CY8CKIT-042-BLE

Bluetooth® Low Energy (BLE) Pioneer Kit Guide

Doc. # 001-93731 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone (USA): 800.858.1810
Phone (Intnl): +1.408.943.2600
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

The CY8CKIT-042-BLE Bluetooth Low Energy (BLE) Pioneer Kit is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. For this reason, the board may cause interference with 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.

The CY8CKIT-042-BLE Bluetooth Low Energy (BLE) Pioneer Kit is intended for use as a development, demonstration and evaluation platform for hardware or software in a laboratory environment. The kit is not intended for general consumer use. It generates, uses and can radiate radio frequency energy. It has not been tested for compliance with the limits applicable under any standard. Operation of the equipment may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference. Cypress recommends that the kit only be used in a shielded room.

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-BLE boards contain electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, which can cause a 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-BLE boards in the protective shipping package.

End-of-Life/Product Recycling

The end-of-life cycle for this kit is five years from the date of manufacture mentioned on the back of the box. Contact your nearest recycler to discard 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 an ESD workstation is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to a grounded metal object.

Handling Boards

CY8CKIT-042-BLE 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 the board over any surface.

Battery Care and Use

- Use the correct size and type of battery specified in this guide.
- Keep battery contact surfaces and battery compartment contacts clean by rubbing them with a clean pencil eraser or a rough cloth each time you replace batteries.
- Remove the battery from a device when it is not expected to be in use for several months.
- Make sure that you insert the battery into your device properly, with the + (plus) and – (minus) terminals aligned correctly.
- Do not place the battery next to metallic objects such as keys and coins.
- Never throw the battery into fire.
- Do not open up the battery.
- Do not short the battery.
- Do not subject the battery to high temperatures or high humidity.
- Store the battery in a dry place.
- Do not recharge a battery unless it is marked “rechargeable.”

Battery Disposal

Batteries can be safely disposed of with normal household waste. Never dispose of batteries in fire because they can explode.

It is important not to dispose of large amounts of batteries in a group. Used batteries are often not completely “dead.” Grouping used batteries together can bring these “live” batteries into contact with one another, creating safety risks.

1. Introduction

Thank you for your interest in the CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit. This kit is designed to showcase the functionality and ease of use of PSoC® 4 BLE and PRoC™ BLE devices while developing Bluetooth Low Energy (Bluetooth Smart) applications. Cypress's BLE solution provides an easy-to-use, intuitive GUI to configure the BLE protocol stack using a BLE component available in the Cypress standard integrated development environment (IDE), PSoC Creator. The CySmart PC tool allows emulation of a central device and quick access to peripheral connections and debugging. The solution provides a true single-chip solution with an integrated balun, Cypress's industry-leading capacitive sensing technology, an analog front end (AFE) for biometric sensors, and digital peripherals suited to a wide variety of applications. Designed for flexibility, this kit offers footprint compatibility with several third-party Arduino™ shields. The kit includes 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 push-button switch, an integrated USB programmer, a program and debug header, an F-RAM, and USB-UART/I^2C bridges.

1.1 Kit Contents

The BLE Pioneer Kit contains the following, as shown in Figure 1-1.

- BLE Pioneer Baseboard preloaded with the CY8CKIT-142 PSoC 4 BLE Module
- CY5671 PRoC BLE Module
- CY5670 CySmart USB Dongle
- Quick start guide
- USB standard A to mini-B cable
- Four jumper wires (4 inch) and two proximity sensor wires (5 inch)
- Coin cell

Figure 1-1. Kit Contents

If any part of the kit is missing, contact your nearest Cypress sales office for help: www.cypress.com/go/support.

1.2 Board Details

The BLE Pioneer Baseboard consists of the blocks shown in Figure 1-2.

1. RGB LED
2. BLE module reset button
3. CapSense proximity header
4. User button
5. CapSense slider
6. LDO 1.9 V~5 V
7. Arduino compatible I/O header (J2)
8. Arduino compatible power header (J1)
9. Digilent Pmod compatible I/O header (J5)
10. Cypress F-RAM 1 Mb (FM24V10-G)
11. PSoC 5LP I/O header (J8)
12. PSoC 5LP - programmer and debugger (CY8C5868LTI-LP039)
13. Coin cell holder (bottom side)
14. USB connector (J13)
15. Power LED
16. Status LED

17. System power supply jumper (J16)
18. Arduino compatible I/O header (J3)
19. Arduino compatible I/O header (J4)
20. BLE power supply jumper (J15)
21. BLE module headers (J10/J11)

Figure 1-2. BLE Pioneer Baseboard Markup

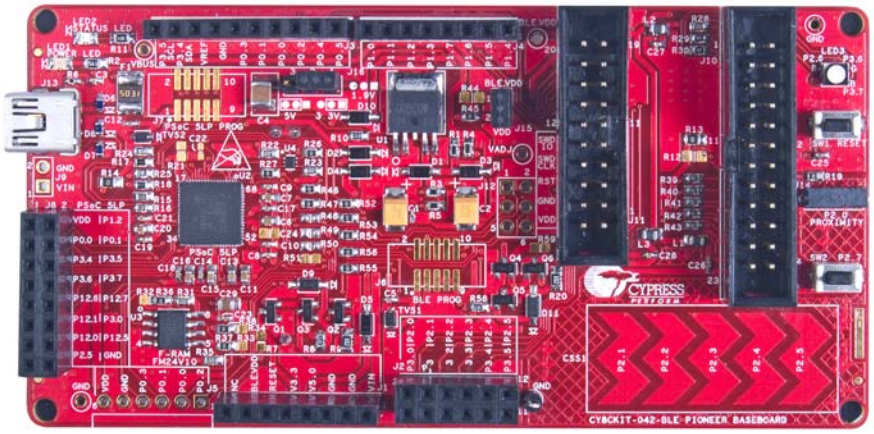
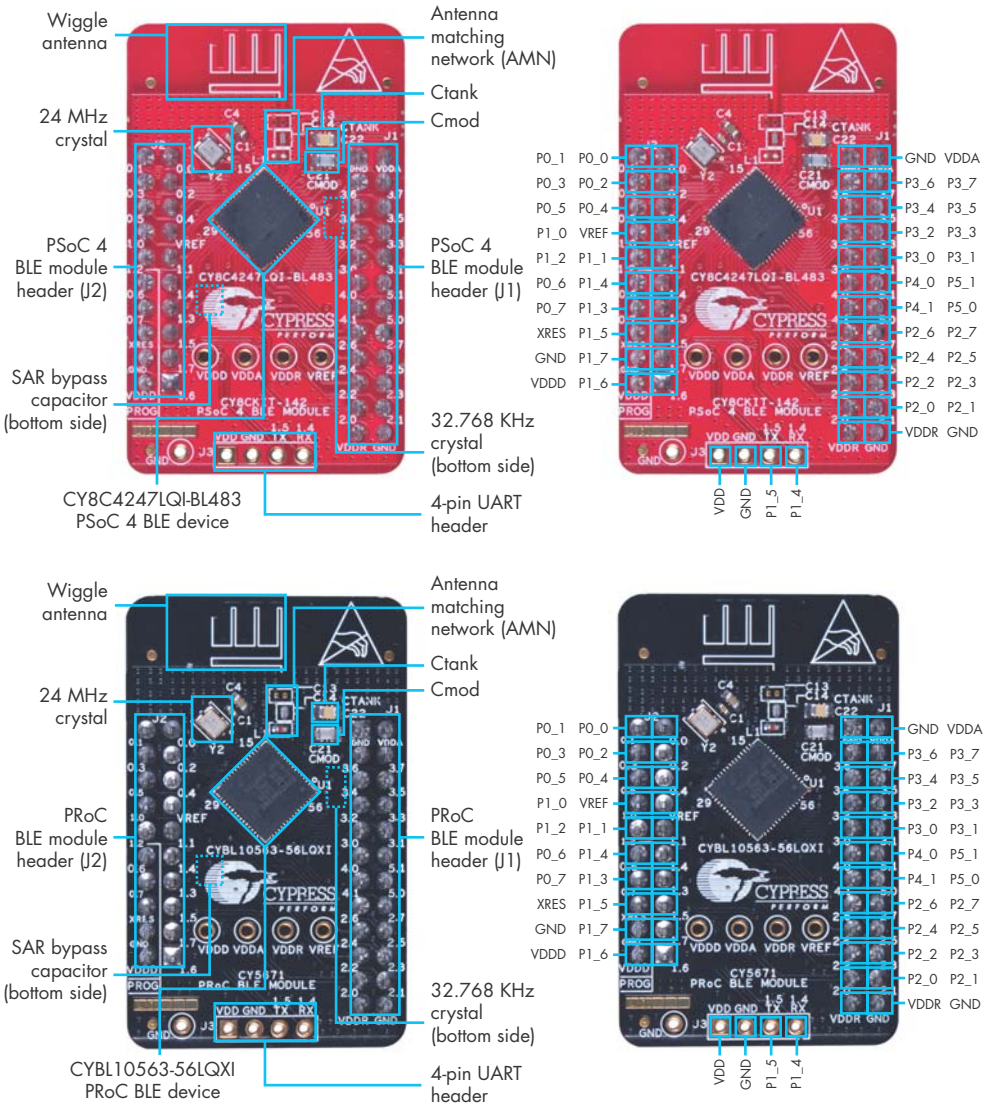


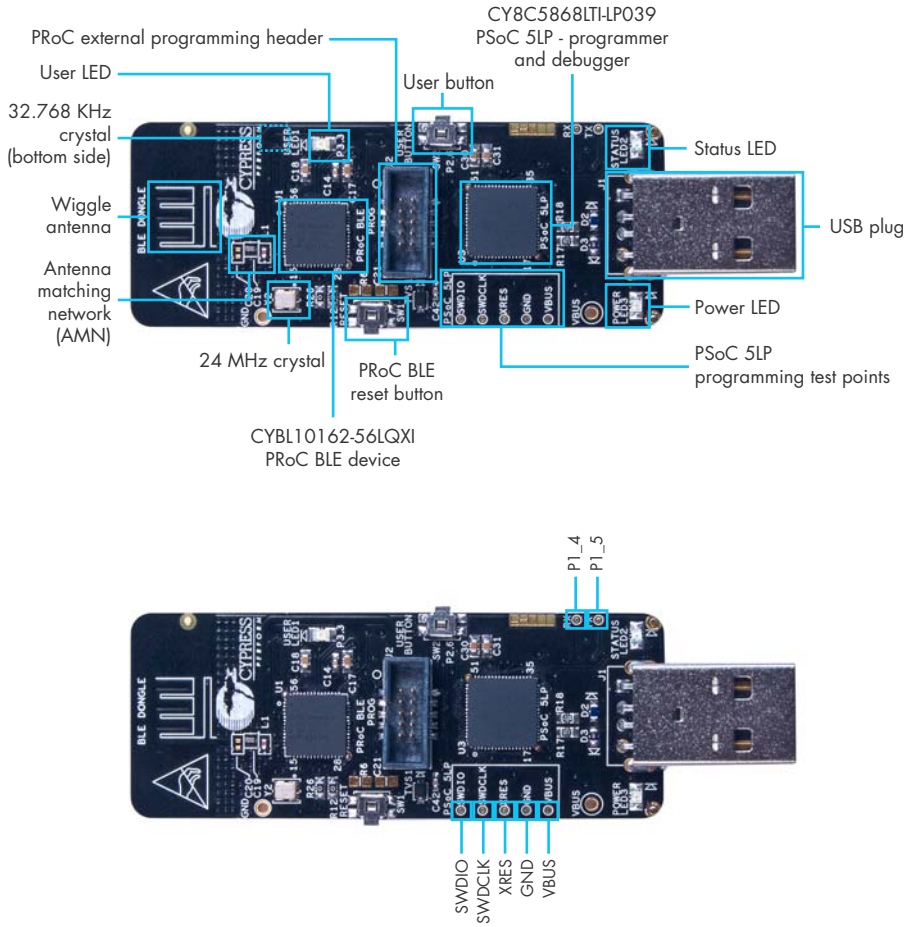
Figure 1-3 shows a markup of the onboard components, where red BLE module denotes the PSoC 4 BLE module and black BLE module denotes the PRoC BLE module.

Figure 1-3. BLE Module Markup



The Dongle board consists of the blocks shown in Figure 1-4.

Figure 1-4. BLE Dongle Markup



1.3 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 preverified and precharacterized PSoC Components™.

With PSoC Creator, you 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 you to tap into an entire tool ecosystem with integrated compiler chains and production programmers for PSoC devices.

For more information, visit www.cypress.com/psoccreator.

1.4 Getting Started

This guide will help you get acquainted with the BLE Pioneer Kit:

- The Software Installation chapter on page 19 describes the installation of the kit software. This includes the PSoC Creator IDE for development and debugging applications, PSoC Programmer for programming hex files, and the CySmart PC Tool for BLE host emulation.
- The Kit Operation chapter on page 23 describes the major features of the BLE Pioneer Kit and functionalities such as programming, debugging the USB-UART and USB-I^{2}C bridges.
- The Example Projects chapter on page 37 describes multiple PSoC 4 BLE and PRoC BLE code examples that will help you understand how to create your own BLE application using the BLE component and device.
- The Hardware chapter on page 83 details the hardware content of the kit and the hardware operation.
- The Advanced Topics chapter on page 113 explains the functionality of some features of the kit, such as the USB-UART bridge, USB-I^{2}C bridge, F-RAM, iOS app, and CySmart PC Tool.
- The Appendix on page 167 provides schematics, board layouts, and the bill of materials (BOM).

1.5 Additional Learning Resources

Visit www.cypress.com/go/psoc4ble and www.cypress.com/procble for additional learning resources in the form of datasheets, technical reference manuals, and application notes.

Visit www.cypress.com/go/cysmart for information on the CySmart PC tool.

1.5.1 Beginner Resources

PSoC Creator Training: www.cypress.com/go/creatorstart/creatortraining

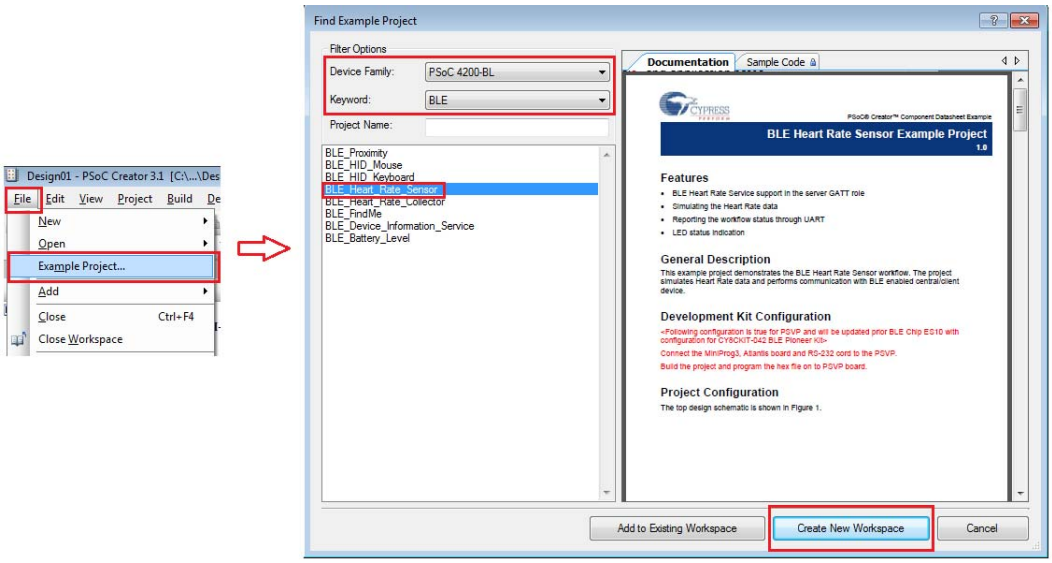
1.5.2 Application Notes

Visit www.cypress.com/appnotes to view a growing list of application notes for PSoC 3, PSoC 4, PSoC 4 BLE, PRoC BLE, and PSoC 5LP.

1.5.3 PSoC Creator Example Projects

These simple example projects demonstrate how to configure and use PSoC Creator components. To open an example project in PSoC Creator, go to **File > Example Project** (see Figure 1-5) and choose the required example project.

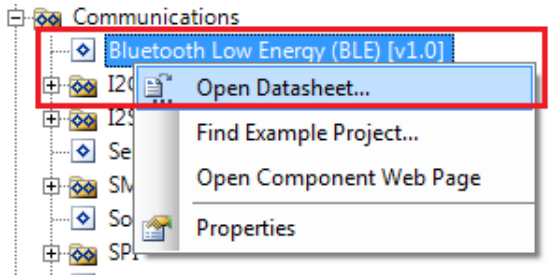
Figure 1-5. PSoC Creator Example Projects



1.5.4 Component Datasheets

Right-click a component and select **Open Datasheet** (see Figure 1-6). Visit this page for the BLE component datasheet.

Figure 1-6. Opening Component Datasheet



1.5.5 Bluetooth Learning Resources

The Bluetooth Developer Portal provides material by the Special Interest Group (SIG) for learning various aspects of the Bluetooth Low Energy protocol and systems. Some of them are:

- Training videos
- GATT profiles
- Bluetooth community forum

1.5.6 Learning From Peers

Cypress Developer Community Forums: Visit www.cypress.com/forums

1.5.7 Other Related Resources

- Digilent PMod: www.digilentinc.com/pmods/
- Arduino: http://arduino.cc/en/Main/ArduinoBoardUno

1.6 Technical Support

For assistance, go to our support web page, www.cypress.com/support, or contact our customer support at +1 (800) 541-4736 Ext. 2 (in the USA) or +1 (408) 943-2600 Ext. 2 (International).

1.7 Documentation Conventions

Table 1-1. Document Conventions for Guides

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

1.8 Acronyms

Table 1-2. Acronyms Used in this Document

Acronym	Definition
ADC	Analog-to-Digital Converter
API	Application Programming Interface
BLE	Bluetooth Low Energy
CD	Compact Disc
CDC	Communications Device Class
COM	Communication Port
DVD	Digital Video Disc
ESD	Electrostatic Discharge
F-RAM	Ferroelectric Random Access Memory
GUI	Graphical User Interface
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
IAS	Immediate Alert Service
IDAC	Interconnecting Digital-Analog Converter
IDE	Integrated Development Environment
ISO	International Organization for Standardization
LDO	Low Drop Out (voltage regulator)
LED	Light-Emitting Diode
LP	Low Power
LPT	Line Print Terminal
PrISM	Precise Illumination Signal Modulation
PRoC	Programmable Radio-on-Chip
PRM	Protocol Service Multiplexer
PSoC	Programmable Systems-on-Chip
PWM	Pulse-Width Modulation
QFN	Quad Flat No-lead (package)
RGB	Red Green Blue
SAR	Successive Approximation Register
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB CDC	Universal Serial Bus Communications Device Class

2. Software Installation

This chapter describes the steps to install the software tools and packages on a PC for using the BLE Pioneer Kit. This includes the IDE on which the projects will be built and used for programming.

2.1 Before You Begin

All Cypress software installations require administrator privileges. However, this is not the case for installed software. Before you install the kit software, close any other Cypress software that is currently running.

2.2 Install Software

Follow these steps to install the BLE Pioneer Kit software:

1. Download the BLE Pioneer Kit software from www.cypress.com/CY8CKIT-042-BLE. The kit software is available in the following formats:
   a. CY8CKIT-042-BLE Kit Setup: This installation package contains the files related to the kit. However, it does not include the Windows Installer or Microsoft .NET framework packages. If these packages are not on your computer, the installer directs you to download and install them from the Internet.
   b. CY8CKIT-042-BLE Kit Only: This executable file installs only the kit contents, which include kit code examples, hardware files, and user documents. This package can be used if all the software prerequisites (listed in step 5) are installed on your PC.
   c. CY8CKIT-042-BLE DVD ISO: This file is a complete package, stored in a DVD-ROM image format, that you can use to create a DVD or extract using an ISO extraction program such as WinZip or WinRAR. The file can also be mounted similar to a virtual CD/DVD using virtual drive programs such as ‘Virtual CloneDrive’ and ‘MagicISO’. This file includes all the required software, utilities, drivers, hardware files, and user documents.
2. If you have downloaded the ISO file, mount it on a virtual drive; if you do not have a virtual drive to mount, extract the ISO contents. Double-click *cyautorun.exe* in the root directory of the extracted content or mounted ISO if “Autorun from CD/DVD” is not enabled on the PC. The installation window will appear automatically.

   Note: If you are using the “Kit Setup” or “Kit Only” file, then go to step 4 for installation.

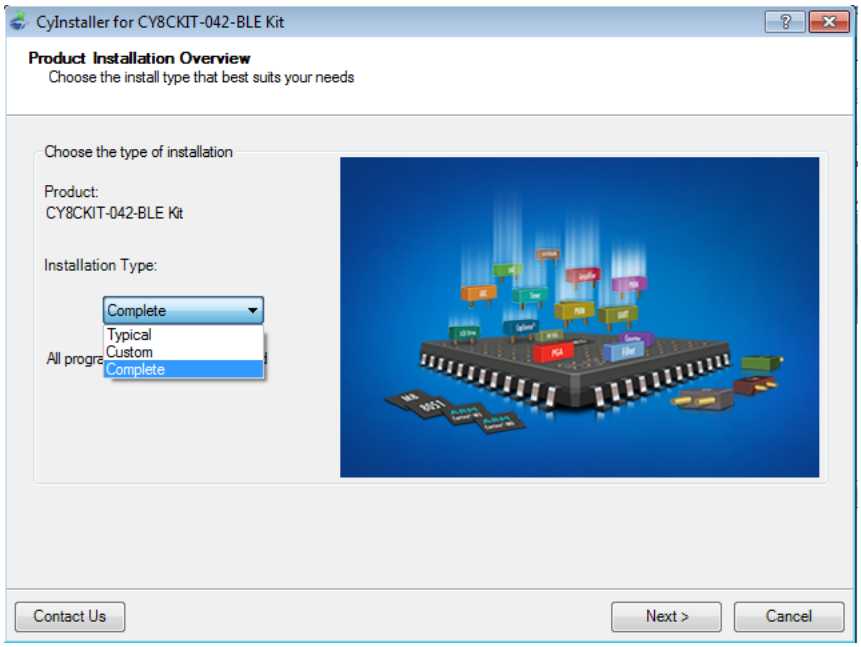
3. Click I**nstall CY8CKIT-042-BLE Kit** to start the kit installation, as shown in Figure 2-1.

Figure 2-1. Kit Installer Screen

4. Select the folder in which you want to install the CY8CKIT-042-BLE kit-related files. Choose the directory and click **Next**.
5. When you click **Next**, the CY8CKIT-042-BLE Kit installer automatically installs the required software, if it is not present on your computer. Following are the required software:
   Note: Incase of Setup Only installer package, please download and install below prerequisites.
   a. PSoC Creator 3.1 or later: Download the latest version from www.cypress.com/psoccreator.
   b. PSoC Programmer 3.21.1 or later: This is installed as part of PSoC Creator installation (www.cypress.com/programmer).
   c. CySmart 1.0 or later: Download the latest version from www.cypress.com/go/cysmart.

6. Choose the Typical/Custom/Complete installation type in the **Product Installation Overview** window, as shown in Figure 2-2. Click **Next** after you select the installation type.

Figure 2-2. Product Installation Overview



7. Read the license agreement and select **I accept the terms in the license agreement** to continue with installation. Click **Next**.
8. When the installation begins, a list of packages appears on the installation page. A green check mark appears next to each package after successful installation.
9. Click **Finish** to complete the CY8CKIT-042-BLE kit installation.
10. Enter your contact information or select the check box **Continue Without Contact Information**. Click **Finish** to complete the CY8CKIT-042-BLE kit installation.
11. After the installation is complete, the kit contents are available at the following location:

`<Install_Directory>\CY8CKIT-042-BLE Kit`

Default location:

Windows 7 (64-bit): `C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit`

Windows 7 (32-bit): `C:\Program Files\Cypress\CY8CKIT-042-BLE Kit`

Note: For Windows 7/8/8.1 users, the installed files and the folder are read only. To use the installer example project, follow the steps outlined in the Example Projects chapter on page 37.

The kit installer also installs the CySmart PC Tool for PC. This software, along with the Dongle, allows the PC to emulate as a BLE central device. Refer to CySmart PC Tool on page 158 for more details on how to use the CySmart PC Tool.

2.3 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** for Windows 7 or **Add/Remove Programs** for Windows XP; select the **Uninstall/Change** button.

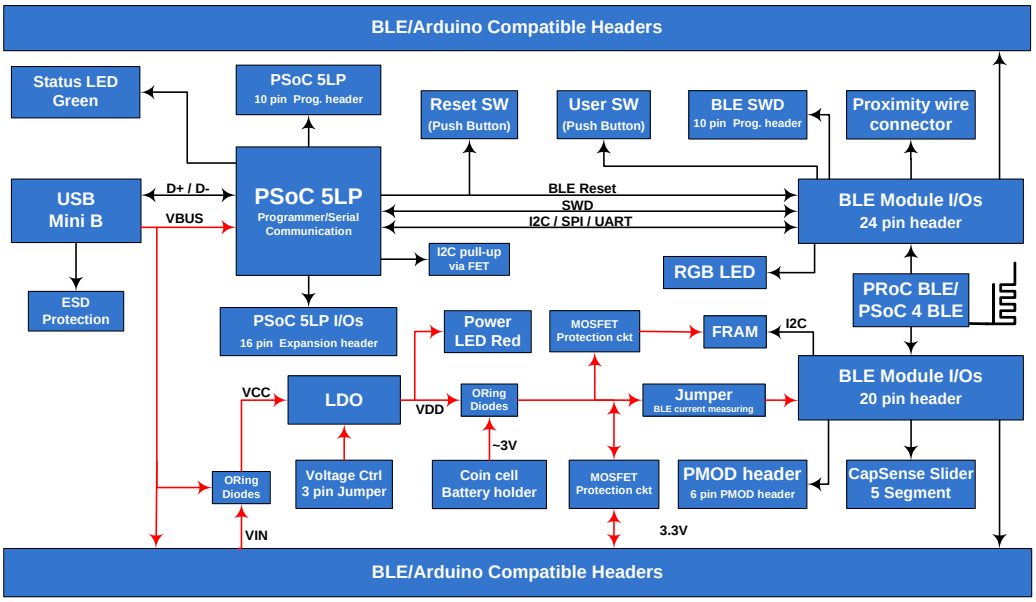
3. Kit Operation

This chapter introduces you to the BLE Pioneer kit and the features that will be used as part of the kit operation. These primarily include USB connection, programming/debugging, and programmer firmware update. The chapter also describes the USB-UART and USB-I^{2}C bridges along with the PC tools that can be used to communicate with the BLE device on the kit.

3.1 Theory of Operation

Figure 3-1, Figure 3-2, and Figure 3-3 show the block diagrams for the BLE Pioneer Baseboard, BLE Module board, and Dongle.

Figure 3-1. BLE Pioneer Baseboard Block Diagram

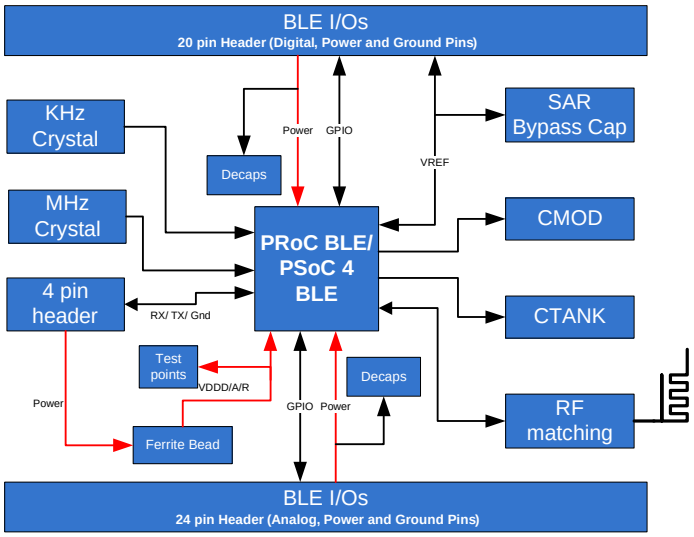


The BLE Pioneer board acts as the baseboard for the PSoC 4 BLE (red module) and PRoC BLE (black module), which can be connected to the BLE Pioneer board. The Pioneer board contains a PSoC 5LP that is used as an onboard programmer or debugger, and for the USB-Serial interface.

The Baseboard is Arduino form factor compatible, enabling Arduino shields to be connected on top of the board to extend the functionality of BLE modules. The board also features a 1-Mb F-RAM, an RGB LED, a five-segment CapSense slider, a proximity header, a user switch, and a reset switch for the PSoC 4 BLE and PRoC BLE devices on the module. The Pioneer board supports three voltage levels: 1.9 V, 3.3 V, and 5 V.

The baseboard can also be used as a standalone programmer to program and debug other PSoC 4 BLE/PRoC BLE devices using serial wire debug (SWD), and as a USB-Serial bridge. The firmware on PSoC 5LP device enables bootloading over USB to upgrade the firmware.

Figure 3-2. BLE Module Block Diagram



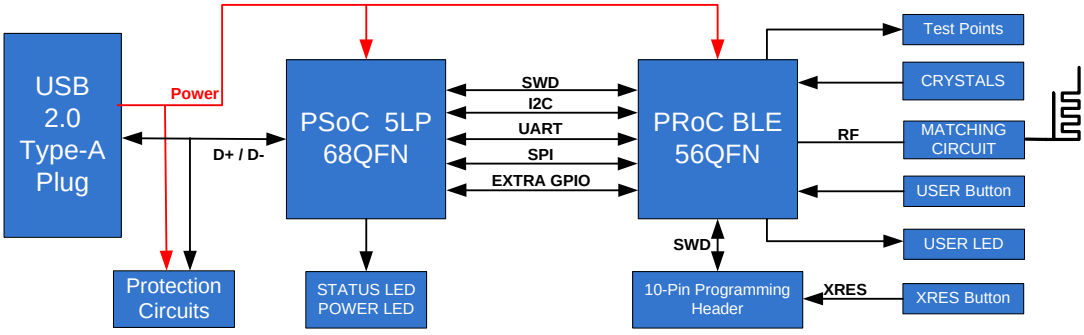
This kit includes two modules boards. These boards act as a basic breakout board for the CY8C4247LQI-BL483 and CYBL10563-56LQXI BLE silicon. The PSoC 4 BLE and PRoC BLE Modules are identical except for the silicon. In addition to including the PSoC 4 BLE and PRoC BLE devices, the module boards also contain the BLE passives (resistors, capacitors, external crystals, and antenna-matching network), an onboard antenna, and headers for connecting to the Baseboard.

The Dongle is the host's wireless interface for the BLE device or project on the baseboard. The dongle has a PRoC BLE device, configurable over various interfaces to work as expected for any project. It also contains a PSoC 5LP, to be used as an onboard programmer or debugger, and for the USB-Serial interface, as shown in Figure 3-3.

The Dongle has a USB A-type plug to connect the PSoC 5LP to the USB port of the host PC. The PSoC 5LP then communicates with the PRoC BLE device over UART or multiplexed I^2C or an SPI bus. The board also features a user LED, a user switch, and a reset switch for the PRoC BLE device. The Dongle is powered directly through the USB port (VBUS) at 5.0 V.

The Dongle can also be used as a standalone programmer to program and debug other PSoC devices (outside the dongle board) using SWD, and as a USB-Serial bridge after removing the resistor between the SWD pins of PSoC 5LP and PRoC BLE.

Figure 3-3. BLE Dongle Block Diagram



3.2 BLE Pioneer Kit USB Connection

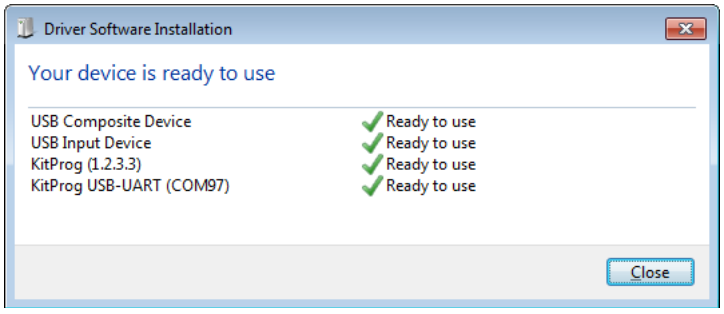
The BLE Pioneer kit connects to a PC over the USB interface (J13) and derives power from it. The kit enumerates as a composite device, as shown in Table 3-1.

Note: Ensure that you install the kit installer on the system for successful enumeration. To download and install the BLE Pioneer Kit, visit www.cypress.com/go/CY8CKIT-042-BLE.

Table 3-1. BLE Pioneer Kit Enumerated Interfaces

Port	Description
USB Composite Device	Composite device
KitProg	Programmer and debugger
KitProg USB-UART	USB-UART bridge, which will appear as a COM# port
USB Input Device	USB-I^{2}C bridge

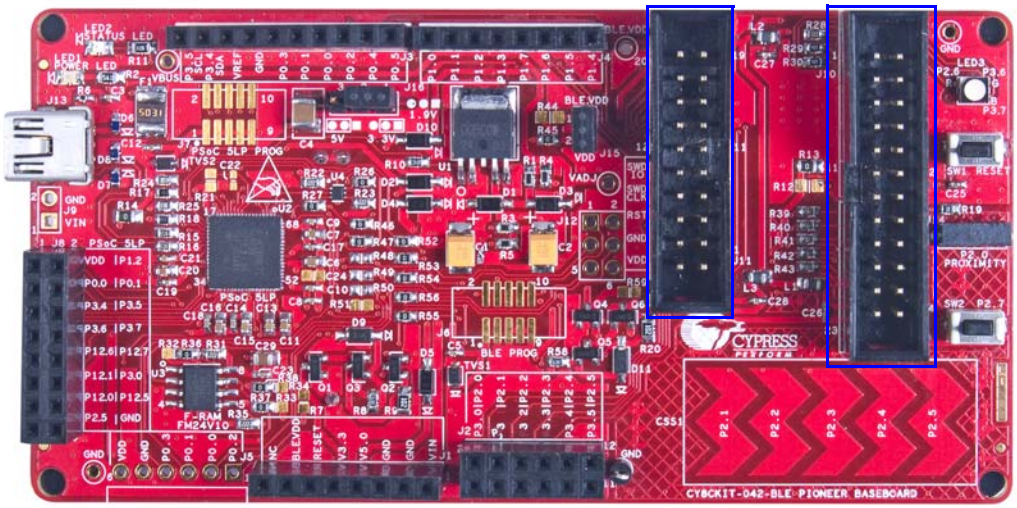
Figure 3-4. KitProg Driver Installation (appearance may differ depending on Windows platform)



3.3 Placing Modules on Baseboard

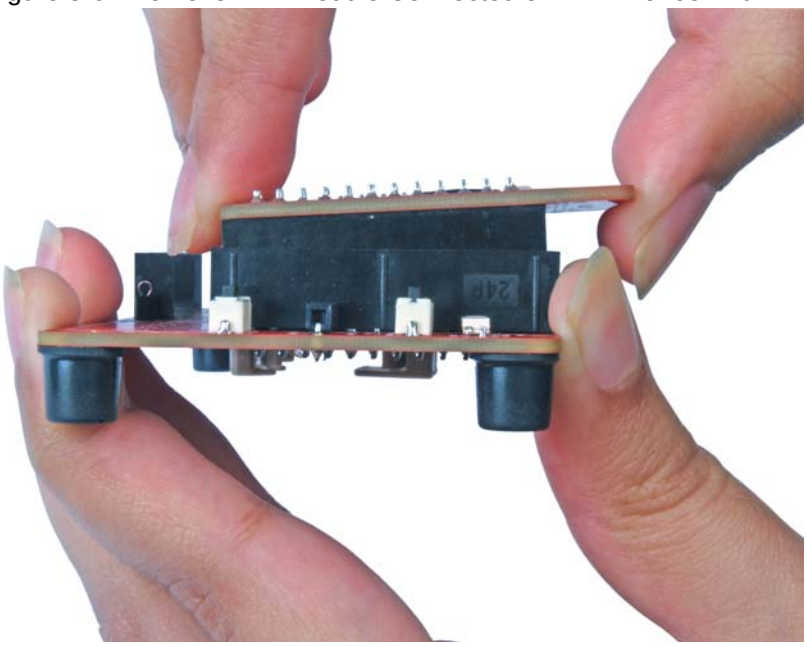
To connect the BLE Modules (PSoC 4 BLE or PRoC BLE) on the Baseboard, place the BLE Module over the headers J10 and J11 while keeping the antenna on the module directing outside the baseboard and press it. Note that the two parallel headers J10 and J11 are not equal (24-pin and 20-pin, respectively) and will not allow the BLE Module to be inserted in the opposite direction.

Figure 3-5. Baseboard with J10 and J11 Headers to connect BLE Modules



To remove the BLE Modules from the BLE Pioneer kit, hold the BLE Pioneer kit in one hand and the BLE Module in the other, as shown in Figure 3-6, and pull it out in a rocking motion.

Figure 3-6. Remove BLE Module Connected on BLE Pioneer Kit

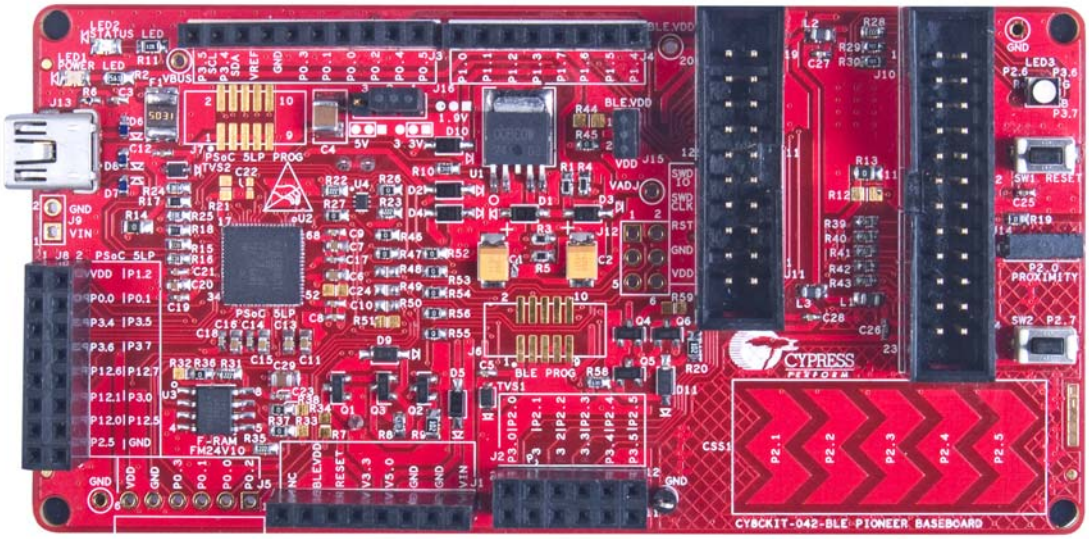


3.4 Programming and Debugging BLE Device

The kit can be programmed and debugged using the onboard PSoC 5LP programmer and debugger. Before programming the device, ensure that PSoC Creator and PSoC Programmer are installed on the PC. See the section Install Software on page 19 for more information.

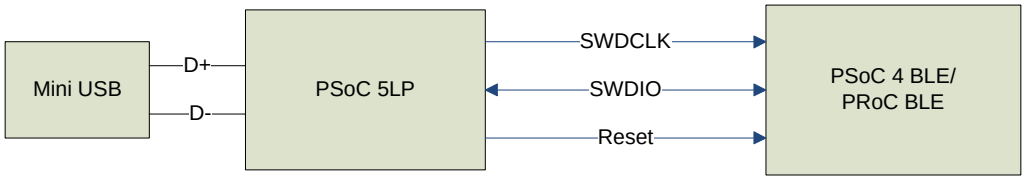
1. To program the device, plug the USB cable into the programming USB connector, J13, as shown in Figure 3-7. The kit will enumerate as a composite device.

Figure 3-7. Connect USB Cable to J13



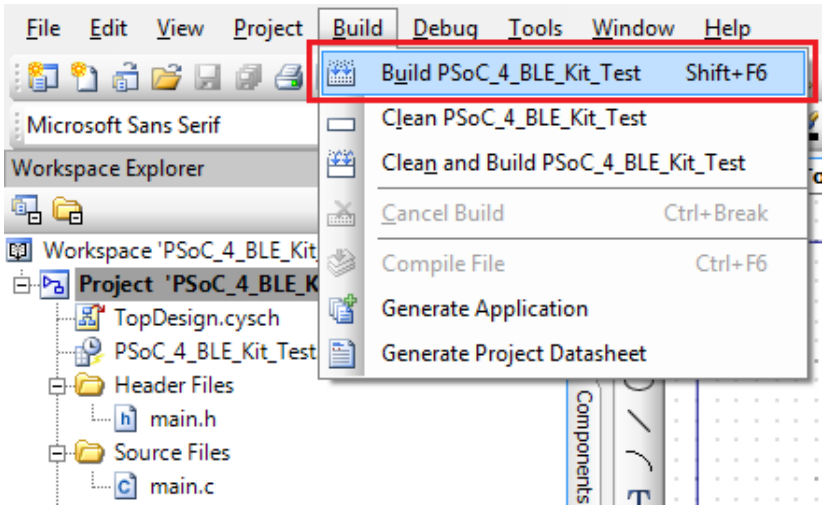
2. The onboard PSoC 5LP uses SWD to program the PSoC 4 BLE or PRoC BLE device. See Figure 3-8 for this implementation.

Figure 3-8. SWD Programming PSoC 4 BLE/PRoC BLE using PSoC 5LP



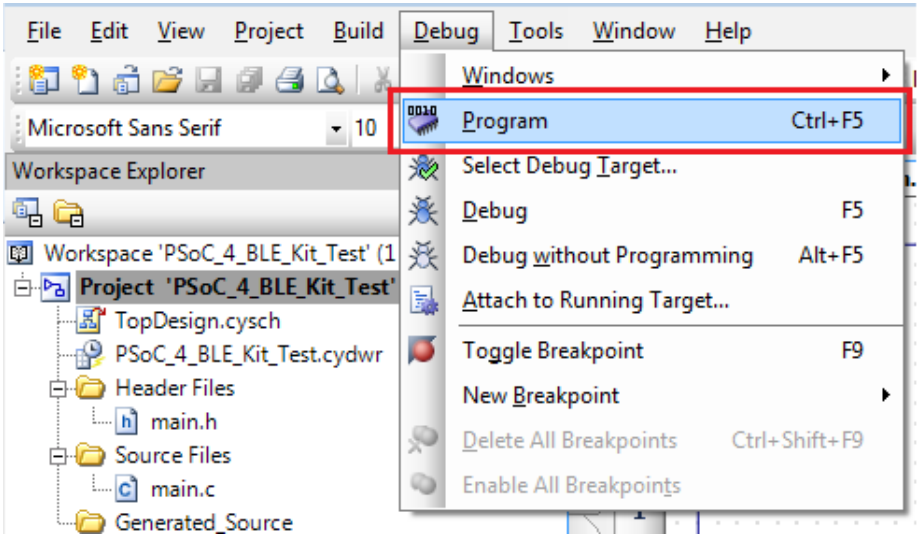
3. To load the desired example project, open PSoC Creator and go to **File > Open > Project/ Workspace**. This will provide the option to browse to and open your saved project.
4. Build the project by choosing **Build > Build <Project Name>** or **[Shift] [F6]**, as shown in Figure 3-9.

Figure 3-9. Build an Example Project



5. If there are no errors during build, program the firmware into the kit by choosing **Debug > Program** or pressing **[Ctrl] [F5]**, as shown in Figure 3-10. This will program the device on the BLE Pioneer Kit and the kit will be ready for use. For example, if the **PSoC_4_BLE_Kit_Test** project is programmed successfully to the PSoC 4 BLE Module on the BLE Pioneer kit, then you will observe the blue LED toggling every second. If debugging is needed on the project, go to step 6.

Figure 3-10. Programming Device From PSoC Creator



6. To debug the project using PSoC Creator, choose **Debug > Debug** or press **[F5]**.
7. When the project is built and programmed into the device on the BLE Pioneer Kit, PSoC Creator will enter the Debug mode; you can use it to debug your application. For more details on using the debug features, see the Cypress application note Getting Started with PSoC 4 BLE.

3.5 Dongle Connection

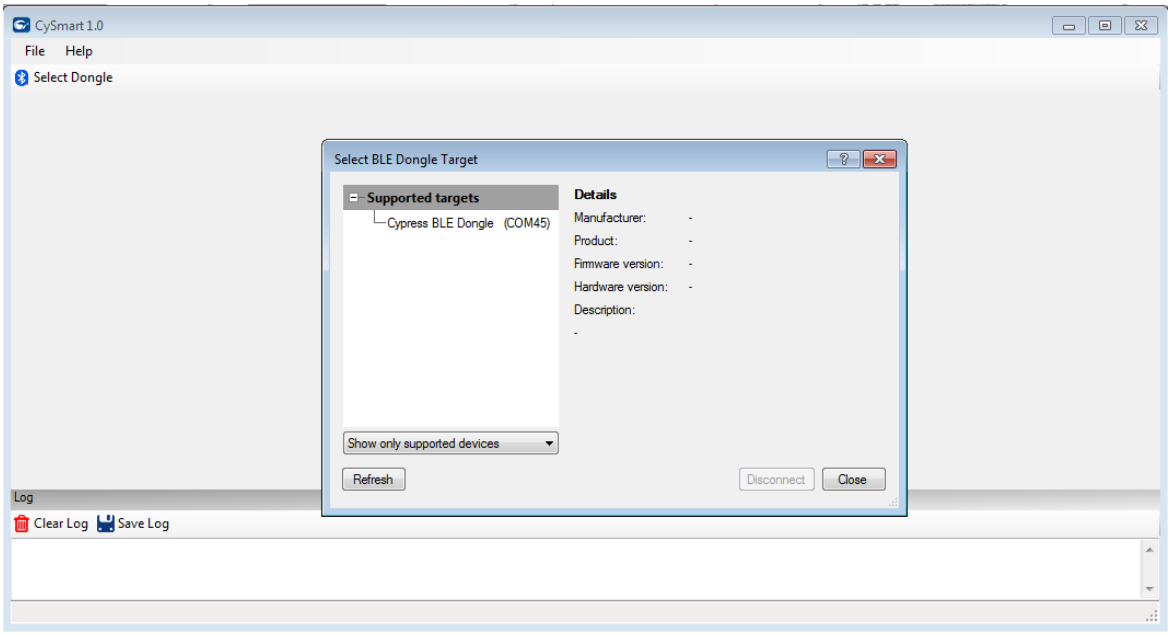
The Dongle, shown in Figure 3-11, provides the BLE Central device capability using the CySmart PC Tool (see CySmart PC Tool on page 158) on the PC. It helps in connecting and validating the example projects loaded on the BLE Pioneer Kit (Baseboard with one of the modules) through BLE. The CySmart PC Tool on the PC is the interface with which to configure the Dongle and analyze the BLE data transferred after connection with a BLE peripheral.

Figure 3-11. Dongle

After being connected to the PC through a USB port, the Dongle enumerates as a composite device, similar to the BLE Pioneer Kit. When enumerated, it allows similar features, such as programming/debugging of the onboard PRoC BLE, USB-UART bridge, and USB-I^{2}C bridge. Additionally, the interface is used to communicate with the CySmart PC Tool and emulate a BLE central device on PRoC BLE.

The Dongle works along with the CySmart PC Tool, as shown in Figure 3-12. The CySmart PC Tool is installed as part of the BLE Pioneer Kit installation. The CySmart PC Tool operation is explained in CySmart PC Tool on page 158.

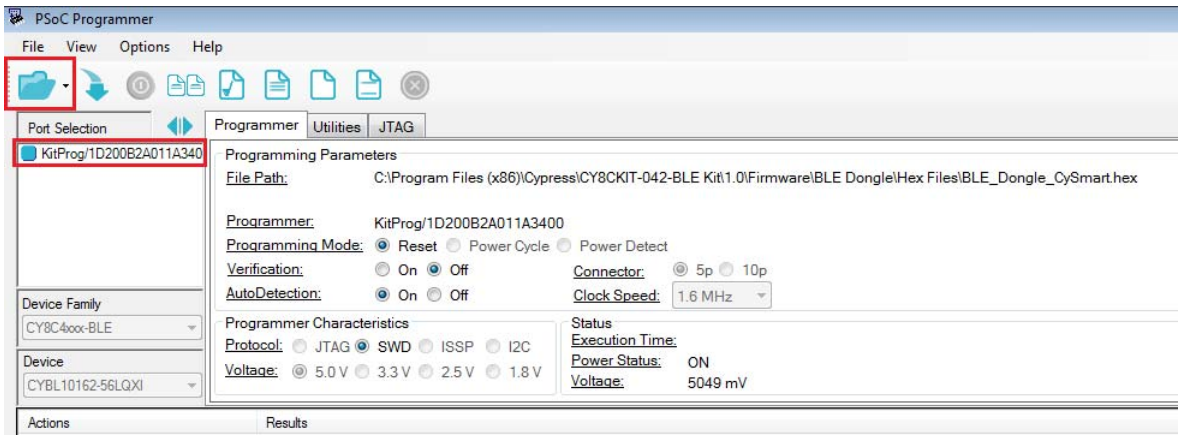
Figure 3-12. Dongle Interface on CySmart PC Tool



In case the Dongle contains custom firmware on PRoC BLE, the original CySmart firmware can be programmed back to the Dongle to restore the Dongle functionality. Follow these steps to do the same:

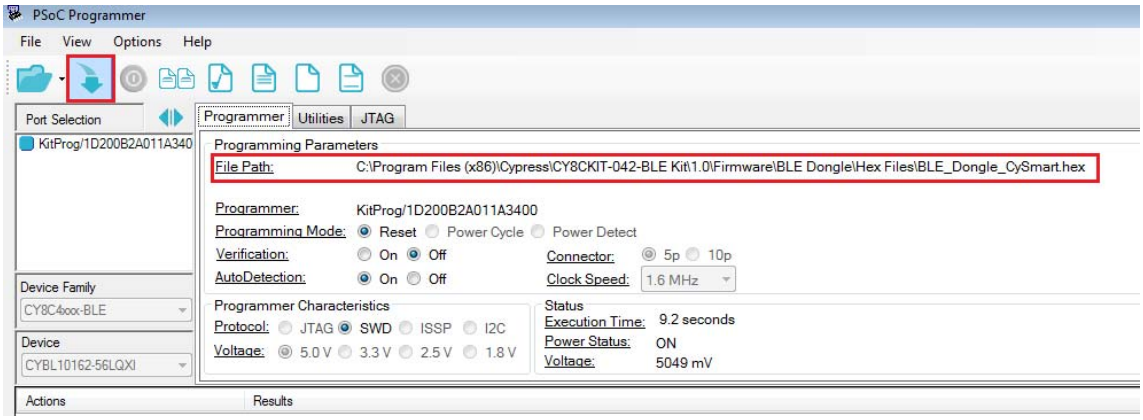
1. Connect the Dongle to the USB port on the PC.
2. Open PSoC Programmer 3.21.1 by going to **Start** > **All Programs** > **Cypress** > **PSoC Programmer 3.21.1** > **PSoC Programmer 3.21.1**.
3. When PSoC Programmer opens, it will automatically detect the KitProg on the Dongle. Click the **File Load** button and browse to the location of the *BLE_Dongle_CySmart.hex* file. The hex file is located at:
   `C:\Program Files (x86)\Cypress\`
   `CY8CKIT-042-BLE Kit\<version>\Firmware\BLE Dongle\Hex Files\`

Figure 3-13. Open Hex File



4. Ensure the other setting match as shown in Figure 3-13. Click the **Program** button to start programming. The status bar at the bottom of the PSoC Programmer window will show the programming status and the result (Successful/Failed).

Figure 3-14. Programming Hex File to Dongle



5. After programming is completed successfully, the Dongle firmware is updated and can be used to connect to the CySmart PC tool.

3.6 USB-UART Bridge

The onboard PSoC 5LP on both the Baseboard and Dongle also acts as a USB-UART bridge to transfer and receive data from the PSoC 4 BLE or PRoC BLE device to the PC via the COM terminal software. When the USB mini-B cable is connected to J13 of the Baseboard or the Dongle is connected to the PC, a device named **"KitProg USB-UART"** is available under **Ports (COM & LPT)** in the Device Manager. For more details about the USB-UART functionality, see Using PSoC 5LP as USB-UART Bridge on page 113.

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

For both the Baseboard and the Dongle, the UART lines are hardwired onboard between the PSoC 5LP and BLE Modules. No external UART connection between the two devices is needed. Simply place the UART component in the PSoC 4 BLE or PRoC BLE and assign the UART pins as shown in Table 3-2.

Table 3-2. UART Pin Assignment in BLE Devices for USB-UART Bridge

Pin	BLE Pioneer Kit	BLE Dongle
UART_RX	P1_4	P1_4
UART_TX	P1_5	P1_5

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

Table 3-3. 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)

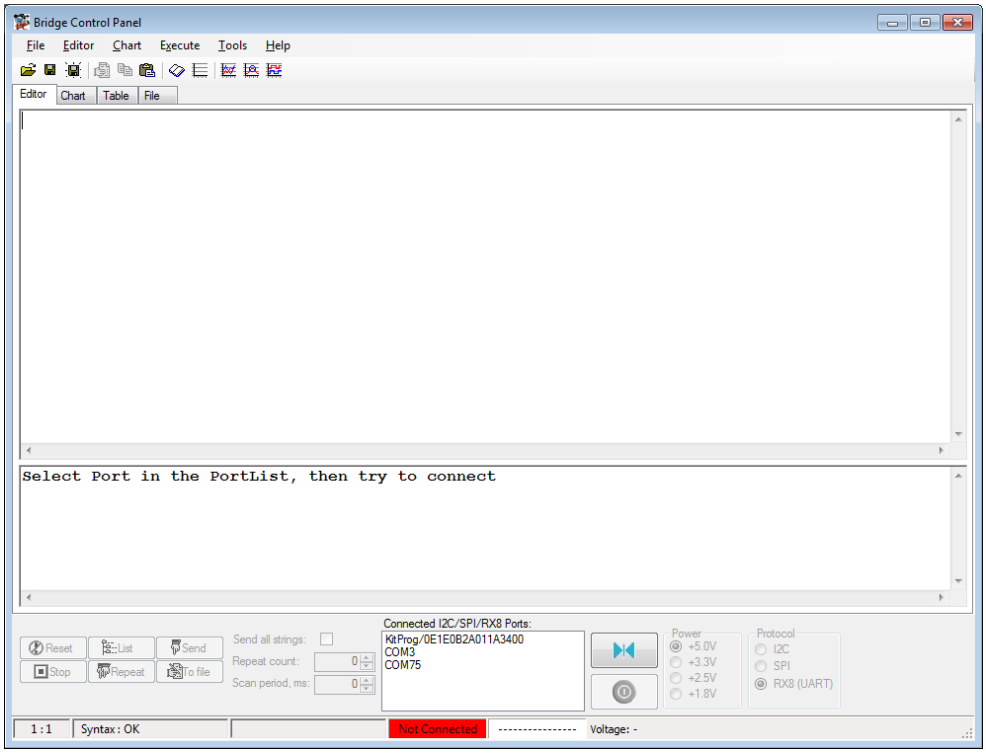
3.7 USB-I^{2}C Bridge

The PSoC 5LP also functions as a USB-I^{2}C bridge. In this role, PSoC 5LP communicates with PSoC 4 BLE/PRoC BLE using an I^{2}C interface, and sends that data over the USB to the USB-I^{2}C software utility running on the PC, called the **Bridge Control Panel** (BCP). This feature is available for both the BLE Pioneer Kit and the Dongle.

The BCP is available as part of the PSoC Programmer installation. This software can be used to send and receive USB-I^{2}C data from the PSoC 5LP. When the USB mini-B cable is connected to header J13 on the BLE Pioneer Kit or when the Dongle is connected to the PC, the **KitProg USB-I^{2}C** is available under **Connected I2C/SPI/RX8 Ports** in the BCP, as shown in Figure 3-15.

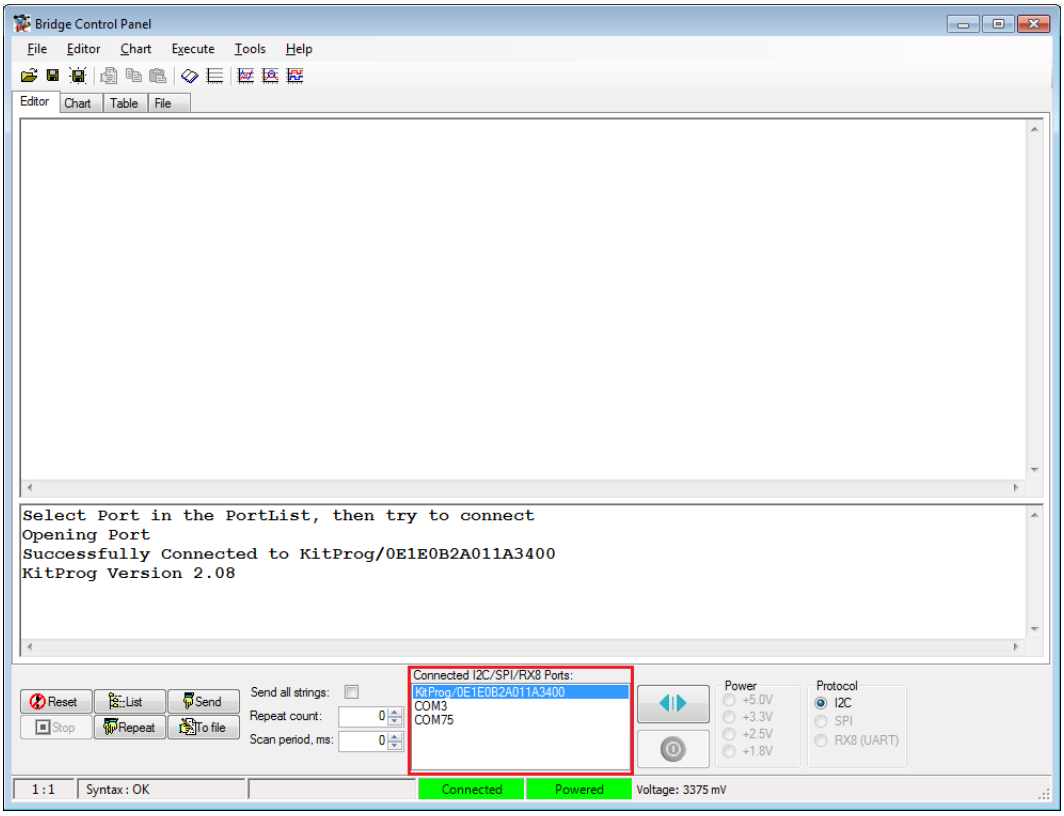
To open BCP in your system, go to **Start > All Programs > Cypress > Bridge Control Panel**.

Figure 3-15. Bridge Control Panel



To use the USB-I^{2}C functionality, select the **KitProg** USB-I^{2}C in the BCP. On successful connection, the **Connected** and **Powered** tabs turn green, as shown in Figure 3-16.

Figure 3-16. KitProg USB-I^{2}C Connected in Bridge Control Panel



USB-I^{2}C is implemented using the USB and I^{2}C components of PSoC 5LP. For the BLE Pioneer Kit, the SCL (P12_0) and SDA (P12_1) lines from the PSoC 5LP are connected to the SCL (P3_5) and SDA (P3_4) lines of the BLE Module header. For the Dongle, the SCL (P12_0) and SDA (P12_1) lines from the PSoC 5LP are connected to the SCL (P3_5) and SDA (P3_4) lines. The USB-I^{2}C bridge currently supports I^{2}C speed of 50 kHz, 100 kHz, 400 kHz, and 1 MHz.

See Using PSoC 5LP as USB-I2C Bridge on page 124 for building a project that uses the USB-I^{2}C bridge functionality.

3.8 Updating the Onboard Programmer Firmware

The BLE Pioneer kit and Dongle contains the modified KitProg that is required to reliably use the BLE Pioneer kit's functionality. Do not update the KitProg firmware on PSoC 5LP on this kit. If the KitProg has been updated with the firmware provided with PSoC Programmer, then you can restore the original KitProg for this kit using the Bootloader Host tool. The required CYACD file is present in the installed location of the kit:

```
<Install_Directory>\CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\KitProg
\KitProg.cyacd
```

Follow the steps in the Advanced Topics chapter on page 113 to restore the KitProg using the Bootloader Host tool.

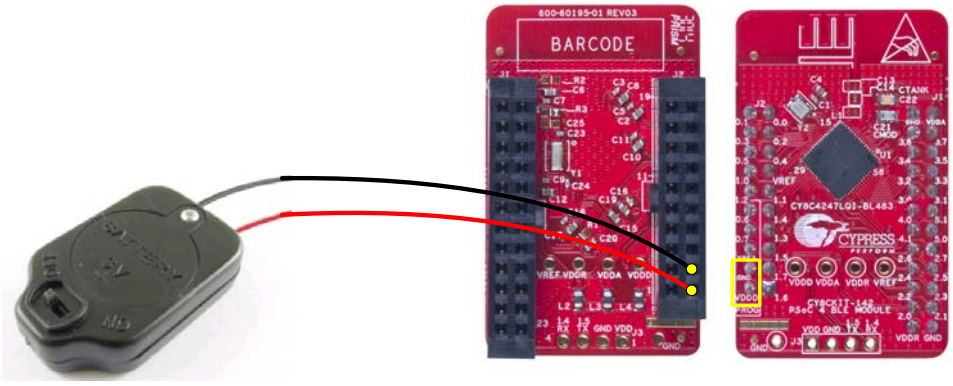
3.9 Measure Coin-cell Power Consumption

To measure the power consumption of a project with coin-cell battery, connect the coin-cell battery directly to the BLE Modules as shown in the Figure 3-17. The baseboard is designed with additional circuits to protect the PSoC 4 BLE/PRoC BLE device and the F-RAM in Arduino environment. Note that the power consumption measurements on the baseboard will also include the power consumed by these additional circuits.

After you have programmed your application on the CY8CKIT-142 PSoC® 4 BLE Module or the CY5671 PRoC™ BLE Module, remove the BLE Module from the baseboard and connect the coin-cell battery (Figure 3-17). This setup enables an accurate power consumption measurement for the application. The other pins on the BLE Module can be used to build the desired application.

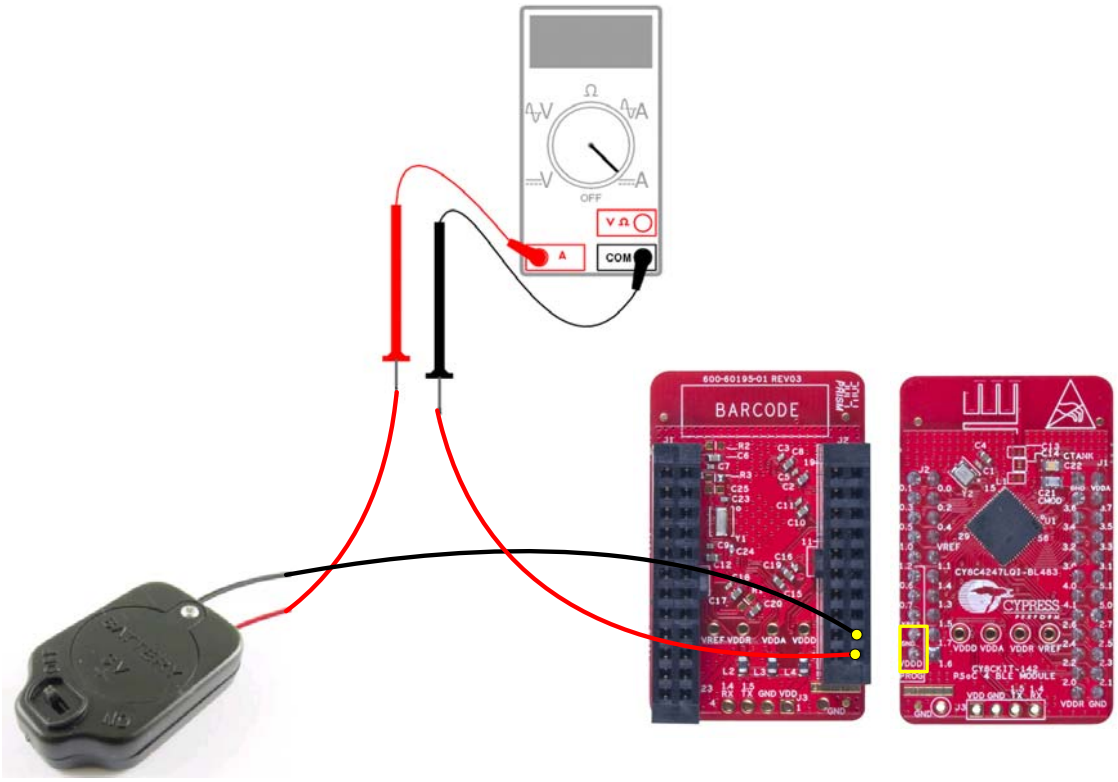
Connect the positive terminal of the coin cell to pin J2.2 and negative terminal to pin J2.4 using wires.

Figure 3-17. Powering the BLE Module using a Coin-Cell Battery



Connect an ammeter in series with the battery to measure the power consumption as shown in Figure 3-18.

Figure 3-18. Current Measurement of BLE Module when Powered from a Coin-cell Battery



4. Example Projects

This chapter demonstrates the functionality of the PSoC 4 BLE and PRoC BLE devices using the BLE Pioneer kit-based example projects. To access these example projects, download and install the kit setup file from the kit web page. The example projects are available in the firmware folder in the installation location.

The Cypress BLE device comes in two variants:

- PSoC 4 BLE: The PSoC 4 BLE product family is the new wireless member of the PSoC 4 architecture platform. The family provides a full programmable analog and digital system and a complete schematic view of PSoC Creator. The PSoC 4 BLE family provides a 32-bit ARM Cortex-M0 based MCU subsystem with programmable analog and digital peripherals, such as universal digital blocks (UDBs), 12-bit SAR ADC, Op-amp, LP comparator, IDACs, UART, I^2C, SPI, and timer/counter/PWM block. It also has a dedicated CapSense block (in select part numbers) to implement the touch-sensing solution, with a practical system SNR of 100:1.
- PRoC BLE: This family provides prebuilt part numbers for applications such as human interface devices (HID), remote control, trackpad, and toys. PRoC BLE also supports up to two-finger gestures for trackpad and remote control applications. The PRoC BLE product family enriches PRoC wireless capacitive touch devices with the Bluetooth Low Energy protocol. The PRoC BLE family has embedded gestures (in select part numbers) to implement the touch-sensing solution for trackpad implementation. It also provides a 32-bit ARM Cortex-M0 based MCU subsystem with analog and digital peripherals, such as 12-bit SAR ADC, UART, I^2C, PWMs, and timer/counter/PWM blocks. The family uses a special PSoC Creator schematic view for easy configuration of PRoC BLE devices.

The example projects that are part of this kit installer include projects for both these devices. Unless otherwise indicated, the functionality of the example projects is the same for PSoC 4 BLE and PRoC BLE. The mode of use and the PSoC Creator schematic view will vary slightly for each device.

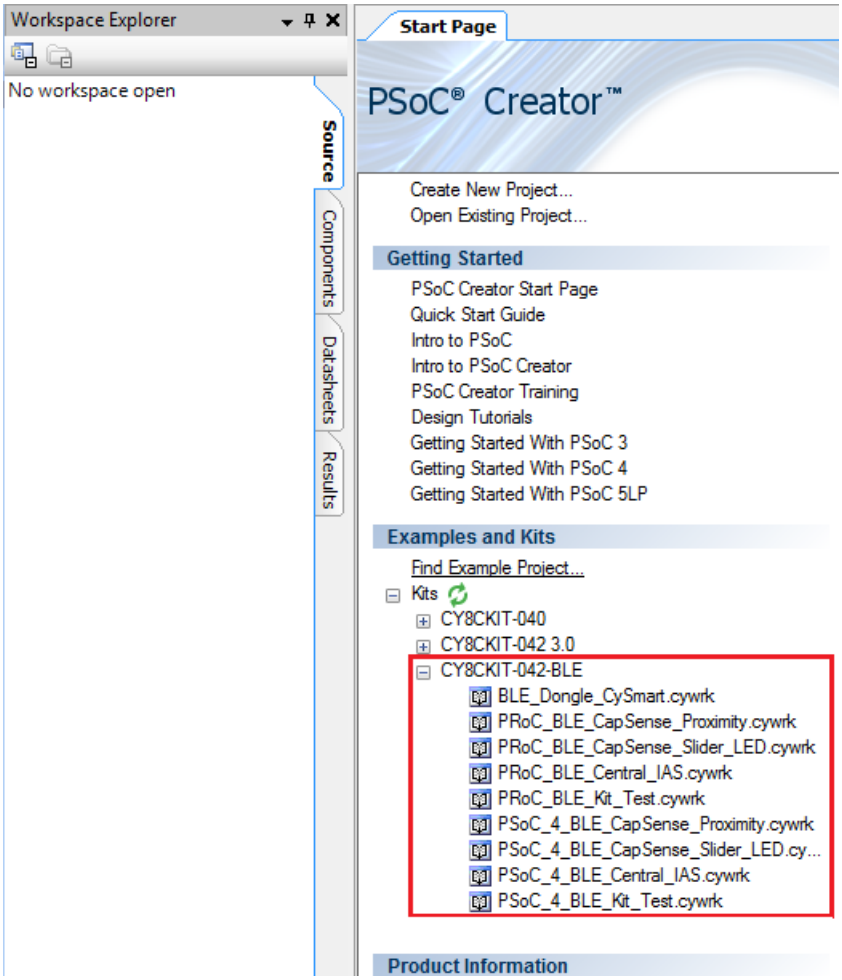
4.1 Using Example Projects

Follow these steps to open and use the example projects:

1. Launch PSoC Creator from **Start > All Programs > Cypress > PSoC Creator 3.1 > PSoC Creator 3.1**.

2. On the Start Page, choose **Examples and Kits > Kits > CY8CKIT-042-BLE**. A list of example projects appears, as shown in Figure 4-1. Projects named with the prefix '*PSoC_4_BLE_*' work on the BLE Pioneer kit with the PSoC 4 BLE Module; projects named with the prefix '*PRoC_BLE_*' work on the BLE Pioneer kit with the PRoC BLE Module.
3. Click on the desired example project.

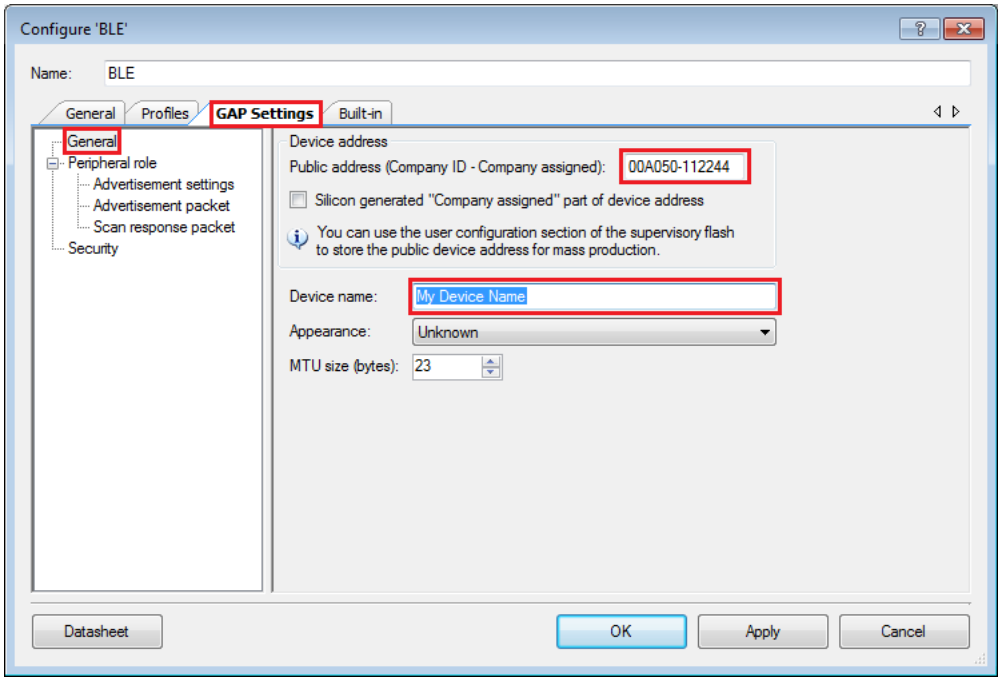
Figure 4-1. Open Example Project from PSoC Creator



4. Select the folder where you want to save the project and click **OK**.

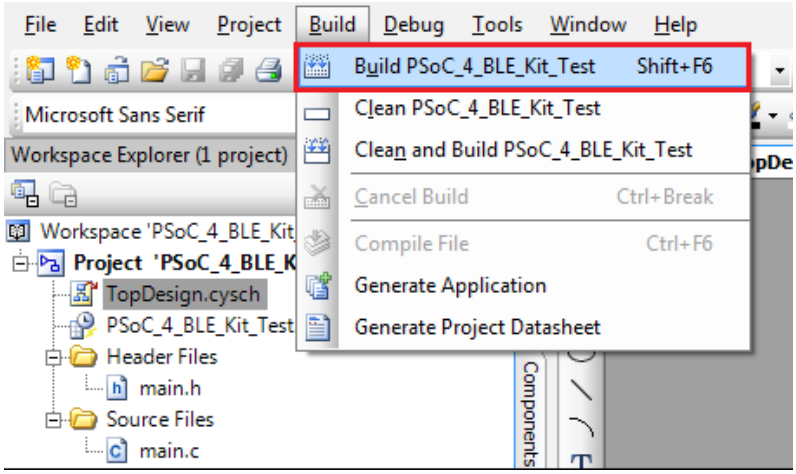
5. BLE projects use a public device address set in the BLE component GUI to advertise and scan, depending on the role: peripheral or central mode. If there are other kits in close proximity, which have the same public device address, then wrong devices may be connected or connections can fail. To prevent this, you can change the **Public device address** (and preferably **Device name**) in the BLE component. To do this, double-click the BLE component in TopDesign, go to the **GAP Settings** tab, and choose the **General** setting. Add the desired public device address (non-zero) and device name in the respective fields, as shown in Figure 4-2. Click **OK**.

Figure 4-2. Change BLE Public Device Address and Name



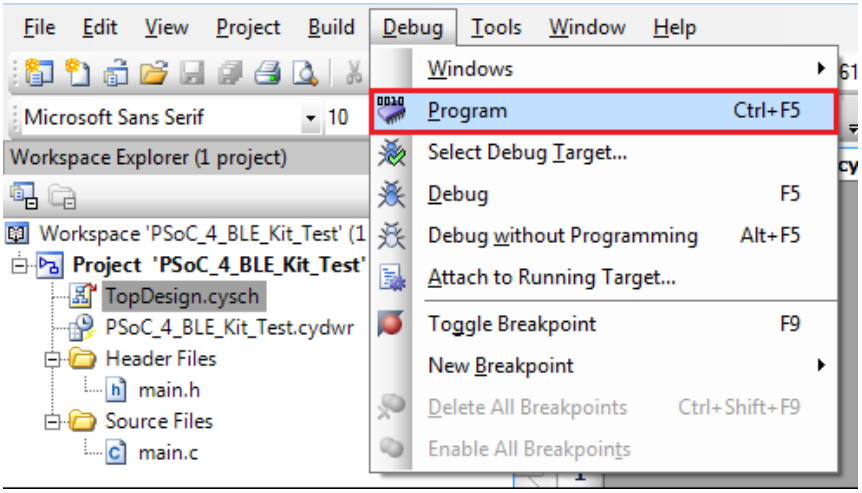
6. Build the example project by choosing **Build > Build <Project Name>**, as shown in Figure 4-3. A hex file will be generated.

Figure 4-3. Build Project from PSoC Creator



7. To program the kit with this example project, connect the Baseboard to the PC by plugging it into the USB mini-B connector (J13) on the baseboard, as described in BLE Pioneer Kit USB Connection on page 25. Ensure that the correct BLE Module is placed on the baseboard, depending on the project opened.
8. Choose **Debug > Program** in PSoC Creator, as shown in Figure 4-4.

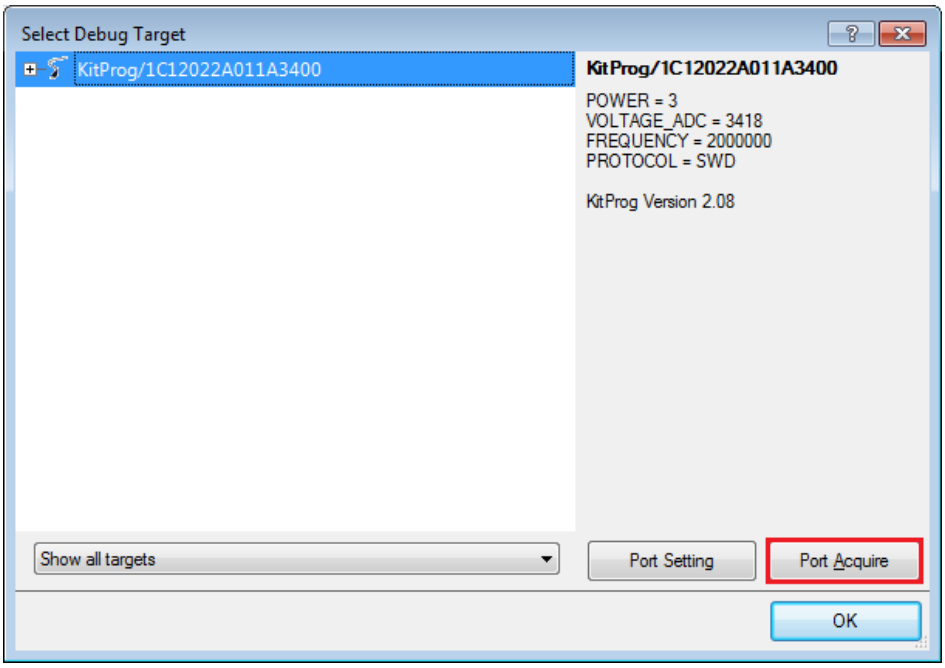
Figure 4-4. Program Device in PSoC Creator



9. If the device is not yet acquired, PSoC Creator will open the programming window. Select **KitProg** and click the **Port Acquire** button, as shown in Figure 4-5.

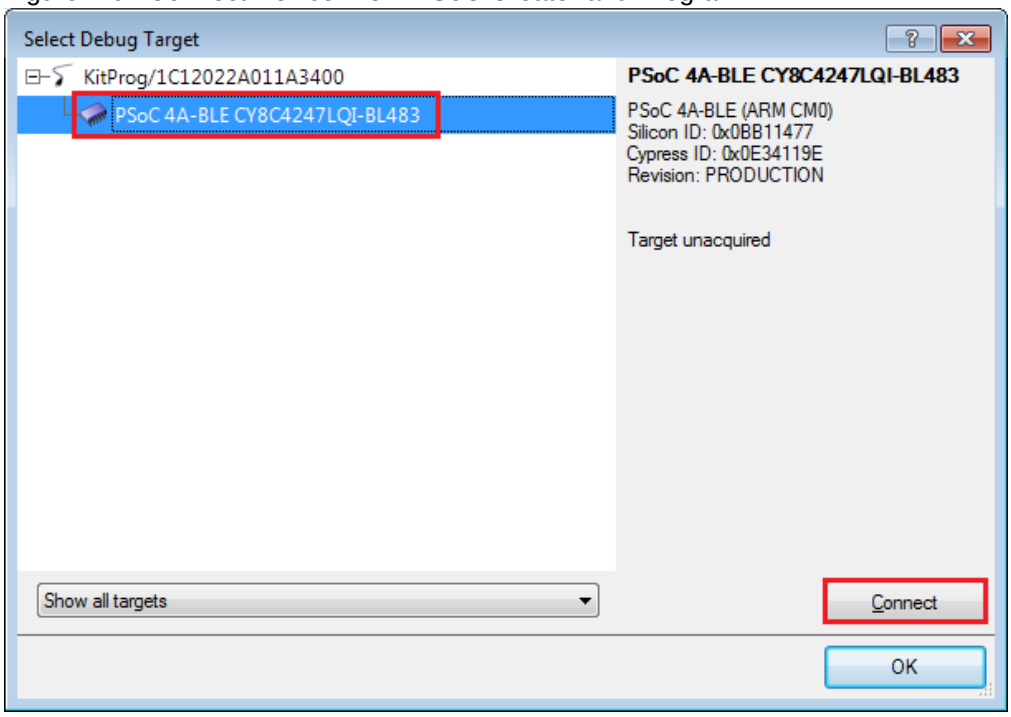
   Note: The string following the 'KitProg' is the serial ID for the programmer on the kit. Each kit will have their unique serial ID. If various kits are connected to the same system, the serial ID can be used to select the correct kit to program the firmware. Additionally, the serial ID starting with 'BLE' belongs to the Dongle (see Dongle Connection on page 29) and provides visual confirmation for Dongles connected to the system.

Figure 4-5. Port Acquire



10. After the device is acquired, it is shown in a tree structure below the KitProg. Click the **Connect** button and then **OK** to exit the window and start programming, as shown in Figure 4-6.

Figure 4-6. Connect Device From PSoC Creator and Program



Note: As stated previously, the BLE Pioneer Kit supports both Cypress BLE devices: PSoC 4 BLE and PRoC BLE. Thus, there are two versions of each of the kit example projects demonstrating the same functionality. Projects named with the prefix 'PSoC_4_BLE_' work with the PSoC 4 BLE Module placed on the Baseboard. Projects named with the prefix 'PRoC_BLE_' work with the PRoC BLE Module placed on the Baseboard. Ensure that the correct module is placed on the Baseboard before programming the device with the corresponding kit example projects.

The description, hardware configurations, and verification method of the kit example projects explained in the following sections are valid for both PSoC 4 BLE and PRoC BLE devices. Unless explicitly mentioned, the theory and usability for these example projects should be considered same for both the modules/devices.

This document refers to the BLE Pioneer kits, Dongle, and PC/Mobile as BLE central or peripheral devices. A BLE central device is normally the master and requests/commands data from peripheral device. BLE-enabled phones and PCs are one such example. BLE peripheral devices, on the other hand, store the actual data and send it to central devices when requested. Examples include BLE-enabled sensors, proximity beacons, and so on.

4.2 Kit Test

4.2.1 Project Description

This example project is a non-BLE project that is meant to validate the BLE Pioneer Kit, if desired. This example project verifies whether the onboard device is working properly. It uses one timer that periodically generates an interrupt and causes toggling of the onboard LED (Figure 4-7).

Two projects demonstrate this functionality on two different devices:

- **PSoC_4_BLE_Kit_Test** works with the PSoC 4 BLE Module.
- **PRoC_BLE_Kit_Test** works with PRoC BLE Module.

Figure 4-7. TopDesign for PSoC_4_BLE_Kit_Test

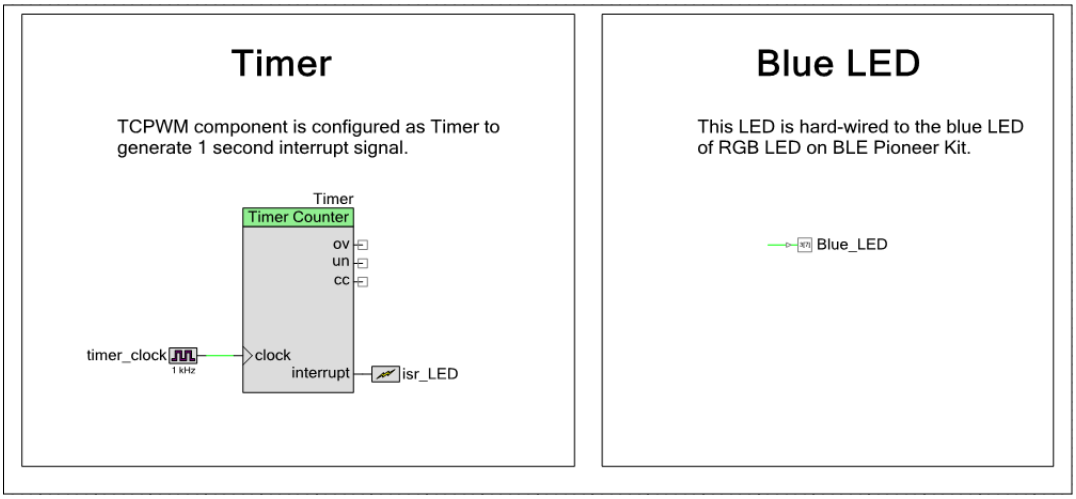
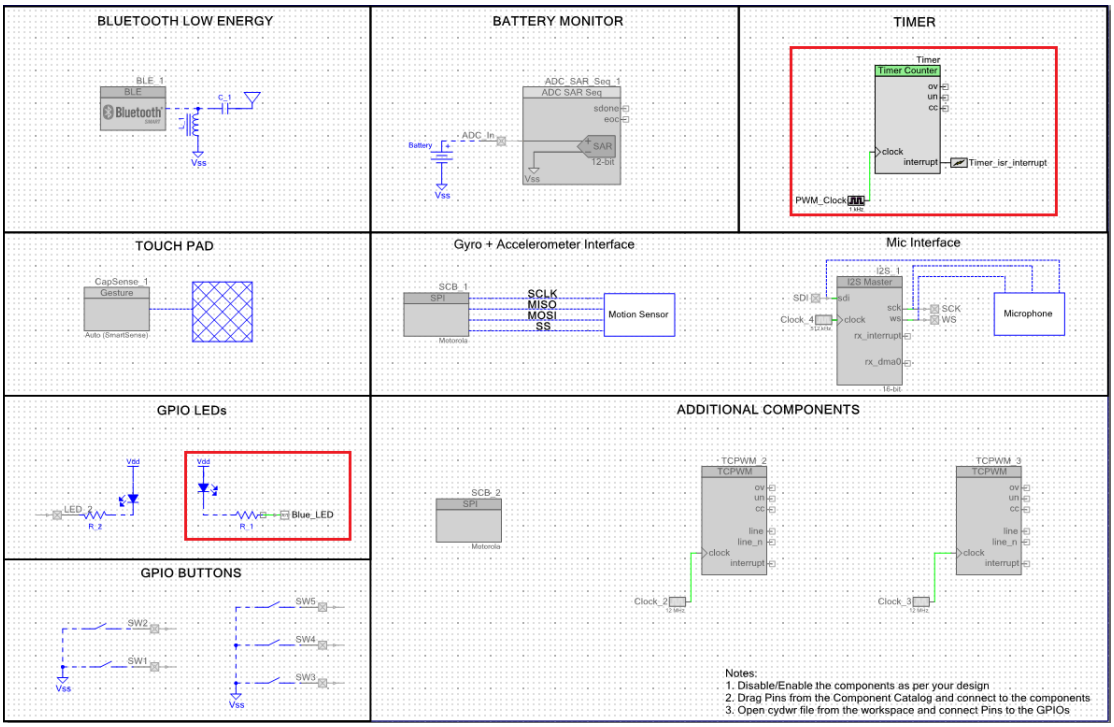


Figure 4-8. Top Design for PRoC_BLE_Kit_Test



4.2.2 Hardware Connections

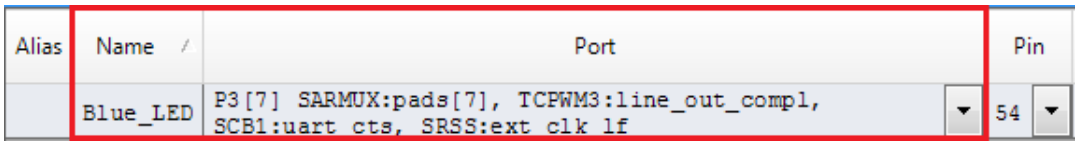
No specific hardware connections are required for this project because all connections are hardwired on the board. On the Workspace Explorer, double-click **PSoC_4_BLE_Kit_Test.cydwr/ PRoC_BLE_Kit_Test.cydwr** on the left and select the pins shown in Figure 4-9.

Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC_4_BLE_Kit_Test works with the PSoC 4 BLE Module (red BLE Module) and PRoC_BLE_Kit_Test works with the PRoC BLE Module (black BLE Module).

Table 4-1. Pin Assignment for Test Project

Pin Name	Port Name
Blue_LED	P3_7

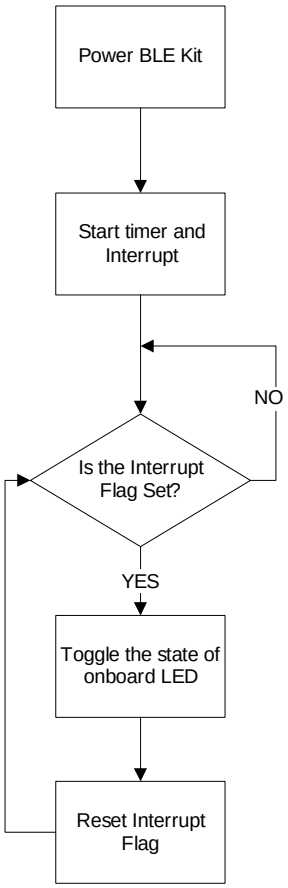
Figure 4-9. Pin Selection for Test Project



4.2.3 Flow Chart

Figure 4-10 shows the flow chart of code implemented in the main.c file.

Figure 4-10. Test Project Flow Chart



4.2.4 Verify Output

The project is intended to validate the kit operation. See Using Example Projects on page 37 for steps to program the kit with the project. After powering the kit and successfully programming it with the test project, the blue LED of the onboard RGB LED should periodically toggle (Figure 4-11).

Figure 4-11. Toggling LED on BLE Pioneer Kit with PSoC 4 BLE Module

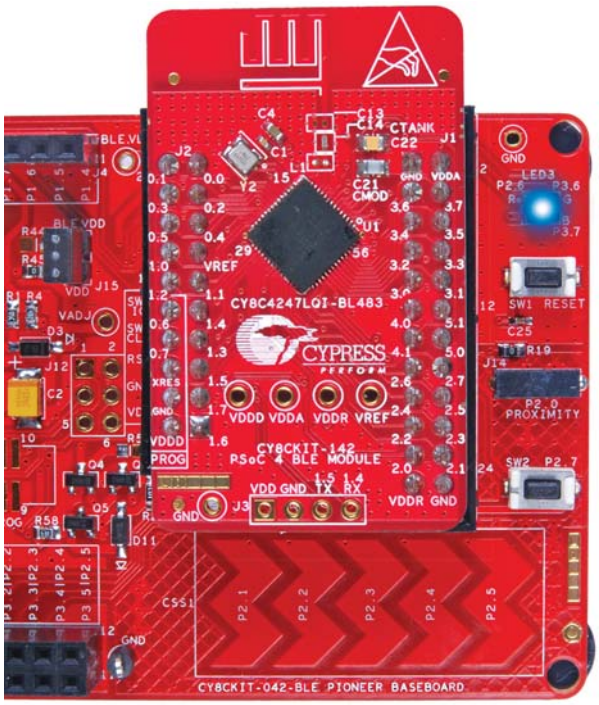
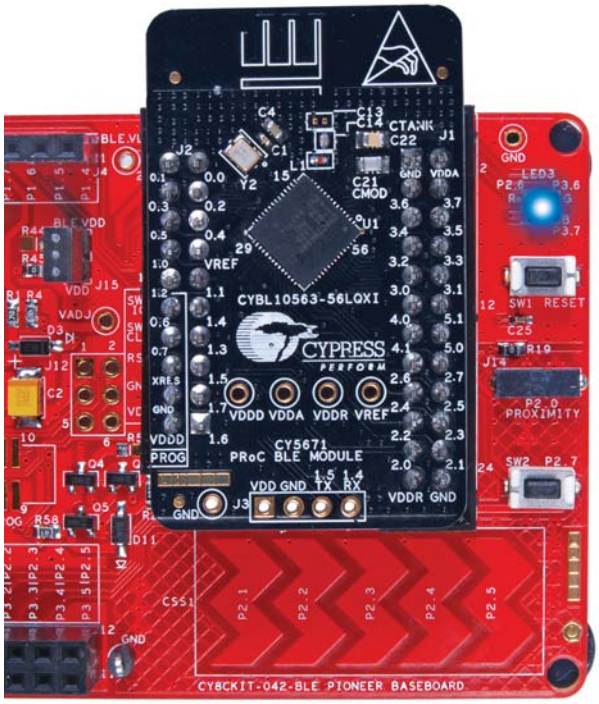


Figure 4-12. Toggling LED on BLE Pioneer Kit with PRoC BLE Module



4.3 CapSense Slider and LED

4.3.1 Project Description

This example project demonstrates bidirectional data transfer between a BLE central device (Dongle or CySmart App) and the kit working as a peripheral with the CapSense slider and RGB LED functionalities. The example project allows advertisement, connection, and data transfer over a custom BLE profile. It uses a CapSense component to read finger location on the CapSense slider and sends the value to a connected BLE central device. The project receives RGB color and intensity values from the BLE central device, which are then translated into the proper color and intensity on the onboard RGB LED.

The project also supports placing the system into sleep in the event of advertising timeout or BLE disconnection. The system then wakes up and resumes when the user button (**SW2**) on the kit is pressed.

This is the default firmware that comes in BLE Modules shipped with the kit.

Two projects demonstrate this functionality on two different devices:

- **PSoC_4_BLE_CapSense_Slider_LED** works with the PSoC 4 BLE Module
- **PRoC_BLE_CapSense_Slider_LED** works with the PRoC BLE Module.

The PSoC 4 BLE project implements RGB color and intensity control using the PRiSM component whereas the PRoC BLE uses the software implementation of the PRiSM mode.

Figure 4-13. TopDesign for PSoC_4_BLE_CapSense_Slider_LED Project

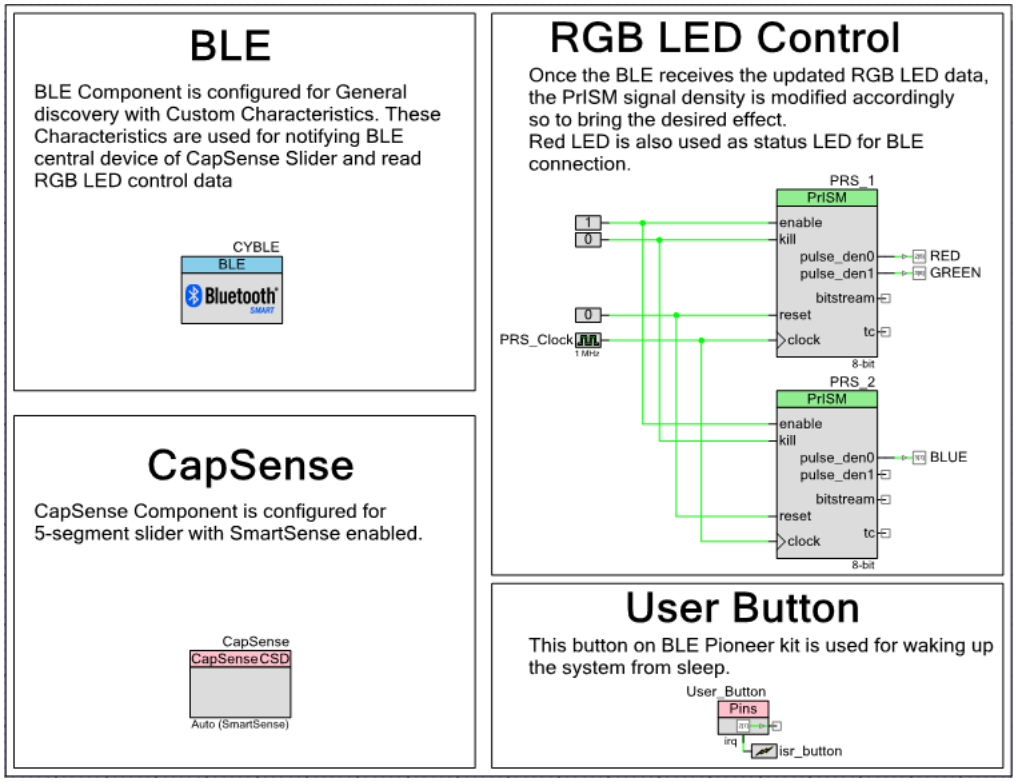
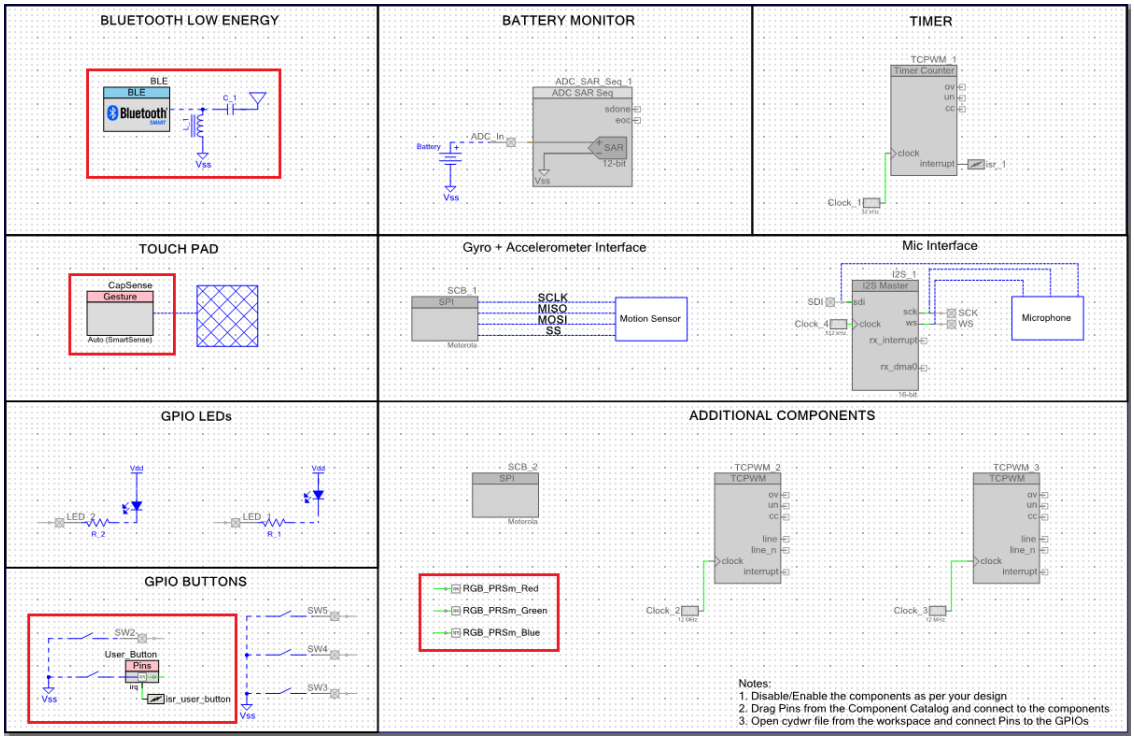


Figure 4-14. TopDesign for PRoC_BLE_CapSense_Slider_LED Project



4.3.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the board. Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC_4_BLE_CapSense_Slider_LED works with the PSoC 4 BLE Module. PRoC_BLE_CapSense_Slider_LED works with the PRoC BLE Module.

On the Workspace Explorer, double-click on **PSoC_4_BLE_CapSense_Slider_LED.cydwr/ PRoC_BLE_CapSense_Slider_LED.cydwr** on the left, and select the pins shown in Figure 4-15.

Table 4-2. Pin Assignments for CapSense Slider and LED Project

Pin Name	Port Name
CapSense CMOD	P4_0
CapSense Slider 1	P2_1
CapSense Slider 2	P2_2
CapSense Slider 3	P2_3
CapSense Slider 4	P2_4
CapSense Slider 5	P2_5
BLUE	P3_7
GREEN	P3_6
RED	P2_6
User_Button	P2_7

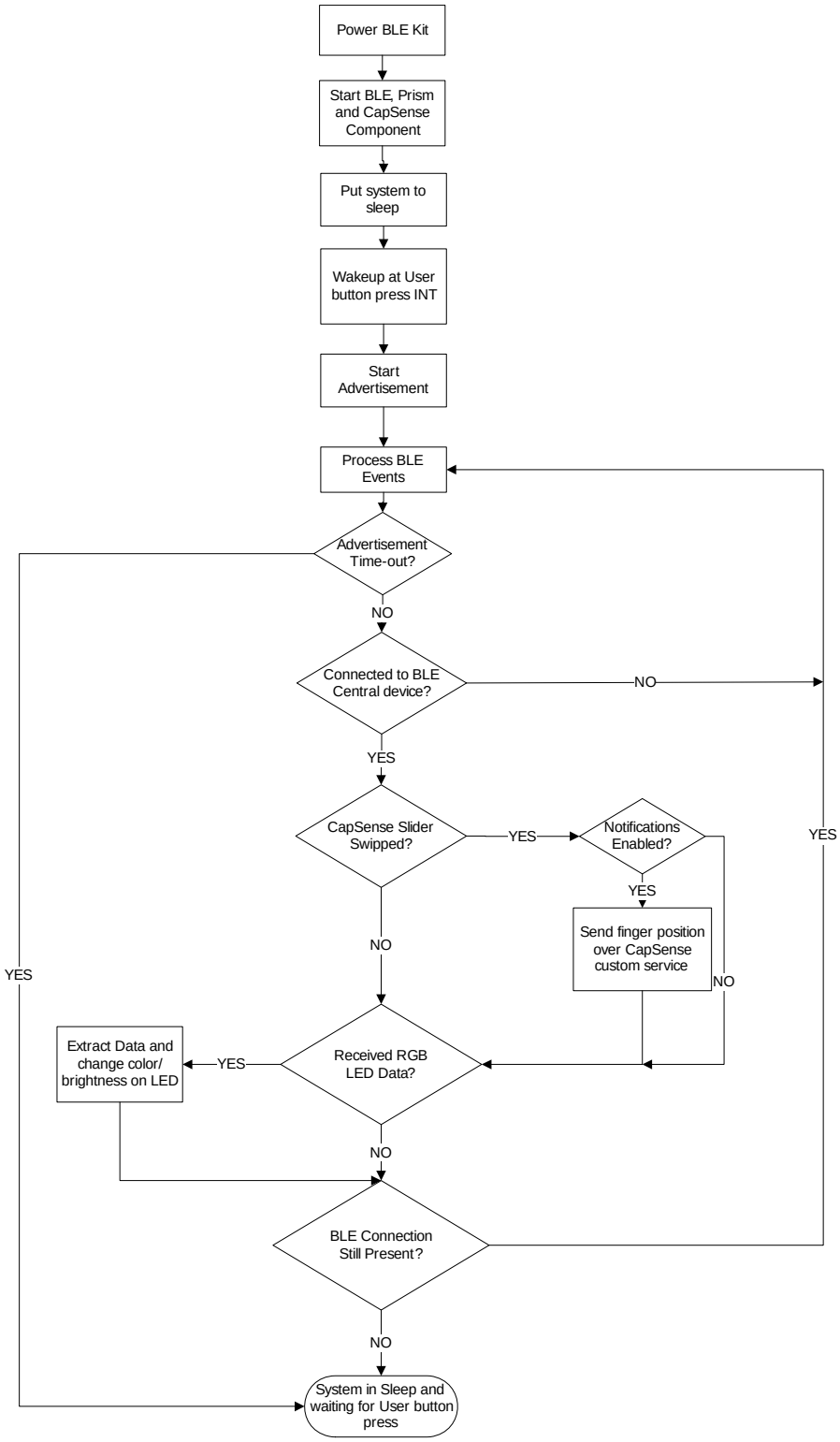
Figure 4-15. Pin Selection for CapSense Slider and LED Project

Alias	Name	Port	Pin	Lock
Cmod	\CapSense:Cmod\	P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi	5	☑
LinearSlider0_e0__LS	\CapSense:Sns[0]\	P2[1] OA0:vminus, SCB0:spi_select[2]	38	☑
LinearSlider0_e1__LS	\CapSense:Sns[1]\	P2[2] OA0:vout_10x, SRSS:wakeup, SCB0:spi select[3]	39	☑
LinearSlider0_e2__LS	\CapSense:Sns[2]\	P2[3] OA1:vout_10x, SRSS:ext_clk_lf	40	☑
LinearSlider0_e3__LS	\CapSense:Sns[3]\	P2[4] OA1:vminus	41	☑
LinearSlider0_e4__LS	\CapSense:Sns[4]\	P2[5] OA1:vplus	42	☑
	BLUE	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf	54	☑
	GREEN	P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart rts	53	☑
	RED	P2[6] OA0:vplus_alt	43	☑
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	44	☑

4.3.3 Flow Chart

Figure 4-16 shows the flow chart of code implemented.

Figure 4-16. CapSense Slider and LED Project Flow Chart



4.3.4 Verify Output

The project can be verified by two methods: one using the CySmart PC Tool and Dongle, and the other using the CySmart iOS BLE app.

To install and use the CySmart PC Tool, see CySmart PC Tool on page 158.

To install and use the CySmart iOS app, see CySmart iOS Application on page 150.

4.3.4.1 CySmart PC Tool

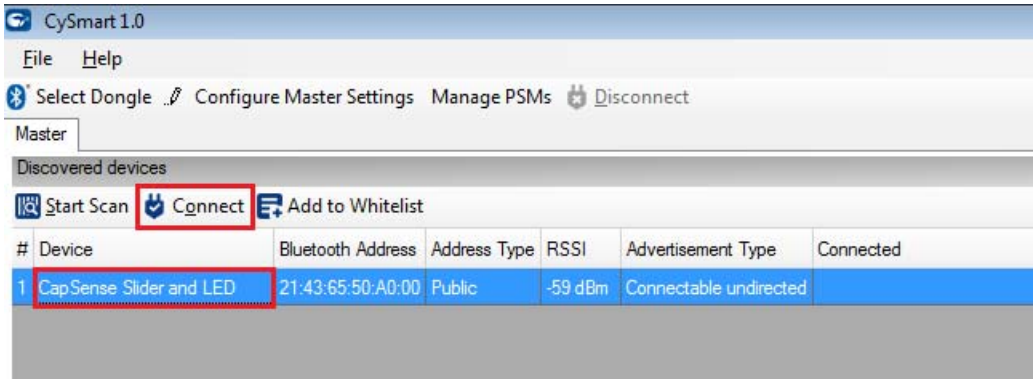
To verify the CapSense and LED project using the CySmart PC tool, follow these steps:

Note: See CySmart PC Tool on page 158 to learn how to use the tool.

1. Connect the Dongle to one of the USB ports on the PC.
2. Start CySmart PC Tool on the PC by going to **Start > All Programs > Cypress > CySmart 1.0 > CySmart**. You will see a list of Dongles connected to it. If no Dongle is found, click **Refresh**. Select the Dongle and click **Connect**.
3. Place the desired BLE Module, either PSoC 4 BLE or PRoC BLE, on the BLE Pioneer kit, depending on the project chosen.
4. Power the BLE Pioneer kit through the USB connector J13.
5. Program the BLE Pioneer kit with the CapSense and LED example projects. Follow steps in Using Example Projects on page 37 to program the device.
6. After programming successfully, press the user button (**SW2**) on the BLE Pioneer Kit to start advertisement. Advertisement is indicated by a blinking red LED on the Baseboard.
7. Click **Start Scan** to see the list of available BLE peripheral devices. You can confirm your device by the device name and Bluetooth address in the list. This should be same as the one configured in the BLE component GUI.

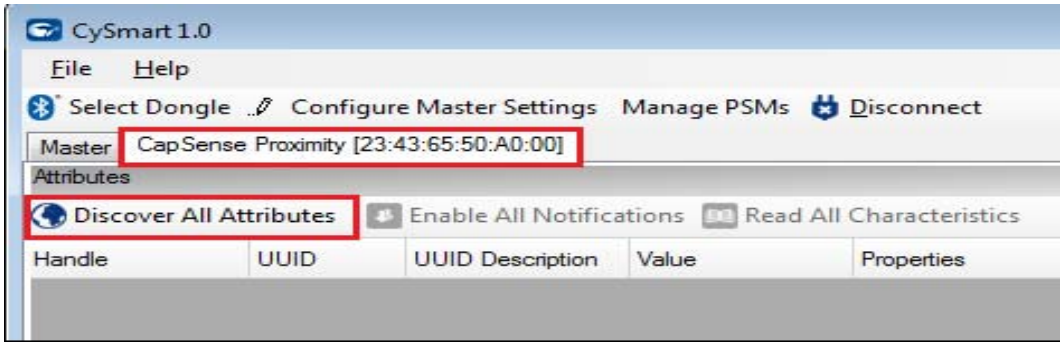
8. After the available devices are listed, double-click the **CapSense Slider and LED** device to connect or click **Stop Scan** and then click **Connect**.

Figure 4-17. Connect to CapSense Slider and LED Peripheral



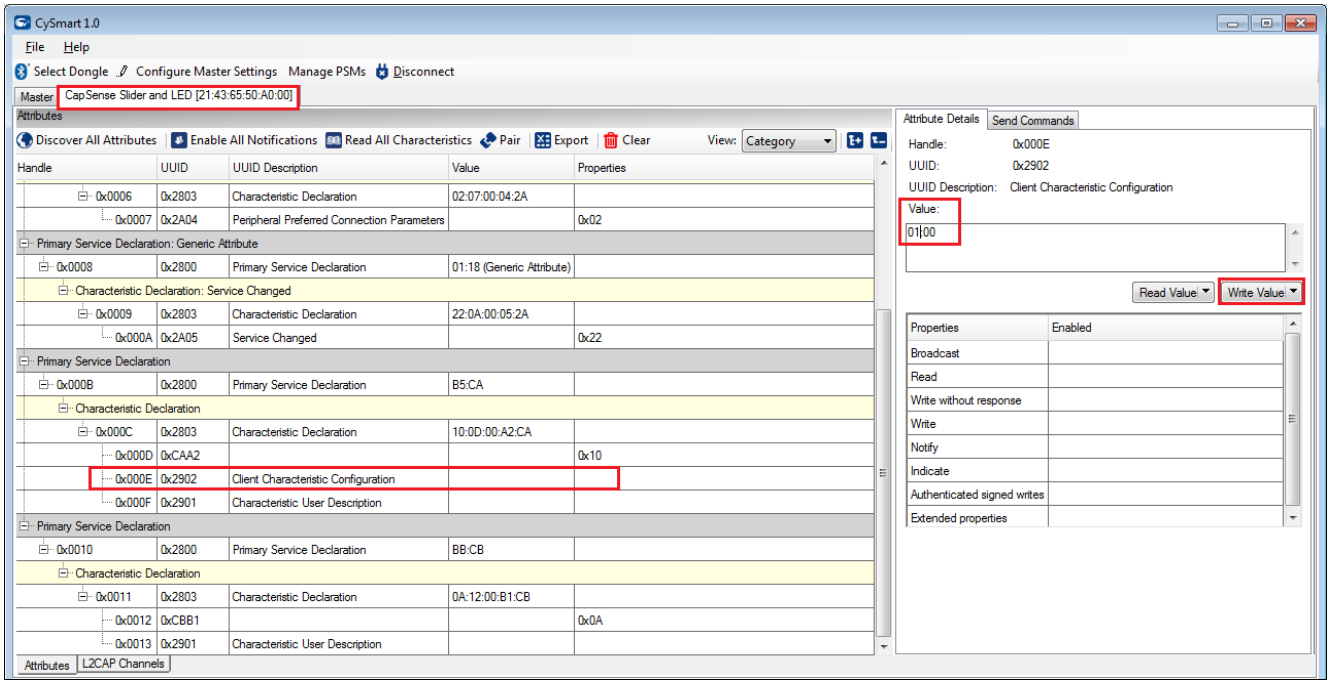
9. If the connection is successful, you will see another tab opening besides the **Master** tab. Go to this tab and click **Discover All Attributes**.

Figure 4-18. Discover All Attributes



10. When all the attributes are listed, locate the attribute with **UUID** value of **0xCAA2**, which is the characteristic under **CapSense Slider** custom service. It will also have a **Client Characteristic Configuration** descriptor (UUID 0x2902). The properties of the descriptor appear in the Attribute Details window, as shown in Figure 4-19.

Figure 4-19. Write to CapSense Slider Characteristic Descriptor



11. Enter '01:00' in the value field and click **Write Value**. This enables the notifications on the **CapSense Slider** characteristic. The notifications received are displayed in the **Value** field of the CapSense Slider characteristic.

12. Swipe your finger on the CapSense slider on the BLE Pioneer kit, as shown in Figure 4-21 and see the notification values in the CapSense Slider value field, as shown in Figure 4-20.

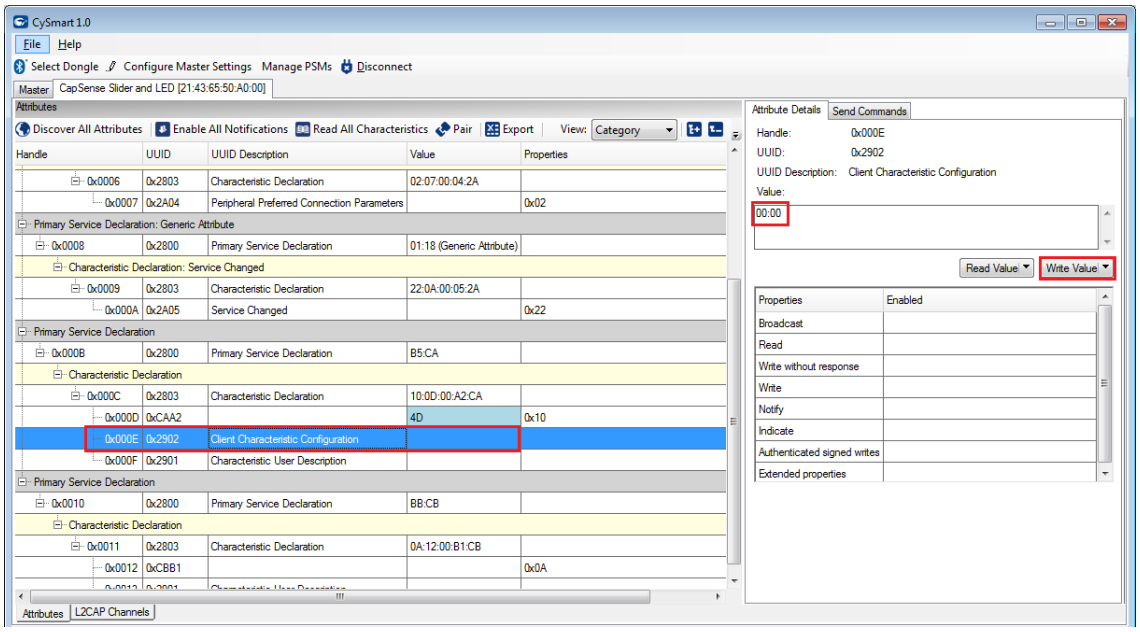
Figure 4-20. CapSense Slider notification received

Primary Service Declaration				
0x000B	0x2800	Primary Service Declaration	B5:CA	
Characteristic Declaration				
0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA	
0x000D	0xCAA2		49	0x10
0x000E	0x2902	Client Characteristic Configuration		
0x000F	0x2901	Characteristic User Description		

Figure 4-21. CapSense Slider

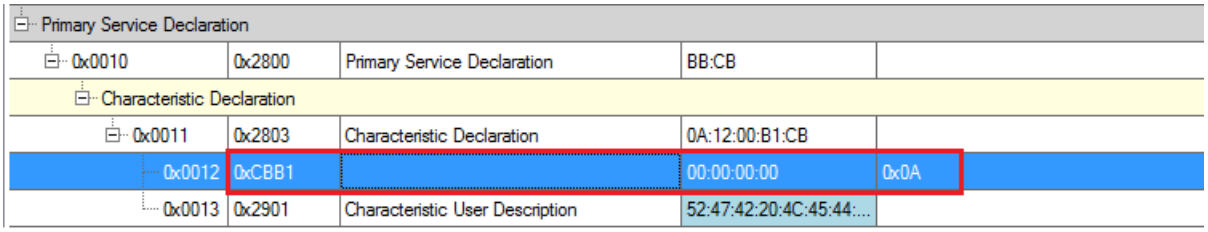
13. Write '00:00' to the **Client Characteristic Configuration** descriptor to disable notifications.

Figure 4-22. Disable Notifications



14. Now, locate the **UUID** value of **0xCBB1**, which is the **RGB LED Control** characteristic. On the right side under the Attribute Details tab, the read and write properties are enabled. Click the **Read value** to read the existing 4-byte of onboard RGB LED color information, as shown in Figure 4-23. The first three bytes have the color values Red, Green, and Blue; the fourth byte is the overall intensity value, all in the range 0 to 255.

Figure 4-23. Read RGB LED Control Characteristic Value



15. Modify the four bytes of data in the **Value** field and click **Write value**. You will see the corresponding change in the color and intensity of the RGB LED on the BLE Pioneer kit, as shown in Figure 4-24.

    Note: In case the BLE Pioneer kit is powered from the coin cell and not the USB Vbus, then the color mixing and intensity will vary. This is because the coin cell provides a lower driving voltage for RGB LEDs.

Figure 4-24. Write RGB LED Control Characteristic Value

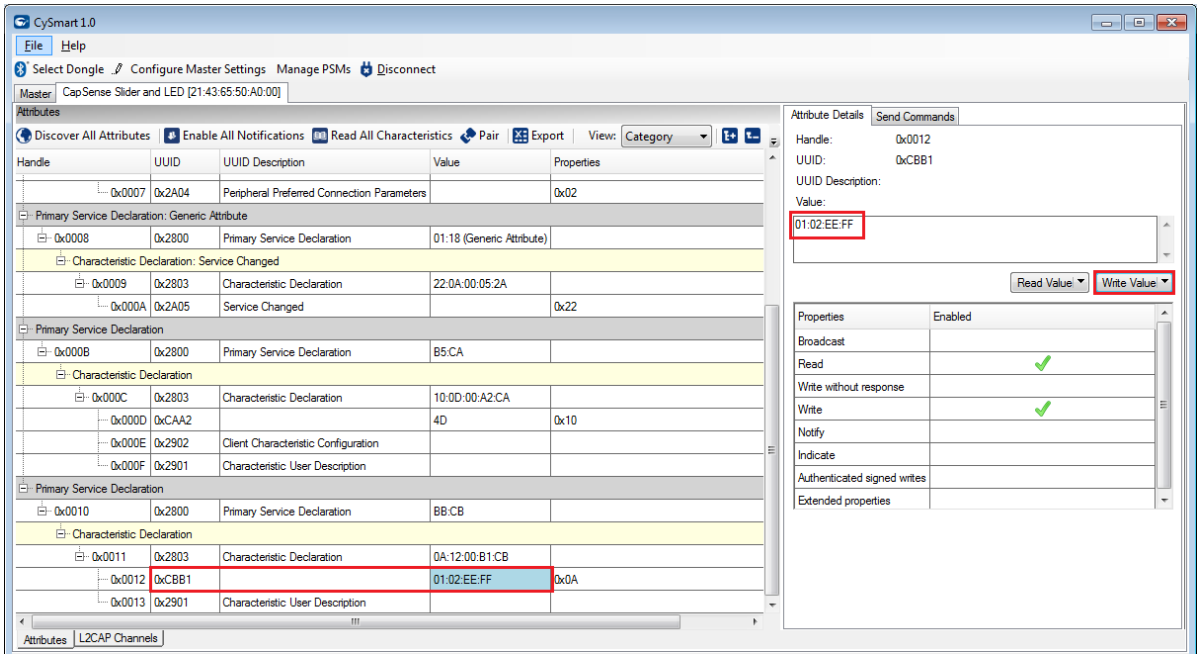


Figure 4-25. RGB LED Control with PSoC 4 BLE Module

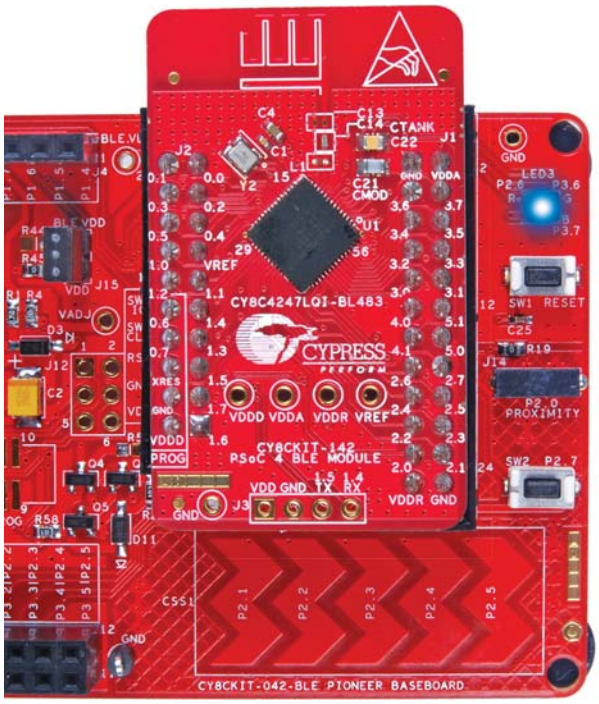
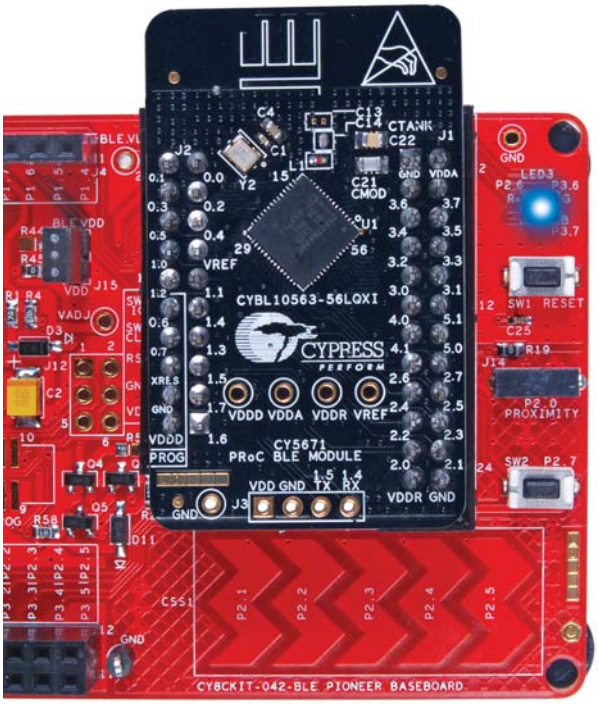
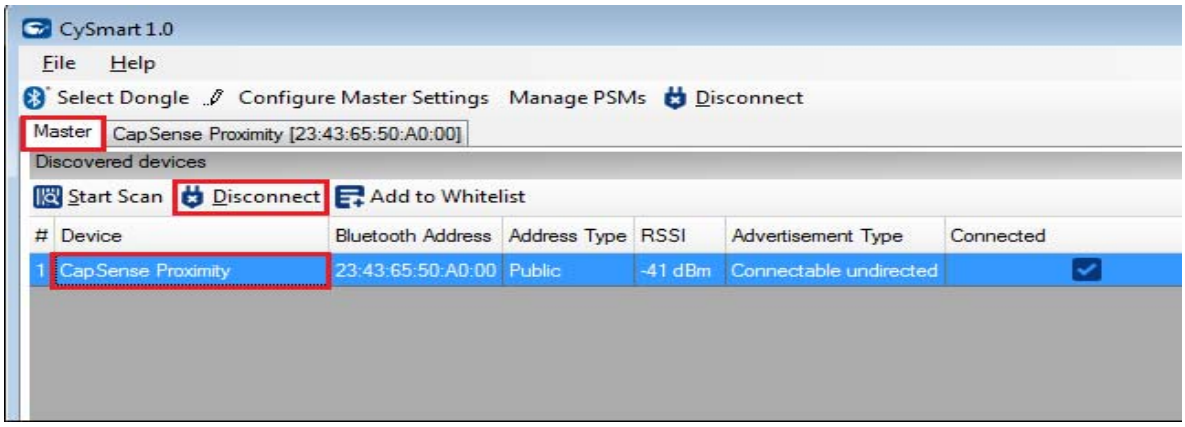


Figure 4-26. RGB LED Control with PRoC BLE Module



16. To disconnect from the device, go to the **Master** tab, select the device and click **Disconnect**, as shown in the Figure 4-27.

Figure 4-27. Disconnect from the Device



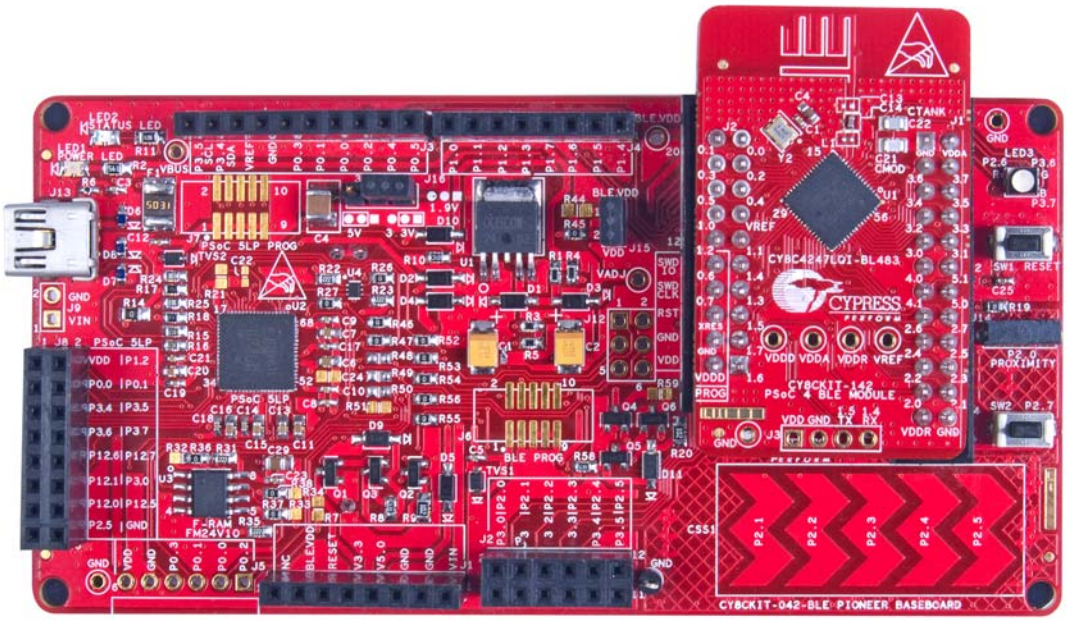
17. Press the user button (**SW2**) on the BLE pioneer kit to restart the advertisement.

4.3.4.2 CySmart iOS App

To verify the CapSense and LED project using the CySmart mobile application (see CySmart iOS Application on page 150), follow these steps:

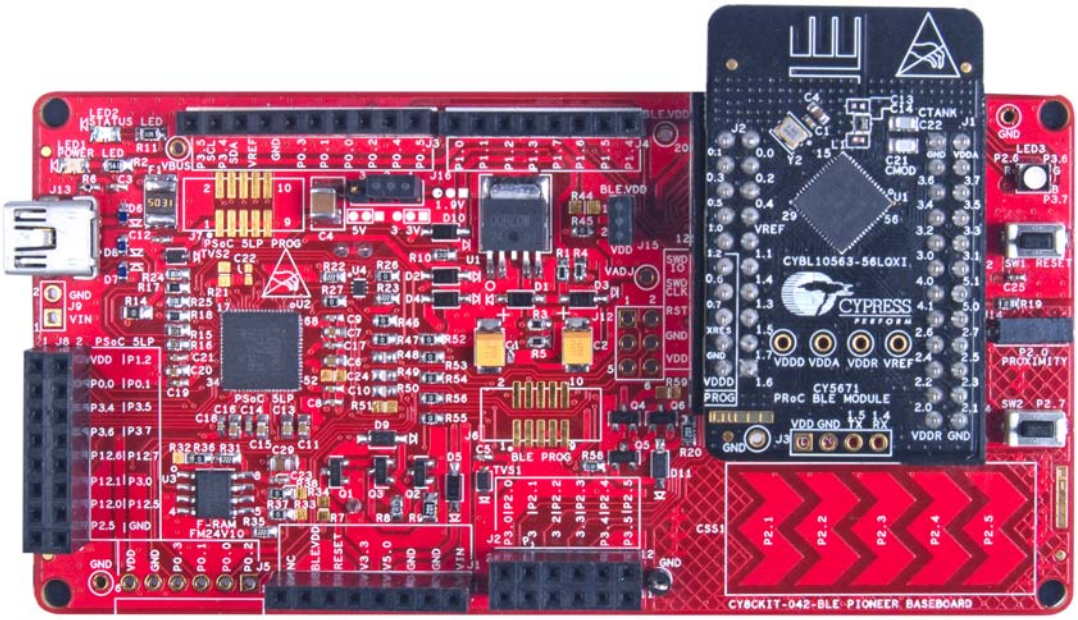
1. To verify the PSoC_4_BLE_CapSense_Slider_LED project, plug in the PSoC 4 BLE Module on the Baseboard.

Figure 4-28. BLE Pioneer Kit with PSoC 4 BLE Module



To verify the PRoC_BLE_CapSense_Sider_LED project, plug in the PRoC BLE Module on the Baseboard.

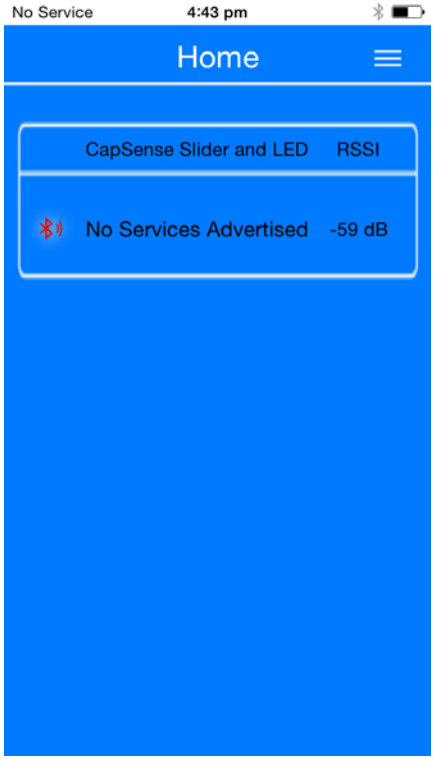
Figure 4-29. BLE Pioneer Kit with PRoC BLE Module



2. Plug the BLE Pioneer kit into the PC for power using the J13 USB connector.
3. Program the kit with the CapSense and LED example projects. See Using Example Projects on page 37 for programming instructions.
4. Press the user button (**SW2**) on the BLE Pioneer kit to start the advertisement. This is indicated by the blinking red LED on the BLE Pioneer kit.
5. Open the application on the mobile device. If Bluetooth is not enabled on the device, the application will ask to enable it.

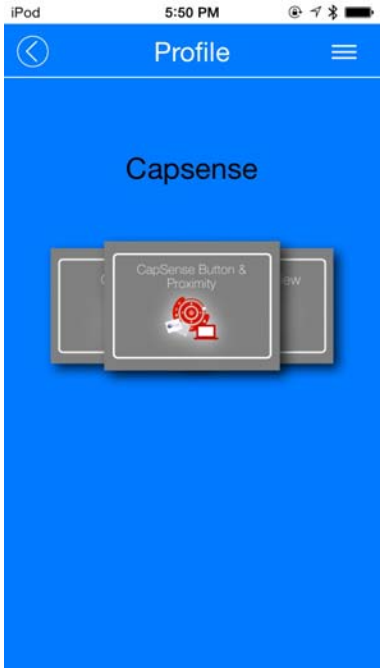
6. After Bluetooth is enabled, the application will automatically search for available BLE peripherals and list them. Select the **CapSense Slider and LED** peripheral as shown in Figure 4-30.

Figure 4-30. CapSense Slider and LED Peripheral



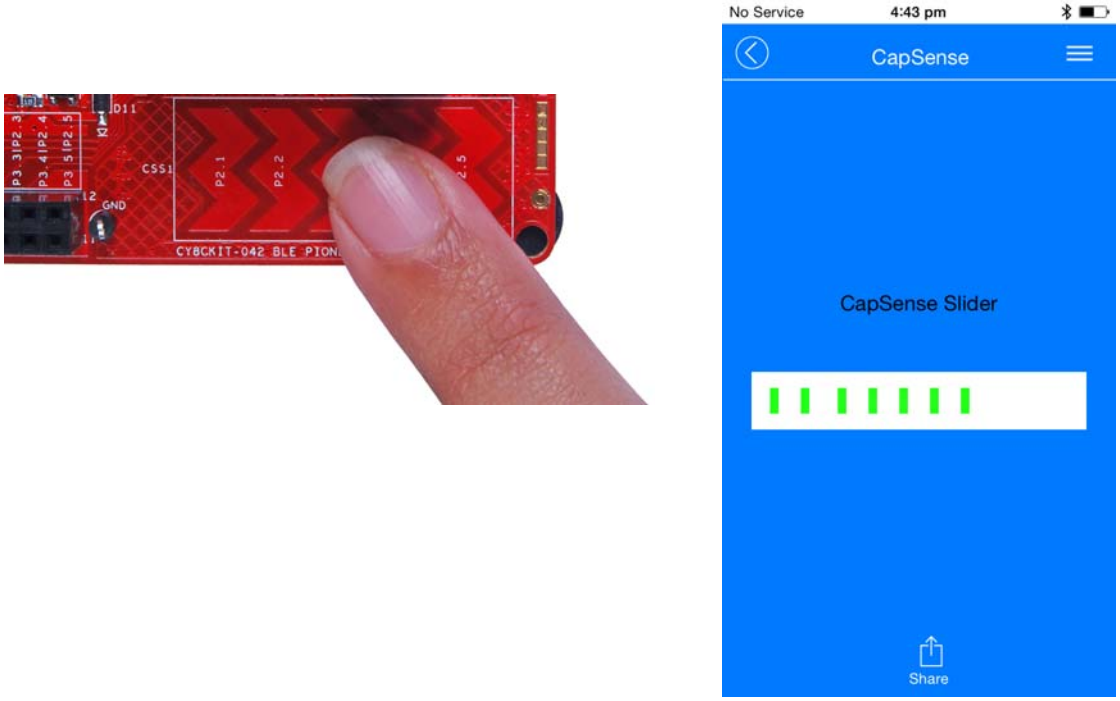
7. When connected, the application will list the supported profiles by the peripherals. Scroll and select the **CapSense** page, as shown in Figure 4-31.

Figure 4-31. CapSense Service Page



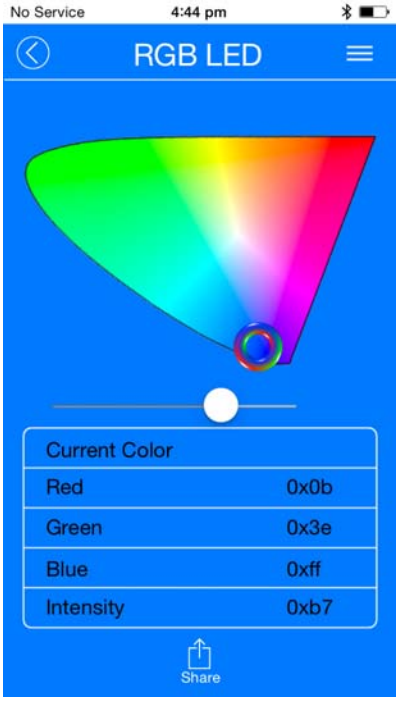
8. Swipe your finger on the CapSense slider on the BLE Pioneer kit and see a similar response on **CapSense** page in the CySmart application (Figure 4-32).

Figure 4-32. CapSense Slider



9. Press the back button in the top left corner of the application to return to the service selection page. Scroll and tap on the **RGB LED** service shown in Figure 4-33.

Figure 4-33. RGB LED Service Page



10. On the RGB LED service page, swipe over the color gamut to see a similar color response on the BLE Pioneer kit RGB LED. The slider below the color gamut controls the intensity of the RGB LED color.

    If the BLE Pioneer kit is powered from the coin cell and not the USB Vbus power, then the color mixing and intensity will vary. This is because the coin cell provides a lower driving voltage for RGB LEDs.

Figure 4-34. RGB LED Control with PSoC 4 BLE Module

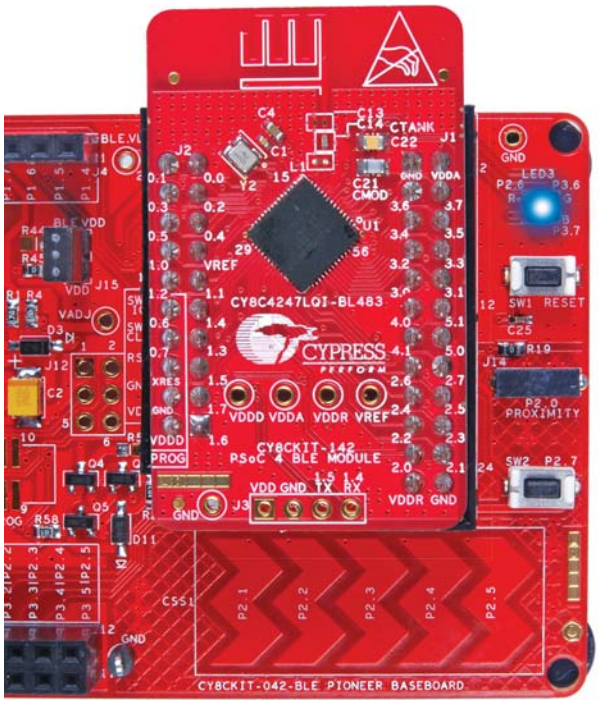
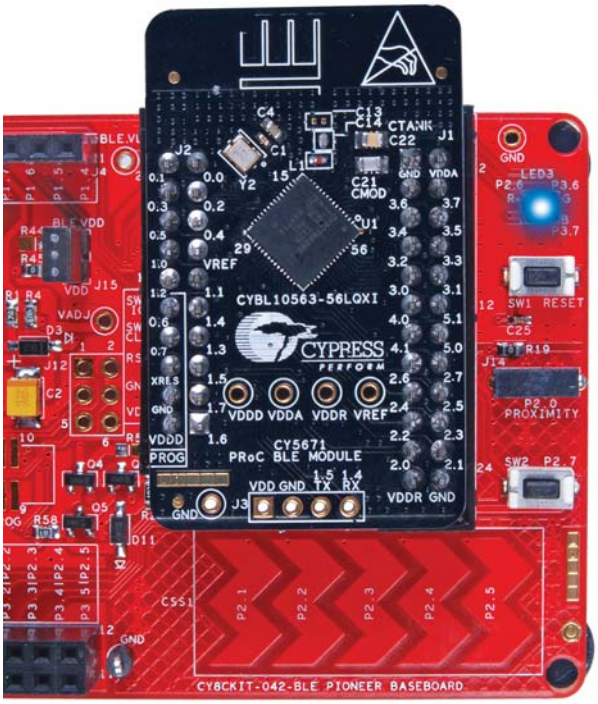


Figure 4-35. RGB LED Control with PRoC BLE Module



11. To disconnect from the BLE Pioneer kit, return to the CySmart app home screen by pressing the back button.
12. To reconnect to the peripheral, press the user button (**SW2**) on the BLE Pioneer kit again and then scan for devices using CySmart mobile app.

4.4 CapSense Proximity

4.4.1 Project Description

This example project demonstrates proximity sensing using CapSense and sending notifications over custom BLE service to a central device (Figure 4-36 and Figure 4-37). This project allows advertisement, connection, and notification (on proximity change) to be sent to a connected central device. It uses the CapSense component configured for single-sensor proximity sensing.

The project also supports sleep mode in the event of advertising timeout or BLE disconnection from the central device. The user button (**SW2**) on the BLE Pioneer kit is used to wake up from deep-sleep mode and restart the advertising procedure.

The blue LED is used as the status LED and provides visual confirmation on advertising or connection states. A blinking blue LED indicates advertising state.

Two projects demonstrate this functionality on two different devices:

- **PSoC_4_BLE_CapSense_Proximity** works with the PSoC 4 BLE Module.
- **PRoC_BLE_CapSense_Proximity** works with the PRoC BLE Module.

Figure 4-36. Top Design for PSoC_4_BLE_CapSense_Proximity Project

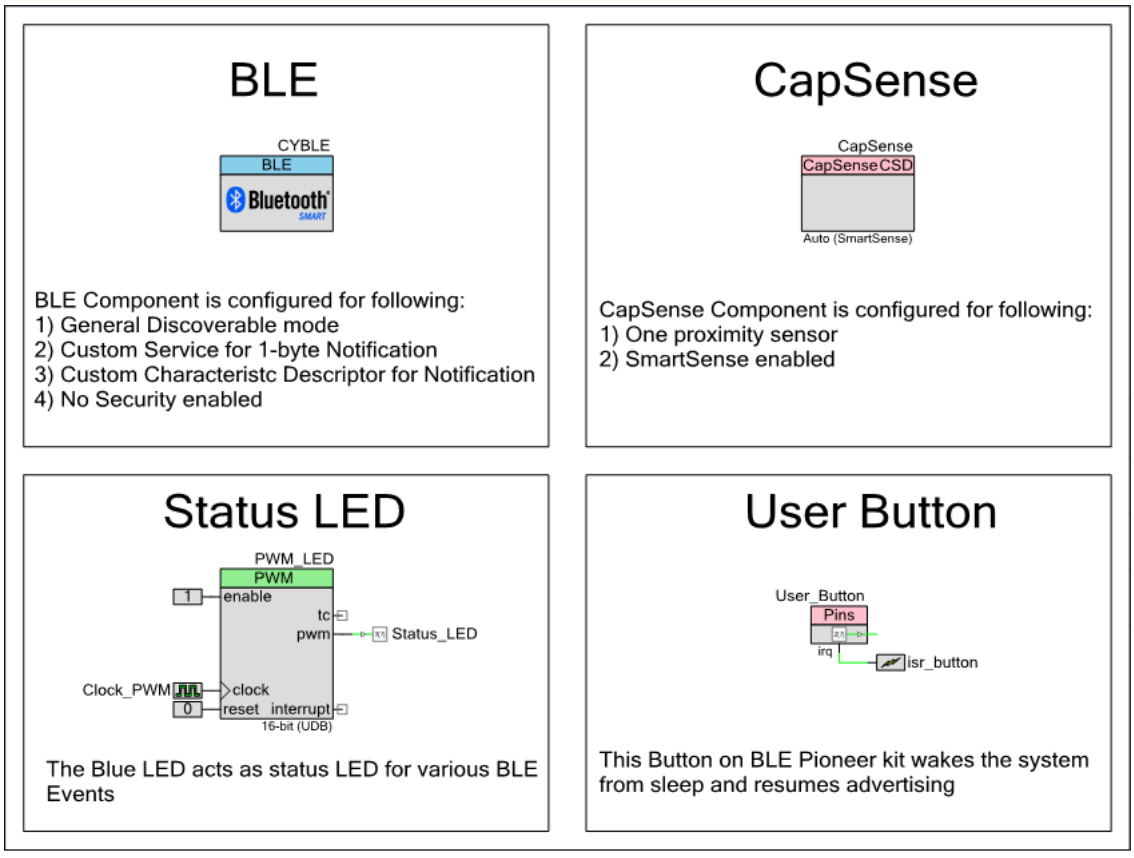
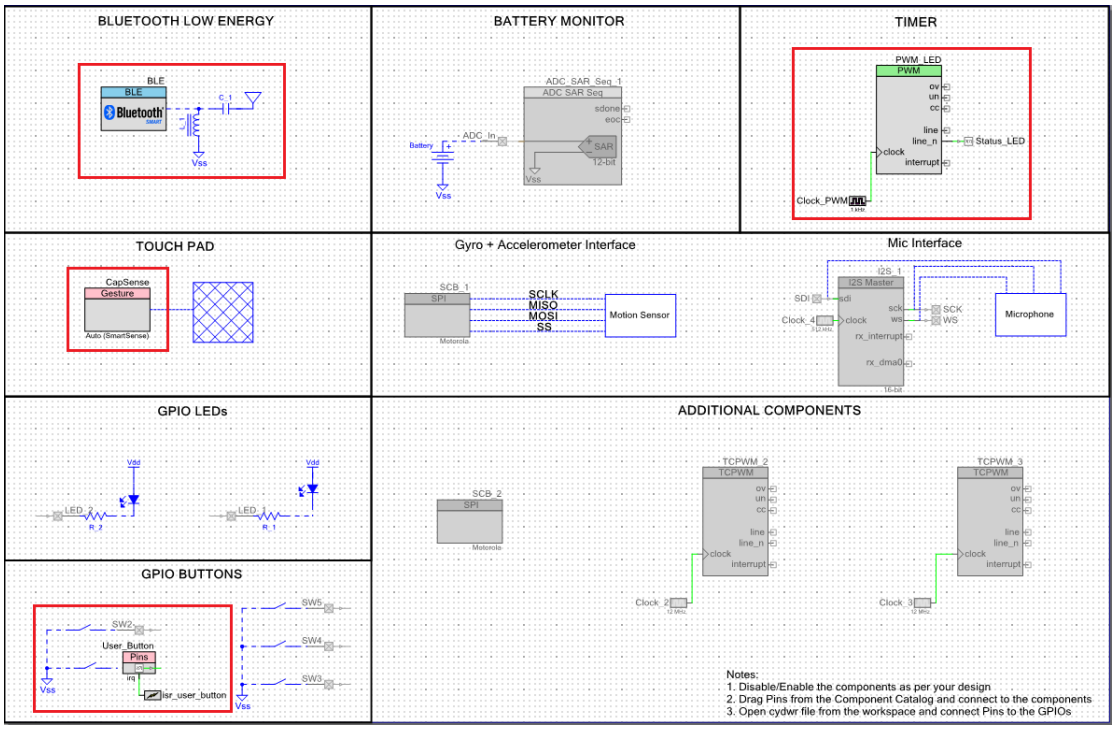


Figure 4-37. Top Design for PRoC_BLE_CapSense_Proximity Project



4.4.2 Hardware Connections

- Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC_4_BLE_CapSense_Proximity works with the PSoC 4 BLE Module. PRoC_BLE_CapSense_Proximity works with the PRoC BLE Module.
- Connect a five-inch wire (provided as part of this kit) to the proximity connector J14 on the baseboard. Loop the wire as shown in Figure 4-38.

  Note: Ensure that the proximity sensor loop wire is kept away as much as possible from the BLE antenna on the modules.

Figure 4-38. Proximity Sensor Connection on BLE Pioneer Kit with PSoC 4 BLE Module

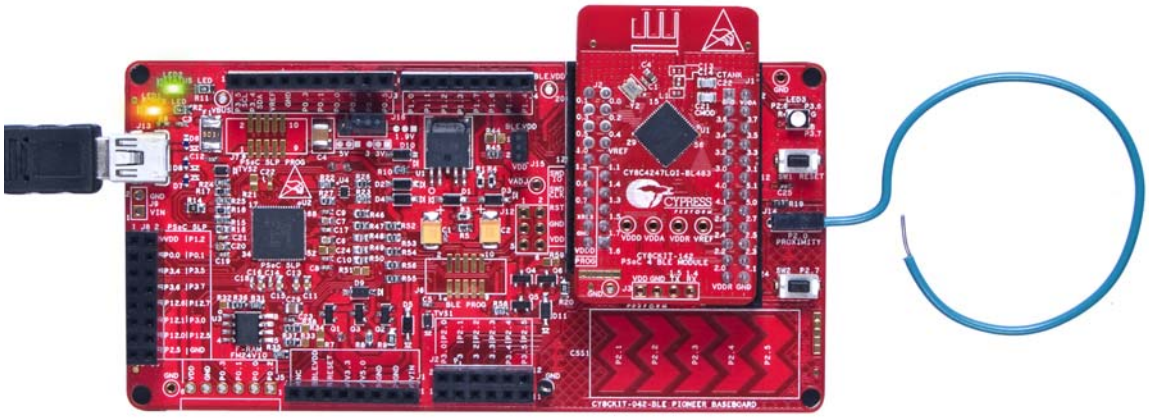
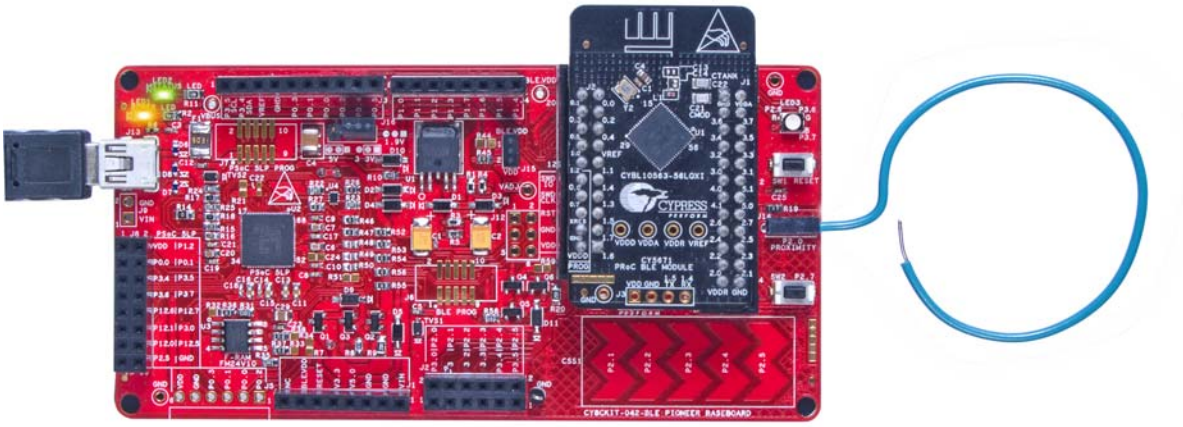


Figure 4-39. Proximity Sensor Connection on BLE Pioneer Kit with PRoC BLE Module



- On the Workspace Explorer, double-click **PSoC_4_BLE_CapSense_Proximity.cydwr/ PRoC_BLE_CapSense_Proximity.cydwr** on the left, and select the pins shown in Figure 4-40.

Table 4-3. Pin Assignments for CapSense Proximity Project

Pin Name	Port Name
CapSense CMOD	P4_0
CapSense Proximity Sensor	P2_0
User Button	P2_7
Status LED	P3_7

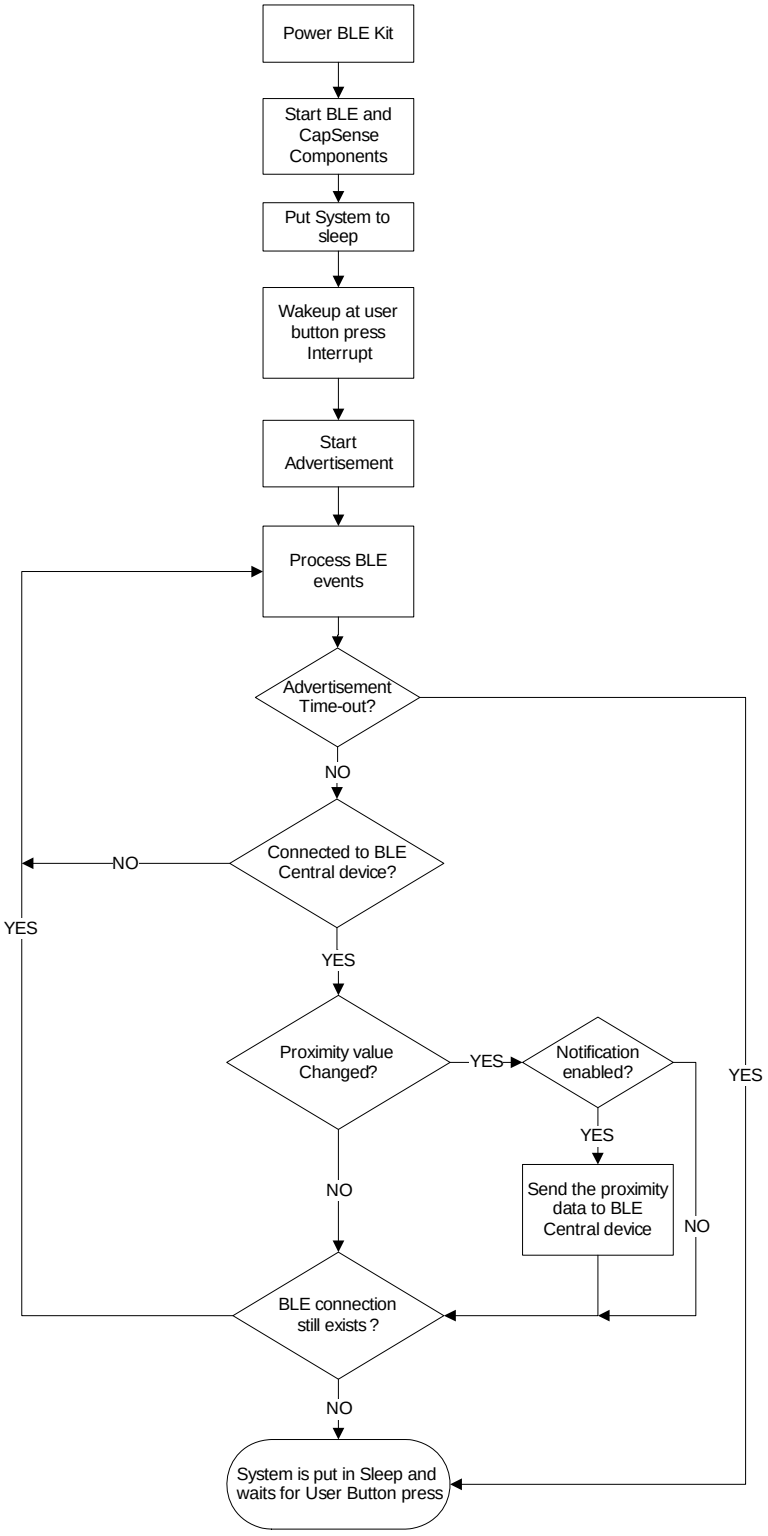
Figure 4-40. Pin Selection for CapSense Proximity Project

Alias	Name	Port	Pin
Cmod	\CapSense:Cmod\	P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi	5
ProximitySensor0_0__PROX	\CapSense:Sns\	P2[0] OA0:vplus, SCB0:spi_select[1]	37
	Status_LED	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf	54
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	44

4.4.3 Flow Chart

Figure 4-41 shows the flow chart of code implemented.

Figure 4-41. CapSense Proximity Project Flow Chart



4.4.4 Verify Output

The project can be verified by two methods: one using the CySmart PC Tool and Dongle, and the other using the CySmart iOS BLE app.

To install and use the CySmart PC Tool, see CySmart PC Tool on page 158.

To install and use the CySmart iOS app, see CySmart iOS Application on page 150.

4.4.4.1 CySmart PC Tool

To verify the CapSense proximity project using the CySmart PC tool, follow these steps:

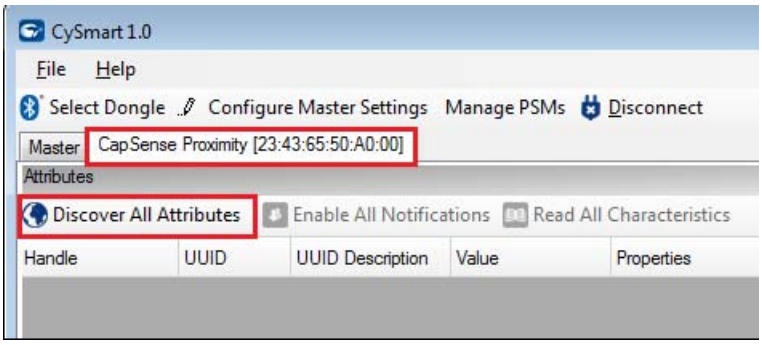
Note: See CySmart PC Tool on page 158 to know how to use the tool.

1. Connect the Dongle to one of the USB ports on the PC.
2. Start CySmart PC Tool on the PC by going to **Start > All Programs > Cypress > CySmart 1.0 > CySmart**. You will see a list of Dongles connected to it. If Dongle is not listed, click **Refresh**. Select the Dongle and click **Connect**.
3. Depending on the example project chosen, place the PSoC 4 BLE or PRoC BLE Module on the Baseboard.
4. Power the BLE Pioneer kit through the USB connector J13.
5. Connect a five-inch wire (included in the kit) to the proximity sensor connector J14 and make a loop of it from the other side.
6. Program the BLE Pioneer kit with the CapSense proximity example project. Follow the steps in Using Example Projects on page 37 to program the device.
7. After programming successfully, press the user button (**SW2**) on the BLE Pioneer Kit to start the advertisement. This is indicated by blinking blue LED on the Baseboard.
8. On the CySmart PC Tool, click **Start Scan** to see the list of available BLE peripheral devices. You can recognize your device by the device name and Bluetooth address shown in the list. This is the same as the one in the Gap Settings tab of the BLE component GUI in PSoC Creator.
9. After the available devices are listed, double-click **CapSense Proximity** to connect or click **Stop Scan** and then click **Connect** to connect to the device.

Figure 4-42. Connect to CapSense Proximity peripheral

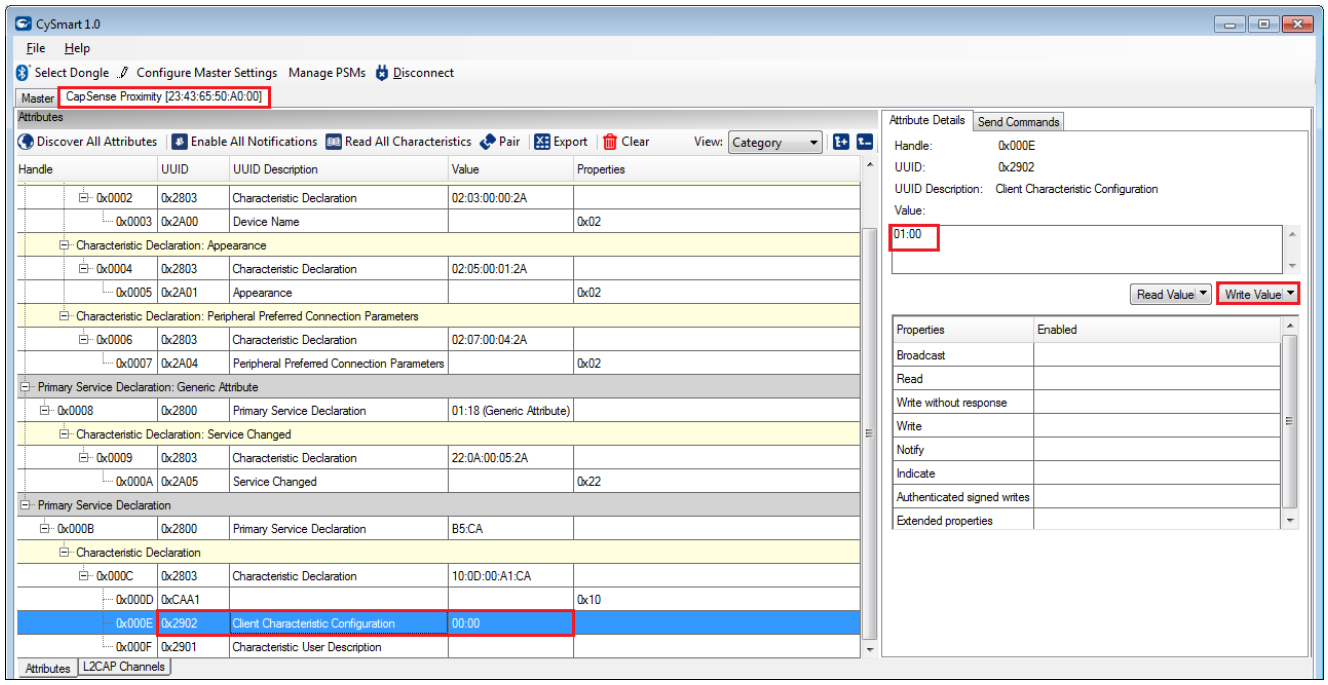
10. If the connection is successful, you will see another tab opening besides the Master tab. Go to this tab and click **Discover All Attributes**.

Figure 4-43. Discover All Attributes



11. When all the attributes are listed, locate the **UUID** value of **0xCAA1**, which is the characteristic for **CapSense Proximity**. It will also have a **Client Characteristic Configuration** descriptor (UUID 0x2902). The properties of the descriptor will appear under the Attribute Details window, as shown in Figure 4-44.

Figure 4-44. Write to CapSense Proximity Characteristic descriptor



12. Enter '01:00' in the **Value** field and click **Write Value**. This enables the notifications on the CapSense proximity characteristic. The notifications received are displayed in the value field of the CapSense proximity characteristic.

13. Bring your hand closer to the proximity sensor on the kit as shown in Figure 4-46 and Figure 4-47 and observe the value changing in the above characteristic value field, as shown in Figure 4-45.

Figure 4-45. CapSense Proximity notification received

Primary Service Declaration				
0x000B	0x2800	Primary Service Declaration	B5:CA	
Characteristic Declaration				
0x000C	0x2803	Characteristic Declaration	10:0D:00:A1:CA	
0x000D	0xCAA1		12	0x10
0x000E	0x2902	Client Characteristic Configuration	00:00	
0x000F	0x2901	Characteristic User Description		

Figure 4-46. CapSense Proximity Sensing with PSoC 4 BLE Module

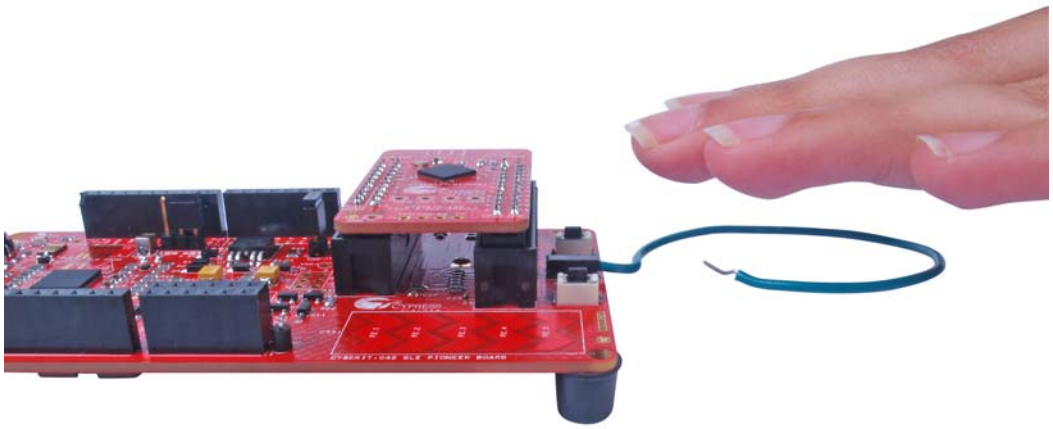
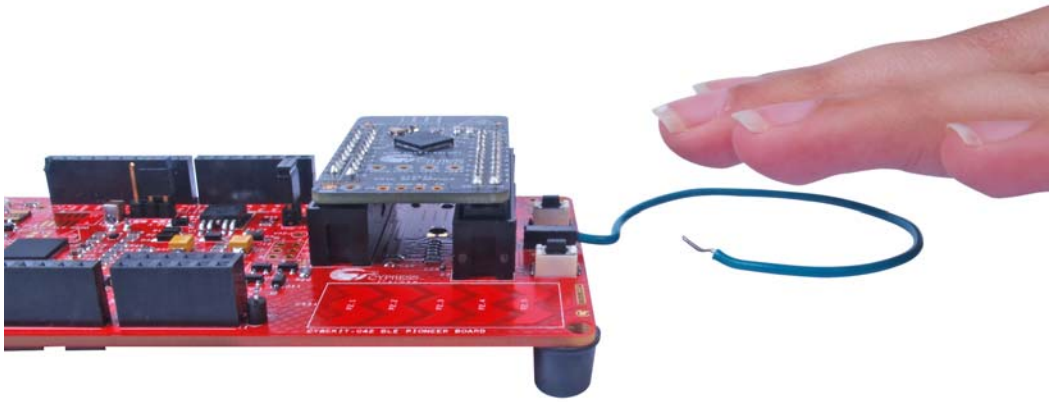


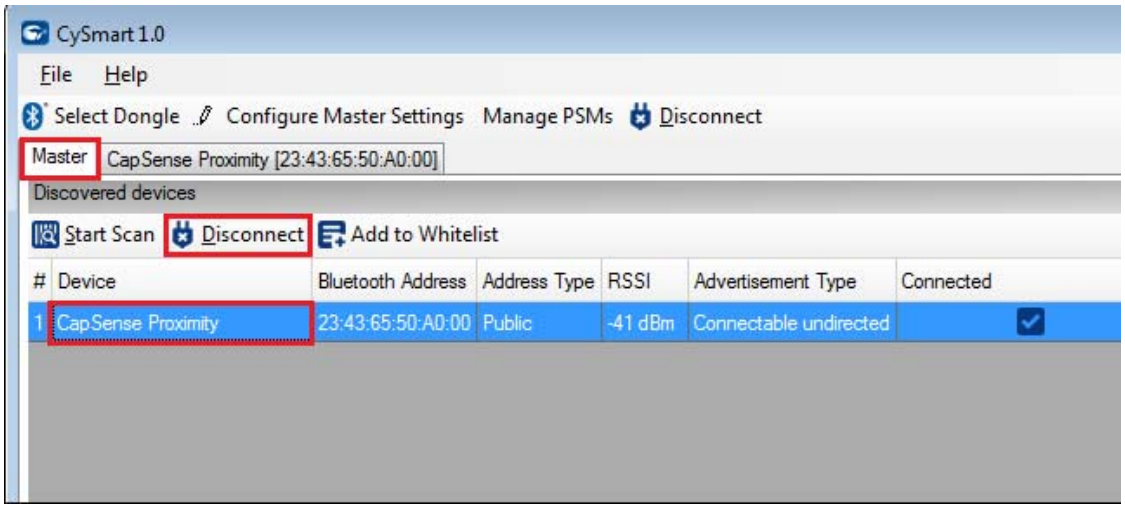
Figure 4-47. CapSense Proximity Sensing with PRoC BLE Module



14. Write '00:00' to the Client Characteristic Configuration descriptor to disable notifications.

15. To disconnect from the device, go to the **Master** tab, select the device and click **Disconnect**, as shown in Figure 4-48.

Figure 4-48. Disconnect from the Device



16. Press user button (**SW2**) to wake up from sleep and restart the advertisement for the next connection.

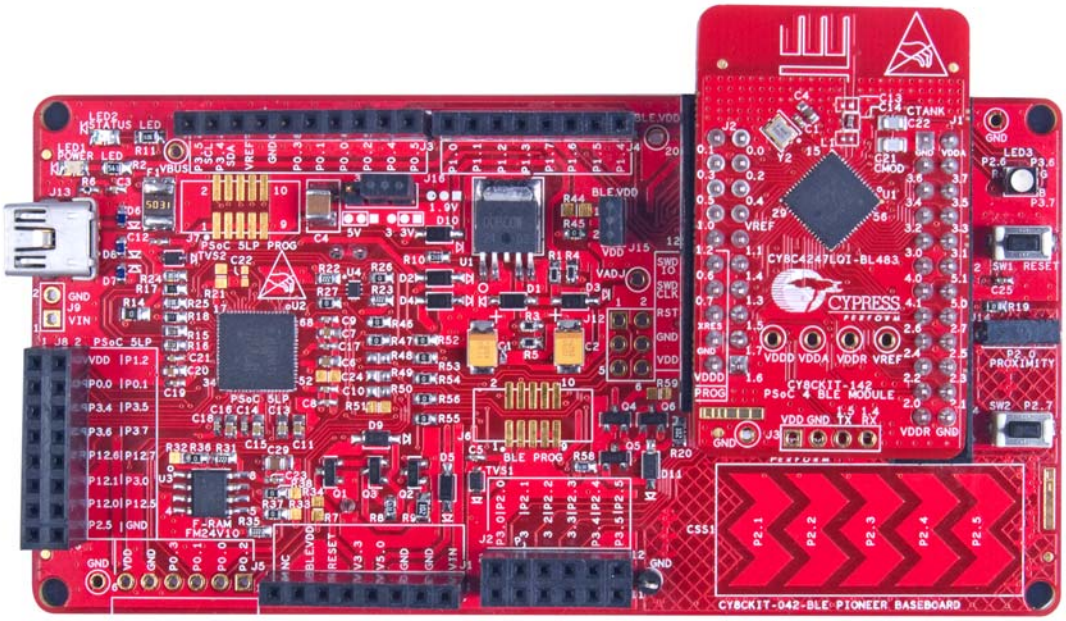
4.4.4.2 *CySmart iOS App*

To learn how to use the CySmart iOS application, see CySmart iOS Application on page 150 or the app user guide.

To verify the CapSense proximity project using the CySmart mobile app, follow these steps:

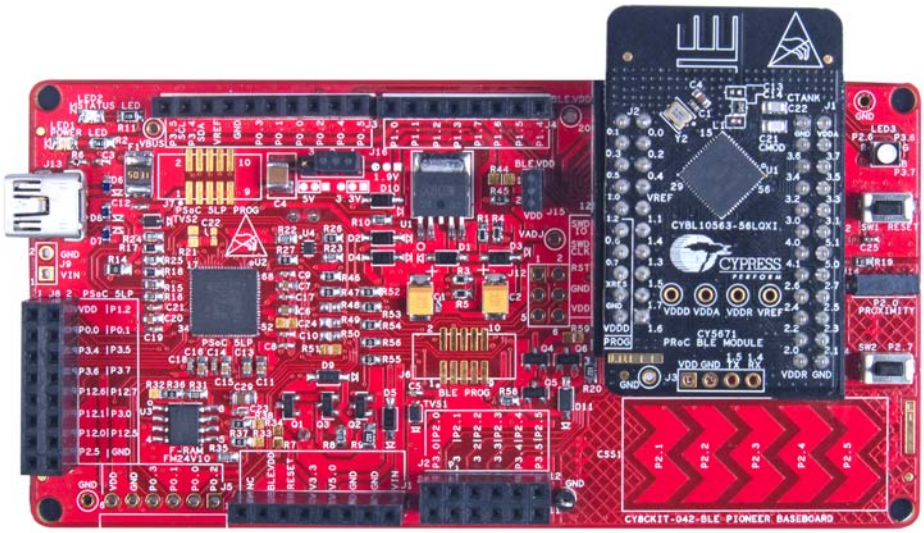
1. To verify the PSoC_4_BLE_CapSense_Proximity project, plug in the PSoC 4 BLE Module on the Baseboard.

Figure 4-49. BLE Pioneer Kit with PSoC 4 BLE Module



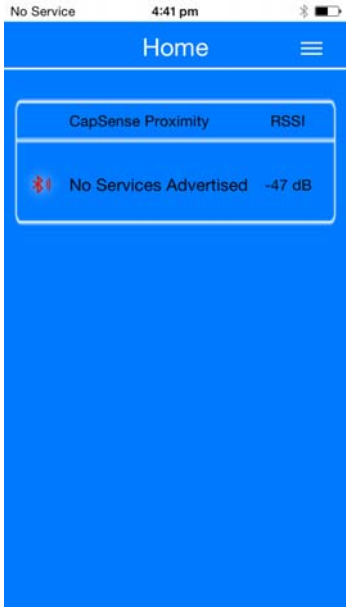
To verify the PRoC_BLE_CapSense_Proximity project, plug in the PRoC BLE Module on the Baseboard.

Figure 4-50. BLE Pioneer Kit with PRoC BLE Module



2. Connect the five-inch wire as a loop to the proximity connector J14 on the Baseboard.
3. Plug the BLE Pioneer kit into the PC for power, using the J13 USB connector.
4. Program the kit with the CapSense proximity example project. Follow steps in Using Example Projects on page 37 to program the device.
5. Press the user button (**SW2**) on the BLE Pioneer kit to start the advertisement.
6. Open the CySmart app on the mobile device. If Bluetooth is not enabled on the device, the app will ask to enable it.
7. The app will automatically search for available BLE peripherals and list them. Select the **CapSense Proximity** peripheral, as shown in Figure 4-51.

Figure 4-51. Connect to CapSense Proximity Peripheral



8. When connected, the app will list the services supported by the peripherals. Scroll and select the **CapSense** service.
9. When the CapSense service page opens, bring your hand near the sensor wire on the BLE Pioneer kit and see a similar response in the app as a bar graph, as shown in Figure 4-52.

Figure 4-52. CapSense Proximity Sensing with PSoC 4 BLE Module

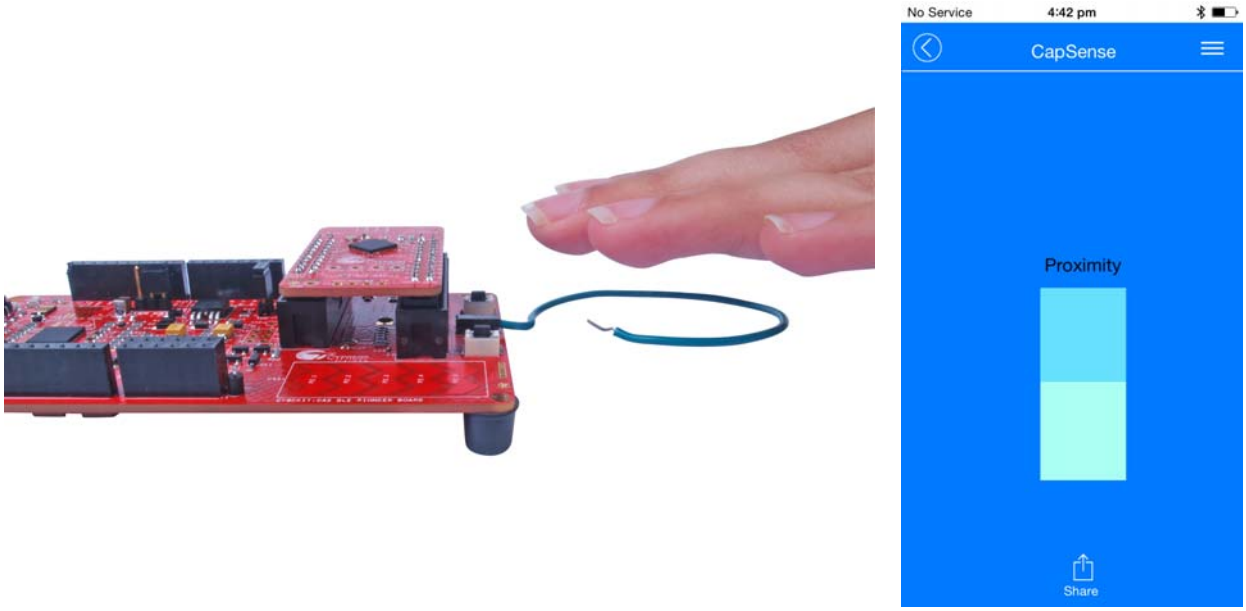
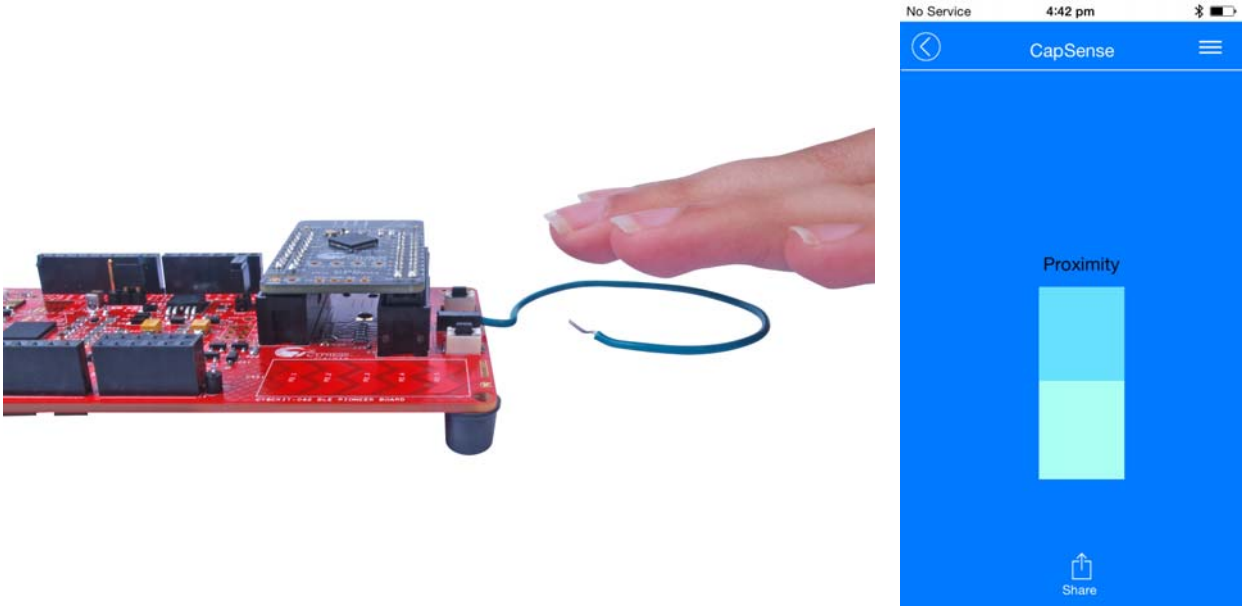


Figure 4-53. CapSense Proximity Sensing with PRoC BLE Module



10. To disconnect from the BLE Pioneer kit, return to the device selection screen on the CySmart app.
11. To reconnect to the peripheral, press the user button (**SW2**) on the BLE Pioneer Kit to restart the advertisement and scan for the device in the CySmart app.

4.5 BLE Central Mode

4.5.1 Project Description

The BLE projects described above have been functioning as BLE peripheral devices. This means that the firmware role was set to be as a server; another device, acting as BLE central (such as the CySmart PC tool or CySmart mobile application) will connect to it and collect the data.

This example project demonstrates the BLE client mode where it will scan for peripheral devices and connect to a particular device supporting Immediate Alert Service (IAS). Connection is initiated when the advertising peripheral Bluetooth address matches the pre-determined Bluetooth address that is saved inside the central device firmware. When connected, it sends immediate alert levels with a press of the button (**SW2**) on the Pioneer kit. To aid in evaluation, an additional BLE IAS peripheral example project is provided for the Dongle. The BLE IAS peripheral example project is present in the same workspace of the PSoC Creator central IAS example project workspace.

The BLE Pioneer kit scans for button presses and sends all three alert levels in circular fashion. The Dongle acting as an IAS peripheral changes the LED behavior according to the received alert level. Upon each successive button press, the LED state changes in a circular fashion. The default level is **No Alert**.

Two projects demonstrate the BLE central functionality on the two devices:

- **PSoC_4_BLE_Central_IAS** works with the PSoC 4 BLE Module.
- **PRoC_BLE_Central_IAS** works with the PRoC BLE Module.

Additionally, the **BLE_Dongle_Peripheral_IAS** project is to be programmed on the Dongle. This project is present in both PSoC_4_BLE_Central_IAS and PRoC_BLE_Central_IAS workspace and can be used to program the Dongle separately.

Note: If the Dongle is programmed with the **BLE_Dongle_Peripheral_IAS** example, it will not work with the CySmart PC utility. Reprogram the Dongle with the CySmart firmware according to BLE Dongle and LED Control on page 79 to use the CySmart PC Tool.

Figure 4-54. PSoC_4_BLE_Central_IAS TopDesign

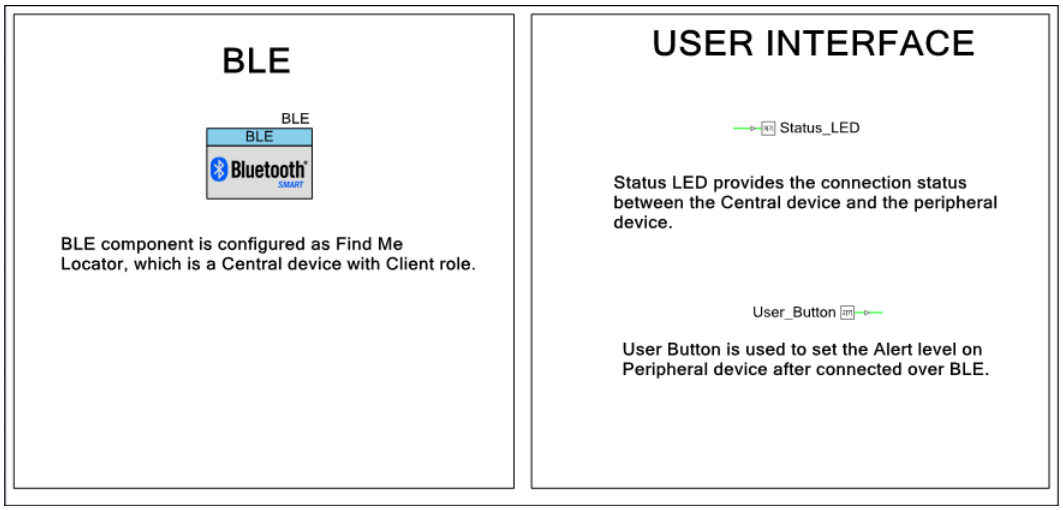


Figure 4-55. PRoC_BLE_Central_IAS TopDesign

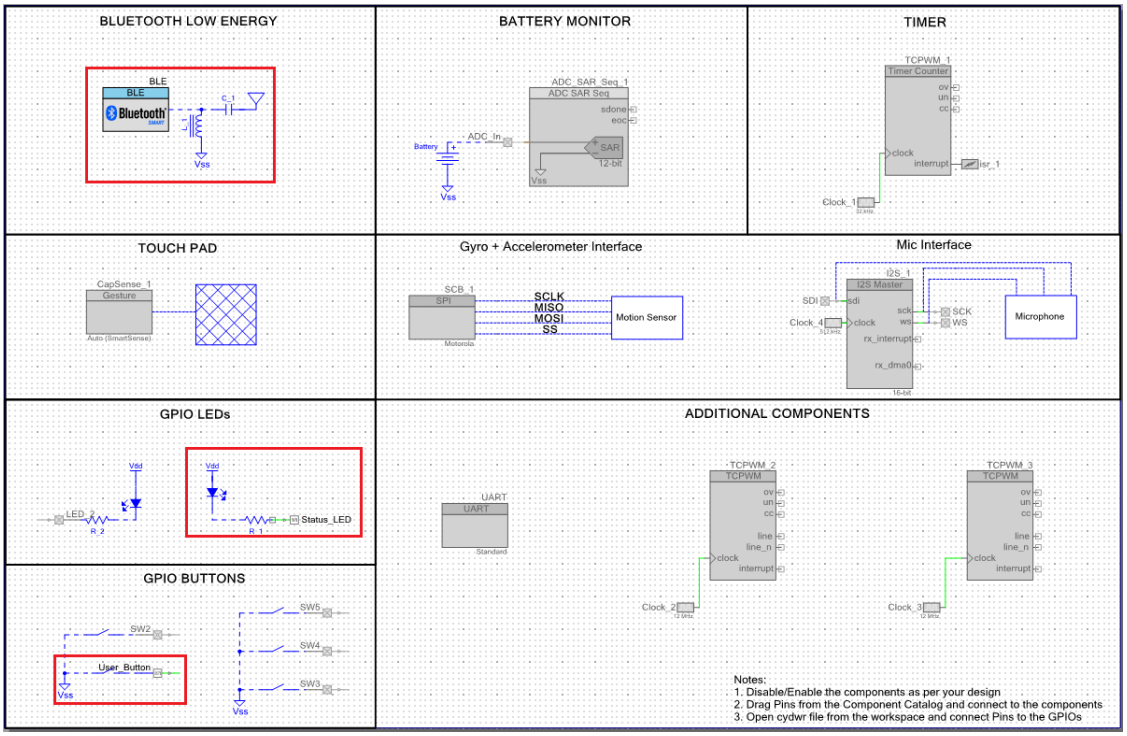
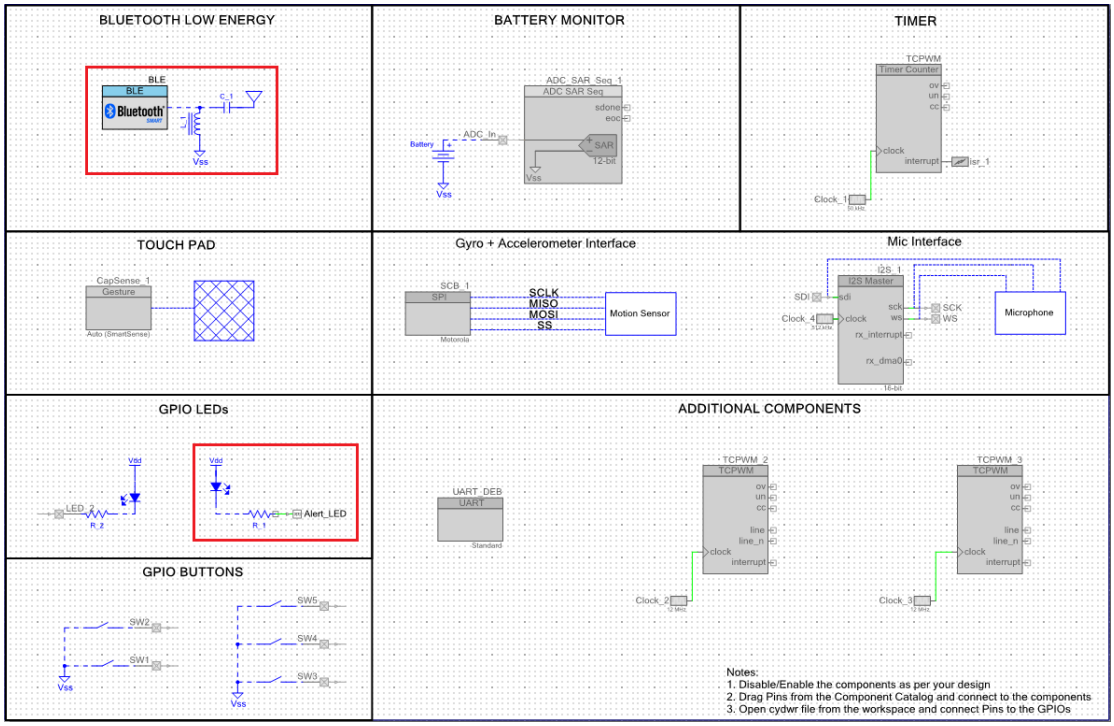


Figure 4-56. BLE_Dongle_Peripheral_IAS TopDesign



4.5.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the board.

Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC_4_BLE_Central_IAS works with the PSoC 4 BLE Module. PRoC_BLE_Central_IAS works with the PRoC BLE Module. BLE_Dongle_Peripheral_IAS is the common project for both workspaces and programs the Dongle with peripheral mode firmware.

On the Workspace Explorer, double-click **PSoC_4_BLE_Central_IAS.cydwr/ PRoC_BLE_Central_IAS.cydwr** on the left, and select the pins shown in Figure 4-57.

Figure 4-57. Pin Selection for BLE IAS Central Example Project

Alias	Name	Port	Pin	Lock
	Status_LED	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart_cts, SRSS:ext_clk_lf	54	☑
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	44	☑

Similarly, double-click **BLE_Dongle_Peripheral_IAS.cydwr** in Workspace Explorer and select the pins shown in Figure 4-58.

Figure 4-58. Pin Selection for BLE IAS Peripheral Example Project

Alias	Name	Port	Pin	Lock
	Alert_LED	P3[3] SARMUX:pads[3], TCPWM1:line_out_compl, SCB0:uart_cts	50	☑

4.5.3 Flow Chart

Figure 4-59 shows the flow chart for the IAS client mode example project.

Figure 4-59. IAS Client Mode Flow Chart

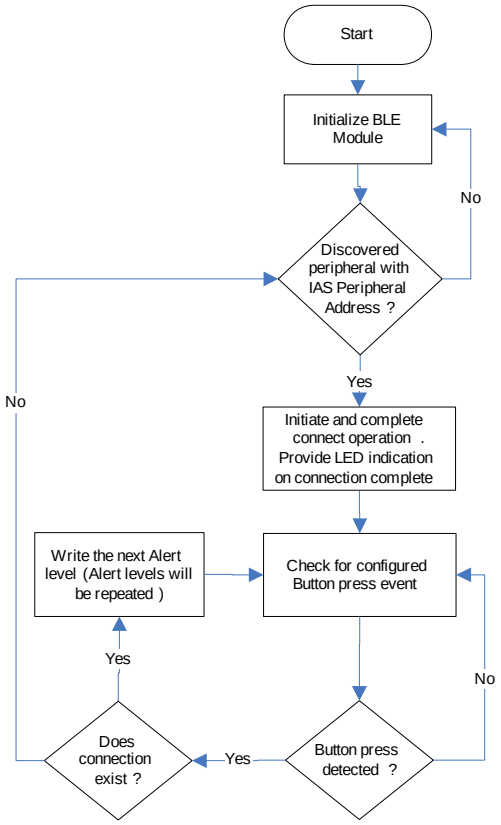
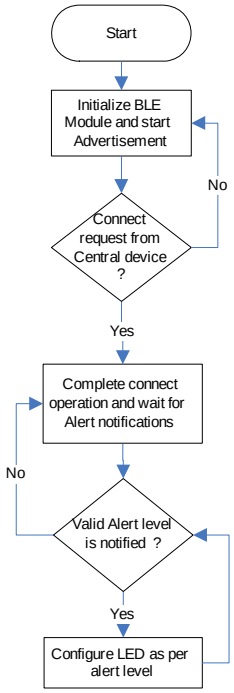


Figure 4-60 shows the flow chart for the IAS peripheral mode example project.

Figure 4-60. IAS Peripheral Mode Flow Chart



4.5.4 Verify Output

1. Connect the Dongle to one of the USB ports on the PC.

Figure 4-61. Connect Dongle to USB Port

2. Program the Dongle with the BLE_Dongle_Peripheral_IAS project described in Using Example Projects on page 37.

   Note: Do not update the public device address (inside the BLE component) for the **BLE_Dongle_Peripheral_IAS** example project. This is because the central example project looks for a particular public address equal to the one set on the peripheral BLE component. Changing the **BLE_Dongle_Peripheral_IAS** example project public address will lead to no connection with the central device on the BLE Pioneer kit.
3. Power the BLE Pioneer kit through USB connector J13.
4. Program the BLE Pioneer kit with either PSoC_4_BLE_Central_IAS or PRoC_BLE_Central_IAS project, depending on the BLE Module placed on the BLE Pioneer kit.
5. Wait for the BLE connection between the dongle and the Pioneer baseboard. The connection success status is indicated on the Baseboard in the following three stages:
   a. Fast blinking blue LED represents scanning mode. During this mode, the BLE Pioneer kit is scanning for peripheral devices.
   b. Slow blinking blue LED represents discovery mode. During this mode, the BLE Pioneer kit has found the Dongle peripheral device and has started the connection procedure.
   c. The blue LED remains on, representing the connected mode. This mode means that the peripheral device has been connected and the application can now send alert levels.
6. Press the user button (**SW2**) on the BLE Pioneer kit to send the next alert level to the Dongle. The alert level will keep rotating from No Alert to Mid Alert to High Alert.
7. Check if the LED behavior changes for each alert notification on the Dongle according to the following table:

Alert Level	LED State
No Alert	LED OFF
Mild Alert	LED blinking
High Alert	LED ON

Figure 4-62. User Button on BLE Pioneer Kit with PSoC 4 BLE Module

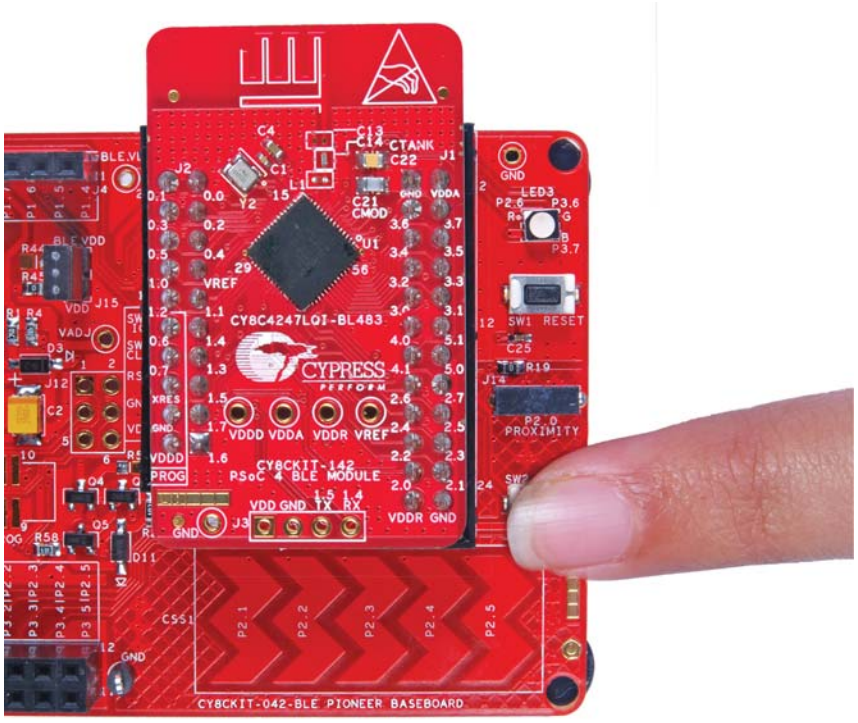
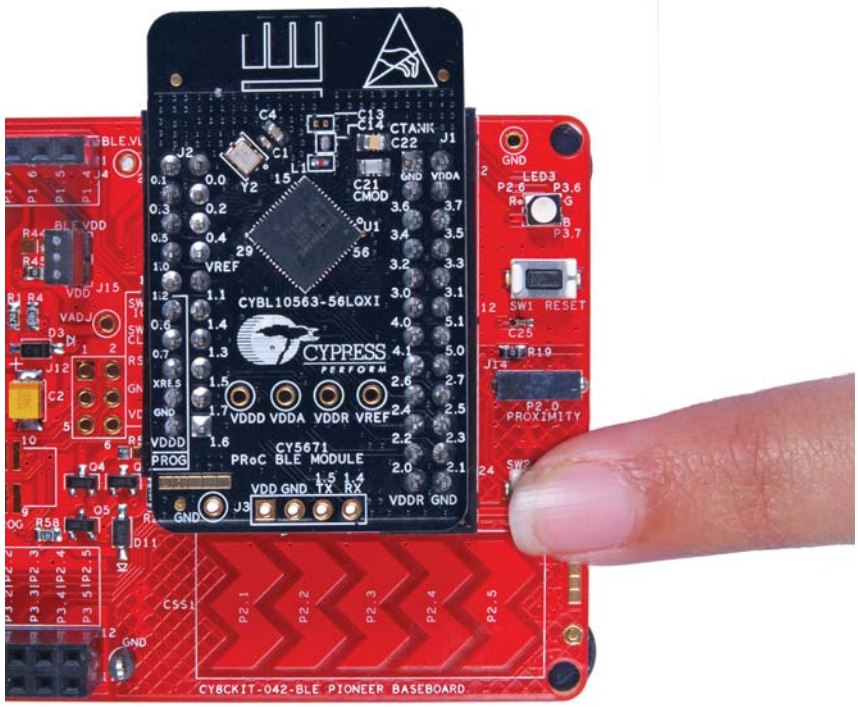


Figure 4-63. User Button on BLE Pioneer Kit with PRoC BLE Module



Note: To revert the CySmart functionality to the Dongle, program the Dongle hex file, as described in Programming and Debugging BLE Device on page 26.

4.6 BLE Dongle and LED Control

4.6.1 Project Description

This firmware supports the CySmart debug tool (see CySmart PC Tool on page 158) by acting as the BLE host emulator. This is the default firmware that comes in Dongle shipped with the kit.

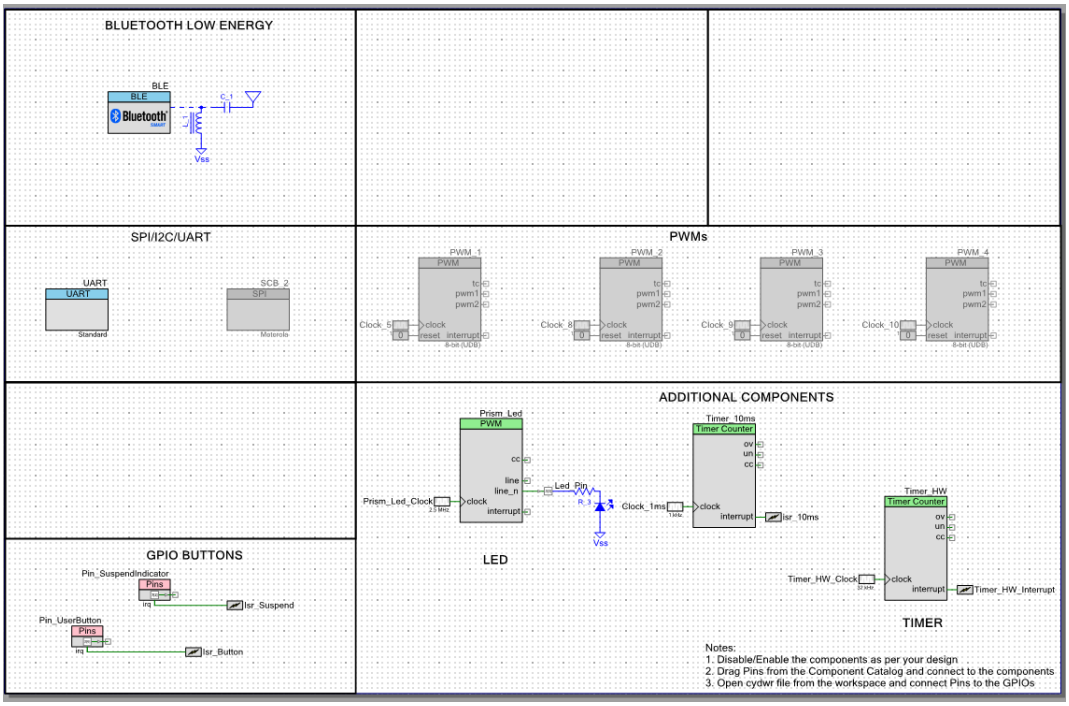
This project additionally demonstrates LED brightness control via a custom BLE profile, which works with the CapSense slider example explained in CapSense Slider and LED on page 46.

The device will scan for the peripheral acting as a CapSense Slider and LED device, and connect to it automatically. Then, it will enable slider notifications and process the received notifications. Whenever CapSense detects activity, it will notify the finger location to the Dongle; the dongle will update the LED brightness using PWM.

The custom client LED control will be stopped if the CySmart PC Tool acquires the Dongle. The dongle will enter the CySmart emulator mode, in which it will process all BLE commands as triggered by the user via the tool. The project uses custom command/event protocol to exchange data between the CySmart PC Tool and the BLE component via a USB-CDC interface. It uses the Cypress USB-UART bridge functionality from the PSoC 5LP based KitProg module described in Using PSoC 5LP as USB-UART Bridge on page 113.

Note: This project is meant only for the PRoC BLE device and works on the Dongle hardware.

Figure 4-64. Top Design for BLE_Dongle_CySmart project



4.6.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the dongle board.

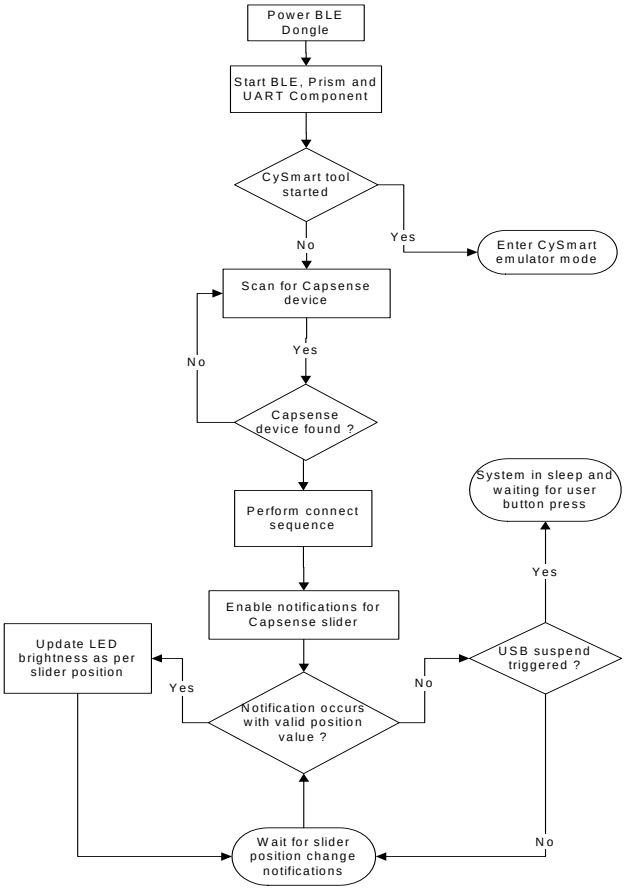
On the Workspace Explorer, double-click **BLE_Dongle_CySmart.cydwr** on the left, and select the pins shown in Figure 4-65.

Figure 4-65. Pin Selection for BLE Dongle Project

Alias	Name	Port	Pin	Lock
	\UART:rx_wake\	P1[4] OA3:vminus, TCPWM2:line_out, SCB0:uart_rx, SCB0:i2c_sda, SCB0:spi_mosi	32	☑
	\UART:tx\	P1[5] OA3:vplus, TCPWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi_miso	33	☑
	Led_Pin	P3[3] SARMUX:pads[3], TCPWM1:line_out_compl, SCB0:uart_cts	50	☑
	Pin_SuspendIndicator	P3[2] SARMUX:pads[2], TCPWM1:line_out, SCB0:uart_rts	49	☑
	Pin_UserButton	P2[6] OA0:vplus_alt	43	☑

4.6.3 Flow Chart

Figure 4-66. Flow Chart for BLE_Dongle_CySmart Project



4.6.4 Verify Output

This project will be used whenever the CySmart PC Tool (see CySmart PC Tool on page 158) is invoked for testing other example projects. In addition, the LED control operation can be verified as follows.

1. Power the BLE Pioneer kit through the USB connector J13.
2. Program the BLE Pioneer kit with the CapSense and LED example project described in CapSense Slider and LED on page 46.
3. Connect the Dongle to one of the USB ports on the PC.
4. Program the Dongle with the **BLE_Dongle_CySmart** project. See Using Example Projects on page 37 for programming instructions.
5. Wait for the BLE connection between the dongle and the Pioneer baseboard. The connection success status will be indicated by a 3-second ON state of the red LED followed by the OFF state on the Baseboard.
6. Move your finger on the CapSense slider and check the LED brightness variation on the dongle.

5. Hardware

This chapter describes the contents of the BLE Pioneer Kit hardware and its different blocks, such as the power block, USB connection, Arduino compatible headers, BLE Module connectors, and CapSense slider.

5.1 Pioneer Baseboard

5.1.1 PSoC 5LP

An onboard PSoC 5LP is used to program and debug the BLE silicon. The PSoC 5LP connects to the USB port of the PC through a USB mini-B connector and to the SWD interface of the BLE 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, visit the PSoC 5LP web page.

See the section Serial Interconnection Between PSoC 5LP and Bluetooth Module on page 100 for more details.

5.1.2 Power 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 connector
- 5-V to 12-V VIN power from Arduino shield
- 3.3 V from I/O header
- 3 V from CR2032 coin cell

An adjustable LDO is used to output three different voltage levels (1.9 V, 3.3 V, and 5 V) to power the BLE Module. These voltages are selected through the J16 jumper, as shown in Figure 5-1.

Figure 5-1. Schematics and Board Highlight of LDO and Power Selection Jumper

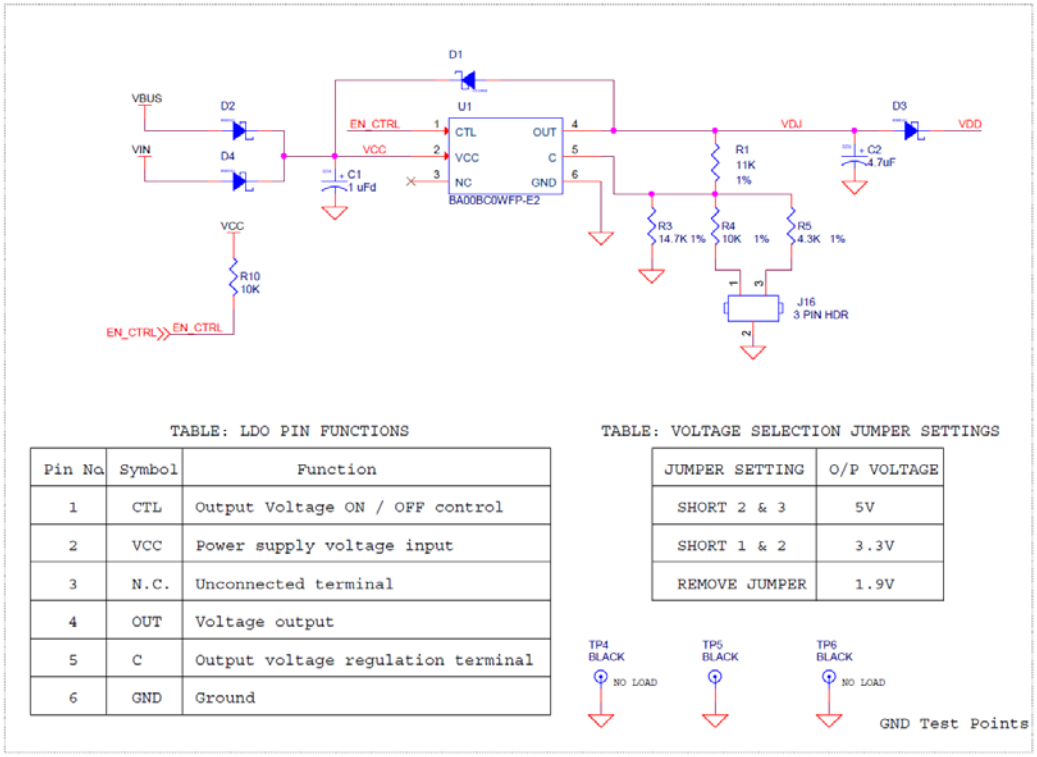
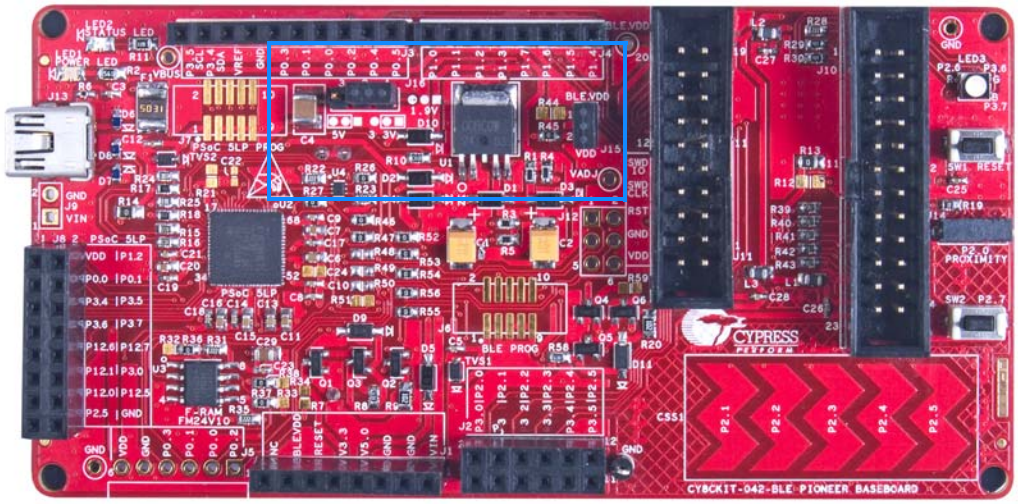


TABLE: LDO PIN FUNCTIONS

Pin No	Symbol	Function
1	CTL	Output Voltage ON / OFF control
2	VCC	Power supply voltage input
3	N.C.	Unconnected terminal
4	OUT	Voltage output
5	C	Output voltage regulation terminal
6	GND	Ground

TABLE: VOLTAGE SELECTION JUMPER SETTINGS

JUMPER SETTING	O/P VOLTAGE
SHORT 2 & 3	5V
SHORT 1 & 2	3.3V
REMOVE JUMPER	1.9V

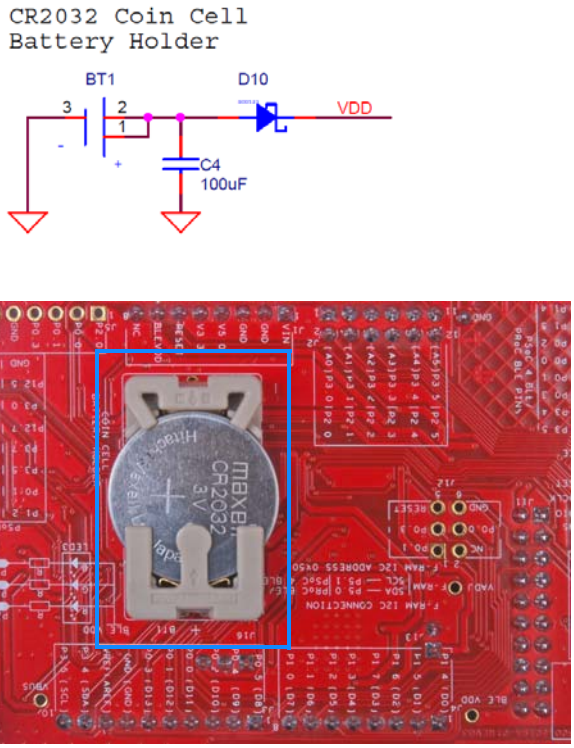


The input to the LDO can come from either the USB or the VIN pin in the Arduino header.

Note: The typical dropout voltage of the selected LDO is 0.3 V at 500-mA output current. This gives a minimum output of 4.6 V from the input voltage of 5 V from the VBUS. This drop also takes into account the voltage drop across the Schottky diode connected at the output of the LDO to protect against voltage applied at the output terminal of the regulator.

The board also contains a CR2032 coin cell holder to power the board using a coin cell, as shown in Figure 5-2.

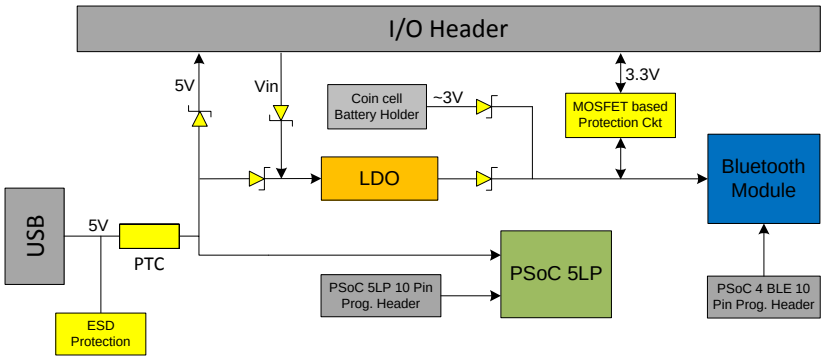
Figure 5-2. Schematics and Board Highlight of Coin Cell Holder



5.1.2.1 *Protection Circuits*

The power supply rail has reverse-voltage, overvoltage, short circuits, and excess current protection features, as shown in Figure 5-3.

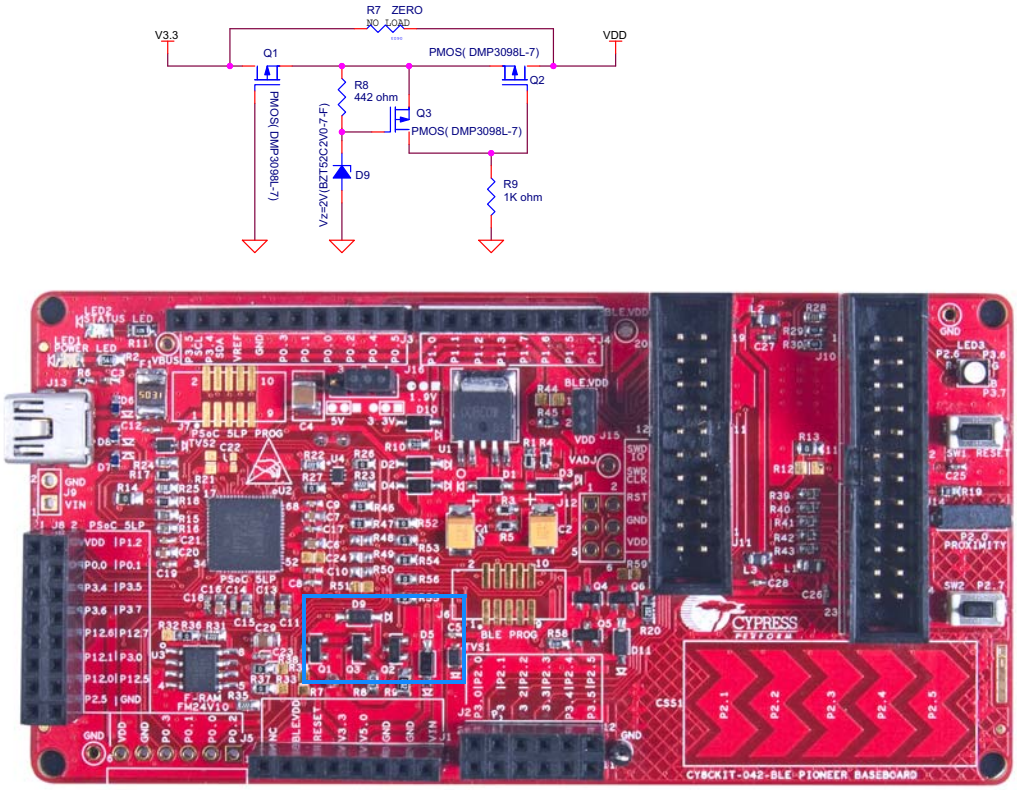
Figure 5-3. Power Supply Block Diagram With Protection Circuits



- A PTC resettable fuse is connected to protect the computer's USB ports from shorts and overcurrent.
- ORing diodes prevent damage to components when the board is powered from different voltage sources at the same time.
- ESD protection is provided for the USB mini-B connector.

- A MOSFET-based protection circuit is provided for overvoltage and reverse-voltage protection for the 3.3-V rail from J1.5, as shown in Figure 5-4. When a voltage greater than 3.6 V is applied from J1.5, the Q2 PMOS will turn off, which will cut off the power to the BLE Module from J1.5. When reverse voltage is applied from J1.5, the Q1 PMOS will turn off, protecting the onboard components from reverse voltage.

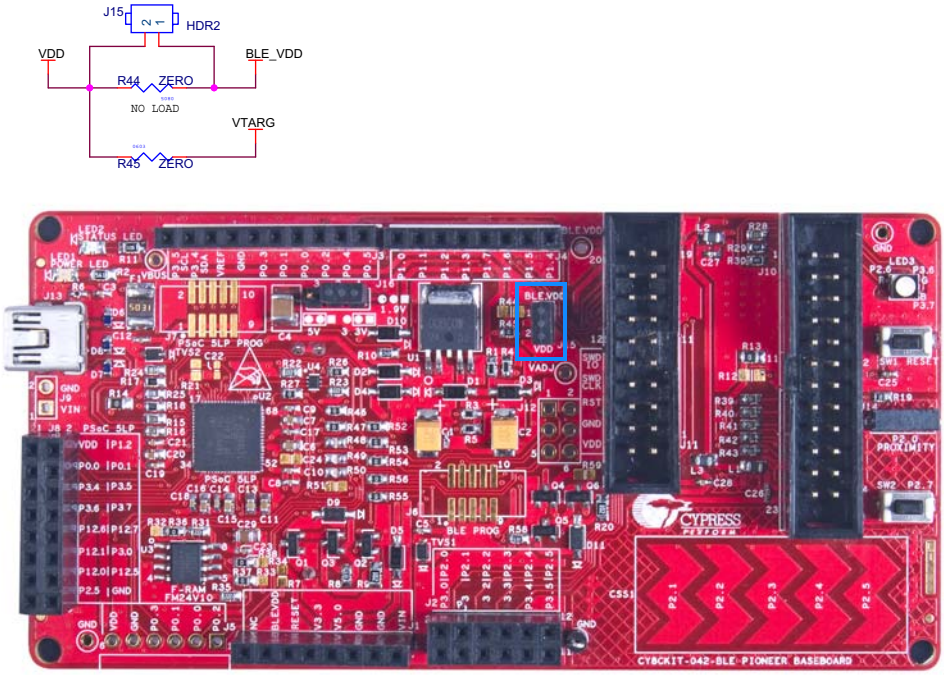
Figure 5-4. Schematics and Board Highlight of MOSFET Protection Circuit for 3.3-V Input



5.1.2.2 *Current Measurement Jumper*

To demonstrate the low power consumption of PSoC 4/PRoC BLE, a two-pin header is populated in series with the power supply to the PSoC 4 BLE. This can be used to measure current using an ammeter without the need to desolder any components from the board, as shown in Figure 5-5.

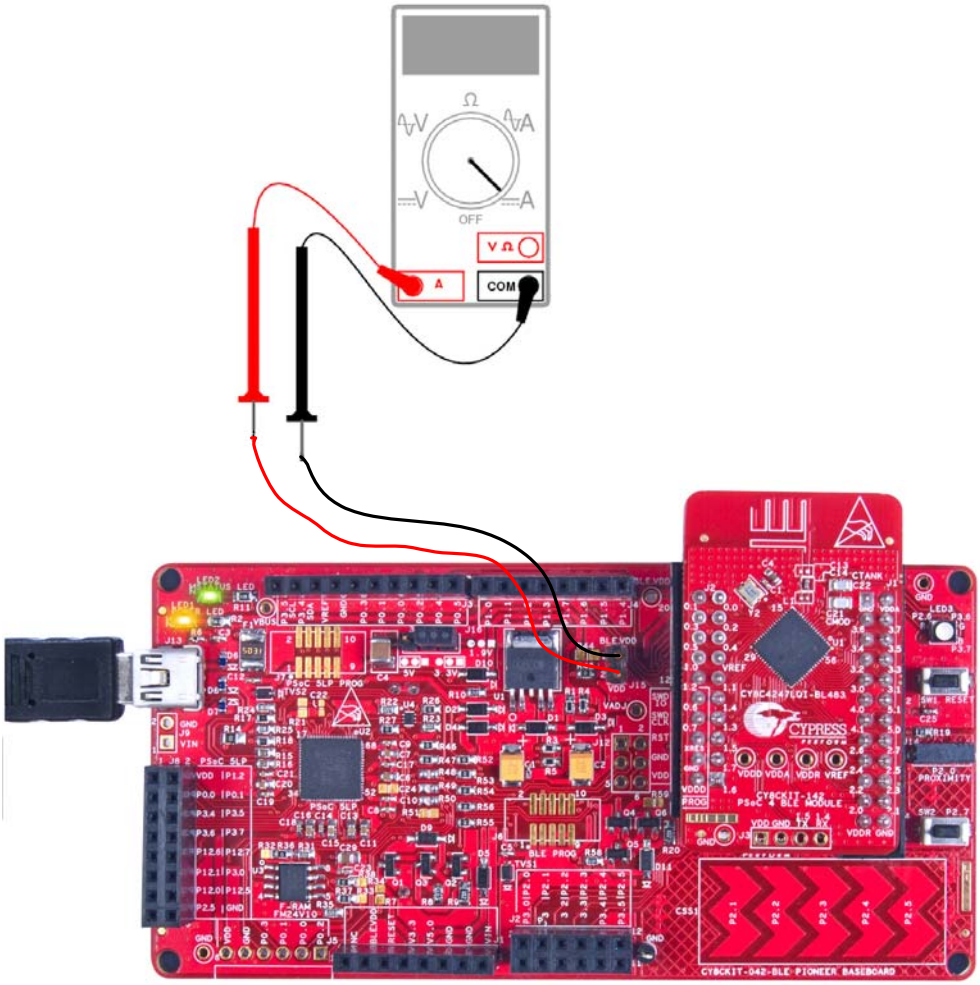
Figure 5-5. Schematics and Board Highlight of Current Measurement Jumper



The following methods are supported for measuring the current consumption of the BLE device.

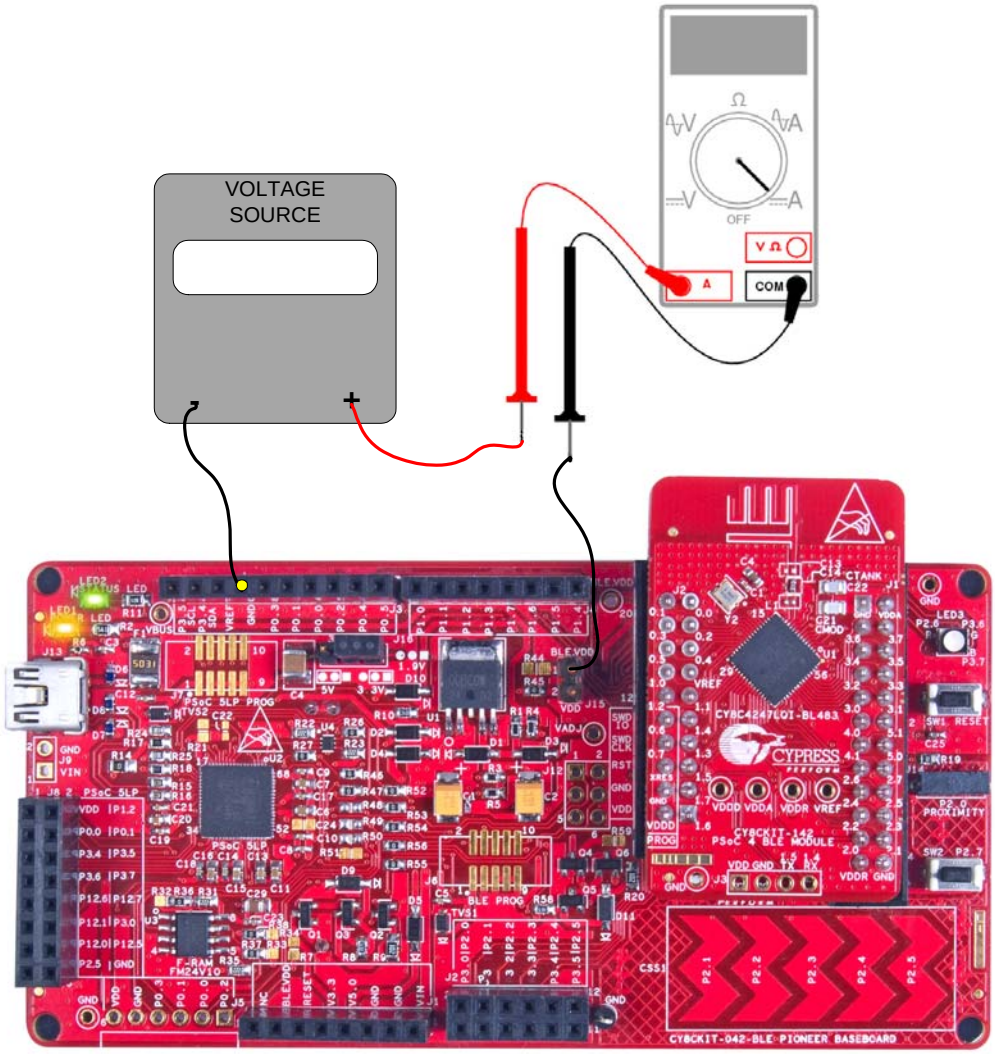
- When the board is powered through the USB port (J13), remove jumper J15 and connect an ammeter, as shown in Figure 5-6.

Figure 5-6. Current Measurement when Powered from USB Port



- When using a separate power supply for the BLE with USB powering (regulator output on the USB supply must be within 0.5 V of the separate power supply), remove jumper J13. Connect the positive terminal of the voltage supply to the positive terminal of the ammeter and the negative terminal of the ammeter to the upper pin of J15. Figure 5-7 shows the required connections.

Figure 5-7. Current Measurement when Powered Separately



5.1.3 Programming Interface

The kit allows you to program and debug the PSoC 4 BLE/PRoC BLE in two ways:

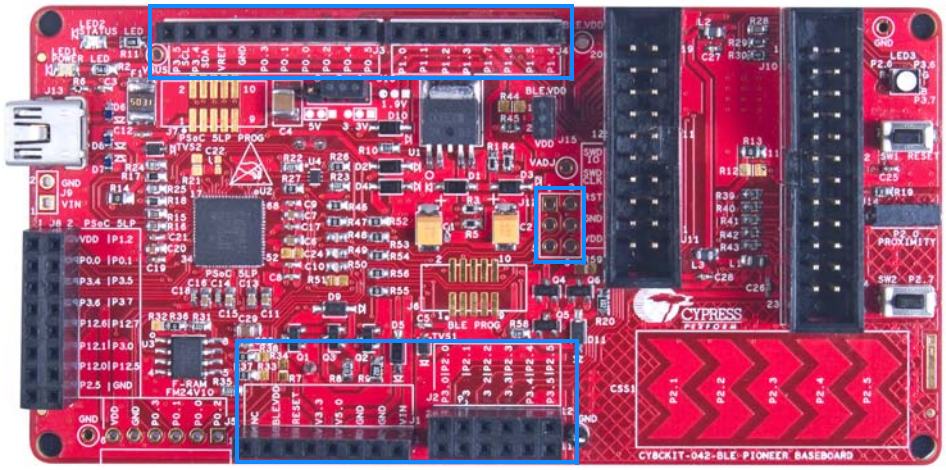
- Using the Onboard PSoC 5LP Programmer and Debugger
- Using CY8CKIT-002 MiniProg3 Programmer and Debugger

5.1.4 Expansion Connectors

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

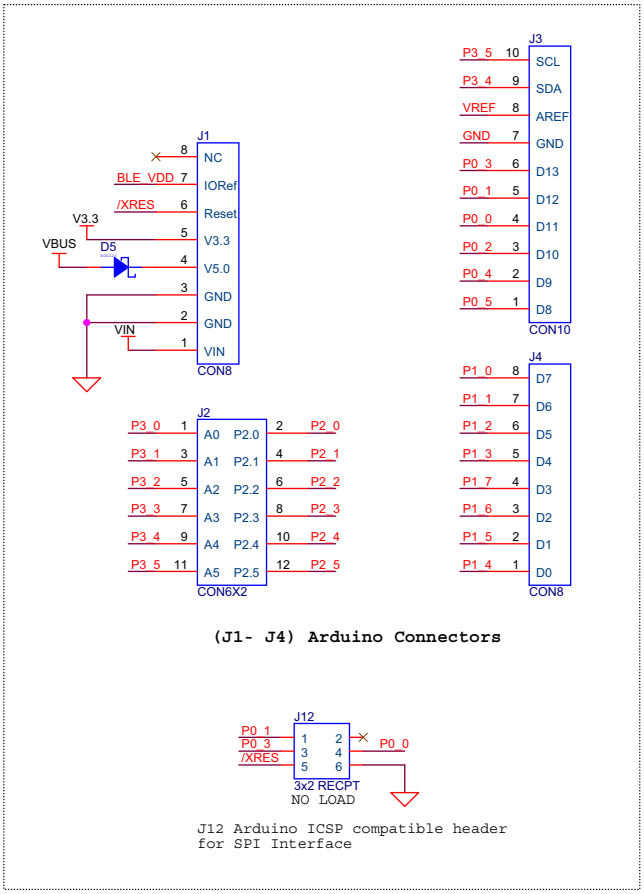
This kit has five Arduino compatible headers: J1, J2, J3, J4, and J12, as shown in Figure 5-8. You can develop applications based on the Arduino shield's hardware.

Figure 5-8. Arduino Headers



The J1 header contains I/O pins for reset, I/O reference voltage (IOREF), and power supply line. The J2 header is an analog port that contains I/O pins for SAR ADC, comparator, and opamp. The J3 header is primarily a digital port that contains I/O pins for PWM, I^2C, SPI, and analog reference. The J4 header is also a digital port that contains I/O pins for UART and PWM. The J12 header is an Arduino ICSP compatible header for the SPI interface and is not populated. Refer to the "No Load Components" section of Bill of Materials (BOM) on page 183 for the header part number.

Figure 5-9. Schematics of Arduino Connectors



Additional Functionality of Header J2

The J2 header is a 6×2 header that supports Arduino shields. The Port 2 and Port 3 pins of PSoC 4 BLE/PRoC BLE are brought to this header. The Port 2 pins also connect to the onboard CapSense slider through 560-ohm resistors. When the CapSense feature is not used, remove these resistors to help ensure better performance with these pins.

Functionality of Unpopulated Header J12

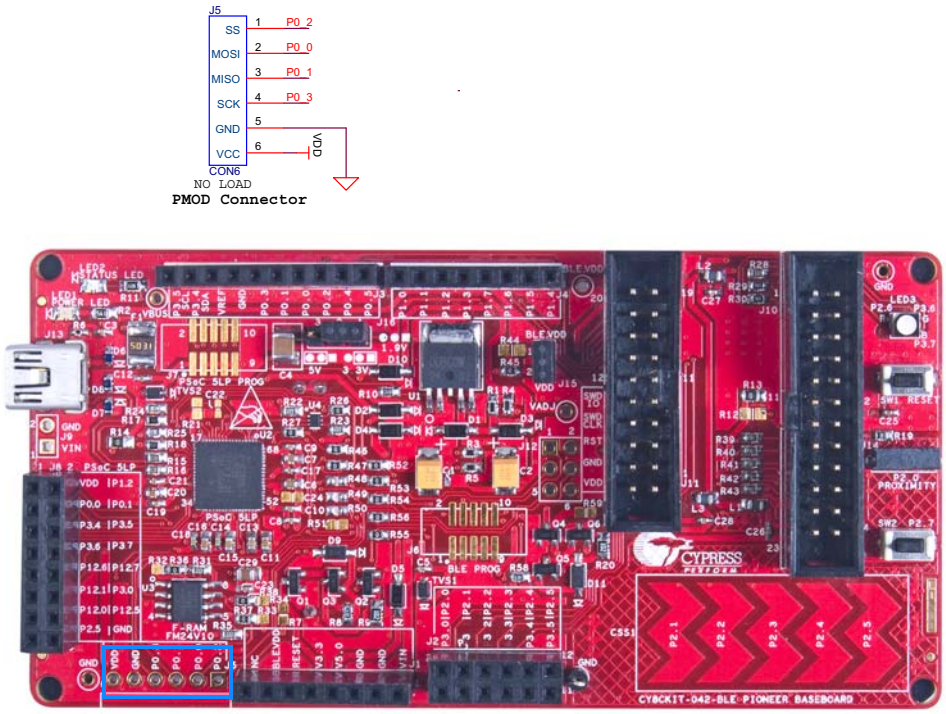
The J12 header is a 2×3 header that supports Arduino shields. It is used on a small subset of shields and is unpopulated on the BLE Pioneer board.

5.1.4.2 PMOD Connector - Digilent Pmod Compatible (J5-unpopulated)

This port supports Digilent Pmod peripheral modules (see Figure 5-10). Pmods are small I/O interfaces that connect with the embedded control boards through either 6- or 12-pin connectors. The BLE 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 BLE Pioneer board. You must populate this header before connecting the PMOD daughter cards. Refer to the “No Load Components” section of Bill of Materials (BOM) on page 183 for the header part number.

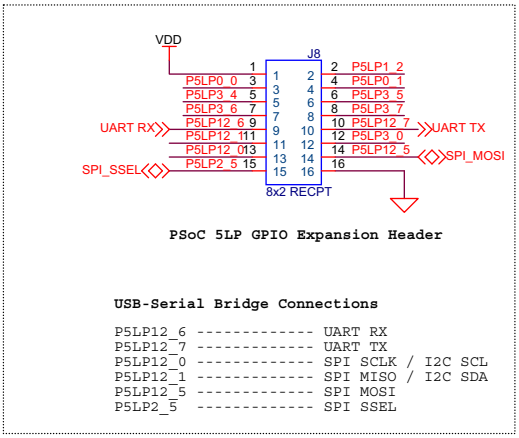
Figure 5-10. Schematics and Board Highlight of PMOD Connector

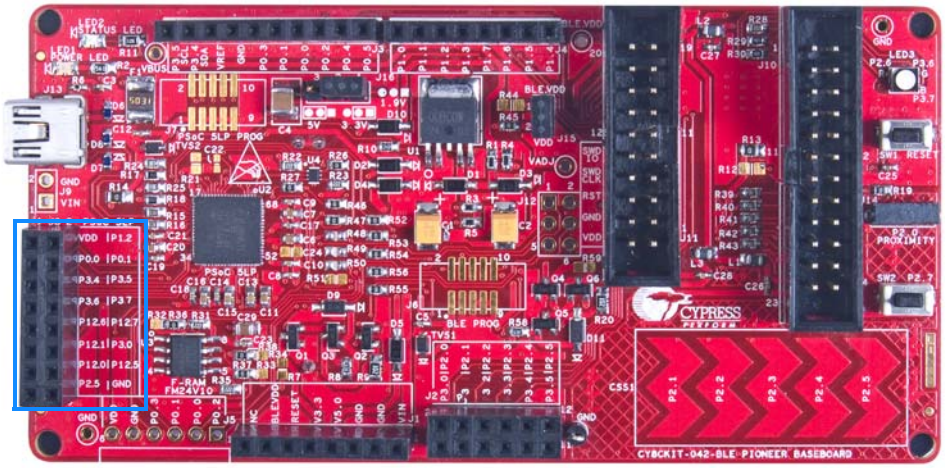


5.1.4.3 PSoC 5LP GPIO Header (J8)

An 8x2 header is provided on the board to pull out several pins of PSoC 5LP to support advanced features such as a low-speed oscilloscope and a low-speed digital logic analyzer (see Figure 5-11). This header also contains the USB-Serial bridge pins that can be used when these pins are not accessible on the Arduino headers because a shield is connected.

Figure 5-11. Schematics and Board Highlight of PSoC 5LP GPIO Expansion Header

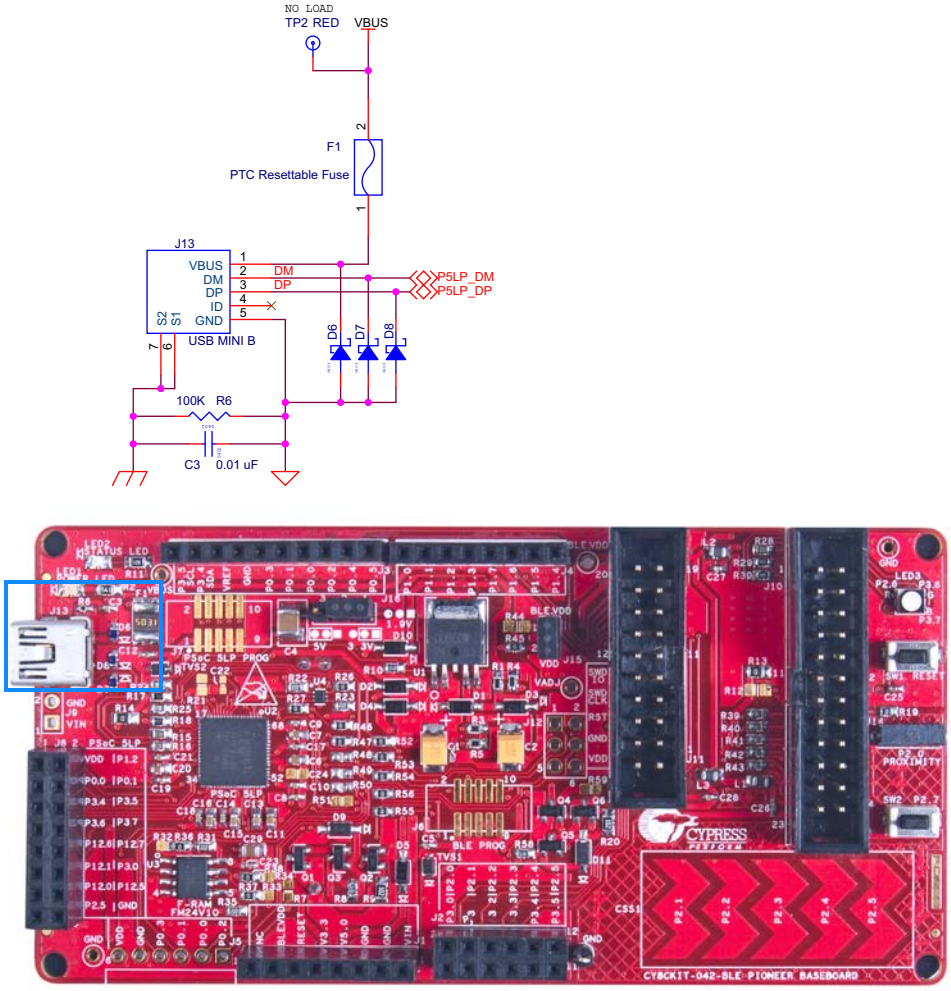




5.1.5 USB Mini-B Connector

The PSoC 5LP connects to the USB port of a PC through a mini-B connector (see Figure 5-12), which can also be used to power up the board. A resettable polyfuse is used to protect the computer's USB ports from shorts and overcurrent. If more than 500 mA is drawn from the USB port, the fuse will automatically break the connection until the short or overload is removed.

Figure 5-12. Schematics and Board Highlight of USB Mini-B Connector

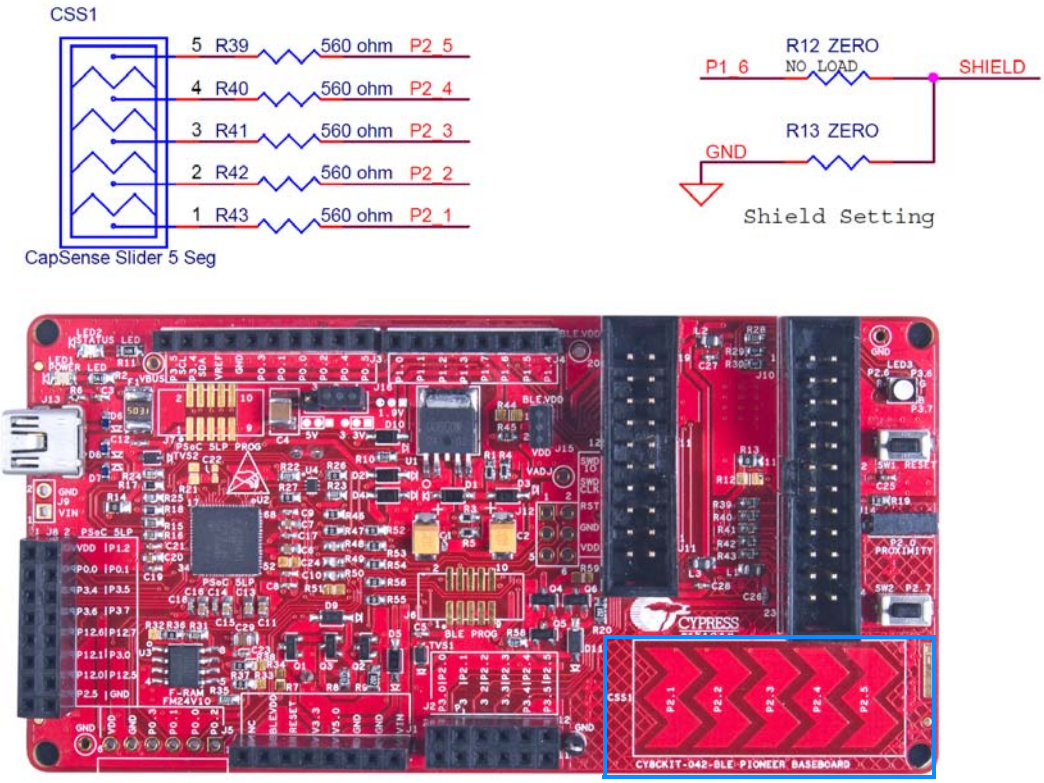


5.1.6 CapSense Circuit

5.1.6.1 CapSense Slider

The kit has a five-segment linear capacitive touch slider on the board, which is connected to BLE Module pins (see Figure 5-13). The CMOD and CTANK capacitors essential for CapSense functionality are connected in the Bluetooth modules (see BLE Module Board on page 103). A 2.2-nF capacitor is present on the CMOD pin, P4[0], for CapSense operation. This kit also supports CapSense designs that enable waterproofing. On this kit, the connection of the shield to the pin or to ground is made by resistors R12 and R13, respectively. By default, R13 is mounted on the board, which connects the shield to ground. Populate R12 and remove R13 when evaluating waterproofing designs, which will connect the shield to the designated pin, P1[6].

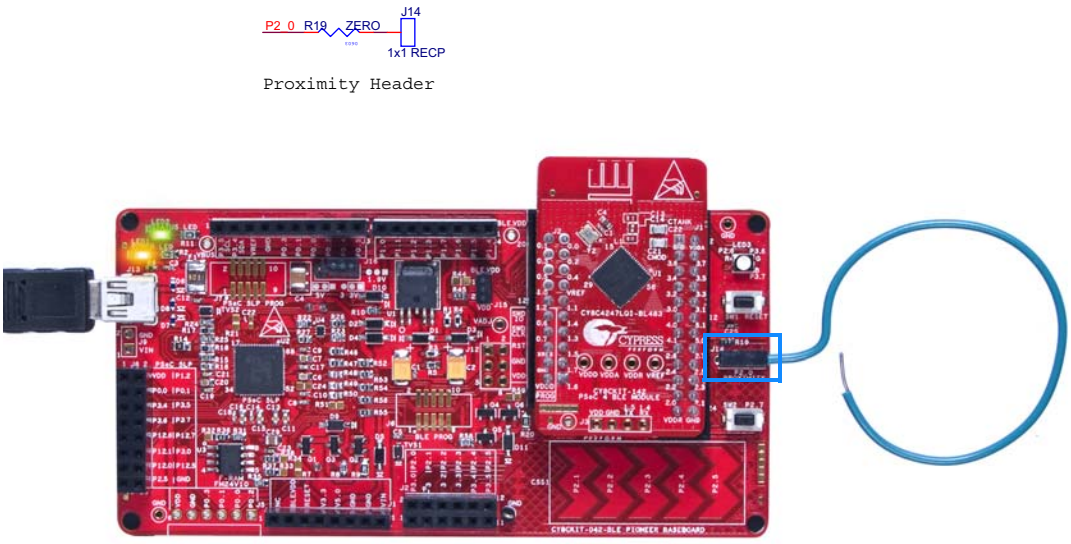
Figure 5-13. Schematics and Board Highlight of CapSense Slider and Shield Setting



Proximity Header

The baseboard contains a header (J14) for CapSense proximity wire connection (see Figure 5-14).

Figure 5-14. Schematics and Board Highlight of Proximity Header



5.1.7 Pioneer Board LEDs

The Pioneer board has three LEDs. A green LED (LED2) indicates the status of the programmer. An amber LED (LED1) indicates the status of power supplied to the board. The kit also has a general-purpose tricolor LED (LED3) for user applications that connect to specific BLE pins. Figure 5-15 and Figure 5-16 show the schematics of these LEDs.

Figure 5-15. Schematics of Status and Power LED

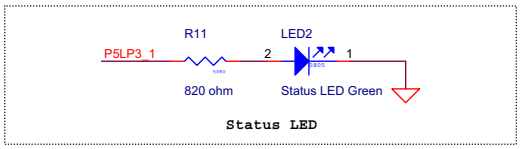
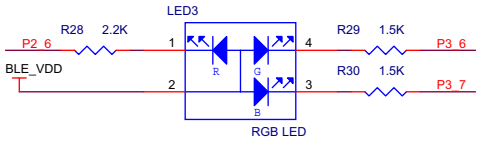
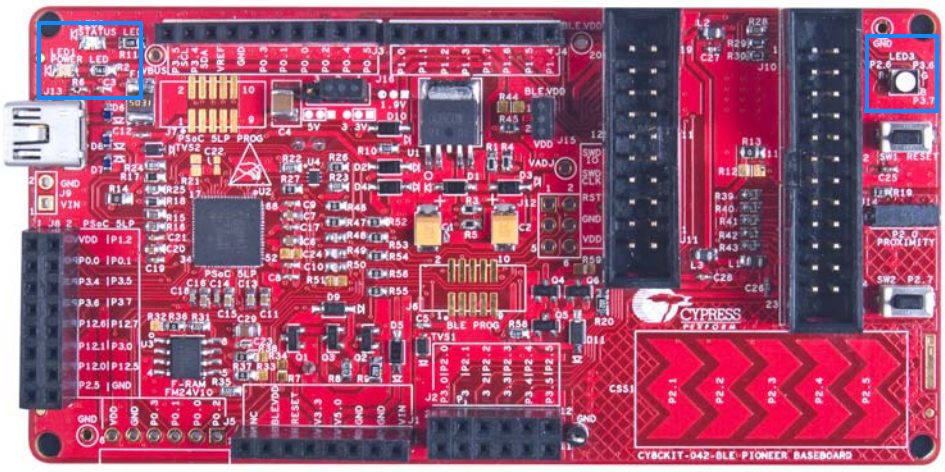


Figure 5-16. Schematics and Board Highlight of RGB LED

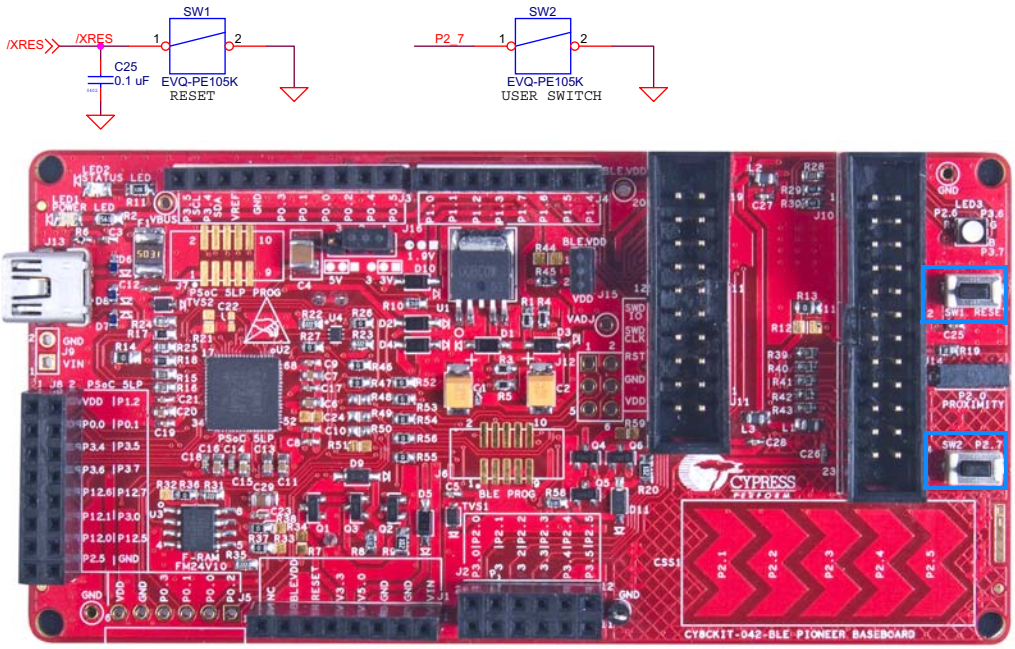




5.1.8 Push Buttons

The board contains a reset push button and a user push button, as shown in Figure 5-17. The reset button is connected to the XRES pin of BLE and is used to reset the BLE device. The user button is connected to P2[7] of the BLE device. Both the push buttons connect to ground on activation (active low).

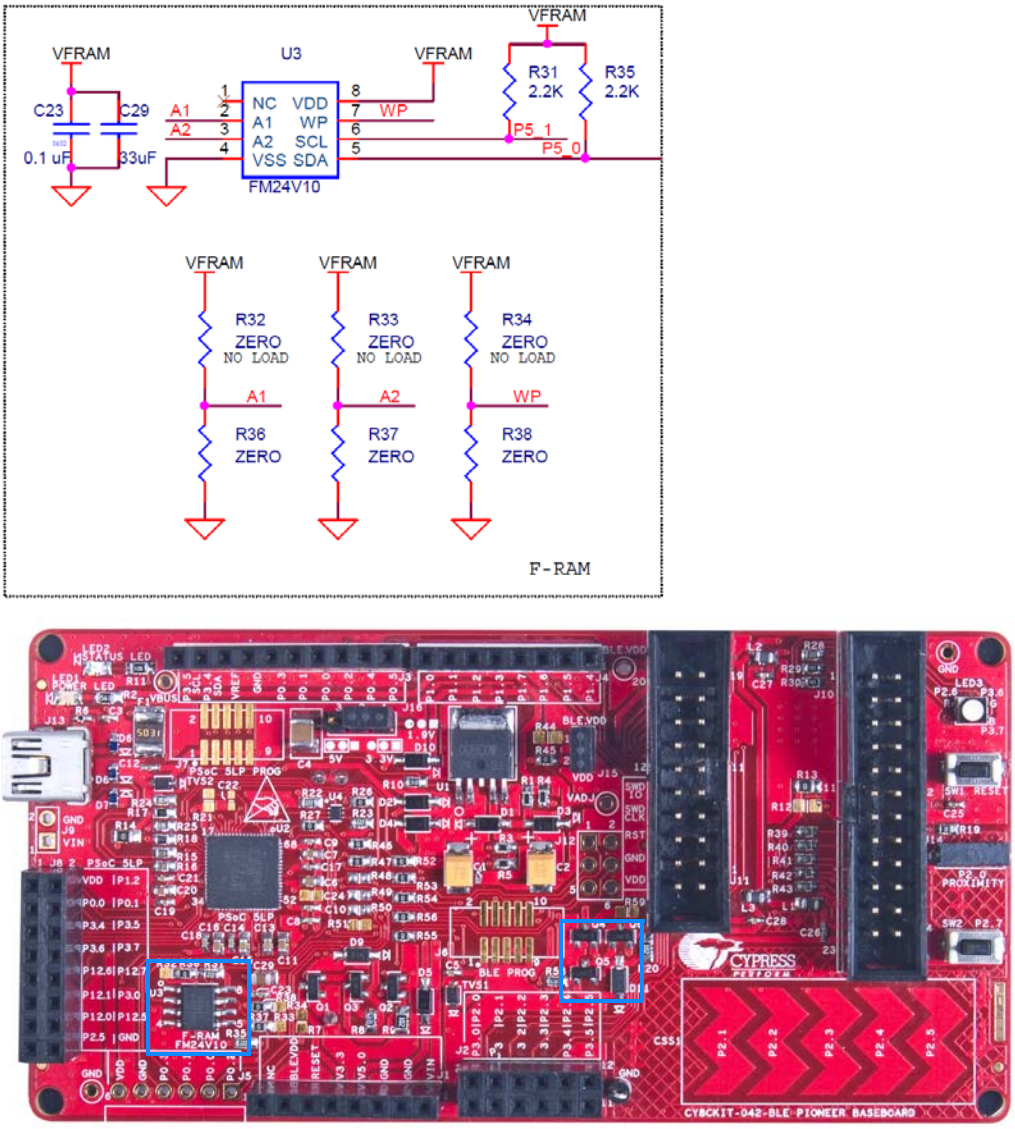
Figure 5-17. Schematics and Board Highlight of Reset Button and User Push Button



5.1.9 Cypress Ferroelectric RAM (F-RAM)

The baseboard contains an F-RAM device (FM24V10) (see Figure 5-18) that can be accessed through I^2C lines P5[0] and P5[1] of the PSoC 4 BLE/PRoC BLE device. The F-RAM is 1-Mbit (128 KB) with an I^2C speed up to 1 Mbps. The I^2C slave address of the F-RAM device is seven bits wide, and the LSB two bits are configurable through physical pins and are hardwired to 000 on the board. By default, the address of the F-RAM device used on the board is 0x50. This address can be modified by changing the R32/R36 and R33/R37 pairs. The operating voltage range of the F-RAM is between 2 V and 3.6 V. To prevent the application of 5 V from the adjustable LDO regulator on the board, a MOSFET based protection circuit similar to the one used for the 3.3-V rail is connected between the output of the regulator and the VDD pin of the F-RAM. The protection circuit cuts off the power to the F-RAM when the output of the regulator is greater than 3.6 V.

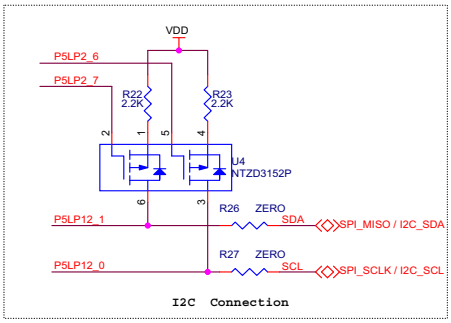
Figure 5-18. Schematics and Board Highlight of F-RAM

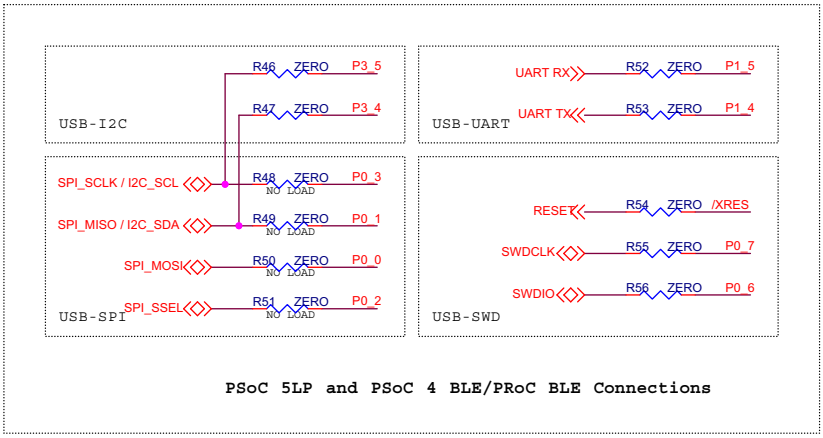


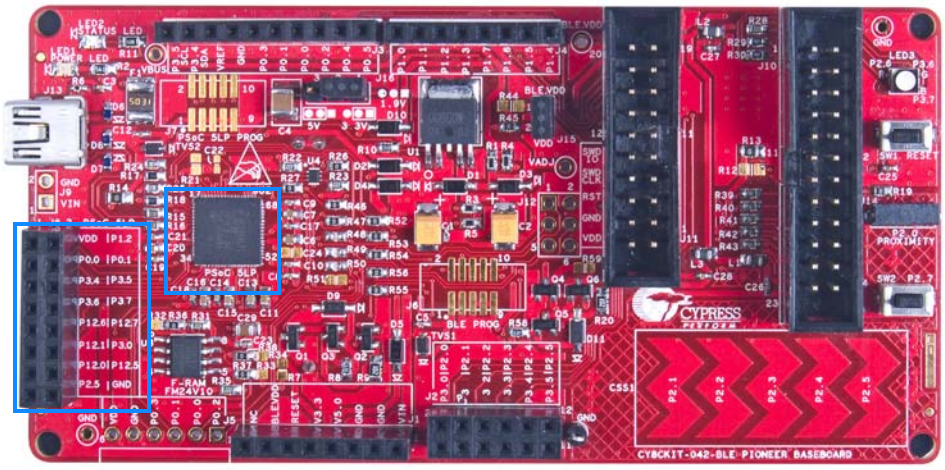
5.1.10 Serial Interconnection Between PSoC 5LP and Bluetooth Module

In addition to use as an onboard programmer, the PSoC 5LP is used as a USB-Serial interface for the USB-UART, USB-I^{2}C, and USB-SPI bridges (see Figure 5-19). The I^{2}C bus contains firmware-controlled pull-ups using FET, which can be enabled or disabled using the PSoC 5LP pins. The USB-Serial pins of the PSoC 5LP are also available on the Arduino header; therefore, the PSoC 5LP can be used to control Arduino shields with the SPI/I^{2}C/UART interface. Moreover, any shield that uses these pins for their functionality can be interfaced with PSoC 5LP by bootloading PSoC 5LP with custom firmware.

Figure 5-19. Schematics and Board Highlight of Serial Interface and I²C Pull-Up via FET



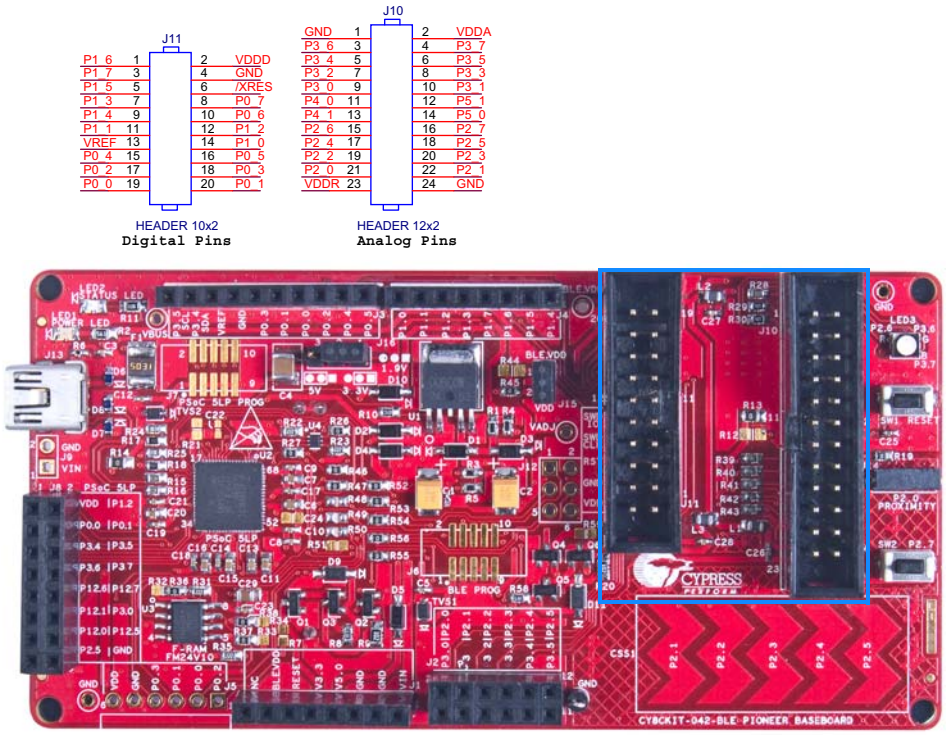




5.1.11 Bluetooth Module Headers

The PSoC 4 BLE and PRoC BLE Modules are connected to the Pioneer board using the two (24-pin and 20-pin) Bluetooth module headers, as shown in Figure 5-20.

Figure 5-20. Schematics and Board Highlight of Bluetooth Module Headers

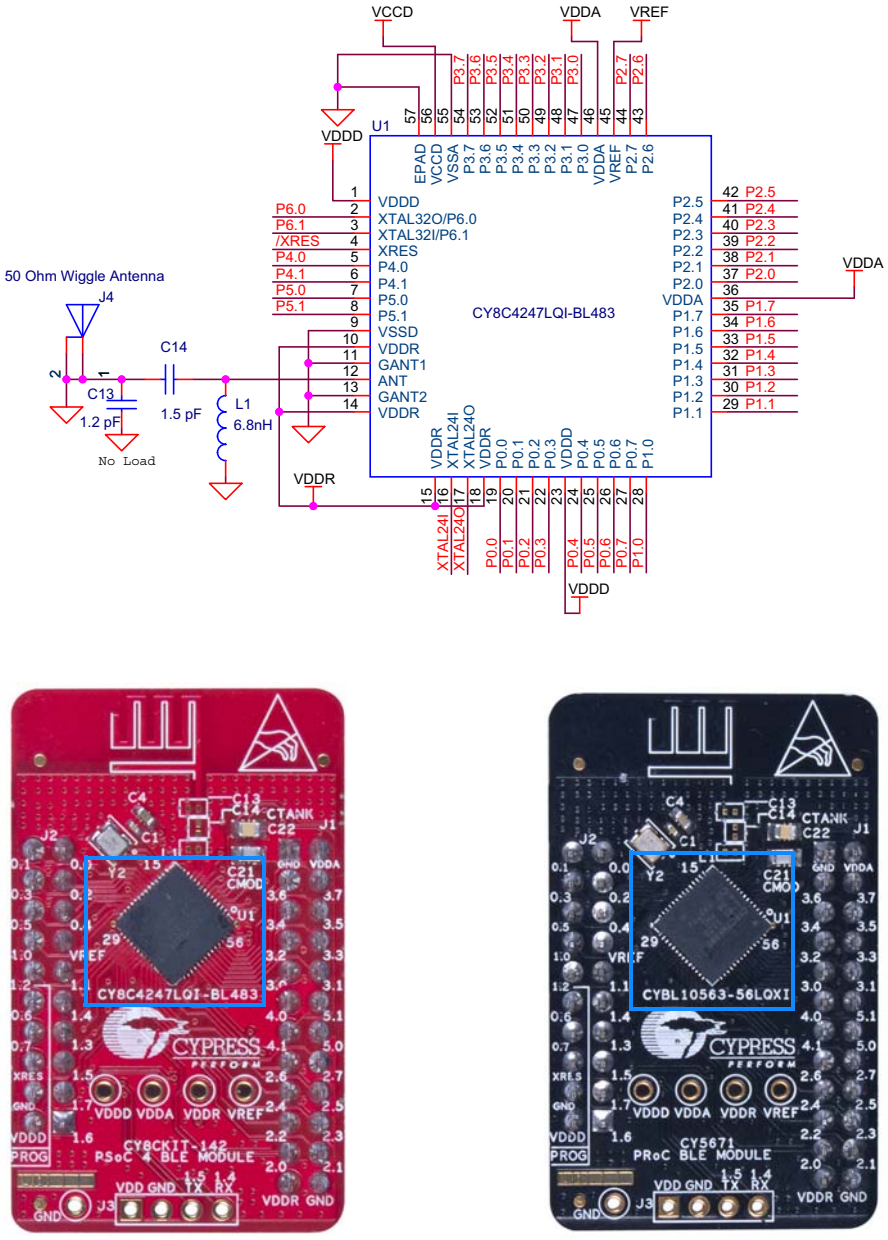


5.2 BLE Module Board

5.2.1 PSoC 4 BLE or PRoC BLE

The PRoC BLE or PSoC 4 BLE is the main component on the BLE Module. It provides the RF interface and analog and digital capability. The PRoC BLE or PSoC 4 BLE pins are mapped to the Bluetooth module headers (see Figure 5-21). For more information, refer to the BLE web page.

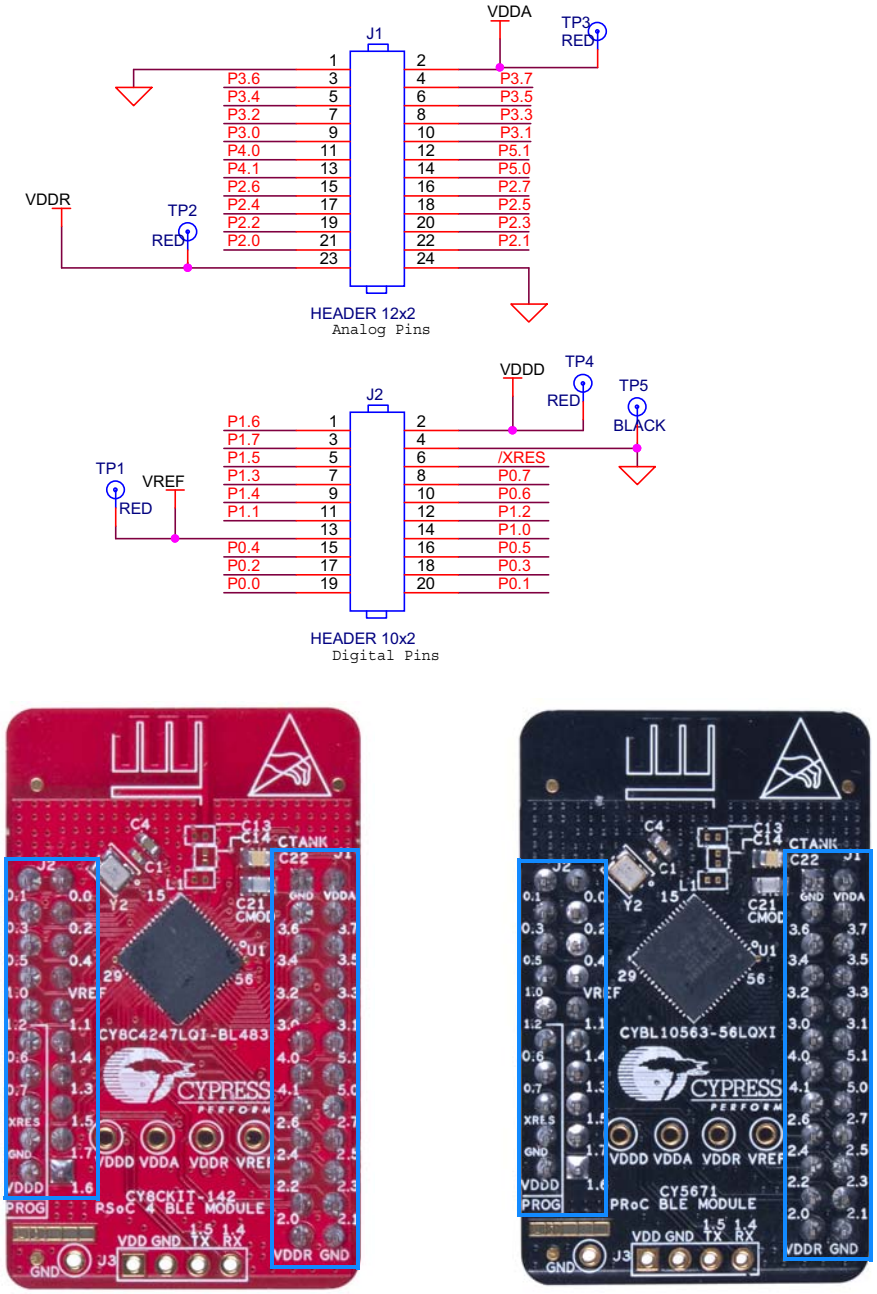
Figure 5-21. Schematics and Board Highlight of Bluetooth Module Headers for BLE pins



5.2.2 Bluetooth Module Headers (20-Pin and 24-Pin Headers)

The PSoC 4 BLE and PRoC BLE Modules connect to the Pioneer board using the two (20-pin and 24-pin) Bluetooth module headers (Figure 5-22). All GPIOs and power domains are brought out to these headers. These headers are the counterparts of the connectors in section 5.1.4 .

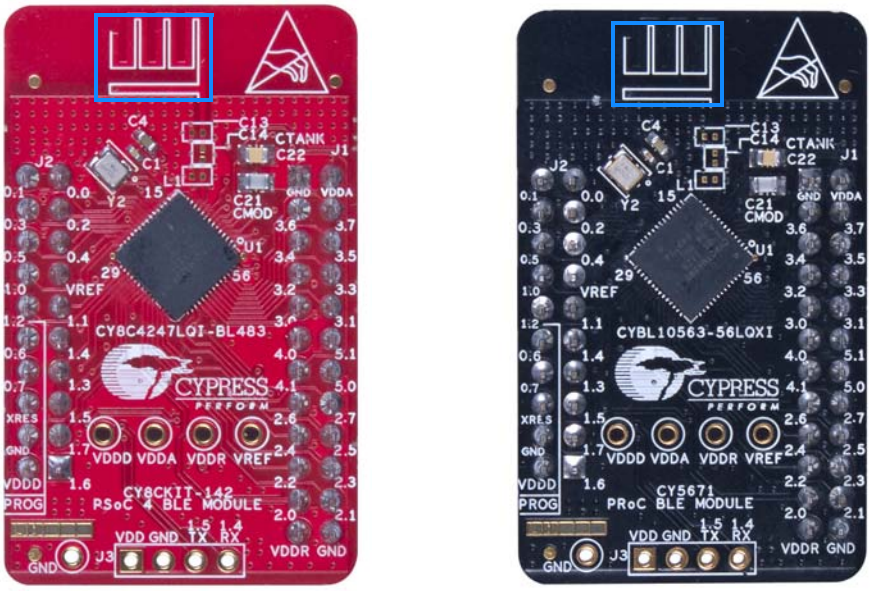
Figure 5-22. Schematics and Board Highlight of Headers



5.2.3 Wiggle Antenna

The Modules/Dongle uses the wiggle antenna. Refer to the Antenna Design Guide (AN91445) for details.

Figure 5-23. Board Highlight of Wiggle Antenna

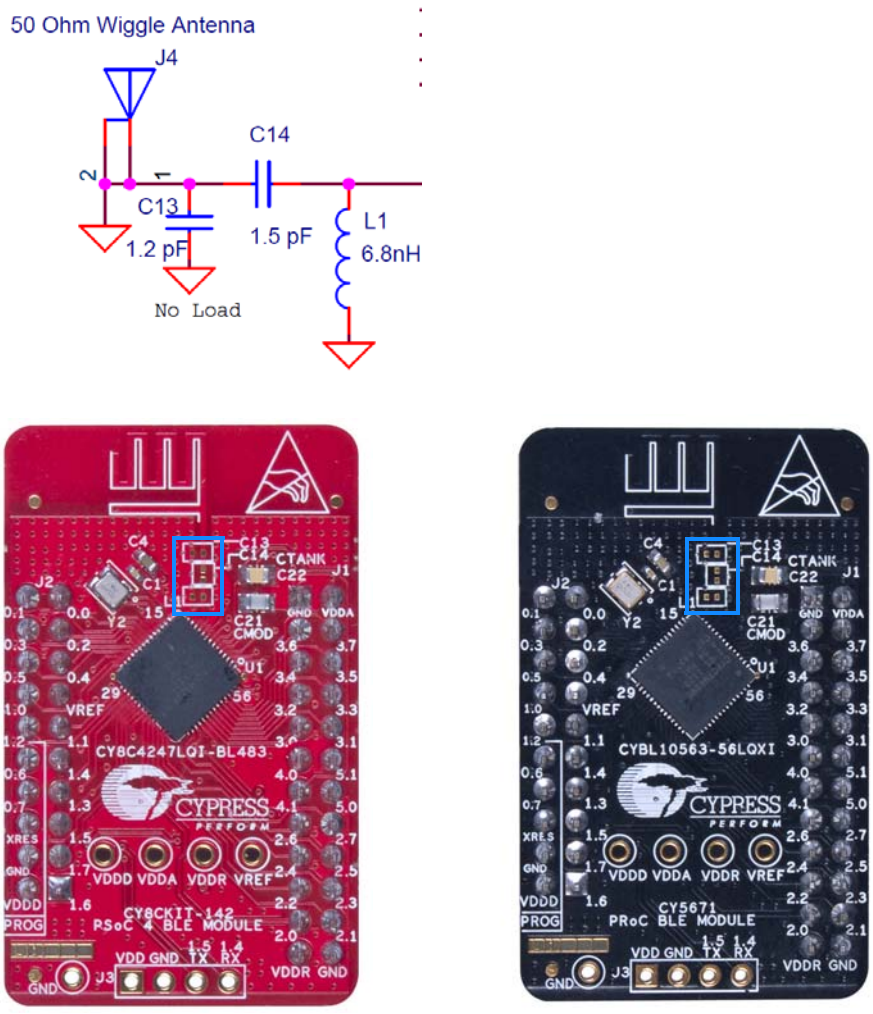


5.2.4 Antenna Matching Network

An Antenna Matching Network is required between the BLE device and the antenna to achieve optimum performance (Figure 5-24). The matching network has four main tasks:

- Transform the balanced output of the radio to an unbalanced connection to the antenna (balun).
- Transform the output impedance of the radio to a 50-ohm antenna.
- Suppress harmonics to a level below the regulations level in TX mode.
- Suppress the LO leakage in RX mode.

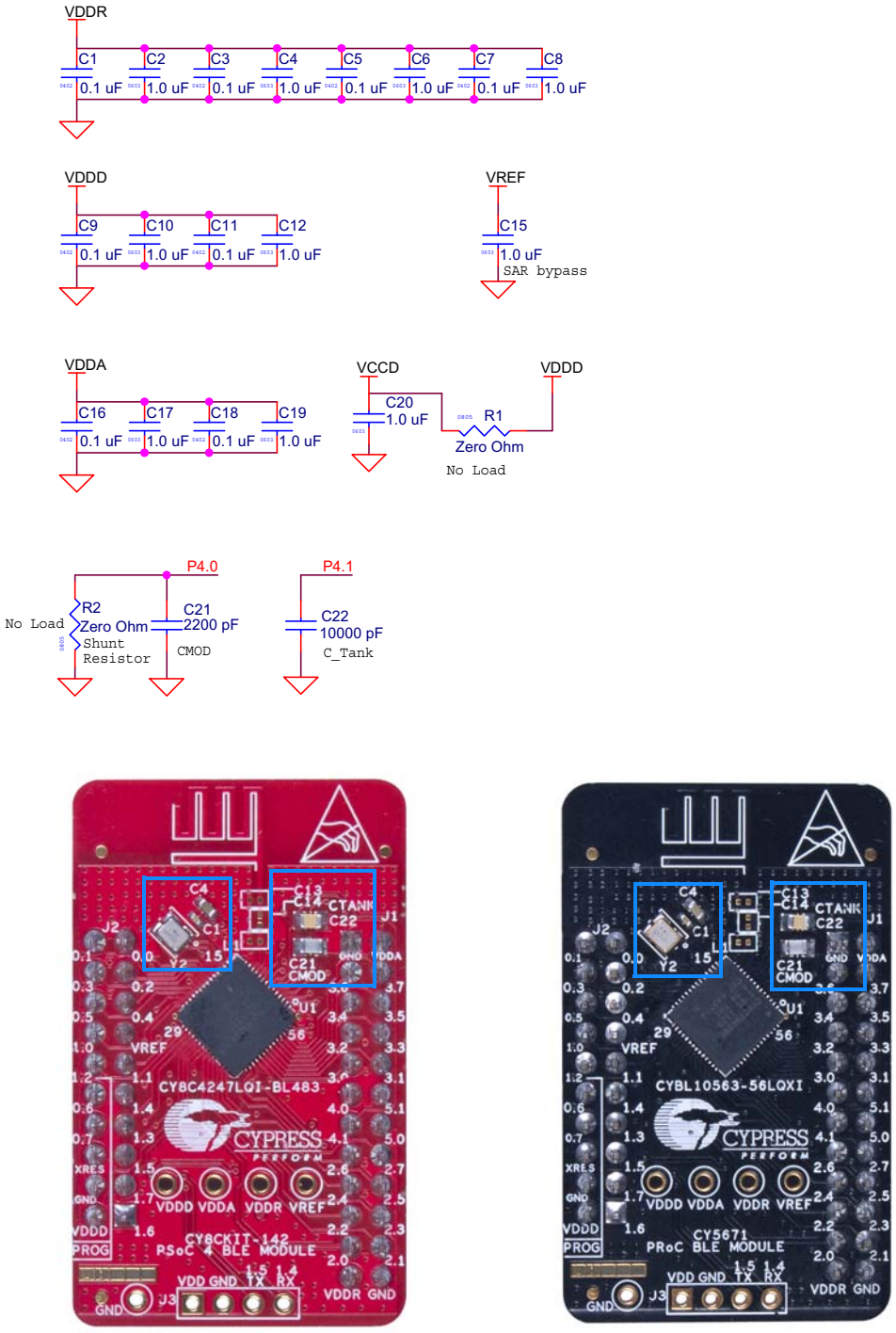
Figure 5-24. Schematics and Board Highlight of Antenna Matching Network and Antenna



5.2.5 BLE Passives

Module boards include a 24-MHz crystal and a 32-kHz crystal, the CMOD and shield capacitor circuit for CapSense, a SAR bypass capacitor, and adequate decoupling capacitors for all power domains, as shown in Figure 5-25.

Figure 5-25. Schematics and Board Highlight of External Crystal, CMOD, CTANK, Decaps, Jumpers



5.2.6 Test Points

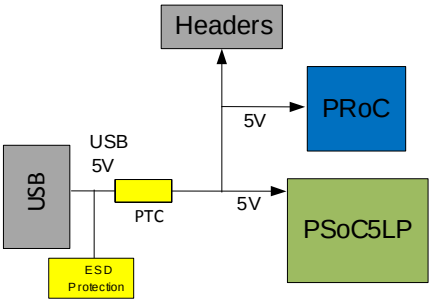
All power domains are brought out as test points for easy probing.

5.3 BLE Dongle Board

5.3.1 Power System

The board is powered directly using 5 V from the USB port, as shown in Figure 5-26.

Figure 5-26. Power Supply Block Diagram With Protection Circuits



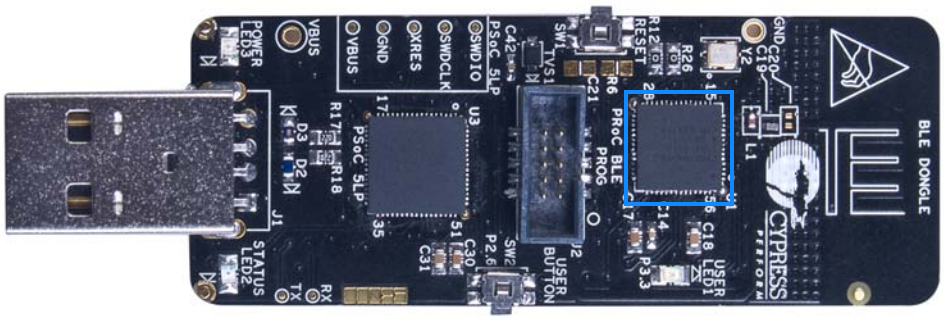
5.3.1.1 Protection Circuits

The PTC resettable fuse is connected to protect the computer's USB ports from shorts and overcurrent.

5.3.2 PRoC BLE

See PSoC 4 BLE or PRoC BLE on page 103.

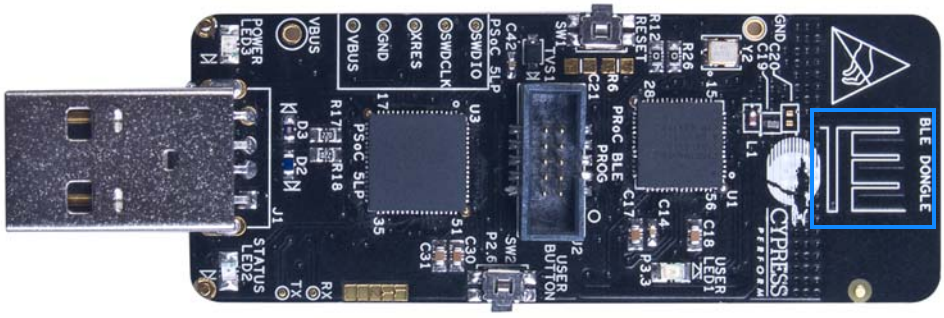
Figure 5-27. Board Highlight of PRoC BLE



5.3.3 Wiggle Antenna

See Wiggle Antenna on page 105.

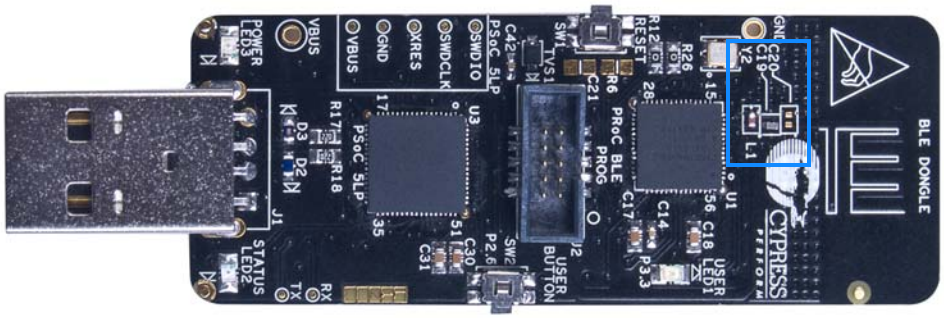
Figure 5-28. Board Highlight of Wiggle Antenna



5.3.4 Antenna Matching Network

See Antenna Matching Network on page 106.

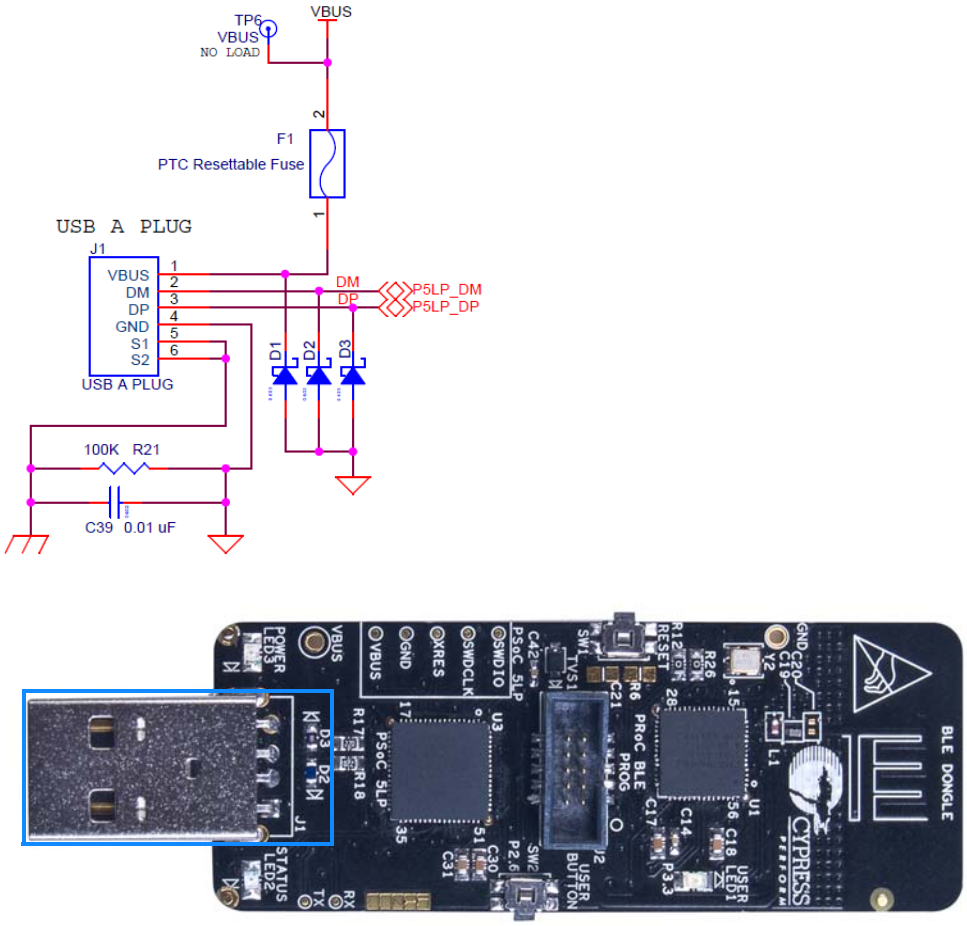
Figure 5-29. Board Highlight of Antenna Matching Network



5.3.5 USB Type A Plug

The PSoC 5LP connects to the USB port of a PC through a USB type A plug (Figure 5-30). This plug can also be used to power up the board. A resettable polyfuse is used to protect the computer's USB ports from shorts and overcurrent. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed. The VBUS, D+, and D– lines from the USB connector are also protected against ESD events using TVS diodes.

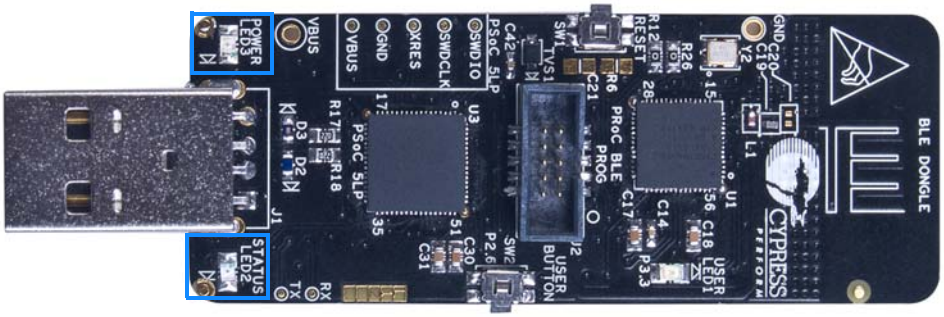
Figure 5-30. Schematics and Board Highlight of USB Type A Plug



5.3.6 System Status LED and Power LED

See Pioneer Board LEDs on page 97.

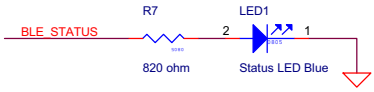
Figure 5-31. Board Highlight of Power and Status LED

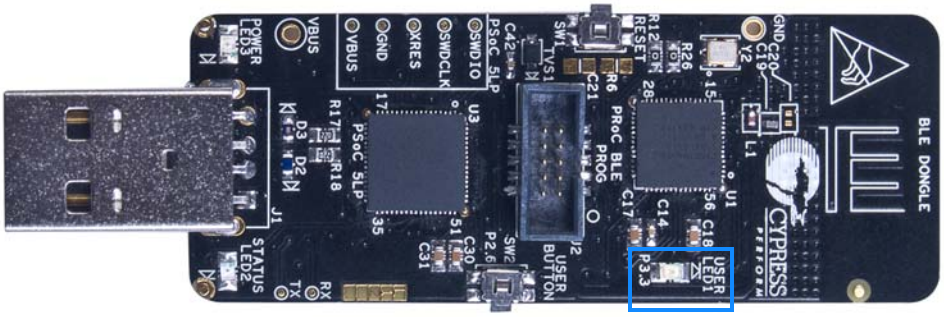


5.3.7 User LED

A user LED is provided to indicate status from the PRoC BLE device (Figure 5-32). It is also used to show the bind status.

Figure 5-32. Schematics and Board Highlight of User LED

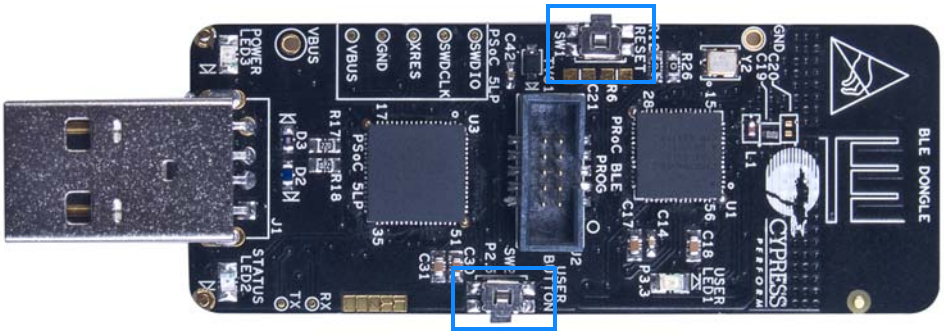




5.3.8 Push Buttons

See Push Buttons on page 98.

Figure 5-33. Push Buttons



6. Advanced Topics

This chapter describes advanced features of the BLE Pioneer kit as well as the corresponding projects. It can be used as reference to exploit these features for other applications, according to project requirements.

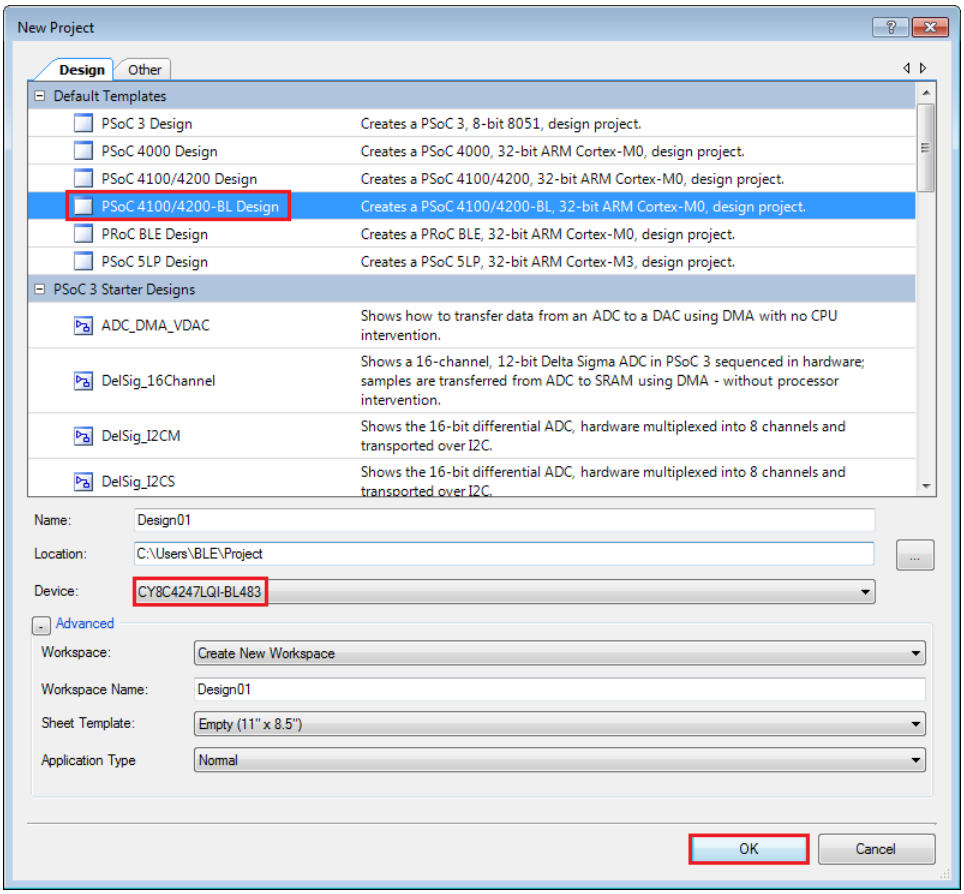
6.1 Using PSoC 5LP as USB-UART Bridge

The PSoC 5LP serves as a USB-UART bridge, which can communicate with the COM terminal software. This section explains how to create a PSoC 4 BLE code example to communicate with the COM terminal software.

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

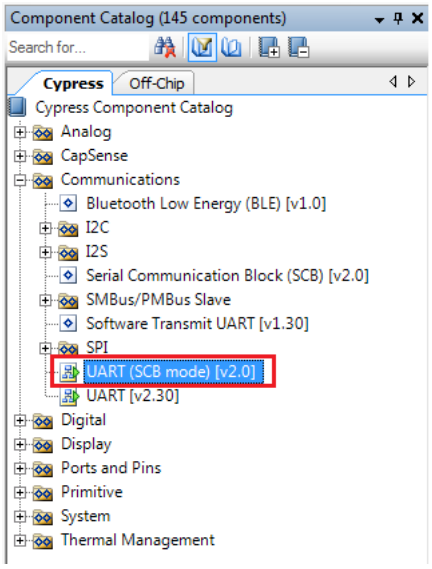
1. Create a new PSoC 4 BLE project in PSoC Creator, as shown in Figure 6-1. Select an appropriate location for your project and rename the project as required.

Figure 6-1. Create New Project in PSoC Creator



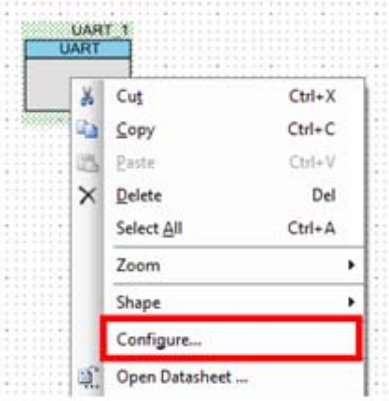
2. Drag and drop a UART (SCB) component (Figure 6-2) to the TopDesign.

Figure 6-2. UART Component in Component Catalog



3. To configure the UART, double-click or right-click on the UART component and select **Configure**, as shown in Figure 6-3.

Figure 6-3. Open UART Configuration Window



4. Change the instance name to UART. Configure the UART as shown in Figure 6-4, Figure 6-5, and Figure 6-6. Click **OK**.

Figure 6-4. UART Configuration Tab Window

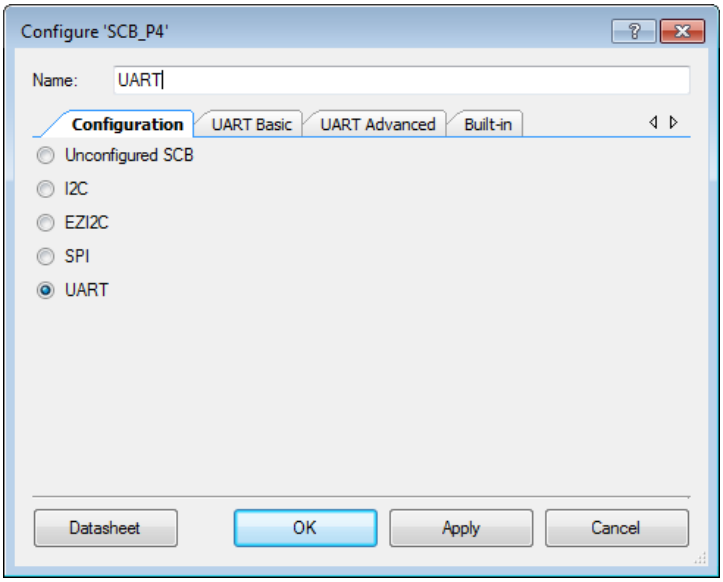


Figure 6-5. UART Basic Tab Window

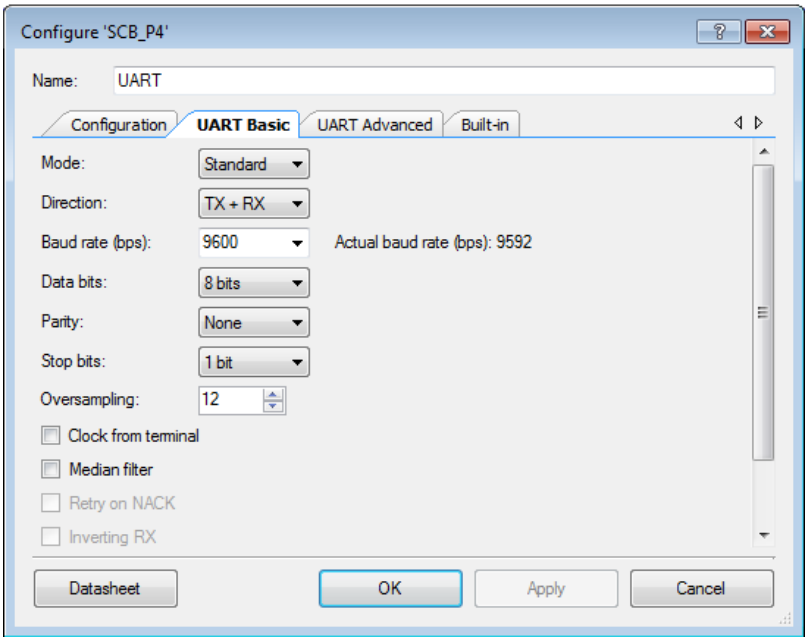
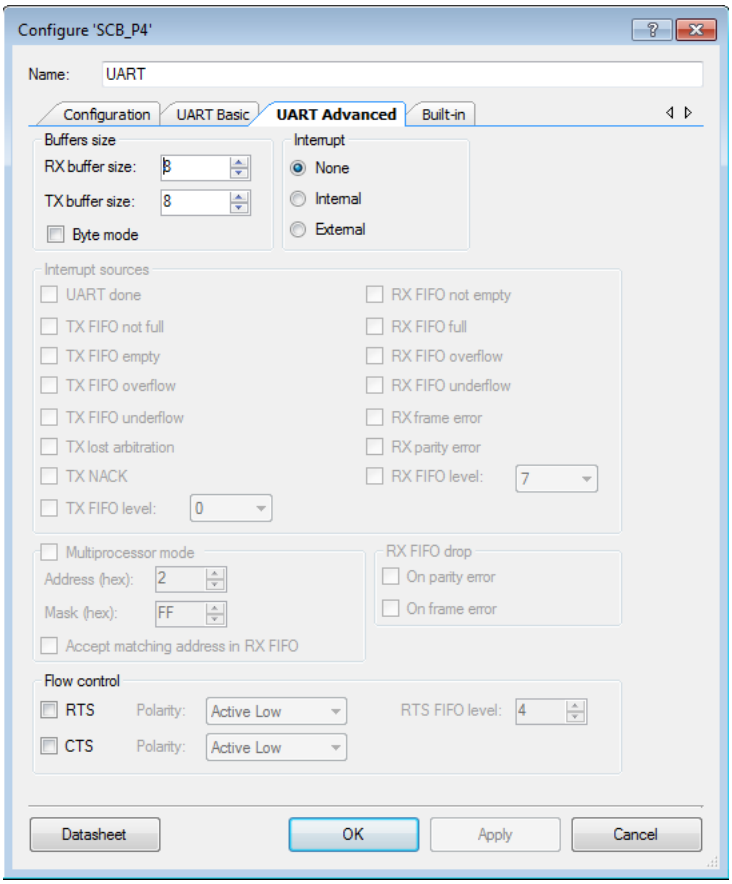


Figure 6-6. UART Advanced Tab Window



5. Select P1[4] for UART RX and P1[5] for UART TX in the **Pins** tab of <*Project_Name*>.*cydwr*, as shown in Figure 6-7.

Figure 6-7. Pin Selection

Alias	Name	Port	Pin	Lock
	\UART_1:rx\	P1[4] OA3:vminus, TCPWM2:line_out, SCB0:uart_rx, SCB0:i2c_sda, SCB0:spi_mosi	32	☑
	\UART_1:tx\	P1[5] OA3:vplus, TCPWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi_miso	33	☑

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

```
int main()
{
    uint8 ch;

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

/* Transmit String through UART TX Line */
UART_UartPutString("CY8CKIT-042-BLE 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 function is blocking and waits until
there is an entry into the TX FIFO. */
            UART_UartPutChar(ch);
        }
    }
}
```

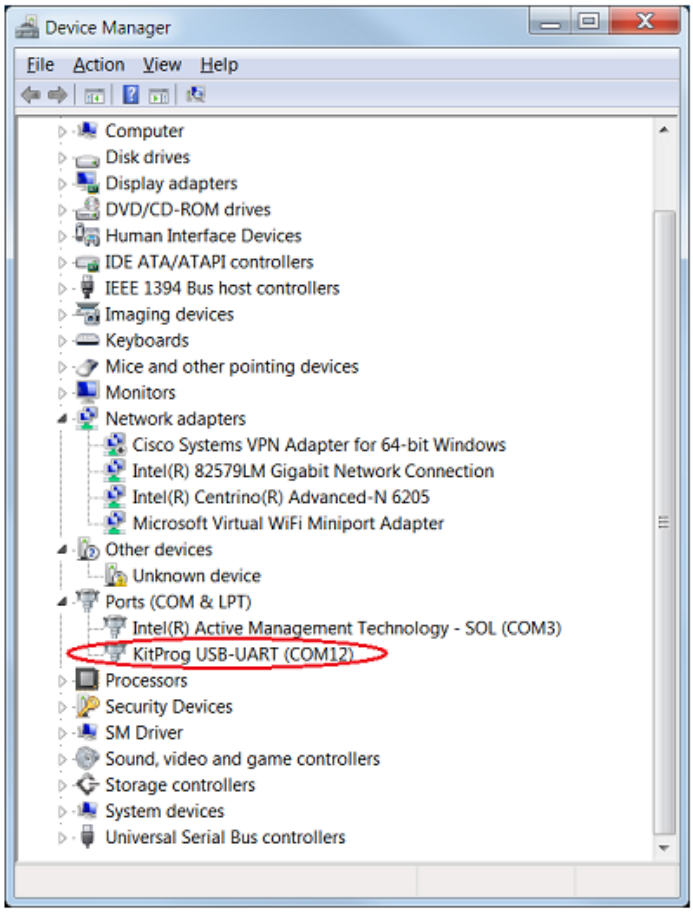
7. Build the project by clicking **Build > Build {Project Name}** or **[Shift][F6]**. After the project is built without errors and warnings, program (by choosing **Debug > Program**) the project to PSoC 4 BLE/PRoC BLE through the PSoC 5LP USB programmer or MiniProg3.

   Note: UART RX and UART TX can be routed to any digital pin on PSoC 4 BLE/PRoC BLE based on the configuration of the UART component. An SCB implementation of UART will route the RX and TX pins to one of the following subsets: (P0[0], P0[1] or P0[4], P0[5] or P1[4], P1[5] or P3[0], P3[1] or P3[4], P3[5] or P5[0], P5[1]).

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

1. Connect USB mini-B to J13. The kit enumerates as a **KitProg USB-UART** and is available in the **Device Manager, Ports (COM & LPT)**. A communication port is assigned to the **KitProg USB-UART**, as shown in Figure 6-8.

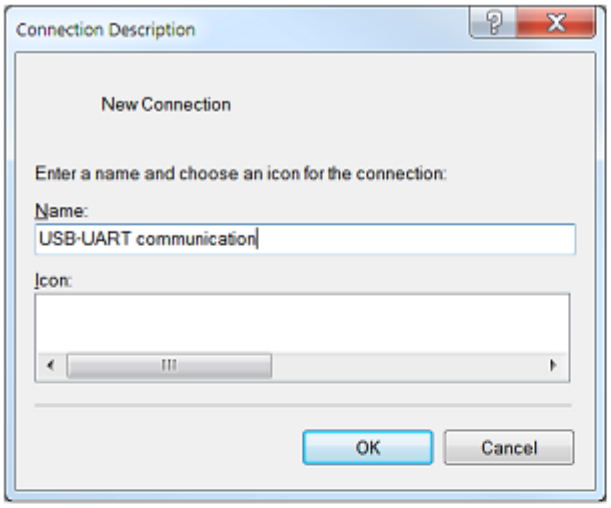
Figure 6-8. KitProg USB-UART in Device Manager



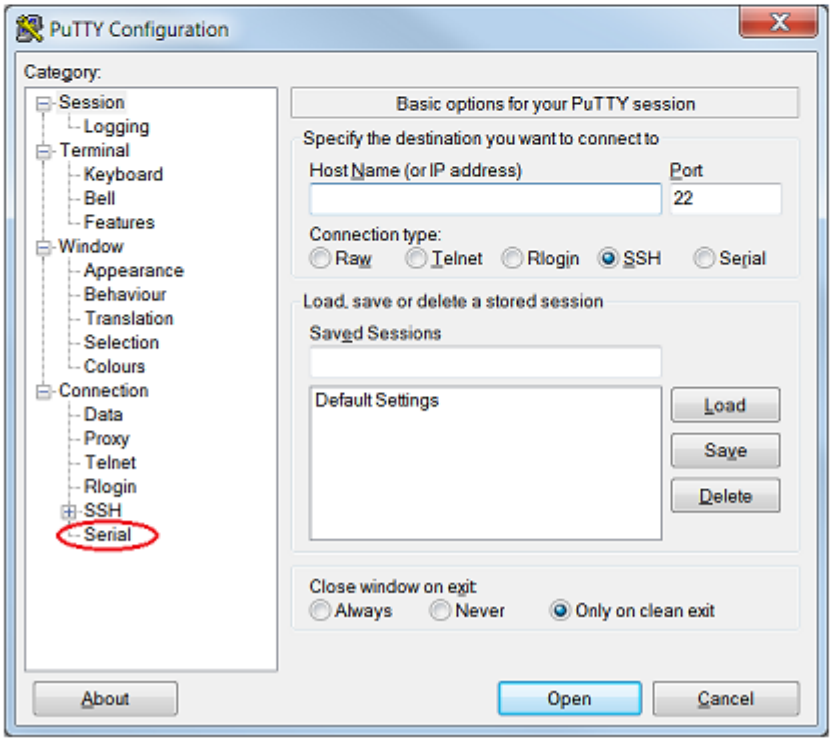
2. Open HyperTerminal and choose **File > New Connection** and enter a name for the new connection and click **OK**, as shown in Figure 6-9. For PuTTY, double-click the PuTTY icon and select **Serial** under **Connection**.

Figure 6-9. Open New Connection

HyperTerminal



PuTTY

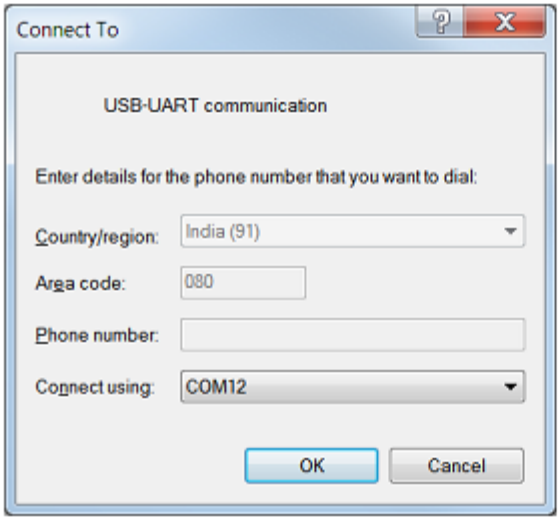


3. A new window opens, where the communication port can be selected.

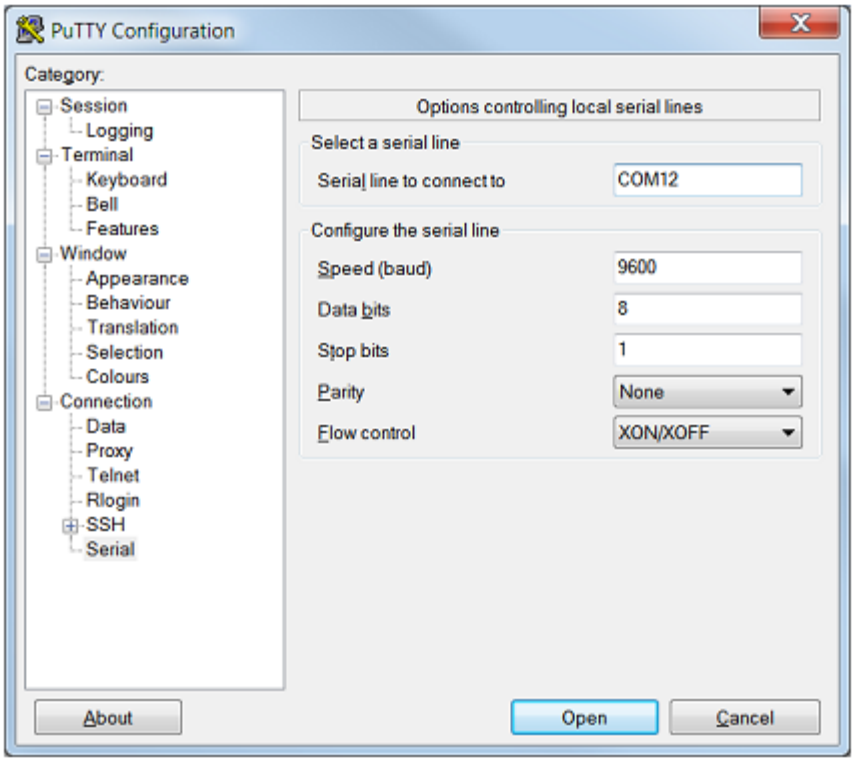
   In HyperTerminal, select COMx (or the specific communication port that is assigned to the KitProg USB-UART) in **Connect using** and click **OK**, as shown in Figure 6-10. In PuTTY enter the COMx in **Serial line to connect to**. This code example uses **COM12**.

Figure 6-10. Select Communication Port

HyperTerminal



PuTTY

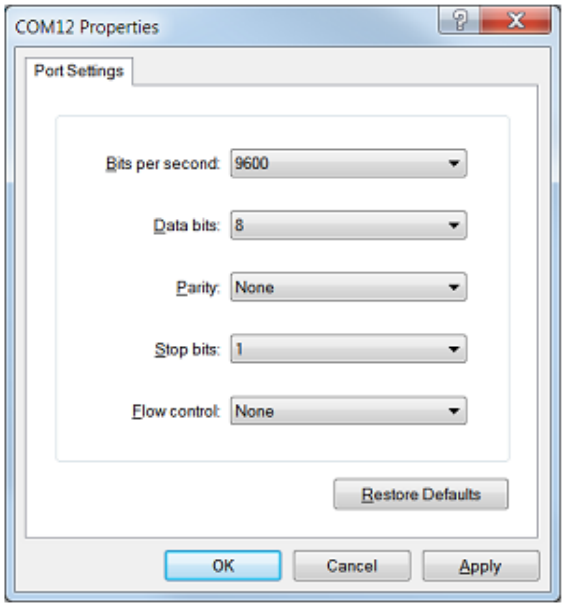


4. In HyperTerminal, select **Bits per second**, **Data bits**, **Parity**, **Stop bits**, and **Flow control** under **Port Settings** and click **OK**, as shown in Figure 6-11. Make sure that the settings are identical to the UART settings configured for the BLE device.

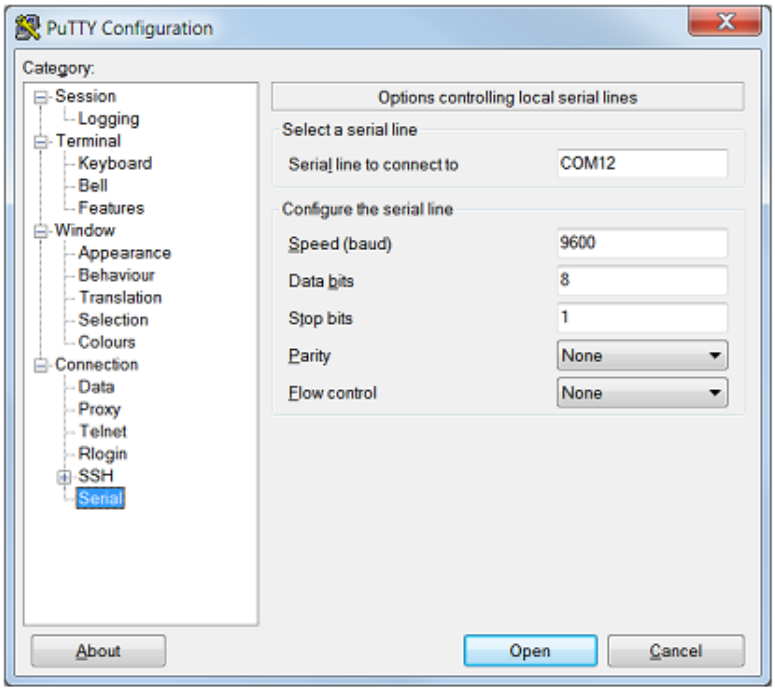
   In PuTTY select **Speed (baud)**, **Data bits**, **Stop bits**, **Parity** and **Flow control** under **Configure the serial line**. Click **Session** and select **Serial** under **Connection type**. **Serial line** shows the communication port (COM12) and **Speed** shows the baud rate selected. Click **Open** to start the communication.

Figure 6-11. Configure the Communication Port

HyperTerminal



PuTTY



5. Enable **Echo typed characters locally** in **File > Properties > Settings > ASCII Setup**, to display the typed characters on HyperTerminal, as shown in Figure 6-12. In PuTTY, select **Force on** in **Terminal > Line discipline options** to display the typed characters on PuTTY, as shown in Figure 6-13.

Figure 6-12. Enable Echo of Typed Characters in HyperTerminal

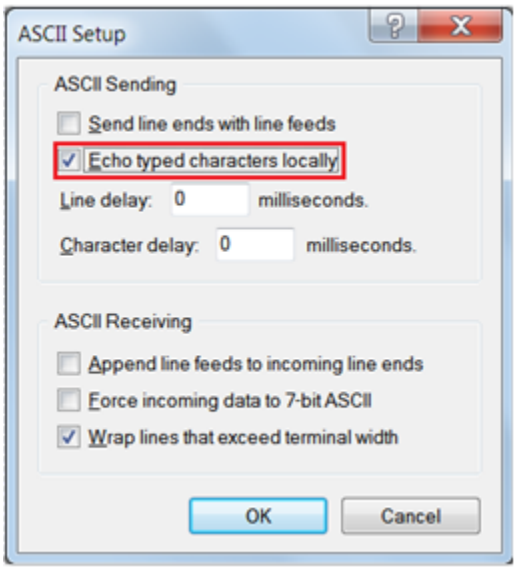
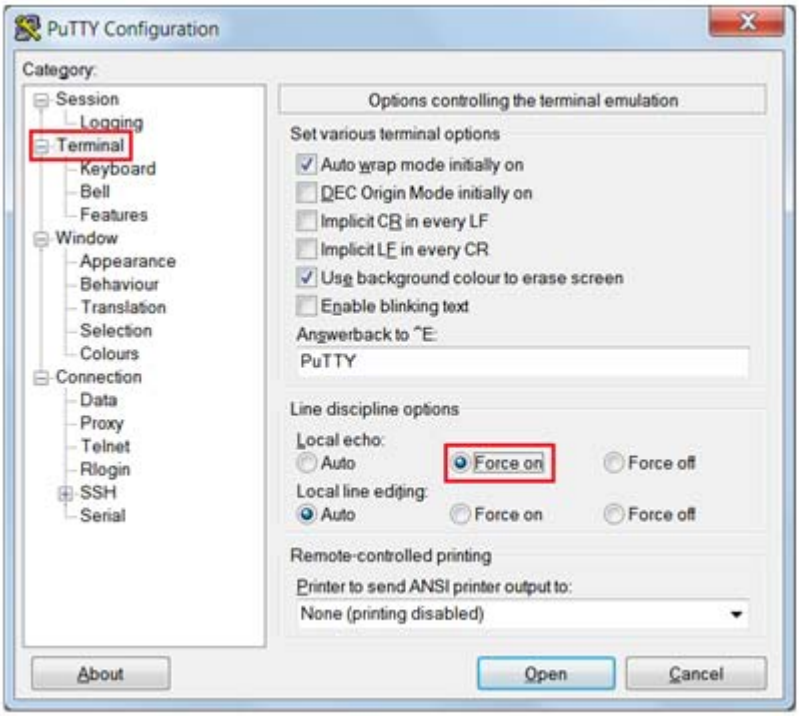


Figure 6-13. Enabling Echo of Typed Characters in PuTTY



6. The COM terminal software displays both the typed data and the echoed data from the PSoC 4 BLE or PRoC BLE UART, as shown in Figure 6-14 and Figure 6-15.

Figure 6-14. Data Displayed on HyperTerminal

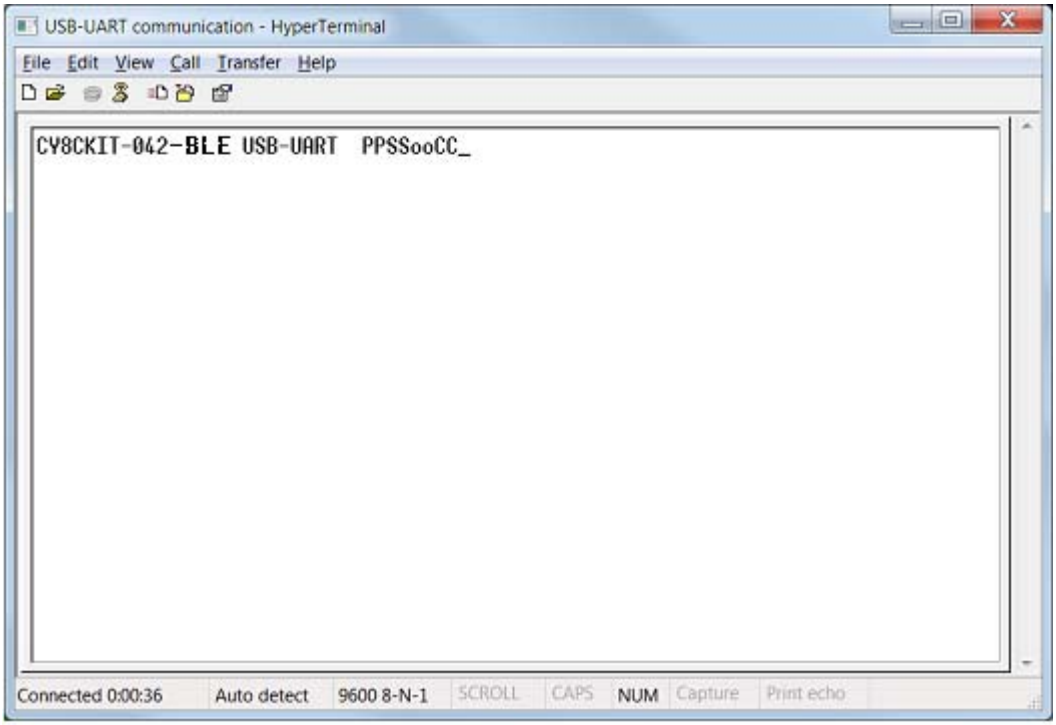
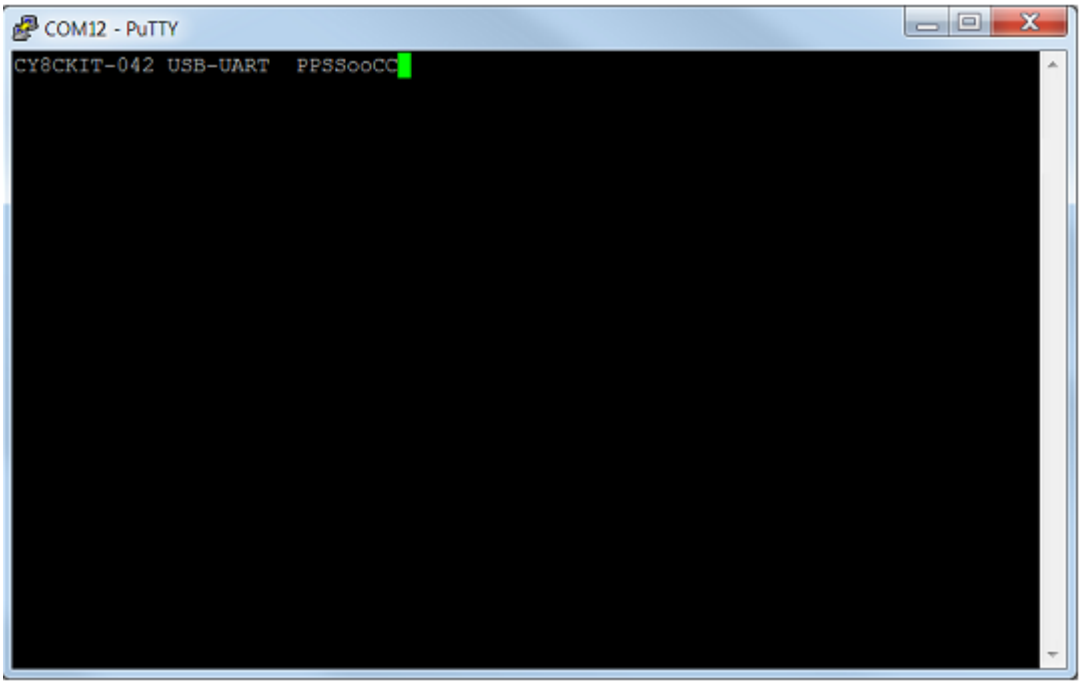


Figure 6-15. Data Displayed on PuTTY

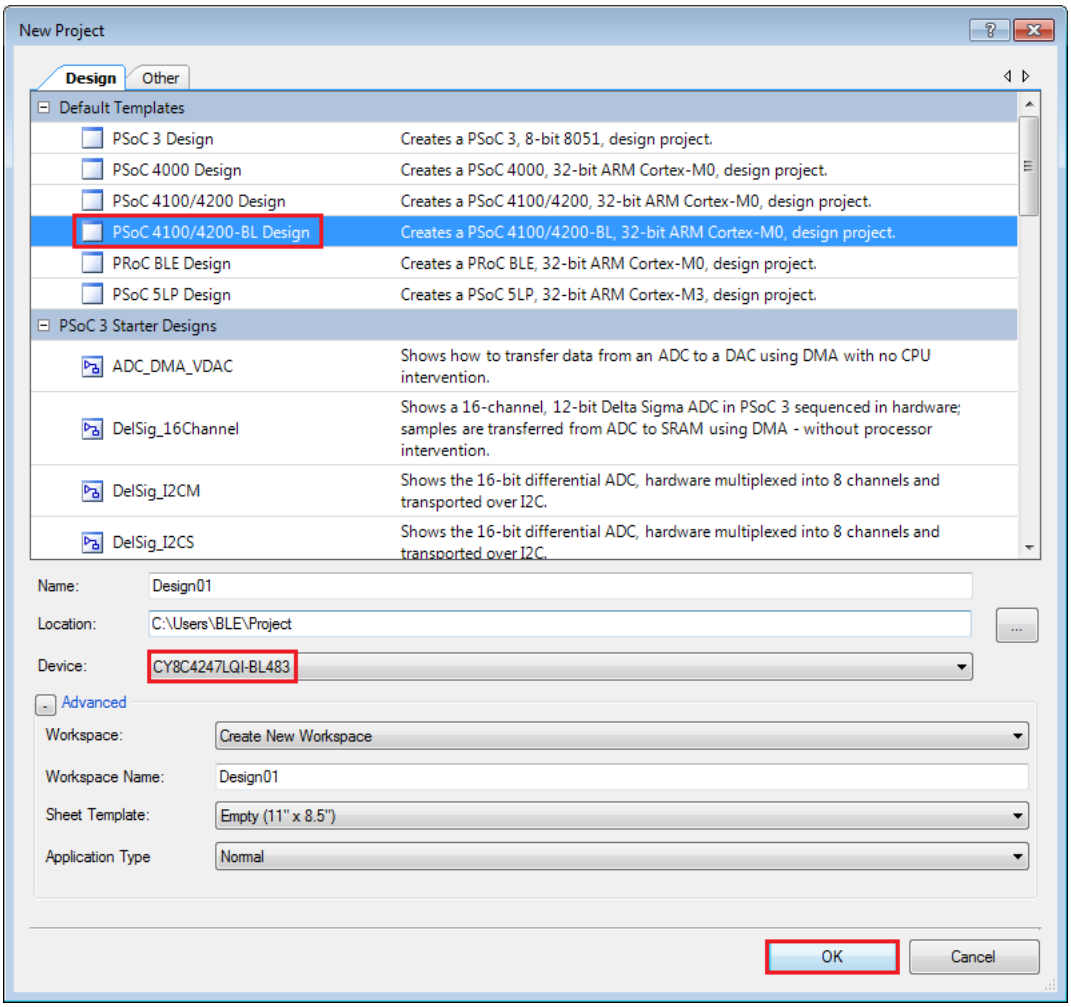


6.2 Using PSoC 5LP as USB-I²C Bridge

The PSoC 5LP serves as a USB-I²C bridge that can be used to communicate with the USB-I²C software running on the PC. The following steps describe how to use the USB-I²C bridge, which can communicate between the BCP and the PSoC 4 BLE/PRoC BLE.

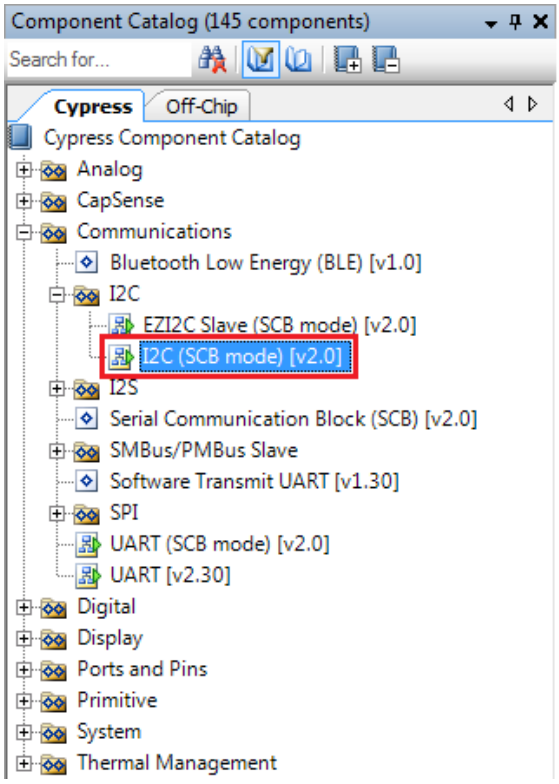
1. Create a new project targeting the PSoC 4 BLE/PRoC BLE device in PSoC Creator, as shown in Figure 6-16.

Figure 6-16. Create New Project in PSoC Creator



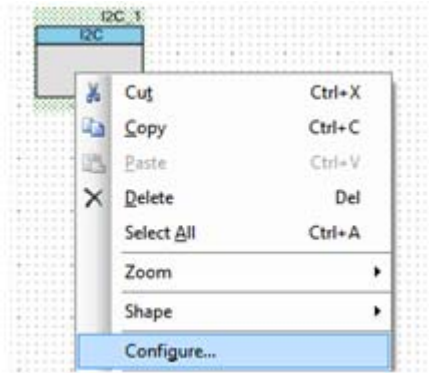
2. Drag and drop an I²C component (Figure 6-17) to the TopDesign.

Figure 6-17. I²C Component in Component Catalog



3. To configure the I²C component, double-click or right-click on the I²C component and select **Configure**, as shown in Figure 6-18.

Figure 6-18. Open I²C Configuration Window



4. Change the instance name to I2C. Configure the I²C component according to the settings in Figure 6-19 and Figure 6-20 and click **OK**.

Figure 6-19. Configuration Tab

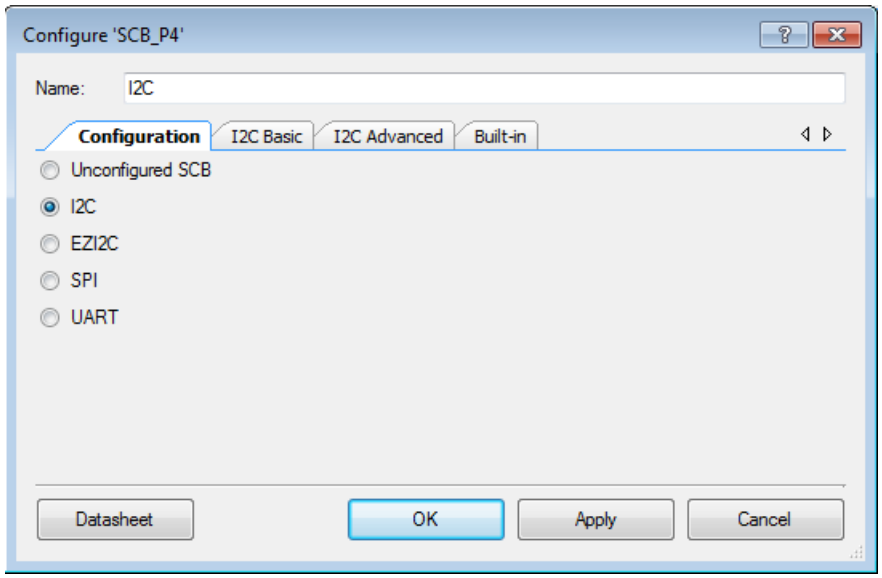
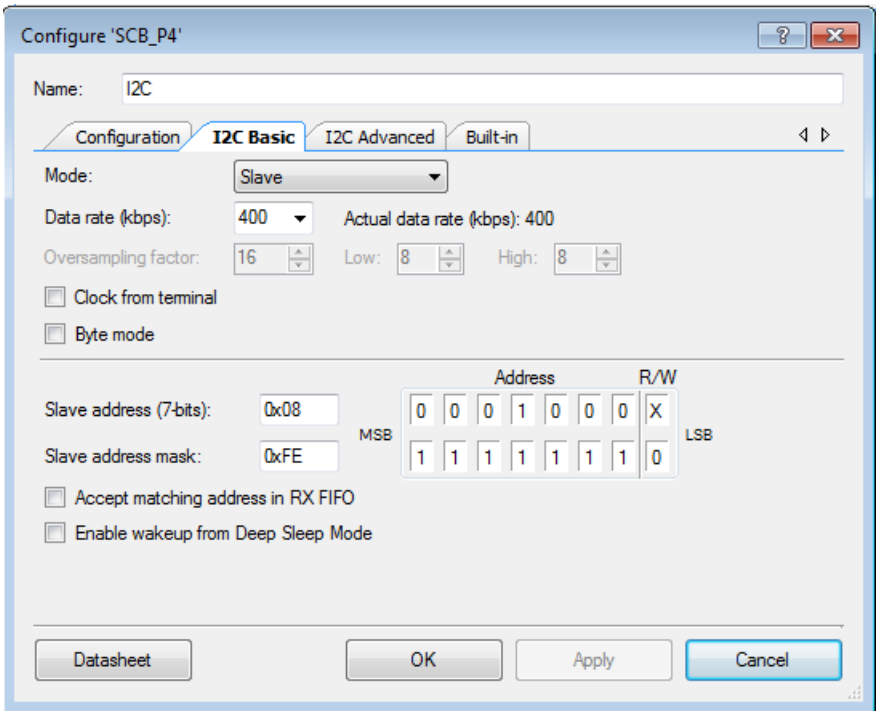
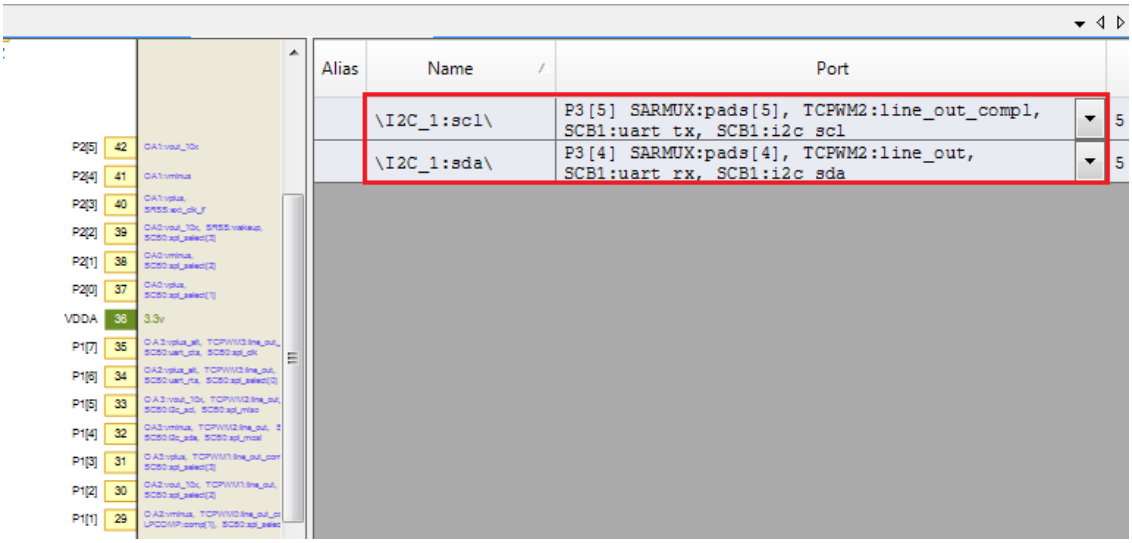


Figure 6-20. I²C Basic and Advanced Tabs



5. Select pin P3[5] for the I²C SCL and pin P3[4] for the I²C SDA in the **Pins** tab of *<Project_Name>.cydwr*, as shown in Figure 6-21.

Figure 6-21. Pin Selection_USBI2C



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

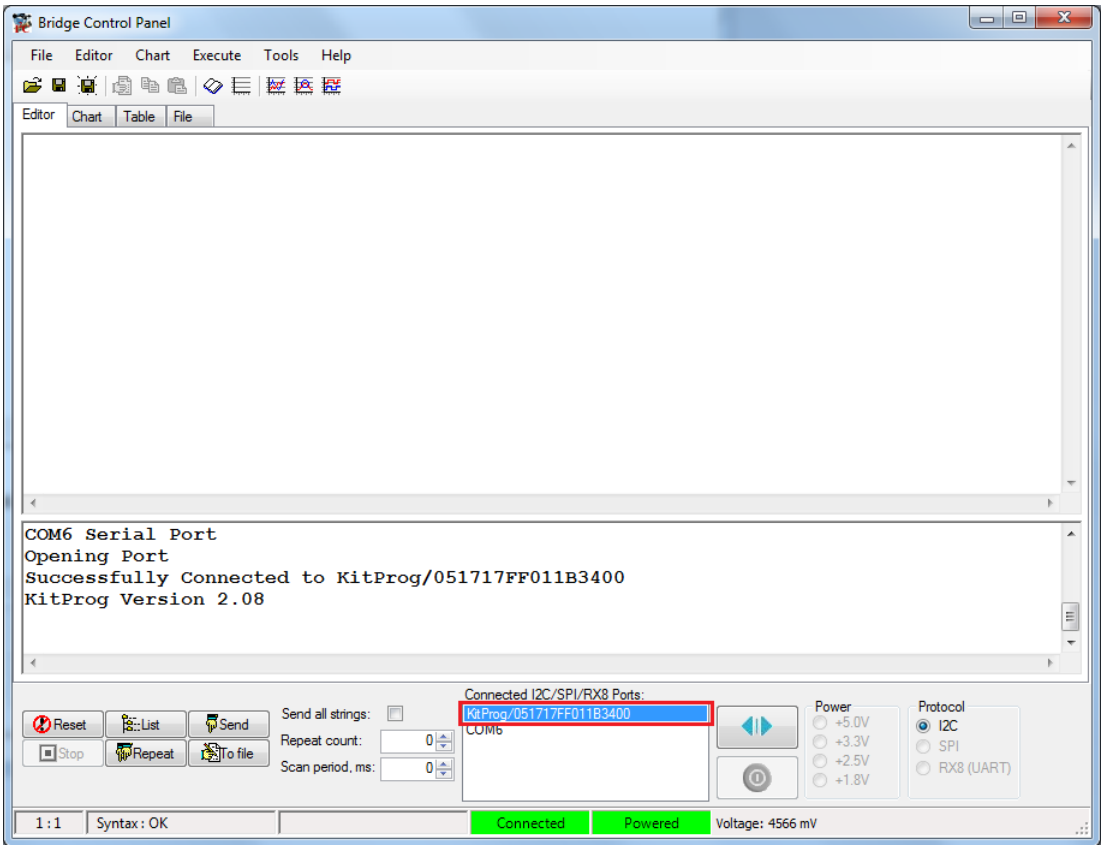
```
int main()
{

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();
 }
}

}
```

7. Build the project by choosing **Build > Build Project** or **[Shift]+[F6]**. After the project is built without errors and warnings, program (**[Ctrl]+[F5]**) this code onto the PSoC 4 BLE/PRoC BLE through the PSoC 5LP programmer or MiniProg3.

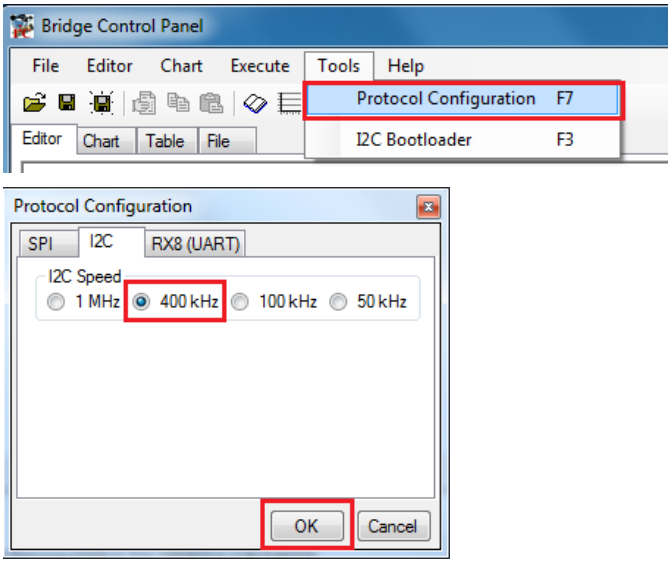
8. Open the BCP from **Start > All Programs > Cypress > Bridge Control Panel <version number>**.
9. Connect to **KitProg/** under **Connected I2C/SPI/RX8 Ports**, as shown in Figure 6-22.

Figure 6-22. Connecting to KitProg/ in BCP



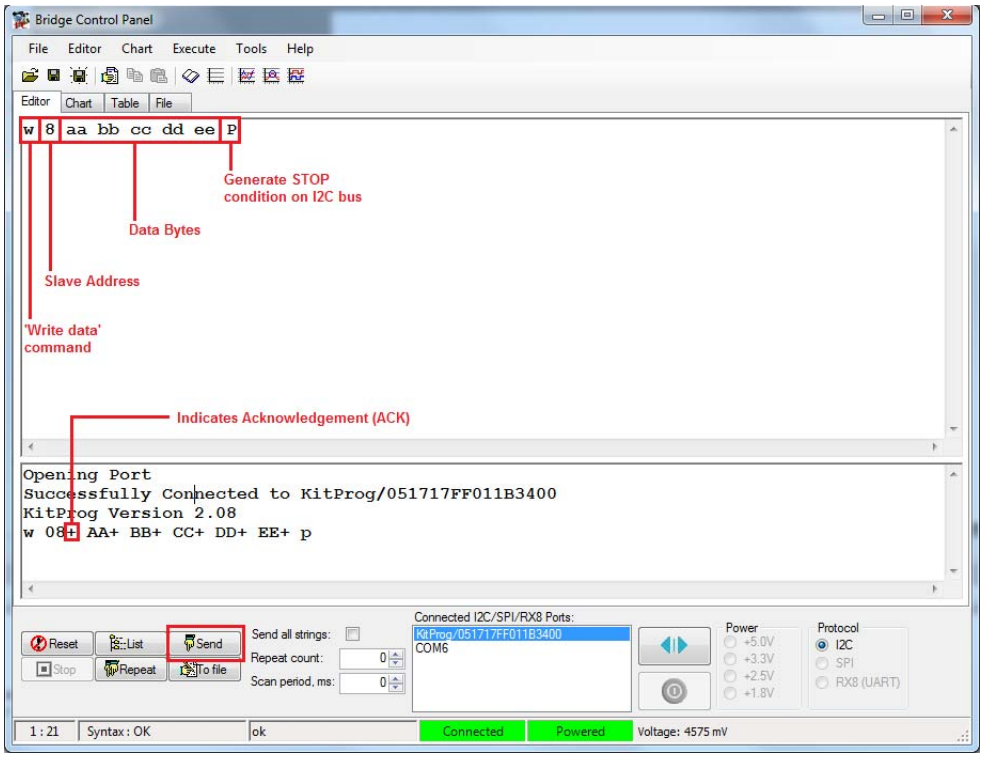
10. Open **Protocol Configuration** from the **Tools** menu and select the appropriate **I2C Speed**, as shown in Figure 6-23. Make sure the I²C speed is the same as the one configured in the I²C component. Click **OK** to close the window.

Figure 6-23. Opening Protocol Configuration Window in BCP



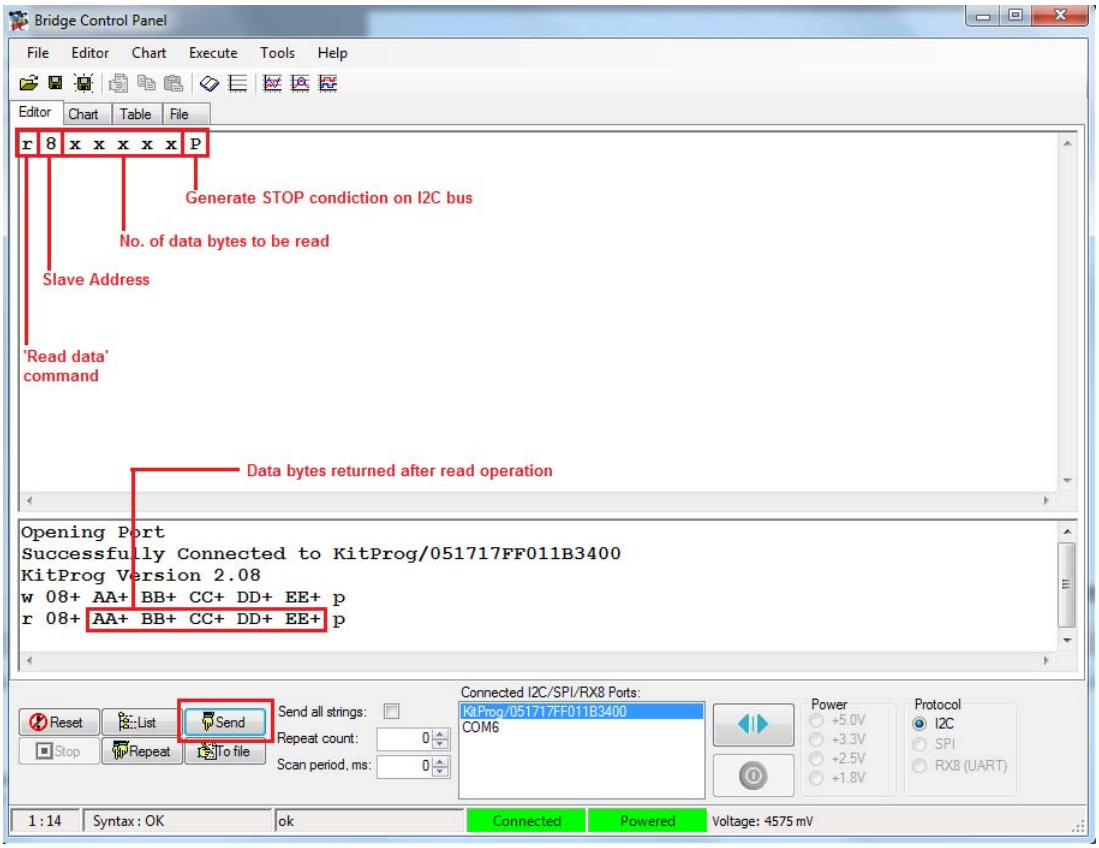
11. From the BCP, transfer five bytes of data to the I²C device with slave address 0x08. Type the command shown in Figure 6-24 and press **[Enter]** or click the **Send** button in the BCP. The log shows whether the transaction was successful. A '+' indication after each byte indicates that the transaction was successful and a '–' indicates that the transaction was a failure.

Figure 6-24. Entering Commands in BCP



12. 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, as shown in Figure 6-25.

Figure 6-25. Read Data Bytes from BCP



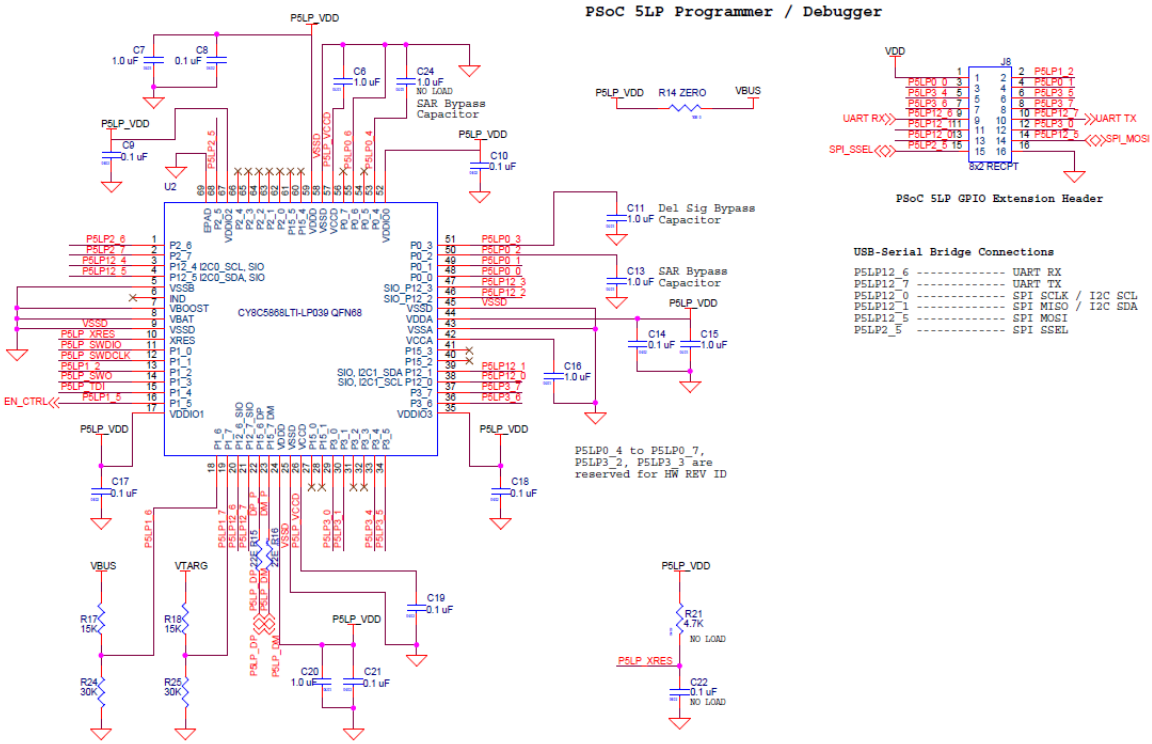
Note: Refer to **Help Contents** under **Help** in BCP or press **[F1]** for details of I²C commands.

6.3 Developing Applications for PSoC 5LP

The BLE Pioneer Kit has an onboard PSoC 5LP whose primary function is that of a programmer and a bridge. You 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 Figure 6-26. J8 is the I/O connector. The USB (J13) is connected and used as the PC interface. However, you can still use this USB connection to create customized USB designs.

Figure 6-26. PSoC 5LP Connections on BLE Pioneer Kit



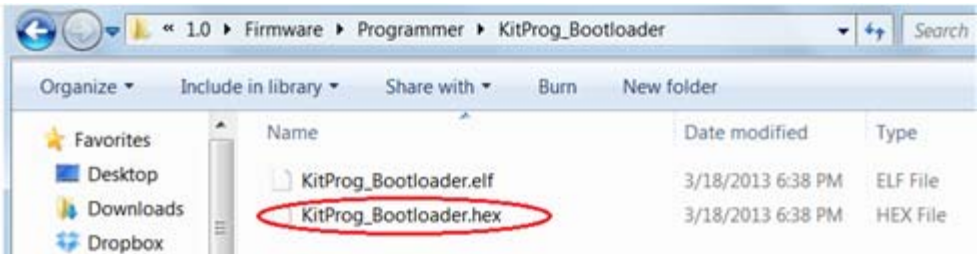
The programming header (J7) is meant for standalone programming. This header needs to be populated. See the 'No Load Components' section in Bill of Materials (BOM) on page 183.

6.3.1 Building a Bootloadable Project for 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 hex files are included in the following kit installer directory:
`<Install_Directory>\CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\KitProg_Bootloader`

Figure 6-27. KitProg Bootloader Hex File Location



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

1. In PSoC Creator, choose **New > Project > PSoC 5LP,** click the expand button adjacent to **Advanced,** select **Launch Device Selector** to bring up the **Select Device Window** and select the **Device** as **CY8C5868LTI-LP039,** as shown in Figure 6-28. Select the **Application Type** as **Bootloadable** from the drop-down list and click **OK**.

Figure 6-28. Create New Project in PSoC Creator_PSoC 5LP

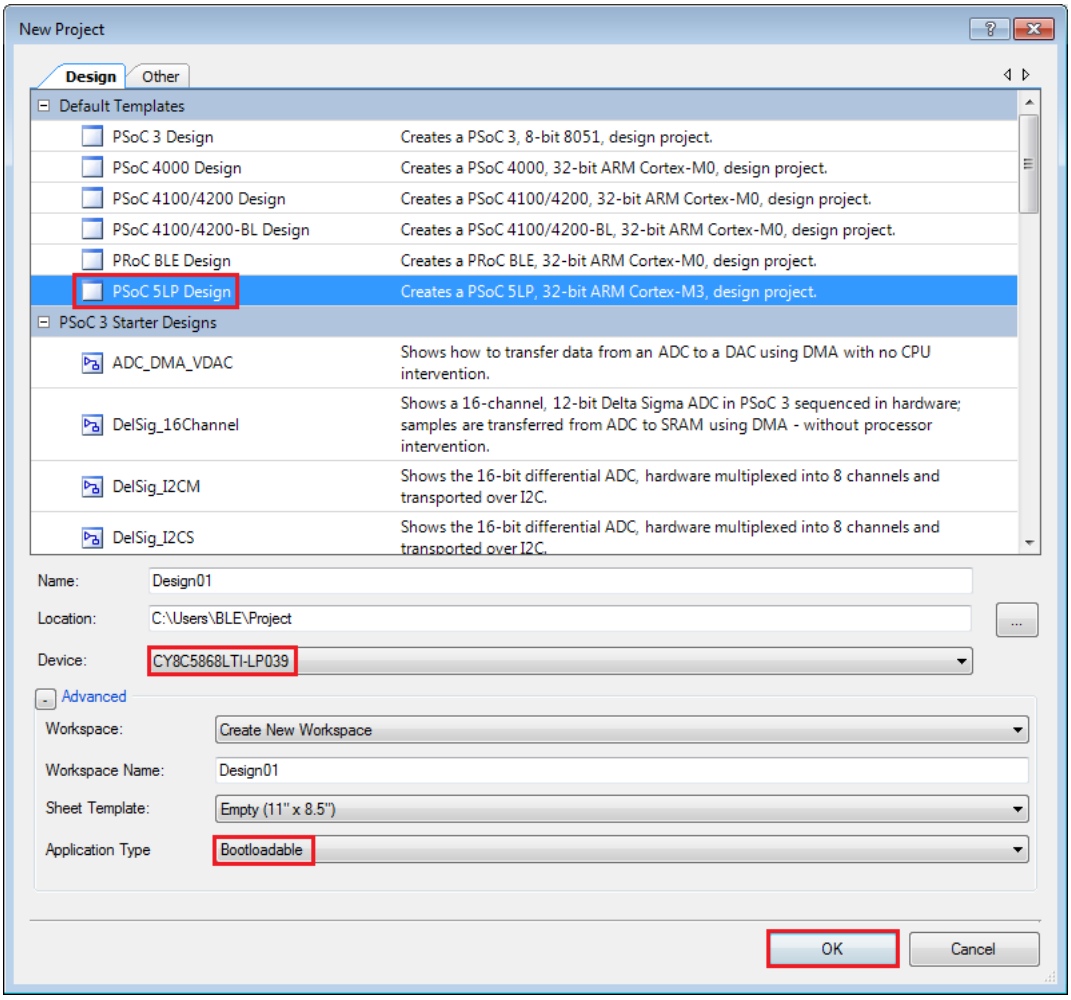
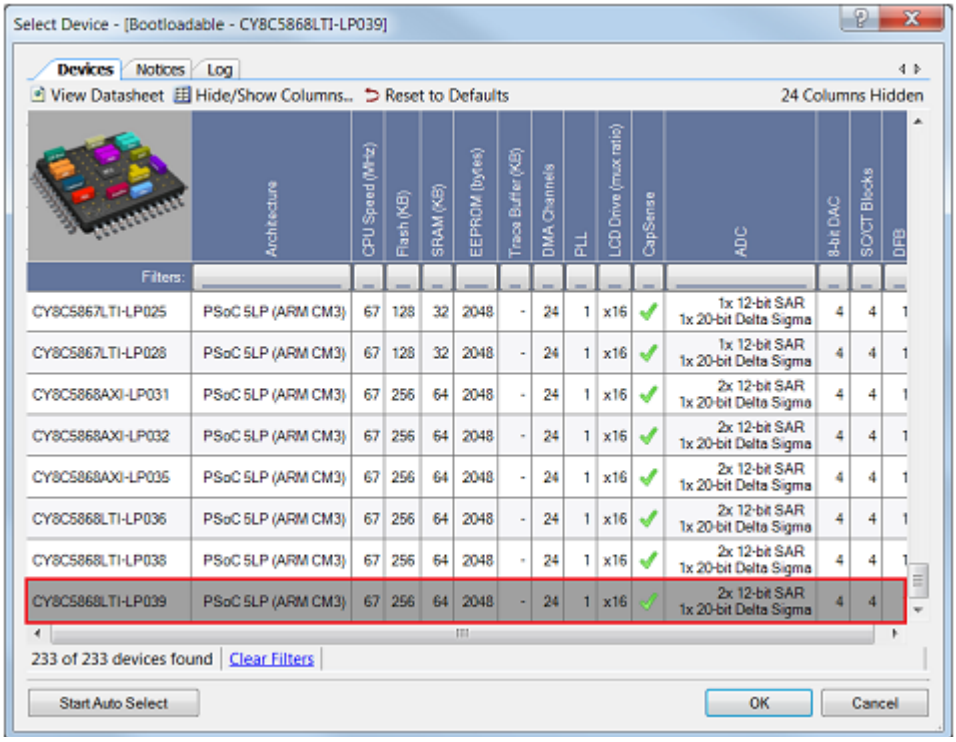
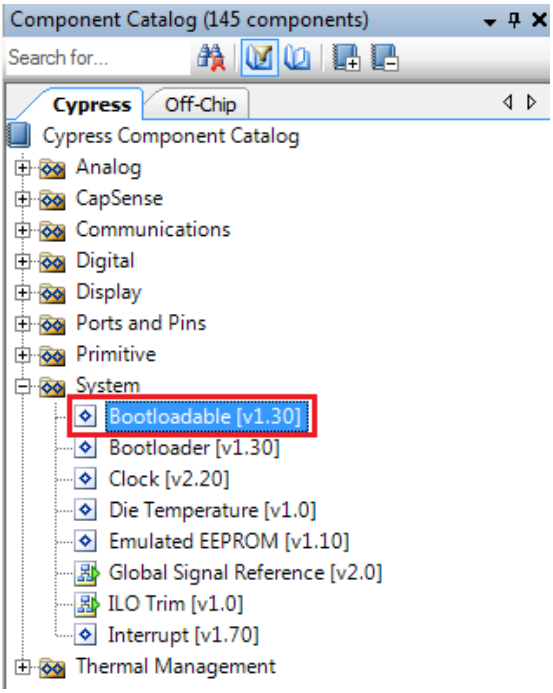


Figure 6-29. Select Device in PSoC Creator



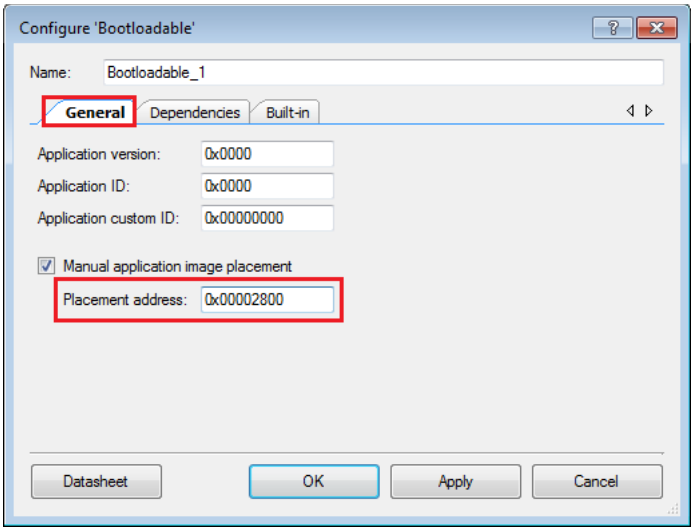
2. Navigate to the Schematic view and drag and drop a Bootloadable component (Figure 6-30) on the TopDesign.

Figure 6-30. Bootloadable Component in Component Catalog



To configure the Bootloadable, double-click or right-click on the Bootloadable component and select **Configure**. In **General** tab setting, enable the checkbox for “Manual application image placement” and set the value of 0x00002800 in Placement address.

Figure 6-31. Configuration Window of Bootloadable Component in “General” tab setting



Set the dependency of the Bootloadable component by selecting the **Dependencies** tab in the configuration window and clicking the **Browse** button, as shown in Figure 6-32. Select the *KitProg_Bootloader.hex* (Figure 6-33) and *KitProg_Bootloader.elf* files (Figure 6-34); click **Open**.

Figure 6-32. Configuration Window of Bootloadable Component in “Dependencies” tab setting

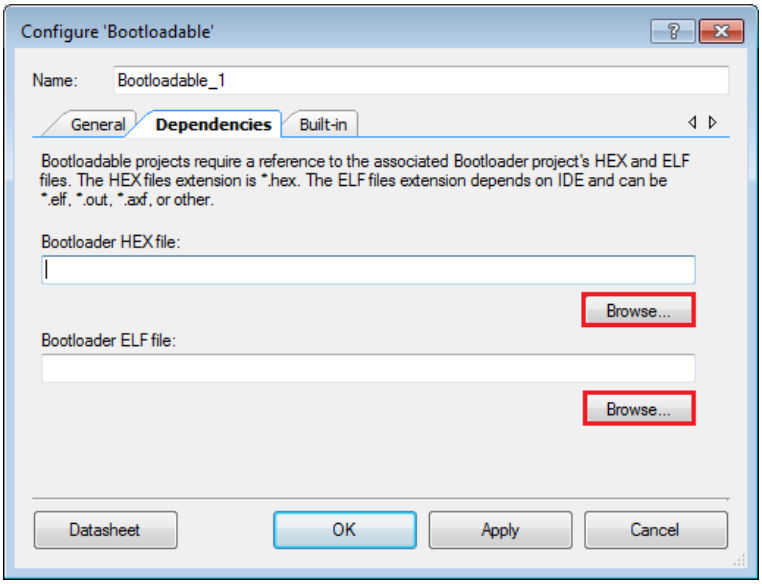


Figure 6-33. Select KitProg Bootloader Hex File

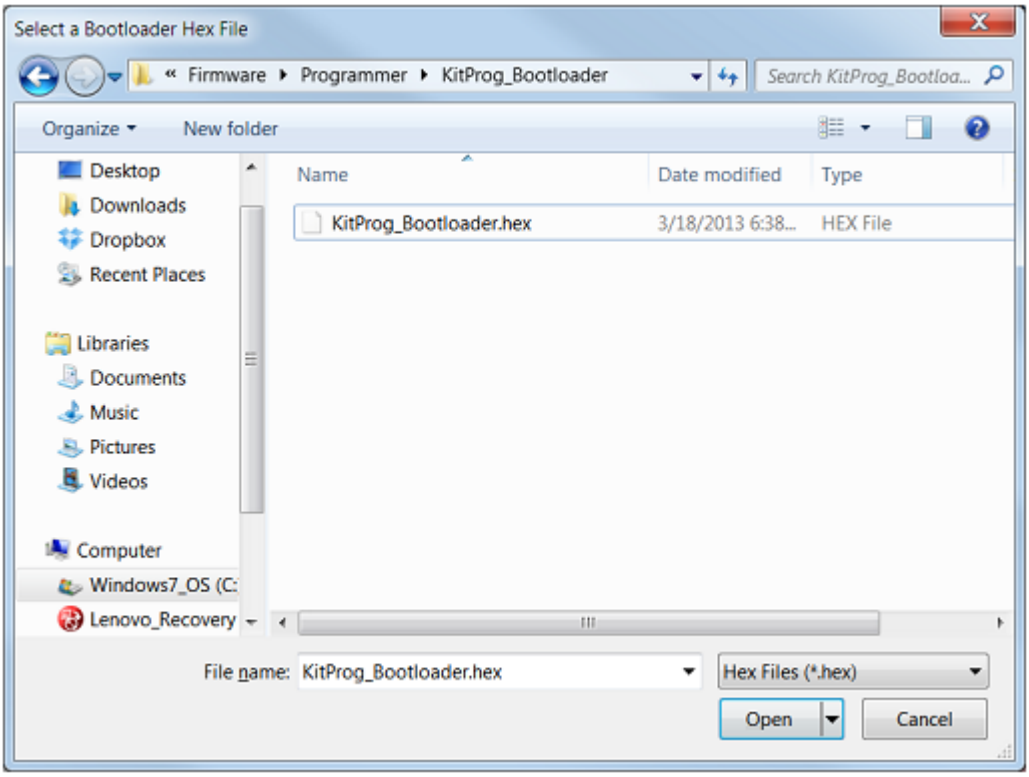
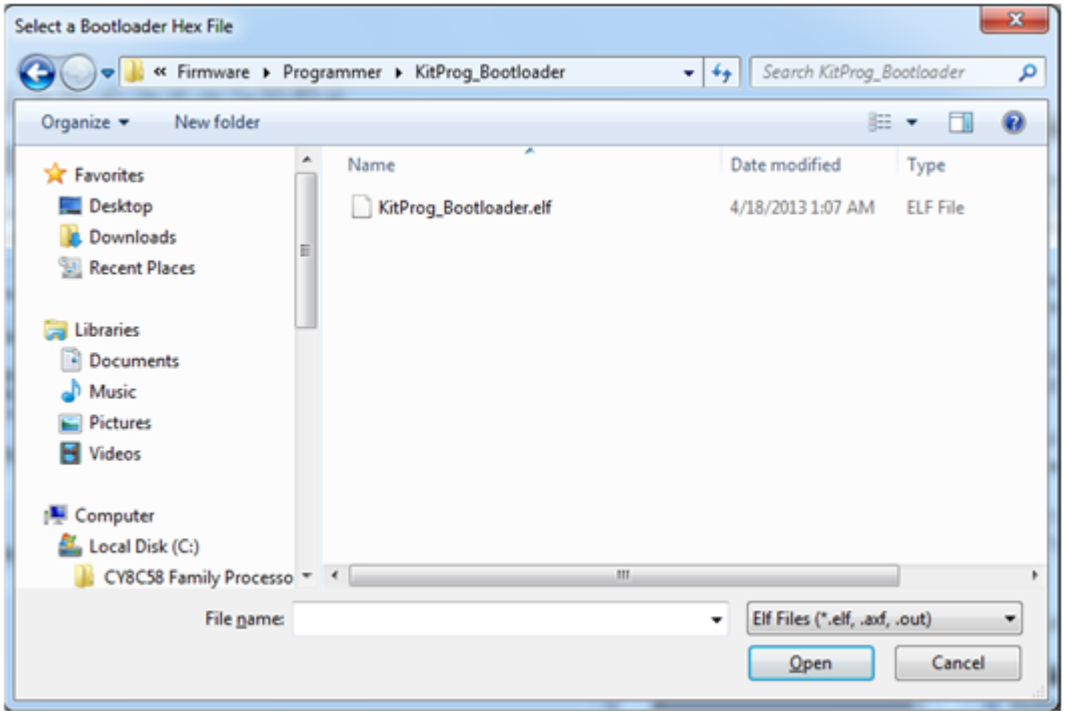


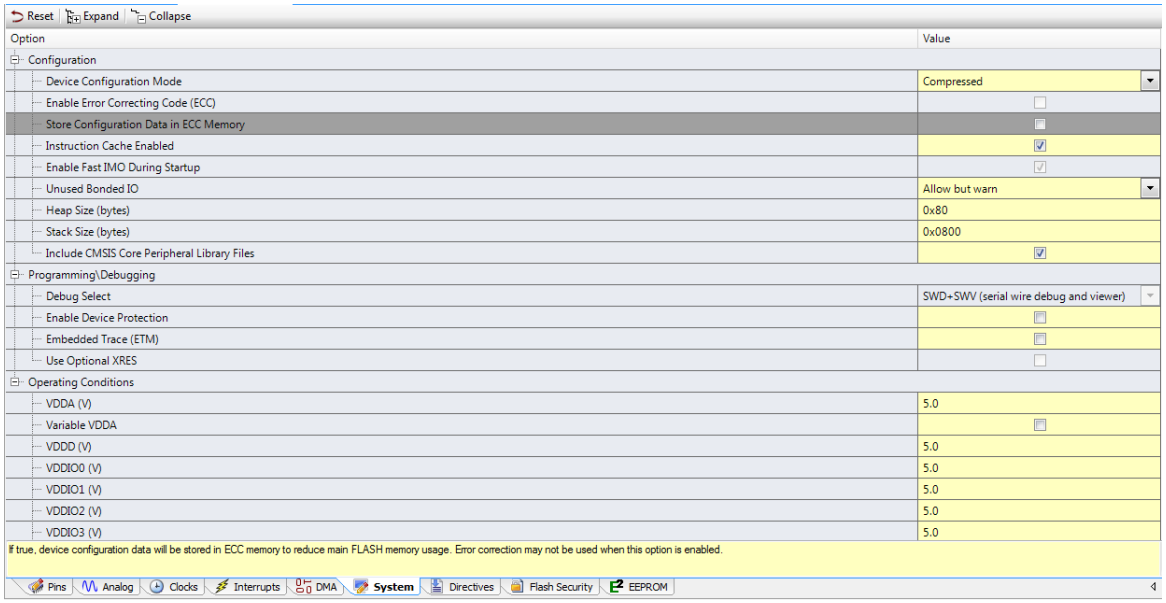
Figure 6-34. Select KitProg Bootloader Elf File



3. Develop your custom project.

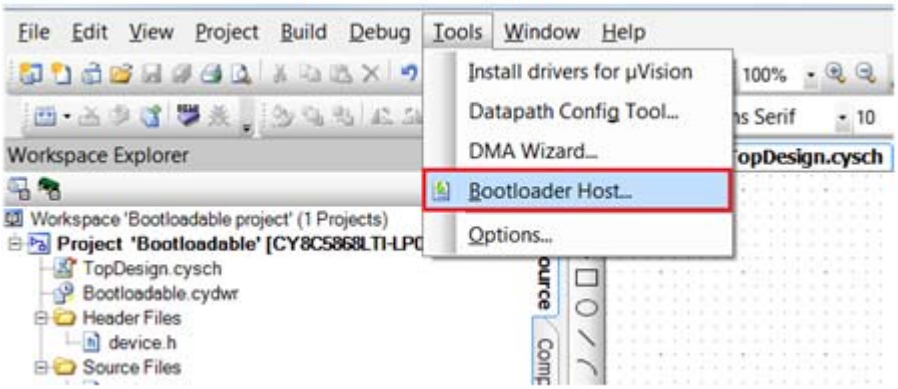
4. Make sure that the NVL setting of the Bootloadable project and the KitProg_Bootloader project is the same. Figure 6-35 shows the *KitProg_Bootloader.cydwr* system settings.

Figure 6-35. KitProg Bootloader System Settings



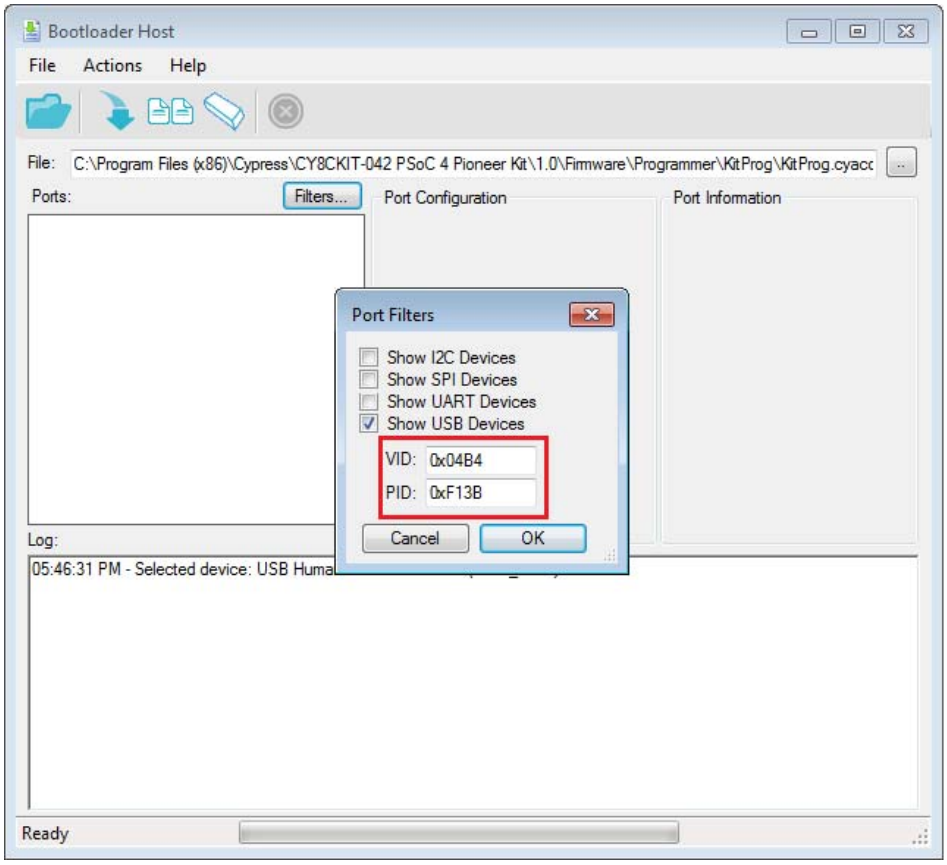
5. Build the project in PSoC Creator by choosing **Build > Build Project** or **[Shift]+[F6]**.
6. To download the project onto the PSoC 5LP device, open the Bootloader Host Tool, which is available in PSoC Creator. Choose **Tools > Bootloader Host**, as shown in Figure 6-36.

Figure 6-36. Open Bootloader Host Tool in PSoC Creator



7. In the Bootloader Host tool, click **Filters** and add a filter to identify the USB device. Ensure that the check box for **Show USB Devices** is enabled. Set VID as **0x04B4**, PID as **0xF13B**, and click **OK**, as shown in Figure 6-37.

Figure 6-37. Port Filters Tab in Bootloader Host Tool



8. In the Bootloader Host tool, click the **Open File** button (Figure 6-38) to browse to the location of the bootloadable file (*.cyacd), as shown in Figure 6-38.

Figure 6-38. Open Bootloadable File in Bootloader Host Tool

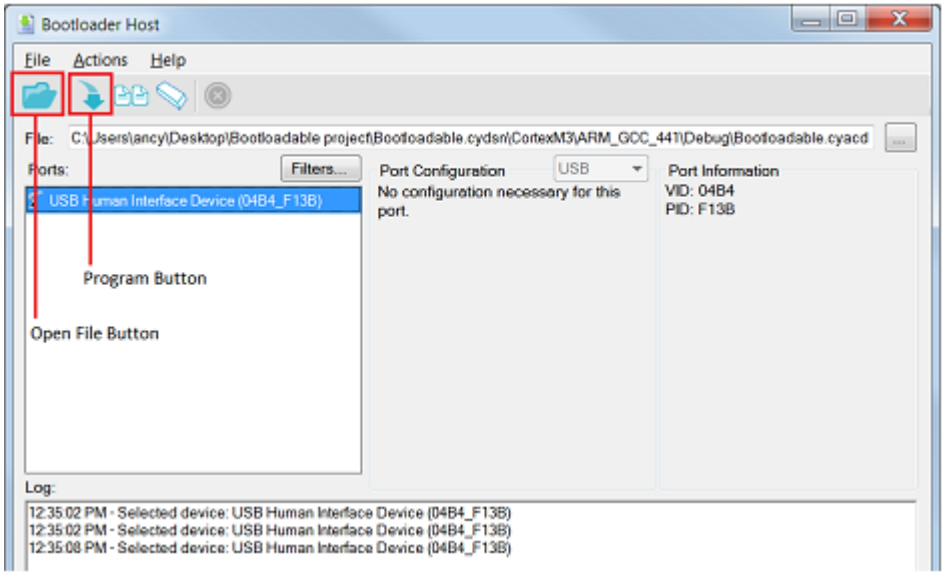
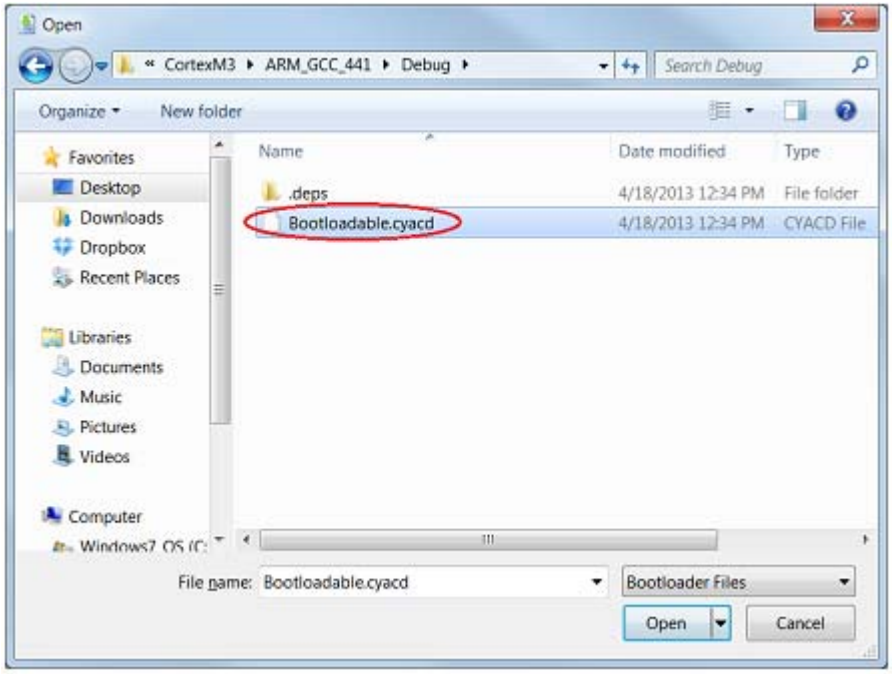


Figure 6-39. Select Bootloadable .cyacd File in Bootloader Host



9. 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. Click the **Program** button (Figure 6-38) in the Bootloader Host tool to program the device.

10. If bootload is successful, the log of the tool displays “Programming Finished Successfully”; otherwise, it displays “Failed” and a reason for the failure.

Notes:

- The PSoC 5LP pins are brought to the PSoC 5LP GPIO header (J8). These pins are selected to support high-performance analog and digital projects. See PSoC 5LP GPIO Header (J8) on page 93 for pin information.
- Take care when allocating the PSoC 5LP pins for custom applications. For example, P2[0]–P2[4] are dedicated for programming the PSoC 4 BLE/PRoC BLE. See Schematics on page 167 before allocating the pins.
- When a custom project is programmed onto the PSoC 5LP, the initial capability of the PSoC 5LP to act as a programmer, USB-UART bridge, or USB-I^2C bridge in not available.
- The status LED does not function unless used by the custom project.

For additional information on bootloaders, refer to Cypress application note, AN73503 - *USB HID Bootloader for PSoC 3 and PSoC 5LP*.

6.3.2 Building a Normal Project for PSoC 5LP

A normal project is a completely new project created for the PSoC 5LP device on the CY8CKIT-042. Here the entire flash of the PSoC 5LP is programmed, overwriting all bootloader and programming code. To recover the programmer, reprogram the PSoC 5LP device with the factory-set *KitProg.hex* file, which is shipped with the kit installer.

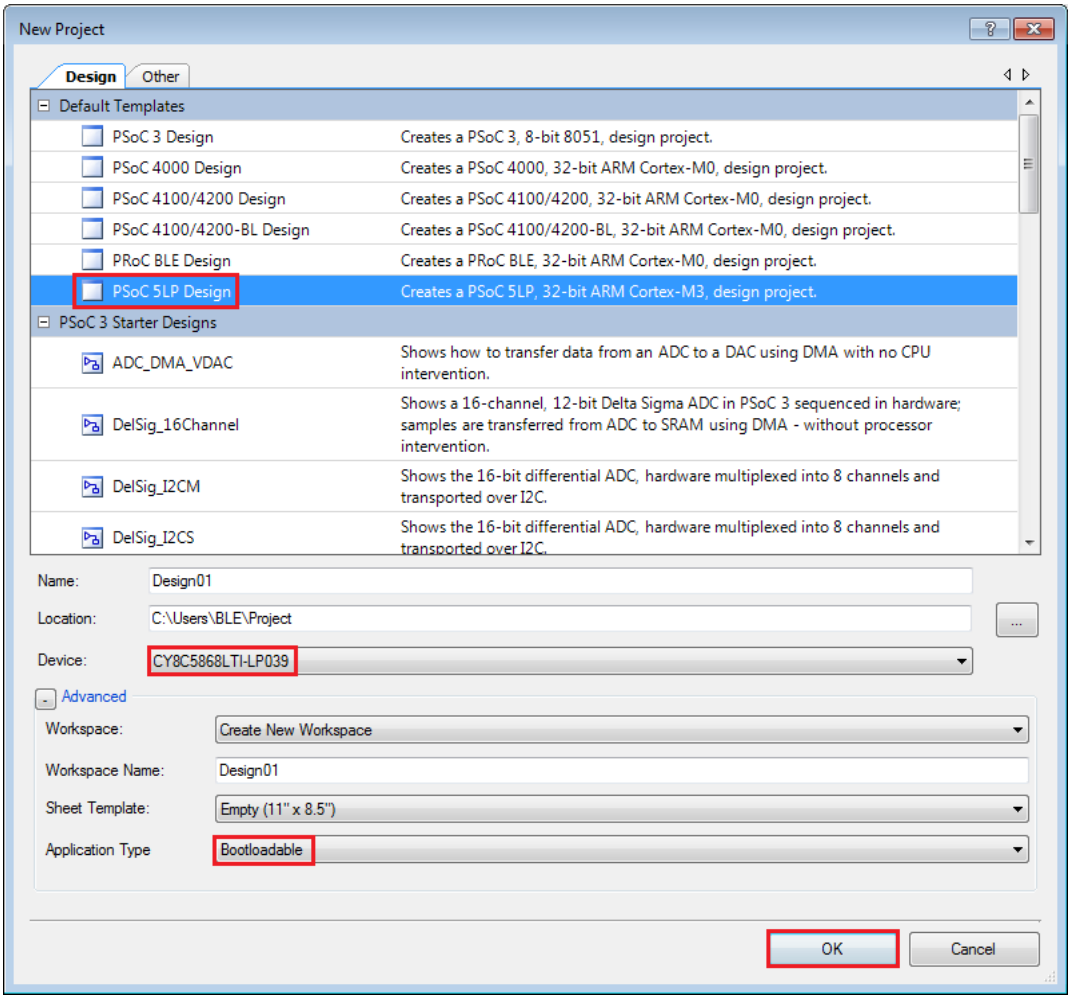
The *KitProg.hex* file is available at the following location:
`<Install_Directory>\CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\KitProg`

This advanced functionality requires a MiniProg3 programmer, which is not included with this kit. The MiniProg3 can be purchased from www.cypress.com/go/CY8CKIT-002.

To build a normal project for the PSoC 5LP, follow these steps:

1. In PSoC Creator, choose **New > Project > PSoC 5LP**, click the expand button adjacent to **Advanced**, select **Device** as **CY8C5868LTI-LP039**, and select **Application Type** as **Normal** from the drop-down list, as shown in Figure 6-40.

Figure 6-40. Create New Project in PSoC Creator_PSoC 5LP



2. Develop your custom project.
3. Build the project in PSoC Creator by choosing **Build > Build Project** or pressing **[Shift]+[F6]**.

4. Connect the 10-pin connector of MiniProg3 to the onboard 10-pin SWD debug and programming header J7 (which needs to be populated).
5. To program the PSoC 5LP with PSoC Creator, choose **Debug > Program** or press **[Ctrl]+[F5]**. If the Programming window appears and shows MiniProg3 and the selected device in the project under it (CY8C5868LTI-LP039); click on the device and click **Connect** to program.

Notes:

- The 10-pin SWD debug and programming header (J7) is not populated. See the 'No Load Components' section of A.3 Bill of Materials (BOM) for details.
- The PSoC 5LP pins are brought to the PSoC 5LP GPIO header (J8). These pins are selected to support high-performance analog and digital projects. See PSoC 5LP GPIO Header (J8) on page 93 for pin information.
- Take care when allocating the PSoC 5LP pins for custom applications. For example, P2[0]–P2[4] are dedicated for programming the PSoC 4. Refer to A.1 Schematics before allocating the pins.
- 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, or USB-I^{2}C bridge is not available.
- The status LED does not function unless it is used by the custom project.

6.4 PSoC 5LP Factory Program Restore Instructions

The BLE Pioneer Kit features a PSoC 5LP device that comes factory-programmed as the onboard programmer and debugger for the PSoC 4 BLE/PRoC BLE device.

In addition to creating applications for the BLE device, you can also create custom applications for the PSoC 5LP device on this kit. For details, see section Developing Applications for PSoC 5LP on page 132. Reprogramming or bootloading the PSoC 5LP device with a new flash image will overwrite the factory program and forfeit the ability to use the PSoC 5LP device as a programmer/debugger for the BLE device. Follow the instructions to restore the factory program on the PSoC 5LP and enable the programmer/debugger functionality.

6.4.1 PSoC 5LP is Programmed with a Bootloadable Application

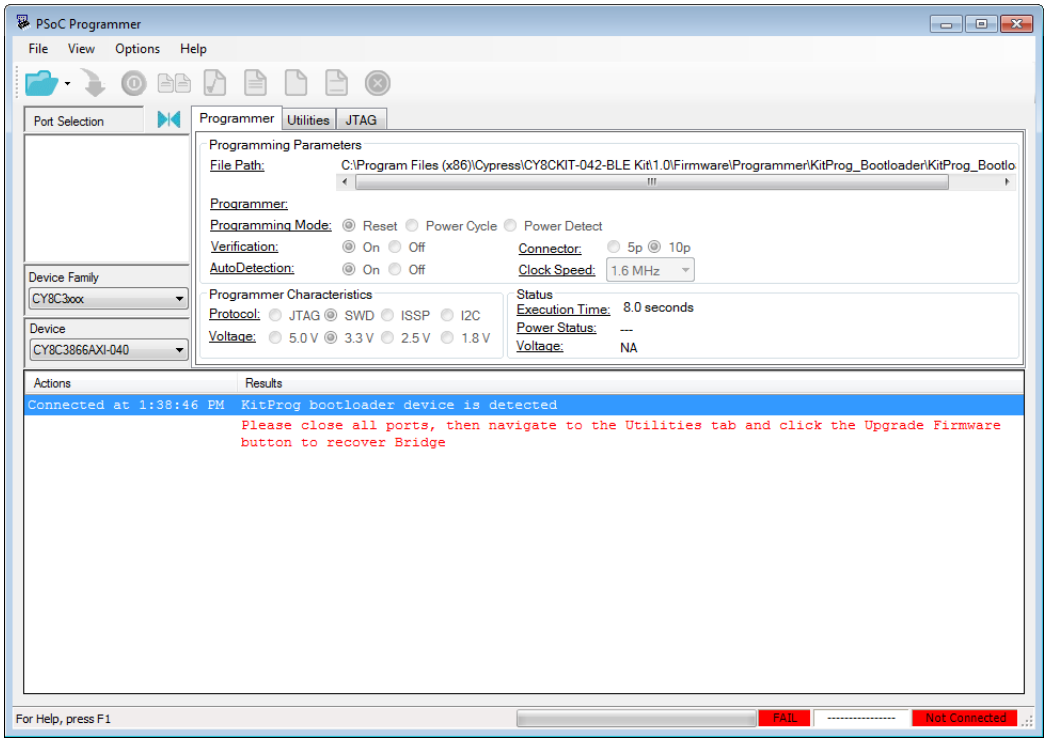
If the PSoC 5LP is programmed with a bootloadable application, restore the factory program by using one of the following two methods.

6.4.1.1 Restore PSoC 5LP Factory Program Using PSoC Programmer

1. Launch **PSoC Programmer 3.21.1** or later from **Start > Cypress > PSoC Programmer**.
2. Configure the BLE Pioneer Kit in service mode. To do this, while holding down the reset button (SW1 Reset), plug in the BLE Pioneer Kit to the computer using the included USB cable (USB A to mini-B). This puts the PSoC 5LP into service mode, which is indicated by the blinking green status LED.

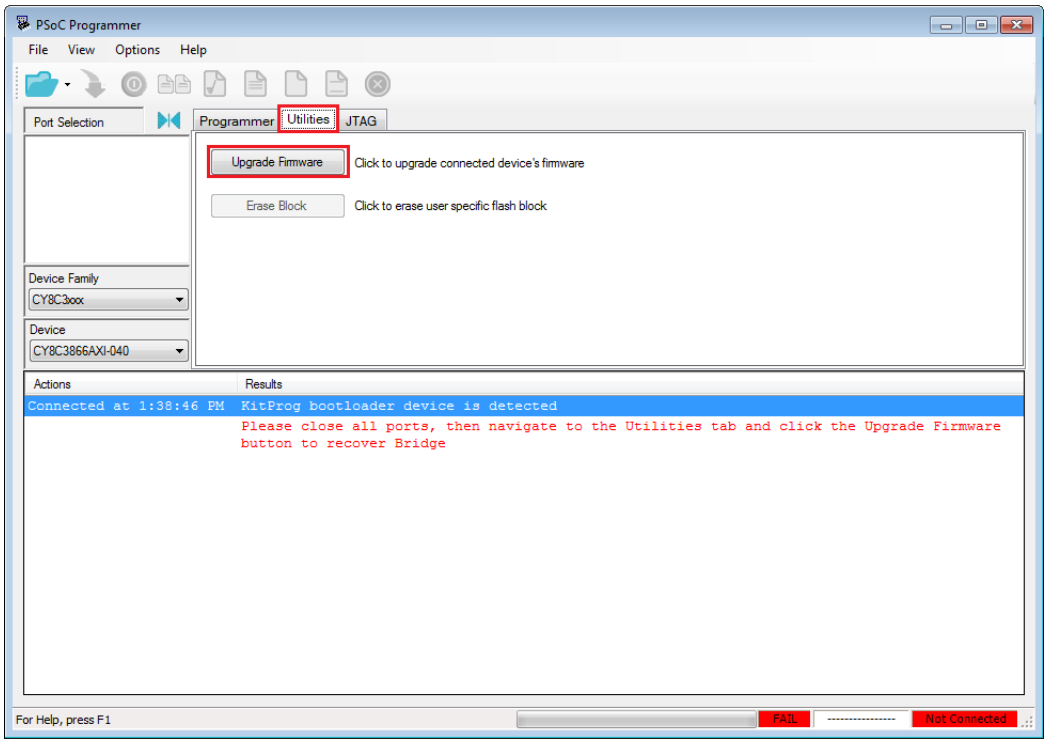
3. The following message appears in the PSoC Programmer **Results** window, as shown in Figure 6-41: “KitProg Bootloader device is detected”.

Figure 6-41. PSoC Programmer Results Window



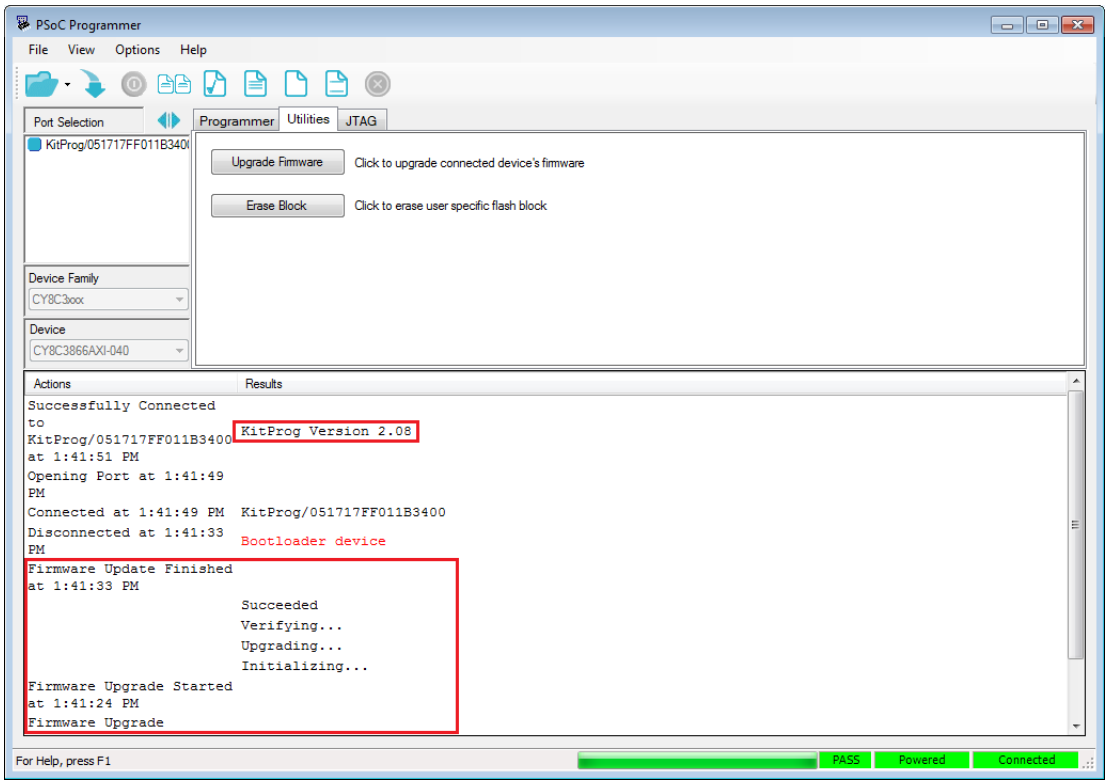
4. Switch to the **Utilities** tab in PSoC Programmer and press the **Upgrade Firmware** button, as shown in Figure 6-42. Unplug all other PSoC programmers (such as MiniProg3 and DVKProg) from the PC before pressing the **Upgrade Firmware** button.

Figure 6-42. Upgrade Firmware



5. After programming has completed, the following message appears, as shown in Figure 6-43: “Firmware Update Finished at <time>”.

Figure 6-43. Firmware Update Completed

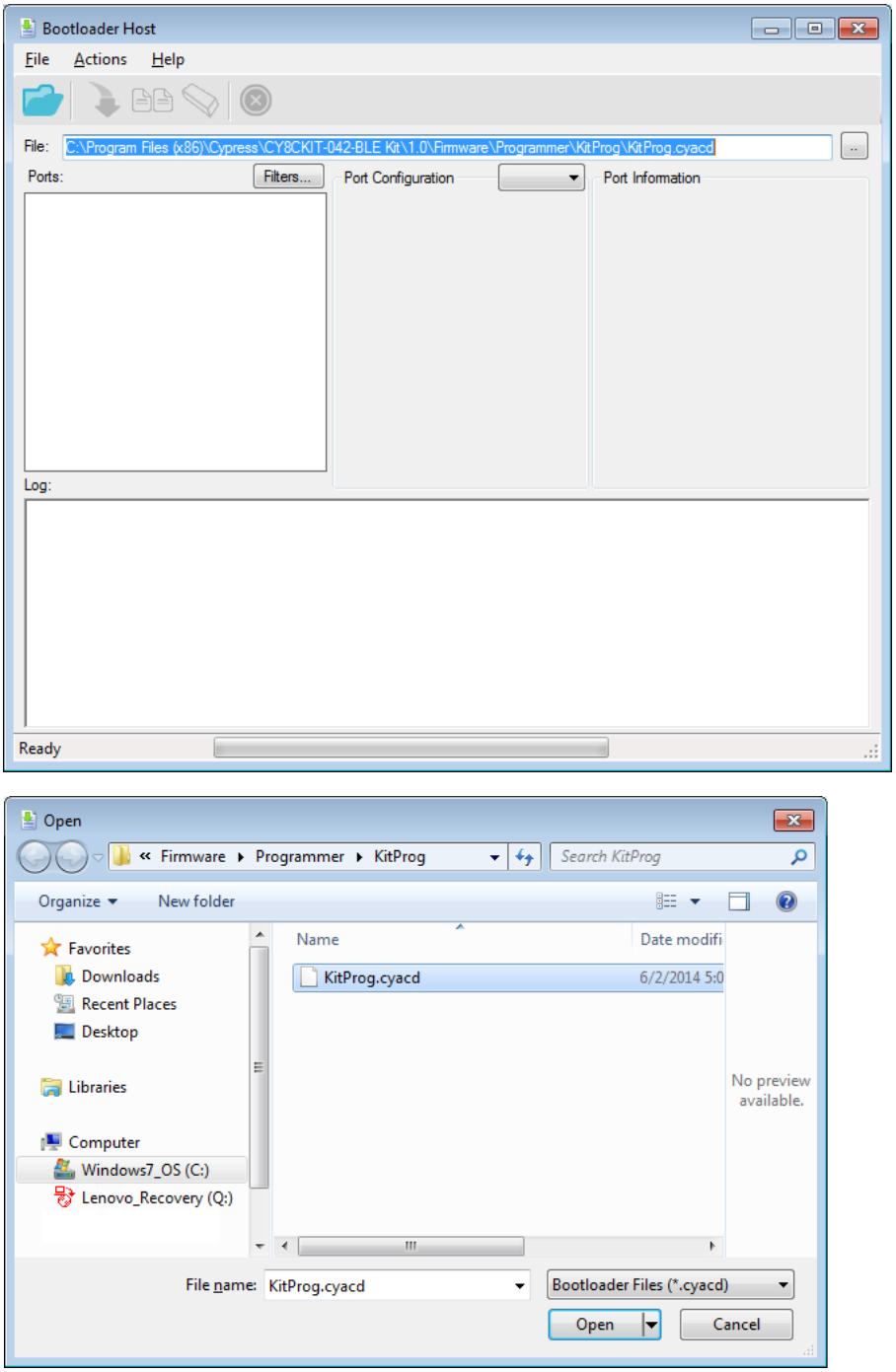


6. The factory program is now successfully restored on the PSoC 5LP. It can be used as the programmer/debugger for the PSoC 4 BLE or PRoC BLE device.

6.4.1.2 *Restore PSoC 5LP Factory Program Using USB Host Tool*

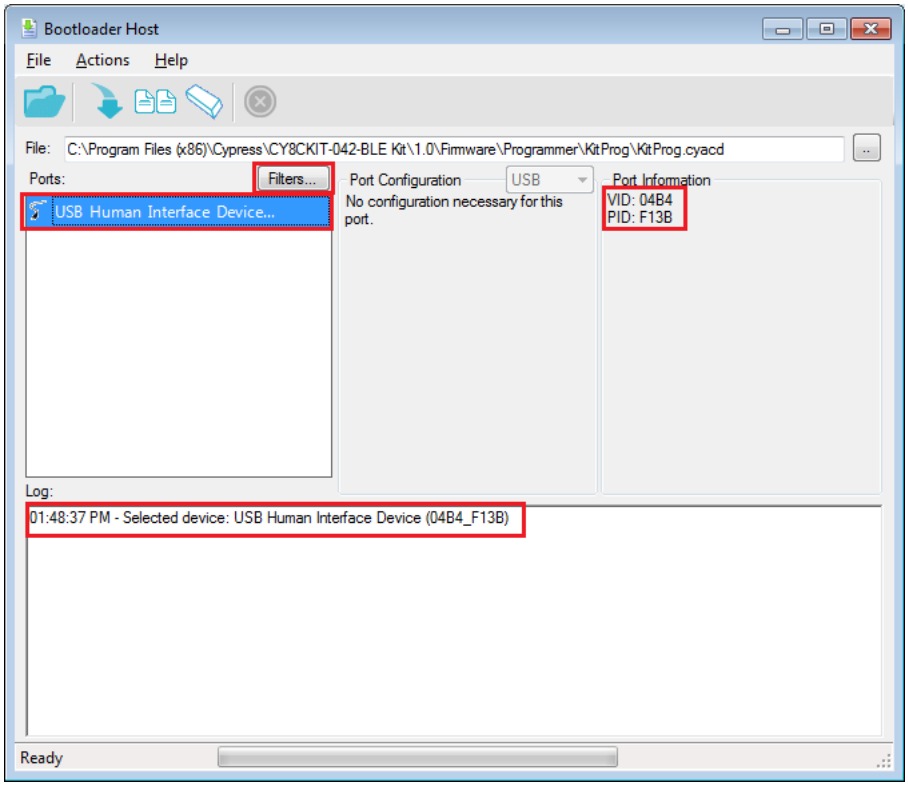
1. Launch the Bootloader Host tool from **Start > Cypress > PSoC Creator**.
2. Using the **File > Open** menu, load the *KitProg.cyacd* file, which is installed with the kit software, as shown in Figure 6-44. The default location for this file is: <Install_Directory>\ `CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\KitProg\KitProg.cyacd`

Figure 6-44. Load KitProg.cyacd File



3. Configure the Pioneer Kit in service mode. To do this, while holding down the reset button (SW1 Reset), plug in the BLE Pioneer Kit to the computer using the included USB cable (USB A to mini-B). 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**. The **USB Human Interface Device** port appears in the **Ports** list. Click that port to select it, as shown in Figure 6-45.

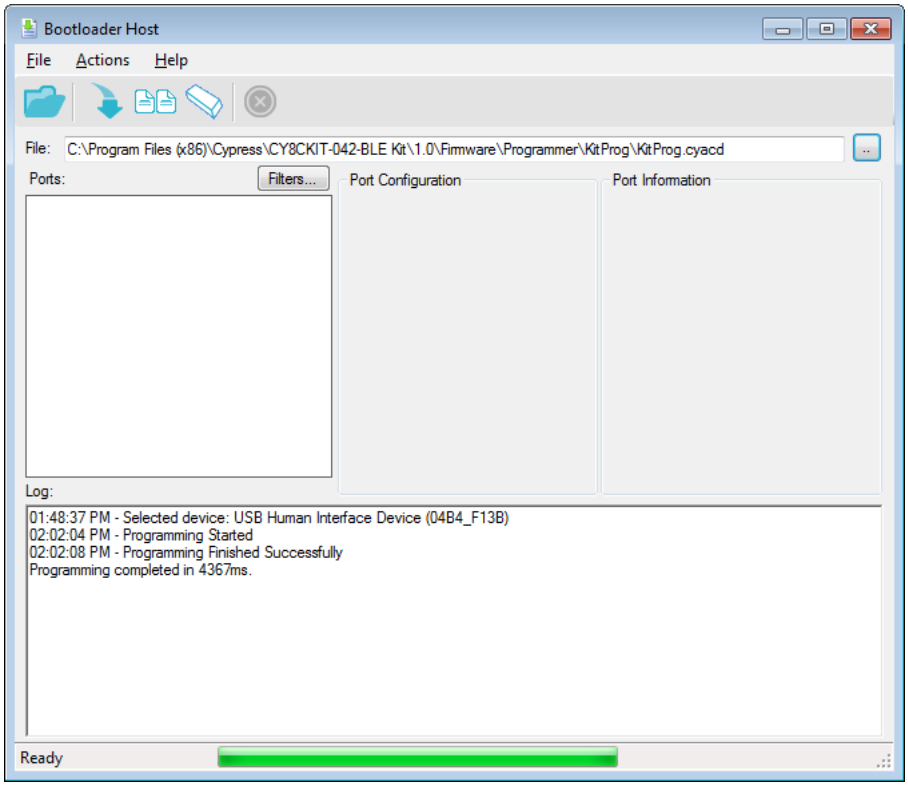
Figure 6-45. Select USB Human Interface Device



5. Click the **Program** button (or choose **Actions > Program**) to restore the factory-program by bootloading it onto the PSoC 5LP.

6. After programming has completed, the following message appears, as shown in Figure 6-46: “Programming Finished Successfully”.

Figure 6-46. Programming Finished Successfully



7. The factory program is now successfully restored on the PSoC 5LP. It can be used as the programmer/debugger for the PSoC 4 BLE/PRoC BLE device.

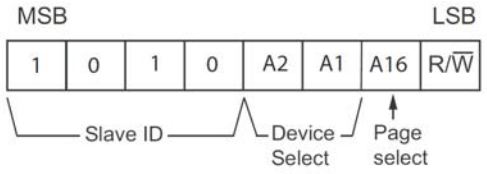
6.5 Using FM24V10 F-RAM

The BLE Pioneer board has an onboard ferroelectric RAM chip that can hold up to 1 Mb of data. The chip provides an I^2C communication interface for data access. It is hardwired to the I^2C lines (P3_4 and P3_5); the same lines are also routed to the PSoC 5LP I^2C lines. Because the F-RAM device is an I^2C slave, it can be accessed or shared among various I^2C masters on the same line. For more details on the F-RAM device, refer to the device datasheet.

6.5.1 Address Selection

The slave address of the F-RAM device consists of three parts, as shown in Figure 6-47: slave ID, device select, and page select. Slave ID is an F-RAM family-specific ID located in the datasheet of the particular F-RAM device. For the device used in BLE Pioneer board (FM24V10), the slave ID is 1010b. Device select bits are set using the two physical pins A2 and A1 in the device. The setting of these two pins on the BLE Pioneer board is controlled by resistors R32/R36 (A1) and R33/R37 (A2). Because the memory location in F-RAM is divided into two pages of 64 KB each, the page select bit is used to refer to one of the two pages in which the read or write operations will take place.

Figure 6-47. F-RAM I^{2}C Address Byte Structure



6.5.2 Write/Read Operation

The device's datasheet includes details on how to perform a write/read operation with the F-RAM. Figure 6-48 and Figure 6-49 provide a snapshot of the write/read packet structure as a quick reference.

Figure 6-48. F-RAM Single-Byte and Multiple-Byte Write Packet Structure

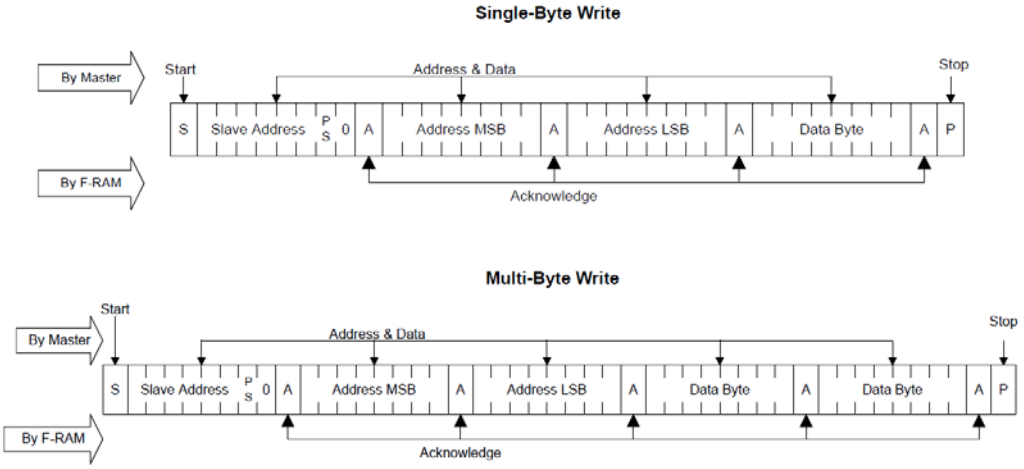
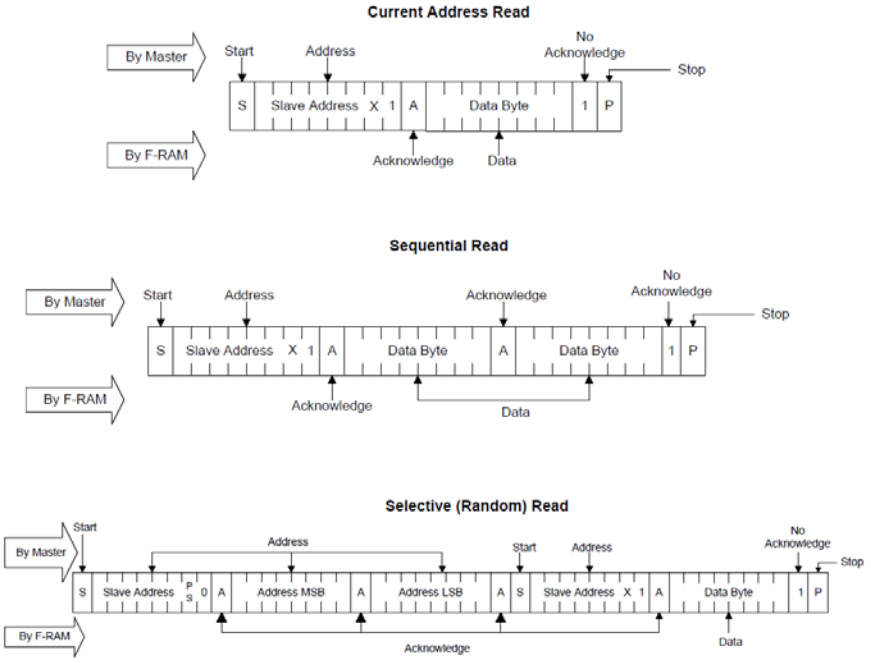


Figure 6-49. F-RAM Single-Byte and Multiple-Byte Read Packet Structure



As shown in the figures, all operations start with the slave address followed by the memory address. For write operations, the bus master sends each byte of data to the memory, and the memory generates an acknowledgement condition. For read operations, after receiving the complete slave address and memory address, the memory begins shifting data from the current address on the next clock.

6.6 CySmart iOS Application

The CySmart mobile application is a powerful tool that allows the mobile device (iOS) with BLE capability to connect to a BLE peripheral device and communicate with it. It supports various standard BLE profiles along with two custom profiles for CapSense and LED control. It also provides a common support for all profiles, standard or custom.

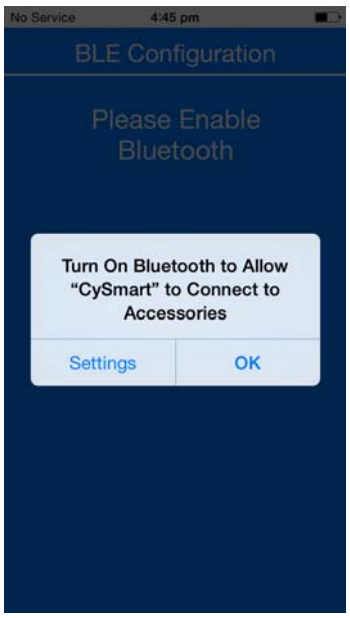
This app is free. You can download and install it for Apple iOS devices from the App Store. Make sure that the mobile device being used supports BLE.

To verify the example project using the CySmart mobile app, follow these steps.

1. Plug the BLE Pioneer Kit into the PC for power, using the J13 USB connector.
2. Program the kit with the desired BLE example project.
3. Open the app on the mobile device.

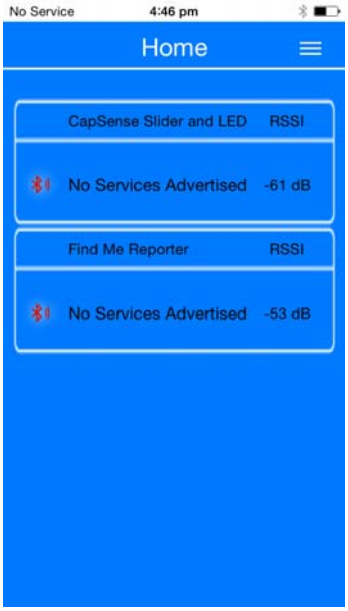
4. If Bluetooth is not enabled on the device, the app will ask to enable it, as shown in Figure 6-50.

Figure 6-50. Turn on Bluetooth on Device



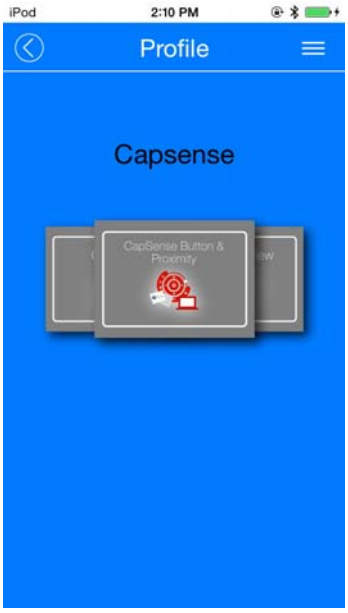
5. After Bluetooth is enabled, the app will automatically search for available BLE peripherals and list them, as shown in Figure 6-51. Select the BLE Pioneer Kit peripheral in the list. The name displayed in the list will be the same as that set in the BLE Component.

Figure 6-51. Peripheral Connections Page



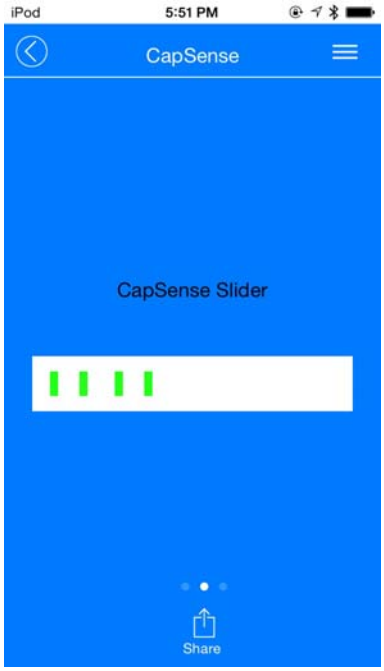
6. When connected, the app will list the supported profiles by the peripherals, as shown in Figure 6-52. Tap on the desired profile.

Figure 6-52. Profiles Page



7. Depending on the type of profile chosen, the app will display options for the profile. Figure 6-53 shows an example for the CapSense slider custom profile, where swiping a finger on the CapSense slider of the BLE Pioneer Kit is reflected in the app. See Pioneer Baseboard on page 83.

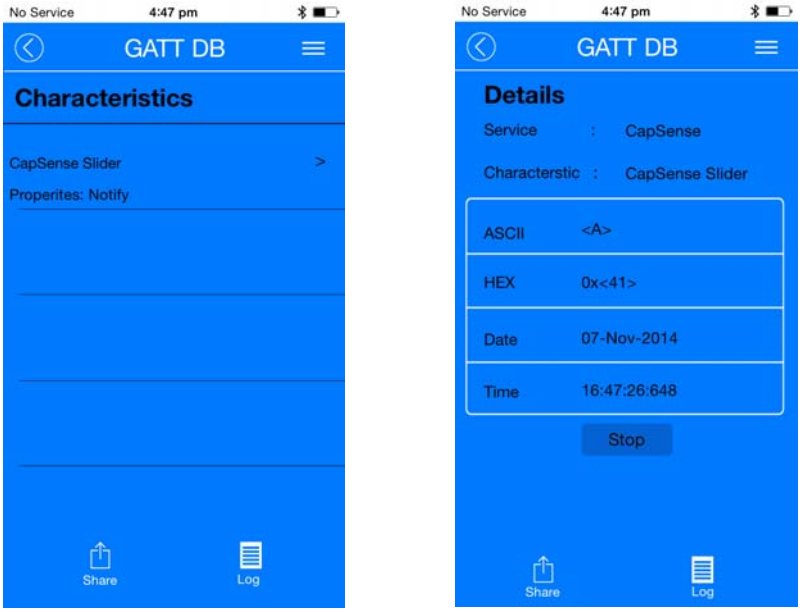
Figure 6-53. CapSense Slider GUI



8. To go to different service, go back to service page in the GUI.

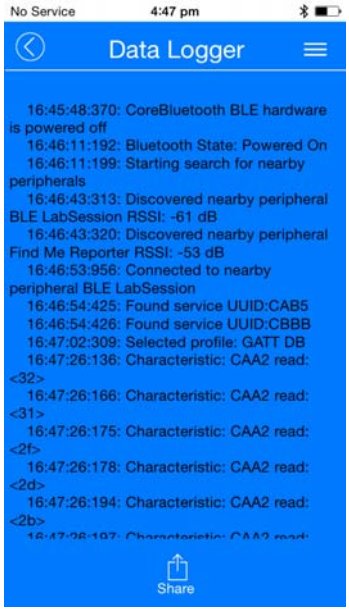
9. To connect to a new BLE peripheral, go back to home page and swipe the screen below to scan for devices.
10. To transfer data/notifications through any other profile that is not listed on the Profiles page after connecting to the peripheral, go to the **GATT DB** option on the Profiles page. The GATT DB allows you to access the services and characteristics of a profile directly, as shown in Figure 6-54, and to modify or receive values through BLE.

Figure 6-54. GATT DB GUI for Characteristics



The **Data Logger** option provides a textual form of all the events that has happened with a particular BLE peripheral device, including scanning and connection.

Figure 6-55. Data Logger

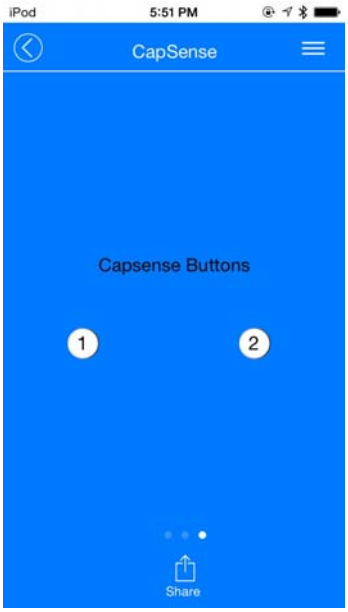


Two custom profiles are created for demonstrating the BLE Pioneer Kit features: the CapSense profile and the RGB LED profile. Both these profiles are integrated into the CySmart mobile app, as easy to use GUI.

The CapSense profile GUI supports three CapSense functionalities.

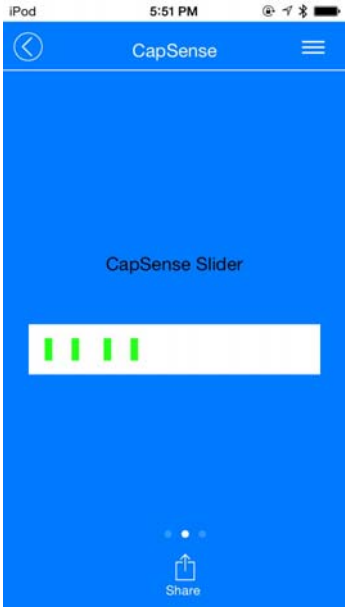
- CapSense Buttons: After connecting to the BLE peripheral, the CapSense Buttons service page displays the number of CapSense buttons supported by the peripheral, as shown in Figure 6-56. Any touch on one of the CapSense buttons on the peripheral is reflected in the CySmart GUI.

Figure 6-56. CapSense Buttons GUI Page



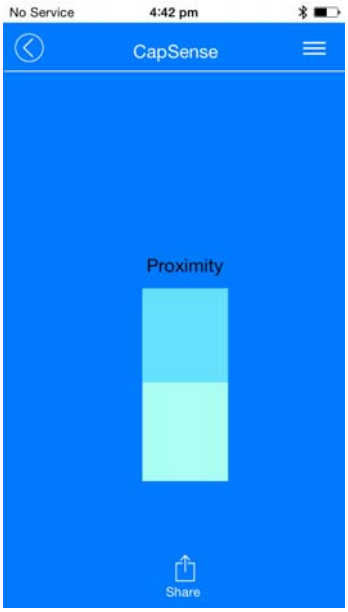
- CapSense Slider: After connecting to the BLE peripheral, the CapSense Slider service page displays the CapSense slider as supported by the peripheral, as shown in Figure 6-57. Swiping a finger on the CapSense slider on the peripheral is reflected in the CySmart GUI.

Figure 6-57. CapSense Slider GUI Page



- CapSense Proximity: After connecting to the BLE peripheral, the CapSense Proximity service page displays the CapSense proximity supported by the peripheral, as shown in Figure 6-58. A change in proximity on the proximity sensor (such as a wire) on the peripheral is reflected in the CySmart GUI.

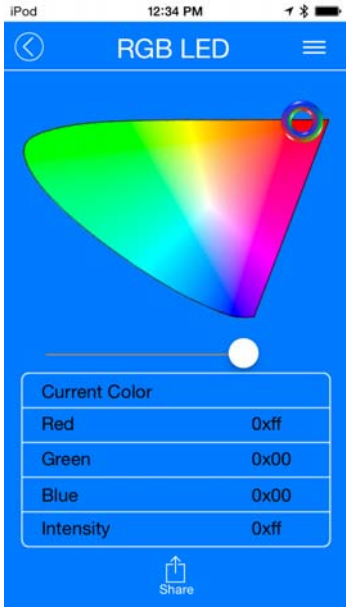
Figure 6-58. CapSense Proximity GUI Page



The RGB LED profile allows you to control the color and intensity of the BLE Pioneer Kit onboard RGB LED, as shown in Figure 6-59. Pressing any part of the color gamut on the GUI is reflected on the BLE peripheral device with the onboard RGB LED.

Note: The onboard RGB LED color range depends on the LED being used. It is possible that the complete color gamut is not reflected on the onboard RGB LED due to limitations on the LED itself.

Figure 6-59. RGB LED Profile



6.7 CySmart PC Tool

The CySmart PC tool is a BLE central host emulation tool that, along with the Dongle, allows you to connect to a BLE peripheral device and transfer data over BLE services. Also, it displays all the packets that are involved during the connection, which can be analyzed for details.

The CySmart PC Tool is installed as part of the BLE Pioneer Kit installer. To launch the software, choose **Start > All Programs > Cypress > CySmart <version> > CySmart <version>**.

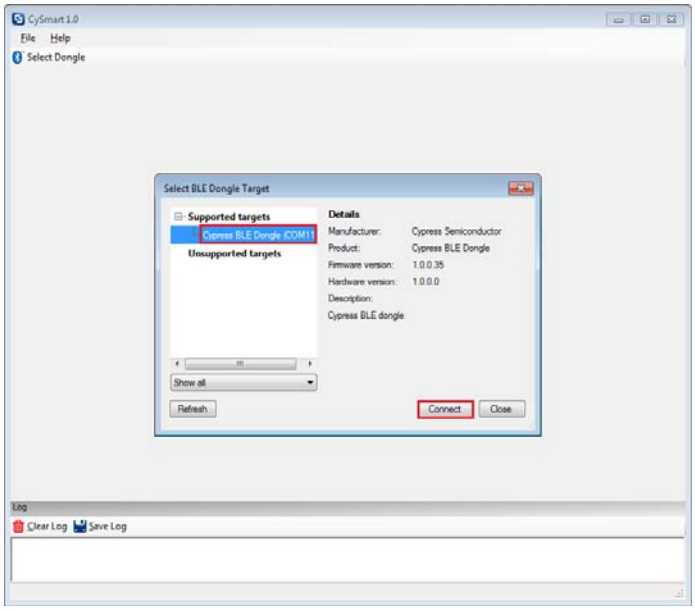
Follow these steps to connect to a BLE peripheral device using the Dongle and CySmart PC tool and to transfer data.

1. Connect the Dongle to one of the USB ports on the PC.

Figure 6-60. Connect Dongle to USB Port

2. Start the CySmart PC Tool on the PC. You will see a list of Dongles connected to it. Select the Dongle you want to use and click **Connect**, as shown in Figure 6-61.

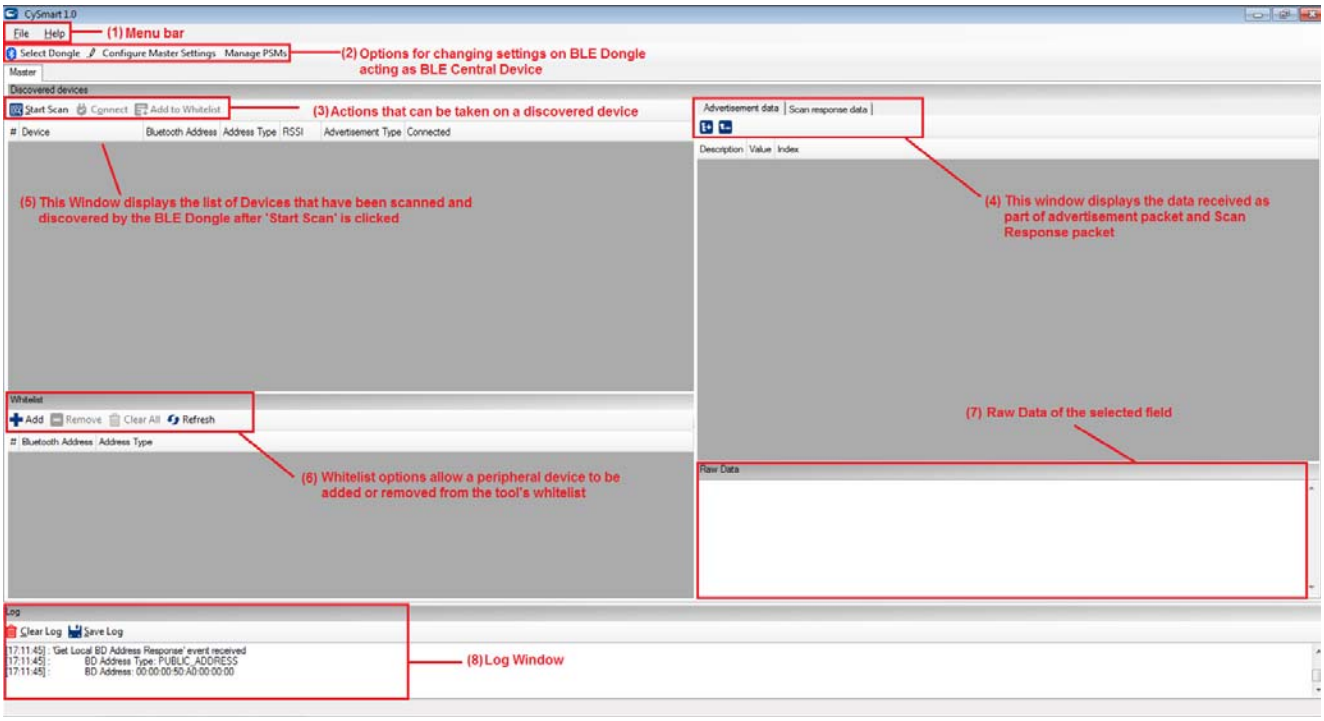
Figure 6-61. Selecting Dongle in CySmart PC Tool



3. The CySmart PC tool can be used to connect to any BLE peripheral device, including the BLE Pioneer kit. To connect to the BLE Pioneer kit, power the kit through the J13 USB connector and program the appropriate BLE peripheral project to it. Follow the steps according to the project description to start advertising.

4. When the Dongle is selected and connected to, the main window shown in Figure 6-62 opens up.

Figure 6-62. CySmart PC Tool Main Window

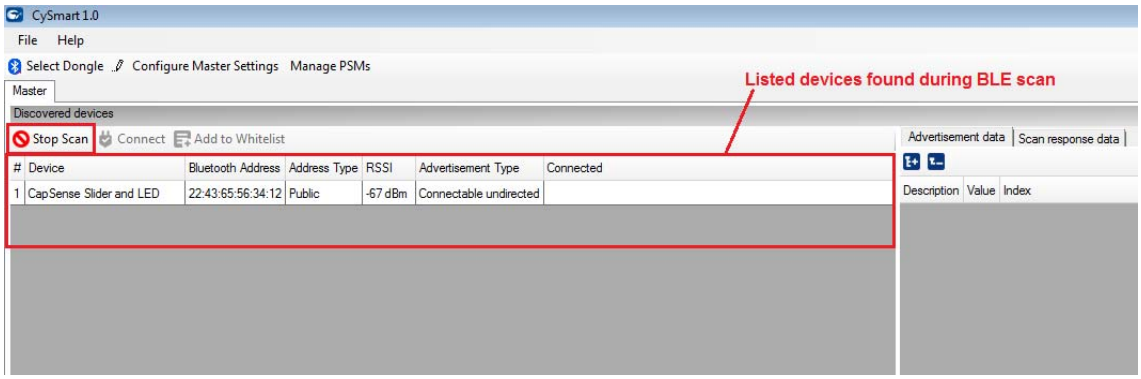


The important parts of this window are as follows:

- **Menu bar:** This contains options to exit or find help about the CySmart PC Tool.
- **Dongle settings:** These settings comprise of **Select Dongle**, **Configure Master Settings**, and **Manage PSMs**. Select Dongle allows to connect to a Dongle that is listed by the system. If a different dongle needs to be connected, then this option can be used. Configure Master Settings option allows to modify the various settings that the Dongle requires to act as a BLE Central device such as connection parameters, scan parameters, or security parameters. Manage PSMs allows to register for PSM or modify them.
- **Discovered devices options:** The **Master** tab provides three options by default: **Start Scan**, **Connect**, and **Add to Whitelist**. The Start Scan button allows the tool to start scanning for available BLE peripheral devices and list them in the Discovered Devices window. This option also allows to stop an ongoing scan. The Connect option allows to connect to a particular BLE peripheral device that is listed in the Discovered Device window. Add to Whitelist allows to add a selected device address to the whitelist.
- **Advertisement Data/Scan response data tabs:** These tabs provide the description of the data received in the advertisement packet and scan response packet from the selected device.
- **Discovered Devices window:** This window lists all the peripheral devices found after starting a scan. Selecting any device populates the information on advertisement data and scan response data on the right side window.
- **Whitelist window:** This window lists the devices that have been added as whitelist and provides options to add, remove, or clear devices from the whitelist.
- **Raw Data window:** This window displays the raw data (in hexadecimal) of the field selected.
- **Log window:** This window displays all the activities that occur on the Dongle and the data communicated. This feature is also useful for debugging.

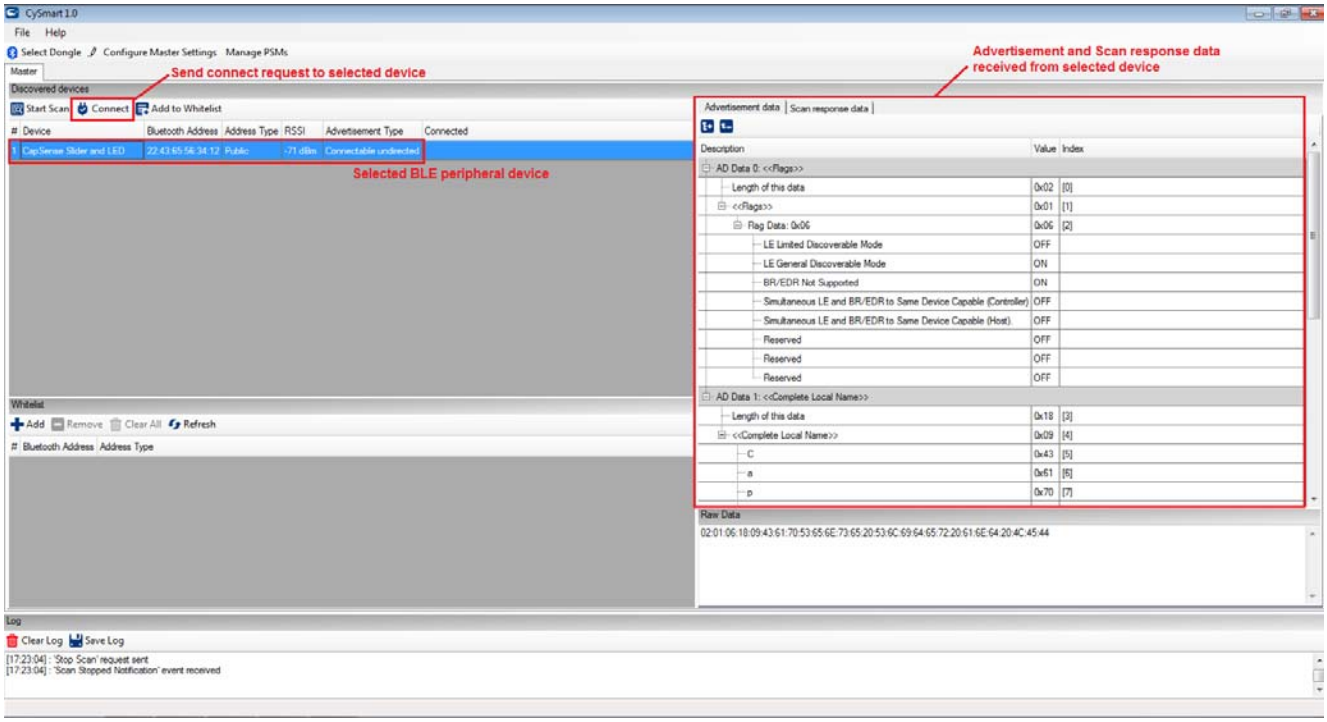
1. Click on **Start Scan** to see the list of available BLE peripheral devices, as shown in Figure 6-63.

Figure 6-63. Scanned Devices Listed in CySmart PC Tool



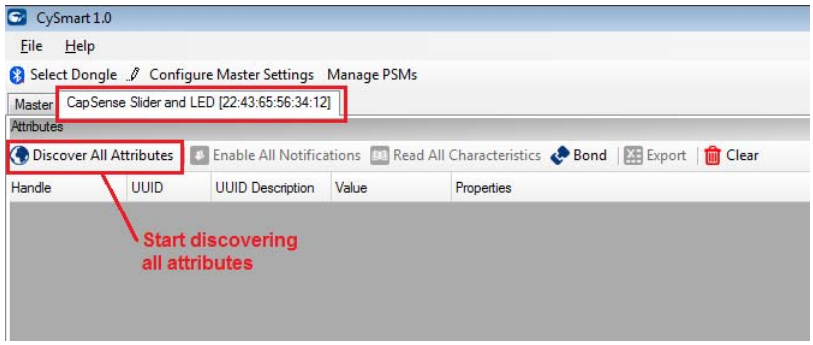
2. After the available devices are listed, choose the desired peripheral and double-click **Connect**, as shown in Figure 6-64.

Figure 6-64. Start Connection with selected device



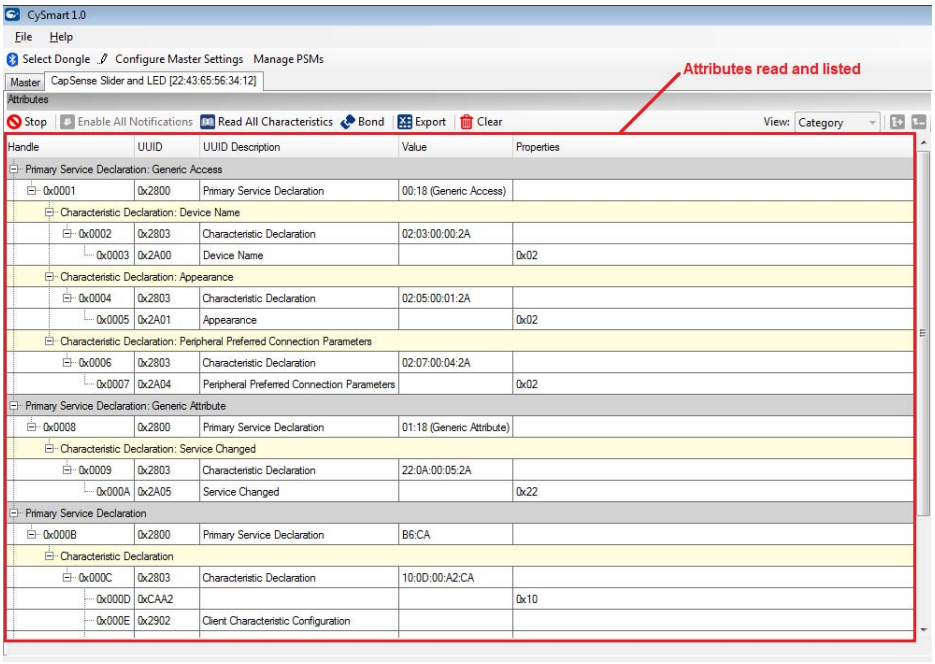
3. If the connection is successful, you will see another tab opening besides the Master tab. This tab provides options with respect to the connected BLE device, as shown in Figure 6-65.

Figure 6-65. Connected Device Tab



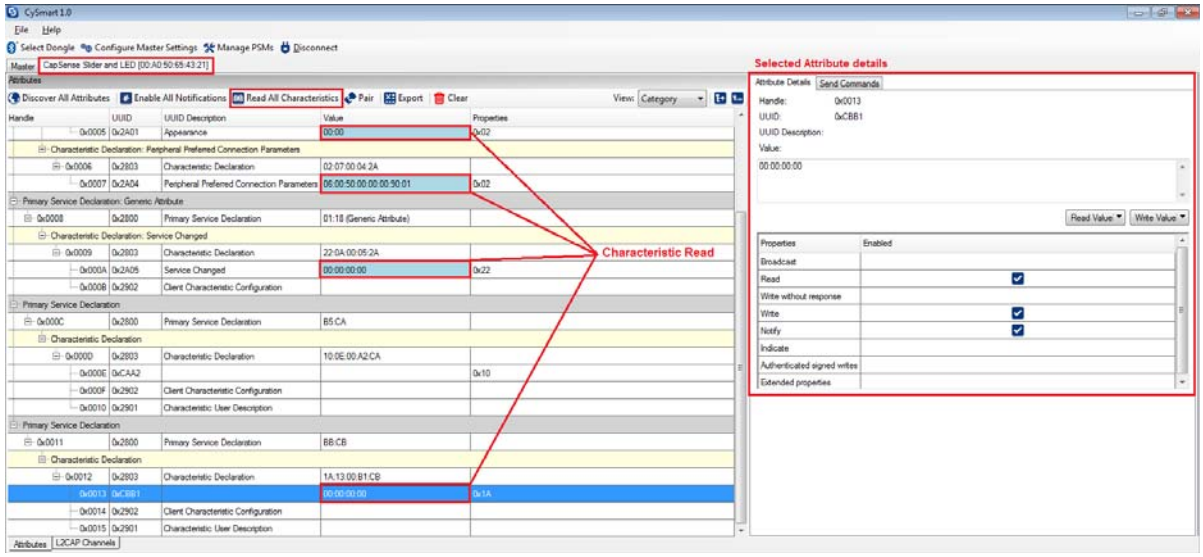
4. On the device tab, click **Discover All Attributes** to find the supported attributes by the connected BLE device. This action populates the list of services and characteristics in the Attribute window along with their values, if any, as shown in Figure 6-66.

Figure 6-66. Discover All Attributes



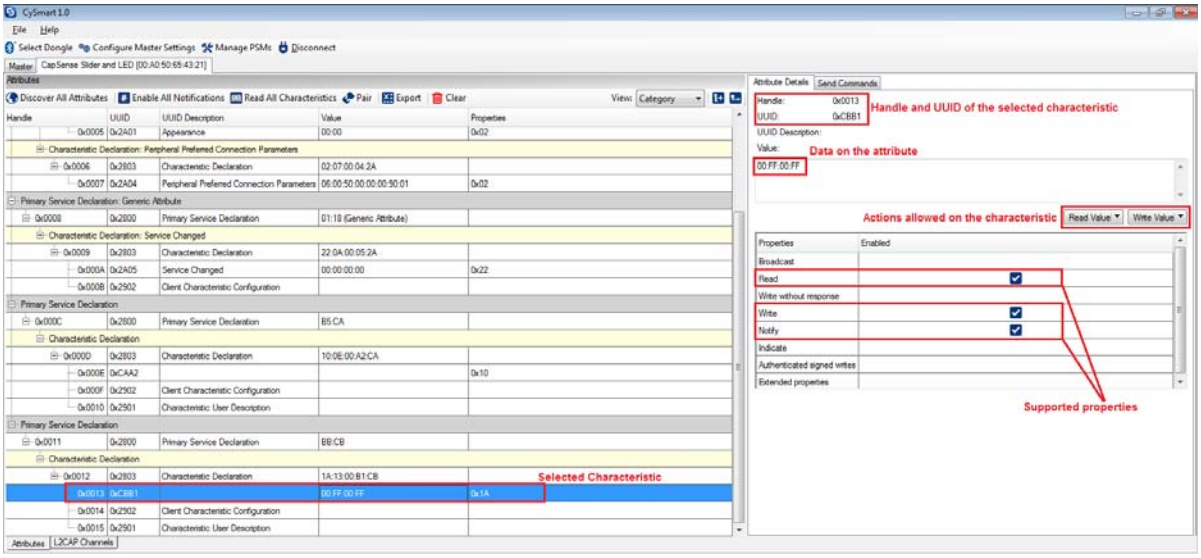
5. You can read the characteristics individually or you can use the **Read All Characteristics** option to update the values for all readable characteristics, as shown in Figure 6-67.

Figure 6-67. Read All Characteristics



6. To modify the value of a characteristic individually, select the particular characteristic from the attribute list. The Attribute Details window on the right will display the properties of the selected characteristics as well as the options to modify or read the values, as shown in Figure 6-68.

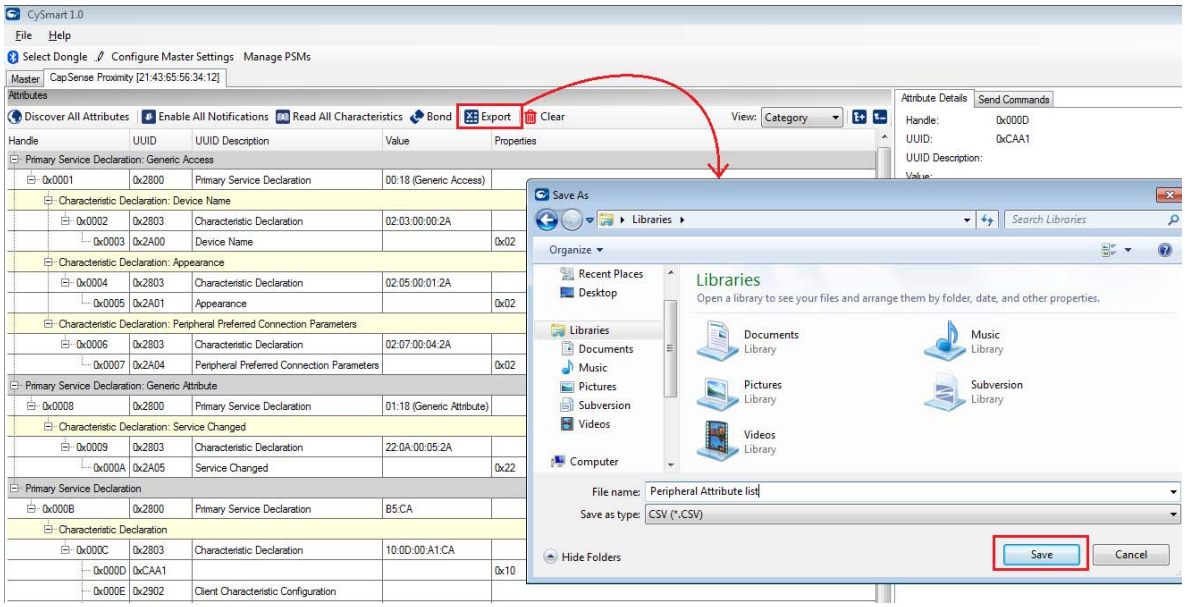
Figure 6-68. Modify a Characteristic



7. Similarly, notifications or indications can be enabled on the characteristics that support those properties.

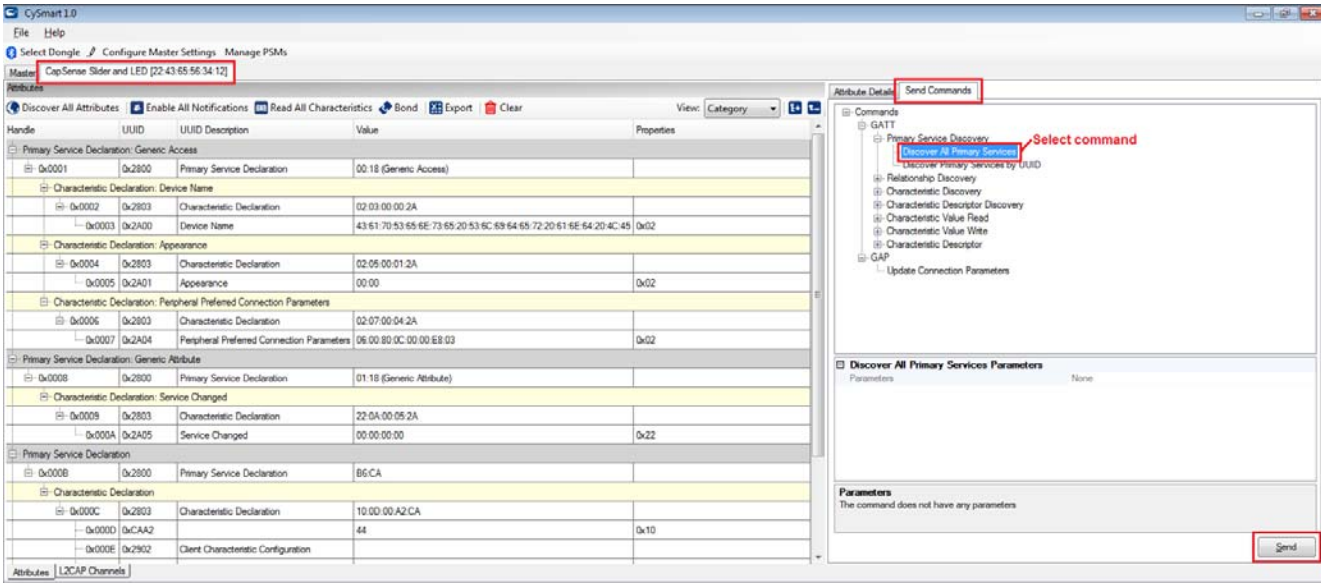
8. The list of attributes of the connected BLE device can also be saved in *.csv* format for later use. For this, click the **Export** button on the device tab and select the location where you want the file to be saved, as shown in Figure 6-69.

Figure 6-69. Save Attribute List to a File



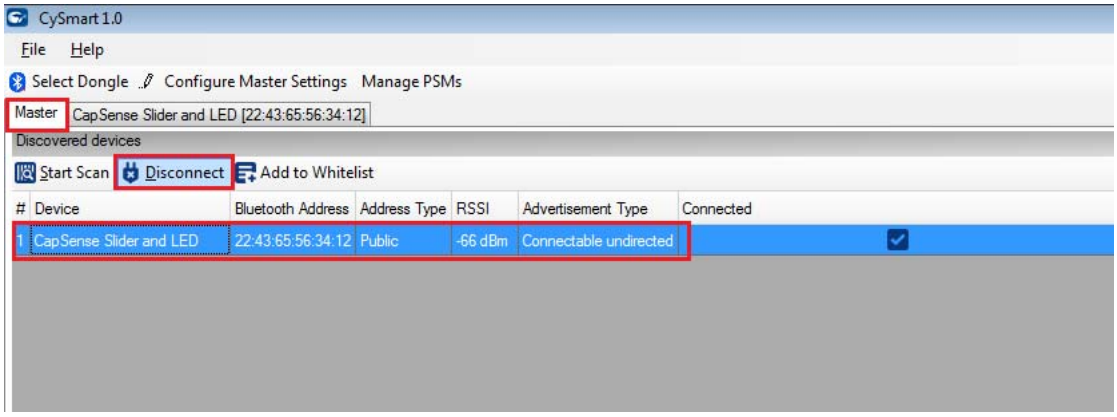
9. The tool also allows sending specific commands to the BLE peripheral device. These commands are present in the **Send Commands** tab on the device window. Select the command to be sent from the list and click **Send**, as shown in Figure 6-70.

Figure 6-70. Send Commands



10. To disconnect from the device, go to the **Master** tab, select the connected device, and click **Disconnect**, as shown in Figure 6-71.

Figure 6-71. Disconnect BLE Device

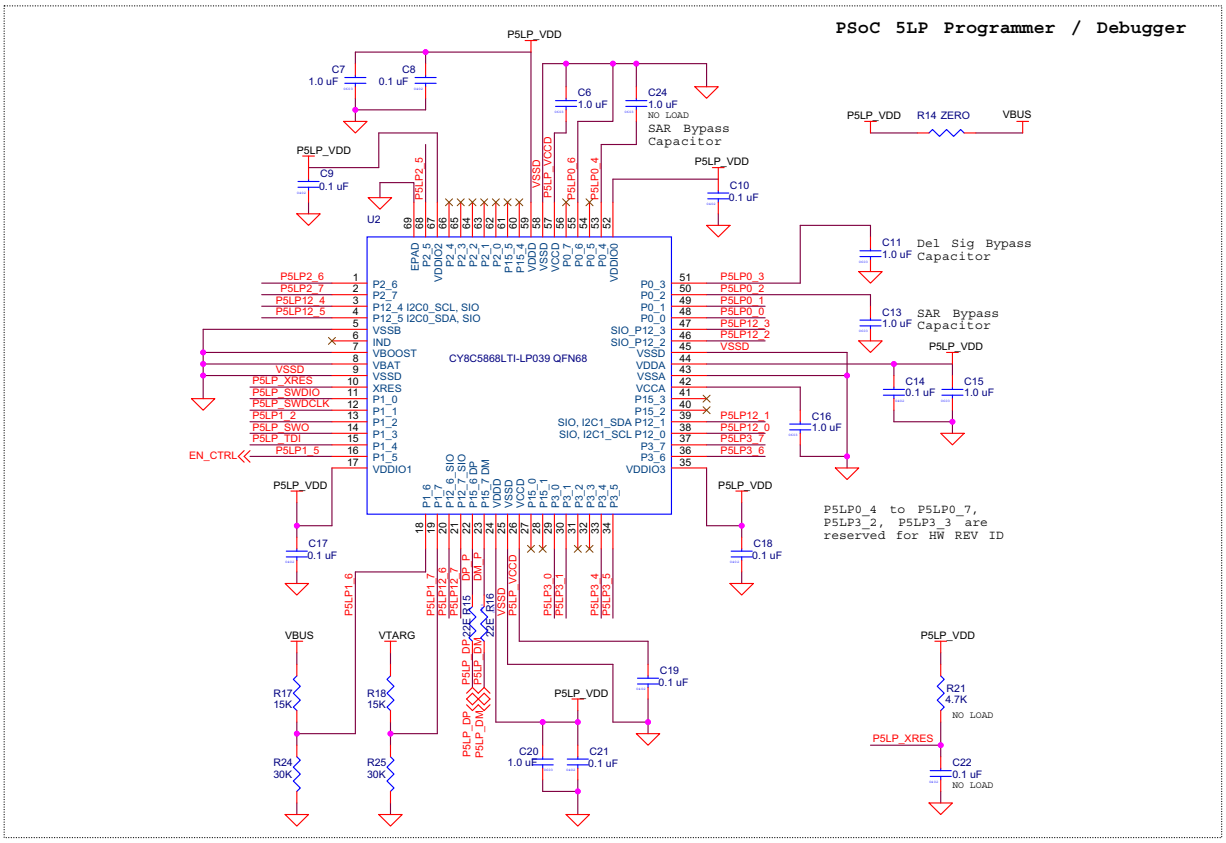


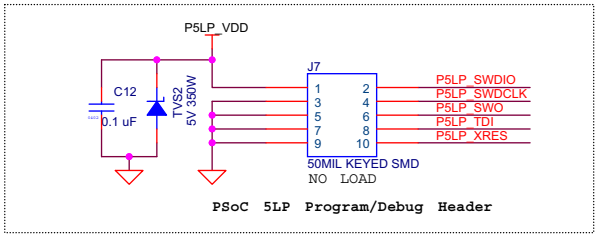
Note: Refer to CySmart PC Tool User guide for more information. This user guide can be accessed from the tool by going to **Help menu > Help Topics**.

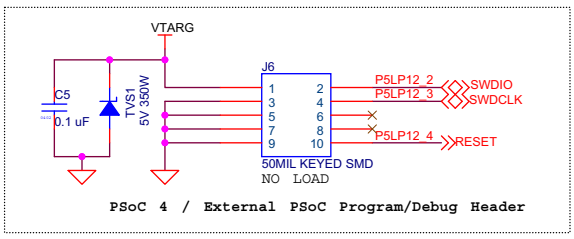
A. Appendix

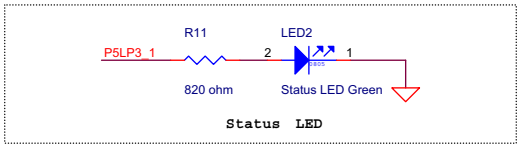
A.1 Schematics

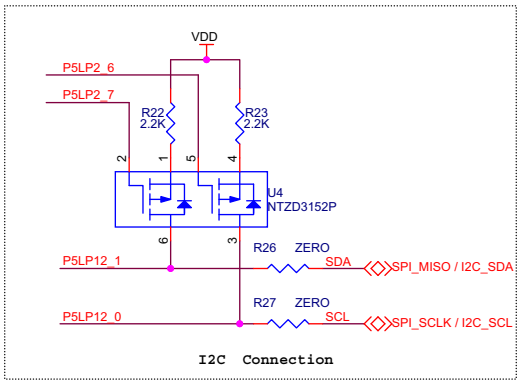
A.1.1 BLE Pioneer Board

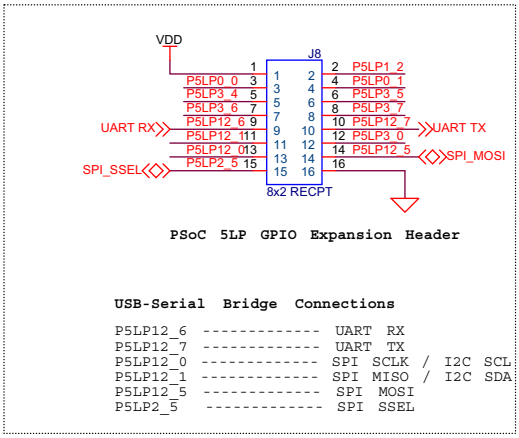


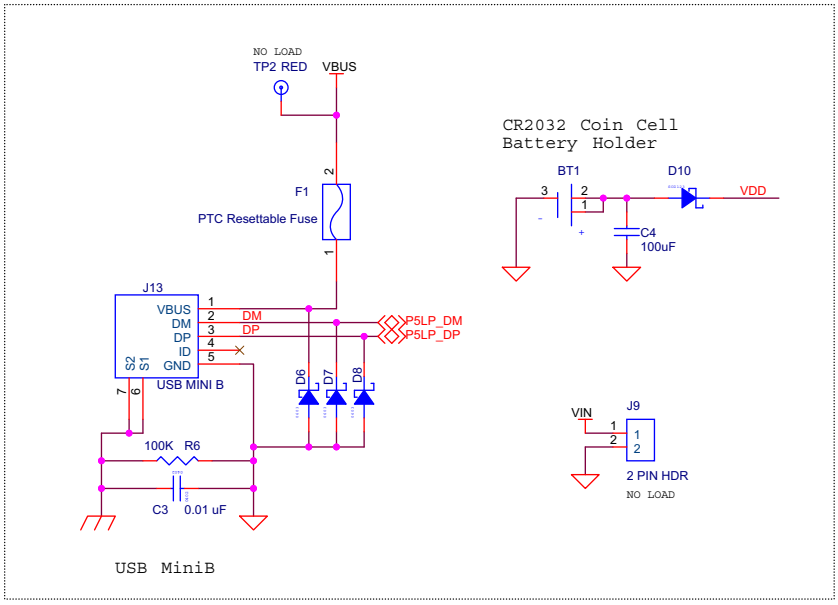


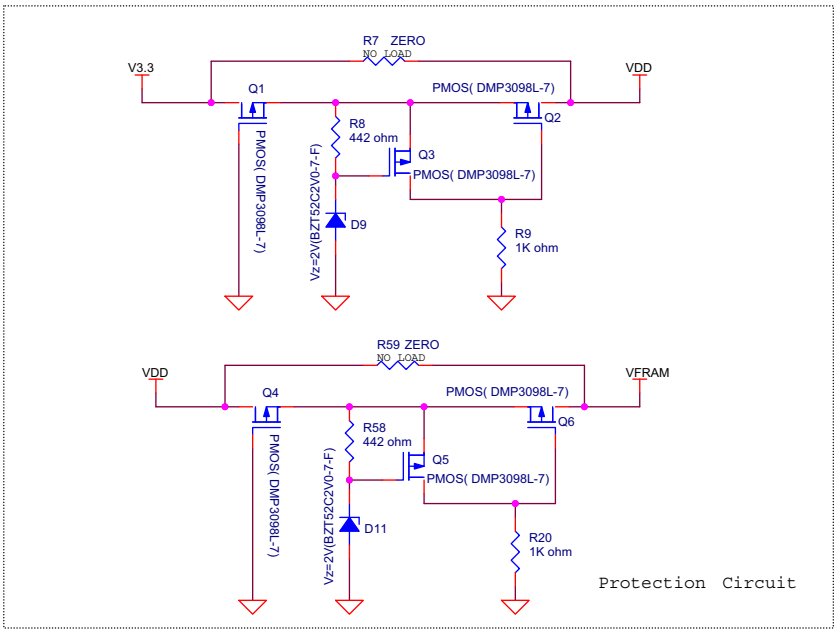



R11
LED2
P5LP3_1
2
1
820 ohm
Status LED Green
Status LED


VDD
P5LP2_6
P5LP2_7
R22
2.2K
R23
2.2K
U4
NTZD3152P
P5LP12_1
R26
ZERO
SDA
SPI_MISO / I2C_SDA
R27
ZERO
P5LP12_0
SCL
SPI_SCLK / I2C_SCL
I2C Connection


VDD
J8
P5LP0_0
P5LP3_4
P5LP3_6
P5LP12_6
P5LP12_1
P5LP12_0
P5LP2_5
P5LP1_2
P5LP0_1
P5LP3_5
P5LP3_7
P5LP12_7
P5LP3_0
P5LP12_5
UART RX
UART TX
SPI_SSEL
SPI_MOSI
8x2 RECPT
PSoC 5LP GPIO Expansion Header
USB-Serial Bridge Connections
P5LP12_6 ------------ UART RX
P5LP12_7 ------------ UART TX
P5LP12_0 ------------ SPI SCLK / I2C SCL
P5LP12_1 ------------ SPI MISO / I2C SDA
P5LP12_5 ------------ SPI MOSI
P5LP2_5 ------------ SPI SSEL


NO LOAD
TP2 RED
VBUS
F1
PTC Resettable Fuse
J13
VBUS
DM
DP
ID
GND
S2
S1
USB MINI B
P5LP_DM
P5LP_DP
D6
D7
D8
100K R6
C3 0.01 uF
USB MiniB
CR2032 Coin Cell
Battery Holder
BT1
D10
VDD
C4
100uF
VIN
J9
2 PIN HDR
NO LOAD



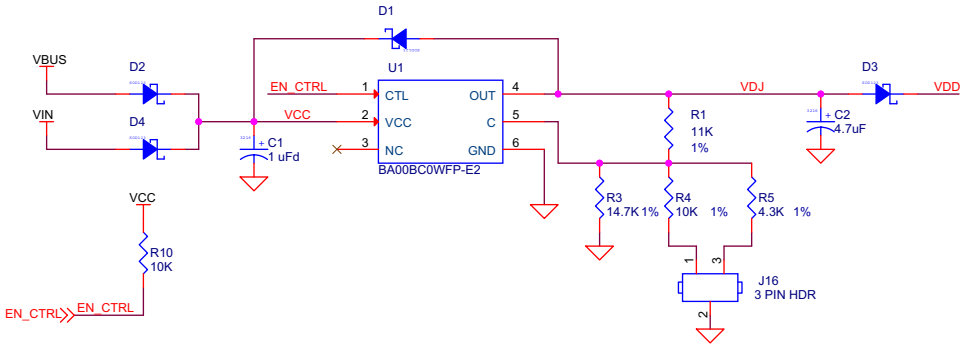
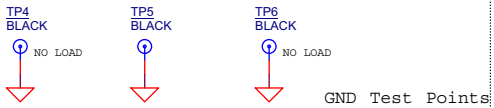


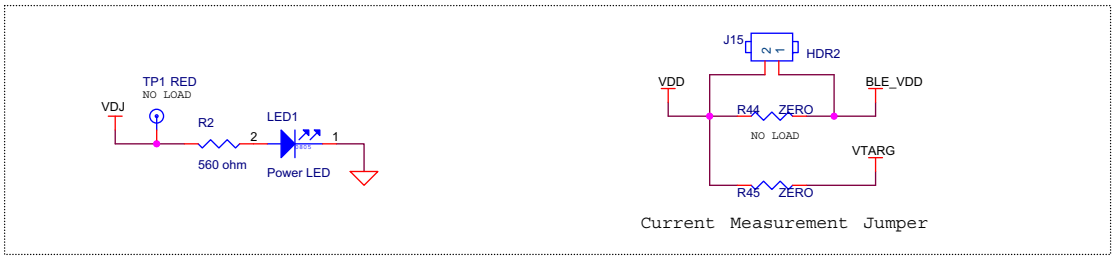
TABLE: LDO PIN FUNCTIONS

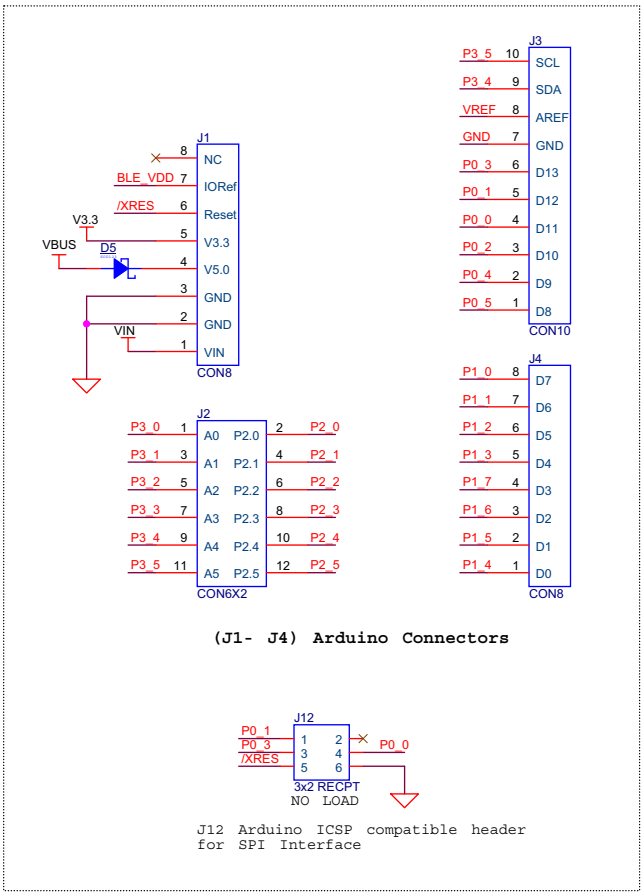
Pin No	Symbol	Function
1	CTL	Output Voltage ON / OFF control
2	VCC	Power supply voltage input
3	N.C.	Unconnected terminal
4	OUT	Voltage output
5	C	Output voltage regulation terminal
6	GND	Ground

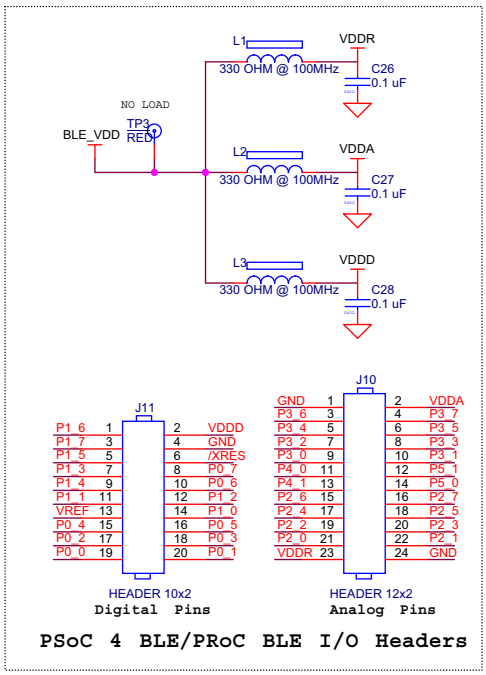
TABLE: VOLTAGE SELECTION JUMPER SETTINGS

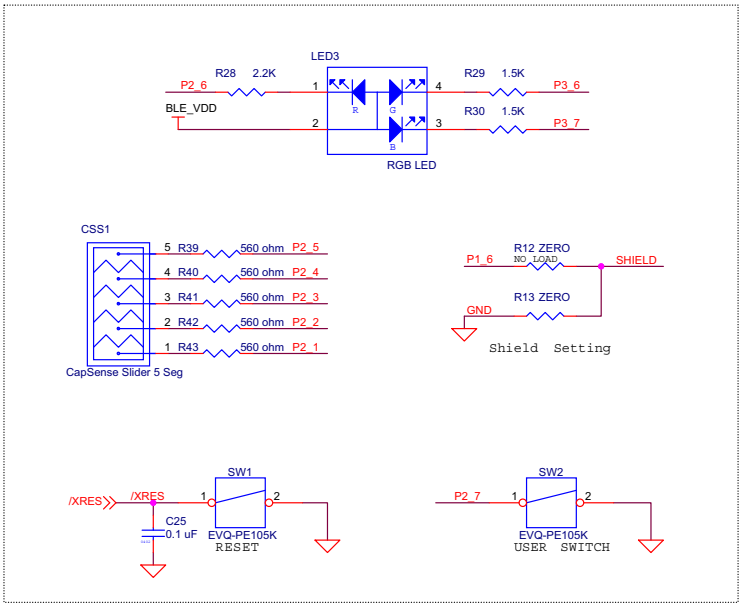
JUMPER SETTING	O/P VOLTAGE
SHORT 2 & 3	5V
SHORT 1 & 2	3.3V
REMOVE JUMPER	1.9V

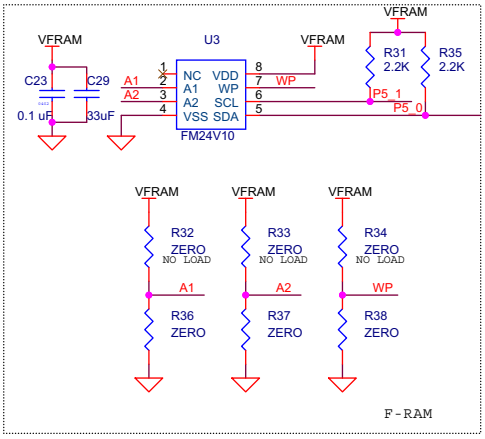


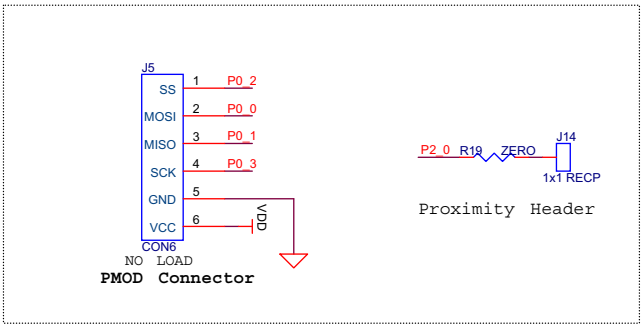

TP1 RED
NO LOAD
VDJ
R2
560 ohm
LED1
Power LED
J15
HDR2
VDD
BLE_VDD
R44
ZERO
NO LOAD
VTARG
R45
ZERO
Current Measurement Jumper

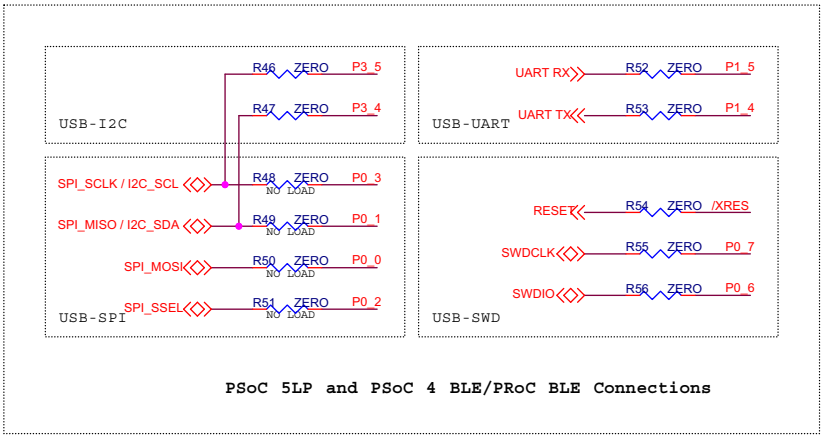

J1
NC
IORef
Reset
V3.3
V5.0
GND
GND
VIN
CON8
BLE_VDD
/XRES
V3.3
VBUS
D5
VIN
J3
SCL
SDA
AREF
GND
D13
D12
D11
D10
D9
D8
CON10
J2
A0 P2.0
A1 P2.1
A2 P2.2
A3 P2.3
A4 P2.4
A5 P2.5
CON6X2
J4
D7
D6
D5
D4
D3
D2
D1
D0
CON8
(J1- J4) Arduino Connectors
J12
3x2 RECPT
NO LOAD
J12 Arduino ICSP compatible header
for SPI Interface


L1
VDDR
330 OHM @ 100MHz
C26
0.1 uF
NO LOAD
TP3
RED
BLE_VDD
L2
VDDA
330 OHM @ 100MHz
C27
0.1 uF
L3
VDDD
330 OHM @ 100MHz
C28
0.1 uF
J11
J10
HEADER 10x2
Digital Pins
HEADER 12x2
Analog Pins
PSoC 4 BLE/PRoC BLE I/O Headers

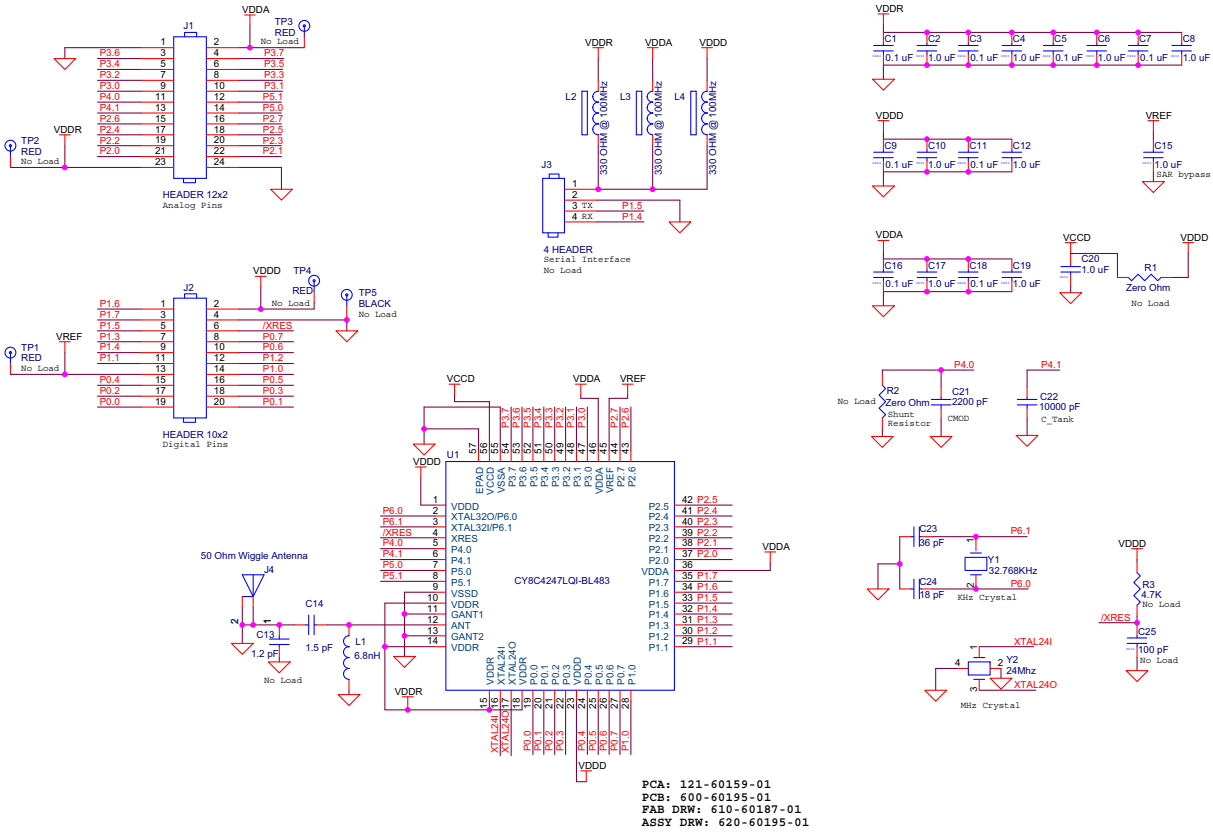

LED3
R28 2.2K
P2_6
BLE_VDD
R29 1.5K
P3_6
R30 1.5K
P3_7
RGB LED
CSS1
5 R39 560 ohm P2_5
4 R40 560 ohm P2_4
3 R41 560 ohm P2_3
2 R42 560 ohm P2_2
1 R43 560 ohm P2_1
CapSense Slider 5 Seg
P1_6
R12 ZERO
NO LOAD
SHIELD
R13 ZERO
GND
Shield Setting
/XRES
C25
0.1 uF
SW1
EVQ-PE105K
RESET
P2_7
SW2
EVQ-PE105K
USER SWITCH


VFRAM
U3
C23
0.1 uF
C29
33uF
NC
A1
A2
VSS
VDD
WP
SCL
SDA
FM24V10
R31 2.2K
R35 2.2K
P5_1
P5_0
R32 ZERO NO LOAD
R33 ZERO NO LOAD
R34 ZERO NO LOAD
R36 ZERO
R37 ZERO
R38 ZERO
F-RAM


J5
SS
MOSI
MISO
SCK
GND
VCC
P0_2
P0_0
P0_1
P0_3
VDD
CON6
NO LOAD
PMOD Connector
P2_0 R19 ZERO
J14
1x1 RECP
Proximity Header

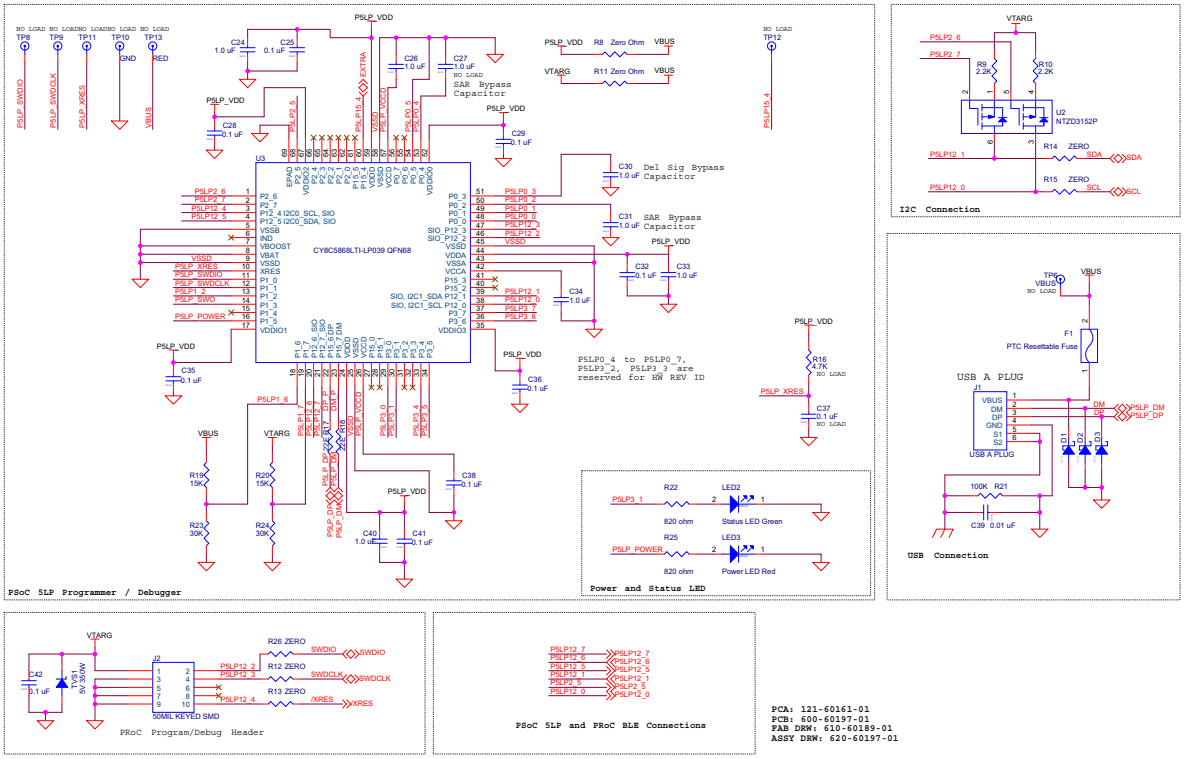

USB-I2C
R46 ZERO P3_5
R47 ZERO P3_4
USB-UART
UART RX
R52 ZERO P1_5
UART TX
R53 ZERO P1_4
SPI_SCLK / I2C_SCL
R48 ZERO P0_3
SPI_MISO / I2C_SDA
R49 ZERO P0_1
SPI_MOSI
R50 ZERO P0_0
SPI_SSEL
R51 ZERO P0_2
USB-SPI
NO LOAD
RESET
R54 ZERO /XRES
SWDCLK
R55 ZERO P0_7
SWDIO
R56 ZERO P0_6
USB-SWD
PSoC 5LP and PSoC 4 BLE/PRoC BLE Connections

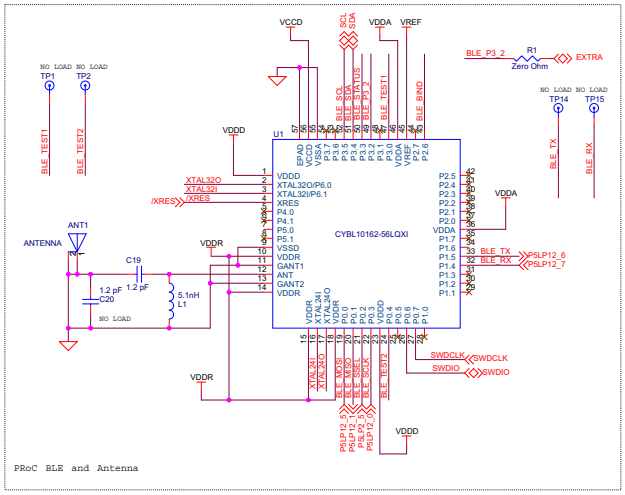
A.1.2 BLE Module

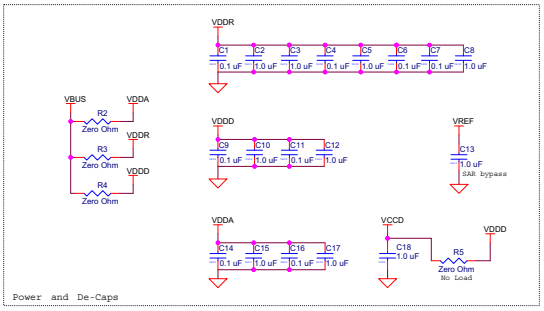


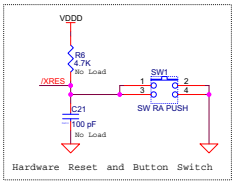
PCA: 121-60159-01
PCB: 600-60195-01
FAB DRW: 610-60187-01
ASSY DRW: 620-60195-01

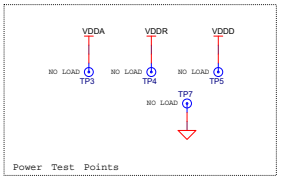
A.1.3 Dongle

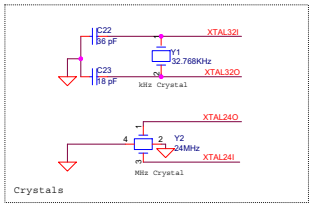

PSoC 5LP Programmer / Debugger
CY8C5868LTI-LP039 QFN68
SAR Bypass Capacitor
Del Sig Bypass Capacitor
SAR Bypass Capacitor
P5LP0_4 to P5LP0_7, P5LP3_2, P5LP3_3 are reserved for HW REV ID
Power and Status LED
Status LED Green
Power LED Red
I2C Connection
USB A PLUG
USB Connection
PTC Resettable Fuse
PRoC Program/Debug Header
PSoC 5LP and PRoC BLE Connections
PCA: 121-60161-01
PCB: 600-60189-01
FAB DRW: 610-60189-01
ASSY DRW: 620-60197-01

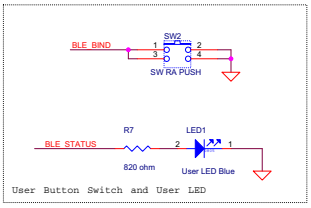












PCA: 121-60161-01
PCB: 600-60197-01
FAB DRW: 610-60189-01
ASSY DRW: 620-60197-01

A.2 Board Layout

A.2.1 BLE Pioneer Board

Figure A-1. Primary Side of BLE Pioneer Board

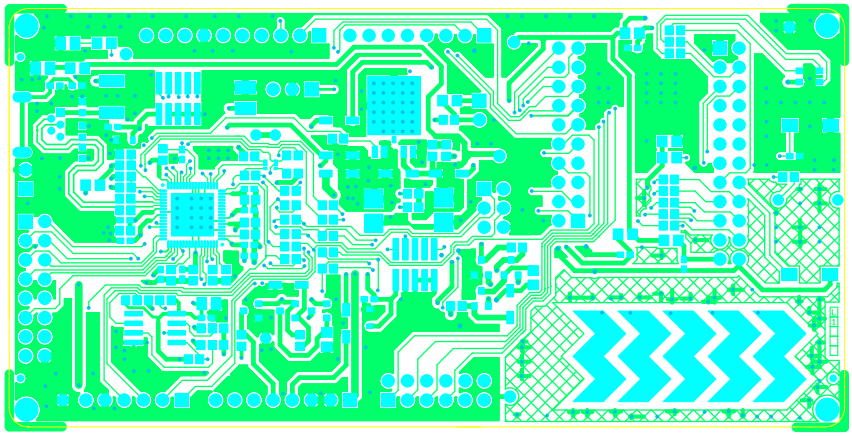


Figure A-2. Ground Layer of BLE Pioneer Board

Figure A-3. Power Layer of BLE Pioneer Board

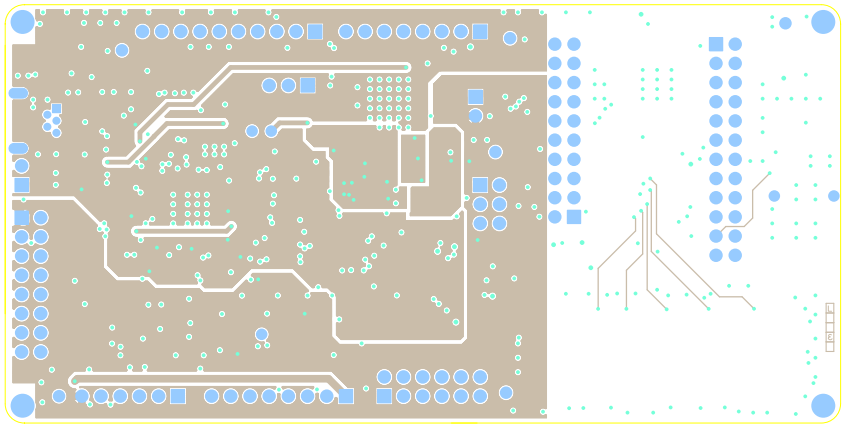


Figure A-4. Secondary Side of BLE Pioneer Board

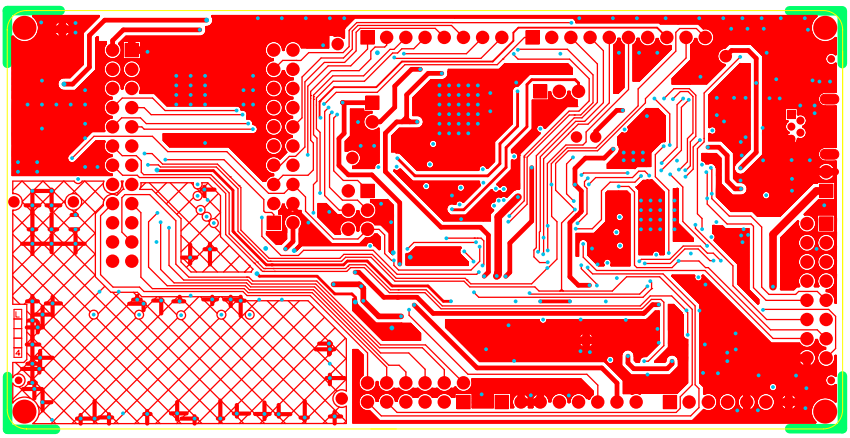


Figure A-5. Primary Silkscreen of BLE Pioneer Board

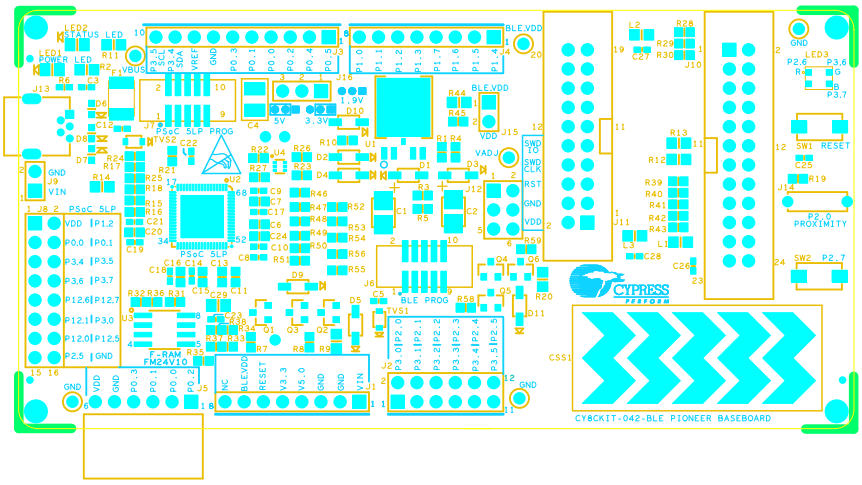
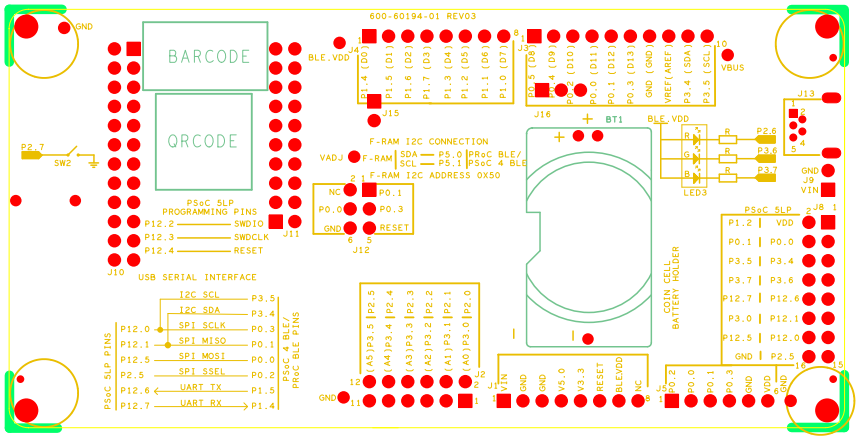


Figure A-6. Secondary Silkscreen of BLE Pioneer Board



A.2.2 PRoC BLE Module

Figure A-7. Primary Side of PRoC BLE Module

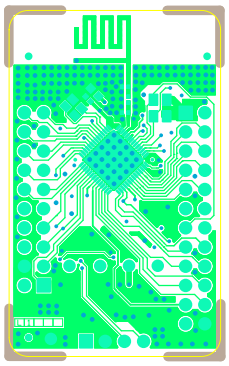


Figure A-8. Ground Layer of PRoC BLE Module

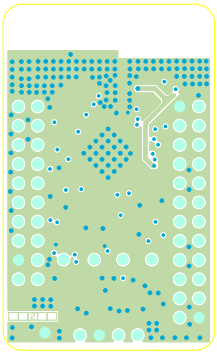


Figure A-9. VCC Layer of PRoC BLE Module

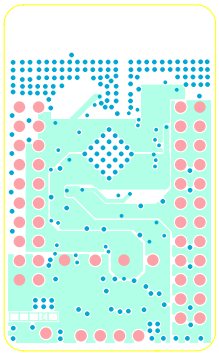


Figure A-10. Secondary Side of PRoC BLE Module

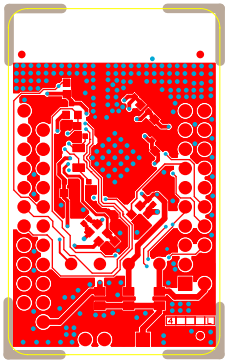


Figure A-11. Primary Silkscreen of PRoC BLE Module

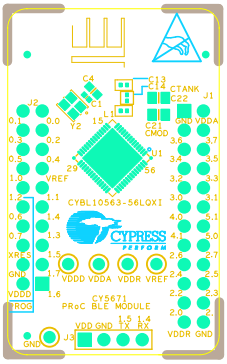
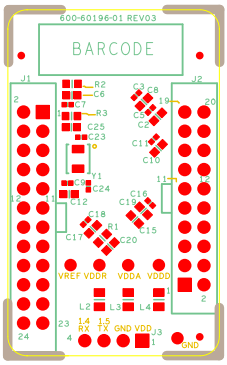


Figure A-12. Secondary Silkscreen of PRoC BLE Module



A.2.3 PSoC 4 BLE Module

Figure A-13. Primary Side of PSoC 4 BLE Module

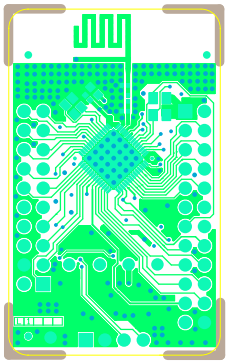


Figure A-14. Ground Layer of PSoC 4 BLE Module

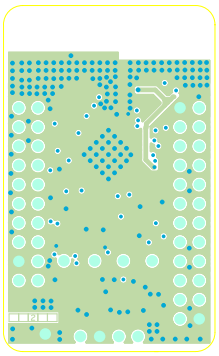


Figure A-15. VCC Layer of PSoC 4 BLE Module

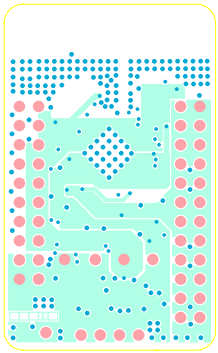


Figure A-16. Secondary Side of PSoC 4 BLE Module

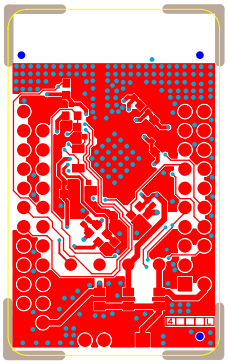


Figure A-17. Primary Silkscreen of PSoC 4 BLE Module

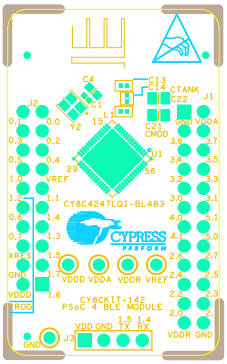
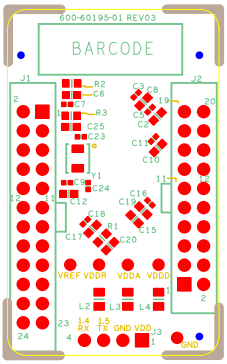


Figure A-18. Secondary Silkscreen of PSoC 4 BLE Module



A.2.4 Dongle

Figure A-19. Primary Side of Dongle

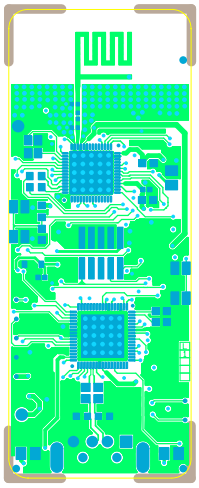


Figure A-20. Ground Layer of Dongle

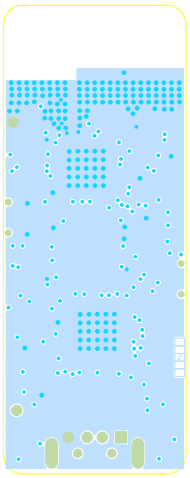


Figure A-21. Power Layer of Dongle

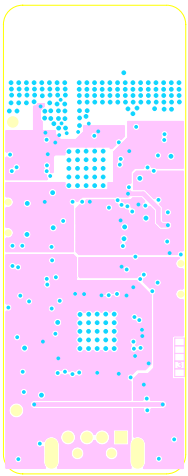


Figure A-22. Secondary Side of Dongle

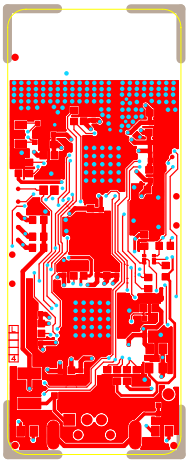


Figure A-23. Primary Silkscreen of Dongle

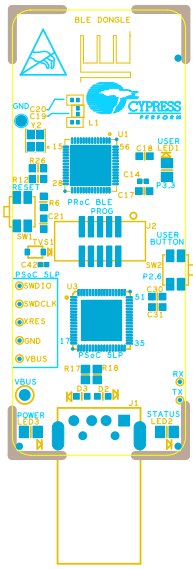
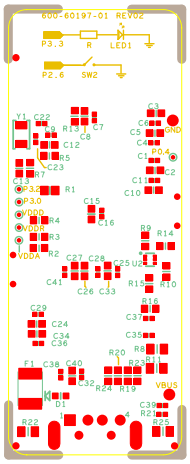


Figure A-24. Secondary Silkscreen of Dongle



A.3 Bill of Materials (BOM)

A.3.1 BLE Pioneer Board

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
1			-	PCB, 106.22 mm x 53.34 mm, High Tg, ENIG finish, 4 layer, Color = RED, Silk = WHITE.	Cypress				N/A	Yes	N/A
2	1	BT1	CR2032 Battery Holder	HOLDER COIN CELL CR2032 EJECT	MPD	BA2032			ASK	Yes	Yes
3	1	C1	1.0 uF	CAP TANT 1UF 35V 10% 1210	AVX Corporation	TAJB105 K035RNJ			ASK	Yes	Yes
4	1	C2	4.7 uF	CAP TANT 4.7UF 20V 10% 1210	AVX Corporation	TAJB475 K020RNJ			ASK	Yes	Yes
5	1	C3	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	TDK Corporation	C1005X7 R1C103K 050BA			ASK	Yes	Yes
6	1	C4	100 uFd	CAP CER 100UF 6.3V 20% X5R 1210	TDK Corporation	C3225X5 R0J107M 250AC			ASK	Yes	Yes
7	15	C5, C8, C9, C10, C12, C14, C17, C18, C19, C21, C23, C25, C26, C27, C28	0.1 uFd	CAP .1UF 16V CERAMIC X5R 0402	TDK Corporation	C1005X5 R1A104K 050BA			ASK	Yes	Yes
8	7	C6, C7, C11, C13, C15, C16, C20	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107B J105KA-T			ASK	Yes	Yes
9	1	C29	33 uF	CAP CER 33UF 6.3V 20% X5R 0805	TDK Corporation	C2012X5 R0J336M 125AC					
10	6	D1, D2, D3, D4, D5, D10	MBR0520L	DIODE SCHOTTKY 0.5A 20V SOD-123	Fairchild Semiconductor	MBR0520 L			ASK	Yes	Yes
11	3	D6, D7, D8	ESD diode	SUPPRES-SOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603M LC-05LE			ASK	Yes	Yes
12	2	D9, D11	2V Zener	DIODE ZENER 2V 500MW SOD123	Diodes Inc	BZT52C2 V0-7-F			ASK	Yes	Yes
13	1	F1	FUSE	PTC RESET-TABLE .50A 15V 1812	Bourns	MF-MSMF050 -2			ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
14	2	J1, J4	8x1 RECP	CONN HEADER FEMALE 8POS .1" GOLD	Protectron Electromech	P9401-08-21	Sullins Connector Solutions	PPPC081LFBN-RC	ASK	Yes	Yes
15	1	J2	6x2 RECP	CONN HEADER FMAL 12PS.1" DL GOLD	Protectron Electromech	P9403-12-21	Sullins Connector Solutions	PPPC062LFBN-RC	ASK	Yes	Yes
16	1	J3	10x1 RECP	CONN HEADER FMALE 10POS .1" GOLD	Protectron Electromech	P9401-10-21	Sullins Connector Solutions	PPPC101LFBN-RC	ASK	Yes	Yes
17	1	J8	8X2 RECP	CONN HEADER FMAL 16PS.1" DL GOLD	Protectron Electromech	P9403-16-21	Sullins Connector Solutions	PPPC082LFBN-RC	ASK	Yes	Yes
18	1	J10	12X2 RECP	CONN HEADER 2.54MM 24POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D12-ST-BK			ASK	Yes	Yes
19	1	J11	10X2 RECP	CONN HEADER 2.54MM 20POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D10-ST-BK			ASK	Yes	Yes
20	1	J13	USB MINI B	MINI USB RCPT R/A DIP	TE Connectivity	1734510-1	Molex Inc	0548190519	ASK	Yes	Yes
21	1	J14	1X1 RECP	CONN RCPT 1POS .100" SNGL HORZ	Samtec Inc	BCS-101-L-S-HE			ASK	Yes	Yes
22	1	J15	2p_jumper	CONN HEADR BRKWAY .100 2POS STR	Protectron Electromech	P9101-02-12-1	TE Connectivity	5-146280-2	ASK	Yes	Yes
23	1	J16	3p_jumper	CONN HEADR BRKWAY .100 3POS STR	Protectron Electromech	P9101-03-12-1	TE Connectivity	5-146280-3	ASK	Yes	Yes
24	1	LED1	Power LED Amber	LED 595NM AMB DIFF 0805 SMD	Avago Technologies	HSMA-C170			ASK	Yes	Yes
25	1	LED2	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8			ASK	Yes	Yes
26	1	LED3	RGB LED	LED RED/GREEN/BLUE PLCC4 SMD	Cree, Inc.	CLV1A-FKB-CJ1M1F1BB7R4S3			ASK	Yes	Yes
	3	L1, L2, L3	330 OHM @ 100MHz	FERRITE CHIP 330 OHM 0805	Murata	BLM21PG331SN1D					
27	6	Q1, Q2, Q3, Q4, Q5, Q6	PMOS	MOSFET P-CH 30V 3.8A SOT23-3	Diodes Inc	DMP3098L-7			ASK	Yes	Yes
28	1	R1	11K 1%	RES 11K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1102V			ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
29	1	R2	560 ohm	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ561V			ASK	Yes	Yes
30	1	R3	14.7K 1%	RES 14.7K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1472V			ASK	Yes	Yes
32	1	R4	10K 1%	RES 10K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1002V			ASK	Yes	Yes
31	1	R5	4.3K 1%	RES 4.3K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF4301V			ASK	Yes	Yes
33	1	R6	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X			ASK	Yes	Yes
34	14	R19, R26, R27, R36, R37, R38, R45, R46, R47, R52, R53, R54, R55, R56	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V			ASK	Yes	Yes
35	2	R8, R58	442 ohm 1%	RES 442 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF4420V			ASK	Yes	Yes
36	2	R9, R20	1K	RES 1K OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ102V			ASK	Yes	Yes
37	1	R10	10K	RES 10K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ103V			ASK	Yes	Yes
38	1	R11	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ821V			ASK	Yes	Yes
39	2	R13, R14	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic- ECG	ERJ-6GEY0R00V			ASK	Yes	Yes
40	2	R15, R16	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V			ASK	Yes	Yes
41	2	R17, R18	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V			ASK	Yes	Yes
42	5	R22, R23, R28, R31, R35	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V			ASK	Yes	Yes
43	2	R24, R25	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V			ASK	Yes	Yes
44	2	R29, R30	1.5K	RES 1.5K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ152V			ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
45	5	R39, R40, R41, R42, R43	560 ohm	RES 560 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ561V			ASK	Yes	Yes
46	2	SW1, SW2	SW PUSH-BUTTON	SWITCH TAC-TILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-PE105K			ASK	Yes	Yes
47	1	TP5	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001			ASK	Yes	Yes
48	2	TVS1,TVS2	5V 350W	TVS UNIDIR 350W 5V SOD-323	Dioded Inc.	SD05-7			ASK	Yes	Yes
49	1	U1	LDO	IC REG LDO ADJ 1A TO252-5	Rohm Semiconductor	BA00BC0WFP-E2			ASK	Yes	Yes
50	1	U2	PSoC 5LP	68QFN PSoC 5LP chip for USB debug channel and USB-Serial interface	Cypress Semiconductor	CY8C5868LTI-LP039			ASK	Yes	Yes
51	1	U3	F-RAM	F-RAM 1-Mbit (128K X 8) I2C interface	Cypress Semiconductor	FM24V10-G			ASK	Yes	Yes
52	1	U4	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G			ASK	Yes	Yes
Install on Bottom of PCB As per the Silk Screen in the Corners											
53	4	N/A	N/A	BUMPER CYLIN 0.375" DIA BLK	3M	SJ61A4			ASK	Yes	Yes
Special Jumper Installation Instructions											
54	2	J15,J16	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E			ASK	Yes	Yes
Label											
56	1	N/A	N/A	LBL, PCA Label, Vendor Code, Date-code, Serial Number 121-60158-01 Rev 04 (YYWWV-VXXXXX)	Cypress Semiconductor				ASK	Yes	Yes
57	1	N/A	N/A	LBL, QR code, 12mm X 12mm	Cypress Semiconductor				ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
No load components											
58	1	C22	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA			ASK	Yes	Yes
59	1	C24	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T			ASK	Yes	Yes
60	9	R7, R59, R32, R33, R34, R48, R49, R50, R51	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1			ASK	Yes	Yes
61	1	R21	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V			ASK	Yes	Yes
62	2	J7, J6	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K	FCI	20021521-00010T1LF	ASK	Yes	Yes
63	1	J9	2 PIN HDR	CONN HEADER FEMALE 2POS .1" GOLD	Sullins Connector Solutions	PPPC021LFBN-RC	Protectron Electromech	P9401-02-21	ASK	Yes	Yes
64	2	TP4, TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001			ASK	Yes	Yes
65	3	TP1, TP2, TP3	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000			ASK	Yes	Yes
66	2	R44, R12	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V			ASK	Yes	Yes
67	1	J12	3x2 RECPT	CONN HEADER FMAL 6PS .1" DL GOLD	Sullins Connector Solutions	PPPC032LFBN-RC	Protectron Electromech	P9403-06-21	ASK	Yes	Yes
68	1	J5	6X1 RECP RA	CONN FEMALE 6POS .100" R/A GOLD	Sullins Connector Solutions	PPPC061LGBN-RC			ASK	Yes	Yes

A.3.2 BLE Module

A.3.2.1 CY5671 PRoC BLE Module

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
1	1	600-60196-01	-	PRoC BLE Module printed circuit board	Cypress qualified vendor	600-60196-01 Rev03	No	Yes	Yes
2	8	C1, C3, C5, C7, C9, C11, C16, C18	0.1 uF	CAP .1UF 16V CERAMIC Y5V 0402	Samsung Electro-Mechanics America, Inc	CL05F104ZO5NNNC	ASK	Yes	Yes
3	10	C2, C4, C6, C8, C10, C12, C15, C17, C19, C20	1.0 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	TDK Corporation	C1608X5R1E105K080AC	ASK	Yes	Yes
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222JA01D	ASK	Yes	Yes
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103JA01D	ASK	Yes	Yes
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D	ASK	Yes	Yes
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D	ASK	Yes	Yes
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T	ASK	Yes	Yes
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC-D12-ST-BK	ASK	Yes	Yes
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC-D10-ST-BK	ASK	Yes	Yes
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technology Inc	L-07C6N8JV6T	ASK	Yes	Yes
12	3	L2, L3, L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1D	ASK	Yes	Yes
13	1	U1	PRoC BLE	56 QFN PRoC BLE	Cypress Semiconductor	CYBL10563-56LQXI	ASK	Yes	Yes
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B	ASK	Yes	Yes
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM	ASK	Yes	Yes
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60160-01 Rev 04 (YYWWVVXXXXX)	Cypress qualified vendor	-	ASK	Yes	Yes
No Load components									
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T	ASK	Yes	Yes
18	1	C25	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU	ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
19	1	R1	Zero Ohm	RES 0.0 OHM 1/ 8W 0605 SMD	TE Connectivity	1623094-1	ASK	Yes	Yes
20	1	R2	Rbleed	No Load	-	-	ASK	Yes	Yes
21	1	R3	4.7K	RES 4.7K OHM 1/ 10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V	ASK	Yes	Yes
22	1	J3	4 HEADER	CONN HEADER 4POS .100 R/A 15AU	FCI	68016-204HLF	ASK	Yes	Yes
23	4	TP1, TP2, TP3, TP4	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000	ASK	Yes	Yes
24	1	TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001	ASK	Yes	Yes

A.3.2.2 CY8CKIT-142 PSoC 4 BLE Module

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
1	1	600-60195-01	-	PSoC 4 BLE Module printed circuit board	Cypress qualified vendor	600-60195-01 Rev03	No	Yes	Yes
2	8	C1, C3, C5, C7, C9, C11, C16, C18	0.1 uF	CAP .1UF 16V CERAMIC Y5V 0402	Samsung Electro-Mechanics America, Inc	CL05F104ZO5NNNC	ASK	Yes	Yes
3	10	C2, C4, C6, C8, C10, C12, C15, C17, C19, C20	1.0 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	TDK Corporation	C1608X5R1E105K080AC	ASK	Yes	Yes
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222JA01D	ASK	Yes	Yes
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103JA01D	ASK	Yes	Yes
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D	ASK	Yes	Yes
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D	ASK	Yes	Yes
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T	ASK	Yes	Yes
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC-D12-ST-BK	ASK	Yes	Yes
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC-D10-ST-BK	ASK	Yes	Yes
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technology Inc	L-07C6N8JV6T	ASK	Yes	Yes
12	3	L2, L3, L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1D	ASK	Yes	Yes
13	1	U1	PSoC 4BLE	56 QFN PSoC 4 BLE	Cypress Semiconductor	CY8C4247LQI-BL483	ASK	Yes	Yes
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B	ASK	Yes	Yes
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM	ASK	Yes	Yes
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60159-01 Rev 04 (YYWWVVXXXXX)	Cypress qualified vendor	-	ASK	Yes	Yes
No Load components									
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T	ASK	Yes	Yes
18	1	C25	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU	ASK	Yes	Yes
19	1	R1	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1	ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
20	1	R2	Rbleed	No Load	-	-	ASK	Yes	Yes
21	1	R3	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V	ASK	Yes	Yes
22	1	J3	4 HEADER	CONN HEADER 4POS .100 R/A 15AU	FCI	68016-204HLF	ASK	Yes	Yes
23	4	TP1, TP2, TP3, TP4	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000	ASK	Yes	Yes
24	1	TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001	ASK	Yes	Yes

A.3.3 Dongle

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Sub Allowed	ROHS	UL Certified
1	1	600-60197-01	-	PCB, 60 mm x 30 mm, High Tg, ENIG finish, 2 layer, Color = BLACK, Silk = WHITE.	Cypress qualified vendor	600-60197-01 Rev02	N/A	Yes	N/A
2	17	C1, C4, C6, C7, C9, C11, C14, C16, C25, C28, C29, C32, C35, C36, C38, C41, C42	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA	ASK	Yes	Yes
3	17	C2, C3, C5, C8, C10, C12, C13, C15, C17, C18, C24, C26, C30, C31, C33, C34, C40	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T	ASK	Yes	Yes
4	1	C19	1.2 pFd	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T	ASK	Yes	Yes
5	1	C22	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D	ASK	Yes	Yes
6	1	C23	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D	ASK	Yes	Yes
7	1	C39	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	TDK Corporation	C1005X7R1C103K050BA	ASK	Yes	Yes
8	3	D1,D2,D3	ESD diode	SUPPRESSOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603MLC-05LE	ASK	Yes	Yes
9	1	F1	FUSE	PTC RESETTABLE .50A 15V 1812	Bourns	MF-MSMF050-2	ASK	Yes	Yes
10	1	J1	USB A PLUG	CONN PLUG USB 4POS RT ANG PCB	Molex Inc	480370001	ASK	Yes	Yes
11	1	J2	50MIL KEYED SMD	CONN HEADER 10POS DUAL SHRD SMD	FCI	20021521-00010T1LF			
12	1	LED1	Status LED Blue	LED BLUE CLEAR THIN 0805 SMD	LiteOn Inc	LTST-C171TBKT	ASK	Yes	Yes
13	1	LED2	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8	ASK	Yes	Yes
14	1	LED3	Power LED Red	LED SUPER RED CLEAR 0805 SMD	LiteOn Inc	LTST-C170KRKT	ASK	Yes	Yes
15	1	L1	5.1 nH	CER INDUCTOR 5.1NH 0402	Johanson Technology Inc	L-07C5N1SV6T	ASK	Yes	Yes
16	2	R8, R11	Zero Ohm	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V	ASK	Yes	Yes
17	1	R7	820 ohm	RES 820 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ821V	ASK	Yes	Yes
18	2	R22, R25	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ821V	ASK	Yes	Yes
19	2	R9, R10	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V	ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Sub Allowed	ROHS	UL Certified
20	9	R1, R2, R3, R4, R12, R13, R14, R15, R26	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V	ASK	Yes	Yes
21	2	R17, R18	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V	ASK	Yes	Yes
22	1	R21	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X	ASK	Yes	Yes
23	2	R19, R20	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V	ASK	Yes	Yes
24	2	R23, R24	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V	ASK	Yes	Yes
25	2	SW1, SW2	SW RA PUSH	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-P3401P	ASK	Yes	Yes
26	1	TVS1	5V 350W	TVS UNIDIR 350W 5V SOD-323	Diodes Inc.	SD05-7	ASK	Yes	Yes
27	1	U1	PRoC BLE	PRoC BLE, Programmable Radio on Chip, 56QFN	Cypress Semiconductor	CYBL10162-56LQXI	ASK	Yes	Yes
28	1	U2	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G	ASK	Yes	Yes
29	1	U3	PSoC 5LP	PSoC 5LP Programmable System on Chip, 68QFN	Cypress Semiconductor	CY8C5868LTI-LP039	ASK	Yes	Yes
30	1	Y1	32.768K Hz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B	ASK	Yes	Yes
31	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM	ASK	Yes	Yes
32	1	N/A	N/A	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60161-01 Rev 03 (YYWWVVXXXXX); Only barcode	Cypress qualified vendor	-	ASK	Yes	N/A
No load components									
33	1	C20	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T	ASK	Yes	Yes
34	1	C21	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU	ASK	Yes	Yes
35	1	C37	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA	ASK	Yes	Yes

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Sub Allowed	ROHS	UL Certified
36	1	C27	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T	ASK	Yes	Yes
37	1	R5	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1	ASK	Yes	Yes
38	2	R6, R16	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V	ASK	Yes	Yes
39	15	TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15	No load	No load	-	-	ASK	Yes	Yes

Revision History

Table 7-1. CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit Guide Revision History

Document Title: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit Guide Document Number: 001-93731			
Revision	**Issue Date**	**Origin of Change**	**Description of Change**
**	11/09/2014	ROIT	New kit guide