CY8CKIT-042-BLE-A

Bluetooth® Low Energy (BLE) Pioneer Kit Guide

Doc. # 002-11468 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

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

CySmart, F-RAM, PRoC, Programmable System-on-Chip, and PSoC Creator are trademarks of Cypress Semiconductor Corporation.

Contents

5. Hardware 87

Safety Information

The CY8CKIT-042-BLE-A Bluetooth® Low Energy (BLE) Pioneer Kit is intended for development purposes only. Users are advised to test and evaluate this kit in an RF development environment.

This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required authorizations are first obtained. Contact support@cypress.com for details.

CY8CKIT-042-BLE-A boards contain ESD-sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-042-BLE-A 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 the chassis ground (any unpainted metal surface) on the board when handling parts.

Handling Boards

CY8CKIT-042-BLE-A 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 with normal household waste. Never dispose batteries in fire because they can explode. It is important not to dispose 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-A Bluetooth® Low Energy (BLE) Pioneer Kit. The BLE Pioneer Kit enables customers to evaluate and develop BLE projects using the PSoC® 4 BLE and PRoC™ BLE (Programmable Radio-on-Chip) devices.

Bluetooth SMART™ or Bluetooth Low Energy (BLE) is a full-featured, layered, communication protocol that includes a 2.4-GHz radio, a link layer, and an application layer. However, you do not need to understand the complex protocol to implement your projects using PSoC 4 BLE or PRoC BLE. The Cypress BLE solution, which includes the device, the BLE Component, and the BLE firmware stack will take care of it for you. The Cypress BLE firmware stack is royalty free.

You will use two software tools, PSoC Creator™ and CySmart™ Central Emulation Tool, to develop and debug your BLE project. PSoC Creator is Cypress' standard integrated design environment (IDE). The BLE protocol has been abstracted into an easy drag-and-drop BLE Component in PSoC Creator. The CySmart Central Emulation Tool is a host tool for Windows PCs, which provides an easy-to-use GUI to enable customers to test and debug their BLE projects.

The BLE Pioneer Kit offers footprint-compatibility with Arduino™ shields as well as 6-pin Digilent® Pmod™ daughter cards. In addition, the kit features a CapSense® slider, an RGB LED, a push-button switch, an onboard programmer/debugger and the USB-UART/I^{2}C bridge functionality block (KitProg), a coin cell battery holder, and a Cypress F-RAM™. The BLE Pioneer Kit supports 1.9 V, 3 V, 3.3 V, or 5 V as operating voltages. The BLE Pioneer Kit supports two devices:

- PSoC 4 BLE is a 32-bit, 48-MHz ARM® Cortex®-M0 BLE solution with CapSense, 12-bit analog front end (1x SAR ADC, 4x low-power opamps, 2x low-power comparators, and 2x current DACs), 4x TCPWM[1], 2x SCBs[2], 4x UDBs[3], LCD[4], I^{2}S[5], and 36 GPIOs. PSoC 4 BLE provides a complete solution for sports and fitness monitors, wearable electronics, medical devices, home automation systems, and sensor-based low-power systems for the Internet of Things (IoT).
- PRoC BLE is a 32-bit, 48-MHz ARM Cortex-M0 BLE solution with CapSense, 12-bit ADC, 4x TCPWM[1], 2x SCBs[2], LCD[4], I^{2}S[5], and 36 GPIOs. PRoC BLE provides a complete solution for BLE connectivity, HID, remote controls, and toys.

Two different versions of the BLE Pioneer Kit are currently available. These kits differ in the BLE modules and the dongle that are included. The baseboard in both kits are identical, so any BLE module can be used interchangeably with either baseboard. Table 1-1 lists the modules and dongle included in each kit.

Table 1-1. Default BLE Modules and BLE Dongle in the Two BLE Pioneer Kits

Kit	PSoC Module	PRoC Module	Dongle	Flash Size	Bluetooth Version
CY8CKIT-042-BLE	CY8CKIT-142	CY5671	CY5670	128 KB	4.1
CY8CKIT-042-BLE-A	CY8CKIT-143A	CY5676A	CY5677	256 KB	4.1 and 4.2

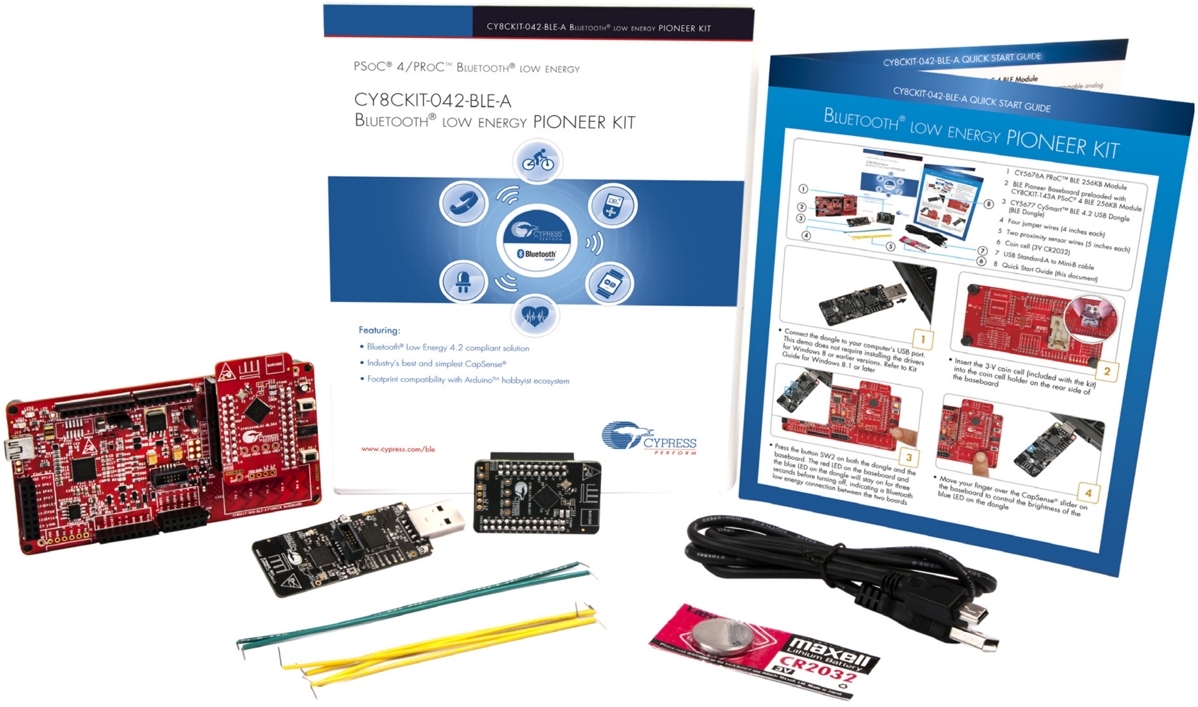
[1] Configurable timer, counter, and pulse-width modulator.
[2] Serial communication blocks (configurable to I^{2}C, SPI, or UART).
[3] Universal digital blocks
[4] Configurable liquid crystal display driver.
[5] Configurable integrated interchip sound serial bus interface.

1.1 Kit Contents

The BLE Pioneer Kit contains the following items (see Figure 1-1):

- BLE Pioneer Baseboard preloaded with the CY8CKIT-143A PSoC 4 BLE 256KB Module
- CY5676A PRoC BLE 256KB Module
- CY5677 CySmart BLE 4.2 USB Dongle (BLE 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 (3-V CR2032)

Figure 1-1. Kit Contents


CY8CKIT-042-BLE-A Bluetooth® low energy PIONEER KIT
PSoC® 4/PRoC™ Bluetooth® low energy
CY8CKIT-042-BLE-A
Bluetooth® low energy PIONEER KIT
Featuring:
• Bluetooth® Low Energy 4.2 compliant solution
• Industry's best and simplest CapSense®
• Footprint compatibility with Arduino™ hobbyist ecosystem
www.cypress.com/ble
CY8CKIT-042-BLE-A QUICK START GUIDE
Bluetooth® low energy PIONEER KIT
maxell
Lithium Battery
CR2032
3V

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

1.2 BLE Pioneer Baseboard Details

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

1. RGB LED
2. BLE device reset button
3. CapSense proximity header
4. User button
5. CapSense slider
6. Arduino-compatible I/O headers (J2/J3/J4)
7. Arduino-compatible power header (J1)
8. Digilent Pmod-compatible I/O header (J5)
9. Cypress F-RAM 1 Mb (FM24V10-G)
10. PSoC 5LP I/O header (J8)
11. PSoC 5LP programmer and debugger (CY8C5868LTI-LP039)
12. Coin cell holder (bottom side)
13. USB connector (J13)
14. Power LED and Status LED
15. System power supply jumper (J16) - LDO 1.9 V~5 V
16. BLE power supply jumper / current measurement (J15)
17. BLE module headers (J10/J11)

Figure 1-2. BLE Pioneer Baseboard

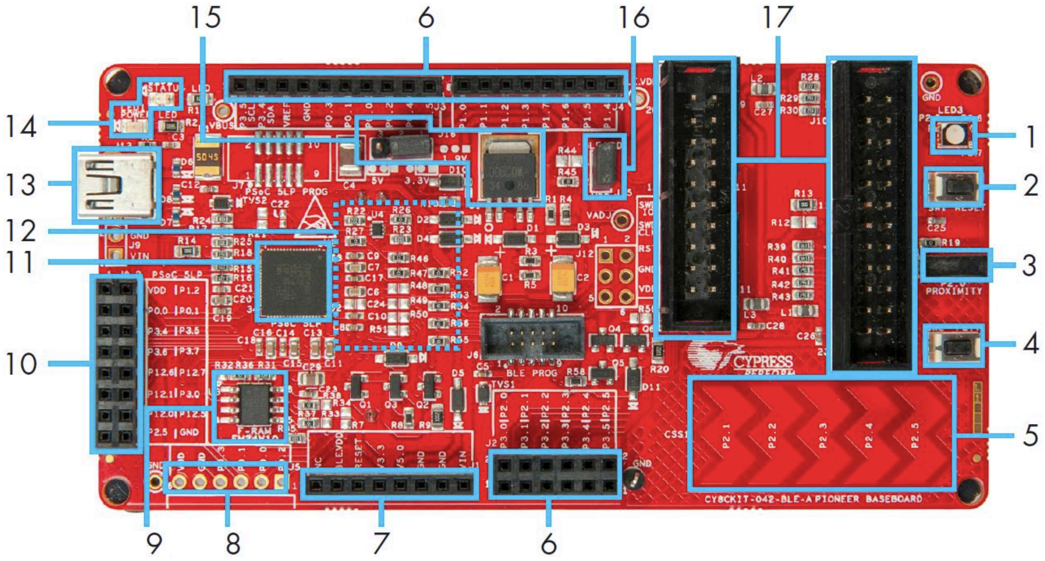


Figure 1-3 shows a markup of the onboard components of the PSoC 4 BLE module (in red) and the PRoC BLE module (in black). See BLE Modules and BLE Dongles Compatible with the BLE Pioneer Kit on page 128 for more details. Figure 1-4 shows the BLE Dongle board blocks.

Figure 1-3. BLE Module Markup

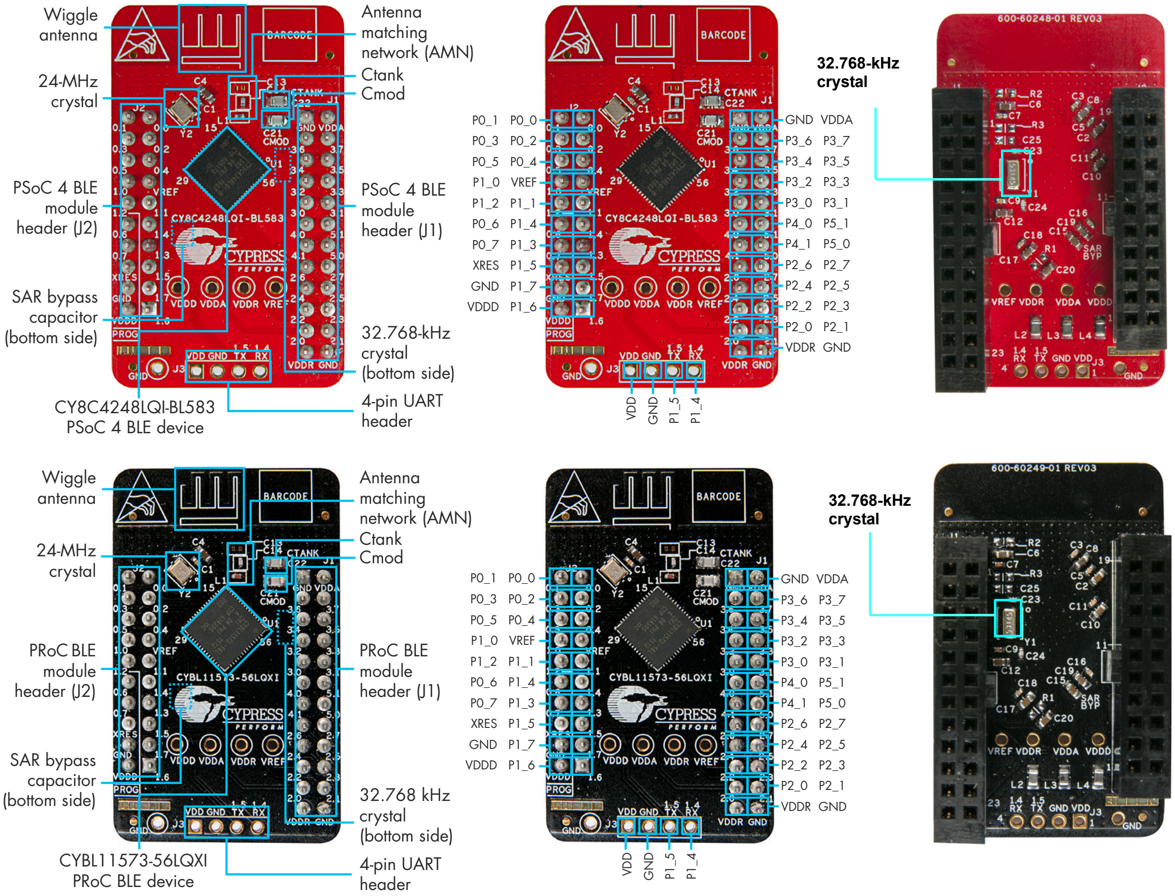
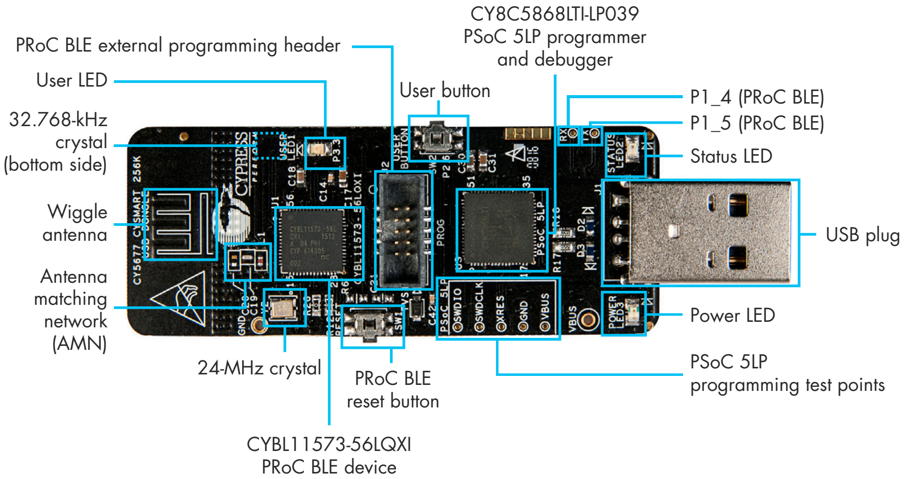


Figure 1-4. BLE Dongle Markup

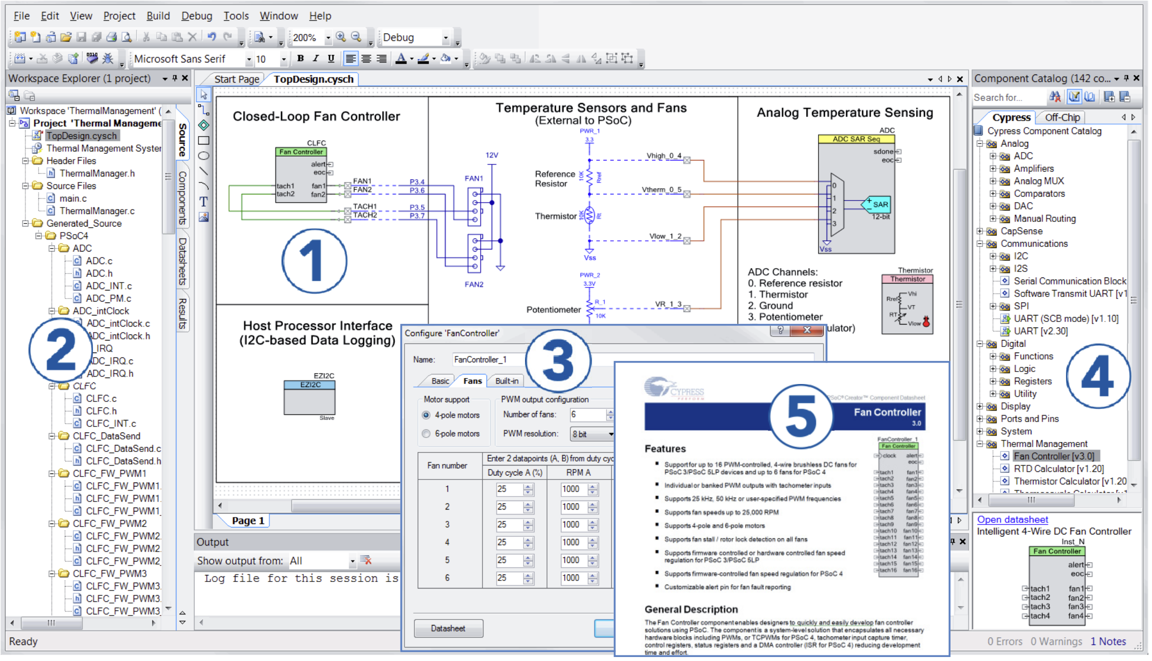

PRoC BLE external programming header
User LED
32.768-kHz crystal (bottom side)
Wiggle antenna
Antenna matching network (AMN)
24-MHz crystal
CYBL11573-56LQXI PRoC BLE device
PRoC BLE reset button
CY8C5868LTI-LP039 PSoC 5LP programmer and debugger
User button
P1_4 (PRoC BLE)
P1_5 (PRoC BLE)
Status LED
USB plug
Power LED
PSoC 5LP programming test points

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 pre-verified and pre-characterized PSoC Components. With PSoC Creator, you can:

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

Figure 1-5. PSoC Creator Features


File Edit View Project Build Debug Tools Window Help
Debug
Workspace Explorer (1 project)
Start Page
TopDesign.cysch
Component Catalog
Closed-Loop Fan Controller
Temperature Sensors and Fans (External to PSoC)
Analog Temperature Sensing
Host Processor Interface (I2C-based Data Logging)
1
2
3
4
5
Configure 'FanController'
Fan Controller
Features
General Description
Open datasheet
Intelligent 4-Wire DC Fan Controller
Output
Ready
0 Errors 0 Warnings 1 Notes

ADC Channels:
0. Reference resistor
1. Thermistor
2. Ground
3. Potentiometer

Name: FanController_1

Motor support: ● 4-pole motors ○ 6-pole motors

PWM output configuration: Number of fans: 6; PWM resolution: 8 bit

Fan number	Duty cycle A (%)	RPM A
1	25	1000
2	25	1000
3	25	1000
4	25	1000
5	25	1000
6	25	1000

Show output from: All

```
Log file for this session is
```

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

For more information, visit www.cypress.com/psoccreator. Visit PSoC Creator training page for video tutorials on learning and using PSoC Creator.

1.3.1 PSoC Creator Code Examples

PSoC Creator includes a large number of code examples. These examples are available from the PSoC Creator Start Page, as Figure 1-6 shows.

Code examples can speed up your design process by starting you off with a complete design, instead of a blank page. The code examples also show how PSoC Creator Components can be used for various applications. Code examples and documentation are included, as shown in Figure 1-7 on page 14.

In the **Find Example Project** dialog shown in Figure 1-7, you have several options:

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

Figure 1-6. Code Examples in PSoC Creator

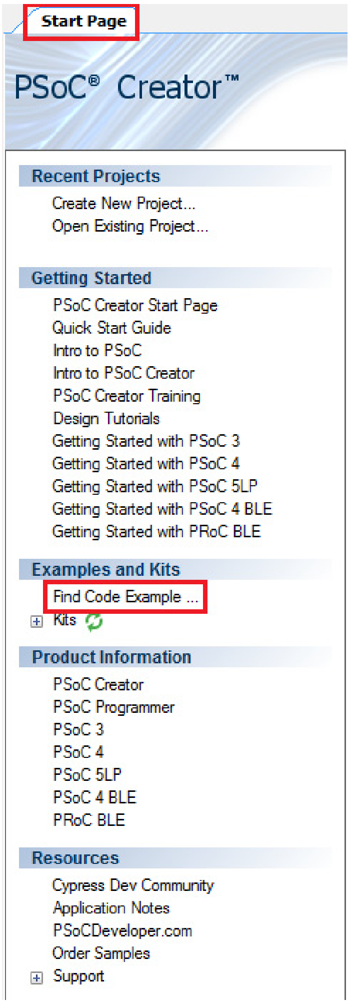
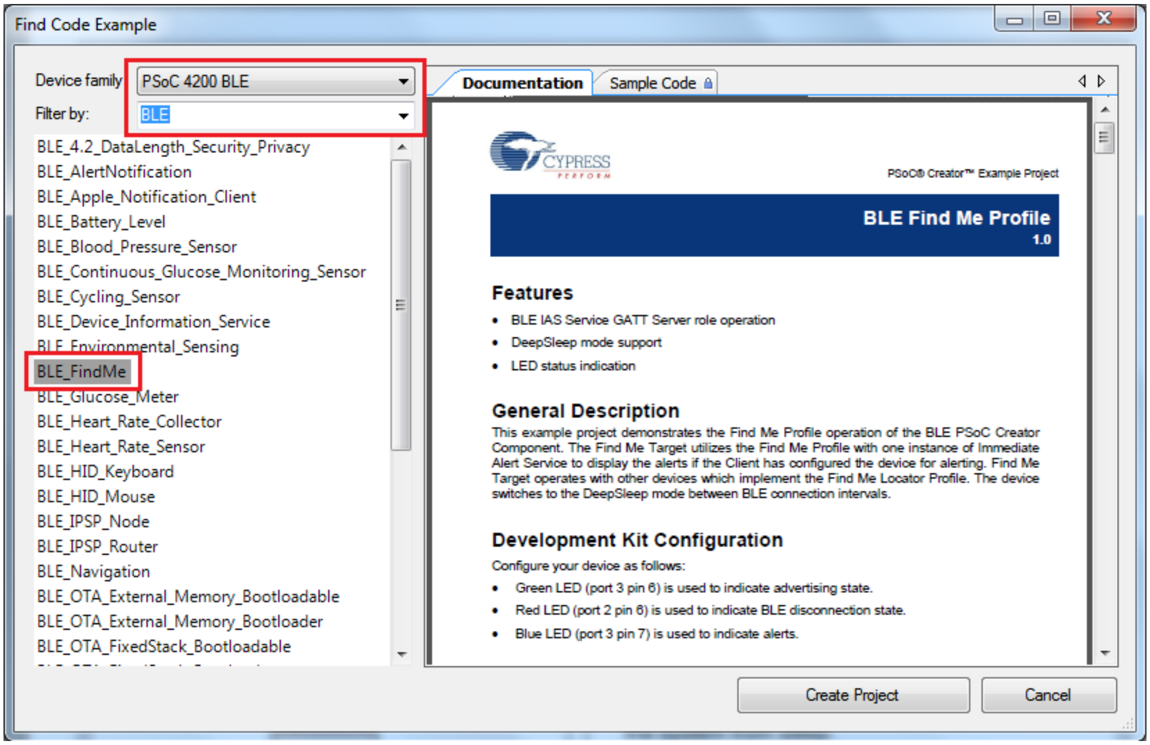


Figure 1-7. Code Example Projects with Sample Code


Find Code Example
Documentation
Sample Code
BLE_4.2_DataLength_Security_Privacy
BLE_AlertNotification
BLE_Apple_Notification_Client
BLE_Battery_Level
BLE_Blood_Pressure_Sensor
BLE_Continuous_Glucose_Monitoring_Sensor
BLE_Cycling_Sensor
BLE_Device_Information_Service
BLE_Environmental_Sensing
BLE_FindMe
BLE_Glucose_Meter
BLE_Heart_Rate_Collector
BLE_Heart_Rate_Sensor
BLE_HID_Keyboard
BLE_HID_Mouse
BLE_IPSP_Node
BLE_IPSP_Router
BLE_Navigation
BLE_OTA_External_Memory_Bootloadable
BLE_OTA_External_Memory_Bootloader
BLE_OTA_FixedStack_Bootloadable
CYPRESS PERFORM
Create Project
Cancel

Device family: PSoC 4200 BLE

Filter by: BLE

PSoC® Creator™ Example Project

BLE Find Me Profile
1.0

Features

- BLE IAS Service GATT Server role operation
- DeepSleep mode support
- LED status indication

General Description

This example project demonstrates the Find Me Profile operation of the BLE PSoC Creator Component. The Find Me Target utilizes the Find Me Profile with one instance of Immediate Alert Service to display the alerts if the Client has configured the device for alerting. Find Me Target operates with other devices which implement the Find Me Locator Profile. The device switches to the DeepSleep mode between BLE connection intervals.

Development Kit Configuration

Configure your device as follows:

- Green LED (port 3 pin 6) is used to indicate advertising state.
- Red LED (port 2 pin 6) is used to indicate BLE disconnection state.
- Blue LED (port 3 pin 7) is used to indicate alerts.

1.3.2 Kit Code Examples

This kit includes a number of code examples, which can be used to quickly evaluate the functionality of this kit. These examples are described in the Example Projects chapter on page 35.

If you are a beginner in BLE, refer to the PSoC Creator code examples such as BLE_FindMe and BLE_Device_Information_Service as described in 1.3.1 PSoC Creator Code Examples. You may also refer the application note Getting Started with PSoC 4 BLE.

The four Kit code examples viz CapSense Slider and LED, CapSense Proximity, BLE Central Mode and Eddystone are intermediatory level examples that will help to design a system around the kit. Refer to the 4.2 CapSense Slider and LED and later for details.

The CySmart Dongle code example is an advanced level example that will demonstrate a complete solution around the kit. Refer to 4.6 BLE Dongle and LED Control for details.

1.3.3 PSoC Creator Help

Visit the PSoC Creator home page to download the latest version of PSoC Creator. Then, launch PSoC Creator and navigate to the following items:

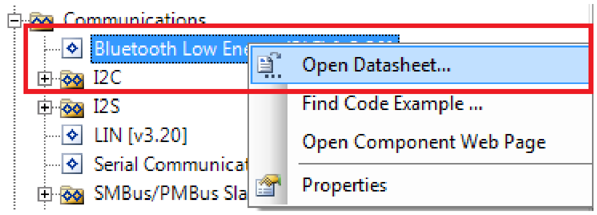
- **Quick Start Guide**: Choose **Help** > **Documentation** > **Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple Component example projects**: Choose **File** > **Code Example**. These example projects demonstrate how to configure and use PSoC Creator Components.
- **System Reference Guide**: Choose **Help** > **System Reference Guide**. This guide lists and describes the system functions provided by PSoC Creator.

- **Component datasheets**: Right-click a Component and select **Open Datasheet**, as shown in Figure 1-8 on page 15. Visit the PSoC 4 Component Datasheets page for a list of all PSoC 4 Component datasheets.
- **Document Manager**: PSoC Creator provides a document manager to help you to easily find and review document resources. To open the document manager, choose the menu item **Help** > **Document Manager**.

1.3.4 Component Datasheets

Right-click a Component and select **Open Datasheet** (see Figure 1-8 on page 15).

Figure 1-8. Opening Component Datasheet



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 Central Emulation Tool for BLE Central emulation.
- The Kit Operation chapter on page 23 describes the major features of the BLE Pioneer Kit such as USB-UART and USB-I^{2}C bridges and functionalities such as programming and debugging.
- The Example Projects chapter on page 35 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 87 details the hardware content of the BLE Pioneer Kit and BLE Dongle, and the hardware operation.
- The Advanced Topics chapter on page 114 explains the functionality of FM24V10 F-RAM.
- The Appendix on page 116 provides the bill of materials (BOM), KitProg LED states, list of BLE modules and dongles compatible with the BLE Pioneer Kit, migrating projects across Pioneer series kits, and programming BLE modules using MiniProg3.

1.5 Additional Learning Resources

Cypress provides a wealth of information at www.cypress.com to help you to select the right PSoC device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see KBA86521, How to Design with PSoC 3, PSoC 4, and PSoC 5LP.

Visit www.cypress.com/go/psoc4ble and www.cypress.com/procble for additional learning resources including datasheets, technical reference manuals, and application notes.The following is an abbreviated list:

- Overview: PSoC 4 BLE Portfolio and PSoC Roadmap
- Product Selectors: PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Creator includes a device selection tool.

- Datasheets: Describe and provide electrical specifications for the PSoC 4 device family
- CapSense Design Guide: Learn how to design capacitive touch-sensing applications with the PSoC 4 family of devices.
- Application Notes and Code Examples: Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples. Visit the PSoC 3/4/5 Code Examples webpage for a list of all available PSoC Creator code examples. To access code examples from within PSoC Creator, see PSoC Creator Code Examples on page 13.
- Technical Reference Manuals (TRM): Provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- Development Kits:
    - CY8CKIT-040, CY8CKIT-042, and CY8CKIT-044 are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino-compatible shields and Digilent Pmod peripheral modules.
    - CY8CKIT-141, CY8CKIT-142, CY8CKIT-143, CY5671, CY5674, and CY5676 are additional BLE Modules compatible with the BLE Pioneer Kit.
    - CY5670 is additional BLE Dongle compatible with the BLE Pioneer Kit.
    - CY8CKIT-049 is a very low-cost prototyping platform for PSoC 4100/4200 families.
    - The MiniProg3 kit provides an interface for flash programming and debug.
    - CY5672 PRoC BLE Remote Control Reference Design Kit provides a production-ready implementation of a Bluetooth Low Energy remote control.
    - CY5682 PRoC BLE Touch Mouse Reference Design Kit provides a production-ready implementation of a Bluetooth Low Energy touch mouse.
- Knowledge Base Articles (KBA): Provide design and application tips from experts on using the device.
- PSoC Creator Training: Visit www.cypress.com/go/creatorstart/creatortraining for a comprehensive list of video trainings on PSoC Creator.
- Cysmart Central Emulation Tool: Visit www.cypress.com/go/cysmart for information on the CySmart Central Emulation Tool.
- Learning From Peers: Visit www.cypress.com/forums to meet enthusiastic PSoC developers discussing the next generation embedded systems on Cypress Developer Community Forums.

1.5.1 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

1.5.2 Other Related Resources

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

1.6 Technical Support

If you have any questions, our technical support team is happy to assist you. You can create a support request on the Cypress Technical Support page.

If you are in the United States, you can talk to our technical support team by calling our toll-free number: +1-800-541-4736. Select option 2 at the prompt.

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

- Self-help
- Local Sales Office Locations

1.7 Documentation Conventions

Table 1-2. Document Conventions for Guides

Convention	Usage
`Courier New`	Displays file locations, user entered text, and source code: `C:\...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC 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 1-3. Acronyms Used in this Document

Acronym	Definition
ADC	analog-to-digital converter
API	application programming interface
BD address	Bluetooth device address
BLE	Bluetooth Low Energy
CDC	Communications Device Class
COM	communication port
DLE	data length extension
ECDH	Elliptic Curve Diffie-Hellman
DVD	digital video disc
ESD	electrostatic discharge
F-RAM	ferroelectric random access memory
GATT	generic attribute profile

Table 1-3. Acronyms Used in this Document

Acronym	Definition
GUI	graphical user interface
GPIO	general-purpose input/output
I^2C	inter-integrated circuit
IAS	immediate alert service
IDAC	current output digital-to-analog converter
IDE	integrated development environment
LDO	low drop out (voltage regulator)
LE	low energy
LED	light-emitting diode
LP	low power
LPT	line print terminal
MTU	maximum transmission unit
OTA	over-the-air
PHY	physical layer
PrISM	Precise Illumination Signal Modulation
PRoC	Programmable Radio-on-Chip
PSM	protocol service multiplexer
PSoC	Programmable System-on-Chip
PWM	pulse width modulation
QFN	quad flat no-lead (package)
RDK	reference design kit
RGB	red green blue
SAR	successive approximation register
SIG	special interest group
SMA	SubMiniature version A
SPI	serial peripheral interface
SWD	serial wire debug
TLM	telemetry
UART	universal asynchronous receiver transmitter
UID	universal identifier
URI	uniform resource identifier
URL	uniform resource locator
USB	Universal Serial Bus
UUID	universal unique identifier

2. Software Installation

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

2.1 Before You Begin

All Cypress software installations require administrator privileges. Ensure you have the required privileges on the system for successful installation. Before you install the kit software, close any other Cypress software that is currently running.

2.2 Install Software

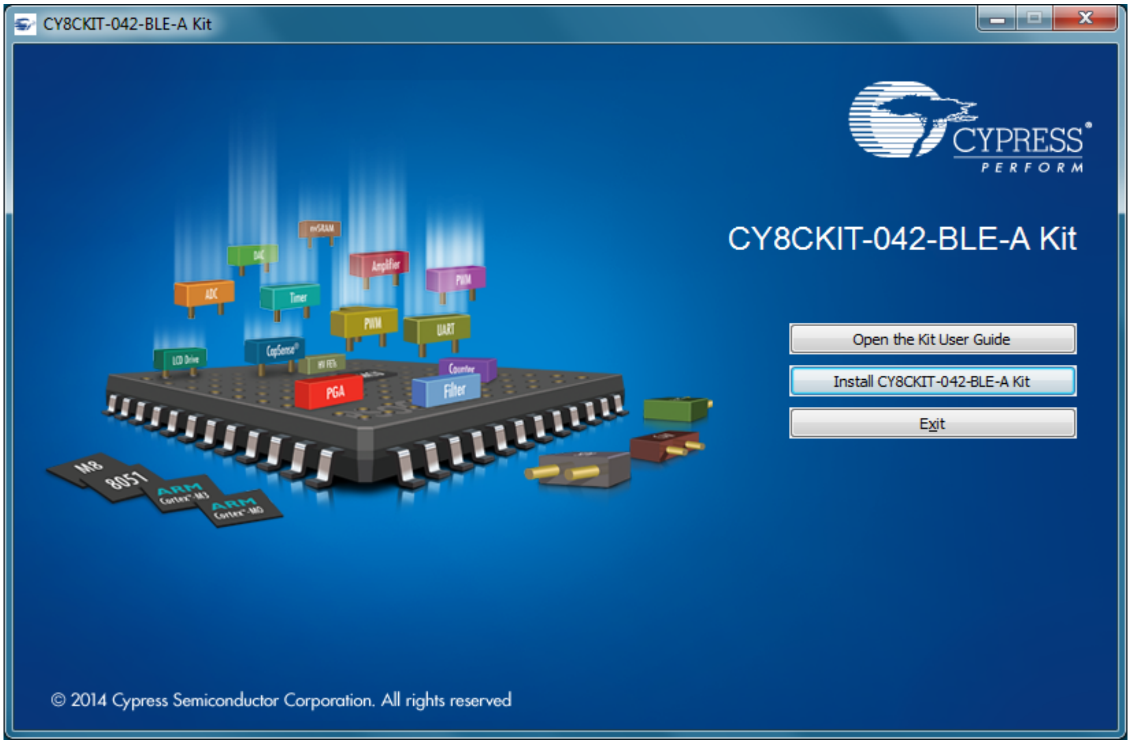
Follow these steps to install the BLE Pioneer Kit software:

1. Download the kit software from www.cypress.com/CY8CKIT-042-BLE-A. The software is available in the following formats:
   a. CY8CKIT-042-BLE-A Kit Complete Setup: This installation package contains the files related to the BLE Pioneer 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-A Kit Only Package: This executable file installs only the BLE Pioneer Kit contents, which include 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 computer.
   c. CY8CKIT-042-BLE-A 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 using the appropriate ISO extractor (such as MagicISO or PowerISO). 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 computer. The installation window will appear automatically.

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

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

Figure 2-1. Installer Screen

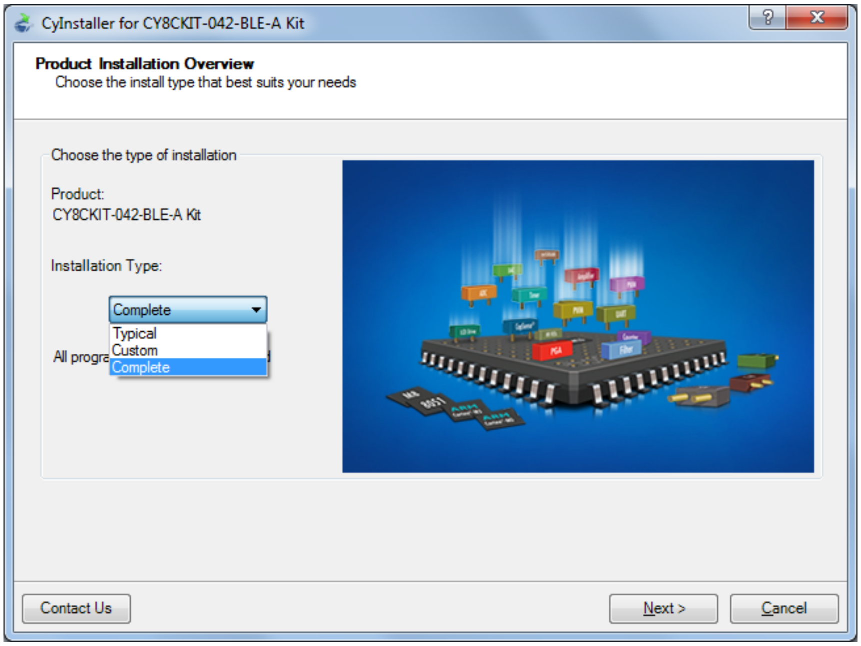


© 2014 Cypress Semiconductor Corporation. All rights reserved

4. Select the folder in which you want to install the CY8CKIT-042-BLE-A kit-related files. Choose the directory and click **Next**.
5. When you click **Next**, the CY8CKIT-042-BLE-A Kit installer automatically installs the required software, if it is not present on your computer. The following software packages are required:
   Note: For the Kit Only Package, download and install the following prerequisites.
   a. PSoC Creator 3.3 or later: Download the latest version from www.cypress.com/psoccreator.
   b. PSoC Programmer 3.24 or later: This is installed as part of PSoC Creator installation (www.cypress.com/programmer).
   c. CySmart 1.2 or later: Download the latest version from www.cypress.com/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



CyInstaller for CY8CKIT-042-BLE-A Kit

Product Installation Overview

Choose the install type that best suits your needs

Choose the type of installation

Product:
CY8CKIT-042-BLE-A Kit

Installation Type: Complete

All progra

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-A kit installation.
10. Enter your contact information or select the **Continue Without Contact Information** check box. Click **Finish** to complete the CY8CKIT-042-BLE-A kit installation.
11. After the installation is complete, the kit contents are available at the following location:

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

Default location:

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

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

Note: For Windows 7/8/8.1 users, the installed files and the folder are read only. To use the installer example projects, follow the steps outlined in the Example Projects chapter on page 35.

The BLE Pioneer Kit installer also installs the CySmart Central Emulation Tool on your computer. This software, along with the BLE Dongle, allows the computer to emulate as a BLE Central device.

2.3 Windows 8.1 and Windows 10 USB Selective Suspend Setting

The Windows 8.1 and Windows 10 operating systems issue a suspend command to a connected USB device if there is no activity on its USB bus. This may happen to the BLE Pioneer Kit/BLE Dongle if the driver installation is not complete or the driver is not found. In this situation, as a response to the suspend command from the operating system, the KitProg will shut down the onboard LDO to remain within the USB suspend current requirement. Due to this, the BLE Pioneer Kit will not be powered and will not be functional.

To prevent this, do either of the following:

- Install the kit software on the Windows 8.1 or Windows 10 operating system before using the BLE Pioneer Kit.
- Disable the USB selective suspend setting for the BLE Pioneer Kit from Power Options on Windows 8.1, by following these steps:
  a. From **Control Panel**, go to **Power Options**.
  b. Click **Change Plan Settings** corresponding to your power plan.
  c. Click **Change Advanced Power Settings**.
  d. Under Advanced settings, expand **USB Settings** > **USB selective suspend setting**.
  e. Select the **Disabled** option for **Plugged in** and **On battery** operation.
  f. Click **OK**.

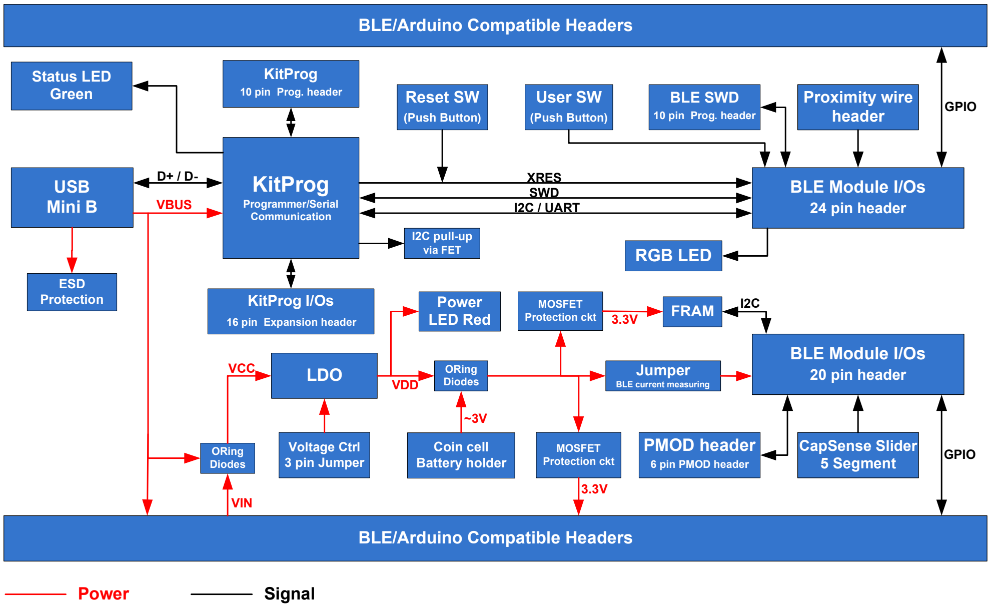
3. Kit Operation

This chapter introduces you to the BLE Pioneer Kit and the features that will be used as part of its operation. We will discuss features such as 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 BLE Pioneer 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, PSoC 4 BLE/PRoC BLE Module, and BLE Dongle.

Figure 3-1. BLE Pioneer Baseboard Block Diagram

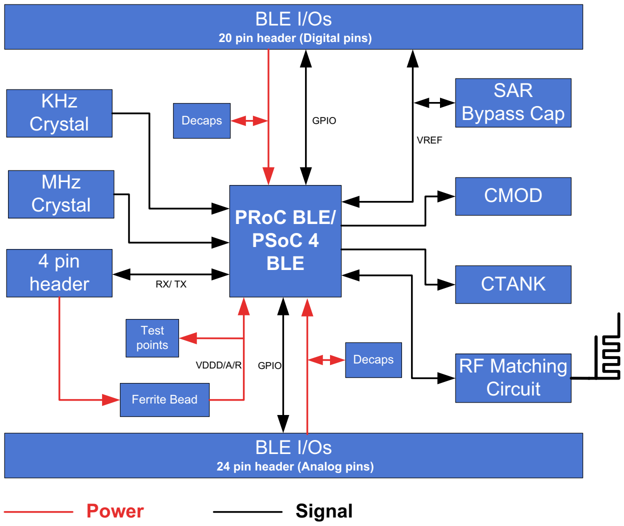

BLE/Arduino Compatible Headers
Status LED Green
KitProg 10 pin Prog. header
Reset SW (Push Button)
User SW (Push Button)
BLE SWD 10 pin Prog. header
Proximity wire header
GPIO
USB Mini B
D+ / D-
VBUS
KitProg Programmer/Serial Communication
XRES
SWD
I2C / UART
BLE Module I/Os 24 pin header
I2C pull-up via FET
RGB LED
ESD Protection
KitProg I/Os 16 pin Expansion header
Power LED Red
MOSFET Protection ckt
3.3V
FRAM
I2C
VCC
LDO
VDD
ORing Diodes
Jumper BLE current measuring
BLE Module I/Os 20 pin header
~3V
ORing Diodes
Voltage Ctrl 3 pin Jumper
Coin cell Battery holder
MOSFET Protection ckt
PMOD header 6 pin PMOD header
CapSense Slider 5 Segment
GPIO
VIN
3.3V
BLE/Arduino Compatible Headers
Power
Signal

The BLE Pioneer Baseboard acts as the baseboard for the PSoC 4 BLE (red module) and PRoC BLE (black module). The BLE Pioneer Baseboard contains a PSoC 5LP device, that has KitProg firmware, 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 BLE Pioneer Baseboard can also be used as a standalone programmer to program and debug other BLE devices using SWD, and as a USB-Serial interface. The KitProg firmware on PSoC 5LP device enables bootloading PSoC 5LP over USB to upgrade the firmware.

Figure 3-2. PSoC 4 BLE/PRoC BLE Module Block Diagram

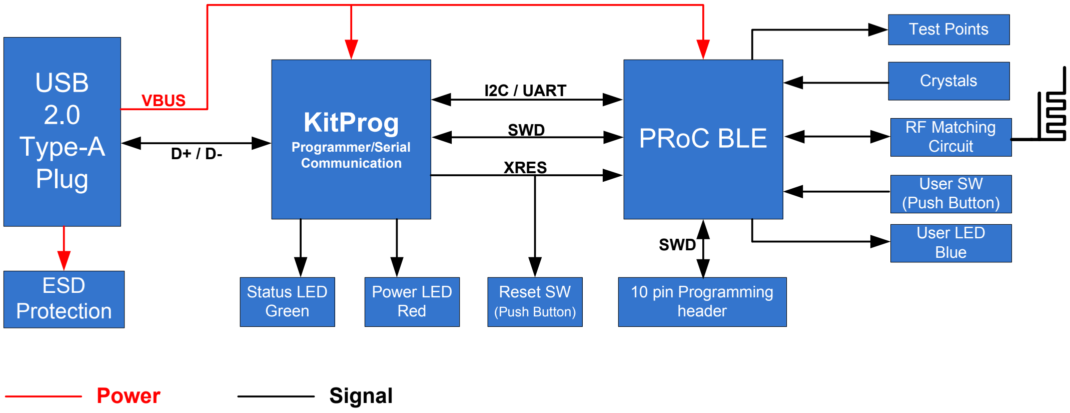


This BLE Pioneer Kit includes two modules. These modules act as a basic breakout board for the CY8C4248LQI-BL583 (PSoC 4 BLE) and CYBL11573-56LQXI (PRoC BLE) device. The PSoC 4 BLE and PRoC BLE Modules are identical except for the BLE device. Besides these two modules, there are additional modules available, which can be ordered separately. The complete list is available in BLE Modules and BLE Dongles Compatible with the BLE Pioneer Kit on page 128.

The BLE Dongle is the wireless interface for the CySmart Central Emulation Tool. It has a PRoC BLE device for BLE communication and KitProg for onboard programming, debugging, and for the USB-Serial interface, as shown in Figure 3-3.

The BLE Dongle has a USB Type-A plug to connect the KitProg to the USB port of the host computer. The KitProg then communicates with the PRoC BLE device over UART or multiplexed I^{2}C or an SPI bus. The BLE Dongle 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.

Figure 3-3. BLE Dongle Block Diagram



3.2 KitProg

KitProg is the hardware/firmware block for onboard programming, debugging, and bridge functionality. It is a common reusable hardware/firmware block used across many Cypress kit platforms. It consists of a PSoC 5LP, which connects to the computer over an USB interface and connects to the PSoC 4 BLE or PRoC BLE device over SWD, I^2C, and UART pins.

The KitProg communicates with PSoC Programmer and PSoC Creator software to program/debug the target PSoC 4 BLE or PRoC BLE over the SWD interface. The main advantage of an onboard programmer/debugger is that users do not have to buy an extra programmer/debugger hardware.

3.3 BLE Pioneer Kit USB Connection

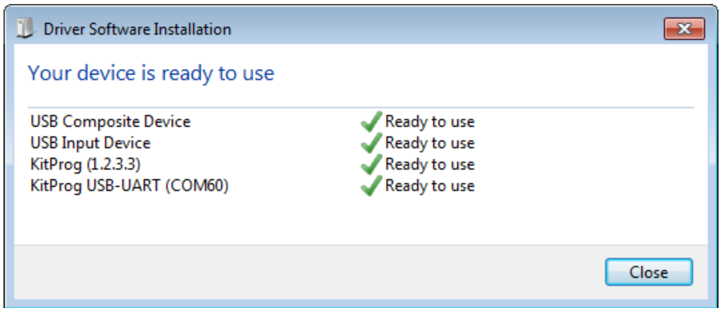
The BLE Pioneer Kit powers from a computer over the USB interface (J13). It enumerates as a composite device, as shown in Table 3-1. USB drivers required for this enumeration are part of the kit installer. The kit should be installed properly for its correct operation.

Visit www.cypress.com/CY8CKIT-042-BLE for the latest kit installer.

Table 3-1. BLE Pioneer Kit Enumerated Interfaces

Port	Description
USB Composite Device	Composite device
USB Input Device	USB-I^2C bridge, KitProg command interface
KitProg	USB-I^2C bridge, programmer, and debugger
KitProg USB-UART	USB-UART bridge, which appears as a COM# port

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



Driver Software Installation

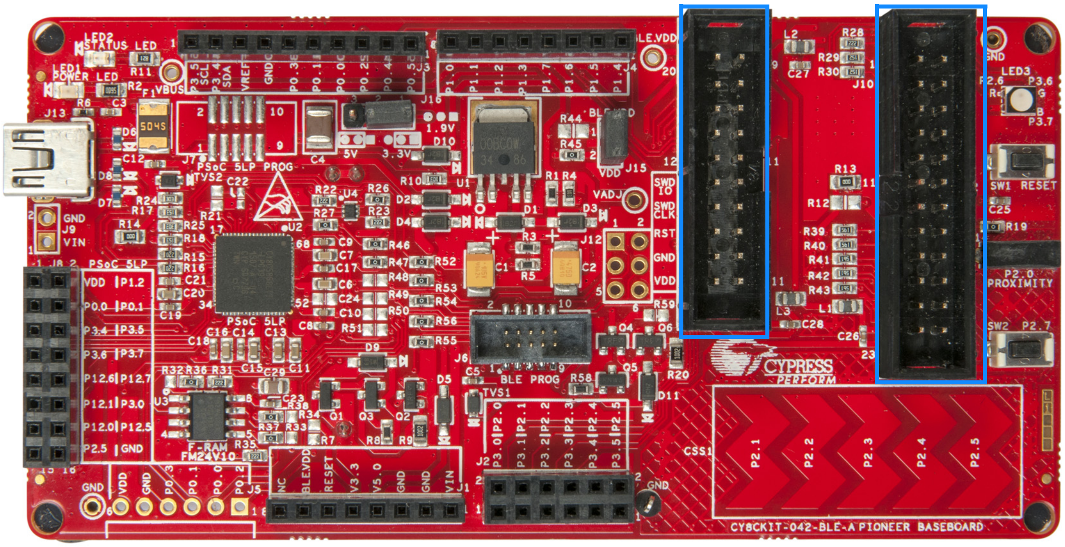
Your device is ready to use

USB Composite Device	✓ Ready to use
USB Input Device	✓ Ready to use
KitProg (1.2.3.3)	✓ Ready to use
KitProg USB-UART (COM60)	✓ Ready to use

3.4 Placing Modules on Baseboard

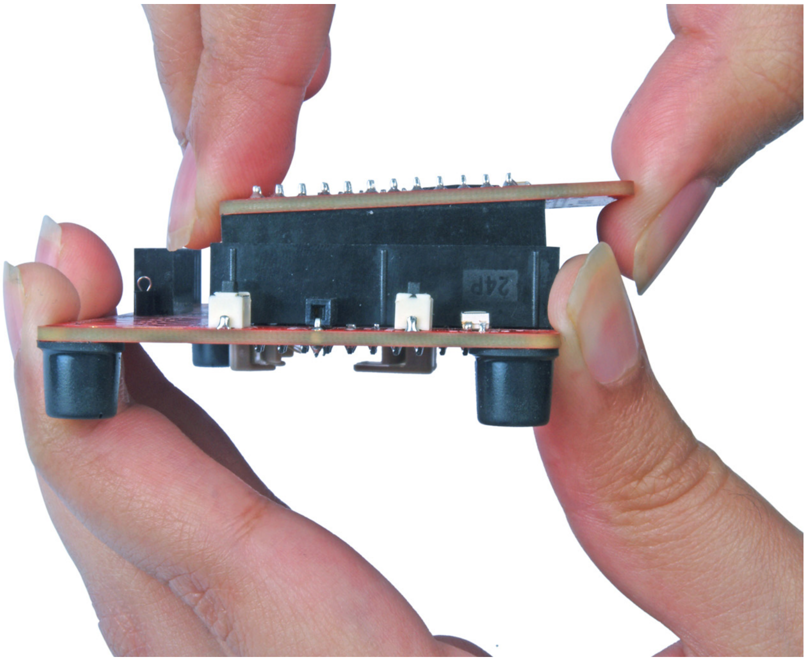
Plug the module into the BLE Pioneer Baseboard on headers J10 and J11, while keeping the antenna directed outside. Note that the two parallel headers J10 and J11 are not equal (24-pin and 20-pin, respectively) and will not allow the module to be inserted in the opposite direction.

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



To remove the modules from the BLE Pioneer Kit, hold the BLE Pioneer Kit in one hand and the module in the other, as shown in Figure 3-6, and pull it out using a rocking motion.

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



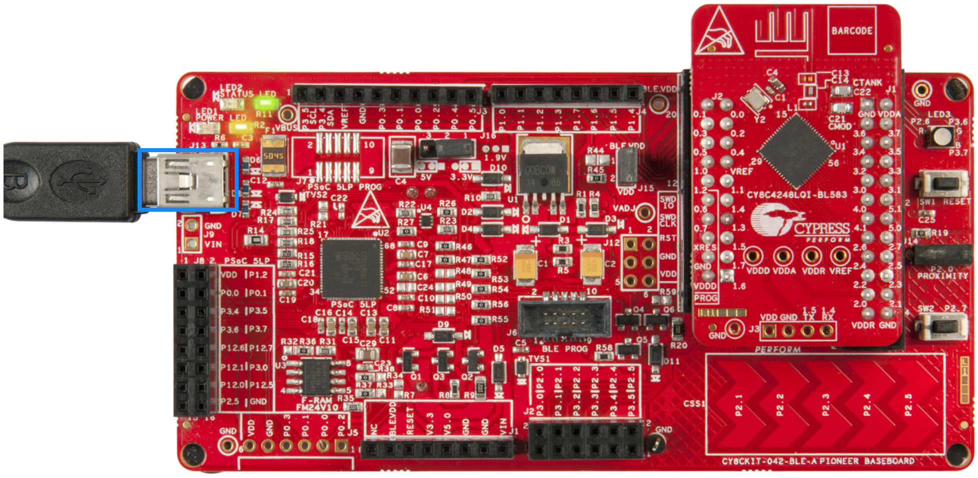
3.5 Programming and Debugging BLE Device

The BLE Pioneer Kit and BLE Dongle can be programmed and debugged using the KitProg. Before programming the device, ensure that PSoC Creator and PSoC Programmer are installed on the computer. See Install Software on page 19 for more information.

3.5.1 Programming using PSoC Creator

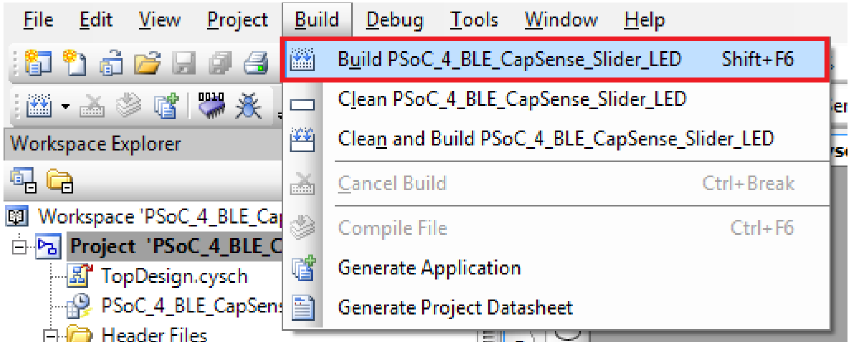
1. Connect the BLE Pioneer Kit/BLE Dongle to the computer's USB port, as shown in Figure 3-7.

Figure 3-7. Connect USB Cable to J13



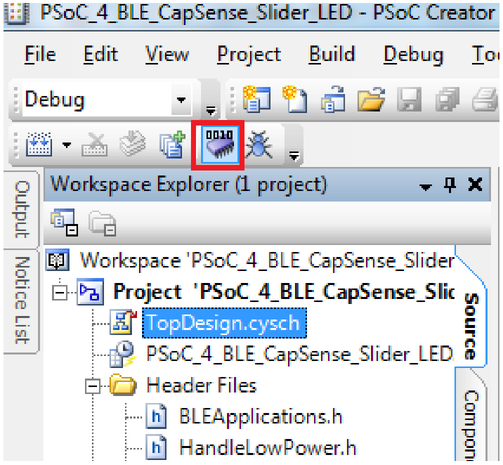
2. Load the desired example project in PSoC Creator from **File** > **Open** > **Project/Workspace**.
3. Build the project by choosing **Build** > **Build <Project Name>** or **[Shift] [F6]**, as shown in Figure 3-8.

Figure 3-8. Build an Example Project



4. If there are no errors during build, program the firmware by clicking the **Program** button on the tool bar or pressing **[Ctrl] [F5]**, as shown in Figure 3-9. This will program the device on the BLE Pioneer Kit/BLE Dongle and it will be ready for use.

Figure 3-9. Programming Device From PSoC Creator

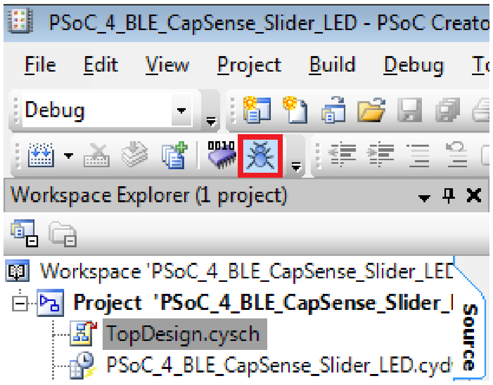

PSoC_4_BLE_CapSense_Slider_LED - PSoC Creator
File Edit View Project Build Debug To
Debug
Output
Notice List
Workspace Explorer (1 project)
Workspace 'PSoC_4_BLE_CapSense_Slider
Project 'PSoC_4_BLE_CapSense_Slic
TopDesign.cysch
PSoC_4_BLE_CapSense_Slider_LED
Header Files
BLEApplications.h
HandleLowPower.h
Source
Compon

3.5.2 Debugging using PSoC Creator

For debugging the project using PSoC Creator, follow steps 1 to 5 from Programming using PSoC Creator on page 27 followed by:

1. Click the **Debug** icon or press **[F5]**, as shown in Figure 3-10.

Figure 3-10. Start Debug on PSoC Creator


PSoC_4_BLE_CapSense_Slider_LED - PSoC Creato
File Edit View Project Build Debug To
Debug
Workspace Explorer (1 project)
Workspace 'PSoC_4_BLE_CapSense_Slider_LED
Project 'PSoC_4_BLE_CapSense_Slider_l
TopDesign.cysch
PSoC_4_BLE_CapSense_Slider_LED.cyd
Source

2. When PSoC Creator opens in debug mode, use the buttons on the toolbar for debugging.

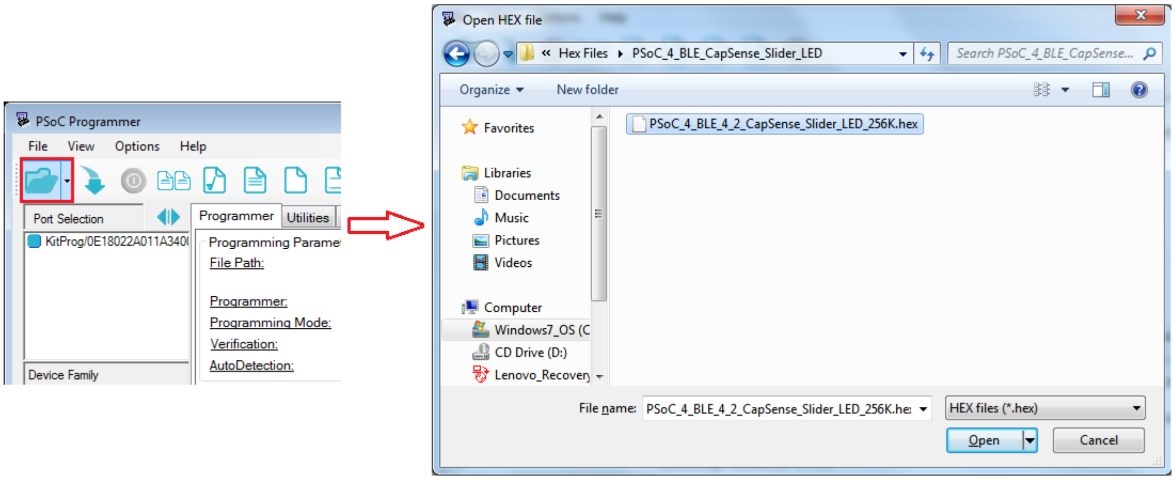
For more details on using the debug features, see the Cypress application note Getting Started with PSoC 4 BLE.

3.5.3 Programming using PSoC Programmer

PSoC Programmer (3.24 or later) can be used to program existing hex files into both BLE Pioneer Kit or BLE Dongle. To do this, follow these steps.

1. Connect the BLE Pioneer Kit or BLE Dongle to a computer and open PSoC Programmer from **Start** > **All Programs** > **Cypress** > **PSoC Programmer <version>** > **PSoC Programmer <version>**.
2. Click the **File Load** button at the top left corner of the window. Browse for the desired hex file and click **Open**.

Figure 3-11. Select Hex File

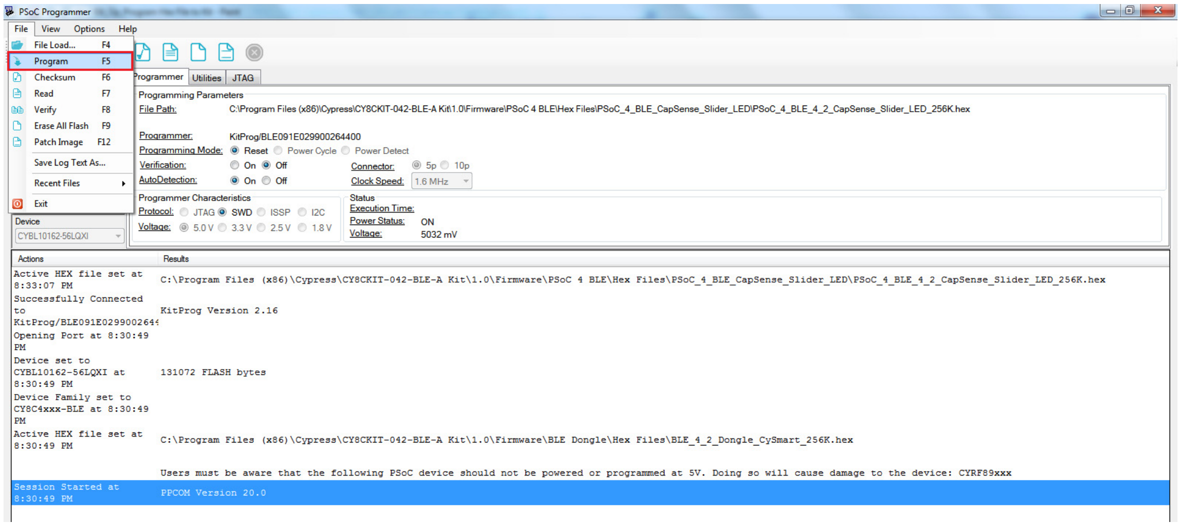


File name: PSoC_4_BLE_4_2_CapSense_Slider_LED_256K.he

3. Go to **File** > **Program** to start programing the kit with the selected file.

   Note: If the hex file does not match the device selected, then PSoC Programmer will throw an error of device mismatch and terminate programming.

Figure 3-12. Program Hex File to Kit



CYBL10162-56LQXI

Programming Parameters

File Path: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE-A Kit\1.0\Firmware\PSoC 4 BLE\Hex Files\PSoC_4_BLE_CapSense_Slider_LED\PSoC_4_BLE_4_2_CapSense_Slider_LED_256K.hex

Programmer: KitProg/BLE091E029900264400

Programming Mode: ● Reset ○ Power Cycle ○ Power Detect

Verification: ○ On ● Off

Connector: ● 5p ○ 10p

AutoDetection: ● On ○ Off

Clock Speed: 1.6 MHz

Programmer Characteristics

Protocol: ○ JTAG ● SWD ○ ISSP ○ I2C

Voltage: ● 5.0 V ○ 3.3 V ○ 2.5 V ○ 1.8 V

Status

Execution Time:

Power Status: ON

Voltage: 5032 mV

```
Actions                      Results
Active HEX file set at       C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE-A Kit\1.0\Firmware\PSoC 4 BLE\Hex Files\PSoC_4_BLE_CapSense_Slider_LED\PSoC_4_BLE_4_2_CapSense_Slider_LED_256K.hex
8:33:07 PM
Successfully Connected
to                           KitProg Version 2.16
KitProg/BLE091E0299002644
Opening Port at 8:30:49
PM
Device set to
CYBL10162-56LQXI at          131072 FLASH bytes
8:30:49 PM
Device Family set to
CY8C4xxx-BLE at 8:30:49
PM
Active HEX file set at       C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE-A Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_4_2_Dongle_CySmart_256K.hex
8:30:49 PM

                             Users must be aware that the following PSoC device should not be powered or programmed at 5V. Doing so will cause damage to the device: CYRF89xxx
Session Started at           PPCOM Version 20.0
8:30:49 PM
```

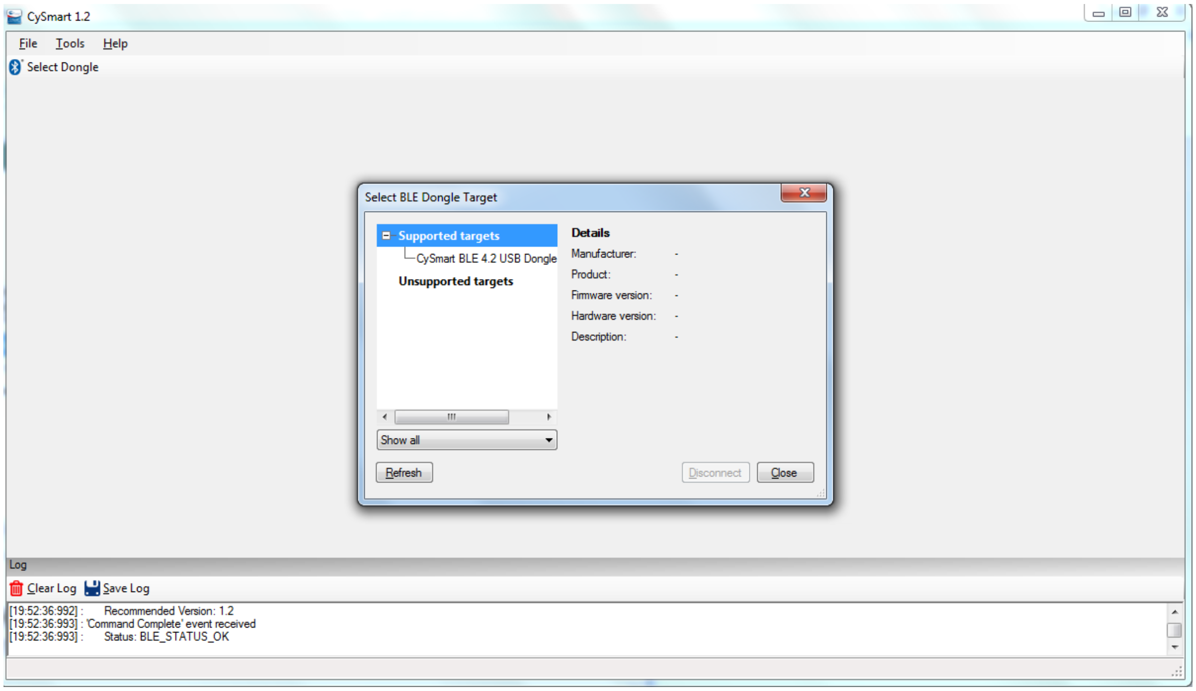
4. When the programming is finished successfully, indicated by a PASS message on the status bar, the BLE Pioneer Kit/BLE Dongle is ready for use. Close PSoC Programmer.

3.6 Updating BLE Dongle for CySmart Central Emulation Tool

The BLE Dongle, provides a BLE Central mode capability using the CySmart Central Emulation Tool on the computer. The CySmart Central Emulation Tool on the PC is the interface with which to configure the BLE Dongle and analyze the data transferred after connecting with a BLE Peripheral.

The BLE Dongle works along with the CySmart Central Emulation Tool, as shown in Figure 3-13. The CySmart Central Emulation Tool is installed as part of the BLE Pioneer Kit installation and can be opened from **Start** > **All Programs** > **Cypress** > **CySmart <version>** > **CySmart <version>**. The tool operation is explained in the user guide, which can be accessed from **Help** > **Help Topics**.

Figure 3-13. BLE Dongle Interface on CySmart Central Emulation Tool



Manufacturer: -
Product: -
Firmware version: -
Hardware version: -
Description: -

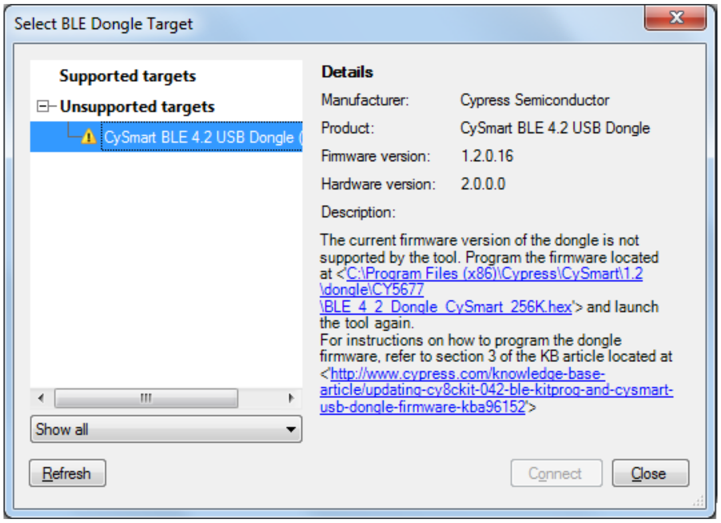
```
[19:52:36:992] :       Recommended Version: 1.2
[19:52:36:993] : 'Command Complete' event received
[19:52:36:993] :       Status: BLE_STATUS_OK
```

If the BLE Dongle contains custom firmware on PRoC BLE, the original CySmart firmware can be programmed back to restore the CySmart functionality. It must be connected through the USB and enumerated as KitProg. To do this, follow these steps:

1. Connect the BLE Dongle to the USB port on the computer.
2. Open PSoC Programmer by going to **Start** > **All Programs** > **Cypress** > **PSoC Programmer <version>** > **PSoC Programmer <version>**.
3. Click the **File Load** button and browse to the location of the *BLE_4_2_Dongle_CySmart_256K.hex* file. The hex file is located at:
   `<Install_Directory>\CY8CKIT-042-BLE-A Kit\<version>\Firmware\BLE Dongle\Hex Files\`

   Note: If Cypress releases new versions of the CySmart Central Emulation Tool and the BLE Dongle firmware, then the CySmart Central Emulation Tool will display a message requesting to update the firmware, as shown in the following figures.

Figure 3-14. Update BLE Dongle Firmware with Hex from Latest Kit Installer



Details

Manufacturer: Cypress Semiconductor

Product: CySmart BLE 4.2 USB Dongle

Firmware version: 1.2.0.16

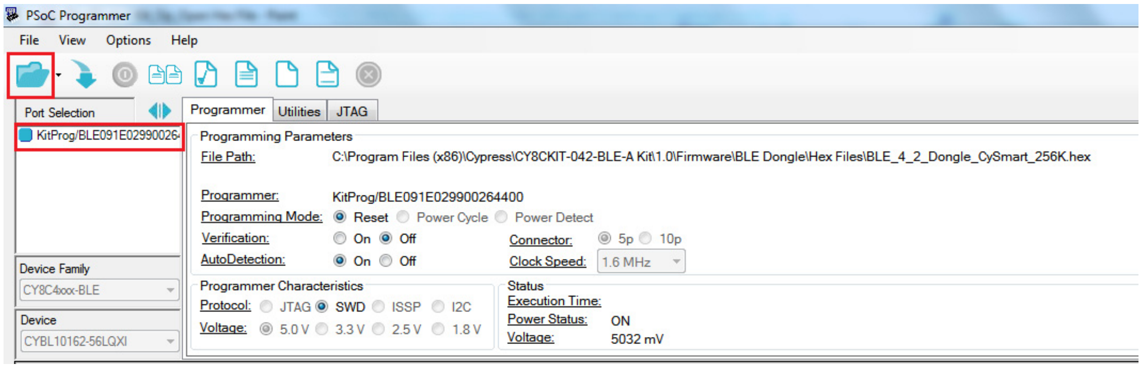
Hardware version: 2.0.0.0

Description:

The current firmware version of the dongle is not supported by the tool. Program the firmware located at <C:\Program Files (x86)\Cypress\CySmart\1.2\dongle\CY5677\BLE_4_2_Dongle_CySmart_256K.hex> and launch the tool again.
For instructions on how to program the dongle firmware, refer to section 3 of the KB article located at <http://www.cypress.com/knowledge-base-article/updating-cy8ckit-042-ble-kitprog-and-cysmart-usb-dongle-firmware-kba96152>

Choose the .hex file from the respective location and update the BLE Dongle firmware.

Figure 3-15. Open Hex File



Programming Parameters

File Path: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE-A Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_4_2_Dongle_CySmart_256K.hex

Programmer: KitProg/BLE091E029900264400

Programming Mode: ● Reset ○ Power Cycle ○ Power Detect

Verification: ○ On ● Off

Connector: ● 5p ○ 10p

AutoDetection: ● On ○ Off

Clock Speed: 1.6 MHz

Programmer Characteristics

Protocol: ○ JTAG ● SWD ○ ISSP ○ I2C

Voltage: ● 5.0 V ○ 3.3 V ○ 2.5 V ○ 1.8 V

Status

Execution Time:

Power Status: ON

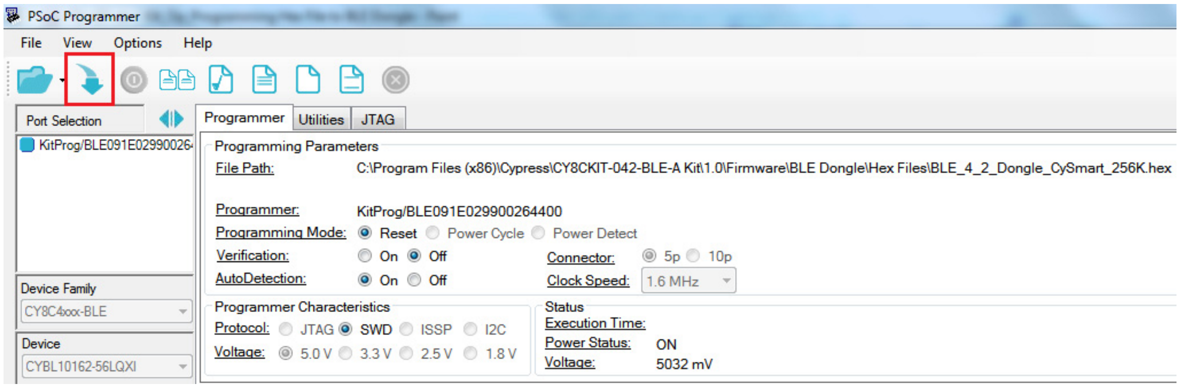
Voltage: 5032 mV

Device Family: CY8C4xxx-BLE

Device: CYBL10162-56LQXI

4. Ensure the other settings match as shown in Figure 3-15. 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 (Pass/Fail).

Figure 3-16. Programming Hex File to Dongle



Programming Parameters

File Path: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE-A Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_4_2_Dongle_CySmart_256K.hex

Programmer: KitProg/BLE091E029900264400

Programming Mode: ● Reset ○ Power Cycle ○ Power Detect

Verification: ○ On ● Off

Connector: ● 5p ○ 10p

AutoDetection: ● On ○ Off

Clock Speed: 1.6 MHz

Programmer Characteristics

Protocol: ○ JTAG ● SWD ○ ISSP ○ I2C

Voltage: ● 5.0 V ○ 3.3 V ○ 2.5 V ○ 1.8 V

Status

Execution Time:

Power Status: ON

Voltage: 5032 mV

Device Family: CY8C4xxx-BLE

Device: CYBL10162-56LQXI

5. After programming is completed successfully, the BLE Dongle firmware is updated and can be used to connect to the CySmart Central Emulation Tool.

3.7 USB-UART Bridge

The KitProg on both the BLE Pioneer Baseboard and BLE Dongle acts as a USB-UART bridge. When connected to a computer, a device named **KitProg USB-UART** is available under **Ports (COM & LPT)** in the Device Manager. The UART lines between modules and KitProg are hard-wired onboard, with UART_RX assigned to **P1_4** and UART_TX assigned to **P1_5** on PSoC 4 BLE/PRoC BLE device.

COM terminal software, such as Hyperterminal or TeraTerm, can be used to send and receive data. UART data sent from PSoC 4 BLE/PRoC BLE device on UART_TX line will be received by the software. Data entered in the software will be received by PSoC 4 BLE/PRoC BLE on UART_RX line. See CY8CKIT-042 PSoC 4 Pioneer Kit User Guide for more details.

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

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

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

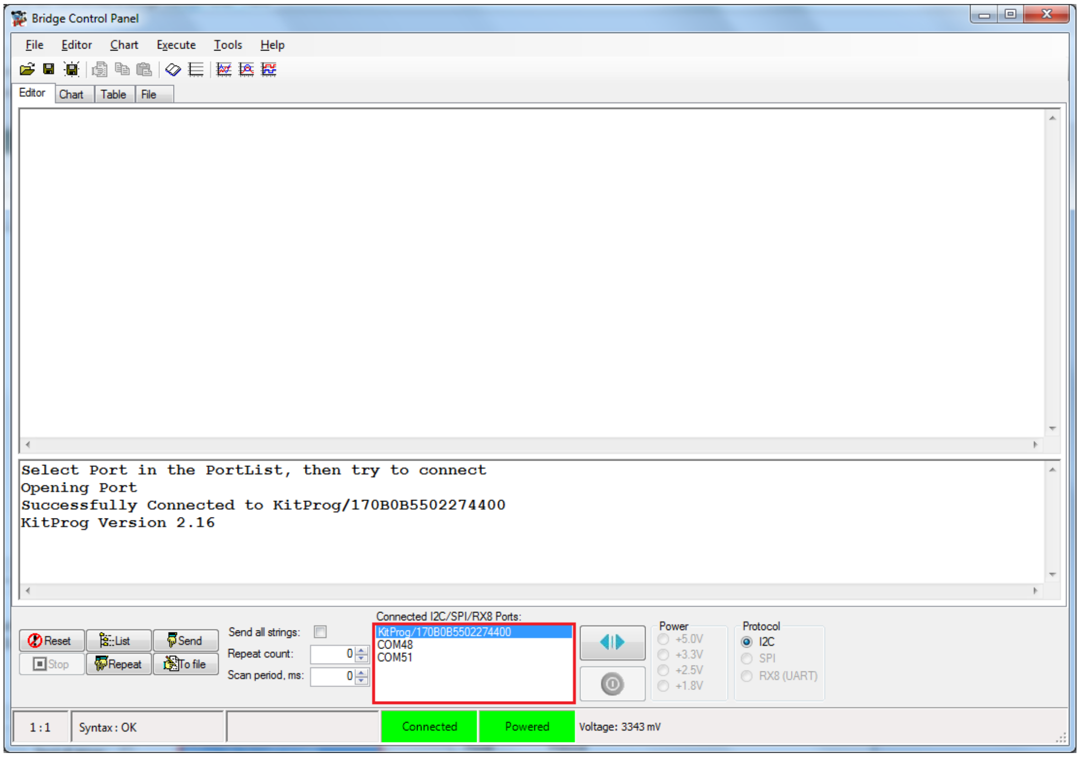
3.8 USB-I^{2}C Bridge

The KitProg can function as USB-I^{2}C bridge and communicate with the Bridge Control Panel (BCP) software utility. When connected to BCP, the **KitProg\<serial number>** is available under **Connected I2C/SPI/RX8 Ports** in the BCP. The I^{2}C connection between PSoC 4 BLE/PRoC BLE device and KitProg is used to transfer data between BCP and the PSoC 4 BLE/PRoC BLE device. The I^{2}C lines on PSoC 4 BLE/PRoC BLE device are **P3_4 (SDA)** and **P3_5 (SCL)**, which are hard-wired onboard to I^{2}C lines of KitProg. The USB-I^{2}C supports I^{2}C speed of 50 kHz, 100 kHz, 400 kHz and 1 MHz.

BCP is installed as part of the PSoC Programmer installation and can be accessed from **Start** > **All Programs** > **Cypress** > **Bridge Control Panel**. Refer to the Advanced section in the CY8CKIT-042 PSoC® 4 Pioneer Kit User Guide for more details.

To use the USB-I^{2}C functionality, select the **KitProg\<serial number>** in the BCP. On successful connection, the **Connected** and **Powered** status box turn green, as shown in Figure 3-17.

Figure 3-17. KitProg USB-I^{2}C Connected in Bridge Control Panel



```
Select Port in the PortList, then try to connect
Opening Port
Successfully Connected to KitProg/170B0B5502274400
KitProg Version 2.16
```

Send all strings: □
Repeat count: 0
Scan period, ms: 0

Connected I2C/SPI/RX8 Ports:
KitProg/170B0B5502274400
COM48
COM51

Power
○ +5.0V
○ +3.3V
○ +2.5V
○ +1.8V

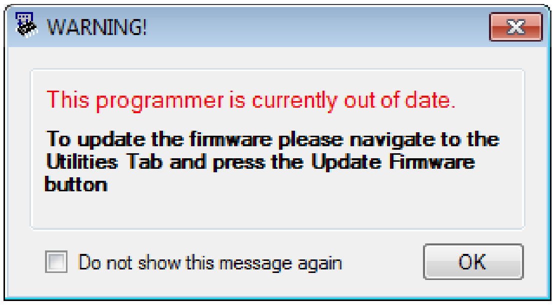
Protocol
● I2C
○ SPI
○ RX8 (UART)

1:1 | Syntax: OK | Connected | Powered | Voltage: 3343 mV

3.9 Updating the KitProg Firmware

The KitProg firmware normally does not require any update. If an update is required, then PSoC Programmer will display a warning message when the kit is connected to it, as shown in Figure 3-18.

Figure 3-18. Update KitProg



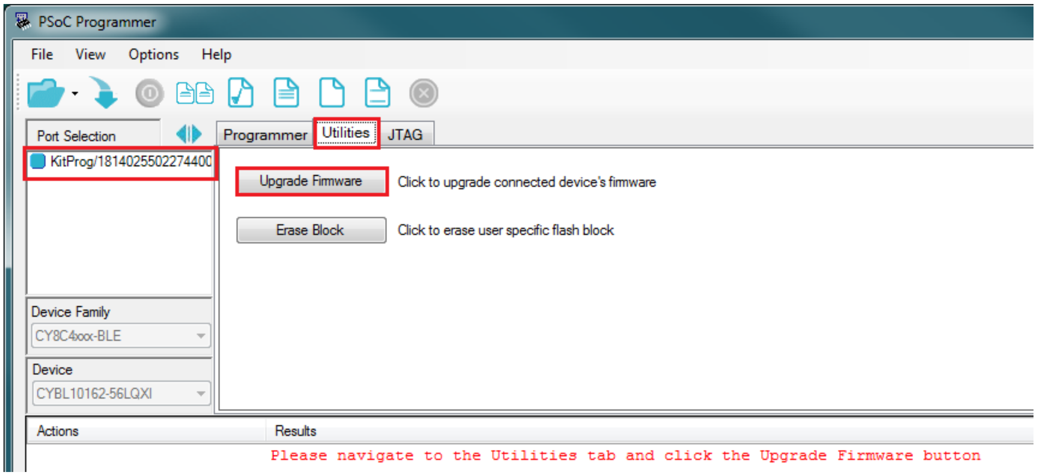
WARNING!

This programmer is currently out of date.

To update the firmware please navigate to the Utilities Tab and press the Update Firmware button

To update the KitProg, go to the **Utilities** tab on PSoC Programmer and click **Upgrade Firmware**, as shown in Figure 3-19.

Figure 3-19. Update KitProg from PSoC Programmer



KitProg/1814025502274400

Click to upgrade connected device's firmware

Click to erase user specific flash block

Device Family: CY8C4xxx-BLE

Device: CYBL10162-56LQXI

```
Please navigate to the Utilities tab and click the Upgrade Firmware button
```

4. Example Projects

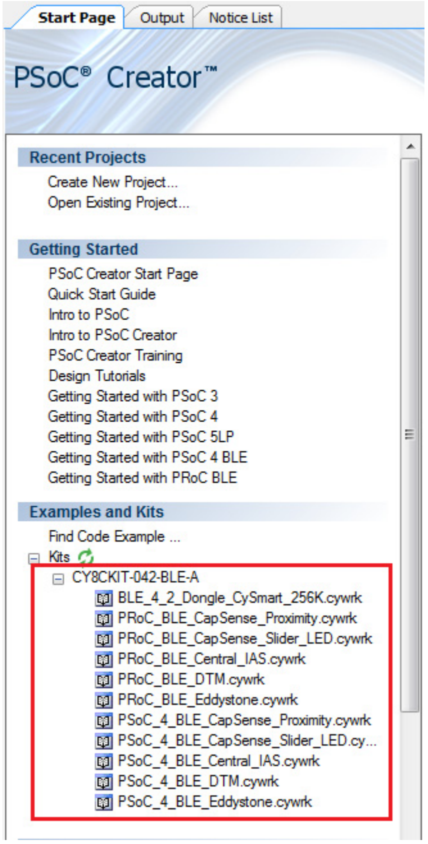
This chapter demonstrates the functionality of PSoC 4 BLE and PRoC BLE devices using the BLE Pioneer Kit example projects. Download and install the kit setup file from the kit web page. The example projects can be accessed on the Start Page of PSoC Creator under **Examples and Kits**.

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 3.3** > **PSoC Creator 3.3**.
2. On the Start Page, under the **Examples and Kits** section, choose **Kits** > **CY8CKIT-042-BLE-A**. 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 the desired example project.

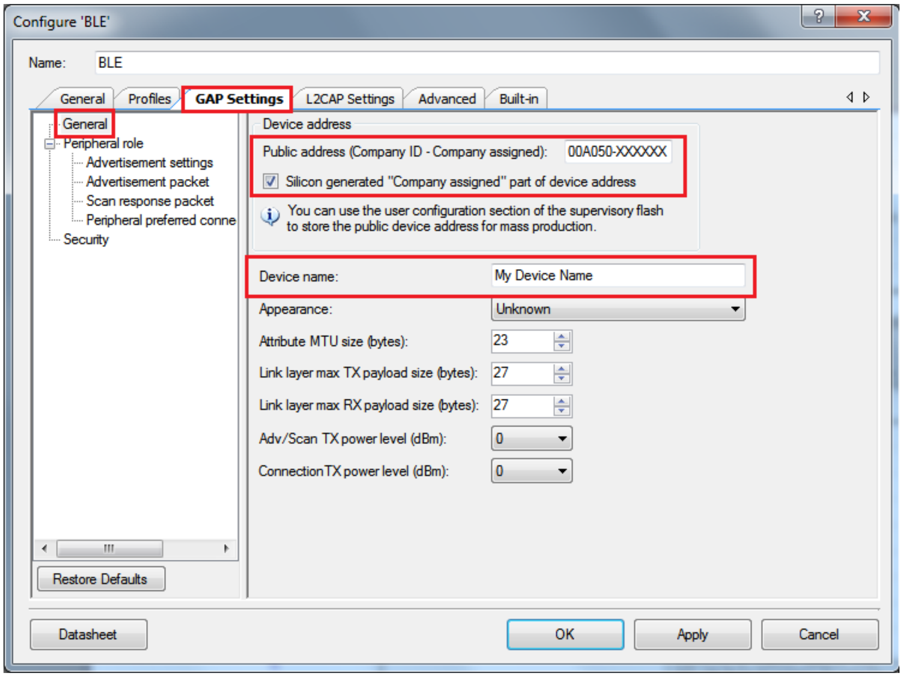
Figure 4-1. Open Example Project from PSoC Creator


Start Page
Output
Notice List
PSoC® Creator™
Recent Projects
Create New Project...
Open Existing Project...
Getting Started
PSoC Creator Start Page
Quick Start Guide
Intro to PSoC
Intro to PSoC Creator
PSoC Creator Training
Design Tutorials
Getting Started with PSoC 3
Getting Started with PSoC 4
Getting Started with PSoC 5LP
Getting Started with PSoC 4 BLE
Getting Started with PRoC BLE
Examples and Kits
Find Code Example ...
Kits
CY8CKIT-042-BLE-A
BLE_4_2_Dongle_CySmart_256K.cywrk
PRoC_BLE_CapSense_Proximity.cywrk
PRoC_BLE_CapSense_Slider_LED.cywrk
PRoC_BLE_Central_IAS.cywrk
PRoC_BLE_DTM.cywrk
PRoC_BLE_Eddystone.cywrk
PSoC_4_BLE_CapSense_Proximity.cywrk
PSoC_4_BLE_CapSense_Slider_LED.cy...
PSoC_4_BLE_Central_IAS.cywrk
PSoC_4_BLE_DTM.cywrk
PSoC_4_BLE_Eddystone.cywrk

4. Select the folder where you want to save the project and click **OK**.
5. Every BLE project use a public address set in the BLE Component GUI to advertise and scan, depending on the role: Peripheral or Central mode. If multiple kits in close proximity have the same public address, then wrong devices may be connected or connections may fail. To prevent this, change the **Public address** (and preferably **Device name**) in the BLE Component **GAP Settings** tab as shown in Figure 4-2. Click **OK**.

   Alternatively, you can select the 'Silicon generated' device address by selecting the check box. This way, the Bluetooth device (BD) address is generated using the silicon ID, unique to each device. Click **OK**.

Figure 4-2. Change BLE Public Address and Name



Name: BLE

Device address

Public address (Company ID - Company assigned): 00A050-XXXXXX

☑ Silicon generated "Company assigned" part of device address

You can use the user configuration section of the supervisory flash to store the public device address for mass production.

Device name: My Device Name

Appearance: Unknown

Attribute MTU size (bytes): 23

Link layer max TX payload size (bytes): 27

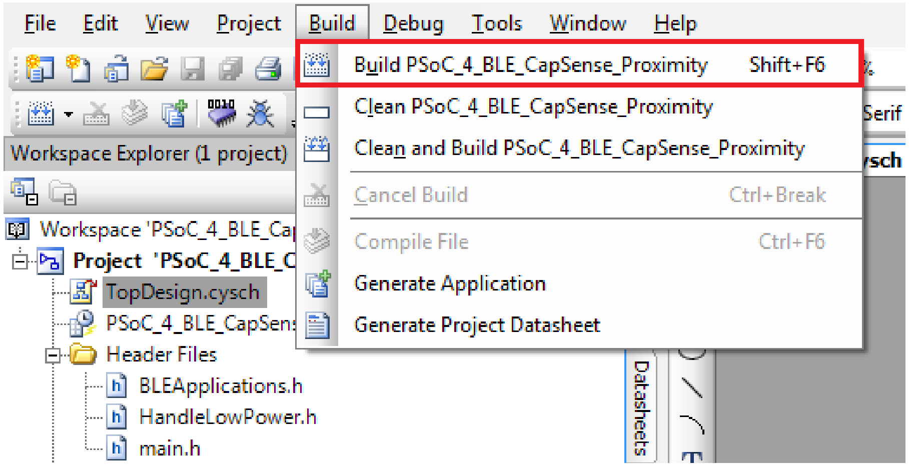
Link layer max RX payload size (bytes): 27

Adv/Scan TX power level (dBm): 0

Connection TX power level (dBm): 0

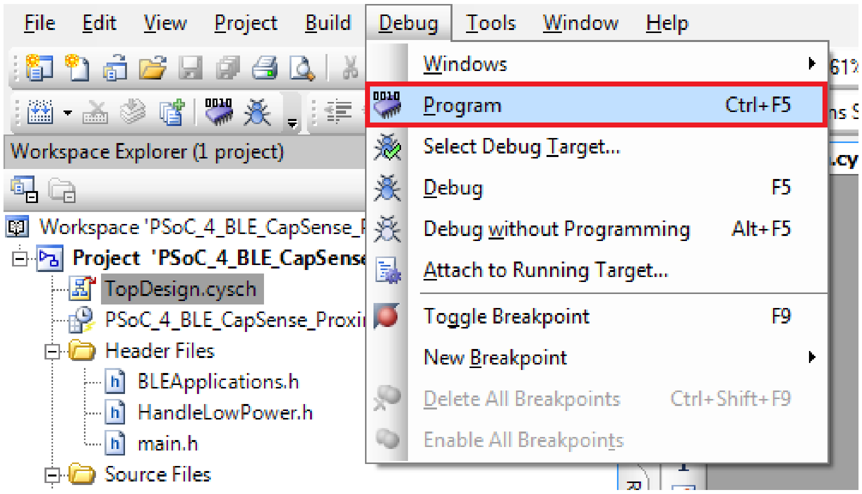
6. Build the example project by choosing **Build** > **Build <Project Name>**, as shown in Figure 4-3. A hex file will be generated.

Figure 4-3. Build Project from PSoC Creator



7. Connect the BLE Pioneer Baseboard to the computer through the USB Mini-B connector J13. Ensure that the correct module (PSoC 4 BLE or PRoC BLE) is placed on the baseboard, depending on the project opened.
8. Choose **Debug** > **Program** in PSoC Creator, as shown in Figure 4-4.

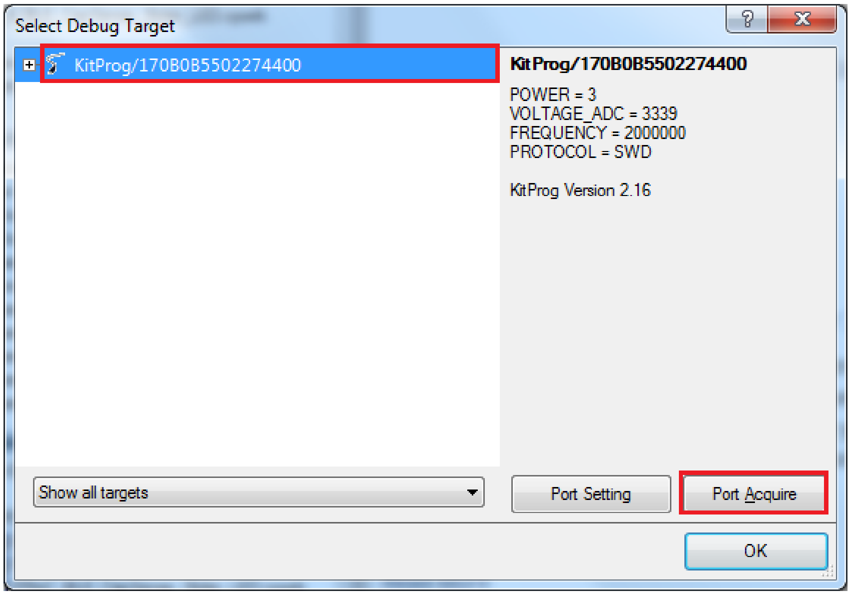
Figure 4-4. Program Device in PSoC Creator



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 serial ID starting with *'BLE'* belongs to the BLE Dongle (see Updating BLE Dongle for CySmart Central Emulation Tool on page 30).

Figure 4-5. Port Acquire



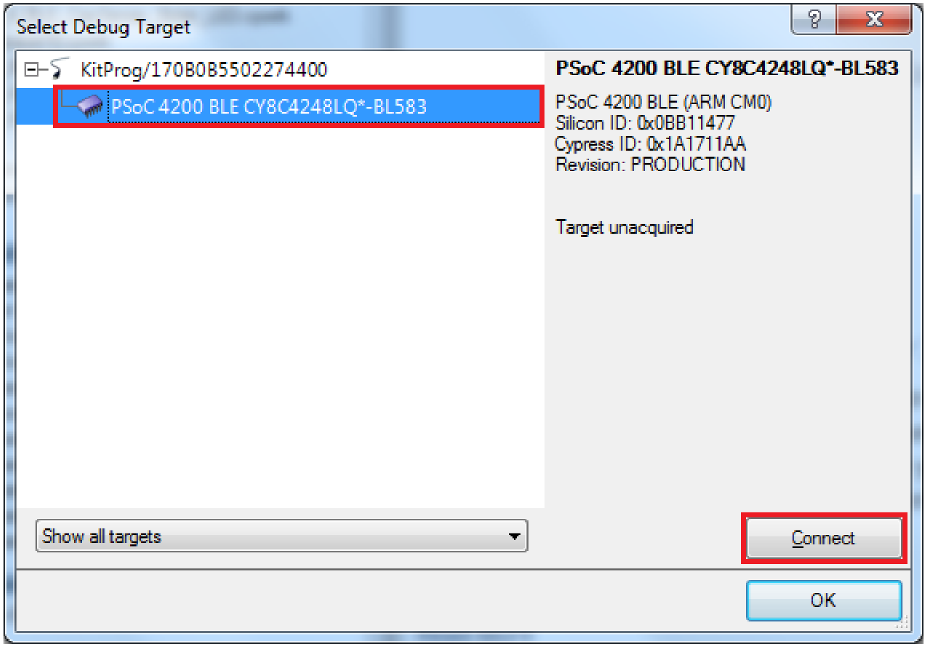
KitProg/170B0B5502274400

POWER = 3
VOLTAGE_ADC = 3339
FREQUENCY = 2000000
PROTOCOL = SWD

KitProg Version 2.16

10. After the device is acquired, it is shown in a 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



Select Debug Target

KitProg/170B0B5502274400

PSoC 4200 BLE CY8C4248LQ*-BL583

PSoC 4200 BLE CY8C4248LQ*-BL583

PSoC 4200 BLE (ARM CM0)
Silicon ID: 0x0BB11477
Cypress ID: 0x1A1711AA
Revision: PRODUCTION

Target unacquired

Note: As stated previously, the BLE Pioneer Kit supports both Cypress BLE devices: PSoC 4 BLE and PRoC BLE. The description, hardware configurations, and verification method of the example projects explained in the following sections are valid for both these devices. Unless explicitly mentioned, the theory and usability for these example projects should be considered the same for both the modules.

This document refers to the BLE Pioneer Kits, BLE Dongle, and PC/mobile as Central or Peripheral devices. A Central device is normally the master and requests/commands data from the Peripheral device. BLE-enabled phones and computers are one such example. Peripheral devices store the actual data and send it to the Central device when requested. Examples include BLE-enabled sensors, proximity beacons, and so on.

If you are a beginner in BLE, refer to the PSoC Creator code examples such as BLE_FindMe and BLE_Device_Information_Service as described in PSoC Creator Code Examples on page 13. You may also refer the application note Getting Started with PSoC 4 BLE.

The four Kit code examples viz. CapSense Slider and LED, CapSense Proximity, BLE Central Mode and Eddystone are intermediatory level examples that will help to design a system around the kit. Refer to the 4.2 CapSense Slider and LED and later for details.

The CySmart Dongle code example is an advanced level example that will demonstrate a complete solution around the kit. Refer to 4.6 BLE Dongle and LED Control for details.

4.2 CapSense Slider and LED

4.2.1 Project Description

This project demonstrates connectivity between the BLE Pioneer Kit (acting as a Peripheral and GATT server device) and CySmart Central Emulation tool or mobile device running the CySmart mobile application (acting as a Central and GATT client device). This project demonstrates the following:

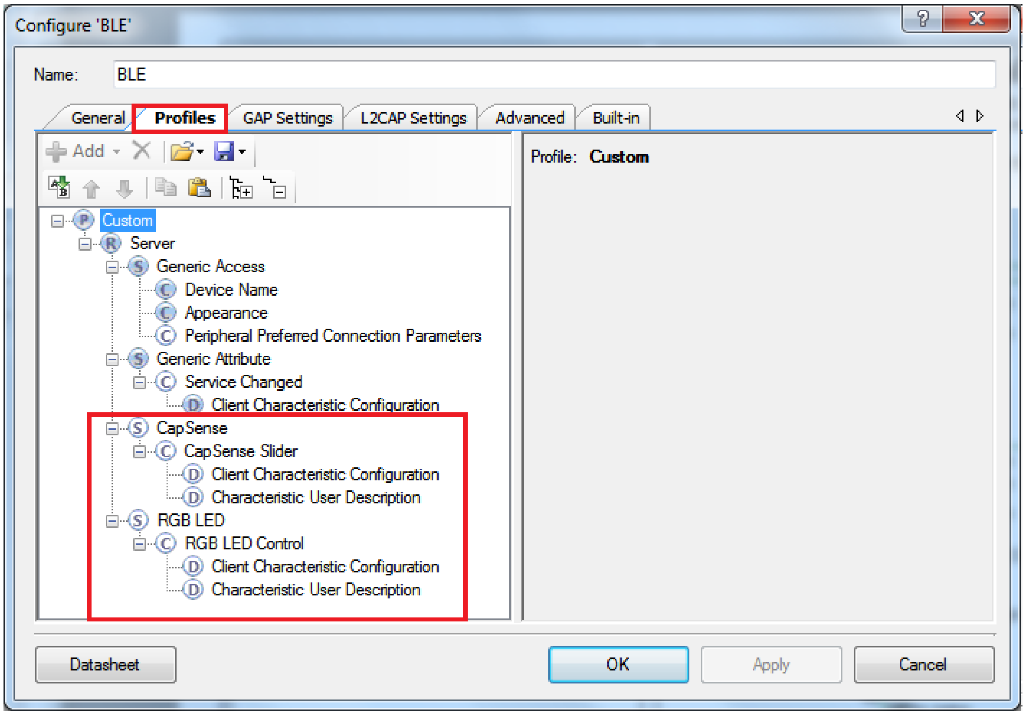
- Advertisement with timeout
- Connection with any Central device
- Two custom services in single profile
- Data transfer over BLE custom service using notifications, read, and write
- Low-power mode implementation for coin-cell operation

The BLE profile in this project consists of two BLE custom services: CapSense and RGB LED. The **CapSense service** consists of one custom characteristic, termed as CapSense Slider. The CapSense slider characteristic is used to send one byte data, ranging from 0 to 100, as notification to the GATT client device. This data is the finger location read by the CapSense Component on the five-segment slider (CSS1) present on the kit. This characteristics supports notification, which allows the GATT server to send data to the connected client device whenever new data is available.

The **RGB LED service** also consists of one custom characteristic, termed as RGB LED Control. This characteristic supports two operations, read and write, through which the connected GATT client device can read data as well as write a new value to the characteristic. This data has four byte values indicating red, green, blue, and the intensity values to control the onboard RGB LED.

The properties for the custom service/characteristics are configured in the BLE Component under the Profiles tab, as shown in Figure 4-7.

Figure 4-7. Attributes Configuration in BLE Component for Custom Services



Name: BLE

Profile: **Custom**

- Custom
  - Server
    - Generic Access
      - Device Name
      - Appearance
      - Peripheral Preferred Connection Parameters
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration
    - CapSense
      - CapSense Slider
        - Client Characteristic Configuration
        - Characteristic User Description
    - RGB LED
      - RGB LED Control
        - Client Characteristic Configuration
        - Characteristic User Description

The project consists of the following files:

- **main.c/.h**

  These files contain the main function, which is the entry point and execution of the firmware application. They also contain the function definition for initialization of the system and reading the CapSense slider data from the CapSense Component.

- **BLEApplications.c/.h**

  These files contain all the macros and function definitions related to BLE communication and operation. They include the event callback function definition that is registered with the BLE Component startup and used to send BLE-related events from the BLE stack to the application layer for processing. These files contain a method to send CapSense notifications to the GATT client device and process the Read and Write commands on the RGB LED characteristic by the GATT client device. They update the BLE Connection parameter, which is important for low-power mode usage.

- **HandleLowPower.c/.h**

  These files contain the function to handle low-power mode. This function is continuously called in the main loop and is responsible for pushing the BLE hardware block (BLESS) as well as the CPU to Deep Sleep mode as much as possible. The wakeup source is either the BLE hardware block link layer internal timer or the interrupt from the user button press (**SW2**). This allows for very low power mode implementation and operation using a coin cell.

Additionally, the PRoC BLE version of this project consists of the *RGB_PRSm.c/.h* file, which contains the function to drive the software-based PrISM method; it also drives the color and intensity on the RGB LED.

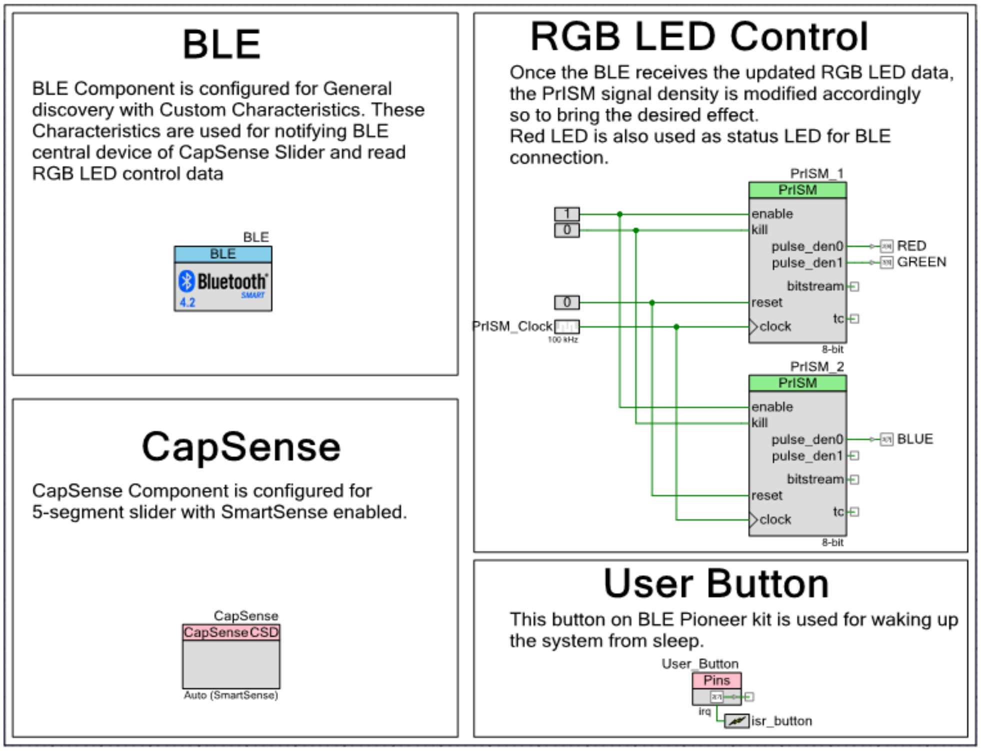
This is the default firmware that comes in the modules shipped with the kit.

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 4-8. TopDesign for PSoC_4_BLE_CapSense_Slider_LED Project



BLE

BLE Component is configured for General discovery with Custom Characteristics. These Characteristics are used for notifying BLE central device of CapSense Slider and read RGB LED control data

RGB LED Control

Once the BLE receives the updated RGB LED data, the PrISM signal density is modified accordingly so to bring the desired effect.
Red LED is also used as status LED for BLE connection.

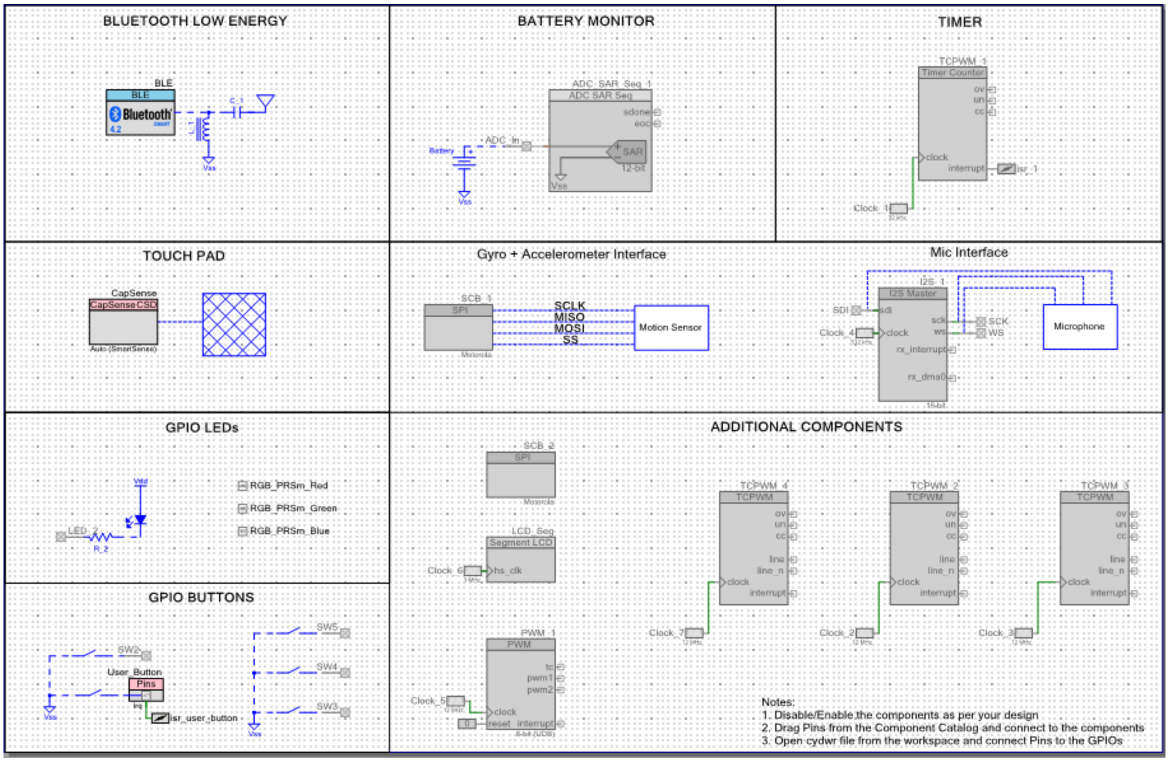
CapSense

CapSense Component is configured for 5-segment slider with SmartSense enabled.

User Button

This button on BLE Pioneer kit is used for waking up the system from sleep.

Figure 4-9. TopDesign for PRoC_BLE_CapSense_Slider_LED Project



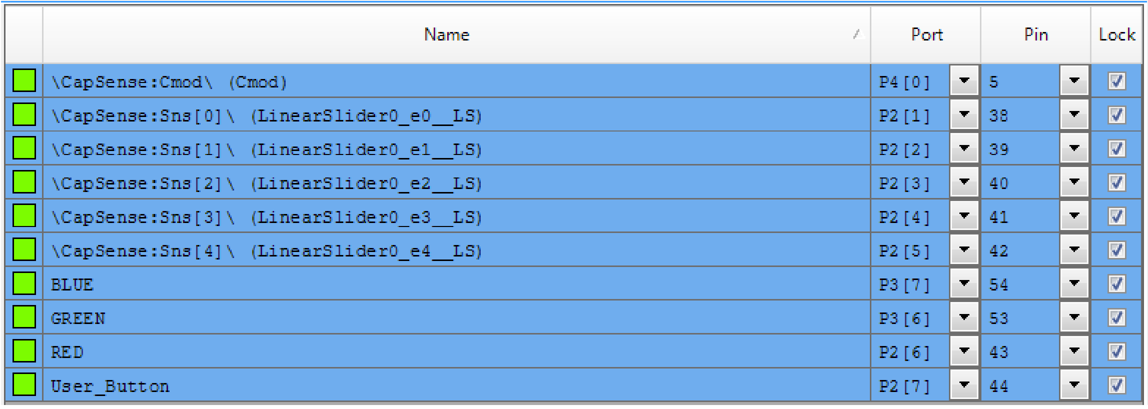
Notes:
1. Disable/Enable the components as per your design
2. Drag Pins from the Component Catalog and connect to the components
3. Open cydwr file from the workspace and connect Pins to the GPIOs

4.2.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the BLE Pioneer Baseboard. Ensure that the correct 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.

The pin assignment for this project is in **PSoC_4_BLE_CapSense_Slider_LED.cydwr**/ **PRoC_BLE_CapSense_Slider_LED.cydwr** in the Workspace Explorer, as shown in Figure 4-10.

Figure 4-10. Pin Selection for CapSense Slider and LED Project

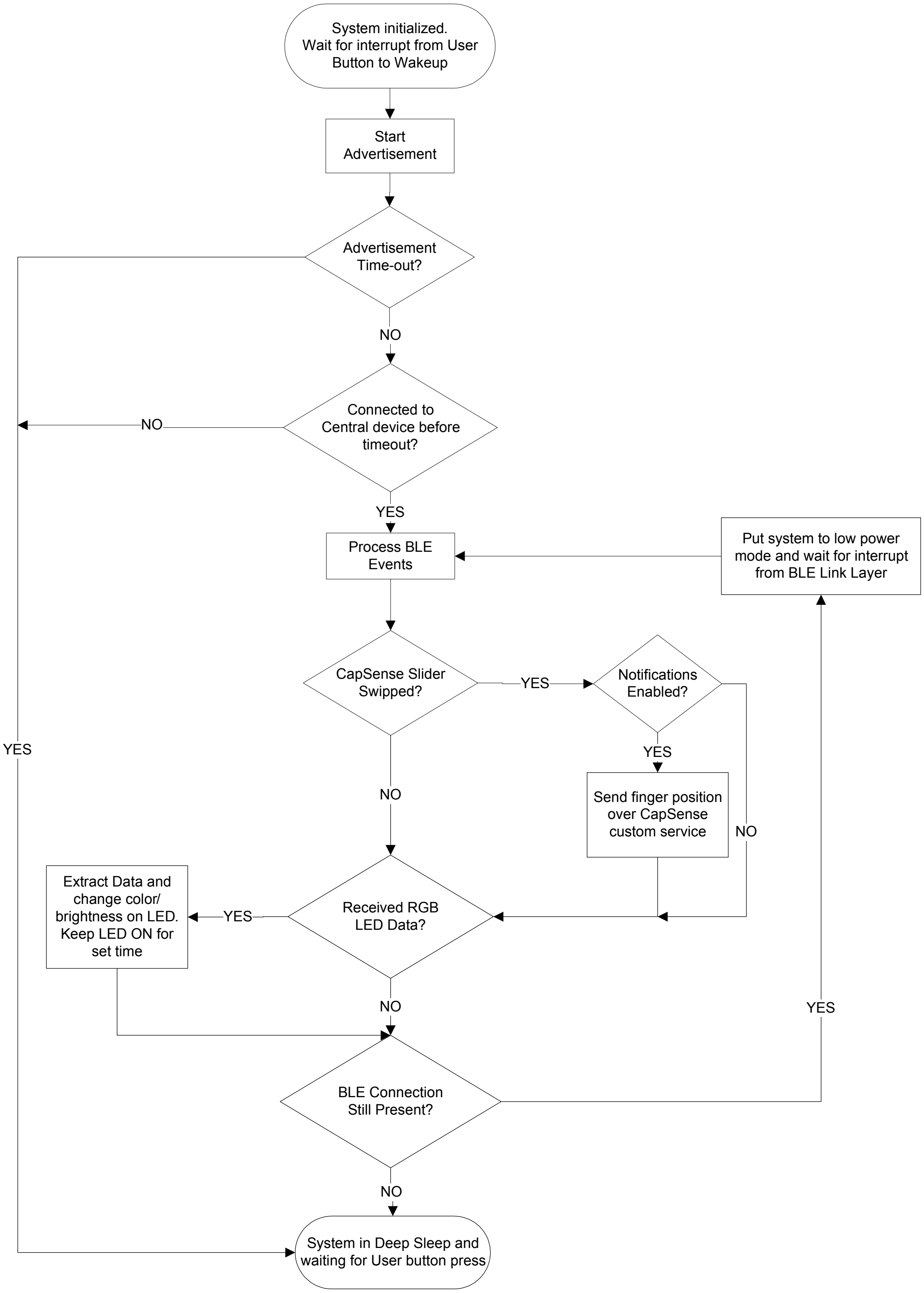


	Name	Port	Pin	Lock
🟩	\CapSense:Cmod\ (Cmod)	P4[0]	5	☑
🟩	\CapSense:Sns[0]\ (LinearSlider0_e0__LS)	P2[1]	38	☑
🟩	\CapSense:Sns[1]\ (LinearSlider0_e1__LS)	P2[2]	39	☑
🟩	\CapSense:Sns[2]\ (LinearSlider0_e2__LS)	P2[3]	40	☑
🟩	\CapSense:Sns[3]\ (LinearSlider0_e3__LS)	P2[4]	41	☑
🟩	\CapSense:Sns[4]\ (LinearSlider0_e4__LS)	P2[5]	42	☑
🟩	BLUE	P3[7]	54	☑
🟩	GREEN	P3[6]	53	☑
🟩	RED	P2[6]	43	☑
🟩	User_Button	P2[7]	44	☑

4.2.3 Flow Chart

Figure 4-11 shows the flow chart of the code implemented.

Figure 4-11. CapSense Slider and LED Project Flow Chart


System initialized.
Wait for interrupt from User Button to Wakeup
Start Advertisement
Advertisement Time-out?
NO
Connected to Central device before timeout?
NO
YES
Process BLE Events
Put system to low power mode and wait for interrupt from BLE Link Layer
CapSense Slider Swipped?
YES
Notifications Enabled?
YES
Send finger position over CapSense custom service
NO
YES
NO
Extract Data and change color/ brightness on LED. Keep LED ON for set time
YES
Received RGB LED Data?
NO
YES
BLE Connection Still Present?
NO
System in Deep Sleep and waiting for User button press

4.2.4 Verify Output

The project can be verified by two methods: using the CySmart Central Emulation Tool and BLE Dongle or using the CySmart mobile application.

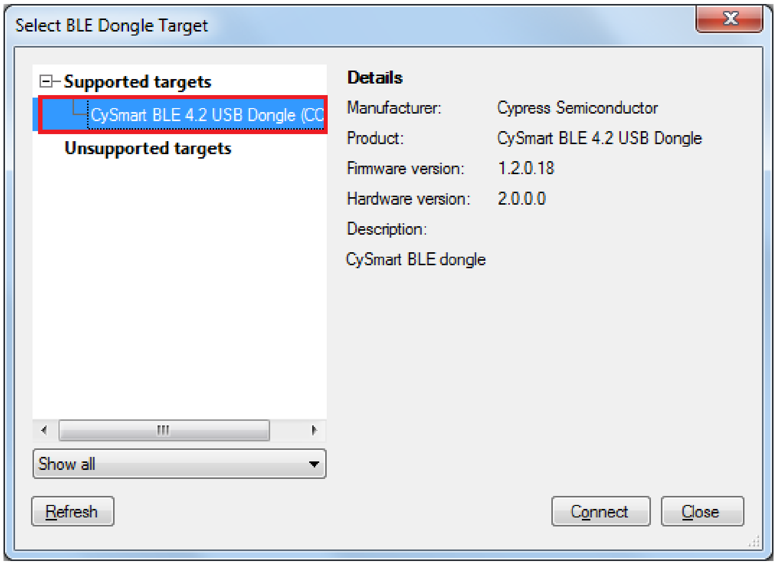
4.2.4.1 CySmart Central Emulation Tool

To verify the CapSense and LED project using the CySmart Central Emulation Tool, follow these steps:

Note: Refer CySmart Central Emulation tool to learn how to use the tool.

1. Connect the BLE Dongle to one of the USB ports on the computer.
2. Start the CySmart Central Emulation Tool on the computer by going to **Start** > **All Programs** > **Cypress** > **CySmart <version>** > **CySmart <version>**. You will see a list of BLE Dongles connected to it. If no dongle is found, click **Refresh**. Select the BLE Dongle and click **Connect**.

Figure 4-12. Connect to BLE Dongle



Details

Manufacturer: Cypress Semiconductor

Product: CySmart BLE 4.2 USB Dongle

Firmware version: 1.2.0.18

Hardware version: 2.0.0.0

Description:

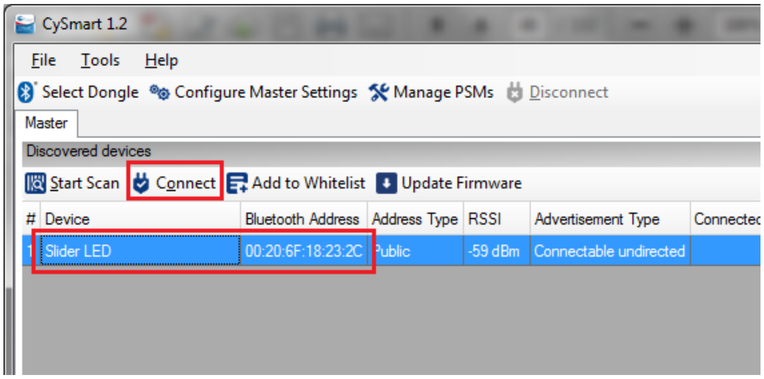
CySmart BLE dongle

3. Place the module 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 project. Follow steps in Using Example Projects on page 35 to program the device.
6. After programming successfully, press the user button (**SW2**) on the BLE Pioneer Kit to start the advertisement. Advertisement is indicated by a blinking red LED on the baseboard.

   Note: The project has an advertisement timeout of 30 seconds after which it returns to Deep Sleep mode. Press **SW2** again to restart the advertisement.
7. On the CySmart Central Emulation Tool, click **Start Scan** to see the list of available BLE Peripheral devices.

8. Double-click the **Slider LED** device to connect, or click **Slider LED** and then click **Connect**.

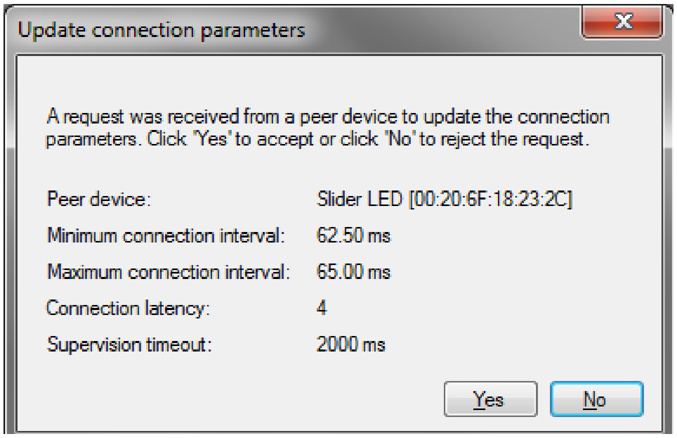
Figure 4-13. Connect to BLE Slider and LED Peripheral



#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connectec
1	Slider LED	00:20:6F:18:23:2C	Public	-59 dBm	Connectable undirected	

9. When connected, the CySmart Central Emulation Tool will display a message for the **Update connection parameters**. Select **Yes**, as shown in Figure 4-14.

Figure 4-14. Update Connection Parameter Option


Update connection parameters

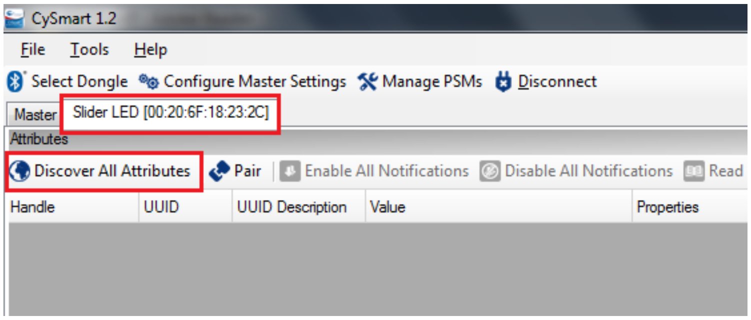
A request was received from a peer device to update the connection parameters. Click 'Yes' to accept or click 'No' to reject the request.

Peer device:	Slider LED [00:20:6F:18:23:2C]
Minimum connection interval:	62.50 ms
Maximum connection interval:	65.00 ms
Connection latency:	4
Supervision timeout:	2000 ms

Note: If you select **No**, the project will still work. However, the current consumption will be higher due to faster connection interval.

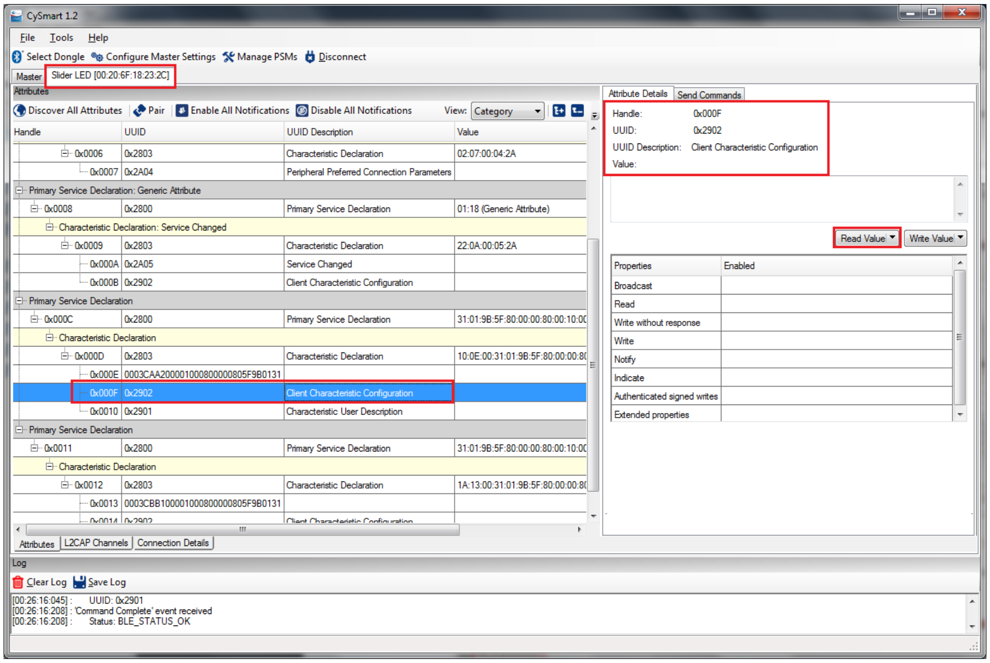
10. Click **Discover All Attributes** to find all attributes supported.

Figure 4-15. Discover All Attributes



11. Locate the attribute **Client Characteristic Configuration** descriptor (UUID 0x2902) under CapSense slider characteristic (**UUID 0x0003CAA2-0000-1000-8000-00805F9B0131**). Click **Read Value** to read the existing Client Characteristic Configuration Descriptor (CCCD) value as shown in Figure 4-16.

Figure 4-16. Read CCCD for CapSense Slider Characteristic



Handle	UUID	UUID Description	Value
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A
0x0007	0x2A04	Peripheral Preferred Connection Parameters	
Primary Service Declaration: Generic Attribute			
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)
Characteristic Declaration: Service Changed			
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
Primary Service Declaration			
0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10:00
Characteristic Declaration			
0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5F:80:00:00:80
0x000E	0003CAA2000010008000000805F9B0131		
0x000F	0x2902	Client Characteristic Configuration	
0x0010	0x2901	Characteristic User Description	
Primary Service Declaration			
0x0011	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10:00
Characteristic Declaration			
0x0012	0x2803	Characteristic Declaration	1A:13:00:31:01:9B:5F:80:00:00:80
0x0013	0003CBB1000010008000000805F9B0131		
0x0014	0x2902	Client Characteristic Configuration	

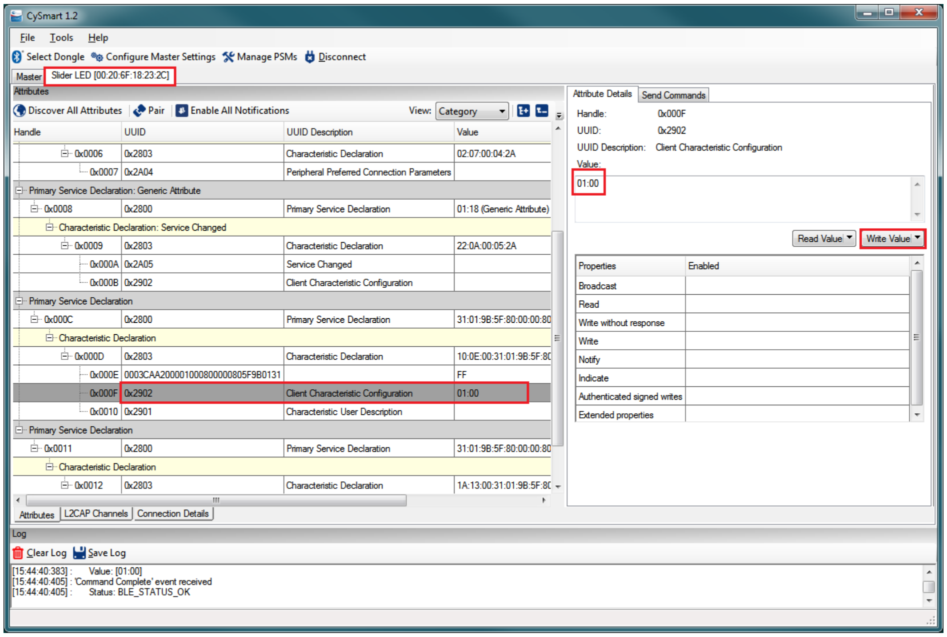
Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

```
[00:26:16:045] :       UUID: 0x2901
[00:26:16:208] : 'Command Complete' event received
[00:26:16:208] :       Status: BLE_STATUS_OK
```

12. Modify the **Value** field of CCCD to '01:00' and click **Write Value**. This enables the notifications on the CapSense slider characteristic.

Figure 4-17. Write CCCD to Enable Notifications



Handle	UUID	UUID Description	Value
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A
0x0007	0x2A04	Peripheral Preferred Connection Parameters	
Primary Service Declaration: Generic Attribute			
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)
Characteristic Declaration: Service Changed			
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
Primary Service Declaration			
0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80
Characteristic Declaration			
0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5F:80
0x000E	0003CAA2000010008000000805F9B0131		FF
0x000F	0x2902	Client Characteristic Configuration	01:00
0x0010	0x2901	Characteristic User Description	
Primary Service Declaration			
0x0011	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80
Characteristic Declaration			
0x0012	0x2803	Characteristic Declaration	1A:13:00:31:01:9B:5F:80

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:
01:00

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

```
[15:44:40:383] :       Value: [01:00]
[15:44:40:405] : 'Command Complete' event received
[15:44:40:405] :       Status: BLE_STATUS_OK
```

13. Swipe your finger on the CapSense slider on the BLE Pioneer kit, as shown in Figure 4-18 and see the notification values in the CapSense Slider value field, as shown in Figure 4-19.

Figure 4-18. CapSense Slider

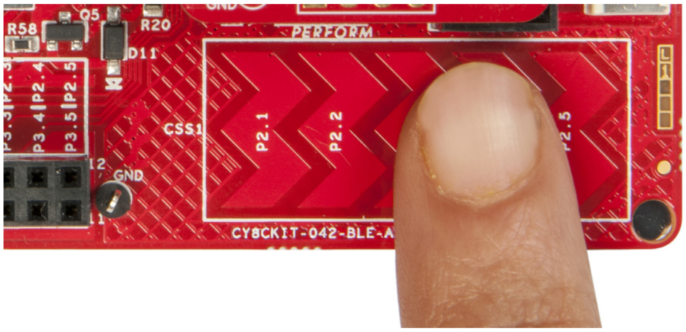
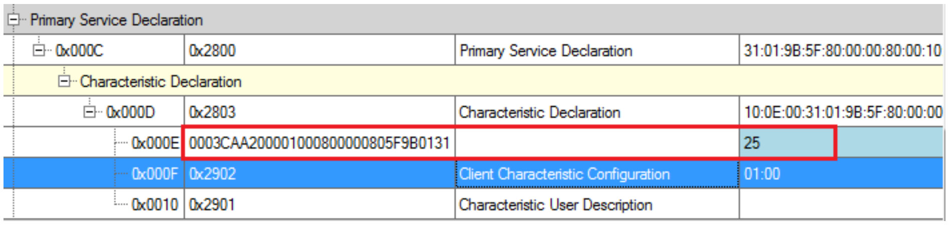


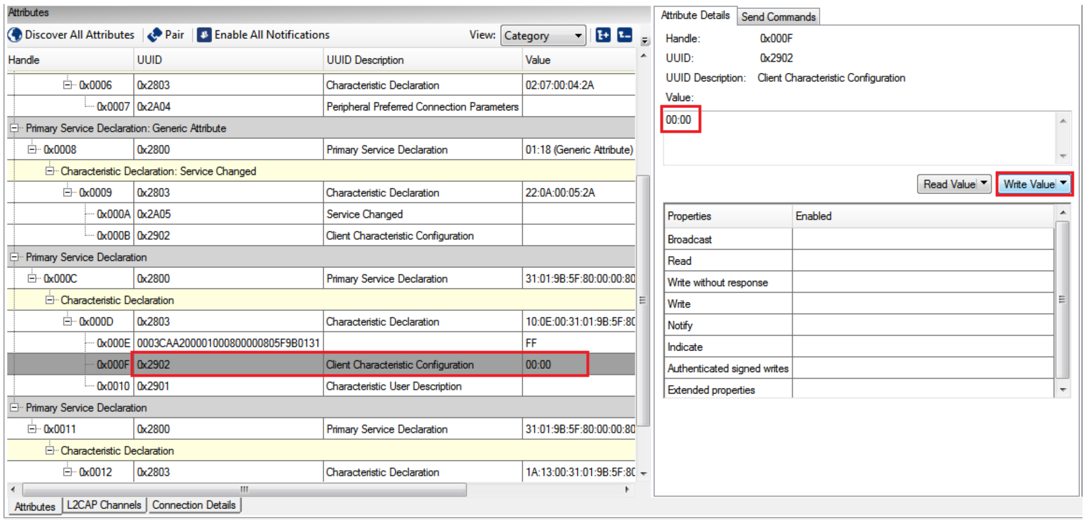
Figure 4-19. CapSense Slider Notification Received



⊟ Primary Service Declaration			
⊟ 0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10
⊟ Characteristic Declaration			
⊟ 0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5F:80:00:00
0x000E	0003CAA2000010008000000805F9B0131		25
0x000F	0x2902	Client Characteristic Configuration	01:00
0x0010	0x2901	Characteristic User Description	

14. To disable notifications, modify the **Value** field of the **Client Characteristic Configuration** descriptor to '00:00' and click **Write Value**.

Figure 4-20. Disable Notifications



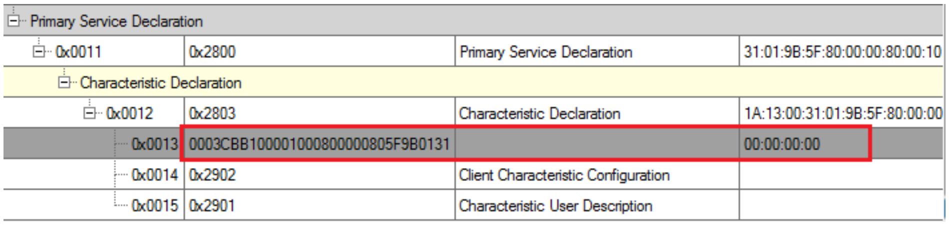
Handle	UUID	UUID Description	Value
⊟ 0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A
0x0007	0x2A04	Peripheral Preferred Connection Parameters	
⊟ Primary Service Declaration: Generic Attribute			
⊟ 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)
⊟ Characteristic Declaration: Service Changed			
⊟ 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
⊟ Primary Service Declaration			
⊟ 0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80
⊟ Characteristic Declaration			
⊟ 0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5F:80
0x000E	0003CAA2000010008000000805F9B0131		FF
0x000F	0x2902	Client Characteristic Configuration	00:00
0x0010	0x2901	Characteristic User Description	
⊟ Primary Service Declaration			
⊟ 0x0011	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80
⊟ Characteristic Declaration			
⊟ 0x0012	0x2803	Characteristic Declaration	1A:13:00:31:01:9B:5F:80

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:
00:00

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

15. Locate the **RGB LED Control** characteristic (**UUID 0x0003CBB1-0000-1000-8000-00805F9B0131**). Click **Read Value** to read the existing 4-byte onboard RGB LED color information, as shown in Figure 4-21. The four bytes indicate red, green, blue, and the overall brightness, respectively.

Figure 4-21. Read RGB LED Control Characteristic Value

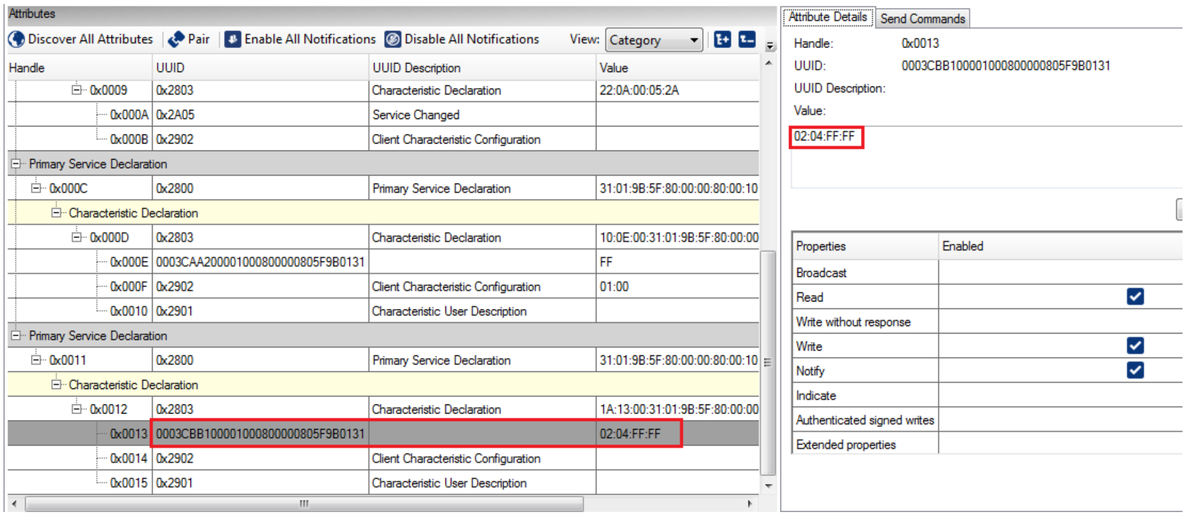


⊟ Primary Service Declaration			
⊟ 0x0011	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10
⊟ Characteristic Declaration			
⊟ 0x0012	0x2803	Characteristic Declaration	1A:13:00:31:01:9B:5F:80:00:00
0x0013	0003CBB100001000800000805F9B0131		00:00:00:00
0x0014	0x2902	Client Characteristic Configuration	
0x0015	0x2901	Characteristic User Description	

16. 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 4-22. The RGB LED will be on for 3 seconds before switching off to conserve power.

    Note: If the kit is powered from a 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 4-22. Write RGB LED Control Characteristic Value

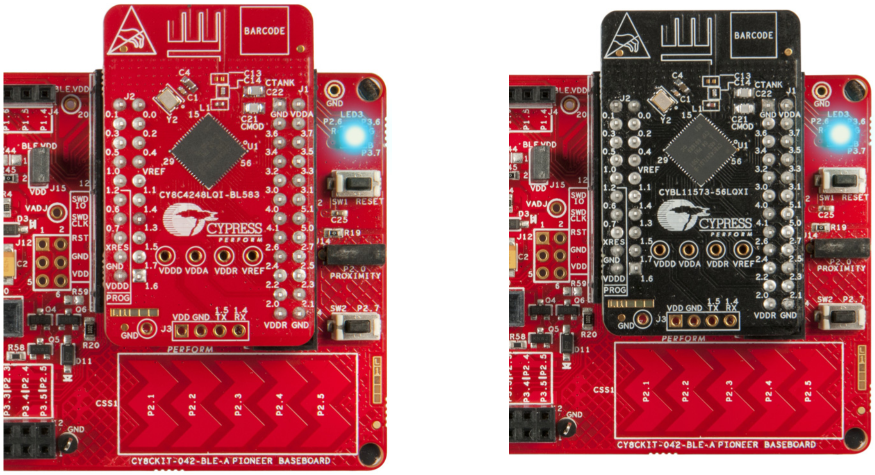


Handle	UUID	UUID Description	Value
⊟ 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
⊟ Primary Service Declaration			
⊟ 0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10
⊟ Characteristic Declaration			
⊟ 0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5F:80:00:00
0x000E	0003CAA200001000800000805F9B0131		FF
0x000F	0x2902	Client Characteristic Configuration	01:00
0x0010	0x2901	Characteristic User Description	
⊟ Primary Service Declaration			
⊟ 0x0011	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:00:80:00:10
⊟ Characteristic Declaration			
⊟ 0x0012	0x2803	Characteristic Declaration	1A:13:00:31:01:9B:5F:80:00:00
0x0013	0003CBB100001000800000805F9B0131		02:04:FF:FF
0x0014	0x2902	Client Characteristic Configuration	
0x0015	0x2901	Characteristic User Description	

Handle: 0x0013
UUID: 0003CBB100001000800000805F9B0131
UUID Description:
Value:
02:04:FF:FF

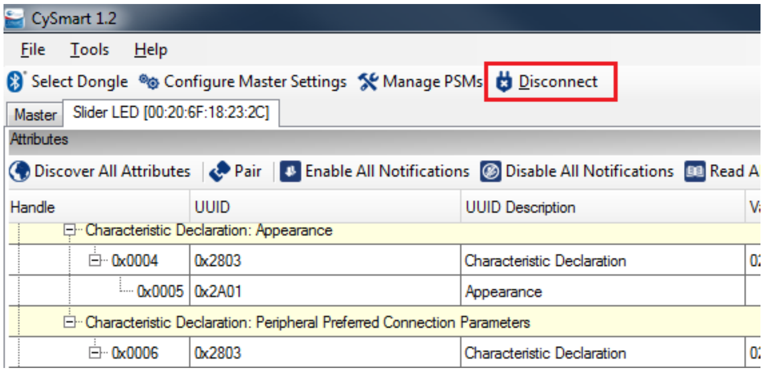
Properties	Enabled
Broadcast	
Read	☑
Write without response	
Write	☑
Notify	☑
Indicate	
Authenticated signed writes	
Extended properties	

Figure 4-23. RGB LED Control with PSoC 4 BLE Module and PRoC BLE Module



17. To disconnect from the device, click **Disconnect**, as shown in Figure 4-24.

Figure 4-24. Disconnect from the Device



Handle	UUID	UUID Description	V
Characteristic Declaration: Appearance			
0x0004	0x2803	Characteristic Declaration	0
0x0005	0x2A01	Appearance	
Characteristic Declaration: Peripheral Preferred Connection Parameters			
0x0006	0x2803	Characteristic Declaration	0

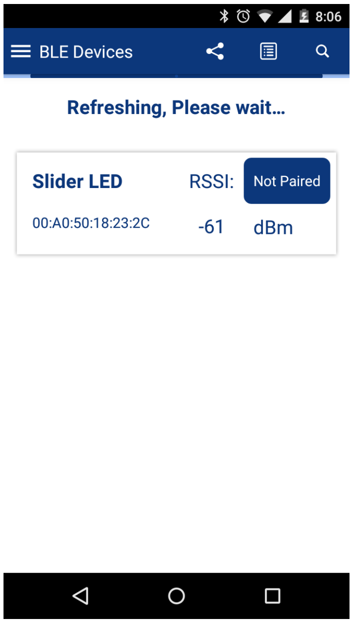
18. To connect to this peripheral again, restart advertising by pressing the user button (**SW2**) on the BLE Pioneer Kit. Advertising is indicated by the blinking red LED.

4.2.4.2 *CySmart Mobile Application*

To verify the CapSense and LED project using the CySmart mobile application (refer CySmart Mobile App webpage), follow these steps:

1. Plug the desired module on the BLE Pioneer Baseboard.
2. Connect the BLE Pioneer Kit into the computer using the J13 USB connector.
3. Program the kit with the CapSense Slider and LED example project. See Using Example Projects on page 35 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 CySmart mobile application will automatically search for available peripherals and list them. Select the **Slider LED** peripheral as shown in Figure 4-25.

Figure 4-25. Slider LED Peripheral



8:06

BLE Devices

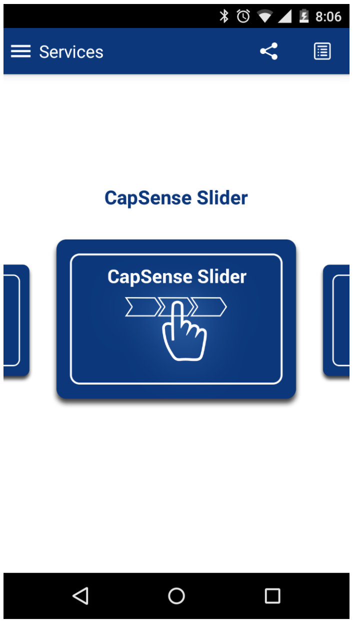
Refreshing, Please wait...

Slider LED   RSSI: Not Paired

00:A0:50:18:23:2C   -61 dBm

7. When connected, the CySmart mobile application will list the profiles supported by the peripherals. Scroll and select the CapSense icon, as shown in Figure 4-26.

Figure 4-26. CapSense Service Page



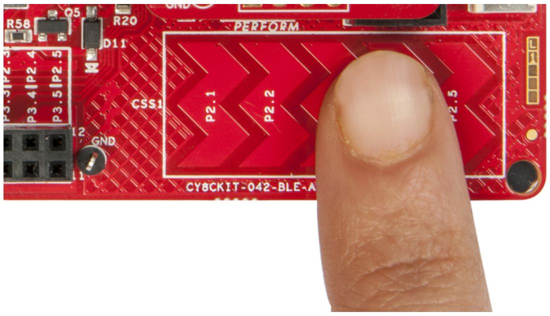
8:06

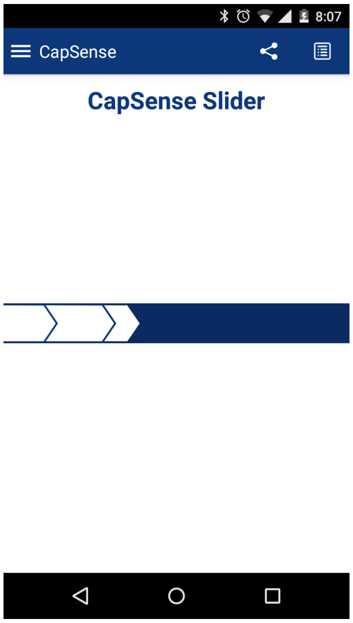
Services

CapSense Slider

8. Swipe your finger on the CapSense slider on the BLE Pioneer Kit and see a similar response on the **CapSense** page in the CySmart application (Figure 4-27).

Figure 4-27. CapSense Slider





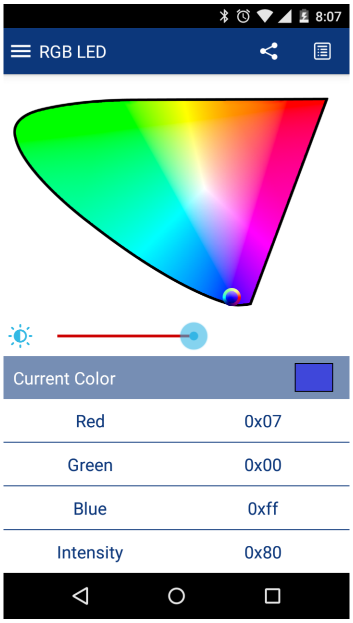
8:07

CapSense

CapSense Slider

9. Press the back button to return to the service selection page. Scroll and tap on the RGB LED service.
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. The RGB LED will be on for 3 seconds before switching off. This is done to conserve power.

Figure 4-28. RGB LED Control with CySmart Mobile Application

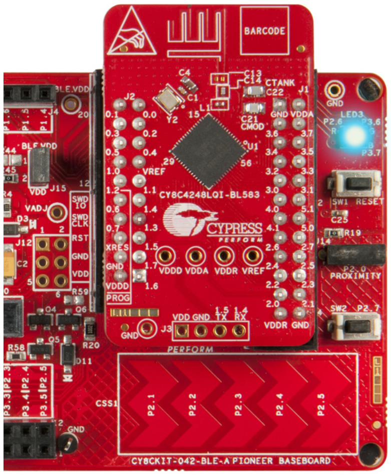


8:07

RGB LED

Current Color

Red	0x07
Green	0x00
Blue	0xff
Intensity	0x80



11. To disconnect from the BLE Pioneer Kit, return to the CySmart mobile application 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 application.

4.3 CapSense Proximity

4.3.1 Project Description

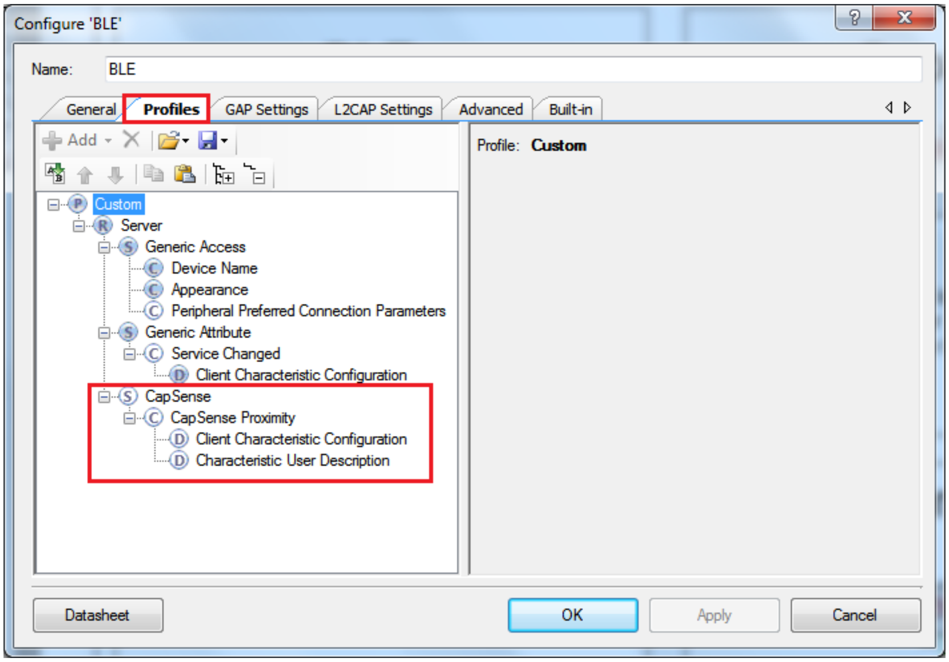
This project demonstrates connectivity between the BLE Pioneer Kit (acting as a Peripheral and GATT server device) and the CySmart Central Emulation Tool or mobile device running the CySmart mobile application (acting as a Central and GATT client device). This project demonstrates the following:

- Advertisement with timeout
- Connection with any Central device
- One custom service
- Data transfer over BLE custom service using notifications
- Low-power mode implementation for coin cell operation

The BLE profile in this project consists of a single BLE custom service, called CapSense. The **CapSense service** consists of a custom characteristic, termed as **CapSense Proximity**. The CapSense proximity characteristic is used to send one byte data, ranging from 0 to 255, as notification to the GATT client device. This data is the difference count read by the CapSense Component on the one-wire proximity sensor (J14) connected on the kit. This characteristic supports notification, which allows the GATT server to send data to the connected GATT client device whenever new data is available.

The properties for the custom attributes are configured in the BLE Component under the Profiles tab, as shown in Figure 4-29.

Figure 4-29. Attributes Configuration in BLE Component for CapSense Proximity



Configure 'BLE'

Name: BLE

Profile: **Custom**

The project consists the following files:

- **main.c/.h**

  These files contain the main function, which is the entry point and execution of the firmware application. They contain function definition for initialization of the system and reading the CapSense proximity data from the CapSense Component.

- **BLEApplications.c/.h**

  These files contain all the macros and function definitions related to BLE communication and operation. They include the event callback function definition that is registered with the BLE Component startup and used to send BLE-related events from the BLE stack to the application layer for processing. These files contain a method to send CapSense notifications to the GATT client device. They update the BLE Connection parameter, which is important for low-power mode usage.

- **HandleLowPower.c/.h**

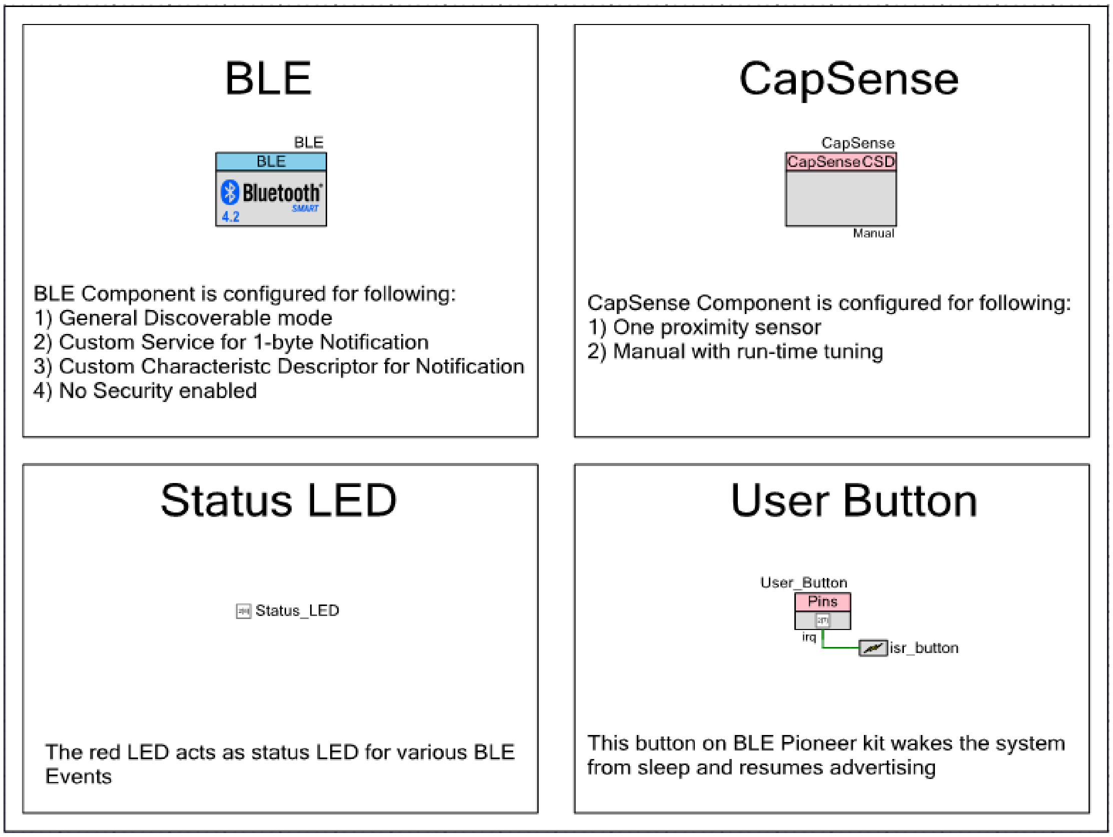
  These files contain the function to handle low-power mode. This function is continuously called in the main loop and is responsible for pushing the BLE hardware block (BLESS) as well as the CPU to Deep Sleep mode as much as possible. The wakeup source is either the BLE hardware block link layer internal timer or the interrupt from the user button press (**SW2**). This allows for very low-power mode implementation and operation using a coin cell.

The red LED is used as the status LED and provides visual confirmation on advertising or connection states. A blinking red 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 4-30. Top Design for PSoC_4_BLE_CapSense_Proximity Project



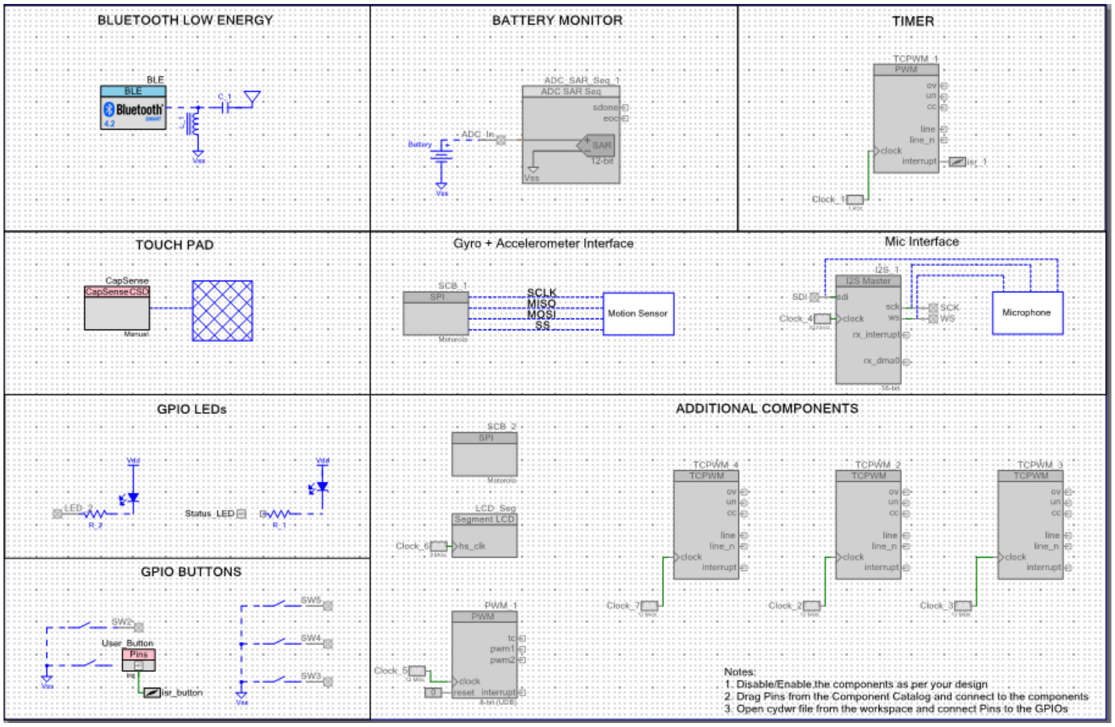
BLE Component is configured for following:
1) General Discoverable mode
2) Custom Service for 1-byte Notification
3) Custom Characteristc Descriptor for Notification
4) No Security enabled

CapSense Component is configured for following:
1) One proximity sensor
2) Manual with run-time tuning

The red LED acts as status LED for various BLE Events

This button on BLE Pioneer kit wakes the system from sleep and resumes advertising

Figure 4-31. Top Design for PRoC_BLE_CapSense_Proximity Project



Notes:
1. Disable/Enable the components as per your design
2. Drag Pins from the Component Catalog and connect to the components
3. Open cydwr file from the workspace and connect Pins to the GPIOs

4.3.2 Hardware Connections

- Ensure that the correct 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 4-32.

  The proximity range is approximately equal to the diameter of the proximity loop. For more information, refer to AN92239 - Proximity Sensing with CapSense.

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

Figure 4-32. Proximity Sensor Connection on BLE Pioneer Kit with PSoC 4 BLE Module

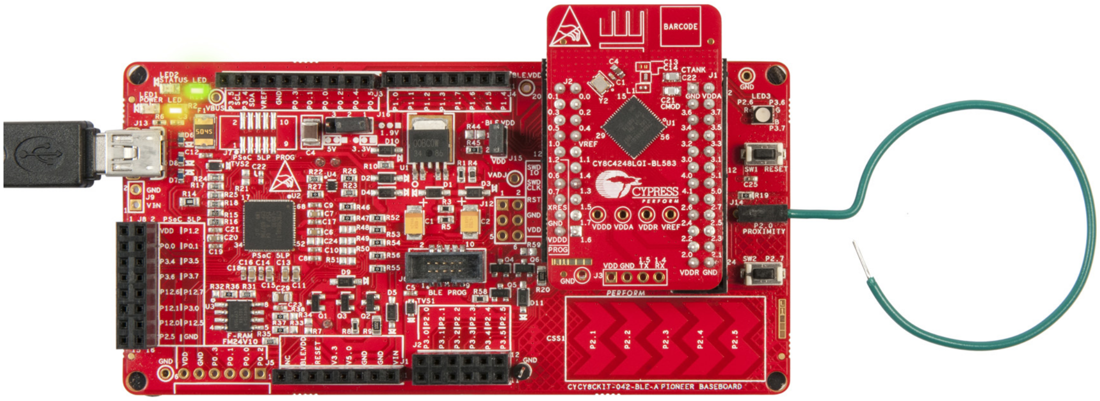
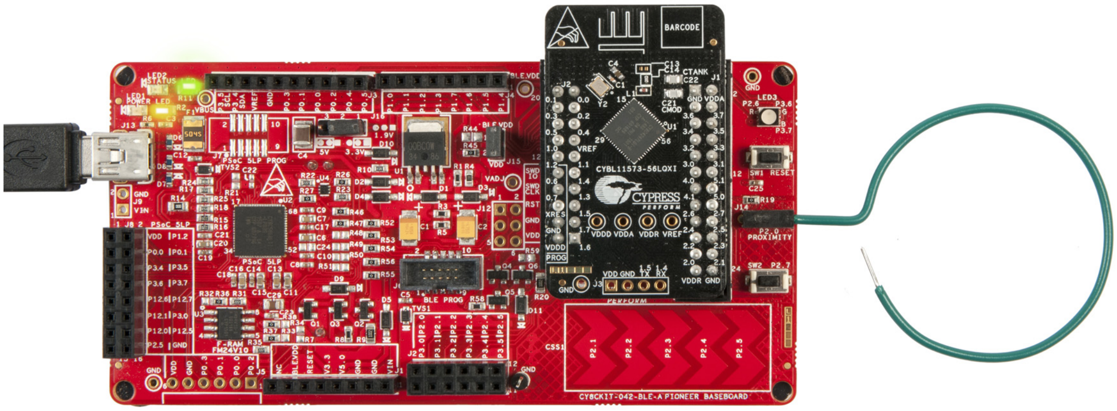
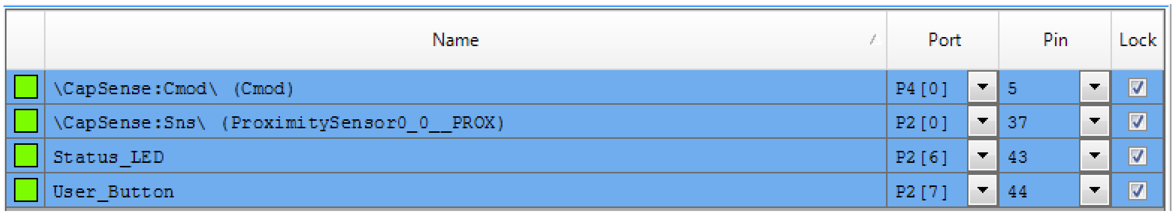


Figure 4-33. Proximity Sensor Connection on BLE Pioneer Kit with PRoC BLE Module



The pin assignment for this project is in **PSoC_4_BLE_CapSense_Proximity.cydwr**/ **PRoC_BLE_CapSense_Proximity.cydwr** in the Workspace Explorer, as shown in Figure 4-34.

Figure 4-34. Pin Selection for CapSense Proximity Project

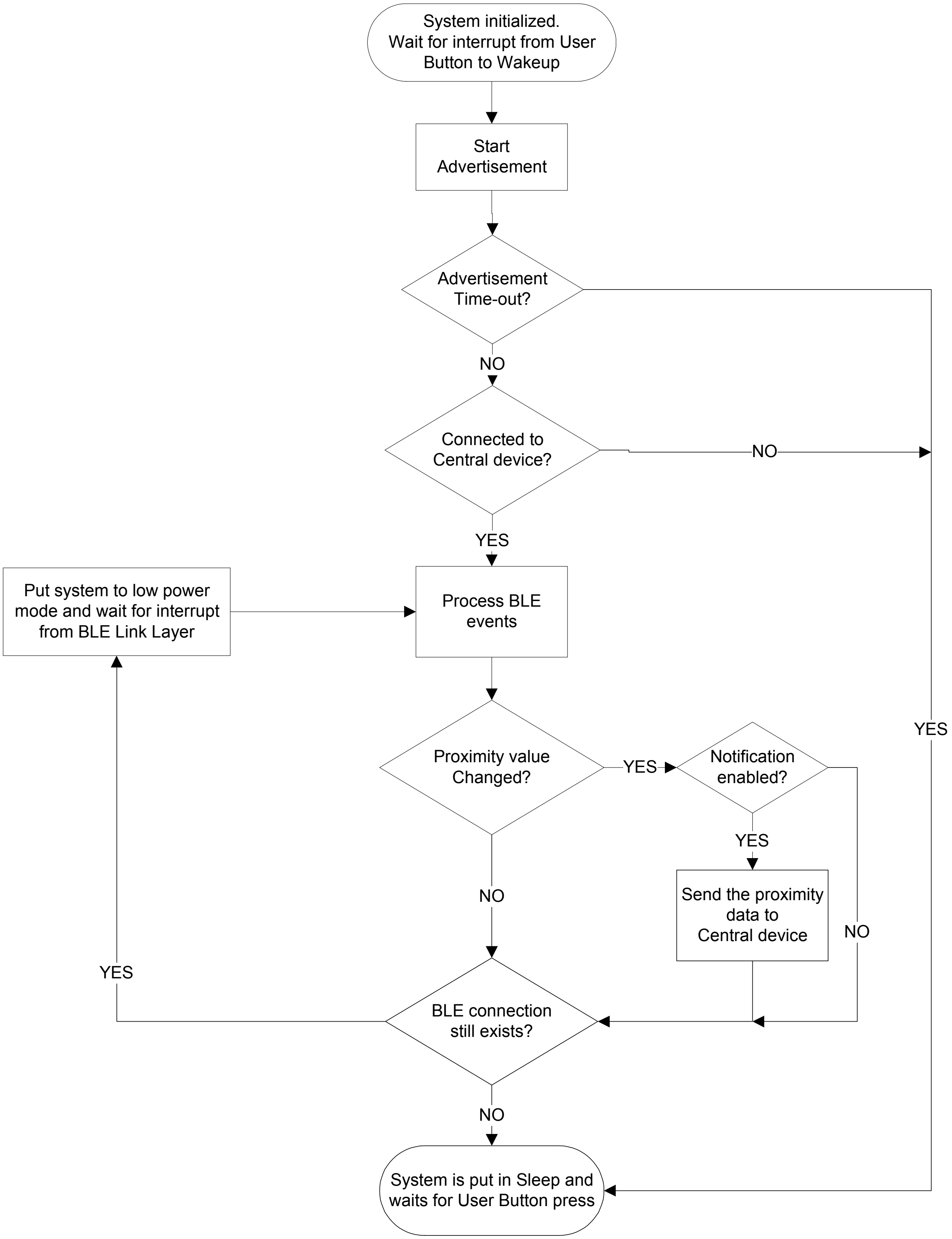


	Name	Port	Pin	Lock
🟩	\CapSense:Cmod\ (Cmod)	P4[0]	5	☑
🟩	\CapSense:Sns\ (ProximitySensor0_0__PROX)	P2[0]	37	☑
🟩	Status_LED	P2[6]	43	☑
🟩	User_Button	P2[7]	44	☑

4.3.3 Flow Chart

Figure 4-35 shows the flow chart of code implemented.

Figure 4-35. CapSense Proximity Project Flow Chart


System initialized.
Wait for interrupt from User Button to Wakeup
Start Advertisement
Advertisement Time-out?
NO
Connected to Central device?
NO
YES
Put system to low power mode and wait for interrupt from BLE Link Layer
Process BLE events
YES
Proximity value Changed?
YES
Notification enabled?
YES
Send the proximity data to Central device
NO
NO
YES
BLE connection still exists?
NO
System is put in Sleep and waits for User Button press

4.3.4 Verify Output

The project can be verified by two methods: using the CySmart Central Emulation Tool and BLE Dongle or using the CySmart iOS/Android app.

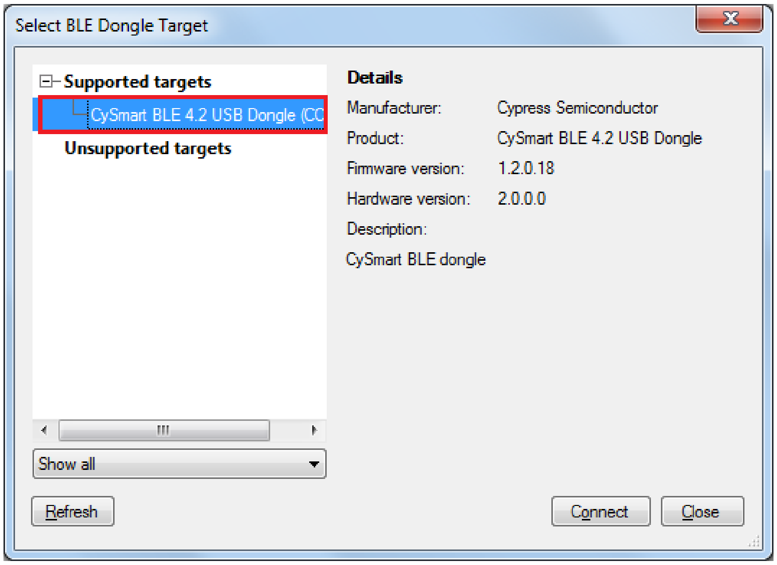
4.3.4.1 CySmart Central Emulation Tool

To verify the CapSense Proximity project using the CySmart Central Emulation Tool, follow these steps:

Note: Refer CySmart Central Emulation tool to learn how to use the tool.

1. Connect the BLE Dongle to one of the USB ports on the computer.
2. Start the CySmart Central Emulation Tool on the computer by going to **Start** > **All Programs** > **Cypress** > **CySmart <version>** > **CySmart <version>**. You will see a list of dongles connected to it. If no dongle is found, click **Refresh**. Select the BLE Dongle and click **Connect**.

Figure 4-36. Connect to BLE Dongle



Select BLE Dongle Target

Supported targets
- CySmart BLE 4.2 USB Dongle (CC

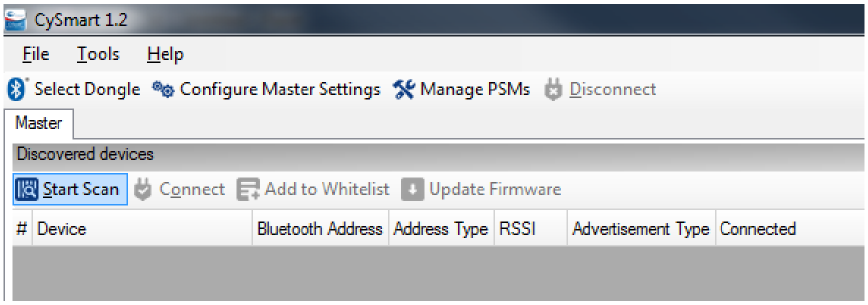
Unsupported targets

Details

Manufacturer: Cypress Semiconductor
Product: CySmart BLE 4.2 USB Dongle
Firmware version: 1.2.0.18
Hardware version: 2.0.0.0
Description:
CySmart BLE dongle

3. Connect a five-inch wire (included in the kit) to the proximity sensor connector J14 and make a loop of it.
4. Power the BLE Pioneer Kit through the USB connector **J13**.
5. Program the BLE Pioneer Kit with the CapSense proximity example project. Follow the steps in Using Example Projects on page 35 to program the device.
6. After programming successfully, press the user button (**SW2**) on the BLE Pioneer kit to start the advertisement. This is indicated by a blinking red LED on the baseboard.
   Note: The project has an advertisement timeout of 30 seconds after which it returns to Deep Sleep mode. Press **SW2** again to restart the advertisement.
7. On the CySmart Central Emulation Tool, click **Start Scan** to see the list of available BLE Peripheral devices.

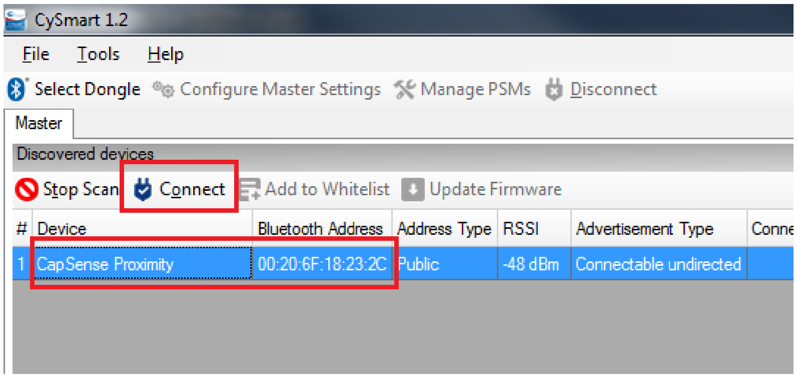
Figure 4-37. Start Scanning



#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected

8. Double-click **CapSense Proximity** to connect or click **Connect** to connect to the BLE Pioneer Kit.

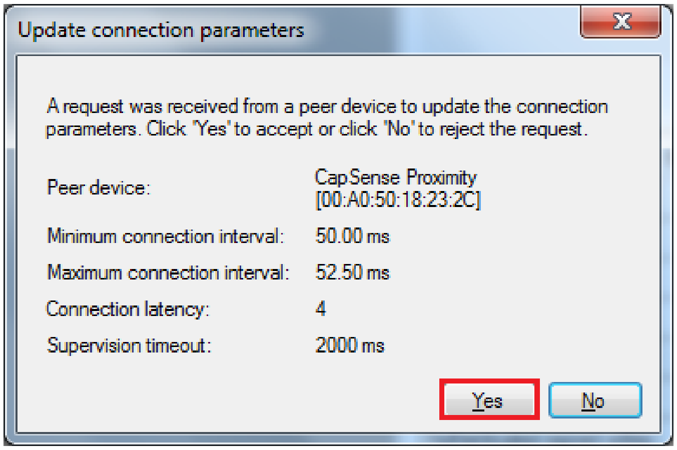
Figure 4-38. Connect to CapSense Proximity Peripheral



#	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Conne
1	CapSense Proximity	00:20:6F:18:23:2C	Public	-48 dBm	Connectable undirected	

9. When connected, the CySmart Central Emulation Tool will display a message for the **Update connection parameters**. Select **Yes**, as shown in Figure 4-39.

Figure 4-39. Update Connection Parameter Option



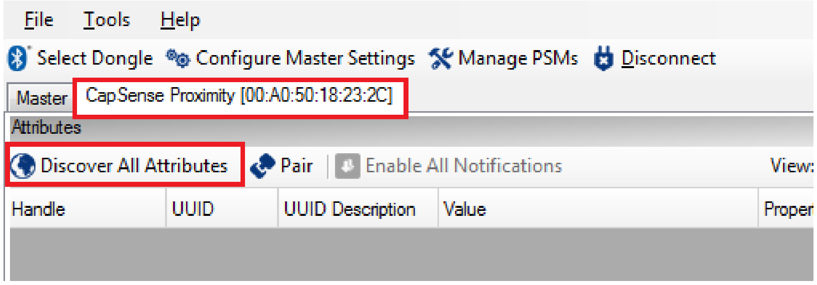
A request was received from a peer device to update the connection parameters. Click 'Yes' to accept or click 'No' to reject the request.

Peer device:	CapSense Proximity [00:A0:50:18:23:2C]
Minimum connection interval:	50.00 ms
Maximum connection interval:	52.50 ms
Connection latency:	4
Supervision timeout:	2000 ms

Note: If you select **No**, the project will still work. However, the current consumption will be higher due to faster connection interval.

10. Click **Discover All Attributes** to find all attributes supported by the Peripheral.

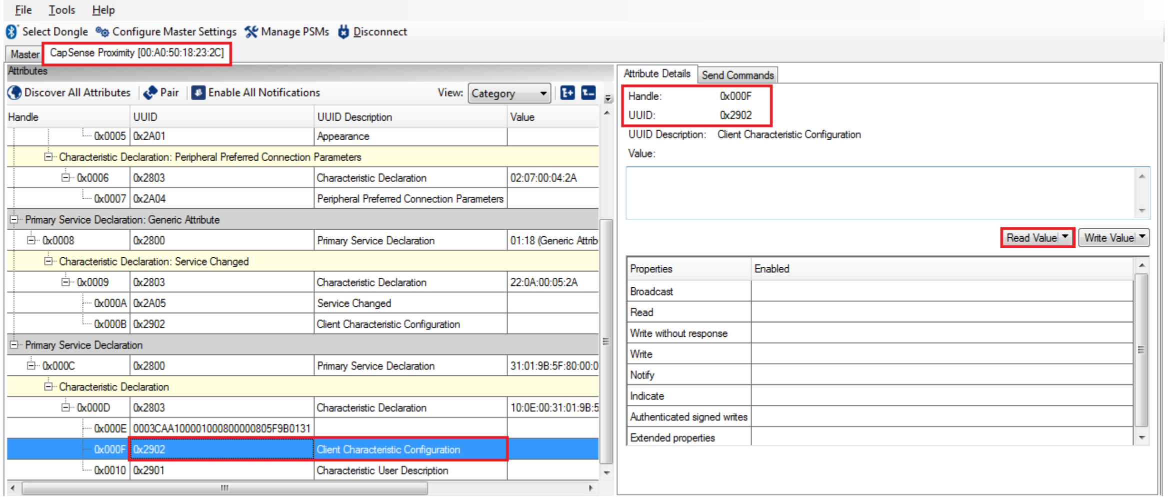
Figure 4-40. Discover All Attributes



Handle	UUID	UUID Description	Value	Proper

11. When all the attributes are listed, locate the **Client Characteristic Configuration** descriptor (UUID 0x2902) under CapSense Proximity characteristic (**UUID 0x0003CAA1-0000-1000-8000-00805F9B0131**). Click **Read Value** to read the existing CCCD value as shown in Figure 4-41.

Figure 4-41. Read CapSense Proximity CCCD



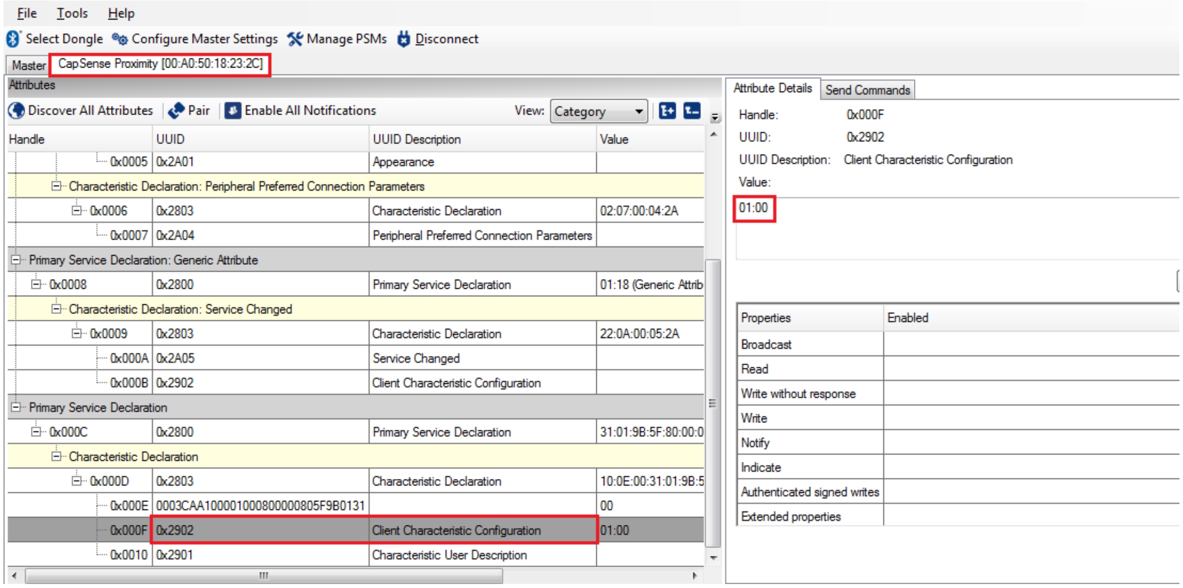
Handle	UUID	UUID Description	Value
0x0005	0x2A01	Appearance	
Characteristic Declaration: Peripheral Preferred Connection Parameters			
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A
0x0007	0x2A04	Peripheral Preferred Connection Parameters	
Primary Service Declaration: Generic Attribute			
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attrib
Characteristic Declaration: Service Changed			
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
Primary Service Declaration			
0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:0
Characteristic Declaration			
0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5
0x000E	0003CAA1000010008000000805F9B0131		
0x000F	0x2902	Client Characteristic Configuration	
0x0010	0x2901	Characteristic User Description	

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

12. Modify the **Value** field to '01:00' and click **Write Value**. This enables the notifications on the CapSense proximity characteristic.

Figure 4-42. Write CCCD to Enable Notifications



CapSense Proximity [00:A0:50:18:23:2C]

View: Category

Handle	UUID	UUID Description	Value
0x0005	0x2A01	Appearance	
Characteristic Declaration: Peripheral Preferred Connection Parameters			
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A
0x0007	0x2A04	Peripheral Preferred Connection Parameters	
Primary Service Declaration: Generic Attribute			
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attrib
Characteristic Declaration: Service Changed			
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A
0x000A	0x2A05	Service Changed	
0x000B	0x2902	Client Characteristic Configuration	
Primary Service Declaration			
0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:0
Characteristic Declaration			
0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5
0x000E	0003CAA100001000800000805F9B0131		00
0x000F	0x2902	Client Characteristic Configuration	01:00
0x0010	0x2901	Characteristic User Description	

Handle: 0x000F
UUID: 0x2902
UUID Description: Client Characteristic Configuration
Value:
01:00

Properties	Enabled
Broadcast	
Read	
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

13. Bring your hand closer to the proximity sensor on the BLE Pioneer Kit, as shown in Figure 4-43 and observe the value changing in the characteristic value field, as shown in Figure 4-44.

Figure 4-43. CapSense Proximity Sensing with PSoC 4 BLE Module

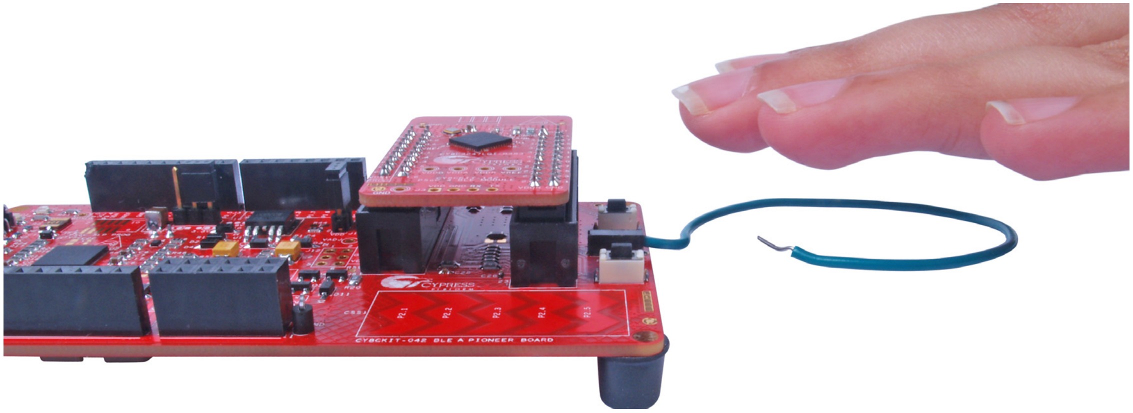
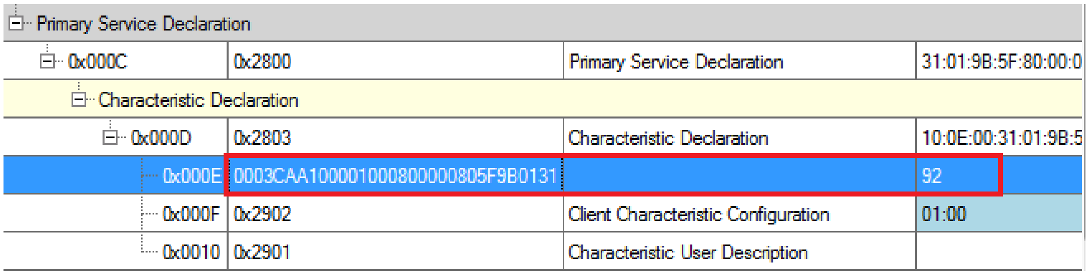


Figure 4-44. CapSense Proximity Notification Received

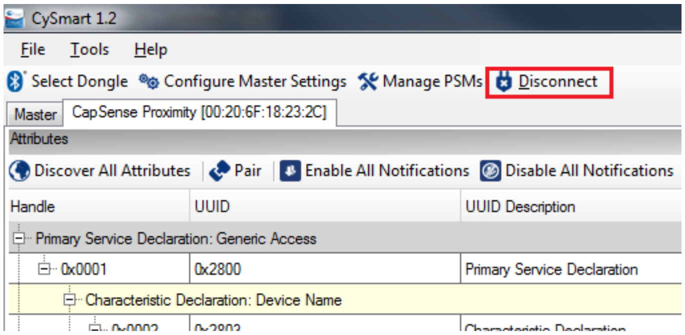


Primary Service Declaration			
0x000C	0x2800	Primary Service Declaration	31:01:9B:5F:80:00:0
Characteristic Declaration			
0x000D	0x2803	Characteristic Declaration	10:0E:00:31:01:9B:5
0x000E	0003CAA100001000800000805F9B0131		92
0x000F	0x2902	Client Characteristic Configuration	01:00
0x0010	0x2901	Characteristic User Description	

14. Modify the **Value** field of the Client Characteristic Configuration descriptor to '00:00' to disable notifications.

15. To disconnect from the device, click **Disconnect**, as shown in Figure 4-45.

Figure 4-45. Disconnect from the Device



Handle	UUID	UUID Description
Primary Service Declaration: Generic Access		
0x0001	0x2800	Primary Service Declaration
Characteristic Declaration: Device Name		
0x0002	0x2803	Characteristic Declaration

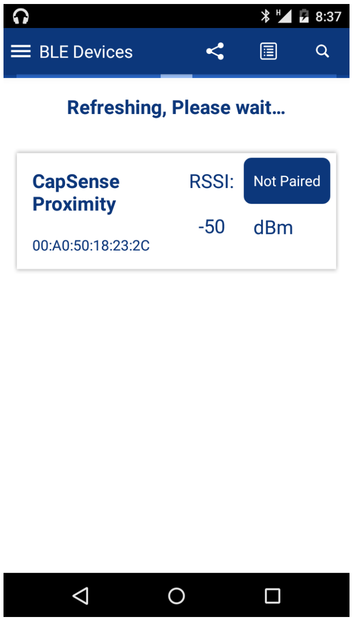
16. Press user button (**SW2**) to wake up from sleep and restart the advertisement for the next connection.

4.3.4.2 *CySmart Mobile Application*

To verify the CapSense Proximity project using the CySmart mobile application (refer CySmart Mobile App webpage), follow these steps:

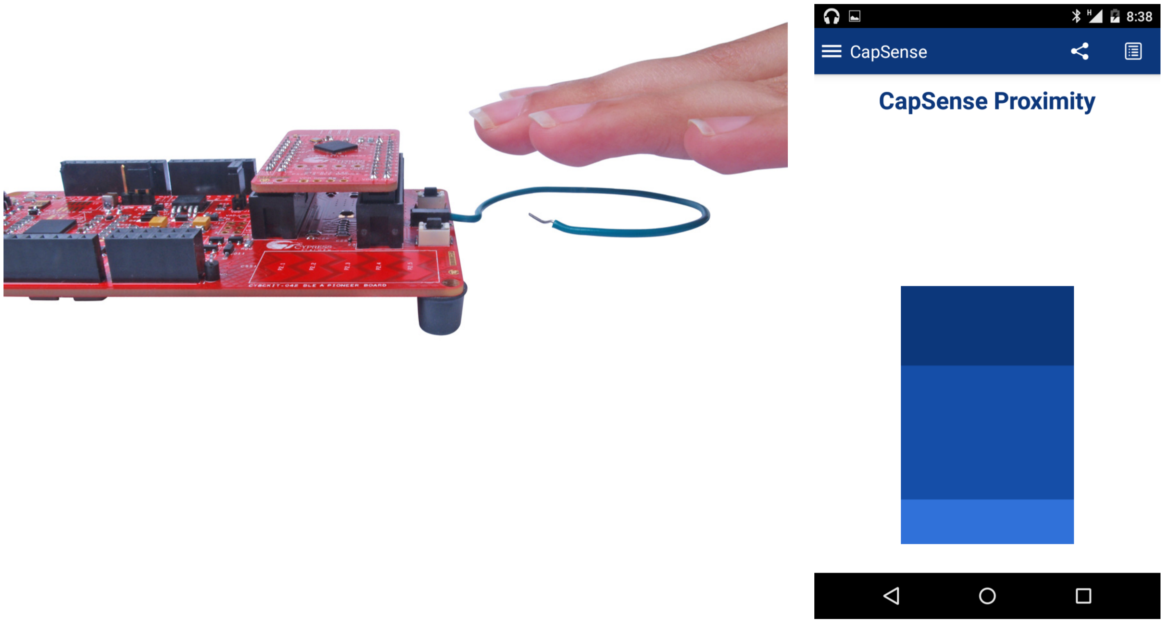
1. Place the desired module on BLE Pioneer Baseboard.
2. Connect the five-inch wire as a loop to the proximity connector J14 on the baseboard.
3. Plug the BLE Pioneer Kit into the computer, using the J13 USB connector.
4. Program the kit with the CapSense proximity example project. Follow steps in Using Example Projects on page 35 to program the device.
5. Press the user button (**SW2**) on the BLE Pioneer Kit to start the advertisement.
6. Open the CySmart mobile application 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 peripherals and list them. Select the **CapSense Proximity** peripheral, as shown in Figure 4-46.

Figure 4-46. Connect to CapSense Proximity Peripheral



8. When connected, the app will list the services supported by the peripherals. Scroll and select the **CapSense Proximity** service.
9. When the CapSense service page opens, bring your hand near the sensor wire on the BLE Pioneer Kit; the proximity level is represented in the app as a bar graph, as shown in Figure 4-47. The darker shade in the graph indicates closer proximity. The app beeps when the graph fills up by 50 percent. Refer to CySmart iOS App User Guide for details.

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



10. To disconnect from the BLE Pioneer Kit, return to the device selection screen on the CySmart mobile application.
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 mobile application.

4.4 BLE Central Mode

4.4.1 Project Description

The BLE projects described above have been functioning as Peripheral devices. This means that the firmware role was set to be a Peripheral and GATT server; another device such as the CySmart Central Emulation Tool or CySmart mobile application will connect to it and collect the data.

This example project demonstrates the Central and GATT client mode where it will scan for a Peripheral device, connect to it, and send commands. In this project, the BLE Pioneer Kit scans and autoconnects to a particular Peripheral device supporting **Immediate Alert Service (IAS)**. Whenever the Peripheral with a predetermined public address is found, a connection request is sent followed by discovering the attributes. When the discovery is over, you can send one of the three alert levels to the Peripheral device over the IAS. This is done by pressing the **SW2** button on the BLE Pioneer Kit and cycling through the alert levels.

The BLE Central project supports low-power mode operation, where the firmware supports BLE Hardware block and CPU Deep Sleep mode whenever possible. The system remains in Deep Sleep when disconnected. Press **SW2** to wake up the system and start scanning (blinking blue LED). The scanning timeout interval is set to 30 seconds. If a particular Peripheral device is found advertising before timeout, a connection is made (blue LED always ON). If no such device is found, then the system stops scanning and returns to Deep Sleep mode (LED OFF). Press **SW2** again to wake the system and restart scanning.

To aid in evaluation, the Peripheral project with the particular public address is provided in the same workspace. This Peripheral project supports IAS and has fixed public address that the Central device will recognize and auto-connect to. The project should be programmed on the BLE Dongle and powered through the USB port of the computer. The received alert levels (No, Mid, and High alert) on the BLE Dongle are represented by different LED status. No Alert is represented by LED OFF, Mid Alert by blinking LED, and High Alert with LED always ON. Upon each successive button press on the BLE Pioneer Kit, the LED state on the BLE Dongle changes in a circular fashion.

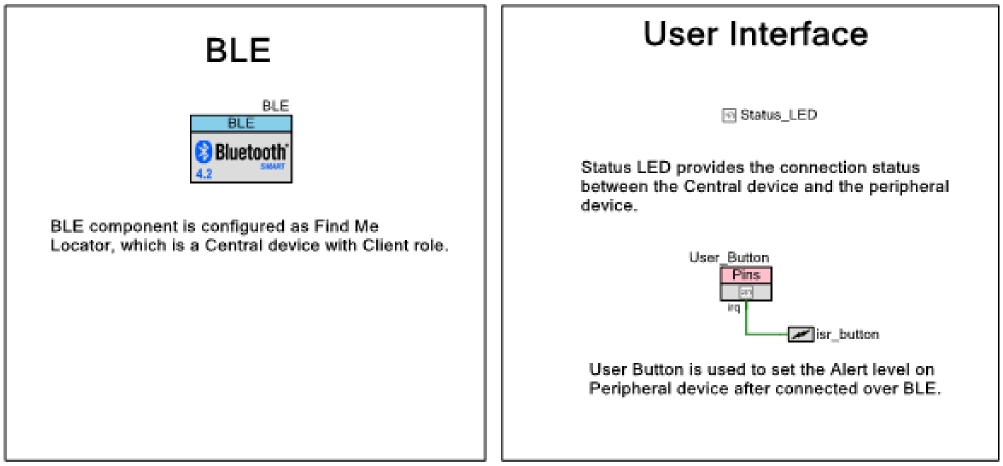
Two projects demonstrate the BLE Central functionality on the two devices:

- **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 BLE Dongle. This project is present in both the PSoC_4_BLE_Central_IAS and the PRoC_BLE_Central_IAS workspace and can be used to program the BLE Dongle separately.

Note: If the BLE Dongle is programmed with the **BLE_Dongle_Peripheral_IAS** example, it will not work with the CySmart PC utility. Reprogram the BLE Dongle with the CySmart firmware according to Updating BLE Dongle for CySmart Central Emulation Tool on page 30 to use the CySmart Central Emulation Tool.

Figure 4-48. PSoC_4_BLE_Central_IAS TopDesign

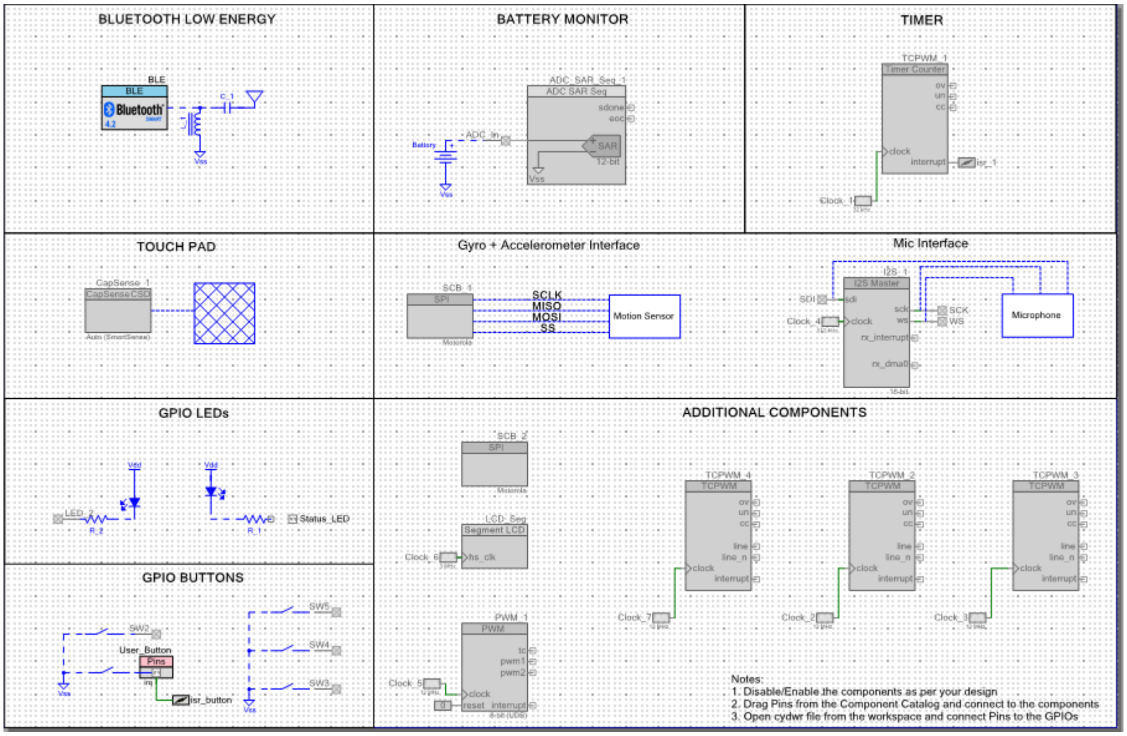


BLE component is configured as Find Me Locator, which is a Central device with Client role.

Status LED provides the connection status between the Central device and the peripheral device.

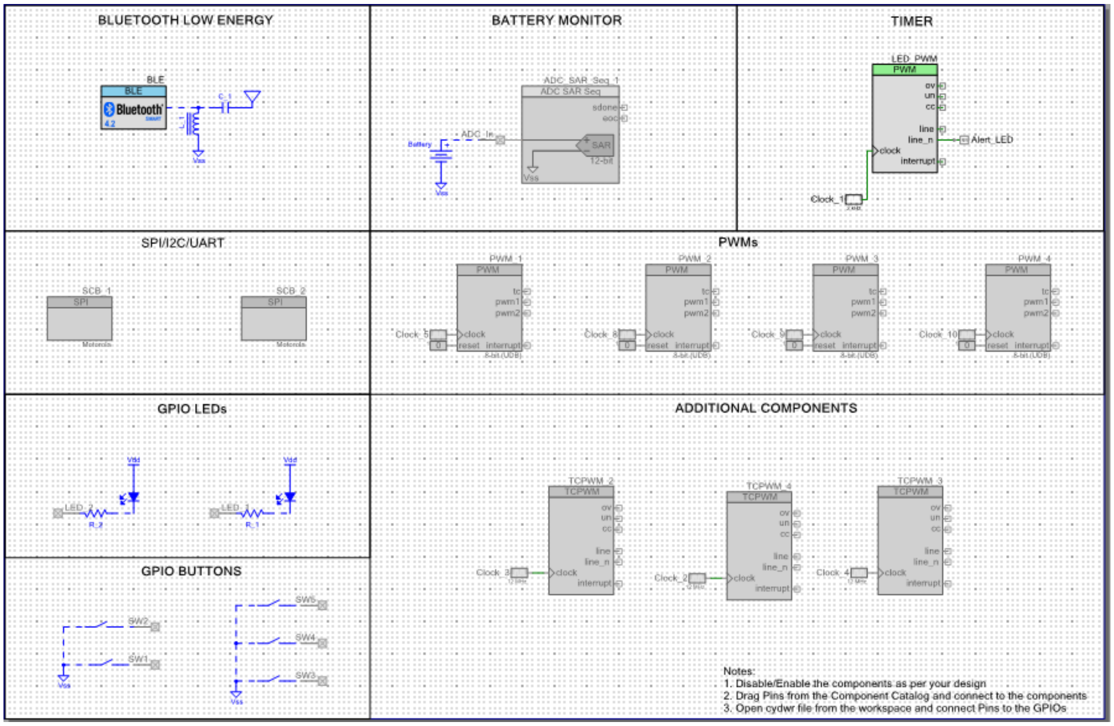
User Button is used to set the Alert level on Peripheral device after connected over BLE.

Figure 4-49. PRoC_BLE_Central_IAS TopDesign



Notes:
1. Disable/Enable the components as per your design
2. Drag Pins from the Component Catalog and connect to the components
3. Open cydwr file from the workspace and connect Pins to the GPIOs

Figure 4-50. BLE_Dongle_Peripheral_IAS TopDesign



Notes:
1. Disable/Enable the components as per your design
2. Drag Pins from the Component Catalog and connect to the components
3. Open cydwr file from the workspace and connect Pins to the GPIOs

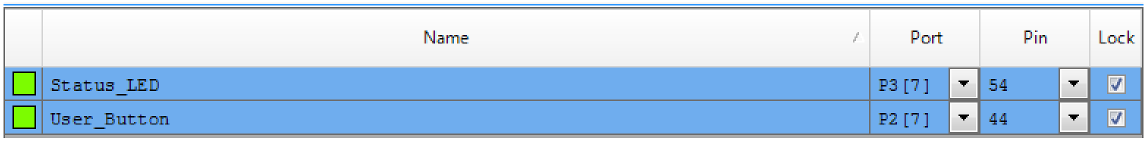
4.4.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the BLE Pioneer Baseboard.

Ensure that the correct module is placed on the BLE Pioneer 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 BLE Dongle with Peripheral mode firmware.

The pin assignment for this project is in **PSoC_4_BLE_Central_IAS.cydwr/ PRoC_BLE_Central_IAS.cydwr** in the Workspace Explorer, as shown in Figure 4-51.

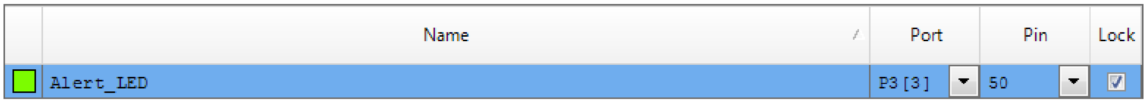
Figure 4-51. Pin Selection for BLE IAS Central Example Project



	Name	Port	Pin	Lock
🟩	Status_LED	P3[7]	54	☑
🟩	User_Button	P2[7]	44	☑

Similarly, the pin assignment for the BLE Dongle Peripheral project is in **BLE_Dongle_Peripheral_IAS.cydwr** in the Workspace Explorer as shown in Figure 4-52.

Figure 4-52. Pin Selection for BLE IAS Peripheral Example Project



	Name	Port	Pin	Lock
🟩	Alert_LED	P3[3]	50	☑

4.4.3 Flow Chart

Figure 4-53 shows the flow chart for the IAS GATT client mode example project.

Figure 4-53. IAS GATT Client Mode Flow Chart

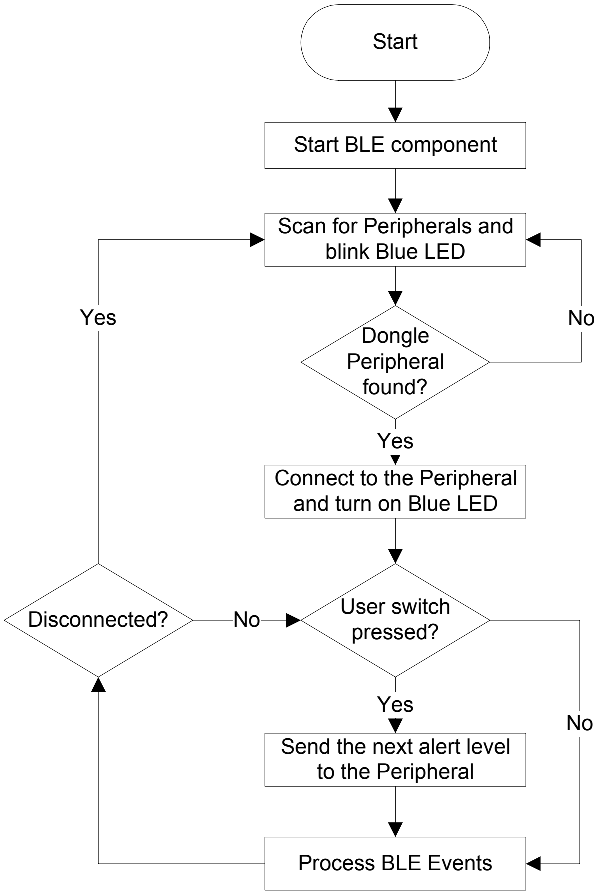
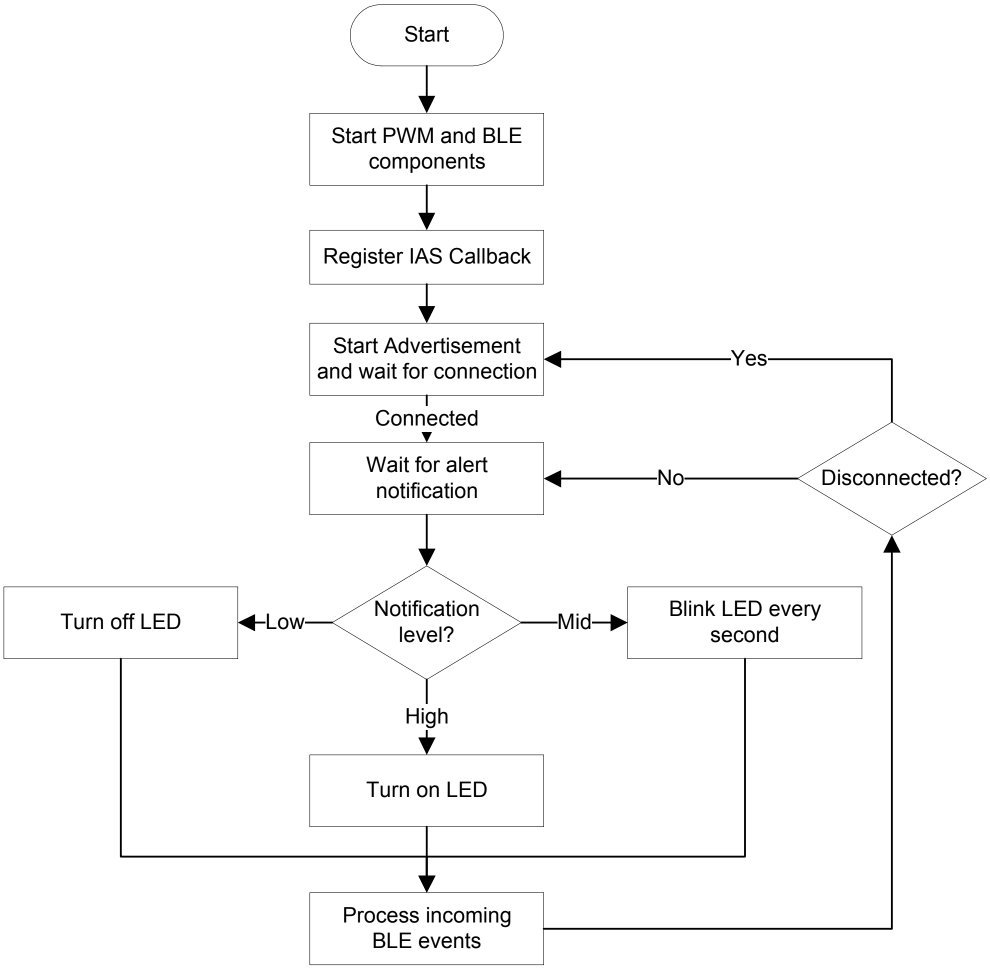

Start
Start BLE component
Scan for Peripherals and blink Blue LED
Yes
No
Dongle Peripheral found?
Yes
Connect to the Peripheral and turn on Blue LED
Disconnected?
No
User switch pressed?
Yes
No
Send the next alert level to the Peripheral
Process BLE Events

Figure 4-54 shows the flow chart for the IAS GATT server mode example project.

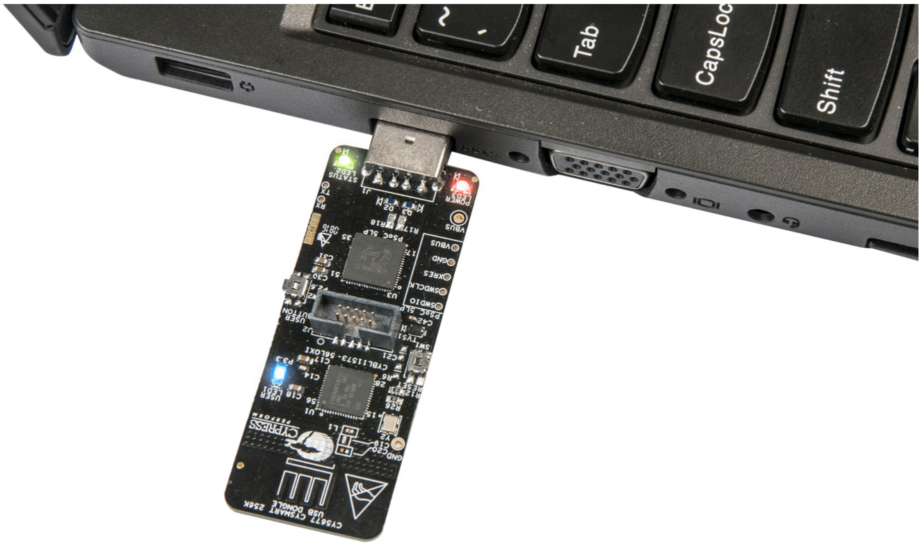
Figure 4-54. IAS GATT Server Mode Flow Chart



4.4.4 Verify Output

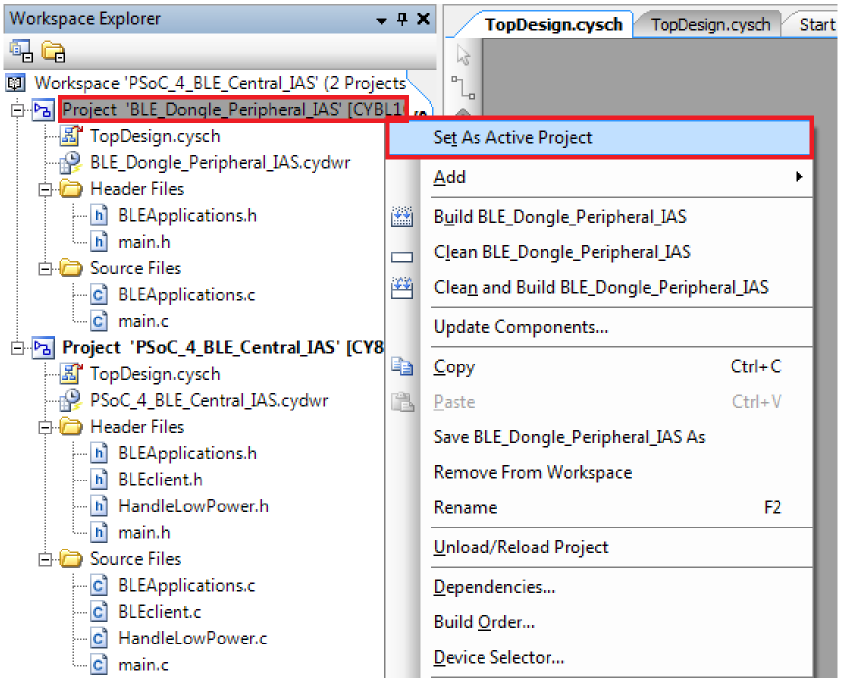
1. Connect the BLE Dongle to one of the USB ports on the computer.

Figure 4-55. Connect BLE Dongle to USB Port



2. In the PSoC Creator Workspace Explorer, right-click the **BLE_Dongle_Peripheral_IAS** project and select **Set As Active Project**, as shown in Figure 4-56.

Figure 4-56. Set Dongle Peripheral Project as Active

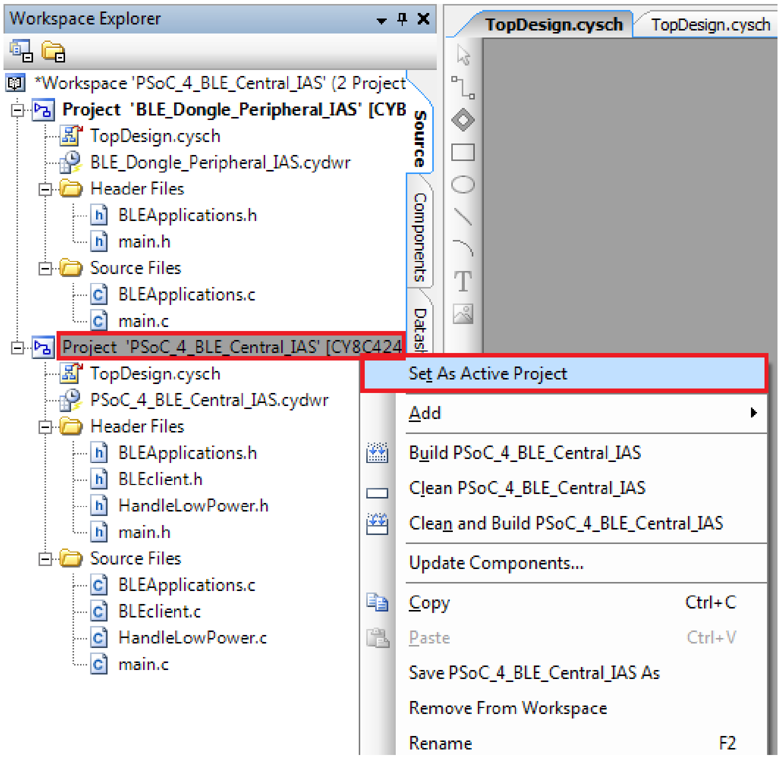


3. Program the BLE Dongle with the BLE_Dongle_Peripheral_IAS project described in Using Example Projects on page 35.

   Note: Do not update the public device address (inside the BLE Component) for the **BLE_Dongle_Peripheral_IAS** example project. Changing the **BLE_Dongle_Peripheral_IAS** example project public address will lead to no connection with the BLE Central device on the BLE Pioneer kit.

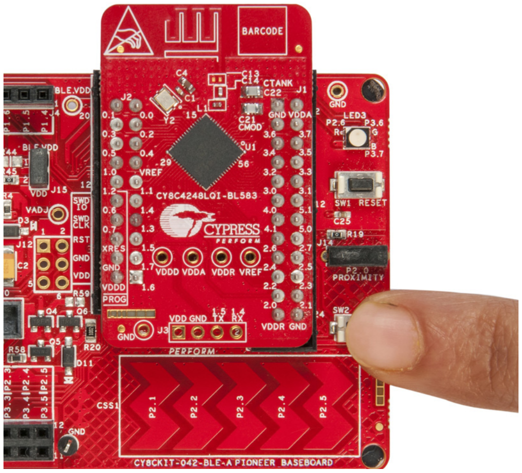
4. Power the BLE Pioneer Kit through USB connector **J13**.
5. In the Workspace Explorer, right-click the **PSoC_4_BLE_Central_IAS** project and select **Set As Active Project**, as shown in Figure 4-57.

Figure 4-57. Set Central IAS Project as Active



6. Program the BLE Pioneer Kit with either the **PSoC_4_BLE_Central_IAS** or the **PRoC_BLE_Central_IAS** project, depending on the module placed on the BLE Pioneer Kit.
7. Press the **SW2** button on the BLE Pioneer Kit to wake the system and start scanning. Scanning is indicated by a blinking LED.
8. Wait for the BLE connection between the BLE Dongle and the BLE Pioneer Kit. 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 BLE Dongle Peripheral device and has started the connection procedure.
   c. The blue LED remains on, representing the connected mode. This mode indicates that the Peripheral device is connected and the application can now send alert levels.
9. Press the **SW2** button on the BLE Pioneer Kit to send the next alert level to the BLE Dongle. The alert level will rotate from No Alert to Mid Alert to High Alert.

Figure 4-58. User Button on BLE Pioneer Kit with PSoC 4 BLE Module



10. Check if the LED behavior changes for each alert notification on the BLE Dongle according to the following table:

Alert Level	LED State
No Alert	LED Off
Mild Alert	LED Blinking
High Alert	LED On

Note: To revert the CySmart functionality to the dongle, program the dongle hex file, as described in Updating BLE Dongle for CySmart Central Emulation Tool on page 30.

4.5 Eddystone

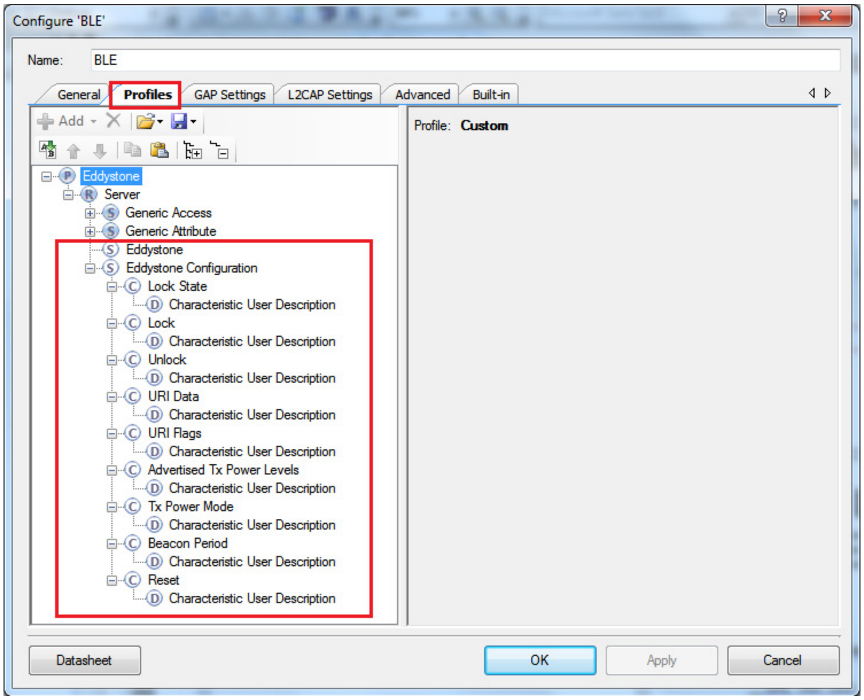
4.5.1 Project Description

This project demonstrates a BLE beacon based on Google's Eddystone™ protocol on the BLE Pioneer Kit. A beacon is a wireless device that broadcasts data (such as temperature) over a periodic radio signal from a known location. BLE-based beacons use the BLE advertisement packets to broadcast data. This project demonstrates the following:

- Beacon (non-connectable advertisement)
- Beacon configuration (connectable advertisement)

Google's Eddystone is a protocol that defines BLE advertisement data formats for beacons. In Eddystone parlance, the advertisement data is called a frame and the advertisement data format is called a frame type. The protocol supports multiple frame types that may be used individually or in combinations to create beacons for a variety of applications.

Figure 4-59. Characteristic Configuration in BLE Component for Eddystone Beacon


Configure 'BLE'
Name: BLE
General
Profiles
GAP Settings
L2CAP Settings
Advanced
Built-in
Add
Eddystone
Server
Generic Access
Generic Attribute
Eddystone
Eddystone Configuration
Lock State
Characteristic User Description
Lock
Characteristic User Description
Unlock
Characteristic User Description
URI Data
Characteristic User Description
URI Flags
Characteristic User Description
Advertised Tx Power Levels
Characteristic User Description
Tx Power Mode
Characteristic User Description
Beacon Period
Characteristic User Description
Reset
Characteristic User Description
Profile: Custom
Datasheet
OK
Apply
Cancel

The BLE profile in this project consists of two BLE custom services, **Eddystone** and **Eddystone Configuration** (see Figure 4-59). The project broadcasts one of the following beacon frames in a non-connectable advertisement packet.

- **Eddystone-UID** frame broadcasts a unique 16-byte Beacon ID composed of a 10-byte namespace and a 6-byte instance. The Beacon ID may be useful in mapping a device to a record in external storage. The namespace portion of the ID may be used to group a particular set of beacons, while the instance ID identifies individual devices in the group. The division of the ID into namespace and instance components may also be used to optimize BLE scanning strategies, for example, by filtering only on the namespace. More details are available here.
- **Eddystone-URL** frame broadcasts a URL using a compressed encoding format to fit more within the limited advertisement packet. More details are available here.
- **Eddystone-TLM** frame broadcasts telemetry information about the beacon itself such as battery voltage, device temperature, and counts of broadcast packets. More details are available here.

The Eddystone Configuration Service consists of nine characteristics as follows.

- **Lock State** – Read returns true if the device is locked.
- **Lock** – Locks the beacon and sets the single-use lock-code.
- **Unlock** – Unlocks the beacon and clears the single-use lock-code.
- **URI Data** – Reads/writes the URI.
- **URI Flags** – Reads/writes the flags.
- **Advertised Tx Power Levels** – Reads/writes the Advertised Power Levels array.
- **Tx Power Mode** – Reads/writes the TX Power Mode.
- **Beacon Period** – The period in milliseconds that an Eddystone-URL packet is transmitted.
- **Reset** – Resets to default values.

More details on the Eddystone Configuration service can be found here.

The project consists of the following files:

- **main.c/.h**

  These files contain the main function, which is the entry point and execution of the firmware application. They contain function definitions for initialization and handling low power mode of the system. By default on power-on-reset, the system will do a non-connectable advertisement of URL/ URI and TLM packets. When the **SW2** button is pressed, the device will enter connectable mode and various characteristics can be configured.
- **Eddystone.c/.h**

  These files contain the functions required to implement the Eddystone protocol including advertisement packet creation (for UID/URL and TLM), advertisement scheduling, and configuration read/write. The battery and temperature data are also updated before the TLM packets are constructed and advertised.
- **WatchdogTimer.c/.h**

  These files contain functions that provide necessary timing for the operation of the system.
- **Battery.c/.h**

  These files contain a function that will measure the battery voltage of the kit. This information is advertised as part of the TLM packets.
- **Temperature.c/.h**

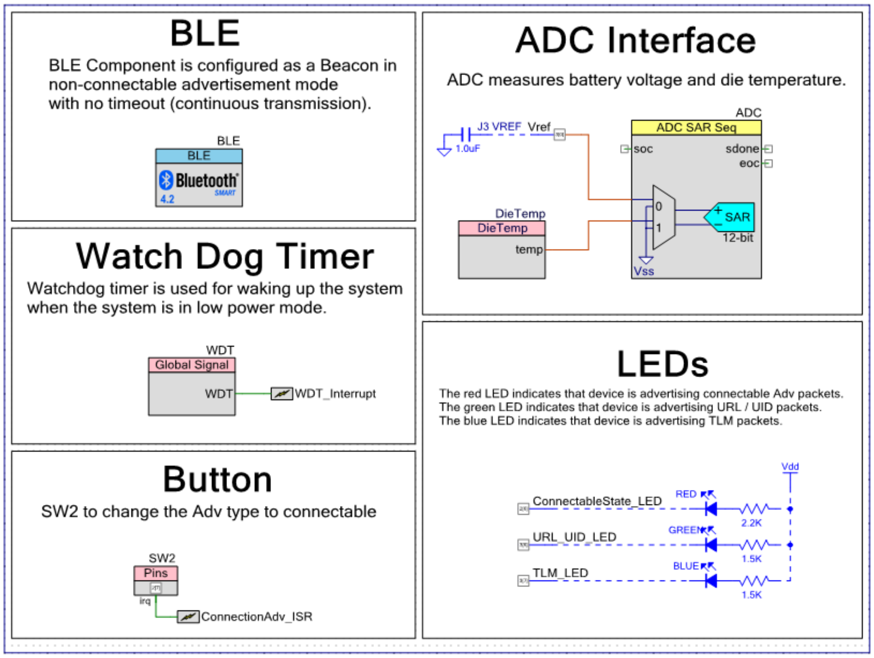
  These files contain a function that will measure the die temperature of the PSoC or PRoC device. An external sensor may be attached to measure ambient temperature. This information is advertised as part of the TLM packets.

The green LED indicates UID/URL packet transmissions, the blue LED indicates TLM packet transmissions, and the red LED indicates that the device is in connectable mode. In this mode, when connected, the beacon configuration can be modified.

Two projects demonstrate this functionality on two different devices:

- PSoC_4_BLE_ Eddystone works with the PSoC 4 BLE Module
- PRoC_BLE_Eddystone works with the PRoC BLE Module

Figure 4-60. Top Design for PSoC_4_BLE_Eddystone and PRoC_BLE_Eddystone Project



BLE

BLE Component is configured as a Beacon in non-connectable advertisement mode with no timeout (continuous transmission).

ADC Interface

ADC measures battery voltage and die temperature.

Watch Dog Timer

Watchdog timer is used for waking up the system when the system is in low power mode.

LEDs

The red LED indicates that device is advertising connectable Adv packets.
The green LED indicates that device is advertising URL / UID packets.
The blue LED indicates that device is advertising TLM packets.

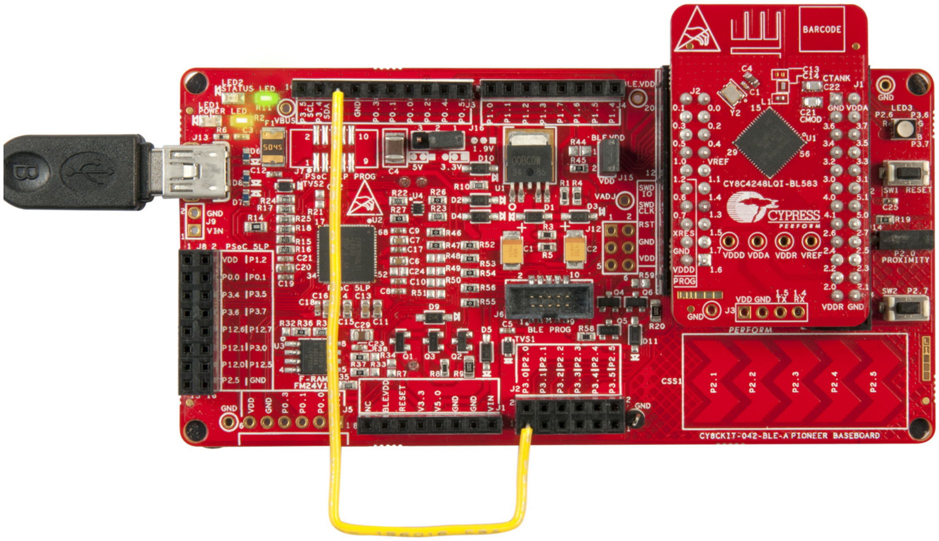
Button

SW2 to change the Adv type to connectable

4.5.2 Hardware Connection

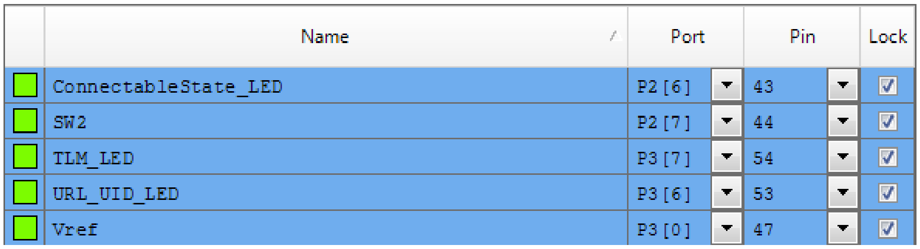
- Ensure that the correct module is placed on the baseboard corresponding to the project being used. PSoC_4_BLE_Eddystone works with the PSoC 4 BLE Module. PRoC_BLE_Eddystone works with the PRoC BLE Module.
- Connect a wire (provided as part of this kit) between VREF (connector J3) and P3.0 (connector J2) on the baseboard (see Figure 4-61).

Figure 4-61. VREF (J3) and P3.0 (J2) Connectors on BLE Pioneer Kit with PSoC 4 BLE Module



The pin assignment for this project is in *PSoC_4_BLE_Eddystone.cydwr* or *PRoC_BLE_Eddystone.cydwr* in the Workspace Explorer, as shown in Figure 4-62.

Figure 4-62. Pin Selection for Eddystone Project

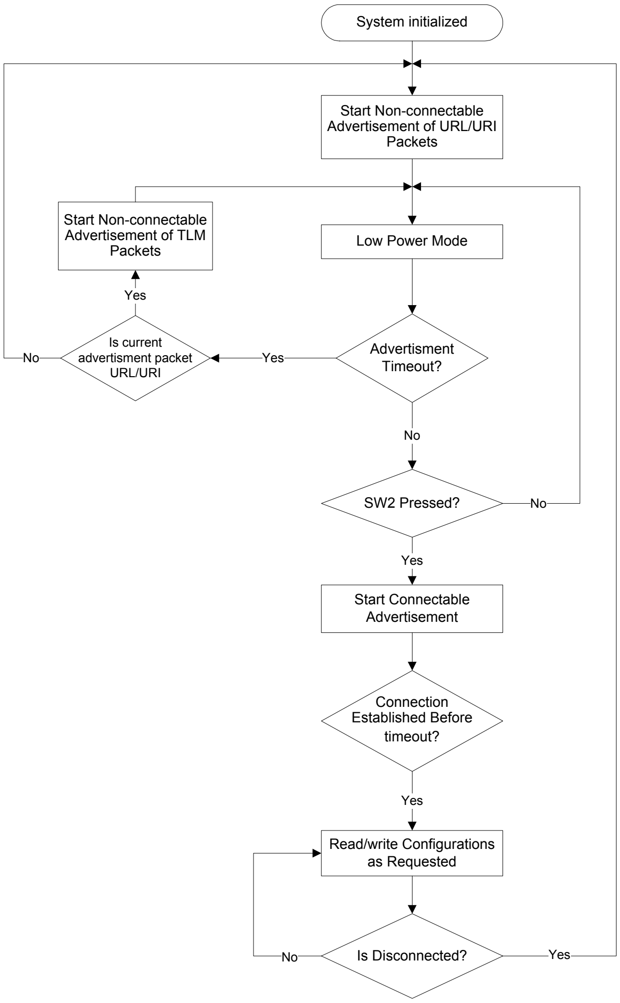


	Name	Port	Pin	Lock
🟩	ConnectableState_LED	P2[6]	43	☑
🟩	SW2	P2[7]	44	☑
🟩	TLM_LED	P3[7]	54	☑
🟩	URL_UID_LED	P3[6]	53	☑
🟩	Vref	P3[0]	47	☑

4.5.3 Flow Chart

Figure 4-63 shows the flow chart of code implemented.

Figure 4-63. Eddystone Project Flow Chart


System initialized
Start Non-connectable Advertisement of URL/URI Packets
Start Non-connectable Advertisement of TLM Packets
Low Power Mode
Yes
No
Is current advertisment packet URL/URI
Yes
Advertisment Timeout?
No
SW2 Pressed?
No
Yes
Start Connectable Advertisement
Connection Established Before timeout?
Yes
Read/write Configurations as Requested
No
Is Disconnected?
Yes

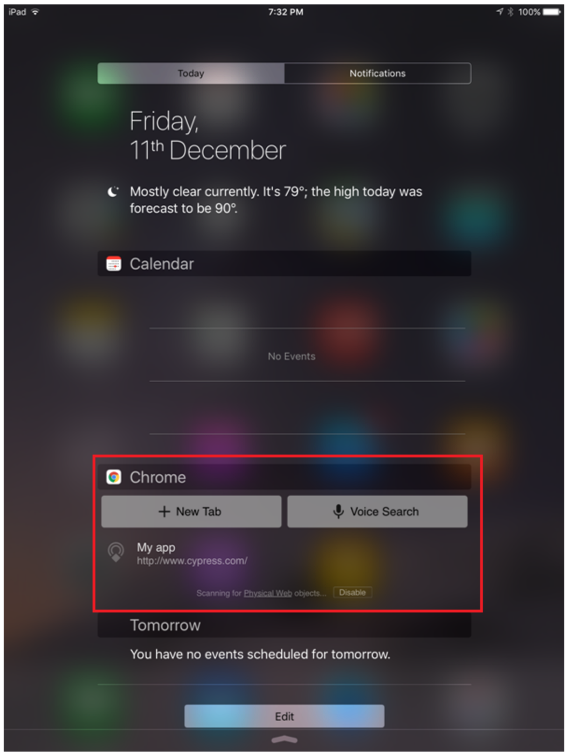
4.5.4 Verify Output

The project can be verified using an Android smart phone or an iOS device (iPhone or iPad). Before proceeding further make sure that a beacon application (such as Locate Beacon for Android or Chrome browser for iOS) and the CySmart app are installed on the smart phone/device. Also make sure that Bluetooth is turned on in the device, and you have a working internet connection.

To verify the Eddystone project in iOS, using Chrome for iOS browser, follow these steps. Before you start, make sure the Chrome browser for iOS is installed in your iOS device and the Chrome widget is added to the widgets view.

1. Place the module on the BLE Pioneer Kit, depending on the project chosen.
2. Power the BLE Pioneer Kit through the USB connector J13.
3. Program the BLE Pioneer Kit with the Eddystone example project. Follow steps in Using Example Projects on page 35 to program the device.
4. After programming successfully, the firmware starts the non-connectable advertisement of UID/URL and TLM packets. Advertisement of UID/URL packets is indicated by the green LED and TLM packets is indicated by blue LED on the baseboard.
5. Pull down the notification area. You should be able to see the link in the notification area as shown in Figure 4-64 on page 76

Figure 4-64. iOS Notification Shade Showing the Beacon Advertised Web Link



Friday,
11th December

Mostly clear currently. It's 79°; the high today was forecast to be 90°.

No Events

My app
http://www.cypress.com/

Scanning for Physical Web objects...

You have no events scheduled for tomorrow.

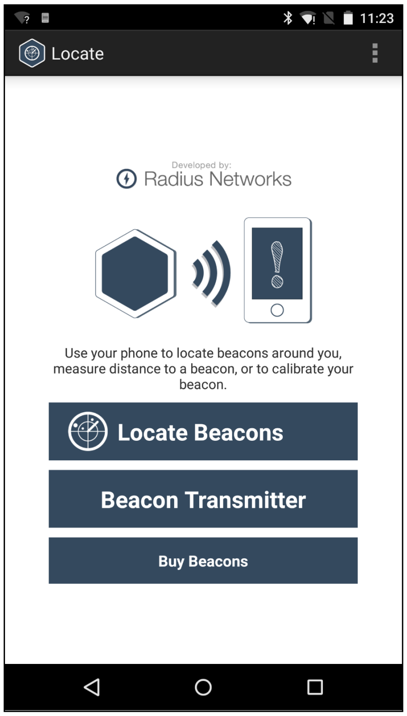
6. Touching the link will take you to the website.

To verify the Eddystone project in Android devices using the Locate Beacon application, follow these steps.

1. Place the module on the BLE Pioneer Kit, depending on the project chosen.
2. Power the BLE Pioneer Kit through the USB connector J13.
3. Program the BLE Pioneer Kit with the Eddystone example project. Follow steps in Using Example Projects on page 35 to program the device.
4. After programming successfully, the firmware starts the non-connectable advertisement of UID/URL and TLM packets. Advertisement of UID/URL packets is indicated by the green LED and TLM packets is indicated by blue LED on the baseboard.
5. Launch the **Locate** application on the smart phone/device.

6. Click the **Locate Beacons** button to start scanning for beacons (see Figure 4-65).

Figure 4-65. Locate Beacon App Home Screen



11:23

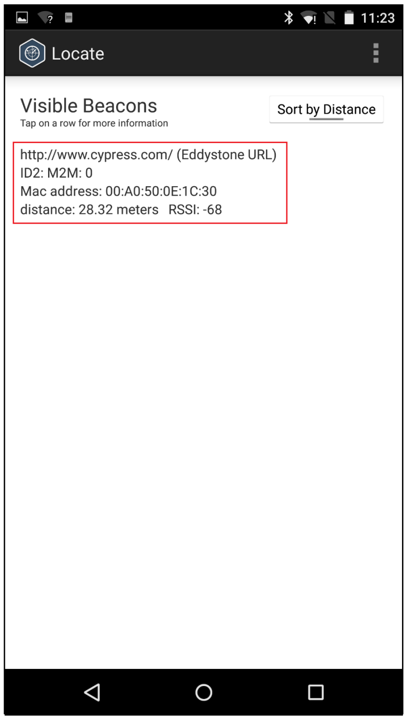
Locate

Developed by:
Radius Networks

Use your phone to locate beacons around you, measure distance to a beacon, or to calibrate your beacon.

7. In the next screen a list of **Visible Beacons** will be displayed. Click the desired beacon (see Figure 4-66) to display its status and information (see Figure 4-67).

Figure 4-66. Locate Beacon App Showing Discovered Beacons



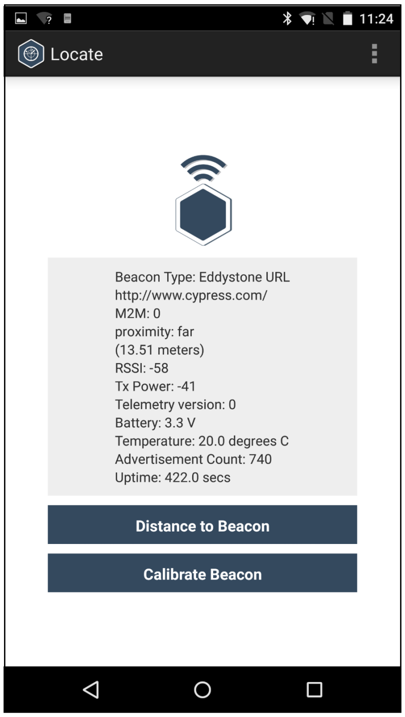
11:23

Locate

Visible Beacons
Tap on a row for more information

http://www.cypress.com/ (Eddystone URL)
ID2: M2M: 0
Mac address: 00:A0:50:0E:1C:30
distance: 28.32 meters RSSI: -68

Figure 4-67. Locate Beacon Showing Details about Selected Beacon

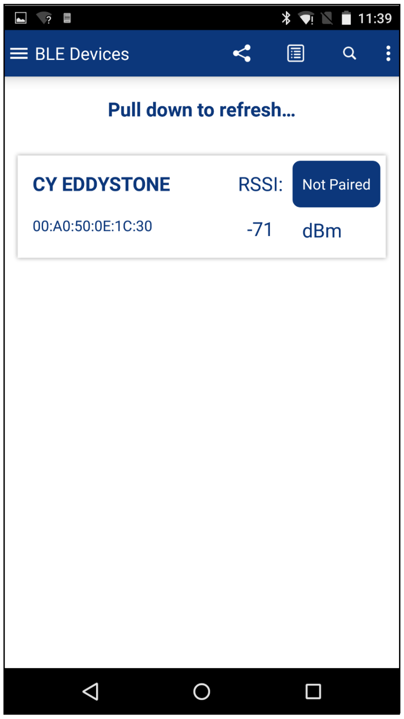


Beacon Type: Eddystone URL
http://www.cypress.com/
M2M: 0
proximity: far
(13.51 meters)
RSSI: -58
Tx Power: -41
Telemetry version: 0
Battery: 3.3 V
Temperature: 20.0 degrees C
Advertisement Count: 740
Uptime: 422.0 secs

To read and write characteristics (using an Android or iOS device, and CySmart app) of the beacon follow these steps.

1. Press **SW2** on the BLE Pioneer Kit baseboard to start the connectable advertisement. The advertisement state is indicated by the red LED.
2. Launch the CySmart app on your Android smart phone/device.
3. Connect to the **CY Eddystone** device from the list by clicking on it (see Figure 4-68).

Figure 4-68. CySmart App Showing the CY Eddystone Beacon



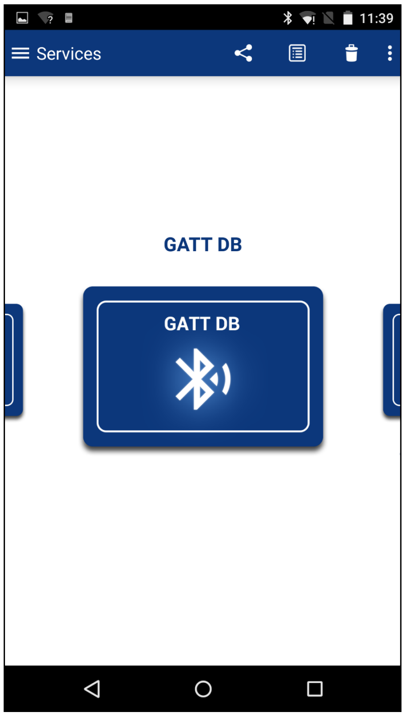
Pull down to refresh...

CY EDDYSTONE RSSI: -71 dBm

00:A0:50:0E:1C:30

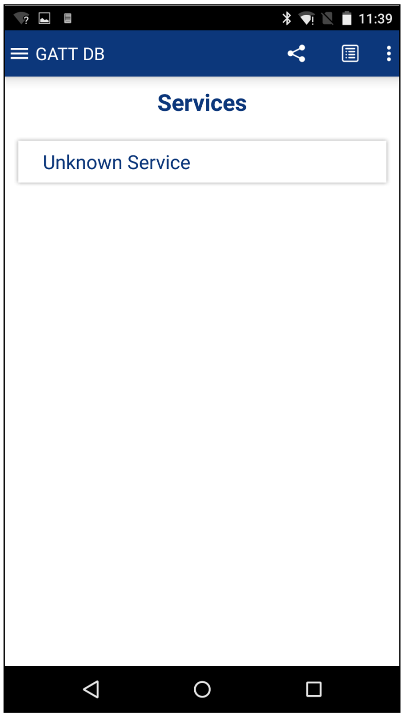
4. Click the **GATT DB** menu option (see Figure 4-69).

Figure 4-69. CySmart Showing GATT DB Option for CY Eddystone Device



5. Click on **Unknown Service** (see Figure 4-70) to see all available characteristics.

Figure 4-70. CySmart Showing Unknown Service in GATT DB



6. Click on any of the characteristics to read and modify it. For reference on Eddystone Configuration characteristics, refer to the Google Eddystone page. To understand more about the CySmart app, refer to the CySmart user guide.
7. Disconnect from the device to go back to the beacon mode.

4.6 BLE Dongle and LED Control

4.6.1 Project Description

This firmware supports the CySmart debug tool (refer CySmart Central Emulation Tool) by acting as the BLE host emulator. This is the default firmware that comes in the BLE Dongle shipped with the kit.

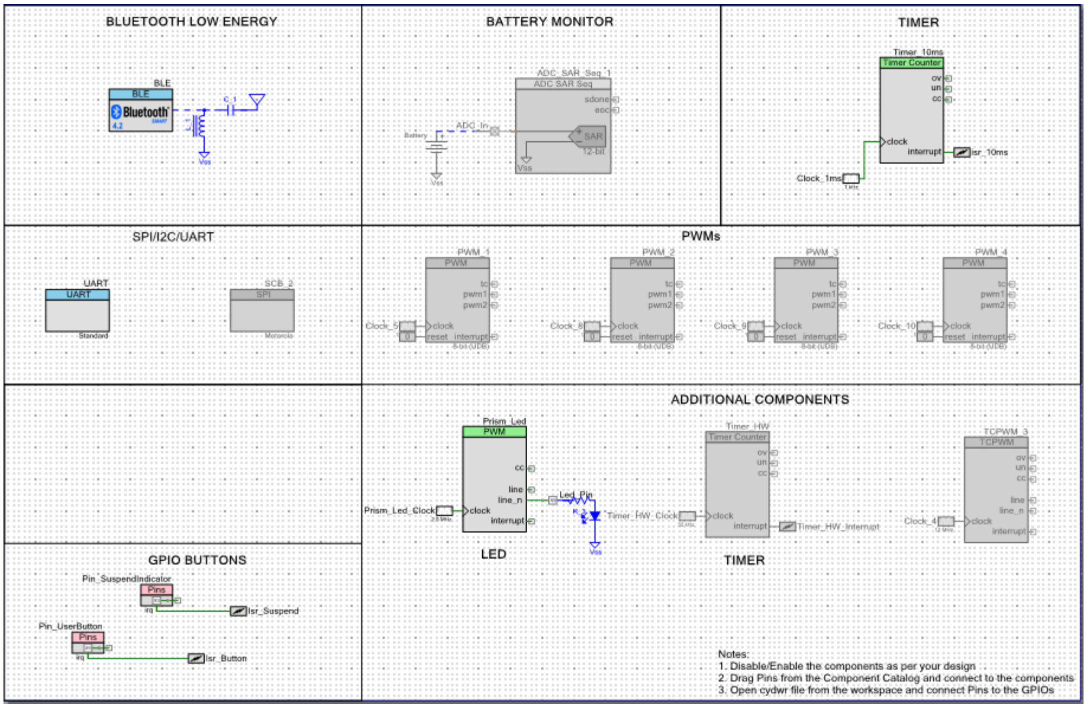
This project additionally demonstrates LED brightness control via a custom BLE profile, which works with the CapSense slider example explained in CapSense Slider and LED on page 39.

The device will scan for the Peripheral, which acts as a CapSense slider and LED device, and connect to it automatically. This is achieved by filtering the advertisement packets for the CapSense slider service data. Then, it will enable slider notifications and process the received notifications. Whenever CapSense detects activity, it will notify the finger location to the BLE Dongle; it will update the LED brightness using PWM.

The custom GATT client LED control will be stopped if the CySmart Central Emulation 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 Central Emulation Tool and the BLE device via a USB-CDC interface. It uses the Cypress USB-UART bridge functionality from the PSoC 5LP-based KitProg module.

Note: This project is meant only for the PRoC BLE device and works on the CY5677 CySmart BLE 4.2 USB Dongle hardware. The example project for CY5670 CySmart USB Dongle is available in the CY8CKIT-042-BLE Pioneer Kit.

Figure 4-71. Top Design for BLE_4_2_Dongle_CySmart_256K Project

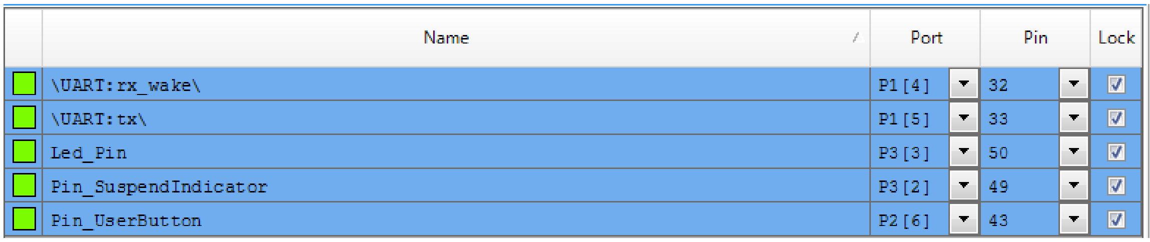


4.6.2 Hardware Connections

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

The pin assignment for this project is in **BLE_4_2_Dongle_CySmart_256K.cydwr** in the Workspace Explorer, as shown in Figure 4-72.

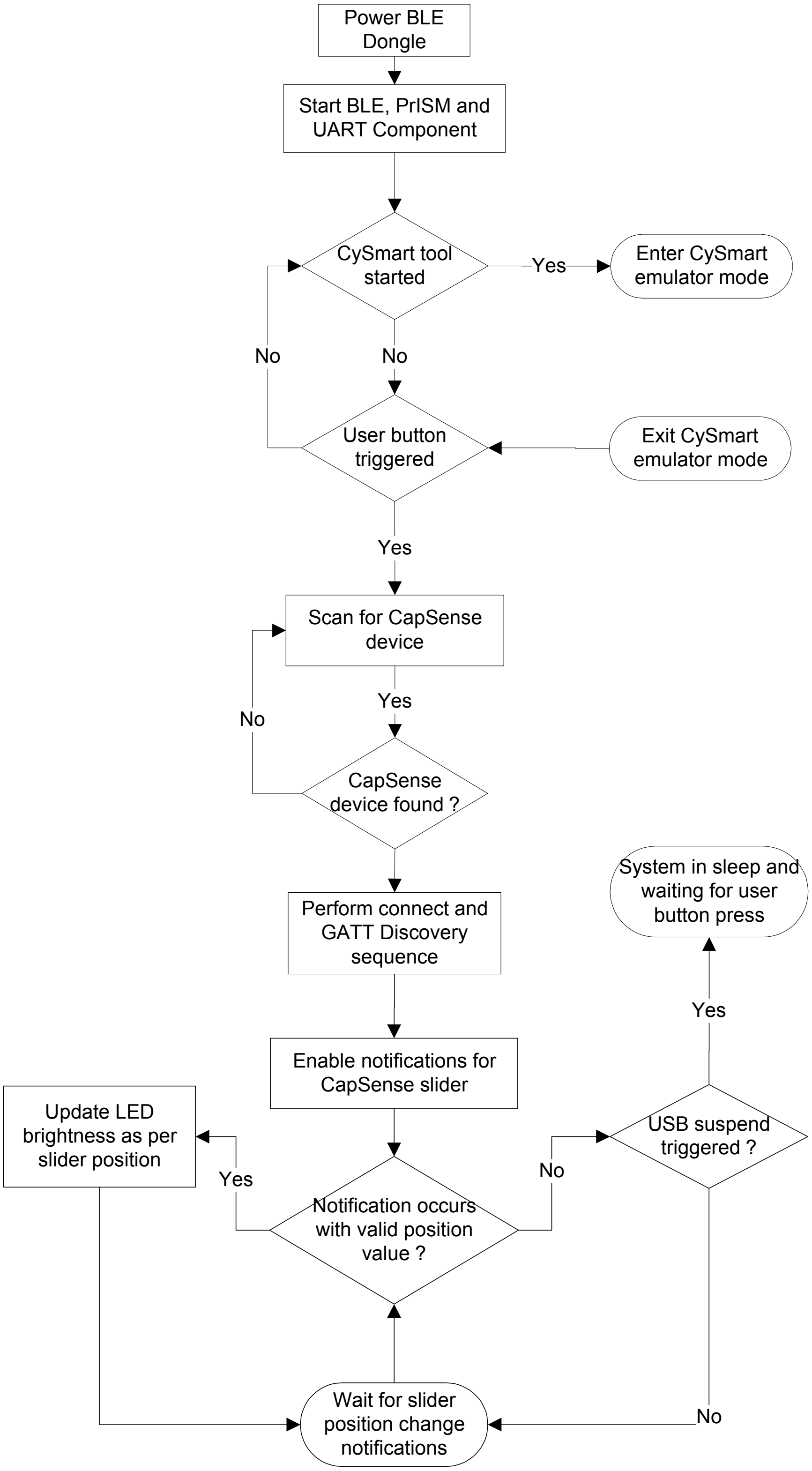
Figure 4-72. Pin Selection for BLE Dongle Project



	Name	Port	Pin	Lock
🟩	\UART:rx_wake\	P1[4]	32	☑
🟩	\UART:tx\	P1[5]	33	☑
🟩	Led_Pin	P3[3]	50	☑
🟩	Pin_SuspendIndicator	P3[2]	49	☑
🟩	Pin_UserButton	P2[6]	43	☑

4.6.3 Flow Chart

Figure 4-73. Flow Chart for BLE_4_2_Dongle_CySmart_256K Project


Power BLE Dongle
Start BLE, PrISM and UART Component
CySmart tool started
Yes
Enter CySmart emulator mode
No
No
User button triggered
Exit CySmart emulator mode
Yes
Scan for CapSense device
Yes
No
CapSense device found ?
Perform connect and GATT Discovery sequence
System in sleep and waiting for user button press
Yes
Enable notifications for CapSense slider
Update LED brightness as per slider position
USB suspend triggered ?
No
Yes
Notification occurs with valid position value ?
Wait for slider position change notifications
No

4.6.4 Verify Output

This project will be used when the CySmart Central Emulation Tool 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 39.
3. Connect the BLE Dongle to one of the USB ports on the computer.
4. Program the BLE Dongle with the **BLE_4_2_Dongle_CySmart_256K** project. See Using Example Projects on page 35 for programming instructions.
5. Press the user button **SW2** on both the BLE Dongle and the BLE Pioneer Kit. The BLE Dongle will start scanning and the BLE Pioneer Kit will start advertising.
6. Wait for the BLE connection between the BLE Dongle and the BLE 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 BLE Pioneer Baseboard.
7. Swipe your finger on the CapSense slider and check the LED brightness variation on the BLE Dongle.

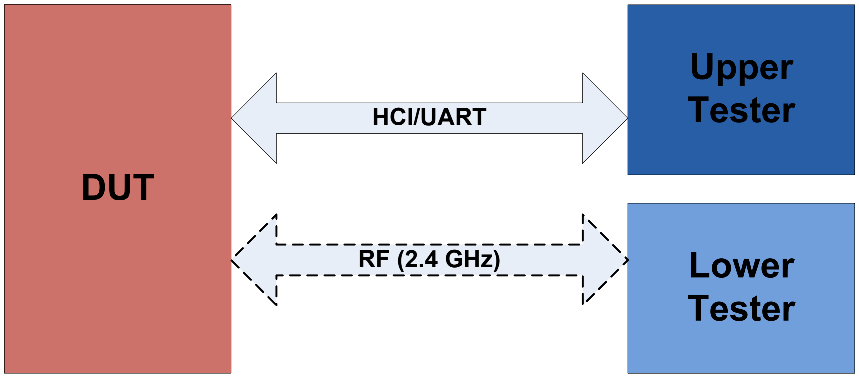
4.7 Direct Test Mode (DTM)

4.7.1 Project Description

Bluetooth Core specification (v4.0 and later), Volume 6, Part F defines Direct Test Mode (DTM) as a method to test the BLE PHY layer and provide a report back to the tester. It uses a **Host Controller Interface (HCI)** with a **two-wire UART** as the communication protocol.

The Device Under Test (DUT) is the BLE system that is to be tested (for example, BLE Pioneer Kit). With DTM, the RF performance of the BLE system can be verified during development or on a production line. The environment consists of the DUT and a tester. The tester has two parts; the **upper tester** sends commands through the HCI and the **lower tester** performs the corresponding action over the RF link. The tester compares the command sent over the HCI and the response received over RF, and provides a result of the performance.

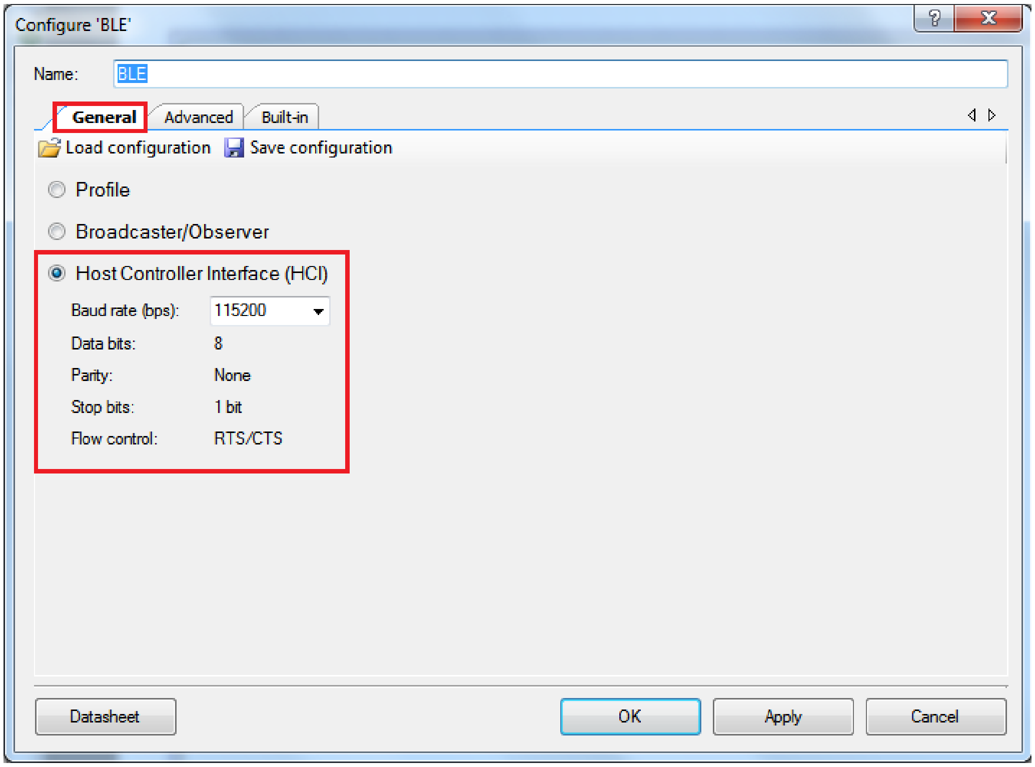
Figure 4-74. Direct Test Mode (DTM) Setup



The BLE Component allows configuring the device in DTM by enabling the HCI. The appropriate responses to commands from the tester are performed by the BLE protocol stack and does not involve separate application handling. The only task required are to start the BLE Component and call the API to process the events.

The HCI is enabled in the BLE Component under the General settings. Note that when the HCI mode is selected, all other tabs are hidden and cannot be configured. This is because in HCI mode, there are no upper layer processes. On enabling HCI mode, the components automatically reserves a UART block to allow communication between the tester and BLE stack. The UART exposes the pins that can be assigned in <*project.cydwr*> file under the Pins tab. The only options to be configured for HCI mode are the baud rate and the pins for communication with the tester.

Figure 4-75. HCI Mode in BLE Component



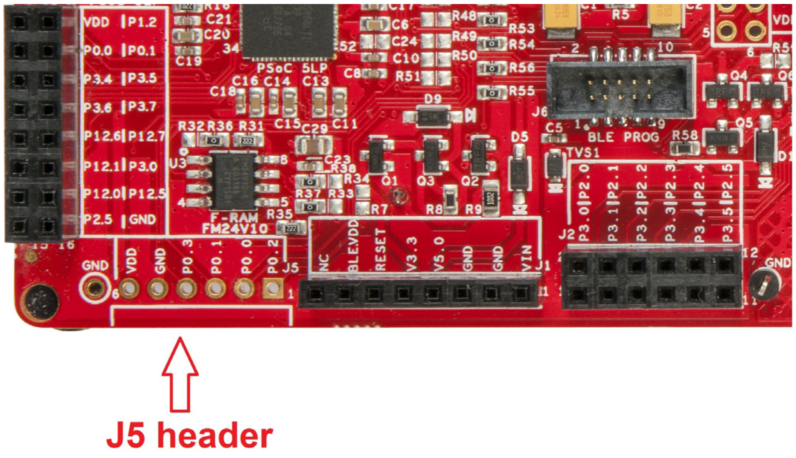
Baud rate (bps): 115200
Data bits: 8
Parity: None
Stop bits: 1 bit
Flow control: RTS/CTS

Many companies develop BLE testers for Direct Test Mode. It is also possible to create PC-based software tools that will send HCI commands over serial communication links.

For PC-based software, the serial communication link is the COM port, which is enumerated by the KitProg on the PSoC 5LP of the BLE Pioneer kit. In such a case, the UART pins on the PSoC 4 BLE/PRoC BLE should be assigned to P1_4 and P1_5. These pins are hardwired to pins on the PSoC 5LP which allows USB-UART data communication between the PC-based software and the BLE device.

For external BLE testers, the serial communication is mostly over RS232. To test with that type of system, an external RS232 voltage translator is required, such as Digilent's PmodRS232. This translator will modify the signal levels of the serial communication between the BLE device and the RS232 port on the tester. The UART pins of the BLE device can be assigned to P0_0 and P0_1 and header J5 can be used to connect to the RS232 translator.

Figure 4-76. J5 Header to Interface RS232 Translator



4.7.2 Hardware Connection

For DTM test mode, it is recommended to use SMA connectors and connect the tester and DUT using an SMA to SMA connector cable. This ensures that there is minimum interference to RF communication between the DUT and tester, and the performance measured is the true RF performance of the device. The BLE Pioneer Kit module with SMA connector (CY8CKIT-141 PSoC 4 BLE) is available separately and can be ordered from the Cypress web page.

Four UART pins are exposed when HCI mode is selected in the BLE Component. These pins should be assigned to allow communication with the external tester. The connection depends on the tester being used.

If the tester is PC-based software that communicates with HCI over a serial link, then the onboard PSoC 5LP on the BLE Pioneer Kit can act as the USB-UART bridge. The KitProg on the PSoC 5LP enumerates as a USB-UART interface and opens a COM port on the computer. This COM port is then used by the software tool to communicate commands to the BLE device. In this case, the UART pins should be assigned as follows.

Table 4-1. UART Pin Assignment for PC Software Tester

UART Pins	Pin Assigned
RX	P1_4
TX	P1_5
RTS	P1_6
CTS	P1_7

The UART for HCI communication exposes hardware flow control lines CTS and RTS. They can either be connected to the hardware control lines of the tester or CTS connected to ground for operation without hardware flow control.

If the tester is an external hardware tester (CBT), then connect any of the RS232 voltage translators to header J5 on the BLE Pioneer Kit. The UART pins should be assigned as follows.

Table 4-2. UART Pin Assignment for RS232 Voltage Translator

UART Pins	Pin Assigned
RX	P0_0
TX	P0_1
RTS	P0_2
CTS	P0_3

4.7.3 Verify Output

1. Connect the BLE Pioneer Kit through the USB connector J13.
2. Program the BLE Pioneer Kit with the PSoC_4_BLE_DTM or PRoC_BLE_DTM project, depending on the module used (PSoC 4 BLE or PRoC BLE), as described in Using Example Projects on page 35. Programming should complete successfully.
3. Connect the serial link, UART, or RS232 to the tester.
4. On the software tool for tester, configure the UART communication with the correct COM port and baud rate, as set in the BLE Component.
5. Start the test. The tool will generate the report after the end of the test. This depends on the tester/tool being used.

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, module connectors, and CapSense slider.

The schematic and board layouts are available at the following location:
`<Install_Directory>\Cypress\CY8CKIT-042-BLE-A Kit\<version>\Hardware`.

5.1 BLE Pioneer Baseboard

5.1.1 PSoC 5LP

An onboard PSoC 5LP contains the KitProg, which is used to program and debug the BLE device. The PSoC 5LP connects to the USB port of the computer 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. The analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals.

For more information, visit the PSoC 5LP web page.

See Serial Interconnection between KitProg and Module on page 103 for more details.

5.1.2 Power System

The power supply system on the BLE Pioneer Baseboard is versatile, allowing the input supply to come from the following sources:

- 5-V power from the onboard USB connector
- 5-V to 12-V VIN power from the Arduino power header (J1)
- 3-V from the 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 module. These voltages are selected with the J16 jumper, as shown in Figure 5-1.

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

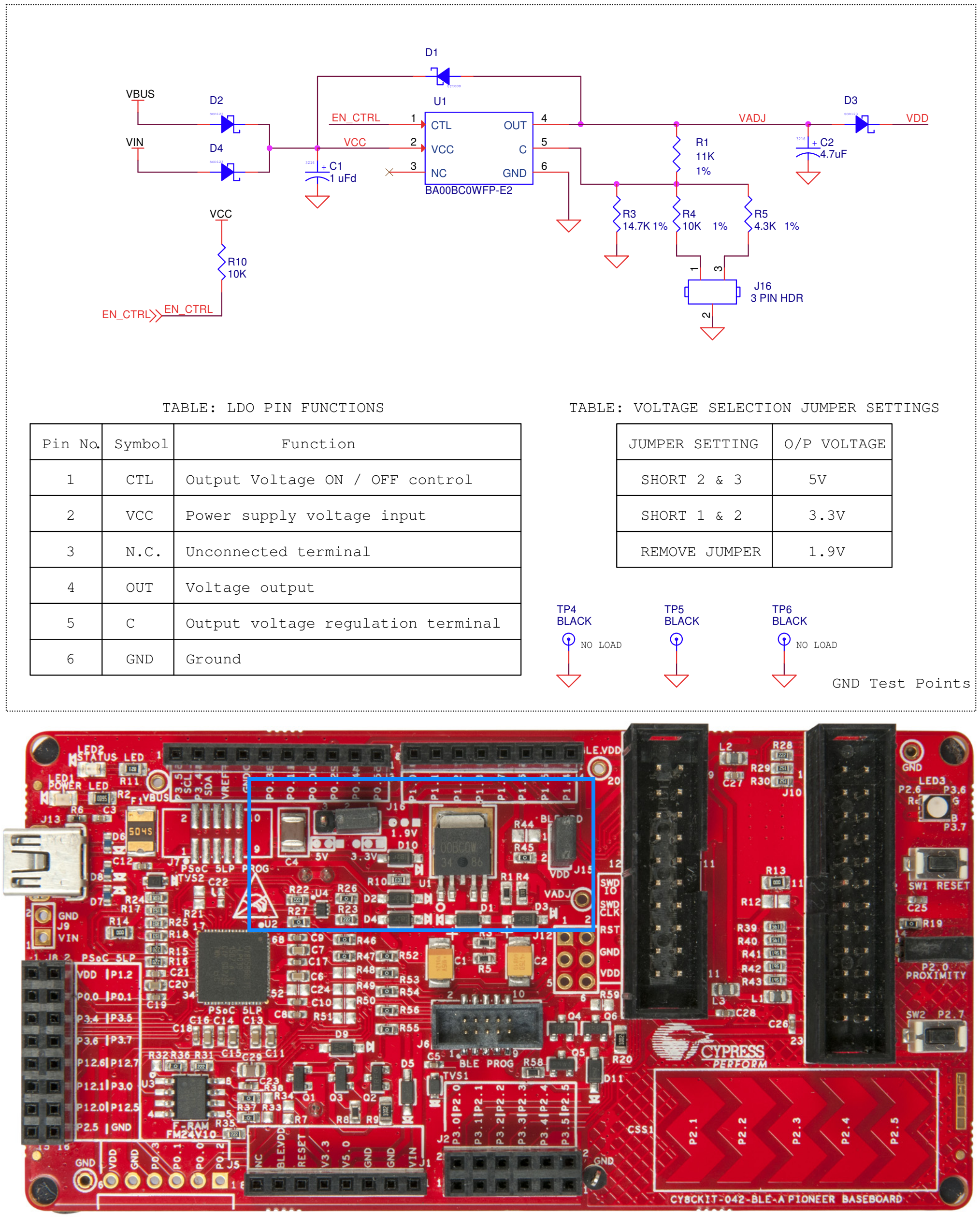


TABLE: LDO PIN FUNCTIONS

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

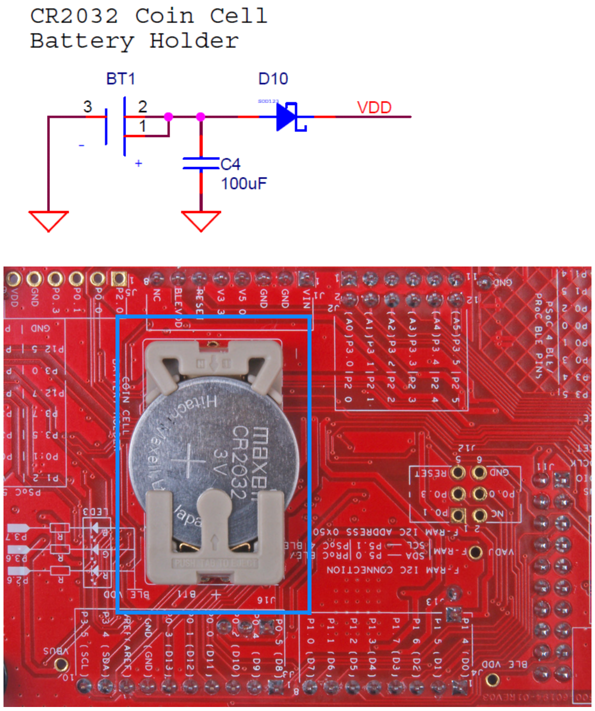
JUMPER SETTING	O/P VOLTAGE
SHORT 2 & 3	5V
SHORT 1 & 2	3.3V
REMOVE JUMPER	1.9V

The input to the LDO can come from either the USB, the VIN pin in the Arduino header J1, or header J9.

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 considers 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. An input voltage supply over 12 V can damage the board.

The BLE Pioneer Baseboard also contains a CR2032 coin cell holder to power it using a coin cell, as shown in Figure 5-2.

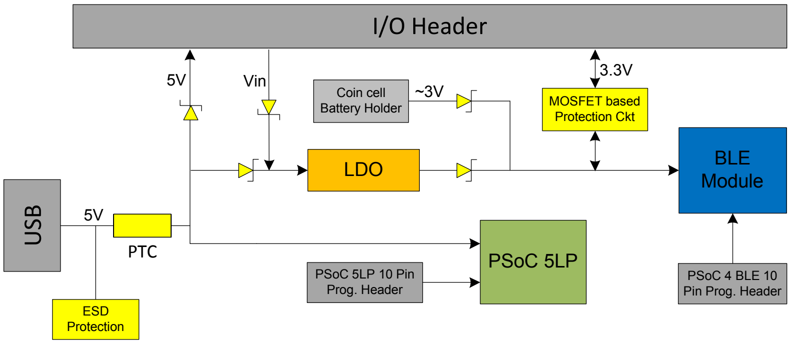
Figure 5-2. Schematics and Board Highlight of Coin Cell Holder



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.

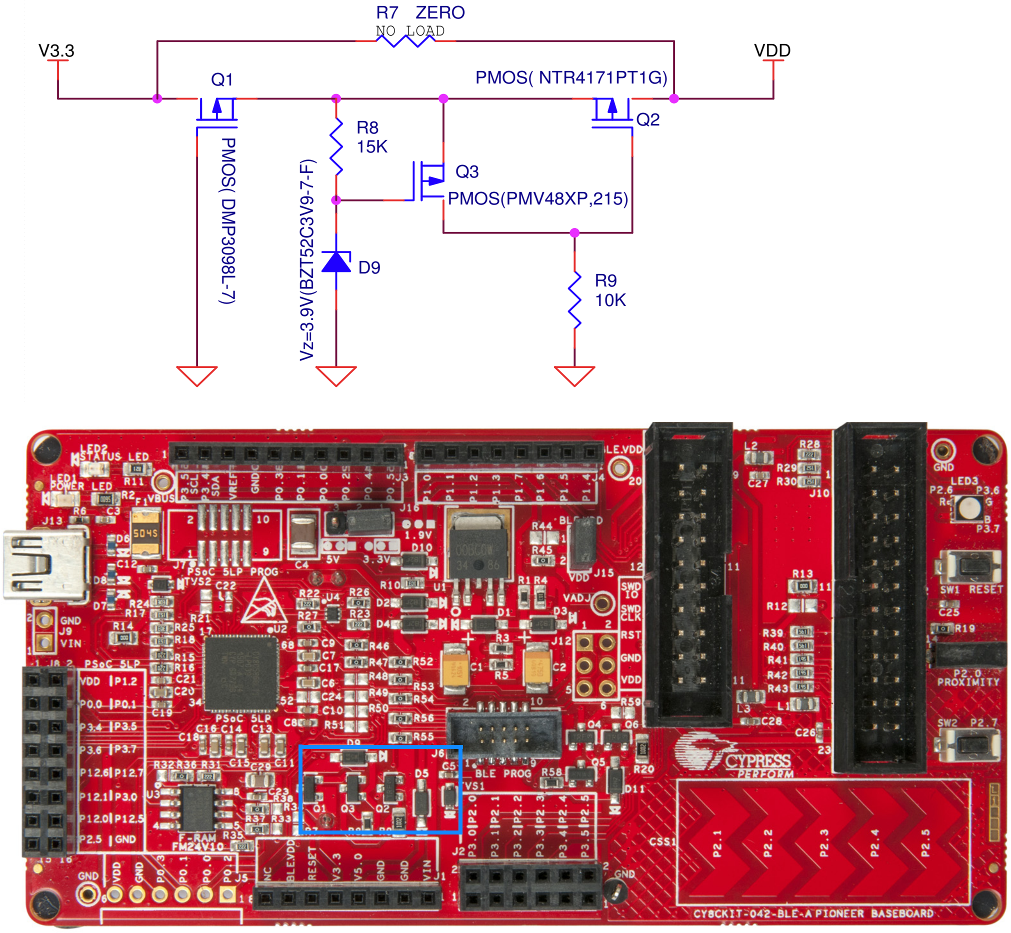
Figure 5-3. Power Supply Block Diagram With Protection Circuits



- 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 BLE Pioneer Baseboard 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.

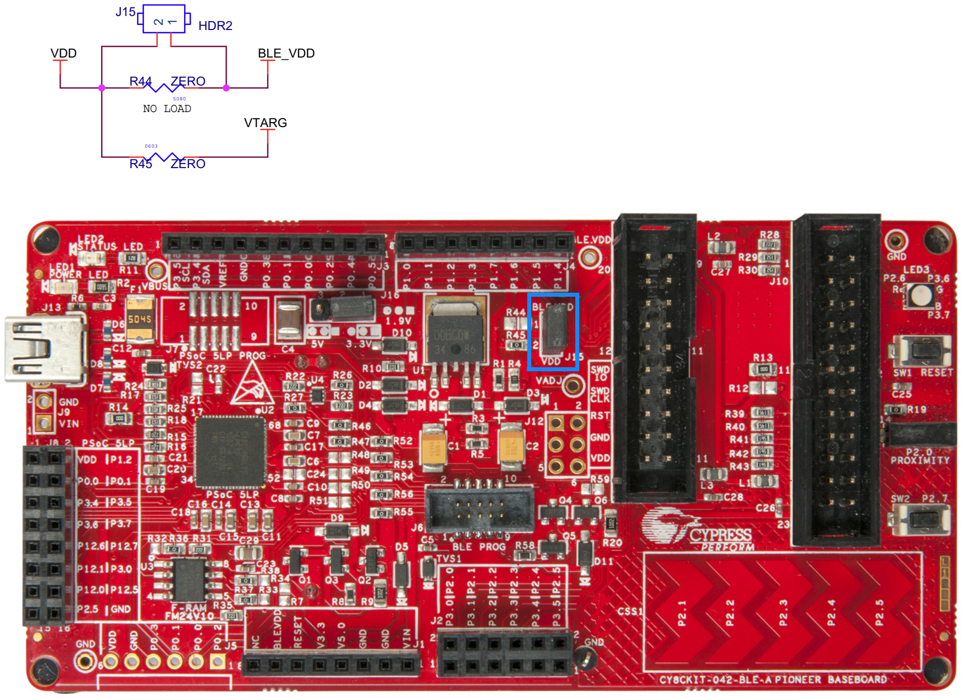
Figure 5-4. Schematics and Board Highlight of MOSFET Protection Circuit for 3.3-V Rail from J1.5



5.1.2.2 Current Measurement Jumper

To demonstrate the low power consumption of PSoC 4 BLE/PRoC BLE Module, a two-pin header (J15) is populated in series with the power supply to the module. This can be used to measure current using an ammeter without the need to desolder any component from the BLE Pioneer Baseboard, as shown in Figure 5-5.

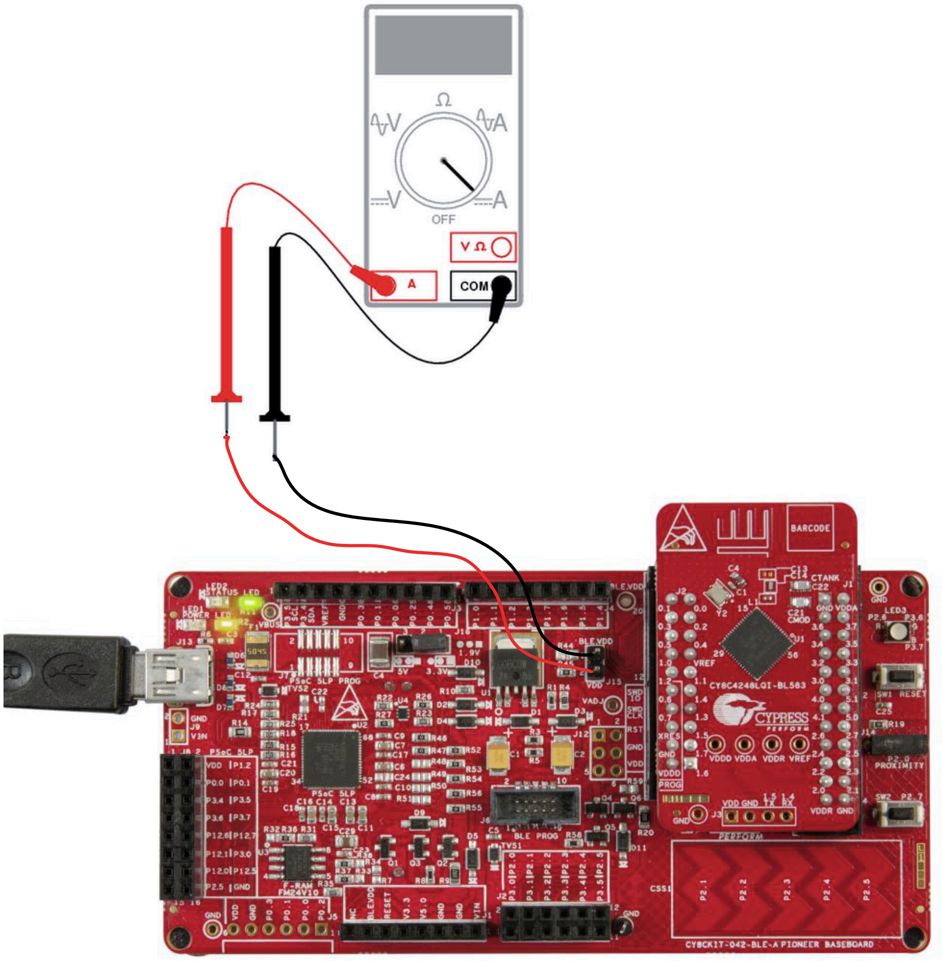
Figure 5-5. Schematics and Board Highlight of Current Measurement Jumper



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

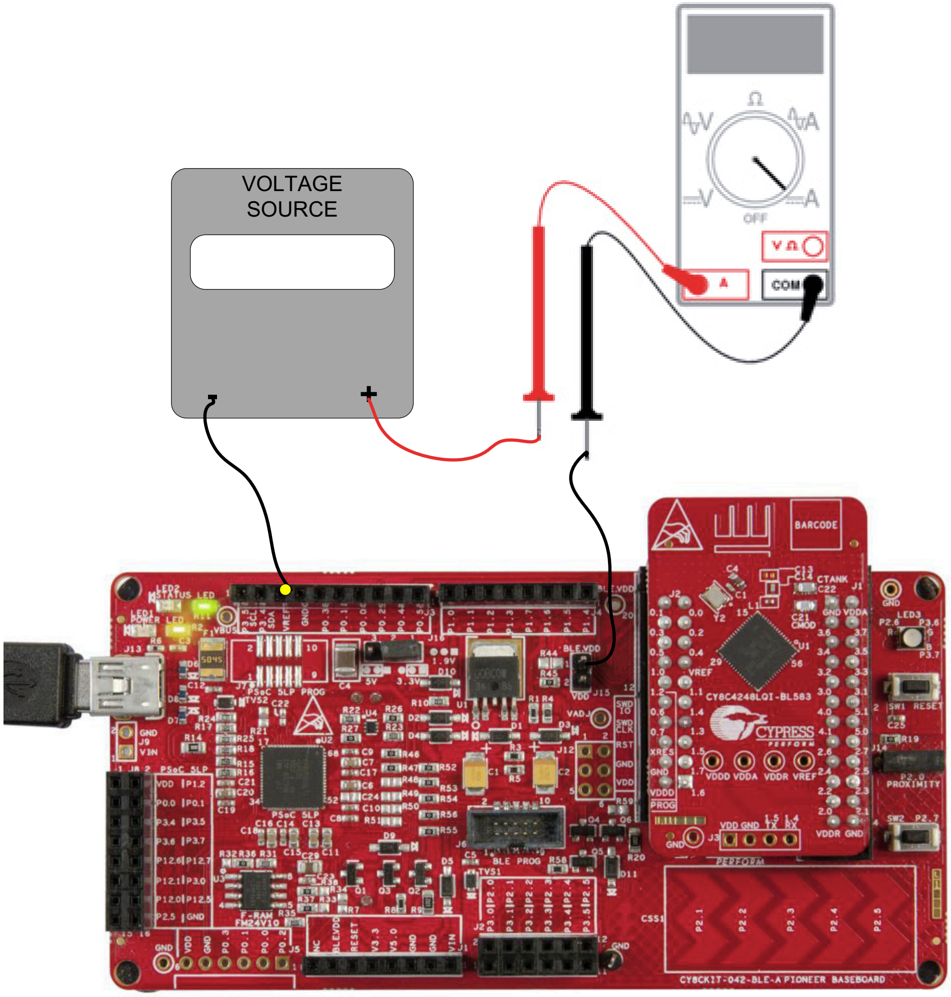
■ When the BLE Pioneer Baseboard is powered through the USB port (J13), remove jumper J15 and connect an ammeter, as shown in Figure 5-6.

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



- When the BLE Pioneer Baseboard is powered from an external voltage supply, remove the USB cable from J13. Connect the positive terminal of the external voltage supply to the positive terminal of the ammeter and the negative terminal of the ammeter to the upper pin of J15. Connect the negative terminal of the external voltage supply to an onboard GND pin. Figure 5-7 shows the required connections.

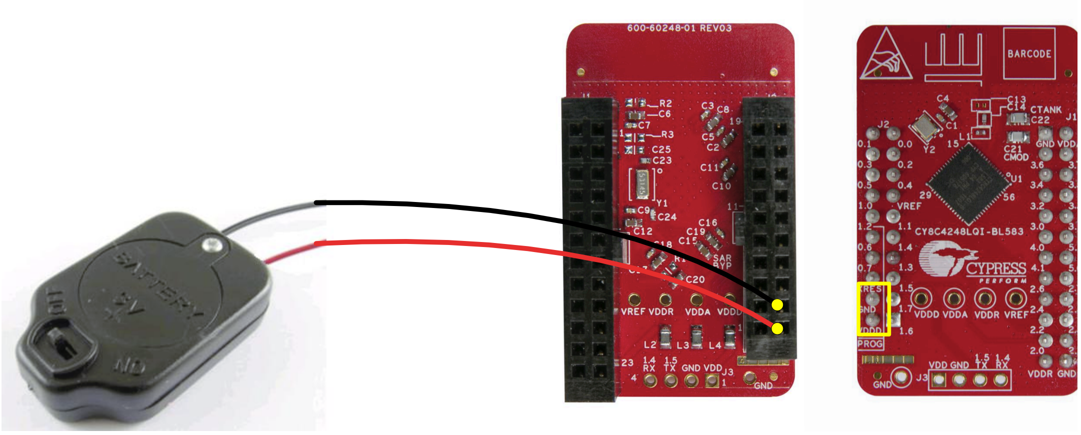
Figure 5-7. Current Measurement when Powered Separately



To measure the power consumption of only the module with coin cell, connect the coin cell directly to the modules, as shown in Figure 5-8. The BLE Pioneer Baseboard is designed with additional circuits to protect the BLE device and the F-RAM in an Arduino environment. Note that power consumption measurements on the BLE Pioneer Baseboard will also include the power consumed by these additional circuits.

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

Figure 5-8. Powering the Module using a Coin Cell



5.1.3 Programming Interface

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

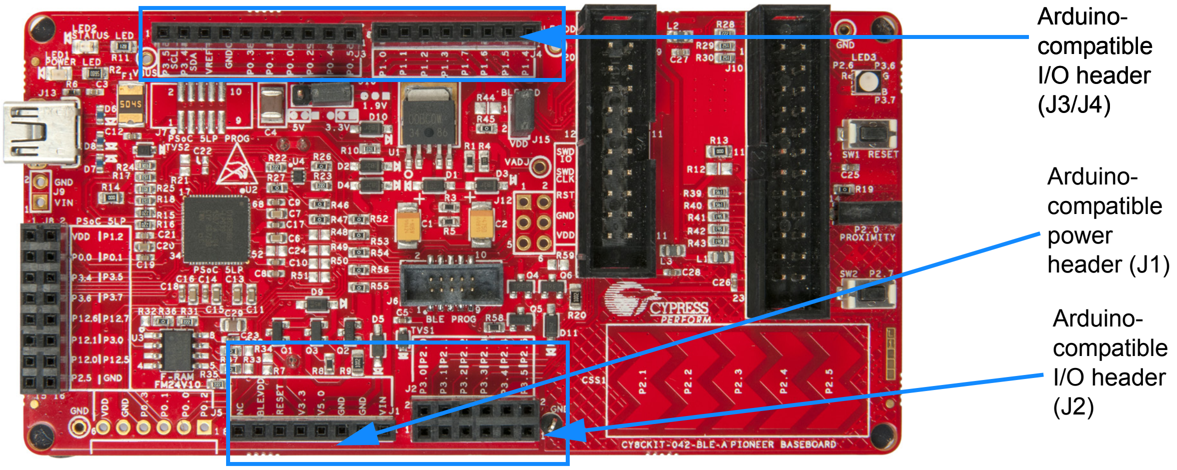
- Using the onboard KitProg
- Using a CY8CKIT-002 MiniProg3 programmer and debugger

5.1.4 Expansion Connectors

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

The BLE Pioneer Kit has five Arduino-compatible headers: J1, J2, J3, J4, and J12, as shown in Figure 5-9. You can develop applications based on the Arduino shield's hardware.

Figure 5-9. Arduino Headers


Arduino-compatible I/O header (J3/J4)
Arduino-compatible power header (J1)
Arduino-compatible I/O header (J2)

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^{2}C, 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 116 for the header part number.

Note: Take care when powering the Arduino shields via Arduino-compatible power header (J1). The V3.3 pin will output 5 V when the board is powered from USB/VIN and the system power supply jumper (J16) is set to 5 V operation.

Additional Functionality of Header J2

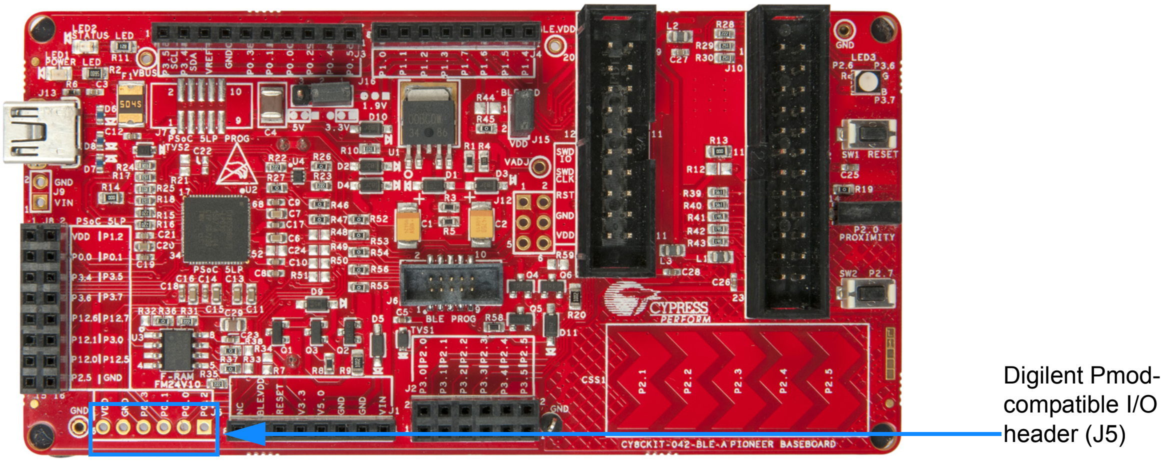
The J2 header is a 6×2 header that supports Arduino shields. The Port 2 and Port 3 pins of PSoC 4BLE and PRoC BLE are brought to this header. The Port 2 pins also connect to the onboard CapSense slider through 560-ohm resistors. When the CapSense feature is not used, remove these resistors to help ensure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

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

This port supports Digilent Pmod 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 Baseboard. 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 116 for the header part number.

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

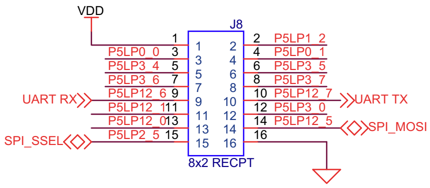


5.1.4.3 PSoC 5LP GPIO Header (J8)

An 8×2 header is provided on the BLE Pioneer Baseboard 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 interface pins that can be used when these pins are not accessible on the Arduino headers because a shield is connected.

Note: You can use PSoC 5LP for your own custom firmware.

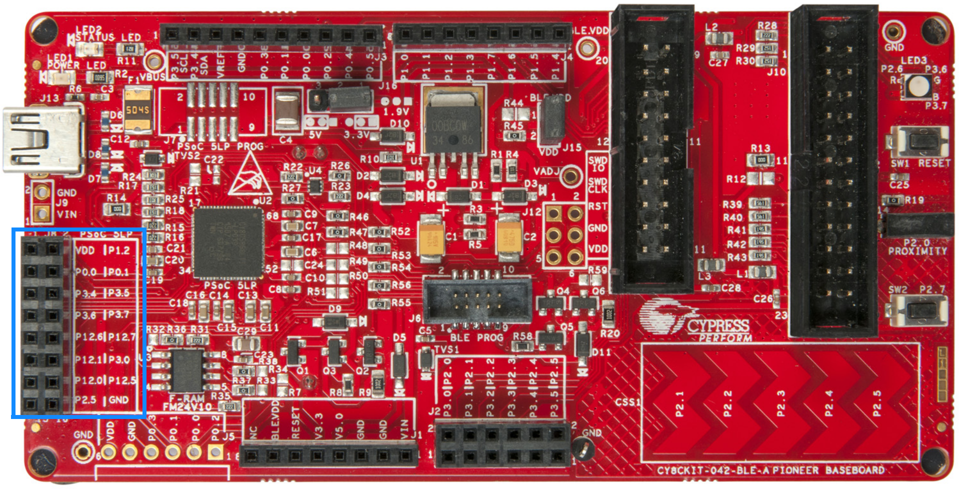
Figure 5-11. Schematics and Board Highlight of PSoC 5LP GPIO Expansion Header



PSoC 5LP GPIO Expansion Header

USB-Serial Bridge Connections

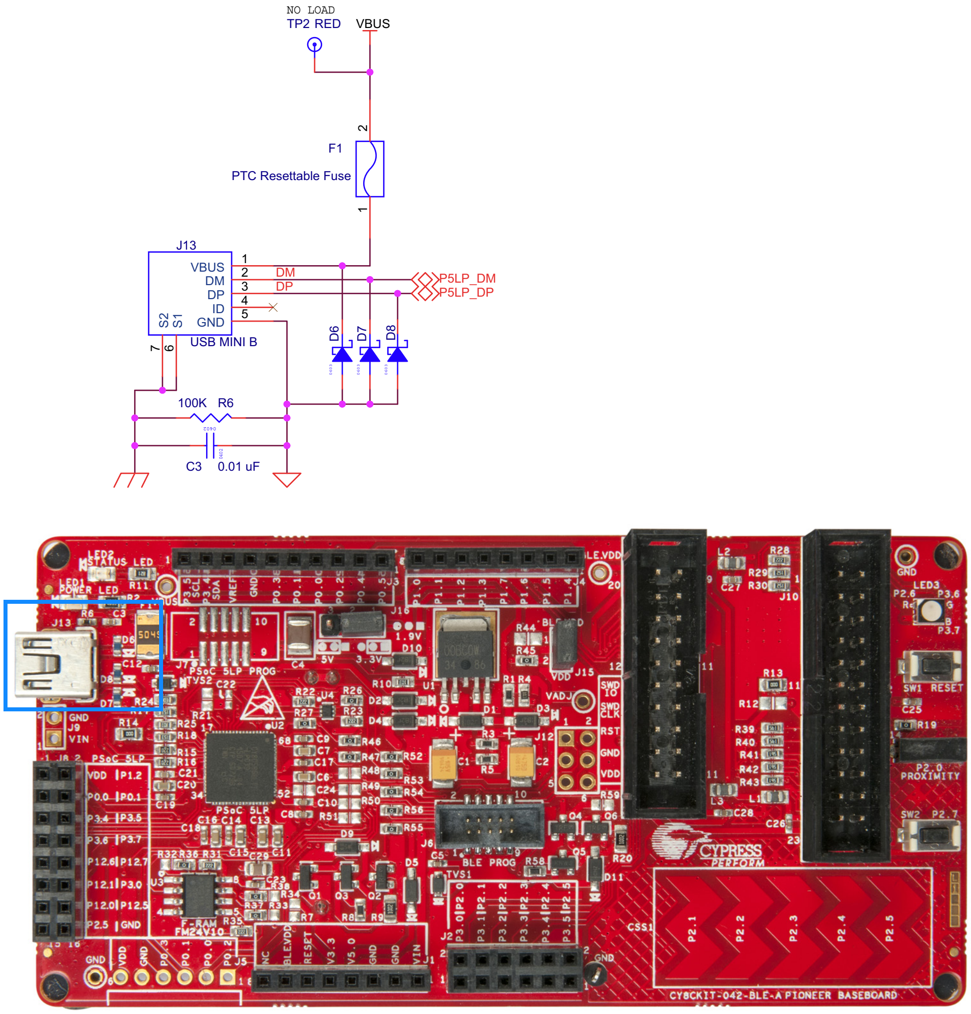
```
P5LP12_6 ------------- UART RX
P5LP12_7 ------------- UART TX
P5LP12_0 ------------- SPI SCLK / I2C SCL
P5LP12_1 ------------- SPI MISO / I2C SDA
P5LP12_5 ------------- SPI MOSI
P5LP2_5  ------------- SPI SSEL
```



5.1.5 USB Mini-B Connector

The PSoC 5LP connects to the USB port of a computer through a Mini-B connector (see Figure 5-12), which can also be used to power the BLE Pineer Baseboard. 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

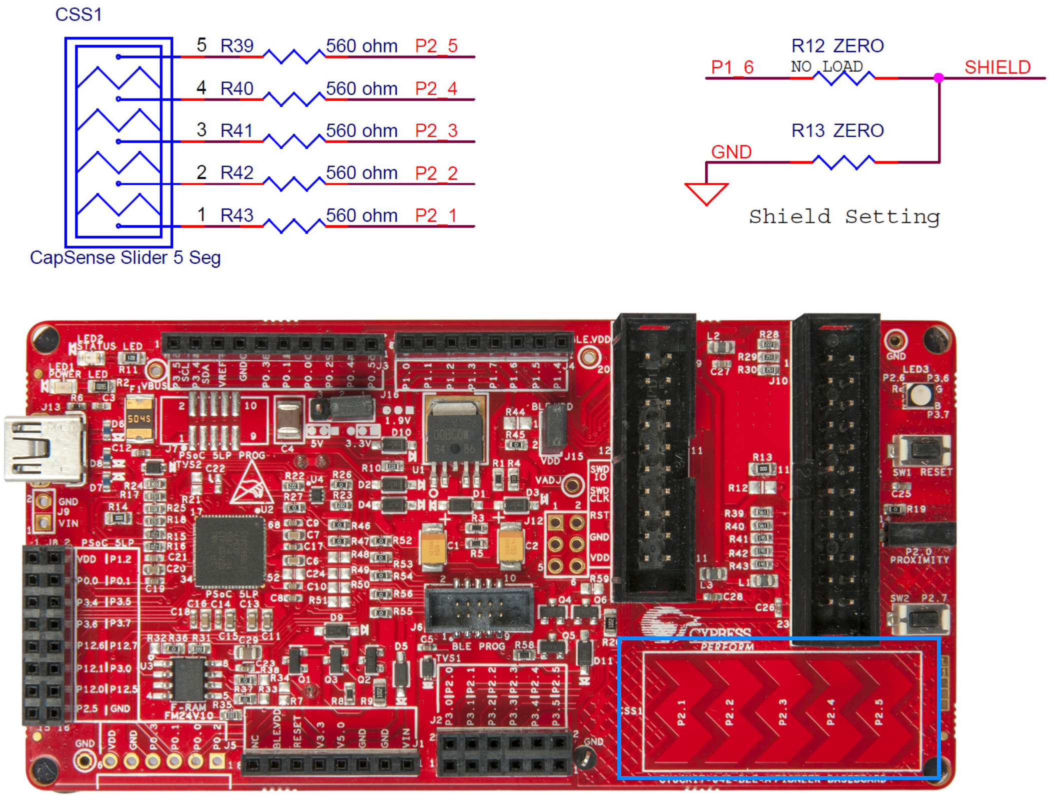


5.1.6 CapSense Circuit

5.1.6.1 *CapSense Slider*

The BLE Pioneer Kit has a five-segment linear capacitive touch slider, which is connected to the PSoC 4 BLE/PRoC BLE Module pins (see Figure 5-13). The CMOD and CTANK capacitors are required for CapSense functionality and are provided on the modules (see Module Board on page 105). A 2.2-nF capacitor is present on the CMOD pin, P4[0], for CapSense operation. BLE Pioneer Kit also supports CapSense designs that enable waterproofing. 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 BLE Pioneer Baseboard, 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].

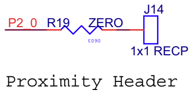
Figure 5-13. Schematics and Board Highlight of CapSense Slider and Shield Setting

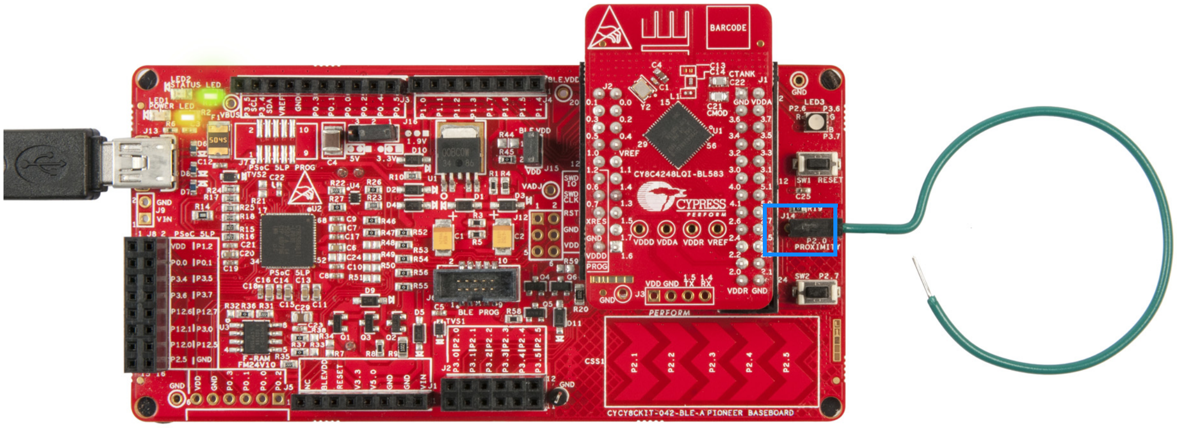


5.1.6.2 *Proximity Header*

The BLE Pioneer 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 BLE Pioneer Baseboard LEDs

The BLE Pioneer Baseboard 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 BLE Pioneer Kit also has a general-purpose tricolor LED (LED3) for user applications. These are connected to P2_6 (red LED), P3_6 (green LED) and P3_7 (blue LED). Figure 5-15 and Figure 5-16 show the schematics of these LEDs.

Figure 5-15. Schematics of Status and Power LED

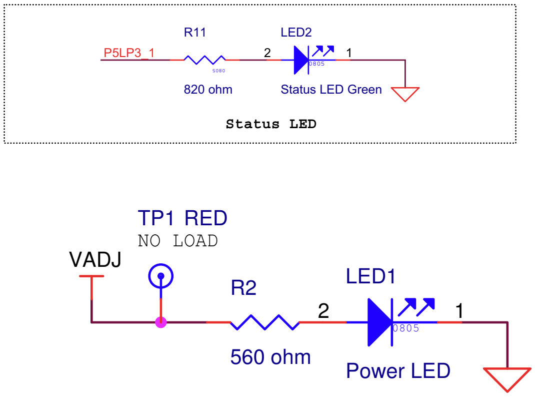
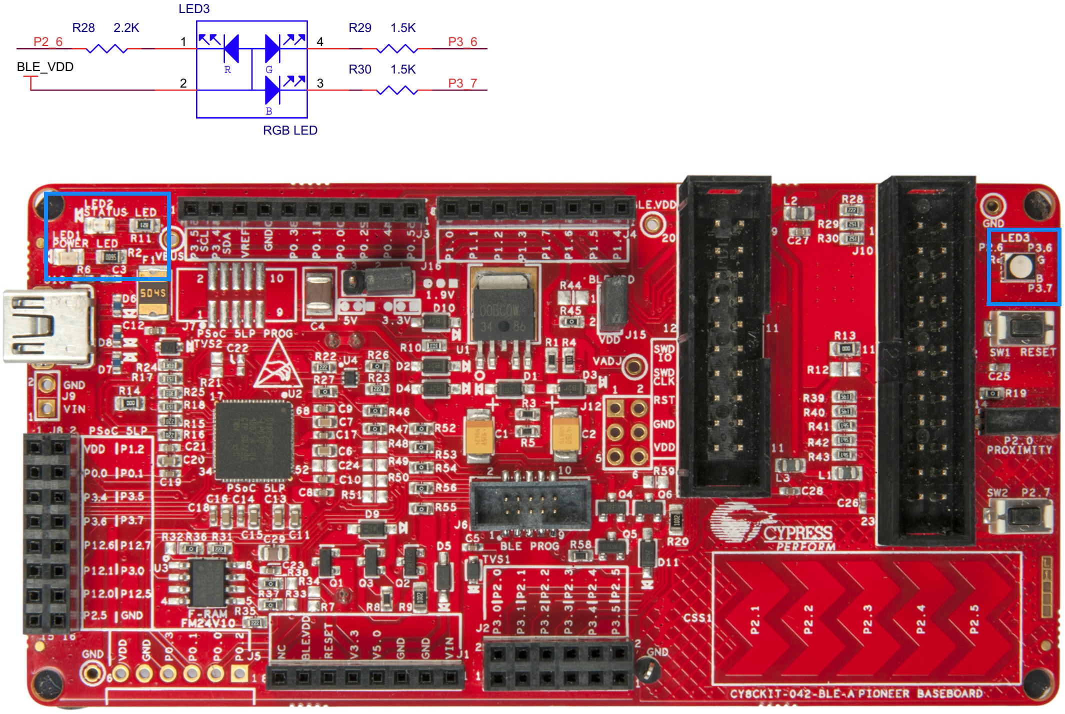


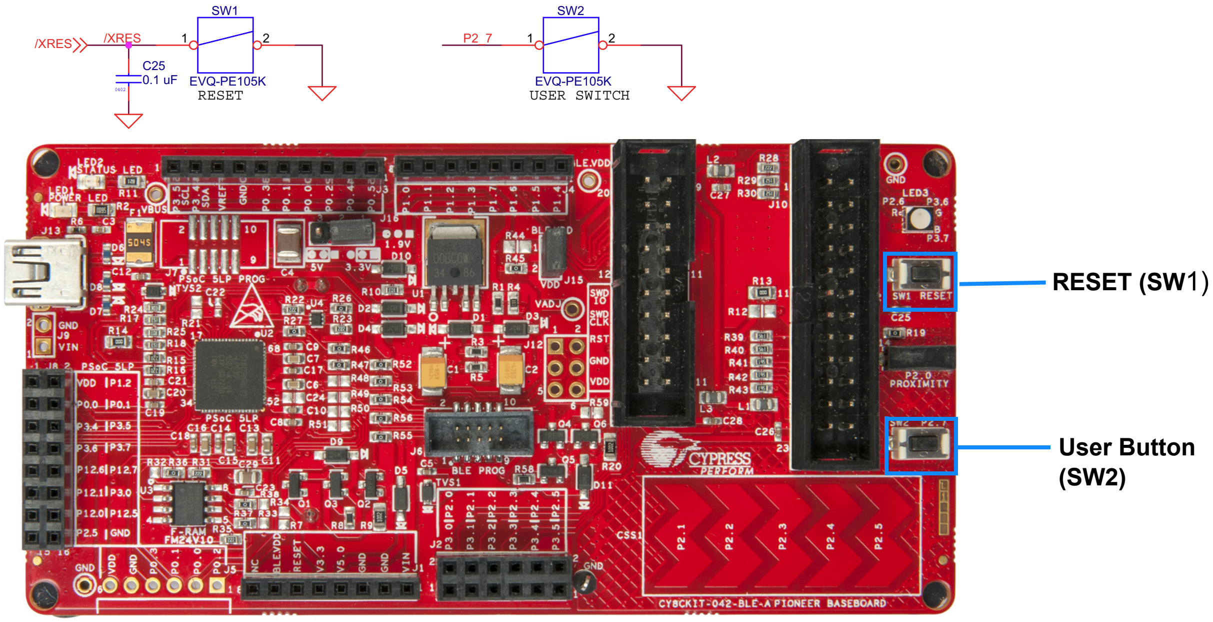
Figure 5-16. Schematics and Board Highlight of RGB LED



5.1.8 Push-Buttons

The BLE Pioneer Baseboard 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 device and is used to reset it. The user button is connected to P2[7] of the BLE device. Both the buttons connect to ground on activation (active low).

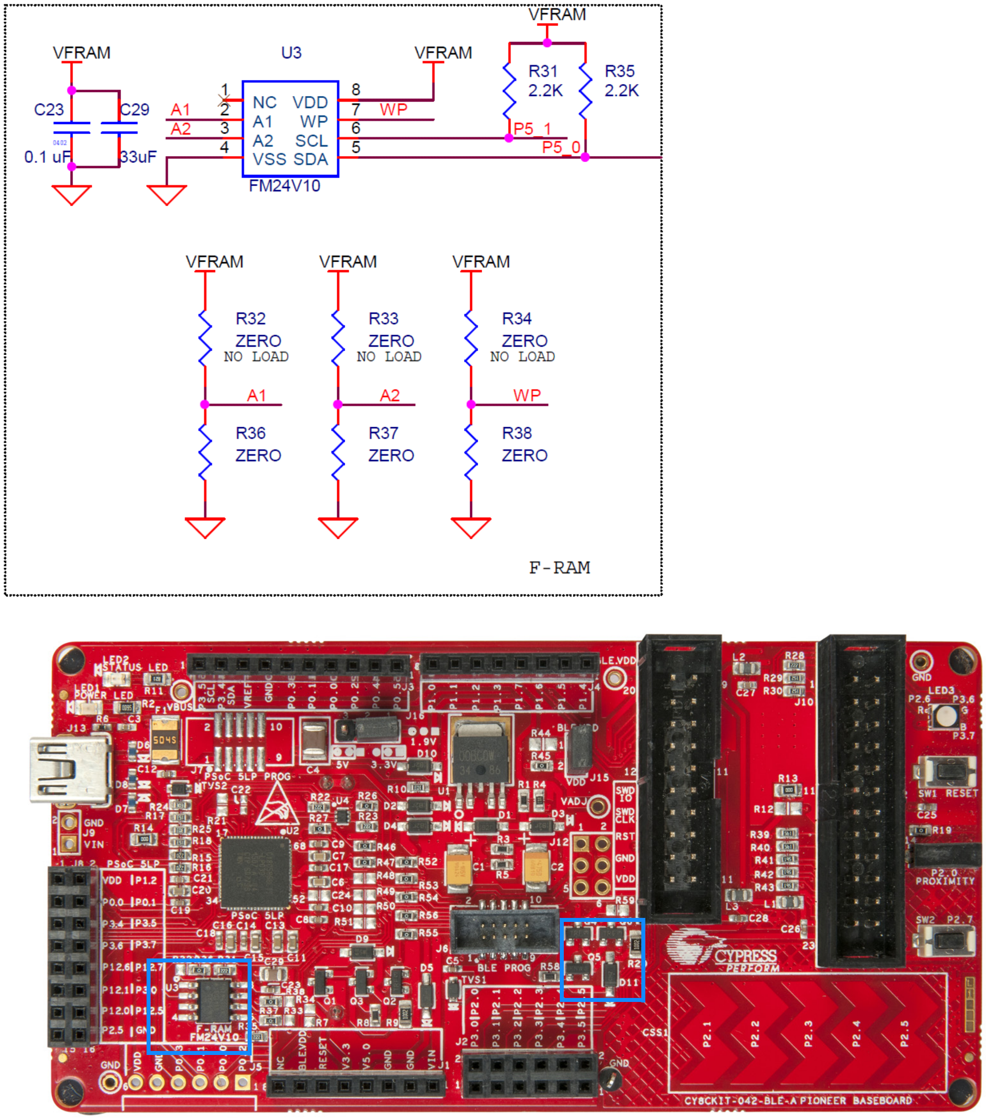
Figure 5-17. Schematics and Board Highlight of Reset Button and User Button



5.1.9 Cypress Ferroelectric RAM (F-RAM)

The BLE Pioneer Baseboard contains the FM24V10-G F-RAM device (see Figure 5-18), which can be accessed through I^{2}C lines P5[0] and P5[1] of the PSoC 4 BLE/PRoC BLE Module. The F-RAM is 1-Mbit (128KB) with an I^{2}C speed up to 1 Mbps. The I^{2}C 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 00 on the board. By default, the address of the F-RAM device used on the BLE Pioneer Baseboard 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 BLE Pioneer Baseboard, 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.

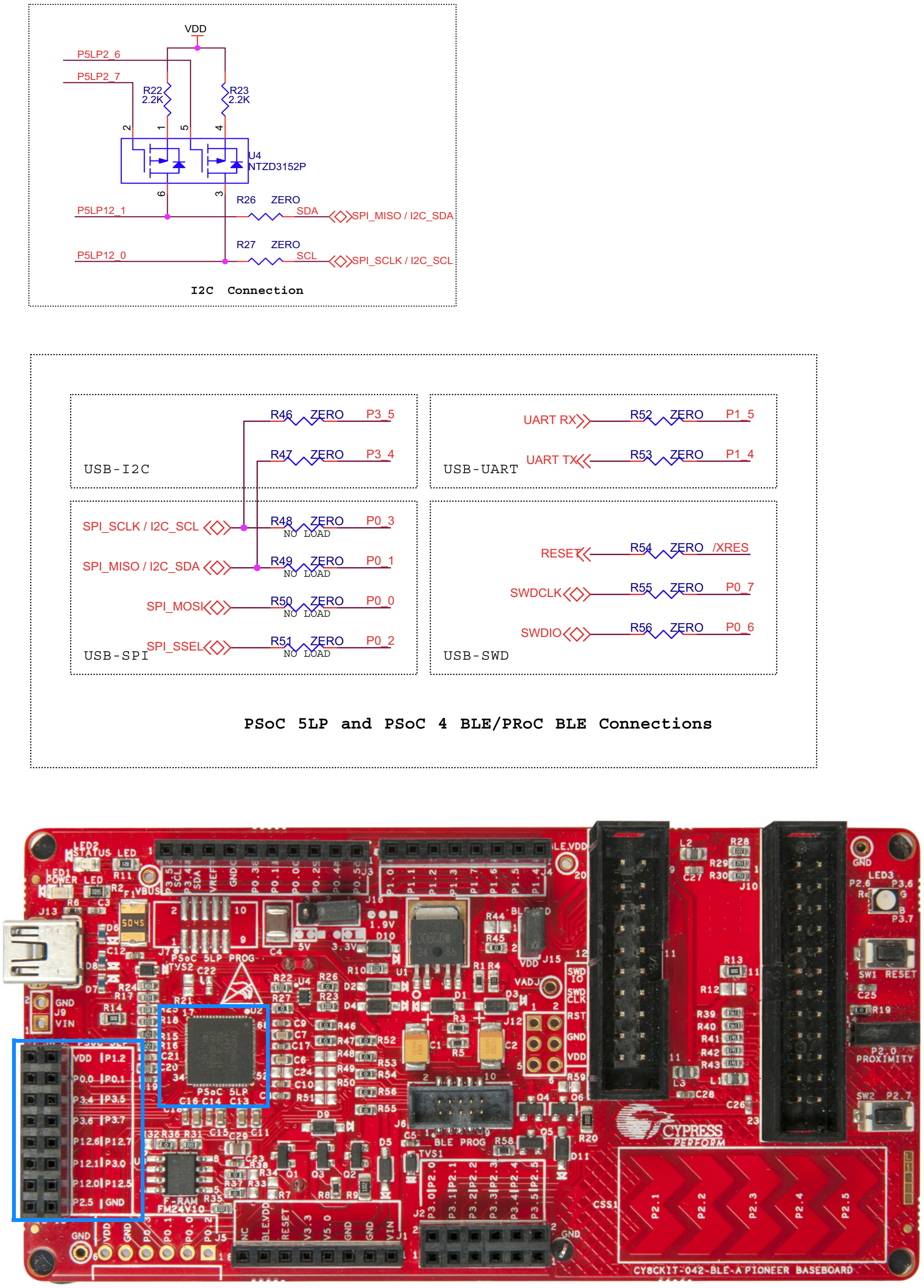
Figure 5-18. Schematics and Board Highlight of F-RAM



5.1.10 Serial Interconnection between KitProg and Module

The KitProg is also a USB-Serial interface. It supports USB-UART and USB-I^{2}C bridges (see Figure 5-19). The pull-up resistors on the I^{2}C bus are enabled when the protocol is selected from the user interface (such as Bridge Control Panel). The USB-Serial pins of the KitProg are also available on the Arduino header; therefore, it can be used to control Arduino shields with the SPI/I^{2}C/UART interface. Refer USB-UART Bridge on page 32 and USB-I2C Bridge on page 33 for more information on how to use these serial interconnections.

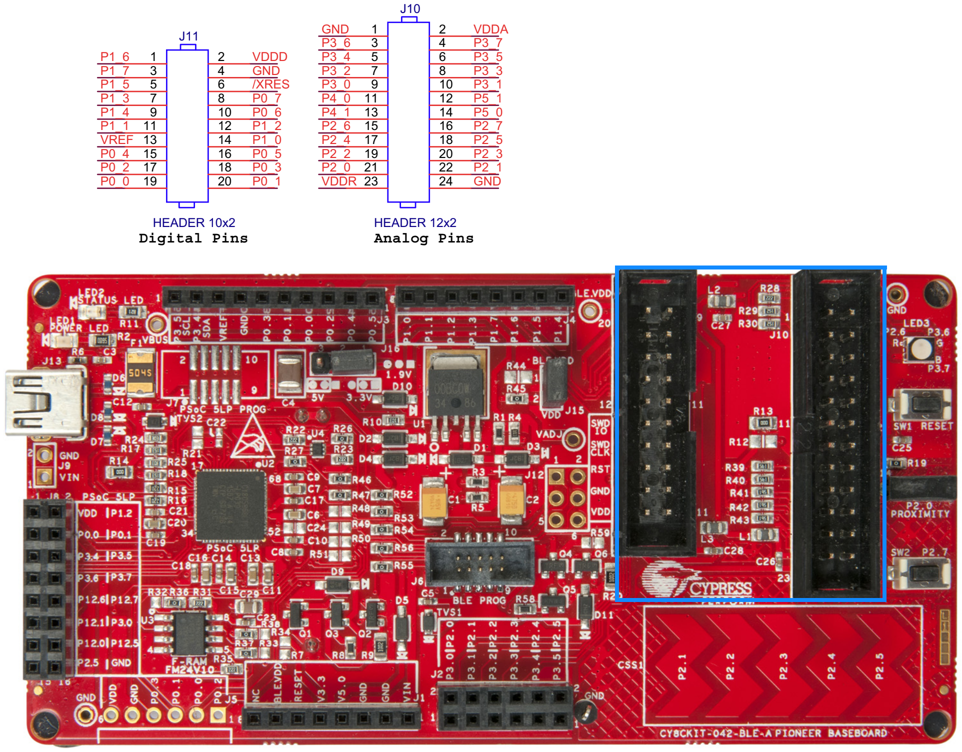
Figure 5-19. Schematics and Board Highlight of Serial Interface and I^{2}C Pull-Up via FET



5.1.11 Module Headers

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

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



For information on how to add these on your own board, refer to Adding BLE Module-Compatible Headers on Your Baseboard on page 126.

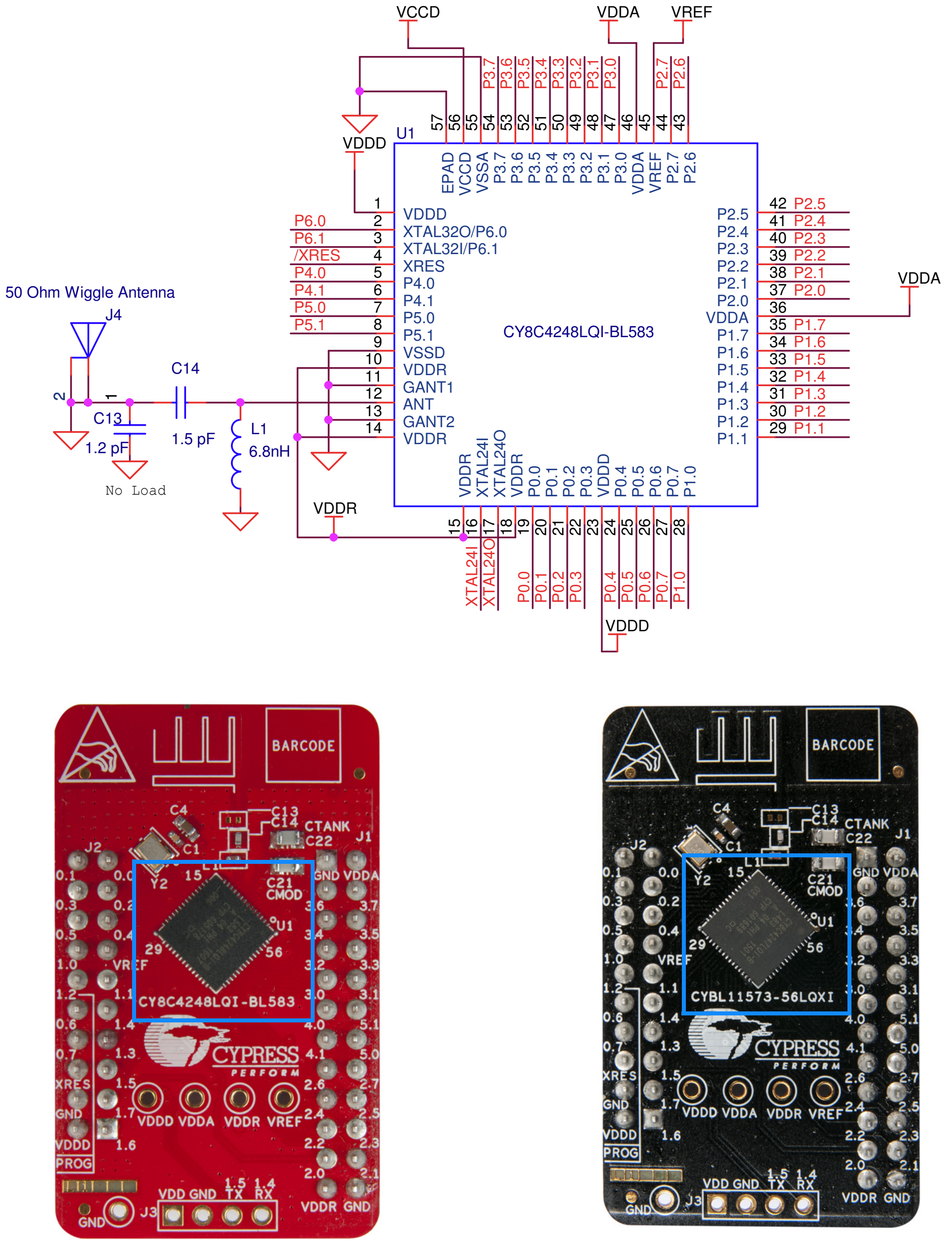
5.2 Module Board

5.2.1 PSoC 4 BLE or PRoC BLE Device

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

See BLE Modules and BLE Dongles Compatible with the BLE Pioneer Kit on page 128 for details.

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

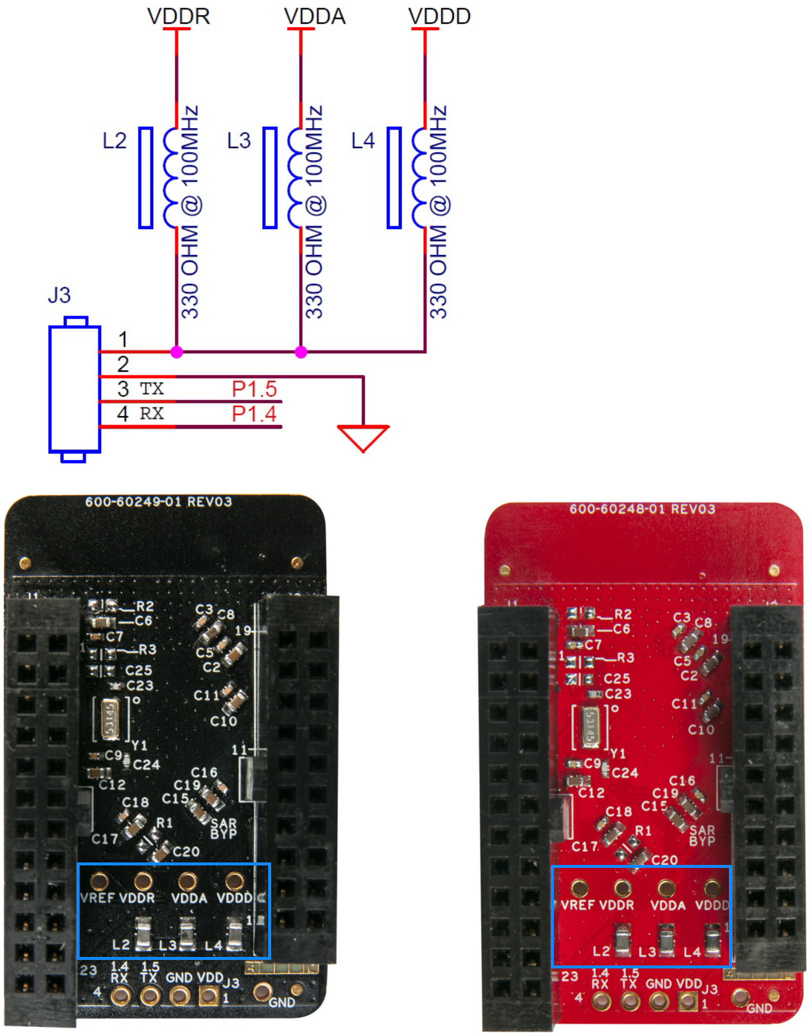


5.2.2 Module Power Connections

The module has three power domains: VDDD, VDDA, and VDDR. The VDDD connection supplies power for digital device operation, VDDA supplies power for analog device operation, and VDDR connection supplies power for the device radio. By default, these domains are shorted using a 330-ohm, 100-MHz ferrite bead. The domains are shorted for standalone usage scenarios of module, such as programming the module using MiniProg 3 or using the module as a standalone data acquisition unit.

It is recommended to place the ferrite bead between the supply to avoid ripple between VDDR and the other two domains. If the supply ripple is less that 100 mV, these can be changed to a zero-ohm resistor.

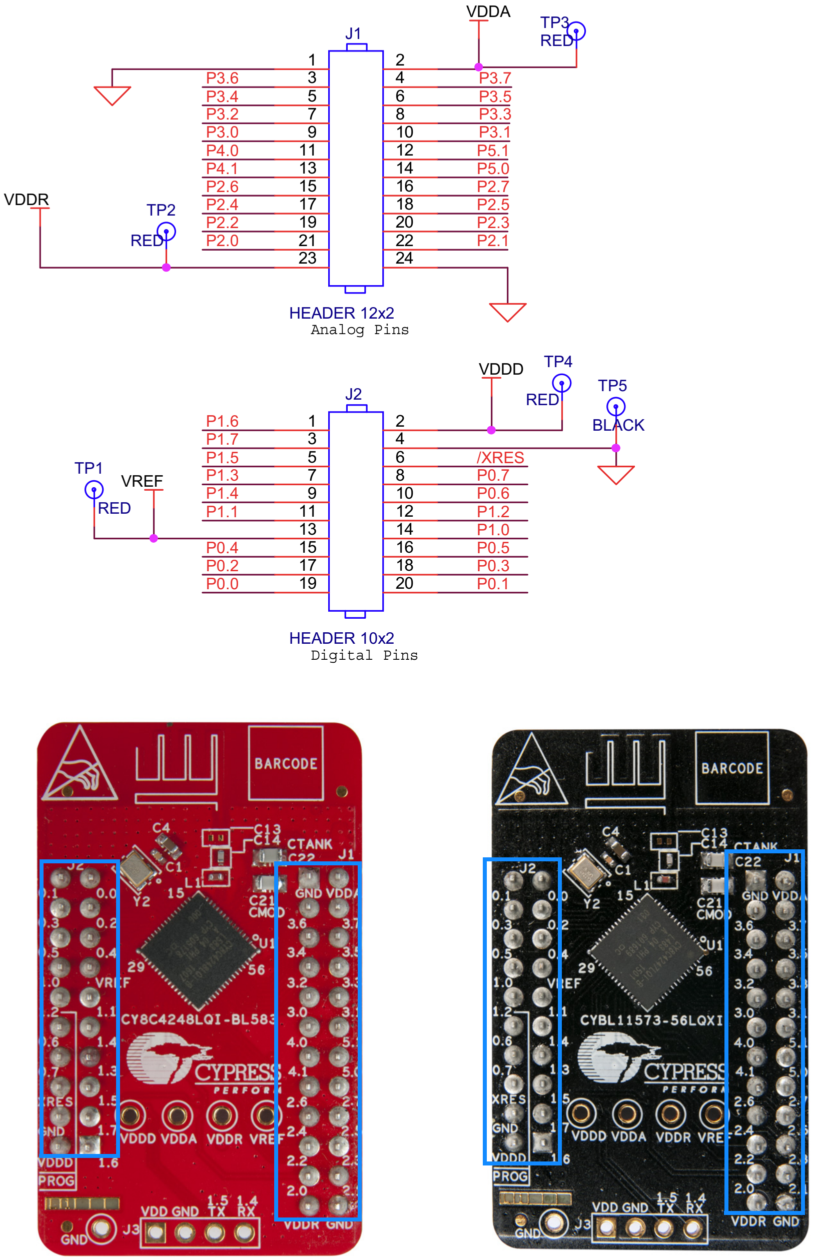
Figure 5-22. Schematics and Board Highlight of Ferrite Bead and Power Pin



5.2.3 Module Headers (20-Pin and 24-Pin Headers)

The PSoC 4 BLE and PRoC BLE Modules connect to the BLE Pioneer Baseboard using two (20-pin and 24-pin) module headers (Figure 5-23). All GPIOs and power domains are brought out to these headers. These headers are the counterparts of the connectors in Expansion Connectors on page 94.

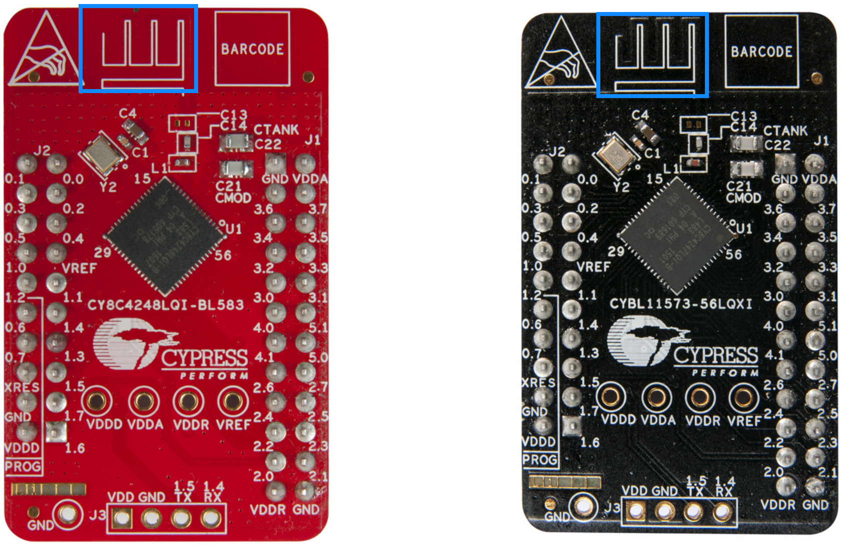
Figure 5-23. Schematics and Board Highlight of Headers



5.2.4 Wiggle Antenna

Both the modules use the wiggle antenna. Refer to the Antenna Design Guide (AN91445) for details.

Figure 5-24. Board Highlight of Wiggle Antenna

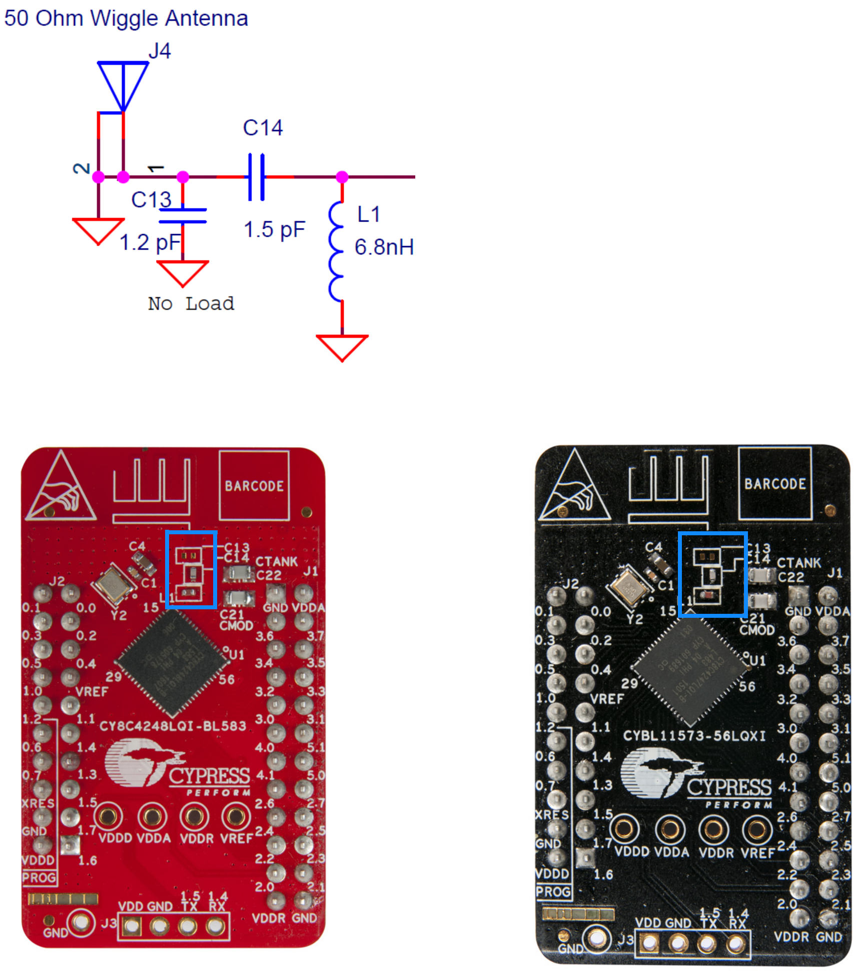


5.2.5 Antenna Matching Network

An Antenna Matching Network is required between the BLE device and the antenna to achieve optimum performance (Figure 5-25). 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 local oscillator (LO) leakage in RX mode.

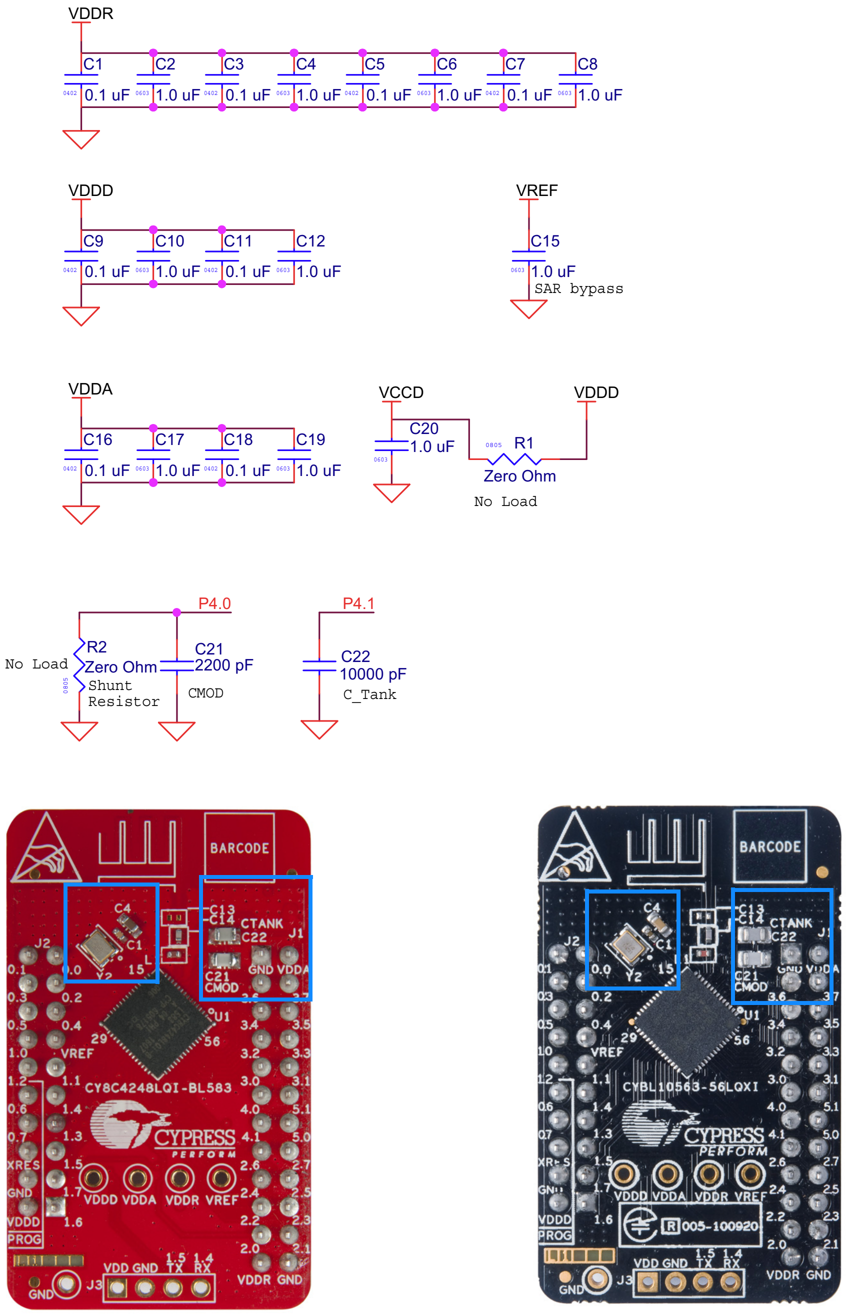
Figure 5-25. Schematics and Board Highlight of Antenna Matching Network and Antenna

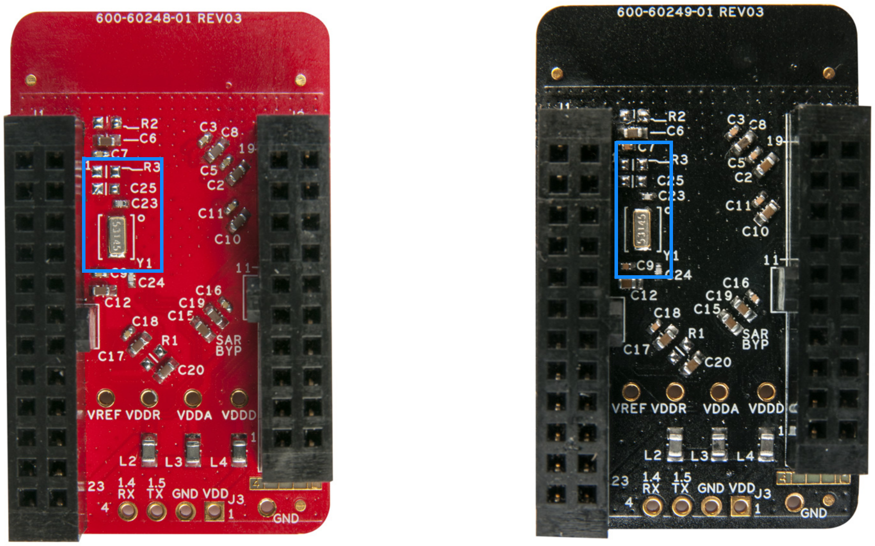


5.2.6 BLE Passives

Module boards include a 24-MHz crystal and a 32-kHz crystal, the CMOD and shield (CTANK) circuit for CapSense, a SAR bypass capacitor, and adequate decoupling capacitors for all the power domains, as shown in Figure 5-26.

Figure 5-26. Schematics and Board Highlight – External Crystal, CMOD, CTANK, Decaps, Jumpers





5.2.7 Test Points

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

5.3 BLE Dongle Board

See PSoC 4 BLE or PRoC BLE Device on page 105 for schematics of PRoC BLE pins.

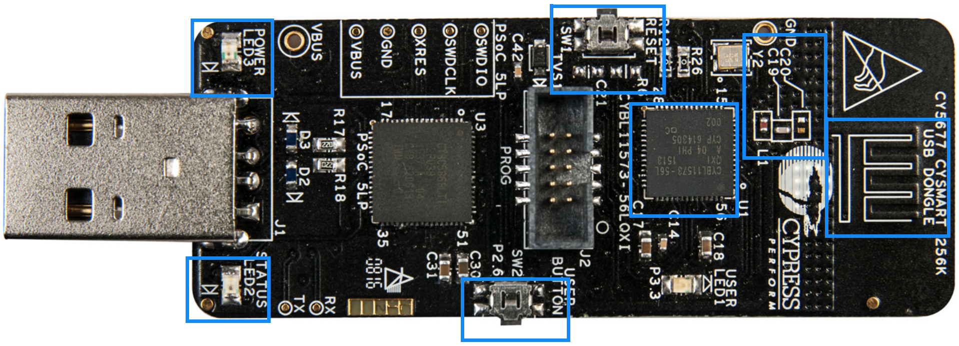
See Wiggle Antenna on page 108 for schematics of wiggle antenna.

See Antenna Matching Network on page 109 for schematics of antenna matching network.

See BLE Pioneer Baseboard LEDs on page 99 for schematics of power and status LED.

See Push-Buttons on page 101 for schematics of push-buttons.

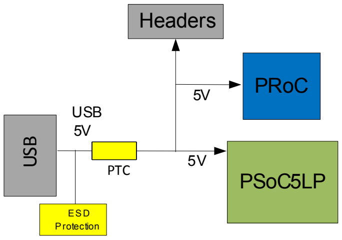
Figure 5-27. Board Highlight



5.3.1 Power System

The BLE Dongle is powered directly using 5 V from the USB port, as shown in Figure 5-28.

Figure 5-28. Power Supply Block Diagram With Protection Circuits



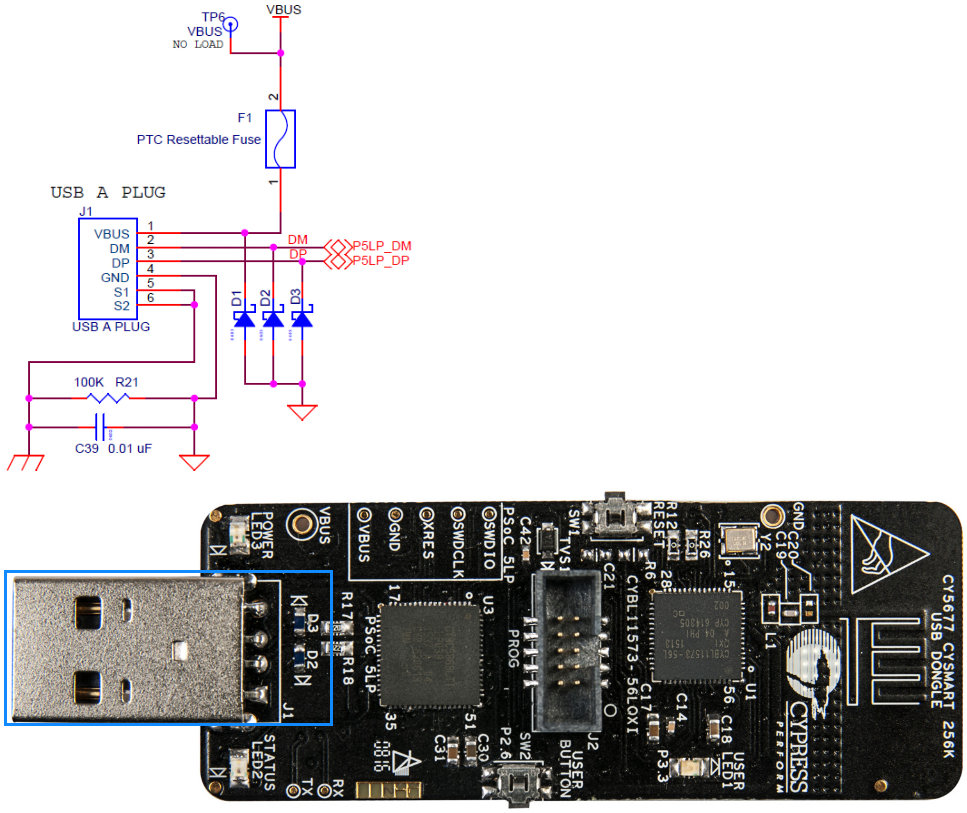
5.3.1.1 Protection Circuits

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

5.3.2 USB Type-A Plug

The KitProg on the BLE Dongle connects to the USB port of a computer through a USB Type-A plug (Figure 5-29). The BLE Dongle is powered using the same plug. 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.

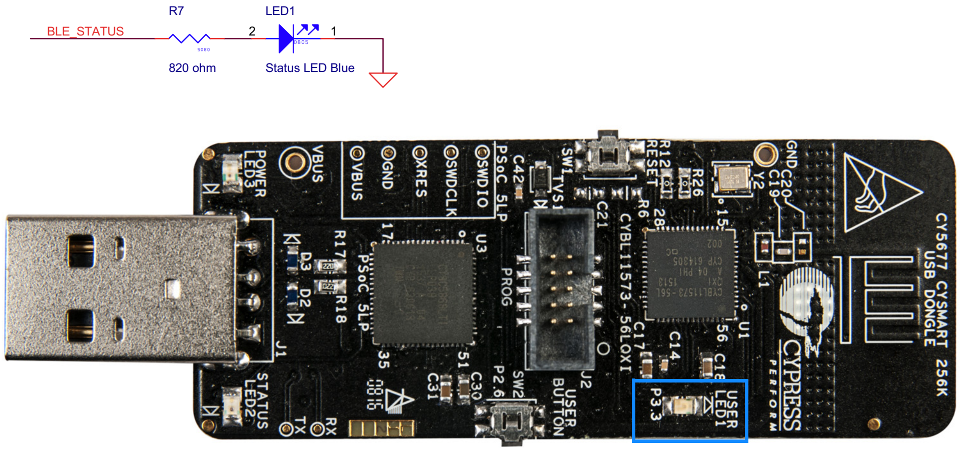
Figure 5-29. Schematics and Board Highlight of USB Type-A Plug



5.3.3 User LED

A user LED is provided to indicate status from the PRoC BLE device (Figure 5-30). It is also used to show the bind status.

Figure 5-30. Schematics and Board Highlight of User LED



6. Advanced Topics

This chapter describes the functionality of the FM24V10 F-RAM in the BLE Pioneer Kit.

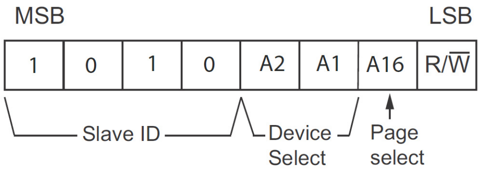
6.1 Using FM24V10 F-RAM

The BLE Pioneer Baseboard 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 (P5_0 and P5_1). 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.

6.1.1 Address Selection

The slave address of the F-RAM device consists of three parts, as shown in Figure 6-1: 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 the BLE Pioneer Baseboard (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 Baseboard is controlled by resistors R32/R36 (A1) and R33/R37 (A2). Because the memory location in F-RAM is divided into two pages of 64KB 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 6-1. F-RAM I^2C Address Byte Structure



6.1.2 Write/Read Operation

The device datasheet includes details on how to perform a write/read operation with the F-RAM. Figure 6-2 and Figure 6-3 provide a snapshot of the write/read packet structure as a quick reference.

Figure 6-2. F-RAM Single-Byte and Multiple-Byte Write Packet Structure

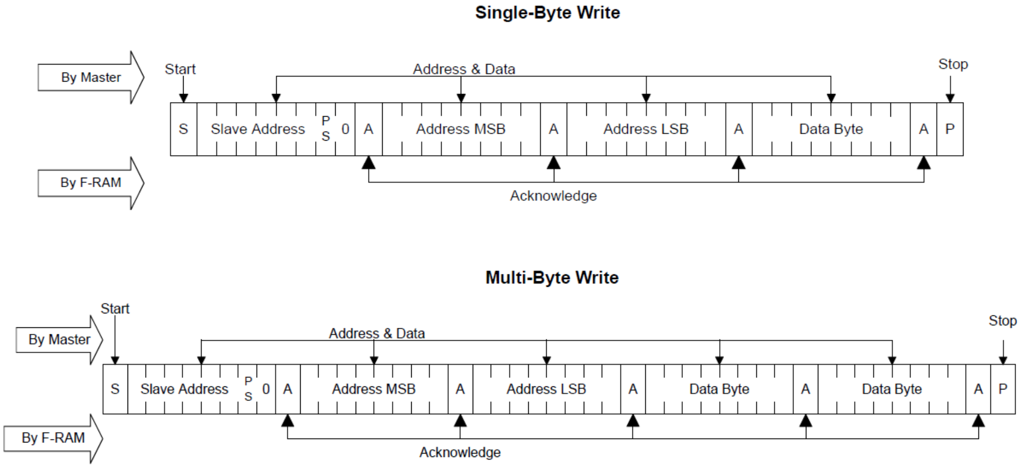
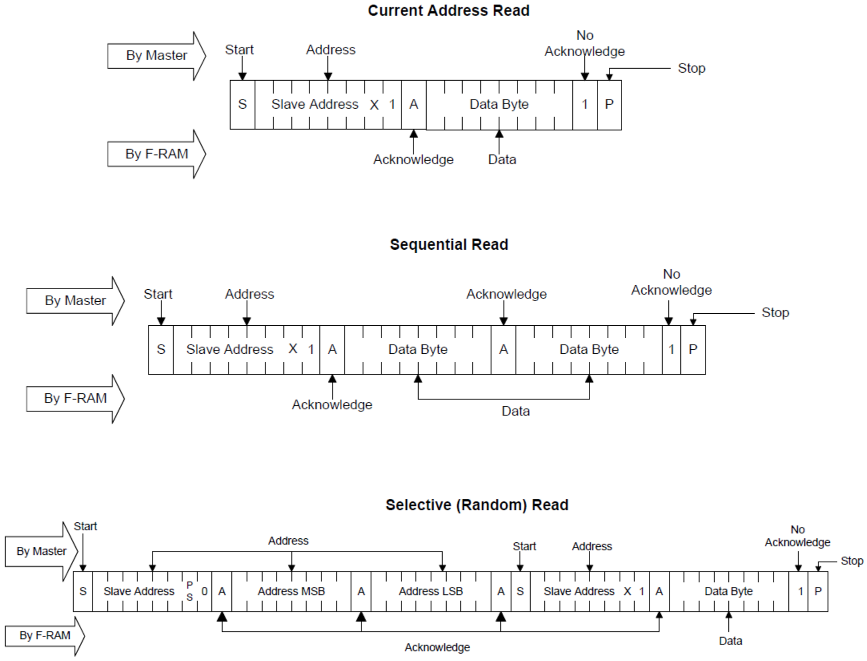


Figure 6-3. F-RAM Single-Byte and Multiple-Byte Read Packet Structure



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.

A. Appendix

A.1 Bill of Materials (BOM)

A.1.1 BLE Pioneer Baseboard

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
1	1	PCB	-	Printed circuit board	Cypress	600-60194-01, Rev04
2	1	BT1	CR2032 Battery Holder	HOLDER COIN CELL CR2032 EJECT	MPD	BA2032
3	1	C1	1.0 uF	CAP TANT 1UF 35V 10% 1210	AVX Corporation	TAJB105K035RNJ
4	1	C2	4.7 uF	CAP TANT 4.7UF 20V 10% 1210	AVX Corporation	TAJB475K020RNJ
5	1	C3	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	TDK Corporation	C1005X7R1C103K050BA
6	1	C4	100 uFd	CAP CER 100UF 6.3V 20% X5R 1210	TDK Corporation	C3225X5R0J107M250AC
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	C1005X5R1A104K050BA
8	7	C6,C7,C11,C13,C15,C16,C20	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
9	1	C29	33 uF	CAP CER 33UF 6.3V 20% X5R 0805	TDK Corporation	C2012X5R0J336M125AC
10	6	D1,D2,D3,D4,D5, D10	MBR0520L	DIODE SCHOTTKY 0.5A 20V SOD-123	Fairchild Semiconductor	MBR0520L
11	3	D6,D7,D8	ESD diode	SUPPRESSOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603MLC-05LE
12	1	D9	3.9V Zener	DIODE ZENER 3.9V 500MW SOD12	Diodes Inc	BZT52C3V9-7-F
13	1	D11	2.7V Zener	DIODE ZENER 2.7V 500MW SOD123	ON Semiconductor	MMSZ4682T1G
14	1	F1	FUSE	PTC RESETTABLE .50A 15V 1812	Bourns	MF-MSMF050-2
15	2	J1, J4	8x1 RECP	CONN HEADER FEMALE 8POS .1" GOLD	Protectron Electromech	P9401-08-21
16	1	J2	6x2 RECP	CONN HEADER FMAL 12PS.1" DL GOLD	Protectron Electromech	P9403-12-21

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
17	1	J3	10x1 RECP	CONN HEADER FMALE 10POS .1" GOLD	Protectron Electro-mech	P9401-10-21
18	1	J6	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K
19	1	J8	8X2 RECP	CONN HEADER FMAL 16PS.1" DL GOLD	Protectron Electro-mech	P9403-16-21
20	1	J10	12X2 RECP	CONN HEADER 2.54MM 24POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D12-ST-BK
21	1	J11	10X2 RECP	CONN HEADER 2.54MM 20POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D10-ST-BK
22	1	J13	USB MINI B	MINI USB RCPT R/A DIP	TE Connectivity	1734510-1
23	1	J14	1X1 RECP	CONN RCPT 1POS .100" SNGL HORZ	Samtec Inc	BCS-101-L-S-HE
24	1	J15	2p_jumper	CONN HEADR BRKWAY .100 2POS STR	Protectron Electro-mech	P9101-02-12-1
25	1	J16	3p_jumper	CONN HEADR BRKWAY .100 3POS STR	Protectron Electro-mech	P9101-03-12-1
26	1	LED1	Power LED Amber	LED 595NM AMB DIFF 0805 SMD	Avago Technologies	HSMA-C170
27	1	LED2	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
28	1	LED3	RGB LED	LED RED/GREEN/BLUE PLCC4 SMD	Cree, Inc.	CLV1A-FKB-CJ1M1F1BB7R4S3
29	3	L1,L2,L3	330 OHM @ 100MHz	FERRITE CHIP 330 OHM 0805	Murata	BLM21PG331SN1D
30	3	Q2,Q4,Q6	PMOS	MOSFET P-CH 30V 2.2A SOT23	ON Semiconductor	NTR4171PT1G
31	1	Q1,	PMOS	MOSFET P-CH 30V 3.8A SOT23-3	Diodes Inc	DMP3098L-7
32	2	Q3,Q5	PMOS	MOSFET P-CH 20V 3.5A SOT23	NXP Semiconductors	PMV48XP,215
33	1	R1	11K 1%	RES 11K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1102V
34	1	R2	560 ohm	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ561V
35	1	R3	14.7K 1%	RES 14.7K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1472V
36	1	R4	10K 1%	RES 10K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1002V
37	1	R5	4.3K 1%	RES 4.3K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF4301V
38	1	R6	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X
39	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

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
40	2	R8,R58	15K	RES 15K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1502V
41	2	R9,R20	10K 1%	RES 10K OHM 1/8W 1% 0805 SMD	Stackpole Electronics Inc	RMCF0805FT10K0
42	1	R10	10K	RES 10K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ103V
43	1	R11	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ821V
44	2	R13,R14	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
45	2	R15,R16	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V
46	2	R17,R18	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V
47	5	R22,R23,R28,R31,R35	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V
48	2	R24,R25	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V
49	2	R29,R30	1.5K	RES 1.5K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ152V
50	5	R39,R40,R41,R42,R43	560 ohm	RES 560 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ561V
51	2	SW1,SW2	SW PUSH-BUTTON	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-PE105K
52	1	TP5	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001
53	2	TVS1,TVS2	5V 350W	TVS UNIDIR 350W 5V SOD-323	Dioded Inc.	SD05-7
54	1	U1	LDO	IC REG LDO ADJ 1A TO252-5	Rohm Semiconductor	BA00BC0WFP-E2
55	1	U2	PSoC 5LP	68QFN PSoC 5LP chip for USB debug channel and USB-Serial interface	Cypress Semiconductor	CY8C5868LTI-LP039
56	1	U3	F-RAM	F-RAM 1-Mbit (128K X 8) I2C interface	Cypress Semiconductor	FM24V10-G
57	1	U4	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G
Install on Bottom of PCB As per the Silk Screen in the Corners						
58	4	N/A	N/A	BUMPER CYLIN 0.375" DIA BLK	3M	SJ61A4
Special Jumper Installation Instructions						
59	2	J15,J16	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
Label						
60	1	N/A	N/A	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60329-01 Rev 04 (YYWWVVXXXXX)	Cypress Semiconductor	
61	1	N/A	N/A	LBL, QR code, 12mm X 12mm	Cypress Semiconductor	
No load components						
62	1	C22	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA
63	1	C24	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
64	9	R7,R59,R32,R33,R34,R48,R49,R50,R51	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1
65	1	R21	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V
66	1	J7	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K
67	1	J9	2 PIN HDR	CONN HEADER FEMALE 2POS .1" GOLD	Sullins Connector Solutions	PPPC021LFBN-RC
68	2	TP4,TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001
69	3	TP1,TP2,TP3	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000
70	2	R44,R12	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
71	1	J12	3x2 RECPT	CONN HEADER FMAL 6PS .1" DL GOLD	Sullins Connector Solutions	PPPC032LFBN-RC
72	1	J5	6X1 RECP RA	CONN FEMALE 6POS .100" R/A GOLD	Sullins Connector Solutions	PPPC061LGBN-RC

A.1.2 Module

A.1.2.1 CY5676A PRoC BLE 256KB Module

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number
1	1	PCB	-	PRoC BLE 256KB Module printed circuit board	Cypress qualified vendor	600-60249-01 Rev 03
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
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
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222JA01D
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103JA01D
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC-D12-ST-BK
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC-D10-ST-BK
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technology Inc	L-07C6N8JV6T
12	3	L2,L3,L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1D
13	1	U1	PRoC BLE	56 QFN PRoC BLE - 256KB	Cypress Semiconductor	CYBL11573-56LQXI
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60216-01 Rev 01 (YYWWV-VXXXXX)	Cypress qualified vendor	-
No Load components						
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T
18	1	C25	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU
19	1	R1	Zero Ohm	RES 0.0 OHM 1/8W 0605 SMD	TE Connectivity	1623094-1

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

A.1.2.2 CY8CKIT-143A PSoC 4 BLE 256KB Module

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number
1	1	PCB	-	PSoC 4 BLE 256KB Module printed circuit board	Cypress qualified vendor	600-60248-01 Rev 03
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
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
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222JA01D
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103JA01D
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC-D12-ST-BK
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC-D10-ST-BK
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technology Inc	L-07C6N8JV6T
12	3	L2,L3,L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1D
13	1	U1	PSoC 4BLE	56 QFN PSoC 4 BLE - 256KB	Cypress Semiconductor	CY8C4248LQI-BL583
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60215-01 Rev 01 (YYWWVVXXXXX)	Cypress qualified vendor	-
No Load components						
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T
18	1	C25	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU
19	1	R1	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1
20	1	R2	Rbleed	No Load	-	-
21	1	R3	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V
22	1	J3	4 HEADER	CONN HEADER 4POS .100 R/A 15AU	FCI	68016-204HLF
23	4	TP1,TP2,TP3,TP4	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000
24	1	TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001

A.1.3 CY5677 CySmart BLE 4.2 USB Dongle

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
1	1	-	-	BLE Dongle Printed circuit board	Cypress qualified vendor	600-60326-01 Rev01
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
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
4	1	C19	1.2 pFd	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T
5	1	C22	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D
6	1	C23	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D
7	1	C39	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	TDK Corporation	C1005X7R1C103K050BA
8	3	D1,D2,D3	ESD diode	SUPPRESSOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603MLC-05LE
9	1	F1	FUSE	PTC RESETTABLE .50A 15V 1812	Bourns	MF-MSMF050-2
10	1	J1	USB A PLUG	CONN PLUG USB 4POS RT ANG PCB	Molex Inc	480370001
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
13	1	LED2	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
14	1	LED3	Power LED Red	LED SUPER RED CLEAR 0805 SMD	LiteOn Inc	LTST-C170KRKT
15	1	L1	5.1 nH	CER INDUCTOR 5.1NH 0402	Johanson Technology Inc	L-07C5N1SV6T
16	2	R8,R11	Zero Ohm	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
17	1	R7	820 ohm	RES 820 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ821V
18	2	R22,R25	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ821V
19	2	R9,R10	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
20	9	R1,R2,R3,R4,R12,R13,R14,R15,R26	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V
21	2	R17,R18	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V
22	1	R21	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X
23	2	R19,R20	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V
24	2	R23,R24	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V
25	2	SW1,SW2	SW RA PUSH	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-P3401P
26	1	TVS1	5V 350W	TVS UNIDIR 350W 5V SOD-323	Diodes Inc.	SD05-7
27	1	U1	PRoC BLE	PRoC BLE, 56QFN 256KB	Cypress Semiconductor	CYBL11573-56LQXI
28	1	U2	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G
29	1	U3	PSoC 5LP	PSoC 5LP Programmable System on Chip, 68QFN	Cypress Semiconductor	CY8C5868LTI-LP039
30	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-34B
31	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM
32	1	N/A	N/A	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60161-01 Rev 03 (YYWWV-VXXXXX); Only barcode	Cypress qualified vendor	-
No load components						
33	1	C20	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T
34	1	C21	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU
35	1	C37	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA
36	1	C27	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
37	1	R5	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1
38	2	R6,R16	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V
39	15	TP1,TP2,TP3,TP4,TP5,TP6,TP7,TP8,TP9,TP10,TP11,TP12,TP13,TP14,TP15	No load	No load	-	-

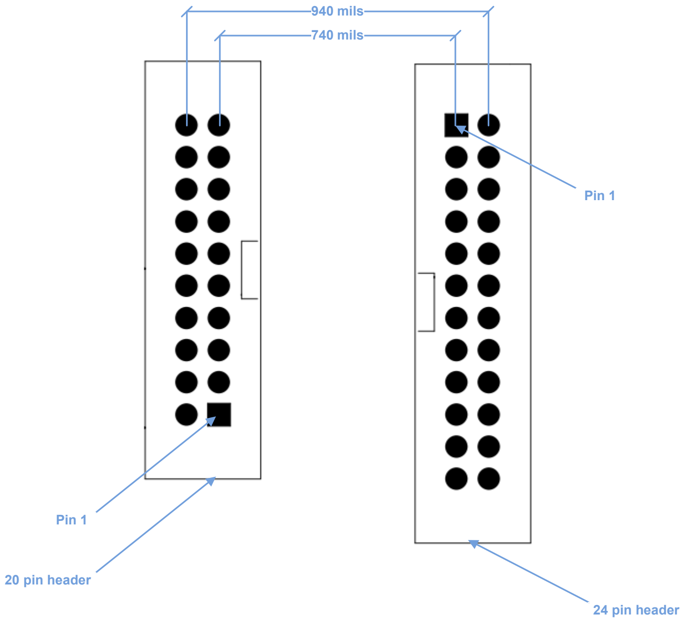
A.2 KitProg Status LED States

User Indication	Scenario	Action Required by user
LED blinks fast: Frequency = 4.00 Hz	LED starts blinking at power up, if bootloadable file is corrupt.	Bootload the *KitProg.cyacd* file: In PSoC Programmer, connect to the kit, open the **Utilities** tab, and press the **Upgrade Firmware** button.
LED blinks slow: Frequency = 0.67 Hz	Entered Bootloader mode by holding the Reset button of the BLE Pioneer Kit/BLE Dongle during kit power up.	Release the Reset button and replug power if you entered this mode by mistake. If the mode entry was intentional, bootload the new *.cyacd* file using the Bootloader Host tool shipped with PSoC Creator.
LED blinks very fast: Frequency = 15.0 Hz	SWD operation is in progress. Any I^{2}C traffic. Kit's COM port connect/disconnect event (one blink).	In PSoC Programmer, watch the log window for status messages for SWD operations. In the Bridge Control Panel, the LED blinks on I^{2}C command requests. In Bridge Control Panel or any other serial port terminal program, distinguish the kit's COM port number by the blinking LED when the port is connected or disconnected.
LED is ON.	USB enumeration successful. Kit is in the idle state waiting for commands.	The kit functions can be used by PSoC Creator, PSoC Programmer, Bridge Control Panel, and any serial port terminal program.
LED is OFF.	Power LED is ON.	This means that the USB enumeration was unsuccessful. This can happen if the kit is not powered from the USB host or the kit is not connected to the USB host through the USB cable. Verify the USB cable and check if PSoC Programmer is installed on the PC.

A.3 Adding BLE Module-Compatible Headers on Your Baseboard

The baseboard should have a 20-pin header and a 24-pin header. Dimension of these connectors are given here.

Figure A-1. Connectors on BLE Pioneer Kit Baseboard



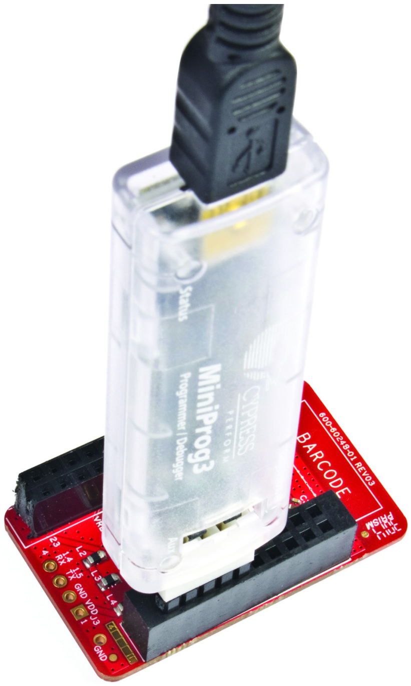
These headers are available for purchase from Digikey.

Description	Manufacturer	Manufacturer Part Number	Digikey Part Number
CONN HEADER 2.54MM 24POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D12-ST-BK	SBH11-PBPC-D12-ST-BK-ND
CONN HEADER 2.54MM 20POS GOLD	Sullins Connector Solutions	SBH11-PBPC-D10-ST-BK	S9172-ND

A.4 Programming BLE Modules via MiniProg3

If the BLE Modules are to be used without the BLE Pioneer Baseboard, they can be programmed using MiniProg3. The J2 header has five adjacent pins – VDDD, GND, XRES, P0[7], and P0[6]. These pins can be used to program the BLE Module using MiniProg3.

Figure A-2. Programming a BLE Module via MiniProg3



Follow these steps to program the module:

1. Connect the MiniProg3 to the J2 connector, with the VDD of the MiniProg3 aligned to the VDDD on the module.
2. Click **Start** > **All Programs** > **Cypress** > **PSoC Programmer <version>** > **PSoC Programmer <version>**.
3. Open the desired *.hex* file in PSoC Programmer.
4. On the **Programmer** tab, set the **Programming Mode** to **Reset**.
5. Set **AutoDetection** to **On**.
6. Set **Programmer Characteristics > Protocol** to **SWD**.
7. Set **Programmer Characteristics > Voltage** to the desired value.
8. Click the **Toggle Power** icon below the menu bar to power the module.
9. Click the **Program** icon below the menu bar to program the module.

A.5 BLE Modules and BLE Dongles Compatible with the BLE Pioneer Kit

Different BLE modules and BLE dongles can work with the BLE Pioneer Kit, as listed in the following tables.

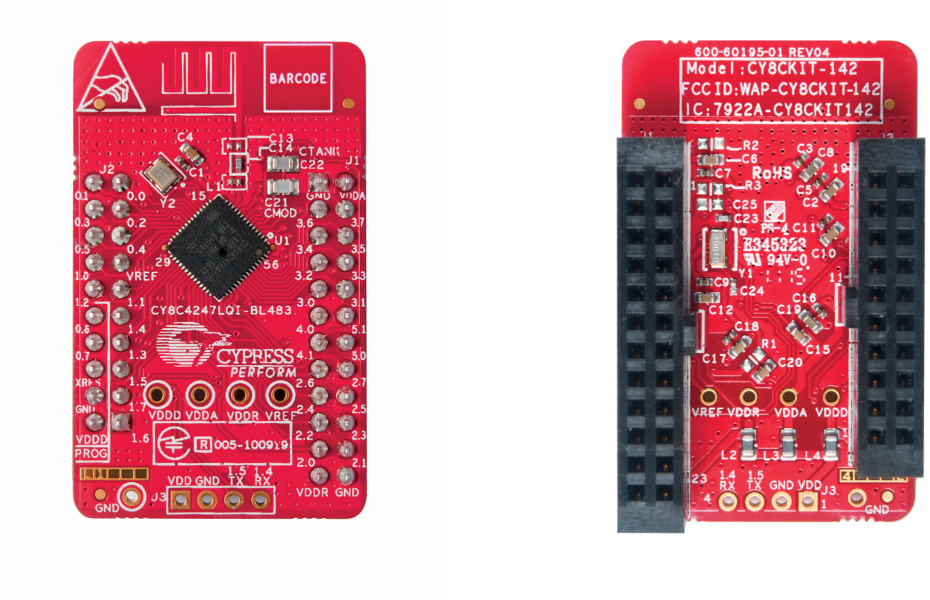
BLE Module	Availability	Flash Size	Bluetooth Version
CY8CKIT-142 PSoC 4 BLE Module	Available separately	128KB	Bluetooth 4.1
CY8CKIT-141 PSoC 4 BLE SMA Module	Available separately	128KB	Bluetooth 4.1
CY8CKIT-143 PSoC 4 BLE 256KB Module	Available separately	256KB	Bluetooth 4.1
CY8CKIT-143A PSoC 4 BLE 256KB Module	As part of the kit	256KB	Bluetooth 4.2
CY5671 PRoC BLE Module	Available separately	128KB	Bluetooth 4.1
CY5674 PRoC BLE SMA Module	Available separately	128KB	Bluetooth 4.1
CY5676 PRoC BLE 256KB Module	Available separately	256KB	Bluetooth 4.1
CY5676A PRoC BLE 256KB Module	As part of the kit	256KB	Bluetooth 4.2

BLE Dongle	Availability	Flash Size	Bluetooth Version
CY5670 CySmart USB Dongle (BLE Dongle)	Available separately	128KB	Bluetooth 4.1
CY5677 CySmart BLE 4.2 USB Dongle (BLE Dongle)	As part of the kit	256KB	Bluetooth 4.2

A.5.1 CY8CKIT-142 PSoC 4 BLE Module

This is the lower flash version of the PSoC 4 BLE Module. It can be ordered separately. This module has the CY8C4247LQI-BL483 silicon, with 128KB flash and 16KB RAM. It supports Bluetooth 4.1.

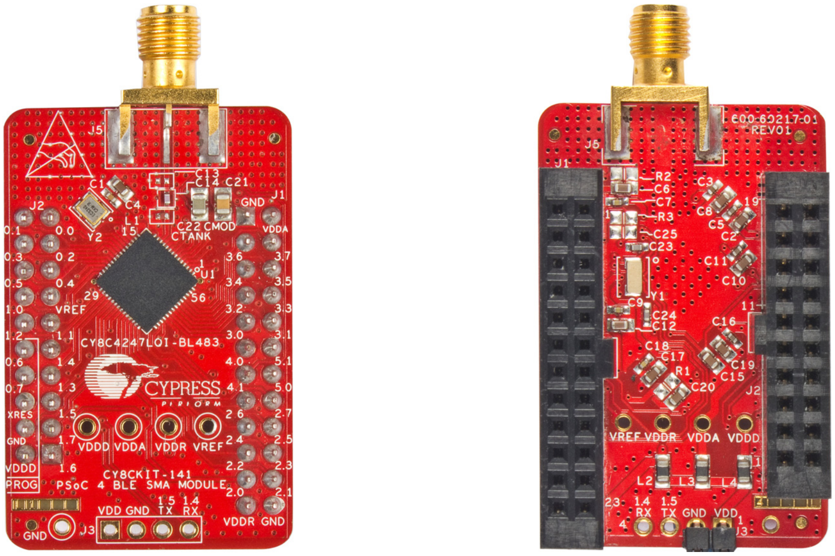
Figure A-3. CY8CKIT-142 PSoC 4 BLE Module



A.5.2 CY8CKIT-141 PSoC 4 BLE SMA Module

This module is identical to the CY8CKIT-142 PSoC 4 BLE Module, except that it has an SMA connector instead of a wiggle antenna; this connector can be used to connect to an external antenna. This module can be ordered separately.

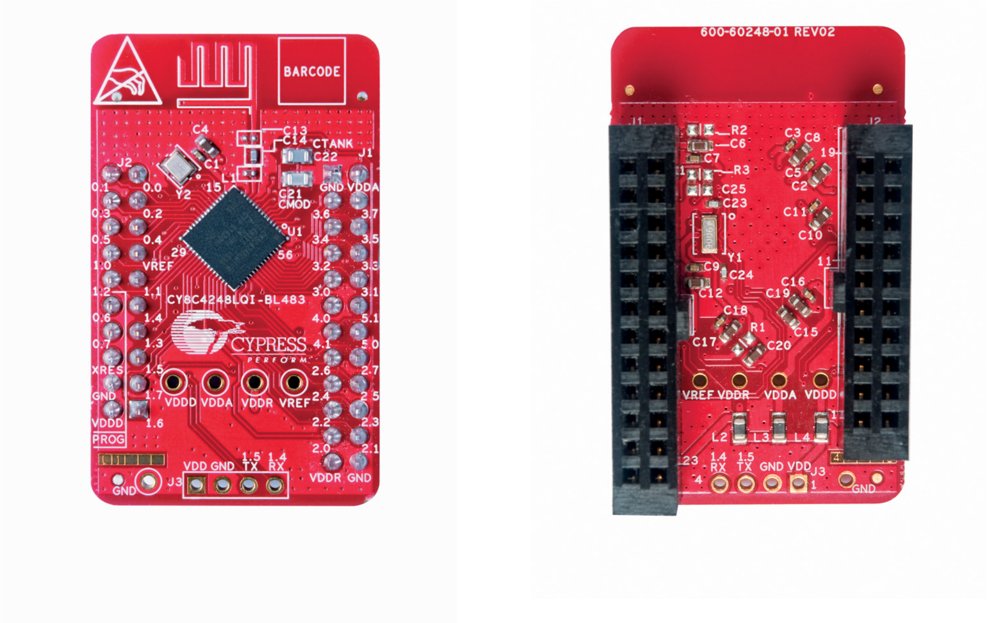
Figure A-4. CY8CKIT-141 PSoC 4 BLE SMA Module



A.5.3 CY8CKIT-143 PSoC 4 BLE 256KB Module

This is the Bluetooth 4.1 equivalent of the CY8CKIT-143A PSoC 4 BLE 256KB Module. It has the CY8C4248LQI-BL483 silicon, with 256KB flash and 32KB RAM. It can be ordered separately.

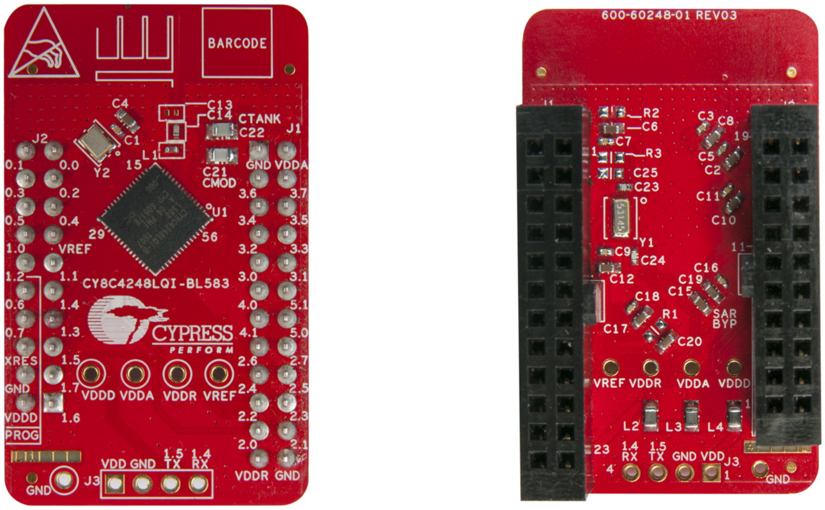
Figure A-5. CY8CKIT-143 PSoC 4 BLE 256KB Module



A.5.4 CY8CKIT-143A PSoC 4 BLE 256KB Module

This is the default PSoC 4 BLE Module shipped as part of the BLE Pioneer Kit. It supports Bluetooth 4.2 features and DMA. It has the CY8C4248LQI-BL583 silicon, with 256KB flash and 32KB RAM.

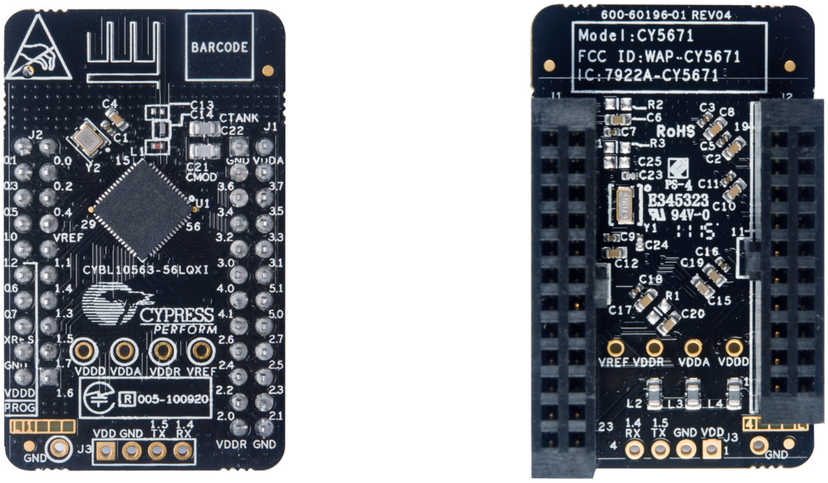
Figure A-6. CY8CKIT-143A PSoC 4 BLE 256KB Module



A.5.5 CY5671 PRoC BLE Module

This is the lower flash version of the PRoC BLE Module. It can be ordered separately. This module has the CYBL10563-56LQXI silicon, with 128KB flash and 16KB RAM. It supports Bluetooth 4.1.

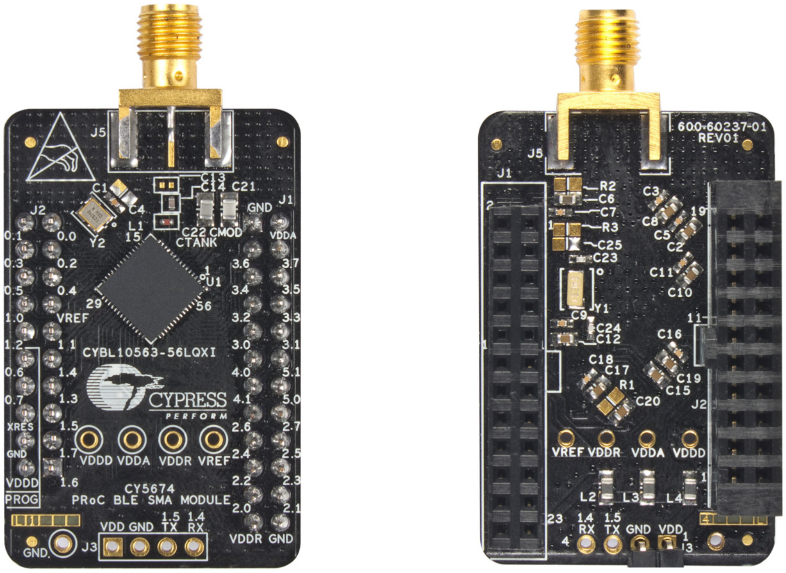
Figure A-7. CY5671 PRoC BLE Module



A.5.6 CY5674 PRoC BLE SMA Module

This module is identical to the CY5671 PRoC BLE Module, except that it has an SMA connector instead of a wiggle antenna; this connector can be used to connect to an external antenna. This module can be ordered separately.

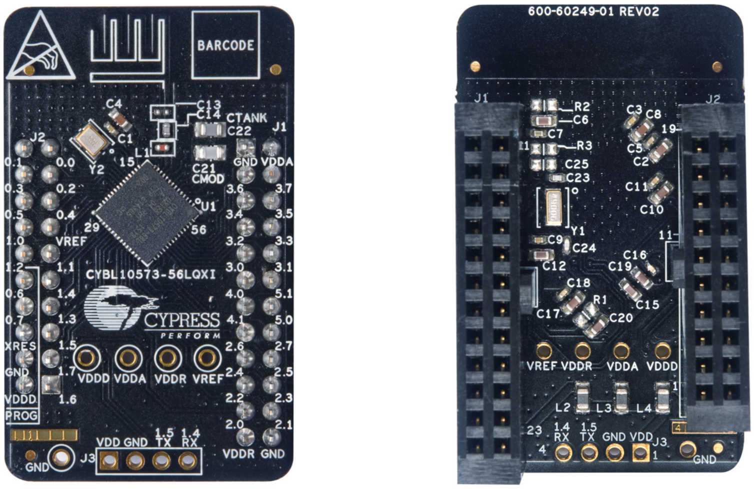
Figure A-8. CY5674 PRoC BLE SMA Module



A.5.7 CY5676 PRoC BLE 256KB Module

This is the Bluetooth 4.1 equivalent of the CY5676A PRoC BLE 256KB Module. It has the CYBL10573-56LQXI silicon, with 256KB flash and 32KB RAM. It can be ordered separately.

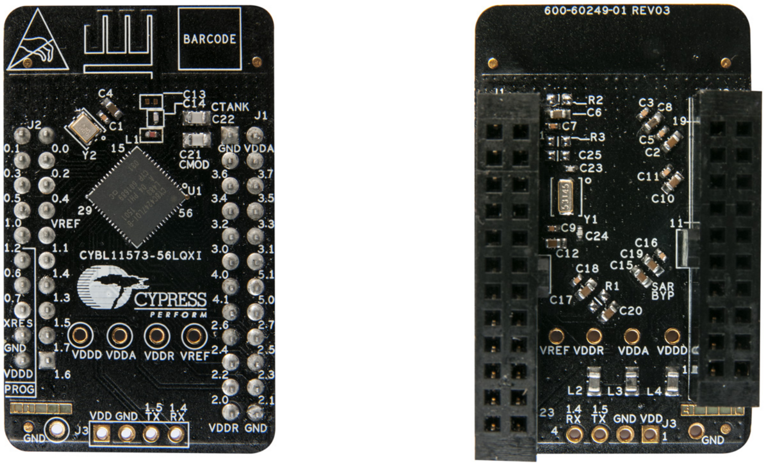
Figure A-9. CY5676 PRoC BLE 256KB Module



A.5.8 CY5676A PRoC BLE 256KB Module

This is the default PRoC BLE Module shipped as part of the BLE Pioneer Kit. It supports Bluetooth 4.2 features and DMA. It has the CYBL11573-56LQXI silicon, with 256KB flash and 32KB RAM.

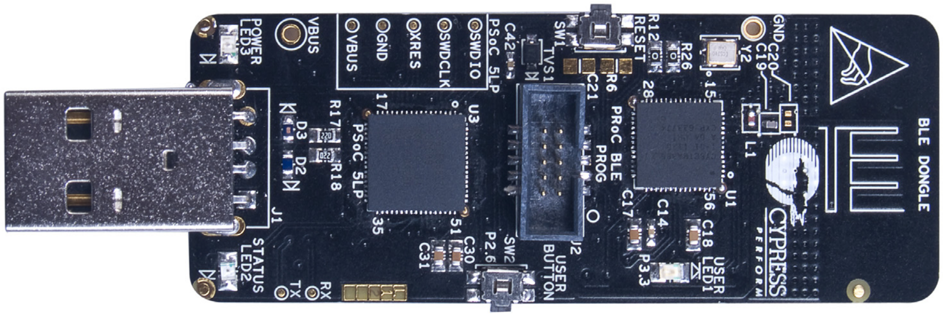
Figure A-10. CY5676A PRoC BLE 256KB Module



A.5.9 CY5670 CySmart USB Dongle (BLE Dongle)

This is the lower flash equivalent of the CY5677 CySmart BLE 4.2 USB. It can be ordered separately. This dongle has the CYBL10162-56LQXI silicon, with 128KB flash and 16KB RAM. It supports Bluetooth 4.1.

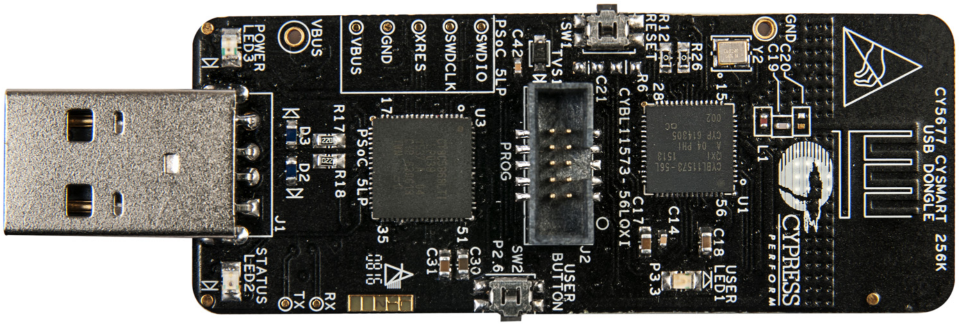
Figure A-11. CY5670 CySmart USB Dongle (BLE Dongle)



A.5.10 CY5677 CySmart BLE 4.2 USB Dongle (BLE Dongle)

This is the default BLE Dongle shipped as part of the BLE Pioneer Kit. It has the CYBL11573-56LQXI silicon, with 256KB flash and 32KB RAM. It supports Bluetooth 4.2 and DMA.

Figure A-12. CY5677 CySmart BLE 4.2 USB Dongle (BLE Dongle)

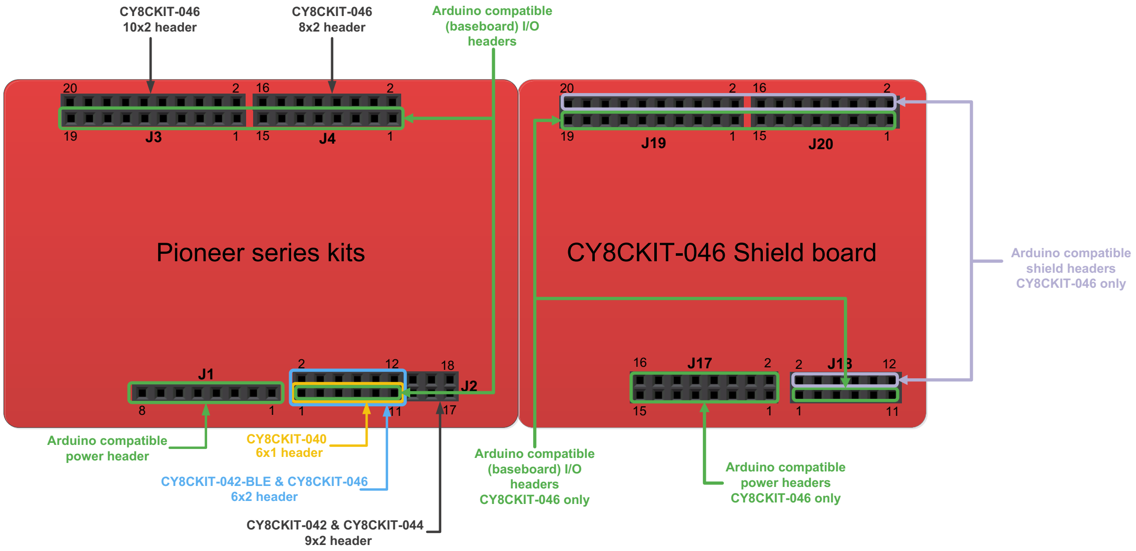


A.6 Migrating Projects Across Different Pioneer Series Kits

All Cypress Pioneer series kits are Arduino Uno-compatible and have some common onboard peripherals such as RGB LED, CapSense, and a user switch. However, the pin mapping in each of the boards is different due to differences in pin functions of the PSoC device used. This guide lists the pin maps of the Pioneer series kits to allow easy migration of projects across different kits.

In some cases, the pins available on the Pioneer kit headers are a super set of the standard Arduino Uno pins. For example, J2 contains only one row of pins on the Arduino Uno pinout while it contains two rows of pins on many of the Pioneer series kits.

Figure A-13. Pioneer Series Kits Pin Map



A.6.1 Arduino Uno-Compatible Headers

J1 Arduino-Compatible Header Pin Map						
Pin #	Arduino Pin	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	VIN	VIN	VIN	VIN	VIN	VIN
2	GND	GND	GND	GND	GND	GND
3	GND	GND	GND	GND	GND	GND
4	5V	V5.0	V5.0	V5.0	V5.0	V5.0
5	3.3V	V3.3	V3.3	V3.3	V3.3	V3.3
6	RESET	RESET	RESET	RESET	RESET	RESET
7	IOREF	P4.VDD	P4.VDD	BLE.VDD	P4.VDD	P4L.VDD
8	NC	NC	NC	NC	NC	NC

J2 Arduino-Compatible Header Pin Map						
Pin #	Arduino Pin	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	A0	P2[0]	P0[0]	P3[0]	P2[0]	P2[0]
2	–	P0[2][a]	–	P2[0]	P2[6][a]	P3[6][a]
3	A1	P2[1]	P0[1]	P3[1]	P2[1]	P2[1]
4	–-	P0[3][a]	–	P2[1][a]	P6[5][a]	P3[7][a]
5	A2	P2[2]	P0[2][a]	P3[2]	P2[2]	P2[2]
6	–	P4_VDD	–	P2[2][a]	P0[6][a]	P9[0]
7	A3	P2[3]	P0[4][a]	P3[3]	P2[3]	P2[3]
8	–	P1[5][a]	–	P2[3][a]	P4[4][a]	P9[1]
9	A4	P2[4]	P1[3]	P3[4]	P2[4]	P2[4]
10	–	P1[4][a]	–	P2[4][a]	P4[5][a]	P9[2]
11	A5	P2[5]	P1[2]	P3[5]	P2[5]	P2[5]
12	–	P1[3][a]	–	P2[5][a]	P4[6][a]	P9[3]
13	–	P0[0]	–	–	P0[0]	–
14	–	GND	–	–	GND	–
15	–	P0[1]	–	–	P0[1]	–
16	–	P1[2][a]	–	–	P3[4][a]	–
17	–	P1[0]	–	–	P0[7][a]	–
18	–	P1[1][a]	–	–	P3[5][a]	–

[a] These pins are also used for onboard peripherals. See the tables in "Onboard Peripherals" on page 138 for connection details.

J3 Arduino-Compatible Header Pin Map						
#	Arduino Pin	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	D8	P2[6]	P1[4]	P0[5]	P0[2]	P0[2]
2	-	-	-	-	-	P3[4]
3	D9	P3[6]	P1[5]	P0[4]	P0[3]	P0[3]
4	-	-	-	-	-	P6[5]
5	D10	P3[4]	P1[6]	P0[2]	P2[7]	P6[3]
6	-	-	-	-	-	P6[3]
7	D11	P3[0]	P1[1][a]	P0[0]	P6[0]	P6[0]
8	-	-	-	-	-	P6[0]
9	D12	P3[1]	P3[1]	P0[1]	P6[1]	P6[1]
10	-	-	-	-	-	P6[1]
11	D13	P0[6]	P1[7]	P0[3]	P6[2]	P6[2]
12	-	-	-	-	-	P6[2]
13	GND	GND	GND	GND	GND	GND
14	-	-	-	-	-	GND
15	AREF	P1[7]	NC	VREF	P1[7]	VREF
16	-	-	-	-	-	VREF
17	SDA	P4[1]	P1[3]	P3[4]	P4[1]	P4[1]
18	-	-	-	-	-	P4[1]
19	SCL	P4[0]	P1[2]	P3[5]	P4[0]	P4[0]
20	-	-	-	-	-	P4[0]

[a] These pins are also used for onboard peripheral connections. Refer to the A.6.2 Onboard Peripherals section for connection details.

J4 Arduino-Compatible Header Pin Map						
#	Arduino Pin	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	D0	P0[4]	P0[5]	P1[4]	P3[0]	P3[0]
2	-	-	-	-	-	P8[0]
3	D1	P0[5]	P0[6]	P1[5]	P3[1]	P3[1]
4	-	-	-	-	-	P8[1]
5	D2	P0[7][a]	P0[7]	P1[6]	P1[0]	P1[0]
6	-	-	-	-	-	P8[2]
7	D3	P3[7]	P3[2][a]	P1[7]	P1[1]	P1[1]
8	-	-	-	-	-	P8[3]
9	D4	P0[0]	P0[3]	P1[3]	P1[2]	P1[2]
10	-	-	-	-	-	P8[4]
11	D5	P3[5]	P3[0]	P1[2]	P1[3]	P1[3]
12	-	-	-	-	-	P8[5]
13	D6	P1[0]	P1[0]	P1[1]	P5[3]	P5[6]
14	-	-	-	-	-	P8[6]
15	D7	P2[7]	P2[0][a]	P1[0]	P5[5]	P5[5]
16	-	-	-	-	-	P8[7]

a. These pins are also used for onboard peripheral connections. Refer to the A.6.2 Onboard Peripherals section for connection details.

A.6.2 Onboard Peripherals

#	CapSense Pin	Pioneer Series Kits				
		CY8CKIT-042 (Linear Slider)	CY8CKIT-040	CY8CKIT-042-BLE (Linear Slider)	CY8CKIT-044 (Gesture Pad)	CY8CKIT-046 (Gesture Pad with Radial Slider)[a]
1	CapSense Sensor 1	P1[1]/ CS_LS_E0	–	P2[1]/CS_LS_E0	P4[4]/ CS_GES_CR	P0[6]/ CS_GES_CR
2	CapSense Sensor 2	P1[2]/ CS_LS_E1	–	P2[2]/CS_LS_E1	P4[5]/ CS_GES_UP	P4[5]/ CS_GES_LT
3	CapSense Sensor 3	P1[3]/ CS_LS_E2	–	P2[3]/CS_LS_E2	P4[6] /CS_GES_LT	P4[4]/ CS_GES_UP
4	CapSense Sensor 4	P1[4]/ CS_LS_E3	–	P2[4]/CS_LS_E3	P3[4]/ CS_GES_DN	P4[7]/ CS_GES_RT
5	CapSense Sensor 5	P1[5]/ CS_LS_E4	–	P2[5]/CS_LS_E4	P3[5]/ CS_GES_RT	P4[6]/ CS_GES_DN
6	CapSense Sensor 10	–	–	–	–	P7[4]/ CS_RS_E0
7	CapSense Sensor 11	–	–	–	–	P7[5]/ CS_RS_E1
8	CapSense Sensor 12	–	–	–	–	P7[6]/ CS_RS_E2
9	CapSense Sensor 13	–	–	–	–	P7[7]/ CS_RS_E3
10	CapSense Sensor 6	–	–	–	–	P0[0]/ CS_RS_E4
11	CapSense Sensor 7	–	–	–	–	P0[1]/ CS_RS_E5
12	CapSense Sensor 8	–	–	–	–	P7[2]/ CS_RS_E6
13	CapSense Sensor 9	–	–	–	–	P7[3]/ CS_RS_E7
14	CMOD[b]	P4[2]	P0[4]	P4[0]	P4[2]	P4[2]
15	CTANK[b]	P4[3]	P0[2]	P4[1]	P4[3]	P4[3]
16	CMOD[b]	–	–	–	–	P5[0]
17	CTANK[b]	–	–	–	–	P5[1]
18	CapSense Shield	P0[1]	–	P1[6]	P0[1]	P0[2]

[a] The CapSense elements are present on the CY8CKIT-046 shield board. The radial slider (CapSense sensors 6 to 13) is symmetric and the sensor order can be shifted to fit your requirement, that is, the desired zero position on the slider.

[b] CMOD0, CTANK0, CMOD1, and CTANK1 are only present in the CY8CKIT-046 PSoC 4 L-Series Pioneer Kit.

Proximity Header Pin Map						
Pin #	Description	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	PROXIMITY	–	P2[0]	P2[0]	P3[7]	P9[4]
2		–	–	–	P3[6]	P9[5]

RGB LED Pin Map						
Pin #	Color	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	Red	P1[6]	P3[2]	P2[6]	P0[6]	P5[2]
2	Green	P0[2]	P1[1]	P3[6]	P2[6]	P5[3]
3	Blue	P0[3]	P0[2]	P3[7]	P6[5]	P5[4]

User Switch Pin Map						
Pin #	Description	Pioneer Series Kits				
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046
1	SW2	P0[7]	–	P2[7]	P0[7]	P0[7]

Revision History

CY8CKIT-042-BLE-A Bluetooth® Low Energy (BLE) Pioneer Kit Guide Revision History

Document Title: CY8CKIT-042-BLE-A Bluetooth® Low Energy (BLE) Pioneer Kit Guide Document Number: 002-11468			
Revision	**Issue Date**	**Origin of Change**	**Description of Change**
**	03/04/2016	UDYG	New kit guide.