

Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



CY8CPROTO-063-BLE

PSoC 6 BLE Prototyping Board Guide

Doc. # 002-24993 Rev. *A

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

Copyrights

© Cypress Semiconductor Corporation, 2018–2020. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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

Contents



Safety and Regulatory Compliance Information	5
1. Introduction	7
1.1 Kit Contents	8
1.2 PSoC Creator	9
1.2.1 PSoC Creator Code Examples	10
1.2.2 PSoC Creator Help	11
1.2.3 Component Datasheets	11
1.3 Getting Started.....	12
1.4 Additional Learning Resources	12
1.5 Technical Support.....	13
1.6 Documentation Conventions.....	13
1.7 Acronyms	13
2. Software Installation	15
2.1 Before You Begin.....	15
2.2 Install Software	15
2.3 Uninstall Software	16
3. Kit Operation	17
3.1 Theory of Operation.....	17
3.2 Programming and Debugging the Target PSoC 6 MCU	19
3.2.1 Programming using PSoC Creator.....	19
3.2.2 Debugging using PSoC Creator.....	20
3.2.3 Programming using PSoC Programmer.....	20
3.3 USB-UART Bridge	21
3.4 USB-I2C Bridge	21
3.5 Updating the KitProg2 Firmware.....	21
4. Code Examples	22
4.1 Using Built-in PSoC Creator Code Examples with the Kit	22

A. Appendix	26
A.1 Board Details	26
A.2 Hardware Details	28
A.2.1 Target Board	28
A.2.2 KitProg2 Board.....	29
A.2.3 Power Supply System	30
A.2.4 Board Separation (Snapping).....	31
A.2.5 Header Connections	32
A.2.6 PSoC 6 BLE Prototyping Board features	36
A.3 Programming PSoC 6 BLE Prototyping Board using MiniProg3/KitProg2.....	40
 Revision History	 41

Safety and Regulatory Compliance Information



The CY8CPROTO-063-BLE PSoC® 6 BLE Prototyping Board is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. This may cause interference to other electrical or electronic devices in close proximity. In a domestic environment, this product may cause radio interference. In such cases, you may be required to take adequate preventive measures. In addition, this board should not be used near any medical equipment or RF devices.

Attaching additional wiring to this product or modifying the product operation from the factory default settings 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 CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board Kit boards in the protective shipping package.



End-of-Life/Product Recycling

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

General Safety Instructions

ESD Protection

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

Handling Boards

The CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board is sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad if available. Do not slide board over any surface.

1. Introduction



Thank you for your interest in the CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board.

The CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board is designed as an easy-to-use and inexpensive prototyping platform. The CY8CPROTO-063-BLE Prototyping Board includes the CYBLE-416045-02 EZ-BLE™ Creator Module, delivering a complete system solution for a wide range of embedded applications at a very low cost. PSoC 6 MCU is a programmable embedded system-on chip, integrating custom digital peripheral functions, memory, and both an Arm® Cortex®-M0+ CPU and Cortex-M4 CPU on a single chip with flexible automatic routing. Programmable digital peripheral functions allow higher flexibility, in-field tuning of the design, and faster time-to-market. It is a combination of MCU with standard communication and timing peripherals, and a capacitive touch sensing system with best-in-class performance.

The CY8CPROTO-063-BLE PSoC BLE Prototyping Board offers an open footprint breakout board to maximize the end utility of the PSoC 6 MCU with Bluetooth Low Energy Connectivity (PSoC 63 Line) device. This kit provides a low-cost alternative to device samples while providing a platform to easily develop and integrate the PSoC 6 MCU into your end-system. In addition, the board includes the following features:

- Two LEDs to provide feedback
- A push button to provide a simple user input
- Footprint for external ECO (2.0 mm × 1.6 mm, 24 MHz)
- External WCO (32.768 kHz)
- 1.8 V to 3.3 V operation

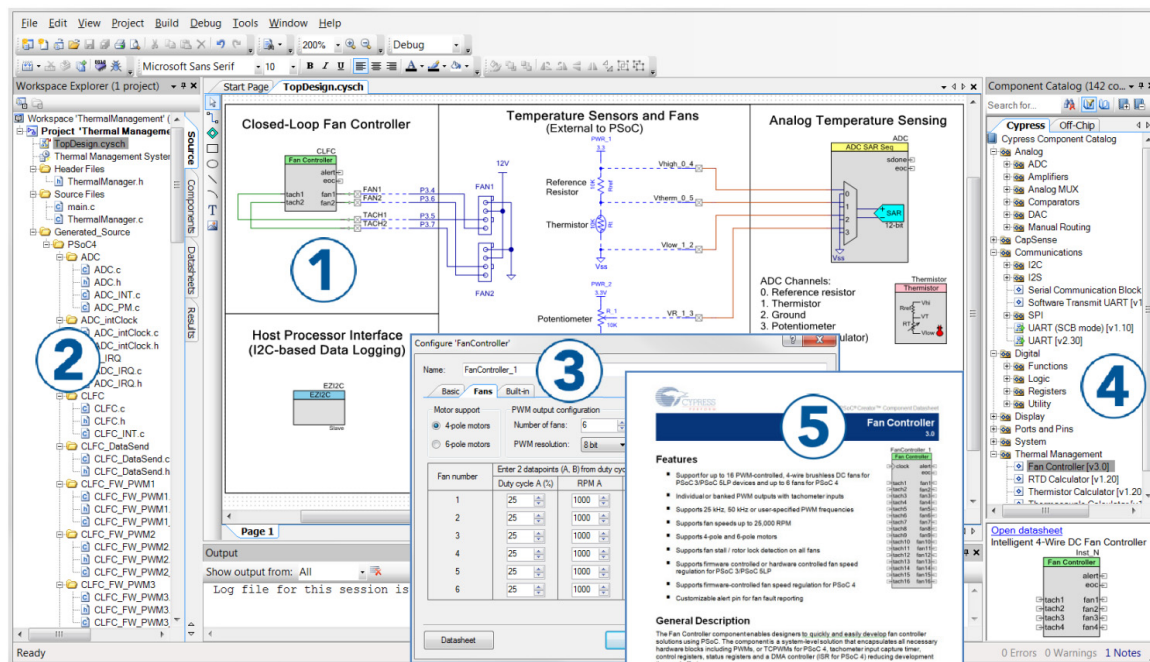
The CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board also integrates Cypress KitProg2 that enables onboard programming, debugging, and bridging functionalities, such as USB-UART and USB-I2C. KitProg2 is used to program and debug the target PSoC 6 MCU which is part of the BLE Module (see [Figure 1-1](#)). The Prototyping board allows you to separate the KitProg2 board from the target PSoC 6 BLE board.

1.2 PSoC Creator

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

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

Figure 1-3. PSoC Creator Features



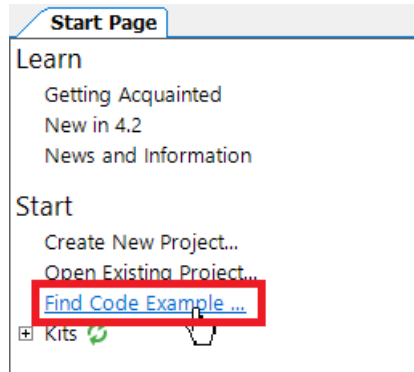
PSoC Creator also enables you to tap into an entire tool ecosystem with integrated compiler chains and programmers for PSoC devices.

For more information, visit www.cypress.com/psoccreator. Visit the [PSoC Creator Video Training Page](#) for video tutorials on learning and using PSoC Creator.

1.2.1 PSoC Creator Code Examples

PSoC Creator includes a large number of code examples. These examples are available from the PSoC Creator Start Page, as shown in [Figure 1-4](#) or from the menu **File > Code Example**.

Figure 1-4. Code Examples in PSoC Creator

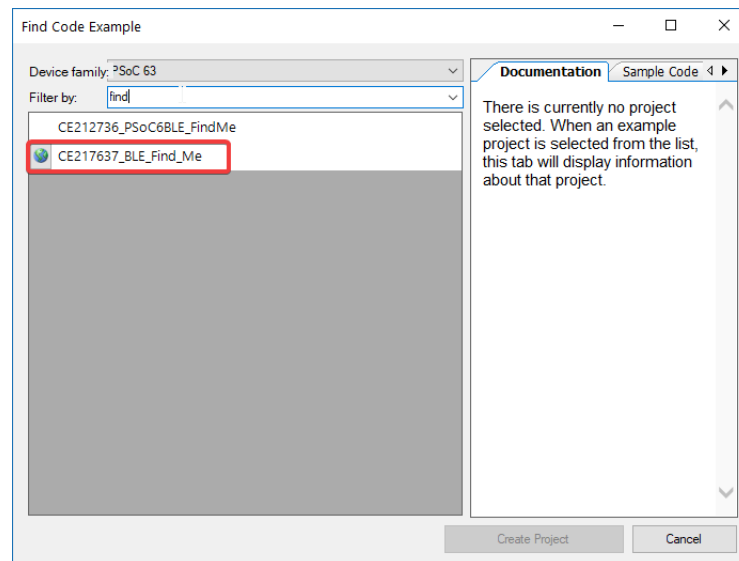


Code examples can speed up your design process by starting you off with a complete design, instead of a blank page. The code examples also show how PSoC Creator Components can be used for various applications.

From the **Find Code Example** dialog shown in [Figure 1-5](#), you can

- Filter for examples based on architecture or device family, that is, PSoC 3, PSoC 4, PSoC 5LP, PSoC 63, or keyword.
- Select from the list of examples offered based on the **Filter Options**.
- Review the code example's description (on the **Documentation** tab).
- Review the code from the **Sample Code** tab. You can copy the code from this window and paste to your project, which can help speed up code development.
- Create a new project (and a new workspace if desired) based on the selection. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 1-5. Code Example Projects with Sample Code



1.2.2 PSoC Creator Help

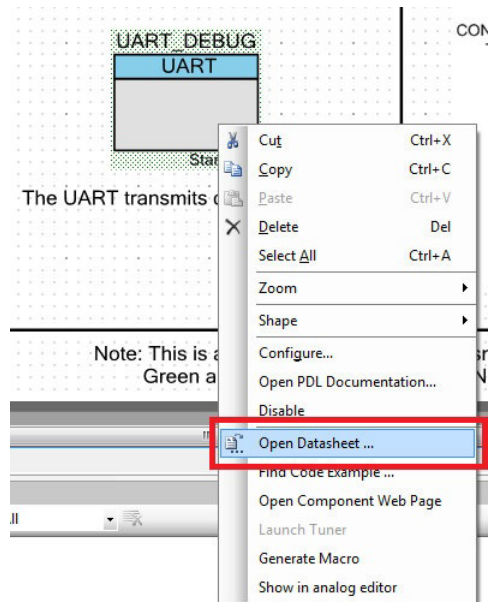
Launch PSoC Creator and navigate to the following items:

- **Quick Start Guide:** Choose **Help > Documentation > Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple Component Code Examples:** Choose **File > Code Example**. These examples demonstrate how to configure and use PSoC Creator Components. To access examples related to a specific Component, right-click the Component in the schematic or in the Component Catalog. Select **Find Code Example** in the context menu that appears.
- **System Reference Guide:** Choose **Help > System Reference Guides**. This guide lists and describes the system functions provided by PSoC Creator.

1.2.3 Component Datasheets

Right-click a Component and select Open Datasheet (see [Figure 1-6](#)).

Figure 1-6. Opening Component Datasheet



1.3 Getting Started

This guide will help you get acquainted with the CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board:

- The [Software Installation chapter on page 15](#) describes the installation of the required software to operate this kit. This includes the installation of the PSoC Creator IDE, Peripheral Driver Library (PDL) and CySmart for development and debugging the applications, and PSoC Programmer for programming the .hex files.
- The [Kit Operation chapter on page 17](#) describes the major features of the CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board and functionalities such as programming, debugging, and the USB-UART and USB-I2C bridges.
- The [Code Examples chapter on page 22](#) describes how to easily adapt a code example that was originally developed for PSoC 6 BLE Pioneer Kit (CY8CKIT-062-BLE).
- The [Appendix on page 26](#) describes the hardware content of the kit and the hardware operation, details about programming the kit using MiniProg3, Kit Schematics, and the bill of materials (BOM).

1.4 Additional Learning Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see [How to Design with PSoC 6 MCU - KBA223067](#). The following is an abbreviated list for PSoC 6 MCU:

- Overview: [PSoC Portfolio](#) and [PSoC Roadmap](#).
- Product Selectors: [PSoC 6 MCU Product Selector](#). In addition, [PSoC Creator](#) includes a device selection tool.
- Datasheets: Describe and provide electrical specifications for the [PSoC 61](#), [PSoC 62](#), [PSoC 63](#) devices and [CYBLE-416045-02 EZ-BLE Creator Module](#).
- [CapSense Design Guide](#): Learn how to design capacitive touch-sensing applications with the PSoC 6 MCU and PSoC 4 family of devices.
- [Application Notes](#) and [Code Examples](#): Cover a broad range of topics, from basic to advanced. Many of the application notes include code examples. Visit the [PSoC 6 MCU Code Examples web page](#) for a list of all available PSoC Creator code examples. To access code examples from within PSoC Creator, see [PSoC Creator Code Examples on page 10](#).
- [Technical Reference Manuals \(TRM\)](#): Provide detailed descriptions of the architecture and registers in each PSoC 6 MCU family.
- [Development Kits](#):
 - [PSoC 6 BLE Pioneer Kit \(CY8CKIT-062-BLE\)](#) and [PSoC 6 WiFi-BT Pioneer Kit \(CY8CKIT-062-WiFi-BT\)](#) are easy-to use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent Pmod Peripheral Modules.
 - [PSoC 6 BLE Prototyping Kit \(CY8CPROTO-063-BLE\)](#) are very low-cost prototyping platform for sampling PSoC 6 MCU.
 - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
 - The [MiniProg3](#) device provides an interface for flash programming and debug.
Note: MiniProg3 is not required to use kits such as CY8CPROTO-063-BLE that have KitProg2 provided on-board.
- [Knowledge Base Articles \(KBA\)](#): Provide design and application tips from experts on using the device.

- PSoC Creator Training: Visit www.cypress.com/go/creatorstart/creatortraining for a comprehensive list of video training on PSoC Creator.
- Learning from Peers: Visit community.cypress.com/welcome to meet enthusiastic PSoC developers discussing the next-generation embedded systems on Cypress Developer Community Forums.

1.5 Technical Support

For assistance, visit [Cypress Support](#).

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

- [Self-help \(Technical Documents\)](#)
- [Local Sales Office Locations](#)

1.6 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays source code: C:\...cd\icc\
<i>Italics</i>	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Creator User Guide</i> .
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the File icon and then click Open .
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes cautions or unique functionality of the product.

1.7 Acronyms

Table 1-2. Acronyms Used in this Document

Acronym	Definition
ADC	Analog-to-Digital Converter
BCP	Bridge Control Panel
BLE	Bluetooth Low Energy
BOM	Bill of Materials
CE	Code Example
C _{MOD}	Modulator Capacitor
DAC	Digital to Analog Converter
ECO	External Crystal Oscillator
ESD	Electrostatic Discharge
FLL	Frequency Locked Loop

Table 1-2. Acronyms Used in this Document (*continued*)

Acronym	Definition
GPIO	General-Purpose Input/Output
I ² C	Inter-Integrated Circuit
IDAC	Current DAC
IDE	Integrated Design Environment
IMO	Internal Main Oscillator
KBA	Knowledge Based Article
LED	Light-Emitting Diode
PLL	Phase Locked Loop
PSoC	Programmable System-on-Chip
SAR	Successive Approximation Register
SCB	Serial Communication Block
SRAM	Serial Random Access Memory
SWD	Serial Wire Debug
TCPWM	Timer, Counter, Pulse Width Modulator
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WCO	Watch Crystal Oscillator

2. Software Installation



This chapter describes the steps to install the software tools and packages on a PC for using the CY8CPROTO-063-BLE Prototyping Board. This includes the IDE on which the projects will be built and used for programming.

2.1 Before You Begin

To install Cypress software, you will require administrator privileges. However, they are not required to run the software that is already installed. 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 required software to operate this kit:

1. Download and install the following if it is not present on your PC.
 - a. PSoC Creator 4.2 or later: This software is available for download at www.cypress.com/psoccreator.
 - b. Peripheral Driver Library (PDL) 3.0.4 or later: Peripheral Driver Library is available for download at www.cypress.com/pdl.
 - c. PSoC Programmer 3.27.3 or later: This software is available for download at (www.cypress.com/programmer).
 - d. CySmart 1.3 or later: This software is available for download at www.cypress.com/cysmart.
2. Choose the **Typical**, **Custom**, or **Complete** installation type in the Product Installation Overview window. Click **Next** after selecting the installation type.

Note: Typical installation is recommended, however you can choose according to your requirement.
3. Read the License agreement and select **I accept the terms in the license agreement** to continue with the installation. Click **Next**. 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.
4. Enter your contact information or select the check box **Continue Without Contact Information**.
5. Click **Finish** to complete the installation.
6. Please visit [EZ-BLE Creator Module Support page](#) and follow the instructions to install support for the PSoC 63-based EZ-BLE Creator Module.

2.3 Uninstall Software

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

1. Go to **Start > All Programs > Cypress > Cypress Update Manager** and select **Uninstall**.
2. Go to **Start > Control Panel > Programs and Features for Windows 7** or **Add/Remove Programs for Windows XP** and select **Uninstall/Change**.

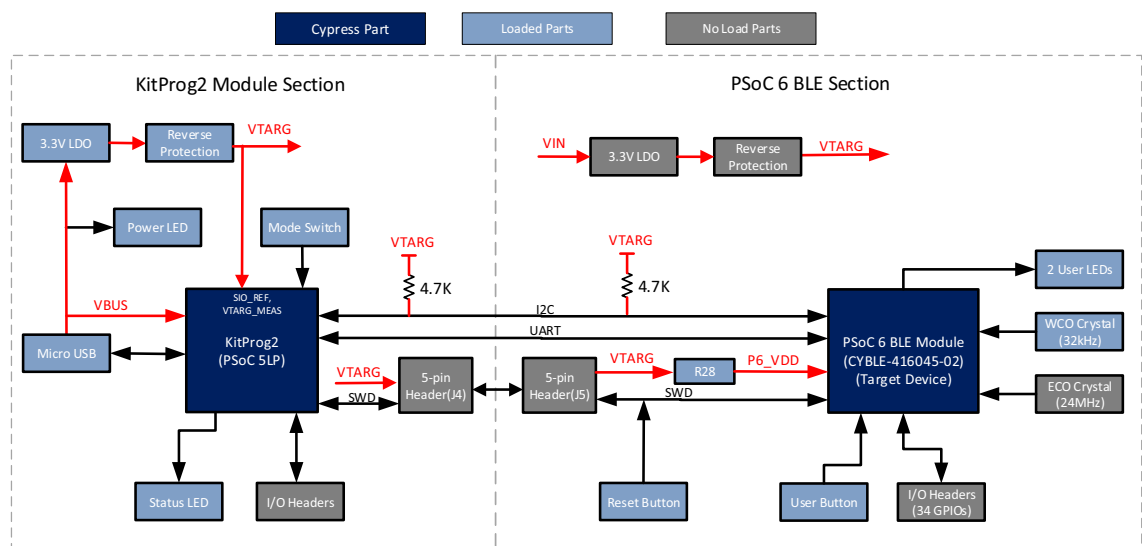
3. Kit Operation



This chapter introduces you to different features of the CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Kit. This primarily includes the programming and debugging functionalities, KitProg2 USB-UART and USB-I2C bridges, and the procedure to update the KitProg2 firmware.

3.1 Theory of Operation

Figure 3-1. Block Diagram of CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board



The CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Kit is simplistic in design and focuses on providing you with complete access to develop applications using the PSoC 6 MCU. This kit supports the following features:

- **PSoC 6 BLE I/O Headers:** The Prototyping Board brings all GPIOs of the target CYBLE-416045-02 module to the two expansion headers, except the pins used for the WCO crystal. Although this is a subset of the actual PSoC 6 MCU, it is sufficient to evaluate most of the features.
- **User LEDs:** The onboard LEDs, (LED3 and LED4) can be used to display outputs from the PSoC 6 BLE device. This includes modulating the brightness of the LED to notify different states of the device. You may interface other LEDs as well by connecting to the expansion headers directly or by connecting them to a breadboard.

- **User Button:** This kit has a push button, which can be used to provide an input to the PSoC 6 MCU.
Note: The button connects the PSoC 6 MCU pin (P0[4]) to ground when pressed. Therefore, you need to configure the PSoC 6 MCU pin as 'resistive pull-up' for detecting the button press.
- **EZ-BLE Module:** The EZ-BLE Module (CYBLE-416045-02) from Cypress is used to demonstrate how easily an end product can be created with BLE. The EZ-BLE module contains the CY8C6347BZI-BLD53 PSoC 6 MCU, crystal used for the BLE radio, and an onboard antenna. Most code examples that work for the PSoC 6 MCU will work with the EZ-BLE module, depending only on the GPIO requirements.
- **ECO Crystal footprint:** A footprint for an external crystal is provided so that you may attach a crystal with your choice of frequency.
 The recommended crystal is [ECS-240-8-37CKM](#) from ECS Inc., which provides 24 MHz at ±50 ppm accuracy. Any crystal with compatible footprint (2.0 mm × 1.6 mm) can be used.
- **WCO Crystal:** A 32.768-kHz crystal is provided and is used for BLE applications to provide accurate timing including when the device is in low power mode.
- **Reset Button:** This button is used to reset the PSoC 6 MCU when pressed.
- **Power LED:** This LED indicates power is being supplied to the board.
- **5-pin Programming Headers:** These are programming headers that can be used to program the EZ-BLE Creator Module using a MiniProg3 programmer/debugger. Typically, this connection is not required because the kit contains an integrated KitProg2 for programming and debugging.
- **PSoC 5LP I/O Headers:** Using these headers, you can evaluate the PSoC 5LP (CY8C5868LTI-LP039) device. The PSoC 5LP device is used to implement the KitProg2 functionality by default.
- **Current Measurement Jumper:** Using this jumper, you can measure the current consumed by the EZ-BLE Creator Module. This jumper is not populated by default. To use this jumper, you must remove the zero-ohm resistor R28.
- **USB 2.0 Micro-B Connector:** This connector is used to connect the board to the PC with the provided USB cable, enable to program and/or debug the device. It is used to provide power to the device.
- **KitProg2:** This is an onboard programmer/debugger utilizing a PSoC 5LP device (CY8C5868LTI-LP039), which enables you to program and debug the target PSoC 6 MCU. It can also act as a USB-UART and USB-I2C bridge.
 When used as a standalone module, KitProg2 can be used to program devices of PSoC 3, PSoC 4, PSoC 5LP, or PSoC 6 families through the SWD interface. This version of the KitProg2 supplies 3.3 volts to the PSoC 6 MCU section of the board. For more details on the KitProg2 functionality, see the [KitProg2 User Guide](#).
- **KitProg2 Mode Selection Button:** This button is used to switch between KitProg2 modes as shown in [Table 3-1](#). You can also use this button to provide an input to the PSoC 5LP device in custom applications. Note that the button connects the PSoC 5LP pin to ground when pressed. For more details, see the [KitProg2 User Guide](#).
- **KitProg2 Status LED:** This LED gives the programming/mode status of KitProg2.

Table 3-1. KitProg2 Mode Switching

KitProg2 Programming Modes *	Status LED (LED2)
KitProg2 Program/Debug mode (default)	ON
CMSIS-DAP Programming mode	OFF

* **Note:** Toggling between programming modes can be done by pressing mode switch SW3.

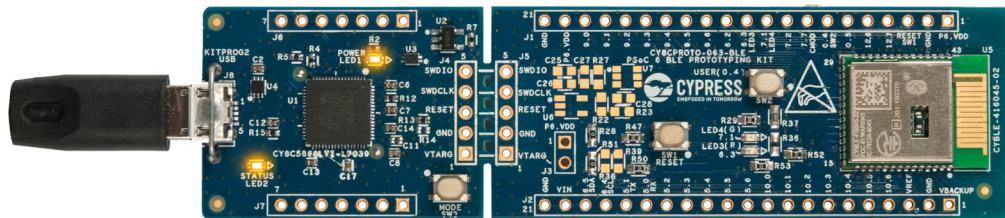
3.2 Programming and Debugging the Target PSoC 6 MCU

The target PSoC 6 MCU can be programmed and debugged using KitProg2. Before programming the device, make sure that PSoC Creator and PSoC Programmer software are installed on the PC. See [Install Software chapter on page 15](#) for more information.

3.2.1 Programming using PSoC Creator

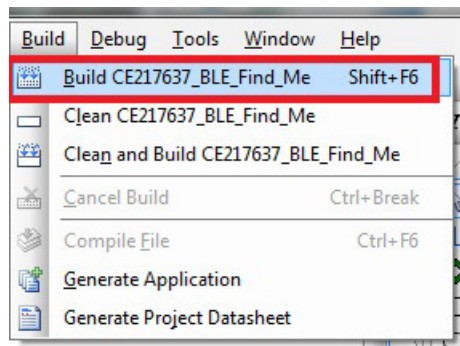
1. Connect the Prototyping Board to the PC through the USB Micro-B Connector (J8) as shown in [Figure 3-2](#). The kit enumerates as a composite device. If you are plugging in the PSoC 6 BLE Prototyping Board to your PC for the first time, the kit drivers will get installed automatically. The amber status LED (LED2) will turn ON to indicate successful enumeration. If the amber status LED (LED2) is OFF, see the [KitProg2 User Guide](#) for details on the KitProg2 status and troubleshooting instructions.

Figure 3-2. Connecting the CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board to a PC



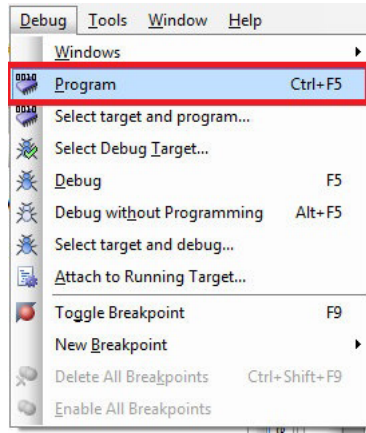
2. Open the desired project in PSoC Creator. From the PSoC Creator, go to **File > Open > Project/Workspace**. You can browse and open a previously saved project. If you want to open one of the code examples compatible with this kit, follow the instructions in the [Code Examples chapter on page 22](#).
3. Select the option **Build > Build <Project_Name>** or press **Shift + F6** to build the project as shown in [Figure 3-3](#).

Figure 3-3. Build a Code Example




- If there are no errors during build, program the firmware onto the kit by choosing **Debug > Program** or press **Ctrl + F5**, as shown in [Figure 3-4](#). This programs the target PSoC 6 MCU on the Prototyping Board.

Figure 3-4. Program Device in PSoC Creator



3.2.2 Debugging using PSoC Creator

For debugging the project using PSoC Creator, follow steps 1–3 from [Programming using PSoC Creator on page 19](#) followed by:

- Click the **Debug** icon  or press **F5**. Alternatively, you can use the option **Debug > Debug**.
- Once PSoC Creator opens in Debug mode, use the buttons on the toolbar to debug your application.

For more details on using the debug features, see the **Debugging using PSoC Programmer** section of the [KitProg2 User Guide](#).

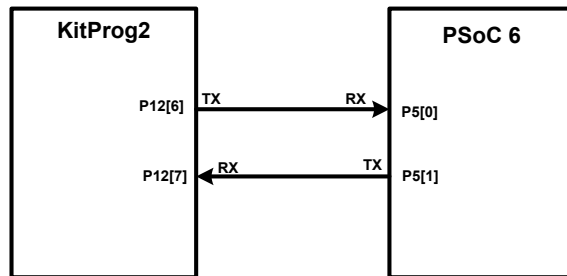
3.2.3 Programming using PSoC Programmer

PSoC Programmer (3.27.3 or later) can be used to program existing `.hex` files into the PSoC 6 BLE Prototyping Board. For a detailed explanation on how to program using PSoC Programmer, see the **Programming using PSoC Programmer** section in the [KitProg2 User Guide](#).

3.3 USB-UART Bridge

KitProg2 on the PSoC 6 BLE Prototyping Board can act as a USB-UART bridge. The UART lines between KitProg2 and the target are hard-wired on the board, through the snappable area, with UART_RX assigned to **P5[0]** and UART_TX assigned to **P5[1]** on target PSoC 6 MCU. For more details on the KitProg2 USB-UART functionality, see the **USB-UART Bridge** section in the [KitProg2 User Guide](#).

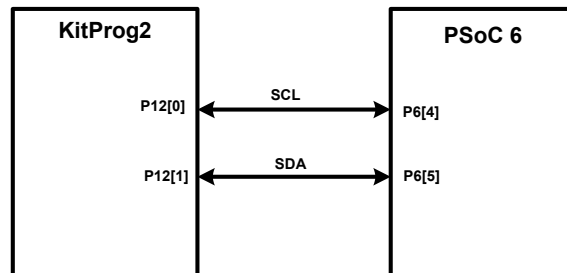
Figure 3-5. UART Connection between KitProg2 and EZ-BLE Creator Module



3.4 USB-I2C Bridge

KitProg2 can function as a USB-I2C bridge and communicate with the software utility Bridge Control Panel (BCP). The I2C lines on target EZ-BLE Creator Module are **P6[5] (SDA)** and **P6[4] (SCL)**, which are hardwired on the board to the I2C lines of KitProg2. The USB-I2C bridge supports I2C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg2 USB-I2C functionality, see the **USB-I2C Bridge** section in the [KitProg2 User Guide](#).

Figure 3-6. I2C Connection between KitProg2 and EZ-BLE Creator Module



3.5 Updating the KitProg2 Firmware

The KitProg2 firmware normally does not require any update. You can use the PSoC Programmer software to update the KitProg2 firmware if necessary. For a detailed explanation on how to update the KitProg2 firmware, see **Updating the KitProg2 Firmware** in the [KitProg2 User Guide](#).

4. Code Examples



This chapter explains the code example to be adapted to the PSoC 6 Prototyping Board. Most existing PSoC 6 MCU code examples only require the device to be changed and GPIOs to be reassigned to match the board.

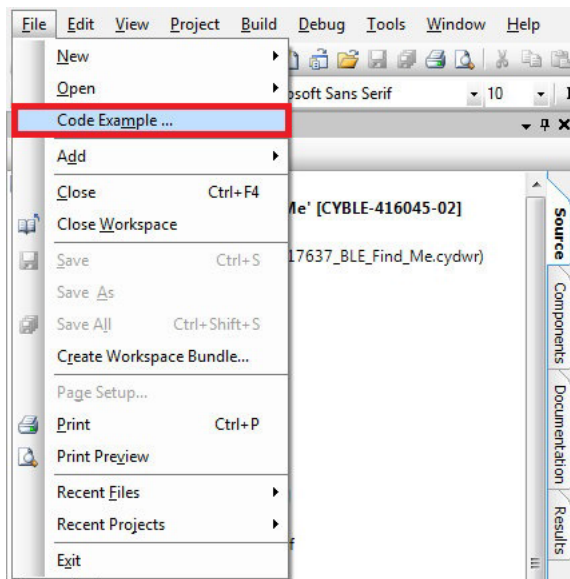
If you have an existing installation of PSoC Creator, please make sure that you install the EZ-BLE module device support by following the instructions provided in [EZ-BLE Creator Module Support page](#).

4.1 Using Built-in PSoC Creator Code Examples with the Kit

Follow these steps to open and use the built-in PSoC Creator examples.

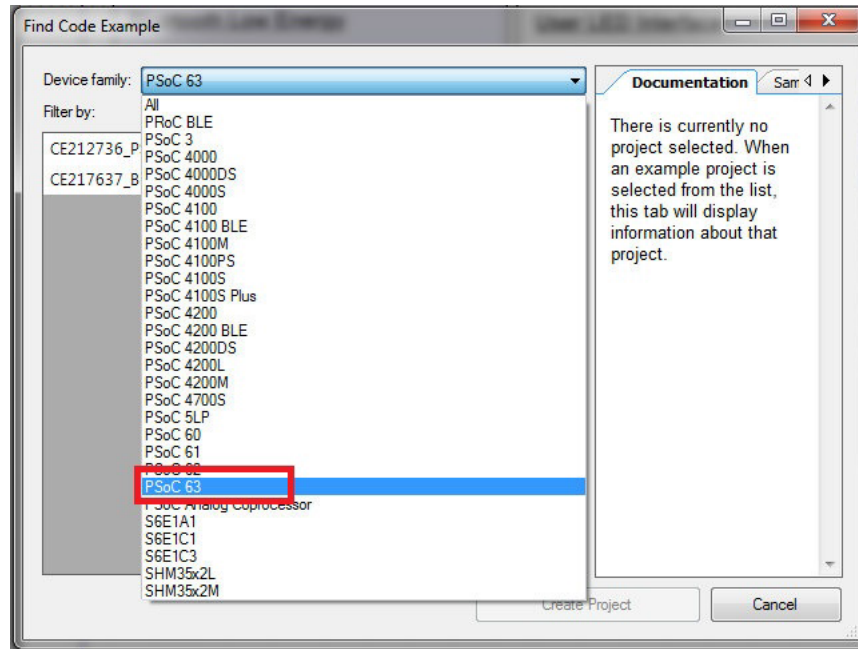
1. Launch PSoC Creator from **Start > All Programs > Cypress > PSoC Creator <version> > PSoC Creator <version>**.
2. On the Start page, click **Find Code Example...** under **Start** or follow the menu **File > Code Example...** as shown in [Figure 4-1](#).

Figure 4-1. PSoC Creator Find Code Example

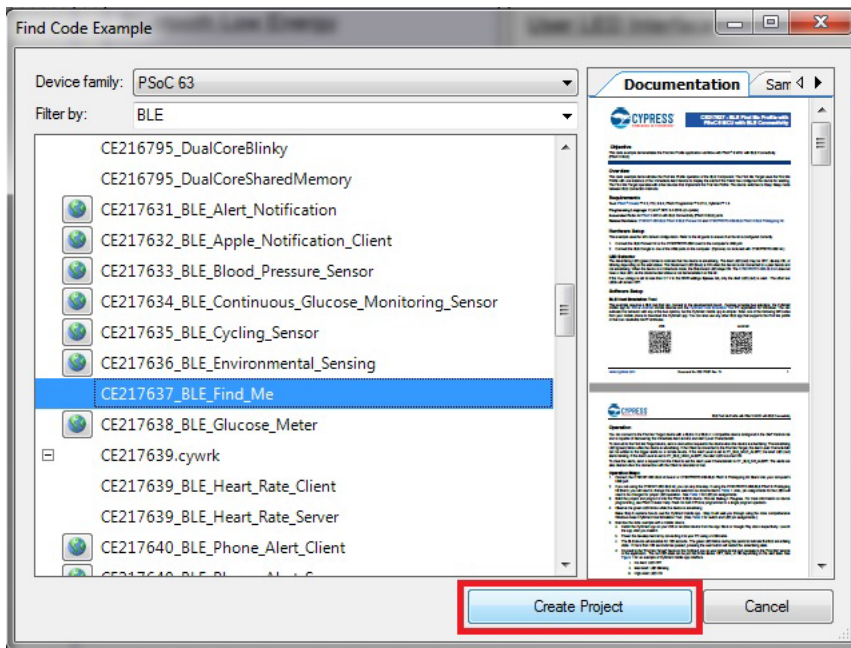


3. In the Find Code Example window, set the **Device family** to **PSoC 63** as shown in Figure 4-2.

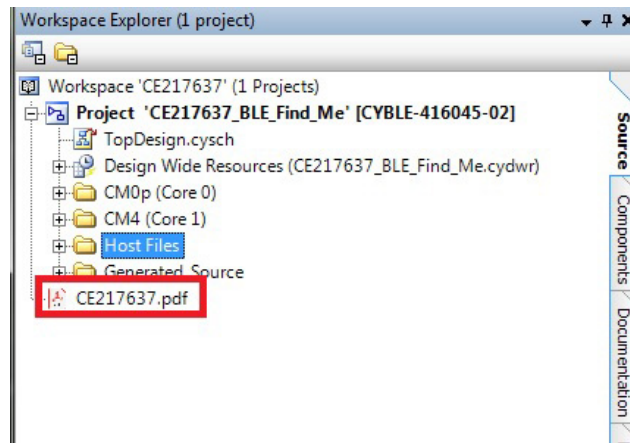
Figure 4-2. Selecting PSoC 63 Device in Find Code Example Window



4. You can select the *CE217637_BLE_Find_Me* project and see how to use it with the CY8CPROTO-063-BLE Prototyping Board.
5. Select the *CE217637_BLE_Find_Me* project and click **Create Project** as shown in Figure 4-3. Save the workspace to a desired location.

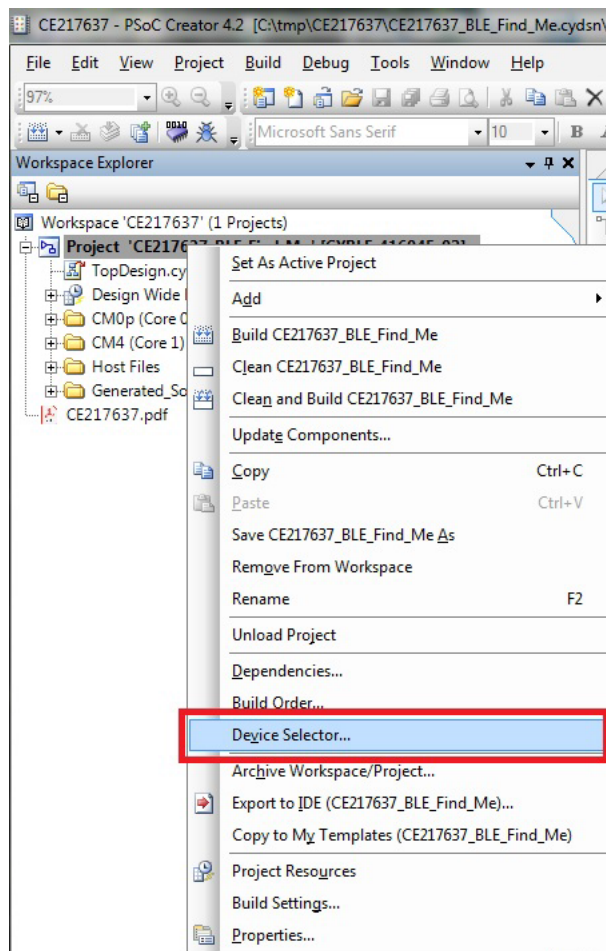
 Figure 4-3. *CE217637_BLE_Find_Me* Code Example


- Open *CE217637.pdf* from the Workspace Explorer to learn more about the code example and its configuration. See [Figure 4-4](#).

 Figure 4-4. Project Datasheet: *CE217637*


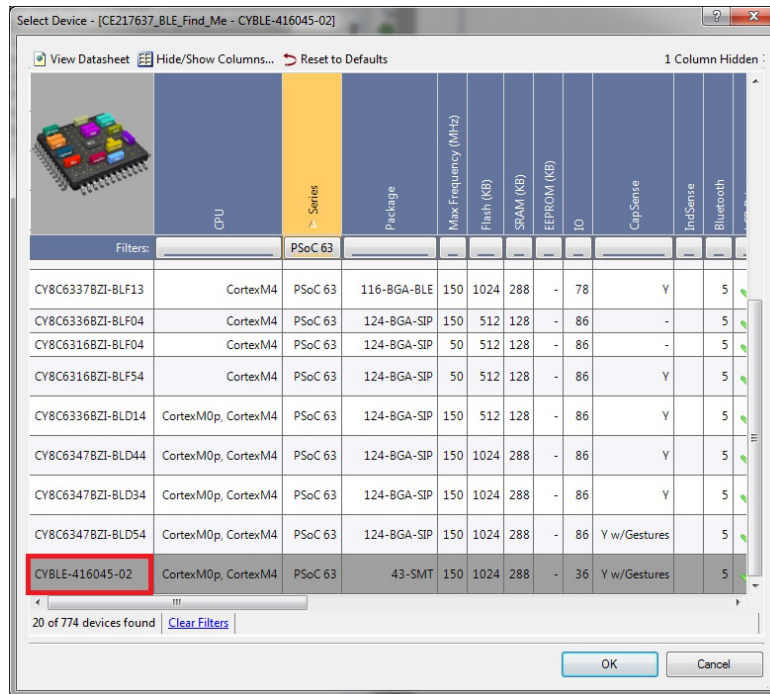
- Right-click the project name, and select **Device Selector**.

Figure 4-5. Project Configuration



Next from the list of devices, select the **CYBLE-416054-02** device from the list.

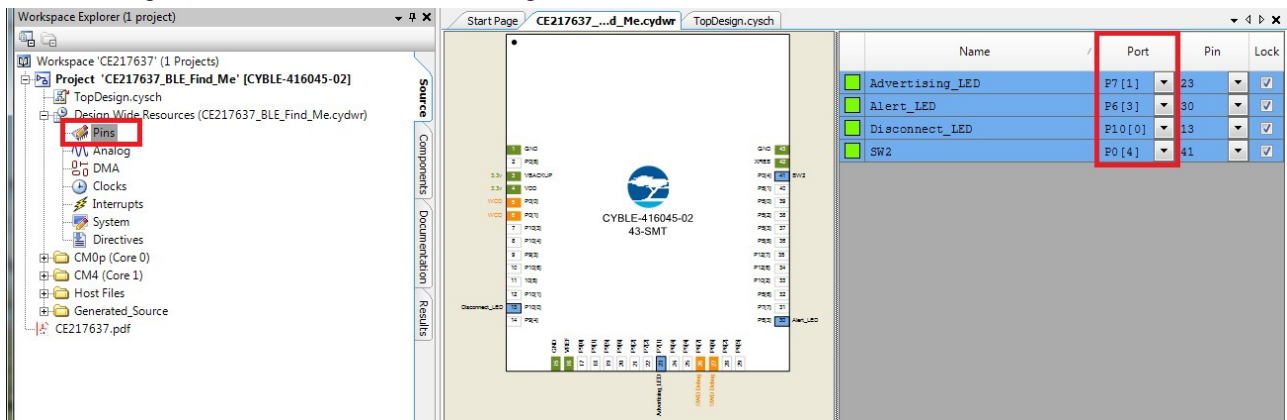
Figure 4-6. Device Selector Wizard



8. On the left-hand side in the Workspace Menu, select the “Pins” option under the “Design Wide Resources” tab. The Pins tab will appear on the right as shown, see Figure 4-7. Because this code example was originally created for a different board, you will need to reassign the LED pins to be compatible with the CY8CPROTO-063-BLE Prototyping Board. Make the following assignments.

- Advertising_LED => P7[1]
- Alert_LED => P6[3]
- Disconnect_LED => P9[4] (This one is not used for this kit since there are only 2 LEDs. It can be assigned to any unused pin.)

Figure 4-7. Pin Selection / Reassignment



9. Build and program the project as described in the [Programming and Debugging the Target PSoC 6 MCU](#) on page 19.

A. Appendix

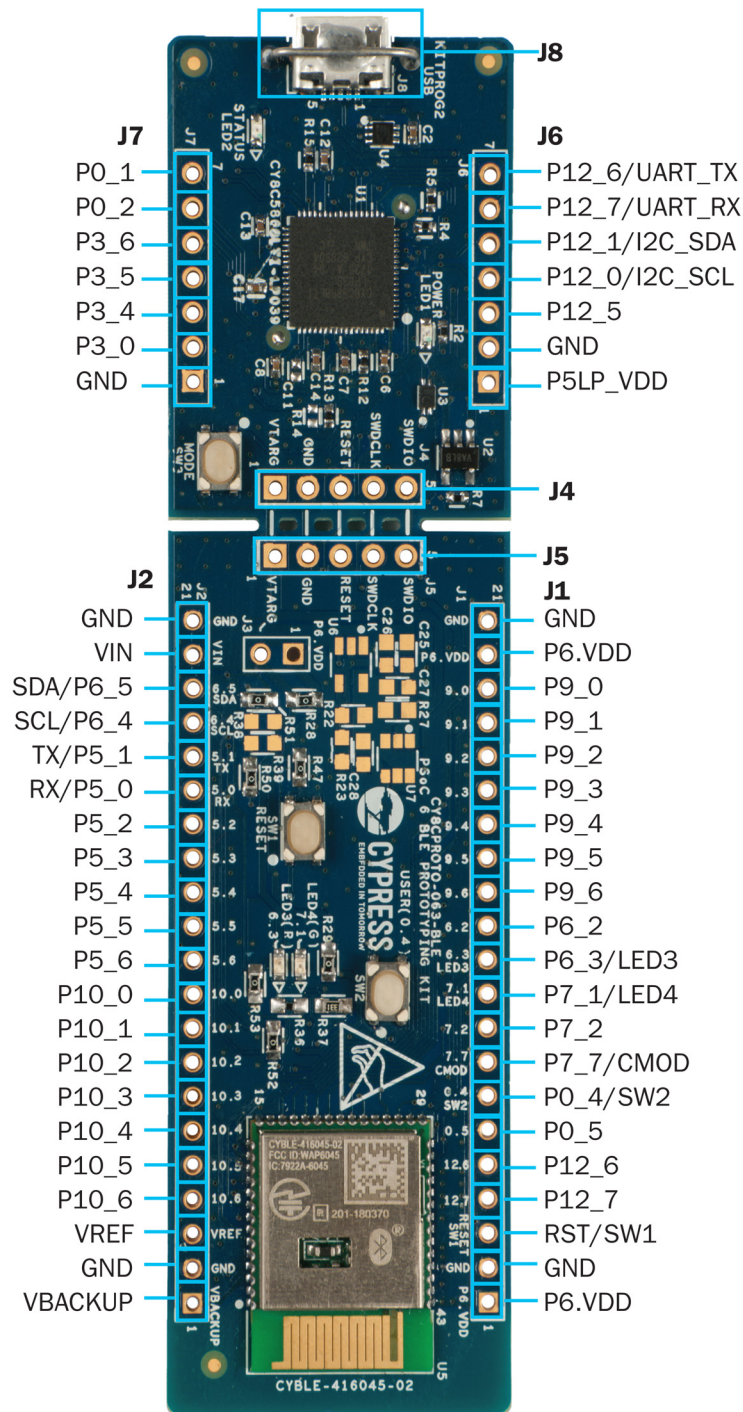


A.1 Board Details

The PSoC 6 BLE Prototyping Board consists of the following blocks:

- EZ-BLE Creator Module (CYBLE-416045-02)
- PSoC 6 MCU I/O headers J1 and J2
- KitProg2 (PSoC 5LP) device (CY8C5868LTI-LP039)
- KitProg2 I/O headers J6 and J7
- SWD connection headers J4 and J5
- USB 2.0 Micro-B connector J8
- Power LED, LED1 (amber)
- KitProg2 Status LED, LED2 (amber)
- User LED, LED3 (red)
- User LED, LED4 (green)
- ECO Crystal (footprint)
- WCO Crystal (32.768 kHz)
- One Push Button SW2 (User)
- One Push Button SW1 (Reset)
- One Push Button SW3 (KitProg2 Mode)
- Current Measurement Jumper J3 (foot-print only, shorted by the zero-ohm resistor R28)
- Perforated 'snappable' board design

Figure A-1. PSoC 6 BLE Prototyping Board Pin Details

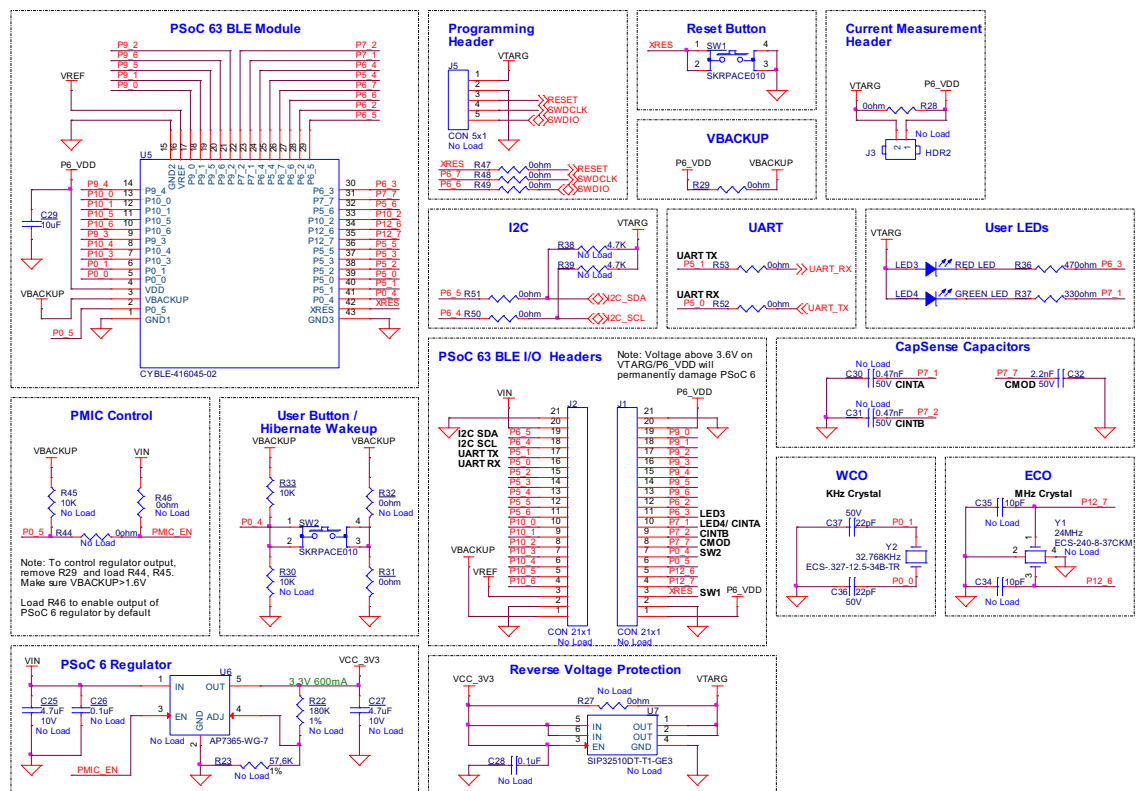


A.2 Hardware Details

A.2.1 Target Board

The target board uses the CYBLE-416045-02 EZ-BLE Creator Module, which contains a PSoC 6 MCU, CY8C6347BZI-BLD53. PSoC 6 MCU is a scalable and reconfigurable platform architecture for a family of programmable embedded system controllers with both Arm Cortex-M0+ CPU and Cortex-M4 CPU on a single chip. It combines programmable and reconfigurable digital blocks with flexible automatic routing. The PSoC 6 MCU, based on this platform architecture, is a combination of an MCU with digital programmable logic, programmable interconnect, and standard communication and timing peripherals. The digital subsystems allow flexibility and in-field tuning of the design. For more information, see the [PSoC 6 MCU Datasheet](#) and the [CYBLE-416045-02 EZ-BLE Creator Module](#).

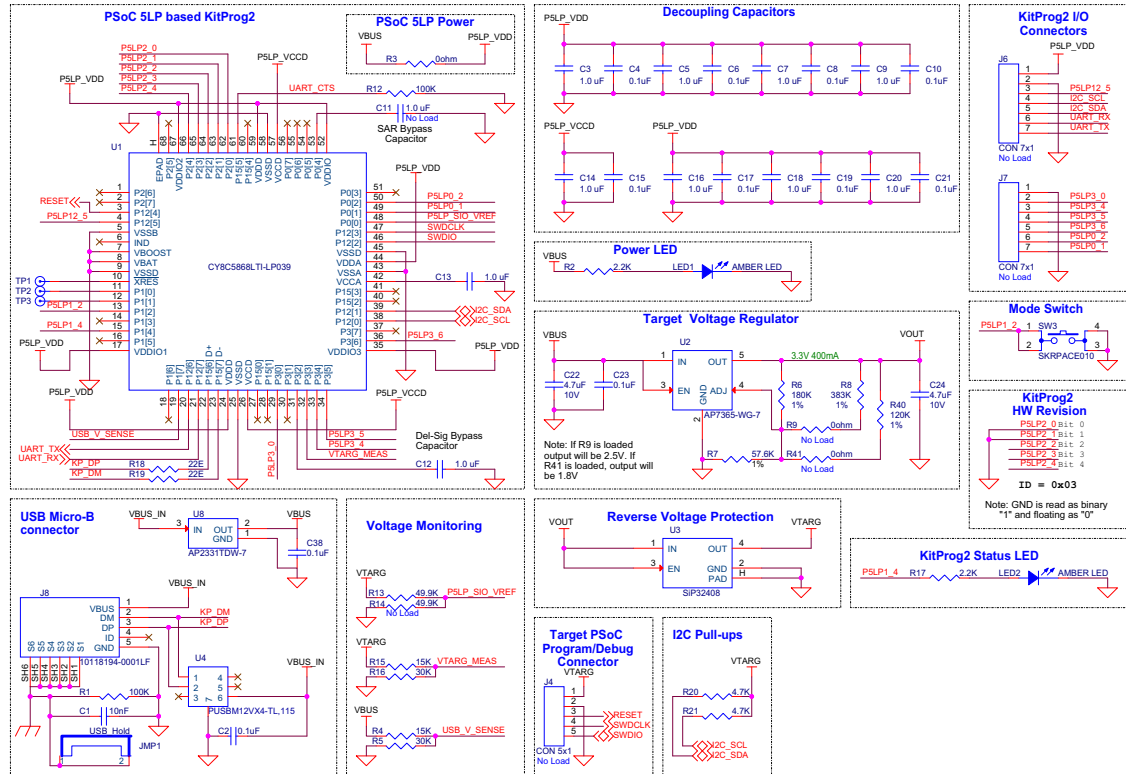
Figure A-2. Schematic of PSoC 6 MCU (Target)



KitProg2 Board

A PSoC 5LP device on the KitProg2 board is used to program and debug the target PSoC 6 MCU. The PSoC 5LP-based KitProg2 connects to the USB port of the PC through the USB 2.0 Micro-B connector and to the SWD interface of the target EZ-BLE Creator Module.

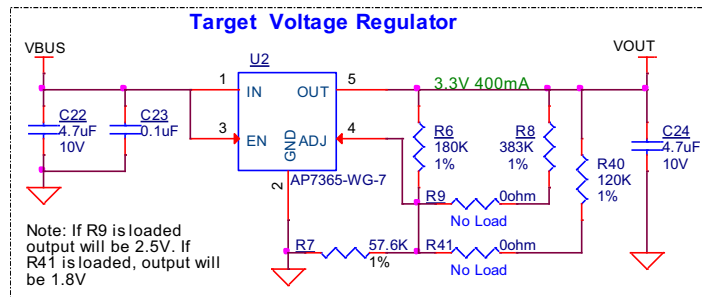
Figure A-3. Schematic of PSoC 5LP (KitProg2)



A.2.3 Power Supply System

The power supply system uses 5 V from the USB connector as the input to a voltage regulator on the KitProg2 section of the board. This regulator provides 3.3 V to the PSoC 6 MCU by default. Provisions have been added so that the output can be easily changed to 2.5 V or 1.8 V by adding either R9 or R41 zero-ohm resistors respectively. You can also connect an external power supply to the board at P6.VDD that is present on pin 20 of J1 for low-voltage applications.

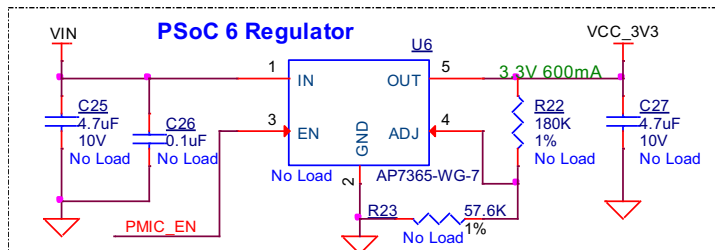
Figure A-4. Target Power Supply



A.2.3.1 Optional PSoC 6 MCU Voltage Regulator

If the PSoC 6 MCU section of the board is disconnected from the KitProg2 side by breaking the board between J4 and J5, there are two power options. The first option is to power the board directly with the required voltage at the target voltage (P6.VDD) on connector J1, pin 20. The other alternative is to load the optional components for the PSoC 6 MCU regulator and apply a voltage at the VIN connector (J2, pin 20). See schematic below for the PSoC 6 MCU regulator.

Figure A-5. Optional PSoC 6 MCU Regulator



Caution: Note that the PSoC 6 BLE Prototyping Board does not have any onboard ESD protection circuitry. Therefore, the power source for the PSoC 6 BLE Prototyping Board must be of high quality to make sure that the board is protected from any over-current conditions and swapped-power connections.

A.2.3.2 Measure EZ-BLE Creator Module Current Consumption

To measure the EZ-BLE Creator Module current, follow these steps:

1. Remove the zero-ohm resistor R28 and install a 2-pin jumper in the supplied holes of J3.
2. Connect an ammeter across the 2-pin jumper to measure the current to the EZ-BLE Creator Module.

This method can be used either with USB power or with the power supplied to one of the VTARG pins (J5.1 or J4.1), but NOT when supplying power to one of the P6.VDD pins (J1.1 or J1.20).

After measuring the current consumption, populate resistor R28 or place a shorting jumper across the two jumper pins for normal operation of the kit.

Note: After removing R28, system may have some leakage power such as the following:

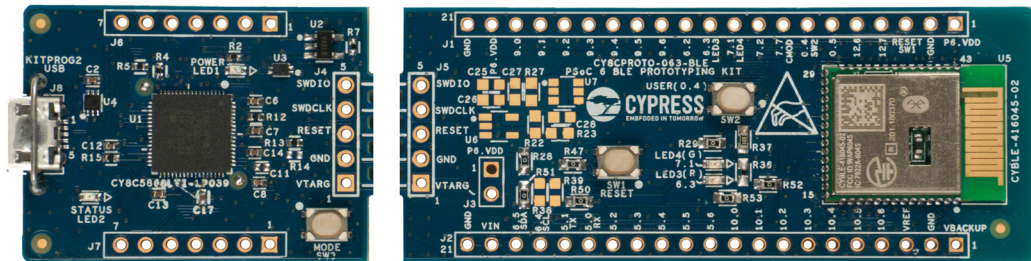
1. Communication circuitry between KitProg2 and Target (I2C, UART, SWDIO)
2. Reset switch between KitProg2 and target device along with PSoC 6 MCU

A.2.4 Board Separation (Snapping)

The PSoC 6 BLE Prototyping Board consists of a PSoC 6 MCU and a KitProg2 board. To separate the two boards for evaluation or development, break the two boards apart at the built-in perforated edge between **J4** and **J5**.

The easiest method of separating the two boards is to place the kit on the edge of a table, where the edge of the table is directly below the perforated edge and the smaller KitProg2 board is off the table edge. Press gently on the KitProg2 board and snap the two boards apart. If any material is removed from the edge of the boards, use sheers to clean up the edge of the kit.

Figure A-6. PSoC 6 BLE Prototyping Board broken into two parts



After the PSoC 6 BLE Prototyping Board is broken into two, the two halves can be reconnected for programming and debugging by adding connectors at J5 and J6. Also, a Cypress MiniProg3 device can be connected to a connector at J5 to perform programming and debug.



Caution: Once the boards are separated, direct UART and I2C connections between the EZ-BLE Creator Module and KitProg2 are lost. This is because the traces connecting the UART and I2C lines are cut off during the separation. However, you can access KitProg2's UART and I2C lines through header J6 and make the following connections to J2.

UART Reconnect:

- J6.7 to J2.16 (PSoC 5LP P12.6 to PSoC 6 P5.0)
- J6.6 to J2.17 (PSoC 5LP P12.7 to PSoC 6 P5.1)

I2C Reconnect:

- J6.5 to J2.19 (PSoC 5LP P12.1 to PSoC 6 P6.5)
- J6.4 to J2.18 (PSoC 5LP P12.0 to PSoC 6 P6.4)

A.2.5 Header Connections

The PSoC 6 BLE Prototyping Board supports a number of unpopulated headers on both the Kit-Prog2 and the target PSoC 6 MCU sections.

A.2.5.1 Functionality of the J1 and J2 Headers (Target Board)

The target PSoC 6 MCU section contains two single inline headers (J1 and J2). These 1×21-pin headers have 0.1-inch spacing and include all of the GPIOs available on the EZ-BLE Creator Module. Power pins, reference input and reset signal are also available on these connectors. See [Figure A-7](#) for details.

The target headers J1 and J2 are partially compatible with mikroBUS™. It supports only 3.3 V operation with [Click Boards](#), and does not have the reset pin functionality.

Figure A-7. J1 and J2 Headers

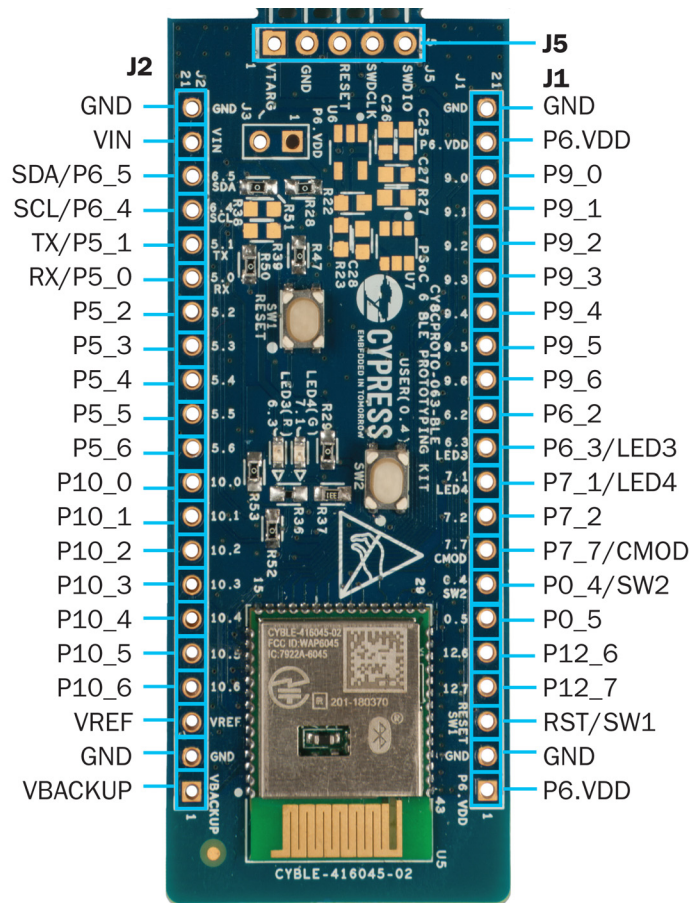


Table A-1. Pin Details of J1 and J2 Headers

PSoC 6 BLE Prototyping Board GPIO Header (J2)			PSoC 6 BLE Prototyping Board GPIO Header (J1)		
Pin	Signal	Description	Pin	Signal	Description
J2_01	VBKUP*	Backup Power	J1_01	P6_VDD*	Power
J2_02	GND	Ground	J1_02	GND	Ground
J2_03	VREF	SAR ADC Vref	J1_03	RST	PSoC 6 MCU Reset
J2_04	P10.6	GPIO	J1_04	P12.7	GPIO
J2_05	P10.5	GPIO	J1_05	P12.6	GPIO
J2_06	P10.4	GPIO	J1_06	P0.5	GPIO
J2_07	P10.3	GPIO	J1_07	P0.4	GPIO/User SW2
J2_08	P10.2	GPIO	J1_08	P7.7	GPIO/CMOD
J2_09	P10.1	GPIO	J1_09	P7.2	GPIO
J2_10	P10.0	GPIO	J1_10	P7.1	GPIO/LED4
J2_11	P5.6	GPIO	J1_11	P6.3	GPIO/LED3
J2_12	P5.5	GPIO	J1_12	P6.2	GPIO
J2_13	P5.4	GPIO	J1_13	P9.6	GPIO
J2_14	P5.3	GPIO	J1_14	P9.5	GPIO
J2_15	P5.2	GPIO	J1_15	P9.4	GPIO
J2_16	P5.0	GPIO/UART_RX	J1_16	P9.3	GPIO/SPI-CS
J2_17	P5.1	GPIO/UART_TX	J1_17	P9.2	GPIO/SPI-SCK
J2_18	P6.4	GPIO/I2C-SCL	J1_18	P9.1	GPIO/SPI-MISO
J2_19	P6.5	GPIO/I2C-SDA	J1_19	P9.0	GPIO/SPI-MOSI
J2_20	VIN	Input Voltage	J1_20	P6_VDD*	Target Voltage Input
J2_21	GND	Ground	J1_21	GND	Ground

* **Note:** P6_VDD and VBKUP should never exceed 3.6 V.

A.2.5.2 Functionality of J4 and J5 Headers (KitProg2 to PSoC 6 MCU)

The KitProg2 and target boards each contain a 1×5-pin header. These headers provide a physical connection between the two devices. Specifically, the connection includes the SWD interface, required to program/debug the target PSoC 6 MCU in the EZ-BLE Creator Module, power, ground, and reset.

Figure A-8. J4 and J5 Headers

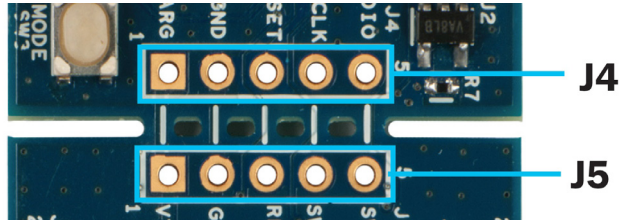


Table A-2. Pin Details of J4 and J5 Headers

KitProg2 Header (J4)			PSoC 6 MCU Header (J5)		
Pin	Signal	Description	Pin	Signal	Description
J4_1	VTARG	Power	J5_1	VTARG	Power
J4_2	GND	Ground	J5_2	GND	Ground
J4_3	P12.4	Reset	J5_3	XRES	Reset
J4_4	P12.3	SWD_CLK	J5_4	P6.7	SWD_CLK
J4_5	P12.2	SWD_IO	J5_5	P6.6	SWD_IO

When the boards are separated, the KitProg2 board can be used to program any other PSoC 3, PSoC 4, PSoC 5LP and PSoC 6 family of devices via J4.

A.2.5.3 Functionality of J6 and J7 Headers (KitProg2)

The KitProg2 board contains two single in-line headers (J6 and J7). Both are 1×7-pin-headers, used to pull out several pins of the PSoC 5LP device to support advanced features such as low-speed oscilloscope and a low-speed digital logic analyzer. This header also contains KitProg2 UART and I2C bridge pins that can be used when the two boards are separated.

The J6 and J7 headers support 100-mil spacing, so you can solder connectors to connect the KitProg2 board to a development breadboard.

Figure A-9. J6 and J7 Headers

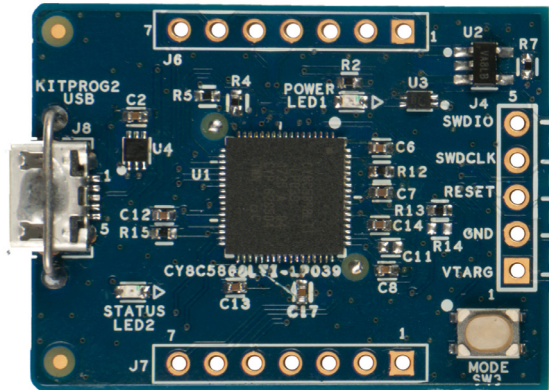


Table A-3. Pin Details of J6 and J7 Headers

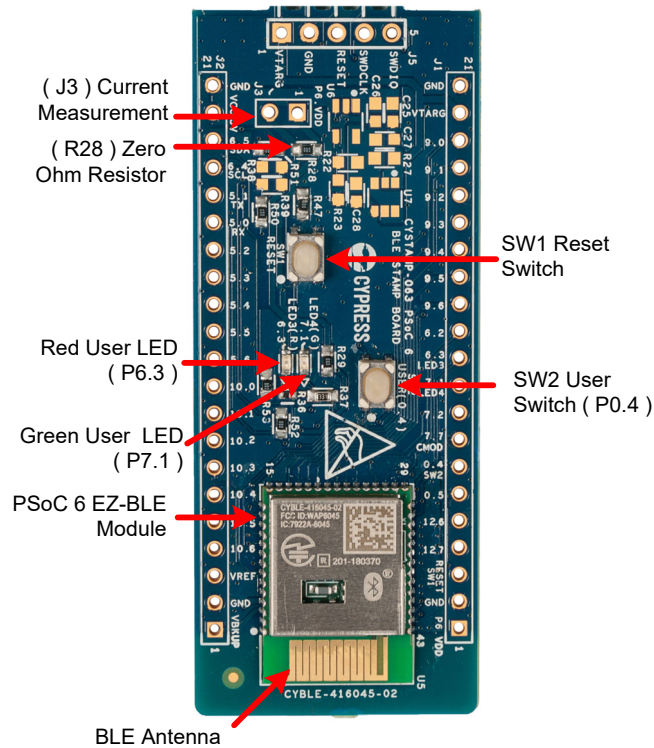
PSoC 5LP KitProg2 Header (J6)			PSoC 5LP KitProg2 Header (J7)		
Pin	Signal	Description	Pin	Signal	Description
J6_01	P5LP_VDD	Power	J7_01	GND	Ground
J6_02	GND	Ground	J7_02	P3.0	GPIO
J6_03	P12.5	GPIO	J7_03	P3.4	GPIO
J6_04	P12.0	GPIO/I2C_SCL	J7_04	P3.5	GPIO
J6_05	P12.1	GPIO/I2C_SDA	J7_05	P3.6	GPIO
J6_06	P12.7	GPIO/UART_RX	J7_06	P0.2	GPIO
J6_07	P12.6	GPIO/UART_TX	J7_07	P0.1	GPIO

A.2.6 PSoC 6 BLE Prototyping Board features

Use the following features of the Prototyping Board to perform a quick evaluation right out of the box.

- Two user LEDs connected to GPIO pins for feedback
- User switch for user input
- Jumper for current measurement (when R28 is removed)
- Onboard BLE Antenna
- Onboard reset switch

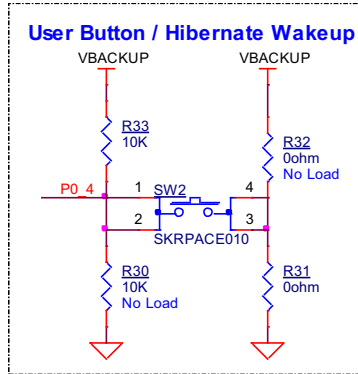
Figure A-10. PSoC 6 BLE Prototyping Board Features



A.2.6.1 User Button

The target PSoC 6 MCU board contains a button connected to the **P0.4** pin on the PSoC 6 MCU. This button can be used for general user inputs or for wakeup during Hibernate mode. The buttons connect to ground on activation (active low) by default. User button (SW2) can be changed to active high mode by changing the zero resistors shown below.

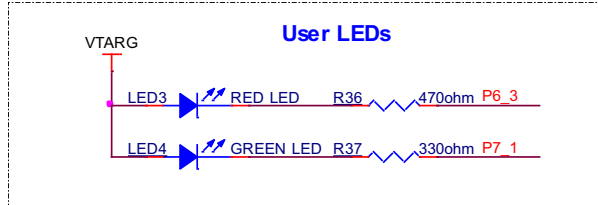
Figure A-11. User Button on the Board



A.2.6.2 User LEDs

There are two user LEDs on the board connected to pins P6.3 (red) and P7.1 (green) to allow the user to indicate an event or mode. The drive is active LOW so the LED will be ON when the GPIO is driven to a '0' and OFF when driven to a '1'.

Figure A-12. User LEDs



A.2.6.3 Reset Button

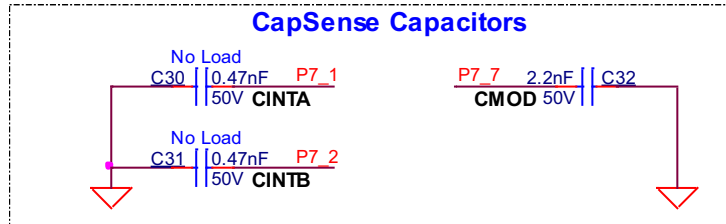
When the Reset button (SW1) is pressed, the XRES line of the PSoC 6 MCU is pulled to ground, which resets the target device.

A.2.6.4 CapSense Capacitors

There are footprints for three CapSense capacitors - one for self-capacitance and one for mutual Capacitance sensing modes, but only the CMOD capacitor used for self-capacitance mode is loaded. If CapSense is not used but P7.7 is used for a digital or analog function, C32 may need to be removed. See the datasheet on the CapSense component for more information on how to select and use these capacitors (C30-CINTA, C31-CINTB, and C32-CMOD).

Note: P7[1] is used for LED4 (green LED) by default. This must be disconnected by removing R37 before loading C30 for mutual capacitive sensing (CSX).

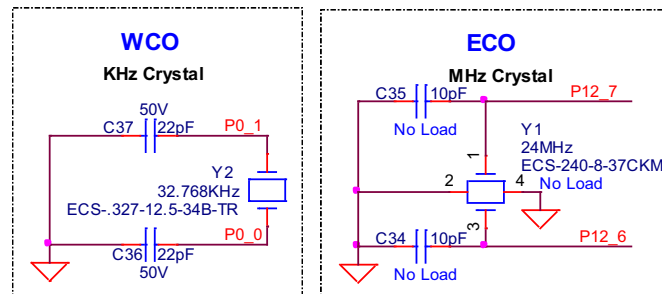
Figure A-13. CapSense Capacitors



A.2.6.5 System Crystals

Two different crystal oscillator inputs are available on the board. The WCO kHz crystal (32.768 kHz) is populated and is used for BLE timing. Pads for the ECO MHz crystal and load crystals are on the board so that you can easily select the crystal of your choice. The ECO is optional and only required when the internal clock must be more accurate than the IMO (Internal Main Oscillator). The internal FLL and PLL help to provide a wide range of options with either the ECO or IMO. The crystal used for the BLE radio is pre-installed in the EZ-BLE Creator Module.

Figure A-14. System Crystals



A.2.6.6 EZ-BLE Creator Module

CYBLE-416045-02 is a fully certified and qualified module supporting Bluetooth Low Energy (BLE) wireless communication. CYBLE-416045-02 is a turnkey solution and includes onboard crystal oscillators, trace antenna, passive components, and the Cypress PSoC 6 MCU.

The EZ-BLE Creator Module is a scalable and reconfigurable platform architecture. It combines programmable and reconfigurable analog and digital blocks with flexible automatic routing. CYBLE-416045-02 also includes digital programmable logic, high-performance analog-to-digital conversion (ADC), low-power comparators, and standard communication and timing peripherals.

CYBLE-416045-02 includes a royalty-free BLE stack compatible with Bluetooth 5.0 and provides up to 36 GPIOs in a 14 × 18.5 × 2.00 mm package. CYBLE-416045-02 is a complete solution and an ideal fit for applications seeking a high-performance BLE wireless solution. See the [CYBLE-416045-02 EZ-BLE Creator Module web page](#).

The module is based on CY8C6347BZI-BLD53 but gives access only to a sub-set of the I/O pins and functionality. Most PSoC 63 code examples can be adapted to run on this module.

A.2.6.7 KitProg2 Switch and LEDs

The KitProg2 board contains a push button connected to P1.2 of PSoC 5LP. When this button is pressed, it toggles between KitProg2 mode and CMSIS-DAP mode. For more details, see the [KitProg2 User Guide](#).

A power indicator LED (LED1) will turn ON when the board is plugged into an active USB port to indicate that power is being supplied.

A second LED (LED2) is used to indicate the status of KitProg2. The Amber Status LED (LED2) indicates the KitProg2 status connected to **P1[4]** of the KitProg2 PSoC 5LP device. See [Table A-4](#) for a summary of the status LED states. For more details on the KitProg2 status LED, see the [KitProg2 User Guide](#).

Table A-4. LED status for KitProg2 Modes

KitProg2 Programming Modes *	Status LED (LED2)
KitProg2 Program / Debug mode (default)	ON
CMSIS-DAP mode	OFF

* **Note:** Toggling between programming modes can be done by pressing mode switch SW3.

A.3 Programming PSoC 6 BLE Prototyping Board using MiniProg3/KitProg2

The target board provides a provision to program the PSoC 6 MCU using MiniProg3 or an external KitProg2 via the 5-pin SWD header. To do this, connect wires or a 5-pin 100-mil spaced header to J5. The PSoC 6 BLE Prototyping Board supports both power cycle and reset programming modes.



Caution: MiniProg3 can provide up to 5 V at VTARG which will result in permanent damage of the device. Please make sure that VTARG is in the range 1.8 V to 3.3 V. Ideally, this configuration is only used when the KitProg2 section is separated.

Figure A-15. Connecting the CY8CPROTO-063-BLE to a MiniProg3



Note: CY8CKIT-002 MiniProg3 is not part of the PSoC 6 BLE Prototyping Board contents and can be purchased from the [Cypress Online Store](#).

Revision History



Document Revision History

Document Title: CY8CPROTO-063-BLE PSoC 6 BLE Prototyping Board Guide			
Document Number: 002-24993			
Revision	ECN Number	Issue Date	Description of Change
**	6372164	11/02/2018	New kit guide.
*A	6776886	01/17/2020	Updated Appendix chapter on page 26 : Updated "Hardware Details" on page 28 : Updated "Target Board" on page 28 : Updated Figure A-2 . Updated "PSoC 6 BLE Prototyping Board features" on page 36 : Updated "System Crystals" on page 38 : Updated Figure A-14 . Updated to new template.