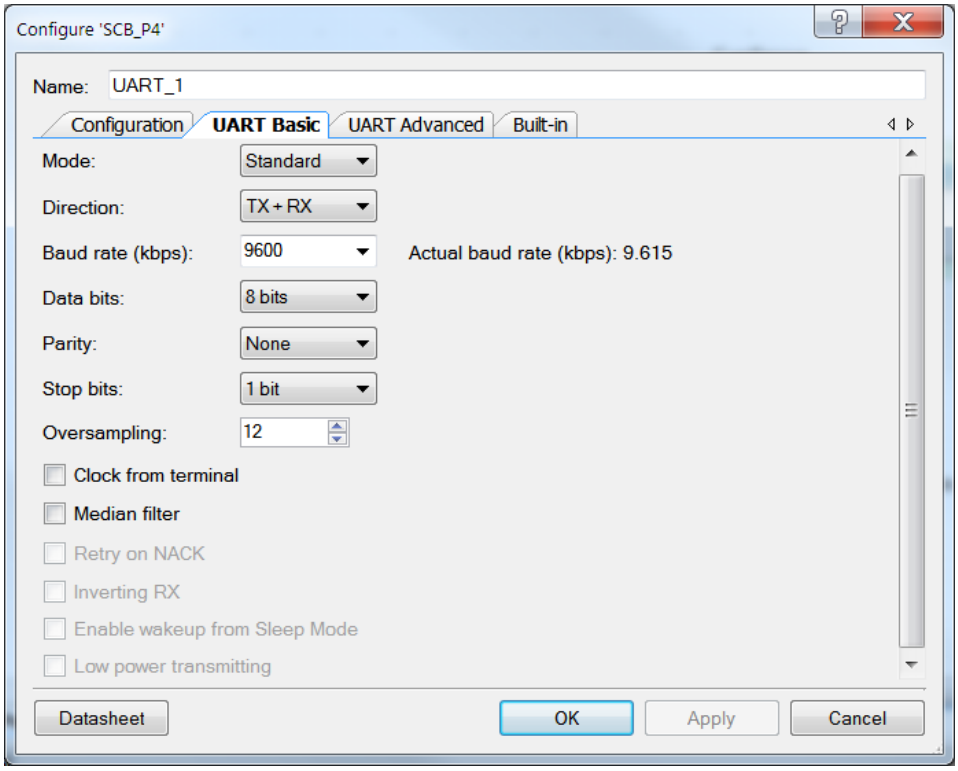
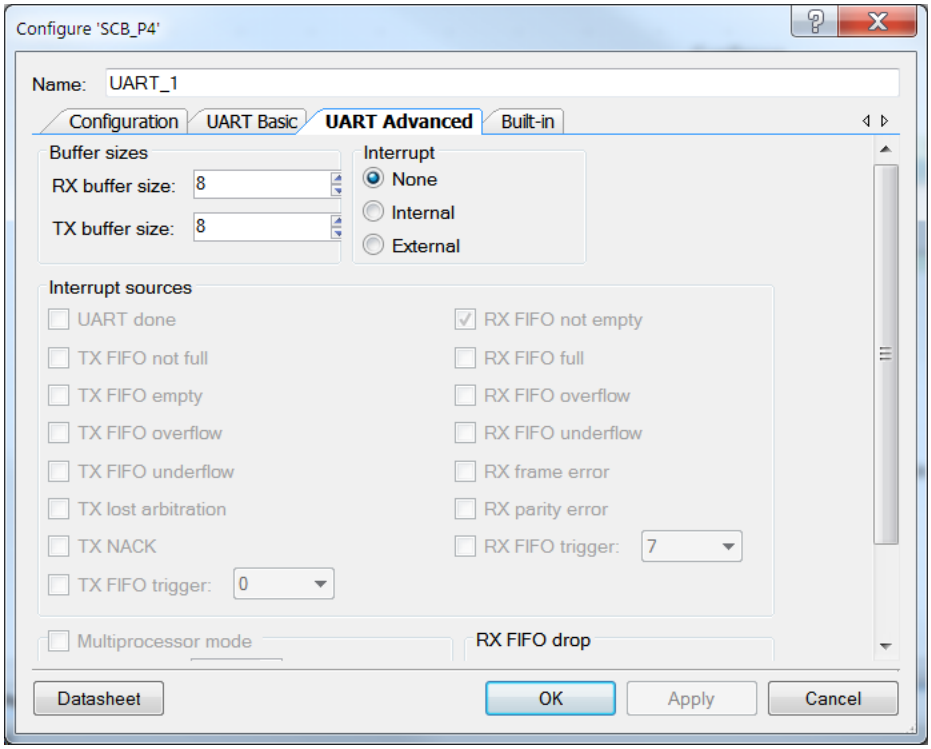
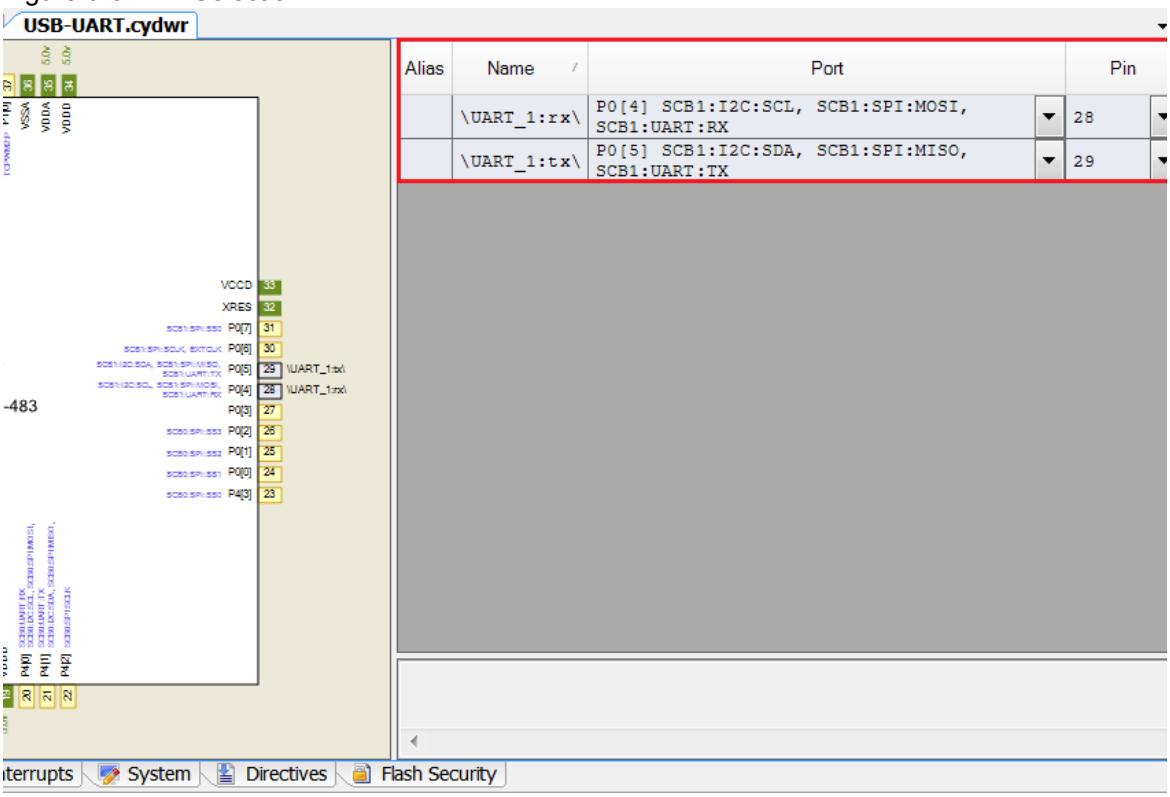


Figure 6-5. UART Advanced Configuration Window



5. Select pin P0[4] for the UART RX and pin P0[5] for the UART TX in the **Pins** tab of <Project.cydwr>.

Figure 6-6. Pin Selection



6. Place the following code in your main.c project file. The code will echo any UART data received.

```
#include <device.h>

void main()
{
    uint8 ch;

    /* Start SCB UART TX+RX operation */
    UART_Start();

/* Transmit String through UART TX Line */
UART_UartPutString("CY8CKIT-042 USB-UART");

    for(;;)
    {
        /* Get received character or zero if nothing has been received yet
*/
        ch = UART_UartGetChar();

        if(0u != ch)
        {
 /* Send the data through UART. This functions is blocking and waits until
there is an entry into the TX FIFO. */
            UART_UartPutChar(ch);
        }
    }
}
```

7. Build the project by clicking **Build > Build {Project Name}** or [Shift] + [F6]. After the project is built without errors and warnings, program (by clicking **Debug > Program**) the project to the PSoC 4 through the PSoC 5LP USB programmer or MiniProg3.

Connect the RX line of the PSoC 4 to J8_10 and TX line of the PSoC 4 to J8_9 as shown in the below figures.

Figure 6-7. UART Connection Between the PSoC 4 and the PSoC 5LP

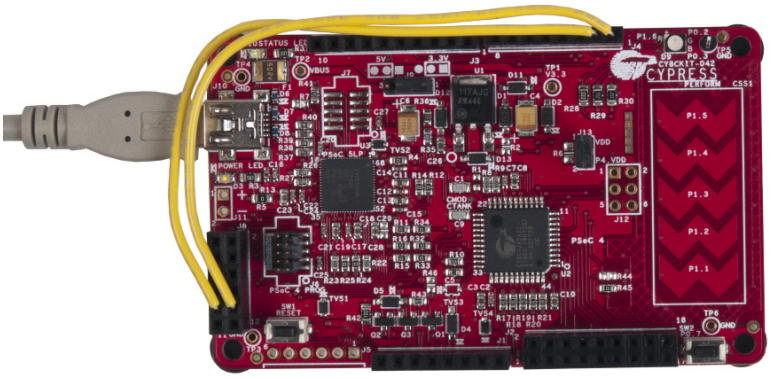
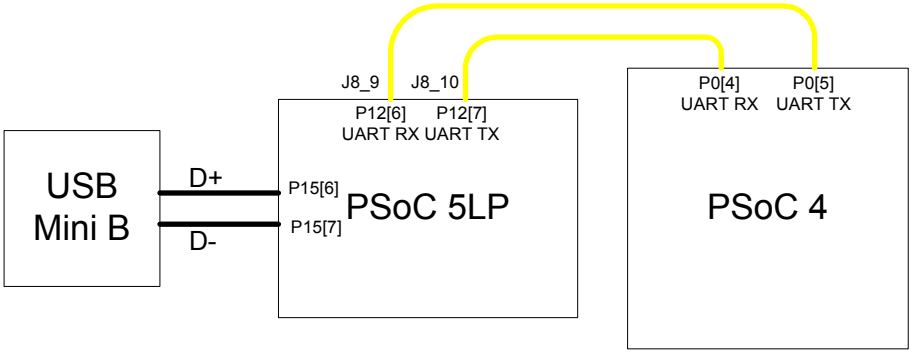


Figure 6-8. Block Diagram of UART Connection Between the PSoC 4 and the PSoC 5LP

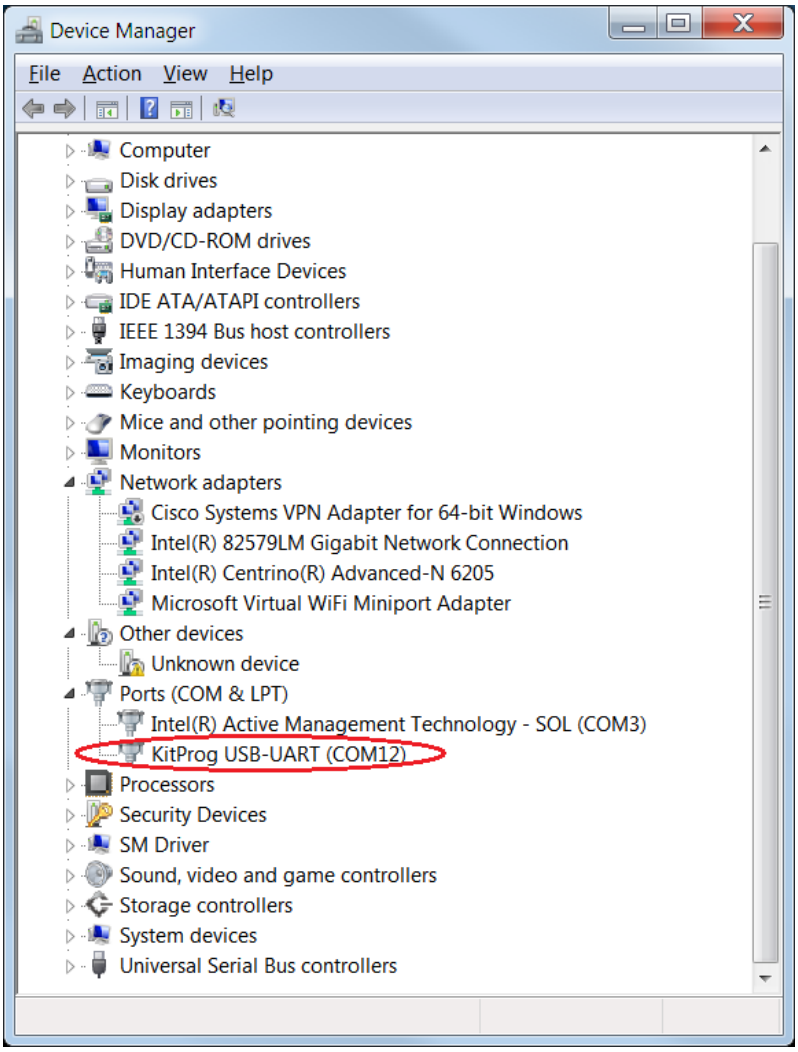


Note: UART RX and UART TX can be routed to any digital pin on PSoC 4 based on the configuration of UART component. A SCB implementation of UART will route the RX and TX pins to either one of the following subsets: (P0[4], P0[5]) or (P3[0],P3[1]) or (P4[0],P4[1]).

To communicate with the PSoC 4 from the terminal software, follow this procedure:

1. Connect USB Mini B to J10. The kit enumerates as a **KitProg USB-UART** and is available under the Device Manager, Ports (COM & LPT). A communication port is assigned to the **KitProg USB-UART**.

Figure 6-9. KitProg USB-UART in Device Manager

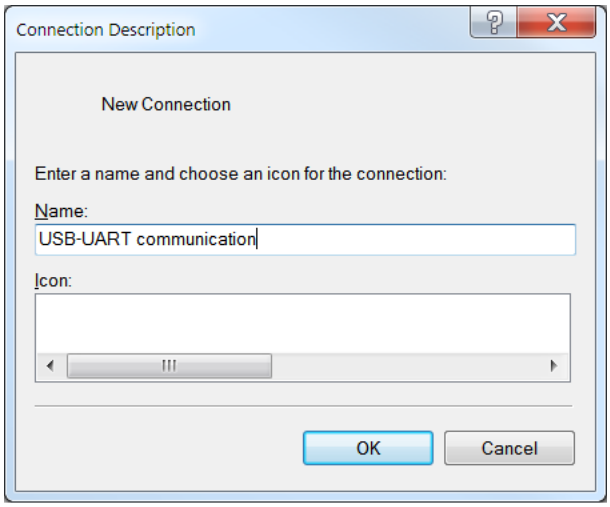


2. Open the hyperterminal and select **File > New Connection** and enter a name for the new connection and click **OK**.

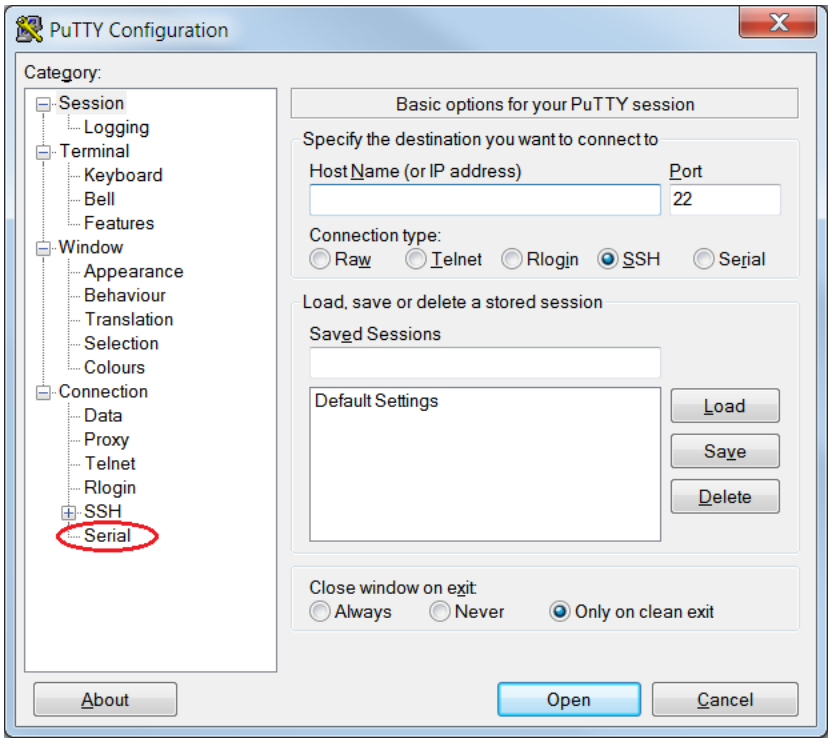
For PuTTY, double click the putty icon and select **Serial** under **Connection**.

Figure 6-10. Open New Connection

Hyperterminal:



PuTTY:



3. A new window opens, where the communication port can be selected.
   In Hyperterminal select COMX (or the specific communication port that is assigned to KitProg USB-UART) in **'Connect using'** and click **OK**.
   In PuTTY enter the COMX in **'Serial line to connect to'**.
   In this example project we use COM12.

Figure 6-11. Select the Communication Port

Hyperterminal:

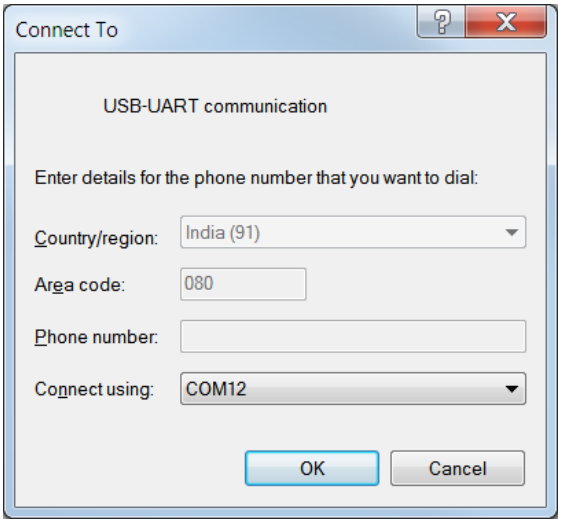
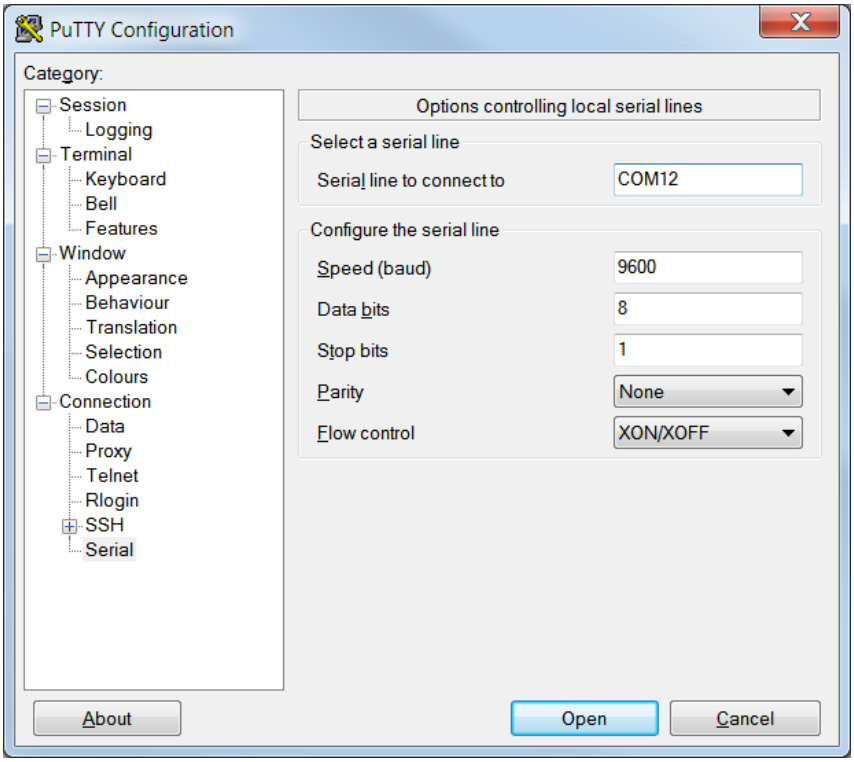


Figure 6-12. Select the Communication Port - PuTTY



4. In Hyperterminal select 'Bits per second', 'Data bits', 'Parity', 'Stop bits', and 'Flow control' under **'Port Settings'** and click **OK**.

Make sure that the settings are identical to the UART settings configured for PSoC 4.

In PuTTY select 'Speed (baud)', 'Data bits', 'Stop bits', 'Parity' and 'Flow control' under **'Configure the serial line'**. Click on **'Session'** select **'Serial'** under **'Connection type'**.

'Serial line' shows the communication port selected (COM12) and the **'Speed'** shows the baud rate selected. Click **Open** to start the communication.

Figure 6-13. Configure the Communication Port - Hyperterminal:

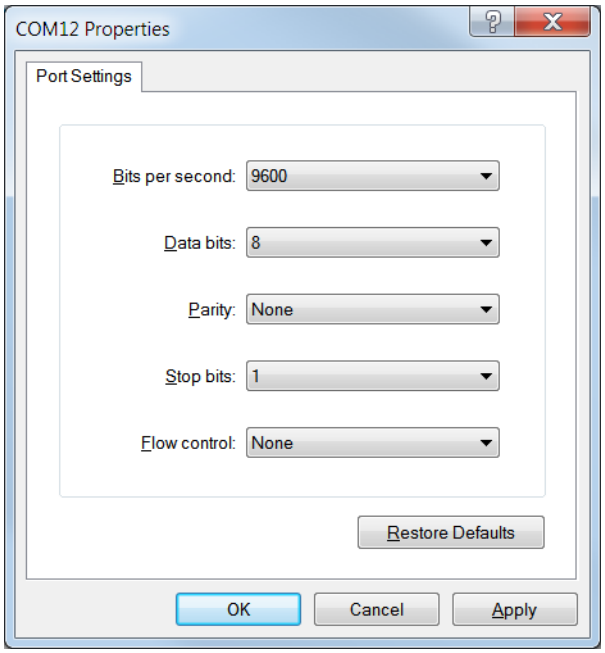


Figure 6-14. Configure the Communication Port - PuTTY:

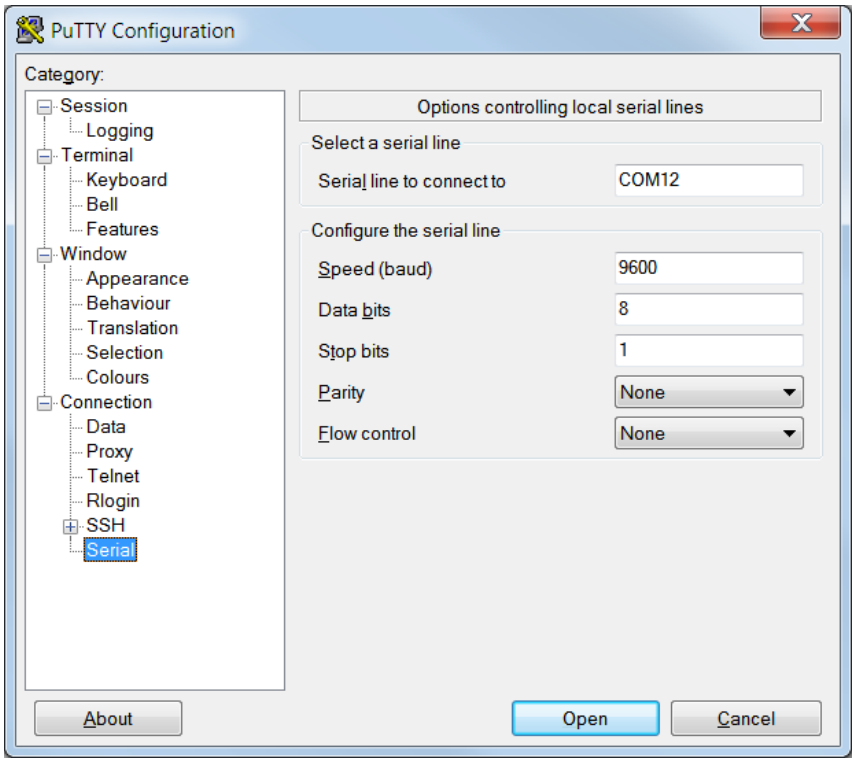
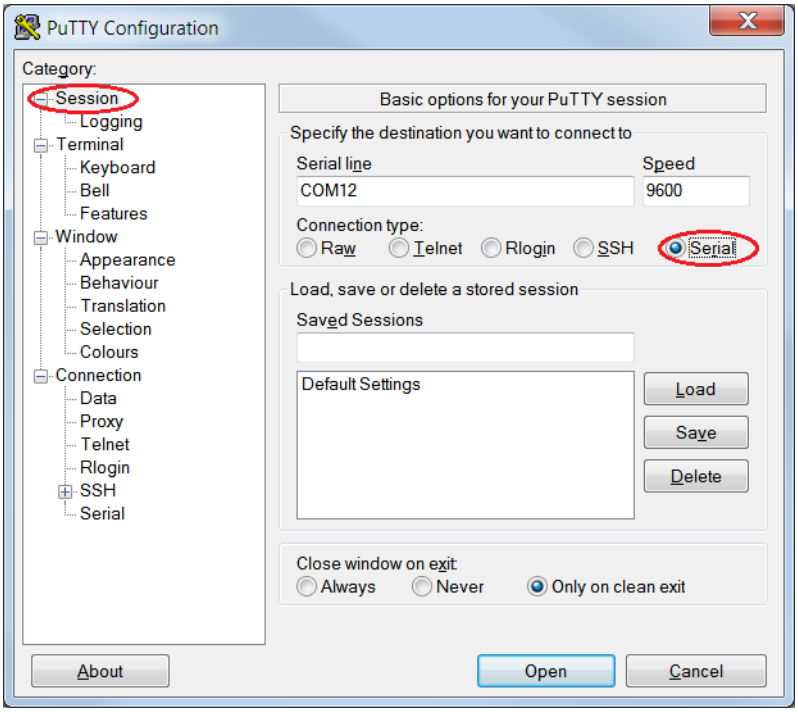


Figure 6-15. Select the Communication Type in PuTTY



5. Enable **Echo typed characters locally** under **File > Properties > Settings > ASCII Setup**, to display the typed characters on the Hyperterminal. In PuTTY enable the **Force on** under **Terminal > Line discipline** options to display the typed characters on the PuTTY.

Figure 6-16. Enabling echo of typed characters in Hyperterminal

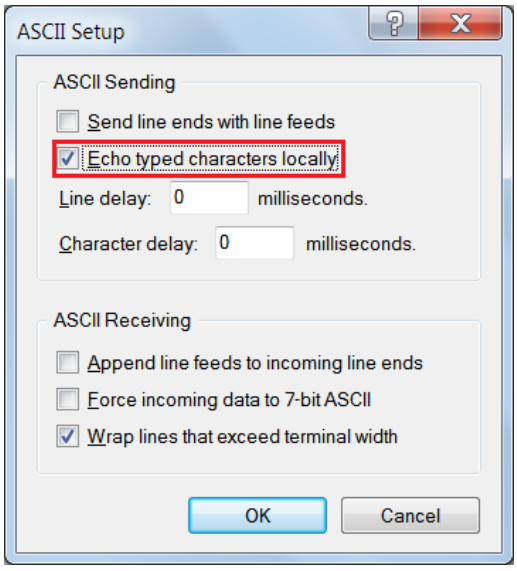
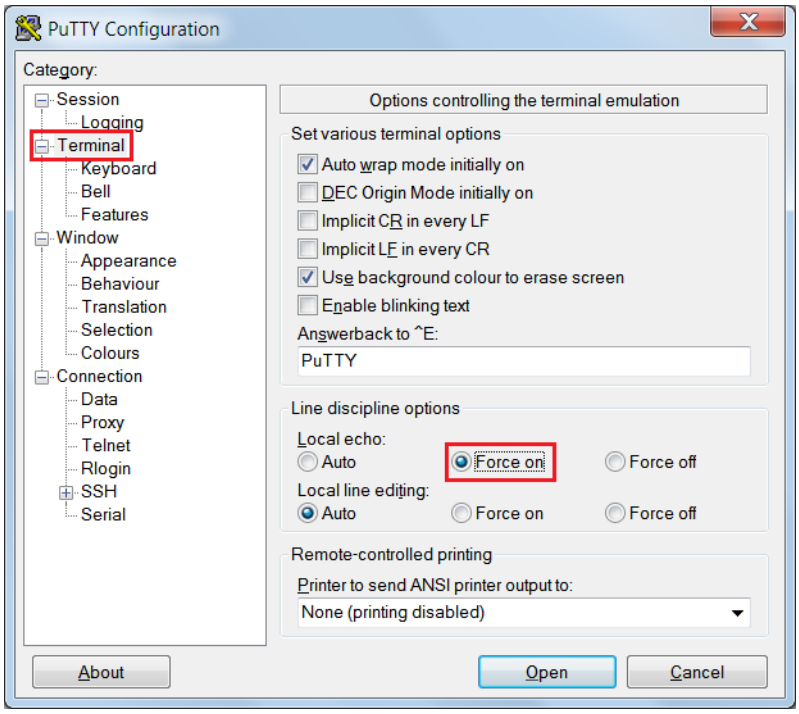


Figure 6-17. Enabling echo of typed characters in PuTTY



6. The COM terminal software displays both the typed data and the looped back data from the UART of PSoC 4.

Figure 6-18. Data Displayed on Hyperterminal

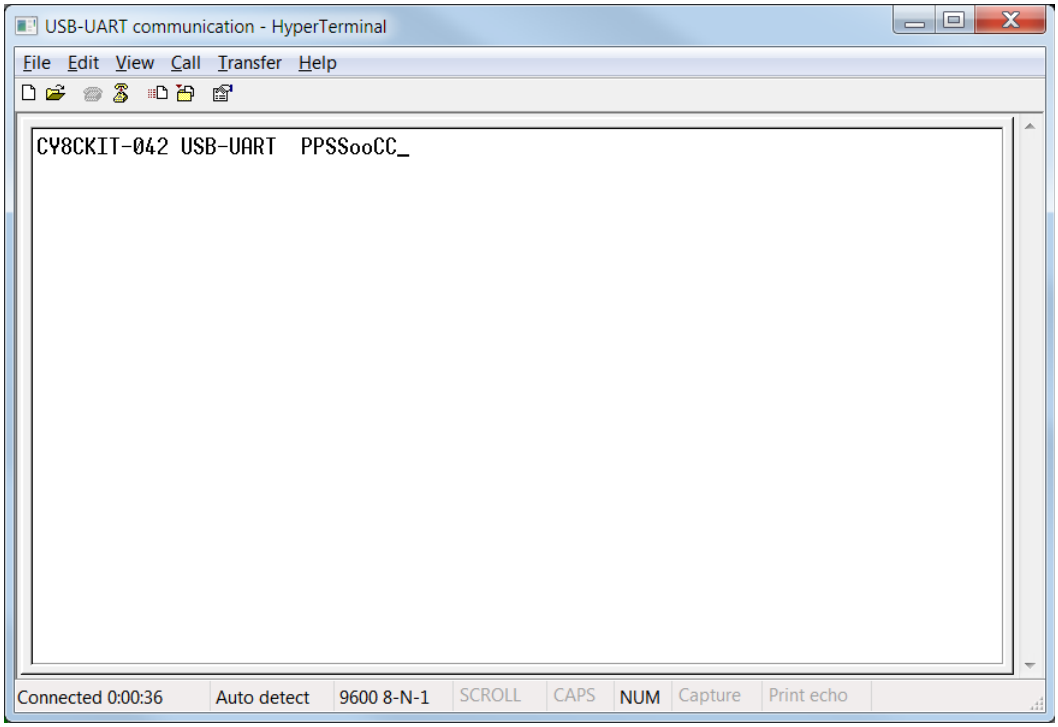
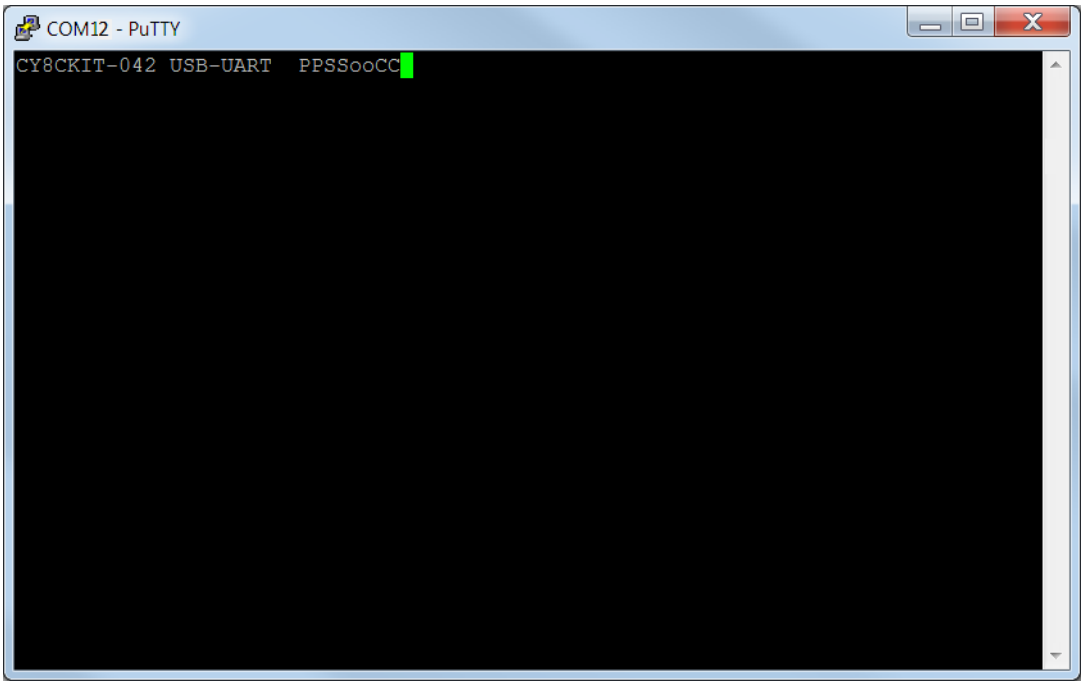


Figure 6-19. Data Displayed on PuTTY



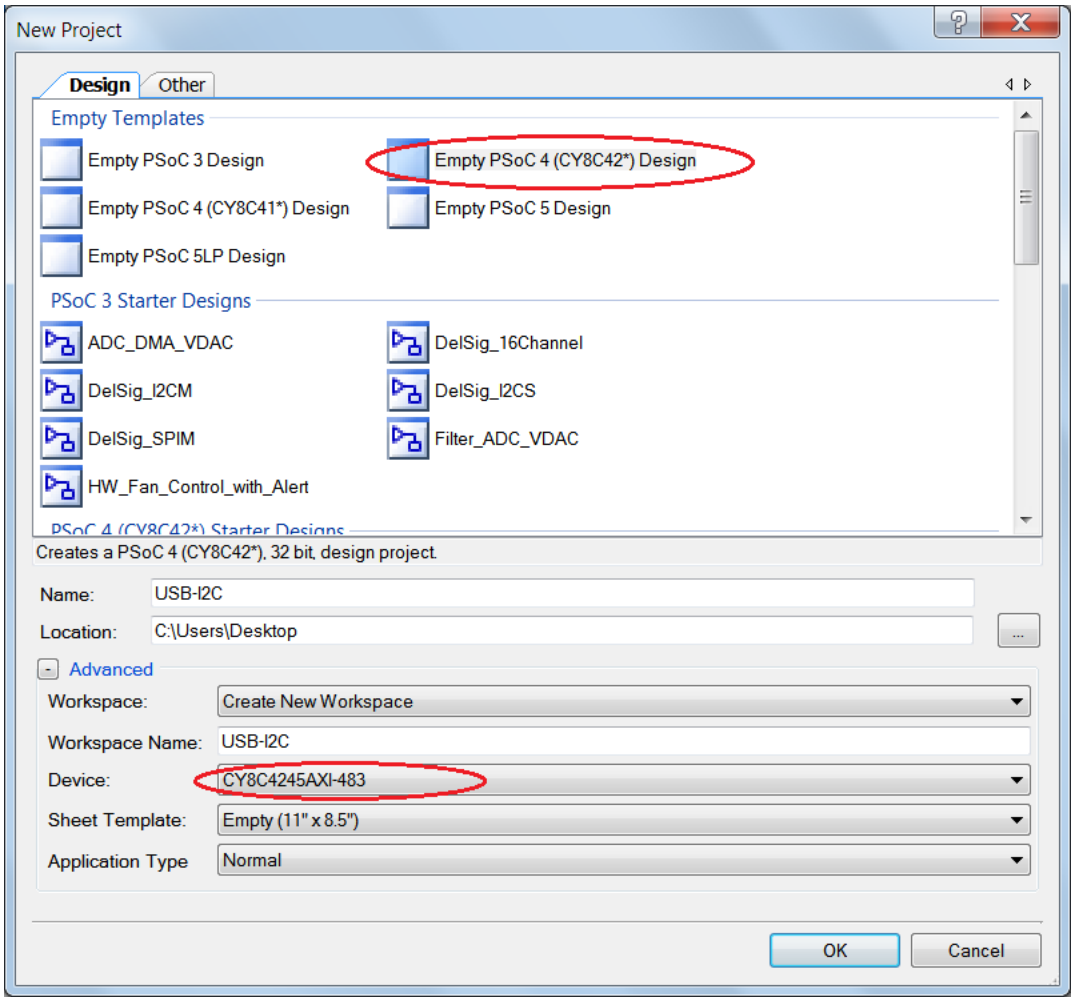
6.2 Using PSoC 5LP as USB-I2C Bridge

The PSoC 5LP serves as a USB-I2C Bridge, which can be used to communicate with the USB-I2C software running on the PC. This project is available with other example projects for the PSoC 4 Pioneer kit at the element14 partner web page, 100 Projects in 100 days.

The following is a detailed explanation of using the USB-I2C Bridge, which can communicate between the BCP and the PSoC 4.

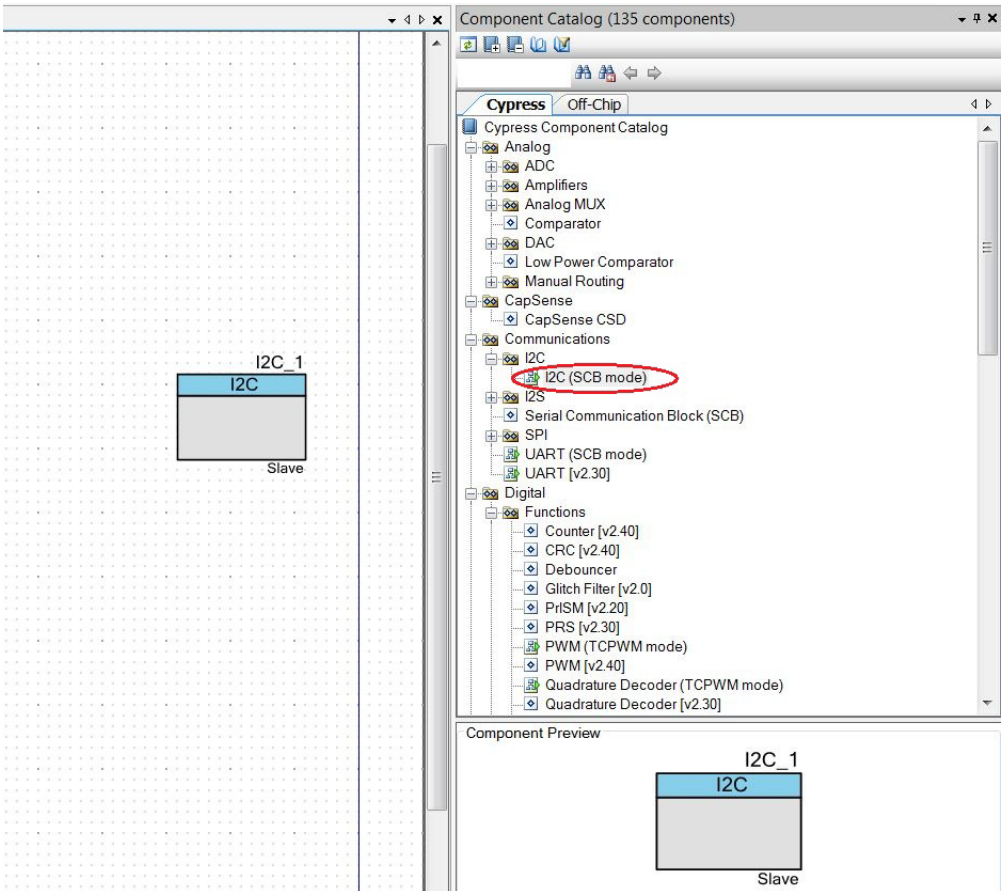
1. Open a new project targeting the PSoC 4 device in the PSoC Creator.

Figure 6-20. Opening a New Project in PSoC Creator



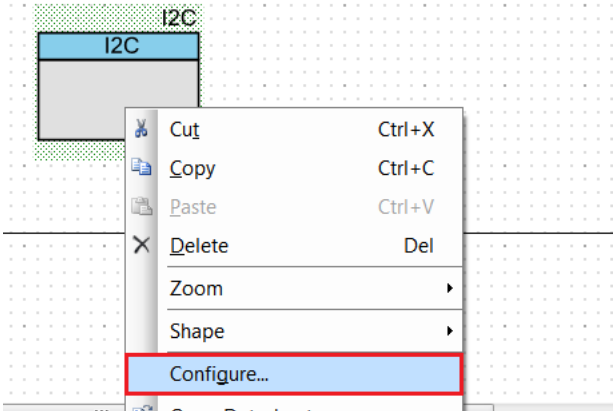
2. Drag and drop an I2C component to the TopDesign.

Figure 6-21. I2C Component in Component Catalog



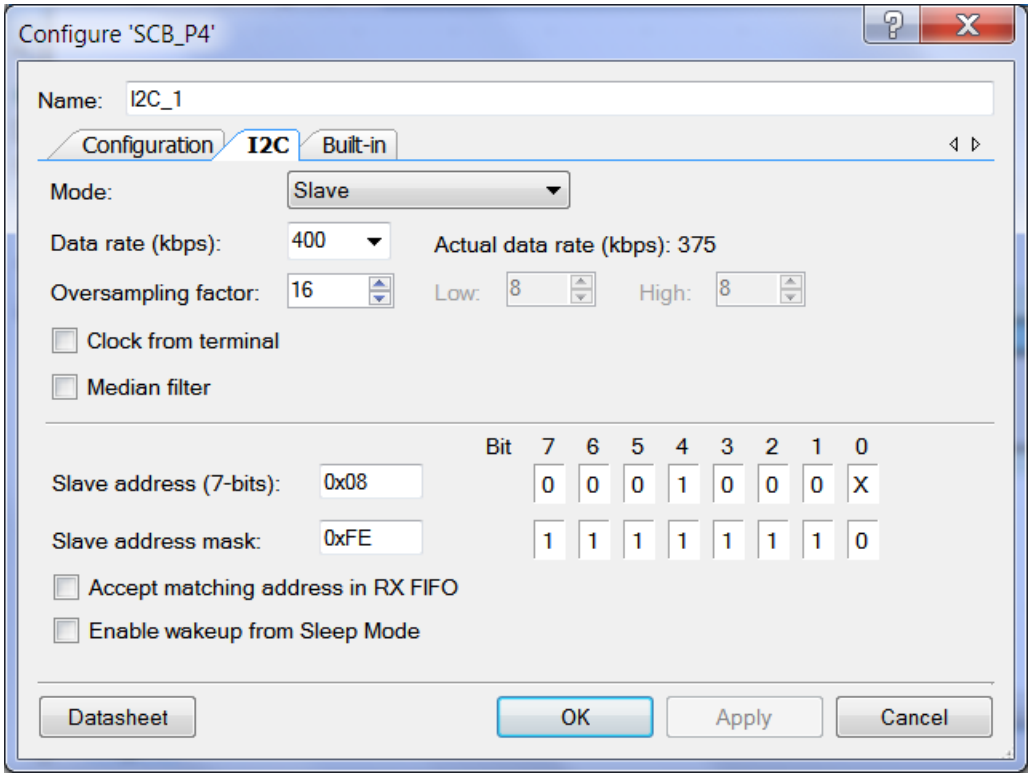
3. To configure the I2C component, double-click or right-click on the I2C component and select **Configure**.

Figure 6-22. Open the I2C Configuration Window



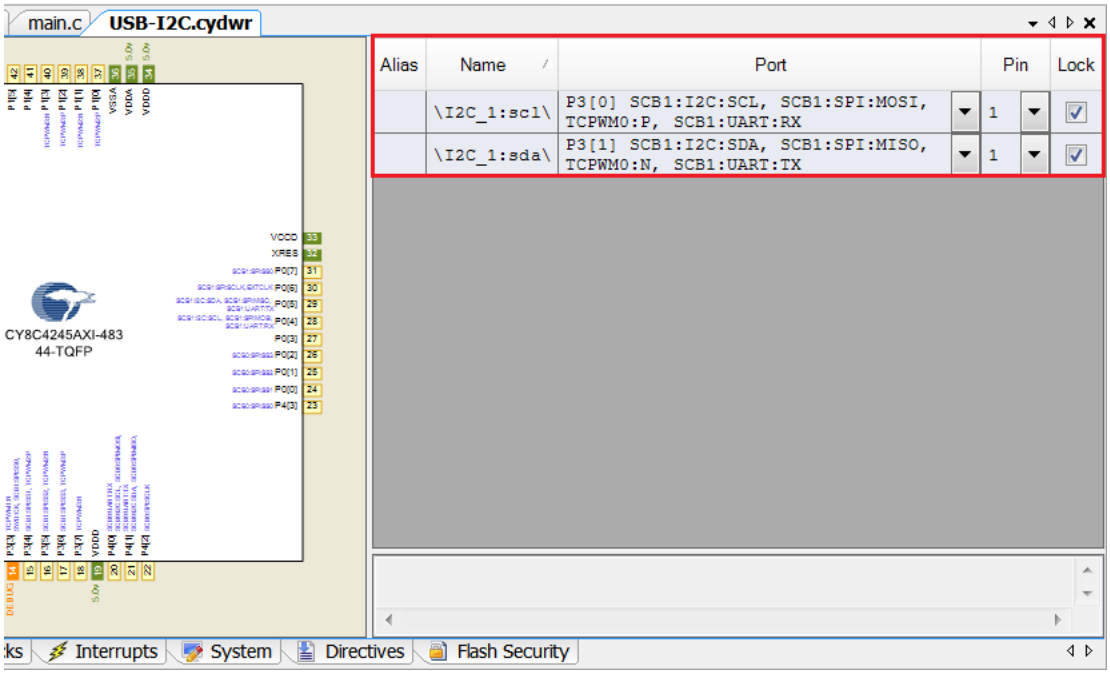
4. Configure the I2C with the following settings.

Figure 6-23. I2C Configuration Window



5. Select pin P3[0] for the I2C SCL and pin P3[1] for the I2C SDA in the **Pins** tab of <poject.cydwr>.

Figure 6-24. Pin Selection



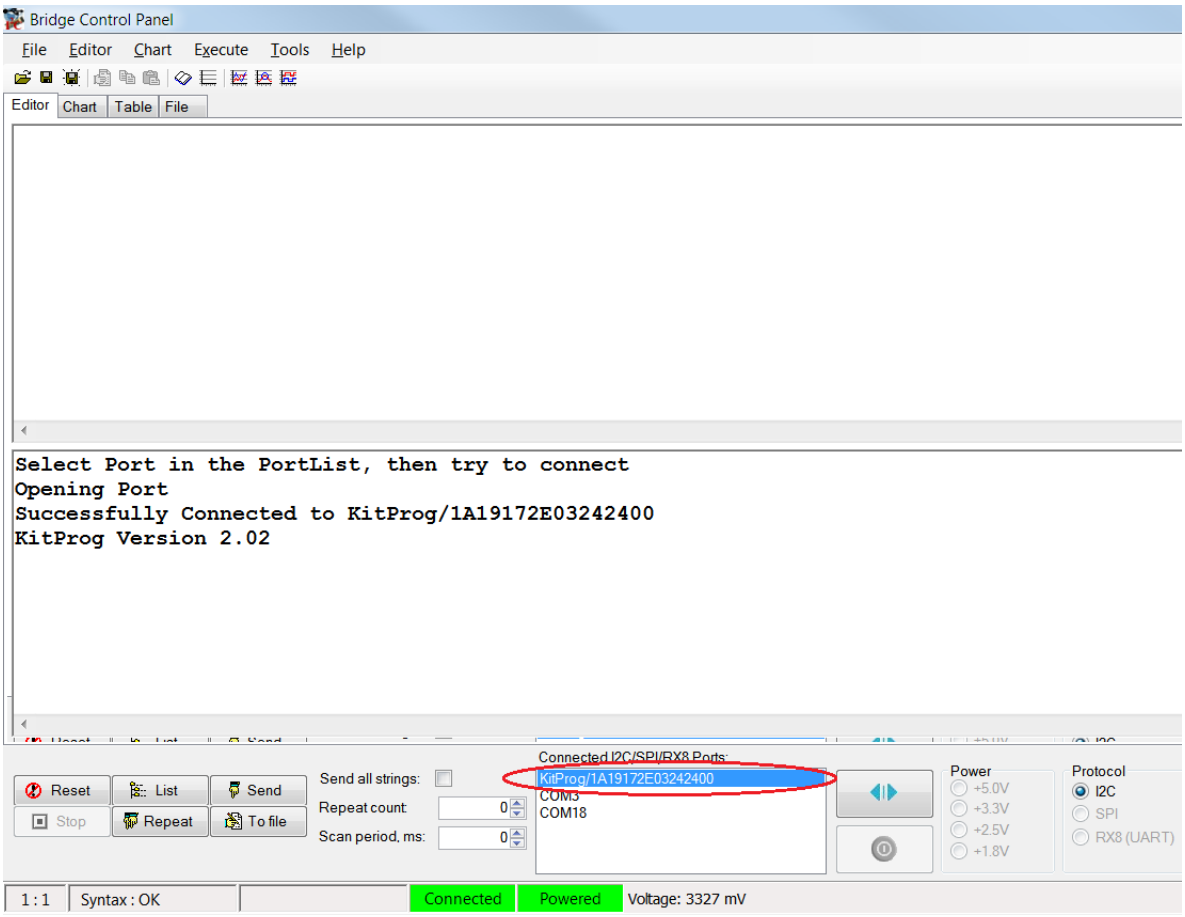
6. Place the following code in your main.c project file. The code will enable the PSoC 4 device to transmit and receive I2C data to and from the BCP application.

```
#include <device.h>

void main()
{

uint8 wrBuf[10];  /* I2C write buffer */
uint8 rdBuf[10];  /* I2C read buffer */
uint8 indexCntr;
uint32 byteCnt;

/* Enable the Global Interrupt */
CyGlobalIntEnable;

/* Start I2C Slave operation */
I2C_Start();

/* Initialize write buffer */
I2C_I2CSlaveInitWriteBuf((uint8 *) wrBuf, 10);

/* Initialize read buffer */
I2C_I2CSlaveInitReadBuf((uint8 *) rdBuf, 10);

for(;;) /* Loop forever */
{

 /* Wait for I2C master to complete a write */
 if(0u != (I2C_I2CSlaveStatus() & I2C_I2C_SSTAT_WR_CMPLT))
 {

       /* Read the number of bytes transferred */
       byteCnt = I2C_I2CSlaveGetWriteBufSize();

       /* Clear the write status bits*/
       I2C_I2CSlaveClearWriteStatus();

/* Move the data written by the master to the read buffer so that the
       master can read back the data */
       for(indexCntr = 0; indexCntr < byteCnt; indexCntr++)
       {
rdBuf [indexCntr] = wrBuf[indexCntr]; /* Loop back the data to the read
             buffer */
       }
```

```
/* Clear the write buffer pointer so that the next write operation will
       start from index 0 */
       I2C_I2CSlaveClearWriteBuf();

/* Clear the read buffer pointer so that the next read operations starts
       from index 0 */
       I2C_I2CSlaveClearReadBuf();
 }
/* If the master has read the data , reset the read buffer pointer to 0
 and clear the read status */
 if(0u != (I2C_I2CSlaveStatus() & I2C_I2C_SSTAT_RD_CMPLT))
 {
/* Clear the read buffer pointer so that the next read operations starts
 from index 0 */
       I2C_I2CSlaveClearReadBuf();

       /* Clear the read status bits */
       I2C_I2CSlaveClearReadStatus();
 }
}

}
```

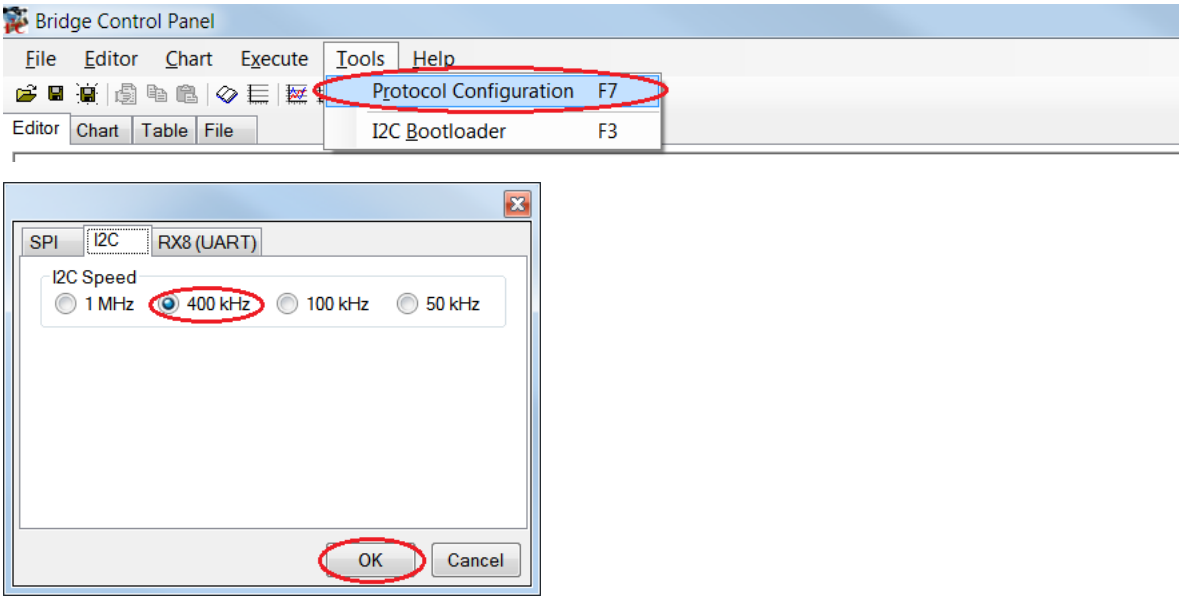
6. Build the project by clicking **Build > Build Project** or [Shift]+[F6]. After the project is built without errors and warnings, program ([Ctrl]+[F5]) this code on to the PSoC 4 through the PSoC 5LP programmer or MiniProg3.
7. Open the BCP from **Start > All Programs > Cypress > Bridge Control Panel <version number>**
8. Connect to **KitProg/** under **Connected I2C/SPI/RX8 Ports**.

Figure 6-25. Connecting to KitProg/ in BCP



9. Open the **Protocol configuration** from **Tools** menu and select the appropriate **I2C speed**. Make sure I2C speed is same as the one configured in I2C component. Click **OK** to close the window.

Figure 6-26. Opening Protocol Configuration Window in BCP



10. From the BCP transfer five bytes of data to the I2C device with slave address 0x08. Log shows whether the transaction was successful or a failure. A '+'indication after each byte indicates that the transaction was successful and a '-' indicates that the transaction was a failure.

Figure 6-27. Entering Commands in BCP

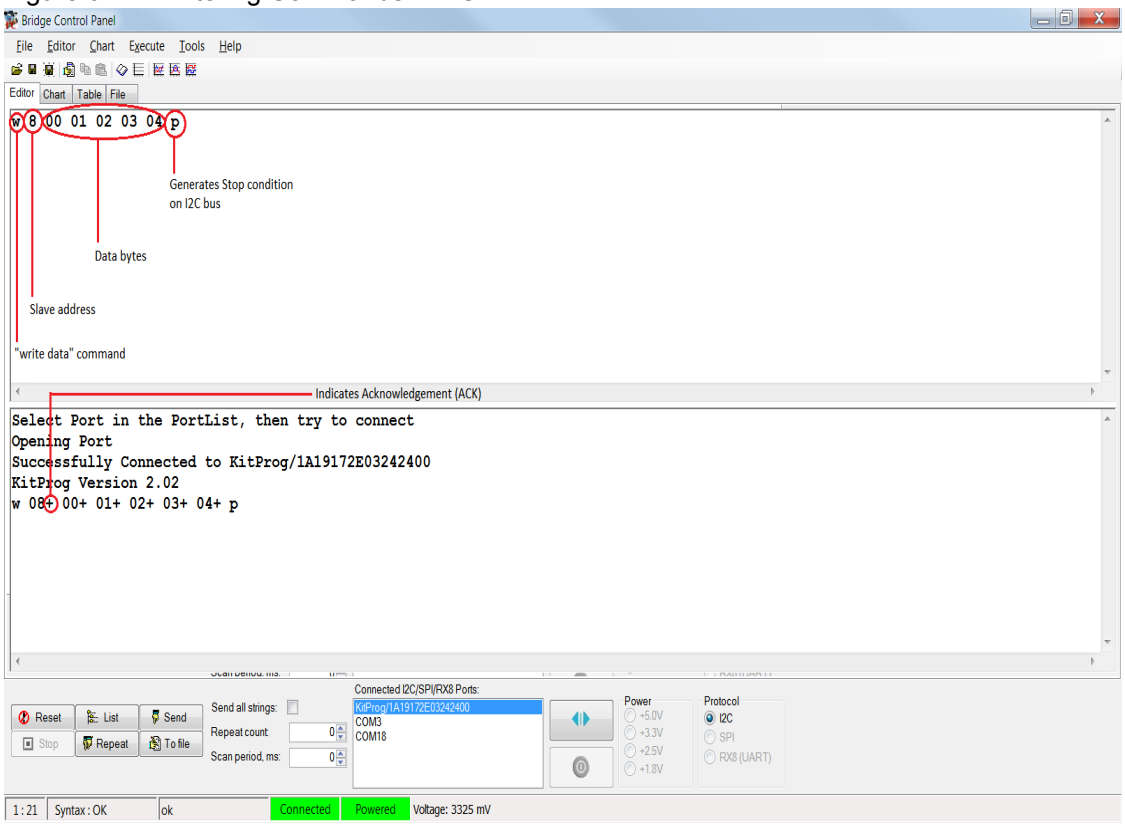
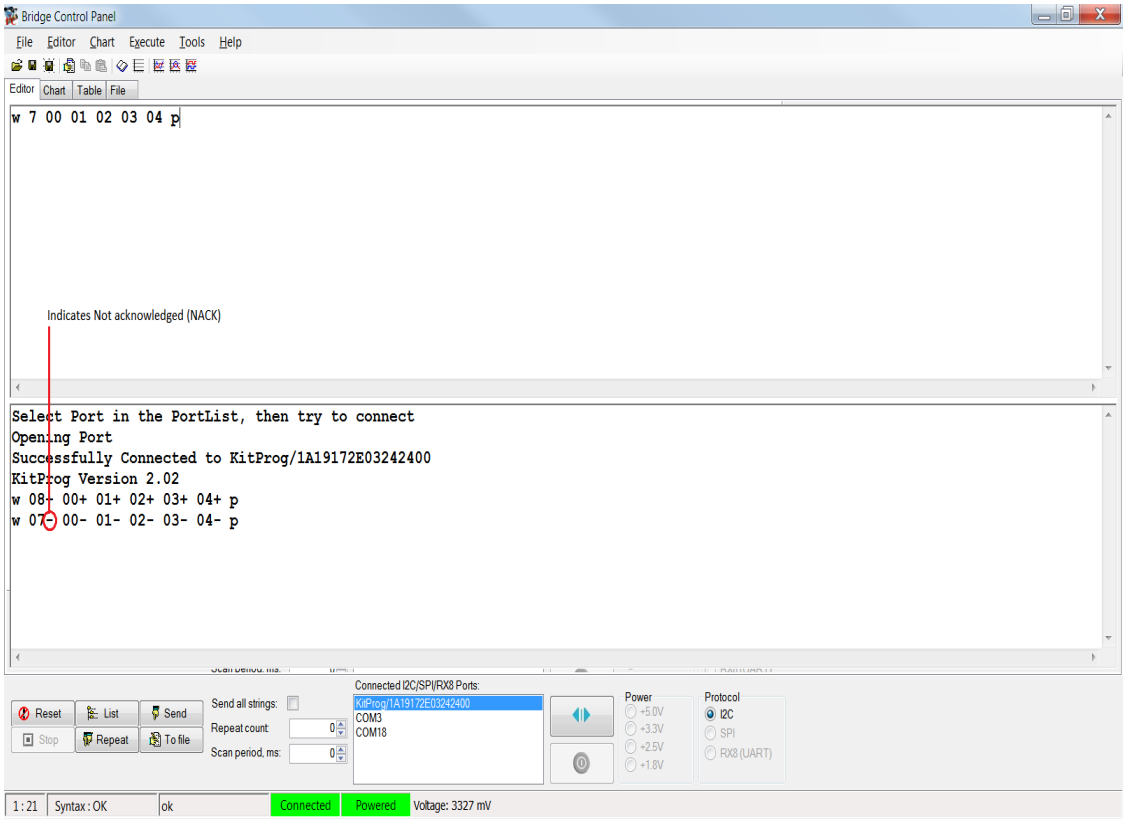
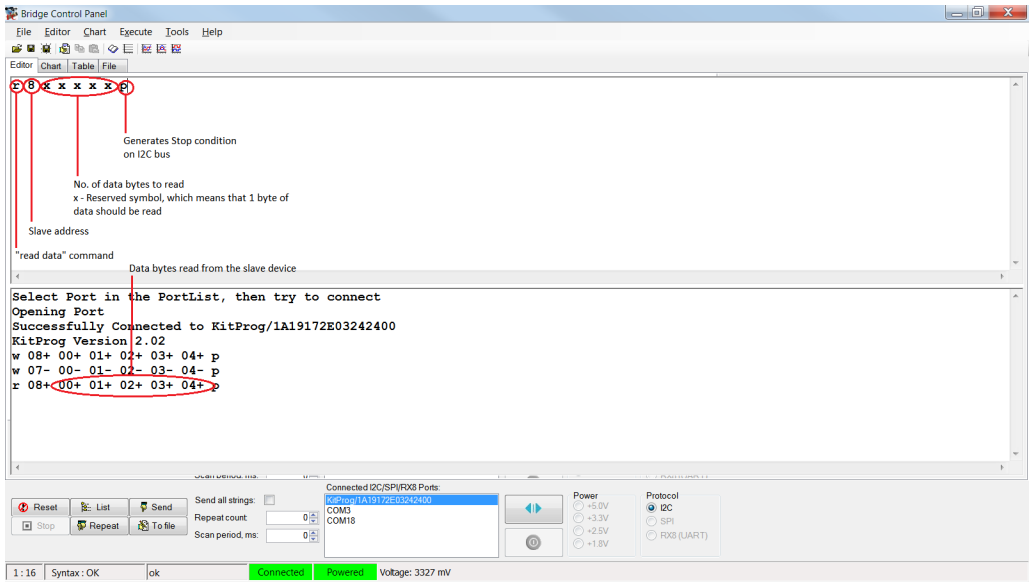


Figure 6-28. Indication of NACK in BCP



11. From the BCP read five bytes of data from the I2C slave device with slave address 0x08. The Log shows whether the transaction was successful or a failure.

Figure 6-29. Read Data Bytes from the BCP



Note: Refer **Help Contents** or [F1] under **Help** in BCP for details of I2C commands.

6.3 Developing Applications for the PSoC 5LP

The PSoC 4 Pioneer kit has an onboard PSoC 5LP whose primary function is that of a programmer and a bridge. In addition, the user can also build standalone applications for the PSoC 5LP. The user can build either a normal project or a bootloadable project using the PSoC 5LP.

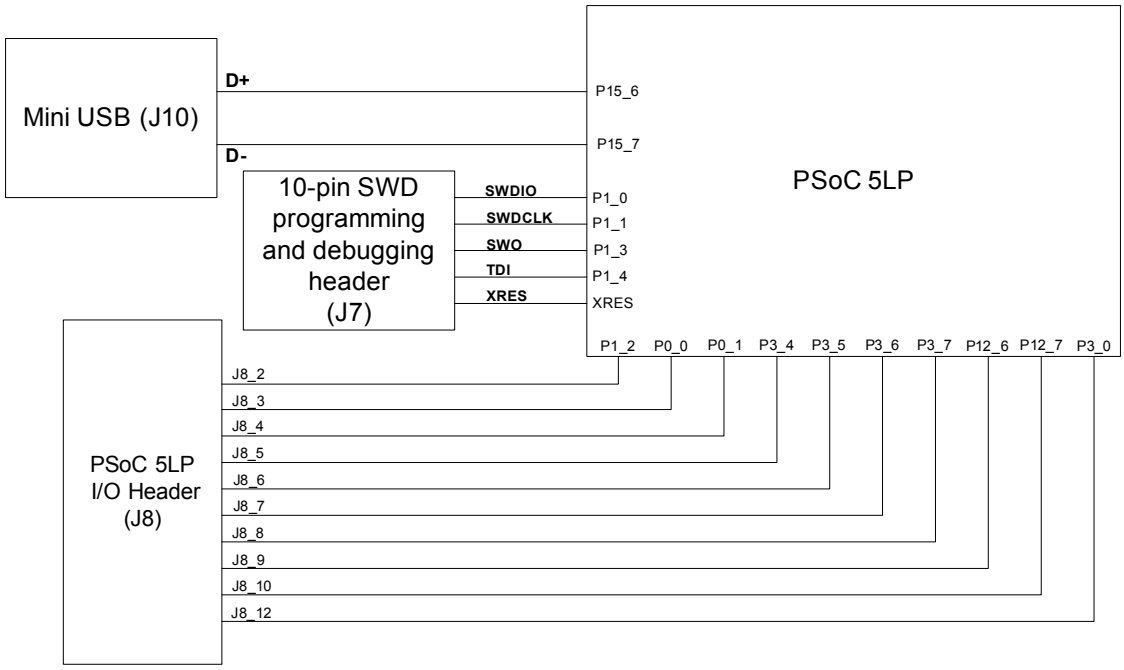
The PSoC 5LP connections in the Pioneer board are summarized in the Figure 6-30 on page 77.

J8 is the I/O connector (refer to section 4.3.7 PSoC 5LP GPIO Header (J8)).

The USB (J10) is connected and is used as the PC interface. But the user can still use this USB connection to create customized USB designs.

The programming header (J7) is meant for standalone programming. This header needs to be populated. Refer to section 'No Load Components' in A.6 Bill of Materials (BOM) on page 100.

Figure 6-30. Block Diagram of PSoc 5LP



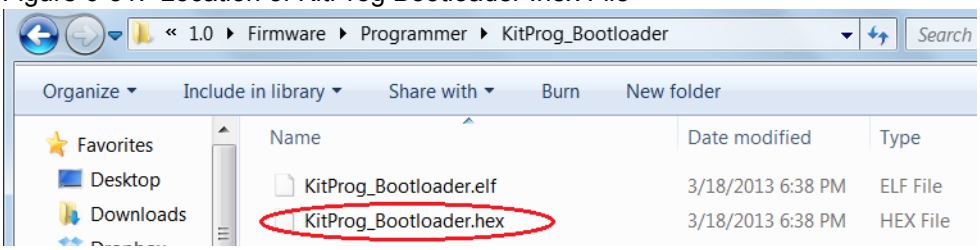
6.3.1 Building a Bootloadable project for the PSoC 5LP

All bootloadable applications developed for the PSoC 5LP should be based on the Bootloader hex file which is programmed onto the kit. The Bootloader hex file is available in the kit files or can be downloaded from the kit web page.

The .hex files are included in the following kit installer directory location:

```
<Install Path>\CY8CKIT-042 PSoC 4 Pioneer Kit\
<version>\Firmware\Programmer\KitProg_Bootloader
```

Figure 6-31. Location of KitProg Bootloader .hex File



To build any bootloadable application for the PSoC 5LP follow this procedure:

1. Open PSoC Creator select **New > Project > PSoC 5LP**, click expand button adjacent to **Advanced** and select the **Device** as **CY8C5868LTI-LP039** as shown in Figure 6-33 on page 78, **Application type** as **Bootloadable** from drop-down list.

Figure 6-32. Opening New Project in PSoC Creator

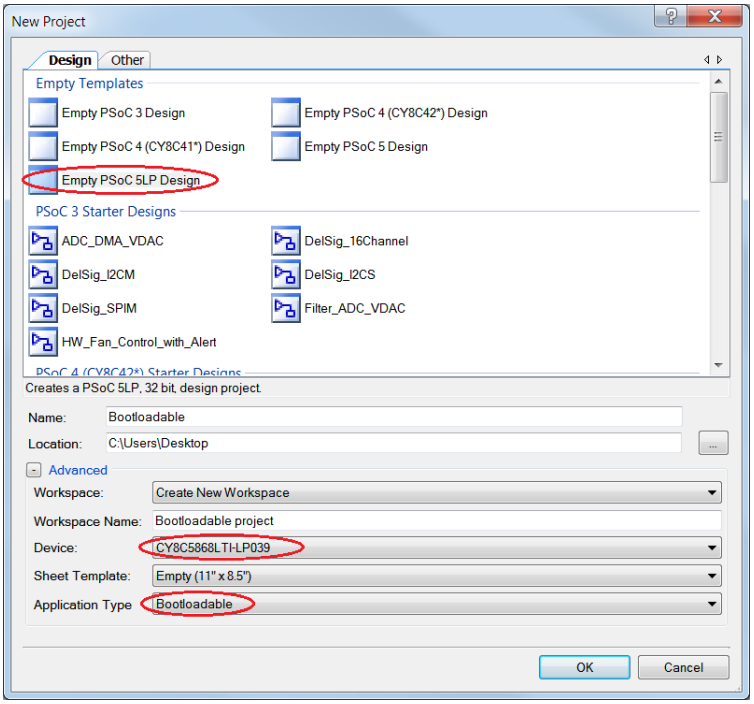
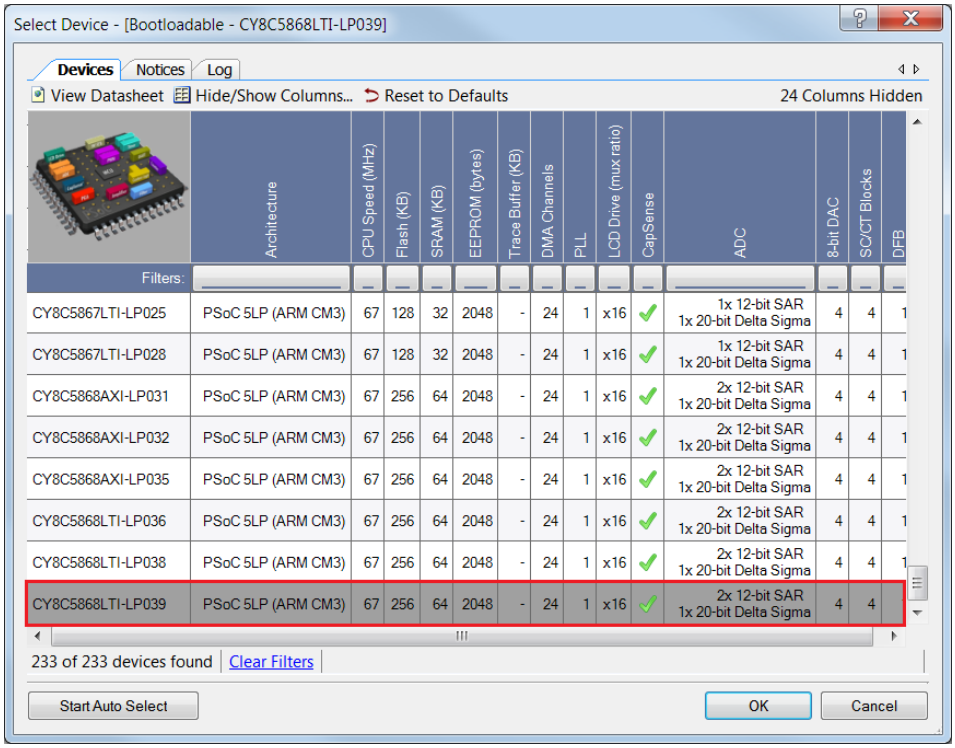
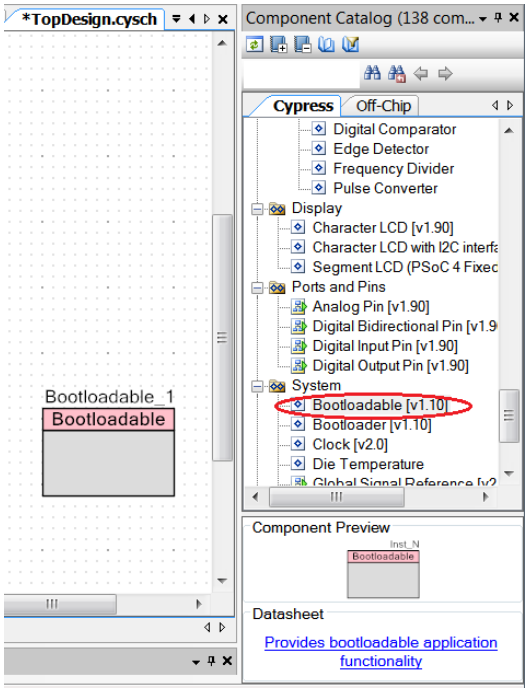


Figure 6-33. Selecting Device in PSoC Creator



2. Navigate to the Schematic view and drag and drop a Bootloadable component on the top design.

Figure 6-34. Bootloadable Component in Component Catalog



Set the dependency of the Bootloadable component by selecting the **Dependencies** tab in the configuration window and clicking the **Browse** button. Select the KitProg_Bootloader.hex file and click Open.

Figure 6-35. Configuration Window of Bootloadable Component

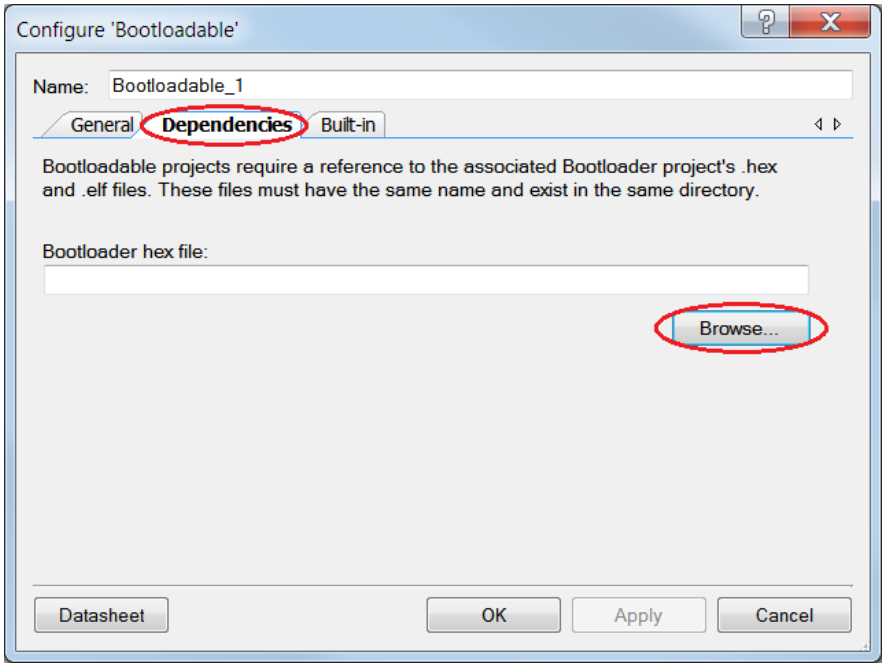
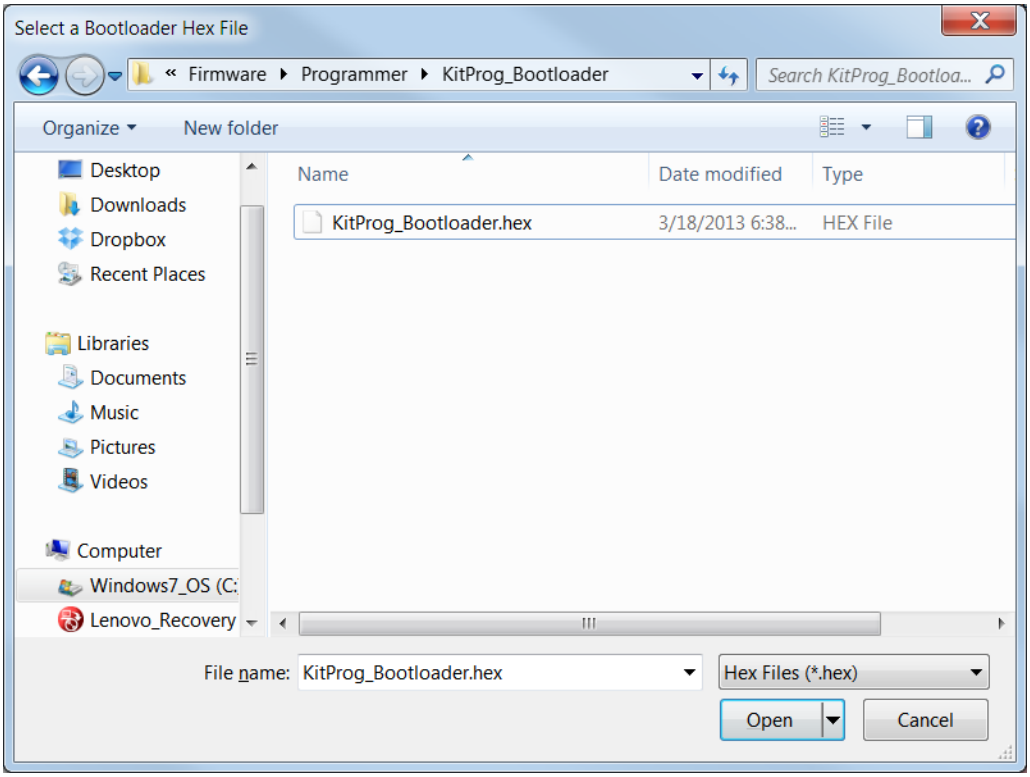
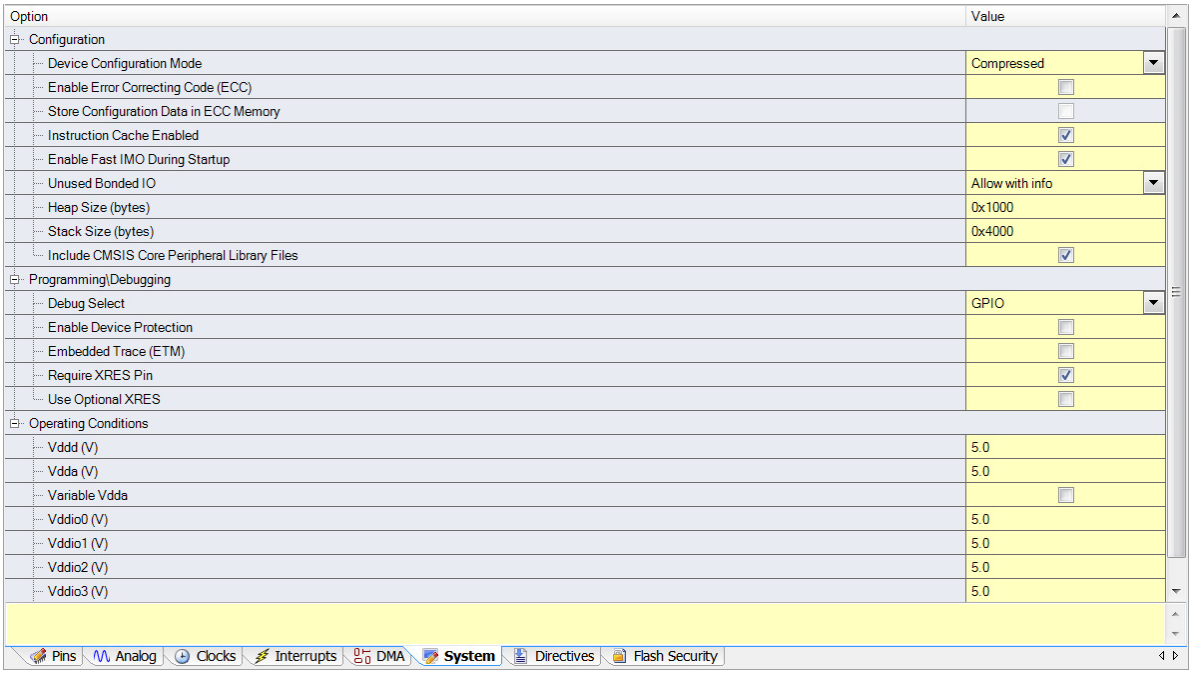


Figure 6-36. Selecting KitProg Bootloader hex File



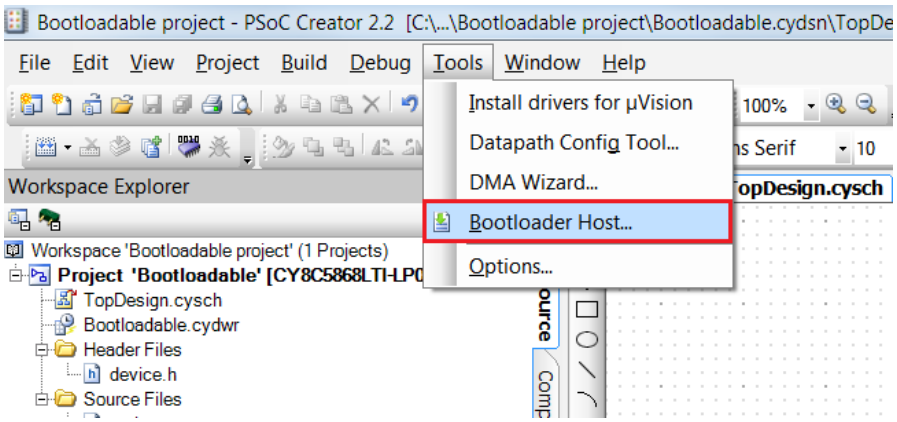
3. Develop your custom project.
4. The NVL setting of Bootloadable project and the KitProg_Bootloader project must be the same. The KitProg_Bootloader.cydwr system settings is shown in the below figure.

Figure 6-37. KitProg Bootloader System Settings



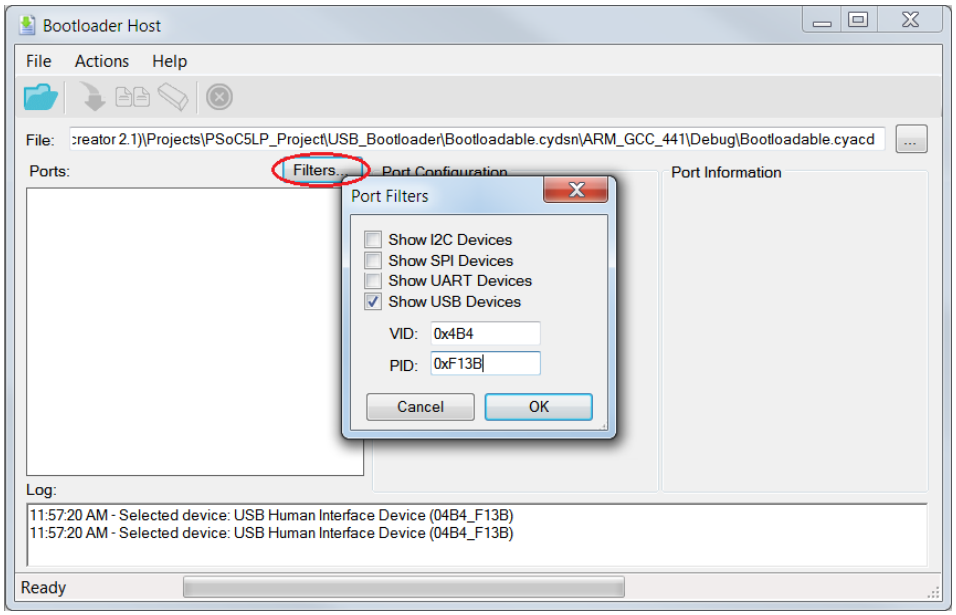
5. Build the project in PSoC Creator by selecting **Build > Build Project** or [Shift]+[F6].
6. To download the project on to the PSoC 5LP device, open Bootloader Host Tool which is available from **PSoC Creator select > Tools > Bootloader Host**.

Figure 6-38. Opening Bootloader Host Tool from PSoC Creator



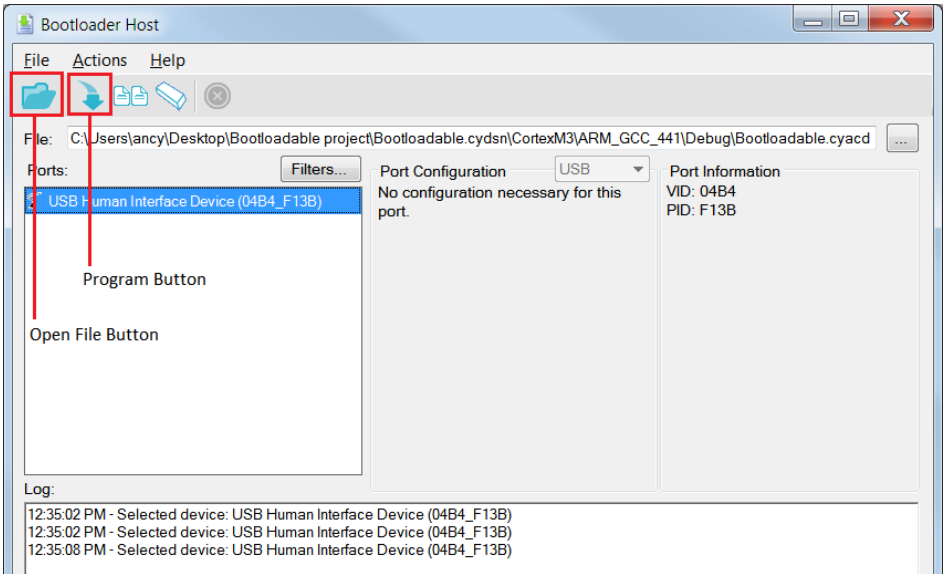
7. In the Bootloader Host tool click **Filters** and add a filter to identify the USB device. Set VID as 0x04B4, PID as 0xF13B and click **OK**.

Figure 6-39. Port Filters Tab in Bootloader Host Tool



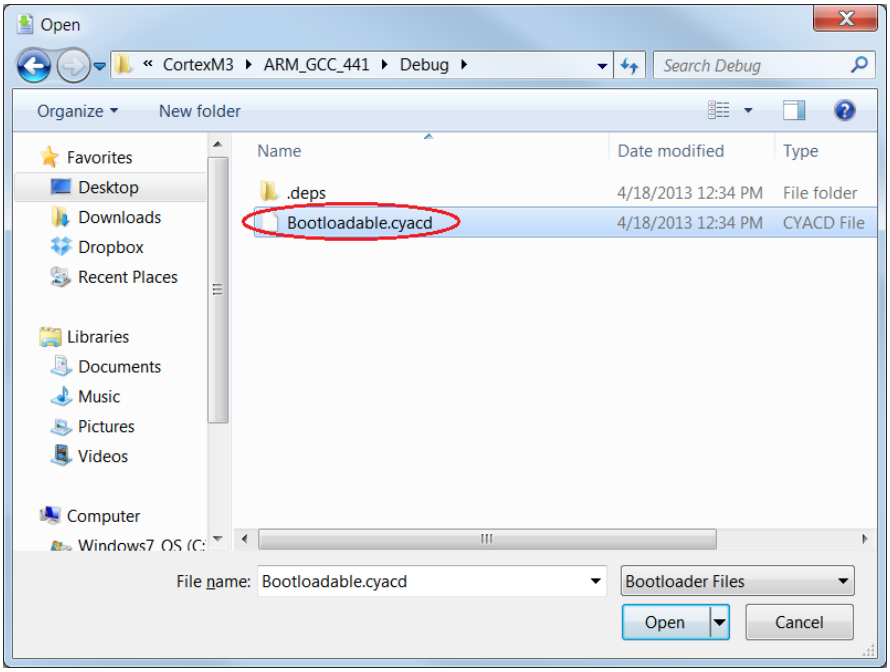
8. In the Bootloader Host tool, click **Open File** button to browse to the location of the bootloadable file(*.cyacd).

Figure 6-40. Opening Bootloadable File from Bootloader Host Tool



9. Now keep the reset switch (SW1) pressed and plug in the USB Mini-B connector. If the switch is pressed for more than 100 ms, the PSoC 5LP enters into bootloader. Now press the Program button in the Bootloader Host tool to program the device. The PSoC 5LP also enters into bootloader when the power supply jumper for the PSoC 4 (J13) is removed and subsequently the USB Mini-B connector is plugged into J10 header.

Figure 6-41. Selecting Bootloadable .cyacd File from Bootloader Host tool



10. If bootload is successful the log of the tool displays "Successful" or else "Failed" and a statement for the failure.

Notes:

1. Pins of the PSoC 5LP are brought to the PSoC 5LP GPIO header (J8). These pins have been selected to support high-performance Analog and Digital projects. Refer to A.2 Pin Assignment Table on page 96 for pin information.
2. Care should be taken when allocating pins of the PSoC 5LP for custom application. For example P2 [0] - P2 [4] are dedicated for programming the PSoC 4. Refer to A.1 CY8CKIT-042 Schematics on page 93 before allocating the pins.
3. When a normal project is programmed onto the PSoC 5LP, the initial capability of the PSoC 5LP to act as a Programmer / USB-UART Bridge/USB-I2C Bridge in not available.
4. The status LED does not function unless used by the custom project.

For additional information on bootloaders, please refer to AN73503 - USB HID Bootloader for PSoC 3 and PSoC 5LP application note from Cypress to learn more about USB Bootloader applications.

6.3.2 Building a Normal Project for PSoC 5LP

A normal project means that the user will create a completely new project for the PSoC 5LP device on the CY8CKit-042. Here the user programs the entire flash of the PSoC 5LP overwriting all boot-loader and programming code. To recover the programmer the user has to reprogram the PSoC 5LP device with the factory settings KitProg.hex file shipped with the kit installer.

The factory KitProg.hex file is included in the installation directory of the kit installer:
`<Install Path>/CY8CKIT-042 PSoC 4 Pioneer Kit\<version>\Firmware\Programmer\KitProg`

This advanced functionality requires a MiniProg3 programmer which is not included with this kit.

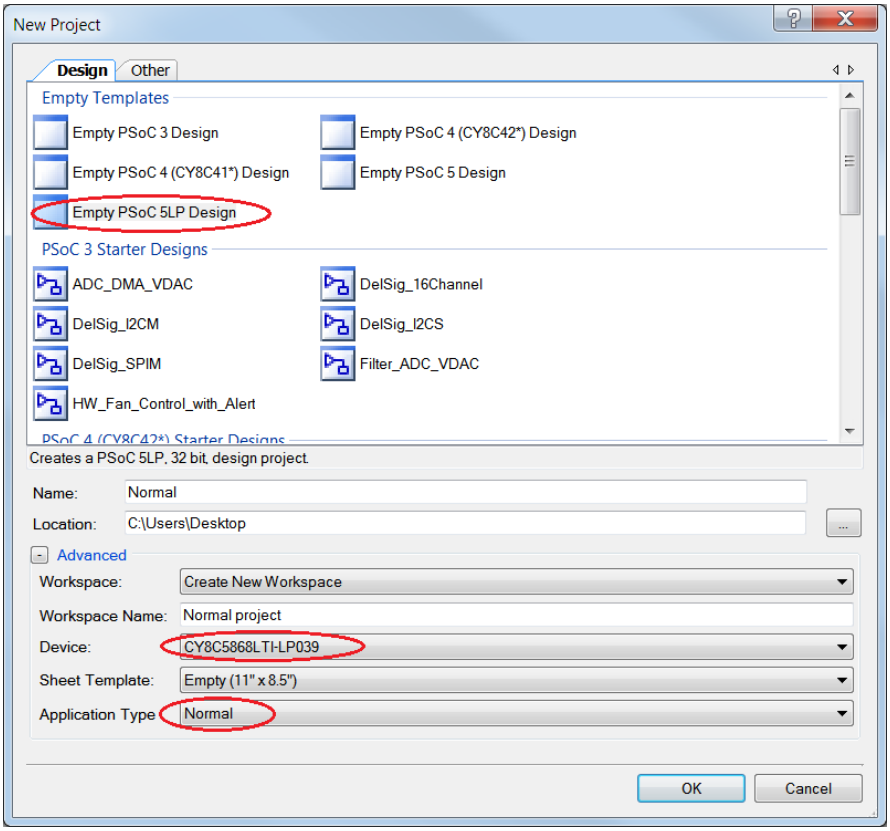
The MiniProg3 can be purchased at the following web page:

www.cypress.com/go/CY8CKit-002

To build a normal project for the PSoC 5LP follow the below procedure

1. Open PSoC Creator select **New > Project > PSoC 5LP**, click expand button adjacent to **Advanced** and select **Device** as **CY8C5868LTI-LP039**, **Application type** as **Normal** from drop-down list.

Figure 6-42. Opening New Project in PSoC Creator



2. Develop your custom project.
3. Build the project in PSoC Creator by selecting **Build > Build Project** or [Shift]+[F6].
4. Connect 10 pin connector of MiniProg3 to onboard 10-pin SWD debug and programming header J7 (which needs to be populated).

5. To program the PSoC 5LP with PSoC Creator, click **Debug > Program** or [Ctrl]+[F5]. The Programming window shows MiniProg3 and the selected device in the project (CY8C5868LTI-LP039) under it.
6. Click on the device and click **Connect** to program.

Notes:

1. The 10-pin SWD Debug and programming header (J7) is not populated. Refer to the Appendix section 'No Load Components' of A.6 Bill of Materials (BOM) for details.
2. Pins of the PSoC 5LP are brought to the PSoC 5LP GPIO header (J8). These pins have been selected to support high-performance analog and digital projects. Refer to the Appendix section A.2 Pin Assignment Table for pin information.
3. Care should be taken when allocating pins of the PSoC 5LP for custom application. For example P2 [0] - P2 [4] are dedicated for programming the PSoC 4. Refer to Appendix section A.1 CY8CKIT-042 Schematics before allocating the pins.
4. When a normal project is programmed onto the PSoC 5LP, the initial capability of the PSoC 5LP to act as a Programmer / USB-UART Bridge/USB-I2C Bridge in not available.
5. The status LED does not function unless used by the custom project.

6.4 PSoC 5LP Factory Program Restore Instructions

The PSoC 4 Pioneer Kit (CY8CKIT-042) features a PSoC 5LP device that comes factory-programmed as the onboard Programmer and Debugger for the PSoC 4 device.

In addition to creating applications for the PSoC 4 device, the user also has the ability to create custom applications for the PSoC 5LP device on this kit. The user can either create a Bootloadable or a Standard application for the PSoC 5LP device. For details, refer section 6.3 Developing Applications for the PSoC 5LP on page 77. Reprogramming or bootloading the PSoC 5LP device with a new flash image will overwrite the factory program and will forfeit the ability to use the PSoC 5LP device as a programmer/debugger for the PSoC 4 device. Follow the instructions to restore the factory program on the PSoC 5LP and enable the programmer/debugger functionality.

6.4.1 Restoring Factory Program on the PSoC 5LP

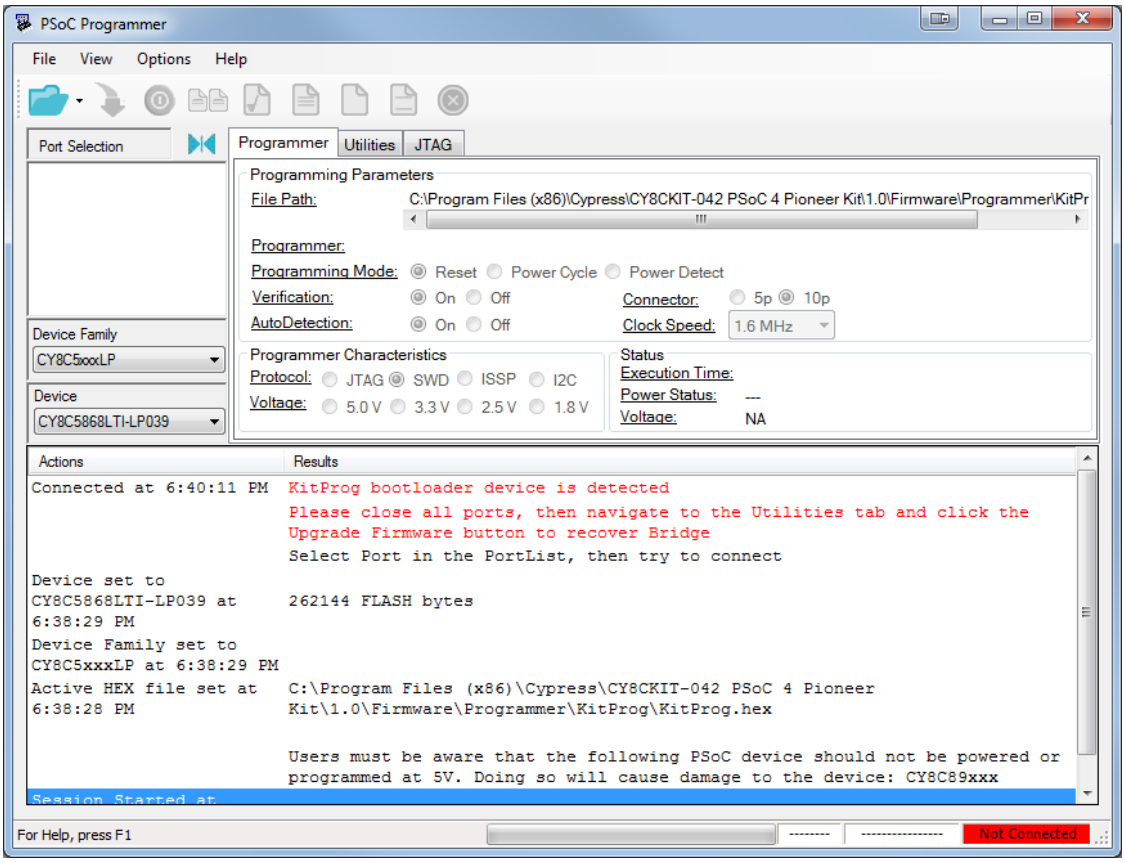
If the PSoC 5LP is programmed with a Bootloadable application, the user can restore the factory program by using one of the following two methods.

6.4.1.1 Restore PSoC 5LP factory program using PSoC Programmer

1. Launch **PSoC Programmer 3.18** or later from within the **Cypress > PSoC Programmer** start menu.
2. Configure the Pioneer Kit in Service Mode - While holding down the reset button (“SW1 Reset”), plug in the PSoC 4 Pioneer Kit to the computer using the included USB cable (USB A to USB Mini B). Doing this puts the PSoC 5LP into service mode which is indicated by the blinking green status LED.

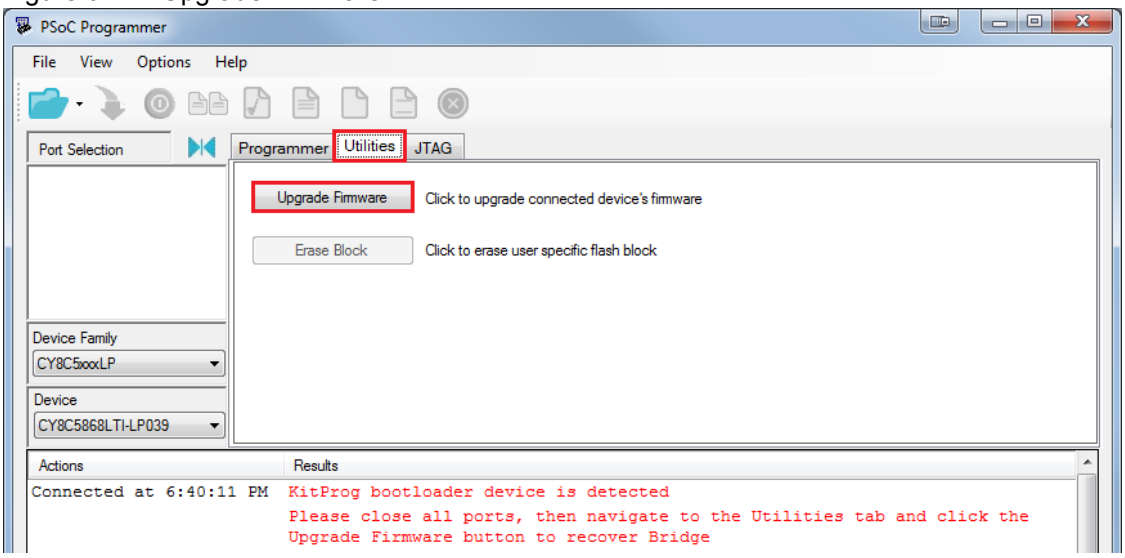
3. The following message appears in the PSoC Programmer results window **"KitProg Bootloader device is detected"**.

Figure 6-43. PSoC Programmer Results Window



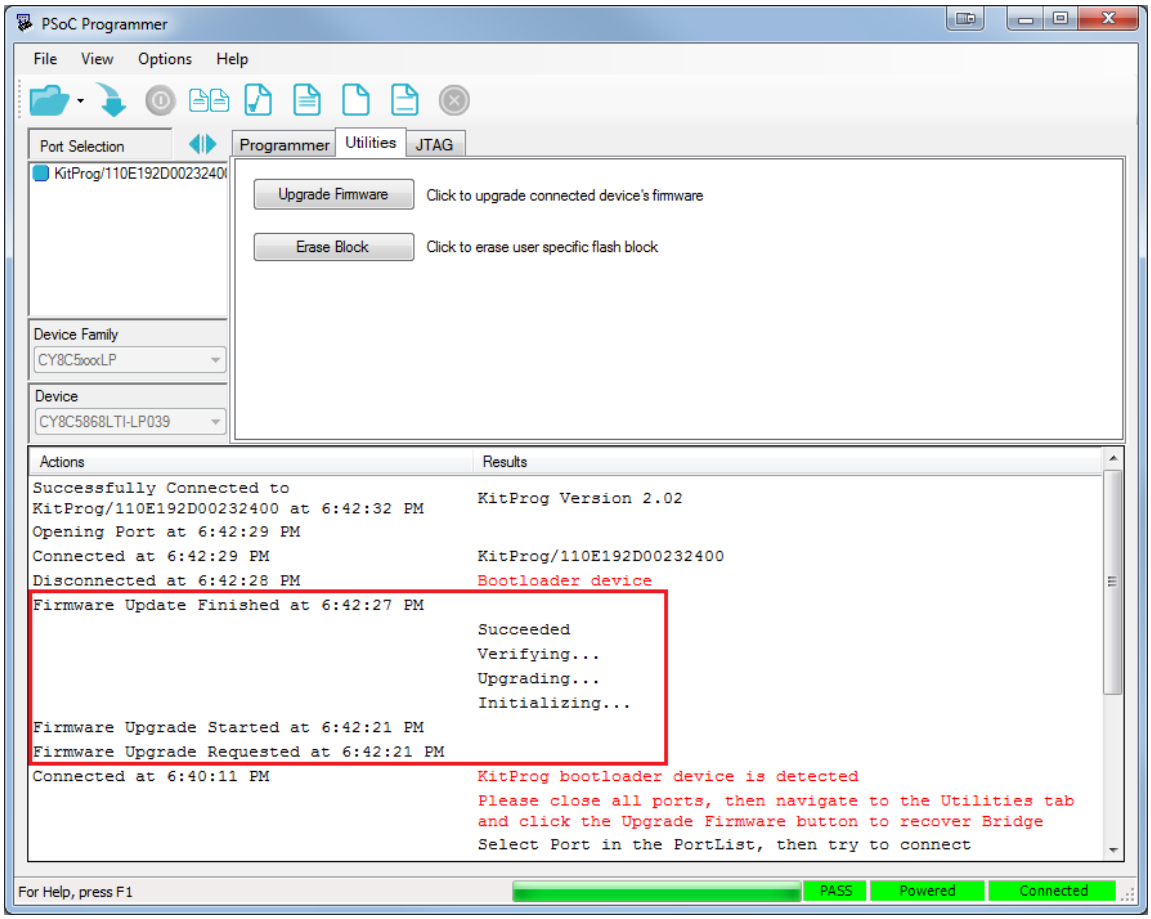
4. Switch to the **Utilities** tab in PSoC Programmer and press the Upgrade Firmware button. The user needs to unplug all other PSoC programmers (MiniProg3, DVKProg and so on.) from the PC before pressing the Upgrade Firmware button.

Figure 6-44. Upgrade Firmware



5. After programming has completed, the user will see the following message “Firmware Update Finished at <time>”.

Figure 6-45. Firmware Update Finished

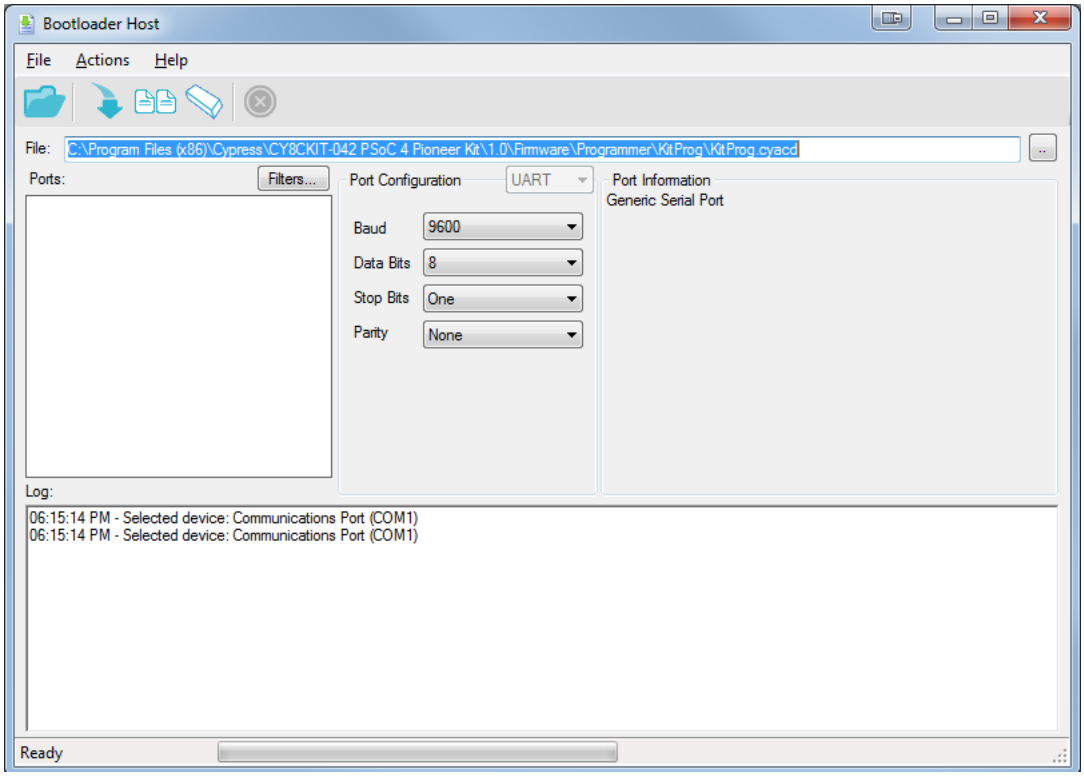


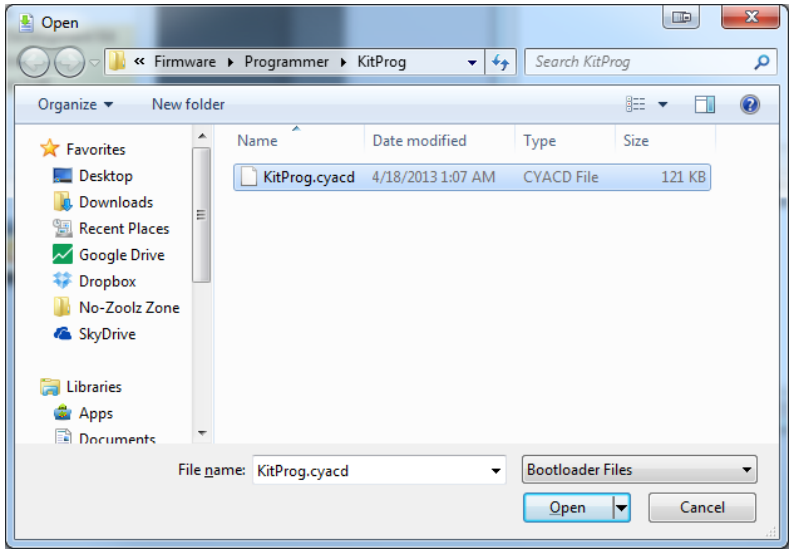
6. The user has now successfully restored the factory program on the PSoC 5LP. It can now be used as the Programmer/Debugger for the PSoC 4 device on the Pioneer Kit.

6.4.1.2 *Restore PSoC 5LP factory program using the USB Host Tool*

1. Launch the Bootloader Host tool from within the **Cypress > PSoC Creator** start menu.
2. After the Bootloader Host tool has launched, using the **File > Open** menu, load the “Kit Prog.cyacd” file that is installed with the kit software. The default location for this file is: `<Install Path>\CY8CKIT-042 PSoC 4 Pioneer Kit\<version>\Firmware\Programmer\KitProg\KitProg.cyacd`

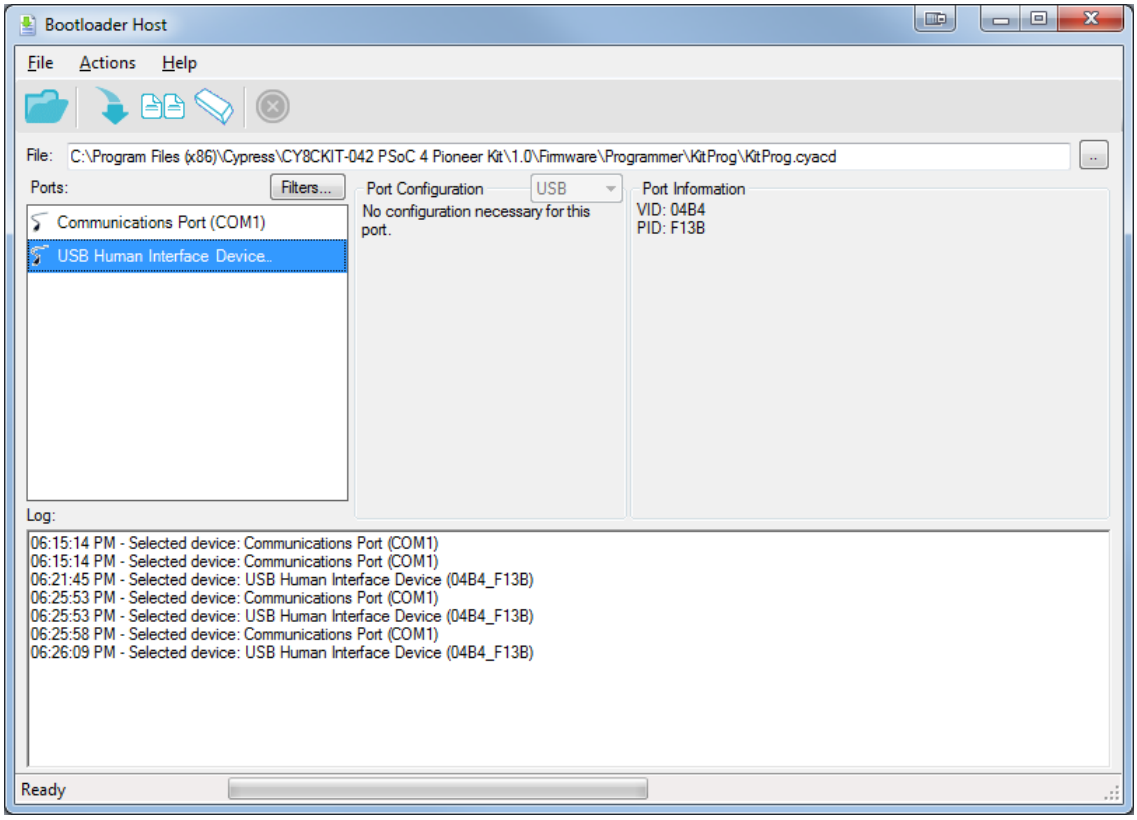
Figure 6-46. Load KitProg.cyacd File





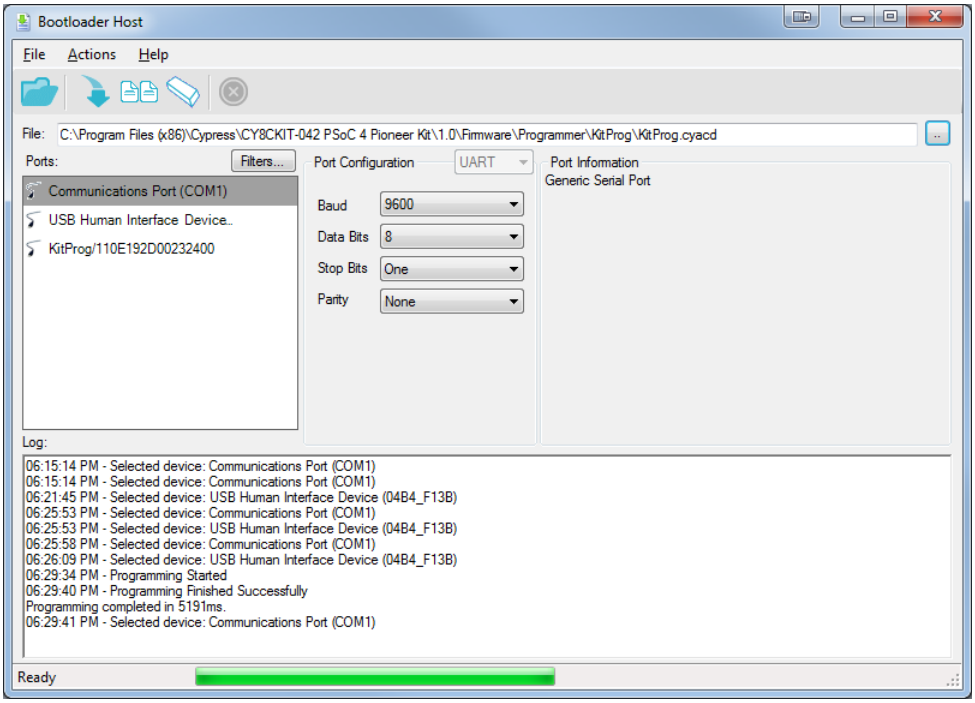
3. Configure the Pioneer Kit in Service Mode - While holding down the reset button (“SW1 Reset”), plug in the PSoC 4 Pioneer Kit to the computer using the included USB cable (USB A to USB Mini B). Doing this puts the PSoC 5LP into service mode, which is indicated by the blinking green status LED.
4. In the Bootloader Host tool, set the Filters for the USB devices with VID: 04B4 and PID: F13B. “USB Human Interface Device” port appears in the Ports list. Choose that port by clicking on it.

Figure 6-47. Select USB Human Interface Device



5. The user is now ready to restore the factory-program, by Bootloading it onto the PSoC 5LP on the Pioneer Kit. To do this, press the Program button (or menu item **Actions > Program**).
6. After programming has completed, the user will see the following message “Programming Finished Successfully”.

Figure 6-48. Programming Finished Successfully



7. The user has now successfully restored the factory program on the PSoC 5LP. It can now be used as the Programmer/Debugger for the PSoC 4 device on the Pioneer Kit.

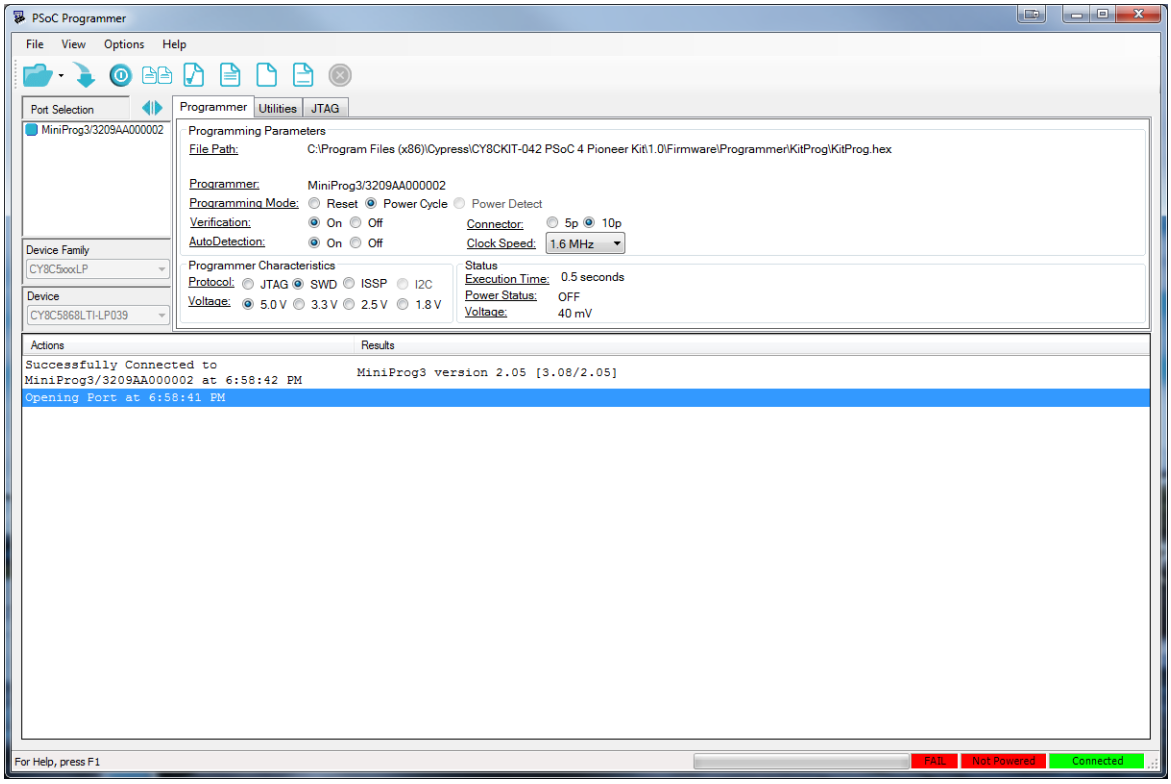
6.4.2 PSoC 5LP is programmed with a Standard Application

If PSoC 5LP is programmed with a Standard Application, the user can restore the factory program by using the following method.

1. Launch **PSoC Programmer 3.18** from within the **Cypress > PSoC Programmer** start menu.
2. Once PSoC Programmer has launched, use the **File > Open** menu to load the “KitProg.hex” factory program HEX file that is shipped with the kit. The default location for this file is: `<Install Path>\CY8CKIT-042 PSoC 4 Pioneer Kit\<version>\Firmware\Programmer\KitProg`
3. Connect a MiniProg3 (CY8CKIT-002 – not included, sold separately) to the computer. The 10-pin connector cable on the MiniProg3 plugs into the header [J7]. Note that the J7 header is unpopulated. For more details, refer to the Appendix section A.6 Bill of Materials (BOM) on page 100.

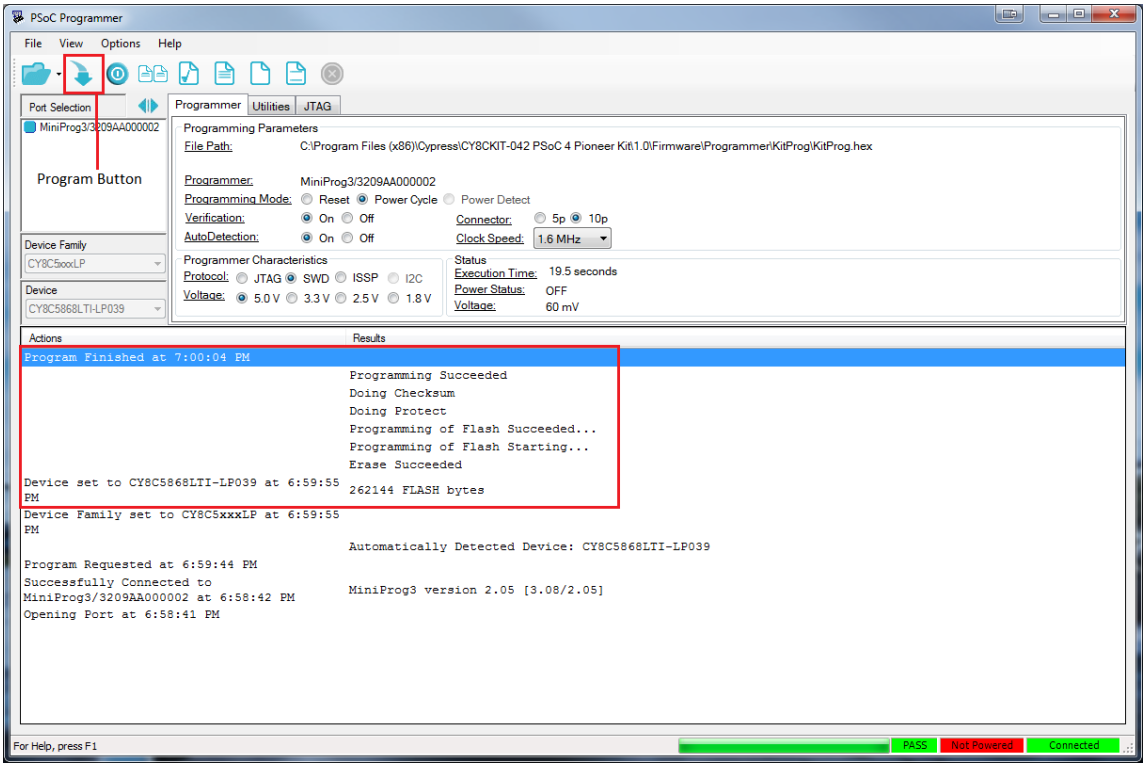
4. Ensure that the MiniProg3 is the selected port in PSoC Programmer, and the 10-pin Connector option is selected as shown in the image below. If the board isn't powered over USB, the user will also need to select "Power Cycle" as the Programming Mode.

Figure 6-49. Select MiniProg3



5. When ready, press the Program button (or menu item **File > Program**) to program the PSoC 5LP device.
6. After programming has completed, the user will see the following message **"Program Finished at <time>"**.

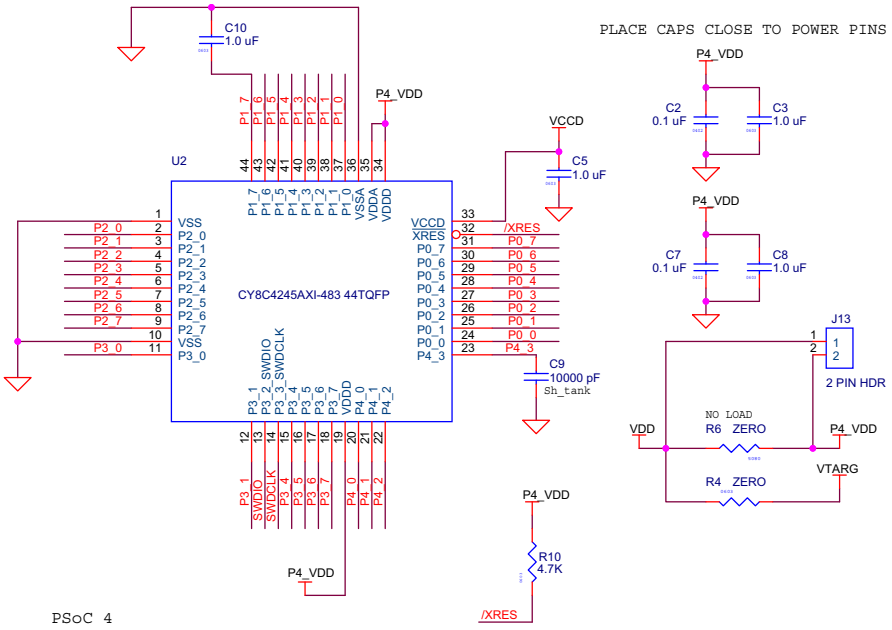
Figure 6-50. Program Finished

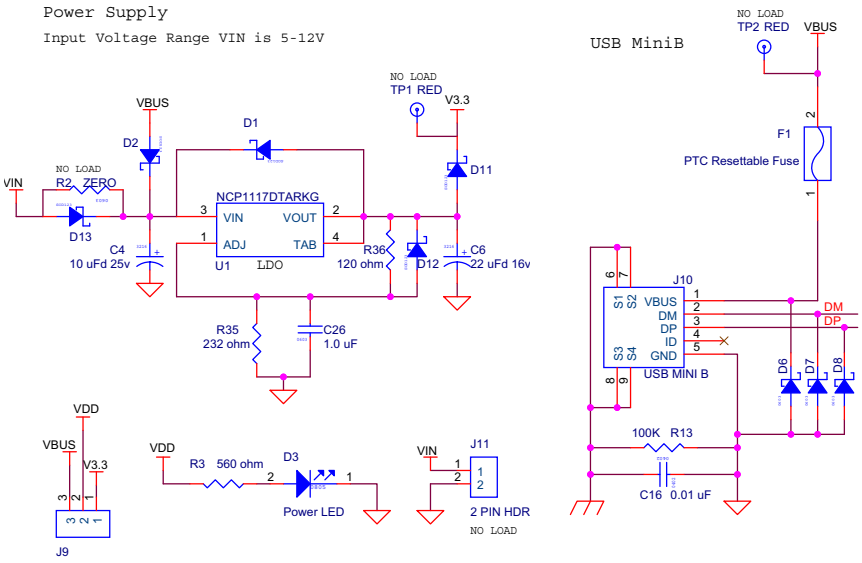


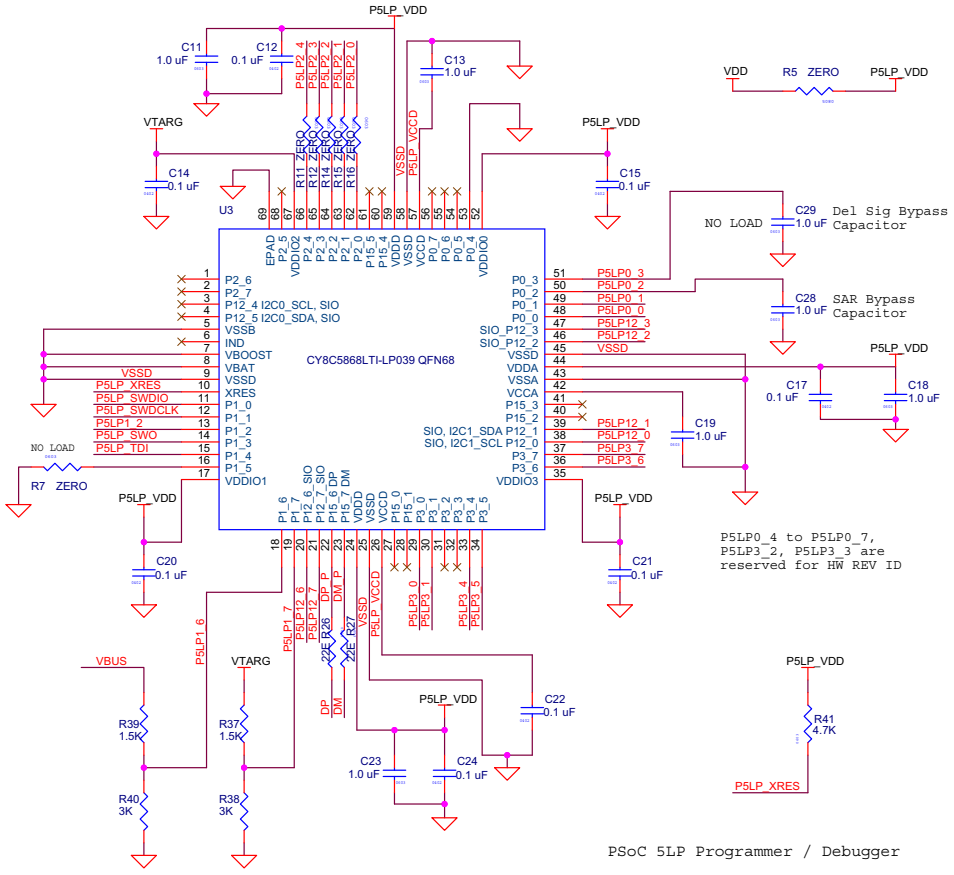
7. The user has now successfully restored the factory program on the PSoC 5LP. It can now be used as the Programmer/Debugger for the PSoC 4 device on the Pioneer Kit.

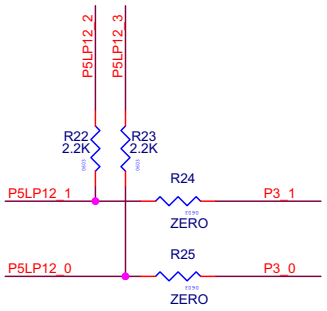
A. Appendix

A.1 CY8CKIT-042 Schematics

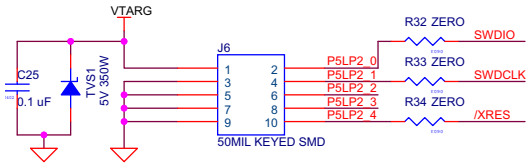




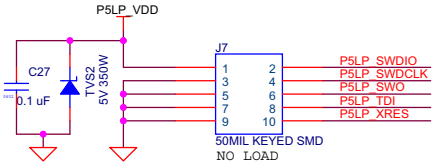




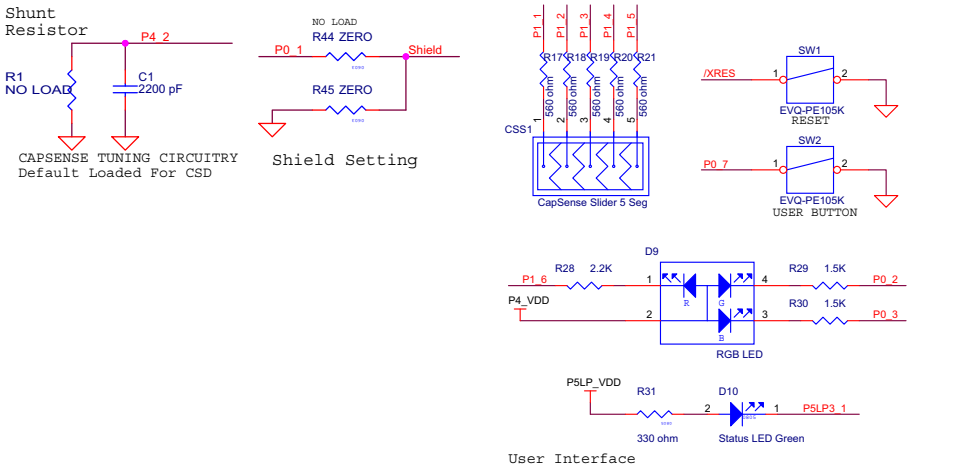
I2C Connection b/w PSoC 5LP and PSoC 4

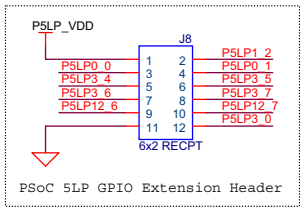


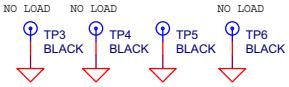
PSoC 4 / External PSoC Program/Debug Header

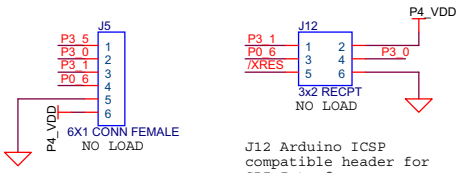


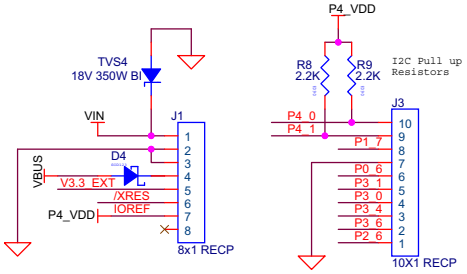
PSoC 5LP Program/Debug Header

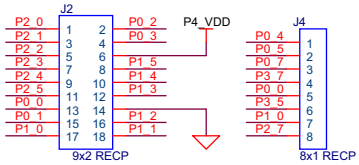




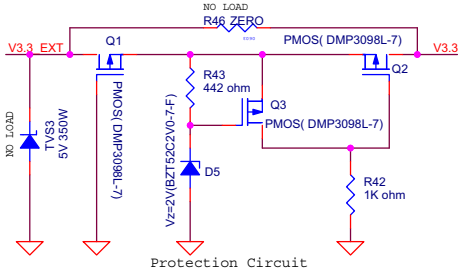








(J1-J4) Arduino Compatible Headers



A.2 Pin Assignment Table

This section provides the pin map of the headers and their usage.

A.2.1 Arduino Compatible headers (J1, J2, J3, J4, and J12)

J1		
Pin	**Kit Signal**	**Description**
J1_01	VIN	Input voltage to the board
J1_02	GND	GND
J1_03	GND	GND
J1_04	5V	5V voltage
J1_05	3.3V	3.3V voltage
J1_06	RESET	/XRES
J1_07	IOREF	IO voltage reference
J1_08	NC	Not connected

J2					
Pin	**PSoC 4 Signal**	**PSoC 4 Description**	**Pin**	**PSoC 4 Signal**	**PSoC 4 Description**
J2_01	P2[0]	A0(SARADC input)	J2_02	P0[2]	Comparator 2+
J2_03	P2[1]	A1(SARADC input)	J2_04	P0[3]	Comparator 2-
J2_05	P2[2]	A2(SARADC input)	J2_06	VDD	VDD
J2_07	P2[3]	A3(SARADC input)	J2_08	P1[5]	Opamp 2+
J2_09	P2[4]	A4(SARADC input)	J2_10	P1[4]	Opamp 2-
J2_11	P2[5]	A5(SARADC input)	J2_12	P1[3]	Opamp 2out
J2_13	P0[0]	Comparator 1+	J2_14	GND	GND
J2_15	P0[1]	Comparator 1-	J2_16	P1[2]	Opamp 1out
J2_17	P1[0]	Opamp 1+	J2_18	P1[1]	Opamp 1-

J3		
Pin	**PSoC 4 Signal**	**PSoC 4 Description**
J3_01	P2[6]	D8
J3_02	P3[6]	D9(PWM)
J3_03	P3[4]	D10(PWM/SS)
J3_04	P3[0]	D11(PWM/MOSI)
J3_05	P3[1]	D12(MISO)
J3_06	P0[6]	D13(SCK)
J3_07	GND	GND
J3_08	P1[7]	AREF
J3_09	P4[1]	SDA
J3_10	P4[0]	SCL

J4		
Pin	**PSoC 4 Signal**	**PSoC 4 Description**
J4_01	P0[4]	D0(RX)
J4_02	P0[5]	D1(TX)
J4_03	P0[7]	D2
J4_04	P3[7]	D3(PWM)
J4_05	P0[0]	D4
J4_06	P3[5]	D5(PWM)
J4_07	P1[0]	D6(PWM)
J4_08	P2[7]	D7

J12		
Pin	**Kit Signal**	**PSoC 4 Description**
J12_01	P3[1]	MISO
J12_02	PSoC 4_VDD	VDD
J12_03	P0[6]	SCK
J12_04	P3[0]	MOSI
J12_05	/XRES	PSoC 4 RESET
J12_06	GND	GND

A.2.2 Digilent Pmod Cards Support Header (J5)

J5		
Pin	**Kit Signal**	**PSoC 4 Description (Default Pmod signals)**
J5_01	P3[5]	SPI_SS (multiplex with J4_06)
J5_02	P3[0]	SPI_MOSI
J5_03	P3[1]	SPI_MISO
J5_04	P0[6]	SPI_SCK
J5_05	GND	GND
J5_06	VDD	VCC

A.2.3 PSoC 5LP GPIO Header (J8)

J8 is a 2x6 header that connects pins of PSoC 5LP to support GPIO controls for custom PSoC 5LP projects.

J8					
Pin	**PSoC 5 LP Signal**	**PSoC 5LP Description**	**Pin**	**PSoC 5 LP Signal**	**PSoC 5LP Description**
J8_01	PSoC 5 LP_VDD	VDD	J8_02	P1[2]	Digital I/O
J8_03	P0[0]	Delta Sigma ADC + input	J8_04	P0[1]	Delta Sigma ADC – input
J8_05	P3[4]	SAR – input	J8_06	P3[5]	SAR + input
J8_07	P3[6]	Buffered VDAC	J8_08	P3[7]	Buffered VDAC
J8_09	P12[6]	UART RX	J8_10	P12[7]	UART TX
J8_11	GND	GND	J8_12	P3[0]	IDAC output

A.3 Program and Debug Headers

A.3.1 PSoC 4 Direct Program/Debug Header (J6)

J6							
Pin	**PSoC 5 LP Signal**	**PSoC 4 Signal**	**Description**	**Pin**	**PSoC 5 LP Signal**	**PSoC 4 Signal**	**Description**
J6_01	VDD	VDD	VCC	J6_02	P2[0]	P3[2]	TMS/SWDIO
J6_03	GND	GND	GND	J6_04	P2[1]	P3[3]	TCLK/SWCLK
J6_05	GND	GND	GND	J6_06	P2[2]	NC	TDO/SWO
J6_07	NC	GND	GND	J6_08	P2[3]	NC	TDI
J6_09	GND	GND	GND	J6_10	P2[4]	XRES	RESET

A.3.2 PSoC 5LP Direct Program/Debug Header (J7)

J7					
Pin	**PSoC 5 LP Signal**	**Description**	**Pin**	**PSoC 5 LP Signal**	**Description**
J7_01	VDD	VCC	J7_02	P1[0]	TMS/SWDIO
J7_03	GND	GND	J7_04	P1[1]	TCLK/SWCLK
J7_05	GND	GND	J7_06	P1[3]	TDO/SWO
J7_07	GND	GND	J7_08	P1[4]	TDI
J7_09	GND	GND	J7_10	XRES	RESET

A.4 Use of Zero ohm resistors and NOLOAD

Unit	Resistor	Usage
Power supply	R2	Solder zero ohm resistance to access voltage from VBUS (USB).
I2C connection between PSoC 5LP and PSoC 4	R24 and R25	Unsolder the resistors to communicate with an external PSoC using the PSoC 5LP. Removing these will disable the PSoC 4 programming by the PSoC 5LP device.
PSoC 4/ External PSoC Program / debug header	R32, R33 and R34	Unsolder the resistors to disconnect SWD lines from the PSoC 4. Use J6 to connect and program an external PSoC.
Protection Circuit	R46	Solder zero ohm resistance to bypass the entire protection circuitry.
CapSense tuning circuitry	R1	Used when RBleed mode of CSD is used. To use this feature the user needs to populate an Rbleed resistor. Refer to the CapSense component datasheet.
CapSense Shield Setting	R44, R45	Unsolder R45 which connects Shield to ground and solder R44 with zero ohm resistance to connect Vref via P0_1.
PSoC 4	R4, R6	Unsolder R4 to remove supply to VTARG and solder zero ohm resistance R6 to supply P4_VDD with VDD instead of J13.
PSoC 5 LP Programmer/Debugger	R11, R12, R14, R15, R16	Meant for future use.
	R5	Unsolder the zero ohm resistor to cut-off the VDD supply to PSoC 5LP.
	R7	Meant for future use.

A.5 Error in Firmware / Status Indication in Status LED

	User Indication	Scenario	Action Required by user
1	LED blinking at a fast rate (ON Time = 0.25s, OFF Time = 0.25s)	Bootloadable is corrupt	Bootload the *.cyacd file over USB interface that is shipped with PSoC Programmer using the Bootloader Host GUI shipped with PSoC Creator. The files are located in the PSoC programmer root installation directory.
2	LED blinking at slow rate (ON Time = 1.5s, OFF Time = 1.5s)	Entered Boot-loader by press-ing Reset switch of PSoC 4	a) Unplug the power and plug it in again if the user has entered this mode by mistake. The LED gives and indication to the user. b) If the mode entry was intentional, Bootload the new *.cyacd file using the Bootloader Host tool shipped with PSoC creator.
3	LED glowing steadily	Programmer Application is run-ning successfully	USB is enumerated successfully and the programmer is up and running.The PSoC 4 device can now be pro-grammed any time using the onboard PSoC 5LP pro-grammer.

Note: LED status is not applicable when a custom project is running in PSoC 5LP.

A.6 Bill of Materials (BOM)

No.	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
1				PCB,3.32"x2.1" CAF resistant High Tg ENIG finish, 4 layer, Color = RED, Silk = WHITE.	Cypress	
2	1	C1	2200 pFd	CAP CER 2200PF 50V 5% NP0 0805	Murata	GRM2165C1H222JA01 D
3	12	C2,C7,C12,C14,C1 5,C17,C20,C21,C2 2,C24,C25,C27	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	Panasonic - ECG	ECJ-0EF1C104Z
4	11	C3,C5,C8,C10,C11, C13,C18,C19,C23, C26,C28	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
5	1	C4	10 uF 25V	CAP TANT 10UF 25V 10% 1210	AVX Corporation	TPSB106K025R1800
6	1	C6	22 uF 16V	CAP TANT 22UF 16V 10% 1210	AVX Corporation	TPSB226K016R0600
7	1	C9	10000 pFd	CAP CER 10000PF 50V 5% NP0 0805	Murata	GRM2195C1H103JA01 D
8	1	C16	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	Panasonic - ECG	ECJ-0EB1C103K
9	6	D1,D2,D4,D11,D12, D13	MBR05	DIODE SCHOTTKY 0.5A 20V SOD-123	Fairchild Semi-conductor	MBR0520L

No.	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
10	1	D3	Power LED Amber	LED AMBER 591NM DIFF LENS 2012	Sharp Microelectronics	LT1ZV40A
11	1	D5	2V Zener	DIODE ZENER 2V 500MW SOD123	Diodes Inc	BZT52C2V0-7-F
12	3	D6, D7, D8	ESD diode	SUPPRESSOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603MLC-05LE
13	1	D9	RGB LED	LED RED/GREEN/BLUE PLCC4 SMD	Cree, Inc.	CLV1A-FKB-CJ1M1F1BB7R4S3
14	1	D10	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
15	1	F1	FUSE	PTC Resettable Fuses 15Volts 100Amps	Bourns	MF-MSMF050-2
16	2	J1, J4	8x1 RECP	CONN HEADER FEMALE 8POS .1" GOLD	Sullins Connector Solutions	PPPC081LFBN-RC
17	1	J2	9x2 RECP	CONN HEADER FMAL 18PS.1" DL GOLD	Sullins Connector Solutions	PPPC092LFBN-RC
18	1	J3	10x1 RECP	CONN HEADER FMALE 10POS .1" GOLD	Sullins Connector Solutions	PPPC101LFBN-RC
19	1	J6	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K
20	1	J8	6x2 RECP	CONN HEADER FMAL 12PS.1" DL GOLD	Sullins Connector Solutions	PPPC062LFBN-RC
21	1	J9	3p_jumper	CONN HEADER VERT SGL 3POS GOLD	3M	961103-6404-AR
22	1	J10	USB Mini B	CONN USB MINI AB SMT RIGHT ANGLE	TE Connectivity	1734035-2
23	1	J13	2p_jumper	CONN HEADER VERT SGL 2POS GOLD	3M	961102-6404-AR
24	3	Q1,Q2,Q3	PMOS	MOSFET P-CH 30V 3.8A SOT23-3	Diodes Inc	DMP3098L-7
25	1	R3	560 ohm	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ561V
26	12	R4,R11,R12,R14,R15,R16,R24,R25,R32,R33,R34,R45	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic-ECG	ERJ-3GEY0R00V
27	1	R5	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
28	4	R8,R9,R22,R23	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V
29	2	R10,R41	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic-ECG	ERJ-3GEYJ472V

No.	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
30	1	R13	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X
31	5	R17,R18,R19,R20, R21	560 ohm	RES 560 OHM 1/10W 5% 0603 SMD	Panasonic-ECG	ERJ-3GEYJ561V
32	2	R26, R27	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V
33	1	R28	2.2K	RES 2.2K OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ222V
34	2	R29,R30	1.5K	RES 1.5K OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ152V
35	1	R31	330 ohm	RES 330 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ331V
36	1	R35	232 ohm	RES 232 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF2320V
37	1	R36	120 ohm	RES 120 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1200V
38	2	R37,R39	1.5K	RES 1.5K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ152V
39	2	R38,R40	3K	RES 3.0K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ302V
40	1	R42	1K	RES 1K OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ102V
41	1	R43	442 ohm	RES 442 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF4420V
42	2	SW1,SW2	SW PUSH-BUTTON	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-PE105K
43	1	TP5	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001
44	2	TVS1,TVS2	5V 350W	TVS UNIDIR 350W 5V SOD-323	Dioded Inc.	SD05-7
45	1	TVS4	18V 350W	TVS DIODE 18V 1CH BI SMD	Bourns Inc.	CDSOD323-T18C
46	1	U1	NCP1117D TARKG	NCP1117DTARKG	ON Semiconductor	NCP1117DTARKG
47	1	U2	PSoC 4 (CY8C4245 AXI-483)	44TQFP PSoC4A target chip	Cypress Semiconductor	CY8C4245AXI-483
48	1	U3	PSoC 5LP (CY8C5868 LTI-LP039)	68QFN PSoC 5LP chip for USB debug channel and USB-Serial interface	Cypress Semiconductor	CY8C5868LTI-LP039
No Load Components						
49	1	C29	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T

No.	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
50	1	J5	6X1 RECP RA	CONN FEMALE 6POS .100" R/A GOLD	Sullins Connector Solutions	PPPC061LGBN-RC
51	1	J7	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K
52	1	J11	2 PIN HDR	CONN HEADER FEMALE 2POS .1" GOLD	Sullins Connector Solutions	PPPC021LFBN-RC
53	1	J12	3x2 RECPT	CONN HEADER FMAL 6PS .1" DL GOLD	Sullins Connector Solutions	PPPC032LFBN-RC
54	5	R1,R2,R7,R44,R46	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic-ECG	ERJ-3GEY0R00V
55	1	R6	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
56	2	TP1,TP2	RED	TEST POINT PC MINI .040"D RED	Keystone Electronics	5000
57	3	TP3,TP4,TP6	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001
58	1	TVS3	5V 350W	TVS UNIDIR 350W 5V SOD-323	Dioded Inc.	SD05-7
Install on Bottom of PCB As per the Silk Screen in the Corners						
59	4	N/A	N/A	BUMPON CYLINDRICAL.312X.215 BLACK	3M	SJ61A6
Special Jumper Installation Instructions						
60	1	J9	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E
61	1	J13	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E
Label						
62	1	N/A	N/A	LBL, Kit Product Identification Label, Vendor Code, Datecode, Serial Number CY8CKIT-042 Rev** (YYWWVVXXXXX)	Cypress Semiconductor	
63	1	N/A	N/A	LBL, PCBA Anti-Static Warning, 10mm X 10mm	Cypress Semiconductor	
64	1	N/A	N/A	Assembly Adhesive Label, Manufacturing ID	Cypress Semiconductor	
65	1	N/A	N/A	Kit QR code	Cypress Semiconductor	

A.7 Regulatory Compliance Information

CY8CKIT-042 has been tested and verified to comply with the following electromagnetic compatibility (EMC) regulations:

- EN 55022:2010 Class A - Emissions
- EN 55024:2010 Class A - Immunity