

**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-048

# PSoC<sup>®</sup> Analog Coprocessor Pioneer Kit Guide

Doc. # 002-11190 Rev. \*C

Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
[www.cypress.com](http://www.cypress.com)

© Cypress Semiconductor Corporation, 2016-2019. 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](http://cypress.com). Other names and brands may be claimed as property of their respective owners.

# Contents



<b>Safety Information</b>	<b>4</b>
<b>1. Introduction</b>	<b>5</b>
1.1 Kit Contents .....	6
1.2 Board Details .....	7
1.3 PSoC Creator .....	9
1.4 Getting Started.....	12
1.5 Additional Learning Resources .....	12
1.6 Technical Support.....	13
1.7 Documentation Conventions.....	13
1.8 Acronyms .....	13
<b>2. Software Installation</b>	<b>15</b>
2.1 Before You Begin.....	15
2.2 Install Software .....	15
2.3 Uninstall Software .....	17
<b>3. Kit Operation</b>	<b>18</b>
3.1 Theory of Operation.....	18
3.2 Flexible Prototyping .....	32
3.3 KitProg2 Functions .....	34
<b>4. Code Examples</b>	<b>36</b>
4.1 Using the Kit Code Example.....	36
4.2 Code Examples .....	38
<b>A. Appendix</b>	<b>39</b>
A.1 Schematics .....	39
A.2 Hardware Functional Description.....	39
A.3 Using the FM24V10 F-RAM.....	44
A.4 Migrating Projects Across Different Pioneer Series Kits .....	47
A.5 Bill of Materials .....	51
<b>Revision History</b>	<b>52</b>

# Safety Information



The CY8CKIT-048 PSoC<sup>®</sup> Analog Coprocessor 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 PSoC Analog Coprocessor Pioneer Kit, as shipped from the factory, has been verified to meet with the requirements of CE as a Class A product.



The PSoC Analog Coprocessor Pioneer Kit contains 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 PSoC Analog Coprocessor Pioneer Kit in the protective shipping package.



## End-of-Life/Product Recycling

The end-of-life cycle for this kit is five years from the date of manufacture mentioned on the back of the box. Contact the nearest recycler to discard the kit.

## General Safety Instructions

### ESD Protection

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

### Handling Boards

PSoC Analog Coprocessor Pioneer Kit is sensitive to ESD. Hold the board only by its edges. After removing the board from the 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-048 PSoC® Analog Coprocessor Pioneer Kit. This kit enables you to evaluate and develop

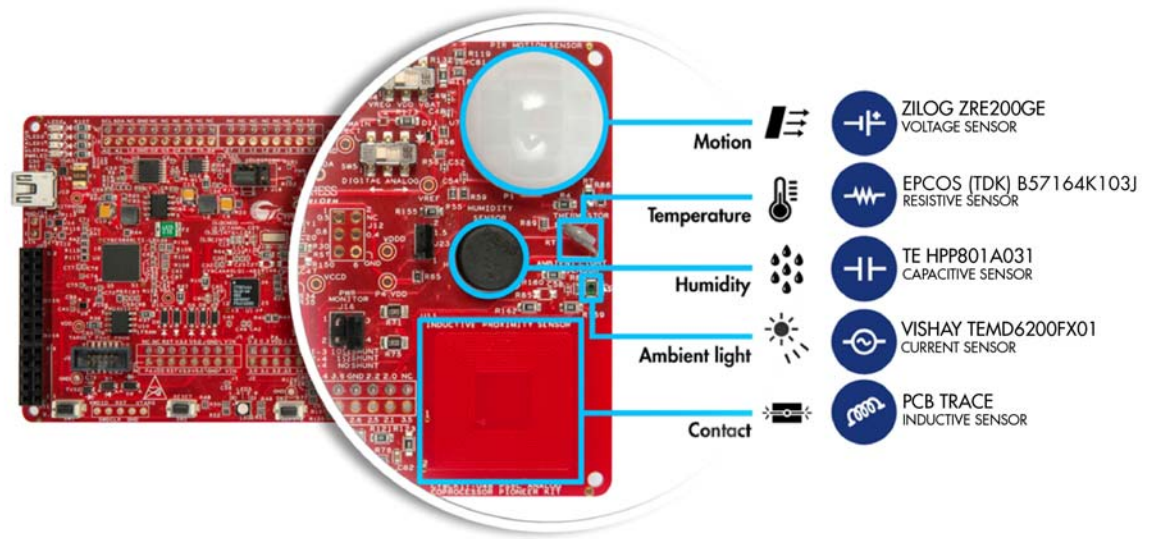
our application using the PSoC Analog Coprocessor device family. The PSoC Analog Coprocessor simplifies the design of sensor-based systems by delivering a scalable and reconfigurable architecture that integrates programmable Analog Front Ends (AFE). It has a signal processing engine (32-bit Arm® Cortex®-M0+) to calibrate and tune the AFE in software. Additionally, the PSoC Analog Coprocessor enables applications to send aggregated, pre-processed, and formatted sensor data over serial communication interfaces to host processors.

Systems that use multiple analog sensors usually require multiple specialized ICs to implement the AFE, which increases BOM cost and PCB size. Systems designed for IoT applications must combine data from multiple sensors to enable new sensing capabilities, commonly known as sensor fusion. Sensor fusion solutions often require custom AFEs. The PSoC Analog Coprocessor reduces the need for specialized ICs, offering the ability to create custom AFEs in a single-chip solution. [Figure 1-1](#) shows the analog sensors interface on PSoC Analog Coprocessor Pioneer Kit.

The PSoC Analog Coprocessor Pioneer Kit offers footprint-compatibility with Arduino™ shields and baseboards. This kit features five analog sensors. They are pyroelectric infrared (PIR) motion sensor, thermistor, humidity sensor, ambient light sensor (ALS), and inductive proximity sensor (IPS). It has an RGB LED, two user-configurable push-button switches, one reset push-button switch, an onboard programmer/debugger with USB-UART/I2C bridge, and a Cypress F-RAM™. This kit supports operating voltages of 1.8 V, 3.0 V, 3.3 V, or 5 V.

If you are new to PSoC Creator, see the documentation on the [PSoC Creator home page](#). You can also refer the application note, [AN211293 – Getting Started with PSoC Analog Coprocessor](#), which gives an introduction to the PSoC Analog Coprocessor.

Figure 1-1. Analog Sensors



## 1.1 Kit Contents

The PSoC Analog Coprocessor Pioneer Kit contains the following:

- PSoC Analog Coprocessor Pioneer board
- USB Standard-A to Mini-B cable
- Four jumper wires (4 inches each)
- Five connectors (one 10x1, two 8x1, one 6x1, and one 4x1)
- One metal disk
- Quick Start Guide

Figure 1-2. Kit Contents



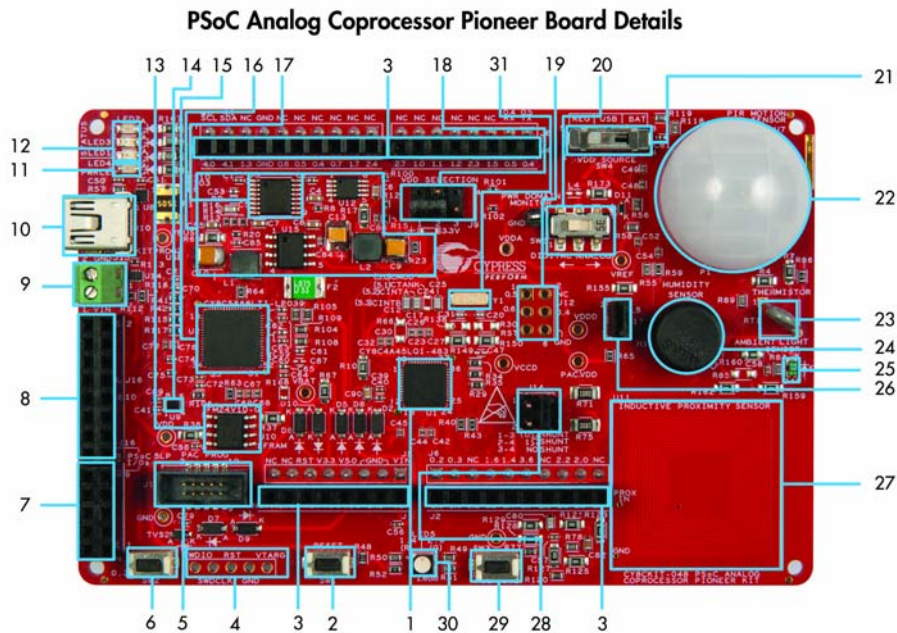
Inspect the contents of the kit; if you find any part missing, contact your nearest Cypress sales office for help: [www.cypress.com/support](http://www.cypress.com/support).



## 1.2 Board Details

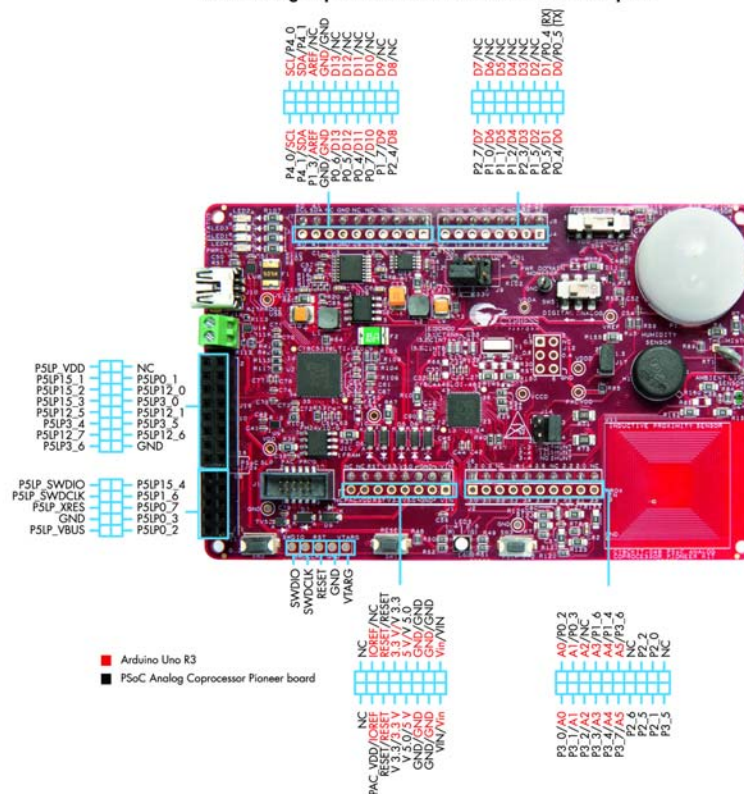
As shown in [Figure 1-3](#), the PSoC Analog Coprocessor Pioneer Kit features five onboard sensors. The kit has two user buttons and an RGB LED. It also has a reset button, a power LED, and three status LEDs. The kit can be powered from three power sources: USB, coin cell, or an external power supply. Refer to [Power Supply System on page 40](#) for details. There is a power selection jumper (J9), which allows you to select the kit operating voltage. The kit can operate at 1.8 V, 3.3 V, or 5 V when powered from the USB connector or external power supply and will operate at 3.0 V when powered from the coin cell. The USB connector is used as a power source and for programming and debugging your application on the PSoC Analog Coprocessor.

Figure 1-3. PSoC Analog Coprocessor Pioneer Kit Top View



- |  |   |
|--|---|
| 1. PSoC Analog Coprocessor (CY8C4A45LQI-483, U1)                         | 17. Cypress PMIC (MB39C011APFT-G-BND-ERE1, U3)                  |
| 2. PSoC Analog Coprocessor reset button (SW1)                            | 18. System Power (VDD) selection jumper (J9)                    |
| 3. Arduino™ Uno shield compatible power and I/O headers (J1, J2, J3, J4) | 19. Arduino ICSP compatible header (J12)                        |
| 4. PSoC Analog Coprocessor 5-pin program and debug test points           | 20. Power domain selection switch for current measurement (SW5) |
| 5. PSoC Analog Coprocessor 10-pin program and debug header (J13)         | 21. VDD source selection switch (SW4)                           |
| 6. User button (SW2)   | 22. PIR Motion Sensor (ZRE200GE, U7)                            |
| 7. KitProg2 custom application/programming header (J11)                  | 23. Thermistor (B57164K103J, RT1)                               |
| 8. KitProg2 I/O header (J16)   | 24. Humidity Sensor (HPP801A031, H1)                            |
| 9. External power supply connector (J15)                                 | 25. Ambient Light Sensor (TMD6200FX01, D4)                      |
| 10. KitProg2 USB Mini-B connector (J10)                                  | 26. Humidity Sensor calibration jumper (J17)                    |
| 11. Power LED (LED4)   | 27. Inductive Proximity Sensor (U11)                            |
| 12. KitProg2 status LEDs (LED1, LED2, LED3)                              | 28. Current measurement jumper (J14)                            |
| 13. Cypress F-RAM 1 Mb (FM24V10-G, U10)                                  | 29. KitProg2 user button (SW3)                                  |
| 14. Voltage Level Translator for F-RAM (U9)                              | 30. RGB LED (LED5)  |
| 15. KitProg2 (PSoC 5LP) programmer and debugger (CY8C5868LTI-PO39, U2)   | 31. 32.768 kHz crystal oscillator (Y1)                          |
| 16. Voltage regulator circuit  |   |

## PSoC Analog Coprocessor Pioneer Board Pinout Description



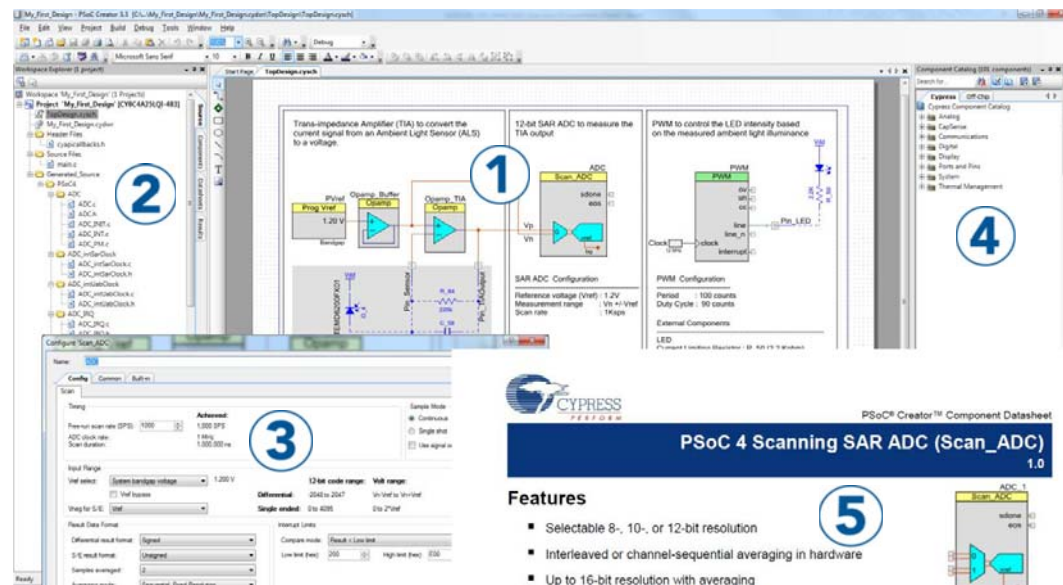
Jumper/Switch	Purpose	Default Position
J9	System Power (VDD Voltage) Selection	1–2 (3.3 V)
J14	Shunt Selection for Current Measurement	3–4 (NO SHUNT)
J17	Humidity Sensor Calibration	1–2 (SENSOR CONNECTED)
SW4	VDD Source Selection	REG
SW5	Power Domain Monitor Selection	DIGITAL

## 1.3 PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, PSoC 5LP, and PSoC Analog Coprocessor. See Figure 1-5. With PSoC Creator, you can:

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

Figure 1-5. 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](http://www.cypress.com/psoccreator).

### 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 1-6.

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 1-7.

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

- Filter for examples based on a device family or keyword.
- Select from the list of examples offered based on the **Filter Options**.
- View the project documentation for the selection (on the **Documentation** tab).
- View the code for the selection on the **Sample Code** tab. You can also copy and paste code from this window to your project, which can help speed up code development.

- Create a new workspace for the code example or add to your existing workspace. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 1-6. Code Examples in PSoC Creator

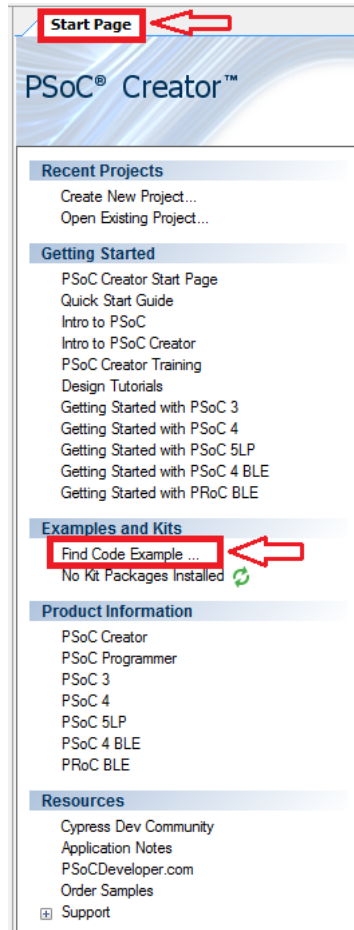
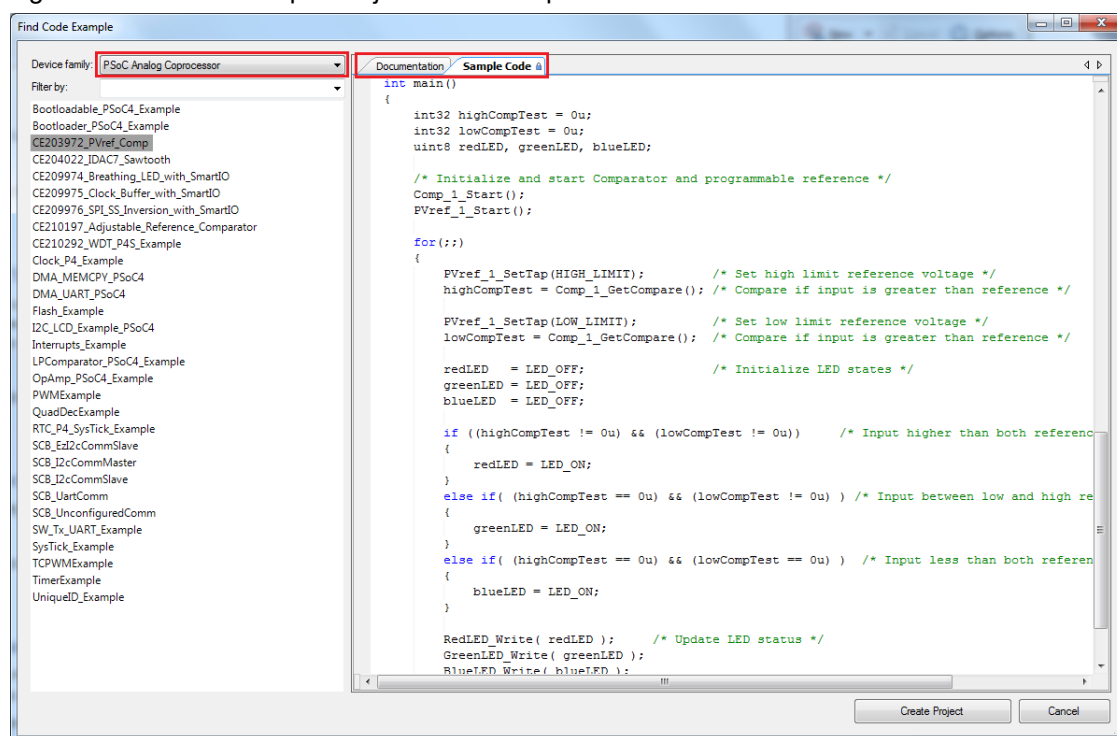


Figure 1-7. Code Example Projects with Sample Code



## Kit Code Examples:

You can access the installed kit code examples from the PSoC Creator Start Page. To access these examples, expand **Kits** under the section **Examples and Kits**; then, expand the specific kit to see the code examples. Refer to the [Code Examples chapter on page 36](#) for a list of code examples that you can use on this kit.

## PSoC Creator Help:

Launch PSoC Creator and navigate to the following items:

- **Quick Start Guide:** Choose **Help > Documentation > Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **Simple Component Code Examples:** Choose **File > Code Example**. These examples demonstrate how to configure and use PSoC Creator Components. To access examples related to a specific Component, right-click on the Component in the schematic or in the Component Catalog. Select the **Find Code Example** option in the context menu that appears.
- **System Reference Guide:** Choose **Help > System Reference > 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 Analog Coprocessor Component Datasheets](#) page for a list of all PSoC Analog Coprocessor Component datasheets.
- **Document Manager:** PSoC Creator provides a document manager to help you easily find and access the document resources. To open the document manager, choose the menu item **Help > Document Manager**.



## 1.4 Getting Started

This guide will help you get acquainted with the PSoC Analog Coprocessor Pioneer Kit:

- The [Software Installation chapter on page 15](#) 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 on page 18](#) describes the major features of the PSoC Analog Coprocessor Pioneer Kit and functionalities such as programming, debugging, and the USB-UART and USB-I2C bridges.
- The [Code Examples chapter on page 36](#) describes multiple PSoC Analog Coprocessor code examples that will help you understand how to create your own PSoC projects.
- The [Appendix on page 39](#) 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](http://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 Analog Coprocessor:

- Overview: [PSoC Portfolio](#) and [PSoC Roadmap](#).
- Product Selectors: [PSoC Analog Coprocessor 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](#), [PSoC 4200M](#), [PSoC 4200L](#), [PSoC 4000S](#) and [PSoC Analog Coprocessor](#) device families.
- [Getting Started with PSoC Analog Coprocessor](#) provides information on how to get started with PSoC Analog Coprocessor.
- [CapSense Design Guide](#): Learn how to design capacitive touch-sensing applications with the PSoC 4 and PSoC Analog Coprocessor 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/PSoC 4/PSoC 5LP Code Examples](#) web page for a list of all available PSoC Creator code examples. To access code examples from within PSoC Creator – see [PSoC Creator Code Examples on page 9](#).
- [Technical Reference Manuals \(TRM\)](#) provide detailed descriptions of the architecture and registers for each of the PSoC 4 and PSoC Analog Coprocessor family of devices.
- [Development Kits](#):
  - [CY8CKIT-041](#), [CY8CKIT-046](#), [CY8CKIT-044](#), [CY8CKIT-042](#), [CY8CKIT-040](#) and [CY8CKIT-048](#) are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino-compatible Shields and Digilent Pmod Peripheral Modules.
  - [CY8CKIT-049](#) and [CY8CKIT-043](#) are very low-cost prototyping platforms for sampling PSoC 4 devices.
  - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
  - The [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.
- PSoC Creator Training: Visit [www.cypress.com/go/creatorstart/creatortraining](http://www.cypress.com/go/creatorstart/creatortraining) for a comprehensive list of video trainings on PSoC Creator.

- Learning from Peers: Visit [www.cypress.com/forums](http://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. 2 (in the USA) or +1 (408) 943-2600 Ext. 2 (International).

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

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

## 1.7 Documentation Conventions

Table 1-2. 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
<b>Bold</b>	Displays commands, menu paths, and icon names in procedures: Click the <b>File</b> icon and then click <b>Open</b> .
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes cautions or unique functionality of the product.

## 1.8 Acronyms

Table 1-3. Acronyms Used in this Document

Acronym	Definition
ADC	Analog-to-Digital Converter
AFE	Analog Front End
AMUX	Analog Multiplexer
BOM	Bill of Materials
CMP	Comparator
CPU	Central Processing Unit
DAC	Digital to Analog Converter
DC	Direct Current
DNL	Differential Non-Linearity
DPDT	Double-Pole, Double-Throw
ESD	Electrostatic Discharge
F-RAM	Ferroelectric Random Access Memory

Table 1-3. Acronyms Used in this Document (*continued*)

Acronym	Definition
FET	Field-effect Transistor
GPIO	General-Purpose Input/Output
HPF	High Pass Filter
IC	Integrated Circuit
ICSP	In-Circuit Serial Programming
IDAC	Current DAC
IDE	Integrated Design Environment
IoT	Internet of Things
IR	Infrared
I <sup>2</sup> C	Inter-Integrated Circuit
LCD	Liquid Crystal Display
LED	Light-emitting Diode
LPF	Low Pass Filter
LSB	Least Significant Bit
PCB	Printed Circuit Board
PGA	Programmable Gain Amplifier
PIR	Pyroelectric Infrared
PPTC	Polymeric Positive Temperature Coefficient
PRB	Programmable Reference Block
PSoC	Programmable System-on-Chip
PWM	Pulse Width Modulation
RGB	Red Green Blue
SAR	Successive Approximation Register
SCB	Serial Communication Block
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
SRAM	Serial Random Access Memory
SWD	Serial Wire Debug
TIA	Transimpedance Amplifier
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WCO	Watch Crystal Oscillator



## 2. Software Installation



This chapter describes the steps to install the software tools and packages on a PC for using the PSoC Analog Coprocessor 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, this is not required to run 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 PSoC Analog Coprocessor Pioneer Kit software:

1. Download the PSoC Analog Coprocessor Pioneer Kit software from [www.cypress.com/CY8CKIT-048](http://www.cypress.com/CY8CKIT-048). The kit software is available in three different formats for download:
  - a. **CY8CKIT-048 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-048 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-048 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 similar to a virtual CD/DVD using virtual drive programs such as Virtual CloneDrive and MagicISO. This file includes all the required software, utilities, drivers, hardware files, and user documents.
2. If you have downloaded the ISO file, mount it on a virtual drive. 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 computer. 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-048** to start the PSoC Analog Coprocessor Pioneer Kit installation, as shown in Figure 2-1.

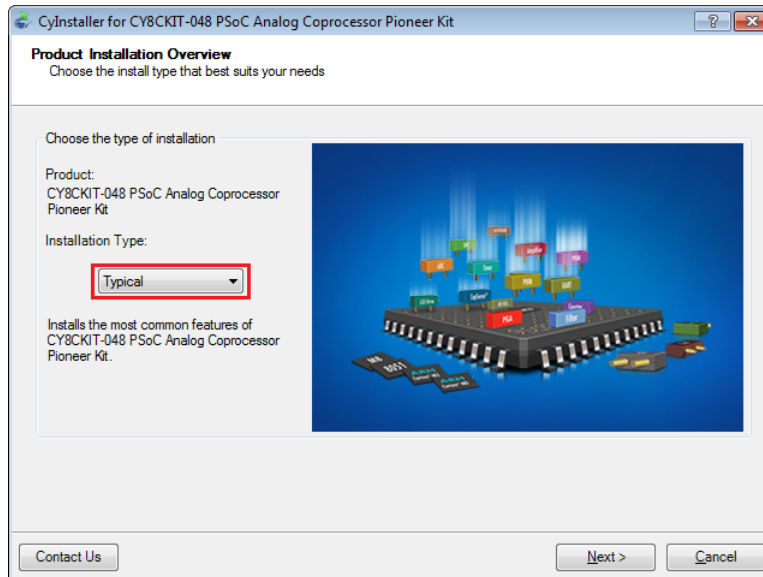
Figure 2-1. Kit Installer Screen



4. Select the folder in which you want to install the PSoC Analog Coprocessor Pioneer Kit-related files. Choose the directory and click **Next**.
5. When you click **Next**, the installer automatically installs the required software, if it is not present on your computer. The following are the required software:
  - a. PSoC Creator 3.3 Component Pack 3 (CP3) or later: This software is available separately from [www.cypress.com/psoccreator](http://www.cypress.com/psoccreator).
  - b. PSoC Programmer 3.24.2 or later: This is installed as part of PSoC Creator installation or is available separately from [www.cypress.com/programmer](http://www.cypress.com/programmer).

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

Figure 2-2. Product Installation Overview



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

<Install\_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit

Default location:

Windows 7 (64-bit): C:\Program Files (x86)\Cypress\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit

Windows 7 (32-bit): C:\Program Files\Cypress\CY8CKIT-048 PSoC Analog Coprocessor 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 on page 36](#). 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; choose the product and select the **Uninstall/Change** button.

## 3. Kit Operation



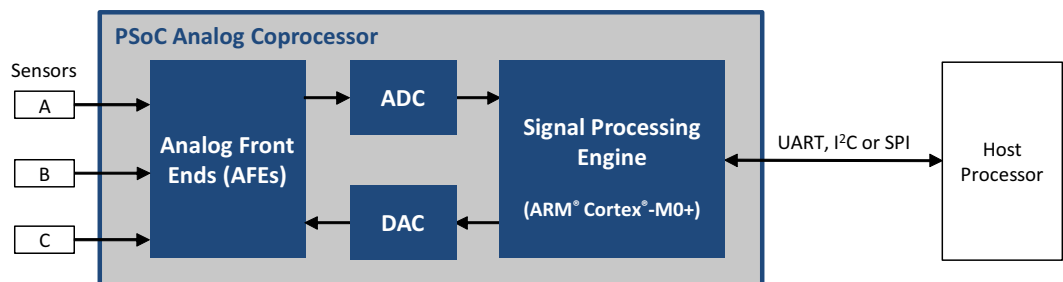
This chapter introduces you to the various features of the PSoC Analog Coprocessor Pioneer Kit, including the theory of operation and the onboard programming and debugging functionality.

### 3.1 Theory of Operation

Figure 3-1 shows a generic block diagram of a sensor-based system, which includes:

1. An AFE to condition the sensor outputs by amplifying and filtering the signals.
2. An analog-to-digital converter (ADC) or a comparator (not shown in below image) to convert the conditioned sensor output into digital data.
3. A programmable signal processing engine with a serial communication interface, to format the sensor data and send it to the host processor.

Figure 3-1. Sensor-Based System



The PSoC Analog Coprocessor Pioneer Kit consists of four blocks. They are PSoC Analog Coprocessor, analog sensors, KitProg2, and power supply.

The PSoC Analog Coprocessor block consists of the PSoC Analog Coprocessor device. Figure 3-2 shows the block diagram of the PSoC Analog Coprocessor block. See [PSoC Analog Coprocessor on page 20](#) for more details on the feature set.

The analog sensors block consists of five onboard sensors. They are PIR motion sensor, ambient light sensor, thermistor, inductive proximity sensor, and humidity sensor. The output of the analog sensors is fed to the PSoC Analog Coprocessor block. Figure 3-2 shows the block diagram of the sensors block. See [Analog Sensors on page 23](#) for more details on this block.

The KitProg2 block is used to program and debug the PSoC Analog Coprocessor through an onboard PSoC 5LP device. It connects to the USB port of a PC through a USB connector. It supports SWD programming/debugging and also includes a USB-I2C/USB-UART bridge interface for the PSoC Analog Coprocessor. Figure 3-3 shows the block diagram of the KitProg2 block. See [KitProg2 on page 31](#) for more details on this block.

The power supply block enables the board to operate at 1.8 V, 3.0 V, 3.3 V, and 5.0 V. It has the provision to be powered from three power sources: external power supply, USB, or a coin cell. The

multi-voltage operation is achieved using a Cypress Power Management IC (PMIC). The operating voltage can be selected using a power selection jumper (J9). [Figure 3-4](#) shows the block diagram of the power supply block. See [Power Supply](#) on [page 31](#) for more details on this block.

Figure 3-2. Block Diagram of PSoC Analog Coprocessor and Analog Sensors

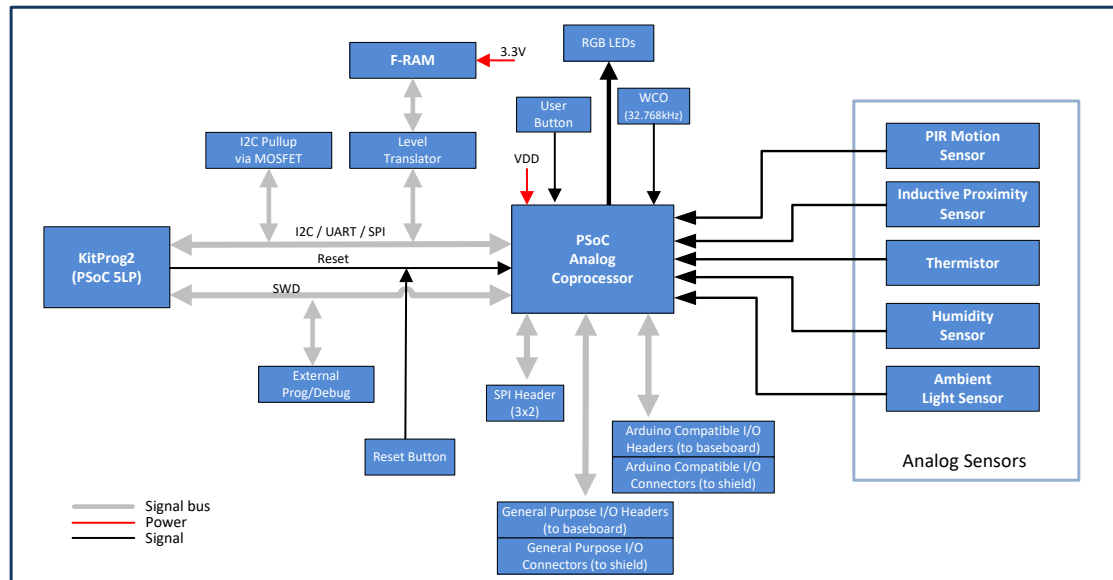


Figure 3-3. Block Diagram of KitProg2

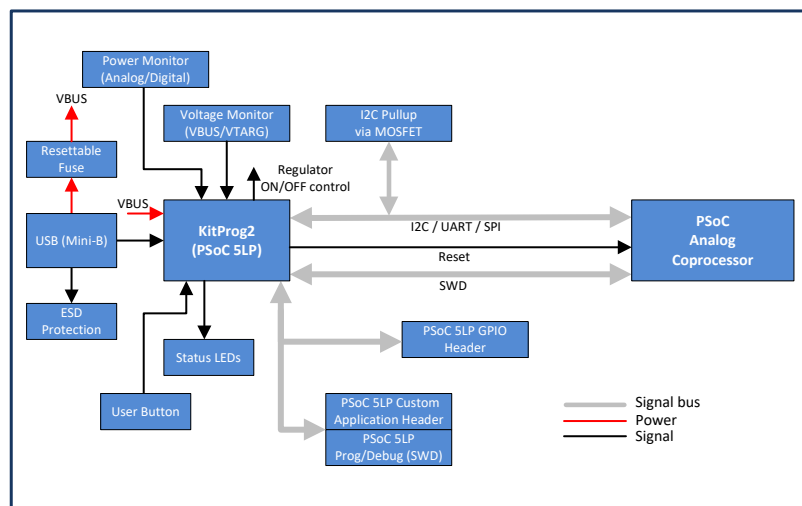
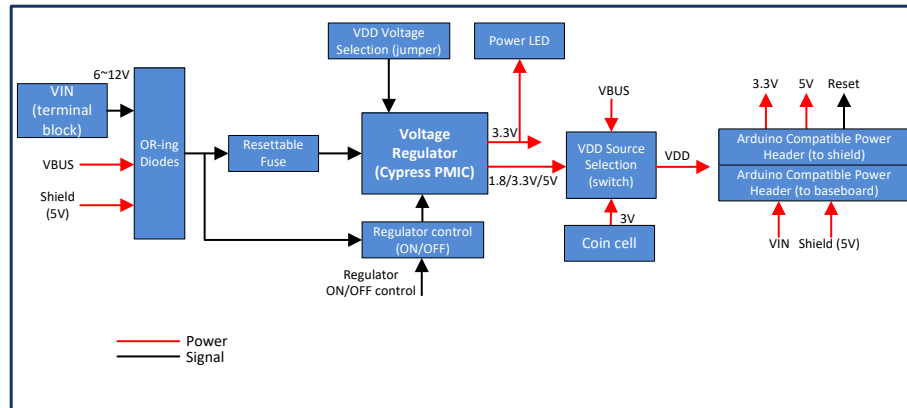


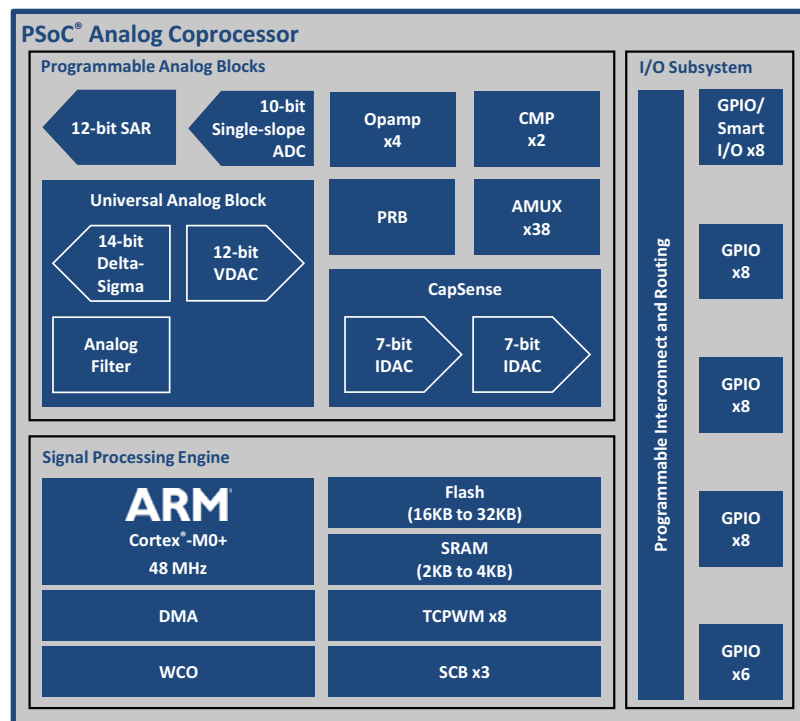
Figure 3-4. Block Diagram of Power supply



### 3.1.1 PSoC Analog Coprocessor

The PSoC Analog Coprocessor device has an extensive set of analog features and other resources.

Figure 3-5. PSoC Analog Coprocessor Block Diagram



Given below is a list of major features of the PSoC Analog Coprocessor device.

### Operating Range and Low-Power Modes

- Device operating voltage 1.71 V to 5.5 V
- Sleep mode switches off clocks to the CPU. 3.1 mA typical current at 12 MHz.
- Deep-Sleep mode with operational analog. 2.5  $\mu$ A typical current.

### Programmable Analog Blocks

- Universal Analog Block (UAB)
  - The UAB can be configured as one of the following:
    - 12-bit buffered voltage DAC (VDAC), with a sample rate of 500 kHz
    - 2<sup>nd</sup>-order bi-quad filter, as a low-pass, high-pass, band-pass, or notch filter
    - 12-bit delta-sigma ADC, with a sample rate of 7.8 ksps and a DNL of  $\pm 1$  LSB <sup>1</sup>
    - 14-bit incremental delta-sigma ADC, with a sample rate of 100 sps and a DNL of  $\pm 2$  LSB <sup>2</sup>
- Four programmable opamps
  - 90 dB open-loop gain, rail-to-rail operation
  - Can be used with external components to form standard opamp circuits
  - Can use an internal resistor array to form a programmable gain amplifier (PGA) with gain up to 32
  - 6 MHz gain-bandwidth when driving external I/Os, with up to 10-mA drive
  - 8 MHz gain-bandwidth when driving internal nodes such as the SAR ADC
  - $\pm 1$  mV input offset voltage
  - 15  $\mu$ A operating current in Deep-Sleep mode
- Two low-power comparators (CMP)
  - Wake up the device from low-power modes
- 12-bit SAR ADC
  - Sample rate up to 1 Msps
  - Selectable resolution 8-, 10-, or 12-bit
  - Automated hardware sequencer with 16 input channels
  - Each channel can be differential or single-ended
  - Integrated hardware averaging per channel
  - Programmable input channels, for example external pins, opamps, and UAB
- Single-Slope ADC
  - Selectable 8- or 10-bit resolution
  - Sample rate up to 11.6 ksps with 10-bit resolution
  - Input measurement range from  $V_{SS}$  to  $V_{DDA}$  on any GPIO pin
  - Implemented in the CapSense® block
- Programmable Reference Block (PRB)
  - Four voltage references, independently adjustable in 16 steps, from  $V_{DDA}$  to  $V_{SS}$ , or 1.2 V to  $V_{SS}$

1. Component support to be made available in future.

2. Component support to be made available in future.

- ❑ References can be routed to internal high-impedance analog resources: ADC, VDAC, comparator, and opamps.
- ❑ References can also be routed to a GPIO if buffered through an opamp.
- **CapSense**
  - ❑ Measures capacitance; can be used with capacitive sensors such as in liquid level or touch sensing applications
  - ❑ Self- and mutual-capacitive sensing methods
  - ❑ Improved electromagnetic interference (EMI) using spread spectrum clock and programmable slew rate control
- **IDACs**
  - ❑ Two 7-bit current DACs (IDACs) for use with CapSense or for general-purpose applications
  - ❑ A single 8-bit IDAC can be created by combining the two IDACs in parallel
  - ❑ 37.5 nA LSB current for precise capacitance measurements
  - ❑ Six output current ranges (4.76  $\mu$ A to 609  $\mu$ A) in source or sink configuration

### **32-bit Signal Processing Engine**

- Arm Cortex-M0+ CPU, operating at up to 48 MHz
- Up to 32 KB of flash with read accelerator
- Up to 4 KB of SRAM
- 8-channel direct memory access (DMA) controller
- Watch crystal oscillator (WCO) for real-time clock (RTC) applications
- Three serial communication blocks (SCBs), each configurable as SPI, I<sup>2</sup>C, or UART
- Eight 16-bit timer / counter / pulse-width modulator (TCPWM) blocks

### **I/O Subsystem**

- Up to 38 GPIOs that can be used for analog, digital, CapSense, or segment LCD functions
- Programmable drive modes and slew rates
- Eight Smart I/Os that can implement Boolean operations on pin signals



### 3.1.2 Analog Sensors

Analog sensors generally come in five different types, depending on their output electrical signal: voltage, current, resistance, capacitance, or inductance. Each sensor type requires a specific AFE design as shown in [Table 3-1](#). The PSoC Analog Coprocessor provides resources to build all of these AFEs, thus reducing the BOM cost and PCB size. The PSoC Analog Coprocessor contains opamp, comparators, programmable gain amplifiers (PGA), programmable reference block (PRB) and analog multiplexers (AMUX), which are the building blocks of different type of AFEs.

The remaining part of this section contains details of interfacing the PSoC Analog Coprocessor with each of the sensors mentioned in [Table 3-1](#). For example, [Interfacing with a PIR Motion Sensor on page 23](#) explains how to use opamps inside the PSoC Analog Coprocessor to implement a voltage amplifier circuit.

Table 3-1. AFE Requirements for Different Sensors

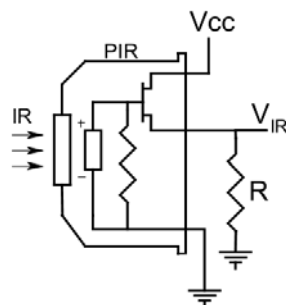
Analog Sensor	Output Signal	AFE	Sensor Part
PIR motion sensor (Refer to <a href="#">Interfacing with a PIR Motion Sensor on page 23</a> )	Voltage	Voltage amplifier	Zilog ZRE200GE
Ambient light sensor (Refer to <a href="#">Interfacing with an Ambient Light Sensor on page 26</a> )	Current	Transimpedance amplifier	Vishay TEMD6200FX01
Thermistor (Refer to <a href="#">Interfacing with a Thermistor on page 27</a> )	Resistance	Programmable voltage reference	EPCOS (TDK) B57164K103J
Inductive proximity sensor (Refer to <a href="#">Interfacing with an Inductive Proximity Sensor on page 28</a> )	Inductance	HPF-Down Mixer-Rectifier	N/A (PCB trace)
Humidity sensor (Refer to <a href="#">Interfacing with a Humidity Sensor on page 29</a> )	Capacitance	Charge transfer circuit	HPP801A031

#### 3.1.2.1 Interfacing with a PIR Motion Sensor

The PIR motion sensor is based on the pyroelectric effect, where certain materials generate a voltage when exposed to infrared radiation. This radiation is the portion of the electromagnetic spectrum that falls between microwaves and visible light. Infrared radiation has wavelengths longer than the visible light but shorter than microwaves. Humans at normal body temperature radiate strongest in the infrared range at an approximate wavelength of 10  $\mu\text{m}$ .

The PIR motion sensor uses infrared-sensitive materials as the sensing elements. It is packaged with a field-effect transistor (FET) in the Source Follower mode, as [Figure 3-6](#) shows. An FET is required to buffer the high-impedance output of the sensor element. When the sensor element is exposed to infrared radiation, a voltage is generated across the element.

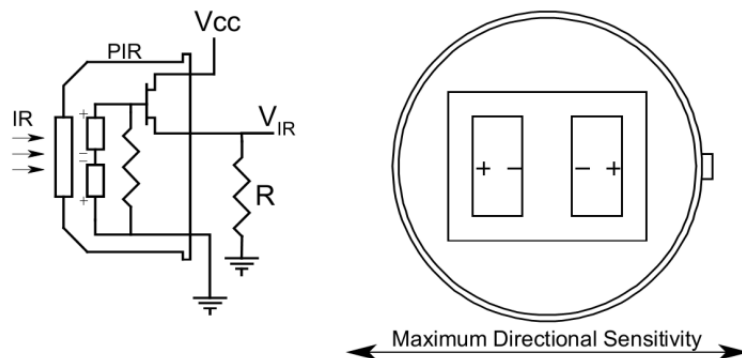
Figure 3-6. PIR Motion Sensor – Single-Element



Most of the common PIR motion sensors have two or four sensing elements. These elements are arranged such that the voltage generated by one is subtracted by the other. This arrangement cancels the common signal and generates a voltage only when there is a difference in the incident infrared radiation level on the sensing elements.

Figure 3-7 shows the dual-element PIR motion sensor with the elements connected in series but with an opposite phase, because of which it has the maximum sensitivity along a particular axis.

Figure 3-7. PIR Motion Sensor Dual-Element



The sensor package is designed to have a unique field-of-view for each element. When an IR radiating source moves across the fields of view, the sensor generates a differential signal (see Figure 3-8). For a 90° field-of-view or more, a Fresnel lens is mounted on the PIR motion sensor. It improves the sensitivity and thus the detection distance.

Figure 3-8. PIR Motion Sensor Output Response

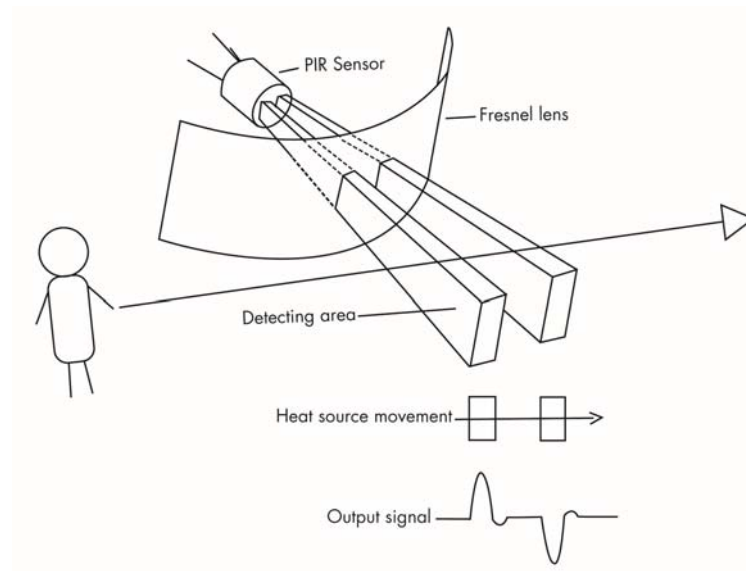
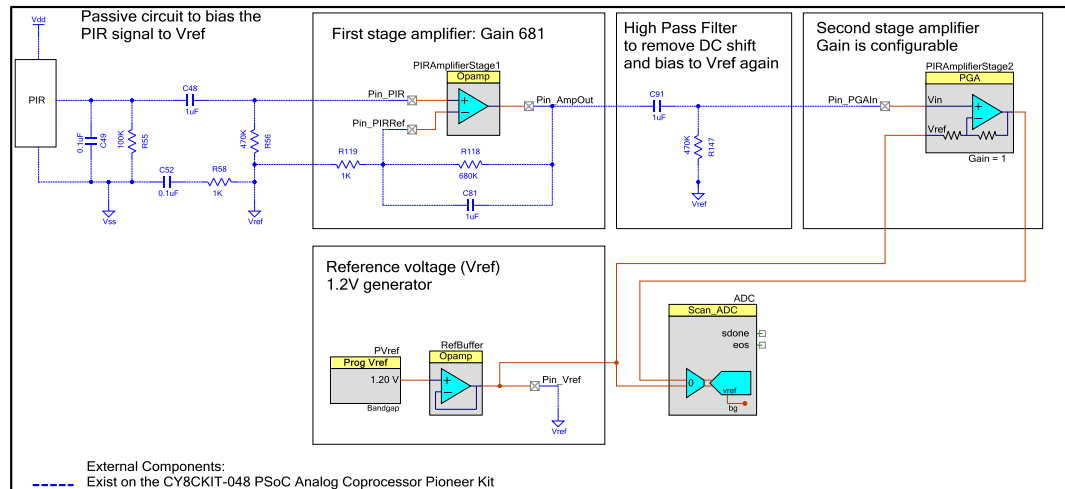


Figure 3-9 shows the PSoC Creator schematic for interfacing a PIR motion sensor with the PSoC Analog Coprocessor.

Figure 3-9. Interfacing PSoC Analog Coprocessor with PIR Motion Sensor



The PIR motion sensor implementation on the PSoC Analog Coprocessor Pioneer Kit consists of five stages: a bias circuit for the PIR motion sensor, a first-stage amplifier, a high-pass filter (HPF), a second-stage amplifier, and an ADC. In the main stages of the PIR motion sensor implementation, the two amplifiers are implemented in the PSoC Analog Coprocessor.

The PSoC Analog Coprocessor Pioneer Kit has a dual-element PIR motion sensor (ZRE200GE). The voltage signal generated by the sensor is AC-coupled (using C48) and clamped to the internal reference voltage generated from PVref Component. The typical sensor output voltage is in the order of several millivolts and varies depending on the strength of the incident infrared radiation. To detect the motion of a human body at a distance of 10 feet, a gain of >1000 is required. A single-stage amplifier with such a high gain causes the amplifier output to saturate due to the amplification of the input offset voltage. Thus, a two-stage amplifier is best suited for amplifying with a high gain.

The total gain is split between two stages. The first-stage amplifier uses a noninverting amplifier configuration using an internal opamp and external gain setting resistors, R118 and R119. The second-stage amplifier uses a PGA Component. The first-stage amplifier gain is set to 681 and the PGA gain is set to 1 on startup. However, the second-stage amplifier gain changes depending on the detection distance required – 3 feet, 10 feet, or 20 feet.

An HPF, made using external passive components C91 and R147, is introduced between the first and the second amplifier stages to eliminate the offset voltage.

The PIR motion sensor and gain stages use a 1.2-V system bandgap voltage as the reference voltage. The bandgap voltage is independent of supply voltage fluctuations and hence provides a stable voltage reference. This voltage is generated using a programmable voltage reference component, PVref, and is buffered using an opamp.

The output of the second-stage PGA is connected to the Scanning SAR ADC Component. The Scanning SAR ADC results are compared against threshold values to detect the motion of an IR emitting object.

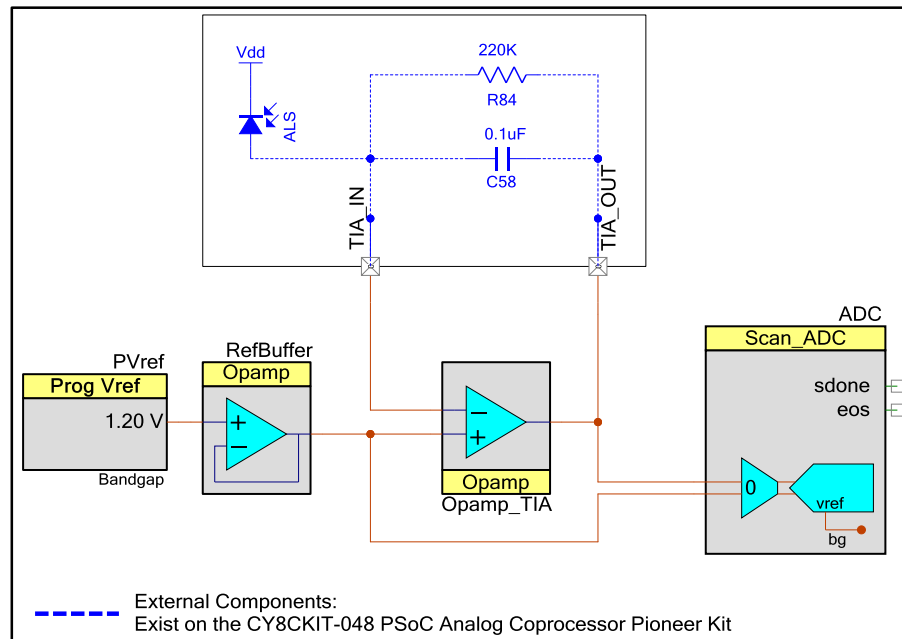
For more details of firmware implementation, refer to the Code Example CE211301\_PIR\_Motion\_Sensing available in the kit installation directory.

### 3.1.2.2 Interfacing with an Ambient Light Sensor

An ambient light sensor (ALS) is a current sensor, which gives different current outputs based on the intensity of incident light on the sensor. The change in the current output is measured to detect the ambient light intensity.

Figure 3-10 shows PSoC Creator schematic to interface an ALS to the PSoC Analog Coprocessor.

Figure 3-10. Interfacing PSoC Analog Coprocessor with Ambient Light Sensor



The current output from the ALS is converted to a voltage signal using a transimpedance amplifier (TIA). The TIA is built using one of the opamps in the PSoC Analog Coprocessor as well as external passive components (R84, C58).

The reference voltage for the TIA is set to the 1.2 V bandgap reference. The bandgap voltage is independent of supply voltage fluctuations and hence provides a stable voltage reference. This voltage is generated using the programmable reference Component, PVref and is buffered using an opamp.

The output of the TIA is measured using the 12-bit Scanning SAR ADC Component with the positive input connected to the TIA output and the negative input connected to the reference voltage of the TIA. The Scanning SAR ADC Component is configured in the differential mode with its reference ( $V_{ref}$ ) connected to the bandgap voltage. This gives the ADC measurement range as  $V_n \pm V_{ref}$ , where  $V_n$  is the negative input voltage of Scanning SAR ADC. The CPU takes the output of Scanning SAR ADC, and calculates the photodiode current and then the light illuminance value.

The PSoC Analog Coprocessor Pioneer Kit has an ambient light sensor (TEMD6200FX01) on the board connected to the PSoC Analog Coprocessor.

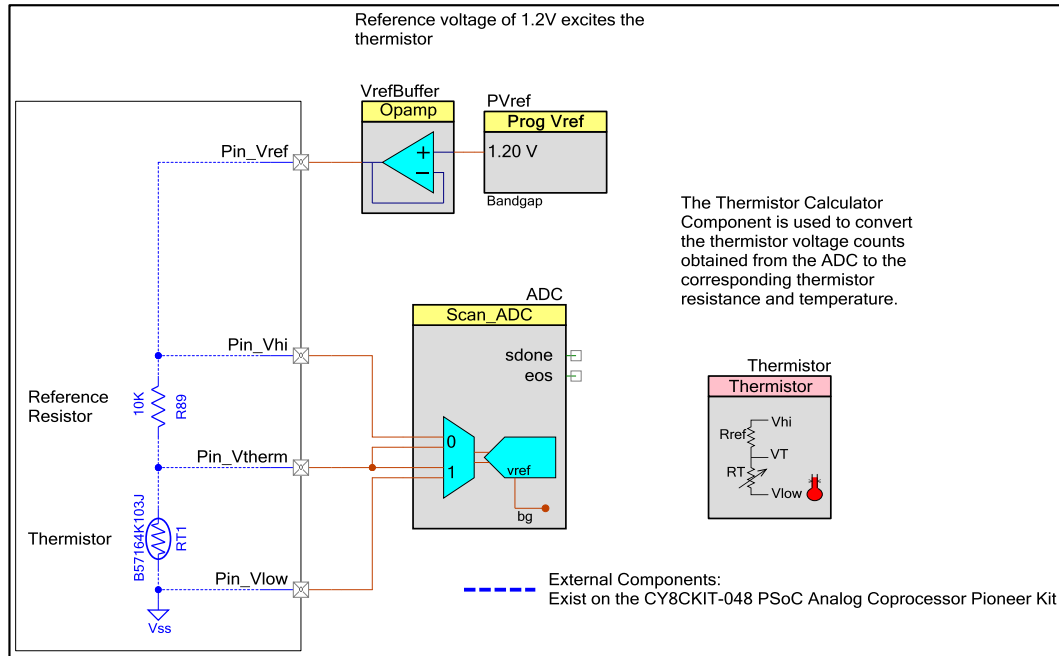
For details of firmware implementation, refer to the Code Example *CE211252\_Ambient\_Light\_Sensing* available in the kit installation directory.

### 3.1.2.3 Interfacing with a Thermistor

Temperature can be calculated by measuring the thermistor resistance. The PSoC Creator Thermistor Calculator Component simplifies the math-intensive resistance-to-temperature conversion.

Figure 3-11 shows the PSoC Creator schematic for interfacing a thermistor with the PSoC Analog Coprocessor.

Figure 3-11. Interfacing PSoC Analog Coprocessor with a Thermistor



The PSoC Analog Coprocessor Pioneer Kit has a thermistor RT1 (B57164K103J). A 10-kΩ reference resistor, R89 ( $R_{ref}$ ), is connected in series with the thermistor.

The thermistor and  $R_{ref}$  are excited using a 1.2-V bandgap voltage used as a reference voltage. The bandgap voltage is independent of supply voltage fluctuations and hence provides a stable voltage reference. This voltage is generated using the programmable reference Component, PVref, and is buffered using an opamp. The same bandgap voltage is used as the ADC reference. This enables measurement in the full-scale range of the ADC and results in increased resolution of voltage measurement.

Three voltage signals ( $V_{therm}$ ,  $V_{low}$ , and  $V_{hi}$ ) from the resistor divider are connected to the two differential channels of the Scanning SAR ADC Component. The thermistor ( $R_T$ ) resistance is calculated from the ADC count using the following equation:

$$R_T = R_{ref} \times \left( \frac{V_{therm} - V_{low}}{V_{hi} - V_{therm}} \right)$$

The temperature value is then derived by passing the measured resistance to the Thermistor Calculator component. The Thermistor Calculator component uses a lookup table method to calculate the temperature with a resolution of 0.1 °C.

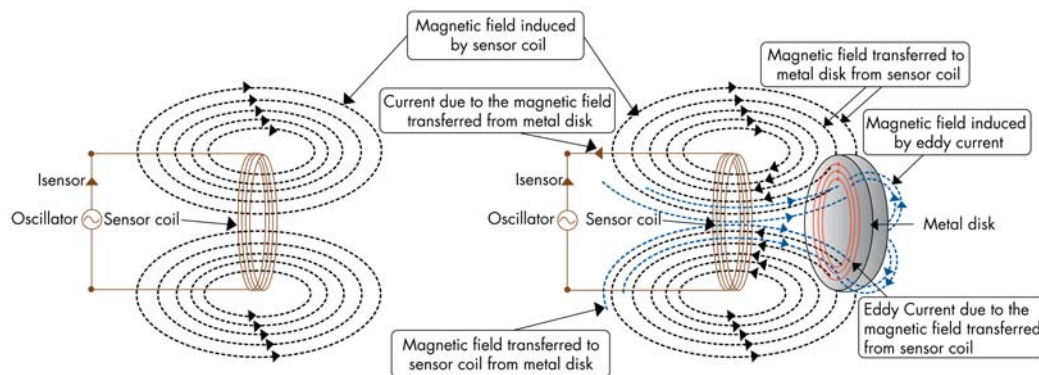
For details of firmware implementation, refer to the Code Example CE211321\_Temperature\_Sensing available in the kit installation directory.

### 3.1.2.4 Interfacing with an Inductive Proximity Sensor

Inductive proximity sensing works on the principle of electromagnetic coupling between a sensor coil and the metal disk to be detected. When the metal disk enters the electromagnetic field induced by a sensor coil, some of the electromagnetic energy is transferred into the metal disk as shown in Figure 3-12. This transferred energy causes a circulating electrical current called eddy current. The eddy current flowing in the metal disk induces a reverse electromagnetic field on the sensor coil, which results in a reduction of effective inductance of the sensor coil.

The sensor coil is placed in parallel with a capacitor to form a tank oscillator circuit. The reduction in the sensor coil inductance causes a shift in the resonant frequency of the tank oscillator circuit. This shift in resonant frequency changes the amplitude of the signal across the sensor coil. The change in the amplitude of the sensor coil signal is measured by the PSoC Analog Coprocessor to detect the presence of the metal disk.

Figure 3-12. Field Coupling between Sensor and Metal



The inductive proximity sensor implementation on the CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit consists of six stages: a sensor excitation circuit, a tank circuit, a high-pass filter (HPF), a down mixer, a rectifier and an ADC. The main stages of the inductive proximity sensing implementation – PWM of sensor excitation circuit, down mixer, rectifier, and ADC – are implemented in the PSoC Analog Coprocessor. Figure 3-13 shows the PSoC Creator schematic for inductive proximity sensing.

The inductive proximity sensor on the PSoC Analog Coprocessor Pioneer Kit is an onboard coil implemented using PCB trace; its inductance is 3  $\mu\text{H}$  (L).

The inductive proximity sensor (L) in parallel with a capacitor (C80) forms a tank circuit as shown in Figure 3-13. The tank circuit is excited using a PWM. This PWM is called Excitation PWM in Figure 3-13. The resistor (R77) controls the excitation current of the inductive proximity sensor. The resonant frequency ( $f_r$ ) of the tank circuit is determined by the following equation.

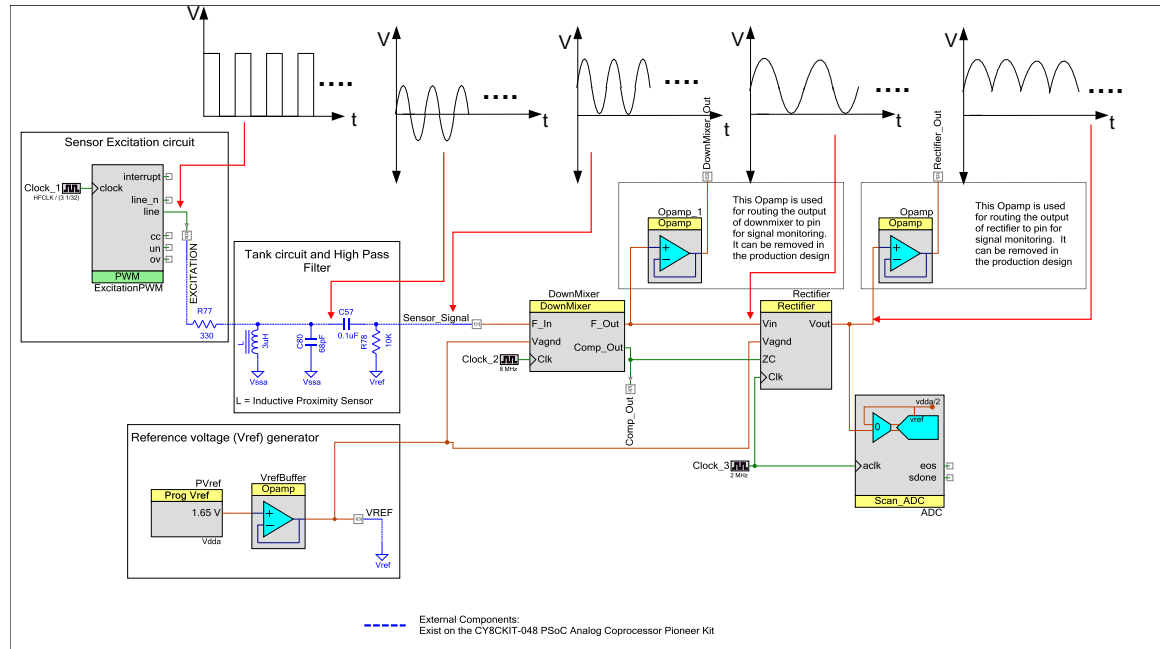
$$f_r = \frac{1}{(2\pi * \sqrt{LC})}$$

The HPF formed by the capacitor (C57) and resistor (R78) prepares the signal for the next stage. It removes the offset voltage and biases the filtered signal to a known reference voltage generated from PVref Component as shown in Figure 3-13.

The signal across resistor R78 is measured by passing it through a down mixer and a rectifier that are implemented using Universal Analog Block (UAB). The output of rectifier is connected to an ADC that is operating on synchronous clock as shown in Figure 3-13. The ADC is set to operate in

averaging mode so that ADC count reflects the DC content of rectifier output. The ADC output count is compared against the threshold value to detect the presence of the metal disk. The threshold value is selected such that it is less than the peak noise observed on the ADC count to avoid false triggers.

Figure 3-13. Interfacing PSoC Analog Coprocessor with Inductive Proximity Sensor



For details of firmware implementation, refer to the Code Example *CE211252\_Inductive\_Proximity\_Sensing* available in the kit installation directory.

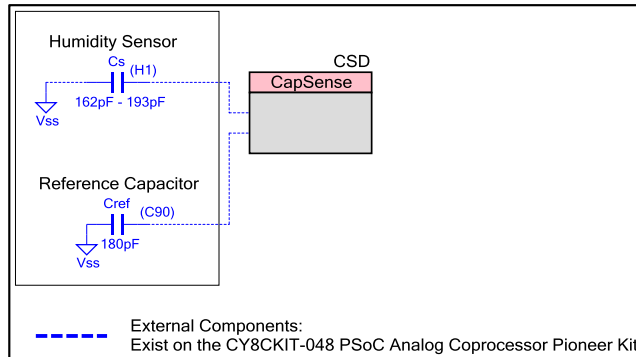
### 3.1.2.5 Interfacing with a Humidity Sensor

Humidity can be measured by measuring the capacitance of a capacitive humidity sensor. A Cypress CapSense Component is used to measure the capacitance of the humidity sensor. The CapSense Component provides the best-in-class signal-to-noise ratio (SNR) (>5:1) for sensors with capacitance up to 200 pF. It also provides high-performance sensing across several varying environmental factors such as temperature and humidity.



Figure 3-14 shows the PSoC Creator schematic for interfacing a PSoC Analog Coprocessor with a humidity sensor.

Figure 3-14. Interfacing PSoC Analog Coprocessor with a Humidity Sensor



The PSoC Analog Coprocessor Pioneer Kit has a humidity sensor (HPP801A031) and a reference capacitor ( $C_{ref}$ ) of 180 pF. The reference capacitor,  $C_{ref}$ , is used to calibrate the capacitance measurement. The capacitance of the humidity sensor is calculated using the following equation:

$$C_s = \frac{C_{ref} * (RawCount_{Cs} - RawCount_{cos})}{RawCount_{C_{ref}} - RawCount_{cos}} + C_{OS}$$

Where,  $C_s$  is the capacitance of the humidity sensor

$C_{ref}$  is the capacitance of the reference capacitor

$RawCount_{Cs}$  is the raw count obtained from the CapSense Component when the humidity sensor is being measured.

$RawCount_{C_{ref}}$  is the raw count obtained from the CapSense Component when the reference capacitor is being measured.

$RawCount_{cos}$  is the raw count that corresponds to the offset capacitance (trace capacitance with sensor disconnected).

$C_{OS}$  is the offset capacitance (trace capacitance with sensor disconnected from the pin).

Humidity is calculated from the measured capacitance  $C_s$  using the following equation:

$$\%RH = \frac{C_s - C_{nom}}{Sensitivity} + \%RH_{nom}$$

Where,

$C_s$  is the capacitance of the humidity sensor

$C_{nom}$  is the nominal capacitance of the humidity sensor

Sensitivity is the sensitivity of the humidity sensor (pF/%RH)

$\%RH_{nom}$  is the nominal humidity value

Note that  $C_{nom}$ , sensitivity, and  $\%RH_{nom}$  are specific to the sensors and are provided by sensor manufacturer. For the humidity sensor (HPP801A031) used on PSoC Analog Coprocessor Pioneer Kit, the  $C_{nom}$ ,  $\%RH_{nom}$ , and sensitivity are 180 pF, 55%RH and 0.31 pF/%RH respectively.



The capacitance of the humidity sensor used in the code example [CE211322](#) varies from 162 pF to 193 pF for 1% to 100% relative humidity (RH). The calibration capacitor is selected closer to the upper range of capacitance measurement to reduce the gain error in measurement.

For details of firmware implementation, refer to the Code Example *CE211322\_Humidity\_Sensing* available in the kit installation directory.

### 3.1.3 KitProg2

The KitProg2 block consists of an onboard PSoC 5LP (CY8C5868LTI-LP039) device, which is used to program and debug the PSoC Analog Coprocessor. The PSoC 5LP device connects to the USB port of a PC through a USB connector, and to the SWD and other communication interfaces of the PSoC Analog Coprocessor. PSoC 5LP is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C58LPxx family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals. For more information, refer to the [KitProg2 User Guide](#), visit the [PSoC 5LP web page](#) and refer to the [CY8C58LPxx Family Datasheet](#).

### 3.1.4 Power Supply

The power supply system is designed to support 1.8-V, 3.3-V, or 5-V operation when supplied from the external VIN or USB connector. The selection between 1.8 V, 3.3 V, and 5 V is made through a 4-pin jumper J9. A slider switch SW4 is used to select the power supply from the voltage regulator (Cypress PMIC - MB39C011APFT-G-BND-ERE1), USB or a 3-V coin cell. [Table 3-2](#) lists the jumper J9 settings for different voltages.

If the kit is used as a shield, it can be powered from the 5-V supply of the Arduino-compatible baseboard. [Figure A-3](#) shows the schematics of the power regulator circuit.

Table 3-2. Jumper J9 and switch SW4 Settings for Different Voltage Selections

Power Supply Source	VDD (in Volts)	SW4 (Position)	J9 (Jumper Position)
USB	1.8	REG	open
	3.3	REG	1–2
	5.0	USB	Any position except 2–3
	1.8–3.3	REG	4–2
External VIN	1.8	REG	open
	3.3	REG	1–2
	5.0	REG	2–3
Arduino baseboard	1.8	REG	open
	3.3	REG	1–2
Coin Cell	3.0	BAT	NA

## 3.2 Flexible Prototyping

PSoC Analog Coprocessor Pioneer Kit is footprint-compatible with Arduino Uno R3 shields and baseboards. It can interface with shields, or be used as a shield for any host processor board. It includes onboard KitProg2 programmer and debugger with Arm CMSIS-DAP.

All the analog sensors on the PSoC Analog Coprocessor Pioneer Kit can be disconnected from the PSoC device or the header by removing the  $0\Omega$  resistors. Figure 3-15 shows the I/Os of the PSoC Analog Coprocessor connected to both the sensors as well as the headers through the  $0\Omega$  resistors. Table 3-3 lists all the  $0\Omega$  resistors used to connect PSoC pins to sensors and the headers. If a PSoC pin is required to be used from the header for some other purpose than the onboard sensor, then the pin can be disconnected from the sensor by removing the corresponding  $0\Omega$  resistor.

Figure 3-15. Connection Between Sensors, Port Pins, and Headers

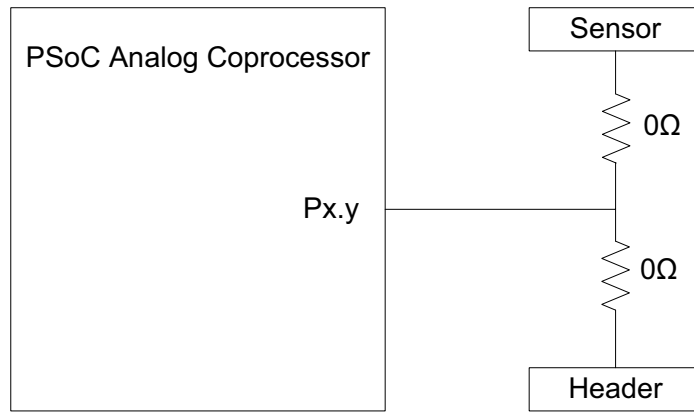


Table 3-3. List of  $0\Omega$  Resistors Connecting PSoC Pins to Sensors and Headers

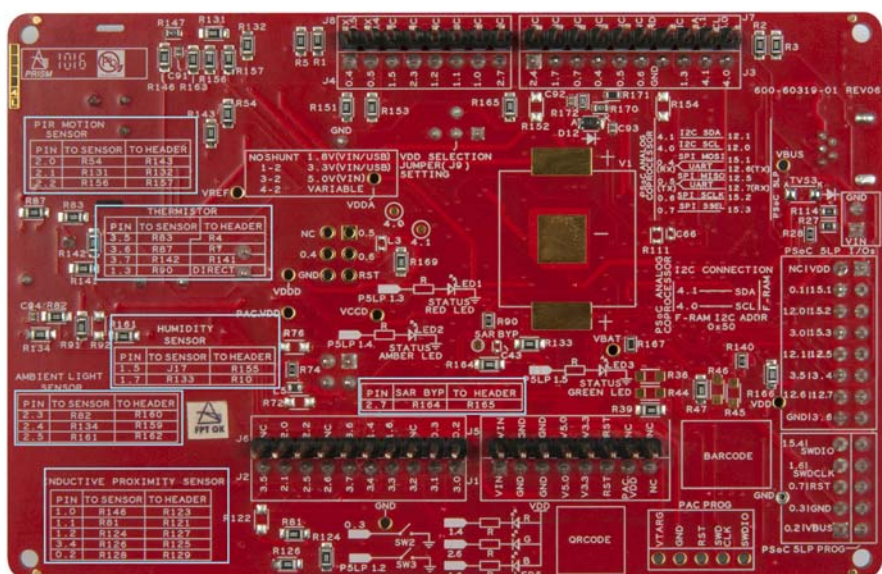
Pin	Sensor	Where on Sensor Circuit	$0\Omega$ for sensor	Header	$0\Omega$ for header
P2[0]	PIR motion sensor	HPF at output of PIR sensor	R54	J6.9	R143
P2[1]	PIR motion sensor	Feedback resistor for first-stage Amplifier	R131	J2.9	R132
P2[2]	PIR motion sensor	HPF between first-stage and second-stage amplifier	R156	J6.8	R157
P2[3]	Ambient light sensor	Feedback resistor of TIA	R82	J4.4	R160
P2[4]	Ambient light sensor	Photo Diode Anode	R134	J3.1	R159
P2[5]	Ambient light sensor	VREF	R161	J2.8	R162
P3[5]	Thermistor	Junction of thermistor and reference resistor	R83	J2.10	R4
P3[6]	Thermistor	VSSA	R87	J6.6	R7
P3[7]	Thermistor	Reference resistor	R142	J2.6	R141
P1[3]	Thermistor	VREF	R90	J3.8	N/A (Direct)
P1[0]	Inductive proximity sensor	VREF	R146	J4.7	R123
P1[1]	Inductive proximity sensor	HPF	R81	J4.6	R121
P1[2]	Inductive proximity sensor	Input of LPF	R124	J4.5	R127

Table 3-3. List of 0Ω Resistors Connecting PSoC Pins to Sensors and Headers (*continued*)

Pin	Sensor	Where on Sensor Circuit	0Ω for sensor	Header	0Ω for header
P3[4]	Inductive proximity sensor	Output of LPF	R126	J2.5	R125
P0[2]	Inductive proximity sensor	Series resistor of Inductive sensor	R128	J6.1	R129
P1[5]	Humidity sensor	Connector J17	N/A (J17)	J4.3	R155
P1[7]	Humidity sensor	Reference capacitor	R133	J3.2	R10

Tables to identify the 0Ω resistors connected to each of the sensors and the headers are also provided on the bottom side of the PSoC Analog Coprocessor Pioneer board, as shown in Figure 3-16.

Figure 3-16. Sensor and Header Pin Connections



### 3.3 KitProg2 Functions

The PSoC Analog Coprocessor Pioneer Kit can be programmed and debugged using the onboard KitProg2. KitProg2 is a multifunctional system, which includes a programmer, debugger, USB-I2C bridge, and a USB-UART bridge. A Cypress PSoC 5LP device is used to implement the KitProg2 functionality. KitProg2 is integrated in most PSoC development kits. For more details on the KitProg2 functionality, refer to the *KitProg2 User Guide* available in the kit installation directory: <Install\_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit\<version>\Documentation\KitProg2\_User\_Guide.pdf.

Before programming the device, ensure that PSoC Creator and PSoC Programmer software are installed on the computer. See [Install Software on page 15](#) for more information.

#### 3.3.1 Programming and Debugging Using PSoC Creator

1. Connect the PSoC Analog Coprocessor Pioneer Kit to the PC using the USB cable, as shown in [Figure 3-17](#). You can proceed to the next step if you see the amber status LED turned ON, the green and red status LEDs turned OFF, and the power LED is turned ON. If you do not see the desired LED status, see the [KitProg2 User Guide](#) for details on the KitProg2 status and troubleshooting instructions.

Figure 3-17. Connect USB Cable to USB Connector on the Kit



2. Open the desired project in PSoC Creator. For this, go to **File > Open > Project/Workspace**. This provides the option to browse and open your saved project.
3. Build the project by selecting **Build > Build Project** or pressing **[Shift] [F6]**.
4. After the project is built without errors, select **Debug > Program** or **[Ctrl] [F5]** to program the device.

PSoC Creator has an integrated debugger. You can start the debugger by selecting **Debug > Debug** or by pressing **[F5]**. Refer to the **Debugging Using PSoC Creator** section in the [KitProg2 User Guide](#) for a detailed explanation on how to debug using PSoC Creator.

### 3.3.2 Programming Using PSoC Programmer

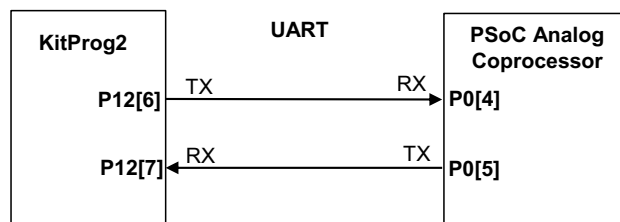
PSoC Programmer (3.24.2 or later) can be used to program the existing `.hex` files into the PSoC Analog Coprocessor Pioneer Kit. Refer to the **Programming Using PSoC Programmer** section in the [KitProg2 User Guide](#) for a detailed explanation on how to program using PSoC Programmer.

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

### 3.3.3 USB-UART Bridge

KitProg2 on the PSoC Analog Coprocessor Pioneer Kit can act as a USB-UART bridge. UART lines between PSoC Analog Coprocessor and KitProg2 are hard-wired on the board, with UART\_RX assigned to P0[5] and UART\_TX assigned to P0[4] of the PSoC Analog Coprocessor. For more details on the KitProg2 USB-UART functionality, see the [KitProg2 User Guide](#).

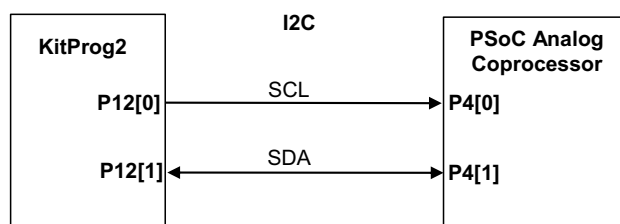
Figure 3-18. UART Connection Between KitProg2 and PSoC Analog Coprocessor



### 3.3.4 USB-I2C Bridge

The KitProg2 can function as a USB-I2C bridge and communicate with the Bridge Control Panel (BCP) software. The I2C lines on the PSoC Analog Coprocessor are P4[1] (SDA) and P4[0] (SCL), which are hard-wired on the board to the I2C lines of KitProg2. USB-I2C supports I2C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg2 USB-I2C functionality, see the [KitProg2 User Guide](#).

Figure 3-19. I2C Connection Between KitProg2 and PSoC Analog Coprocessor



## 4. Code Examples



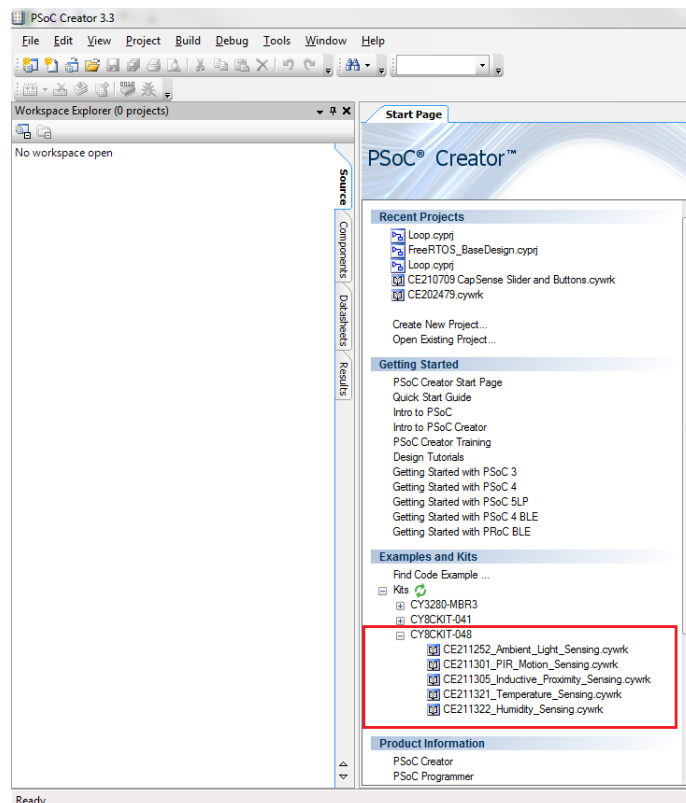
The PSoC Analog Coprocessor Pioneer Kit includes five code examples. To access these code examples, download and install the PSoC Analog Coprocessor Pioneer Kit setup file from [www.cypress.com/CY8CKIT-048](http://www.cypress.com/CY8CKIT-048). After installation, the code examples will be available from **Examples and Kits > Kits** under PSoC Creator Start Page. For more code examples, visit the [PSoC 3, PSoC 4, and PSoC 5LP code examples](#) page, which lists all PSoC Creator code examples available across application notes, kits, and PSoC Creator.

### 4.1 Using the Kit Code Example

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, expand **CY8CKIT-048** under **Examples and Kits > 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**.

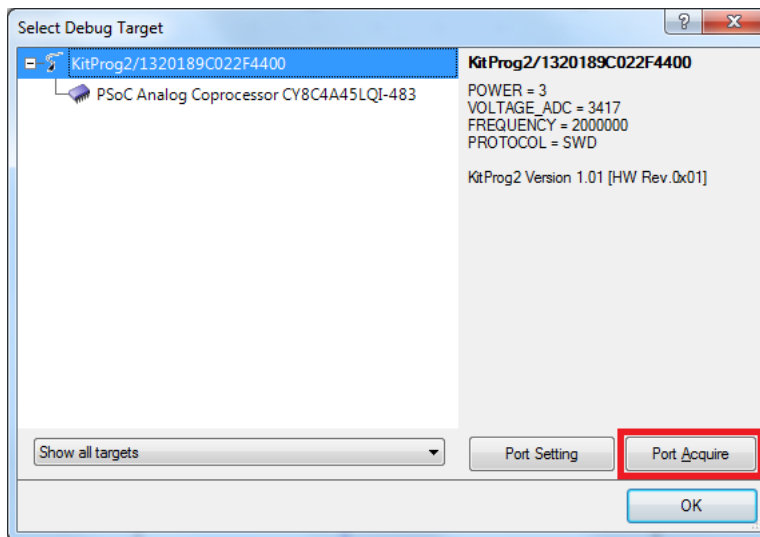
Figure 4-1. Open Code Example from PSoC Creator





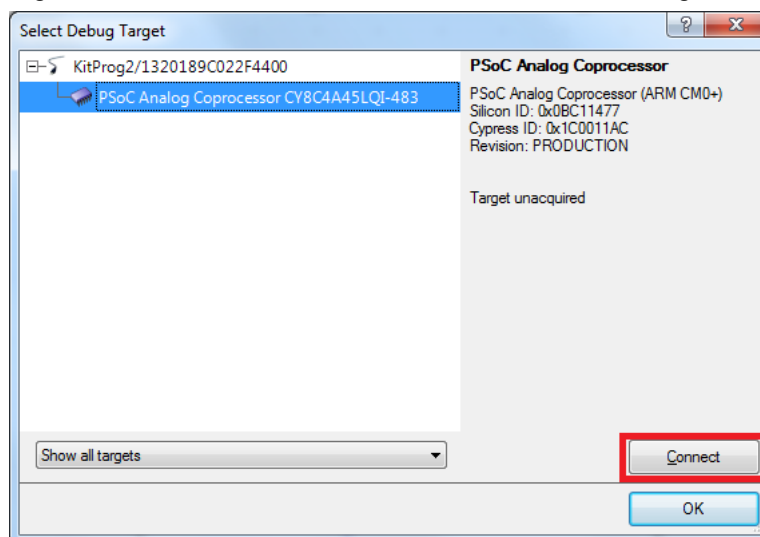
4. Build the code example by choosing **Build > Build <Project Name>**. After the build process is successful, a .hex file is generated.
5. To program the kit with the code example, connect the PSoC Analog Coprocessor to PC using USB cable as shown in [Figure 3-17](#).
6. Choose **Debug > Program** in PSoC Creator.
7. If the device is already acquired, programming will complete automatically – the result will appear in the PSoC Creator status bar at the bottom left side of the screen. If the device is yet to be acquired, the Select Debug Target window will appear. Select **KitProg2/<serial number>** and click the **Port Acquire** button, as shown in [Figure 4-2](#).

Figure 4-2. Port Acquire



8. After the device is acquired, it is shown in a tree structure below **KitProg2/<serial number>**. Click **Connect** as shown in [Figure 4-3](#) and then click **OK** to exit the window and start programming.

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



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

## 4.2 Code Examples

Table 4-1 shows a list of code examples that can be used with this kit.

Table 4-1. Code Examples

#	Project	Title/Description
1	<a href="#">CE211301</a> PIR Motion Sensing	This code example demonstrates how to measure the voltage signal from a PIR sensor to detect the movement of an IR emitting object. The sensor data is communicated over I2C. The RGB LED is also turned ON once motion is detected.
2	<a href="#">CE211252</a> Ambient Light Sensing	This code example demonstrates how to measure the current output from the ALS sensor and calculate the ambient light illuminance. The measured current and the calculated light illuminance are sent over I2C. The light illuminance is used to control the RGB LED light intensity.
3	<a href="#">CE211321</a> Temperature Sensing	This code example demonstrates how to measure the thermistor resistance and calculate the temperature. The measured thermistor resistance and the calculated temperature are sent over I2C. The RGB LED is controlled based on the calculated temperature value.
4	<a href="#">CE211305</a> Inductive Proximity Sensing	This code example demonstrates how to interface PSoC Analog Coprocessor with an inductive proximity sensor. The code example measures the change in onboard inductance to detect the presence of metal in close proximity to an onboard coil. The measured sensor data is sent over I2C. The brightness of the RGB LED is varied based on the proximity distance between the sensor and metal.
5	<a href="#">CE211322</a> Humidity Sensing	This code example demonstrates how to interface PSoC Analog Coprocessor with a capacitive humidity sensor. The code example measures the output capacitance of the humidity sensor and calculates the value of humidity from the measured capacitance. The measured capacitance and calculated humidity value are sent over I2C. The intensity of the RGB LED is varied with the calculated humidity value.



# A. Appendix



## A.1 Schematics

Refer to the schematics file in the following path:

```
<Install_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit\1.0\  
Hardware\CY8CKIT-048 Schematic.pdf
```

## A.2 Hardware Functional Description

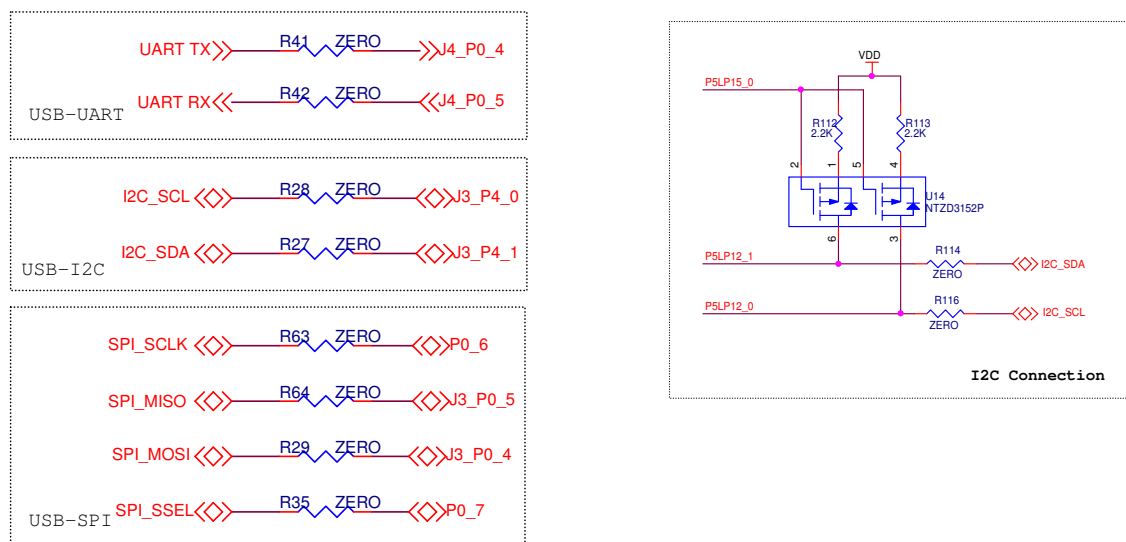
This section provides a detailed explanation of individual hardware blocks of the PSoC Analog Coprocessor Pioneer Kit. The details of PSoC Analog Coprocessor and KitProg2 are provided in “PSoC Analog Coprocessor” on page 20 and “KitProg2” on page 31 respectively.

### A.2.1 Serial Interconnection Between PSoC 5LP and PSoC Analog Coprocessor

In addition to the use as an onboard programmer, the PSoC 5LP device is used as a USB-Serial interface for the USB-UART bridge and the USB-I2C bridge, as shown in Figure A-1. The I2C bus contains firmware-controlled resistive pull-ups using FETs, which can be enabled or disabled using PSoC 5LP pins. The USB-Serial pins of the PSoC 5LP device are also available on the Arduino-compatible header; therefore, the PSoC 5LP device can be used to control Arduino shields with an SPI/I2C/UART interface.

**Note:** KitProg2 firmware does not support the USB-SPI bridge functionality at present. SPI pins of PSoC Analog Coprocessor are connected to the PSoC 5LP device for enabling SPI communication between PSoC 5LP and PSoC Analog Coprocessor for custom PSoC 5LP applications.

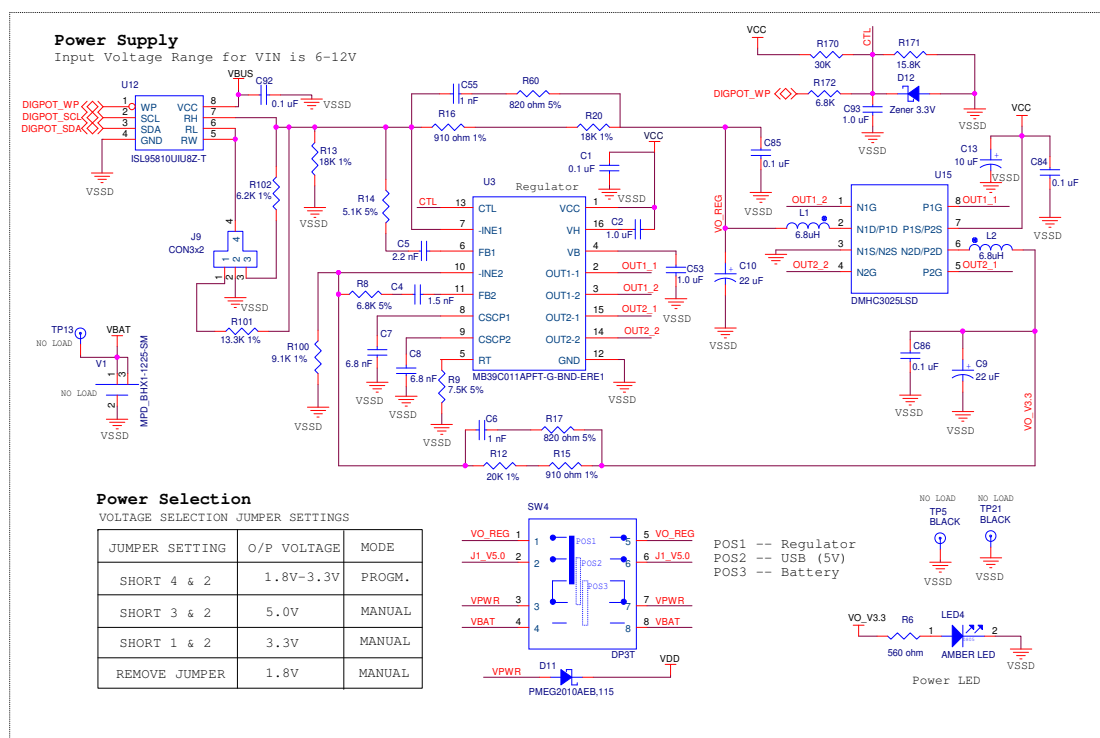
Figure A-1. Schematics of Serial Interface Connections and I2C Pull-Up via FETs



## A.2.2 Power Supply System

The power supply system is designed to support 1.8-V, 3.3-V, or 5-V operation when supplied from the external VIN or USB connector. The selection between 1.8 V, 3.3 V, and 5 V is made through a 4-pin jumper J9. If the board is powered from the USB connector, it provides 3.3 V and 5 V to the Arduino-compatible power headers. The board also has a provision for a 3-V coin cell. However, the coin cell holder is not loaded by default. If you want to use the 3-V coin cell, you should populate the coin cell holder (BHX1-1225-SM) at the V1 position on the bottom side of PSoC Analog Coprocessor Pioneer Kit. If the kit is used as a shield, then it can be powered from the 5-V supply of the Arduino-compatible baseboard. [Figure A-2](#) shows the schematics of power regulator circuit. Refer to [Table 3-2 on page 31](#).

Figure A-2. Schematics of Power Regulator Circuit

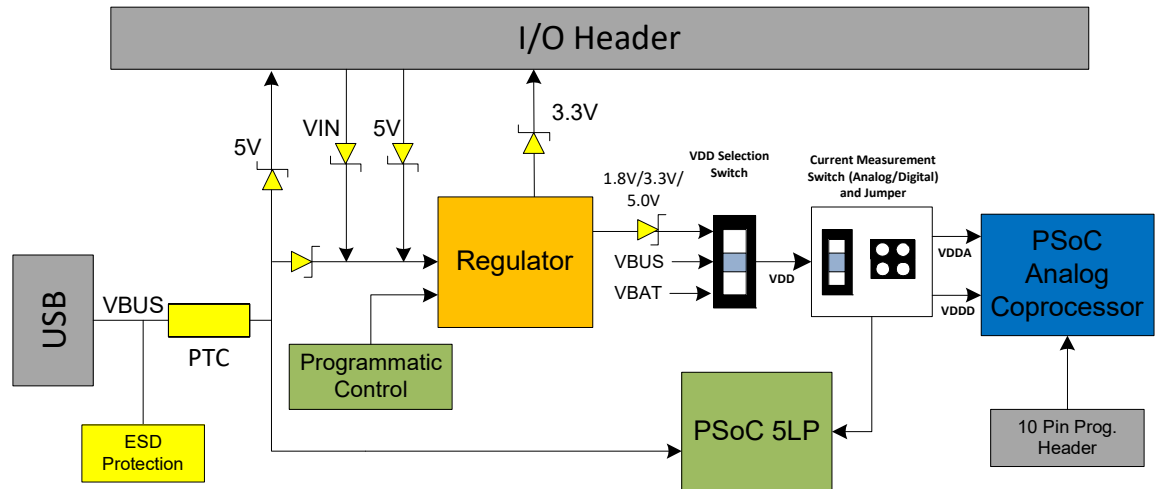


The voltage regulator (U3, MB39C011APFT-G-BND-ERE1) from Cypress has two channels that provide different voltage levels. The regulator generates a constant 3.3 V on one channel. The other channel can be configured to generate 1.8 V, 3.3 V, or 5.0 V by setting the jumper J9 to desired position. The digital potentiometer (U12) can be used to generate voltages in the 1.8 V–3.3 V range when powered from USB. This feature will be available in future versions of KitProg2. A 4-pin jumper (J9) is used to select any one of the three voltage levels (1.8 V, 3.3 V, or 5.0 V) or a programmable voltage level (1.8 V–3.3 V). VDD source selection slider switch (SW4) is used to select the power supply from the voltage regulator, USB or the 3-V coin cell.

### A.2.3 Protection Circuits

A positive temperature coefficient (PTC) resettable fuse is connected to protect the computer's USB ports from shorts and over current ( $> 500 \text{ mA}$ ). ORing diodes are provided to prevent damage to components when the board is powered from different voltage sources at the same time. ESD protection is provided on the USB connector.

Figure A-3. Power Supply Block Diagram with Protection Circuits

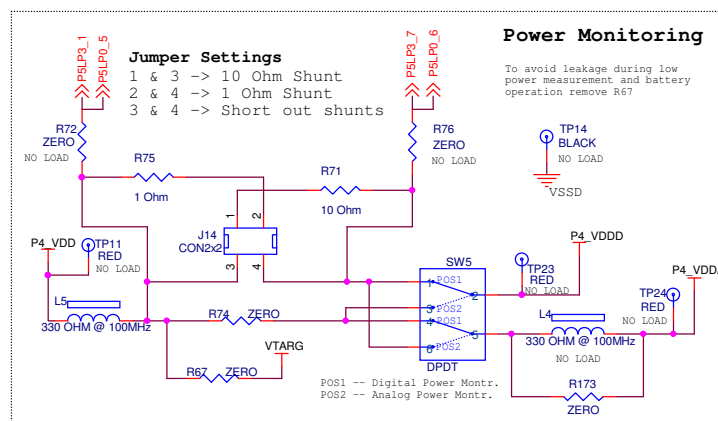


### A.2.4 Current Measurement Jumper

The kit has an onboard facility to monitor both analog and digital power consumed by the PSoc Analog Coprocessor. To measure the PSoc Analog Coprocessor power consumption, the jumper J14 is populated in series with the power supplies (Analog/Digital) to the PSoc Analog Coprocessor and can be used to measure current using an ammeter without the need to remove any components from the board. The switch SW5 is provided to select between Analog/Digital power domain. The KitProg2 on the board is also able to measure the power consumption of the PSoc Analog Coprocessor using its internal Del-Sig ADC and SAR ADC. For this purpose, either a 1 Ohm shunt resistor or 10 Ohm shunt resistor can be selected using the jumper J14. Connect an ammeter between terminals 3 and 4 of the jumper J14 to measure the PSoc Analog Coprocessor current consumption manually.

Figure A-4 shows the detailed circuit for analog/digital power monitoring.

Figure A-4. Detailed Circuit of Current Measurement Switch



## A.2.5 Expansion Connectors

### A.2.5.1 *Arduino-compatible Headers (J1 to J8, and J12)*

This kit has nine Arduino-compatible headers: J1, J2, J3, J4, J5, J6, J7, J8, and J12. You can develop applications based on the Arduino shield's hardware. The J1 header contains I/O pins for reset, internal reference voltage (IOREF), and power supply lines. The J2 header is an analog port that contains I/O pins for the SAR ADC. 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 (In-Circuit Serial Programming) compatible header for the SPI interface. Refer to the "No Load Components" section of the Bill of Materials for the header part number. The J5, J6, J7 and J8 headers are used to connect the PSoC Analog Coprocessor Pioneer Kit as shield. The reset and power supply lines are routed to J5, few I/Os are routed to J6, I2C lines are routed to J7, and UART lines are routed to J8. While using the kit as a shield, the connectors provided with the kit can be used to provide extra clearance between the shield and the base board.

### A.2.5.2 *Functionality of Header J12 (Unpopulated)*

The J12 header is a 2 × 3 header that supports Arduino shields. This header is used on a small subset of shields and is not populated on the PSoC Analog Coprocessor Pioneer Kit.

### A.2.5.3 *PSoC 5LP GPIO Header (J16)*

An 8 × 2 header is provided on the board to bring out several pins of the PSoC 5LP device to support advanced features such as a low-speed oscilloscope and a low-speed digital logic analyzer. This header also contains USB-UART bridge pins and USB-I2C bridge pins that can be used when these pins are not accessible on Arduino-compatible headers because a shield is connected. The additional pins on this header are direct connections to the internal programmable analog logic of PSoC 5LP. This header also has GPIOs for custom application usage.

### A.2.5.4 *KitProg2 Custom Application Header (J11)*

A 5 × 2 header is provided on the board to bring out more GPIOs of PSoC 5LP for custom application usage. This header also brings out the PSoC 5LP programming pins and can be programmed using [MiniProg3](#) and 5-pin programming connector.

## A.2.6 USB Mini-B Connector

The PSoC 5LP device connects to the USB port of a PC through a USB Mini-B connector, which can also be used to power the PSoC Analog Coprocessor Pioneer Kit. A resettable fuse is used to protect the computer's USB port from shorts and overcurrent. If more than 500 mA is drawn from the USB connector, the fuse will automatically break the connection until the short or overload is removed.

## A.2.7 Analog Sensors

Refer to the [Analog Sensors chapter on page 23](#).

## A.2.8 Pioneer Board LEDs

The Pioneer board has five LEDs. Three status LEDs (Red, Amber and Green) indicate the status of the KitProg2. Refer to the Troubleshooting section in [KitProg2 User Guide](#) for more information on

LED states. An amber LED indicates the status of power supplied to the kit. The kit also has a general-purpose RGB LED for user applications.

#### A.2.9 Push Buttons

The board contains a reset button (SW1) and two user buttons. The reset button is connected to the RESET pin of the PSoC Analog Coprocessor and is used to reset the device. One user button (SW2) is connected to P0[3] of the PSoC Analog Coprocessor. The other user button (SW3) is connected to P1[2] of the PSoC 5LP device. All the push buttons connect to ground on activation (active LOW).

#### A.2.10 Cypress Ferroelectric RAM (F-RAM)

The PSoC Analog Coprocessor Pioneer Kit contains an F-RAM device ([FM24V10-G](#)) that can be accessed through I2C lines P4 [0] and P4 [1] of the PSoC Analog Coprocessor. The F-RAM has a capacity of 1-Mbit (128 KB) with an I2C speed up to 3.4 MHz. 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 hard-wired 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 R44/R39 and R36/R37 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 3.6 V, 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 allow the F-RAM device to communicate with the PSoC Analog Coprocessor operating at 5 V. For more information on using the F-RAM, see [Using the FM24V10 F-RAM chapter on page 44](#).

#### A.2.11 External Crystals

The PSoC Analog Coprocessor Pioneer board has a provision to use a 32.768-kHz (Y1) external crystal for the WCO input. The WCO is used to provide an accurate low-frequency clock to PSoC Analog Coprocessor for Deep Sleep wake up intervals, and WDT reset intervals.

## A.3 Using the FM24V10 F-RAM

This section describes advanced features of the PSoC Analog Coprocessor Pioneer Kit as well as the corresponding projects. It can be used as a reference to use these features for other applications, according to the needs of the project.

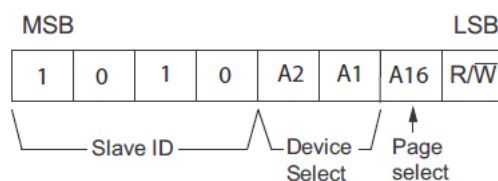
The PSoC Analog Coprocessor Pioneer board has an onboard ferroelectric RAM chip that can hold up to 1 Mb of data. The chip provides an I2C communication interface for data access. It is hard-wired to the I2C interface (P4[0] and P4[1] of the PSoC Analog Coprocessor); the same lines are routed to the 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-5](#): 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 Analog Coprocessor Pioneer board (FM24V10-G), 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 Analog Coprocessor Pioneer board is controlled by resistors R44/R39 (A1) and R36/R37 (A2). Because the memory location in the F-RAM device 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 read or write operations will take place.

**Note:** The 8-pin SOIC footprint provided for the F-RAM FM24V10 device on the PSoC Analog Processor Pioneer Kit is compatible with all I2C-based F-RAM devices from Cypress (FM24Vxx, FM24-CLxx, and CY15BxxxJ parts). F-RAM parts with more than 64 KB size support only four addresses (four devices of the same type on the same I2C bus); resistors connected to A1 (R44/R39) and A2 (R36/ R37) pins can be used to select any of the four addresses. F-RAM parts with less than 64 KB and FM24CLxx parts support eight addresses; resistors connected to A0 (R46/R47), A1 (R44/R39), and A2 (R36/R37) pins can be used to select one of the eight addresses.

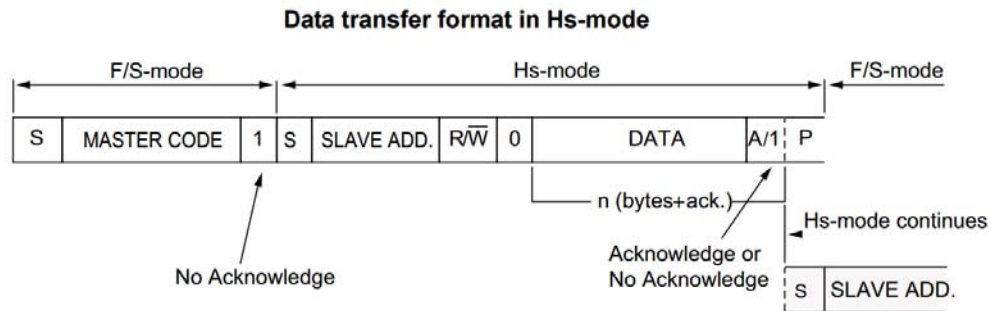
Figure A-5. F-RAM I2C Address Byte Structure



### A.3.2 High-Speed Mode (HS-mode)

The FM24V10 supports a 3.4-MHz high-speed mode. A master code (00001XXXb) must be issued to place the device into high-speed mode. Communication between the master and the slave will then be enabled for speeds up to 3.4 MHz. A STOP condition will exit the Hs-mode. Single- and multiple-byte reads and writes are supported.

Figure A-6. F-RAM I2C Data Format for HS Mode





### A.3.3 Write/Read Operation

The F-RAM datasheet includes details on how to perform a write/read operation with an F-RAM device. [Figure A-7](#) and [Figure A-8](#) provide a snapshot of the write/read packet structure as a quick reference.

Figure A-7. F-RAM Single-Byte and Multiple-Byte Write Packet Structure

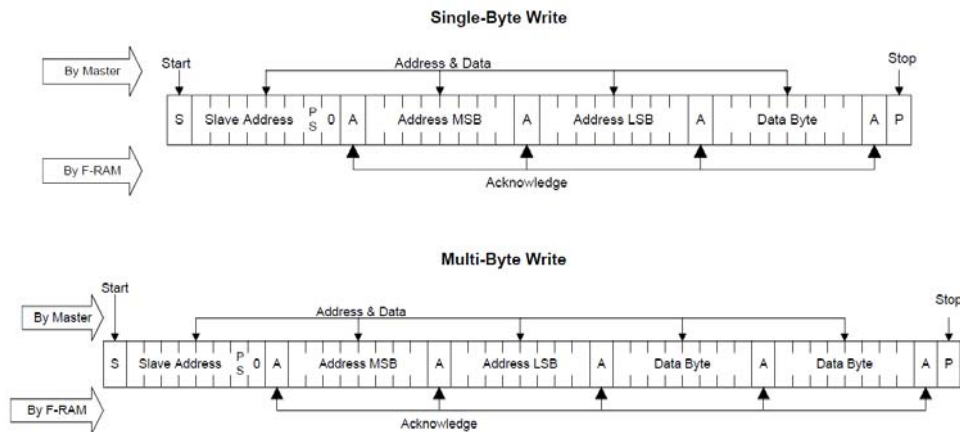
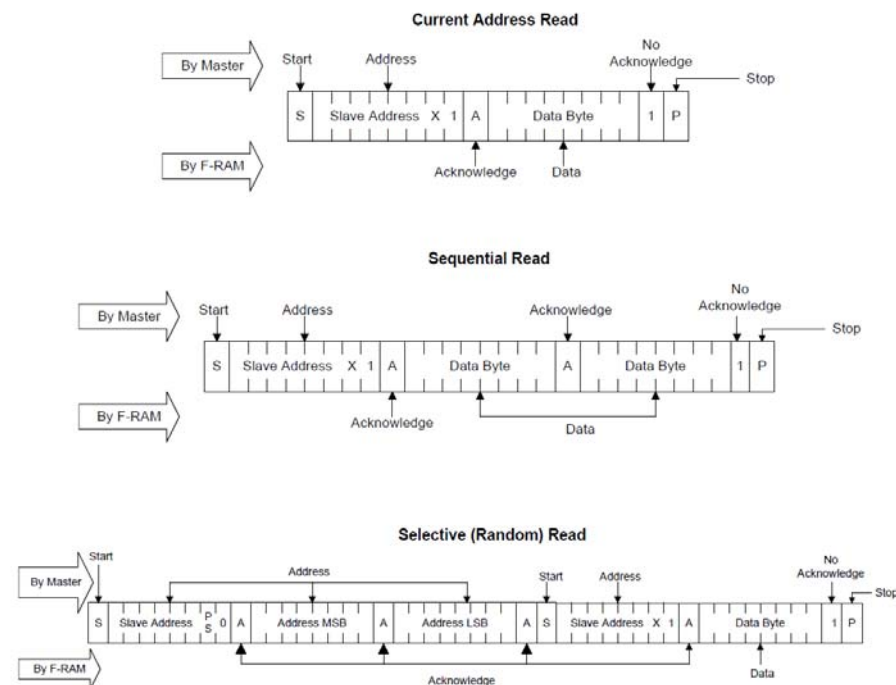


Figure A-8. F-RAM Single-Byte and Multiple-Byte Read Packet Structure



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

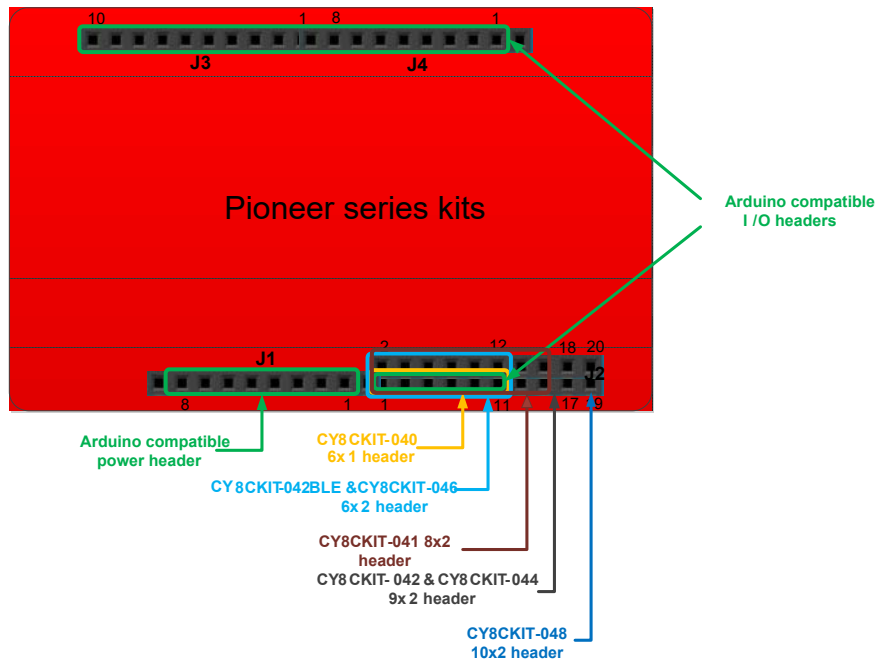
**Note:** You can communicate with the F-RAM using the Bridge Control Panel (BCP) software similar to the way you communicate with any other I2C slave device. Refer to the [KitProg2 User Guide](#) for more details on how to use the BCP software to communicate with an I2C slave device. Visit the [CY15FRAMKIT-001 Serial F-RAM Development Kit web page](#) for code examples and the Arduino library for details on interfacing I2C F-RAM devices with the PSoC Analog Coprocessor family.

## A.4 Migrating Projects Across Different Pioneer Series Kits

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-9. 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	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	VIN	VIN	VIN	VIN	VIN	VIN	VIN	VIN
2	GND	GND	GND	GND	GND	GND	GND	GND
3	GND	GND	GND	GND	GND	GND	GND	GND
4	5V	V5.0	V5.0	V5.0	V5.0	V5.0	V5.0	V5.0
5	3.3V	V3.3	V3.3	V3.3	V3.3	V3.3	V3.3	V3.3
6	RESET	RESET	RESET	RESET	RESET	RESET	RESET	RESET
7	IOREF	P4.VDD	P4.VDD	BLE.VDD	P4.VDD	P4.VDD	P4.VDD	PAC.VDD
8	NC	NC	NC	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	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	A0	P2[0]	P0[0]	P3[0]	P2[0]	P2[0]	P2[0] <sup>1</sup>	P3[0]
2	—	P0[2] <sup>1</sup>	—	P2[0]	P2[6] <sup>1</sup>	P3[6] <sup>1</sup>	P1[6] <sup>1</sup>	—
3	A1	P2[1]	P0[1]	P3[1]	P2[1]	P2[1]	P2[1]	P3[1]
4	—	P0[3] <sup>1</sup>	—	P2[1] <sup>1</sup>	P6[5] <sup>1</sup>	P3[7] <sup>1</sup>	P4[0] <sup>1</sup>	—
5	A2	P2[2]	P0[2] <sup>1</sup>	P3[2]	P2[2]	P2[2]	P2[2] <sup>1</sup>	P3[2]
6	—	P4.VDD	—	P2[2] <sup>1</sup>	P0[6] <sup>1</sup>	P9[0]	P4.VDD	—
7	A3	P2[3]	P0[4] <sup>1</sup>	P3[3]	P2[3]	P2[3]	P2[3] <sup>1</sup>	P3[3]
8	—	P1[5] <sup>1</sup>	—	P2[3] <sup>1</sup>	P4[4] <sup>1</sup>	P9[1]	P0[1] <sup>1</sup>	—
9	A4	P2[4]	P1[3]	P3[4]	P2[4]	P2[4]	P2[4] <sup>1</sup>	P3[4] <sup>1</sup>
10	—	P1[4] <sup>1</sup>	—	P2[4] <sup>1</sup>	P4[5] <sup>1</sup>	P9[2]	P0[6] <sup>1</sup>	—
11	A5	P2[5]	P1[2]	P3[5]	P2[5]	P2[5]	P2[5]	P3[7] <sup>1</sup>
12	—	P1[3] <sup>1</sup>	—	P2[5] <sup>1</sup>	P4[6] <sup>1</sup>	P9[3]	P0[7] <sup>1</sup>	—
13	—	P0[0]	—	—	P0[0]	—	P2[6] <sup>1</sup>	P2[6] <sup>1</sup>
14	—	GND	—	—	GND	—	GND	—
15	—	P0[1]	—	—	P0[1]	—	P3[4] <sup>1</sup>	P2[5] <sup>1</sup>
16	—	P1[2] <sup>1</sup>	—	—	P3[4] <sup>1</sup>	—	P3[6] <sup>1</sup>	—
17	—	P1[0]	—	—	P0[7] <sup>1</sup>	—	—	P2[1] <sup>1</sup>
18	—	P1[1] <sup>1</sup>	—	—	P3[5] <sup>1</sup>	—	—	—
19	—	—	—	—	—	—	—	P3[5] <sup>1</sup>
20	—	—	—	—	—	—	—	—

**Note 1:** These pins are also used for onboard peripherals. Refer to the [Table 3-3 on page 32](#), [Table A-5 on page 50](#), and [Table A-6 on page 50](#) 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	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	D8	P2[6]	P1[4]	P0[5]	P0[2]	P0[2]	P0[3] <sup>1</sup>	P2[4] <sup>1</sup>
2	D9	P3[6]	P1[5]	P0[4]	P0[3]	P0[3]	P2[7]	P1[7] <sup>1</sup>
3	D10	P3[4]	P1[6]	P0[2]	P2[7]	P6[3]	P0[0] <sup>1</sup>	P0[7]
4	D11	P3[0]	P1[1] <sup>1</sup>	P0[0]	P6[0]	P6[0]	P1[0] <sup>1</sup>	P0[4]
5	D12	P3[1]	P3[1]	P0[1]	P6[1]	P6[1]	P1[1] <sup>1</sup>	P0[5]
6	D13	P0[6]	P1[7]	P0[3]	P6[2]	P6[2]	P1[2] <sup>1</sup>	P0[6]
7	GND	GND	GND	GND	GND	GND	GND	GND
8	AREF	P1[7]	NC	VREF	P1[7]	VREF	P1[7] <sup>1</sup>	P1[3] <sup>1</sup>
9	SDA	P4[1]	P1[3]	P3[4]	P4[1]	P4[1]	P3[1]	P4[1]
10	SCL	P4[0]	P1[2]	P3[5]	P4[0]	P4[0]	P3[0]	P4[0]

**Note 1:** These pins are also used for onboard peripherals. Refer to the [Table 3-3 on page 32](#), [Table A-5 on page 50](#), and [Table A-6 on page 50](#) 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	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	D0	P0[4]	P0[5]	P1[4]	P3[0]	P3[0]	P0[4]	P0[4]
2	—	—	—	—	—	P8[0]	—	—
3	D1	P0[5]	P0[6]	P1[5]	P3[1]	P3[1]	P0[5]	P0[5]
4	—	—	—	—	—	P8[1]	—	—
5	D2	P0[7] <sup>1</sup>	P0[7]	P1[6]	P1[0]	P1[0]	P1[4] <sup>1</sup>	P1[5] <sup>1</sup>
6	—	—	—	—	—	P8[2]	—	—
7	D3	P3[7]	P3[2] <sup>1</sup>	P1[7]	P1[1]	P1[1]	P1[3] <sup>1</sup>	P2[3] <sup>1</sup>
8	—	—	—	—	—	P8[3]	—	—
9	D4	P0[0]	P0[3]	P1[3]	P1[2]	P1[2]	P1[5] <sup>1</sup>	P1[2] <sup>1</sup>
10	—	—	—	—	—	P8[4]	—	—
11	D5	P3[5]	P3[0]	P1[2]	P1[3]	P1[3]	P3[5] <sup>1</sup>	P1[1] <sup>1</sup>
12	—	—	—	—	—	P8[5]	—	—
13	D6	P1[0]	P1[0]	P1[1]	P5[3]	P5[6]	P3[7] <sup>1</sup>	P1[0] <sup>1</sup>
14	—	—	—	—	—	P8[6]	—	—
15	D7	P2[7] <sup>1</sup>	P2[0] <sup>1</sup>	P1[0]	P5[5]	P5[5]	P0[2] <sup>1</sup>	P2[7] <sup>1</sup>
16	—	—	—	—	—	P8[7]	—	—

**Note 1:** These pins are also used for onboard peripherals. Refer to the [Table 3-3 on page 32](#), [Table A-5 on page 50](#), and [Table A-6 on page 50](#) for connection details.

Table A-5. RGB LED Pin Map

#	Arduino Pin	Pioneer Series Kits						
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	Red	P1[6]	P3[2]	P2[6]	P0[6]	P5[2]	P3[4]	P1[4]
2	Green	P0[2]	P1[1]	P3[6]	P2[6]	P5[3]	P2[6]	P2[6]
3	Blue	P0[3]	P0[2]	P3[7]	P6[5]	P5[4]	P3[6]	P1[6]

Table A-6. User Switch Pin Map

#	Arduino Pin	Pioneer Series Kits						
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046	CY8CKIT-041	CY8CKIT-048
1	SW2	P0[7]	—	P2[7]	P0[7]	P0[7]	P0[7]	P0[3]

## A.5 Bill of Materials

BOM file is located in the following path in the kit software installation:

```
<Install_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit\1.0  
\Hardware\CY8CKIT-048 PCBA BOM.xlsx
```

# Revision History



## CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide Revision History

Document Title: CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide				
Document Number: 002-11190				
Revision	ECN Number	Issue Date	Origin of Change	Description of Change
**	5185681	06/04/2016	DIMA	Initial version of the kit guide.
*A	5302951	06/10/2016	SRDS	<p>Updated <a href="#">Introduction</a> chapter on page 5:  Updated <a href="#">Figure 1-1</a>.  Updated <a href="#">Figure 1-2</a>.  Updated <a href="#">Figure 1-3</a>.  Updated <a href="#">Figure 1-4</a>.</p> <p>Updated <a href="#">Software Installation</a> chapter on page 15:  Updated <a href="#">"Install Software"</a> on page 15:  Updated description.</p> <p>Updated <a href="#">Kit Operation</a> chapter on page 18:  Updated <a href="#">"Theory of Operation"</a> on page 18:  Updated description.  Updated <a href="#">"Analog Sensors"</a> on page 23:  Updated <a href="#">Table 3-1</a>:  Updated details in "Sensor Part" column corresponding to "Thermistor" and "Humidity sensor".  Updated <a href="#">"Interfacing with a PIR Motion Sensor"</a> on page 23:  Updated description.  Updated <a href="#">"Interfacing with an Ambient Light Sensor"</a> on page 26:  Updated description.  Updated <a href="#">"Interfacing with a Thermistor"</a> on page 27:  Updated description.  Updated <a href="#">"Interfacing with an Inductive Proximity Sensor"</a> on page 28:  Updated description.  Updated <a href="#">"Interfacing with a Humidity Sensor"</a> on page 29:  Updated description.  Updated <a href="#">"Power Supply"</a> on page 31:  Updated description.</p>



## CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide Revision History

Document Title: CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide				
Document Number: 002-11190				
Revision	ECN Number	Issue Date	Origin of Change	Description of Change
*A (cont.)	5302951	06/10/2016	SRDS	<p>Updated <a href="#">Kit Operation</a> chapter on page 18:</p> <p>Updated “<a href="#">Theory of Operation</a>” on page 18:</p> <p>Updated “<a href="#">Analog Sensors</a>” on page 23:</p> <p>Updated “<a href="#">Interfacing with an Inductive Proximity Sensor</a>” on page 28:</p> <p>Updated <a href="#">Figure 3-13</a>.</p> <p>Updated “<a href="#">Power Supply</a>” on page 31:</p> <p>Updated <a href="#">Table 3-2</a>.</p> <p>Updated “<a href="#">Flexible Prototyping</a>” on page 32:</p> <p>Updated <a href="#">Table 3-3</a>:</p> <p>Updated details in “0Ω for sensor” column corresponding to P1[0] and P1[5] pins.</p> <p>Updated details in “Header” column for all pins.</p> <p>Updated details in “0Ω for header” column corresponding to P1[3] pin.</p> <p>Updated “<a href="#">Programming and Debugging Using PSoC Creator</a>” on <a href="#">page 34</a>:</p> <p>Updated description.</p> <p>Updated <a href="#">Code Examples</a> chapter on page 36:</p> <p>Updated description.</p> <p>Updated “<a href="#">Using the Kit Code Example</a>” on page 36:</p> <p>Updated description.</p> <p>Updated <a href="#">Appendix</a> chapter on page 39:</p> <p>Updated “<a href="#">Hardware Functional Description</a>” on page 39:</p> <p>Updated “<a href="#">Power Supply System</a>” on page 40:</p> <p>Updated description.</p> <p>Updated <a href="#">Figure A-2</a>.</p> <p>Updated “<a href="#">Current Measurement Jumper</a>” on page 41:</p> <p>Updated description.</p> <p>Removed table “Jumper (J14) Settings for Different Load Selection”.</p> <p>Updated “<a href="#">Expansion Connectors</a>” on page 42:</p> <p>Updated “<a href="#">Arduino-compatible Headers (J1 to J8, and J12)</a>” on page 42:</p> <p>Updated description.</p> <p>Updated “<a href="#">PSoC 5LP GPIO Header (J16)</a>” on page 42:</p> <p>Updated description.</p>

## CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide Revision History

Document Title: CY8CKIT-048 PSoC® Analog Coprocessor Pioneer Kit Guide				
Document Number: 002-11190				
Revision	ECN Number	Issue Date	Origin of Change	Description of Change
*A (cont.)	5302951	06/10/2016	SRDS	Updated <a href="#">“KitProg2 Custom Application Header (J11)”</a> on page 42: Replaced “PSoC 5LP” with “KitProg2” in heading. Updated description. Updated <a href="#">“USB Mini-B Connector”</a> on page 42: Updated description. Updated <a href="#">“Push Buttons”</a> on page 43: Updated description. Updated <a href="#">“Cypress Ferroelectric RAM (F-RAM)”</a> on page 43: Updated description. Updated <a href="#">“External Crystals”</a> on page 43: Updated description. Updated <a href="#">“Using the FM24V10 F-RAM”</a> on page 44: Updated <a href="#">“Address Selection”</a> on page 44: Updated description. Updated <a href="#">“Migrating Projects Across Different Pioneer Series Kits”</a> on page 47: Updated <a href="#">“Arduino Uno-Compatible Headers”</a> on page 48: Updated <a href="#">Table A-1</a> . Updated <a href="#">Table A-2</a> . Added <a href="#">Table A-4</a> . Updated <a href="#">Table A-6</a> : Removed SW1 pin and details.
*B	5767600	06/13/2017	AESATP12	Updated logo and copyright.
*C	6497247	02/28/2019	DIMA	Updated the PSoC Analog Coprocessor datasheet link. Updated the copyright information.