CY8CKIT-062-WiFi-BT

PSoC® 6 WiFi-BT Pioneer Kit Guide

Doc. # 002-22677 Rev. *B

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

Copyrights

Contents

Safety Information

Regulatory Compliance

The CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT 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. 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 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 PSoC 6 WiFi-BT Pioneer Kit 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 PSoC 6 WiFi-BT Pioneer Kits in the protective shipping package.

End-of-Life/Product Recycling

The end-of life for this kit is five years from the date of manufacture mentioned as a bar code on the back of the kit box. Contact your nearest recycler for information on 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 an ESD workstation is unavailable, use appropriate ESD protection by wearing an anti-static wrist strap attached to a grounded metal object.

Handling Boards

The PSoC 6 WiFi-BT Pioneer Kit 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 the board over any surface.

1. Introduction

Thank you for your interest in the CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit. The PSoC 6 WiFi-BT Pioneer Kit enables you to evaluate and develop your applications using the PSoC 6 MCU.

The PSoC 6 MCU is Cypress' latest, ultra-low-power PSoC specifically designed for wearables and IoT products. It is a programmable embedded system-on-chip, integrating a 150-MHz Arm® Cortex®-M4 as the primary application processor, a 100-MHz CM0+ that supports low-power operations, up to 1 MB Flash and 288 KB SRAM, CapSense® touch-sensing, and programmable analog and digital peripherals that allow higher flexibility, in-field tuning of the design, and faster time-to-market.

The PSoC 6 WiFi-BT Pioneer board offers compatibility with Arduino™ shields. The board features a PSoC 6 MCU, a 512-Mb NOR flash, an onboard programmer/debugger (KitProg2), a 2.4-GHz WLAN and Bluetooth functionality module (CYW4343W), a USB Type-C power delivery system (EZ-PD™ CCG3), a five-segment CapSense slider, two CapSense buttons, one CapSense proximity sensing header, an RGB LED, two user LEDs, USB host and device features, and one push button. The board supports operating voltages from 1.8 V to 3.3 V for the PSoC 6 MCU.

The CY8CKIT-062-WiFi-BT package includes a CY8CKIT-028-TFT Display Shield that contains a 2.4-inch TFT display, a motion sensor, ambient light sensor, a 32-bit audio codec, and a PDM microphone.

This is the first kit that enables development of PSoC 6 MCU + WiFi applications. To develop a PSoC 6 MCU + WiFi application, WICED™ Studio 6.1 or later must be used. WICED Studio is Cypress' integrated development environment (IDE) for developing WiFi applications. If you are not adding WiFi to your design then you can develop and debug PSoC 6 MCU project using PSoC Creator™. PSoC Creator supports exporting your designs to other third-party firmware development tools.

1.1 Kit Contents

The CY8CKIT-062-WiFi-BT package has the following contents, as shown in Figure 1-1.

- PSoC 6 WiFi-BT Pioneer Board
- CY8CKIT-028-TFT Display Shield
- USB Type-A to Type-C cable
- Four jumper wires (4 inches each)
- Two proximity sensor wires (5 inches each)
- Quick Start Guide

Figure 1-1. Kit Contents

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

1.2 Hardware Introduction

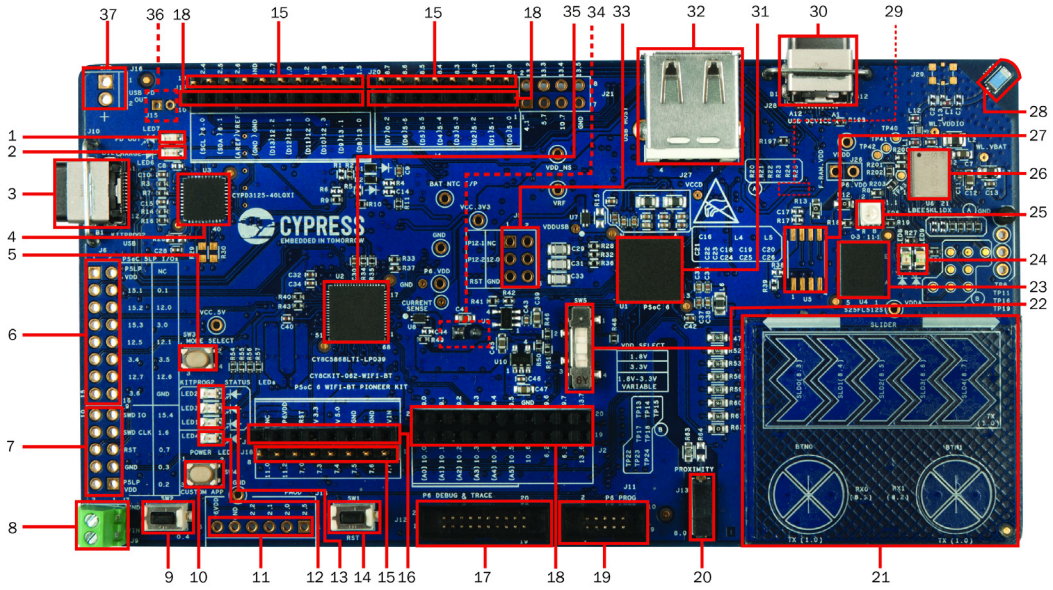
1.2.1 CY8CKIT-062-WiFi-BT Board Details

Figure 1-2 shows the Pioneer board, which has the following features:

- PSoC 6 MCU
- Expansion headers that are compatible with Arduino Uno 3.3-V shields[1] and Digilent® Pmod™ modules
- Type 1DX ultra-small 2.4-GHz WLAN and Bluetooth functionality module
- 512-Mbit external quad-SPI NOR flash that provides a fast, expandable memory for data and code
- KitProg2 onboard programmer/debugger with mass storage programming, USB to UART/I2C/SPI bridge functionality, and custom applications support
- EZ-PD CCG3 USB Type-C power delivery (PD) system with rechargeable lithium-ion polymer (Li-Po) battery support[2]
- CapSense touch-sensing slider (five elements) and two buttons, all of which are capable of both self-capacitance (CSD) and mutual-capacitance (CSX) operation, and a CSD proximity sensor that allows you to evaluate Cypress' fourth-generation CapSense technology
- 1.8-V to 3.3-V operation of PSoC 6 MCU is supported. An additional 330-mF super-capacitor is provided for backup domain supply (VBACKUP)
- Two user LEDs, an RGB LED, a user button, and a reset button for PSoC 6 MCU.
- Two buttons and three LEDs for KitProg2.

1. 5-V shields are not supported
2. Battery and power-delivery capable USB Type-C to Type-C cable are not included in the kit package and should be purchased separately.

Figure 1-2. Pioneer Board



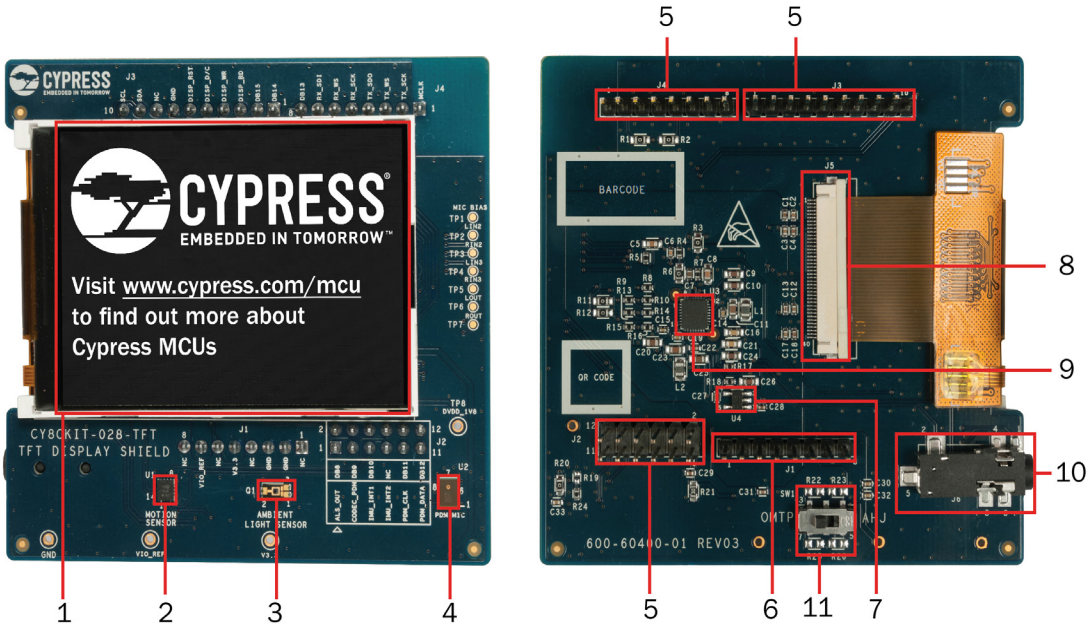
1. USB PD output voltage availability indicator (LED7)
2. Battery charging indicator (LED6)
3. KitProg2 USB Type-C connector (J10)
4. Cypress EZ-PD™ CCG3 Type-C Port Controller with PD (CYPD3125-40LQXI, U3)
5. KitProg2 programming mode selection button (SW3)
6. KitProg2 I/O header (J6)1
7. KitProg2 programming/custom application header (J7)1
8. External power supply connector (J9)
9. PSoC 6 user button (SW2)
10. KitProg2 application selection button (SW4)
11. Digilent® Pmod™ compatible I/O header (J14)1
12. Power LED (LED4)
13. KitProg2 status LEDs (LED1, LED2, and LED3)
14. PSoC 6 reset button (SW1)
15. PSoC 6 I/O header (J18, J19 and J20)
16. Arduino™ Uno R3 compatible power header (J1)
17. PSoC 6 debug and trace header (J12)
18. Arduino Uno R3 compatible PSoC 6 I/O header (J2, J3 and J4)
19. PSoC 6 program and debug header (J11)
20. CapSense proximity header (J13)
21. CapSense slider and buttons
22. PSoC 6 VDD selection switch (SW5)
23. Cypress 512-Mbit serial NOR Flash memory (S25-FL512S, U4)
24. PSoC 6 user LEDs (LED8 and LED9)
25. RGB LED (LED5)
26. WiFi/BT module (LBEE5KL 1DX, U6)
27. Cypress serial Ferroelectric RAM (U5)1
28. WiFi-BT Antenna
29. VBACKUP and PMIC control selection switch (SW7)2
30. PSoC 6 USB device Type-C connector (J28)
31. Cypress PSoC 6 (CY8C6247BZI-D54, U1)
32. PSoC 6 USB Host Type-A connector (J27)
33. Arduino Uno R3 compatible ICSP header (J5)1
34. PSoC 6 power monitoring jumper (J8)2
35. KitProg2 (PSoC 5LP) programmer and debugger (CY8C5868LTI-LP039, U2)
36. Battery connector (J15)1,2
37. USB PD output voltage (9V/12V) connector (J16)1

1.2.2 CY8CKIT-028-TFT Board Details

Figure 1-3 shows the TFT display shield that has the following features:

- A 2.4-inch Thin-Film Transistor (TFT) LCD module with 240x320 pixel resolution.
- A three-axis acceleration and three-axis gyroscopic motion sensor.
- A PDM microphone for voice input.
- A 32-bit stereo codec with microphone, headphone, and speaker amplifier capability.
- An audio jack with a provision of connecting both AHJ and OMTP headphones. The headset standard can be set by an onboard switch.
- An ambient light sensor IC made of an NPN phototransistor.
- An LDO that converts 3.3 V to 1.8 V for the digital supply of the audio codec.

Figure 1-3. TFT Display Shield



1. 2.4-inch TFT display
2. Motion Sensor (U1)
3. Ambient Light Sensor (Q1)
4. PDM microphone (U2)
5. Ardunio compatible I/O headers (J2, J3, J4)
6. Ardunio compatible power header (J1)
7. TFT display power control load switch (U4)
8. TFT display connector (J5)
9. Audio CODEC (U3)
10. Audio Jack (J6)
11. Audio Jack Selection (OMTP/AHJ) Switch (SW1)

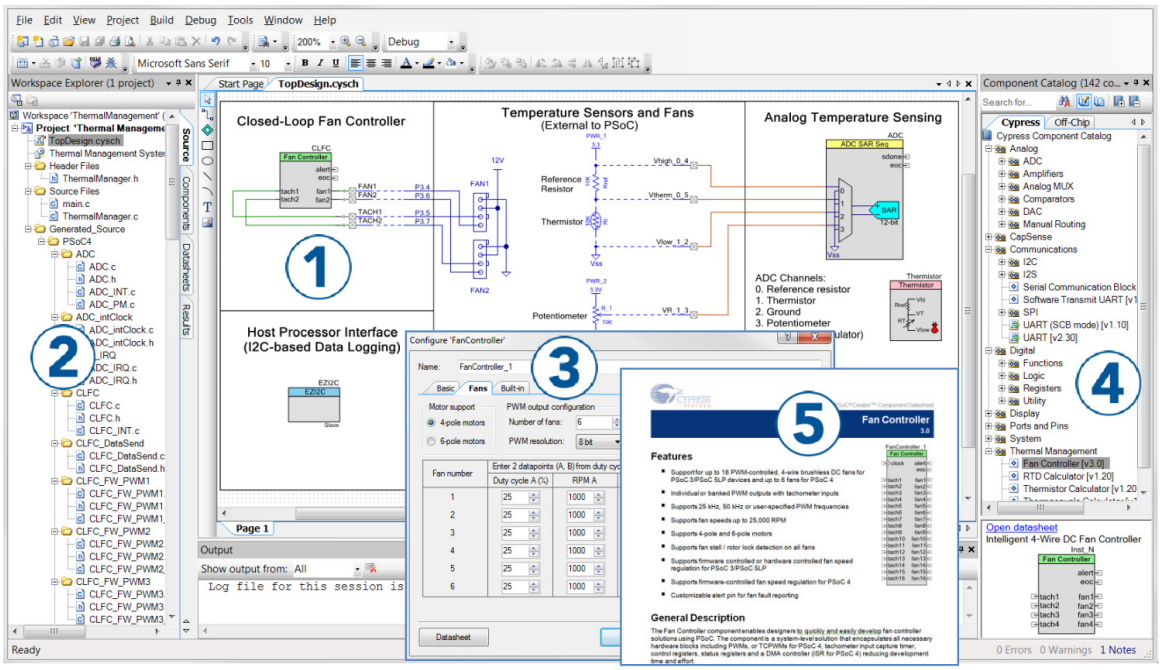
1.3 Software Introduction

1.3.1 PSoC Creator Overview

PSoC Creator is a state-of-the-art, easy-to-use IDE. It uses revolutionary hardware and software co-design, powered by a library of fully verified and characterized PSoC Components™ and peripheral driver libraries (PDL), as shown in Figure 1-4. 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 or PDL.
4. Explore the library of 100+ Components.
5. Access Component datasheets.
6. Export your design to third-party firmware development tools.

Figure 1-4. PSoC Creator Features



PSoC Creator also enables you to tap into an entire tool ecosystem with integrated compiler chains and production programmers for PSoC devices. Use PSoC Creator for all PSoC 6 MCU designs that do not require WiFi connectivity. If WiFi connectivity is required, use WICED Studio 6.1 or higher.

1.3.1.1 PSoC Creator Code Examples

PSoC Creator includes a large number of code examples. These examples are accessible from the PSoC Creator Start Page, as shown in Figure 1-5 or from the menu **File** > **Code Example**.

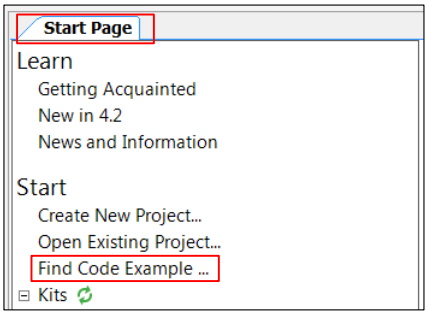
Code examples can speed up your design process by starting you off with a complete design. The code examples also show how to use PSoC Creator Components for various applications. Code examples and documentation are included.

In the **Find Code Example** dialog, you have several options:

- Filter for examples based on device family or keyword.
- Select from the list of examples offered based on the **Filter Options**.

- View the project documentation for the selection (on the **Documentation** tab).
- View the code for the selection on the **Sample Code** tab. You can also copy and paste code from this window to your project, which can help speed up code development.
- Create a new workspace for the code example or add to your existing workspace. 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 Examples in PSoC Creator



1.3.1.2 *Kit Code Examples*

You can access the installed kit code examples from the PSoC Creator Start Page. To access these examples, expand the Kits under the section Examples and Kits; then, expand the specific kit to see the code examples. To work with the code examples using this kit, see PSoC Creator chapter on page 18.

1.3.1.3 *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 the **Find Code Example** option in the context menu that appears.
- **System Reference Guide:** Choose **Help** > **System Reference Guide**. This guide lists and describes the system functions provided by PSoC Creator.
- **Component Datasheets:** Right-click a Component and select **Open Datasheet**.

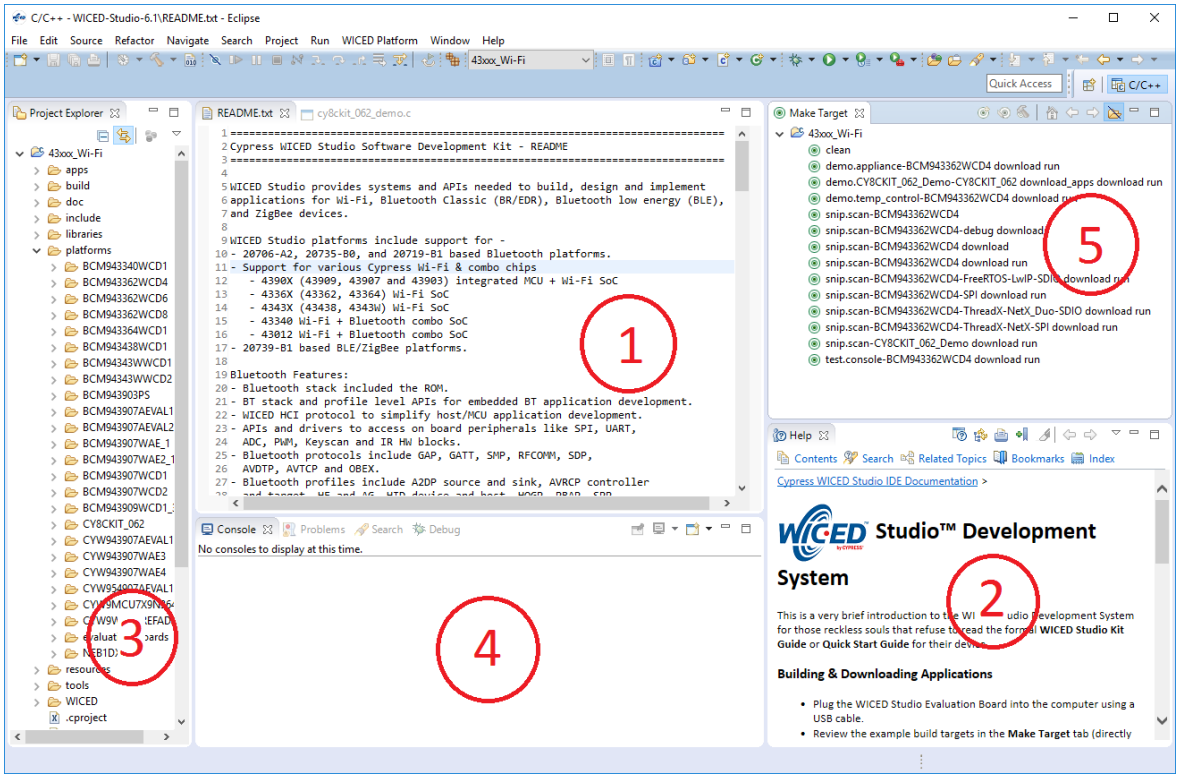
1.3.2 WICED Studio Development System Overview

WICED Studio 6.1 (or later) supports WiFi and Bluetooth application development using the CY8CKIT-062-WiFi-BT kit. Tabs and their location in the WICED IDE are as shown in Figure 1-6.

Figure 1-6 illustrates the following:

1. Edit your application firmware.
2. Help Window that contains instructions on building and downloading applications.
3. Explore existing applications/firmware and library of the Software Development Kit (SDK).
4. View Build messages in the Console window.
5. Create and edit Make Targets for the platform to build your application/project.

Figure 1-6. WICED IDE



1.3.2.1 *WICED Studio Code Examples*

WICED Studio includes libraries and code examples supporting both Bluetooth and WiFi platforms. Selecting the 43xxx_Wi-fi Filter will show only WiFi platform related files in the project explorer as shown in Figure 1-7.

Application examples can speed up the design process by serving as templates for development. Code examples are located under the *apps* category (in the Project Explorer window), as shown in Figure 1-8. Code examples under *apps* are further grouped into *demo*, *snip*, *test*, *waf* (WICED application framework), and *wwd* (WICED WiFi Driver Application) directories.

The *demo* directory contains applications that combine various WICED features into a single application. The *snip* directory contains application snippets that demonstrate how to use various WICED libraries and API functions. The `test` directory contains applications that are used for simple test and utility. The *waf* directory contains applications that are part of WICED application framework, for instance, the bootloader. The *wwd* directory contains applications that are developed using the low level `wwd` API calls and do not rely on higher level WICED APIs. Located within each subdirectory in the *apps* folder is a *README.txt* that lists and summarizes the applications located within the folder. Note that not all applications are supported in all platforms. The *snip* directory contains a *README.txt* with a matrix on what applications are supported in what platforms. For more details on the WICED software stack and APIs, review the application notes and documents available in the doc folder *<WICED SDK installation folder>/WICED-Studio-6.1/43xxx_Wi-Fi/doc. WICED-QSG.pdf,* available in the same path, is a good document to start with.

Figure 1-7. Filter for WiFi Code Example in WICED Studio

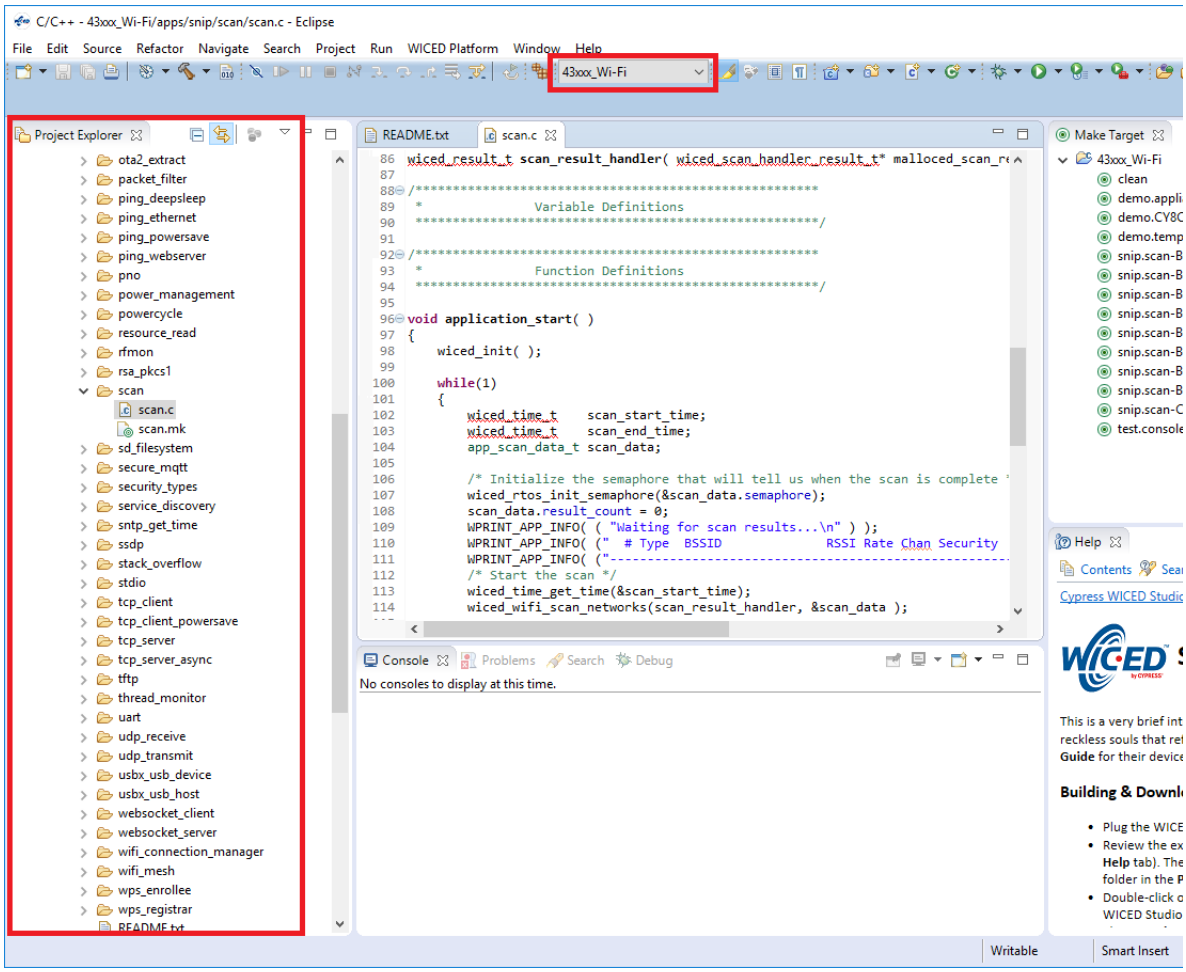
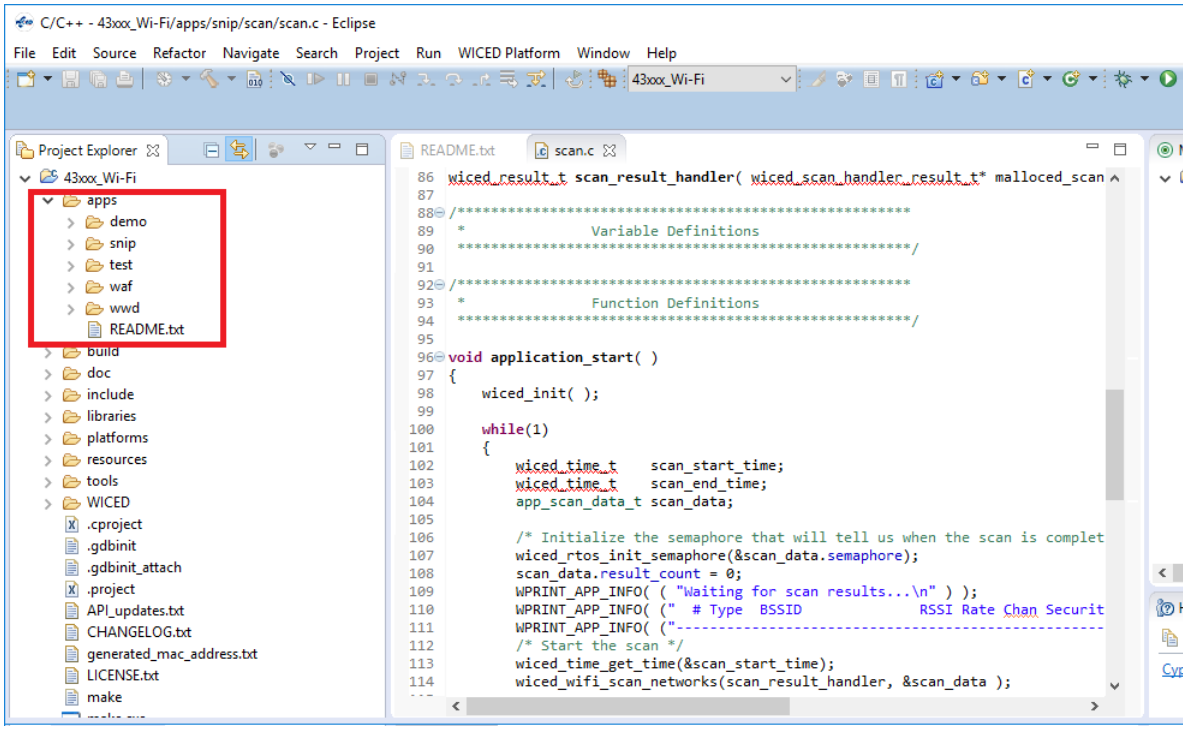


Figure 1-8. Code Examples under *apps* Category



1.4 Getting Started

This guide will help you to get acquainted with the PSoC 6 WiFi-BT Pioneer Kit:

- The PSoC Creator chapter on page 18 describes the installation of the kit software. The kit software includes the PSoC Creator IDE and PDL to develop and debug the applications, the PSoC Programmer to program the *.hex* files on to the device.
- The WICED chapter on page 24 describes the installation of WICED and getting started guidelines of the WICED-based example project of the kit.
- The Kit Hardware chapter on page 38 describes the CY8CKIT-062-WiFi-BT base board and CY8CKIT-028-TFT shield hardware features and functionalities.
- The Appendix on page 58 provides a detailed hardware description, methods to use the onboard components, kit schematics, the bill of materials (BOM), and an FAQ.

1.5 Additional Learning Resources

Cypress provides a wealth of data at www.cypress.com/psoc6 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.

1.6 Technical Support

For assistance, visit Cypress Support or contact customer support at +1(800) 541-4736 Ext. 3 (in the USA) or +1 (408) 943-2600 Ext. 3 (International).

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

- Self-help (Technical Documents)
- Local Sales Office Locations

1.7 Documentation Conventions

Table 1-1. Document Conventions for Guides

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

1.8 Acronyms

Table 1-2. Acronyms Used in this Document

Acronym	Definition
ADC	analog-to-digital converter
BOM	bill of materials
BT	Bluetooth
CINT	integration capacitor
CMOD	modulator capacitor
CPU	central processing unit
CSD	CapSense sigma delta
CTANK	shield tank capacitor
DC	direct current
Del-Sig	delta-sigma
ECO	external crystal oscillator
ESD	electrostatic discharge
F-RAM	Ferroelectric Random Access Memory
FPC	flexible printed circuit
GPIO	general-purpose input/output
HID	human interface device
I^2C	Inter-Integrated Circuit
IC	integrated circuit
ICSP	in-circuit serial programming
IDAC	current digital-to-analog converter
IDE	integrated development environment
LED	light-emitting diode

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

Acronym	Definition
PC	personal computer
PCM	pulse code modulation
PD	power delivery
PDM	pulse density modulation
PTC	positive temperature coefficient
PSoC	Programmable System-on-Chip
PWM	pulse width modulation
RGB	red green blue
SAR	successive approximation register
SMIF	serial memory interfac
SPI	serial peripheral interface
SRAM	serial random access memory
SWD	serial wire debug
TFT	thin-film transistor
UART	universal asynchronous receiver transmitter
USB	Universal Serial Bus
WCO	watch crystal oscillator

2. PSoC Creator

This chapter describes the steps to install PSoC Creator and the packages required to use the PSoC 6 WiFi-BT Pioneer Kit, for non WiFi applications. This includes the IDE on which the projects will be built and used for programming. For developing WiFi applications, WICED Studio 6.1 or later must be used (see the WICED chapter on page 24).

2.1 Before You Begin

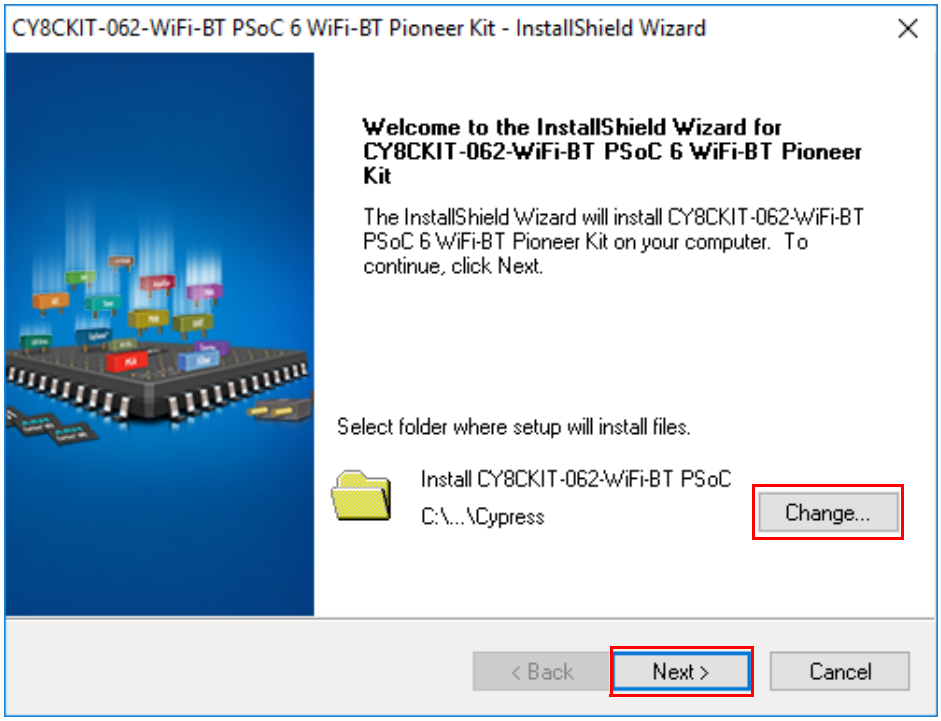
To install Cypress software, you will require administrator privileges. However, they are not required to run the software that is already installed. Before you install the kit software, close any other Cypress software that is currently running.

2.2 Install Kit Software

Follow these steps to install the PSoC 6 WiFi-BT Pioneer Kit software:

1. Download and run the CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit software from www.cypress.com/CY8CKIT-062-WiFi-BT. The kit software is available in two different formats for download.
   a. **CY8CKIT-062-WiFi-BT Kit Complete Setup**: This installation package contains the files related to the kit including PSoC Creator, PSoC Programmer, and PDL. However, it does not include the Windows Installer or Microsoft .NET framework packages. If these packages are not on your computer, the installer will direct you to download and install them from the Internet.
   b. **CY8CKIT-062-WiFi-BT 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 3) are installed on your PC.
2. Select the folder in which you want to install the PSoC 6 WiFi-BT Pioneer Kit-related files. Choose the directory and click **Next**.

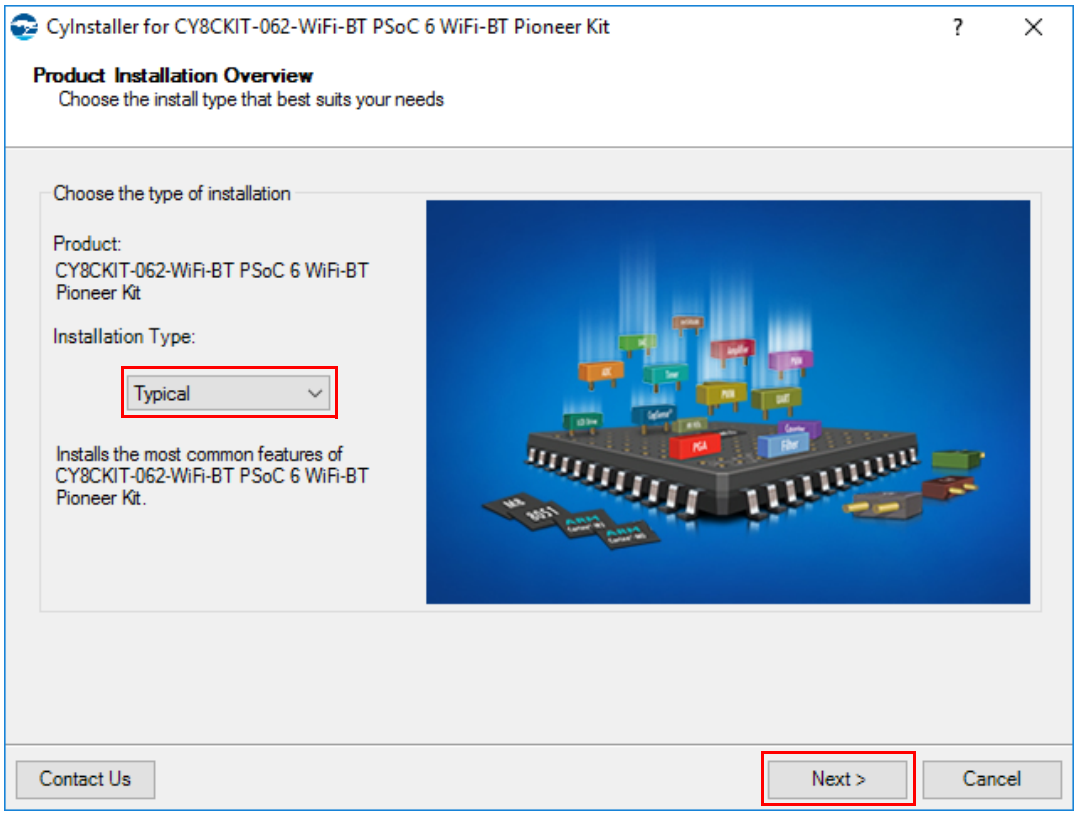
Figure 2-1. Kit Installer Screen



3. When you click **Next**, the installer automatically installs the required software, if it is not present on your computer. Following are the required software:
   a. PSoC Creator 4.2: This software is available for download separately at www.cypress.com/psoccreator. PSoC Creator 4.2 installer automatically installs the following additional software:
   PSoC Programmer 3.27.0
   Peripheral Driver Library 2.1.0
   Peripheral Driver Library 3.0.1

4. Choose the **Typical**, **Custom**, or **Complete** installation type (select **Typical** if you do not know which one to select) in the Product Installation Overview window, as shown in Figure 2-2. Click **Next** after selecting the installation type.

Figure 2-2. Product Installation Overview



5. Read the License agreement and select **I accept the terms in the license agreement** to continue with installation. Click **Next**.
6. 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.
7. Enter your contact information or select the check box **Continue Without Contact Information**. Click **Finish** to complete the CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit software installation.
8. After the installation is complete, the kit contents are available at the following location: *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit*

   Default location:

   Windows 7 (64-bit): *C:\Program Files (x86)\Cypress\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit*

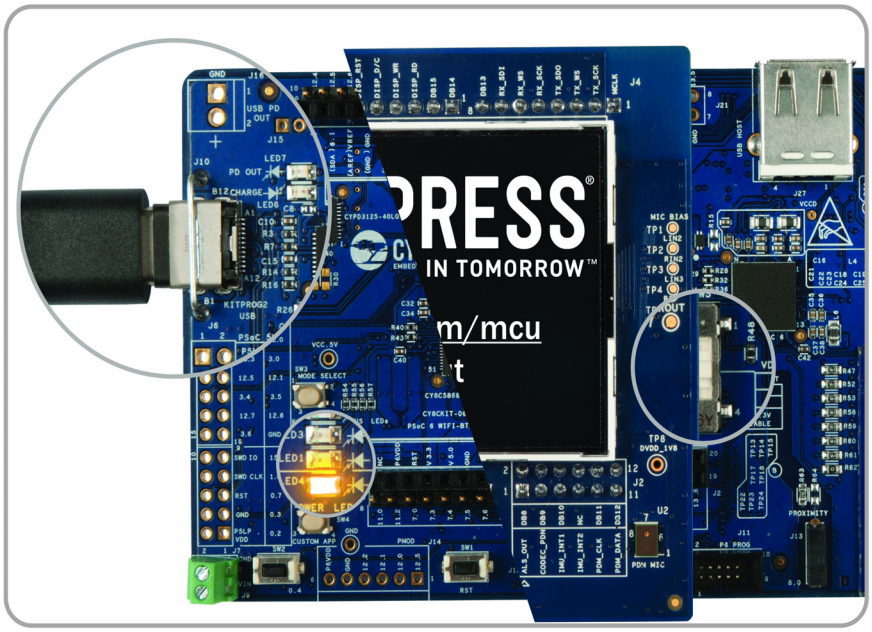
   Windows 7 (32-bit): *C:\Program Files\Cypress\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit*

Note: For Windows 7/8/8.1/10 users, the installed files and the folder are read-only. To use the installed code examples, create an editable copy of the example in a path that you choose, so that the original installed example is not modified. These steps will create an editable copy of the example in a path that you choose, so the original installed example is not modified.

2.3 Programming and Debugging using PSoC Creator

1. Connect the PSoC 6 WiFi-BT Pioneer Kit to the PC using the USB cable, as shown in Figure 2-3. The kit enumerates as a composite device if you are connecting it to your PC for the first time. See section 4.3.3 Kit Enumeration and Programming Modes of KitProg2 to know whether the kit is successfully enumerated or not.

Figure 2-3. Connect USB Cable to USB Connector on the Kit



2. Open the desired project in PSoC Creator. To do this, go to **File** > **Open** > **Project/Workspace**. This provides the option to browse and open your saved project.
3. Select the option **Build** > **Build Project** or pressing **[Shift] [F6]** to build the project.
4. If there are no errors during build, select **Debug** > **Program** or press **[Ctrl] [F5]**. This programs the device on the PSoC 6 WiFi-BT Pioneer Kit.

PSoC Creator has an integrated debugger. You can start the debugger by selecting **Debug** > **Debug** or by pressing **[F5]**. For more details, see the "Debugging Using PSoC Creator" section in the KitProg2 User Guide.

2.4 Kit Code Examples

The PSoC 6 WiFi-BT Pioneer Kit includes two code examples. One of these two code examples is developed in PSoC Creator. To access this code example, first download and install the PSoC 6 WiFi-BT Pioneer Kit setup file from www.cypress.com/CY8CKIT-062-WiFi-BT. After the kit package is installed on your PC, the PSoC Creator-based code examples will be available from **Start** > **Kits** on the PSoC Creator Start Page. This code example can be accessed in the following directory: *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\<version>\Firmware\PSoC 6 MCU\CE222221.*

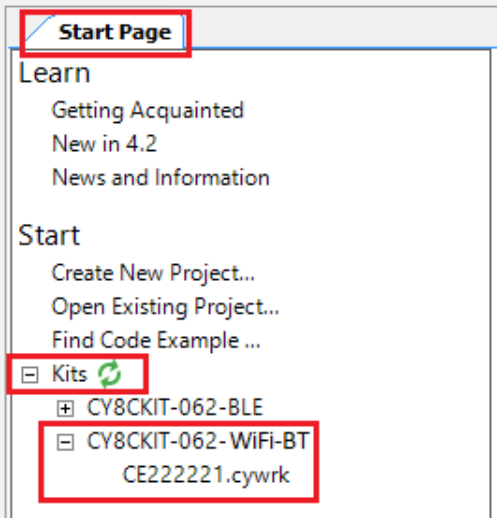
2.4.1 Using the Kit Code Examples Built in PSoC Creator

Follow these steps to open and use the code examples:

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

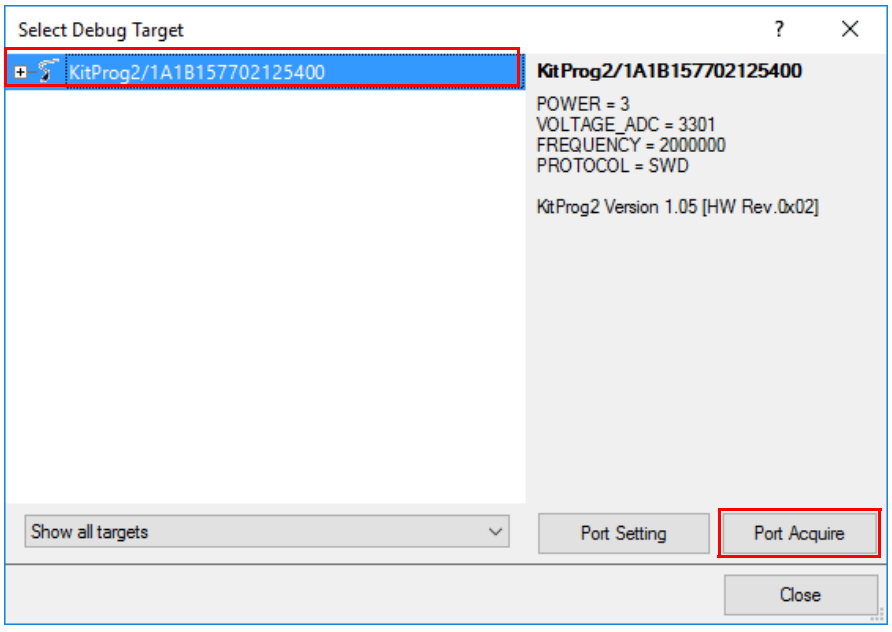
2. On the Start Page, click **CY8CKIT-062-WiFi-BT** under **Start** > **Kits**. A list of code examples appears, as shown in Figure 2-4.
3. Click the desired code example, select a location to save the project, and click **OK**.

Figure 2-4. Open Code Example from PSoC Creator



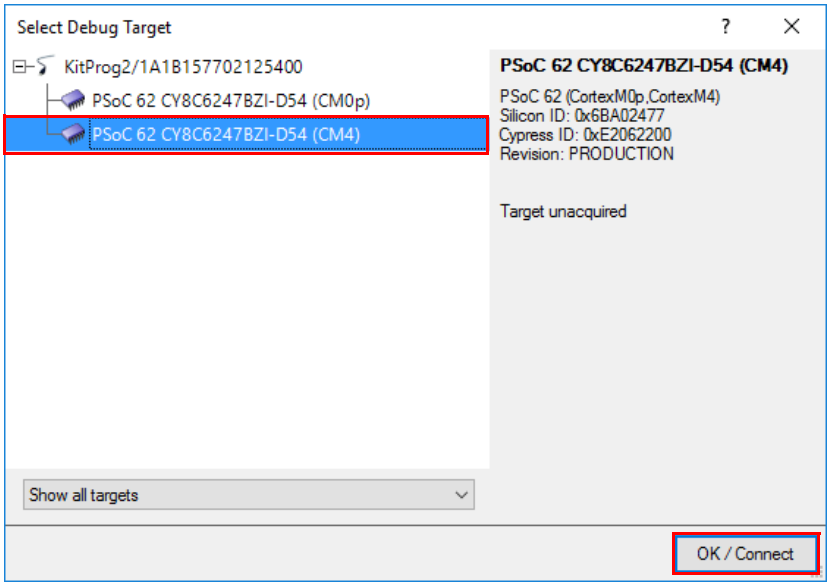
4. Build the code example by choosing **Build** > **Build <Project Name>**. After the build process is successful, a *.hex* file is generated.
5. Connect PSoC 6 WiFi-BT Pioneer Kit to the PC using the USB cable, as shown in Figure 2-3 on page 21, to program the kit with the code example.
6. Choose **Debug** > **Program** in PSoC Creator.
7. If the device is already acquired, programming will complete automatically – the result will appear in the PSoC Creator status bar at the bottom left side of the screen. If the device is yet to be acquired, the Select Debug Target window will appear. Select **KitProg2/<serial_number>** and click **Port Acquire**, as shown in Figure 2-5.

Figure 2-5. Port Acquire



8. After the device is acquired, it is shown in a tree structure below the **KitProg2/<serial_number>**. Click **Connect** and then **OK** to exit the window and start programming, as shown in Figure 2-6. **Note:** PSoC 6 MCUs have both CM0+ and CM4 CPUs. To program, select one of them and click **Connect**. To debug, select the CPU that needs to be debugged.

Figure 2-6. Connect Device from PSoC Creator and Program



9. After programming is successful, the code example is ready to use.

Table 2-1 shows the code example, developed in PSoC Creator, which can be used with this kit.

Table 2-1. Code Example in PSoC Creator

Project	Description
CE222221_TFT_VoiceRecorder	This code example shows how PSoC 6 MCU can be used to record audio data, store it, and play it back. It uses a digital microphone with the PDM/PCM hardware block. All the audio data captured by the microphone is stored in an external flash memory. After the recording is completed, you can play the audio data over I2S, which interfaces with an audio codec. You can record/play/pause/resume with CapSense buttons. You control the audio volume with a CapSense slider. The TFT LCD displays the current state of the voice recorder, the volume, and the time of the record/play.

3. WICED

3.1 Introduction

This chapter describes the steps to install the WICED software tools and packages required to use the PSoC 6 WiFi-BT Pioneer Kit for developing WiFi applications. This chapter describes basic quick start guidelines of the WICED-based example project that can be used for further development purposes.

3.2 Before You Begin

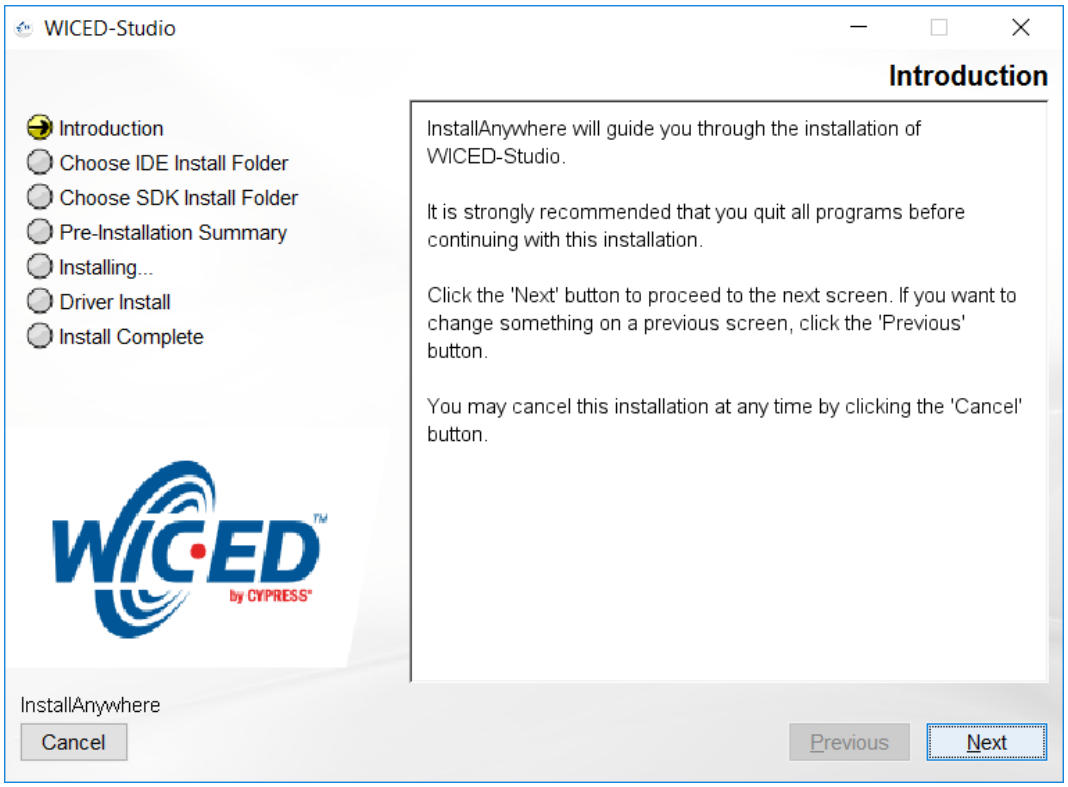
To install Cypress software, you will require administrator privileges. However, they are not required to run the software that is already installed. Before you install the kit software, close any other Cypress software that is currently running.

3.3 Install WICED

The CY8CKIT-062-WiFi-BT kit needs the WICED 6.1 (or later) software to be installed in your system. WICED 6.1 needs separate download and installation.

1. Download and install WICED Studio 6.1 (or later) from www.cypress.com/products/wiced-software.
2. Select two folders, one for the IDE and the other for the SDK. The SDK folder contains the Framework for developing WiFi applications.
3. The installer will ask to select between WiFi and Bluetooth platforms. Select **43xxx_Wi-Fi** as the default.

Figure 3-1. Installer Window Screenshot



3.4 Programming and Debugging in WICED

3.4.1 Building and Programming a Project for CY8CKIT-062-WiFi-BT in WICED Studio IDE

To build and program a project for CY8CKIT-062-WiFi-BT, perform the following steps:

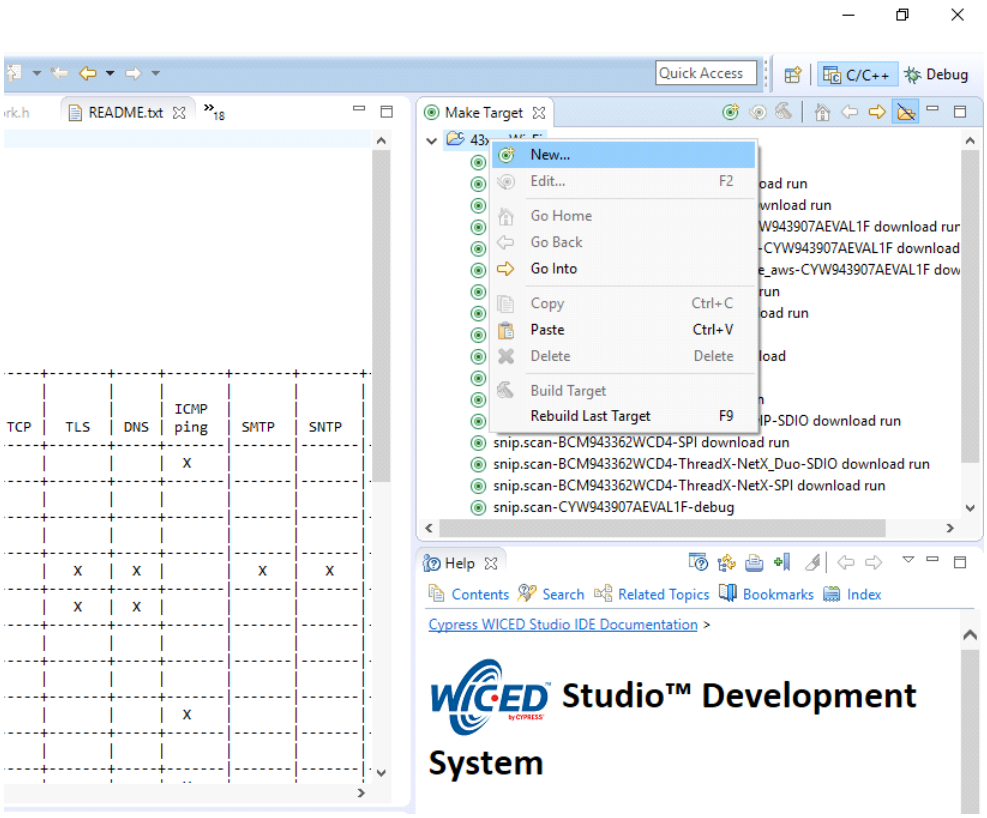
1. To open the WICED IDE on the Windows PC, go to **Start** > **All Programs** > **Cypress** > **WICED-Studio**.
2. Select **43xxx_Wi-Fi** in the WICED Target selector drop-down box as shown in Figure 1-7. Building a project requires a corresponding make target, located in the Make Target window. All applications go under the *apps* directory. The make target path will contain the directory hierarchy starting from *apps* with directory names separated by a period. The project name is followed by a hyphen and then the platform name. Finally, the actions to be performed after the build are specified, such as download and run. For example, to build, download, and run the application scan which exists in *apps**snip**scan*, create the following make target:

   ```
   snip.scan-CY8CKIT_062 download_apps download run
   ```

   This project will periodically scan for Wi-Fi access points and will list them using the serial-to-USB connection on the kit.

   Perform these steps to create the make target, build, program, and test application scan:
3. Right-click **43xxx_Wi-Fi** in the Make Target window as shown in Figure 3-2 and click **New**.

Figure 3-2. Creating New Make Target



4. Enter `snip.scan-CY8CKIT_062 download_apps download run` in the **Target name** field and click **OK**.

   Note: The list of all commands that can be provided in the Make target is listed in <*WICED-SDK installation directory>/ 43xxx_Wi-Fi/Makefile*.

   `snip.scan-CY8CKIT_062 download_apps download run` indicates the following:

   a. `snip` = Directory inside *apps* folder

   b. `scan` = Sub-directory and name of the application to be built. For example, to build the console application under test directory in apps, then use `test.console` instead of snip.scan.

   c. CY8CKIT_062 = Board/platform name

   d. download_apps = Download application resources into QSPI Flash

   e. `download` = Indicates download to target

   f. `run` = Resets the target and starts execution

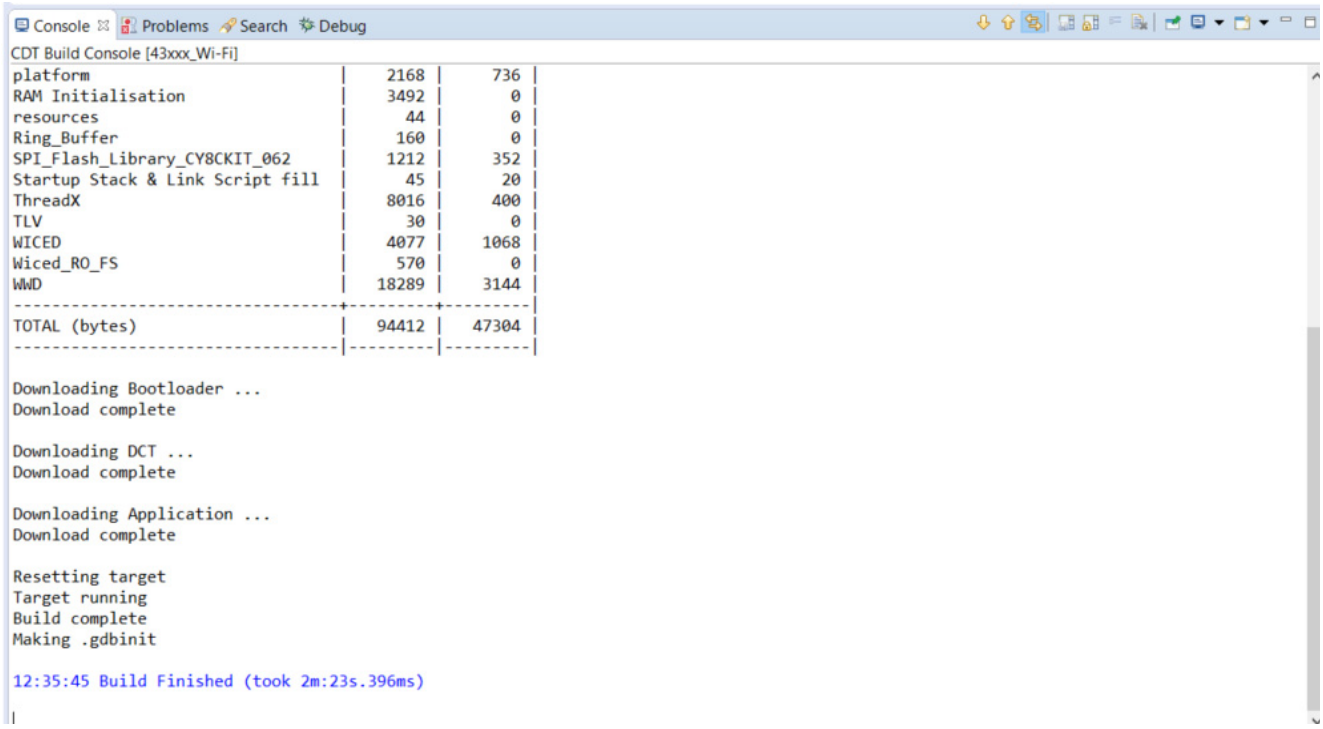
5. Double-click (alternately, right-click and select **Build Target**) the `Clean` Make Target to remove any output from the previous build. It is recommended to do Make clean when any new files are added or removed to the corresponding Target.

   Note: Before executing the next step, ensure that you connect the CY8CKIT-062-WiFi-BT kit to the same PC through the Type-C USB cable connected to the J10 port. See 4.3.3 Kit Enumeration and Programming Modes of KitProg2 to ensure that the kit is successfully enumerated and **Port Selection** is set to CMSIS-DAP mode.

6. Double-click (alternatively right-click and select **Build Target**) the `snip.scan-CY8CKIT_062 download_apps download run` make target to build and download it to the CY8CKIT-062-WiFi-BT.

   The project is built and programmed into the CY8CKIT-062-WiFi-BT, as shown in Figure 3-3.

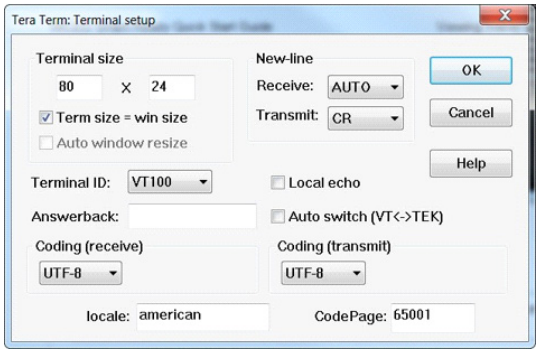
Figure 3-3. Successful Build and Program



7. To view output messages with a terminal emulation program (such as Tera Term), follow these steps:

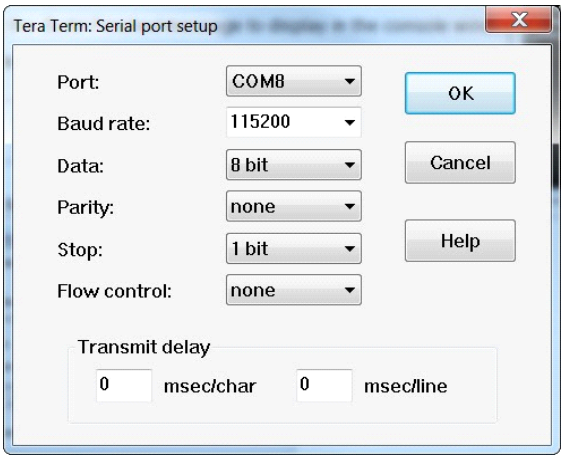
   a. Start the terminal emulation program.

   b. Go to **Setup** > **Terminal**, set the Terminal ID to **VT100** and New-Line Receive to **AUTO**. Other settings should be left at the default settings.



   c. Go to **Setup** > **Serial port**, initiate a connection with the Serial port number from the Device Manager on the PC.

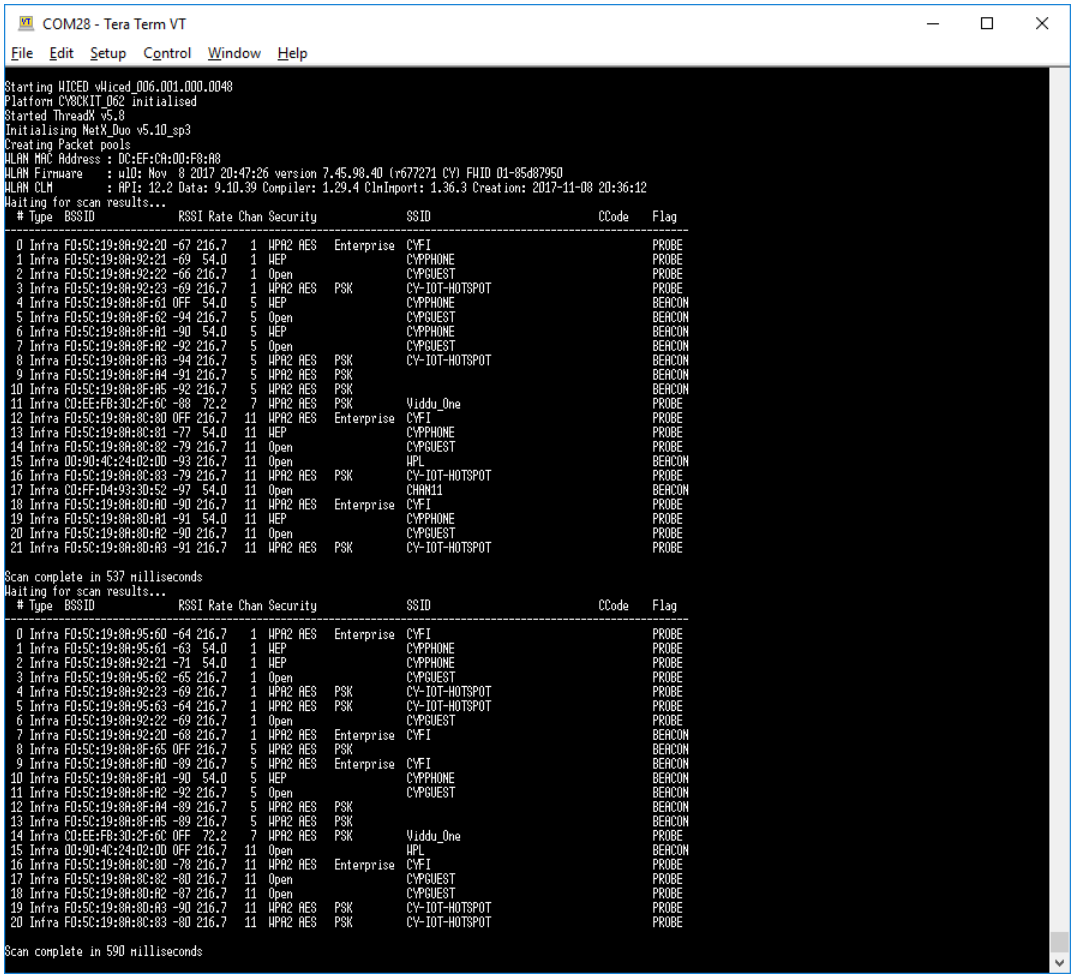
Note: The exact Port number will vary with the corresponding PC port.



d. Press the Reset button on the CY8CKIT-062-WiFi-BT to view the application start-up messages.

8. The output of the Terminal Emulation program should be similar to what is shown in Figure 3-4.

Figure 3-4. Console Output



3.4.2 Debugging a Project using Breakpoints

After programming a project, it is possible to debug it in CY8CKIT-062-WiFi-BT using the built-in debugger.

Note that the scan example used in section 3.4.1 is also used here. The steps outlined in section 3.4.1 should be first followed with a slight change (adding `-debug` to the Make Target command and removing run). Instead of

```
snip.scan-CY8CKIT_062 download_apps download run
```

The following make command should be used:

```
snip.scan-CY8CKIT_062-debug download_apps download
```

If `-debug` is not added, then it will be built for release. The important difference between the debug and release configurations is optimization. Debug is built with no optimization and release is built with optimization. It is possible to debug without using `-debug` as well, but with many variables and lines optimized away, many breakpoints may not get hit.

Note that breakpoints must be placed after starting a debug session in WICED Studio 6.1 or later. If there are any breakpoints that were created prior to the start of debug session, their properties must be changed to be enabled for all threads.

Perform these steps to debug the project:

1. Execute the make target described above to download the project to the device.
2. Click the arrow next to the **Debug** icon as shown in Figure 3-5 and select 43xxx-Wi-Fi_Debug_Windows. The Confirm Perspective Switch dialog appears; click **Yes**. The Debug session starts and halts in the *start_GCC.s* file.

   Notes:

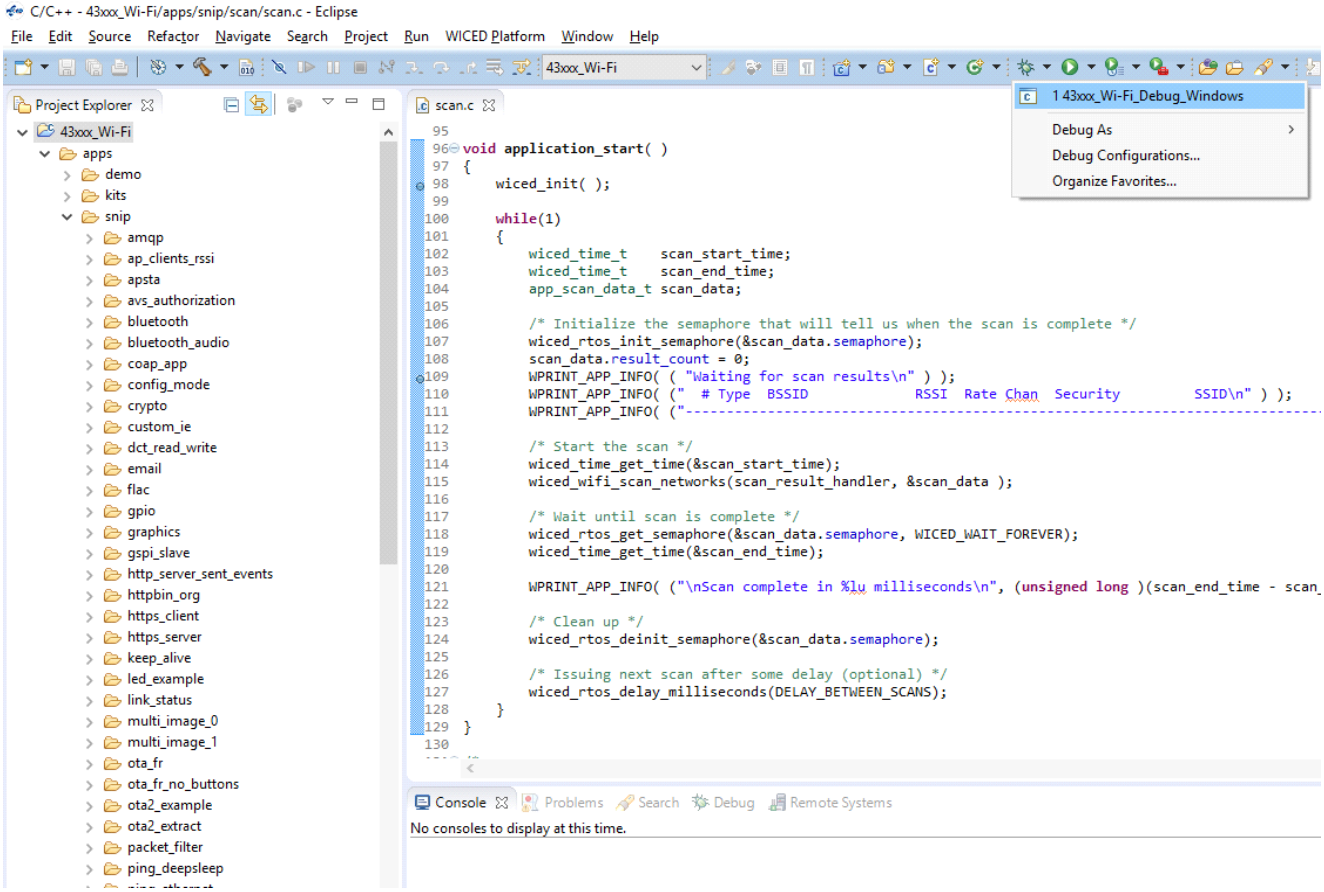
   a. The Confirm Perspective Switch dialog is not displayed if you previously selected the **Remember my decision** check box in the Confirm Perspective Switch dialog.

   b. If any MakeFile/Build error occurs, then clean (using the Clean make target), re-build, and download to the CY8CKIT-062-WiFi-BT again.

   c. In the Debug Perspective, the Project explorer window goes away by default. To view the source files, switch back to the "C/C++" perspective.

   d. To switch between perspectives use the "C/C++" or "Debug" icon at the top right corner of screen.

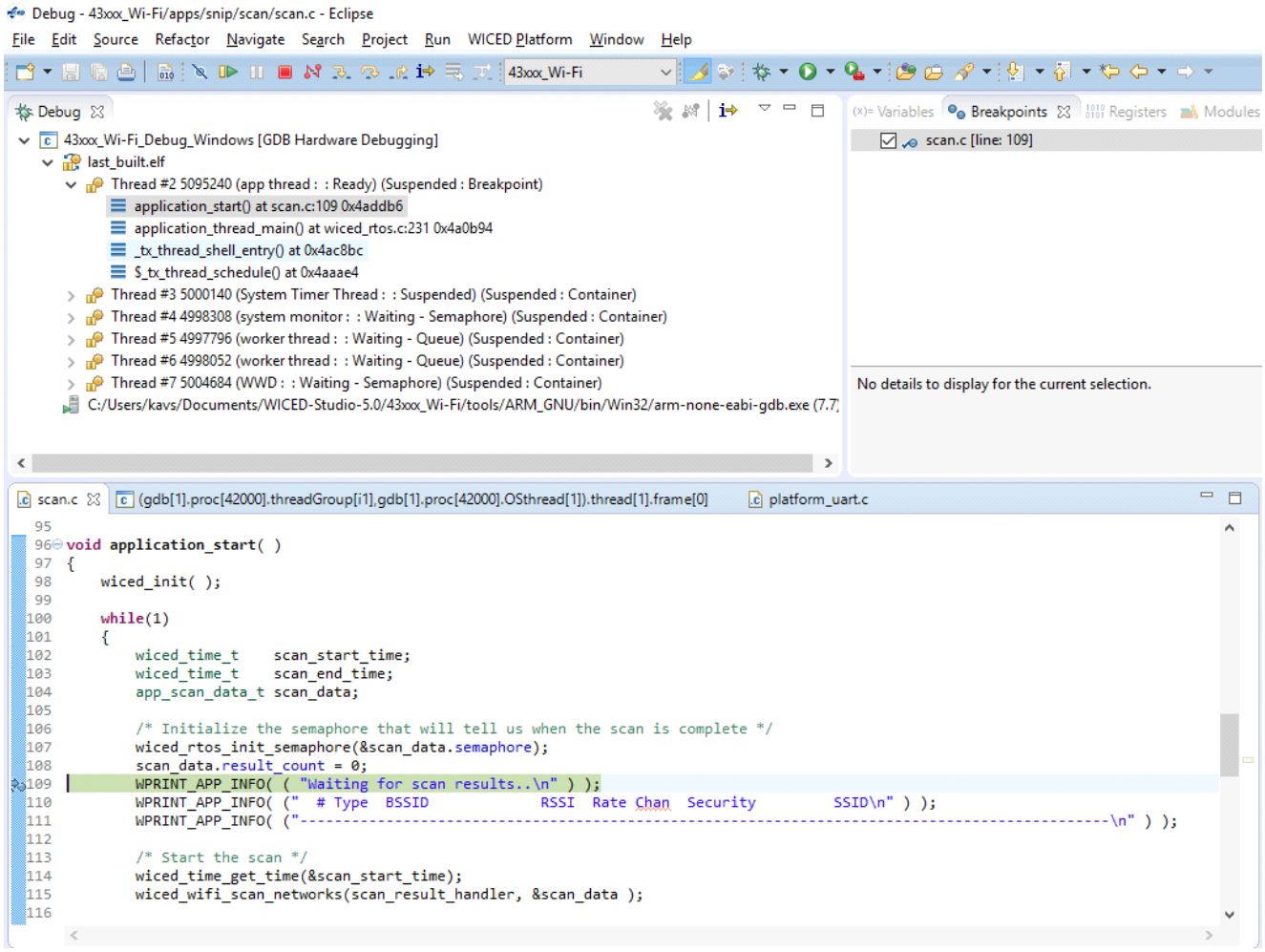
Figure 3-5. Debugging Project



3. Open the *scan.c* file from the Project Explorer window. Click the line with `WPRINT_APP_INFO( ( "Waiting for scan results...\n" ) );` and press the **[Ctrl +Shift+B]** keys on your keyboard. A blue hollow circle along with a check mark appears next to the line number, as shown in Figure 3-6.
4. From the main menu, click **Run** > **Resume**. Execution will stop at the breakpoint that you added. To continue after hitting the breakpoint, click **Resume** again.
5. To disable the breakpoint, press the **[Ctrl+Shift+B]** keys again on the same line, or deselect the corresponding check box in the Breakpoints window.

   Note: If the Breakpoint window does not appear, then choose **Window** > **Show View** > **Breakpoints**.
6. To terminate the Debugging session, click **Run** > **Terminate**, or click on the red Square icon. Once you terminate the session, click **C/C++** in the upper right corner to return to the C/C++ perspective.

Figure 3-6. Placing Breakpoint in Code



7. If Breakpoints are created prior to starting the current Debug session, they will not be associated with the current thread and will be indicated with a Blue circle without a check mark. To enable the Breakpoints in the current thread, associate the properties from the Breakpoints window with the current thread.

   Note: If you do not see any breakpoints in the Breakpoints window, click the **Show Breakpoints Supported by Selected Target** icon as shown in Figure 3-7. The breakpoints are displayed.
8. Right-click the desired breakpoint check box and click **Breakpoint Properties…**. Click the **last_built.elf** check box, as shown in Figure 3-8. The check mark appears before the actual breakpoint indicating its association with the current execution.

Figure 3-7. Show Breakpoints Icon

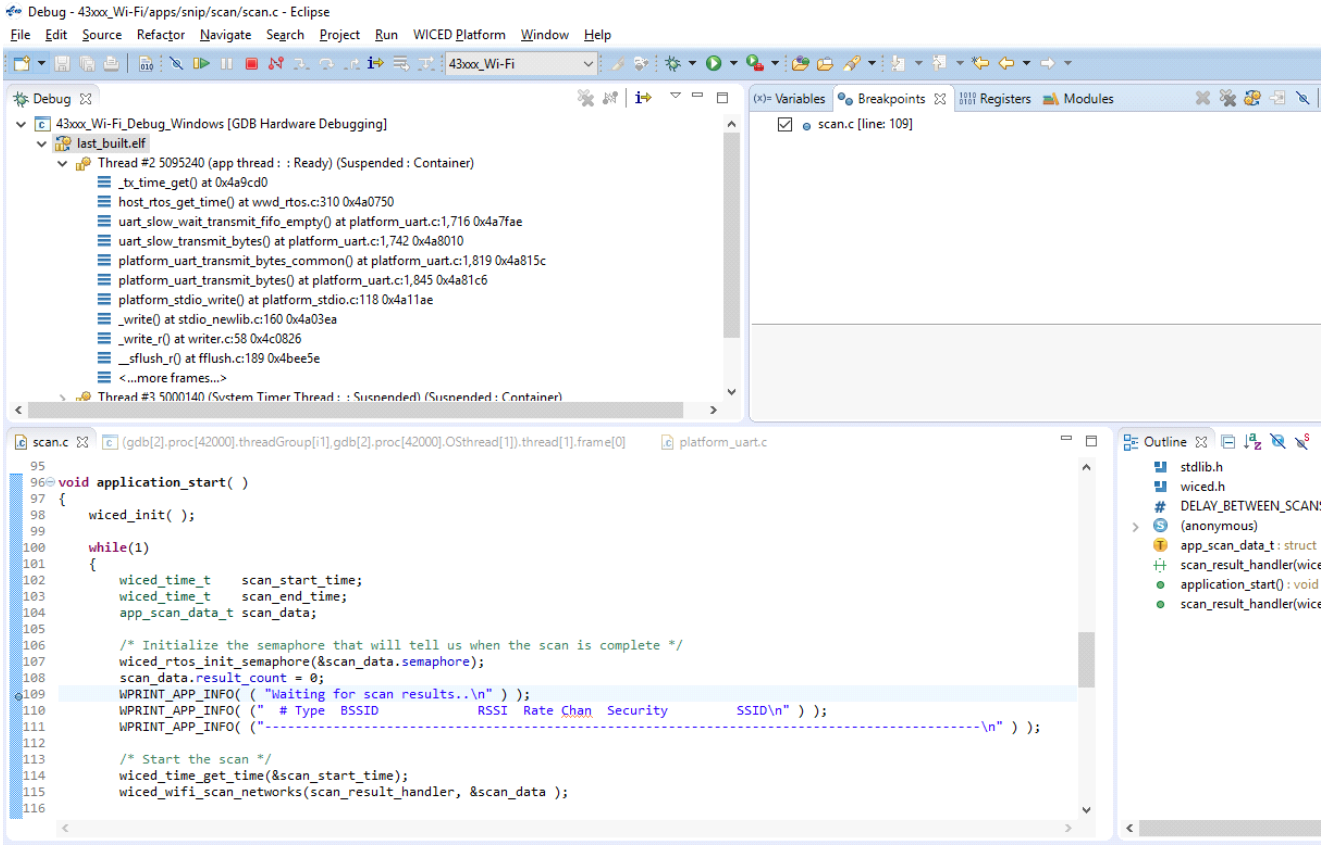
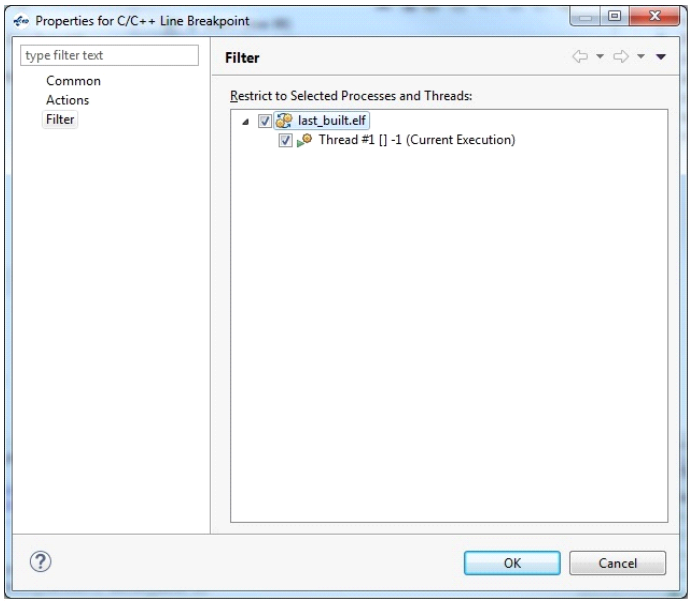


Figure 3-8. Enabling Breakpoint for Current Execution



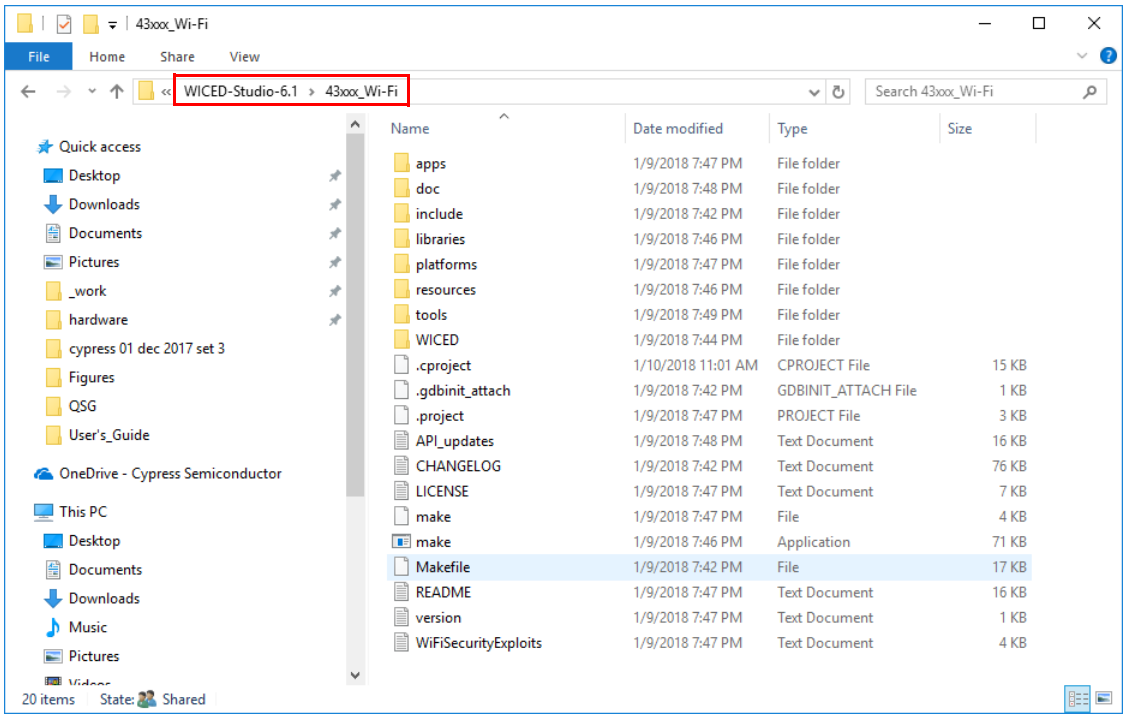
3.5 Kit Code Example

3.5.1 Building and Programming a Project in the WICED Studio IDE

Before starting with the kit code example, make sure you install the kit software and WICED 6.1 to your PC. The steps to install the kit installer is explained in Install Kit Software on page 18. The installation of WICED 6.1 is explained Install WICED on page 24. To build and program the WICED based project of the CY8CKIT-062-WiFi-BT kit, perform the following steps:

1. To open the WICED IDE on Windows PC, go to **Start** > **All Programs** > **Cypress** > **WICED-Studio**.
2. Locate the WICED WiFi-SDK directory in your PC. The default location is *C:\Users\<user name>\Documents\WICED-Studio-6.1\43xxx_WiFi*, as shown in Figure 3-9. However, it may be in a different location depending on the path you choose when installing WICED Studio.

Figure 3-9. WICED SDK Directory



3. Also, locate the CY8CKIT-062-WiFi-BT Kit Code Example at the location *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\<version>\Firmware\PSoC 6 MCU\CE222494_PSoC6_WICED_WiFi_Demo.* This has two directories called *apps* and *resources*. Open the *apps\demo* folder and copy the *CE222494_PSoC6_WICED_WiFi* folder into the *WICED-Studio-6.1\43xxx_Wi-Fi\apps\demo* folder.
4. Open the *resources\apps* folder and copy the *CE222494_PSoC6_WICED_WiFi* folder into *WICED-Studio-6.1\43xxx_Wi-Fi\resources\apps*.

   If WICED Studio 6.1 (or later) is opened with `43xxx_Wi-Fi` as the WICED Filter, then the new folders appear as shown in Figure 3-10.

   If the projects are not visible in WICED Studio 6.1 (or later), then right-click the *43xxx_Wi-Fi* folder in Project Explorer and click **Refresh**, as shown in Figure 3-10.

Figure 3-10. Setup Package in WICED Studio 6.1 (or later)

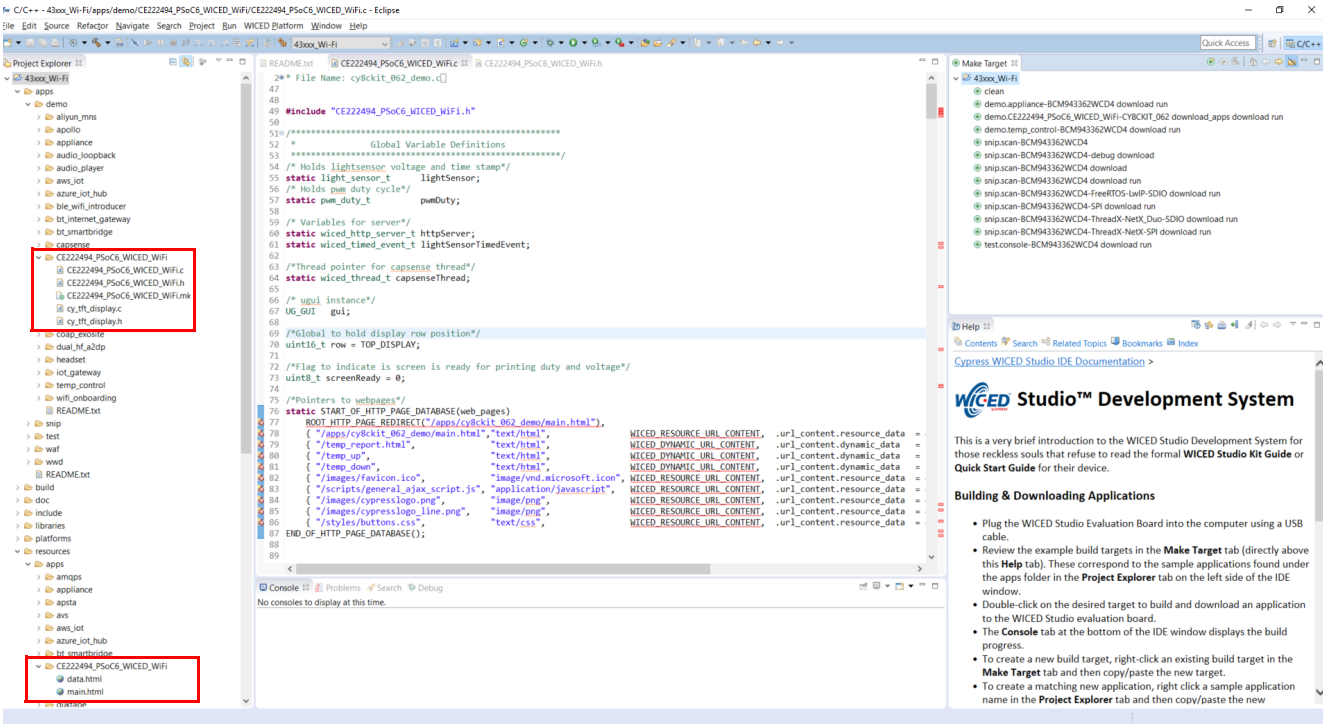
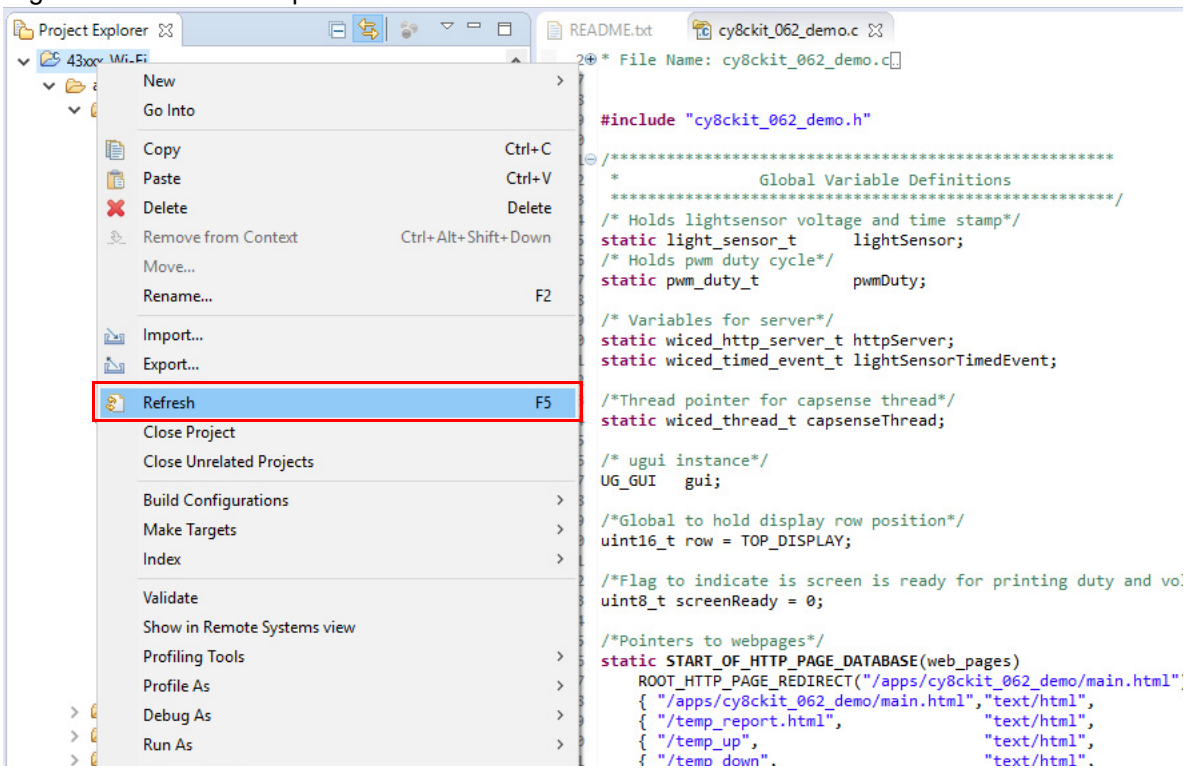
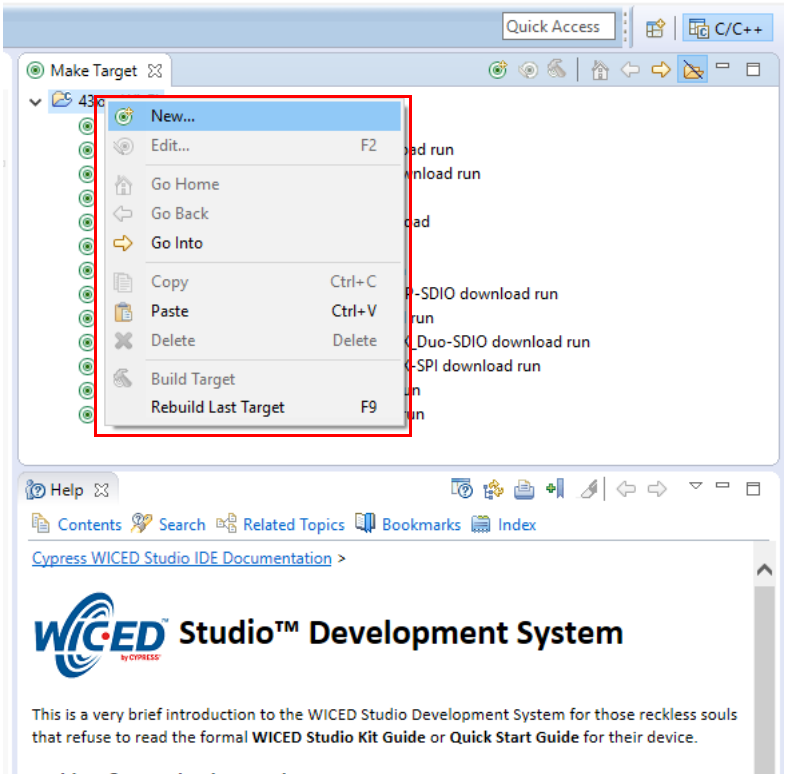


Figure 3-11. Refresh Top Folder



5. In the Make Target window, right-click and select **New**, and give it the following Target Name: `demo.CE222494_PSoC6_WICED_WiFi-CY8CKIT_062 download_apps download run` (see Figure 3-12).

Figure 3-12. Creating New Make Target



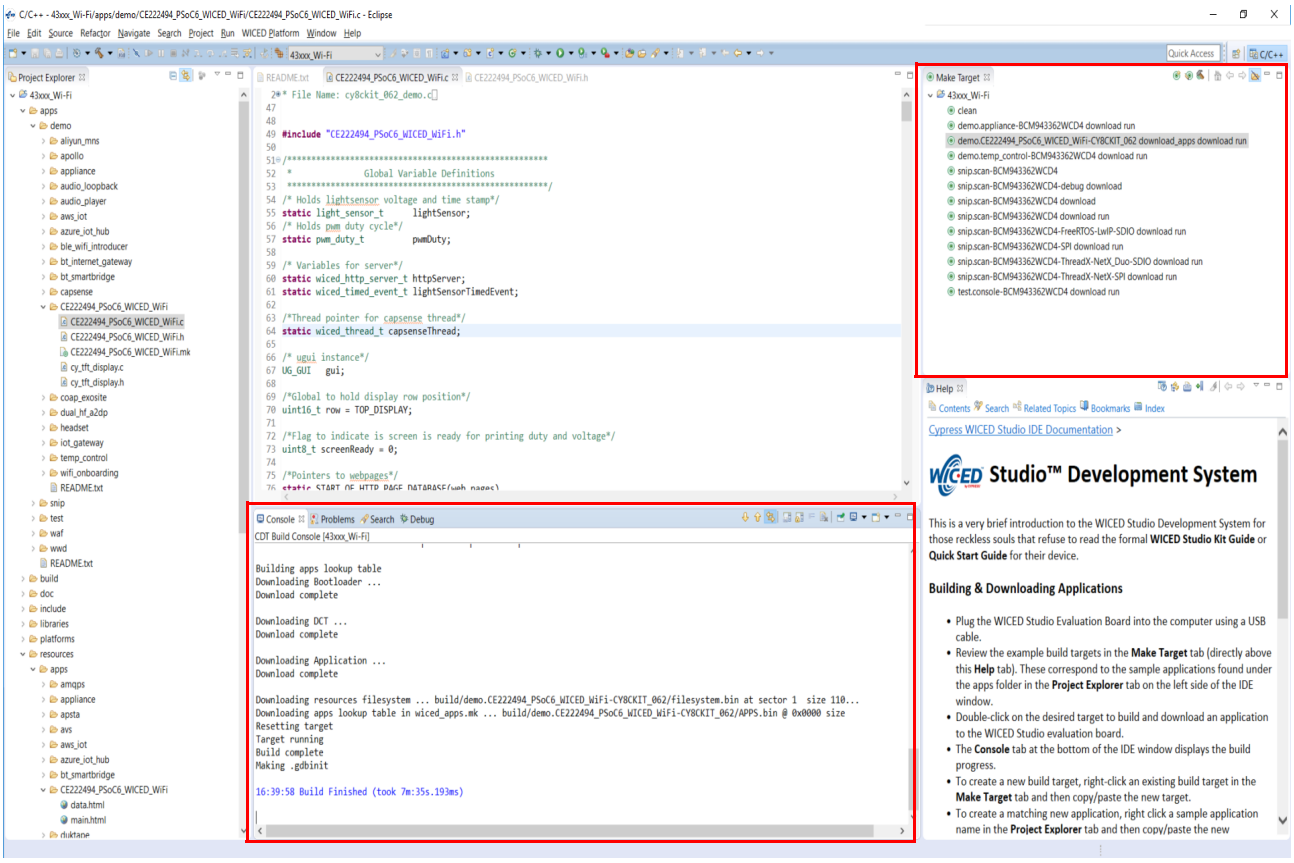
The Make Target window will show the following added target: `demo.CE222494_PSoC6_WICED_WiFi-CY8CKIT_062 download_apps download run` (see Figure 3-13).

Note: Before executing the next step, ensure that you connect the CY8CKIT-062-WiFi-BT kit to the same PC through the Type-C USB cable connected to the J10 port. See 4.3.3 Kit Enumeration and Programming Modes of KitProg2 to ensure that the kit is successfully enumerated and **Port Selection** is set to CMSIS-DAP mode.

6. Double-click the newly created make target to build the code, program the kit, and run the example. Note that this may take a few minutes.

   The Console window will display a message "Build Finished" as shown in Figure 3-13.

Figure 3-13. Building and Programming the Target



To know more about this code example, refer to CE222494_WiFi_BT_WICED_WiFi_Demo.

This kit is shipped with this code example preprogrammed to it. While power is on, and when you peel the sticker on the TFT display, you can see the instructions on the display as shown in Figure 3-14.

Figure 3-14. TFT Screen Instruction Display

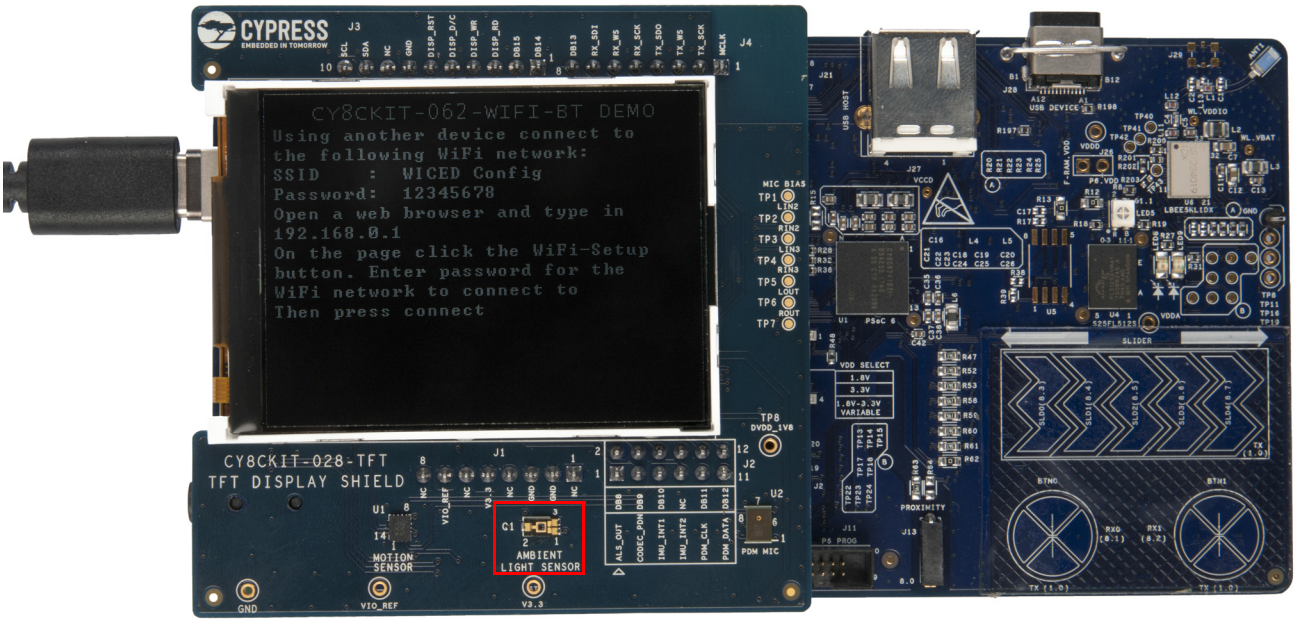


Table 3-1 shows the code example, developed in WICED, which can be used with this kit. See the WICED 6.1 code example documents for additional details.

Table 3-1. Code Example in WICED

Project	Description
CE222494_WiFi_BT_WICED_WiFi_Demo	This code example demonstrates how to use PSoC 6 MCU and WICED to enable WiFi communication. It demonstrates how the PSoC 6 MCU with the 4343W module can be used as a configuration access point (AP) to allow a user to enter the credentials of their personal network.

4. Kit Hardware

4.1 CY8CKIT-062-WiFi-BT Details

The PSoC 6 WiFi-BT Pioneer Kit is built around the PSoC 6 MCU; Figure 4-1 shows the block diagram of the device. For details of the PSoC 6 MCU features, see the device datasheet.

Figure 4-1. PSoC 6 MCU Block Diagram

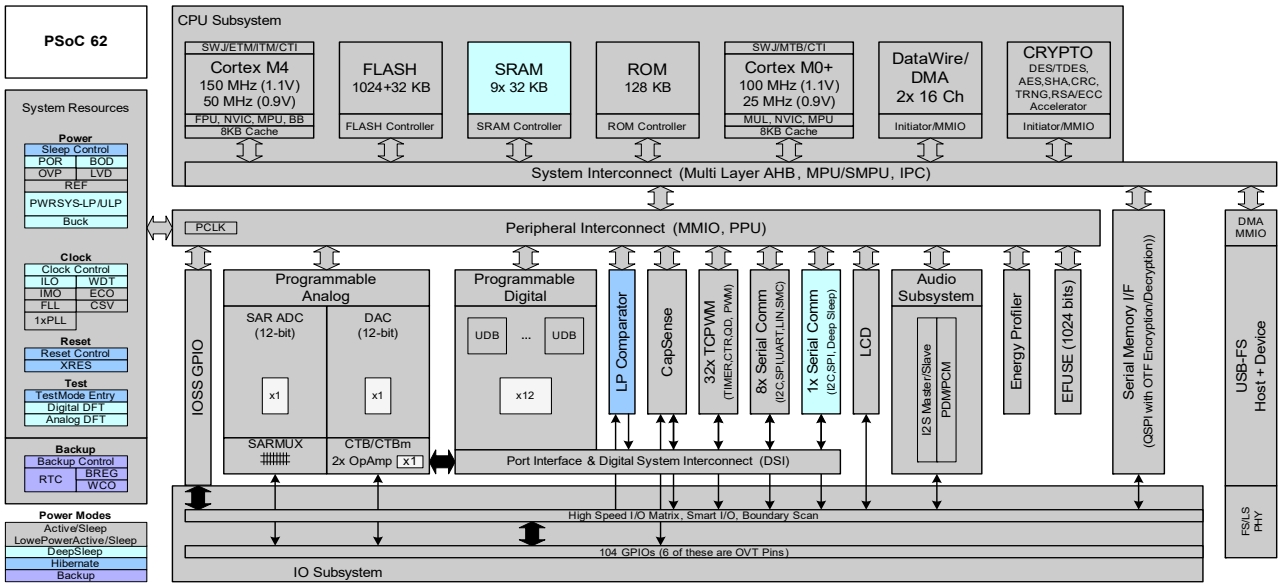
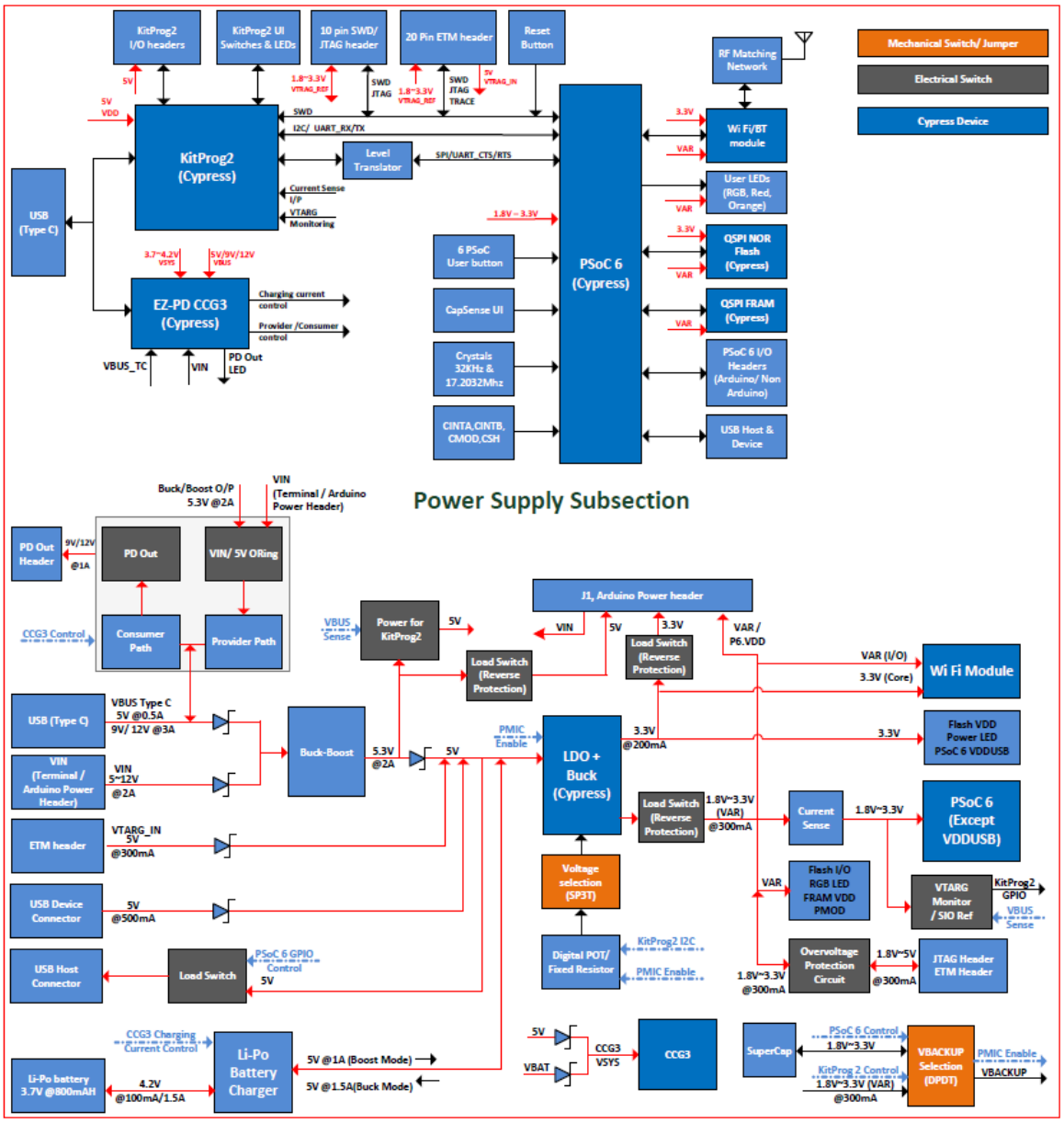


Figure 4-2 shows the block diagram of the Pioneer board.

Figure 4-2. Pioneer Board Block Diagram



The CY8CKIT-062-WiFi-BT Pioneer Kit comes with the PSoC 6 WiFi-BT Pioneer board, which has the CY8CKIT-028-TFT display shield connected, as Figure 4-3 shows.

Figure 4-3. PSoC 6 WiFi-BT Pioneer Board and TFT Display Shield

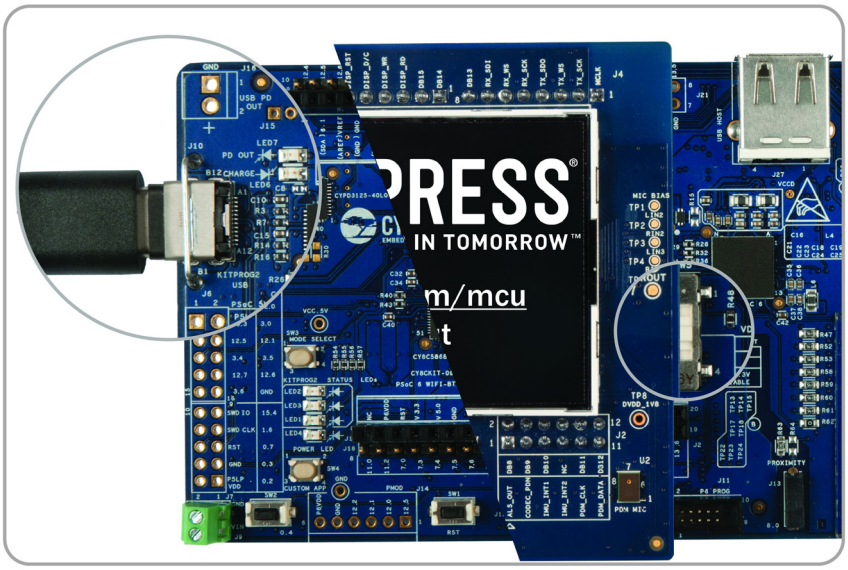
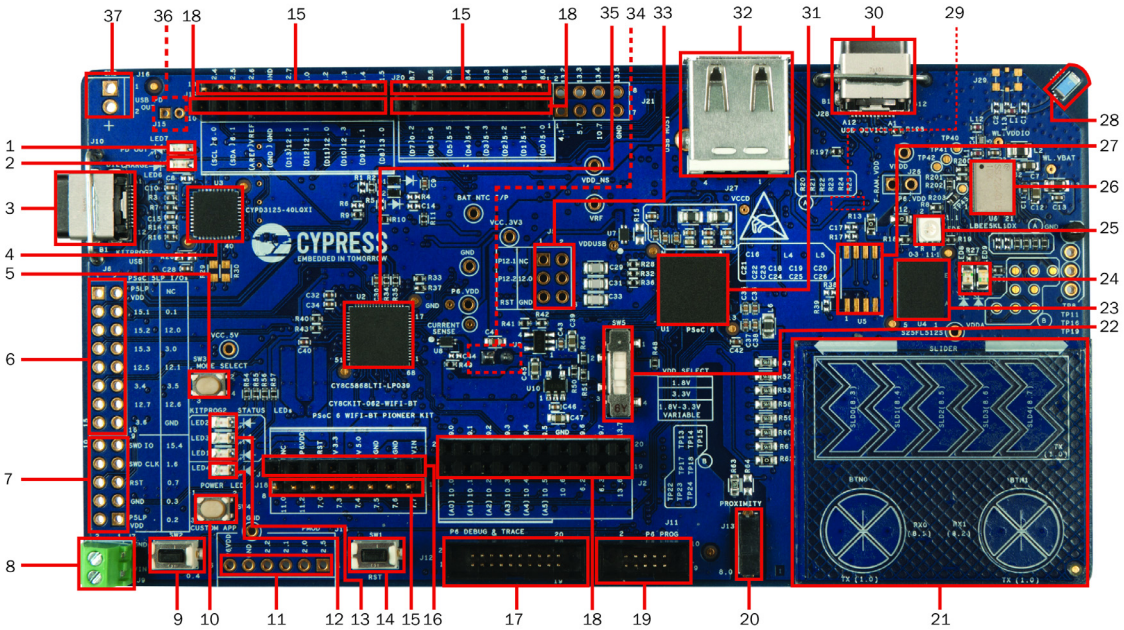


Figure 4-4 shows the markup of the Pioneer board.

Figure 4-4. PSoC 6 WiFi-BT Pioneer Board – Top View



The PSoC 6 WiFi-BT Pioneer board has the following peripherals:

1. **USB PD out indicator (LED7):** This LED turns ON when the USB Type-C power delivery output is available for use.
2. **Battery charging indicator (LED6):** This LED turns ON when the onboard battery charger is charging a lithium-ion polymer battery connected to **J15**. Note that the battery connector and

battery are not included in the kit and should be purchased separately if you want to test the battery charging functionality.

3. **KitProg2 USB connector (J10):** The USB cable provided along with the PSoC 6 WiFi-BT Pioneer Kit connects between this USB connector and the PC to use the KitProg2 onboard programmer and debugger and to provide power to the Pioneer board. **J10** is also used for the USB Type-C power delivery system. See "EZ-PD CCG3 Type-C Power Delivery" **on page 56** for more details.
4. **Cypress EZ-PD CCG3 Type-C Port Controller with PD (CYPD3125-40LQXIT, U3):** The Pioneer board includes an EZ-PD CCG3 USB Type-C port controller with power delivery system. This device is pre-programmed and can deliver power from a Type-C port to an onboard header **J16**, while simultaneously charging a lithium-ion polymer battery connected to **J15**. In addition, the power delivery system can deliver power to a Type-C power sink or consumer, such as a mobile phone, with the power derived from the VIN supply. See "EZ-PD CCG3 Type-C Power Delivery" **on page 56** for more details.
5. **KitProg2 programming button (SW3):** This button can be used to switch between the KitProg2 operation modes (proprietary SWD programming/CMSIS-DAP mode). This button can also be used to provide input to the PSoC 5LP in custom application mode. For more details, see the KitProg2 User Guide.
6. **KitProg2 I/O header (J6):** This header brings out several GPIOs of the onboard KitProg2 PSoC 5LP device. This includes the USB-I2C, USB-UART, and USB-SPI bridge lines. The additional PSoC 5LP pins are direct connections to the internal programmable analog logic of the PSoC 5LP. You can also use these pins for custom applications. For more details on KitProg2, see the KitProg2 User Guide.
7. **KitProg2 programming/custom application header (J7):** This header brings out more GPIOs of the PSoC 5LP, which can be used for custom applications. It also contains a five-pin SWD programming header for the PSoC 5LP.
8. **External power supply VIN connector (J9):** This connector connects an external DC power supply input to the onboard regulators and the USB Type-C power delivery system. The voltage input from the external supply should be between 5 V and 12 V. Moreover, when used as an input to the USB Type-C power delivery system, the external power supply should have enough current capacity to support the load connected via the Type-C port. See "EZ-PD CCG3 Type-C Power Delivery" **on page 56** for more details.
9. **PSoC 6 MCU user button (SW2):** This button can be used to provide an input to PSoC 6 MCU. Note that by default the button connects the PSoC 6 MCU pin to ground when pressed, so you need to configure the PSoC 6 MCU pin as a digital input with resistive pull-up for detecting the button press. This button also provides a wake-up source from low-power modes of the device.
10. **KitProg2 application selection button (SW4):** This button can be used to switch between KitProg2 programming mode and custom application mode. For more details, see the KitProg2 User Guide.
11. **Digilent Pmod-compatible I/O header (J14):** This header can be used to connect Digilent Pmod 1 x 6 pin modules.
12. **Power LED (LED4):** This is the amber LED that indicates the status of power supplied to PSoC 6 MCU.
13. **Kitprog2 status LEDs (LED1, LED2, and LED3):** Red, amber, and green LEDs (LED1, LED2, and LED3 respectively) indicate the status of KitProg2. For more details on the KitProg2 status, see the KitProg2 User Guide.
14. **PSoC 6 MCU reset button (SW1):** This button is used to reset PSoC 6 MCU. It connects the PSoC 6 MCU reset (XRES) pin to ground.
15. **PSoC 6 MCU I/O headers (J18, J19, and J20):** These headers provide connectivity to PSoC 6 MCU GPIOs that are not connected to the Arduino-compatible headers. Most of these pins are

multiplexed with onbroad peripherals and are not connected to PSoC 6 MCU by default. For detailed information on how to rework the kit to access these pins, see Table 4-2 **on page 45**.

16. **Arduino-compatible power header (J1):** This header powers the Arduino shields. It also has a provision to power the kit though the VIN input.
17. **PSoC 6 MCU debug and trace header (J12):** This header can be connected to an Embedded Trace Macrocell (ETM)-compatible programmer/debugger.
18. **Arduino Uno R3-compatible I/O headers (J2, J3, and J4):** These I/O headers bring out pins from PSoC 6 MCU to interface with the Arduino shields. Some of these pins are multiplexed with onboard peripherals and are not connected to PSoC 6 MCU by default. For a detailed information on how to rework the kit to access these pins, see Table 4-2 **on page 45**
19. **PSoC 6 MCU program and debug header (J11):** This 10-pin header allows you to program and debug the PSoC 6 MCU using an external programmer such as MiniProg3. In addition, an external PSoC 4, PSoC 5LP, or PSoC 6 device can be connected to this header and programmed using KitProg2.
20. **CapSense proximity header (J13):** A wire can be connected to this header to evaluate the proximity sensing feature of CapSense.
21. **CapSense slider (SLIDER) and buttons (BTN0 and BTN1):** The CapSense touch-sensing slider and two buttons, all of which are capable of both self-capacitance (CSD) and mutual-capacitance (CSX) operation, allow you to evaluate Cypress' fourth-generation CapSense technology. The slider and buttons have a 1-mm acrylic overlay for smooth touch sensing.
22. **System power V_{DD} selection switch (SW5):** This switch is used to select the PSoC 6 MCU V_{DD} supply voltage between constant 1.8 V, constant 3.3 V, and variable 1.8 to 3.3 V. In the variable 1.8 to 3.3 V mode, the PSoC Programmer software can control the voltage via KitProg2.
23. **Cypress 512-Mbit serial NOR flash memory (S25FL512S, U4):** The S25FL512S NOR flash of 512Mb capacity is connected to the serial memory interface (SMIF) of the PSoC 6 MCU. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption.
24. **PSoC 6 MCU user LEDs (LED8 and LED9):** These two user LEDs can operate at the entire operating voltage range of PSoC 6 MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs.
25. **RGB LED (LED5):** This onboard RGB LED can be controlled by the PSoC 6 MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs.
26. **WiFi and Bluetooth module (LBEE5KL1DX-883, U6):** This kit features the onboard WiFi and Bluetooth combination module to demonstrate the IoT features. The LBEE5KL1DX is a Type 1DX module available with 2.4-GHz WLAN and Bluetooth functionality. Based on Cypress CYW4343W, this module provides high-efficiency RF front-end circuits.
27. **Cypress 4-Mbit serial Ferroelectric RAM (FM25V10, U5):** Footprint to connect a FM25V10 or any other pin-compatible F-RAM.
28. **WiFi-BT antenna (ANT1):** This is the onboard antenna connected to the WiFi and Bluetooth module.
29. **VBACKUP and PMIC control selection switch (SW7, on the bottom side of the board):** This switches the VBACKUP supply connection to PSoC 6 MCU between V_{DDD}/KitProg2 and the Super Cap/PSoC 6. When V_{DDD}/KitProg2 is selected, the regulator ON/OFF is controlled by KitProg2. When the super-capacitor is selected, the regulator ON/OFF is controlled by PSoC 6 MCU.
30. **PSoC 6 USB Type-C connector (J28):** The USB cable provided with the PSoC 6 WiFi-BT Pioneer Kit can also be connected between this USB connector and the PC to use the PSoC 6 MCU USB device applications.

31. **Cypress PSoC 6 MCU (CY8C6247BZI-D54, U1):** This kit is designed to highlight the features of the PSoC 6 MCU. For details on PSoC 6 MCU pin mapping, see Table 4-2 **on page 45**.
32. **PSoC 6 USB Type-A connector (J27):** USB devices can be connected to this USB Type-A connector and communicate with PSoC 6 MCU USB host applications.
33. **Arduino-compatible ICSP header (J5):** This header provides an SPI interface for Arduino ICSP-compatible shields.
34. **PSoC 6 MCU current measurement jumper (J8, on the bottom side of the board):** An ammeter can be connected to this jumper to measure the current consumed by the PSoC 6 MCU.
35. **KitProg2 (PSoC 5LP) programmer and debugger (CY8C5868LTI-LP039, U2):** The PSoC 5LP device (CY8C5868LTI-LP039) serving as KitProg2, is a multi-functional system, which includes a programmer, debugger, USB-I2C bridge, USB-UART bridge, and a USB-SPI bridge. KitProg2 also supports custom applications. For more details, see the KitProg2 User Guide.
36. **Battery connector (J15, on the bottom side of the board):** This connector can be used to connect a lithium-ion polymer battery. Note that a battery is not included in the kit package and should be purchased separately if you want to demonstrate battery charging.
37. **USB PD output (J16):** This header provides a voltage output when the USB Type-C power delivery system receives power from an external host connected to **J10**. See "EZ-PD CCG3 Type-C Power Delivery" **on page 56** for more details.

See "Hardware Functional Description" on page 58 for details on various hardware blocks.

Table 4-1 shows the functionalities of the on-board selection switches.

Table 4-1. Selection Switches on the Pioneer Board

Switch	Location on the Board	Purpose	Default Position
SW5	Front	Selects the V_{DD} supply of the PSoC 6 MCU between 1.8 V, 3.3 V, and the variable 1.8 V to 3.3 V, which is controlled by KitProg2.	3.3 V
SW7	Back	Selects the VBACKUP supply connection of the PSoC 6 MCU between V_{DDD} and the super-capacitor. When V_{DDD} is selected, the regulator can be turned ON/OFF by KitProg2. When the super-capacitor is selected, the PSoC 6 MCU can turn the regulator ON/OFF.	V_{DDD}/KitProg2

Figure 4-5 shows the pinout of the Pioneer board.

Figure 4-5. Pioneer Board Pinout

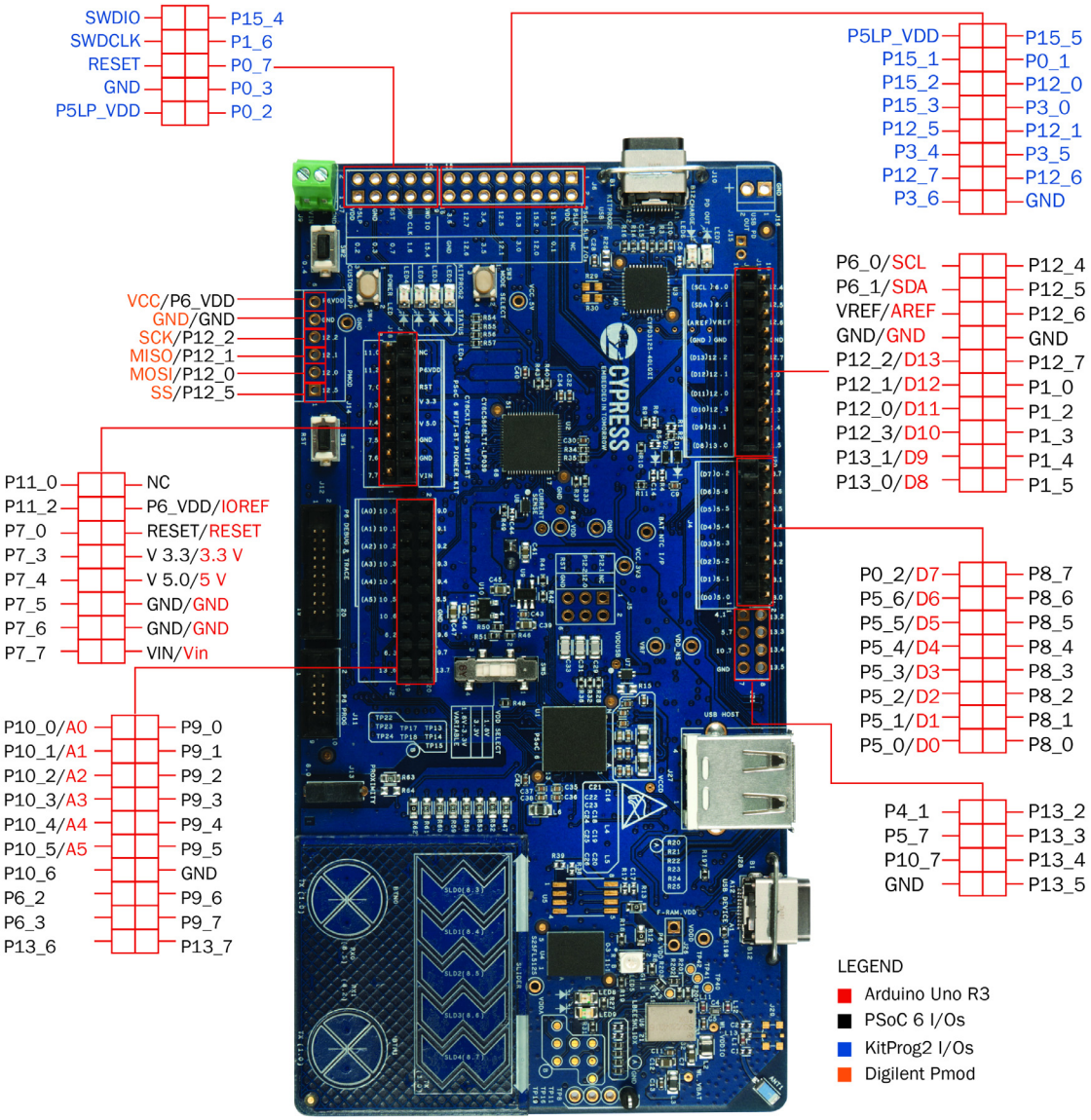


Table 4-2. Pioneer Board Pinout

PSoC 6 Pin	Primary Onboard Function	Secondary Onboard Function	Connection Details
XRES	Reset	–	
P0.0	WCO IN	–	
P0.1	WCO OUT	–	
P0.2	Arduino header J4.8, D7	–	
P0.3	RGB red LED	–	
P0.4	User button with Hibernate wakeup capability	–	
P0.5	PMIC control	–	
P1.0	CapSense Tx	GPIO on non-Arduino header (J19.5)	Populate R174 to connect to the header or remove R62 to disconnect from CapSense.
P1.1	RGB green LED	–	
P1.2	GPIO on non-Arduino header (J19.4)	–	
P1.3	GPIO on non-Arduino header (J19.3)	–	
P1.4	GPIO on non-Arduino header (J19.2)	–	
P1.5	Orange user LED	GPIO on non-Arduino header (J19.1)	Connected to primary and secondary functions by default. Remove R27 to disconnect from LED.
P2.0	SDIO DATA0	J25.7 (WL_JTAG_TMS)	Remove R122 (or R110) and mount R115 to disconnect from PSoC 6 and connect the WiFi/BT module to the JTAG connector.
P2.1	SDIO DATA1	J25.3 (WL_JTAG_TDI)	Remove R32 (or R111) and mount R116 to disconnect from PSoC 6 and connect the WiFi/BT module to the JTAG connector.
P2.2	SDIO DATA2	J25.5 (WL_JTAG_TDO)	Remove R128 (or R112) and mount R117 to disconnect from PSoC 6 and connect the WiFi/BT module to the JTAG connector.
P2.3	SDIO DATA3	J25.1 (WL_JTAG_TRSTN)	Remove R109 (or R132) and mount R114 to disconnect from PSoC 6 and connect the WiFi/BT module to the JTAG connector.
P2.4	SDIO CMD		
P2.5	SDIO CLK	J25.9 (WL_JTAG_TCK)	Remove R108 (or R28) and mount R113 to disconnect from PSoC 6 and connect the WiFi/BT module to the JTAG connector.
P3.0	BT UART TXD		BT UART TXD pin of the WiFi/BT module
P3.1	BT UART RXD		BT UART RXD pin of the WiFi/BT module
P3.2	BT UART CTS		BT UART CTS pin of the WiFi/BT module
P3.3	BT UART RTS		BT UART RTS pin of the WiFi/BT module

Table 4-2. Pioneer Board Pinout *(continued)*

PSoC 6 Pin	Primary Onboard Function	Secondary Onboard Function	Connection Details
P3.4	BT REG ON		BT REG ON pin of the WiFi/BT module
P3.5	BT HOST WAKE		BT HOST WAKE pin of the WiFi/BT module
P4.0	BT DEV WAKE		BT DEV WAKE pin of the WiFi/BT module
P4.1	Header J21.1		
P5.0	Arduino J4.1, D0 UART RX KitProg2 UART TX	–	Remove R159 to disconnect from KitProg2 UART TX.
P5.1	Arduino J4.2, D1 UART TX KitProg2 UART RX	–	Remove R156 to disconnect from KitProg2 UART RX.
P5.2	Arduino J4.3, D2 UART RTS KitProg2 UART CTS	–	Remove R93 to disconnect from KitProg2 UART CTS. This will also disconnect RTS and SPI lines from KitProg2.
P5.3	Arduino J4.4, D3 UART CTS KitProg2 UART RTS	–	Remove R88 to disconnect from KitProg2 UART CTS. This will also disconnect RTS and SPI lines from KitProg2.
P5.4	Arduino J4.5, D4	–	
P5.5	Arduino J4.6, D5	–	
P5.6	Arduino J4.7, D6	–	
P5.7	Header J21.3		
P6.0	Arduino J3.10, SCL KitProg2 I2C SCL	–	Remove R141 to disconnect from KitProg2 I2C SCL.
P6.1	Arduino J3.9, SDA KitProg2 I2C SDA	–	Remove R150 to disconnect from KitProg2 I2C SDA.
P6.2	GPIO on non-Arduino header (J2.15)	–	
P6.3	GPIO on non-Arduino header (J2.17)	CapSense Shield	Remove R44 to disconnect from GND and populate R145 to connect to the CapSense shield (hash pattern on the board).
P6.4	TDO/SWO	–	
P6.5	TDI	–	
P6.6	TMS/SWDIO	–	Remove R194 to disconnect from KitProg2 SWDIO.
P6.7	TCLK/SWCLK	–	Remove R183 to disconnect from KitProg2 SWCLK.
P7.0	TRACECLK	GPIO on non-Arduino header (J18.6)	Populate R181 to connect to J18 header.
P7.1	CINTA	–	
P7.2	CINTB	CSH	Remove C31 (0.47 nF) and populate 10 nF for CSH.
P7.3	GPIO on non-Arduino header (J18.5)	CSH	Remove R146 to disconnect from header and populate C29 (10 nF) for CSH.

Table 4-2. Pioneer Board Pinout *(continued)*

PSoC 6 Pin	Primary Onboard Function	Secondary Onboard Function	Connection Details
P7.4	TRACEDATA[3]	GPIO on non-Arduino header (J18.4)	Populate R178 to connect to J18.
P7.5	TRACEDATA[2]	GPIO on non-Arduino header (J18.3)	Populate R179 to connect to J18.
P7.6	TRACEDATA[1]	GPIO on non-Arduino header (J18.2)	Populate R180 to connect to J18.
P7.7	CMOD	GPIO on non-Arduino header (J18.1)	Populate R142 to connect to J18.
P8.0	Proximity	GPIO on non-Arduino header (J20.1)	Populate R64 with zero ohm to connect to header.
P8.1	CapSense Button0 Rx	GPIO on non-Arduino header (J20.2)	Remove R61 to disconnect CapSense pad and populate R172 to connect to header.
P8.2	CapSense Button1 Rx	GPIO on non-Arduino header (J20.3)	Remove R60 to disconnect CapSense pad and populate R166 to connect to header.
P8.3	CapSense Silder0 Rx	GPIO on non-Arduino header (J20.4)	Remove R53 to disconnect CapSense pad and populate R153 to connect to header.
P8.4	CapSense Silder1 Rx	GPIO on non-Arduino header (J20.5)	Remove R52 to disconnect CapSense pad and populate R152 to connect to header.
P8.5	CapSense Silder2 Rx	GPIO on non-Arduino header (J20.6)	Remove R47 to disconnect CapSense pad and populate R149 to connect to header.
P8.6	CapSense Silder3 Rx	GPIO on non-Arduino header (J20.7)	Remove R58 to disconnect CapSense pad and populate R158 to connect to header.
P8.7	CapSense Silder4 Rx	GPIO on non-Arduino header (J20.8)	Remove R59 to disconnect CapSense pad and populate R160 to connect to header.
P9.0	GPIO on non-Arduino header (J2.2)	–	
P9.1	GPIO on non-Arduino header (J2.4)	–	
P9.2	GPIO on non-Arduino header (J2.6)	–	
P9.3	TRACEDATA[0]	GPIO on non-Arduino header (J2.8)	Populate R162 to connect to header.
P9.4	GPIO on non-Arduino header (J2.10)	–	

Table 4-2. Pioneer Board Pinout *(continued)*

PSoC 6 Pin	Primary Onboard Function	Secondary Onboard Function	Connection Details
P9.5	GPIO on non-Arduino header (J2.12)	–	
P9.6	GPIO on non-Arduino header (J2.16)	–	
P9.7	GPIO on non-Arduino header (J2.18)	–	
P10.0	GPIO on Arduino header J2.1, A0	–	
P10.1	GPIO on Arduino header J2.3, A1	–	
P10.2	GPIO on Arduino header J2.5, A2	–	
P10.3	GPIO on Arduino header J2.7, A3	–	
P10.4	GPIO on Arduino header J2.9, A4 PDM_CLK	–	
P10.5	GPIO on Arduino header J2.11, A5 PDM_DAT	–	
P10.6	GPIO on non-Arduino header (J2.13)	–	
P10.7	Header J21.5		
P11.0	FRAM CS	GPIO on non-Arduino header (J18.8)	Connected to primary and secondary functions by default. Remove R175 to disconnect from J18 and load R39 (10K) as FRAM pull-up.
P11.1	RGB Blue LED	–	
P11.2	QSPI FLASH CS	GPIO on non-Arduino header (J18.7)	Populate R177 to connect to J18, remove R176 to disconnect from Flash.
P11.3	QSPI FLASH/ FRAM DATA3	–	
P11.4	QSPI FLASH/ FRAM DATA2	–	
P11.5	QSPI FLASH/ FRAM DATA1	–	
P11.6	QSPI FLASH/ FRAM DATA0	–	
P11.7	QSPI FLASH/ FRAM CLK	–	
P12.0	Arduino J3.4, D11 SPI MOSI	ICSP header (J5.4) and Pmod header (J14.2)	Remove R77 to disconnect from KitProg2_SPI lines.
P12.1	Arduino J3.5, D12 SPI MISO	ICSP header (J5.1) and Pmod header (J14.3)	Remove R85 to disconnect from KitProg2_SPI lines.

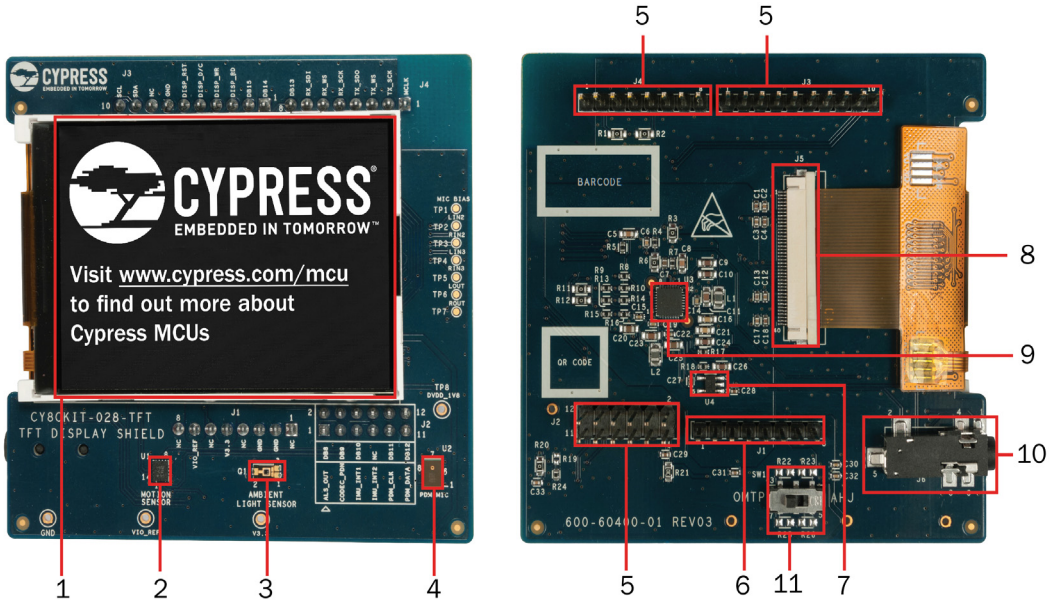
Table 4-2. Pioneer Board Pinout *(continued)*

PSoC 6 Pin	Primary Onboard Function	Secondary Onboard Function	Connection Details
P12.2	Arduino J3.6, D13 SPI CLK	ICSP header (J5.3) and Pmod header (J14.4)	Remove R81 to disconnect from KitProg2_SPI lines.
P12.4	KitProg2 SPI SELECT	GPIO on non-Arduino header (J19.10)	Connected to primary function by default. Populate R74 to connect to J19 or remove R83 to disconnect KitProg2_SPI_SELECT.
P12.5	PMOD SPI SELECT, J14.1	GPIO on non-Arduino header (J19.9)	Connected to primary and secondary functions by default. Remove R73 to disconnect from J19 or remove R82 to disconnect PMOD_SPI_SELECT.
P12.6	ECO IN	GPIO on non-Arduino header (J19.8)	
P12.7	ECO OUT	GPIO on non-Arduino header (J19.6)	
P13.0	GPIO on Arduino Header J3.1, D8		
P13.1	GPIO on Arduino Header J3.2, D9		
P13.2	USB HOST EN	GPIO on non-Arduino header (J21.2)	
P13.3	USB_INT_L	GPIO on non-Arduino header (J21.4)	
P13.4	USB_DEV_VBUS_DET	GPIO on non-Arduino header (J21.6)	
P13.5	GPIO on non-Arduino header J21.8		
P13.6	GPIO on non-Arduino header J2.19	CapSense shield	Remove R44 to disconnect from GND and populate R45 to connect to the CapSense shield (hash pattern on the board).
P13.7	Red user LED	GPIO on non-Arduino header (J2.20)	Remove R31 to disconnect from LED

4.2 CY8CKIT-028-TFT Details

4.2.1 CY8CKIT-028-TFT Display Shield

Figure 4-6. TFT Display Shield



The TFT display shield has the following peripherals:

1. **2.4-inch TFT display:** This is a 2.4-inch TFT LCD module with 240x320 pixel resolution. This display module is configured for an 8-bit parallel pinout connection to interface with the PSoC 6 device on the baseboard.
2. **Inertial Measurement Unit (U1):** This Inertial Measurement Unit (IMU) is a three-axis acceleration and three-axis gyroscopic motion sensor that can be used to count steps to emulate a pedometer or similar application.
3. **Ambient light sensor (Q1):** This is a high photosensitive NPN phototransistor IC that can be used to detect intensity of ambient light.
4. **PDM microphone (U2):** This microphone converts voice inputs to pulse-density modulated (PDM) digital signals.
5. **Arduino-compatible I/O header (J2, J3, and J4):** These headers interface with the PSoC 6 MCU GPIOs on the baseboard.
6. **Arduino-compatible power header (J1):** This header receives power from header J1 on the board.
7. **Voltage regulator - 1.8 V (U4):** An LDO that converts 3.3 V to 1.8 V for the digital supply of the audio codec.
8. **TFT display connector (J5):** This connector is used to connect the TFT display to the circuits on the TFT display shield.
9. **Audio Codec (U3)**: This is a low power consumption, 32-bit stereo codec with speaker amplifiers. The left channel and right channel amplifier output pins of the device are connected to the on-board audio jack.
10. **Audio Jack (J6)**: The on-board audio jack provides a provision of connecting both AHJ and OMTP headphones. The headset standard can be set by an onboard switch SW1.

11. **Audio Jack Selection Switch (SW1)**: This on-board selection switch can set the headphone type either to AHJ and OMTP standard connected to the audio jack.

Table 4-3. TFT Shield Pinout

Pin #	Arduino Pin	Arduino Function	TFT Shield Function	Pioneer Board Connection
J1.1	VIN	VIN	NC	VIN
J1.2	GND	GND	GND	GND
J1.3	GND	GND	GND	GND
J1.4	5 V	5 V	NC	5 V
J1.5	3.3 V	3.3 V	VCC 3.3V	3.3 V
J1.6	RESET	RESET	NC	SWD RESET
J1.7	I/O REF	I/O REF	VIO REF	P6 VDD
J1.8	--	--	NC	NC
J2.1	A0	ADC0	ALS OUT	P10[0]
J2.2	--	--	TFT DISP DB8	P9[0]
J2.3	A1	ADC1	Codec PDN SW	P10[1]
J2.4	--	--	TFT DISP DB9	P9[1]
J2.5	A2	ADC2	IMU INT1	P10[2]
J2.6	--	--	TFT DISP DB10	P9[2]
J2.7	A3	ADC3	IMU INT2	P10[3]
J2.8	--	--	NC	P9[3]
J2.9	A4	ADC4 / SDA (I2C)	PDM CLK	P10[4]
J2.10	--	--	TFT DISP DB11	P9[4][a]
J2.11	A5	ADC5	PDM DATA	P10[5]
J2.12	--	--	TFT DISP DB12	P9[5]
J3.1	D8	DIGITAL I/O	TFT DISP DB14	P13[0]
J3.2	D9	PWM	TFT DISP DB15	P13[1]
J3.3	D10	SS/PWM	TFT DISP RD_L	P12[3]
J3.4	D11	MOSI/PWM	TFT DISP WR_L	P12[0]
J3.5	D12	MISO	TFT DISP D/C	P12[1]
J3.6	D13	SCK	TFT DISP RST_L	P12[2]
J3.7	GND	GND	GND	GND
J3.8	AREF	analog ref i/p	NC	VREF
J3.9	SDA	SDA	I2C SDA (IMU and audio codec)	P6[1]
J3.10	SCL	SCL	I2C SCL (IMU and audio codec)	P6[0]
J4.1	D0	RX	I2S MCLK	P5[0]
J4.2	D1	TX	I2S TX SCK	P5[1]
J4.3	D2	DIGITAL I/O	I2S TX WS	P5[2]
J4.4	D3	PWM, I/O	I2S TX SDO	P5[3]
J4.5	D4	DIGITAL I/O	I2S RX SCK	P5[4]
J4.6	D5	PWM, I/O	I2S RX WS	P5[5]
J4.7	D6	PWM, I/O	I2S RX SDI	P5[6]

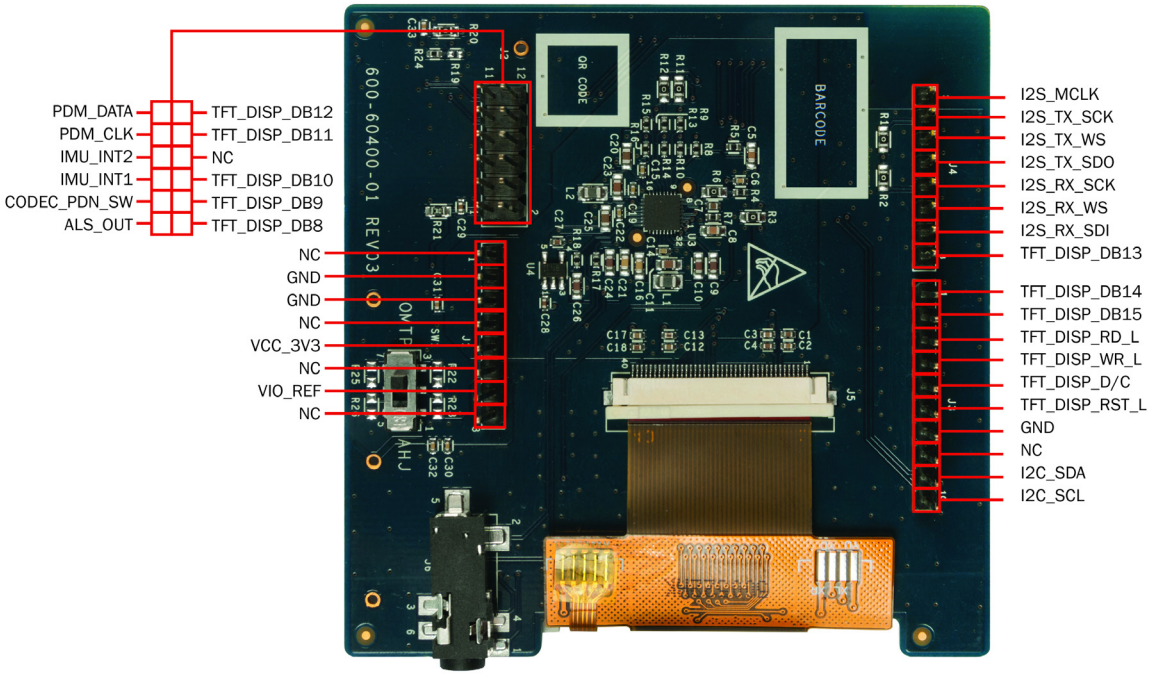
Table 4-3. TFT Shield Pinout

J4.8	D7	DIGITAL I/O	TFT DISP DB13	P0[2]

a. Mount R162 (on the Pioneer board) to connect J2.10 of the TFT board to P9[4].

Figure 4-7 shows the connectivity of the TFT display shield with the CY8CKIT-062-WiFi-BT baseboard through the headers.

Figure 4-7. TFT Shield Pinout



Notes:

The TFT display operation at 1.8 V is currently not supported in this version of the kit. Ensure the following conditions are met when the CY8CKIT-028-TFT Display Shield is mounted on the PSoC 6 WiFi-BT Pioneer Board.

1. Ensure that SW5 is either set to 3.3 V or set to the 1.8 V–3.3 V VARIABLE with PSoC Programmer or PSoC Creator selecting a voltage of 2.5 V or higher.
2. If you want to erase the internal flash of the PSoC 6 MCU, ensure that PSoC Programmer or PSoC Creator setting is not 1.8 V when the SW5 is set to the 1.8 V–3.3 V VARIABLE position.

4.3 KitProg2

4.3.1 Introduction

The PSoC 6 WiFi-BT Pioneer Kit can be programmed and debugged using the onboard KitProg2. The KitProg2 is a multi-functional system, which includes a programmer, debugger, USB-I2C bridge, USB-UART bridge, and USB-SPI bridge. KitProg2 also supports mass storage programming and CMSIS-DAP, and custom applications. A PSoC 5LP device is used to implement KitProg2 functionality. The KitProg2 is integrated in most PSoC development kits. For more details on the KitProg2 functionality, see the KitProg2 User Guide.

Before programming the device, ensure that PSoC Creator and PSoC Programmer software are installed on the computer. See “Install Kit Software” on page 18 for more information.

4.3.2 Programming using PSoC Programmer

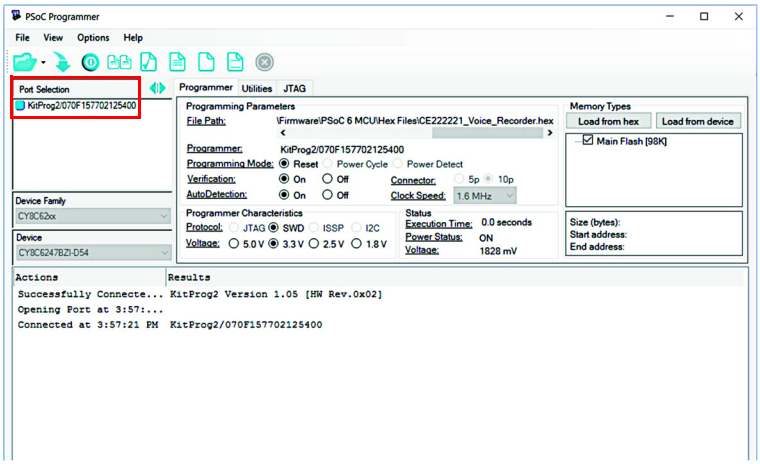
PSoC Programmer can be used to program existing *.hex* files into the PSoC 6 WiFi-BT Pioneer Kit. For more details, see the “Programming Using PSoC Programmer” section in the KitProg2 User Guide.

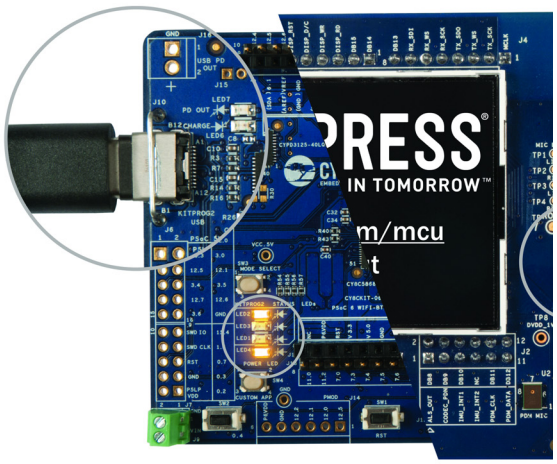
The KitProg2 firmware normally does not require any update. If necessary you can use the PSoC Programmer software to update the KitProg2 firmware. For more details, see the “Updating the KitProg2 Firmware” section in the KitProg2 User Guide.

4.3.3 Kit Enumeration and Programming Modes of KitProg2

The PSoC 5LP device in the PSoC 6 WiFi-BT Pioneer Kit supports two types of programming interfaces (proprietary SWD programming and CMSIS-DAP mode programming). You can press the mode select button (**SW3**) to switch between KitProg2 SWD and CMSIS-DAP modes. When the SWD-based programming mode is active, both the amber LEDs (LED2 and LED4) will turn ON (see Figure 4-8). In the CMSIS-DAP mode, only LED4 will be ON, and LED1, LED2, and LED3 are OFF (see Figure 4-9). For more details on the KitProg2 programming interfaces, see the KitProg2 User Guide.

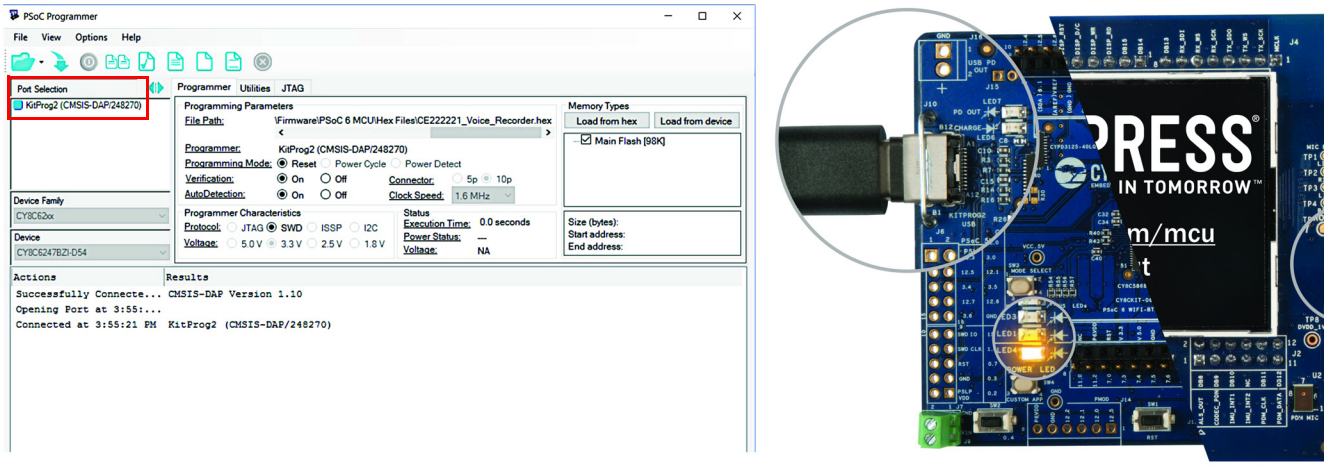
Figure 4-8. KitProg2 SWD Programming Interface





To know whether the kit is successfully enumerated, open PSoC Programmer software and connect the kit to your PC. Check the status of the onboard LEDs and the **Port Selection** window in PSoC Programmer to know the KitProg2 programming interface, see Figure 4-8 and Figure 4-9.

Figure 4-9. CMSIS-DAP Programming Interface



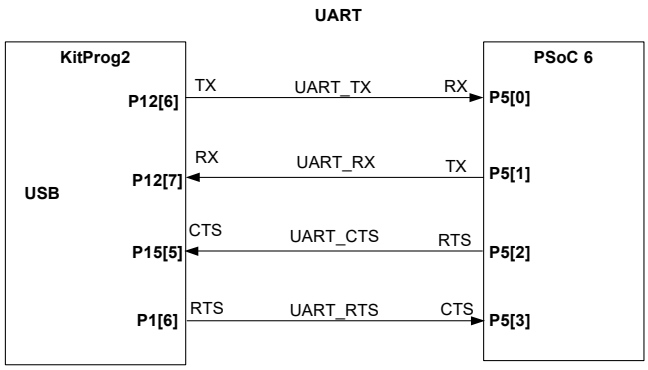
Note that if you are programming the kit using WICED 6.1 (or later), the CMSIS-DAP programming mode should be enabled. After confirming the successful enumeration of the kit, you can close PSoC Programmer if you want to program the kit either through PSoC Creator or through WICED 6.1.

The KitProg2-based programming interface allows you to program PSoC 6 MCU with the desired .hex files.

4.3.4 USB-UART Bridge

The KitProg2 on the PSoC 6 WiFi-BT Pioneer Kit can act as a USB-UART bridge. The UART and flow-control lines between the PSoC 6 MCU and the KitProg2 are hard-wired on the board, as Figure 4-10 shows. For more details on the KitProg2 USB-UART functionality, see the KitProg2 User Guide.

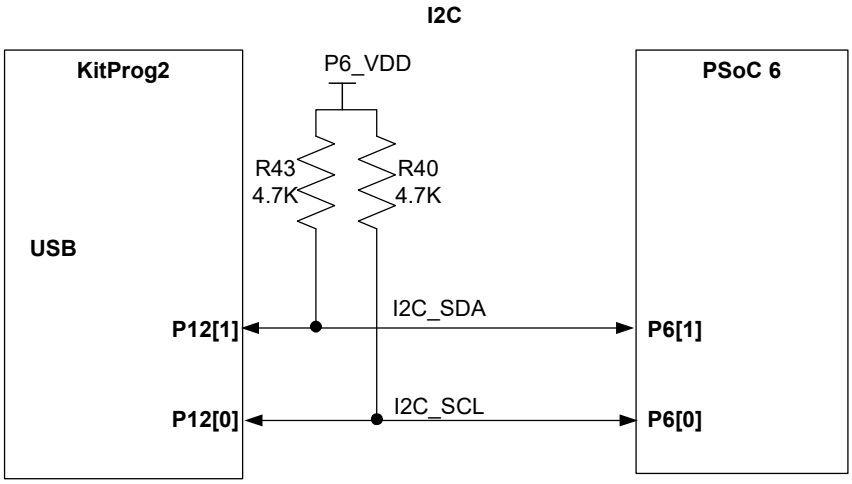
Figure 4-10. UART Connection between KitProg2 and PSoC 6



4.3.5 USB-I2C Bridge

The KitProg2 can function as a USB-I2C bridge and communicate with the Bridge Control Panel (BCP) software. The I2C lines on the PSoC 6 MCU are hard-wired on the board to the I2C lines of the KitProg2, with onboard pull-up resistors as Figure 4-11 shows. The USB-I2C supports I2C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg2 USB-I2C functionality, see the KitProg2 User Guide.

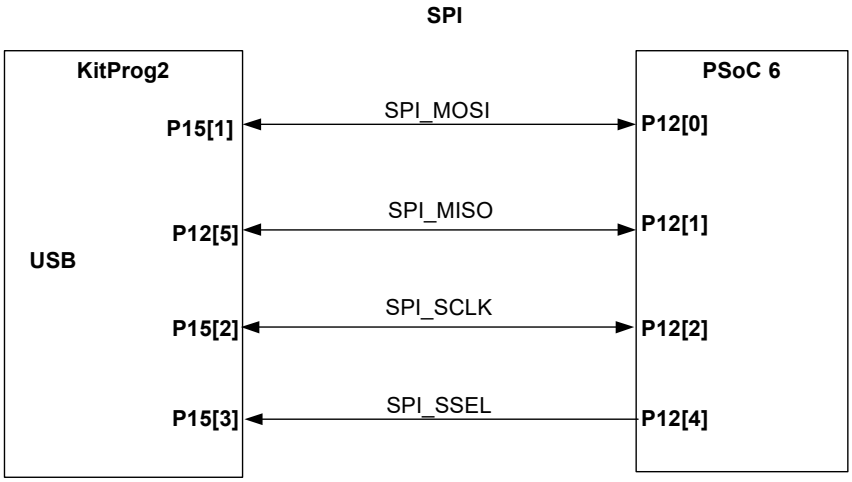
Figure 4-11. I2C Connection between KitProg2 and PSoC 6



4.3.6 USB-SPI Bridge

The KitProg2 can function as a USB-SPI bridge. The SPI lines between the PSoC 6 MCU and the KitProg2 are hard-wired on the board, as Figure 4-12 shows. For more details on the KitProg2 USB-SPI functionality, see the KitProg2 User Guide.

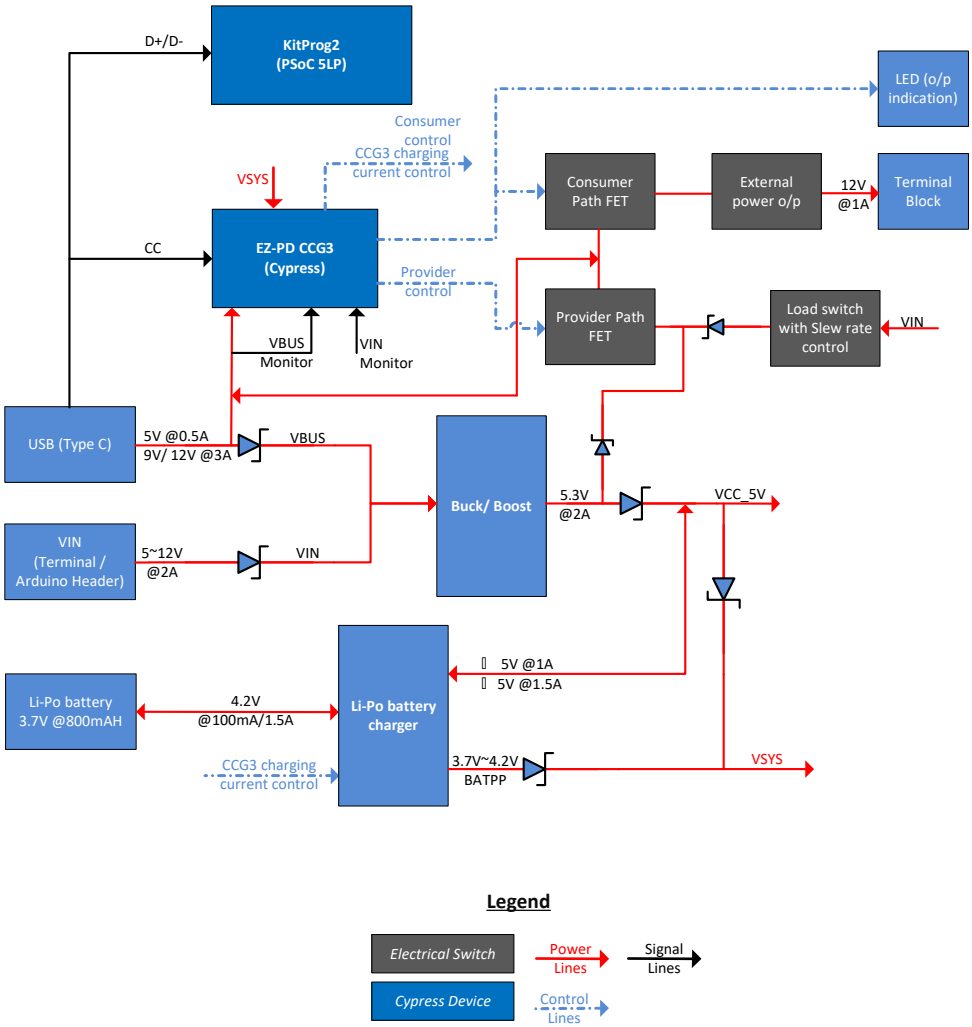
Figure 4-12. SPI Connection between KitProg2 and PSoC 6



4.4 EZ-PD CCG3 Type-C Power Delivery

The Pioneer board includes a Cypress EZ-PD CCG3 power delivery system. The EZ-PD CCG3 is pre-programmed and can deliver power from a Type-C port to the onboard header **J16** (known as the consumer path), while simultaneously charging a 3.7-V, lithium-ion polymer battery connected to **J15**. In addition, the power delivery system can deliver power to a Type-C peripheral, such as a mobile phone, with the power derived from the VIN **(J9)** supply (known as the provider path). Note that to use the EZ-PD CCG3 Type-C power delivery system, a power delivery capable USB Type-C to Type-C cable should be connected to **J10**. This cable is not included in the kit, and should be purchased separately.

Figure 4-13. Type-C Block Diagram



The power delivery system works as follows:

1. If the power delivery system detects a non Type-C power adapter (legacy USB), CCG3 will charge the battery at 100 mA. CCG3 will also disable the consumer and provider paths.
2. On detection of a Type-C power adapter, CCG3 will request 5 V at 3 A, 9 V at 3 A, or 12 V at 3 A depending on the host capability. After the power level is successfully negotiated, the consumer path is enabled by turning ON the load switch **U13**. This load switch is hardware-limited to supply up to 1 A through header **J16** to an external device. CCG3 will use the remaining current to charge the battery connected to **J15** at a higher charging rate up to 1.5 A and PD output voltage availability indicator (**LED7**) will be turned ON.
3. CCG3 will also advertise that it can provide 5 V, 9 V, or 12 V if a DC power supply capable of providing either of these voltages is connected at VIN **(J9)**. The current is limited in this case to 1 A. Note that the external supply must be capable of providing this current. If a connected, Type-C device requests power, the provider path is enabled by turning on load switch **U31**. Table 4-4 details the power delivery scenarios for onboard CCG3.

Table 4-4. Type-C table Power Delivery Scenarios

USB Host/Consumer Capability	VIN	Consumer Capability	Provider Capability	External USB PD Out (J16 Header)	Battery Charging Current
Non Type-C Power adapter (Legacy USB)	<5 V	N/A	0	0	100 mA
	>5 V	N/A	0	0	0
Type-C, PD power adapter (12-V capable)[a]	<12 V	12 V@3A	0	12 V@1A[b]	1.5A max
	>12 V	N/A	0	0	0
Type-C, capable of providing max 9 V[a]	<9 V	9 V@3A	0	9 V@1A	1.5A max
	>9 V	N/A	0	0	0
Type-C only, capable of providing max 5 V[a]	<5	5 V@3A	0	5 V@1A	1.5A max
	>5	N/A	0	0	0
Type-C, requesting 12 V[a]	≠12 V	0	5 V@1A	0	0
	12 V	0	12 V@1A	0	0
Type-C, requesting 9 V[a]	≠9 V	0	5 V@1A	0	0
	9 V	0	9 V@1A	0	0
Type-C, requesting 5 V[a]	≠5 V	0	5 V@1A	0	0
	5 V	0	5 V@1A	0	0
Type-C, requesting another voltage	5 V < VIN <12 V	0	5 V@1A	0	0

a. The table is valid only if the Type-C cable is connected first and then VIN is applied. If VIN is applied first, consumer capability will be N/A.
b. Due to the voltage drop-in series components, the voltage at J16 is ~9 V when 12 V PD power adapter is used.

For more information on USB Type-C power delivery with CCG3 device, see the EZ-PD CCG3 webpage.

A. Appendix

A.1 Schematics

Refer to the schematic files available in the kit installation directory under the following paths:

- *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\1.0\Hardware\CY8CKIT-028-TFT\CY8CKIT-028-TFT Schematic.pdf*
- *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\1.0\Hardware\CY8CKIT-062-WiFi-BT\CY8CKIT-062-WiFi-BT Schematic.pdf*

A.2 Hardware Functional Description

This section explains the individual hardware blocks of the PSoC 6 WiFi-BT Pioneer board.

A.2.1 PSoC 6 MCU (U1)

PSoC 6 MCU is Cypress' latest, ultra-low-power PSoC specifically designed for wearables and IoT products. It is a programmable embedded system-on-chip, integrating a 150-MHz CM4 as the primary application processor, a 100-MHz CM0+ that supports low-power operations, up to 1 MB Flash and 28 8KB SRAM, CapSense touch-sensing, and custom analog and digital peripheral functions. The programmable analog and digital peripheral functions allow higher flexibility, in-field tuning of the design, and faster time-to-market.

For more information, see the PSoC 6 MCU webpage and the datasheet.

A.2.2 PSoC 5LP (U2)

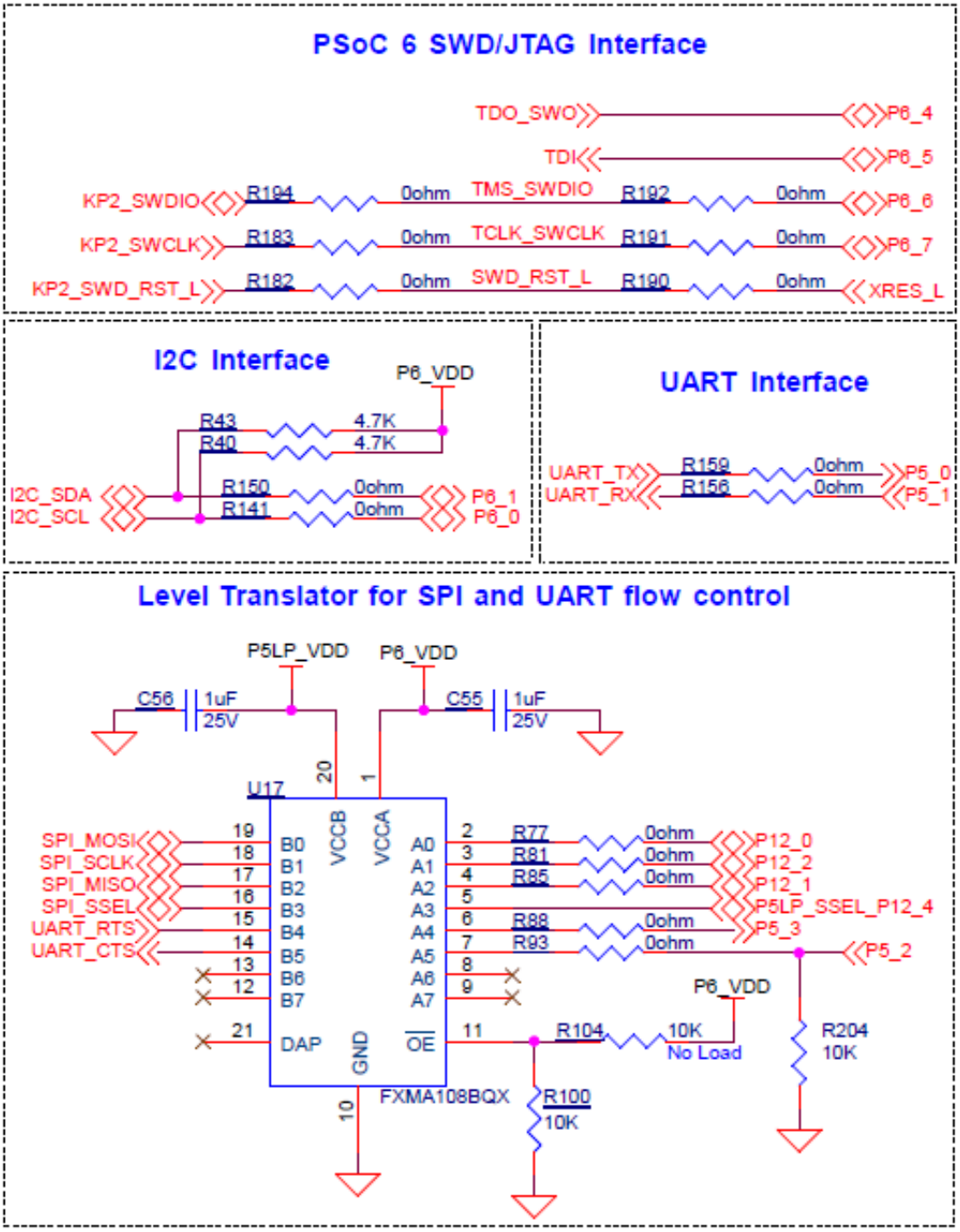
An onboard PSoC 5LP (CY8C5868LTI-LP039) is used as KitProg2 to program and debug the PSoC 6 MCU. The PSoC 5LP connects to the USB port of a PC through a USB connector and to the SWD and other communication interfaces of PSoC 6 MCU. The PSoC 5LP is a system-level solution providing MCU, memory, analog, 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 webpage. Also, see the CY8C58LPxx family datasheet.

A.2.3 Serial Interconnection between PSoC 5LP and PSoC 6 MCU

In addition to being used as an onboard programmer, the PSoC 5LP functions as an interface for the USB-UART, USB-I2C, and USB-SPI bridges, as shown in Figure A-1. The USB-Serial pins of the PSoC 5LP are hard-wired to the I2C/UART/SPI pins of the PSoC 6 MCU. These pins are also available on the Arduino-compatible I/O headers; therefore, the PSoC 5LP can be used to control Arduino shields with an I2C/UART/SPI interface.

Figure A-1. Schematics of Programming and Serial Interface Connections



A.2.4 EZ-PD CCG3 Power Delivery System

Cypress EZ-PD CCG3 provides a complete solution ideal for power adapters, power banks, Type-C dongles, monitors, docks, and notebooks. See EZ-PD CCG3 Type-C Power Delivery on page 56 for more details of the power delivery system implementation on the Pioneer board.

Figure A-2. Schematics of EZ-PD CCG3 Power Delivery System

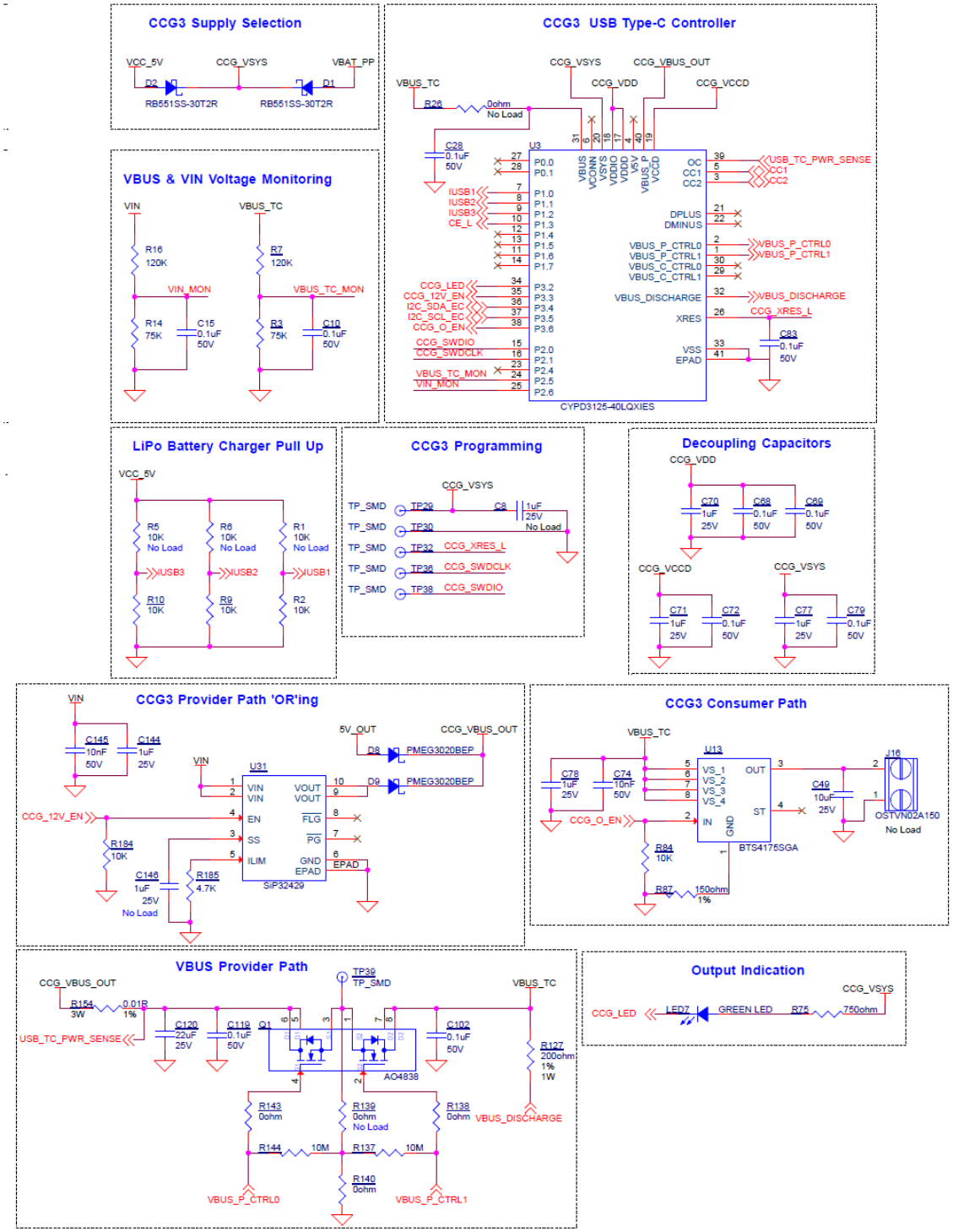
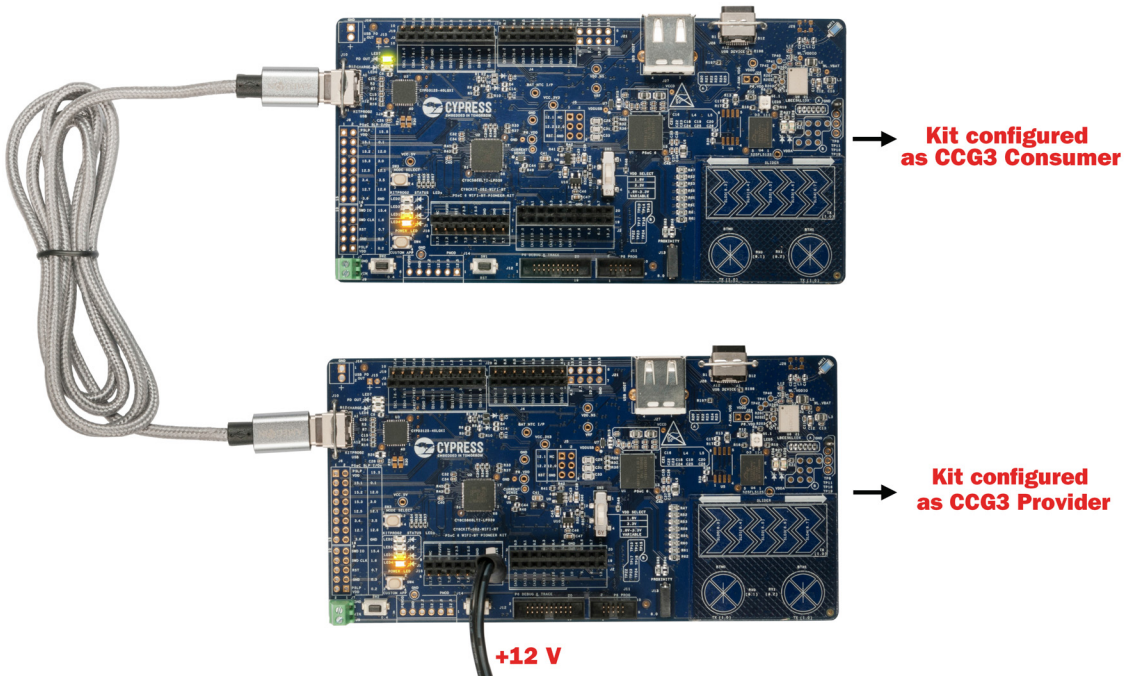


Figure A-3. EZ-PD CCG3 Power Delivery Setup using Two CY8CKIT-062-WiFi-BT Kits



A.2.5 Power Supply System

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

- 5 V, 9 V, or 12 V from the onboard USB Type-C connector
- 5 V to 12 V power from an Arduino shield or from external power supply through VIN header **J9** or **J1**
- 3.7 V from a rechargeable Li-Po battery connected to **J15**
- 5 V from an external programmer/debugger connected to **J11** and **J12**

The power supply system is designed to support 1.8 V to 3.3 V operation of the PSoC 6 MCU. In addition, an intermediate voltage of 5 V is required for operation of the power delivery circuitry and KitProg2. Therefore, three regulators are used to achieve 1.8 V to 3.3 V and 5 V outputs – a buck boost regulator **(U30)** that generates a fixed 5 V from an input of 5 V to 12 V, and a main regulator **(U10)** that generates either a variable 1.8 V to 3.3 V, or a fixed 1.8 V, or a fixed 3.3 V from the output of **U30**. Figure A-4 shows the schematics of the voltage regulator and power selection circuits. In addition to this, the battery charger **U19** also functions as a boost regulator. **U19** boosts the battery voltage to provide a 5 V to the main regulator **U10**. This feature is enabled only when the VIN and the USB supply are unavailable.

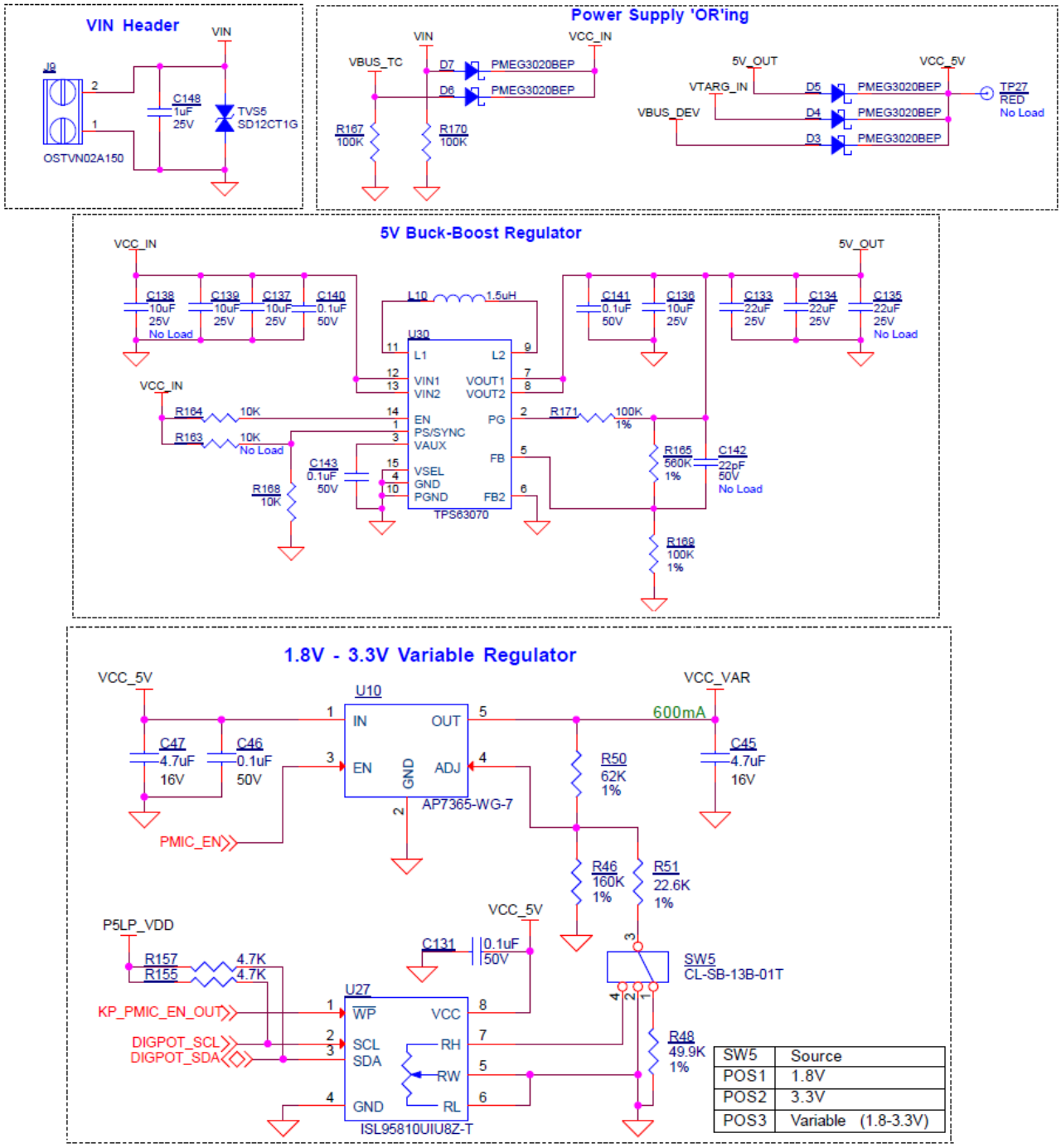
The voltage selection is made through switch **SW5**. In addition, an onboard 330-mF super-capacitor **(C103)** can be used to power the backup domain (VBACKUP) of PSoC 6 MCU. Switch **SW7** selects the VBACKUP supply connection of PSoC 6 MCU between V_{DDD} and the super-capacitor. When V_{DDD} is selected, the variable regulator ON/OFF terminal is controlled by KitProg2. When the super-capacitor is selected, the regulator ON/OFF terminal is controlled by PSoC 6 MCU. To ensure proper operation of the PSoC 6 MCU, the super-capacitor, when selected, must be charged internally by PSoC 6 MCU before turning OFF the regulator. For more details of the PSoC 6 MCU backup system and power supply, see the PSoC 62 Architecture Technical Reference Manual.

Table A-1 details the different powering scenarios for Pioneer board.

Table A-1. Power Supply Scenarios

Power Inputs					Board Condition		
USB	**VIN**	**ETM Header (VTARG_IN)**	**Battery Connected**	**JTAG/SWD Header (VTARG_REF)**	**Main Regulator Powered by**	**PSoC Powered by**	**Battery Charging**
Non Type-C power adapter (legacy USB), 5 V	<5	N/A	Yes	N/A	Type-C	Main Regulator	100 mA
	>5	N/A	N/A	N/A	VIN	Main Regulator	No
Type-C, PD power adapter	< PD power adapter	N/A	Yes	N/A	Type-C	Main Regulator	1.5A
	> PD power adapter, <12 V	N/A	N/A	N/A	VIN	Main Regulator	No
0 V	5 V–12 V	N/A	N/A	N/A	VIN	Main Regulator	No
0 V	0 V	5 V	N/A	N/A	ETM (VTARG_IN)	Main Regulator	No
0 V	0 V	0 V	3.2 V–4.2 V	N/A	Battery	Main Regulator	No
0 V	0 V	0 V	0 V	1.8 V–3.3 V	N/A	JTAG/SWD (VTARG_REF)	No

Figure A-4. Schematics of Power Supply System



A.2.6 Expansion Connectors

A.2.6.1 Arduino-compatible Headers (J1, J2, J3, J4, and J5)

The board has five Arduino-compatible headers: **J1**, **J2**, **J3**, **J4**, and **J5** (**J5** is not populated by default). You can connect 3.3-V Arduino-compatible shields to develop applications based on the shield's hardware. Note that 5-V shields are not supported and connecting a 5-V shield may permanently damage the board. See Table 4-2 on page 45 for details on PSoC 6 MCU pin mapping to these headers.

A.2.6.2 PSoC 6 MCU I/O Headers (J18, J19, and J20)

These headers provide connectivity to PSoC 6 MCU GPIOs that are not connected to the Arduino-compatible headers. Most of these pins are multiplexed with onboard peripherals and are not connected to PSoC 6 MCU by default. For detailed information on how to rework the kit to access these pins, see PSoC 6 WiFi-BT Pioneer Board Reworks on page 69.

A.2.6.3 PSoC 5LP GPIO Header (J6)

J6 is a 8x2 header provided on the board to bring out several pins of the PSoC 5LP to support advanced features such as a low-speed oscilloscope and a low-speed digital logic analyzer. This header also contains the USB-UART, USB-I2C, and USB-SPI bridge pins that can be used when these pins are not accessible on the Arduino headers because a shield is connected. The additional PSoC 5LP pins are connected directly to the internal programmable analog logic of PSoC 5LP. This header also has GPIOs for custom application usage. J6 is not populated by default. Note that the SPI, RTS, and CTS lines on these headers are directly from PSoC 5LP (before level translator).

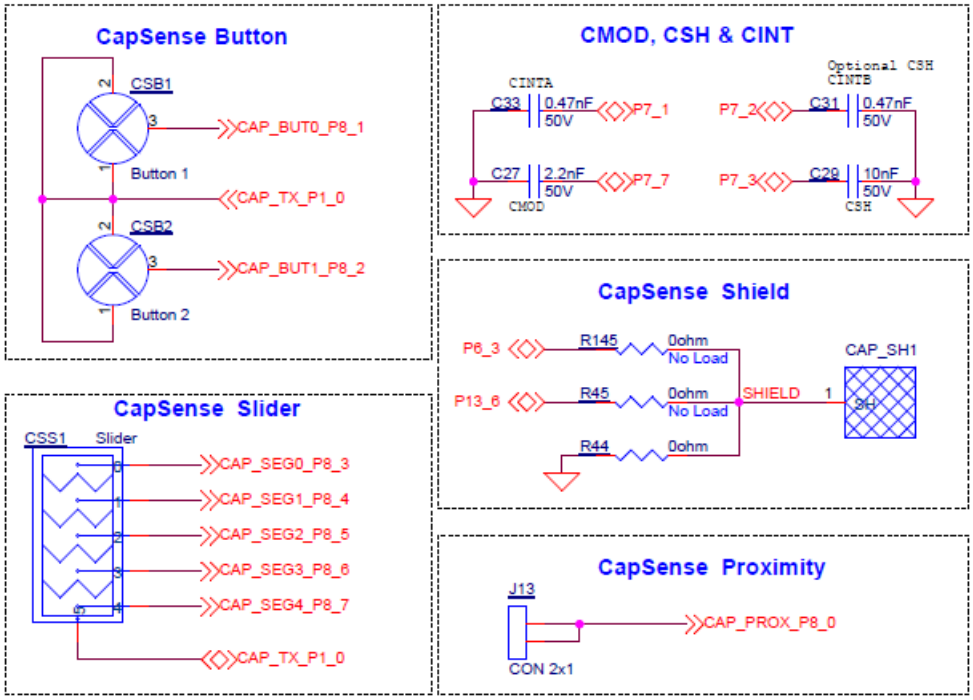
A.2.6.4 KitProg2 Custom Application Header (J7)

A 5x2 header is provided on the board to bring out more PSoC 5LP GPIOs for custom application usage. This header also brings out the PSoC 5LP programming pins and can be programmed using MiniProg3 and a five-pin programming connector. J7 is not populated by default.

A.2.7 CapSense Circuit

A CapSense slider and two buttons, all of which support both self-capacitance (CSD) and mutual-capacitance (CSX) sensing, and a CSD proximity sensor (header) are connected to PSoC 6 MCU as Figure A-5 shows. Four external capacitors – C_{MOD} and C_{SH} for CSD, C_{INTA} and C_{INTB} for CSX are present on the Pioneer board. Note that CSH is not loaded by default. For details on using CapSense including design guidelines, see the *Getting Started with CapSense Design Guide*.

Figure A-5. Schematics of CapSense Circuit



A.2.8 LEDs

LED1, **LED2**, and **LED3** (red, amber, and green respectively) indicate the KitProg2 status (see the KitProg2 User Guide for details). **LED4** (amber) indicates the status of power supplied to PSoC 6 MCU. **LED7** (green) indicates the status of power delivery output on **J16**. **LED6** (red) indicates the battery charger status.

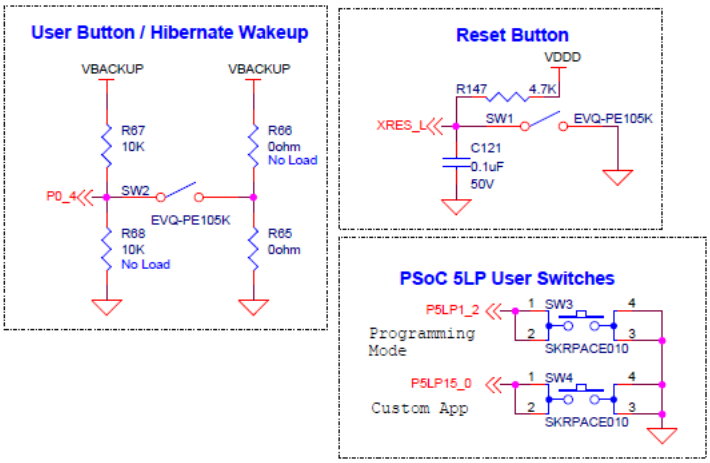
The Pioneer board also has two user-controllable LEDs **(LED8** and **LED9)** and an RGB LED **(LED5)** connected to PSoC 6 MCU pins for user applications.

A.2.9 Push Buttons

The PSoC 6 WiFi-BT Pioneer Kit has a reset button and three user buttons:

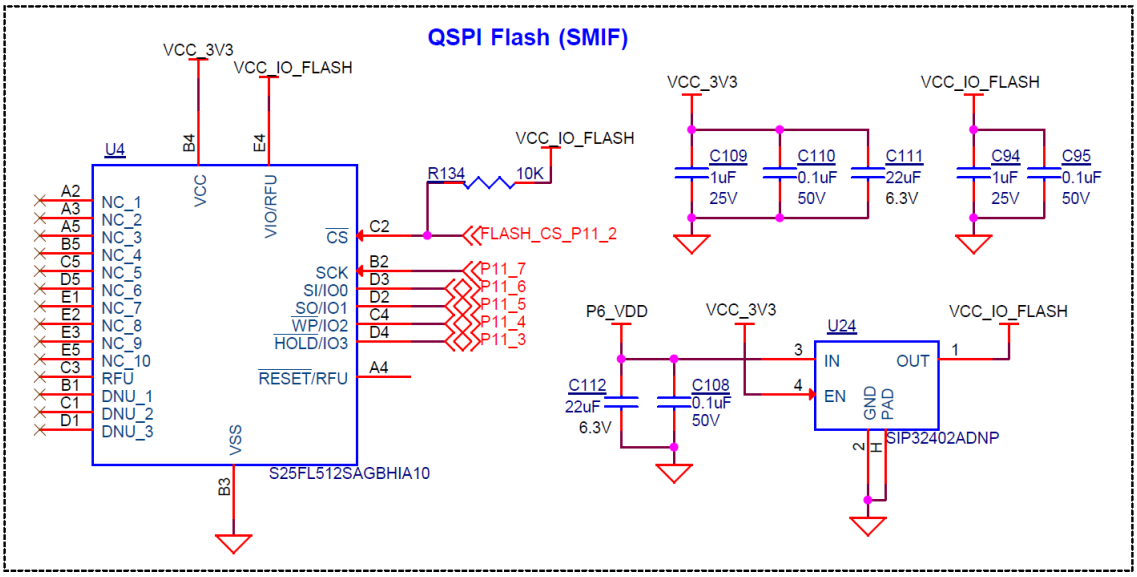
- The reset button **(SW1)** is connected to the XRES pin of the PSoC 6 MCU, and is used to reset the device.
- The user button **(SW2)** is connected to pin P0[4] of the PSoC 6 MCU. **(SW2)** can be changed to active HIGH mode by changing the zero resistors as shown in the figure below.
- The remaining two buttons – **SW3** and **SW4** are connected to the PSoC 5LP device for programming mode and custom application selection respectively (see the KitProg2 User Guide for details).

All the buttons connect to ground on activation (active LOW) by default.



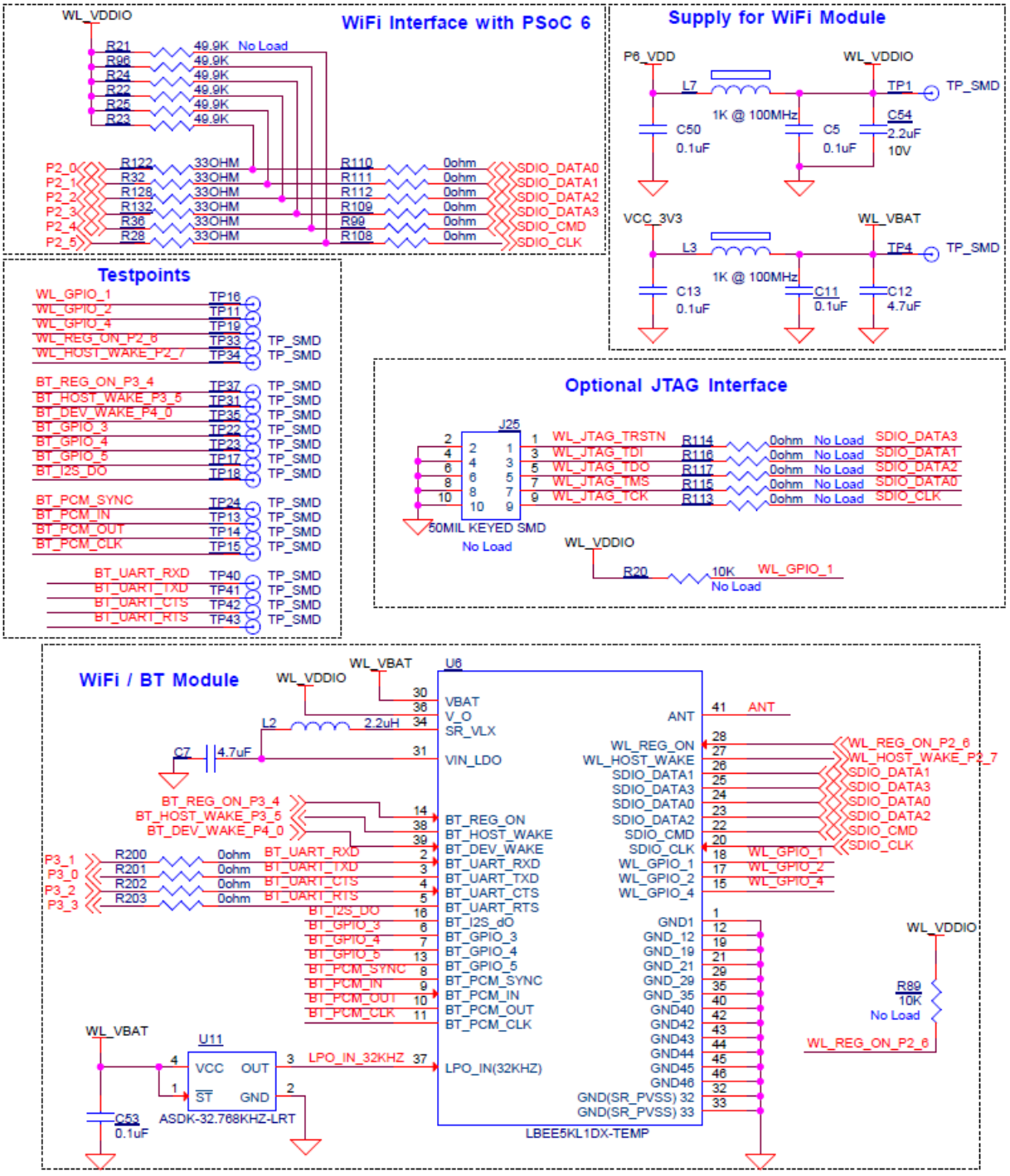
A.2.10 Cypress NOR Flash

The Pioneer board has a Cypress NOR flash memory (S25FL512SAGMFI011) of 512 Mb capacity. The NOR flash is connected to the serial memory interface (SMIF) of PSoC 6 MCU. The NOR flash device can be used for both data and code memory with execute-in-place (XIP) support and encryption.



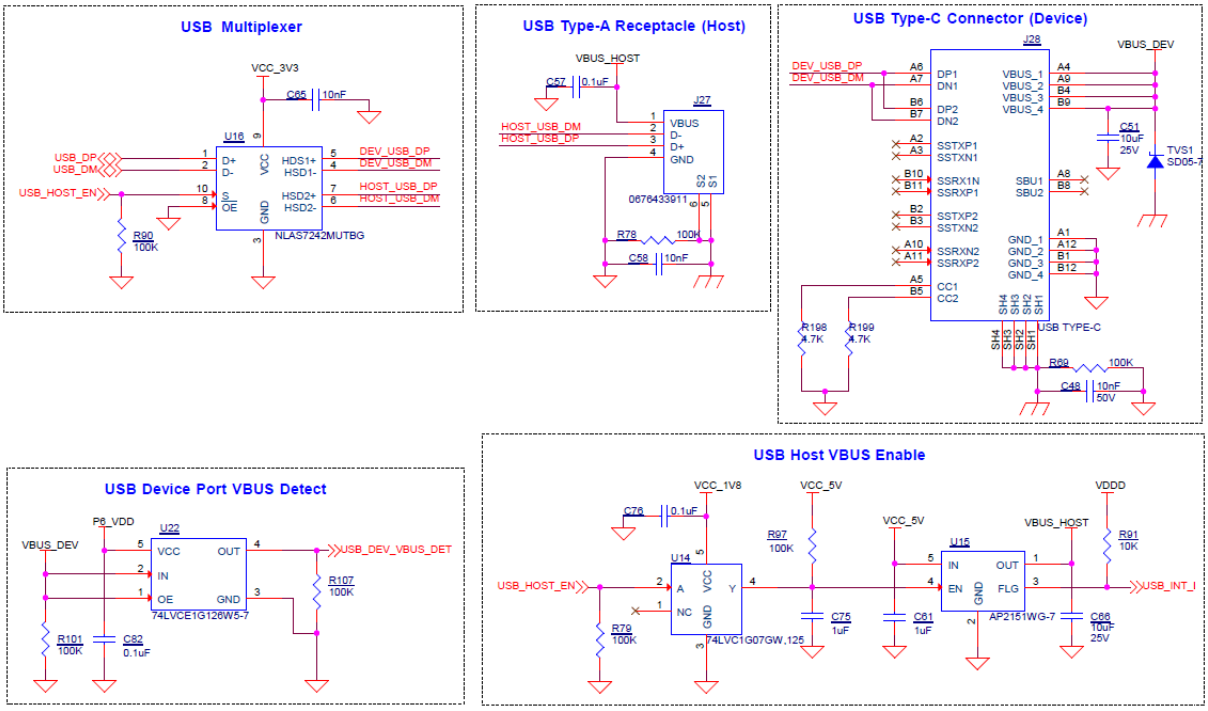
A.2.11 WiFi and Bluetooth Module

The Pioneer board features an onboard WiFi and Bluetooth combination module to demonstrate the wireless communication features. This LBEE5KL1DX is a Type 1DX module available with 2.4-GHz WLAN and Bluetooth functionality. Based on Cypress CYW4343W, this module provides high-efficiency RF front-end circuits.



A.2.12 USB Host and USB Device Connections

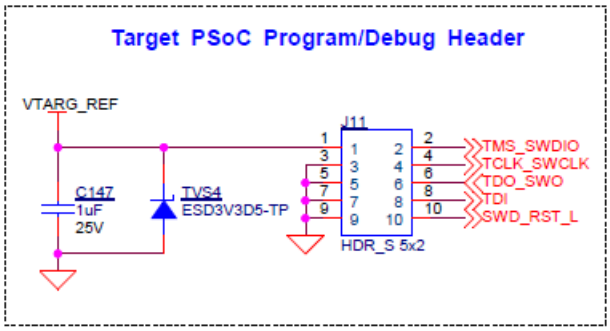
The PSoC 6 MCU can be configured as either a USB host or USB device. When PSoC 6 is programmed as a host controller, you can connect an external device such as mouse, keyboard, and flash memory to the USB Type-A receptacle port (J27). When the PSoC 6 MCU is programmed as a USB device, you can connect the kit either to a PC or to another host controller through a Type-C cable at the USB Type-C Connector (J28).

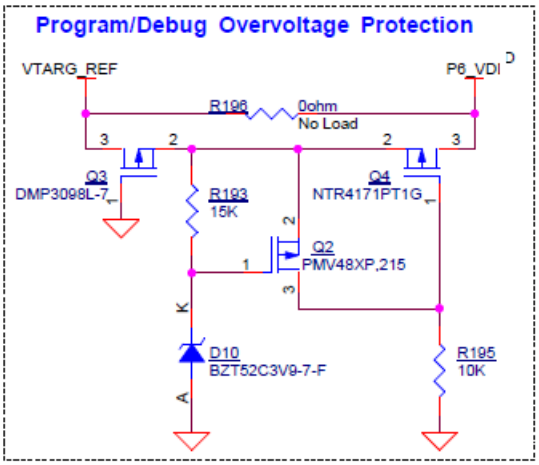


A.3 PSoC 6 WiFi-BT Pioneer Board Reworks

A.3.1 Bypass Protection Circuit on Program and Debug Header (J11)

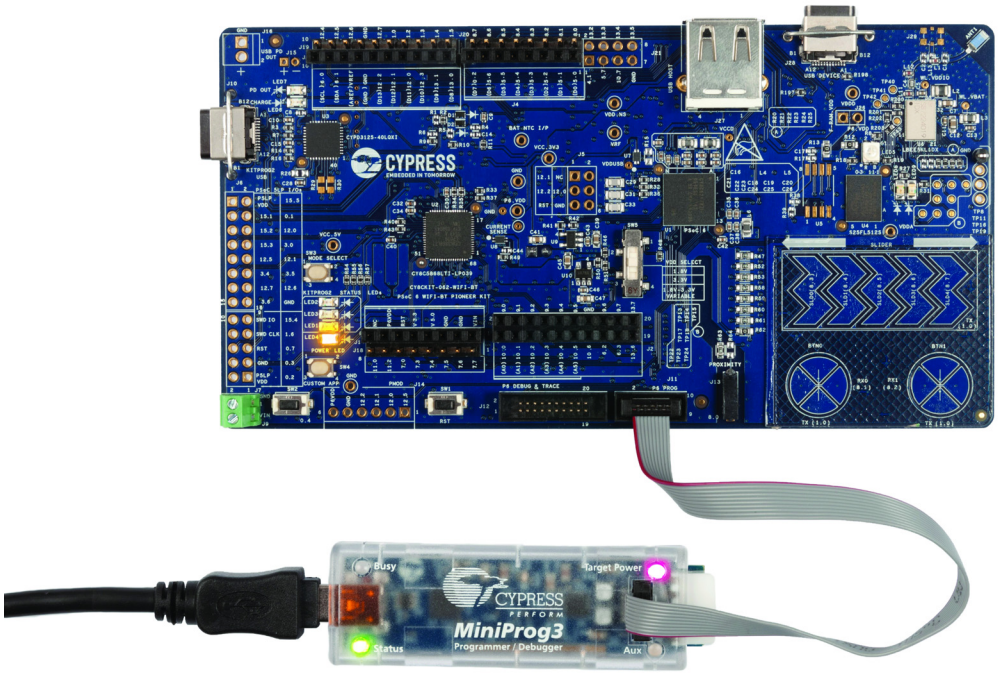
The 10-pin header allows you to program and debug PSoC 6 MCU using an external programmer such as MiniProg3. This header has a protection circuit that cuts-off any voltage greater that 3.4 V on VTARG_REF pin. This is to ensure that PSoC 6 MCU and other 3.3-V devices do not get damaged due to overvoltage.





If the external programmer provides a slightly higher voltage, say 3.42 V, and you still need to use the programmer, you can bypass this protection circuit by populating the bypass zero-ohm resistor R196.

Note that this change will compromise the protection circuit when an external supply is used and will permanently damage any 3.3-V device if the external voltage exceeds absolute maximum limit of the device. See the respective device datasheet for absolute maximum voltage limits.

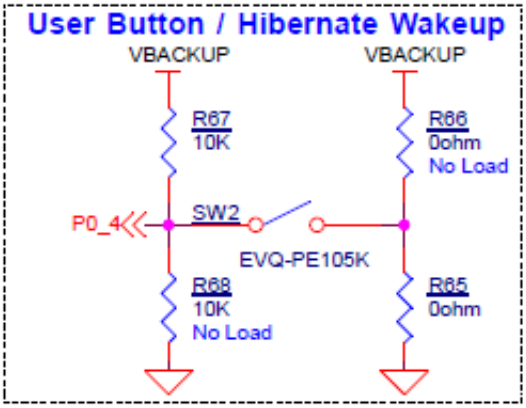


Notes:

1. If you are programming the PSoC 6 MCU using a MiniProg3 connected to J11, make sure the voltage is set at either 2.5 V or at 3.3 V.
2. If you want to program the PSoC 6 MCU using MiniProg3 at the 1.8 V condition, make sure you are populating the 0-ohm resistor R196 on the board. This is to bypass the overvoltage protection circuit, as the protection circuit does not allow programming of the device at 1.8 V.
3. Powering PSoC 6 through MiniProg3 sometimes turns on the LED4. This is due to the reverse conduction from PSoC 6 VDD domain to the VCC_3V3 domain when there is no USB device connected at J10.
4. Do not mount the CY8CKIT-028-Display shield on the PSoC 6 Pioneer board at the time of programming and debugging through the J11 header. This causes extra load on the external programmer, and hence the programmer may not be able to power-up the PSoC 6 supply domain.

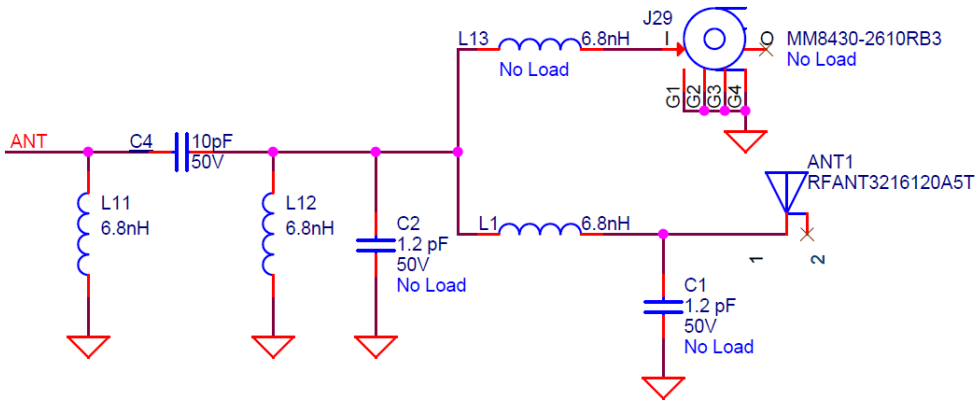
A.3.2 PSoC 6 MCU User Button (SW2)

By default, this button connects the PSoC 6 MCU pin to ground when pressed, and you need to configure the PSoC 6 MCU pin as a digital input with resistive pull-up for detecting the button press. If you want to sense active HIGH on the PSoC 6 MCU pin, resistor R67 should be removed and R68 should be populated. This will connect the button connecting the PSoC 6 MCU pin to VBACKUP when pressed. Additionally, footprints are provided for pull-up and pull-down resistors that can be populated if external pull-up is required.



A.3.3 SWD Connector Receptacle (J29)

This connector can be used for conductive measurements and can also be used to connect external antenna. This is not loaded by default. Remove L1, populate L13, and the SWD connector (J29) to connect the external antenna. See the BOM for recommended part numbers.



Antenna with Matching Network

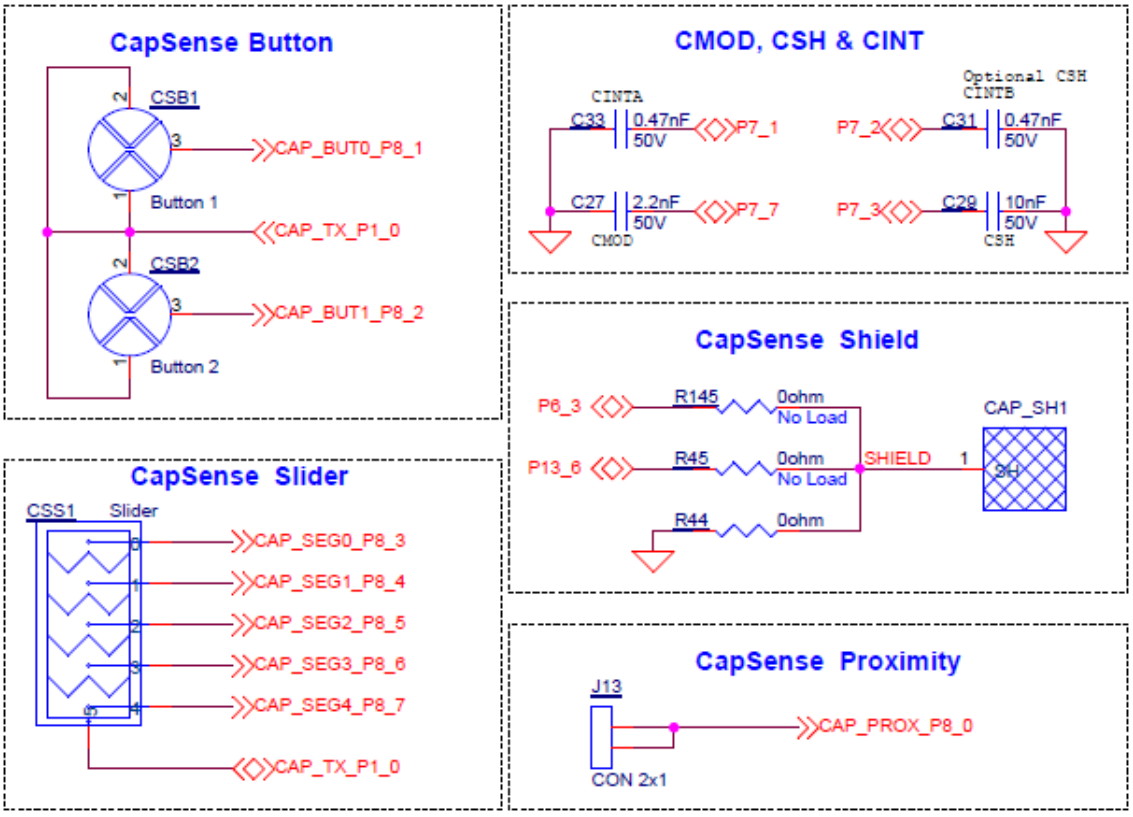
A.3.4 CapSense Shield

The hatched pattern around the CapSense buttons and slider are connected to ground. If liquid tolerance is required, this pattern needs to be connected to the shield pin. This pattern can be connected to either of the two ports P6.3 or P13.6 populated by R138 or R137, respectively. In both cases, resistor R44 connecting the hatched pattern to ground needs to be removed. These pins need to be configured as a shield pin in PSoC Creator.

Connecting the hatched pattern to shield instead of ground will also reduce parasitic capacitance of the sensors.

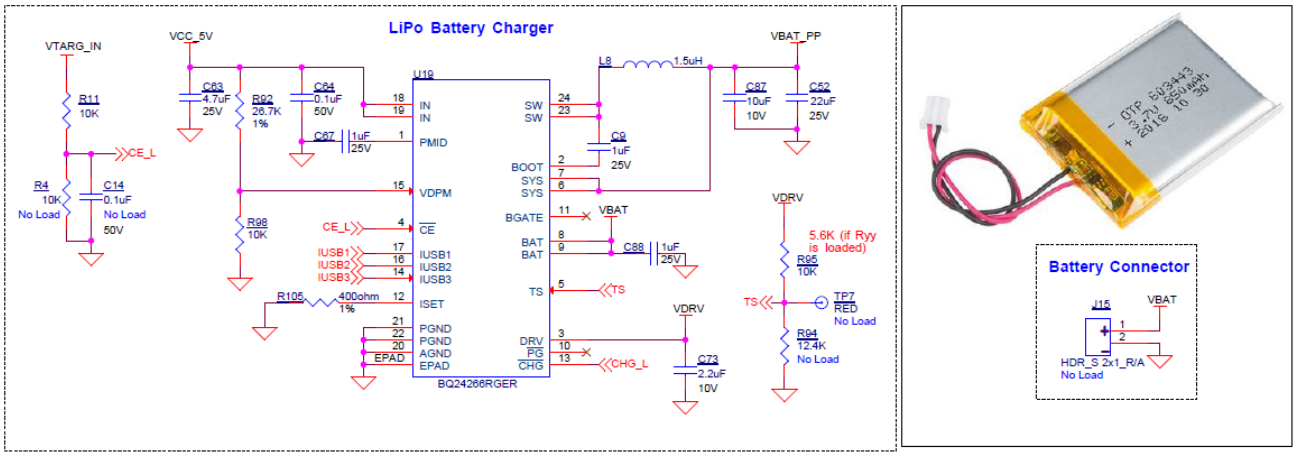
A.3.5 CSH

The shield tank capacitor (CSH) is not populated by default. This capacitor is optional, and can be used for an improved shield electrode driver when CSD sensing is used. You can remove R146 to disconnect port 7.3 from header and populate C29 (10 nF) for CSH. See the bill of material (BOM) for the recommended part number.


CapSense Button
CSB1
3
CAP_BUT0_P8_1
Button 1
CAP_TX_P1_0
CSB2
3
CAP_BUT1_P8_2
Button 2
CMOD, CSH & CINT
CINTA
C33
0.47nF
50V
P7_1
C27
2.2nF
50V
P7_7
CMOD
Optional CSH
CINTB
P7_2
C31
0.47nF
50V
P7_3
C29
10nF
50V
CSH
CapSense Slider
CSS1
Slider
CAP_SEG0_P8_3
CAP_SEG1_P8_4
CAP_SEG2_P8_5
CAP_SEG3_P8_6
CAP_SEG4_P8_7
CAP_TX_P1_0
CapSense Shield
P6_3
R145
0ohm
No Load
P13_6
R45
0ohm
No Load
R44
0ohm
SHIELD
1
CAP_SH1
SH
CapSense Proximity
J13
CAP_PROX_P8_0
CON 2x1

A.3.6 LiPo Battery Charger

Battery connector (J15) for lithium-ion polymer battery charger is not loaded by default; this need to be populated to evaluate battery charging and battery powering option. See the BOM for the recommended part numbers. Recommended lithium-ion polymer rate is 3.7 V at 850 mAH or higher. SparkFun Electronics PRT-13854 or equivalent. batteries can be used



A.3.7 Multiplexed GPIOs

Some PSoC 6 MCU pins are multiplexed with onboard peripherals and are not connected to connectors or other secondary components by default. See Table 4-2 on page 45 for details on modification required to access these pins.

A.4 Bill of Materials

Refer to the BOM files in the following paths in the kit software installed:

- *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\1.0\Hardware\CY8CKIT-062-WiFi-BT\CY8CKIT-062-WiFi-BT PCBA BOM.xlsx*
- *<Install_Directory>\CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit\1.0\Hardware\CY8CKIT-028-TFT\CY8CKIT-028-TFT PCBA BOM.xlsx*

A.5 WICED Configuration

This section describes the parts of the PSoC 6 MCU that are configured by WICED Studio 6.1. This section includes the following tables:

- Items that should not be changed
- Items that should be changed with extreme caution
- Items that can be changed under certain circumstances
- Items that WICED configures only if required

Configurations that should not be changed

System Clocks		
Name	**Configuration**	**Used For**
Clk_Slow	No Divider	Clocking M0+ Core, and DMA/DW. Clk_Slow needs to be the same as Clk_Peri for SDIO to function properly
Peripherial Clocks		
Divider Type	**Divider #**	**Used For**
8 Bit	0	SDIO. Do not share with other peripherals
GPIO		
Port	**Pin**	**Used For**
2	0	SDIO
2	1	SDIO
2	2	SDIO
2	3	SDIO
2	4	SDIO
2	5	SDIO
2	6	SDIO
11	2	SMIF
11	3	SMIF
11	4	SMIF
11	5	SMIF
11	6	SMIF
11	7	SMIF
Interrupts		
Name	**Number**	**Priority**
UDB	122	2
DW1_1	67	3
DW1_3	69	3
SMIF	142	1
Do not disable interrupts for long periods of time; this will have adverse side-effects on the SDIO and SMIF operation.		
UDBs		
UDB configuration should not be changed; it is pre-configured for SDIO and TFT display.		

DMAs		
DMA	**Priority**	**Used For**
DW0_0	1	SDIO
DW0_2	1	SDIO
DW1_1	0	SDIO. Must be highest priority DMA in system
DW1_3	0	SDIO. Must be highest priority DMA in system
Do not create another DMA channel that will block DW1_1 or DW1_3 for a long period. All other channels must be of lower priority and preemptable. Never disable the DMA.		
Bus Access Priority		
DW1 = 0 (Highest) DW0 = 1 CM4 = 2 CM0+ = 2 Do not change, required for SDIO operation		
SysTick		
SysTick is configured to run off CM4 Clock (100 MHz), and is configured to run at 1 msec.		
Trigger Muxes		
The following Trigger Muxes are used for SDIO and should not be modified: (TRIG10_IN_CPUSS_DW0_TR_OUT2, TRIG10_OUT_UDB_TR_DW_ACK4) (TRIG14_IN_UDB_TR_UDB5, TRIG14_OUT_TR_GROUP0_INPUT49) (TRIG0_IN_TR_GROUP14_OUTPUT6, TRIG0_OUT_CPUSS_DW0_TR_IN1) (TRIG14_IN_UDB_TR_UDB0, TRIG14_OUT_TR_GROUP0_INPUT48) (TRIG1_IN_TR_GROUP14_OUTPUT5, TRIG1_OUT_CPUSS_DW1_TR_IN3) (TRIG14_IN_UDB_TR_UDB6, TRIG14_OUT_TR_GROUP0_INPUT47) (TRIG1_IN_TR_GROUP14_OUTPUT4, TRIG1_OUT_CPUSS_DW1_TR_IN1) (TRIG14_IN_UDB_TR_UDB3, TRIG14_OUT_TR_GROUP0_INPUT46) (TRIG0_IN_TR_GROUP14_OUTPUT3, TRIG0_OUT_CPUSS_DW0_TR_IN0)		

Configurations that can be changed, but will have adverse side-effects

System Clocks			
Name	**Configuration**	**Used For**	**Notes**
Clk_Peri	Divder = 1	Source clock for all peripheral	The SDIO clock is always one-quarter of Clk_Peri. Any change in the frequency of Clk_Peri will reduce SDIO clock speed, thus WiFi throughput. Communication with the CY4343W radio will fail if the SDIO clock is slower than 6 MHz. Also the WICED API for UART, SPI, I2C, PWM, CapSense, ADC, and so on, assume that this clock is 100 MHz. So any changes will cause unexpected behavior when using WICED drivers.
Clk_Fast	Divider = 1	Clock for CM4	Reducing the CM4 clock frequency reduces WiFi throughput because the CM4 is unable to process packets as fast
Clk_Hf[0]	Input: Clk_path_0 (FLL) Divider: 1 Frequency: 100 MHz	Clocking CM4, CM0, and Peripheral	Clk_Hf[0] is the root of Clk_Peri, Clk_Fast and Clk_Slow. Any changes will reduce WiFi throughput and cause WICED drivers to have unexpected behavior
Clk_Hf[2]	Input: Clk_path_0 (FLL) Divider: 2 Frequency: 50 MHz	Clock for SMIF (QSPI)	Any changes to this clock will cause the QSPI to run slower
FLL	Input: IMO Output: 100 MHz	Clk_Hf[0] root and Clk_Hf[2]	See Clk_Hf[0] and Clk_HF[2]

Note: If you desire to change these configurations, contact Cypress technical support.

Default configurations in WICED configured; these can be changed if needed.

System Clocks			
Name	**Configuration**	**Used For**	**Notes**
Clk_Hf[1]	Input: Clk_path_1 (PLL) Divider: 1 Frequency: 90 MHz	Clocking audio sub-system	The WICED audio API assume this clock is configured for 90 MHz. If not using WICED audio API feel free to change. If using WICED audio API do not change.
PLL	Input: IMO Output: 90 MHz	Clk_HF[1] root	See note on Clk_Hf[1]
Peripheral Clocks			
Divider Type	**Divider #**	**Used For**	**Notes**
8 Bit	2	CapSense	If not using WICED CapSense API, feel free to change. If using WICED Cap-Sense API, do not change.
16 bit	5	WICED STDIO UART	If not using WICED STDIO UART, feel free to change, otherwise do not change
GPIO			
Port	**Pin**	**Used For**	**Notes**
5	0	WICED STDIO UART	If not using WICED STDIO UART, feel free to change, otherwise do not change
5	1	WICED STDIO UART	If not using WICED STDIO UART, feel free to change, otherwise do not change
0	3	RED LED	Configured for Strong Drive
1	1	Green LED	Configured for Strong Drive
11	1	Blue LED	Configured for Strong Drive
0	4	SW2	Configured for Pull-Up
8	1	CapSense	If not using WICED CapSense, feel free to use
8	2	CapSense	If not using WICED CapSense, feel free to use
8	3	CapSense	If not using WICED CapSense, feel free to use
8	4	CapSense	If not using WICED CapSense, feel free to use
8	5	CapSense	If not using WICED CapSense, feel free to use
8	6	CapSense	If not using WICED CapSense, feel free to use
8	7	CapSense	If not using WICED CapSense, feel free to use

Interrupts			
Name	**Number**	**Priority**	**Notes**
PDM Interrupt	140	6	If not using WICED audio API, feel free to change. If using WICED audio API do not change.
I2S Interrupt	139	6	If not using WICED audio API, feel free to change. If using WICED audio API do not change.
SAR Interrupt	138	6	If not using WICED ADC API, feel free to change. If using WICED ADC API do not change.
All SCB Interrupt	41-48	6	Feel free to change as needed
CSD Interrupt	49	5	Feel free to change as needed
Analog Routing			
When CapSense is used, analog routing is consumed to connect the pins. The following routing code was used to create this CapSense configuration: CY_SET_REG32(CYREG_HSIOM_AMUX_SPLIT_CTL2, 0x00000030u); CY_SET_REG32(CYREG_HSIOM_AMUX_SPLIT_CTL4, 0x00000033u); CY_SET_REG32(CYREG_PASS_AREF_AREF_CTRL, 0x80000000u);			

Table A-2. WICED Names and their Configuration and Mapping in PSoC 6

WICED Name	PSoC Resource and Configuration	Notes
WICED_LED1	P0.3, Strong Drive	Connects to red LED
WICED_LED2	P1.1, Strong Drive	Connects to green LED
WICED_LED3	P11.1, Strong Drive	Connects to blue LED
WICED_BUTTON1	P0.4, Pull_Up	Connects to SW2
STDIO_UART	SCB5: P5.0, P5.1 8N1 115200	
WICED_ADC_1	P10.0 Single Ended (negative input connected to VSSA/2) Unsigned Result Consumes 8 Bit divider # 1	
WICED_PWM_1	Connected to P0.3 Consumes TCPWM Block 0 Counter 1 Consumes 16 bit divider #14	Red LED
WICED_PWM_2	Connected to P1.1 Consumes TCPWM Block 0 Counter 3 Consumes 16 bit divider #14	Green LED
WICED_PWM_3	Connected to P11.1 Consumes TCPWM Block 1 Counter 1 Consumes 16 bit divider #14	Blue LED

Table A-2. WICED Names and their Configuration and Mapping in PSoC 6

WICED Name	PSoC Resource and Configuration	Notes
WICED_SPI_7	Consumes SCB[6] Consumes 16 bit divider #6 MOSI - P12.0 MISO - P12.1 SCLK - P12.2 SS - P12.3	
WICED_I2C_4	Consumes SCB[3] Consumes 16 bit divider #3 SCL - P6.0 SDA - P6.1	
WICED_UART_3	Consumes SCB[2] Consumes 16 bit divider#2 RX - P3.0 TX - P3.1 CTS - P3.2 RTS - P3.3	
WICED_UART_6	Consumes SCB[5] Consumes 16 bit divider#5 RX - P5.0 TX - P5.1	Used for STDIO UART
WICED_I2S_1 WICED_I2S_2	Consumes I2S Block (I2S_1 = Write, I2S_2 = Read) TX SCK - P5.1 TX WS - P5.2 TX_SDO - P5.3 RX_SCLK - P5.4 RX_WS - P5.5 RX_SDI - P5.6	On consumed when WICED Audio API are used

Other Notes:

- WICED has a limited integration of CapSense. This integration only supports the two buttons and five-element slider on the kit. There is no ability to tune in WICED or change the configuration.
- Any WICED names not mentioned in Table A-2 are not fully configured in the *platform.c* file and thus will not function properly.
- Either WICED API or PDL API can be used to configure and use the PSoC 6 MCU peripheral. For ease of use it is recommended to use one or the other. For peripherals such as the STDIO UART and AUDIO, it is recommended to use the WICED API.

A.6 Frequently Asked Questions

1. I don’t have a Type- C connector on my PC. Can I still connect and use this kit?
2. How does CY8CKIT-062-WiFi-BT handle voltage connection when multiple power sources are plugged in?
3. How can I access Smart I/O and other GPIOs connected to onboard peripherals?
4. Why does the Red LED of RGB LED (LED5) light up when switch SW7 is set to SuperCap position?
5. What are the three selection switches on baseboard used for?
6. What is the Jumper onboard for?
7. What are the input voltage tolerances? Are there any overvoltage protection on this kit?
8. Why is the voltage of the kit restricted to 3.3 V? Can it drive external 5-V interfaces?
9. I powered my Arduino board by mistake, while powering the PSoC 6 MCU. Is my PSoC 6 device alive?
10. What type of battery can I use for this kit?
11. I connected the battery with the opposite polarity by mistake. Did I fry the system?
12. Can I charge any kind of Type-C device using this kit?
13. How can I evaluate the USB Type-C provider and consumer features to get started?
14. I am unable to program the target device.
15. Does the kit get powered when I power it from another Cypress kit through the J1 header?
16. What additional overlays can be used with CapSense?
17. What is Pmod?
18. With what type of shield from Cypress can I use this baseboard?
19. Which third-party debuggers does this kit support?
20. Why am I not able to program PSoC 6 MCU using MiniProg3 at 1.8 V?
21. How can SW2 be used for PMIC wake up?

1. I don’t have a Type- C connector on my PC. Can I still connect and use this kit?

   Yes. To evaluate PSoC 6 MCU features, any PC with USB 2.0 connectivity is sufficient. A Type-C power adapter is required only to evaluate the CCG3 section of the kit.

2. How does CY8CKIT-062-WiFi-BT handle voltage connection when multiple power sources are plugged in?

   There are five options to power the baseboard: Type-C USB connector (J10), external DC supply via VIN connector (J9/ J1), debug and trace header (J12, VTARG_IN), program and debug header (J11), and LiPo battery header (J15). Type-C and VIN take priority over other supply options. These inputs are ORed using a diode and the higher voltage between the two take precedence. Output of ORing diode is given to a buck-boost regulator (**U30**) that generates a constant 5.2 V. This output is ORed with ETM supply (J12), which is typically 5 V. For most practical applications, output from the 5.2-V regulator takes priority and the same is given as an input to the voltage regulator (U10). LiPo battery voltage is used when all the above sources are absent. Output of buck regulator (U30) is ORed with supply voltage from the program and debug header (J11), and higher voltage takes precedence. See Table A-1 for more details on voltage input and output scenarios.

3. How can I access Smart I/O and other GPIOs connected to onboard peripherals?

   The Smart I/O (Port 8 and Port 9.3) and GPIO connected to the onboard peripherals are multiplexed with PSoC 6 MCU I/O headers (**J2** and **J20**). By default, some of these I/Os are connected to onboard peripherals using series resistors. These resistors can be changed to route these I/Os to headers. See Table 4-2 on page 45 for the list of resistors that needs to be changed.

4. Why does the red LED (LED5) light up when switch **SW7** is set to SuperCap position?

   This behavior is observed if SuperCap is charged below 1.5 V. The I/Os referring to this domain will leak current, in this case P0[3]. The VBACKUP feature needs to be enabled in the device before switching **SW7** to the SuperCap position. See the device TRM or datasheet for options to enable SuperCap charging.

5. What are the selection switches on baseboard used for?

   Table 4-1 on page 43 gives details on all two selection switches.

6. What is the jumper on board for?

   The jumper J8 can be used to measure current of the PSoC 6 MCU without the need to desolder any component from the board. An ammeter can be connected across this jumper to measure the current consumed by the PSoC 6 MCU. Remove the jumper on J8, connect an ammeter (positive terminal of ammeter to Pin 2), and power the kit though USB connector J10.

7. What are input voltage tolerances? Are there any overvoltage protection on this kit?

   Input voltage level are as follows:

Table A-3. Input voltage levels

Supply	Typical Input Voltage	Absolute Maximum (overvoltage protection)
USB Type-C connector (J10)	4.5 V to 12 V	15 V
VIN connector (J9/J1)	5 V to 12 V	15 V
Debug and trace header (J12)	5 V	5.5 V
Program and debug header (J11)	1.8 V to 3.3 V	3.6 V
Li-Po battery connected (J15)	3.2 V to 4.2 V	5 V

8. Why is the voltage of the kit restricted to 3.3 V? Can it drive external 5-V interfaces?

   PSoC 6 is not meant to be powered for more than 3.6 V. Powering PSoC 6 to more than 3.3 V will damage the chip. You cannot drive the I/O system with more than 3.3-V supply voltages.

9. I powered my Arduino board by mistake, while powering the PSoC 6 MCU. Is my PSoC 6 device alive?

   Yes. The 3.3 V and 5 V on Arduino power header are not input pins and have protection circuit to prevent the voltage from entering the board. VIN is an input pin and this is routed to the regulator, which is capable of taking an absolute maximum of 15 V. The P6.V_{DD} pin is not protected and care should be taken not to supply voltage to this pin.

10. What type of battery can I use for this kit?

    The recommended lithium-ion polymer rating is 3.7 V at 850 mAH or higher. SparkFun Electronics PRT-13854 or equivalent batteries can be used. The LiPo battery charger can charge at 100 mA or 1.5 mA based on whether the USB connection is a legacy device or PD-capable.

11. I connected the battery with the opposite polarity by mistake. Did I fry the system?

    The kit has relevant protection circuits to protect the system from permanent damage. Prolonged connection may lead to damage.

12. Can I charge any kind of Type-C device using this kit?

    The kit is programmed to advertise the VIN voltage with 1 A current rating. 5-V and 12-V devices are the recommended options. VIN needs to be 5 V and 12 V respectively for this to work.

13. How can I evaluate the USB Type-C provider and consumer features to get started?

    You can use any kind of Type-C laptop, mobile phone, or PD adapters based on the feature that you are trying to evaluate. When using as a consumer, note that devices such as laptops may be able to provide only 5 V out and may not support 9 V/12 V without a docking station. To use as a provider, any 5 V/9 V/12 V device that has a current requirement of less that 1 A may be used. Additionally, Cypress has its own USB Type-C evaluation kit, which can be used to evaluate the provider and consumer features and many more. Visit www.cypress.com/products/usb-type-c-and-power-delivery for details on these kits.

14. I am unable to program the target device.

    a. Ensure that SW7 is in VDDD/KITPROG2 position.

b. Make sure that no external devices are connected to J11.
c. Update your KitProg2 version in the programmer to 1.04 or later using the steps mentioned in the KitProg2 User Guide.
d. Ensure that the device used in PSoC Creator is CY8C6247BZI-D54.

15. Is it possible to power the kit from another Cypress kit through the J1 header?

    Yes, VIN pin on the J1 header is the supply input/output pin and can support up to 12 V.

16. What additional overlays can be used with CapSense?

    Any overlay (up to 5 mm thickness) such as wood, acrylic, and glass can be used with CapSense. Note that additional tuning may be required when the overlay is changed.

17. What is Pmod?

    The Peripheral Module or Pmod interface is an open standard defined by Digilent Inc in the Digilent Pmod Interface Specification for peripherals used with FPGAs or microcontrollers. Several module types are available – from simple push buttons to more complex modules with network interfaces, analog to digital converters, or LCD displays. Pmods are available from multiple vendors such as Diligent, Maxim Integrated, Analog Devices, and a variety of hobby sites. This kit supports only 1x6 pin Pmods.

18. What type of shield from Cypress is compatible with this baseboard?

    Any Arduino Uno shield that supports 3.3-V operation is compatible with this Pioneer board. The following Cypress shields are pin-compatible with this board:

    a. CY3280-MBR3
    b. CY8CKIT-022
    c. CY8CKIT-024
    d. CY8CKIT-026
    e. CY8CKIT-040
    f. CY8CKIT-046
    g. CY8CKIT-048

19. Which third-party debuggers does this kit support?

    Multiple third-party IDEs are supported; IAR and uVision are some examples. For more details on all supported devices and procedures to export to these IDEs, see the PSoC Creator ‘Help’ menu.

20. Why am I not able to program PSoC 6 MCU using MiniProg3 at 1.8 V?

    The "Program/Debug Overvoltage Protection" circuit shown in A.3.1 Bypass Protection Circuit on Program and Debug Header (J11) on page 69 does not allow programming of the device at 1.8 V through MiniProg3. If you want to program the PSoC 6 MCU using MiniProg3 at the 1.8-V condition, make sure you are populating the 0-ohm resistor at R196 on the board. This resistor will bypass the protection circuit and will allow programming of the device at 1.8 V. Make sure you are not populating this resistor at any other voltage of operation.

21. How can SW2 be used for PMIC wake up?

    SW2 is connected to the PMIC_Wakeup_In pin (P0.4) of PSoC 6 MCU. A logic high input at the PMIC_Wakeup_In pin can wake up the system and enable the PMIC. See the Backup chapter in PSoC 6 MCU Architecture Technical Reference Manual for more details of this feature.

    SW2 should be externally pulled down to ground to use PMIC control. Moreover, when the switch is pressed, the active HIGH logic should push P0.4 to the VBACKUP supply. However, the kit is configured by default to use active LOW logic as described in A.2.9 Push Buttons on page 66. In addition, in Rev04 version of the kit, the active HIGH logic for SW2 pushes P0.4 to VBACKUP. Therefore, the following re-works on the kit are required to use the PMIC control feature:

    - Remove R65 and populate the 0Ω R66.
    - Remove R67 and populate a 10KΩ resistor at R68.

Revision History

Document Revision History

Document Title: CY8CKIT-062-WiFi-BT, PSoC® 6 WiFi-BT Pioneer Kit Guide Document Number: 002-22677				
Revision	**ECN Number**	**Issue Date**	**Origin of Change**	**Description of Change**
**	6077389	02/21/2018	TAVA/TDU	New kit guide
*A	6114582	03/30/2018	TAVA	Updated Table 4-3. Updated Section 4.2.
*B	6165395	05/03/2018	TAVA	Updated Section A.2.9 Added image in Section A.2.10 Updated Section A.6

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Cypress Semiconductor:
CY8CKIT-062-WIFI-BT