

# CY8CKIT-042-BLE

# Bluetooth<sup>®</sup> Low Energy (BLE) Pioneer Kit Guide

Doc. # 001-93731 Rev. \*\*

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



#### Copyrights

© Cypress Semiconductor Corporation, 2014. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

CySmart, F-RAM, PRoC, Programmable System-on-Chip, and PSoC Creator are trademarks and PSoC and CapSense are registered trademarks of Cypress Semiconductor Corporation. All other products and company names mentioned in this document may be the trademarks of their respective holders.

Purchase of  $I^2C$  components from Cypress or one of its sublicensed Associated Companies conveys a license under the Philips  $I^2C$  Patent Rights to use these components in an  $I^2C$  system, provided that the system conforms to the  $I^2C$  Standard Specification as defined by Philips. As from October 1st, 2006 Philips Semiconductors has a new trade name - NXP Semiconductors.

#### **Flash Code Protection**

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

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

# Contents



Saf	fety Information	7
1. Intr	roduction	9
	1.1 Kit Contents	9
	1.2 Board Details	
	1.3 PSoC Creator™	
	1.4 Getting Started	
	1.5 Additional Learning Resources	
	1.5.1 Beginner Resources	14
	1.5.2 Application Notes	
	1.5.3 PSoC Creator Example Projects	15
	1.5.4 Component Datasheets	15
	1.5.5 Bluetooth Learning Resources	15
	1.5.6 Learning From Peers	15
	1.5.7 Other Related Resources	16
	1.6 Technical Support	16
	1.7 Documentation Conventions	16
	1.8 Acronyms	17
2. Sof	ftware Installation	19
	2.1 Before You Begin	19
	2.2 Install Software	19
	2.3 Uninstall Software	22
3. Kit	Operation	23
	3.1 Theory of Operation	
	3.2 BLE Pioneer Kit USB Connection	25
	3.3 Placing Modules on Baseboard	25
	3.4 Programming and Debugging BLE Device	26
	3.5 Donale Connection	29
	3.6 USB-UART Bridge	31
	3.7 USB-I2C Bridge	32
	3.8 Updating the Opboard Programmer Firmware	34
	3.9 Measure Coin-cell Power Consumption	34
4. Exa	ample Projects	37
	4.1 Using Example Projects	37
	4.2 Kit Test	42
	4.2.1 Project Description	42
	4.2.2 Hardware Connections	43
	4.2.3 Flow Chart	44
	4.2.4 Verify Output	45



4.3	CapSense Slider and LED	
	4.3.1 Project Description	46
	4.3.2 Hardware Connections	
	4.3.3 Flow Chart	
	4.3.4 Verify Output	50
4.4	CapSense Proximity	62
	4.4.1 Project Description	62
	4.4.2 Hardware Connections	63
	4.4.3 Flow Chart	65
	4.4.4 Verify Output	66
4.5	BLE Central Mode	72
	4.5.1 Project Description	72
	4.5.2 Hardware Connections	74
	4.5.3 Flow Chart	75
	4.5.4 Verify Output	76
4.6	BLE Dongle and LED Control	79
	4.6.1 Project Description	79
	4.6.2 Hardware Connections	79
	4.6.3 Flow Chart	80
	4.6.4 Verify Output	81
	/ 1	

## 5. Hardware

#### 83

- 4	<b>D</b> '	- Developed	00
5.1	Pione		83
	5.1.1	PSoC 5LP	
	5.1.2	Power System	83
	5.1.3	Programming Interface	89
	5.1.4	Expansion Connectors	90
	5.1.5	USB Mini-B Connector	94
	5.1.6	CapSense Circuit	95
	5.1.7	Pioneer Board LEDs	97
	5.1.8	Push Buttons	
	5.1.9	Cypress Ferroelectric RAM (F-RAM)	99
	5.1.10	Serial Interconnection Between PSoC 5LP and Bluetooth Module	
	5.1.11	Bluetooth Module Headers	
5.2	BLE M	lodule Board	
	5.2.1	PSoC 4 BLE or PRoC BLE	
	5.2.2	Bluetooth Module Headers (20-Pin and 24-Pin Headers)	
	5.2.3	Wiggle Antenna	
	5.2.4	Antenna Matching Network	
	5.2.5	BLE Passives	
	5.2.6	Test Points	
5.3	BLEC	ongle Board	
0.0	531	Power System	108
	532	PRoC BI F	109
	533	Wiggle Antenna	109
	534	Antenna Matching Network	109
	535		110
	536	System Status   ED and Power   ED	
	527		
	5.5.7	Duch Puttona	
	0.3.8		



195

6.	Advanc	ed Topics	113
	6.1	Using PSoC 5LP as USB-UART Bridge	113
	6.2	Using PSoC 5LP as USB-I2C Bridge	124
	6.3	Developing Applications for PSoC 5LP	132
		6.3.1 Building a Bootloadable Project for PSoC 5LP	132
		6.3.2 Building a Normal Project for PSoC 5LP	141
	6.4	PSoC 5LP Factory Program Restore Instructions	142
		6.4.1 PSoC 5LP is Programmed with a Bootloadable Application	142
	6.5	Using FM24V10 F-RAM	148
		6.5.1 Address Selection	149
		6.5.2 Write/Read Operation	149
	6.6	CySmart iOS Application	150
	6.7	CySmart PC Tool	158
Α.	Append	ix	167
	A.1	Schematics	167
	A.2	Board Layout	175

## **Revision History**

A.3

CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit Guide, Doc. # 001-93731 Rev. \*\*

Contents



# Safety Information



The CY8CKIT-042-BLE Bluetooth Low Energy (BLE) Pioneer Kit is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. For this reason, the board may cause interference with other electrical or electronic devices in close proximity. In a domestic environment, this product may cause radio interference. In such cases, the user may be required to take adequate preventive measures. Also, this board should not be used near any medical equipment or RF devices.

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

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



The CY8CKIT-042-BLE boards contain electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, which can cause a discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-042-BLE boards in the protective shipping package.

#### End-of-Life/Product Recycling





# **General Safety Instructions**

#### **ESD** Protection

ESD can damage boards and associated components. Cypress recommends that the user perform procedures only at an ESD workstation. If an ESD workstation is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to a grounded metal object.

### Handling Boards

CY8CKIT-042-BLE boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad if available. Do not slide the board over any surface.

### Battery Care and Use

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

### **Battery Disposal**

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

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



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

## 1.1 Kit Contents

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

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



#### Figure 1-1. Kit Contents



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

## 1.2 Board Details

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

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



- 17. System power supply jumper (J16)
- 18. Arduino compatible I/O header (J3)
- 19. Arduino compatible I/O header (J4)

20.BLE power supply jumper (J15)

21.BLE module headers (J10/J11)

Figure 1-2. BLE Pioneer Baseboard Markup





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







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

#### Figure 1-4. BLE Dongle Markup



## **1.3 PSoC Creator™**

PSoC Creator is a state-of-the-art, easy-to-use integrated design environment (IDE). It introduces revolutionary hardware and software co-design, powered by a library of preverified and precharacterized PSoC Components<sup>™</sup>.

With PSoC Creator, you can:

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

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

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



# 1.4 Getting Started

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

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

## 1.5 Additional Learning Resources

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

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

#### 1.5.1 Beginner Resources

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

#### 1.5.2 Application Notes

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



## 1.5.3 PSoC Creator Example Projects

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

	Find Example Proje	ct				2	×
	Filter Options						
	Device Family:	PSoC 4200-BL	•	Documentation	Sample Code 🛔		Q P
	Keyword:	BLE	•	6			
	Project Name:			CYPRESS	PSoce	Creator™ Component Datasheet Example	
	BLE Proximity				BLE Heart Rate Se	nsor Example Project	
Design01 - PSoC Creator 3.1 [C:\\Des <u>Eile</u> <u>Edit View Project Build De</u>	BLE_HID_Mouse BLE_HID_Keyboar BLE_Heart_Rate_ BLE_Heart_Rate_ BLE_FindMe	rd Sensor Lollector	-1	Features BLE Heart Rate Se Simulating the Hea	rvice support in the server GATT role rt Rate data		
New	BLE_Device_Infon BLE_Battery_Leve	mation_Service		LED status indication	on status through DART		
Example Project				General Descr This example project d simulates Heart Rate d	iption emonstrates the BLE Heart Rate Sens lata and performs communication with	or workflow. The project BLE enabled central/client	
Glose Ctrl+F4				Development	Kit Configuration	prior BLE Chip ES10 with	
				Connect the MiniProg3	I, Atlantis board and RS-232 cord to the	PSVP.	
				Project Config	uration		
				The top design schem:	atic is shown in Figure 1.		
	2		*				
			Add to	Existing Workspace	Create New Wor	kspace Cancel	

Figure 1-5. PSoC Creator Example Projects

#### 1.5.4 Component Datasheets

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

Figure 1-6. Opening Component Datasheet



#### 1.5.5 Bluetooth Learning Resources

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

- Training videos
- GATT profiles
- Bluetooth community forum

#### 1.5.6 Learning From Peers

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



### 1.5.7 Other Related Resources

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

## 1.6 Technical Support

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

# **1.7** Documentation Conventions

|--|

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 <i>sourcefile.hex</i> file in the <i>PSoC Creator User Guide</i> .
[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 <b>File</b> icon and then click <b>Open</b> .
Times New Roman	Displays an equation: 2+2=4
Text in gray boxes	Describes cautions or unique functionality of the product.



# 1.8 Acronyms

Table 1-2. Acronyms Used in this Document

Acronym	Definition
ADC	Analog-to-Digital Converter
API	Application Programming Interface
BLE	Bluetooth Low Energy
CD	Compact Disc
CDC	Communications Device Class
СОМ	Communication Port
DVD	Digital Video Disc
ESD	Electrostatic Discharge
F-RAM	Ferroelectric Random Access Memory
GUI	Graphical User Interface
GPIO	General Purpose Input/Output
12C	Inter-Integrated Circuit
IAS	Immediate Alert Service
IDAC	Interconnecting Digital-Analog Converter
IDE	Integrated Development Environment
ISO	International Organization for Standardization
LDO	Low Drop Out (voltage regulator)
LED	Light-Emitting Diode
LP	Low Power
LPT	Line Print Terminal
PrISM	Precise Illumination Signal Modulation
PRoC	Programmable Radio-on-Chip
PRM	Protocol Service Multiplexer
PSoC	Programmable Systems-on-Chip
PWM	Pulse-Width Modulation
QFN	Quad Flat No-lead (package)
RGB	Red Green Blue
SAR	Successive Approximation Register
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB CDC	Universal Serial Bus Communications Device Class

Introduction





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

## 2.1 Before You Begin

2.

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

## 2.2 Install Software

Follow these steps to install the BLE Pioneer Kit software:

Software Installation

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

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



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

#### Figure 2-1. Kit Installer Screen



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



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

Figure 2-2. Product Installation Overview

🐳 CyInstaller for CY8CKIT-042-BLE Kit	? 🔀
Product Installation Overview Choose the install type that best suits your needs	
Choose the type of installation Product: CY8CKIT-042-BLE Kit Installation Type: Complete Typical Custom All progra Complete	
Contact Us	Next > Cancel

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

<Install\_Directory>\CY8CKIT-042-BLE Kit

Default location:

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

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

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

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



## 2.3 Uninstall Software

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

- Go to Start > All Programs > Cypress > Cypress Update Manager > Cypress Update Manager; select the Uninstall button.
- Go to Start > Control Panel > Programs and Features for Windows 7 or Add/Remove Programs for Windows XP; select the Uninstall/Change button.



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

## 3.1 Theory of Operation

Kit Operation

3.

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



Figure 3-1. BLE Pioneer Baseboard Block Diagram

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

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

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



Figure 3-2. BLE Module Block Diagram



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

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

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

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



Figure 3-3. BLE Dongle Block Diagram



# 3.2 BLE Pioneer Kit USB Connection

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

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

Table 3-1. BLE Pioneer Kit Enumerated Interfaces

Port	Description
USB Composite Device	Composite device
KitProg	Programmer and debugger
KitProg USB-UART	USB-UART bridge, which will appear as a COM# port
USB Input Device	USB-I <sup>2</sup> C bridge

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

J Driver Software Installation		<b>×</b>
Your device is ready to use		
USB Composite Device USB Input Device KitProg (1.2.3.3) KitProg USB-UART (COM97)	Ready to use Ready to use Ready to use Ready to use	
		Close

## 3.3 Placing Modules on Baseboard

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

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





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

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



## 3.4 Programming and Debugging BLE Device

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

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



Figure 3-7. Connect USB Cable to J13



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

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



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

Figure 3-9. Build an Example Project

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	<u>B</u> uild	<u>D</u> ebug <u>T</u> ools <u>W</u> indow <u>H</u> elp
; 🗊 🎦 着 💕 🗐 🖉 🍊	E E	B <u>u</u> ild PSoC_4_BLE_Kit_Test Shift+F6
Microsoft Sans Serif	(	Clean PSoC_4_BLE_Kit_Test
Workspace Explorer	¥ (	Clean and Build PSoC_4_BLE_Kit_Test
🖫 🔁	<u></u>	Cancel Build Ctrl+Break
Workspace 'PSoC_4_BLE_Kit	۵ (	Compile File Ctrl+F6
Project 'PSoC_4_BLE_K	1	Generate Application
PSoC_4_BLE_Kit_Test		Generate Project Datasheet
⊡-1 ⊡-1 ⊡-1 ⊡-1 Source Files □-1 © main.c		Components



5. If there are no errors during build, program the firmware into the kit by choosing Debug > Program or pressing [Ctrl] [F5], as shown in Figure 3-10. This will program the device on the BLE Pioneer Kit and the kit will be ready for use. For example, if the PSoC\_4\_BLE\_Kit\_Test project is programmed successfully to the PSoC 4 BLE Module on the BLE Pioneer kit, then you will observe the blue LED toggling every second. If debugging is needed on the project, go to step 6.

<u>File Edit View Project Build</u>	Deb	bug <u>T</u> ools <u>W</u> indow <u>H</u> elp
👔 🖞 👌 💕 🖬 🖉 🖪 🐧 🕺		<u>W</u> indows
Microsoft Sans Serif 🔹 10	0010	Program Ctrl+F5
Workspace Explorer	淰	Select Debug <u>T</u> arget
· · · · · · · · · · · · · · · · · · ·	惫	Debug F5
Workspace 'PSoC_4_BLE_Kit_Test' (1	羝	Debug without Programming Alt+F5
Project 'PSoC_4_BLE_Kit_Test'		Attach to Running Target
PSoC_4_BLE_Kit_Test.cydwr	ø	Toggle Breakpoint F9
🖨 🧀 Header Files		New Breakpoint
👘 🛅 main.h	0	Delete All Breakpoints Ctrl+Shift+F9
main.c	0	Enable All Breakpoints
Cenerated_Source	_	

Figure 3-10. Programming Device From PSoC Creator

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



# 3.5 Dongle Connection

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



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



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

Figure 3-12.	Dongle Interface	on CvSmart	PC Tool
1 iguio o 12.	Dongio intonaco	on oyoman	1 0 100

S CySmart 1.0				
File Help				
Select Dongle				
	Select BLE Dongle Target	Details	2	
	Cypress BLE Dongle (COM45)	Product: -		
		Firmware version: -		
		Hardware version: -		
		Description:		
	Show only supported devices			
Log	Refresh		Disconnect Close	
Clear Log 🔛 Save Log				
				*
				· ·

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

- 1. Connect the Dongle to the USB port on the PC.
- 2. Open PSoC Programmer 3.21.1 by going to Start > All Programs > Cypress > PSoC Programmer 3.21.1 > PSoC Programmer 3.21.1.
- 3. When PSoC Programmer opens, it will automatically detect the KitProg on the Dongle. Click the **File Load** button and browse to the location of the *BLE\_Dongle\_CySmart.hex* file. The hex file is located at:

```
C:\Program Files (x86)\Cypress\
CY8CKIT-042-BLE Kit\<version>\Firmware\BLE Dongle\Hex Files\
```

Figure	3-13	Open	Hex	File
riguie	5-15.	Open	1167	1 110

PSoC Programmer				
File View Options He	elp			
🖆 · 🔪 💿 BB		3		
Port Selection	Programmer Utilities	JTAG		
KitProg/1D200B2A011A340	Programming Param	eters		
	File Path:	C:\Program Files (x86)\Cypre	ess\CY8CKIT-042-	BLE Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_Dongle_CySmart.hex
	Programmer: Programming Mode: Verification:	KitProg/1D200B2A011A340 Reset Power Cycle On Off	0 Power Detect	@ 5p 💭 10p
Device Family	AutoDetection:	🖲 On 🔘 Off	Clock Speed:	1.6 MHz 🔻
CY8C4xx-BLE *	Programmer Charact	eristics	Status Execution Time	2°
Device CYBL10162-56LQXI *	Voltage: 0 5.0 V	● SWD ● ISSP ● I2C ) 3.3 V ● 2.5 V ● 1.8 V	Power Status: Voltage:	ON 5049 mV
Actions	Results			



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

|--|

PSoC Programmer		
File View Options Help		
🖆 • 💽 💿 🗗		
Port Selection	rogrammer Utilities JTAG	
KitProg/1D200B2A011A340	Programming Parameters	
	File Path: C:\Program Files (x86)\Cypre	ss\CY8CKIT-042-BLE Kit\1.0\Firmware\BLE Dongle\Hex Files\BLE_Dongle_CySmart.hex
	Programmer:         KitProg/1D200B2A011A3400           Programming Mode:         Image: Comparison of the section of th	Power Detect <u>Connector:</u>
Device Family	AutoDetection:      On      Off	Clock Speed: 1.6 MHz *
CY8C4xx-BLE *	Programmer Characteristics Protocol:	Status Execution Time: 9.2 seconds
Device		Power Status: ON
CYBL10162-56LQXI -	0.00,000,000,000,000,000	Voltage: 5049 mV
Actions	Results	

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

## 3.6 USB-UART Bridge

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

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

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

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

|--|



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

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 sup- ported	Xmodem, 1K Xmodem, Ymodem, Kermit, and Zmodem (only speeds greater than 2400 baud)

Table 3-3	Specifications	Supported by	USB-UART Bridge
	opecifications	Supported by	

# 3.7 USB-I<sup>2</sup>C Bridge

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

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

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

Figure 3-15. Bridge Control Panel





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

Figure 3-16. KitProg USB-I<sup>2</sup>C Connected in Bridge Control Panel

🗱 Bridge Control Panel 📃	
<u>File Editor Chart Execute Iools Help</u>	
☞ ■ 資   過 № 池   ◇ 三   歴 医 歴	
Editor Chart Table File	
	-
	4
Select Port in the PortList, then try to connect Opening Port	^
Successfully Connected to KitProg/0E1E0B2A011A3400	
KitProg Version 2.08	
	-
	•
Connected I2C/SPI/RX8 Pots: RCProg/0E1E082A011A3400 Power Protocol Protocol	
(2) Reset         (2:List         (2:List	
Scan period, ms: 0	
1:1 Syntax: OK Connected Powered Voltage: 3375 mV	

USB-I<sup>2</sup>C is implemented using the USB and I<sup>2</sup>C components of PSoC 5LP. For the BLE Pioneer Kit, the SCL (P12\_0) and SDA (P12\_1) lines from the PSoC 5LP are connected to the SCL (P3\_5) and SDA (P3\_4) lines of the BLE Module header. For the Dongle, the SCL (P12\_0) and SDA (P12\_1) lines from the PSoC 5LP are connected to the SCL (P3\_5) and SDA (P3\_4) lines. The USB-I<sup>2</sup>C bridge currently supports I<sup>2</sup>C speed of 50 kHz, 100 kHz, 400 kHz, and 1 MHz.

See Using PSoC 5LP as USB-I2C Bridge on page 124 for building a project that uses the USB-I<sup>2</sup>C bridge functionality.



## 3.8 Updating the Onboard Programmer Firmware

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

<Install\_Directory>\CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\KitProg
\KitProg.cyacd

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

## 3.9 Measure Coin-cell Power Consumption

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

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

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

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





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

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



Kit Operation


# 4. Example Projects



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

The Cypress BLE device comes in two variants:

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

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

### 4.1 Using Example Projects

Follow these steps to open and use the example projects:

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



- 2. On the Start Page, choose **Examples and Kits > Kits > CY8CKIT-042-BLE**. A list of example projects appears, as shown in Figure 4-1. Projects named with the prefix '*PSoC\_4\_BLE\_*' work on the BLE Pioneer kit with the PSoC 4 BLE Module; projects named with the prefix '*PRoC\_BLE\_*' work on the BLE Pioneer kit with the PRoC BLE Module.
- 3. Click on the desired example project.

Figure 4-1. Open Example Project from PSoC Creator

Workspace Explorer 🛛 👻 👎	×	Start Page
<b>1</b>		
No workspace open	Sourc	PSoC <sup>®</sup> Creator™
	Con	Create New Project Open Existing Project
	por	Getting Started
	ents Datasheets Results	PSoC Creator Start Page Quick Start Guide Intro to PSoC Intro to PSoC Creator PSoC Creator Training Design Tutorials Getting Started With PSoC 3 Getting Started With PSoC 4 Getting Started With PSoC 5LP
		Examples and Kits
		Find Example Project         Kits ∅         CY8CKIT-040         CY8CKIT-042.3.0         CY8CKIT-042-BLE         BLE_Dongle_CySmart.cywrk         PRoC_BLE_CapSense_Proximity.cywrk         PRoC_BLE_CapSense_Slider_LED.cywrk         PRoC_BLE_Central_IAS.cywrk         PSoC_4_BLE_CapSense_Slider_LED.cy         PSoC_4_BLE_Central_IAS.cywrk         PSoC_4_BLE_Central_IAS.cywrk         PSoC_4_BLE_Kit_Test.cywrk
		Product Information

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



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

Configure 'BLE'		?
Configure 'BLE' Name: BLE General Gene	ttings       Built-in         Device address       Public address         Public address (Company ID - Company assigned):       00A050-112244         Silicon generated "Company assigned" part of device address         Vou 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	41
Datasheet	MTU size (bytes): 23 🚖	Cancel

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

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

Figure 4-3. Build Project from PSoC Creator





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

Figure 4-4. Program Device in PSoC Creator



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

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

Select Debug Target	? 💌
■-  KitProg/1C12022A011A3400	KitProg/1C12022A011A3400
	POWER = 3 VOLTAGE_ADC = 3418 FREQUENCY = 2000000 PROTOCOL = SWD
	KitProg Version 2.08
Show all targets	Port Setting Port Acquire
	ок

Figure 4-5. Port Acquire

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

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

Select Debug Target	-?
⊡-5 KitProg/1C12022A011A3400	PSoC 4A-BLE CY8C4247LQI-BL483
PSoC 4A-BLE CY8C4247LQI-BL483	PSoC 4A-BLE (ARM CM0) Silicon ID: 0x0BB11477 Cypress ID: 0x0E34119E Revision: PRODUCTION
	Target unacquired
Show all targets	<u>C</u> onnect OK

**Note:** As stated previously, the BLE Pioneer Kit supports both Cypress BLE devices: PSoC 4 BLE and PRoC BLE. Thus, there are two versions of each of the kit example projects demonstrating the same functionality. Projects named with the prefix 'PSoC\_4\_BLE\_' work with the PSoC 4 BLE Module placed on the Baseboard. Projects named with the prefix 'PRoC\_BLE\_' work with the PRoC BLE Module placed on the Baseboard. Ensure that the correct module is placed on the Baseboard before programming the device with the corresponding kit example projects.

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

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



# 4.2 Kit Test

#### 4.2.1 Project Description

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

Two projects demonstrate this functionality on two different devices:

- **PSoC\_4\_BLE\_Kit\_Test** works with the PSoC 4 BLE Module.
- **PRoC\_BLE\_Kit\_Test** works with PRoC BLE Module.

Figure 4-7. TopDesign for PSoC\_4\_BLE\_Kit\_Test







Figure 4-8. Top Design for PRoC\_BLE\_Kit\_Test

#### 4.2.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the board. On the Workspace Explorer, double-click **PSoC\_4\_BLE\_Kit\_Test.cydwr**/ **PRoC\_BLE\_Kit\_Test.cydwr** on the left and select the pins shown in Figure 4-9.

Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC\_4\_BLE\_Kit\_Test works with the PSoC 4 BLE Module (red BLE Module) and PRoC\_BLE\_Kit\_Test works with the PRoC BLE Module (black BLE Module).

Table 4-1.	Pin	Assignment for	Test Pro	ject
------------	-----	----------------	----------	------

Pin Name	Port Name
Blue_LED	P3_7

Figure 4-9. Pin Selection for Test Project

Alias	Name 🗠	Port	Ι	Pir	n
	Blue_LED	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart_cts, SRSS:ext_clk_lf	•	54	•



# 4.2.3 Flow Chart

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

Figure 4-10. Test Project Flow Chart





# 4.2.4 Verify Output

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

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



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





# 4.3 CapSense Slider and LED

#### 4.3.1 Project Description

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

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

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

Two projects demonstrate this functionality on two different devices:

- PSoC\_4\_BLE\_CapSense\_Slider\_LED works with the PSoC 4 BLE Module
- **PRoC\_BLE\_CapSense\_Slider\_LED** works with the PRoC BLE Module.

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

Figure 4-13. TopDesign for PSoC\_4\_BLE\_CapSense\_Slider\_LED Project







Figure 4-14. TopDesign for PRoC\_BLE\_CapSense\_Slider\_LED Project



#### 4.3.2 Hardware Connections

No specific hardware connections are required for this project because all connections are hardwired on the board. Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC\_4\_BLE\_CapSense\_Slider\_LED works with the PSoC 4 BLE Module. PRoC\_BLE\_CapSense\_Slider\_LED works with the PRoC BLE Module.

On the Workspace Explorer, double-click on **PSoC\_4\_BLE\_CapSense\_Slider\_LED.cydwr**/ **PRoC\_BLE\_CapSense\_Slider\_LED.cydwr** on the left, and select the pins shown in Figure 4-15.

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

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

I IQUIE 4-13. FIII SEIECUUITIUI CAPSEIISE SIIUEI AIIU LED FIUEU	Figure 4-15.	Pin Selection	for CapSense	Slider a	nd LED	Project
---	--------------	---------------	--------------	----------	--------	---------

Alias	Name 🗠	Port		Pi	'n	Lock
Cmod	\CapSense:Cmod\	P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi	•	5	•	<b>V</b>
LinearSlider0_e0_LS	\CapSense:Sns[0]\	P2[1] OA0:vminus, SCB0:spi_select[2]	•	38	•	<b>V</b>
LinearSlider0_e1LS	\CapSense:Sns[1]\	P2[2] OA0:vout_10x, SRSS:wakeup, SCB0:spi select[3]	•	39	•	<b>V</b>
LinearSlider0_e2_LS	\CapSense:Sns[2]\	P2[3] OA1:vout_10x, SRSS:ext_clk_lf	•	40	•	<b>V</b>
LinearSlider0_e3_LS	\CapSense:Sns[3]\	P2[4] OA1:vminus	•	41	•	<b>V</b>
LinearSlider0_e4_LS	\CapSense:Sns[4]\	P2[5] OA1:vplus	•	42	•	<b>V</b>
	BLUE	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf	•	54	•	<b>V</b>
	GREEN	P3[6] SARMUX:pads[6], TCPWM3:line_out, SCB1:uart rts	•	53	•	<b>V</b>
	RED	P2[6] OA0:vplus_alt	•	43	•	<b>V</b>
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	•	44	•	<b>V</b>



# 4.3.3 Flow Chart

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

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





#### 4.3.4 Verify Output

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

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

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

#### 4.3.4.1 CySmart PC Tool

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

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

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



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

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

Ei	le <u>H</u> elp					
1	Select Dongle 🥒 Configu	ure Master Settings	Manage PSN	ls 🖞 Die	sconnect	
la	ster					
Dis	covered devices					
ä	<u>S</u> tart Scan 👹 C <u>o</u> nnect	🛃 Add to Whitelist				
ŧ	Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected
2			the second s	Control Control France	and the second state of the second state	

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

Figure 4-18. Discover All Attributes

Eile	<u>H</u> elp				
🚯 Sele	ct Dongle	<sup>9</sup> Configu	re Master Settings	Manage PSMs	Disconnect
Master	CapSense F	roximity [23	:43:65:50:A0:00]		
Attribute	s				
	cover All Attr	ibutes	Enable All Notific	ations 🛄 Read	All Characteristic
- Disc					



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

S CySmart 1.0											
<u>F</u> ile <u>H</u> elp											
🚯 Select Dongle 🥒 Cor	nfigure Maste	r Settings 🛛 Manage PSMs 👌 <u>D</u> isconnee	t								
Master CapSense Slider a	nd LED [21:43	:65:50:A0:00]									
Attributes						1	Attribute Det	ails Send Com	mands		
Oiscover All Attributes	🛛 🛃 Enable	All Notifications  i Read All Characteri	stics , Pair 🛛 🔛 Exp	ort 🛛 💼	View: Category 👻 🗜 💶		Handle:	0×000	E		
Handle	UUID	UUID Description	Value	Propertie	-	•	UUID:	0x290	2		
	0x2803	Characteristic Declaration	02:07:00:04:2A			L,	UUID Desc	cription: Client C	Characteristic Confi	guration	
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02			Value:				
Primary Service Declarat	tion: Generic At	tribute				ĻĻ	01:00				^
⊡- 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)			μl					~
- Characteristic De	eclaration: Serv	rice Changed							[	Read Value 🔻 Write	Value 🔻
Ė− 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A			11	Presenting		Eashlad		<u> </u>
0x000A	0x2A05	Service Changed		0x22			Properties		Enabled		
Primary Service Declarat	tion						Broadcast				
⊡- 0x000B	0x2800	Primary Service Declaration	B5:CA				Read				
- Characteristic De	eclaration						write witho	out response			
Ė- 0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA				White				
0x000D	0xCAA2			0x10			Notify				
0x000E	0x2902	Client Characteristic Configuration			F		Indicate				
0x000F	0x2901	Characteristic User Description					Authentical	ted signed writes			
Primary Service Declarat	tion					11	Extended p	properties			<b>T</b>
	0x2800	Primary Service Declaration	BB:CB								
- Characteristic De	eclaration										
	0x2803	Characteristic Declaration	0A:12:00:B1:CB								
0x0012	0xCBB1			0x0A							
0x0013	0x2901	Characteristic User Description				-					
Attributes L2CAP Channe	els										

Figure 4-19. Write to CapSense Slider Characteristic Descriptor

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



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

Figure 4-20. CapSense Slider notification received

E	Primary Service Declarati	ion							
	⊡ 0x000B	0x2800	Primary Service Declaration	B5:CA					
	⊡ 0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA					
	<b>0</b> ×000D	0xCAA2		49	0×10				
	<b>0</b> ×000E	0x2902	Client Characteristic Configuration						
	0x000F	0x2901	Characteristic User Description						

Figure 4-21. CapSense Slider



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

Figure 4-22. Disable Notifications

🚭 CySmart 1.0							
<u>File</u> <u>H</u> elp							
🛞 Select Dongle 🍠 Con	ifigure Maste	er Settings Manage PSMs 👹 Disconner	ct				
Master CapSense Slider an	nd LED [21:43	3:65:50:A0:00]					
Attributes					A	ttribute Details Send Comn	ands
Oiscover All Attributes	🔹 Enable	All Notifications 🔟 Read All Characteri	istics \land Pair 🛛 🔛 Exp	ort   View: Category 👻 🛃 📻	8	Handle: 0x000E	
Handle	UUID	UUID Description	Value	Properties		UUID: 0x2902	!
⊡- 0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A			UUID Description: Client C	haracteristic Configuration
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02		Value:	
Primary Service Declarati	ion: Generic A	Itribute				00:00	
⊡ 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)		l		Ψ
- Characteristic De	claration: Serv	vice Changed					Read Value 👻 Write Value 💌
⊟- 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A		ll r	Descention	Factoria A
0x000A	0x2A05	Service Changed		0x22	H	Properties	Enabled
Primary Service Declarati	ion				H	Broadcast	
⊡- 0x000B	0x2800	Primary Service Declaration	B5:CA			Read	
- Characteristic De	claration				H	Write without response	=
⊟- 0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA			Write	
0x000D	0xCAA2		4D	0x10 =		Notify	
- 0x000E		Client Characteristic Configuration			H	Indicate	
0x000F	0x2901	Characteristic User Description				Authenticated signed writes	
E Primary Service Declarati	ion				1	Extended properties	•
⊡··0x0010	0x2800	Primary Service Declaration	BB:CB				
- Characteristic De	claration						
Ė-0x0011	0x2803	Characteristic Declaration	0A:12:00:B1:CB				
- 0x0012	0xCBB1			0x0A			
<	0.0001			· · · · · · · · · · · · · · · · · · ·			
Attributes   L2CAP Channel	ls						



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

En Primary Service Declarat	ion						
En 0x0010 0x2800 Primary Service Declaration BB:CB							
En Characteristic De	eclaration						
⊡ 0x0011	0x2803	Characteristic Declaration	0A:12:00:B1:CB				
0x0012	0xCBB1		00:00:00:00	0x0A			
0x0013	0x2901	Characteristic User Description	52:47:42:20:4C:45:44:				

Figure 4-23. Read RGB LED Control Characteristic Value

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

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

🕞 CySmart 1.0										
Eile Help										
🚯 Select Dongle 🥒 Conf	figure Maste	r Settings 🛛 Manage PSMs 😸 Disconne	t							
Master CapSense Slider an	d LED [21:43	:65:50:A0:00]								
Attributes						Attribute Details	Send Comm	ands		
Oiscover All Attributes	Enable	All Notifications 🔟 Read All Character	stics \land Pair 🛛 🔛 Exp	ort   View: Category 👻 💶	Ŧ	Handle:	0x0012			
Handle	UUID	UUID Description	Value	Properties	^	UUID:	0xCBB	1		
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02		UUID Descriptio	n:			
Primary Service Declaration	on: Generic At	tribute				Value.				
E- 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)			UT.UZ.EE.FF				
⊡ Characteristic De	claration: Serv	vice Changed								Ψ.
⊡ • <b>0</b> x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A						Read Value 💌 🛛 W	'rite Valuel 🔻
0x000A	0x2A05	Service Changed		0x22		Properties		Enabled		*
Primary Service Declaration	on					Properties		Lindblod		
⊡- 0x000B	0x2800	Primary Service Declaration	B5:CA			Pood				
- Characteristic De	claration					Write without m			•	
⊡ • 0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA			Write without re	sponse			
0x000D	0xCAA2		4D	0x10		Natific			<b>V</b>	
0x000E	0x2902	Client Characteristic Configuration			=	Indianta				
0x000F	0x2901	Characteristic User Description				Authoriticated a	ianad writes			
Primary Service Declaration	on					Extended prope	igned writes			
⊡- 0x0010	0x2800	Primary Service Declaration	BB:CB			Cxterided prope	nues			•
- Characteristic De	claration									
	0x2803	Characteristic Declaration	0A:12:00:B1:CB							
0x0012	0xCBB1		01:02:EE:FF	0×0A						
0x0013	0x2901	Characteristic User Description			-					
Attributes L2CAP Channels	s	III		F	L					

Figure 4-24. Write RGB LED Control Characteristic Value





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







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

Figure 4-27. Disconnect from the Device

aster CapSense Proximit	y [23:43:65:50:A0:00]				
Start Scan 😫 Discon	nect 🛃 Add to White	list			
Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected
CapSense Proximity	23:43:65:50:A0:00	Public	-41 dBm	Connectable undirected	

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

#### 4.3.4.2 CySmart iOS App

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

1. To verify the PSoC\_4\_BLE\_CapSense\_Slider\_LED project, plug in the PSoC 4 BLE Module on the Baseboard.

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





To verify the PRoC\_BLE\_CapSense\_Sider\_LED project, plug in the PRoC BLE Module on the Baseboard.





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



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

Figure 4-30. CapSense Slider and LED Peripheral

No Servi	ice 4:43 pm	* 🖚
	Home	≡
	CapSense Slider and LED	RSSI
<b>\$</b> 1	No Services Advertised	-59 dB

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

Figure 4-31. CapSense Service Page





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

Figure 4-32. CapSense Slider



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



Figure 4-33. RGB LED Service Page



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

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

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







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

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



# 4.4 CapSense Proximity

#### 4.4.1 Project Description

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

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

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

Two projects demonstrate this functionality on two different devices:

- **PSoC\_4\_BLE\_CapSense\_Proximity** works with the PSoC 4 BLE Module.
- **PRoC\_BLE\_CapSense\_Proximity** works with the PRoC BLE Module.

Figure 4-36. Top Design for PSoC\_4\_BLE\_CapSense\_Proximity Project







Figure 4-37. Top Design for PRoC\_BLE\_CapSense\_Proximity Project

#### 4.4.2 Hardware Connections

- Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC\_4\_BLE\_CapSense\_Proximity works with the PSoC 4 BLE Module. PRoC\_BLE\_CapSense\_Proximity works with the PRoC BLE Module.
- Connect a five-inch wire (provided as part of this kit) to the proximity connector J14 on the baseboard. Loop the wire as shown in Figure 4-38.

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

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







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

■ On the Workspace Explorer, double-click **PSoC\_4\_BLE\_CapSense\_Proximity.cydwr/ PRoC\_BLE\_CapSense\_Proximity.cydwr** on the left, and select the pins shown in Figure 4-40.

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

Table 4-3. Pin Assignments for CapSense Proximity Project

Figure 4-40. Pin Selection for CapSense Proximity Project

Alias	Name 🗠	Port		Pi	in
Cmod	\CapSense:Cmod\	P4[0] CSD:c_mod, TCPWM0:line_out, SCB1:uart rts, SCB1:spi mosi	•	5	•
ProximitySensor0_0_PROX	\CapSense:Sns\	P2[0] OA0:vplus, SCB0:spi_select[1]	•	37	•
	Status_LED	<pre>P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf</pre>	•	54	•
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	•	44	-



## 4.4.3 Flow Chart

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

Figure 4-41. CapSense Proximity Project Flow Chart





#### 4.4.4 Verify Output

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

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

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

#### 4.4.4.1 CySmart PC Tool

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

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

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

Figure 4-42. Connect to CapSense Proximity peripheral





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

Figure 4-43. Discover All Attributes

<u>File</u> <u>H</u> el	2				
🚯 Select Do	ngle 🥒	Config	ure Master Settings	Manage PSMs	🛱 Disconnect
Master Cap	Sense Pro	oximity [2	3:43:65:50:A0:00]		
musici					
Attributes					
Attributes	All Attrik	outes	Enable All Notific	ations 🔟 Read	I All Characteristic

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

Figure 4-44. Write to CapSense Proximity Characteristic descriptor

CySmart 1.0					
File Help					
Select Dongle 1 Cor	nfigure Maste	er Settings Manage PSMs 📩 Disconned	t		
Master CapSense Proximit	y [23:43:65:50	:A0:00]			
Attributes					Attribute Details Send Commands
Oiscover All Attributes	: 🛛 🛃 Enable	All Notifications 🔟 Read All Characteri	stics \land Pair 🛛 🔛 Exp	ort 📋 Clear View: Category 👻 🛃 🎦	Handle: 0x000E
Handle	UUID	UUID Description	Value	Properties	UUID: 0x2902
	0x2803	Characteristic Declaration	02:03:00:00:2A		UUID Description: Client Characteristic Configuration
0x0003	0x2A00	Device Name		0x02	Value:
- Characteristic De	eclaration: App	earance	1		01:00
⊡ 0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A		· · · · · · · · · · · · · · · · · · ·
0x0005	0x2A01	Appearance		0x02	Read Value Value Value
- Characteristic De	eclaration: Peri	pheral Preferred Connection Parameters			
⊡ 0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A		Properties Enabled
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02	Broadcast
Primary Service Declarat	ion: Generic A	ttribute			Head
⊡ 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)		Withe without response
- Characteristic De	eclaration: Sen	vice Changed			Net .
Ė~ 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A		Indicate
0x000A	0x2A05	Service Changed		0x22	Authenticated signed writes
Primary Service Declarat	ion	1			Extended organization
<u>⊡</u> 0x000B	0x2800	Primary Service Declaration	B5:CA		
Characteristic De	eclaration	T			
⊡- 0x000C	0x2803	Characteristic Declaration	10:0D:00:A1:CA		
0x000D	UxCAA1			0x10	
0x000E	0x2902	Client Characteristic Configuration	00:00		
Auchustana LI 2CAP Chappe	0x2901	Characteristic User Description		•	r
Autobules L2CAI Chaine	//s				

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



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

Primary Service Declarat	ion							
<sup>≟</sup> <b>0x000B</b>	0x2800	Primary Service Declaration	B5:CA					
En Characteristic Declaration								
⊡ 0x000C	0x2803	Characteristic Declaration	10:0D:00:A1:CA					
0x000D	0xCAA1		12	0x10				
0x000E	0x2902	Client Characteristic Configuration	00:00					
0x000F	0x2901	Characteristic User Description						

Figure 4-45. CapSense Proximity notification received

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



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



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



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

Figure 4-48. Disconnect from the Device

File       Help         Select Dongle       Configure Master Settings       Manage PSMs       Disconnect         Master       Cap Sense Proximity [23:43:65:50:A0:00]         Discovered devices         Image Start Scan       Disconnect	
Select Dongle 🧳 Configure Master Settings Manage PSMs 👹 Disconnect Master CapSense Proximity [23:43:65:50:A0:00] Discovered devices           Discovered devices           Image Start Scan         Disconnect	
Master       Cap Sense Proximity [23:43:65:50:A0:00]         Discovered devices         Image: Start Scan       Disconnect         Image: Start Scan       Disconnect	
Discovered devices          Image: Start Scan in the second seco	
🔀 Start Scan 😸 Disconnect 🛃 Add to Whitelist	
# Device Bluetooth Address Address Type RSSI Advertisement Type Connected	
1 CapSense Proximity 23:43:65:50:A0:00 Public -41 dBm Connectable undirected	<b>~</b>

- 16.Press user button (SW2) to wake up from sleep and restart the advertisement for the next connection.
- 4.4.4.2 CySmart iOS App

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

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

1. To verify the PSoC\_4\_BLE\_CapSense\_Proximity project, plug in the PSoC 4 BLE Module on the Baseboard.

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





To verify the PRoC\_BLE\_CapSense\_Proximity project, plug in the PRoC BLE Module on the Baseboard.

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



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

Figure 4-51. Connect to CapSense Proximity Peripheral





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

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



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



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



# 4.5 BLE Central Mode

#### 4.5.1 Project Description

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

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

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

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

- **PSoC\_4\_BLE\_Central\_IAS** works with the PSoC 4 BLE Module.
- **PRoC\_BLE\_Central\_IAS** works with the PRoC BLE Module.

Additionally, the **BLE\_Dongle\_Peripheral\_IAS** project is to be programmed on the Dongle. This project is present in both PSoC\_4\_BLE\_Central\_IAS and PRoC\_BLE\_Central\_IAS workspace and can be used to program the Dongle separately.

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

Figure 4-54. PSoC\_4\_BLE\_Central\_IAS TopDesign

BLE	USER INTERFACE
BLE BLE Bluetooth BLE component is configured as Find Me Locator, which is a Central device with Client role.	→ The Status_LED 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 4-55. PRoC\_BLE\_Central\_IAS TopDesign







#### 4.5.2 Hardware Connections

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

Ensure that the correct BLE Module is placed on the Baseboard corresponding to the project being used. PSoC\_4\_BLE\_Central\_IAS works with the PSoC 4 BLE Module. PRoC\_BLE\_Central\_IAS works with the PRoC BLE Module. BLE\_Dongle\_Peripheral\_IAS is the common project for both workspaces and programs the Dongle with peripheral mode firmware.

On the Workspace Explorer, double-click **PSoC\_4\_BLE\_Central\_IAS.cydwr/ PRoC\_BLE\_Central\_IAS.cydwr** on the left, and select the pins shown in Figure 4-57.

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

Alias	Name 🗠	Port		Pi	n	Lock
	Status_LED	P3[7] SARMUX:pads[7], TCPWM3:line_out_compl, SCB1:uart cts, SRSS:ext clk lf	-	54	•	<b>V</b>
	User_Button	P2[7] OA1:vplus_alt, SRSS:ext_clk	-	44	-	

Similarly, double-click **BLE\_Dongle\_Peripheral\_IAS.cydwr** in Workspace Explorer and select the pins shown in Figure 4-58.

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

Alias	Name /	Port			Pin	
	Alert_LED	P3[3] SARMUX:pads[3], TCPWM1:line_out_compl, SCB0:uart cts	•	50	•	<b>V</b>



## 4.5.3 Flow Chart

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

Figure 4-59. IAS Client Mode Flow Chart





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

Figure 4-60. IAS Peripheral Mode Flow Chart



### 4.5.4 Verify Output

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

Figure 4-61. Connect Dongle to USB Port





2. Program the Dongle with the BLE\_Dongle\_Peripheral\_IAS project described in Using Example Projects on page 37.

**Note:** Do not update the public device address (inside the BLE component) for the **BLE\_Dongle\_Peripheral\_IAS** example project. This is because the central example project looks for a particular public address equal to the one set on the peripheral BLE component. Changing the **BLE\_Dongle\_Peripheral\_IAS** example project public address will lead to no connection with the central device on the BLE Pioneer kit.

- 3. Power the BLE Pioneer kit through USB connector J13.
- 4. Program the BLE Pioneer kit with either PSoC\_4\_BLE\_Central\_IAS or PRoC\_BLE\_Central\_IAS project, depending on the BLE Module placed on the BLE Pioneer kit.
- 5. Wait for the BLE connection between the dongle and the Pioneer baseboard. The connection success status is indicated on the Baseboard in the following three stages:
  - a. Fast blinking blue LED represents scanning mode. During this mode, the BLE Pioneer kit is scanning for peripheral devices.
  - b. Slow blinking blue LED represents discovery mode. During this mode, the BLE Pioneer kit has found the Dongle peripheral device and has started the connection procedure.
  - c. The blue LED remains on, representing the connected mode. This mode means that the peripheral device has been connected and the application can now send alert levels.
- 6. Press the user button (**SW2**) on the BLE Pioneer kit to send the next alert level to the Dongle. The alert level will keep rotating from No Alert to Mid Alert to High Alert.
- 7. Check if the LED behavior changes for each alert notification on the Dongle according to the following table:

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





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

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



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



# 4.6 BLE Dongle and LED Control

#### 4.6.1 Project Description

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

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

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

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

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



Figure 4-64. Top Design for BLE\_Dongle\_CySmart project

#### 4.6.2 Hardware Connections

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



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

Figure 4-65.	Pin Selection	for BLE	Donale	Project
1 19010 1 00.	1 111 0010001011		Dongio	1 10,000

Alias	Name A	Port			Pin	
	\UART:rx_wake\	P1[4] OA3:vminus, TCPWM2:line_out, SCB0:uart_rx, SCB0:i2c_sda, SCB0:spi mosi		32	•	
	\UART:tx\	P1[5] OA3:vplus, TCPWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi miso		33	•	<b>V</b>
	Led_Pin	P3[3] SARMUX:pads[3], TCPWM1:line_out_compl, SCB0:uart cts	•	50	•	<b>V</b>
	Pin_SuspendIndicator	P3[2] SARMUX:pads[2], TCPWM1:line_out, SCB0:uart rts	•	49	•	<b>V</b>
	Pin_UserButton	P2[6] OA0:vplus_alt	•	43	•	<b>V</b>

## 4.6.3 Flow Chart

Figure 4-66. Flow Chart for BLE\_Dongle\_CySmart Project





## 4.6.4 Verify Output

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

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

Example Projects



# 5. Hardware



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

## 5.1 Pioneer Baseboard

#### 5.1.1 PSoC 5LP

An onboard PSoC 5LP is used to program and debug the BLE silicon. The PSoC 5LP connects to the USB port of the PC through a USB mini-B connector and to the SWD interface of the BLE device. PSoC 5LP is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C58LPxx family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals.

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

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

#### 5.1.2 Power System

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

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



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





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

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



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

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





#### 5.1.2.1 Protection Circuits

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

I/O Header 3.3V 5٧ Vin Coin cell ·3V MOSFET based Protection Ckt ttery Holde Bluetooth LDO Module USB 5V PTC PSoC 5LP PSoC 5LP 10 Pir Prog. Header PSoC 4 BLE 10 Pin Prog. Header ESD Protection

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

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



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

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







#### 5.1.2.2 Current Measurement Jumper

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

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







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

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

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





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

Figure 5-7. Current Measurement when Powered Separately



### 5.1.3 Programming Interface

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

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



# 5.1.4 Expansion Connectors

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

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

Figure 5-8. Arduino Headers





The J1 header contains I/O pins for reset, I/O reference voltage (IOREF), and power supply line. The J2 header is an analog port that contains I/O pins for SAR ADC, comparator, and opamp. The J3 header is primarily a digital port that contains I/O pins for PWM, I<sup>2</sup>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 183 for the header part number.





#### **Additional Functionality of Header J2**

The J2 header is a  $6\times2$  header that supports Arduino shields. The Port 2 and Port 3 pins of PSoC 4 BLE/PRoC BLE are brought to this header. The Port 2 pins also connect to the onboard CapSense slider through 560-ohm resistors. When the CapSense feature is not used, remove these resistors to help ensure better performance with these pins.

#### **Functionality of Unpopulated Header J12**

The J12 header is a 2x3 header that supports Arduino shields. It is used on a small subset of shields and is unpopulated on the BLE Pioneer board.



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

This port supports Digilent Pmod peripheral modules (see Figure 5-10). Pmods are small I/O interfaces that connect with the embedded control boards through either 6- or 12-pin connectors. The BLE Pioneer Kit supports the 6-pin PMOD type 2 (SPI) interface. For Digilent PMOD cards, go to www.digilentinc.com.

This header is not populated on the BLE Pioneer board. You must populate this header before connecting the PMOD daughter cards. Refer to the "No Load Components" section of Bill of Materials (BOM) on page 183 for the header part number.

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







#### 5.1.4.3 PSoC 5LP GPIO Header (J8)

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

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







### 5.1.5 USB Mini-B Connector

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

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







## 5.1.6 CapSense Circuit

#### 5.1.6.1 CapSense Slider

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

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





## **Proximity Header**

The baseboard contains a header (J14) for CapSense proximity wire connection (see Figure 5-14). Figure 5-14. Schematics and Board Highlight of Proximity Header





#### 5.1.7 Pioneer Board LEDs

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

Figure 5-15. Schematics of Status and Power LED

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





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



## 5.1.8 Push Buttons

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

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





## 5.1.9 Cypress Ferroelectric RAM (F-RAM)

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

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







## 5.1.10 Serial Interconnection Between PSoC 5LP and Bluetooth Module

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





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







## 5.1.11 Bluetooth Module Headers

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

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





# 5.2 BLE Module Board

## 5.2.1 PSoC 4 BLE or PRoC BLE

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

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





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

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

Figure 5-22. Schematics and Board Highlight of Headers









# 5.2.3 Wiggle Antenna

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

Figure 5-23. Board Highlight of Wiggle Antenna







## 5.2.4 Antenna Matching Network

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

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

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









## 5.2.5 BLE Passives

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

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











### 5.2.6 Test Points

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

# 5.3 BLE Dongle Board

#### 5.3.1 Power System

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

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



#### 5.3.1.1 Protection Circuits

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


# 5.3.2 PRoC BLE

## See PSoC 4 BLE or PRoC BLE on page 103.

Figure 5-27. Board Highlight of PRoC BLE



## 5.3.3 Wiggle Antenna

See Wiggle Antenna on page 105.

Figure 5-28. Board Highlight of Wiggle Antenna



# 5.3.4 Antenna Matching Network

See Antenna Matching Network on page 106.

Figure 5-29. Board Highlight of Antenna Matching Network





## 5.3.5 USB Type A Plug

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

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







# 5.3.6 System Status LED and Power LED

See Pioneer Board LEDs on page 97.

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



## 5.3.7 User LED

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

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







# 5.3.8 Push Buttons

See Push Buttons on page 98.

Figure 5-33. Push Buttons





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

# 6.1 Using PSoC 5LP as USB-UART Bridge

6. Advanced Topics

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

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

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

New Project	8
Design Other	٩ ۵
<ul> <li>Default Templates</li> </ul>	<u>^</u>
PSoC 3 Design	Creates a PSoC 3, 8-bit 8051, design project.
PSoC 4000 Design	Creates a PSoC 4000, 32-bit ARM Cortex-M0, design project.  ≡
PSoC 4100/4200 Design	Creates a PSoC 4100/4200, 32-bit ARM Cortex-M0, design project.
PSoC 4100/4200-BL Design	Creates a PSoC 4100/4200-BL, 32-bit ARM Cortex-M0, design project.
PRoC BLE Design	Creates a PRoC BLE, 32-bit ARM Cortex-M0, design project.
PSoC 5LP Design	Creates a PSoC 5LP, 32-bit ARM Cortex-M3, design project.
PSoC 3 Starter Designs	
ADC_DMA_VDAC	Shows how to transfer data from an ADC to a DAC using DMA with no CPU intervention.
Pa DelSig_16Channel	Shows a 16-channel, 12-bit Delta Sigma ADC in PSoC 3 sequenced in hardware; samples are transferred from ADC to SRAM using DMA - without processor intervention.
DelSig_I2CM	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.
DelSig_I2CS	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.
Name: Design01	
Location: C:\Users\BLE\Project	
Device: CY8C4247LQI-BL483	▼
Advanced	
Workspace: Create New Worksp	ace 🔹
Workspace Name: Design01	
Sheet Template: Empty (11" x 8.5")	•
Application Type Normal	•
	OK Cancel

Figure 6-1. Create New Project in PSoC Creator



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





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

Figure 6-3. Open UART Configuration Window





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

Figure 6-4. UART Configuration Tab Window

Configure 'SCB_P4'	? <mark>×</mark>
Name: UART	
Configuration UART Basic UART Advanced Built-in	4 ۵
Unconfigured SCB	
© 12C	
© EZI2C	
SPI	
O UART	
Datasheet OK Apply	Cancel

Figure 6-5. UART Basic Tab Window

Configure '	SCB_P4'		? 💌
Name:	UART		
Conf	iguration	UART Basic UART Advanced Built-in	4 Þ
Mode:		Standard 👻	<u> </u>
Direction		TX + RX 🔹	
Baud rate	e (bps):	9600 <ul> <li>Actual baud rate (bps): 9592</li> </ul>	
Data bits:		8 bits 🔹	
Parity:		None	E
Stop bits:		1 bit 🔹	
Oversamp	oling:	12	
Clock	from termin	al	
Media	an filter		
Retry	on NACK		
Invert	ting RX		-
Datas	sheet	ОК Арріу	Cancel



Figure 6-6.	UART Advance	ed Tab Window
-------------	--------------	---------------

Configure 'SCB_P4'	? <b>×</b>
Name: UART	
Configuration LIADT Basic LIADT Advanced Built in	4 Þ
Buffers size Interrupt	
RX buffer size: B 🔍 💿 None	
TX buffer size: 8 🔿 Internal	
Byte mode	
Interrupt sources	
UART done RX FIFO not empty	
TX FIFO not full RX FIFO full	
TX FIFO empty RX FIFO overflow	
TX FIFO overflow RX FIFO underflow	
TX FIFO underflow RX frame error	
TX lost arbitration	
TX NACK RX FIFO level: 7 👻	
TX FIFO level:	
Multiprocessor mode	
Address (hex): 2 k	
Mask (hex): FF	
Accept matching address in RX FIFO	
Flow control	
RTS Polarity: Active Low  RTS FIFO level: 4	
CTS Polarity: Active Low	
Pictor Contry. Pictor Com	
Datasheet OK Apply C	ancel

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

Figure 6-7. Pin Selection

Alias	Name 🗠	Port		/ Port		Pin Lo		Lock
	\UART_1:rx\	P1[4] OA3:vminus, TCPWM2:line_out, SCB0:uart_rx, SCB0:i2c_sda, SCB0:spi mosi	Ŧ	32	•	<b>V</b>		
	\UART_1:tx\	<pre>P1[5] OA3:vplus, TCFWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi miso</pre>	Ŧ	33	•	<b>V</b>		



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

```
int main()
{
    uint8 ch;
    /* Start SCB UART TX+RX operation */
    UART_Start();
/* Transmit String through UART TX Line */
UART_UartPutString("CY8CKIT-042-BLE USB-UART");
    for(;;)
    {
        /* Get received character or zero if nothing has been received yet
* /
         ch = UART_UartGetChar();
         if(0u != ch)
         {
 /* Send the data through UART. This function is blocking and waits until
there is an entry into the TX FIFO. */
             UART UartPutChar(ch);
         }
    }
}
7. Build the project by clicking Build > Build {Project Name} or [Shift][F6]. After the project is built
  without errors and warnings, program (by choosing Debug > Program) the project to PSoC 4
  BLE/PRoC BLE through the PSoC 5LP USB programmer or MiniProg3.
```

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



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

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

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

🚔 Device Manager	X
Eile Action ⊻iew Help	
D State Computer	*
Disk drives	
Display adapters	
DVD/CD-ROM drives	
P High Human Interface Devices	
D IDE ATA/ATAPI controllers	
IEEE 1394 Bus host controllers	
Imaging devices	
Keyboards	
Mice and other pointing devices	
Monitors	
Vetwork adapters	
Lisco Systems VPN Adapter for 64-bit Windows	
Intel(R) 82579LM Gigabit Network Connection	
Microsoft Virtual WiEi Miniport Adapter	-
A De Other devices	-
- Inknown device	
A TO Ports (COM & LPT)	
Titel(R) Active Management Technology - SOL (COI	M3)
KitProg USB-UART (COM12)	
Processors	
Security Devices	
SM Driver	
Sound, video and game controllers	
Storage controllers	
System devices	
🖻 🖣 Universal Serial Bus controllers	-



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

Figure 6-9. Open New Connection

### **HyperTerminal**

Connection Description	? X
New Connection	
Enter a name and choose an icon for the connection:	
Name:	
USB-UART communication	
lcon:	
•	•
ОК	Cancel

## PuTTY

Real Putty Configuration		X
Category:		
Session	Basic options for your PuTTY se	ssion
Logging	Specify the destination you want to connect t	0
Keyboard	Host Name (or IP address)	Port
Bell		22
- Features ⊟-Window - Appearance	Connection type:	H 🔘 Serial
Behaviour Translation Selection Colours	Load, save or delete a stored session Saved Sessions	
Connection Data Proxy Telnet Rlogin SSH Serial	Default Settings	Load Saye Delete
	Close window on exit Always ONever Only on c	lean exit
About	Open	<u>C</u> ancel



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

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

Figure 6-10. Select Communication Port

### HyperTerminal

Connect To	? ×	
USB-UART communication		
Enter details for	the phone number that you want to dial:	
Country/region:	India (91) 👻	
Ar <u>e</u> a code:	080	
Phone number:		
Connect using:	COM12 -	
	OK Cancel	

## PuTTY





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

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

Figure 6-11. Configure the Communication Port HyperTerminal

COM12 Properties	2 ×
Port Settings	
Bits per second:	9600 👻
Data bits:	8
Parity:	None 👻
<u>S</u> top bits:	1
Elow control:	None •
	Restore Defaults
0	K Cancel Apply

PuTTY

Reputry Configuration		
Category: Session Logging Terminal - Keyboard - Bell - Features Window - Appearance - Behaviour - Translation - Selection - Colours - Connection - Data - Proxy - Telnet - Riogin - SSH - Senal	Options controlling Select a serial line Serial line to connect to Configure the serial line Speed (baud) Data bits Stop bits Parity Elow control	g local serial lines COM12 9600 8 1 None None
About		Open Cancel



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

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

ASCII Setup									
ASCII Sending									
Send line ends with line feeds									
Echo typed characters locally									
Line delay: 0 milliseconds.									
Character delay: 0 milliseconds.									
ASCII Receiving									
Append line feeds to incoming line ends									
Eorce incoming data to 7-bit ASCII									
✓ <u>W</u> rap lines that exceed terminal width									
OK Cancel									

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

lategory:	
Session	Options controlling the terminal emulation
Logging     Logging     Terminal     Keyboard     Bell     Features     Window     Appearance     Behaviour     Translation     Selection     Colours     Colours     Colours     Data     Proxy     Telnet     Rlogin     SSH     Serial	Options controlling the terminal emulation Set various terminal options Auto grap mode initially on DEC Origin Mode initially on DEC Origin Mode initially on DEC Origin Node initially on DEC Origin Node initially on DEC Origin Mode initial
	Local line editing: Auto Force on Force off Remote-controlled printing Printer to send ANSI printer output to:
	None (printing disabled)



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

Figure 6-14. Data Displayed on HyperTerminal

USB-UART communication - HyperT Eile Edit View Call Iransfer Help	erminal p						X
06 83 08 6							1
CY8CKIT-042-BLE USB-UAR	T PPSSoo(	C					
Connected 0:00:36 Auto detect	9600 8-N-1	SCROLL	CAPS	NUM	Capture	Print echo	

Figure 6-15. Data Displayed on PuTTY





# 6.2 Using PSoC 5LP as USB-I<sup>2</sup>C Bridge

The PSoC 5LP serves as a USB-I<sup>2</sup>C bridge that can be used to communicate with the USB-I<sup>2</sup>C software running on the PC. The following steps describe how to use the USB-I<sup>2</sup>C bridge, which can communicate between the BCP and the PSoC 4 BLE/PRoC BLE.

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

Figure 6-16. Create New Project in PSoC Creator

New Project			? <b>×</b>
Design Other			۵ ۵
<ul> <li>Default Templates</li> </ul>			*
PSoC 3 Design	n	Creates a PSoC 3, 8-bit 8051, design project.	
PSoC 4000 De	esign	Creates a PSoC 4000, 32-bit ARM Cortex-M0, design project.	E
PSoC 4100/4	200 Design	Creates a PSoC 4100/4200, 32-bit ARM Cortex-M0, design project.	
PSoC 4100/4	200-BL Design	Creates a PSoC 4100/4200-BL, 32-bit ARM Cortex-M0, design project.	
PRoC BLE Des	ign	Creates a PRoC BLE, 32-bit ARM Cortex-M0, design project.	
PSoC 5LP Des	ign	Creates a PSoC 5LP, 32-bit ARM Cortex-M3, design project.	
PSoC 3 Starter Design	IS		
ADC_DMA_VE	DAC	Shows how to transfer data from an ADC to a DAC using DMA with no CPU intervention.	
┣┓ DelSig_16Cha	nnel	Shows a 16-channel, 12-bit Delta Sigma ADC in PSoC 3 sequenced in hardware; samples are transferred from ADC to SRAM using DMA - without processor intervention.	
DelSig_I2CM		Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.	
DelSig_I2CS		Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.	-
Name: Design0	1		
Location: C:\Users	\BLE\Project		
Device: CX8C42	47LOLBL483		
	47EQI-DE403		
Advanced			
workspace:	Create New Workspace		•
Workspace Name:	Design01		
Sheet Template:	Empty (11" x 8.5")		•
Application Type	Normal		•
		ОК	ancel



2. Drag and drop an  $I^2C$  component (Figure 6-17) to the TopDesign.

Figure 6-17. I<sup>2</sup>C Component in Component Catalog



3. To configure the I<sup>2</sup>C component, double-click or right-click on the I<sup>2</sup>C component and select **Con-figure**, as shown in Figure 6-18.



Easte

Select All

Zoom

Shape

Configure...

## Figure 6-18. Open I<sup>2</sup>C Configuration Window

Ctrl+V

Ctrl+A

Del

٠

.

### 125



4. Change the instance name to I2C. Configure the I<sup>2</sup>C component according to the settings in Figure 6-19 and Figure 6-20 and click **OK**.

Figure 6-19. Configuration Tab

Configure 'SCB_P4'	? 🗙
Name: I2C	
Configuration I2C Basic I2C Advanced Built-in	4 ۵
Unconfigured SCB	
I2C	
© EZI2C	
SPI	
O UART	
Datasheet OK Apply C	<b>Cancel</b>

Figure 6-20. I<sup>2</sup>C Basic and Advanced Tabs

Configure 'SCB_P4'		? 💌
Name: I2C		
Configuration 12	C Basic I2C Advanced Built-in	4 ۵
Mode:	Slave 🔹	
Data rate (kbps):	400 - Actual data rate (kbps): 400	
Oversampling factor:	16 🔔 Low: 8 🏩 High: 8 🖕	
Clock from terminal		
Byte mode		
	Address R/W	
Slave address (7-bits):	0x08 0 0 0 1 0 0 X	
Slave address mask:	DxFE 1 1 1 1 1 1 1 0 LSB	
Accept matching ad	dress in RX FIFO	
Enable wakeup from	Deep Sleep Mode	
Datasheet	OK Apply	Cancel



5. Select pin P3[5] for the I<sup>2</sup>C SCL and pin P3[4] for the I<sup>2</sup>C SDA in the **Pins** tab of *<Project\_Name>.cydwr*, as shown in Figure 6-21.

Figure 6-21. Pin Selection\_USBI2C

						•	∢	₽
			*	Alias	Name 🛆	Port		
					\I2C_1:scl\	P3[5] SARMUX:pads[5], TCPWM2:line_out_compl, SCB1:uart tx, SCB1:i2c scl	r	5
P2[5]	42	OA1:vout_10x			\T2C 1:eda\	P3[4] SARMUX:pads[4], TCPWM2:line_out,	Л	5
P2[4]	41	OA1:vminus			(120_1.50a)	SCB1:uart rx, SCB1:i2c sda	Ц	-
P2[3]	40	OAhvplus, 5455:ext_ck_f						
P2[2]	39	OA0.voit_10x, SMSS:vakeup, SC50.spl_select(3)						
P2[1]	38	OA0:vminus, SC50:spi_select(2)						
P2[0]	37	OAD:vpkz, SCS0:xpl_select(1)						
VDDA	38	3.3v						
P1[7]	35	O A 3 vplus_st, TOPWW3:line_out, SOS0:usrt_cts, SOS0:spi_ck	=					
P1[6]	34	OA2vplus_st, TCPWW3:lns_out, SC50:ust_rts, SC50:ust_stell(0)	-					
P1[5]	33	O A 3:void_10x, TCPVW2:line_oid SC50:02c_ad, SC50:apl_mlao						
P1[4]	32	CA3:vminus, TCPWW2:line_out, 5 SC50:0c_ada, SC50:api_moal	•					
P1[3]	31	O A3:vplus, TCPWW1:line_out_com SC50:spl_select(3)	•					
P1[2]	30	OA2:voit_10x, TCPWW1:line_out, SC50:spl_select(2)						
P1[1]	29	0 A2:vminus, TCPWW0:line_out_o UPCOMP:comp(1), SC50:sol_select						

 Place the following code in your *main.c* project file. The code will enable the PSoC 4 BLE/PRoC BLE device to transmit and receive I<sup>2</sup>C data to and from the BCP application.

```
int main()
{
uint8 wrBuf[10]; /* I<sup>2</sup>C write buffer */
uint8 rdBuf[10]; /* I<sup>2</sup>C read buffer */
uint8 indexCntr;
uint32 byteCnt;
/* Enable the Global Interrupt */
CyGlobalIntEnable;
/* Start I<sup>2</sup>C Slave operation */
I2C_Start();
/* Initialize write buffer */
I2C_I2CSlaveInitWriteBuf((uint8 *) wrBuf, 10);
/* Initialize read buffer */
```



```
I2C I2CSlaveInitReadBuf((uint8 *) rdBuf, 10);
for(;;) /* Loop forever */
{
/* Wait for I<sup>2</sup>C master to complete a write */
if(Ou != (I2C_I2CSlaveStatus() & I2C_I2C_SSTAT_WR_CMPLT))
       /* Read the number of bytes transferred */
      byteCnt = I2C I2CSlaveGetWriteBufSize();
       /* Clear the write status bits*/
       I2C I2CSlaveClearWriteStatus();
/* Move the data written by the master to the read buffer so that the
      master can read back the data */
      for(indexCntr = 0; indexCntr < byteCnt; indexCntr++)</pre>
       {
rdBuf [indexCntr] = wrBuf[indexCntr]; /* Loop back the data to the read
             buffer */
       }
/* Clear the write buffer pointer so that the next write operation will
       start from index 0 */
       I2C_I2CSlaveClearWriteBuf();
/* Clear the read buffer pointer so that the next read operations starts
       from index 0 */
       I2C I2CSlaveClearReadBuf();
}
/* If the master has read the data , reset the read buffer pointer to 0
and clear the read status */
if(Ou != (I2C_I2CSlaveStatus() & I2C_I2C_SSTAT_RD_CMPLT))
/* Clear the read buffer pointer so that the next read operations starts
from index 0 */
       I2C I2CSlaveClearReadBuf();
       /* Clear the read status bits */
       I2C I2CSlaveClearReadStatus();
}
}
}
```

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



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

Figure 6-22. Connecting to KitProg/ in BCP

W Bridge Control Panel	
File Editor Chart Execute Tools Help	
🖻 🖩 🗑 🕲 🗠 🗢 🧮 🧱 🧱	
Editor Chart Table File	
	A
	-
	4
COM6 Serial Port	A
Opening Port	
KitProg Version 2.08	
	E
4	
Connected I2C/SDI/DV9 Parter	
Connected 22/31/1/06/018.     KtProg/051717FF011B3400	Power Protocol
	0 +3.3V 0 SPI
Scan period, ms: 0	(O) +2.5V ○ +1.8V (UART)
1:1 Syntax: OK Connected Powered	Voltage: 4566 mV .:



10.Open **Protocol Configuration** from the **Tools** menu and select the appropriate **I2C Speed**, as shown in Figure 6-23. Make sure the I<sup>2</sup>C speed is the same as the one configured in the I<sup>2</sup>C component. Click **OK** to close the window.

Figure 6-23. Opening Protocol Configuration Window in BCP



11. From the BCP, transfer five bytes of data to the I<sup>2</sup>C device with slave address 0x08. Type the command shown in Figure 6-24 and press **[Enter]** or click the **Send** button in the BCP. The log shows whether the transaction was successful. A '+' indication after each byte indicates that the transaction was successful and a '-' indicates that the transaction was a failure.

Figure 6-24. Entering Commands in BCP

🗱 Bridge Control Panel	X
File Editor Chart Execute Tools Help	
Editor Chart Table File	
W 8 aa bb cc dd ee P Generate STOP condition on I2C bus Data Bytes Slave Address Write data' command	*
( )	*
Opening Port	
Successfully Connected to KitProg/051717FF011B3400 KitProg Version 2.08 w 08+ AA+ BB+ CC+ DD+ EE+ p	
	-
	P.
Connected I2C/SPI/RX8 Ports:         Power         Protocol           Image: Scop         Send all strings:         Image: Connected I2C/SPI/RX8 Ports:         Power         Protocol           Image: Scop         Image: Scop         Scan period, ms:         Image: Other Science Sci	
1:21 Syntax: OK ok Connected Powered Voltage: 4575 mV	.4



12. From the BCP, read five bytes of data from the I<sup>2</sup>C slave device with slave address 0x08. The log shows whether the transaction was successful, as shown in Figure 6-25.

Figure 6-25. Read Data Bytes from BCP

🗱 Bridge Control Panel		
File Editor Chart Execute Tools Help		
Editor Chart Table File		
r 8 x x x x P		*
Generate STOP condic	tion on I2C bus	
No. of data bytes to be read		
Slave Address		
'Read data'		
command		
Data bytes retu	rned after read operation	-
		E.
Opening Port		*
Successfully Connected to Ki	tProg/051717FF011B3400	
KitProg Version 2.08		E
r 08+AA+BB+CC+DD+EE+p		
		*
4		P
	Connected I2C/SPI/RX8 Ports:	ol
Reset Send all strings	RIProg/051717FF011B3400	
Stop Repeat To file	0 +3.3V O SPI	0.04070
Scan period, m	S. UT	B (UART)
1 · 14 Suntay : OK	Connected Dowered Voltage: 4575 mV	
	Connected Formered Voldge: 4575 IIV	

Note: Refer to Help Contents under Help in BCP or press [F1] for details of I<sup>2</sup>C commands.



# 6.3 Developing Applications for PSoC 5LP

The BLE Pioneer Kit has an onboard PSoC 5LP whose primary function is that of a programmer and a bridge. You can build either a normal project or a bootloadable project using the PSoC 5LP.

The PSoC 5LP connections in the Pioneer board are summarized in Figure 6-26. J8 is the I/O connector. The USB (J13) is connected and used as the PC interface. However, you can still use this USB connection to create customized USB designs.



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

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

## 6.3.1 Building a Bootloadable Project for PSoC 5LP

All bootloadable applications developed for the PSoC 5LP should be based on the bootloader hex file, which is programmed onto the kit.

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

<Install\_Directory>\CY8CKIT-042-BLE Kit\<version>\Firmware\Programmer\ KitProg\_Bootloader

Figure 6-27. KitProg Bootloader Hex File Location

<b>30-</b> • • «	1.0 • 1	Firmware • F	Programmer • K	tProg_Boo	otloader 👻	* <b>7</b> Sea
Organize • 1	Include	in library 🕶	Share with •	Burn	New folder	
Favorites	^	Name	*		Date modified	Туре
E Desktop		KitProg	_Bootloader.elf		3/18/2013 6:38 PM	ELF File
Downloads		KitProg	_Bootloader.hex	>	3/18/2013 6:38 PM	HEX Fil



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

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

Figure 6-28. Create New Project in PSoC Creator\_PSoC 5LP

New Project	
Design Other	4 Þ
<ul> <li>Default Templates</li> </ul>	
PSoC 3 Design	Creates a PSoC 3, 8-bit 8051, design project.
PSoC 4000 Design	Creates a PSoC 4000, 32-bit ARM Cortex-M0, design project.
PSoC 4100/4200 Design	Creates a PSoC 4100/4200, 32-bit ARM Cortex-M0, design project.
PSoC 4100/4200-BL Desig	n Creates a PSoC 4100/4200-BL, 32-bit ARM Cortex-M0, design project.
PRoC BLE Design	Creates a PRoC BLE, 32-bit ARM Cortex-M0, design project.
PSoC 5LP Design	Creates a PSoC 5LP, 32-bit ARM Cortex-M3, design project.
PSoC 3 Starter Designs	
ADC_DMA_VDAC	Shows how to transfer data from an ADC to a DAC using DMA with no CPU intervention.
▶ DelSig_16Channel	Shows a 16-channel, 12-bit Delta Sigma ADC in PSoC 3 sequenced in hardware; samples are transferred from ADC to SRAM using DMA - without processor intervention.
DelSig_I2CM	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.
DelSig_I2CS	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.
Name: Design01	
Location: C:\Users\BLE\Project	
Device: CY8C5868I TI-I P039	
Workenzoe:	Nel:esee
Workspace.	voikspace
Workspace Name: Design01	
Sheet Template: Empty (11" x	8.5") 🔹
Application Type Bootloadable	•
	OK Cancel



	Hide/show columns	-	rvese	1 10 1	Jeraur	IS	_	_	_	_	24 C	sium		
<b>E</b>	Archtecture	CPU Speed (MHz)	Flash (KB)	SRAM (KB)	EEPROM (bytes)	Trace Buffer (KB)	DMA Channels	PLL	LCD Drive (mux ratio)	CapSense	ADC	8-bit DAC	SOUT Blocks	and and a
Filters:														
CY8C5867LTI-LP025	PSoC 5LP (ARM CM3)	67	128	32	2048	-	24	1	x16	1	1x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5867LTI-LP028	PSoC 5LP (ARM CM3)	67	128	32	2048	-	24	1	x16	1	1x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868AXI-LP031	PSoC 5LP (ARM CM3)	67	256	64	2048	-	24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868AXI-LP032	PSoC 5LP (ARM CM3)	67	256	64	2048	-	24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868AXI-LP035	PSoC 5LP (ARM CM3)	67	256	64	2048	-	24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868LTI-LP036	PSoC 5LP (ARM CM3)	67	256	64	2048	-	24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868LTI-LP038	PSoC 5LP (ARM CM3)	67	256	64	2048		24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	
CY8C5868LTI-LP039	PSoC 5LP (ARM CM3)	67	256	64	2048		24	1	x16	1	2x 12-bit SAR 1x 20-bit Delta Sigma	4	4	

Figure 6-29. Select Device in PSoC Creator

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

Figure 6-30. Bootloadable Component in Component Catalog





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

Figure 6-31.	Configuration	Window o	f Bootloadable	Component in	"General" t	ab setting

Configure 'Bootloadable	ť	? 🔀
Name: Bootloadable	_1	
General Deper	dencies Built-in	4 Þ
Application version:	0x0000	
Application ID:	Qx0000	
Application custom ID:	0x00000000	
Manual application i Placement address:	mage placement	
Datasheet	OK Apply	Cancel

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

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

Configure 'Bootloadable'	? 🔀
Name: Bootloadable_1	
General Dependencies Built-in	4 Þ
Bootloadable projects require a reference to the associated Bootloader project's HE files. The HEX files extension is ".hex. The ELF files extension depends on IDE and ".elf, ".out, ".axf, or other.	X and ELF can be
Bootloader HEX file:	
В	rowse
Bootloader ELF file:	
В	rowse
Datasheet OK Apply	Cancel



Select a Bootloader Hex File	Programmer & VitBrog Rootlander	- to Com	X
Organize • New folder	Programmer / Kitriog_boolioader	• • • Jeuro	8 · · · ·
Desktop	Name	Date modified	Туре
Downloads	KitProg_Bootloader.hex	3/18/2013 6:38	HEX File
S Recent Places			
☐ Libraries 3 Documents 4 Music 3 Pictures 5 Videos			
Scomputer			
Windows7_OS (C: Lenovo_Recovery + <			•
File <u>n</u> ame:	KitProg_Bootloader.hex	Hex Files     Open	*.hex)   Cancel

Figure 6-33. Select KitProg Bootloader Hex File

Figure 6-34. Select KitProg Bootloader Elf File

Organize New folder     Organize New folder     Image: Programmer + KitProg_Bootloader     Organize     New folder     Image: Programmer + KitProg_Bootloader     Organize     New folder     Image: Programmer + KitProg_Bootloader     Organize     New folder     Image: Programmer + KitProg_Bootloader	Select a Bootloader Hex File		×
Organize New folder     Favorites   Desktop   Downloads   Recent Places     Documents   Music   Pictures   Videos   Videos   Computer   Local Disk (C:)   CY8C58 Family Processo     Yes     Name     Date modified   Type   Date modified   Type     Videos     Computer     Cy8C58 Family Processo     Tube     Name     Date modified     Type     Name     Date modified     Videos     Image: Computer     Videos     Image: Computer     Videos	G V Sirmware > Pr	ogrammer 🕨 KitProg_Bootloader	✓ 4y Search KitProg_Bootloader
Image: Second secon	Organize 👻 New folder		ii • 🗊 🔞
<ul> <li>Desktop</li> <li>Downloads</li> <li>Recent Places</li> <li>Libraries</li> <li>Documents</li> <li>Music</li> <li>Pictures</li> <li>Videos</li> <li>KitProg_Bootloader.elf</li> <li>4/18/2013 1:07 AM ELF File</li> </ul>	🔆 Favorites	^ Name	Date modified Type
<ul> <li>▶ Downloads</li> <li>▶ Recent Places</li> <li>■ Documents</li> <li>▶ Music</li> <li>▶ Pictures</li> <li>■ Videos</li> <li>▶ Computer</li> <li>▲ Local Disk (C:)</li> <li>▶ CY8C58 Family Processo ▼</li> </ul>	Desktop	KitProg_Bootloader.elf	4/18/2013 1:07 AM ELF File
Ibraries         Ibraries         Documents         Music         Pictures         Videos         Local Disk (C:)         Ibraries         CY8C58 Family Processo ▼	Downloads	E	
Documents Music Pictures Videos Komputer Local Disk (C:) CY8C58 Family Processo * * * * ***************************	🥽 Libraries		
<ul> <li>Pictures</li> <li>Videos</li> <li>Monte Computer</li> <li>Local Disk (C:)</li> <li>UV8C58 Family Processo ▼</li> </ul>	Documents     Music		
Videos   Computer  Local Disk (C:)  CY8C58 Family Processo	Fictures		
r Computer Local Disk (C:) B CY8C58 Family Processo ▼ <	📓 Videos		
Local Disk (C:)	🖳 Computer		
VY8C58 Family Processo 👻 👘 👘	🕌 Local Disk (C:)		
	CY8C58 Family Processo	• • • [	,
File name:	File <u>n</u> ame:		✓ Elf Files (*.elf, .axf, .out) ▼
<u>Open</u> Cancel			Qpen Cancel

3. Develop your custom project.



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

Figure 6-35. KitProg Bootloader System Settings

otion	Value
Configuration	
Device Configuration Mode	Compressed
Enable Error Correcting Code (ECC)	
- Store Configuration Data in ECC Memory	
- Instruction Cache Enabled	
Enable Fast IMO During Startup	
Unused Bonded IO	Allow but warn
Heap Size (bytes)	0x80
Stack Size (bytes)	0x0800
Include CMSIS Core Peripheral Library Files	
Programming\Debugging	
Debug Select	SWD+SWV (serial wire debug and viewer)
Enable Device Protection	
Embedded Trace (ETM)	
Use Optional XRES	
Operating Conditions	
- VDDA (V)	5.0
Variable VDDA	
- VDDD (V)	5.0
- VDDIO0 (V)	5.0
- VDDIO1 (V)	5.0
- VDDIO2 (V)	5.0
- VDDIO3 (V)	5.0
ue, device configuration data will be stored in ECC memory to reduce main FLASH memory usage. Error correction may not be used when this option is enabled.	

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

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





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

Figure 6-37.	Port Filters	Tab in	<b>Bootloader Host</b>	Tool
1 19410 0 011	1 0101 110010	100 111	50000000111000	

ile: C:\Program Files Ports:	(x86)\Cypress\CY8CKIT-042 PSoC 4 Pioneer Kit \1.0\Fi Filters Port Configuration	imware\Programmer\KtProg\KtProg.cyact
og:	Port Filters Show I2C Devices Show SPI Devices Show UART Devices Show USB Devices VID: 0x04B4 PID: 0xF13B Cancel OK	
15:46:31 PM - Selected	d device: USB HumaL	



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

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

Bootloader Host		- C ×
Elle Actions Help		
🖆 🔰 BB 📎 🚳		
Fle: C1_Jsers\ancy\Desktop\Bootloadable project	f(Boofloadable.cydsn(CortexM3)ARM_GCC_	441\Debug\Bootloadable.cyacd
Ports: Filters USB F uman Interface Device (04B4_F13B) Program Button Open File Button	Port Configuration USB * No configuration necessary for this port.	Port Information VID: 0484 PID: F138
Log:		
1235.02 PM - Selected device: USB Human Interfac 1235.02 PM - Selected device: USB Human Interfac 1235.08 PM - Selected device: USB Human Interfac	e Device (0484_F138) e Device (0484_F138) e Device (0484_F138)	

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

		S [ 1] Search break	-
Organize • New folder		恒•	. 0
Favorites	Name	Date modified	Туре
E Desktop	🐌 .deps	4/18/2013 12:34 PM	File folde
👍 Downloads 🛛 🔇	Bootloadable.cyacd	4/18/2013 12:34 PM	CYACD FI
🚆 Libraries			
Libraries			
Ubraries			
Ubraries Documents Music Pictures Videos Computer Mundows7 OS (C: * *	111		

 Keep the reset switch (SW1) pressed and plug in the USB mini-B connector. If the switch is pressed for more than 100 ms, the PSoC 5LP enters into bootloader. Click the Program button (Figure 6-38) in the Bootloader Host tool to program the device.



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

### Notes:

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

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



# 6.3.2 Building a Normal Project for PSoC 5LP

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

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

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

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

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

New Project		? 🗙
Design Other		4 ۵
Default Templates		<b>^</b>
PSoC 3 Design	Creates a PSoC 3, 8-bit 8051, design project.	
PSoC 4000 Design	Creates a PSoC 4000, 32-bit ARM Cortex-M0, design project.	E
PSoC 4100/4200 Design	Creates a PSoC 4100/4200, 32-bit ARM Cortex-M0, design project.	
PSoC 4100/4200-BL Design	Creates a PSoC 4100/4200-BL, 32-bit ARM Cortex-M0, design project.	
PRoC BLE Design	Creates a PRoC BLE, 32-bit ARM Cortex-M0, design project.	
PSoC 5LP Design	Creates a PSoC 5LP, 32-bit ARM Cortex-M3, design project.	
PSoC 3 Starter Designs		
ADC_DMA_VDAC	Shows how to transfer data from an ADC to a DAC using DMA with no CPU intervention.	
▶ DelSig_16Channel	Shows a 16-channel, 12-bit Delta Sigma ADC in PSoC 3 sequenced in hardware samples are transferred from ADC to SRAM using DMA - without processor intervention.	
Pa DelSig_I2CM	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.	
DelSig_I2CS	Shows the 16-bit differential ADC, hardware multiplexed into 8 channels and transported over I2C.	-
Name: Design01		
Location: C:\Users\BLE\Project		
Device: CY8C5868LTI-LP039		•
Advanced		
Workspace: Create New Workspace		•
Workspace Name: Design01		
Sheet Template: Empty (11" x 8.5")		-
Application Type Bootloadable		•
	ОК	ancel

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

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



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

### Notes:

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

# 6.4 **PSoC 5LP Factory Program Restore Instructions**

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

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

## 6.4.1 PSoC 5LP is Programmed with a Bootloadable Application

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

### 6.4.1.1 Restore PSoC 5LP Factory Program Using PSoC Programmer

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



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

Figure 6-41. PSoC Programmer Results Window

👺 PSoC Programmer	
File View Options H	elp
📄 • 🗼 🔘 BB	
Port Selection	Programmer Utilities JTAG
Device Family CY9C3oox Device CY9C3866AXI-040	Programming Parameters         File Path:       C:\Program Files (x86)\Cypress\CY3CKIT-042-BLE Kit1.0\Firmware\Programmer\KitProg_Bootloader\KitProg_Bootloa
Actions	Results
Actions Connected at 1:38:4	Results           6 FM         KitProg bootloader device is detected           Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge



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

Figure 6-42. Upgrade Firmware

PSoC Programmer	
File View Options Help	
🖆 · 🍡 💿 bb 🕻	
Port Selection Progr	ammer Utilities JTAG
	Upgrade Firmware Click to upgrade connected device's firmware
	Erase Block Click to erase user specific flash block
Device Family	
CY8C3xxx	
Device	
CY8C3866AXI-040 -	
Actions	Results
Actions Connected at 1:38:46 PM	Results RitBrog bootloader device is detected
Actions Connected at 1:38:46 PM	Results RitProg bootloader device is detected Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware
Actions Connected at 1:38:46 PM	Results KitProg bootloader device is detected Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results KitProg bootloader device is detected Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results KitProg bootloader device is detected Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results KitProg bootloader device is detected Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge
Actions Connected at 1:38:46 PM	Results <u>KitProg bootloader device is detected</u> Please close all ports, then navigate to the Utilities tab and click the Upgrade Firmware button to recover Bridge


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

Figure 6-43. Firmware Update Completed

🖗 PSoC Programmer	
File View Options Help	
📂 · 🔪 💿 BB 🕻 🖹 🗋 🕲	
Port Selection IVilities JTAG	
KitProg/051717FF011B340     Upgrade Rimware     Click to upgrade connected device's firmware     Erase Block     Click to erase user specific flash block	
Device Family       CYBC300x       Device       CYBC3866AX0-040	
Actions Results	*
Successfully Connected to KitProg/051717FF01B3400 KitProg Version 2.08 at 1:41:51 PM	
PM	
Connected at 1:41:49 PM KitProg/051717FF011B3400	=
Disconnected at 1:41:33 PM Bootloader device	_
Firmware Update Finished at 1:41:33 PM	
Succeeded	
Verifying	
Initializing	
Firmware Upgrade Started at 1:41:24 PM	
Firmware Upgrade	-
For Help, press F1 PASS Powered	Connected .::

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



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

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

🛓 Bootloader Host							×
<u>File Actions H</u> elp							
🔁 🕽 66 🚫	$\otimes$						
File: C:\Program Files (x86)\Cvpre	ess\CY8CKIT-	042-BLE Kit\1.0\Fimwa	are\Programm	er\KîtProa\ł	KitProg.cvacd		
Ports:	Filters	Port Configuration		▼ Port	Information		
Log:							
Ready							-
O v ↓ « Firmware ↓	Programm	er 🕨 KitProg	<b>-  -  -  + - → - - - - - - - - - -</b>	Search Kit	Prog	2	
Organize  New folder					8== 🗸		
A New Tolder	▲ Nam	A			Date modifi		
☆ Favorites ↓ Downloads Recent Places ■ Desktop		CitProg.cyacd			6/2/2014 5:0		
Libraries	E					No preview available.	
▼数 Lenovo_Recovery (Q:)							
File <u>n</u> an	e: KitProg.c	yacd	- E	Bootloader Open	Files (*.cyacd)	<b>→</b> ancel	

Figure 6-44. Load KitProg.cyacd File



- Configure the Pioneer Kit in service mode. To do this, while holding down the reset button (SW1 Reset), plug in the BLE Pioneer Kit to the computer using the included USB cable (USB A to mini-B). This puts the PSoC 5LP into service mode, which is indicated by the blinking green status LED.
- 4. In the Bootloader Host tool, set the filters for the USB devices with VID: 04B4 and PID: F13B. The USB Human Interface Device port appears in the Ports list. Click that port to select it, as shown in Figure 6-45.

Bootloader Host - • • <u>F</u>ile Actions Help File C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Firmware\Programmer\KitProg\KitProg.cyacd Port Filters. Port Configuration USB Port Information VID: 04B4 PID: F13B No configuration necessary for this USB Human Interface D port Log: 01:48:37 PM - Selected device: USB Human Interface Device (04B4\_F13B) Ready

Figure 6-45. Select USB Human Interface Device

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



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

Figure 6-46. Programming Finished Successfully

File       Actions       Help         Image: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Fimware\Programmer\KitProg\KitProg.cyacd       Image: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Fimware\Programmer\KitProg\KitProg.cyacd         Ports:       Filters       Port Configuration       Port Information         Image: C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Fimware\Programmer\KitProg\KitProg.cyacd       Image: C:\Program Files (x86)\Cypress\Cypre		🛓 Bootloader Host
Rie:       C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Fimware\Programmer\KitProg\KitProg.cyacd         Ports:       Filters         Port Configuration       Port Information         Log:       D1:48:37 PM - Selected device: USB Human Interface Device (04B4_F13B)         D2:202 04 PM - Programming Enished Successfully		<u>F</u> ile <u>A</u> ctions <u>H</u> elp
File:       C:\Program Files (x86)\Cypress\CY8CKIT-042-BLE Kit\1.0\Fimware\Programmer\KitProg\KitProg.cyacd         Ports:       Filters         Port Configuration       Port Information         Log:       D:148:37 PM - Selected device: USB Human Interface Device (0484_F13B)         02:20:04 PM - Programming Entished Successfully		🖆 🗼 BB 📎 🛞
Ports:       Filters       Port Configuration       Port Information	Firmware\Programmer\KitProg\KitProg.cyacd	File: C:\Program Files (x86)\Cypress\CY8CKIT-
02:02:04 PM - Programming Started 02:02:08 PM - Programming Finished Successfully	ion Pot Information IB4_F13B)	Ports: Filters Filters Log: 01:48:37 PM - Selected device: USB Human Inte
Programming completed in 4367ms.		02:02:04 WH - Programming Started 02:02:02 HW - Programming Started 02:02:02 HW - Programming Finished Successful Programming completed in 4367ms.

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

# 6.5 Using FM24V10 F-RAM

The BLE Pioneer board has an onboard ferroelectric RAM chip that can hold up to 1 Mb of data. The chip provides an I<sup>2</sup>C communication interface for data access. It is hardwired to the I<sup>2</sup>C lines (P3\_4 and P3\_5); the same lines are also routed to the PSoC 5LP I<sup>2</sup>C lines. Because the F-RAM device is an I<sup>2</sup>C slave, it can be accessed or shared among various I<sup>2</sup>C masters on the same line. For more details on the F-RAM device, refer to the device datasheet.



### 6.5.1 Address Selection

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

Figure 6-47. F-RAM I<sup>2</sup>C Address Byte Structure



#### 6.5.2 Write/Read Operation

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

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



Single-Byte Write





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

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

## 6.6 CySmart iOS Application

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

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

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

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



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

Turn On Bluetooth "CySmart" to Con Accessories	to Allow nect to
Settings	ок

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







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



iPod	2:10 PM	@ \$ 💼+
$\bigotimes$	Profile	
	Capsense	
	CapSonse Button & Proximity	]

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





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



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

Figure 6-54. GATT DB GUI for Characteristics



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

Figure 6-55. Data Logger





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

The CapSense profile GUI supports three CapSense functionalities.

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

Figure 6-56. CapSense Buttons GUI Page





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

Figure 6-57. CapSense Slider GUI Page





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

Figure 6-58. CapSense Proximity GUI Page





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

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

iPod 12:34 PM + ★ ➡ C RGB LED = Current Color Red 0xff Green 0x00 Blue 0x00 Intensity 0xff Share

Figure 6-59. RGB LED Profile



# 6.7 CySmart PC Tool

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

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

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

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

Figure 6-60. Connect Dongle to USB Port





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

Figure 6-61. Selecting Dongle in CySmart PC Tool

CySmart 1.0				
Ele Help				
() <sup>*</sup> Select Dongle				
	Select BLE Dongle Target			
		Details Manufacturer: Product: Fernware version: Hardware version: Description: Cypress BLE dongle	Cypress Semiconductor Cypress BLE Dongle 1.0.0.35 1.0.0.0	
	· •			
	Show all	]		
	Refresh		Connect Close	
Log		_		
🍵 Clear Log 🔡 Save Log				
				-

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



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

Elie       Help       (1) Menu bar            Git State Dongle	
Baser       Compose managers and general gener	
Discovered devices         Image: Start Scan       Add to Whitelot	
Image: Start Start (Start Start Start)       (3) Actions that can be taken on a discovered device       Advetsement data   Scan response data           If Device       Buetooth Address Type (RSS)       Advetsement Type (Connected)       Image: Start St	
# Device Buetoch Address Type RSSI Advertisement Type Connected Decorption Value Index	
Decoption Value Index	
(5) This Window displays the list of Devices that have been scanned and discovered by the BLE Dongle after 'Start Scan' is clicked (4) This window displays the data rec part of advertisement packet and the Response packet	eived as Scan
Whated (7) Data of the extended field	
Add Enemove Clear All & Refresh	
# Buildooth Address Address Type	
(6) Whitelist options allow a peripheral device to be	
added or removed from the too's whitelist	
29	
Clear Log Without Same	
171143]: Und Load 8D Address personal event reacted 171145]: BD Address personal Revent Reacted 171145]: BD Address personal Revent Reacted 171145]: BD Address personal Revent Reven	

Figure 6-62. CySmart PC Tool Main Window

The important parts of this window are as follows:

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



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

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

CySmart 1.0						
File Help						
Select Dongle "Ø Config Master	ure Master Settings	Manage PSN	1s		Listed	devices found during BLE scan
Stop Scan Connect	🛃 Add to Whitelist					Advertisement data   Scan response data
# Device	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected	E 1
1 CapSense Slider and LED	22:43:65:56:34:12	Public	-67 dBm	Connectable undirected		Description Value Index
					<b>**</b>	

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

laster	Send connect request t	o selected device	received	from selected device
Start Scan H Co	annat 🗖 Add to Whitefut		Advertisement data   Scan associate data	
g start scart Q CO	S and Mineral	in the second		
Device	buetoon Address Address Type HISSI	Advertisement Type Connected	Description	Value Index
Captorial Score at	11EU 22.431629634112 PUBK -//1086		D AD Data 0: ccBanaco	These Freedom
		Selected BLE peripheral device	Length of this data	0x02 101
			E- (cflags>)	0:01 [1]
			B Rag Data: 0x06	0x06 [2]
			LE Limited Discoverable Mode	OFF
			- LE General Discoverable Mode	ON
			- BR/EDR Not Supported	ON
			- Simultaneous LE and BR/EDR to Same Device Capable (Controller)	OFF
			- Smultaneous LE and BR/EDR to Same Device Capable (Host).	OFF
			Reserved	OFF
			Reserved	OFF
			Reserved	OFF
Intelst			AD Data 1: < <complete local="" name="">&gt;</complete>	
Add Remove	fill Clear All G Refresh		- Length of this data	0x18 [3]
Distorth Address A	iddaaa Tura		El- < <complete local="" name="">&gt;</complete>	0x09 [4]
Buttour Aburdes A	and an other			0x43 [5]
			- 8	0x61 [6]
			p	0x70 [7]
			Raw Data	
			02:01:06:18:09:43:61:70:53:65:6E:73:65:20:53:6C:69:64:65:72:20:61:6E:64:20:4C:	45:44
a				
Clear Log	log			
23:04] : 'Stop Scan' n	equest sent			
[23:04] : 'Scan Stoppe	ed Notification' event received			



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

Figure 6-65. Connected Device Tab



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

CySmart 1.0					
<u>F</u> ile <u>H</u> elp					
Select Dongle 🥒 Cor	nfigure Ma	ster Settings Manage PSMs			Attributes read and listed
Master CapSense Slider a	nd LED [22:	43:65:56:34:12]			Attributes read and listed
Attributes					
💊 Stop 🛛 🔝 Enable All I	Notification	ns 🔟 Read All Characteristics 💸 Bond	i 🛛 🔛 Export 🛛 🎁 Clear	/	View: Category 👻 陆 🖬
Handle	UUID	UUID Description	Value	Properties	
Primary Service Declarat	tion: Generic	Access			
<b>⊡</b> -0x0001	0x2800	Primary Service Declaration	00:18 (Generic Access)		
Characteristic D	eclaration: D	levice Name		59. 	
⊡ 0x0002	0x2803	Characteristic Declaration	02:03:00:00:2A		
0x0003	0x2A00	Device Name		0x02	
Characteristic D	eclaration: A	ppearance			
⊡ 0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A		
0x0005	0x2A01	Appearance		0x02	
E Characteristic D	eclaration: P	eripheral Preferred Connection Parameters			
	0x2803	Characteristic Declaration	02:07:00:04:2A		
0x0007	0x2A04	Peripheral Preferred Connection Parameter	ers	0x02	
Primary Service Declarat	tion: Generic	: Attribute			
E−0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)		
- Characteristic D	eclaration: S	ervice Changed			
⊡. 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A		
A000x0	0x2A05	Service Changed		0x22	
Primary Service Declarat	tion				
	0x2800	Primary Service Declaration	B6:CA		
Characteristic D	eclaration			58. 	
⊡ 0x000C	0x2803	Characteristic Declaration	10:0D:00:A2:CA		
0x000D	0xCAA2			Ox10	
0x000E	0x2902	Client Characteristic Configuration			



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

CySmart 1.0						
Eile Help						
Select Dongle 👋 Cor	figure Mas	ter Settings 🛠 Manage PSMs 👸 Discor	inect			
Master CapSense Sider a	nd LED (00.	AD 50:65:43:21]			Selected Attribute details	
Whites					Athbute Details Send Commands	
Discover All Attributes	Enak	le All Notifications 🛄 Read All Character	istics 🧬 Pair 🔛 Export 🍵	Clear View: Category 👻 🖬 🍱	Handle: 0x0013	
lande	UUID	UUID Description	Value	Properties	UUID: GcBB1	
0x0005	0x2A01	Appearance	00:00	Q-02	UUID Description:	
E Characterístic D	eclaration: P	erpheral Preferred Connection Parameters			Value:	
⊡ 0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A		00.00.00.00	
0x0007	0x2A04	Perpheral Preferred Connection Parameters	06:00:50:00:00:00:90:01	0x02	en en miliel uperil	
Primary Service Declarat	ton: Generic	Abribute				
B- 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)		1	Read Value * Write Val
- Characteristic D	eclaration: S	ervice Changed			Dente Court	
iii 0x0009	0x2803	Characteristic Declaration	22-0A-00-05-2A	Characteristic Read	Properties Enabled	
- 0x000A	0x2A05	Service Changed	00:00:00:00	0/22	Broadcast	-
- 0x0008	0x2902	Client Characteristic Configuration			Read	<u> </u>
Primary Service Declarat	ton				Wite without response	-
B- 0x000C	0x2800	Primary Service Declaration	BS:CA		Wite	<u> </u>
Characteristic D	eclaration		to constants		Netfy	2
B-0x0000	0x2803	Characteristic Declaration	10.0E-00.A2:CA		Indicate	
0x000E	DICA42			0x10	Authenticated signed writes	
- 0x000F	0x2902	Client Characteristic Configuration			Extended propeties	
- 0x0010	0x2901	Characteristic User Description				
Primary Service Declarat	tion					
E 0x0011	0x2800	Primary Service Declaration	BBCB			
Diaracteristic D	eclaration	In the second second second				
⊡-0x0012	0x2803	Characteristic Declaration	1A.13.00.B1.CB			
0/0013	OKCER1		00:00:00:00	DilA		
- 0x0014	0x2902	Client Characteristic Configuration				
- Dx0015	0x2901	Characteristic User Description				
Atributes L2CAP Channel	la l		1			

Figure 6-67. Read All Characteristics

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

Figure 6-68. Modify a Characteristic

CySmart 1.0								
Eile Help								
Select Dongle 👋 Co	nfigure Ma	ster Settings 🛠 Manage PSMs 岗 <u>D</u> iscor	nect					
Master CapSense Sider a	and LED (00	AD 50:65:43:21]						
Atributes							Athibute Details Send Comma	nds
Discover All Attribute	s 🚺 Ena	ble All Notifications 🧰 Read All Character	istics 🥐 Pair 🔛 Export 🥤	Clear	Views Category -		Handle: 0x0013	Juneally and HURD of the anti-stand abarrantication
Handle	UUID	UUID Description	Value	Properties			UUID: 0xC881	Handle and UUID of the selected characteristic
- 0x0005	0x2A01	Appearance	00:00	0402			UUID Description:	
El·Oharacterístic D	eclaration: A	Perpheral Preferred Connection Parameters					Value: Data on th	e attribute
⊡-0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A				00:FF:00:FF	
0x0007	0x2A04	Perpheral Preferred Connection Parameters	06:00:50:00:00:00:90:01	Dx02				
Primary Service Declara	tion: Generic	: Abibute						
E- 0x0008	Gx2800	Primary Service Declaration	01:18 (Generic Attribute)			1		Actions allowed on the characteristic Read Value * Write Value *
B- Characteristic D	eclaration: 5	Service Changed					former former	
⊡-0x0009	Dx2803	Characteristic Declaration	22:0A:00:05:2A				rioperces	14040
Dx000A	Dx2A05	Service Changed	00:00:00:00	Dx22			Broadcast	
0x0008	0x2902	Client Characteristic Configuration					Plead	¥
Primary Service Declara	tion						Wite without response	
E-0x000C	0x2800	Primary Service Declaration	B5 CA				Wite	
😑 Characteristic D	eclaration	*****					Notfy	
B- 0x0000	0x2803	Characteristic Declaration	10:0E:00:A2:CA				Indicate	11
- 0x000E	DICAA2			De10			Authenticated signed writes	H
- 0x000F	0x2902	Client Characteristic Configuration					Extended properties	
- 0x0010	0x2901	Characteristic User Description						Supported properties
Primary Service Declara	tion							
⊖ 0x0011	0x2800	Primary Service Declaration	BB:CB					
Oharacteristic 0	eclaration							
B- 0x0012	0x2803	Characteristic Declaration	1A:13:00:81:C8		Selected Characteristic			
0.0013			00 FF 00 FF	Di 1A				
- Dx0014	0x2502	Client Characteristic Configuration			1			
0x0015	0x2901	Characteristic User Description						
Abributes   L2CAP Chann	els			1				

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



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

🖸 CySmart 1.0										
<u>File H</u> elp										
😝 Select Dongle 🥒 Co	nfigure Ma	ster Settings Manage PSMs								
Master CapSense Proxim	ity [21:43:65	:56:34:12]				~				
Attributes							Attribute Details Send Commands			
Oiscover All Attribute	es 🛛 🚺 Ena	ble All Notifications   Read All Characteri	istics 🔄 Bond 🔛 E	xport	💼 Clear	View: Category 👻 🖬	Handle: 0x000D			
Handle	UUID	UUID Description	Value	Proper	ties		UUID: 0xCAA1			
Primary Service Declar	ation: Generi	c Access					UUID Description:			
⊡ 0x0001	0x2800	Primary Service Declaration	00:18 (Generic Access)		(	¥ 111	Value			
Characteristic [	eclaration: [	Device Name			Save As					
⊡ • 0x0002	0x2803	Characteristic Declaration	02:03:00:00:2A		🕒 🗢 🧊 🕨 Librari	es 🕨	✓ 4 Search Libraries			
0x0003	0x2A00	Device Name		0x02	Organize 💌		9° - 6			
Characteristic [	eclaration: A	Appearance			and a second second	Organize • Die •				
⊡ 0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A		Recent Places	Libraries				
Cx0008	0x2A01	Appearance		0x02	Desktop	Open a library to see your files and arrange	them by folder, date, and other properties.			
- Characteristic [	eclaration: F	Peripheral Preferred Connection Parameters			I libraries					
⊡0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A		Documents	E Library	Library			
0x0007	0x2A04	Peripheral Preferred Connection Parameters		0x02	Music		~			
Primary Service Declaration: Generic Attribute					Pictures	Pictures	Subversion			
⊡ 0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)		Subversion	Library	Library			
🗄 Characteristic [	eclaration: S	Service Changed			🛃 Videos	Videor				
⊡ • 0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A			Library				
0x000/	0x2A05	Service Changed		0x22	🐏 Computer	- ~				
Primary Service Declar	ation				File name: Pe	ripheral Attribute list				
	0x2800	Primary Service Declaration	B5:CA		Save as type: CS	V (*.CSV)				
- Characteristic [	Declaration									
⊡ 0x000C	0x2803	Characteristic Declaration	10:0D:00:A1:CA		A Hide Folders		Save Cancel			
	0xCAA1			0x10	- nac i olders					
0x0008	0x2902	Client Characteristic Configuration								

Figure 6-69. Save Attribute List to a File

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

Figure 6-70. Send Commands

CySmart 1.0					
Eile Help					
Select Dongle 🥒 Co	onfigure Ma	ster Settings Manage PSMs			
Master CapSense Sider	and LED [22	43:65:56:34:12]			
Atributes					Attribute Details Send Commands
🕞 Discover All Attribute	es 🚺 Enal	ble All Notifications 🔟 Read All Character	istics 🛷 Bond 🛛 🏭 Export 🛛 🍵 Clear	View: Category 💌 🚺	Commands
Handle	UUID	UUID Description	Value	Properties	GATT     General Service Decovery     Select command
Primary Service Declar	ation: Generic	Access			Discover Al Primary Services
E- 0x0001	0x2800	Primary Service Declaration	00:18 (Generic Access)		Decover Planary Services by DUID
- Characteristic D	Declaration: D	Vevice Name			Oharacteristic Discovery
⊖ 0x0002	0x2803	Ovaracteristic Declaration	02:03:00:00:2A		Oharacteristic Descriptor Discovery
- 0x0003	3 0x2A00	Device Name	43 61 70 53 65 6E 73 65 20 53 6C 69 64 65 72 20 61 6E 64 20 4C 45	0x02	(e) Characteristic Value Head     (e) Characteristic Value Wite
B- Characteristic I	Declaration: A	ppearance			Oraracteristic Descriptor
B- 0x0004	0x2803	Characteristic Declaration	02:05:00:01:2A		GAP
0x0005	5 0x2A01	Appearance	00:00	0x02	
Characteristic 0	Declaration: P	enpheral Preferred Connection Parameters			-E
B- 0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A		
Qx0007	7 Gx2A04	Peripheral Preferred Connection Parameters	06 00 80 0C 00 00 E8 03	0x02	
Primary Service Declara	ation. Generic	: Atribute			Discover All Primary Services Parameters
B-0x0008	Dx0008 Dx2800 Primary Service Declaration 01.18 (Generic Attribute)			Parameters None	
E- Characteristic [	Declaration: S	lervice Changed			
⊖-0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A		
0x0004	A 0x2A05	Service Changed	00:00:00:00	0x22	
Primary Service Declari	ation				
B-0x0008	0x2900	Primary Service Declaration	BGICA		
E Characteristic 0	Declaration	11/32		e	Parameters
- 0x000C	0x2803	Characteristic Declaration	10:00:00:A2:CA		The command does not have any parameters
- Cx0000	Dicaa2		44	0x10	
- 0x0008	0x2902	Client Characteristic Configuration			Şend
Athen day 12CAP Charge	wie l	- 15 -	1		



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

Figure 6-71. Disconnect BLE Device

<b>C</b> (	ySmart 1.0								
<u>F</u> ile	<u>H</u> elp								
😵 Se	lect Dongle 🥒 Configur	e Master Settings	Manage PSM	1s					
Maste	er CapSense Slider and LE	D [22:43:65:56:34:1	2]						
Discovered devices									
🕅 Start Scan 👹 Disconnect 🚍 Add to Whitelist									
# D	evice	Bluetooth Address	Address Type	RSSI	Advertisement Type	Connected			
1 Ca	apSense Slider and LED	22:43:65:56:34:12	Public	-66 dBm	Connectable undirected				

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

Advanced Topics



# A. Appendix



# A.1 Schematics

## A.1.1 BLE Pioneer Board



































# A.1.2 BLE Module





# A.1.3 Dongle









# A.2 Board Layout

## A.2.1 BLE Pioneer Board

Figure A-1. Primary Side of BLE Pioneer Board



Figure A-2. Ground Layer of BLE Pioneer Board



Figure A-3. Power Layer of BLE Pioneer Board









Figure A-5. Primary Silkscreen of BLE Pioneer Board



Figure A-6. Secondary Silkscreen of BLE Pioneer Board





# A.2.2 PRoC BLE Module

Figure A-7. Primary Side of PRoC BLE Module



Figure A-8. Ground Layer of PRoC BLE Module



Figure A-9. VCC Layer of PRoC BLE Module





Figure A-10. Secondary Side of PRoC BLE Module



Figure A-11. Primary Silkscreen of PRoC BLE Module



Figure A-12. Secondary Silkscreen of PRoC BLE Module





# A.2.3 PSoC 4 BLE Module

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



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



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





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

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



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




## A.2.4 Dongle

Figure A-19. Primary Side of Dongle



Figure A-20. Ground Layer of Dongle



Figure A-21. Power Layer of Dongle









Figure A-23. Primary Silkscreen of Dongle



Figure A-24. Secondary Silkscreen of Dongle





# A.3 Bill of Materials (BOM)

#### A.3.1 BLE Pioneer Board

ltem	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
1			-	PCB, 106.22 mm x 53.34 mm, High Tg, ENIG finish, 4 layer, Color = RED, Silk = WHITE.	Cypress				N/A	Yes	N/A
2	1	BT1	CR2032 Battery Holder	HOLDER COIN CELL CR2032 EJECT	MPD	BA2032			ASK	Yes	Yes
3	1	C1	1.0 uF	CAP TANT 1UF 35V 10% 1210	AVX Corporation	TAJB105 K035RNJ			ASK	Yes	Yes
4	1	C2	4.7 uF	CAP TANT 4.7UF 20V 10% 1210	AVX Corporation	TAJB475 K020RNJ			ASK	Yes	Yes
5	1	СЗ	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD TDK Corporation C1005X7 R1C103K 050BA ASK CAP CEP C2225X5		ASK	Yes	Yes			
6	1	C4	100 uFd	CAP CER 100UF 6.3V 20% X5R 1210	TDK Corporation	C3225X5 R0J107M 250AC			ASK	Yes	Yes
7	15	C5, C8, C9, C10, C12, C14, C17, C18, C19, C21, C23, C25, C26, C27, C28	0.1 uFd	CAP .1UF 16V CERAMIC X5R 0402	TDK Corporation	C1005X5 R1A104K 050BA			ASK	Yes	Yes
8	7	C6, C7, C11, C13, C15, C16, C20	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107B J105KA-T			ASK	Yes	Yes
9	1	C29	33 uF	CAP CER 33UF 6.3V 20% X5R 0805	TDK Corporation	C2012X5 R0J336M 125AC					
10	6	D1, D2, D3, D4, D5, D10	MBR052 0L	DIODE SCHOTTKY 0.5A 20V SOD- 123	Fairchild Semiconductor	MBR0520 L			ASK	Yes	Yes
11	3	D6, D7, D8	ESD diode	SUPPRES- SOR ESD 5VDC 0603 SMD	Bourns Inc.	CG0603M LC-05LE			ASK	Yes	Yes
12	2	D9, D11	2V Zener	DIODE ZENER 2V 500MW SOD123	Diodes Inc	BZT52C2 V0-7-F			ASK	Yes	Yes
13	1	F1	FUSE	PTC RESET- TABLE .50A 15V 1812	Bourns	MF- MSMF050 -2			ASK	Yes	Yes



ltem	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
14	2	J1, J4	8x1 RECP	CONN HEADER FEMALE 8POS .1" GOLD	Protectron Electromech	P9401-08- 21	Sullins Con- nector Solu- tions	PPPC081LF BN-RC	ASK	Yes	Yes
15	1	J2	6x2 RECP	CONN HEADER FMAL 12PS.1" DL GOLD	Protectron Electromech	P9403-12- 21	Sullins Con- nector Solu- tions	PPPC062LF BN-RC	ASK	Yes	Yes
16	1	J3	10x1 RECP	CONN HEADER FMALE 10POS .1" GOLD	Protectron Electromech	P9401-10- 21	Sullins Con- nector Solu- tions	PPPC101LF BN-RC	ASK	Yes	Yes
17	1	J8	8X2 RECP	CONN HEADER FMAL 16PS.1" DL GOLD	Protectron Electromech	P9403-16- 21	Sullins Con- nector Solu- tions	PPPC082LF BN-RC	ASK	Yes	Yes
18	1	J10	12X2 RECP	CONN HEADER 2.54MM 24POS GOLD	Sullins Connector Solutions	SBH11- PBPC- D12-ST- BK			ASK	Yes	Yes
19	1	J11	10X2 RECP	CONN HEADER 2.54MM 20POS GOLD	Sullins Connector Solutions	SBH11- PBPC- D10-ST- BK			ASK	Yes	Yes
20	1	J13	USB MINI B	MINI USB RCPT R/A DIP	TE Connectivity	1734510- 1	Molex Inc	0548190519	ASK	Yes	Yes
21	1	J14	1X1 RECP	CONN RCPT 1POS .100" SNGL HORZ	Samtec Inc	BCS-101- L-S-HE			ASK	Yes	Yes
22	1	J15	2p_jump er	CONN HEADR BRKWAY .100 2POS STR	Protectron Electromech	P9101-02- 12-1	TE Connectivity	5-146280-2	ASK	Yes	Yes
23	1	J16	3p_jump er	CONN HEADR BRKWAY .100 3POS STR	Protectron Electromech	P9101-03- 12-1	TE Connectivity	5-146280-3	ASK	Yes	Yes
24	1	LED1	Power LED Amber	LED 595NM AMB DIFF 0805 SMD	Avago Technologies	HSMA- C170			ASK	Yes	Yes
25	1	LED2	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17- 21VGC/ TR8			ASK	Yes	Yes
26	1	LED3	RGB LED	LED RED/ GREEN/BLUE PLCC4 SMD	Cree, Inc.	CLV1A- FKB- CJ1M1F1 BB7R4S3			ASK	Yes	Yes
	3	L1, L2, L3	330 OHM @ 100MHz	FERRITE CHIP 330 OHM 0805	Murata	BLM21PG 331SN1D					
27	6	Q1, Q2, Q3, Q4, Q5, Q6	PMOS	MOSFET P-CH 30V 3.8A SOT23-3	Diodes Inc	DMP3098 L-7			ASK	Yes	Yes
28	1	R1	11K 1%	RES 11K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF1102 V			ASK	Yes	Yes



ltem	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
29	1	R2	560 ohm	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ- 6GEYJ56 1V			ASK	Yes	Yes
30	1	R3	14.7K 1%	RES 14.7K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF147 2V			ASK	Yes	Yes
32	1	R4	10K 1%	RES 10K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF100 2V			ASK	Yes	Yes
31	1	R5	4.3K 1%	RES 4.3K OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF430 1V			ASK	Yes	Yes
33	1	R6	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ- 2GEJ104 X			ASK	Yes	Yes
34	14	R19, R26, R27, R36, R37, R38, R45, R46, R47, R52, R53, R54, R55, R56	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ- 3GEY0R0 0V			ASK	Yes	Yes
35	2	R8, R58	442 ohm 1%	RES 442 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF442 0V			ASK	Yes	Yes
36	2	R9, R20	1К	RES 1K OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ- 6GEYJ10 2V			ASK	Yes	Yes
37	1	R10	10K	RES 10K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ10 3V			ASK	Yes	Yes
38	1	R11	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ- 6GEYJ82 1V			ASK	Yes	Yes
39	2	R13, R14	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic- ECG	ERJ- 6GEY0R0 0V			ASK	Yes	Yes
40	2	R15, R16	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ- 3EKF22R 0V			ASK	Yes	Yes
41	2	R17, R18	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ15 3V			ASK	Yes	Yes
42	5	R22, R23, R28, R31, R35	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ22 2V			ASK	Yes	Yes
43	2	R24, R25	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ30 3V			ASK	Yes	Yes
44	2	R29, R30	1.5K	RES 1.5K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ15 2V			ASK	Yes	Yes



ltem	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
45	5	R39, R40, R41, R42, R43	560 ohm	RES 560 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ56 1V			ASK	Yes	Yes
46	2	SW1, SW2	SW PUSH- BUTTON	SWITCH TAC- TILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ- PE105K			ASK	Yes	Yes
47	1	TP5	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001			ASK	Yes	Yes
48	2	TVS1,TVS2	5V 350W	TVS UNIDIR 350W 5V SOD- 323	Dioded Inc.	SD05-7			ASK	Yes	Yes
49	1	U1	LDO	IC REG LDO ADJ 1A TO252-5	Rohm Semiconductor	BA00BC0 WFP-E2			ASK	Yes	Yes
50	1	U2	PSoC 5LP	68QFN PSoC 5LP chip for USB debug channel and USB-Serial interface	Cypress Semiconductor	CY8C586 8LTI- LP039			ASK	Yes	Yes
51	1	U3	F-RAM	F-RAM 1-Mbit (128K X 8) I2C interface	Cypress Semiconductor	FM24V10- G			ASK	Yes	Yes
52	1	U4	DUAL PMOS	MOSFET 2P- CH 20V 430MA SOT-563	ON Semiconductor	NTZD315 2PT1G			ASK	Yes	Yes
Insta	ll on	Bottom of P	CB As per	the Silk Screer	in the Corners						
53	4	N/A	N/A	BUMPER CYLIN 0.375" DIA BLK	ЗМ	SJ61A4			ASK	Yes	Yes
5 <b>рес</b>	2	J15,J16	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010- E			ASK	Yes	Yes
		1	I		1	1	1		1		
56	1	N/A	N/A	LBL, PCA Label, Vendor Code, Date- code, Serial Number 121- 60158-01 Rev 04 (YYWWV- VXXXX)	Cypress Semiconductor				ASK	Yes	Yes
57	1	N/A	N/A	12mm X 12mm	Semiconductor				ASK	Yes	Yes



ltem	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Alternate Manufacturer	Alternate Mfr Part Number	Sub Allowed	ROHS	UL Certified
No lo	ad co	omponents									
58	1	C22	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5 R1A104K 050BA			ASK	Yes	Yes
59	1	C24	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107B J105KA-T			ASK	Yes	Yes
60	9	R7, R59, R32, R33, R34, R48, R49, R50, R51	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094- 1			ASK	Yes	Yes
61	1	R21	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ- 3GEYJ47 2V			ASK	Yes	Yes
62	2	J7, J6	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH- 105-01-L- DV-K	FCI	20021521- 00010T1LF	ASK	Yes	Yes
63	1	19	2 PIN HDR	CONN HEADER FEMALE 2POS .1" GOLD	Sullins Connector Solutions	PPPC021 LFBN-RC	Protectron Electromech	P9401-02- 21	ASK	Yes	Yes
64	2	TP4, TP5	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001			ASK	Yes	Yes
65	3	TP1, TP2, TP3	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000			ASK	Yes	Yes
66	2	R44, R12	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic- ECG	ERJ- 6GEY0R0 0V			ASK	Yes	Yes
67	1	J12	3x2 RECPT	CONN HEADER FMAL 6PS .1" DL GOLD	Sullins Connector Solutions	PPPC032 LFBN-RC	Protectron Electromech	P9403-06- 21	ASK	Yes	Yes
68	1	J5	6X1 RECP RA	CONN FEMALE 6POS .100" R/A GOLD	Sullins Connector Solutions	PPPC061 LGBN-RC			ASK	Yes	Yes



## A.3.2 BLE Module

#### A.3.2.1 CY5671 PRoC BLE Module

Item	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
1	1	600-60196-01	-	PRoC BLE Module printed circuit board	Cypress qualified vendor	600-60196-01 Rev03	No	Yes	Yes
2	8	C1, C3, C5, C7, C9, C11, C16, C18	0.1 uF	CAP .1UF 16V CERAMIC Y5V 0402	Samsung Electro-Mechanics America, Inc	CL05F104ZO5NNNC	ASK	Yes	Yes
3	10	C2, C4, C6, C8, C10, C12, C15, C17, C19, C20	1.0 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	TDK Corporation	C1608X5R1E105K080AC	ASK	Yes	Yes
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222JA01D	ASK	Yes	Yes
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103JA01D	ASK	Yes	Yes
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D	ASK	Yes	Yes
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D	ASK	Yes	Yes
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T	ASK	Yes	Yes
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC-D12-ST-BK	ASK	Yes	Yes
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC-D10-ST-BK	ASK	Yes	Yes
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technology Inc	L-07C6N8JV6T	ASK	Yes	Yes
12	3	L2, L3, L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1D	ASK	Yes	Yes
13	1	U1	PRoC BLE	56 QFN PRoC BLE	Cypress Semiconductor	CYBL10563-56LQXI	ASK	Yes	Yes
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS327-12.5-34B	ASK	Yes	Yes
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM	ASK	Yes	Yes
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121- 60160-01 Rev 04 (YYWWVVXXXX)	Cypress qualified vendor	-	ASK	Yes	Yes
No Lo	ad co	mponents		I	I	I			
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Technology Inc	500R07S1R2BV4T	ASK	Yes	Yes
18	1	C25	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU	ASK	Yes	Yes



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



### A.3.2.2 CY8CKIT-142 PSoC 4 BLE Module

ltem	Qty	Reference	Value	Description	Mfr_Name	Mfr_Part_Number	Sub Allowed	ROHS	UL Certified
1	1	600-60195-01	-	PSoC 4 BLE Module printed circuit board	Cypress qualified vendor	600-60195-01 Rev03	No	Yes	Yes
2	8	C1, C3, C5, C7, C9, C11, C16, C18	0.1 uF	CAP .1UF 16V CERAMIC Y5V 0402	Samsung Electro-Mechanics America, Inc	CL05F104ZO5NN NC	ASK	Yes	Yes
3	10	C2, C4, C6, C8, C10, C12, C15, C17, C19, C20	1.0 uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	TDK Corporation	C1608X5R1E105K 080AC	ASK	Yes	Yes
4	1	C21	2200 pF	CAP CER 2200PF 50V 5% NP0 0805	Murata Electronics	GRM2165C1H222 JA01D	ASK	Yes	Yes
5	1	C22	10000 pF	CAP CER 10000PF 50V 5% NP0 0805	Murata Electronics	GRM2195C1H103 JA01D	ASK	Yes	Yes
6	1	C23	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360 JA01D	ASK	Yes	Yes
7	1	C24	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180 FA01D	ASK	Yes	Yes
8	1	C14	1.5 pF	CAP CER 1.5PF 50V NP0 0402	Johanson Technology Inc	500R07S1R5BV4T	ASK	Yes	Yes
9	1	J1	HEADER 24	CONN HEADR FMALE 24POS .1" DL AU	Sullins Connector	SFH11-PBPC- D12-ST-BK	ASK	Yes	Yes
10	1	J2	HEADER 20	CONN HEADR FMALE 20POS .1" DL AU	Sullins Connector	SFH11-PBPC- D10-ST-BK	ASK	Yes	Yes
11	1	L1	6.8nH	CER INDUCTOR 6.8NH 0402	Johanson Technol- ogy Inc	L-07C6N8JV6T	ASK	Yes	Yes
12	3	L2, L3, L4	330 Ohm @100 MHz	FERRITE CHIP 330 OHM 0805	Murata Electronics	BLM21PG331SN1 D	ASK	Yes	Yes
13	1	U1	PSoC 4BLE	56 QFN PSoC 4 BLE	Cypress Semiconductor	CY8C4247LQI- BL483	ASK	Yes	Yes
14	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS327-12.5- 34B	ASK	Yes	Yes
15	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8- 36CKM	ASK	Yes	Yes
16	1	LBL	-	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121- 60159-01 Rev 04 (YYWWVVXXXXX)	Cypress qualified vendor	-	ASK	Yes	Yes
Noto	ad co	mnonente							
17	1	C13	1.2 pF	CAP CER 1.2PF 50V NP0	Johanson	500R07S1R2BV4T	ASK	Yes	Yes
		005	100 F	0402 CAP CER 100PF 50V	Technology Inc	C0603C101K5RA			
18	1	C25	100pF	10% X7R 0603	Kemet	СТU	ASK	Yes	Yes
19	1	R1	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1	ASK	Yes	Yes



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



# A.3.3 Dongle

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



Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Sub Allowed	ROHS	UL Certified
20	9	R1, R2, R3, R4, R12, R13, R14, R15, R26	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V	ASK	Yes	Yes
21	2	R17, R18	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V	ASK	Yes	Yes
22	1	R21	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X	ASK	Yes	Yes
23	2	R19, R20	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V	ASK	Yes	Yes
24	2	R23, R24	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V	ASK	Yes	Yes
25	2	SW1, SW2	SW RA PUSH	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-P3401P	ASK	Yes	Yes
26	1	TVS1	5V 350W	TVS UNIDIR 350W 5V SOD-323	Diodes Inc.	SD05-7	ASK	Yes	Yes
27	1	U1	PRoC BLE	PRoC BLE, Programmable Radio on Chip, 56QFN	Cypress Semiconductor	CYBL10162-56LQXI	ASK	Yes	Yes
28	1	U2	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G	ASK	Yes	Yes
29	1	U3	PSoC 5LP	PSoC 5LP Programmable System on Chip, 68QFN	Cypress Semiconductor	CY8C5868LTI-LP039	ASK	Yes	Yes
30	1	Y1	32.768K Hz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS327-12.5-34B	ASK	Yes	Yes
31	1	Y2	24MHz	CRYSTAL 24.000 MHZ 8PF SMD	ECS Inc	ECS-240-8-36CKM	ASK	Yes	Yes
32	1	N/A	N/A	LBL, PCA Label, Vendor Code, Datecode, Serial Number 121-60161-01 Rev 03 (YYWWVVXXXXX); Only barcode	Cypress qualified vendor	-	ASK	Yes	N/A
<b>NO IO2</b> 33	1	C20	1.2 pF	CAP CER 1.2PF 50V NP0 0402	Johanson Tech- nology Inc	500R07S1R2BV4T	ASK	Yes	Yes
34	1	C21	100pF	CAP CER 100PF 50V 10% X7R 0603	Kemet	C0603C101K5RACTU	ASK	Yes	Yes
35	1	C37	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corpora- tion	C1005X5R1A104K050BA	ASK	Yes	Yes



Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number	Sub Allowed	ROHS	UL Certified
36	1	C27	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	ТМК107ВЈ105КА-Т	ASK	Yes	Yes
37	1	R5	Zero Ohm	RES 0.0 OHM 1/10W JUMP 0603	TE Connectivity	1623094-1	ASK	Yes	Yes
38	2	R6, R16	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V	ASK	Yes	Yes
39	15	TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15	No load	No load	-	-	ASK	Yes	Yes

# **Revision History**



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

1	Documen	cument Title: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit Guide			
	Document Number: 001-93731				
	Revision	Issue Date	Origin of Change	Description of Change	
1	**	11/09/2014	ROIT	New kit guide	

Index

