



Please note that Cypress is an Infineon Technologies Company.

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

Continuity of document content

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

Continuity of ordering part numbers

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



CY8CKIT-044

PSoC® 4 M-Series Pioneer Kit Guide

Document Number: 001-96598 Rev. *E

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



Copyrights

© Cypress Semiconductor Corporation, 2015-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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

Contents



Safety Information	6
1. Introduction	7
1.1 Kit Contents	7
1.2 Board Details	9
1.3 PSoC Creator	10
1.3.1 PSoC Creator Code Examples	11
1.3.2 Kit Code Examples	11
1.3.3 PSoC Creator Help	12
1.4 Getting Started	12
1.5 Additional Learning Resources	12
1.6 Technical Support	13
1.7 Document Conventions	13
1.8 Acronyms	14
2. Installation	15
2.1 Before You Begin	15
2.2 Install Software	15
2.3 Uninstall Software	17
3. Kit Operation	18
3.1 Theory of Operation	18
3.2 KitProg	21
3.3 Programming and Debugging PSoC 4200M Device	21
3.3.1 Programming using PSoC Creator	21
3.3.2 Debugging using PSoC Creator	22
3.3.3 Programming using PSoC Programmer	22
3.4 USB-UART Bridge	22
3.5 USB-I2C Bridge	22
3.6 Mass Storage Programmer	22
3.7 Updating the KitProg Firmware	22
4. Code Examples	23
4.1 Using the Kit Code Examples	23
4.2 Deep-Sleep Blinky	25
4.2.1 Project Description	25
4.2.2 Hardware Connections	25
4.2.3 Verify Output	25
4.3 CapSense Proximity	26

4.3.1	Project Description.....	26
4.3.2	Hardware Connections	26
4.3.3	Verify Output.....	26
4.4	Proximity Gestures	27
4.4.1	Project Description.....	27
4.4.2	Hardware Connections	28
4.4.3	Verify Output.....	29
4.5	Touch Gestures.....	30
4.5.1	Project Description.....	30
4.5.2	Hardware Connections	32
4.5.3	Verify Output.....	32
4.6	Accelerometer	33
4.6.1	Project Description.....	33
4.6.2	Hardware Connections	34
4.6.3	Verify Output.....	34
4.7	Sensor Hub.....	35
4.7.1	Project Description.....	35
4.7.2	Hardware Connections	36
4.7.3	Verify Output.....	36
4.8	Raspberry Pi.....	38
4.8.1	Project Description.....	38
4.8.2	Hardware Connections	38
4.8.3	Setting Up Raspberry Pi	38
4.8.4	Verify Output.....	38
A.	Appendix	39
A.1	Schematics.....	39
A.2	Hardware Functional Description.....	43
A.2.1	PSoC 4200M	43
A.2.2	PSoC 5LP	43
A.2.3	Power System	44
A.2.4	Expansion Connectors.....	46
A.2.5	USB Mini-B Connector.....	47
A.2.6	CapSense Circuit.....	47
A.2.7	Pioneer Board LEDs.....	47
A.2.8	Push Buttons	47
A.2.9	Cypress Ferroelectric RAM (F-RAM)	47
A.2.10	3-Axis Accelerometer.....	47
A.2.11	PWM Temperature Sensor	48
A.2.12	Ambient Light Sensor	48
A.2.13	Serial Interconnection Between PSoC 5LP and PSoC 4200M	48
A.2.14	Raspberry Pi Compatible Header	49
A.2.15	Level Translators	50
A.2.16	Test Points.....	51
A.3	Using FM24V10 F-RAM	52
A.3.1	Address Selection.....	52
A.3.2	Write/Read Operation	52

A.4	Migrating Projects Across Different Pioneer Series Kits	54
A.4.1	Arduino Uno Compatible Headers	54
A.4.2	Onboard Peripherals.....	56
A.5	KitProg Status LED States	57
A.6	Bill of Materials	58
Revision History	62

Safety Information



The CY8CKIT-044 PSoC® 4 M-Series Pioneer Kit is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. For this reason, the board may cause interference with other electrical or electronic devices in close proximity. In a domestic environment, this product may cause radio interference. In such cases, the user may be required to take adequate preventive measures. Also, this board should not be used near any medical equipment or RF devices.

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

The CY8CKIT-044 as shipped from the factory has been verified to meet with the requirements of CE as a Class A product.

	<p>The CY8CKIT-044 board contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment which can cause a discharge without detection. Permanent damage may occur to devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-044 boards in the protective shipping package.</p>
	<p>End-of-Life/Product Recycling</p> <p>The end-of-life cycle for this kit is five years from the date of manufacture mentioned on the back of the box. Contact the nearest recycler to discard the kit.</p>

General Safety Instructions

ESD Protection

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

Handling Boards

CY8CKIT-044 board is sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad if available. Do not slide the board over any surface.

1. Introduction



Thank you for your interest in the CY8CKIT-044 PSoC® 4 M-Series Pioneer Kit. The PSoC 4 M-Series Pioneer Kit enables customers to evaluate and develop projects using the PSoC 4200M device family.

PSoC 4200M device family is an expansion to the PSoC 4 device family that offers more flash, GPIOs, and programmable analog and digital blocks. PSoC 4200M device is a true programmable embedded system-on chip, integrating custom analog and digital peripheral functions, memory, and an Arm® Cortex®-M0 microcontroller on a single chip. The programmable analog and digital peripheral functions allow higher flexibility, in-field tuning of the design, and faster time-to-market.

The PSoC 4 M-Series Pioneer Kit offers footprint-compatibility with Arduino™ Shields, 6-pin Digilent® Pmod™ Peripheral Modules, and Raspberry Pi boards. This kit features two CapSense® Proximity headers, a CapSense Gesture Pad, an ambient light sensor, a 3-Axis accelerometer, a PWM temperature sensor, an RGB LED, a push button switch, an onboard programmer/debugger and USB-UART/I2C bridge functionality block (KitProg), and a Cypress F-RAM™. This kit supports operating voltages of 3.3 V or 5 V.

You will use PSoC Creator™ to develop and debug your PSoC 4200M device projects. PSoC Creator is Cypress' standard integrated design environment (IDE). If you are new to PSoC Creator, see the documentation on the [PSoC Creator home page](#). You can also refer the application note [AN79953 – Getting Started with PSoC 4](#), which gives an introduction to the PSoC 4 device.

1.1 Kit Contents

The CY8CKIT-044 PSoC 4 M-Series Pioneer Kit contains the following, as shown in [Figure 1-1](#).

- PSoC 4 M-Series Pioneer board
- USB Standard-A to Mini-B cable
- Four jumper wires (4 inches each)
- Two proximity sensor wires (5 inches each)
- Quick Start Guide

Figure 1-1. Kit Contents

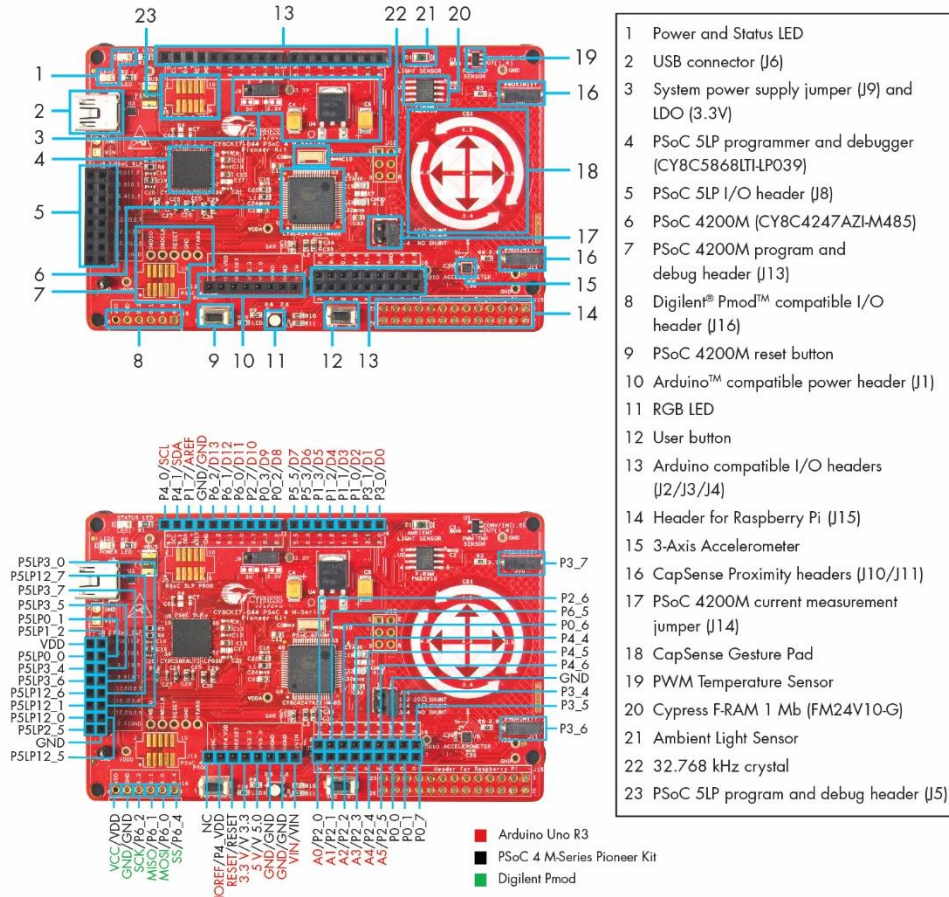


Inspect the contents of the kit; if you find any part missing, contact your nearest Cypress sales office for help: www.cypress.com/support.

1.2 Board Details

The PSoC 4 M-Series Pioneer Kit consists of the following blocks, as shown in [Figure 1-2](#). It also shows the pin mapping of all headers.

Figure 1-2. PSoC 4 M-Series Pioneer Kit Markup

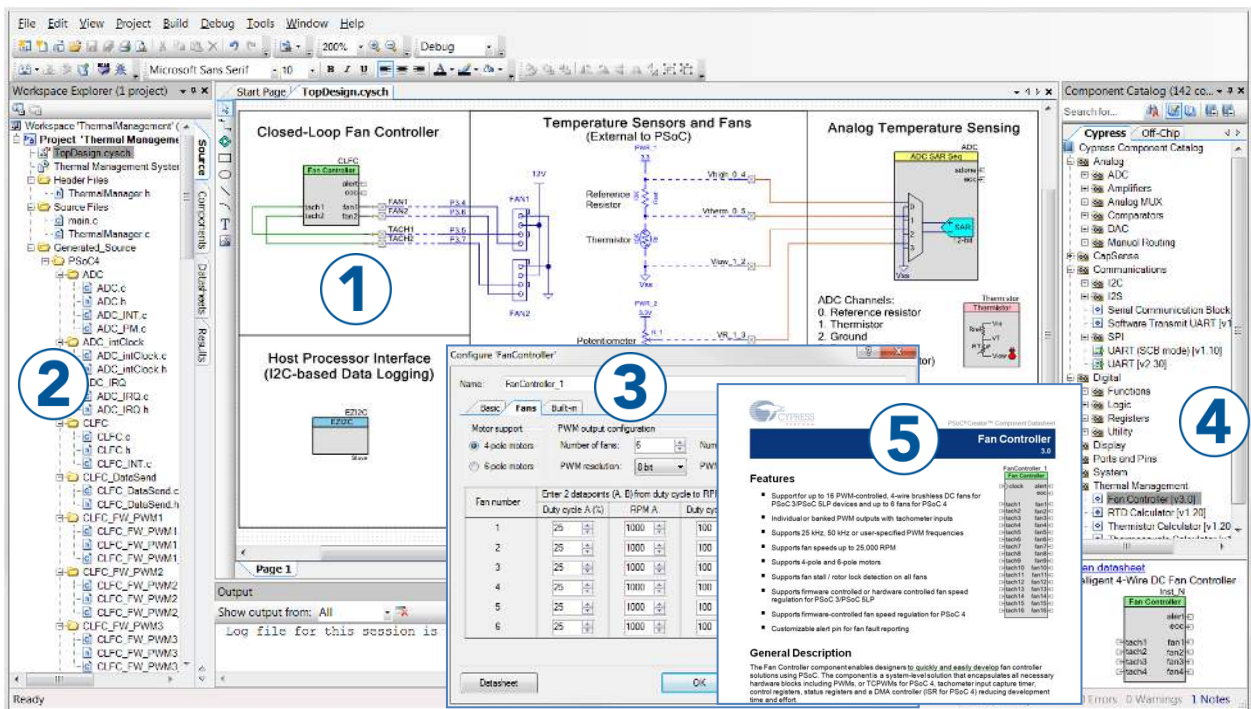


1.3 PSoC Creator

PSoC Creator is a state-of-the-art, easy-to-use IDE. It introduces revolutionary hardware and software codesign, powered by a library of preverified and precharacterized PSoC Components™, as shown in Figure 1-3. With PSoC Creator, you can:

- Drag and drop Components to build your hardware system design in the main design workspace
- Co-design your application firmware with the PSoC hardware
- Configure Components using configuration tools
- Explore the library of 100+ Components
- Access Component datasheets

Figure 1-3. PSoC Creator Features



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

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

1.3.1 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 1-4.

Code examples can speed up your design process by starting you off with a complete design, instead of a blank page. The code examples also show how to use PSoC Creator Components for various applications. Code examples and documentation are included, as shown in Figure 1-5.

In the Find Code Example dialog shown in Figure 1-5, 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. You can also copy and paste code from this window to your project, which can help speed up code development.
- Create new workspace for the code example. 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 1-4. Code Examples in PSoC Creator

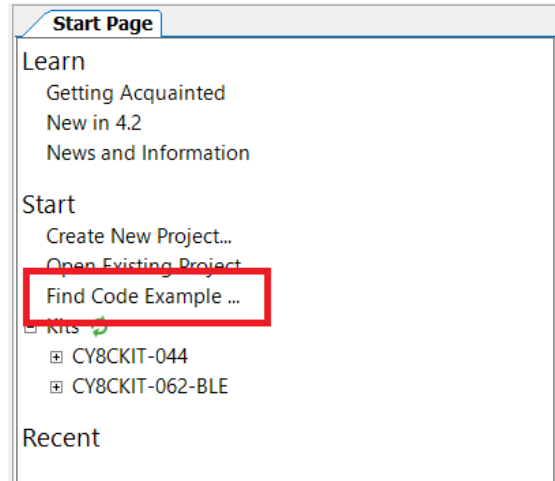
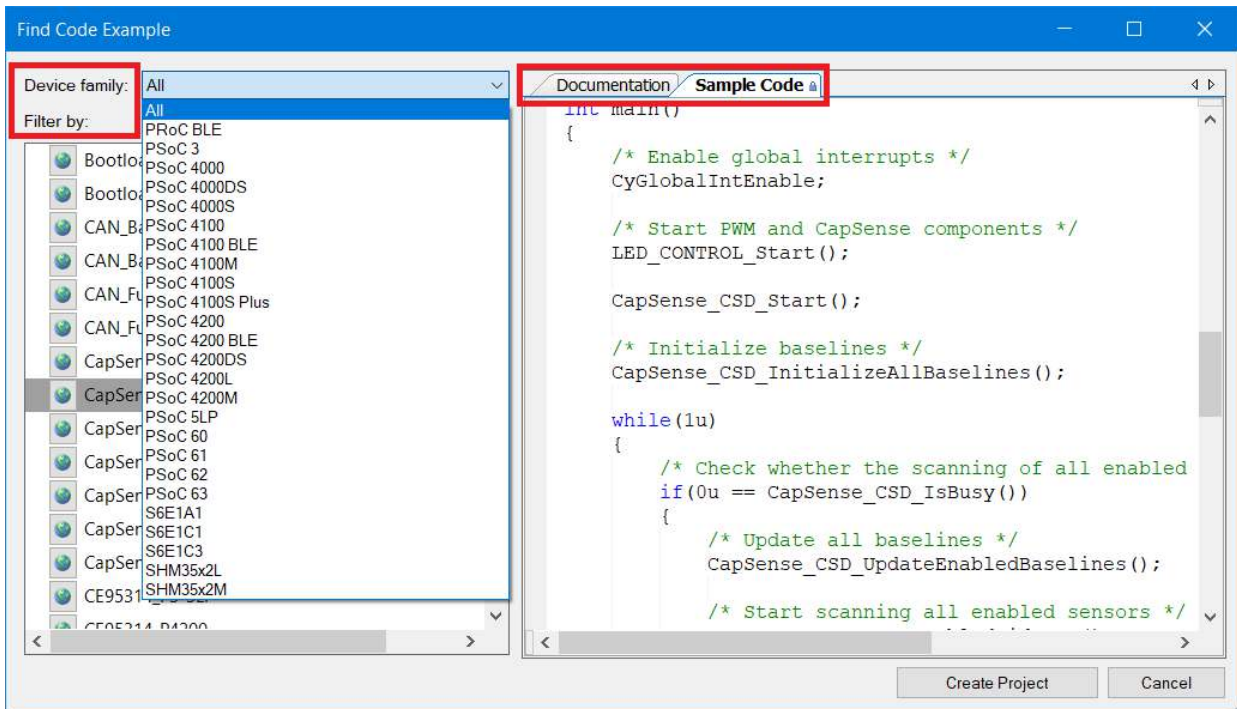


Figure 1-5. Sample Code Example



1.3.2 Kit Code Examples

You can access the installed kit code examples from the PSoC Creator Start Page. To access these code examples, expand the **Kits** under the section **Start**; then, expand the specific kit to see the code examples. Refer to the [Using the Kit Code Examples](#) section for a detailed explanation on how to access the kit code examples.

1.3.3 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 code examples demonstrate how to configure and use PSoC Creator Components. To access code examples related to a specific Component, place the Component on the TopDesign schematic and right-click on the Component. 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**. Visit the [PSoC 4 Component Datasheets](#) page for a list of all PSoC 4 Component datasheets.

1.4 Getting Started

This guide will help you be acquainted with the CY8CKIT-044 PSoC 4 M-Series Pioneer Kit:

- The [Installation](#) chapter describes the installation of the kit software. This includes the PSoC Creator IDE to develop and debug the applications, and PSoC Programmer to program the `.hex` files on to the device.
- The [Kit Operation](#) chapter describes the major features of the PSoC 4 M-Series Pioneer Kit and functionalities such as programming, debugging, and the USB-UART and USB-I2C bridges.
- The [Code Examples](#) chapter describes multiple PSoC 4200M code examples that will help you understand how to create your own PSoC 4 projects.
- The [Appendix](#) provides the detailed hardware description, method to use the onboard F-RAM, kit schematics, and the bill of materials (BOM).

1.5 Additional Learning Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see [KBA86521, How to Design with PSoC 3, PSoC 4, and PSoC 5LP](#). The following is an abbreviated list for PSoC 4:

- **Overview:** [PSoC Portfolio](#) and [PSoC Roadmap](#).
- **Product Selectors:** [PSoC 4 Product Selector](#). In addition, [PSoC Creator](#) includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the [PSoC 4000](#), [PSoC 4100](#), [PSoC 4200](#), [PSoC 4100M](#), and [PSoC 4200M](#) device families.
- **CapSense Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 4 family of devices.
- **Application Notes and Code Examples:** Cover a broad range of topics, from basic to advanced. Many of the application notes include code examples. Visit the [PSoC 3/4/5 Code Examples](#) web page for a list of all available PSoC Creator code examples. To access code examples from within PSoC Creator – see [PSoC Creator Code Examples](#).
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each PSoC 4 device family.
- **Development Kits:**
 - [CY8CKIT-042](#) and [CY8CKIT-040](#) are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino compatible Shields and Digilent Pmod Peripheral Modules.
 - [CY8CKIT-049](#) is a low-cost prototyping platform for sampling PSoC 4 devices.
 - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
 - [MiniProg3](#) device provides an interface for flash programming and debug.
- **Knowledge Base Articles (KBA):** Provide design and application tips from experts on using the device. For instance, [KBA93541](#) explains how to use [CY8CKIT-049](#) to program another PSoC 4.

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

1.6 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).

1.7 Document Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user-entered text, and source code: C:\...cd\icc\
<i>Italics</i>	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Creator User Guide</i> .
[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	Description
ADC	Analog-to-Digital Converter
ESD	Electrostatic Discharge
F-RAM	Ferroelectric Random Access Memory
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
ILO	Internal Low-speed Oscillator
IMO	Internal Main Oscillator
LDO	Low Drop Out (voltage regulator)
LED	Light-Emitting Diode
LFCLK	Low-frequency Clock
LSB	Least Significant Byte
MSB	Most Significant Byte
PSoC	Programmable Systems-on-Chip
PWM	Pulse Width Modulation
RGB	Red Green Blue
SAR	Successive Approximation Register
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

2. Installation



This chapter describes the steps to install the software tools and packages on a PC for using the CY8CKIT-044 PSoC 4 M-Series Pioneer Kit. This includes the IDE on which the projects will be built and used for programming.

2.1 Before You Begin

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

2.2 Install Software

Follow these steps to install the CY8CKIT-044 PSoC 4 M-Series Pioneer Kit software:

1. Download the PSoC 4 M-Series Pioneer Kit software from www.cypress.com/CY8CKIT-044. The kit software is available in three different formats for download.
 - a. **CY8CKIT-044 Kit Complete Setup:** This installation package contains the files related to the kit. However, it does not include the Windows Installer or Microsoft .NET framework packages. If these packages are not on your computer, the installer directs you to download and install them from the Internet.
 - b. **CY8CKIT-044 Kit Only:** This executable file installs only the kit contents, which include kit code examples, hardware files, and user documents. This package can be used if all the software prerequisites (listed in step 5) are installed on your PC.
 - c. **CY8CKIT-044 DVD ISO:** This file is a complete package, stored in a DVD-ROM image format, which you can use to create a DVD or extract using an ISO extraction program such as WinZip® or WinRAR. The file can also be mounted like a virtual CD/DVD using virtual drive programs such as Virtual CloneDrive and MagicISO. This file includes all the required software, utilities, drivers, hardware files, and user documents.
2. If you have downloaded the ISO file, mount it in a virtual drive. Extract the ISO contents if you do not have a virtual drive to mount. Double-click *cyautorun.exe* in the root directory of the extracted content or the mounted ISO if “Autorun from CD/DVD” is not enabled on the PC. The installation window will appear automatically.

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

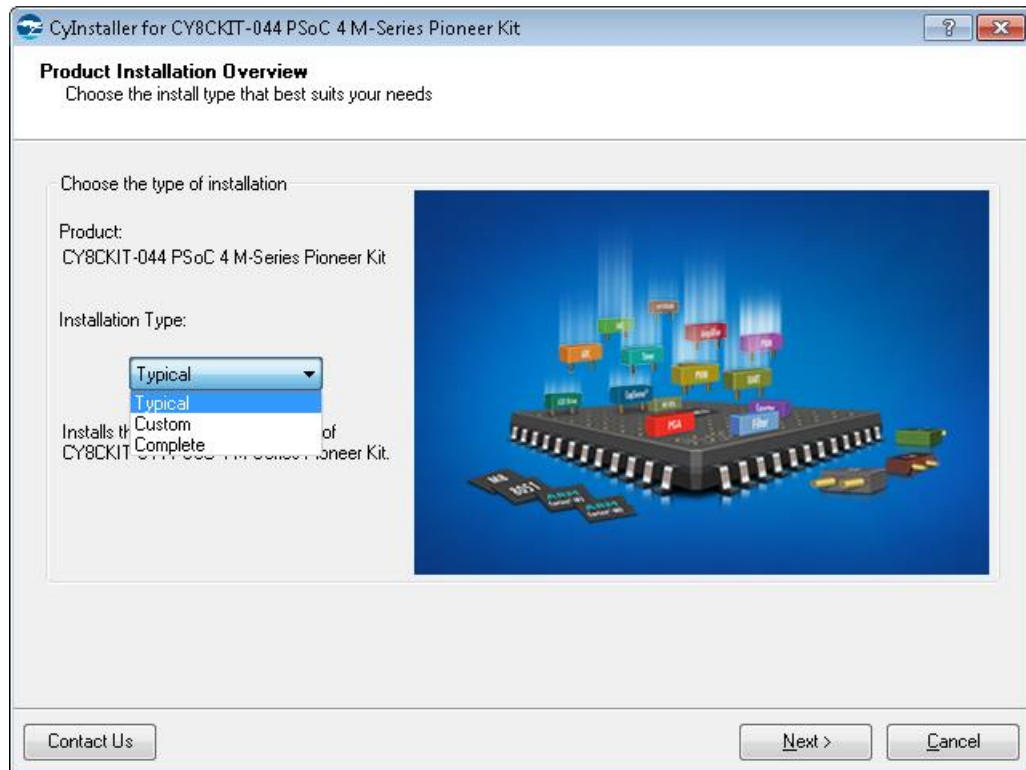
3. Click **Install CY8CKIT-044** to start the kit installation, as shown in [Figure 2-1](#).

Figure 2-1. Kit Installer Screen



4. Select the folder in which you want to install the CY8CKIT-044 kit-related files. Choose the directory and click **Next**.
5. When you click **Next**, the CY8CKIT-044 installer automatically installs the required software, if it is not present on your computer. Following is 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 3.27.1
 - Peripheral Driver Library 3.0.1
6. Choose the Typical/Custom/Complete installation type in the **Product Installation Overview** window, as shown in [Figure 2-2](#). Click **Next** after you select the installation type.

Figure 2-2. Product Installation Overview



7. Read the License agreement and select **I accept the terms in the license agreement** to continue with installation. Click **Next**.
8. When the installation begins, a list of packages appears on the installation page. A green check mark appears next to each package after successful installation.
9. Enter your contact information or select the check box **Continue Without Contact Information**. Click **Finish** to complete the CY8CKIT-044 kit software installation.
10. After the installation is complete, the kit contents are available at the following location:

<Install_Directory>\CY8CKIT-044 PSoC 4 M-Series Pioneer Kit

Default location:

Windows 7 (64-bit): C:\Program Files (x86)\Cypress\CY8CKIT-044 PSoC 4 M-Series Pioneer Kit

Windows 7 (32-bit): C:\Program Files\Cypress\CY8CKIT-044 PSoC 4 M-Series 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, follow the steps outlined in the [Code Examples](#) chapter. 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 Uninstall Software

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

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

3. Kit Operation



This chapter introduces you to the PSoC 4 M-Series Pioneer Kit features.

3.1 Theory of Operation

The PSoC 4 M-Series Pioneer Kit is built around the PSoC 4200M device. Figure 3-1 shows the block diagram of the PSoC 4200M device. Refer to the PSoC 4200M datasheet for details on the device features.

Figure 3-1. PSoC 4200M Block Diagram

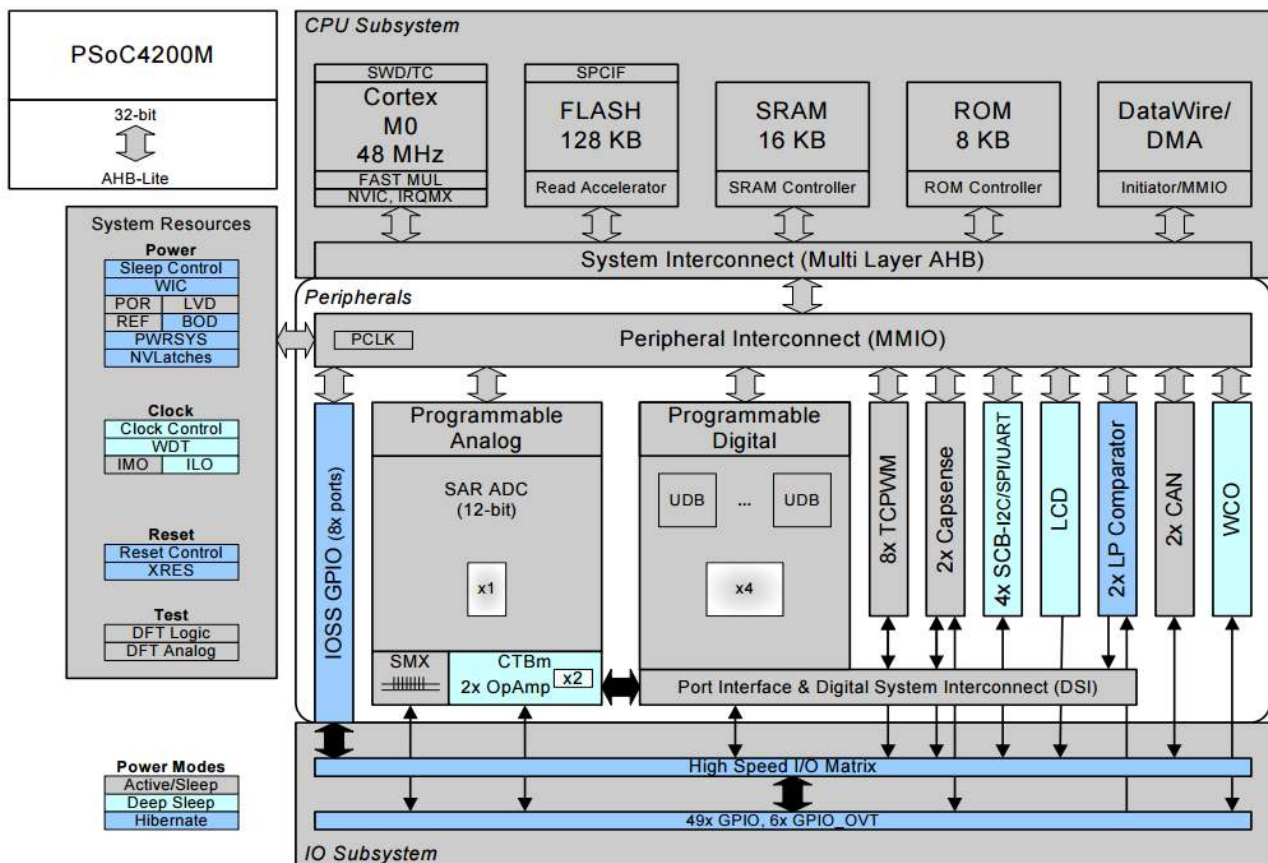


Figure 3-2 shows the block diagram for the PSoC 4 M-Series Pioneer Kit.

Figure 3-2. Block Diagram of PSoC 4 M-Series Pioneer Kit

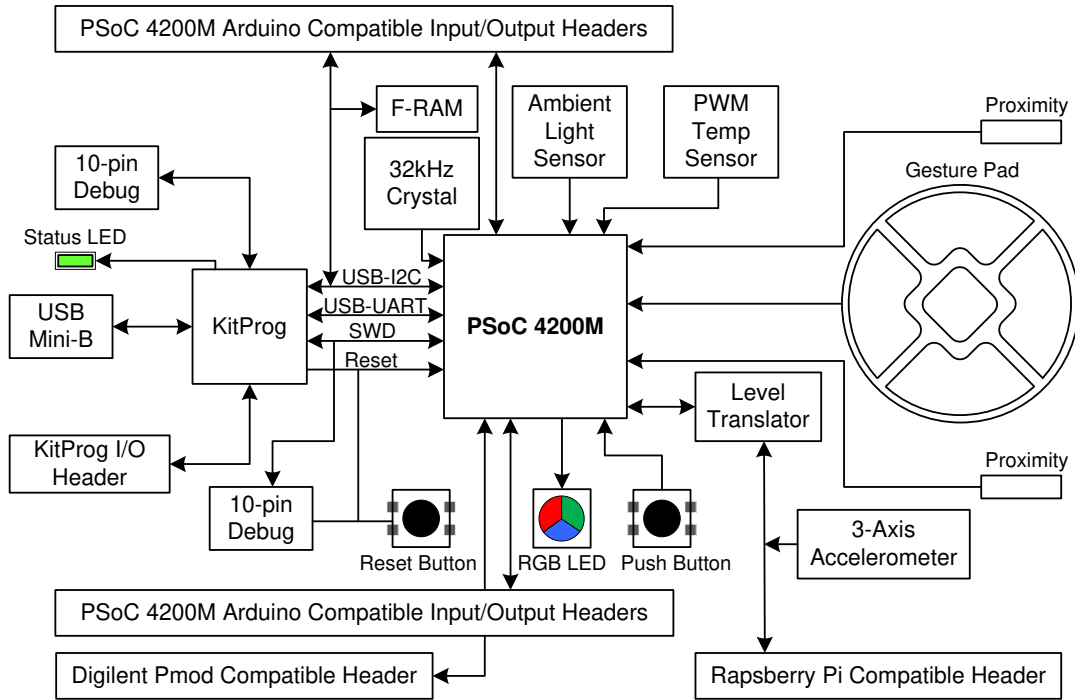
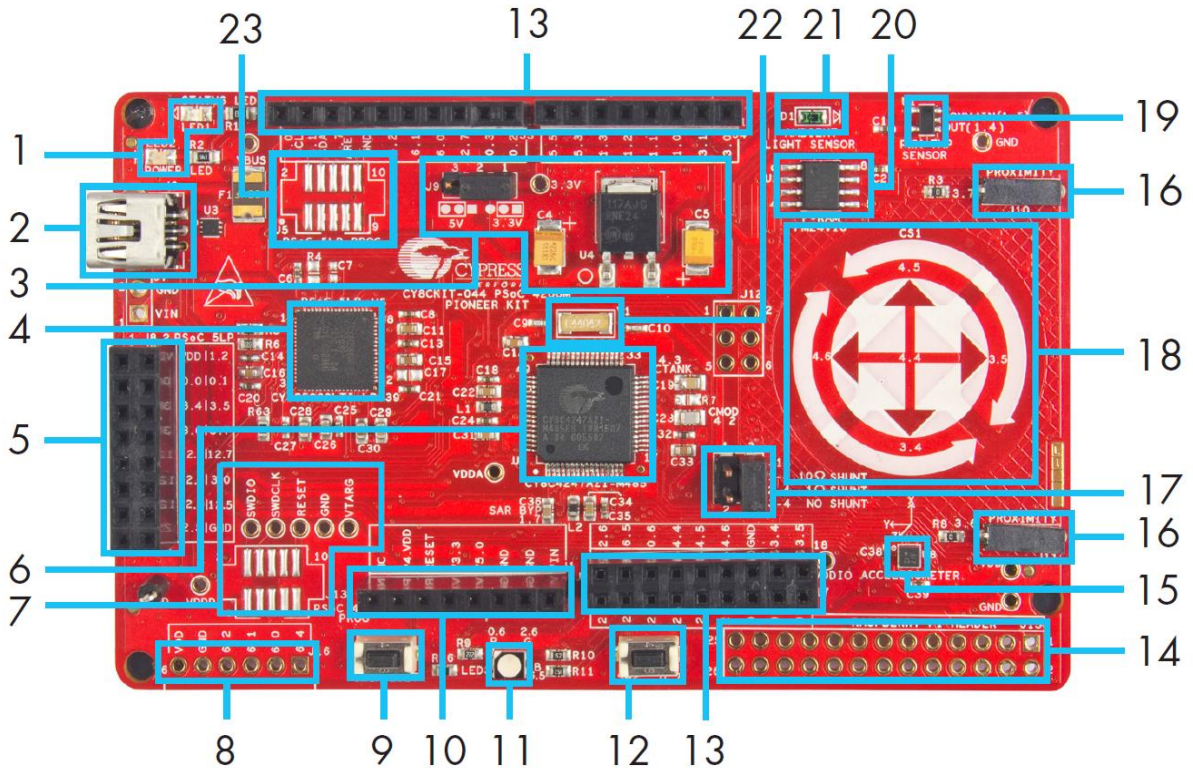


Figure 3-3. PSoC 4 M-Series Pioneer Kit Board Diagram



This kit has the following peripherals:

1. **Power and Status LED:** The amber power LED is turned on when the kit is connected to a power source. The green status LED indicates the status of the [KitProg](#).
2. **USB Mini-B connector:** The USB cable provided along with the PSoC 4 M-Series Pioneer Kit connects between this connector and the PC.
3. **Power Supply:** The kit can operate at two voltage levels: 3.3 V and 5 V. The power selection jumper J9 is used to select between the two voltage levels. The kit has an onboard 3.3-V Low Drop Out (LDO) regulator that converts the 5-V input to 3.3 V.
4. **KitProg:** The KitProg is a multi-functional system, which includes a programmer, debugger, USB-I2C bridge, and a USB-UART bridge. Refer to the *KitProg User Guide* in the installation folder <Install_Directory>\CY8CKIT-044 PSoC 4 M-Series Pioneer Kit\<version>\Documentation\KitProg_User_Guide.pdf for more details on the KitProg.
5. **KitProg (PSoC 5LP) I/O Header:** This header brings out a few Input/Output lines of the onboard PSoC 5LP device. This includes the KitProg USB-I2C bridge lines and USB-UART bridge lines. The additional PSoC 5LP pins are direct connections to the internal programmable analog logic of PSoC 5LP. You can use these pins as General-purpose input/output (GPIO) also. For more details on the KitProg functionality, refer to the *KitProg User Guide*.
6. **PSoC 4200M:** This kit is designed to highlight the features of [PSoC 4200M](#).
7. **10-pin programming header for PSoC 4200M:** This header allows you to program/debug the PSoC 4200M using an external programmer such as [MiniProg3](#). PSoC 4 M-Series Pioneer Kit also has through-hole test points to allow you to program and debug external PSoC 4 and PSoC 5LP devices using the KitProg. You must remove the zero-ohm resistors R51, R53, and R55 to disconnect the connection between KitProg and the onboard PSoC 4200M device before programming an external PSoC.
8. **Digilent Pmod Peripheral Module compatible header:** This header allows you to connect to the PSoC 4200M device to Pmod Peripheral Modules through I2C and SPI interfaces.
9. **Reset Button:** This button is used to reset the PSoC 4200M device.
10. **Arduino Uno Shield compatible power header:** This header powers third-party Arduino Shields. This header also has a provision to power the PSoC 4 M-Series Pioneer Kit through the VIN input on the header.
11. **RGB LED:** The onboard RGB LED can be controlled from the PSoC 4200M device.
12. **User Button:** This button can be used to provide input to the PSoC 4200M. Note that the switch connects the PSoC 4200M pin to ground when pressed, so you need to configure the PSoC 4200M pin as resistive pull-up for detecting the switch press.
13. **Arduino Uno Shield compatible I/O headers:** The PSoC 4 M-Series Pioneer Kit is compatible and operates flawlessly with third-party Arduino Shields.
14. **Raspberry Pi compatible header:** A header which is compatible with Raspberry Pi Model B is included on the kit for enabling advanced embedded systems. A Raspberry Pi can communicate to the PSoC 4200M device using I2C, SPI, or UART communication protocols. One of the applications of this header is to enable the PSoC 4200M to act as an analog input processor for Raspberry Pi. The PSoC 4200M, with its CapSense capability, can also enable user interface functions such as gesture and proximity detection.
15. **3-Axis Accelerometer:** This kit features a 3-Axis digital accelerometer device from Kionix® (KXTJ2-1009). The accelerometer is interfaced to the PSoC 4200M using the I2C protocol.
16. **CapSense Proximity Headers:** These headers are located on both sides of the CapSense Gesture Pad. They can be used as individual proximity sensors or can be used together to detect CapSense proximity gestures. The code example included with this kit demonstrates the detection of wave up and wave down proximity gestures. Remember to connect wires into the proximity headers before using the proximity functionality. Note that if these inputs are not used for proximity sensing, then the connectors and the associated I/Os are available for general-purpose use.
17. **Current Measurement Jumper:** This jumper can be used to measure the current consumed by the PSoC 4200M device by connecting an ammeter between the pins 3 and 4 of the jumper. Remember to connect the jumper between pin 3 and pin 4 if the current measurement functionality is not used.
18. **CapSense Gesture Pad:** The CapSense Gesture Pad is a five-element CapSense peripheral specifically intended for touch gesture recognition. This peripheral can be used to detect swipe left, swipe right, swipe up, swipe down, clockwise swipe, and counter-clockwise swipe gestures.

Note: The Gesture Pad can also be used as five individual capacitive touch buttons. The Gesture Pad is not intended to be used as a radial slider.

19. PWM Temperature Sensor: The kit features a digital temperature sensor from Analog Devices (TMP05). This sensor measures the ambient temperature and reports it to the PSoC 4200M device by varying the duty cycle of the PWM output generated by the sensor.
20. F-RAM: This kit features a Cypress F-RAM device of 1 Mb capacity. The F-RAM is connected to the I2C interface of the PSoC 4200M device. The Cypress F-RAM device can be used similar to an external EEPROM memory for data logging operations. The F-RAM can also be used for bootloading the PSoC 4200M device.
21. Ambient Light Sensor: This is a photodiode connected in photoconductive mode to detect variations in the ambient light intensity. The conductivity of this sensor increases when the ambient light intensity increases and results in a higher current flow through the sensor. The PSoC 4200M internal opamps are used to implement a current to voltage converter to detect these current variations. The resultant output voltage is measured using the internal SAR ADC of the PSoC 4200M device to determine the ambient light intensity. Refer to the TEMD6200FX01 datasheet from Vishay® Semiconductors for more information on the electrical characteristics of the sensor.
22. 32-kHz Crystal Oscillator: The PSoC 4 M-Series Pioneer Kit has a 32-kHz crystal oscillator to provide an accurate low-frequency clock for the PSoC 4200M device.
23. 10-pin programming header for PSoC 5LP: This header is used to program/debug the PSoC 5LP using an external programmer such as [MiniProg3](#).

The subsequent sections explain the method to interface the PSoC 4 M-Series Pioneer Kit to a PC.

3.2 KitProg

The KitProg is a multi-functional system, which includes a programmer, debugger, USB-I2C bridge, and a USB-UART bridge. The PSoC 4 M-Series Pioneer Kit also supports Mass Storage Programming using KitProg. Cypress PSoC 5LP device is used to implement KitProg functionality. The KitProg is integrated in most PSoC development kits. For more details on the KitProg functionality, refer to the *KitProg User Guide* in the kit installation directory: <Install_Directory>\CY8CKIT-044 PSoC 4 M-Series Pioneer Kit\<version>\Documentation\ KitProg_User_Guide.pdf.

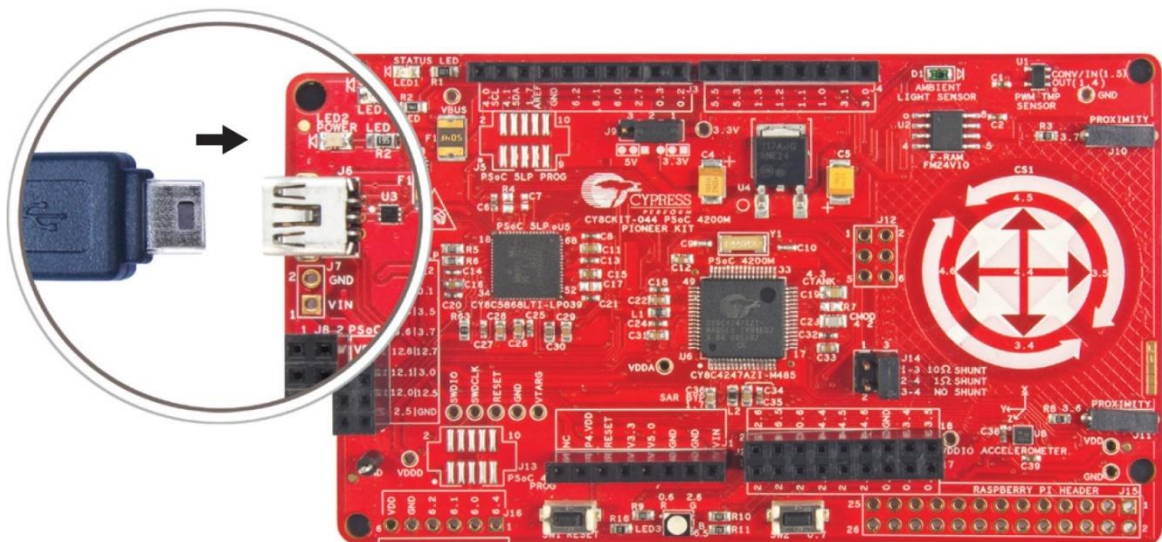
3.3 Programming and Debugging PSoC 4200M Device

The PSoC 4 M-Series Pioneer Kit can be programmed and debugged using the KitProg. Before programming the device, ensure that PSoC Creator and PSoC Programmer software are installed on the PC. See the [Install Software](#) section for more information.

3.3.1 Programming using PSoC Creator

1. Connect the USB cable into the USB connector, J6, as shown in [Figure 3-4](#). The kit enumerates as a composite device if you are plugging in the PSoC 4 M-Series Pioneer Kit to your PC for the first time.

Figure 3-4. Connect USB Cable to J6



2. Open the desired project in PSoC Creator. For this, go to **File > Open > Project/Workspace**. This provides the option to browse to and open your saved project.
3. Select the option **Build > Build <Project Name>** or pressing **[Shift] [F6]** to build the project.
4. If there are no errors during build, program the firmware into the kit by choosing **Debug > Program** or pressing **[Ctrl] [F5]**. This programs the device on the PSoC 4 M-Series Pioneer Kit.

3.3.2 Debugging using PSoC Creator

PSoC Creator has an integrated debugger. You can start the debugger by selecting **Debug > Debug** or by pressing **[F5]**. Refer to section 3.2 of the [KitProg User Guide](#) for a detailed explanation on how to debug using PSoC Creator.

3.3.3 Programming using PSoC Programmer

PSoC Programmer (3.27.1 or later) can be used to program existing *.hex* files on to the PSoC 4 M-Series Pioneer Kit. Refer to section 3.3 of the [KitProg User Guide](#) for a detailed explanation on how to program using PSoC Programmer.

3.4 USB-UART Bridge

The KitProg on the PSoC 4 M-Series Pioneer Kit can act as a USB-UART bridge. The UART lines between the PSoC 4200M and the KitProg are hard-wired on the board, with UART_RX assigned to **P7[0]** and UART_TX assigned to **P7[1]** on PSoC 4200M. For more details on the KitProg USB-UART functionality, refer to the [KitProg User Guide](#).

3.5 USB-I2C Bridge

The KitProg can function as a USB-I2C bridge and communicate with the software utility Bridge Control Panel (BCP). The I2C lines on PSoC 4200M device are **P4[1] (SDA)** and **P4[0] (SCL)**, which are hardwired on the board to the I2C lines of the KitProg. The USB-I2C supports I2C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg USB-I2C functionality, refer to the [KitProg User Guide](#).

3.6 Mass Storage Programmer

The KitProg in PSoC 4 M-Series Pioneer Kit supports programming through a USB Mass Storage interface. This interface allows you to program the PSoC 4200M by copying *.hex* files into an emulated USB Mass Storage device. For more details on KitProg Mass Storage Programmer, refer to the [KitProg User Guide](#).

3.7 Updating the KitProg Firmware

The KitProg firmware normally does not require any update. You can use the PSoC Programmer software to update the KitProg firmware. Refer to the [KitProg User Guide](#) for a detailed explanation on how to update the KitProg firmware.

4. Code Examples



The CY8CKIT-044 PSoC 4 M-Series Pioneer Kit includes seven code examples. This chapter explains how to use these code examples.

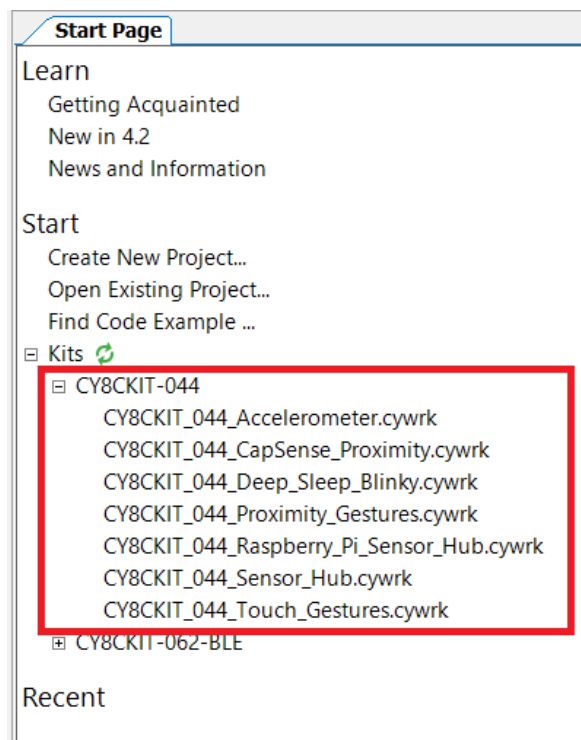
4.1 Using the Kit Code Examples

To access the code examples, download and install the CY8CKIT-044 PSoC 4 M-Series Pioneer Kit setup file from the [kit web page](#). After installation, the code examples are available in the *Firmware* folder in the installation directory.

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-044** under **Start > Kits**. A list of code examples appears, as shown in [Figure 4-1](#).
3. Click on the desired code example, select a location to save the project and click **OK**.

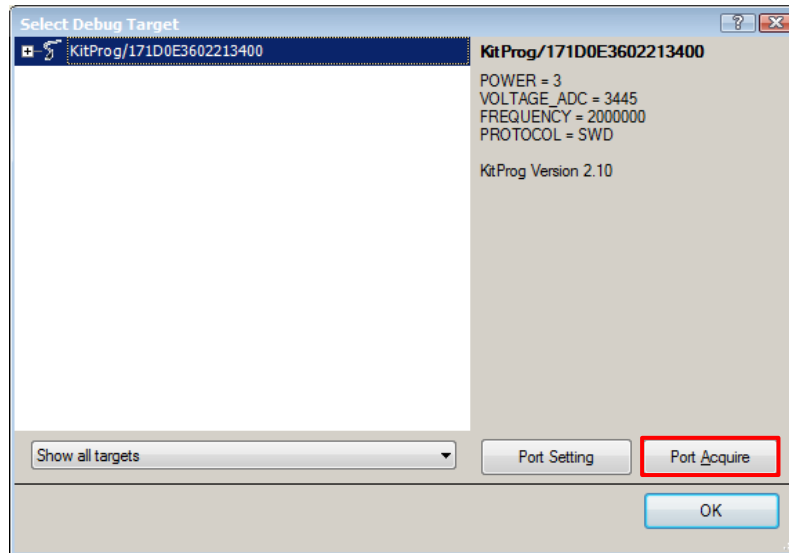
Figure 4-1. Open Code Example from PSoC Creator



4. Build the code example by choosing **Build > Build <Project Name>**. A *.hex* file is generated after the build process.
5. Connect the PSoC 4 M-Series Pioneer Kit to the PC using the USB cable connected to connector J6 as shown in [Figure 3-4](#), to program the kit with this code example.
6. Choose **Debug > Program** in PSoC Creator.

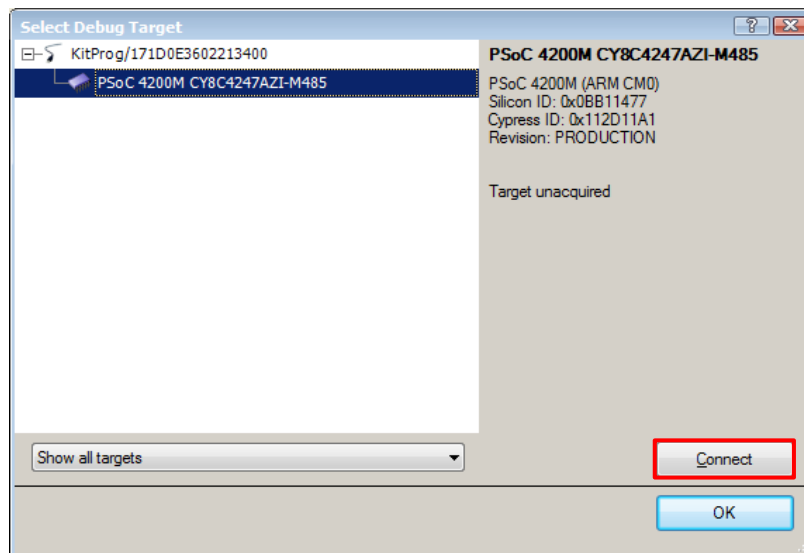
- If the device is already acquired, programming will complete automatically – the result will appear in the PSoC Creator status bar at the bottom left of the screen. If the device is yet to be acquired, the Select Debug Target window will appear. Select **KitProg/<serial_number>** and click the **Port Acquire** button, as shown in [Figure 4-2](#).

Figure 4-2. Port Acquire



- After the device is acquired, it is shown in a tree structure below the **KitProg/<serial_number>**. Click the **Connect** button and then **OK** to exit the window and start programming, as shown in [Figure 4-3](#).

Figure 4-3. Connect Device From PSoC Creator and Program



4.2 Deep-Sleep Blinky

4.2.1 Project Description

This code example demonstrates the Deep-Sleep low-power mode of the PSoC 4200M device. The PSoC 4200M device is configured to be in Deep-Sleep power mode and wakes-up once every second. The PSoC 4200M toggles the state of the GPIO each time the device wakes up from Deep-Sleep power mode. The GPIO is connected to the LED on the kit which turns on and off depending upon the GPIO state. The project also demonstrates the method to calibrate the ILO clock based on the IMO clock signal for better ILO accuracy. The calibrated LFCLK is used to source the Watchdog Timer.

4.2.2 Hardware Connections

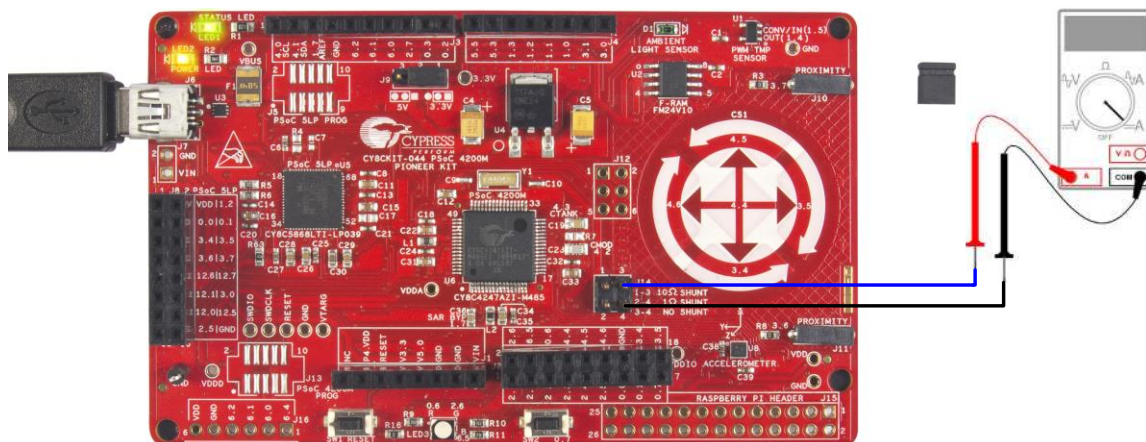
No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.3 Verify Output

To verify the Deep-Sleep Blinky code example, follow these steps.

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with the *CY8CKIT_044_Deep_Sleep_Blinky* code example.
3. After programming successfully, you will see the green LED toggling state every one second.
4. Power off the PSoC 4 M-Series Pioneer Kit by unplugging the USB cable. Remove the jumper shunt from power measurement jumper of PSoC 4200M on the PSoC 4 M-Series Pioneer Kit (J14).
5. Connect an ammeter across the terminals 3 and 4 of the power measurement header, as shown in [Figure 4-4](#).

Figure 4-4. Current Measurement When Powered From USB Port



6. Plug in the USB cable back to power the PSoC 4 M-Series Pioneer Kit.
7. Observe the average power consumed by the PSoC 4200M device on the ammeter. The measured average power must be less than 260 μ A at 3.3V.

Note: Detailed explanation on the power measurement header terminals is provided in the [Current Measurement Jumper](#) section.

4.3 CapSense Proximity

4.3.1 Project Description

This code example demonstrates the proximity sensing capability of the PSoC 4200M device. The CapSense scans for an approaching hand every 100ms and enters a low-power mode if proximity is not detected. This allows the PSoC 4200M to operate at lower average power levels. The scanning interval is increased and the CapSense scan is performed every 30ms if proximity is detected.

The PSoC 4200M gradually increases the brightness of the green LED as the hand approaches the proximity sensor. If proximity is not detected for a period more than 3s, the PSoC 4200M increases the CapSense scanning interval to 100ms.

Note: The time periods of 30ms and 100ms are dependent on the LFCLK accuracy. These time periods can be inaccurate up to 60% when ILO is used as the clock source for LFCLK. Use the ILO Trim Component as demonstrated in the [Deep-Sleep Blinky](#) project to achieve higher accuracy with ILO.

4.3.2 Hardware Connections

Connect the five-inch wire (provided with this kit) to proximity header J10 on the PSoC 4 M-Series Pioneer Kit, as shown in [Figure 4-5](#).

Figure 4-5. Proximity Sensor Connection on PSoC 4 M-Series Pioneer Kit

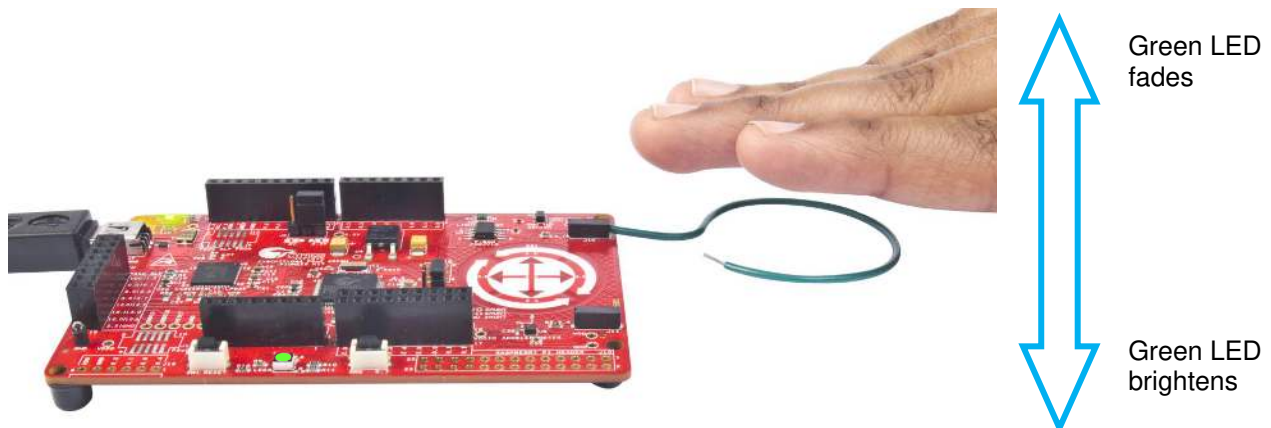


4.3.3 Verify Output

To verify the CapSense Proximity code example, follow these steps:

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with the `CY8CKIT_044_CapSense_Proximity` code example.
3. Plug in the proximity sensor wire to the proximity sensor header as shown in [Figure 4-5](#), and press the reset switch (SW1) of the kit. You should bend the wire to form a loop, as shown in [Figure 4-5](#) to get better proximity range.

Figure 4-6. Bring Hand Close to the Proximity Sensor



4. Bring the hand closer to the proximity sensor, as shown in [Figure 4-6](#). You will see the green LED turn ON.
5. As you move your hand closer to the proximity sensor, the brightness of the LED increases. As you move your hand away from the proximity sensor, the brightness of the LED decreases.

You will get a proximity range of approximately 6cm with the wire supplied with this kit. You can achieve higher proximity range by using a larger wire loop.

Power consumed by PSoC 4200M for this project can be measured using an ammeter as described in the [Deep-Sleep Blinky](#) project.

4.4 Proximity Gestures

4.4.1 Project Description

This code example demonstrates the proximity gesture detection capability of the PSoC 4200M device using CapSense. The PSoC 4 M-Series Pioneer Kit has two headers which are used to connect proximity wires. These wires are used as proximity sensors to detect an approaching hand.

The firmware gangs both proximity sensors and scans them as a single proximity sensor every 200ms by default. If the ganged proximity sensor detects the presence of an approaching hand then PSoC 4200M checks the magnitude of the signal produced by the proximity sensor. If the proximity signal is less than the threshold value, the RGB LED on the kit is maintained at a minimum brightness. If the proximity signal is greater than the threshold value, the brightness of the RGB LED on the kit is varied according to the magnitude of the proximity signal. CapSense scanning interval is changed to 20 ms when the proximity is detected. The project has a proximity range of approximately 5cm.

If proximity is detected, the firmware checks for hand gestures. Two types of gestures are recognized by the firmware – UP gesture and DOWN gesture. [Figure 4-7](#) shows the UP and DOWN gestures. The color of the RGB LED is changed based on the recognized gesture. The RGB LED turns off if proximity is not detected for more than 3 seconds. CapSense scanning interval is changed to 200ms when the proximity is not detected.

Note: The gesture is registered only when the hand moves completely from one end to the other over the proximity sensors.

Note: The time periods of 20ms and 200ms are dependent on the LFCLK accuracy. These time periods can be inaccurate up to 60% when ILO is used as the clock source for LFCLK. Use the ILO Trim Component as demonstrated in the [Deep-Sleep Blinky](#) project to achieve higher accuracy with ILO.

Figure 4-7. Proximity Gesture Recognition with PSoC 4 M-Series Pioneer Kit (Wave DOWN gesture)

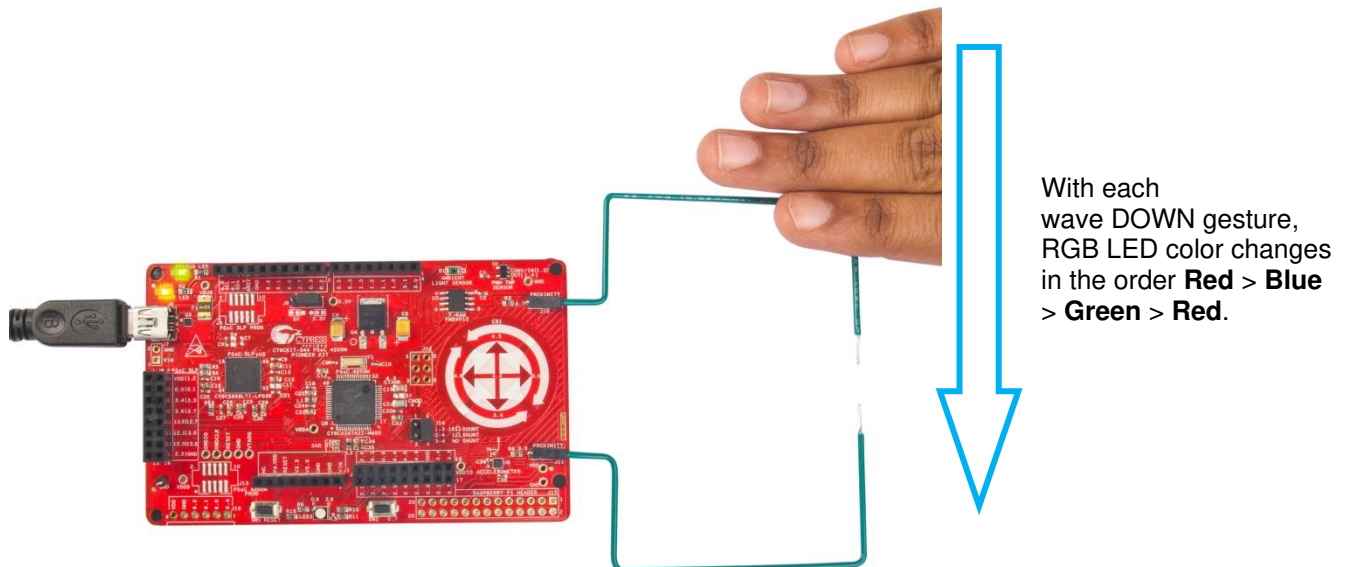


Figure 4-8. Proximity Gesture Recognition with PSoC 4 M-Series Pioneer Kit (Wave UP gesture)

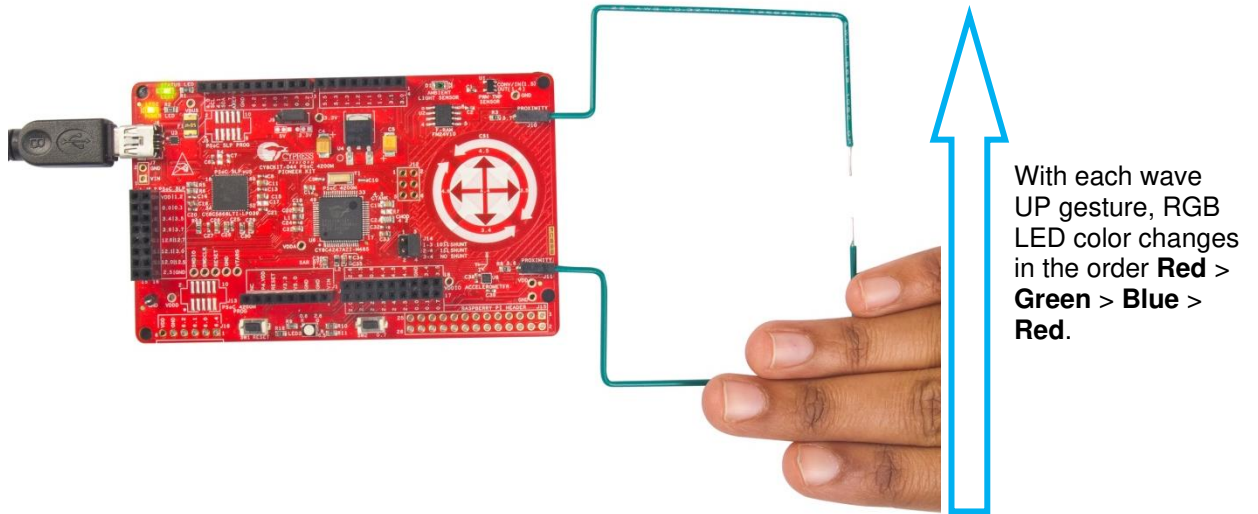


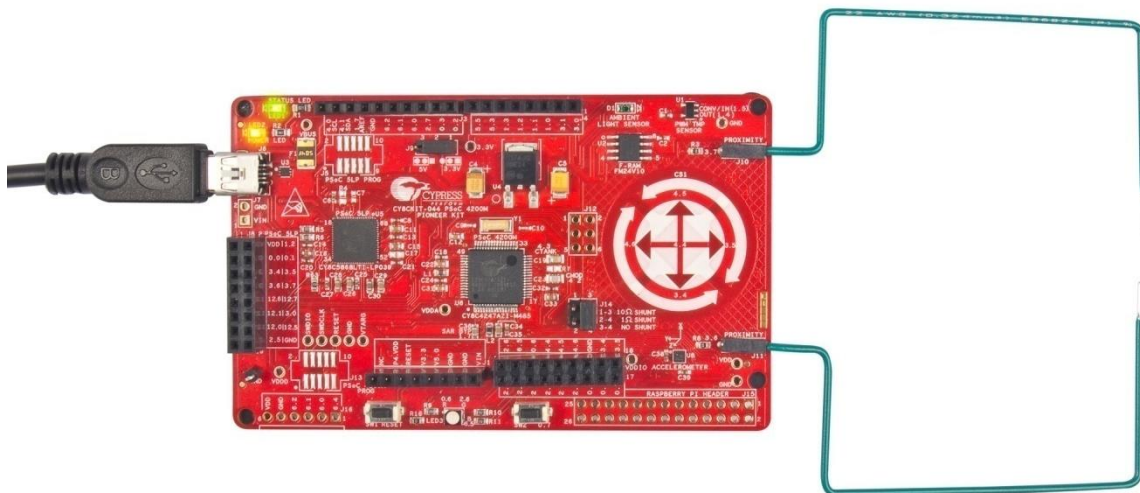
Table 4-1. Gestures and Actions

Gesture	Action (LED Color Change)
UP	Red > Green > Blue > Red
DOWN	Red > Blue > Green > Red

4.4.2 Hardware Connections

Connect five-inch wires (provided with this kit) to proximity headers J10 and J11 on the PSoC 4 M-Series Pioneer Kit, as shown in [Figure 4-9](#).

Figure 4-9. Proximity Sensor Connection on PSoC 4 M-Series Pioneer Kit



4.4.3 Verify Output

To verify the Proximity Gestures code example, follow these steps:

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with the *CY8CKIT_044_Proximity_Gestures* code example.
3. Connect the proximity sensor wires as shown in [Figure 4-9](#), and press the reset switch of the kit (SW1). You should bend the wire to form a loop, as shown in [Figure 4-9](#) to get better proximity range.
4. Bring your hand close to the proximity sensors. The RGB LED will turn ON with white color.
5. The brightness of the LED will increase as you bring your hand closer to the proximity sensors.
6. Wave the hand at a close proximity from the bottom sensor to the top sensor (UP gesture) to see the LED color change in the order Red > Green > Blue > Red, one color at a time with each wave.
7. Wave the hand at a close proximity from the top sensor to the bottom sensor (DOWN gesture) to see the LED color change in the order Red > Blue > Green > Red, one color at a time with each wave.

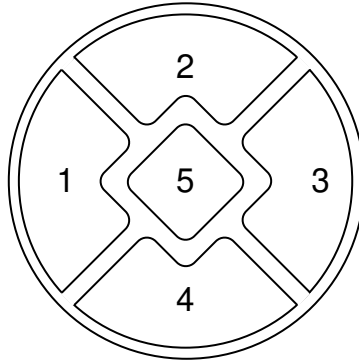
Power consumed by PSoC 4200M for this project can be measured using an ammeter as described in the [Deep-Sleep Blinky](#) project.

4.5 Touch Gestures

4.5.1 Project Description

This code example demonstrates the touch gesture recognition capability of the PSoC 4200M device using the CapSense Gesture Pad. This Gesture Pad has five buttons arranged similar to a joystick. [Figure 4-10](#) shows the CapSense Gesture Pad on the PSoC 4 M-Series Pioneer Kit.

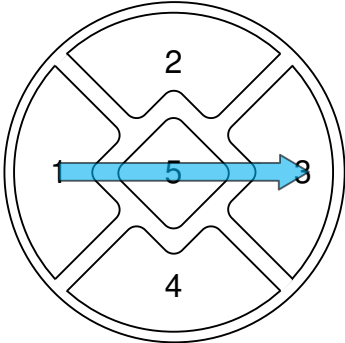
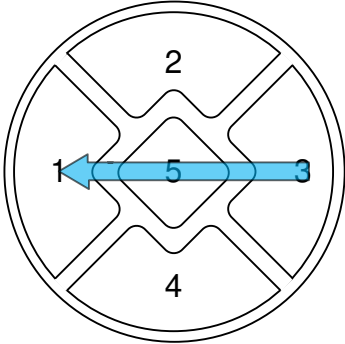
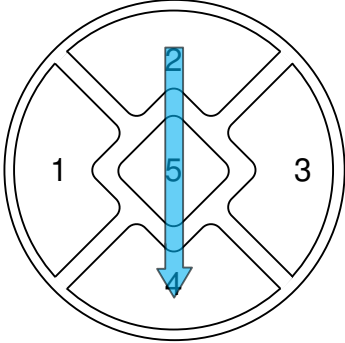
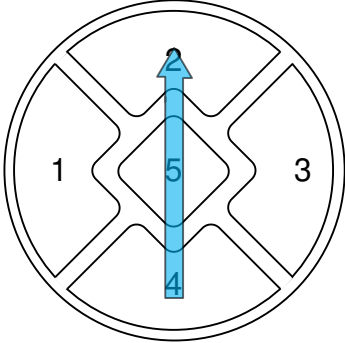
Figure 4-10. CapSense Button Pattern



The project gangs all the five CapSense elements of the Gesture Pad and scans them together as a single proximity sensor every 100ms by default. When the ganged proximity sensor detects an approaching finger, PSoC 4200M turns on the RGB LED and decreases the scanning interval to 10ms. When a finger touch is detected, the firmware tries to recognize if the finger has performed any touch gestures. The PSoC 4200M changes the color of the onboard RGB LED in different patterns depending on the detected gestures. [Table 4-2](#) shows the different gestures recognized by the firmware and the corresponding output action for each gesture.

Table 4-2. CapSense Touch Gestures

Gesture Name	Gesture Example	Action
Swipe Clockwise	A diagram of the gesture pad with a blue arrow starting at button 1 and curving clockwise through buttons 2, 3, 4, and back to 1. The path is highlighted with a light blue shaded area.	Violet > Indigo > Blue > Green > Yellow > Orange > Red > Violet
Swipe Counter-clockwise	A diagram of the gesture pad with a blue arrow starting at button 1 and curving counter-clockwise through buttons 2, 3, 4, and back to 1. The path is highlighted with a light blue shaded area.	Red > Orange > Yellow > Green > Blue > Indigo > Violet > Red

Gesture Name	Gesture Example	Action
Swipe Right		Red > Green > Blue > Red
Swipe Left		Red > Blue > Green > Red
Swipe Down		Decrease Light Intensity
Swipe Up		Increase Light Intensity

If a touch is not detected for more than 3 seconds, the PSoC 4200M turns off the RGB LED. The CapSense scanning interval is changed to 200ms when the proximity is not detected.

Note: The time periods of 10ms and 100ms are dependent on the LFCLK accuracy. These time periods can be inaccurate up to 60% when ILO is used as the clock source for LFCLK. Use the ILO Trim Component as demonstrated in the [Deep-Sleep Blinky](#) project to achieve higher accuracy with ILO.

Note: Clockwise and counter-clockwise gestures can be initiated from any sensor except sensor 5.

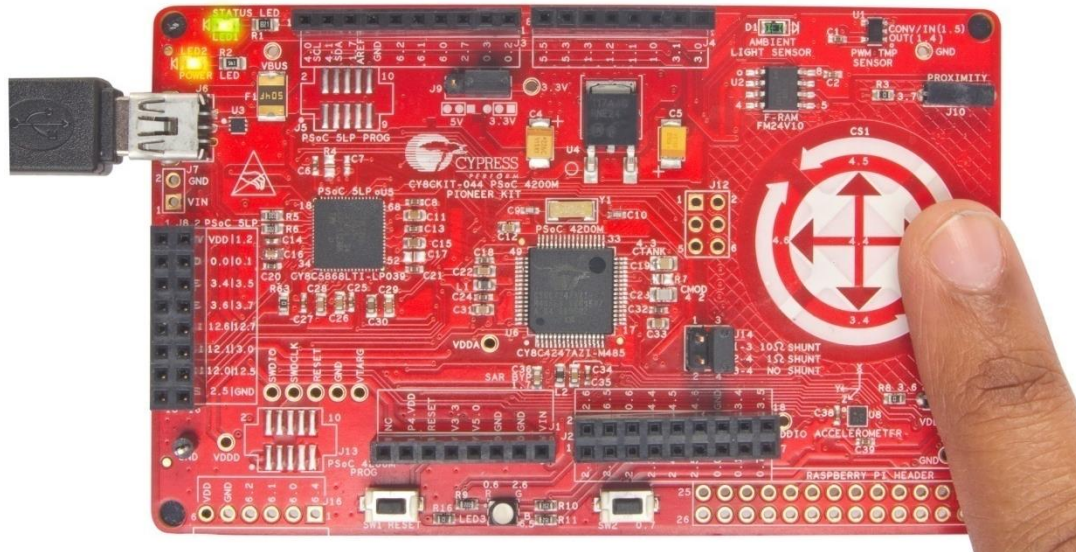
4.5.2 Hardware Connections

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

4.5.3 Verify Output

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with *CY8CKIT_044_Touch_Gestures* project.
3. Bring your finger closer to the Gesture Pad to see the RGB LED turn ON with white color.
4. Move your finger on the Gesture Pad on PSoC 4 M-Series Pioneer Kit as mentioned in [Table 4-2](#) to observe the corresponding LED color and intensity changes. The RGB LED changes the color once after the completion of a gesture. For example, if you complete a clockwise gesture, the RGB LED changes the color to Indigo if the previous color was Violet.

Figure 4-11. Verify CapSense Touch Gestures



5. Clockwise and counter-clockwise gestures are used to make fine color changes. Swipe-left and swipe-right gestures are used to make coarse color changes. The swipe up and swipe down gestures are used to change the brightness of the LED. See [Table 4-2](#) for details on how the LED behaves for each gesture.

Note: It requires one full circular swipe (clockwise or counter-clockwise) on the Gesture Pad to see the change in LED color. However, if the swipe is stopped halfway through the circle and the finger is removed from the board, no gesture is detected by the PSoC 4200M.

Note: You might get erroneous results if you pick up the PSoC 4 M-Series Pioneer Kit in your hand and try to validate the output. This is because the PSoC 4200M pins connected to the CapSense Gesture Pad are also connected to the header J2 and can cause false touches if these pins are touched. You must touch all the required sensors in the right order as provided in [Table 4-2](#) for the firmware to detect gestures correctly.

Note: If you hold the finger in close proximity to the Gesture Pad (without touching the sensors) you will observe that the RGB LED blinks every 3 seconds. This is because, the PSoC 4200M firmware switches to the 100ms scan interval mode if touch is not detected for more than 3 seconds and turns off the LED. Since the finger is in the range of proximity sensor after this event, PSoC 4200M turns the LED back on and switches to the 10ms scan interval mode.

Power consumed by PSoC 4200M for this project can be measured using an ammeter as described in the [Deep-Sleep Blinky](#) project.

4.6 Accelerometer

4.6.1 Project Description

This code example shows a method to interface the onboard digital accelerometer with the PSoC 4200M device. The accelerometer is assigned with the I2C address 0x0F.

The firmware configures the accelerometer to generate an interrupt if motion is detected. PSoC 4200M reads the accelerometer data whenever the accelerometer generates an interrupt. This information is used to change the color and intensity of the onboard RGB LED depending on the direction of motion of the kit. [Table 4-3](#) shows the RGB LED color corresponding to the direction of motion. The intensity of the RGB LED is directly proportional to the magnitude of acceleration experienced by the sensor. [Figure 4-12](#) shows the accelerometer orientation with respect to the kit.

Figure 4-12. Accelerometer Movement in X-Axis

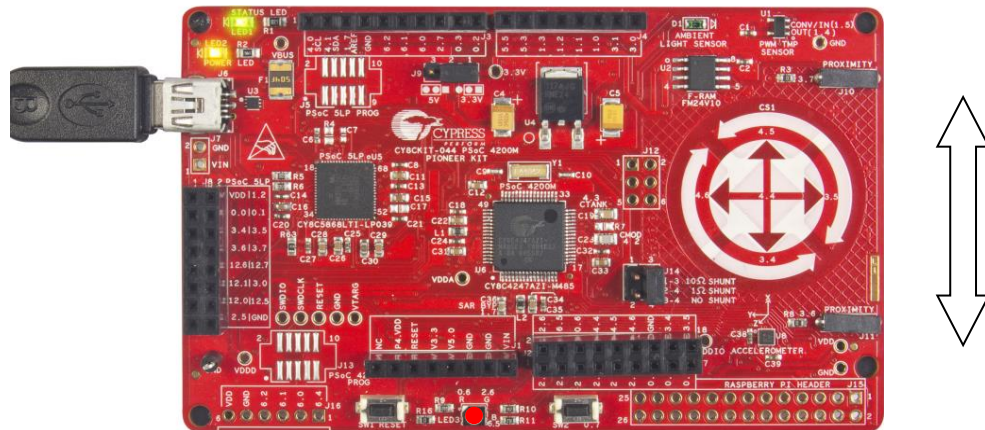


Figure 4-13 Accelerometer Movement in Y-Axis

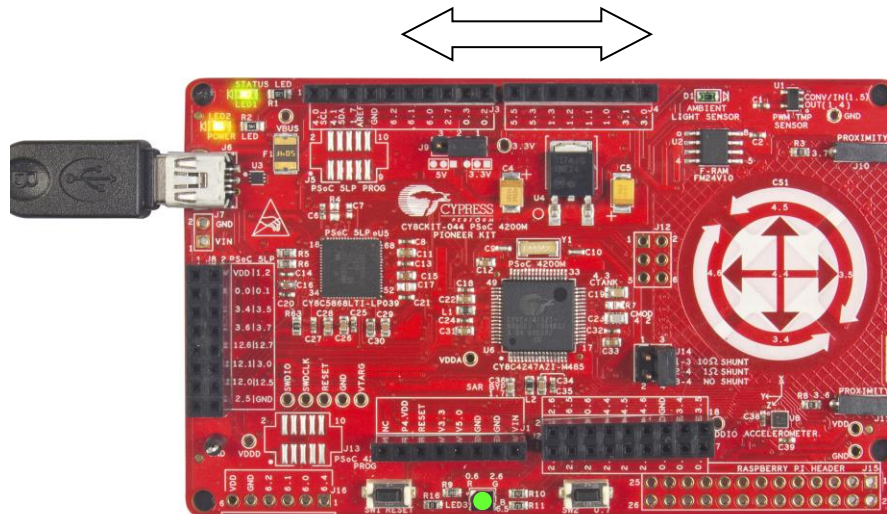


Figure 4-14 Accelerometer Movement in Z-axis

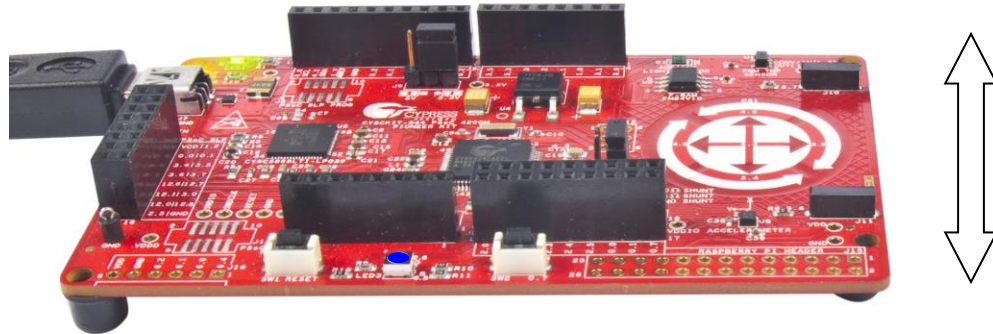


Table 4-3. LED Color Corresponding to Direction of Motion

Direction of Motion	RGB LED Color
x-axis	Red
y-axis	Green
z-axis	Blue

Note: If the kit is moved in multiple axes, the PSoC 4200M modulates the intensity of RGB LED color corresponding to each axis depending on the acceleration detected on that axis. For example, if the kit is moved in both x- and y- axis, the RGB LED shows a combination of red and green colors.

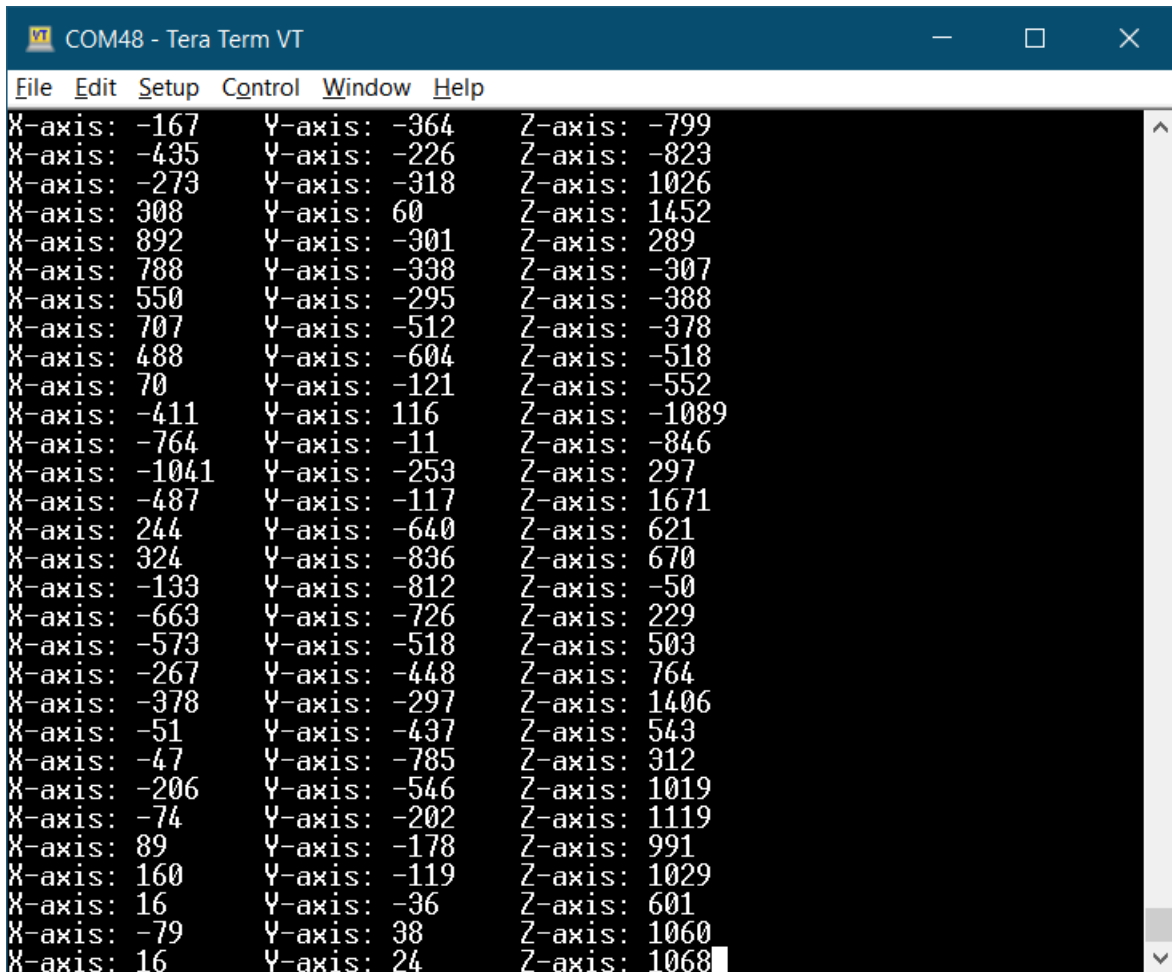
4.6.2 Hardware Connections

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

4.6.3 Verify Output

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with *CY8CKIT_044_Accelerometer* project.
3. Move the PSoC 4 M-Series Pioneer Kit in different directions to see the RGB LED glow as described in [Table 4-3](#).
4. Open the serial terminal software on your PC, such as Tera Term or PuTTY, and connect to the KitProg USBUART. Refer to the *KitProg User Guide* to learn how to use the KitProg USB-UART Bridge with COM terminal software.
5. Set the baud rate for communication as "115200". Set the number of Data bits as '8', Parity option to 'None', number of Stop bits as '1', and Flow Control option to 'None'. Observe that the accelerometer data displayed on the serial terminal, as shown in [Figure 4-15](#).

Figure 4-15. Accelerometer Data on Serial Terminal



```

COM48 - Tera Term VT
File Edit Setup Control Window Help
X-axis: -167   Y-axis: -364   Z-axis: -799
X-axis: -435   Y-axis: -226   Z-axis: -823
X-axis: -273   Y-axis: -318   Z-axis: 1026
X-axis: 308    Y-axis: 60     Z-axis: 1452
X-axis: 892    Y-axis: -301   Z-axis: 289
X-axis: 788    Y-axis: -338   Z-axis: -307
X-axis: 550    Y-axis: -295   Z-axis: -388
X-axis: 707    Y-axis: -512   Z-axis: -378
X-axis: 488    Y-axis: -604   Z-axis: -518
X-axis: 70     Y-axis: -121   Z-axis: -552
X-axis: -411   Y-axis: 116    Z-axis: -1089
X-axis: -764   Y-axis: -11    Z-axis: -846
X-axis: -1041  Y-axis: -253   Z-axis: 297
X-axis: -487   Y-axis: -117   Z-axis: 1671
X-axis: 244    Y-axis: -640   Z-axis: 621
X-axis: 324    Y-axis: -836   Z-axis: 670
X-axis: -133   Y-axis: -812   Z-axis: -50
X-axis: -663   Y-axis: -726   Z-axis: 229
X-axis: -573   Y-axis: -518   Z-axis: 503
X-axis: -267   Y-axis: -448   Z-axis: 764
X-axis: -378   Y-axis: -297   Z-axis: 1406
X-axis: -51    Y-axis: -437   Z-axis: 543
X-axis: -47    Y-axis: -785   Z-axis: 312
X-axis: -206   Y-axis: -546   Z-axis: 1019
X-axis: -74    Y-axis: -202   Z-axis: 1119
X-axis: 89     Y-axis: -178   Z-axis: 991
X-axis: 160    Y-axis: -119   Z-axis: 1029
X-axis: 16     Y-axis: -36    Z-axis: 601
X-axis: -79    Y-axis: 38     Z-axis: 1060
X-axis: 16     Y-axis: 24     Z-axis: 1068
  
```

4.7 Sensor Hub

4.7.1 Project Description

This code example demonstrates the capability of the PSoC 4200M device to function as a sensor hub device. The PSoC 4200M can interface with multiple digital and analog sensors. In this code example, the PSoC 4200M interfaces an I2C-based accelerometer, a PWM-based temperature sensor, and an ambient light sensor. The firmware measures the ambient temperature, ambient light, and the accelerometer values from the sensors and implements Real-Time Clock (RTC) functionality.

The firmware also logs the temperature and acceleration data along with the RTC timestamp information onto the onboard F-RAM device every one second. Data is stored in the format [timestamp hours, timestamp minutes, timestamp seconds, ambient light sensor data, temperature data integer part, temperature data fractional part, x-axis acceleration MSB, x-axis acceleration LSB, y-axis acceleration MSB, y-axis acceleration LSB, z-axis acceleration MSB, z-axis acceleration LSB]. If the F-RAM is full, the firmware rewrites the data from the beginning of the F-RAM.

The firmware also reads back the data logged to the F-RAM every one second and sends it through a UART. This data can be read using the KitProg USB-UART bridge. Refer to the *KitProg User Guide* to learn how to use the KitProg USB-UART bridge.

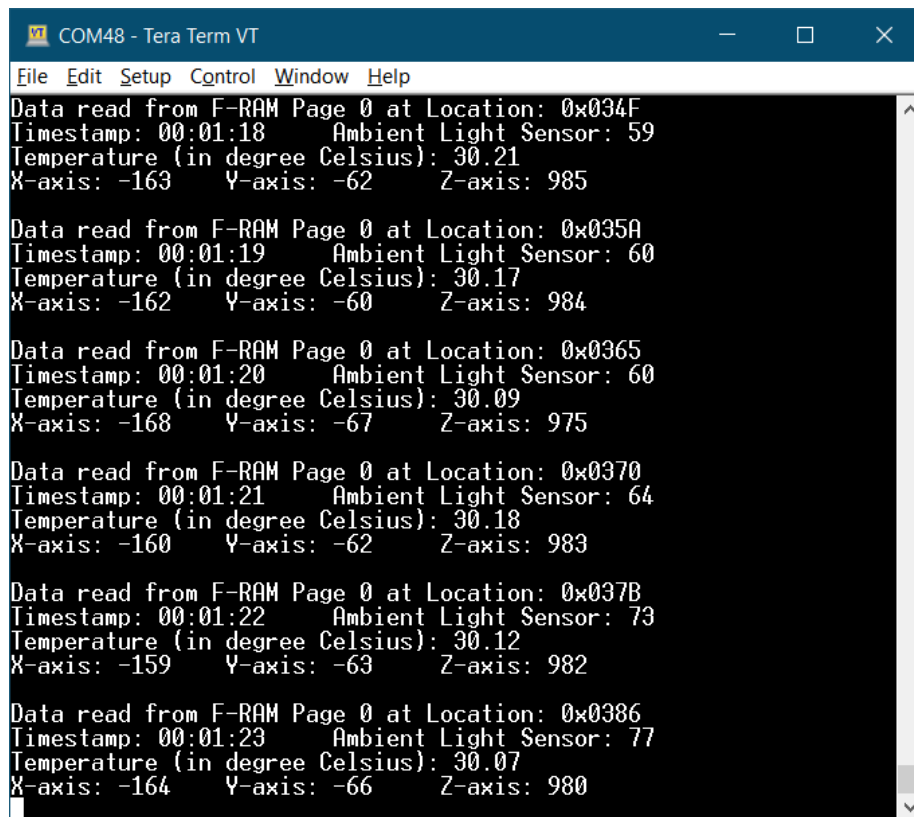
4.7.2 Hardware Connections

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

4.7.3 Verify Output

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with the *CY8CKIT_044_Sensor_Hub* project.
3. Open the serial terminal software on your PC, such as Tera Term or PuTTY, and connect to the KitProg USB-UART. Refer to the *KitProg User Guide* to learn how to use the KitProg USB-UART Bridge with COM terminal software.
4. Set the baud rate for communication as “115200”. Set the number of Data bits as ‘8’, Parity option to ‘None’, number of Stop bits as ‘1’, and Flow Control option to ‘None’. Observe that the sensor data written to the F-RAM is read back and displayed in the serial terminal, as shown in [Figure 4-16](#).

Figure 4-16. F-RAM Data Read by PSoC 4200M



5. Vary the light intensity falling on the ambient light sensor to observe the change in ambient light intensity shown on the serial terminal.
6. Move the kit in different directions to observe the accelerometer data change on the serial terminal.
7. You can read the data logged by the PSoC 4200M to the F-RAM using the Bridge Control Panel (BCP) software and KitProg. Refer to Chapter 5 of the *KitProg User Guide* to learn how to use KitProg USB-I2C bridge with BCP. Type in the following command in BCP to access the F-RAM data from address location 0x0000 of the F-RAM:

```
w 50 00 00 r 50 x x x x x x x x x x x x p
```

You can read from any address location of F-RAM by modifying the above command. For example, to read 13 bytes from address location 0x1234 of F-RAM, use the following command:

```
w 50 12 34 r 50 x x x x x x x x x x x x p
```

Refer to the [Appendix A.3](#) to understand how the F-RAM communicates with an I2C master.

The ambient light sensor and accelerometer data corresponding to each axis are 2 bytes long. Since the complete data packet is 13 bytes long, the last few locations in each F-RAM page are not written. The F-RAM pages are switched if a complete data packet cannot fit in to the same F-RAM page.

Note: This project is not configured to operate in an I2C multi-master environment. The firmware may fail to log the data in F-RAM if you use Bridge Control Panel tool to read the F-RAM data.

4.8 Raspberry Pi

4.8.1 Project Description

This code example is similar to the [Sensor Hub](#) code example, except that the UART data is sent to the Raspberry Pi interface instead of the USB-UART Bridge interface. This example demonstrates the capability of the PSoC 4200M device to function as a sensor hub and output the values to a Raspberry Pi. In this code example, the PSoC 4200M interfaces an I2C-based accelerometer, a PWM-based temperature sensor, an I2C-based F-RAM, and an ambient light sensor.

The firmware writes the timestamp information from the RTC, ambient temperature from the temperature sensor, and the accelerometer values to the F-RAM. These values are read back from the F-RAM and then transmitted through the UART to the Raspberry Pi.

[Minicom](#) is a text-based terminal emulation program running on the Raspberry Pi. This code example uses Minicom to view the UART output on the Raspberry Pi.

4.8.2 Hardware Connections

You must populate the Raspberry Pi compatible header (J15) on the PSoC 4 M-Series Pioneer Kit and connect a Raspberry Pi to the header, as shown in [Figure A-12](#). Refer to the following section for instructions on how to setup the Raspberry Pi for viewing the output.

4.8.3 Setting Up Raspberry Pi

1. Setup the Raspberry Pi hardware, as explained in this [tutorial](#).
2. Login to the Raspberry Pi using the default User ID: pi and Password: raspberry (unless you have explicitly changed it while setting up the Raspberry Pi).
3. Type “startx” at the command prompt to enter the graphical user interface.
4. Double-click on LXTerminal on the desktop to open the terminal program.
5. Install Minicom to view the serial data coming from CY8CKIT-044 by typing “sudo apt-get install minicom”.
6. Wait until the minicom gets installed.

4.8.4 Verify Output

1. Power the PSoC 4 M-Series Pioneer Kit through USB connector J6.
2. Program the PSoC 4 M-Series Pioneer Kit with *CY8CKIT_044_Raspberry_Pi* project.
3. Connect a Raspberry Pi and login as specified in step 2 in [Setting Up Raspberry Pi](#).
4. Launch Minicom by typing “sudo minicom -b 115200 -o -D /dev/ttyAMA0” in LXTerminal.
5. The Minicom terminal opens and the UART data is displayed similar to [Figure 4-16](#).
6. To stop Minicom, press **CTRL-A** and then press **Z**. The Minicom help screen is displayed.
7. Press **Q** to quit the Minicom.
8. Minicom pops-up a window to confirm quitting without reset. Press **Enter** to confirm and quit.

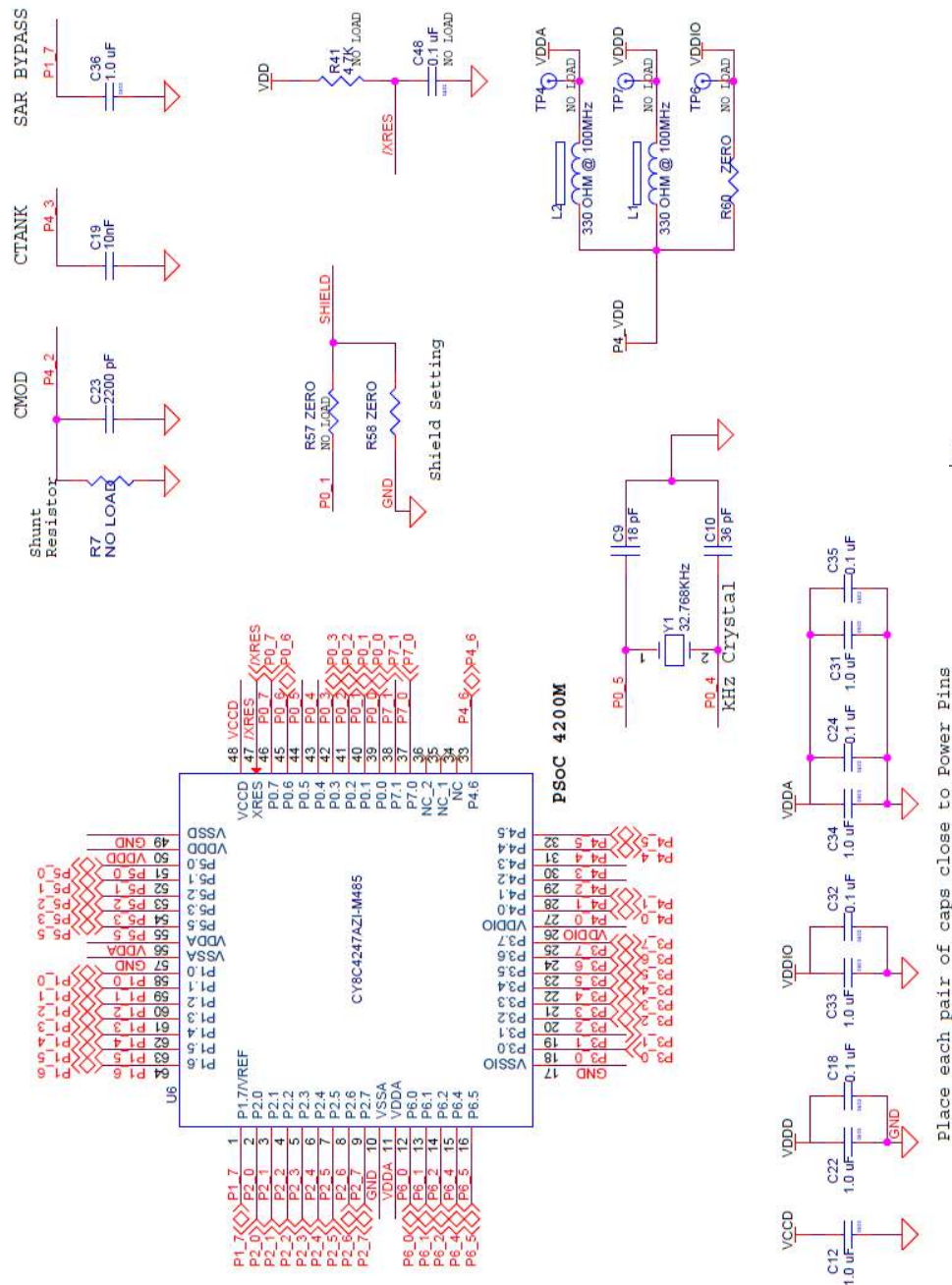
Note: If you did not follow the steps in [Setting Up Raspberry Pi](#) to setup the Raspberry Pi, refer to the online tutorials for details on accessing UART, since the above settings/options may change depending on the installation.

A. Appendix



A.1 Schematics

Figure A-1. PSoC 4200M Schematic



Place each pair of caps close to Power Pins

Figure A-2. KitProg (PSoC 5LP) and Programming Interface Schematic

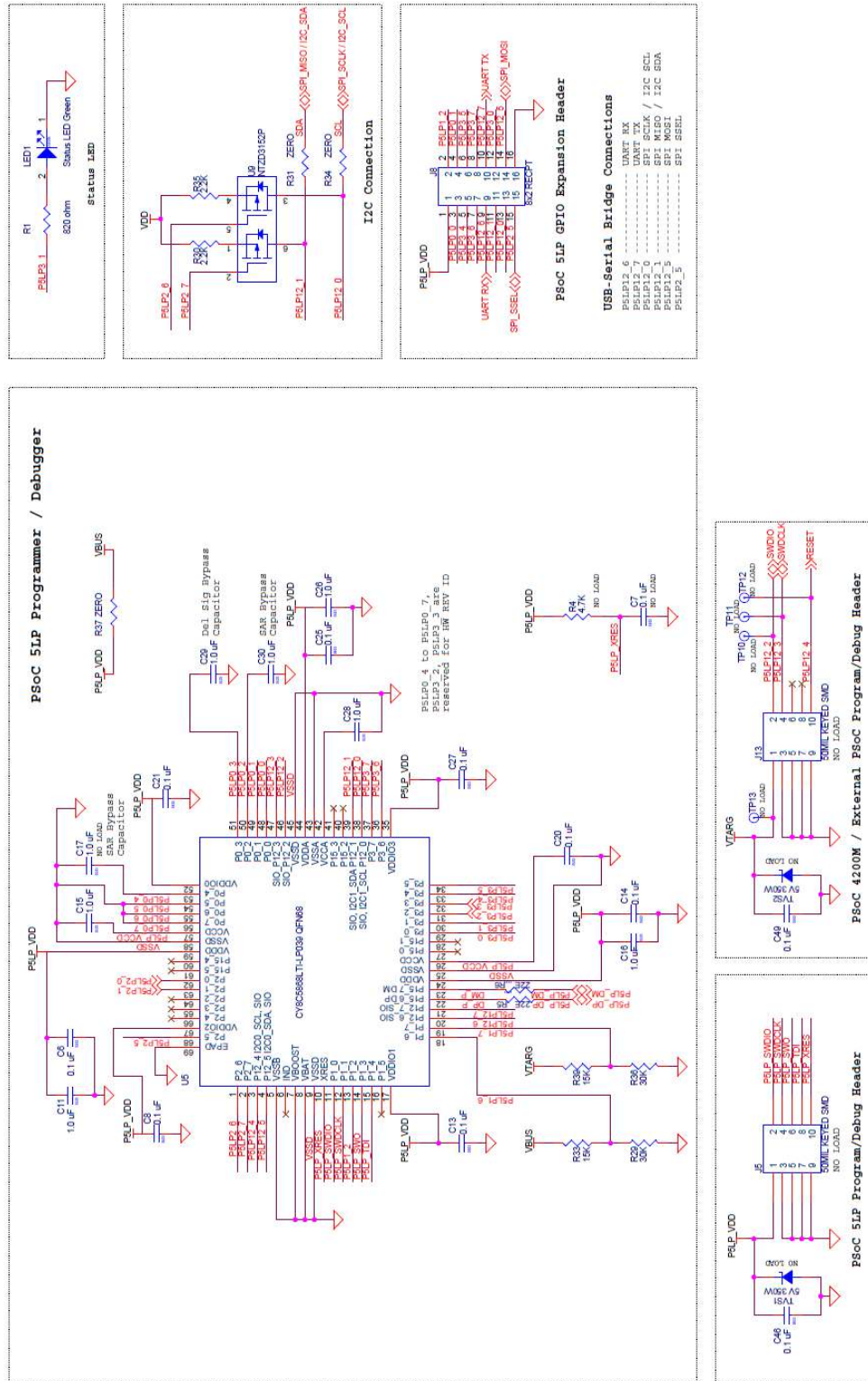
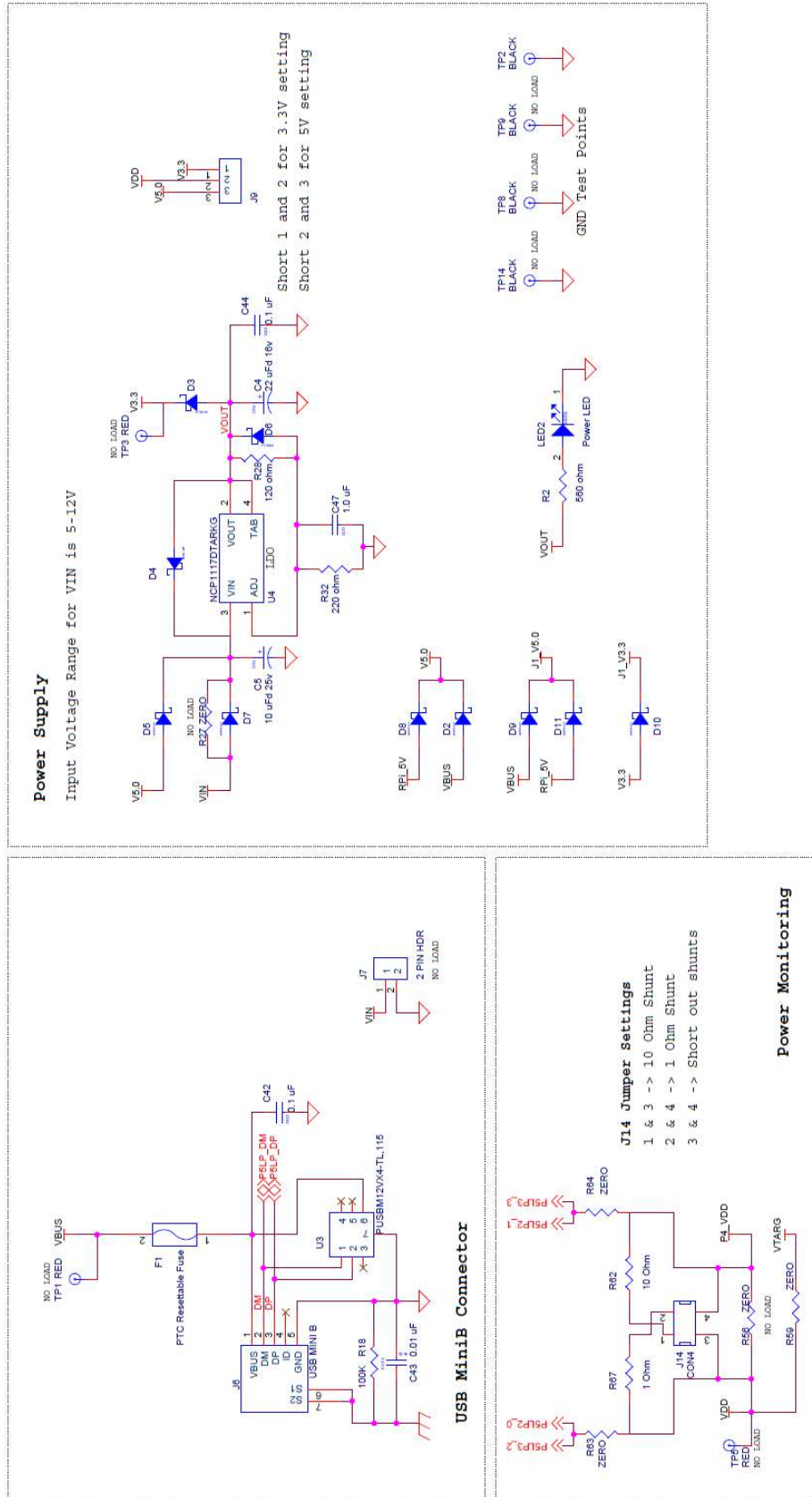


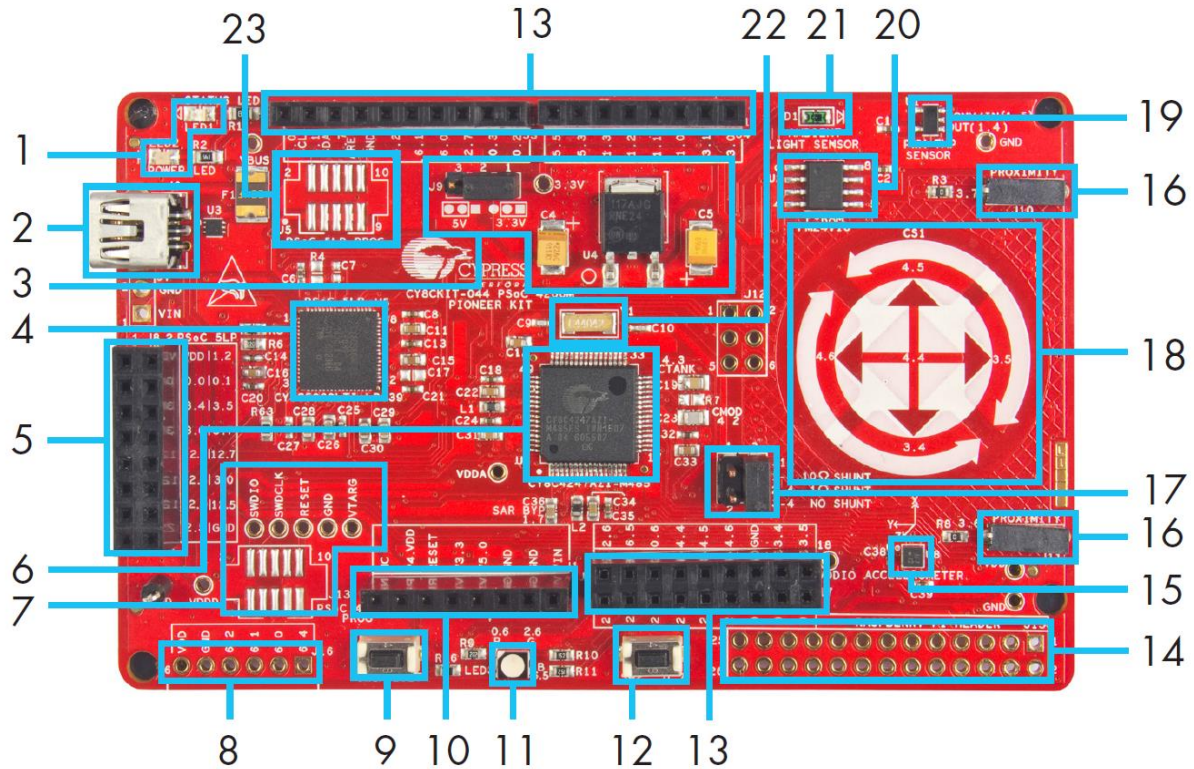
Figure A-3. Power Supply and Power Monitoring Circuit Schematic



A.2 Hardware Functional Description

This section provides detailed explanation on individual hardware blocks of the PSoC 4 M-Series Pioneer Kit. Complete schematic of the PSoC 4 M-Series Pioneer Kit can be accessed from the PSoC 4 M-Series Pioneer Kit web page or the kit install directory.

Figure A-5. Board Highlight of Hardware Components



A.2.1 PSoC 4200M

This kit uses the PSoC 4200M (CY8C4247AZI-M485) family device (marked **6** in [Figure A-5](#)). PSoC 4 is a scalable and reconfigurable platform architecture for a family of programmable embedded system controllers with an Arm Cortex-M0 CPU. It combines programmable and reconfigurable analog and digital blocks with flexible automatic routing. The PSoC 4200M device family, based on this platform architecture, is a combination of a microcontroller with digital programmable logic, programmable analog, programmable interconnect, high-performance analog-to-digital conversion, opamps with comparator mode, and standard communication and timing peripherals. The PSoC 4200M products will be fully compatible with members of the PSoC 4 platform for new applications and design needs. The programmable analog and digital subsystems allow flexibility and in-field tuning of the design. For more information, refer to the [PSoC 4200M web page](#) and the [PSoC 4200M Family Datasheet](#).

A.2.2 PSoC 5LP

An onboard PSoC 5LP (CY8C5868LTI-LP039) is used to program and debug the PSoC 4200M device (marked **4** in [Figure A-5](#)). The PSoC 5LP connects to the USB port of the PC through a USB Mini-B connector and to the SWD interface of the PSoC 4200M device. PSoC 5LP is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C58LP family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals.

For more information, visit the [PSoC 5LP web page](#). Also, refer to the [CY8C58LP Family Datasheet](#).

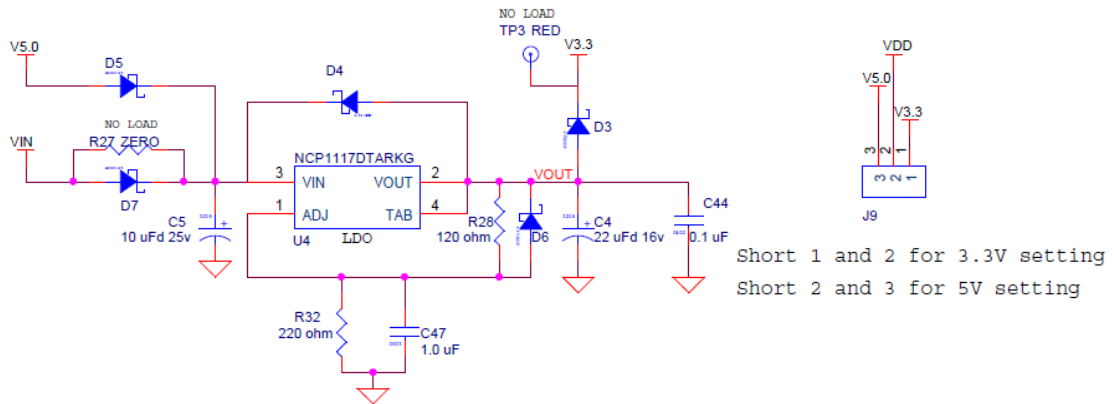
A.2.3 Power System

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

- 5-V power from onboard USB connector
- 5-V to 12-V VIN power from Arduino Shield
- 5-V power from the Raspberry Pi

The PSoC 4 M-Series Pioneer Kit is powered at 5 V directly from the USB power supply. An LDO is used to output 3.3 V to power the PSoC 4200M device (marked **3** in Figure A-5). These voltages are selected through the J9 jumper, as shown in Figure A-6.

Figure A-6. Schematics of LDO and Power Selection Jumper

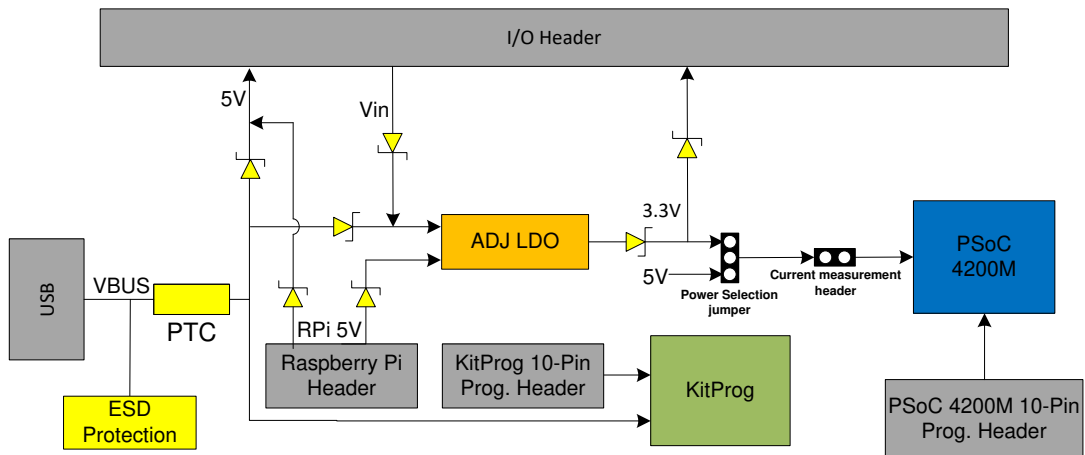


The input to the LDO can be from either the USB or the VIN pin in the Arduino header.

Protection Circuits

The power supply rail has reverse-voltage, overvoltage, short circuit, and excess current protection features, as shown in Figure A-7.

Figure A-7. Power Supply Block Diagram with Protection Circuits

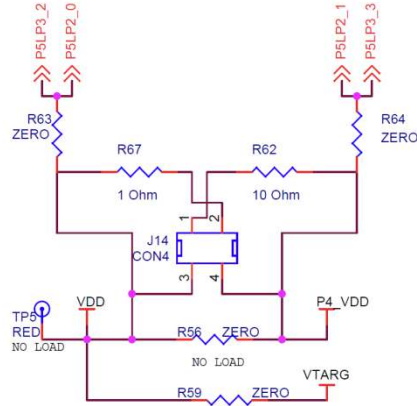


- A PTC resettable fuse is connected to protect the computer's USB Mini-B connectors from shorts and over current.
- ORing diodes prevent damage to components when the board is powered from different voltage sources at the same time.
- ESD protection is provided for the USB Mini-B connector.

Current Measurement Jumper

To measure the PSoC 4200M power consumption, a header is populated in series with the power supply to the PSoC 4200M, which can be used to measure current using an ammeter without the need to de-solder any components from the board (marked 17 in Figure A-5).

Figure A-8. Schematics of Current Measurement Jumper



J14 Jumper Settings

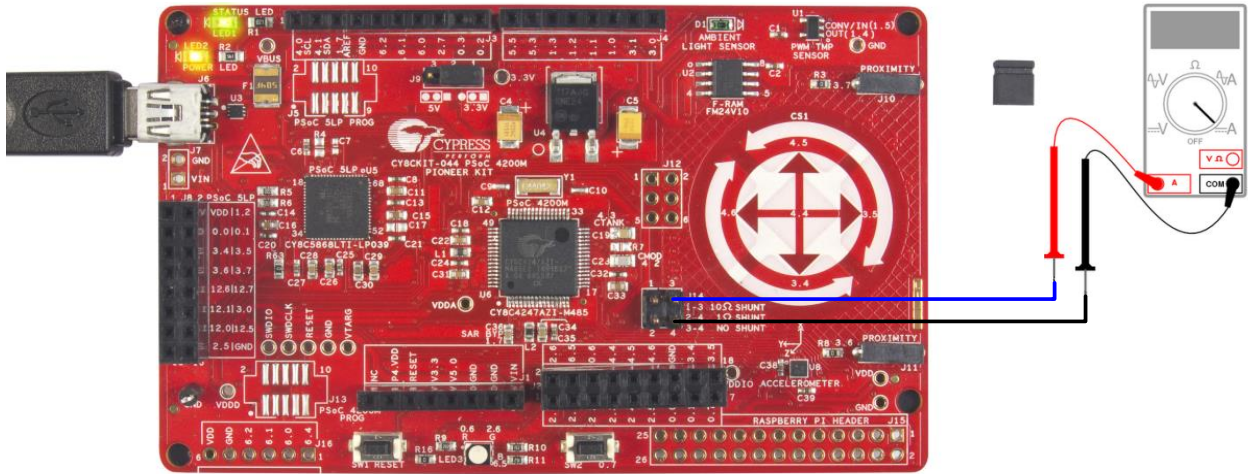
- 1 & 3 -> 10 Ohm Shunt
- 2 & 4 -> 1 Ohm Shunt
- 3 & 4 -> Short out shunts

Power Monitoring

The following two methods are supported for measuring the current consumption of the PSoC 4200M device.

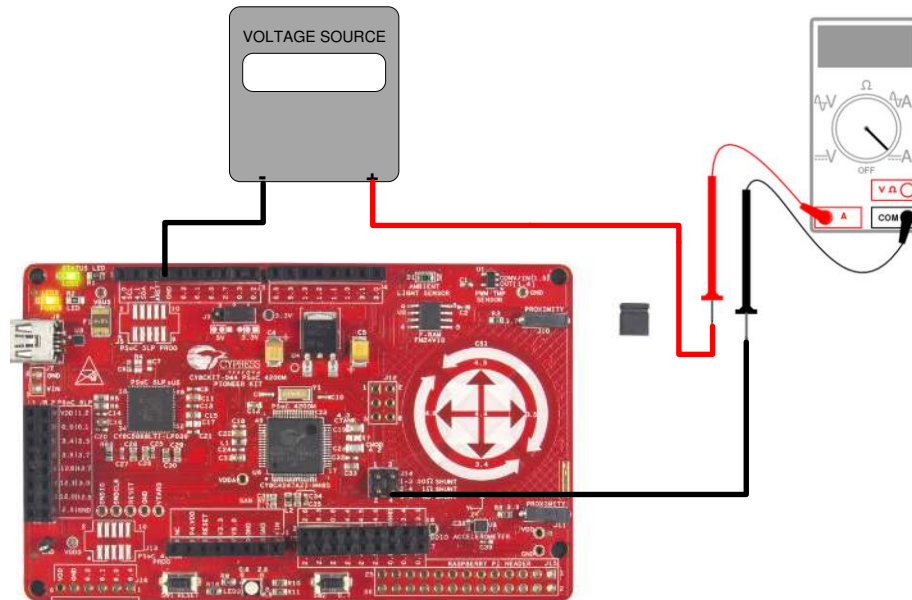
- When the board is powered through the USB Mini-B connector (J6), remove jumper J14 and connect an ammeter between terminals 3 and 4, as shown in Figure A-9.

Figure A-9. Current Measurement When Powered From USB Connector



- When the PSoC 4 M-Series Pioneer Kit is powered from an external voltage supply, remove the USB cable from J6. Connect the positive terminal of the external voltage supply to the positive terminal of the ammeter and the negative terminal of the ammeter to pin 4 of J14. Connect the negative terminal of the external voltage supply to a GND pin on the board. Figure A-10 shows the required connections.

Figure A-10. Current Measurement When Powered Separately



Note: When the PSoC 4200M is powered separately and not through the USB, remove the resistors R63 and R64 to avoid leakage while measuring current.

A.2.4 Expansion Connectors

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

This kit has five Arduino compatible headers—J1, J2, J3, J4, and J12. Header J1 is marked **10** and header J2, J3, and J4 are marked **13** in [Figure A-5](#). You can develop applications based on the Arduino Shield's hardware.

The J1 header contains I/O pins for reset, I/O reference voltage (IOREF), and power supply line. The J2 header is an analog port that contains I/O pins for SAR ADC, comparator, and opamp. The J3 header also has I/O pins that can connect to the PSoC 4200M CAN. The J3 header is primarily a digital port that contains I/O pins for PWM, I2C, SPI, and analog reference. The J4 header is also a digital port that contains I/O pins for UART and PWM. The J12 header is an Arduino ICSP compatible header for the SPI interface and is not populated. Refer to the “No Load Components” section of the [Bill of Materials](#) for the header part number.

Note: Pin P1[7] has a 1.0- μ F capacitor connected to it. This is provided as a bypass capacitor for the PSoC 4200M SAR ADC. Remember to remove this capacitor if you are planning to use this pin for any other purposes.

Additional Functionality of Header J2

The J2 header is a 9 \times 2 header that supports Arduino Shields. Some pins from port 0, port 2, port 3, port 4, and port 6 of PSoC 4200M are brought to this header. The port 3 and port 4 pins also connect to the onboard CapSense Gesture Pad through 560-ohm resistors. When the CapSense feature is not used, you may remove these resistors if necessary to help ensure better performance with these pins.

Functionality of Unpopulated Header J12

The J12 header is a 2 \times 3 header that supports Arduino Shields. This header is used on a small subset of Shields and is unpopulated on the PSoC 4 M-Series Pioneer Kit.

Pmod Connector – Digilent Pmod Compatible (J16—unpopulated)

This port supports Digilent Pmod Peripheral Modules (marked **8** in [Figure A-5](#)). Pmods are small I/O interfaces that connect with embedded control boards through either 6 or 12-pin connectors. The PSoC 4 M-Series Pioneer Kit supports the 6-pin Pmod type 2 (SPI) interface. For Digilent Pmod Peripheral Modules, go to www.digilentinc.com.

This header is not populated on the PSoC 4 M-Series Pioneer board. You must populate this header before connecting Pmod Peripheral Modules. Refer to the “No Load Components” section of the [Bill of Materials](#) for the header part number.

PSoC 5LP GPIO Header (J8)

An 8×2 header is 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 (marked **5** in [Figure A-5](#)). This header also contains the USB-UART bridge pins and USB-I2C bridge pins that can be used when these pins are not accessible on the Arduino headers because a Shield is connected.

A.2.5 USB Mini-B Connector

The PSoC 5LP connects to the USB port of a PC through a USB Mini-B connector (marked **2** in [Figure A-5](#)), which can also be used to power the PSoC 4 M-Series Pioneer Kit. A resettable polyfuse is used to protect the computer's USB port from shorts and over current. If more than 500mA is drawn from the USB Mini-B connector, the fuse will automatically break the connection until the short or overload is removed.

A.2.6 CapSense Circuit

CapSense Gesture Pad

The kit has a five element capacitive Gesture Pad on the board, which is connected to the PSoC 4200M module pins (marked **18** in [Figure A-5](#)). The CMOD and CTANK capacitors essential for CapSense functionality are connected on the board. A 2.2-nF capacitor is present on the CMOD pin, P4[2], for CapSense operation. This kit also supports CapSense designs that enable waterproofing. On this kit, the Shield is connected to the pin or to ground by resistors R57 and R58, respectively. By default, R58 is mounted on the board, which connects the Shield to ground. Remove R58 and populate R57 when evaluating waterproofing designs, which will connect the Shield to the designated pin, P0[1].

Proximity Header

The PSoC 4 M-Series Pioneer board contains two headers (J10 and J11) for CapSense Proximity sensor connection (marked **16** in [Figure A-5](#)). If the proximity functionality is not being used, these can be used as standard GPIO pins.

A.2.7 Pioneer Board LEDs

The Pioneer board has three LEDs. A green LED (LED1) indicates the status of the KitProg. See [A.5 KitProg Status LED States](#) for more information on LED states. An amber LED (LED2) indicates the status of power supplied to the board. The kit also has a general-purpose RGB LED (LED3) for user applications (marked **1** and **11** in [Figure A-5](#)).

A.2.8 Push Buttons

The board contains a reset push button and a user push button (marked **9** and **12** in [Figure A-5](#)). The reset push button is connected to the XRES pin of PSoC 4200M and is used to reset the device. The user push button is connected to P0[7] of the PSoC 4200M device. Both the push buttons connect to ground on activation (active low).

A.2.9 Cypress Ferroelectric RAM (F-RAM)

The PSoC 4 M-Series Pioneer Kit contains an F-RAM device ([FM24V10-G](#)) (marked **20** in [Figure A-5](#)) that can be accessed through I2C lines P4[0] and P4[1] of the PSoC 4200M device. The F-RAM has a capacity of 1-Mbit (128 KB) with an I2C speed up to 1 Mbps. The I2C slave address of the F-RAM device is 7 bits wide, and the two least significant bits are configurable through physical pins. These pins are hardwired to 00 on the board. By default, the address of the F-RAM device used on the kit is 0x50. This address can be modified by changing the R21/R20 and R23/R22 resistor pairs. The operating voltage range of the F-RAM is between 2 V and 3.6 V. To prevent the F-RAM from operating at voltages greater than 3.6V, the power supply to the F-RAM is derived from the output of the 3.3-V regulator. The I2C lines are connected to the 3.3 V side of the onboard level translator to help the F-RAM communicate with PSoC 4200M operating at 5 V.

A.2.10 3-Axis Accelerometer

The PSoC 4 M-Series Pioneer Kit has a 3-Axis accelerometer (marked **15** in [Figure A-5](#)). The acceleration change detected by this device can be used to determine the direction of motion of the kit. The PSoC 4200M communicates with the accelerometer through an I2C interface.

Note that the I2C accelerometer is connected on to the same I2C bus on which the KitProg USB-I2C bridge and the F-RAM is connected. The accelerometer is assigned with the I2C address 0x0F.

Refer to the [KXTJ2-1009 Datasheet](#) for more information on how to configure the accelerometer.

A.2.11 PWM Temperature Sensor

A PWM temperature sensor is included on the PSoC 4 M-Series Pioneer Kit (marked **19** in [Figure A-5](#)). This sensor is also known as a TMP05 sensor and can indicate the measured temperature by varying the duty cycle of the PWM output generated by the sensor.

Refer to the [TMP05 Datasheet](#) for more details on the operation of this sensor. Note that the sensor is configured to operate in daisy-chain mode.

A.2.12 Ambient Light Sensor

The PSoC 4 M-Series Pioneer Kit includes an analog ambient light sensor (marked **21** in [Figure A-5](#)). This is a photodiode configured in photoconductive mode for detecting the changes in the ambient light intensity.

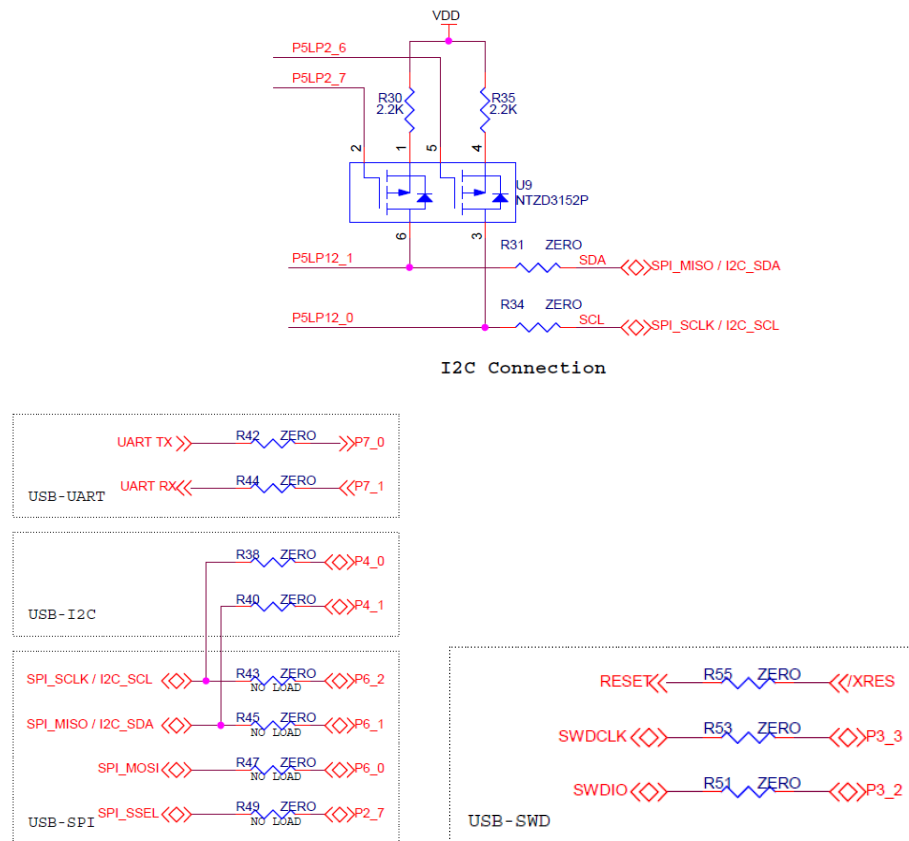
The photodiode is intended to be connected to the input of a transimpedance amplifier. A 220-K Ω resistor and a 0.1-uF capacitor are connected across the pins P5[1] and P5[2] to implement a transimpedance amplifier. Refer to the [TEMD6200FX01 datasheet](#) for the electrical characteristics of the photodiode.

A.2.13 Serial Interconnection Between PSoC 5LP and PSoC 4200M

In addition to use as an onboard programmer, the PSoC 5LP is used as a USB-Serial interface for the USB-UART bridge and USB-I2C bridge, as shown in [Figure A-11](#). The I2C bus contains firmware-controlled resistive pull-ups using FETs, which can be enabled or disabled using the PSoC 5LP pins. The USB-Serial pins of the PSoC 5LP are also available on the Arduino header; therefore, the PSoC 5LP can be used to control Arduino Shields with an SPI/I2C/UART interface.

Note: The KitProg does not support USB-SPI bridge functionality. The SPI pins of the PSoC 4200M are connected to the PSoC 5LP for enabling SPI communication between PSoC 5LP and PSoC 4200M for custom PSoC 5LP applications.

Figure A-11. Schematics of Serial Interface Connections and I2C Pull-Up via FET

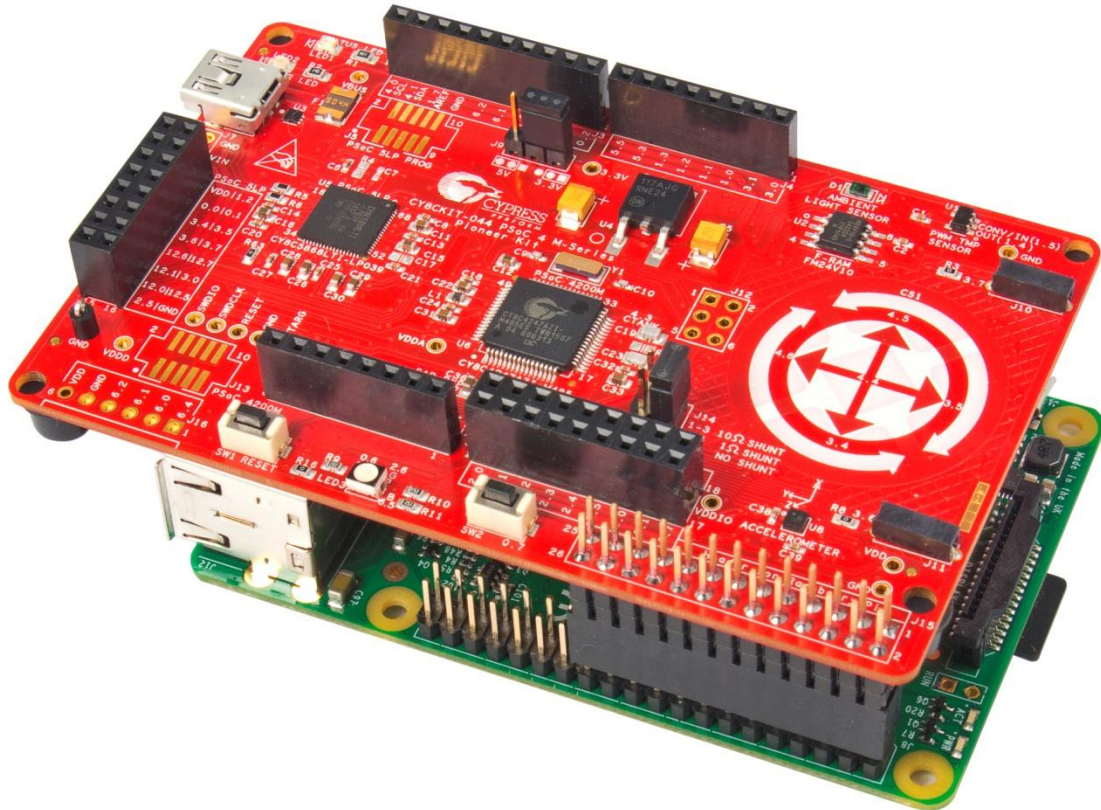


A.2.14 Raspberry Pi Compatible Header

The PSoC 4 M-Series Pioneer Kit has a Raspberry Pi compatible header (marked **14** in [Figure A-5](#)). This header (**J15**) is not populated by default. Refer to the “No Load Components” section of the [Bill of Materials](#) for the header part number. Alternately, you may use the header from Samtec Inc. (part number: [ESQ-113-44-G-D](#)). All the serial communication pins are brought onto this header. Note that the GPIO pins of the Raspberry Pi are not connected to the PSoC 4 M-Series Pioneer Kit using this interface.

The Raspberry Pi pins are intolerant to voltages above 3.3 V. A level translator is included on the board to automatically convert the digital logic voltages to the 3.3-V domain regardless of the power settings on the PSoC 4 M-Series Pioneer Kit.

Figure A-12. PSoC 4 M-Series Pioneer Kit Connected to a Raspberry Pi Model B

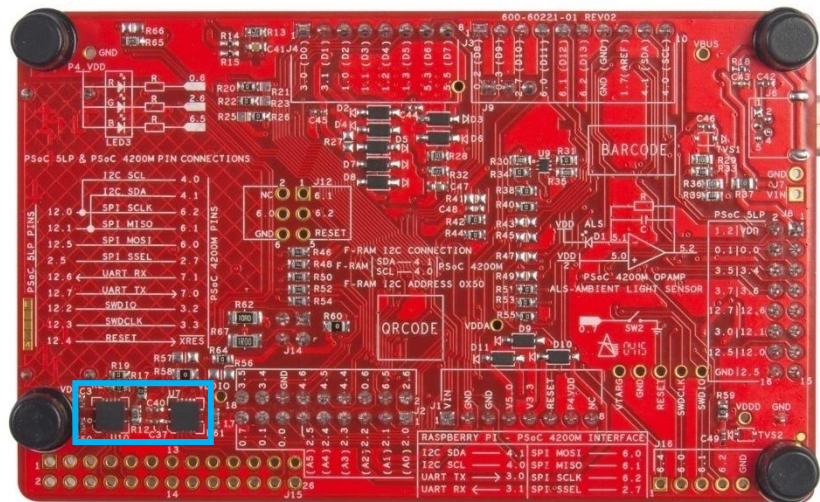
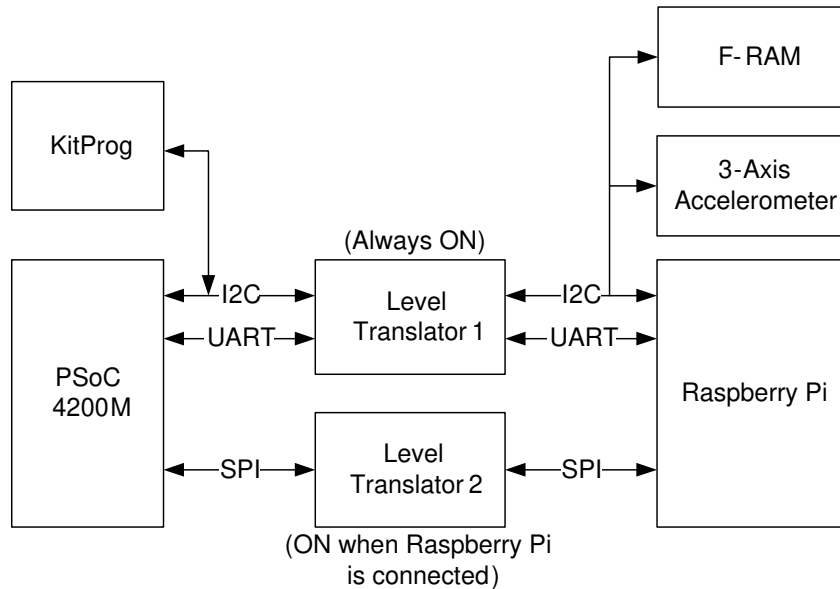


A.2.15 Level Translators

The PSoC 4 M-Series Pioneer Kit has two level translators. The level translators are used on this kit to translate digital 5-V logic signals to 3.3 V. One level translator connects the I2C and UART signals from PSoC 4200M to the Raspberry Pi compatible header. It also connects to the I2C bus of the accelerometer as shown in [Figure A-13](#).

The level translator, which connects the I2C and UART interfaces to the Raspberry Pi is always on. This ensures that you can read the accelerometer data without connecting a Raspberry Pi. This level translator also connects to the 3-Axis accelerometer and the Cypress F-RAM device. The level translator, which interfaces SPI lines of PSoC 4200M to the Raspberry Pi, is switched on automatically when a Raspberry Pi is connected to the PSoC 4 M-Series Pioneer Kit.

Figure A-13. Level Translators on PSoC 4 M-Series Pioneer Kit

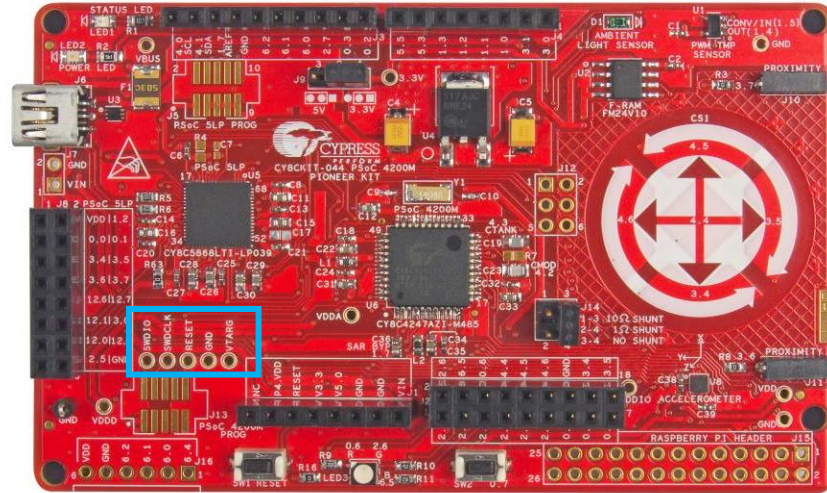


A.2.16 Test Points

The SWD programming pins for the PSoC 4200M are brought out as test points on the PSoC 4 M-Series Pioneer Kit. You can solder standard 100-mil connectors to these test points to program an external PSoC device using the onboard PSoC 5LP.

Note: You must remove the zero ohm resistors R51, R53, and R55 before programming an external PSoC device.

Figure A-14. SWD Programming Pins Brought as Test Points



All power domains (VDDA, VDDD, and VDDIO) including VIN, 3.3V and GND are also brought as test points for easy probing.

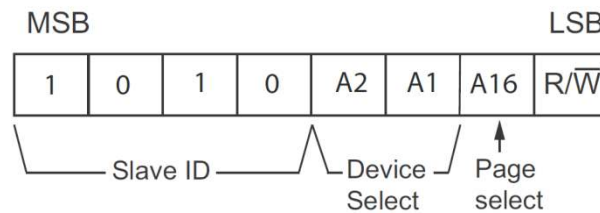
A.3 Using FM24V10 F-RAM

The PSoC 4 M-Series Pioneer board has an onboard ferroelectric RAM chip that can hold up to 1 Mb of data. The chip provides an I2C communication interface for data access. It is hardwired to the I2C interface (P4[0] and P4[1] of the PSoC 4200M); the same lines are routed to the PSoC 5LP I2C interface. Because the F-RAM device is an I2C slave, it can be accessed or shared among various I2C masters on the same lines. For more details on the F-RAM device, refer to the device [datasheet](#).

A.3.1 Address Selection

The slave address of the F-RAM device consists of three parts, as shown in [Figure A-15](#): slave ID, device select, and page select. Slave ID is an F-RAM family-specific ID provided in the datasheet of the particular F-RAM device. For the device used on the PSoC 4 M-Series Pioneer board (FM24V10), the slave ID is 1010b. Device select bits are set using the two physical pins A2 and A1 in the device. The setting of these two pins in PSoC 4 M-Series Pioneer board is controlled by resistors R21/R20 (A1) and R23/R22 (A2). Because the memory location in F-RAM is divided into two pages of 64 KB each, the page select bit is used to refer to one of the two pages in which the read or write operations will take place.

Figure A-15. F-RAM I2C Address Byte Structure



A.3.2 Write/Read Operation

The F-RAM device datasheet includes details on how to perform a write/read operation with the F-RAM. [Figure A-16](#) and [Figure A-17](#) provide a snapshot of the write/read packet structure as a quick reference.

Figure A-16. F-RAM Single-Byte and Multiple-Byte Write Packet Structure

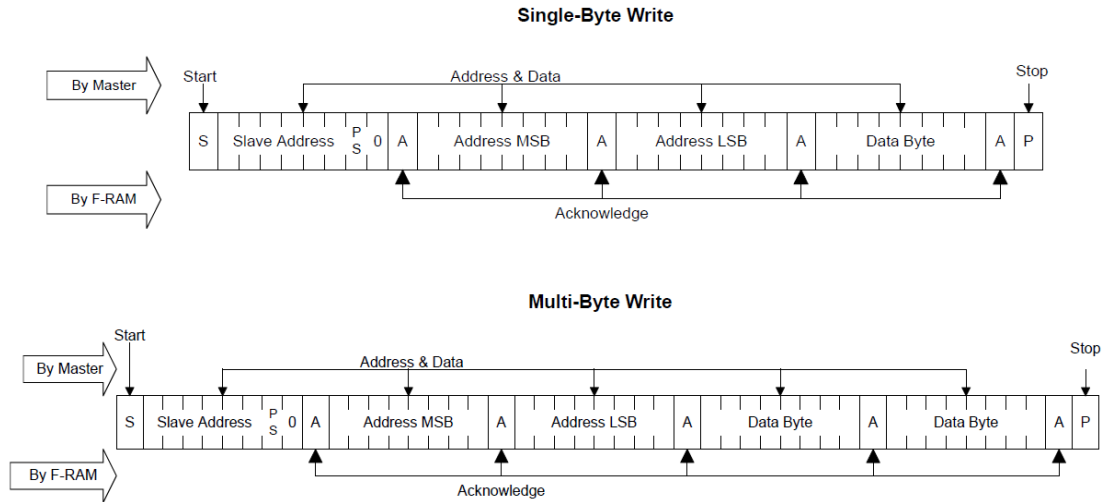
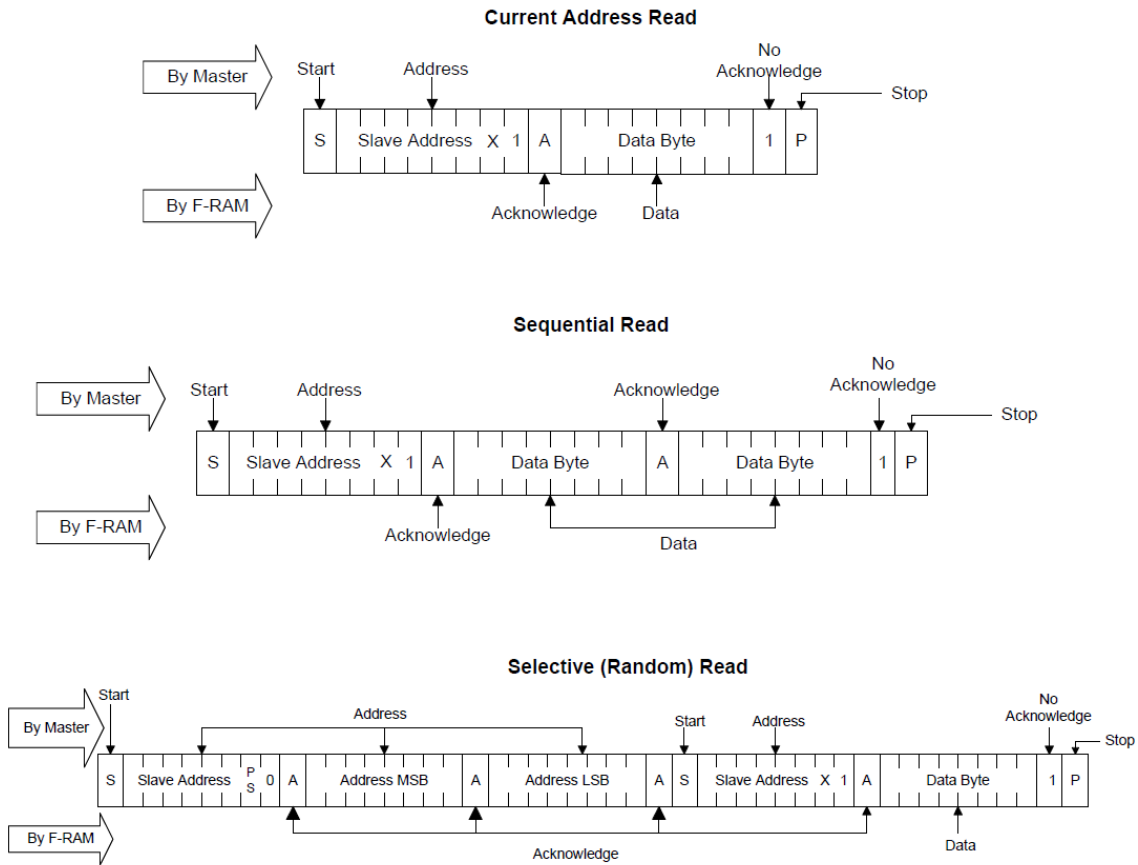


Figure A-17. F-RAM Read Packet Structure



As the figures show, operations start with the slave address followed by the memory address. For write operations, the bus master sends the slave address and memory address followed by one or more data bytes. Each byte of data is written to consecutive locations in the memory, and the memory generates an acknowledgement condition.

For 'Current Address Read' and 'Sequential Read', the bus master sends only the slave address. The memory address used is the same address that was set by the previous 'Write' or 'Selective Read' operation. For 'Selective Read' operations, after receiving the complete slave address and memory address, the memory will begin shifting data from the current address on the next clock

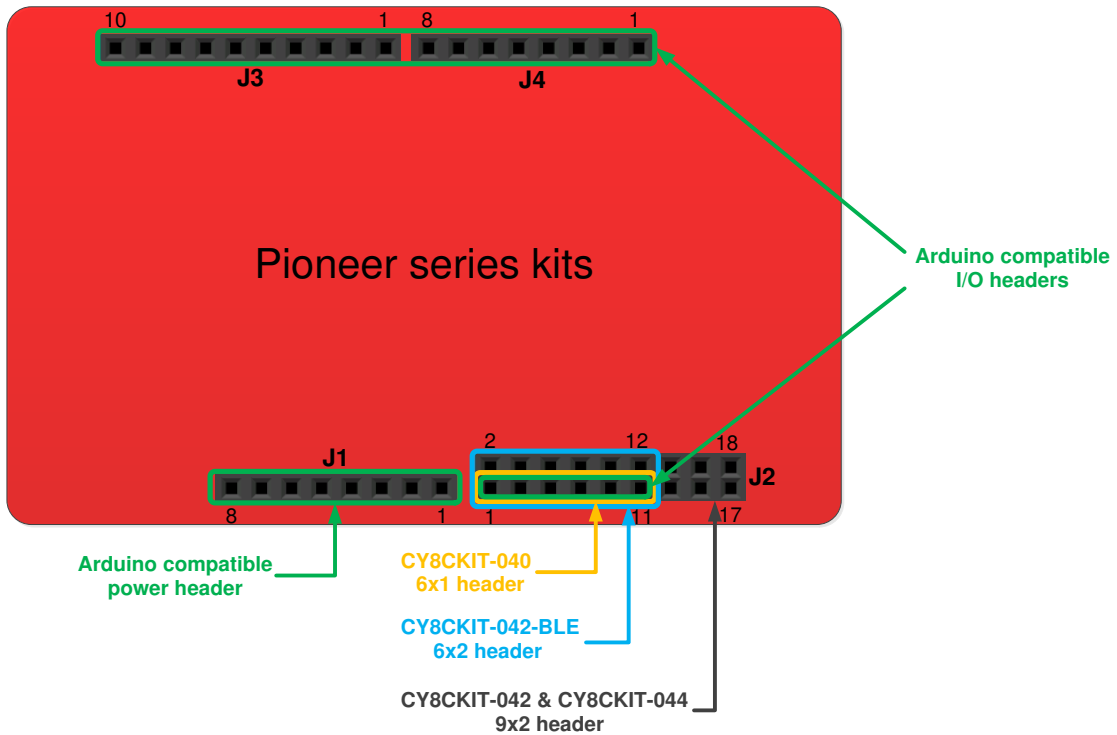
Note: The code example Sensor Hub uses the onboard F-RAM. See the [Sensor Hub](#) project to understand how to communicate with the F-RAM device using PSoC 4200M device. You can also communicate with the F-RAM using the Bridge Control Panel (BCP) software similar to the way you communicate with any other I2C slave device. Refer to the *KitProg User Guide* for more details on how to use the BCP to communicate with an I2C slave device.

A.4 Migrating Projects Across Different Pioneer Series Kits

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

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

Figure A-18. Pioneer series kits pin map



A.4.1 Arduino Uno Compatible Headers

Table A-1. J1 Arduino Compatible Header Pin Map

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

Table A - 2. J2 Arduino Compatible Header Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	A0	P2[0]	P0[0]	P3[0]	P2[0]
2	—	P0[2]*	—	P2[0]	P2[6]*
3	A1	P2[1]	P0[1]	P3[1]	P2[1]
4	—	P0[3]*	—	P2[1]*	P6[5]*
5	A2	P2[2]	P0[2]*	P3[2]	P2[2]
6	—	P4.VDD	—	P2[2]*	P0[6]*
7	A3	P2[3]	P0[4]*	P3[3]	P2[3]
8	—	P1[5]*	—	P2[3]*	P4[4]*
9	A4	P2[4]	P1[3]	P3[4]	P2[4]
10	—	P1[4]*	—	P2[4]*	P4[5]*
11	A5	P2[5]	P1[2]	P3[5]	P2[5]
12	—	P1[3]*	—	P2[5]*	P4[6]*
13	—	P0[0]	—	—	P0[0]
14	—	GND	—	—	GND
15	—	P0[1]	—	—	P0[1]
16	—	P1[2]*	—	—	P3[4]*
17	—	P1[0]	—	—	P0[7]*
18	—	P1[1]*	—	—	P3[5]*

* These pins are also used for onboard peripherals. Refer to the tables in the [Onboard Peripherals](#) section for connection details.

Table A - 3. J3 Arduino Compatible Header Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	D8	P2[6]	P1[4]	P0[5]	P0[2]
2	D9	P3[6]	P1[5]	P0[4]	P0[3]
3	D10	P3[4]	P1[6]	P0[2]	P2[7]
4	D11	P3[0]	P1[1]*	P0[0]	P6[0]
5	D12	P3[1]	P3[1]	P0[1]	P6[1]
6	D13	P0[6]	P1[7]	P0[3]	P6[2]
7	GND	GND	GND	GND	GND
8	AREF	P1[7]	NC	VREF	P1[7]
9	SDA	P4[1]	P1[3]	P3[4]	P4[1]
10	SCL	P4[0]	P1[2]	P3[5]	P4[0]

* These pins are also used for onboard peripherals. Refer to the tables in the [Onboard Peripherals](#) section for connection details.

Table A - 4. J4 Arduino Compatible Header Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	D0	P0[4]	P0[5]	P1[4]	P3[0]
2	D1	P0[5]	P0[6]	P1[5]	P3[1]
3	D2	P0[7]*	P0[7]	P1[6]	P1[0]
4	D3	P3[7]	P3[2]*	P1[7]	P1[1]
5	D4	P0[0]	P0[3]	P1[3]	P1[2]
6	D5	P3[5]	P3[0]	P1[2]	P1[3]
7	D6	P1[0]	P1[0]	P1[1]	P5[3]
8	D7	P2[7]	P2[0]*	P1[0]	P5[5]

* These pins are also used for onboard peripherals. Refer to the tables in the [Onboard Peripherals](#) section for connection details.

A.4.2 Onboard Peripherals

Table A - 5. CapSense Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	CapSense Sensor 1	P1[1]	—	P2[1]	P4[4]
2	CapSense Sensor 2	P1[2]	—	P2[2]	P4[5]
3	CapSense Sensor 3	P1[3]	—	P2[3]	P4[6]
4	CapSense Sensor 4	P1[4]	—	P2[4]	P3[4]
5	CapSense Sensor 5	P1[5]	—	P2[5]	P3[5]
6	CMOD	P4[2]	P0[4]	P4[0]	P4[2]
7	CTANK	P4[3]	P0[2]	P4[1]	P4[3]

Table A - 6. Proximity header Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	PROXIMITY	—	P2[0]	P2[0]	P3[7]
2		—	—	—	P3[6]

Table A - 7. RGB LED Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	Red	P1[6]	P3[2]	P2[6]	P0[6]
2	Green	P0[2]	P1[1]	P3[6]	P2[6]
3	Blue	P0[3]	P0[2]	P3[7]	P6[5]

Table A - 8. User Switch Pin Map

#	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	SW2	P0[7]	—	P2[7]	P0[7]

A.5 KitProg Status LED States

The KitProg Status LED on the PSoC 4 M-Series Pioneer Kit indicates the status of the KitProg operation using different blink rates. Table A - 9 shows the KitProg LED States and the corresponding status of the KitProg.

Table A - 9. Meaning of KitProg LED Indications

User Indication	Scenario	Action Required by User
LED blinks fast: Frequency = 4.00 Hz	LED starts blinking at power up, if bootloadable file is corrupt.	Bootload the <i>KitProg.cyacd</i> file: In PSoC Programmer, connect to the kit, go to the Utilities tab, and press the Upgrade Firmware button.
LED blinks slow: Frequency = 0.67 Hz	Entered Bootloader mode by holding the Reset button during kit power-up.	Release the Reset button and re-plug the kit if you entered this mode by mistake. If the mode entry was intentional, bootload the new <i>.cyacd</i> file using the Bootloader Host tool available in PSoC Creator.
LED blinks very fast: Frequency = 15.0 Hz	SWD or I2C operation is in progress. The Kit's COM port connect / disconnect event (only one blink).	In PSoC Programmer, watch the log window for status messages for SWD operations. In the BCP, the LED blinks on I2C command requests. In BCP or any other serial port terminal program, distinguish the kit's COM port number by the blinking LED when the port is connected or disconnected.
LED is ON	USB enumeration successful. Kit is in the idle state waiting for commands.	PSoC Creator, PSoC Programmer, BCP, and any serial port terminal program can use the kit functions.
LED is OFF	Power LED is ON	This means that the USB enumeration was unsuccessful. This may happen if the kit is not powered from the USB host. Verify the USB cable and check if PSoC Programmer is installed on the PC.

A.6 Bill of Materials

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
1	1	600-60221-01 REV03	—	PCB, 98.66 mm x 61.70 mm, High Tg, ENIG finish, 4 layer, Color = RED, Silk = WHITE	Cypress Semiconductor	600-60221-01
2	25	C1,C2,C3,C6,C8,C13,C14,C18,C20,C21,C24,C25,C27,C32,C35,C37,C38,C39,C40,C42,C44,C45,C46,C49,C50	0.1 uFd	CAP .1UF 16V CERAMIC X5R 0402	TDK Corporation	C1005X5R1A104K050BA
3	1	C4	22 uFd 16v	CAP TANT 22UF 16V 10% 1210	AVX Corporation	TPSB226K016R0600
4	1	C5	10 uFd 20v	CAP TANT 10UF 20V 10% 1210	AVX Corporation	TAJB106K020RNJ
5	1	C9	18 pF	CAP CER 18PF 50V 1% NP0 0402	Murata Electronics	GRM1555C1H180FA01D
6	1	C10	36 pF	CAP CER 36PF 50V 5% NP0 0402	Murata Electronics	GRM1555C1H360JA01D
7	14	C11,C12,C15,C16,C22,C26,C28,C29,C30,C31,C33,C34,C36,C47	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T
8	1	C19	10000 pFd	CAP CER 10000PF 50V 5% NP0 0805	Murata	GRM2195C1H103JA01D
9	1	C23	2200 pFd	CAP CER 2200PF 50V 5% NP0 0805	Murata	GRM2165C1H222JA01D
10	1	C41	0.1 uF	CAP CER 0.1UF 16V 10% X7R 0603	Murata	GRM188R71C104KA01D
11	1	C43	0.01 uFd	CAP 10000PF 16V CERAMIC 0402 SMD	TDK Corporation	C1005X7R1C103K050BA
12	1	D1	TEMD6200FX01	AMBIENT LIGHT SENSOR 0805 SMD	Vishay Semiconductor	TEMD6200FX01
13	10	D2,D3,D4,D5,D6,D7,D8,D9,D10,D11	MBR0520L	DIODE SCHOTTKY 0.5A 20V SOD-123	Fairchild Semiconductor	MBR0520L
14	1	F1	FUSE	PTC RESETTABLE .50A 15V 1812	Bourns	MF-MSMF050-2
15	2	J1,J4	8x1 RECP	CONN HEADER FEMALE 8POS .1" GOLD	Protectron Electromech	P9401-08-21
16	1	J2	9x2 RECP	CONN HEADER FMAL 18PS.1" DL GOLD	Protectron Electromech	P9403-18-21
17	1	J3	10x1 RECP	CONN HEADER FMALE 10POS .1" GOLD	Protectron Electromech	P9401-10-21
18	1	J6	USB MINI B	MINI USB RCPT R/A DIP	Molex Inc	0548190519
19	1	J8	8x2 RECP	CONN HEADER FMAL 16PS.1" DL GOLD	Protectron Electromech	P9403-16-21
20	1	J9	3p_jumper	CONN HEADR BRKWAY .100 3POS STR	Protectron Electromech	P9101-03-12-1

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
21	2	J10,J11	1X1 RECP	CONN RCPT 1POS .100" SNGL HORZ	Samtec Inc	BCS-101-L-S-HE
22	1	J14	2X2 RECP	CONN HEADER .100 DUAL STR 4POS	Protectron Electromech	P9103-04-12-1
23	1	LED1	Status LED Green	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
24	1	LED2	Power LED Amber	LED 595NM AMB DIFF 0805 SMD	Avago Technologies	HSMA-C170
25	1	LED3	RGB LED	LED RED/GREEN/BLUE PLCC4 SMD	Cree, Inc.	CLV1A-FKB- CJ1M1F1BB7R4S3
26	2	L1,L2	330 ohm @ 100MHz	FILTER CHIP 330 OHM 1.5A 0603	Murata Electronics	BLM18SG331TN1D
27	1	R1	820 ohm	RES 820 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ821V
28	1	R2	560 ohm	RES 560 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ561V
29	21	R3,R8,R16,R17,R19,R20,R22,R24,R26,R31,R34,R38,R40,R42,R44,R51,R53,R55,R59,R63,R64	ZERO	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V
30	2	R5,R6	22E	RES 22 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF22R0V
31	3	R9,R30,R35	2.2K	RES 2.2K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ222V
32	2	R10,R11	1.5K	RES 1.5K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ152V
33	3	R12,R61,R65	10K	RES 10K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ103V
34	1	R13	220K	RES SMD 220K OHM 5% 1/10W 0603	Panasonic - ECG	ERJ-3GEYJ224V
35	2	R14,R15	10K	RES 10K OHM 1/16W 5% 0402 SMD	Yageo	RC0402JR-0710KL
36	1	R18	100K	RES 100K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ104X
37	1	R28	120 ohm	RES 120 OHM 1/10W 1% 0603 SMD	Panasonic - ECG	ERJ-3EKF1200V
38	2	R29,R36	30K	RES 30K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ303V
39	1	R32	220 ohm	RES SMD 220 OHM 1% 1/10W 0603	Panasonic - ECG	ERJ-3EKF2200V
40	2	R33,R39	15K	RES 15K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ153V
41	3	R37,R58,R60	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V
42	5	R46,R48,R50,R52,R54	560 ohm	RES 560 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ561V
43	1	R62	10 ohm	RES SMD 10 OHM 1% 1/4W 1206	Stackpole Electronics Inc	RMCF1206FT10R0
44	1	R67	1 ohm	RES SMD 1 OHM 1% 1/4W 1206	Stackpole Electronics Inc	RMCF1206FT1R00

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
45	2	SW1,SW2	SW PUSHBUTTON	SWITCH TACTILE SPST-NO 0.05A 12V	Panasonic - ECG	EVQ-PE105K
46	1	TP8	BLACK	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001
47	1	U1	TMP05	IC PWM TEMP SNSR CMOSTTL SOT23-5	Analog Devices Inc	TMP05ARTZ-500RL7
48	1	U2	F-RAM	F-RAM 1-Mbit (128K X 8) I2C interface	Cypress Semiconductor	FM24V10-G
49	1	U3	PUSBM12VX4- TL,115	TVS DIODE 5.5VWM 12VC 6HXSON	NXP Semiconductors	PUSBM12VX4-TL,115
50	1	U4	NCP1117DTARK G	IC REG LDO ADJ 1A DPAK	ON Semiconductor	NCP1117DTARKG
51	1	U5	PSoC 5LP	68QFN PSoC 5LP chip for USB debug channel and USB-Serial interface	Cypress Semiconductor	CY8C5868LTI-LP039
52	1	U6	PSoC 4200M	68TQFP PSoC 4M	Cypress Semiconductor	CY8C4247AZI-M485
53	2	U7,U10	TXS0104ERGYR	IC 4BIT NON-INV TRANSLTR 14-QFN	Texas Instruments	TXS0104ERGYR
54	1	U8	KXTJ2-1009	IC DIGITAL (I2C) ACCELEROMETER TRI-AXIS 12-LGA	Kionix Inc	KXTJ2-1009
55	1	U9	DUAL PMOS	MOSFET 2P-CH 20V 430MA SOT-563	ON Semiconductor	NTZD3152PT1G
56	1	Y1	32.768KHz	CRYSTAL 32.768KHZ 12.5PF SMD	ECS Inc	ECS-.327-12.5-39-TR
57	4	N/A	N/A	BUMPON CYLINDRICAL .312X.215 BLACK	3M	SJ61A6
58	1	J9	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E
59	1	J14	Install jumper across pins 3 and 4	Rectangular Connectors MINI JUMPER GF 6.0MM CLOSE TYPE BLACK	Kobiconn	151-8010-E
No load components						
60	2	J5,J13	50MIL KEYED SMD	CONN HEADER 10 PIN 50MIL KEYED SMD	Samtec	FTSH-105-01-L-DV-K
61	1	J7	2 PIN HDR	CONN HEADER FEMALE 2POS .1" GOLD	Sullins Connector Solutions	PPPC021LFBN-RC
62	1	J12	3x2 RECPT	CONN HEADER FMAL 6PS .1" DL GOLD	Sullins Connector Solutions	PPPC032LFBN-RC
63	1	J15	CONNECTOR EDGE 13x2	STACKING HEADER, RASPERRY PI	Samtec Inc	ESQ-113-44-G-D
64	1	J16	6X1 RECP RA	CONN FEMALE 6POS .100" R/A GOLD	Sullins Connector Solutions	PPPC061LGBN-RC
65	2	C7,C48	0.1 uFd	CAP .1UF 16V CERAMIC Y5V 0402	TDK Corporation	C1005X5R1A104K050BA
66	1	C17	1.0 uFd	CAP CERAMIC 1.0UF 25V X5R 0603 10%	Taiyo Yuden	TMK107BJ105KA-T

Item	Qty	Reference	Value	Description	Manufacturer	Mfr Part Number
67	9	R21,R23,R25,R27,R43,R45, R47,R49,R66	Zero Ohm	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEY0R00V
68	2	R4,R41	4.7K	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V
69	3	TP8,TP9,TP14	BLACK	TEST POINT 43 HOLE 65 PLATED BLACK	Keystone Electronics	5001
70	10	TP1,TP3,TP4,TP5,TP6,TP7, TP10,TP11,TP12,TP13	RED	TEST POINT 43 HOLE 65 PLATED RED	Keystone Electronics	5000
71	2	TVS1,TVS2	5V 350W	TVS UNIDIR 350W 5V SOD-323	Diodes Inc.	SD05-7
72	3	R7,R56,R57	ZERO	RES 0.0 OHM 1/8W 0805 SMD	Panasonic-ECG	ERJ-6GEY0R00V

Revision History



Document Title: CY8CKIT-044 PSoC® 4 M-Series Pioneer Kit Guide		
Document Number: 001-96598		
Revision	Issue Date	Description of Change
**	02/26/2015	Initial version of the kit guide
*A	04/02/2015	Updated the Code Examples section. Updated the kit name to PSoC 4 M-Series Pioneer Kit. Updated all images with high resolution images. Updated links and cross references. Updated the Project Description section. Updated the Verify Output section. Added "Migrating Projects Across Different Pioneer Series Kits" section. Updated the Expansion Connectors sub section in Appendix section. Updated the kit schematics and BOM in Appendix section.
*B	05/29/2015	Updated KitProg . Added Mass Storage Programmer . Updated the Action for Swipe Right and Swipe Left in Table 4-2 . Updated the second Note in 4.5.3 Verify Output . Added a Note, and updated Steps 12 and 13 in 4.7.3 Verify Output . Added 4.8 Raspberry Pi . Updated Table A - 9 . Updated Figure A-22 .
*C	05/11/2017	Updated the Cypress logo and copyright information.
*D	03/12/2018	Updated PSoC Creator images Changed "example projects" to "code examples"
*E	12/10/2019	Updated Figure 1-1 and Figure 1-2 Added serial terminal figures Figure 4-15 and Figure 4-16