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-025

PSoC® Precision Analog Temperature Sensor Expansion Board Kit Guide

Doc. # 001-65791 Rev. *K

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2011–2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("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. 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. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, 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 or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

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

Contents

Safety Information 5

1. Introduction 7

1.1 Kit Contents 9
1.2 PSoC Creator™ 11
1.3 Getting Started 11
1.4 Additional Learning Resources 11
1.4.1 Beginner Resources 11
1.4.2 Engineers Looking for More 11
1.4.3 Learn from Peers 11
1.4.4 More Code Examples 12
1.4.5 Technical Support 13
1.5 Document Conventions 14

2. Software Installation 15

3. Kit Operation 18

3.1 Kit Overview 18
3.2 Kit Connections 18
3.3 Temperature Sensors 21
3.3.1 Thermocouple 21
3.3.2 Thermistor 21
3.3.3 RTD 22
3.3.4 Diode 22
3.4 Prototype Boards 23

4. Hardware 24

4.1 System Block Diagram and Theory of System Operation 24
4.2 Thermocouple 24
4.3 Thermistor 25
4.4 RTD 26
4.5 Diode 28

5. Example Projects **30**
5.1 Overview 30
5.1.1 Migrating Projects to use with CY8CKIT-050 and CY8CKIT-010 31
5.2 Tempsense 32
5.2.1 Project Description 32
5.2.2 Project Operation 33
5.2.3 Project Details 40
5.3 Sequenced ADC 52
5.3.1 Project Description 52
5.3.2 Project Operation 52
5.3.3 Project Details 58
5.4 Thermal Management 68
5.4.1 Project Description 68
5.4.2 Project Operation 70
5.4.3 Project Details 75

A. Appendix **83**
A.1 Schematic 83
A.2 Board Layout 84
A.2.1 PDC-09802 Top 84
A.2.2 PDC-09802 Bottom 84
A.3 Bill of Materials (BOM) 85
A.4 Regulatory Compliance Information 85

Revision History **86**

Safety Information

Regulatory Compliance

The CY8CKIT-025 is intended for use as a development platform for hardware or software in a laboratory environment. The board is an open system design, which does not include a shielded enclosure. This may cause interference to other electrical or electronic devices in close proximity.

In a domestic environment, this product may cause radio interference. In this case, you may be required to take adequate prevention measures. Also, the board should not be used near any medical equipment or RF devices.

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

The CY8CKIT-025 as shipped from the factory has been verified to meet with requirements of CE as a Class A product.



The CY8CKIT-025 contains electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused CY8CKIT-025 boards in the protective shipping package.



End-of-Life/Product Recycling

This kit has an end-of-life cycle five years from the date of manufacturing mentioned on the back of the box. Contact your nearest recycler for discarding the kit.

General Safety Instructions

Electrostatic Discharge Protection

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

Handling Boards

CY8CKIT-025 boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static free surface. Use a conductive foam pad if available. Do not slide board over any surface.

1. Introduction

Thank you for your interest in PSoC solutions for temperature sensing. The CY8CKIT-025 PSoC® Precision Analog Temperature Sensor Expansion Board Kit (EBK) together with the example projects provide complete solutions demonstrating PSoC-based temperature sensing. The CY8CKIT-025 board connects with the CY8CKIT-030 PSoC 3 Development Kit (DVK), CY8CKIT-050 PSoC 5LP Development Kit, or the CY8CKIT-001 PSoC Development Kit. The kit supports temperature measurement using four temperature sensors:

- Thermocouple
- Thermistor
- Resistive Temperature Detector (RTD)
- Diode

Note: The TempSense example project outputs one more temperature value read from the DS600 IC. This is used for cold junction compensation for the thermocouple and not intended as an option for the onboard temperature sensor.

The four sensors have their own advantages and limitations. The choice of a sensor for an application depends on the cost, accuracy required, and temperature measurement range.

Thermocouples measure temperatures in a wide range, from –250 °C to 2300 °C. However, the voltage change is not linear and takes more CPU cycles for temperature computation. Thermocouples require another temperature sensor, such as the thermistor, diode, or RTD, to measure the cold junction temperature. The DS600 IC or the thermistor can be used for cold junction compensation of the thermocouple. A thermistor, diode, or RTD cannot measure temperatures greater than 850 °C. Thermocouples have a very good response time.

Thermistors have a highly nonlinear curve and take more CPU cycles for temperature computation. They measure temperature in the range –100 °C to 200 °C. Thermistors have a good response time and they are less expensive when compared to RTDs and thermocouples.

RTDs have a linear and repeatable resistance variation with temperature, making it easier to compute temperature accurately. They measure temperature from –200 °C to 850 °C. RTDs are expensive due to their linearity and accuracy. RTDs have a lower response time compared to a thermistor or thermocouple.

Diodes are the cheapest sensors for temperature measurement. They measure temperature in the range –50 °C to 150 °C. Accurate temperature measurement is difficult with a diode. Table 1-1 shows a comparison among the four sensors.

Table 1-1. Temperature Sensor Comparison

Parameter	RTD	Thermocouple	Thermistor	Diode
Temperature range (°C)	–200 to +850	–250 to +2350	–100 to +300	–50 to +150
Sensitivity at 25 °C	0.387 Ω/°C	40 µV/°C (K-type)	416 Ω/°C	250 µV /°C
Accuracy	High	Medium to High	Medium	Low
Linearity	Good	Fair	Poor	Good
Typical cost (US $)	$3–$80	$3–$15	$0.2–$10	<$0.2
Typical distance of sensing	Surface mount for onboard temperature. 3- and 4-wire up to a few hundred meters	<100 meters	Surface mount for onboard temperature. Leaded for <1 meter	Onboard temperature
Resource requirement	Excitation current, amplifier, ADC, reference resistor	Amplifier, ADC, voltage reference, and another temperature sensor for cold junction	Excitation current, ADC, reference resistor	Excitation current, amplifier, ADC
Response time	Slow	Fast	Fast	Slow
Computational complexity (best possible accuracy)	High	Very high	Very high	Medium
Cypress application note	AN70698	AN75511	AN66477	AN60590

Note: The temperature sensor comparison chart does not include values for DS600 IC because it is used for cold junction compensation for the thermocouple; it is not intended for use as an onboard temperature sensor.

This kit includes three example projects, divided into project areas based on end applications. Each project area contains examples for the CY8CKIT-030, CY8CKIT-050, and CY8CKIT-001 development kits. The CY8CKIT-030 DVK contains a PSoC 3 (8051 core) and the CY8CKIT-050 DVK contains a PSoC 5LP (ARM Cortex-M3); both support high-performance analog applications.

The CY8CKIT-001 DVK can be used to develop solutions for PSoC 1 (M8C proprietary core), PSoC 3, or PSoC 5LP (ARM Cortex 3). The included example projects support PSoC 3 (8051) and PSoC 5LP (ARM Cortex-M3).

The Tempsense example project provided with the kit supports most temperature sensing applications. It provides a high-performance temperature sensing example by using the ADC in 20-bit mode and implementing a high-precision linearization algorithm. It includes a resistance temperature detector (RTD), thermocouple, thermistor, two temperature diodes, and an IC temperature sensor. You can evaluate and compare sensor performance. Complete interface and output signal correction solutions can be developed for each type of temperature sensor. With the 20-bit ADC of the PSoC, it is possible to obtain a 0.1-°C resolution easily with all four sensors. The accuracy varies in each case depending on the method of measurement and the sensor used. See the Hardware chapter on page 24 for details.

The Sequenced ADC example project includes three temperatures (RTD, temperature diode, and IC temperature sensor) as well as millivolt and voltage inputs. This project can be used for sensing applications that require temperature measurement (such as RTD) as well as other types of voltage output sensors. It can also be used for system monitoring applications requiring temperature (usually

temperature diode or IC temperature sensor) and voltage rail measurement. The sequenced ADC example project uses the ADC in 16-bit mode and provides a higher throughput. In this project, the ADC readings are automatically repeated at consistent intervals, which allow filtering of ADC readings to remove specific frequencies of interest, such as a 50- and 60-Hz hum.

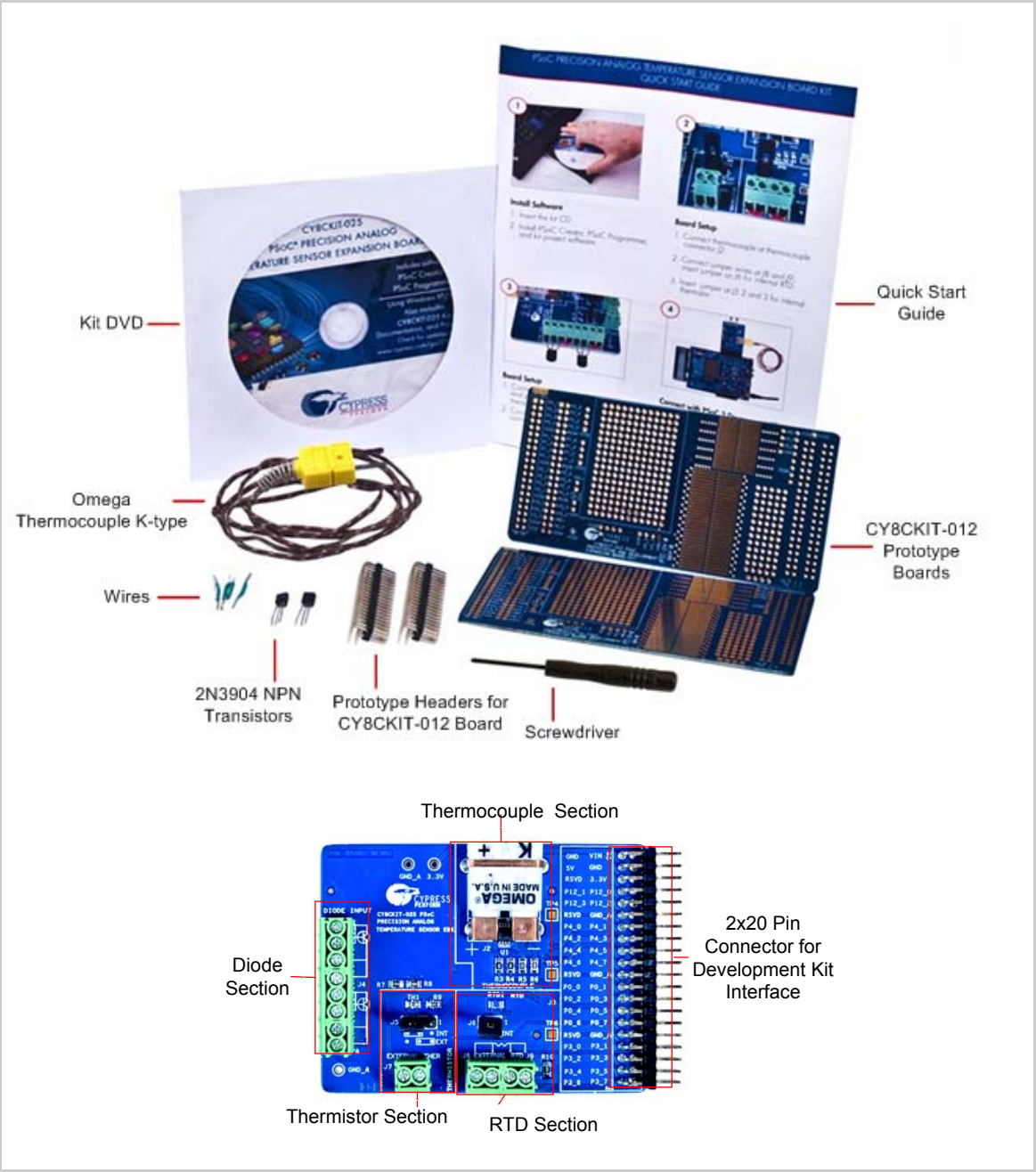
The Thermal Management System example project supports fan control applications. Add the CY8CKIT-036 EBK (with two four-wire fans) to the CY8CKIT-025 EBK and a development kit to support complete temperature-based fan control solutions. For more details on CY8CKIT-036 PSoC Thermal Management EBK, see the Cypress website.

1.1 Kit Contents

The CY8CKIT-025 PSoC Precision Analog Temperature Sensor EBK includes:

- CY8CKIT-025 PSoC Precision Analog Temperature Sensor Board
- Two CY8CKIT-012 prototype boards
- Omega thermocouple K-type 5srtc-gg-20-36
- Two 2N3904 NPN transistors
- Two prototype headers for CY8CKIT-012 prototype board
- Quick start guide
- Resource CD
- Wires
- Mini Philips cross screw driver (black)

Figure 1-1. CY8CKIT-025 Kit Contents



1.2 PSoC Creator™

Cypress's PSoC Creator software is a state-of-the-art, easy-to-use integrated development environment (IDE) that introduces a hardware and software design environment based on classic schematic entry and revolutionary embedded design methodology.

With PSoC Creator, you can:

- Automatically place and route select components and integrate simple glue logic normally located in discrete muxes.
- Trade off hardware and software design considerations, allowing you to focus on what matters and getting to market faster.

PSoC Creator also enables you to tap into an entire tools ecosystem with integrated compiler tool chains, RTOS solutions, and production programmers to support both PSoC 3 and PSoC 5LP devices.

1.3 Getting Started

Follow the steps in the Software Installation chapter on page 15 to install the kit. See the Kit Operation chapter on page 18 and Hardware chapter on page 24 to understand the kit operation and hardware. The Example Projects chapter on page 30 explains the details of the firmware and provides steps to run the projects.

1.4 Additional Learning Resources

Visit www.cypress.com/go/psoc3 for additional learning resources in the form of datasheets, technical reference manual, and application notes.

1.4.1 Beginner Resources

AN54181 - PSoC 3 - Getting Started with a PSoC 3 Design Project

PSoC Creator Training

1.4.2 Engineers Looking for More

AN54460 - PSoC® 3, PSoC 4, and PSoC 5LP Interrupts

AN52705 - PSoC® 3 and PSoC 5LP - Getting Started with DMA

AN52701 - PSoC® 3 and PSoC 5LP - Getting Started with CAN (Controller Area Network)

AN54439 - PSoC® 3 and PSoC 5LP External Crystal Oscillators

AN52927 - PSoC® 3 and PSoC 5LP: Segment LCD Direct Drive

Cypress continually strives to provide the best support. Click here to view a growing list of application notes for PSoC 3 and PSoC 5LP.

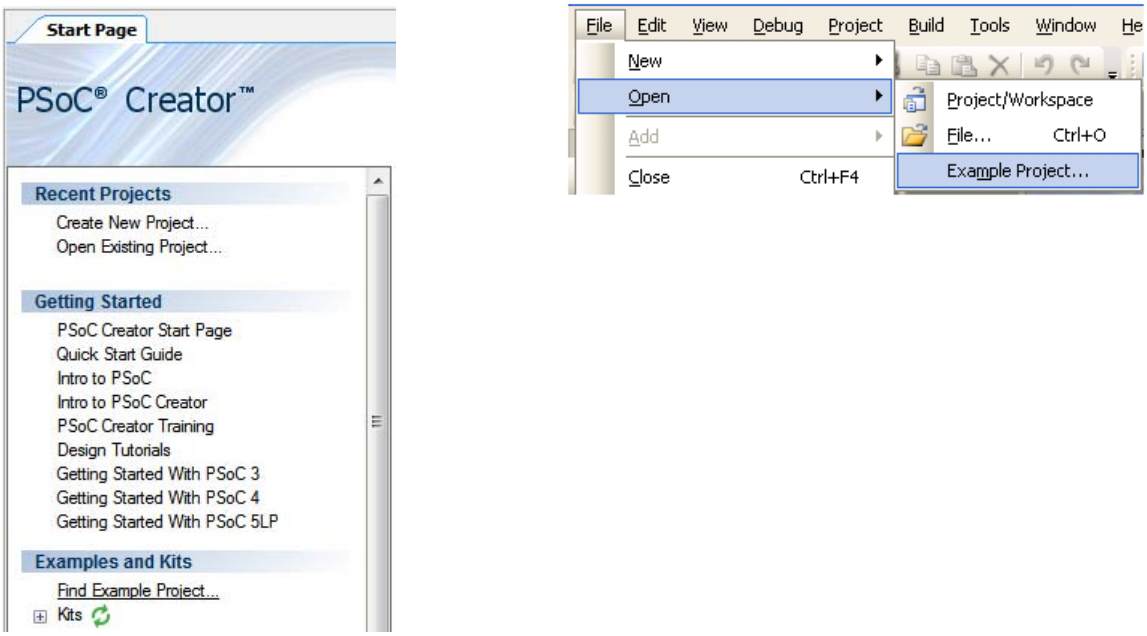
1.4.3 Learn from Peers

Cypress Developer Community Forums

1.4.4 More Code Examples

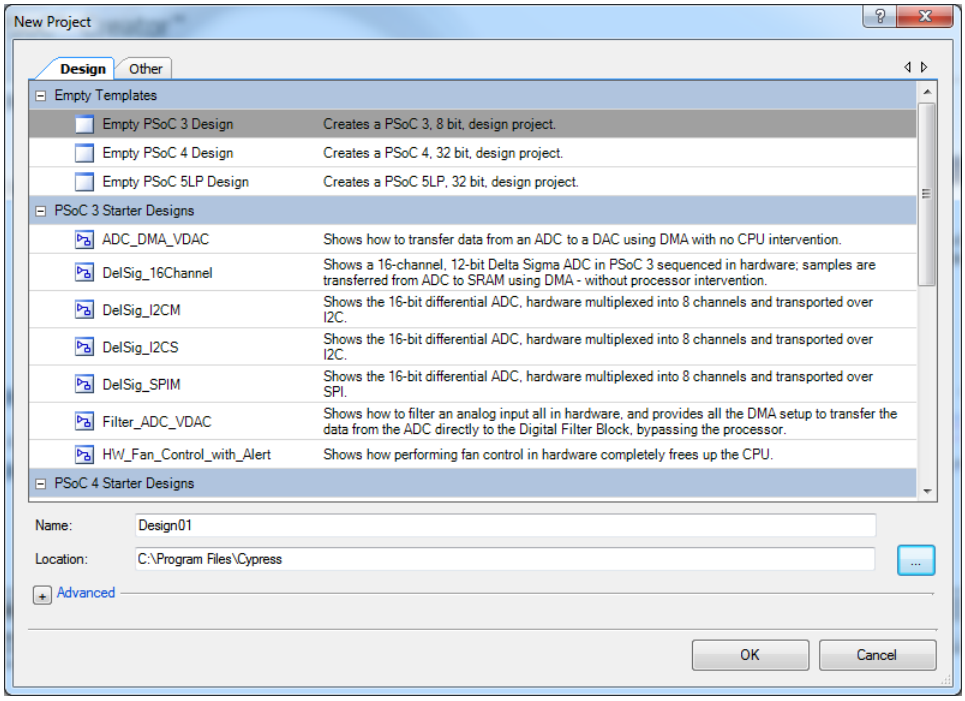
PSoC Creator provides several example projects that make code development fast and easy. To access these example projects, click **Find Example Project…** under the **Example and Kits** section in the **Start Page** of PSoC Creator or navigate to **File > Open > Example Project…**.

Figure 1-2. Find Example Project



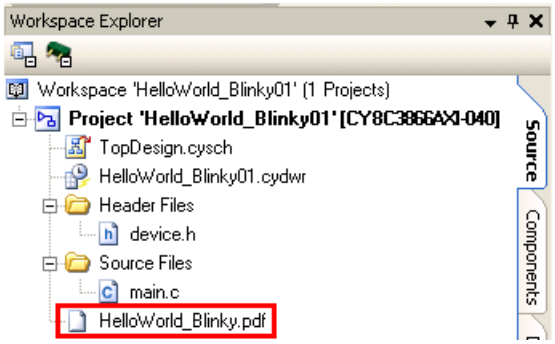
The Find Example Project section has various filters that help you locate the most relevant project. PSoC Creator provides several starter designs. These designs highlight features that are unique to PSoC devices. They allow you to create a design with various components, instead of creating a new empty design; code is also provided. To use a starter design for your project, navigate to **File > New > Project** and select the design required.

Figure 1-3. New Project



The starter designs and the example project contain a PDF within the project that explains the features of the project and its configuration.

Figure 1-4. Project PDF Location



Note: The example projects and starter designs are designed for CY8CKIT-001 PSoC Development Kit. However, these projects can be converted for use with CY8CKIT-030 PSoC 3 Development Kit or CY8CKIT-050 PSoC 5LP Development Kit by following the procedure in the knowledge base article Migrating CY8CKIT-001 DVK project to CY8CKIT 030/ 050.

Apart from the example projects and starter designs that are available within PSoC Creator, Cypress continuously strives to provide the best support. Click here to view a growing list of application notes for PSoC 3, PSoC 4, and PSoC 5LP.

1.4.5 Technical Support

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

1.5 Document Conventions

Table 1-2. Document Conventions for Guides

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

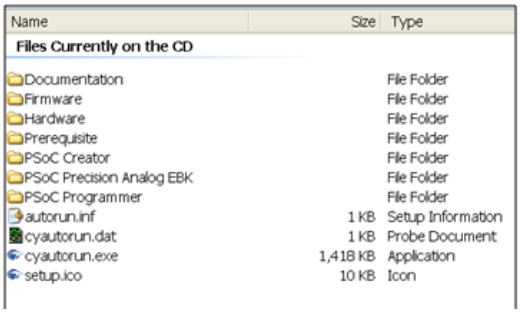
2. Software Installation

The kit CD/DVD contains the PSoC Precision Analog EBK-related software. The installer in the CD/DVD checks if the prerequisites – PSoC Creator, PSoC Programmer, Windows Installer, Windows .NET, and Keil C51 compiler, are installed in your PC. If these applications are not installed, then it installs them before installing the kit. If the Acrobat Reader application is not installed in your PC, then the installer provides the link to install the same; this does not prevent kit installation. Note that Adobe Reader is required to view the kit documents. The installer also installs the example projects, user guide, quick start guide, and other kit-related documents as part of kit installation.

The installation steps are as follows:

1. Insert the kit CD/DVD into the CD/DVD drive of your computer. The CD/DVD is designed to automatically open an installation dialog (see Figure 2-2). **Note**: If auto-run does not execute, double-click ***cyautorun.exe*** in the root directory of the CD/DVD.

Figure 2-1. CD/DVD Root Directory



2. The CD/DVD installation dialog prompts you to open this file or begin installing the development environment software. Click the **Install PSoC Precision Analog EBK** button to begin installation.

Figure 2-2. PSoC Precision Analog EBK Window



3. Choose the **Typical** installation type in the Product Installation Overview window.

PSoC Creator uses the DP8051 Keil 9.51 compiler to build PSoC 3 applications. This compiler is included on the CD/DVD; the installer will prompt you to install the compiler if it is not detected.

Note: The Keil compiler is distributed with a free license. You must activate this license within 30 days of installation. When the Cypress software installation is complete and you run PSoC Creator, activate the compiler license from **Help > Register > Keil**.

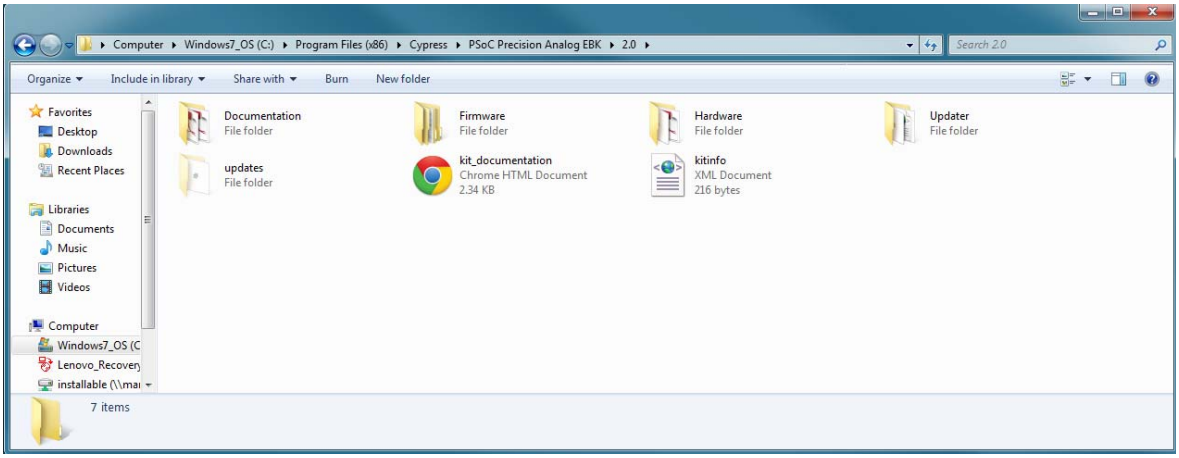
Note: If there is a problem with Keil registration, rename the *_*tools.ini* file in `<Install_Directory>\PSoC Creator\<version>\PSoC Creator\import\keil\pk51\<version>` to "*tools.ini*" for the Keil registration to be successful.

After the installation is complete, the following are installed in your computer:

- PSoC Creator 3.3 CP3 or later
- PSoC Programmer 3.24.2 or later
- Kit documents:
  - Quick Start Guide
  - User Guide
  - Temperature Sensor Datasheets
- Firmware
  - Example Projects
- Hardware
  - Schematic
  - Layout
  - Bill of Materials (BOM)

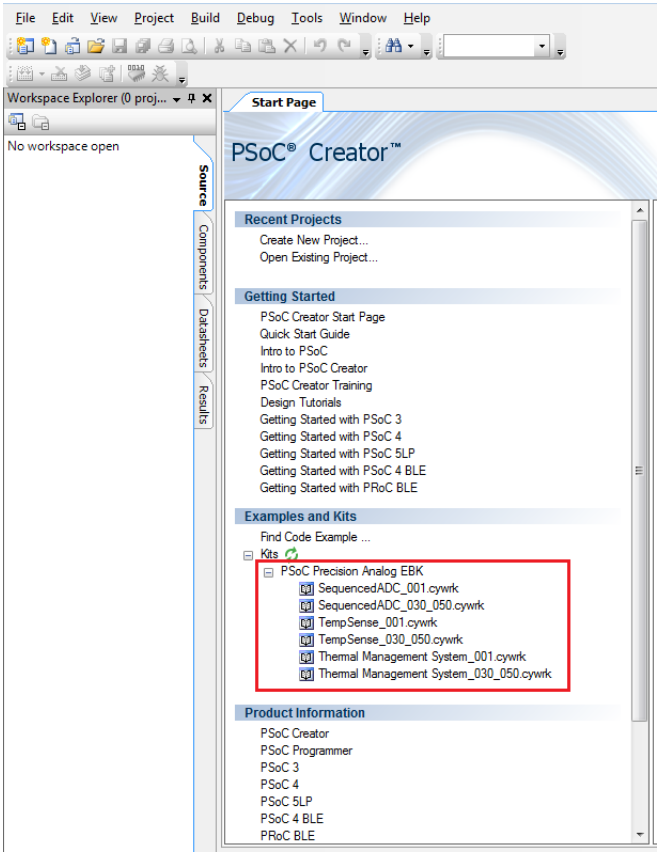
If you have PSoC Creator and PSoC Programmer already installed in your computer, the installer will install only the kit-related files in `<Install_Directory>\PSoC Precision Analog EBK\<version>`, as shown in Figure 2-3.

Figure 2-3. Installed Files



After installation, launch the example projects associated with the kit from the PSoC Creator Start Page, as shown in Figure 2-4.

Figure 2-4. PSoC Creator Start Page

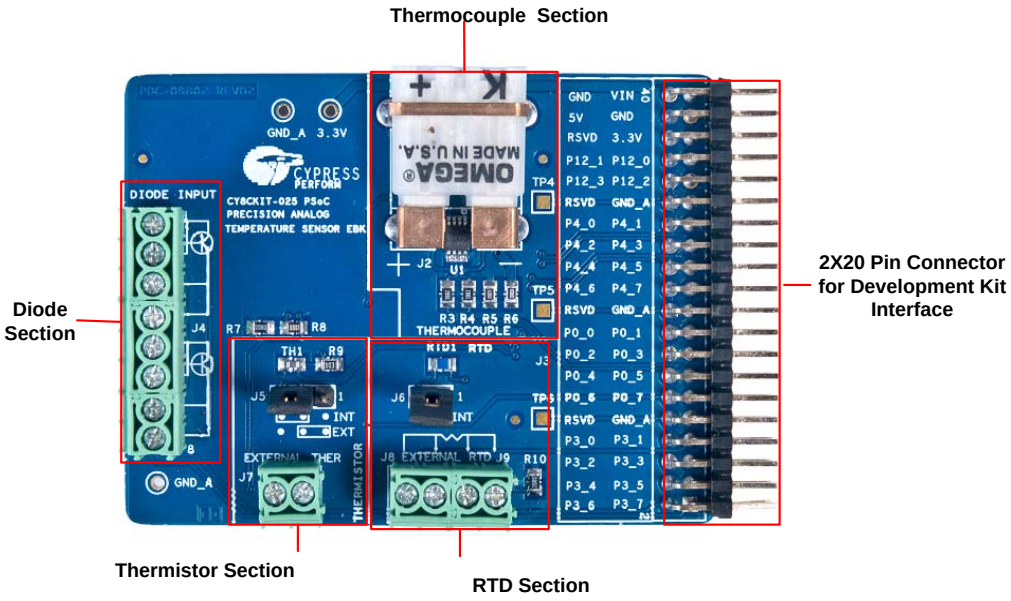


3. Kit Operation

3.1 Kit Overview

Figure 3-1 shows the CY8CKIT-025 PSoC Precision Analog Temperature Sensor EBK. The circuits associated with each sensor are boxed and labeled in the figure. The kit has an RTD, a thermistor, and a DS600 IC sensor onboard and provides interface slots to plug in your thermocouple, diode, thermistor, or external RTD. The kit includes two diodes and a K-type thermocouple.

Figure 3-1. CY8CKIT-025 PSoC Precision Analog Temperature Sensor EBK



3.2 Kit Connections

The CY8CKIT-025 EBK connects to a development kit (DVK) using a 2×20 pin connector. You can use the CY8CKIT-030 PSoC 3 DVK, CY8CKIT-050 PSoC 5LP DVK, or CY8CKIT-001 PSoC DVK. The CY8CKIT-025 EBK can be connected to port E of the CY8CKIT-030 PSoC 3 DVK (see Figure 3-2) or port E of the CY8CKIT-050 PSoC 5LP DVK (see Figure 3-3), or port A of the CY8CKIT-001 PSoC DVK (see Figure 3-4).

Figure 3-2. Connect 2×20 Pin Connector to Port E of CY8CKIT-030 PSoC 3 DVK

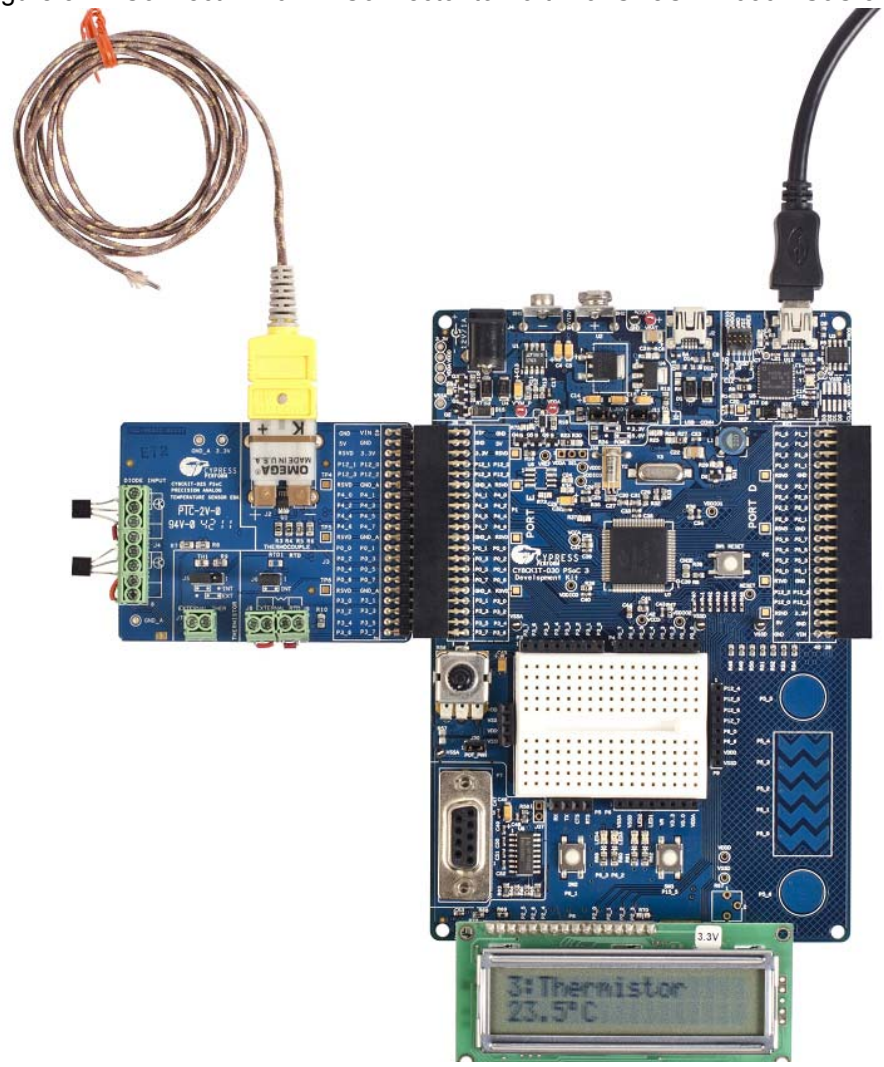


Figure 3-3. Connect 2×20 Pin Connector to Port E of CY8CKIT-050 PSoC 5LP DVK

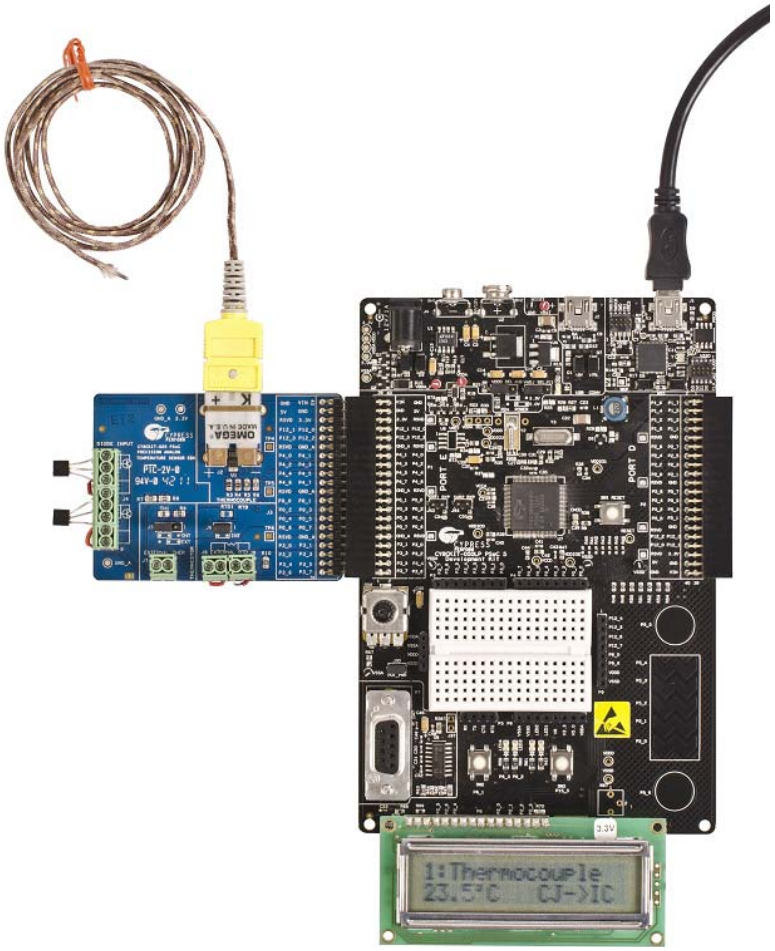
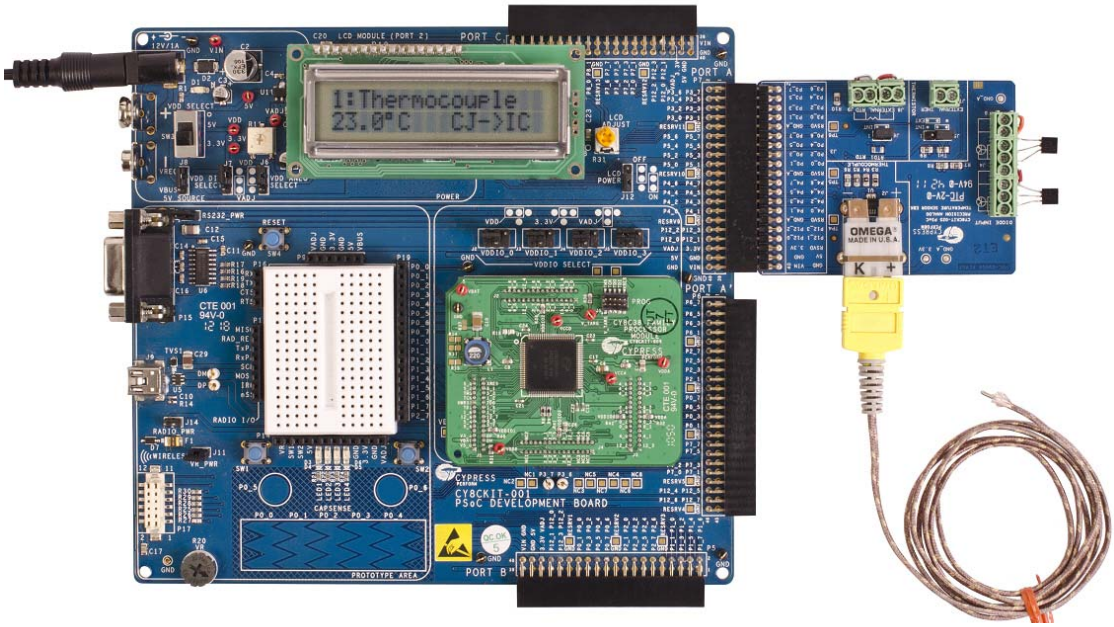


Figure 3-4. Connect 2×20 Pin Connector to Port A of CY8CKIT-001 PSoC DVK



WARNING: Static discharges from the human body can easily reach very high voltages in the order of kV. Such voltages can damage the PSoC device on the development kit. Ensure that you discharge any static before touching the hardware. Follow these steps to avoid any unwanted behavior on the board:

1. Power off the kit before making any connections
2. Connect the CY8CKIT-025 EBK to the development kit being used
3. Power on the development kit

3.3 Temperature Sensors

The PSoC Precision Analog Temperature Sensor EBK supports the following temperature sensors:

- Thermocouple
- Thermistor
- Resistive Temperature Detector (RTD)
- Diode
- DS600 IC

3.3.1 Thermocouple

The board has a K-type thermocouple socket, PCC-SMP-K-5. The K-type Omega thermocouple, 5SRTC-GG-K-20-36, is shipped with the kit. It complies with the ASTM special limits of error tolerance standard. IC DS600 is provided on the board for cold junction compensation. The thermistor on the board can also be used for cold junction compensation. Because the onboard thermistor is not calibrated, the IC (DS600) is provided for cold junction compensation. AN75511 - Temperature Measurement with Thermocouples provides an example project that demonstrates how to use the thermistor for cold junction compensation. For more details about the thermocouple 5SRTC-GG-K-20-36 and IC DS600, see their datasheets in the following location:
`<Install_Directory>\PSoC Precision Analog EBK\<version>\Documentation\DataSheet`.

While connecting the thermocouple to the connector, make sure the markings in the thermocouple match those of the connector.

Figure 3-5. Thermocouple

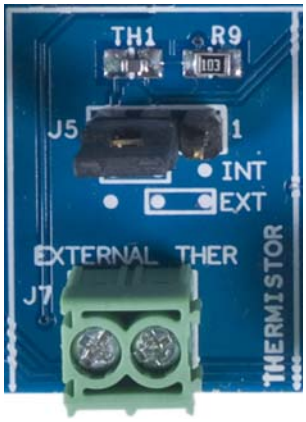


3.3.2 Thermistor

The board has a NCP18XH103F03RB (NTC) thermistor. It has a 1 percent tolerance (10 k ± 1%) at 25 °C. For more details about the sensor, see its datasheet in the following location:
`<Install_Directory>\PSoC Precision Analog EBK\<version>\Documentation\DataSheet`

You can also plug an external thermistor in terminal J7. The choice between using the internal and external thermistor is made using jumper J5. The jumper is inserted on header position 1–2 for using an external thermistor and on header position 2–3 for using the internal (onboard) thermistor. The silk indicates the jumper positions for internal and external thermistors.

Figure 3-6. Thermistor



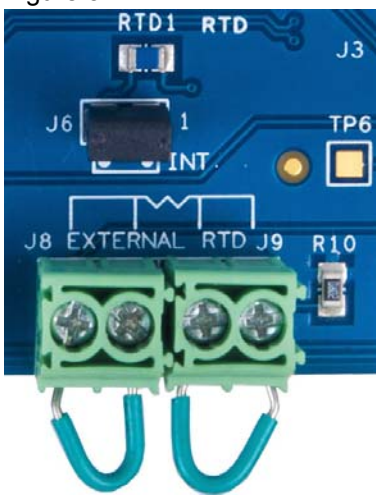
3.3.3 RTD

Platinum RTD PTS080501B100RP100, which is a class B PT-100 RTD, is used on the board. See the RTD datasheet in the following location for more details:
`<Install_Directory>\PSoC Precision Analog EBK\<version>\Documentation\DataSheet`.

You can connect any external four-wire RTD. Follow these instructions to use the internal and external RTDs.

- External RTD: Connect the external RTD to terminal blocks J8 and J9. The silk helps in the connection. Remove jumper J6.
- Internal (onboard) RTD: Place jumper J6. On jumpers J8 and J9, short 1–2. Figure 3-7 shows the connections for an internal RTD.

Figure 3-7. RTD



3.3.4 Diode

Terminal block J4 provides the option to connect up to two diode connected transistors. The diodes are connected anti-parallel (see Figure 4-4 on page 29) to each other so that two temperatures are measured with just four pins – two pins to pass current and two pins to measure V_{BE}.

J4 (1-4) is used to connect transistor 1. Connect the collector, base, and emitter to screws J4 1, 2, and 3, respectively. Use a small wire to short J4, 3–4. This is to support four-wire measurement so

that the series resistance error can be avoided. See the Diode section on page 28 for details. This wire also shorts the base and collector of transistor 2.

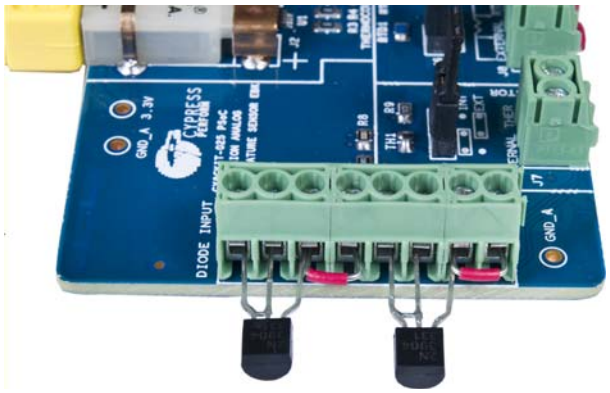
If you are installing only transistor 1, make sure you also have a wire between J4, 7–8 so that the base and collector of transistor 1 are shorted. Alternatively, you can short the base and collector directly on the transistor, as shown in Figure 3-8.

J4 (4–8) is used to connect transistor 2. Connect the collector, base, and emitter to screws J4 5, 6, and 7, respectively. Use a small wire to short J4, 7–8. This is to support four-wire measurement so that the series resistance error can be avoided. See the Diode section on page 28 for details. This wire also shorts the base and collector of transistor 1.

If you are installing only transistor 2, make sure you also have a wire between J4, 3–4 so that the base and collector of transistor 2 are shorted. Alternatively, you can short the base and collector directly on the transistor, as shown in Figure 3-8.

The 2N3904 transistor is shipped with the kit. See the datasheet in the following location for details: `<Install_Directory>\PSoC Precision Analog EBK\<version>\Documentation\DataSheet`.

Figure 3-8. Diode



3.4 Prototype Boards

The kit includes two CY8CKIT-012 prototype boards. The prototype boards plugs into any port of DVKs, CY8CKIT-001, CY8CKIT-030, and CY8CKIT-050 and they can be used for prototyping your custom circuit built around PSoC.

4. Hardware

4.1 System Block Diagram and Theory of System Operation

The schematic of the CY8CKIT-025 PSoC Precision Analog Temperature Sensor EBK is shown in the Appendix on page 83. The circuits associated with each temperature sensor are boxed individually. This section gives a description of each circuit and the temperature equations involved in each method.

4.2 Thermocouple

Cypress application note AN75511 - PSoC 3/PSoC 5LP Temperature Measurement with a Thermocouple explains the theory behind the thermocouple terminology; see the application note to understand thermocouple temperature measurement. This section assumes that you have prior knowledge of thermocouple terminology.

A thermocouple gives an output voltage that is directly related to the temperature difference between two metallic junctions. The output voltage is not linearly related to temperature. The output voltage versus temperature curve can be considered piece-wise linear with different slopes in different temperature ranges. The relation between hot junction voltage and cold junction voltage is given by the following equation.

$$V1 = KT + V2$$

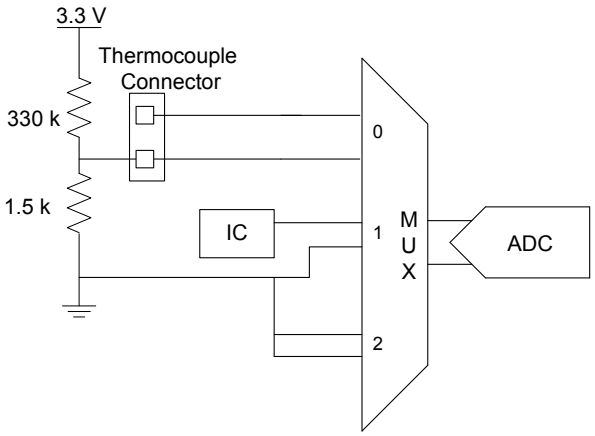
Equation 1

where the value of K depends on the temperature range. $V1$ is the voltage across the hot junction and $V2$ is the cold junction compensation (CJC) voltage (see AN75511 for details on cold junction compensation). T is the hot junction temperature.

Figure 4-1 shows the circuit used for thermocouple temperature measurement. A three-channel ADC is used.

- Channel 0 is used for thermocouple voltage measurement
- Channel 1 is used for cold junction compensation
- Channel 2 is used for offset cancellation

Figure 4-1. Thermocouple Circuit Diagram



The cold junction compensation IC measures the cold junction temperature and outputs a voltage proportional to the temperature. The ADC measures the cold junction output voltage to calculate the cold junction temperature. This temperature is used to calculate the cold junction compensation, $V2$.

$V1$ is measured in the first ADC channel. The temperature can be calculated using equation 1 if $V1$, $V2$, and K are known.

The lower potential lead of the thermocouple connector is biased at a potential above ground so that negative temperatures up to –270 °C can be measured after accommodating for small negative device offset. PSoC Creator provides a thermocouple component that supports all thermocouple types (B, E, J, K, N, R, S, and T) and simplifies voltage-to-temperature conversion and vice versa.

4.3 Thermistor

Application note AN66477 - PSoC 3 and PSoC 5LP Temperature Measurement with Thermistor explains the thermistor theory and temperature measurement with thermistor. This section assumes that you are aware of thermistor fundamentals and terminology.

The thermistor resistance changes with temperature in a nonlinear fashion. Thermistor manufacturers provide the thermistor temperature versus resistance table. The Steinhart-Hart equation (equation 2) characterizes thermistor resistance change with temperature to a good accuracy and can be used to find temperature from thermistor resistance.

$$\frac{1}{T} = A + B\ln R + C\ln(R)^3$$

Equation 2

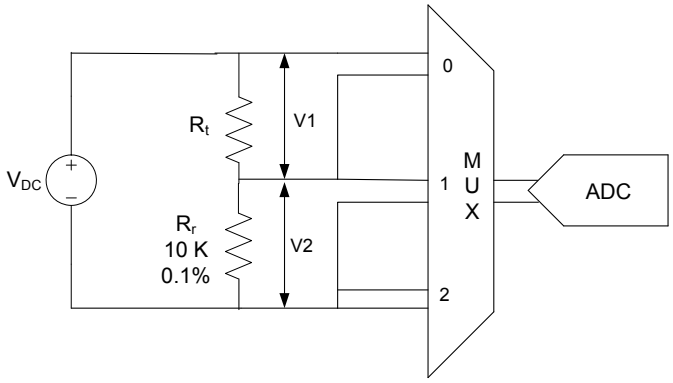
where,

- A, B, and C are the Steinhart-Hart coefficients

- R is the resistance at temperature T expressed in Kelvin

Figure 4-2 shows the circuit diagram used for the measurement. An excitation voltage is applied across a series combination of thermistor and precision reference resistor, and the voltages across thermistor and reference resistor are measured.

Figure 4-2. Thermistor Circuit Diagram



In Figure 4-2, $V1$ is the voltage across the thermistor resistance, R_t and $V2$ are the voltage across the reference resistance R_r. Equation 3 provides the thermistor resistance.

$$R_t = \frac{V1}{V2} * R_r$$

Equation 3

The voltage ratio ensures that any gain error and drift associated with the ADC are eliminated. The accuracy of the thermistor resistance, R_t, is determined mainly by the accuracy of the reference resistance, R_r. From R_t, the temperature can be calculated using the Steinhart-Hart equation. PSoC Creator provides a thermistor component for PSoC Creator, which makes it easier to interface a thermistor to PSoC 3 or PSoC 5LP. The component provides the flexibility of using either a lookup table (LUT) or the Steinhart-Hart equation and provides simple C functions to compute temperature from resistance.

4.4 RTD

Application note AN70698 - PSoC 3 and PSoC 5LP Temperature Measurement with RTDs explains the RTD fundamentals and RTD temperature measurement. This section assumes that you are aware of the RTD terminology.

In an RTD, the resistance variation with temperature is given by Callender Van Dusen equations (equations 4 and 5).

$$R_{RTD} = R_0(1 + AT + BT^2), T > 0$$

Equation 4

$$R_{RTD} = R_0(1 + AT + BT^2 + C(T - 100)T^2), T < 0$$

Equation 5

where,

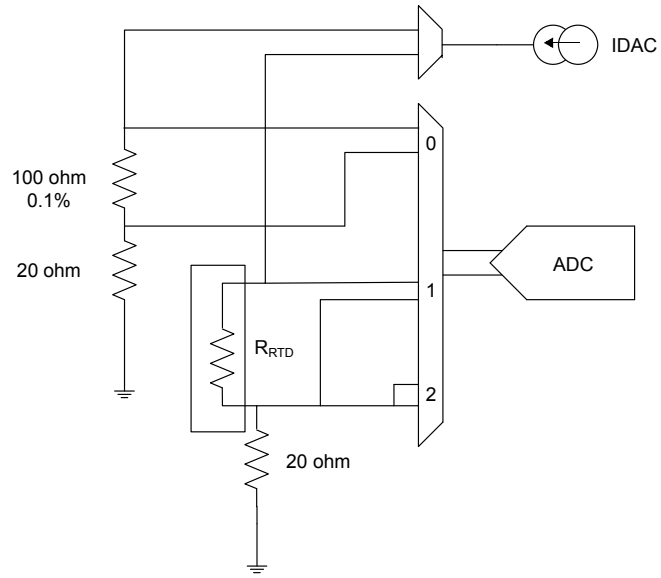
- R_{RTD} is the resistance at temperature T expressed in °C

- R_0 is the resistance at 0 °C (equal to 100 Ω for PT100 RTD, one of the commonly used RTDs)

- A, B, and C are Callender Van Dusen coefficients

By determining the resistance of the RTD, you can find the temperature using equations 4 or 5. Figure 4-3 shows the circuit used to measure RTD resistance.

Figure 4-3. RTD Circuit Diagram



One of the PSoC 3 or PSoC 5LP IDACs is used to force a current through the RTD, and the delta sigma ADC is used to measure the voltage across the RTD. The RTD resistance (V/I) is calculated and hence the temperature is obtained. Precise 100-Ω (0.1%) reference resistance is used to accurately measure the current forced using IDAC, thereby eliminating any gain error and drift associated with the ADC and IDAC. First the IDAC current is passed through the reference resistor (Rref) and the voltage across the reference resistor (Vref) is measured using ADC. The same current is then passed through the RTD and the voltage across RTD (Vrtd) is also measured using ADC. The RTD resistance is calculated using the equation:

$$R_{RTD} = \frac{Vrtd}{(Vref/Rref\ )}$$ Equation 6

PSoC Creator provides an RTD component that helps to convert resistance to temperature.

4.5 Diode

Application note AN60590 - PSoC 3 and PSoC 5LP Temperature Measurement using Diode explains the theory behind diode temperature measurement. This section provides only a short description of the method. The current through a forward-biased diode is given by the equation:

$$I = I_s EXP\left(\frac{v}{\eta V_T}\right)$$

Equation 7

where,

V is the diode forward voltage drop

I_S is the reverse saturation current

η is the constant that has a value between 1 and 2, depending on the material and physical structure of the diode

V_T is the thermal voltage calculated with the equation

$$V_T = \frac{kT}{q}$$

where,

k is the Boltzmann's constant

T is the absolute temperature in Kelvin

q is the magnitude of electron charge

By passing two currents I_1 and I_2 and measuring the respective voltages V_1 and V_2, the temperature can be calculated using the following equation.

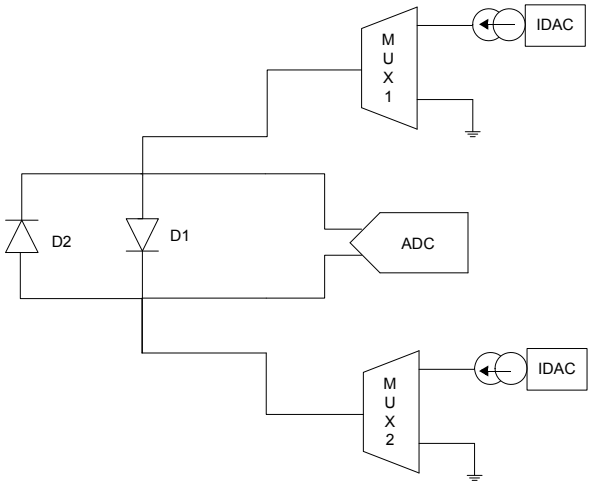
$$T = \frac{q}{\eta k}\left(\frac{V_1 - V_2}{\ln\left(I_1 / I_2\right)}\right)$$

Equation 8

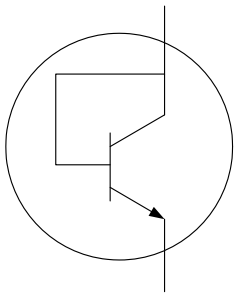
Figure 4-4 shows the circuit diagram used for diode temperature measurement. Note that two diodes are connected antiparallel to each other. This helps to measure two temperatures using only four pins.

Figure 4-4. Diode Circuit Diagram and Diode Connected Transistor

(a) Diode Circuit Diagram



(b) Diode Connected Transistor



In Figure 4-4 (a), D1 and D2 are diode connected transistors (See Figure 4-4 (b)). While measuring the temperature using diode D1, MUX1 channel1 and MUX2 channel2 are active. While measuring the temperature using diode D2, MUX1 channel2 and MUX2 channel1 are active.

By passing two currents I_1 and I_2 and measuring the respective voltages V_1 and V_2, the temperature is calculated using equation 8. The ADC measures a negative voltage in one of the two cases, which can be negated in firmware to make it positive. Measurements are made in four-wire configuration to avoid series resistance error.

5. Example Projects

5.1 Overview

Example projects are divided into three project areas based on end applications. Each project area contains examples for the CY8CKIT-030, CY8CKIT-050, and CY8CKIT-001 development kits (DVK). The CY8CKIT-030 DVK contains a PSoC 3 (8051 core) and CY8CKIT-050 DVK contains a PSoC 5LP (ARM Cortex-M3); both support high-performance analog applications. The CY8CKIT-001 DVK can be used to develop solutions for the PSoC 1 (M8C proprietary core), PSoC 3 (8051 core), or PSoC 5LP (ARM Cortex-M3). The example projects provided with the kit support PSoC 3 (8051) and PSoC 5LP (ARM Cortex-M3).

The Tempsense example project supports most temperature sensing applications. It provides high-performance temperature sensing by using the ADC in 20-bit mode and implementing a high-precision linearization algorithm. It includes a resistive temperature detector (RTD), thermocouple, thermistor, two temperature diodes, and an IC temperature sensor. You can evaluate and compare sensor performance. Complete interface and output signal correction solutions can be developed for each type of temperature sensor.

The Sequenced ADC example project includes three temperatures (RTD, temperature diode, and IC temperature sensor) as well as millivolt and voltage inputs. This example project can be used for sensing applications requiring temperature measurement (such as RTD) and other types of voltage output sensors. The example is also useful in system monitoring applications requiring temperature (usually temperature diode or IC temperature sensor) and voltage rail measurement. This example project uses the ADC in 16-bit mode and provides higher throughput. In this project, the ADC readings are automatically repeated at consistent intervals. This consistent time interval allows filtering of ADC readings to remove specific frequencies of interest, such as 50- and/or 60-Hz hum.

The Thermal Management System supports fan control applications. Adding a CY8CKIT-036 evaluation kit (with two four-wire fans) to the CY8CKIT-025 EBK and a development kit supports complete temperature-based fan control solutions. For more details on the CY8CKIT-036 PSoC Thermal Management EBK, click here.

Example Projects:

- Tempsense - Measures the temperature of six temperature sensors
  - TempSense_030_050 - Works with PSoC 3 DVK CY8CKIT-030 and PSoC 5LP DVK CY8CKIT-050
  - TempSense_001 - Works with PSoC DVK CY8CKIT-001
- Sequenced ADC - Measures three temperatures, two voltage rails, and two generic analog inputs
  - SequencedADC_030_050 - Works with PSoC 3 DVK CY8CKIT-030 and PSoC 5LP DVK CY8CKIT-050
  - SequencedADC_001 - Works with PSoC DVK CY8CKIT-001
- Thermal Management System - Controls fan speed based on temperature
  - Thermal Management System_030_050 - Works with PSoC 3 DVK CY8CKIT-030 and PSoC 5LP DVK CY8CKIT-050
  - Thermal Management System_001 - Works with PSoC DVK CY8CKIT-001

5.1.1 Migrating Projects to use with CY8CKIT-050 and CY8CKIT-010

By default, the project works with PSoC 3. To use the project for PSoC 5LP, change the device to CY8C5868AXI-LP035 (**Project > Device Selector**). If the CY8CKIT-001 project is being updated to be used with the PSoC 5LP processor module, you must also change the CMOD pin mapping from P2_7 to pin 15_5.

To print floating point variables on the LCD, set the 'Use newlib-nano Float Formatting' parameter to **True** under **Project > Build Settings > Linker** as shown in Figure 5-1 and change the **Heap Size** to **0x200** in the **System** tab in the *<project>.cydwr* file as shown in Figure 5-2.

Figure 5-1. Build Settings

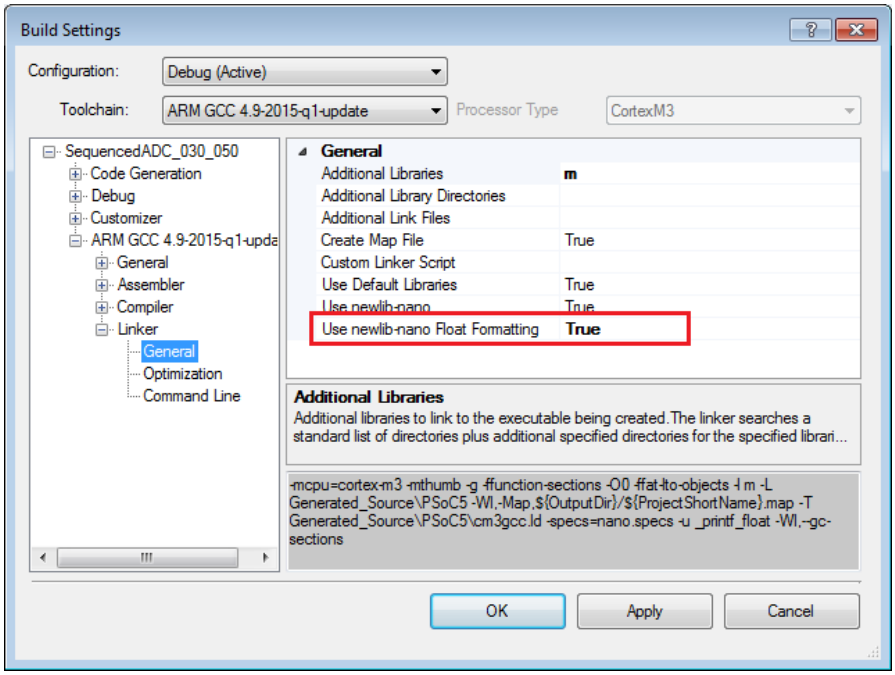
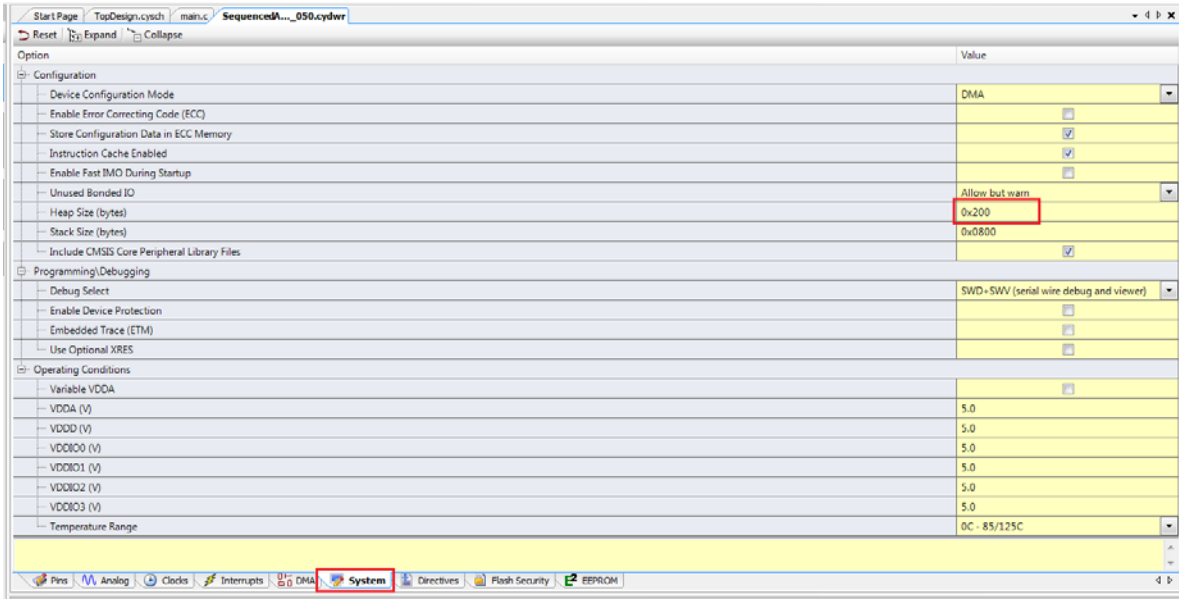


Figure 5-2. System Tab



The following sections provide a detailed description of these example projects.

5.2 Tempsense

This example project demonstrates high-precision, high-accuracy, multi-channel temperature sensing using PSoC.

5.2.1 Project Description

This project displays the temperature measured using a thermocouple, IC sensor, thermistor, RTD, and two diodes. It also provides steps to calibrate the RTD, diode, and IC. The user interface of this example is controlled by two CapSense® buttons, a CapSense slider, two mechanical push buttons, and a character LCD.

When the example project for CY8CKIT-030 is programmed, the LCD displays a welcome message for one second. It then displays the temperature from the thermocouple, which is the first item in the user selectable menu. CapSense buttons P5_5 and P5_6 act as the scroll down and scroll up menu buttons, respectively. A mechanical switch connected to P6_1 is used to proceed through various calibration steps. A CapSense slider is used to set the calibration temperature.

The menu has the following items:

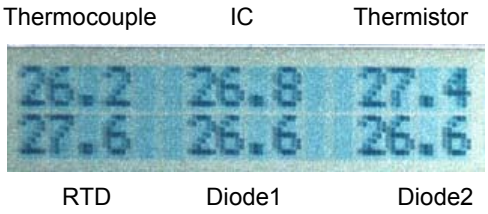
- Thermocouple temperature display
- IC temperature display
- Thermistor temperature display
- RTD temperature display
- Diode1 temperature display
- Diode2 temperature display
- Simultaneous temperature display labels (see Figure 5-3)
- Simultaneous temperature display from all 6 sensors (see Figure 5-4)
- Calibrate RTD temperature
- Calibrate Diode1 temperature
- Calibrate Diode2 temperature
- Calibrate IC sensor temperature

In the simultaneous display, the first row displays the temperature of the thermocouple, IC, and thermistor. The second row displays the temperature of the RTD, diode 1, and diode2, as shown in Figure 5-4.

Figure 5-3. Simultaneous Temperature Display Labels



Figure 5-4. Simultaneous Temperature Display for All Six Sensors



5.2.2 Project Operation

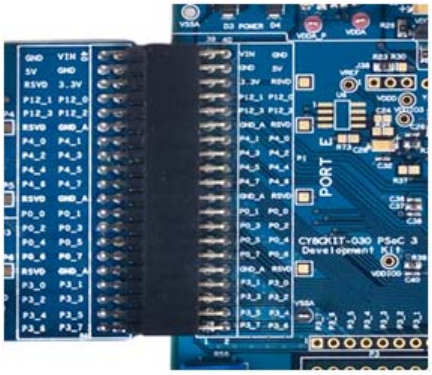
5.2.2.1 Hardware Connections

The kit includes example projects for the CY8CKIT-001 DVK, CY8CKIT-030 DVK, and CY8CKIT-050 DVK hardware platforms. The pin mapping for CY8CKIT-030 and CY8CKIT-050 are identical; follow the instructions provided for CY8CKIT-030 PSoC 3 DVK for CY8CKIT-050 PSoC 5LP DVK as well.

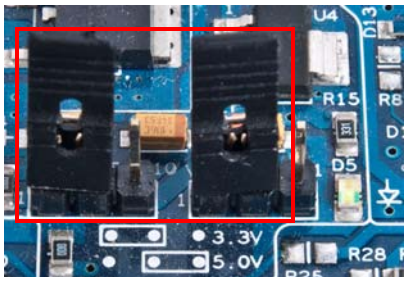
The main difference between the examples for the CY8CKIT-001 and CY8CKIT-030 platforms is the PSoC pin mapping. The project provided for the CY8CKIT-001 PSoC DVK can be used with both PSoC 3 and PSoC 5LP. See Migrating Projects to use with CY8CKIT-050 and CY8CKIT-010 on page 31 for more details.

5.2.2.1.1 CY8CKIT-030 PSoC DVK

1. Plug the PSoC Precision Analog Temperature Sensor EBK to port E of the CY8CKIT-030 DVK.



2. No jumper wires are required for the PSoC 3 DVK examples because the mechanical buttons are hard-wired to GPIOs. Ensure that the LCD included with the PSoC 3 DVK is attached.
3. Set VDDD and VDDA to 3.3 V using J10 and J11.

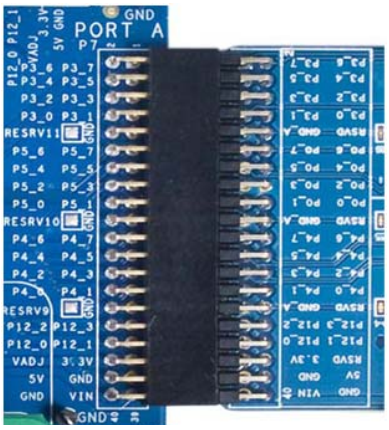


4. In the CY8CKIT-025 EBK, choose appropriate jumper settings for internal sensors; see the Kit Operation chapter on page 18.

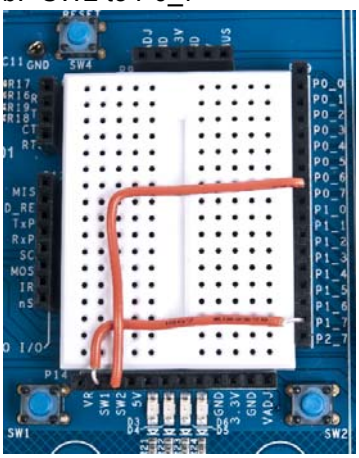
Note: When using the CY8CKIT-025 EBK with CY8CKIT-050 PSoC 5LP DVK, remove the jumpers from J43 and J44 on PSoC 5LP DVK.

5.2.2.1.2 CY8CKIT-001 PSoC DVK

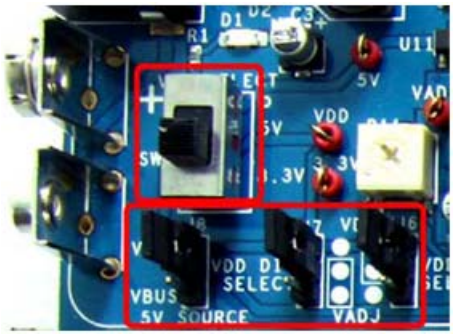
1. Plug the PSoC Precision Analog Temperature Sensor EBK to port A of the CY8CKIT-001 DVK.



2. In the pin header/breadboard area of the CY8CKIT-001 base board, use jumper wires to make the following connections:
   a. SW1 to P1_7
   b. SW2 to P0_7



3. Set the system to run at 3.3 V using SW3; set J6 VDD DIG and J7 VDD ANLG to VDD = 3.3 V.



4. Ensure that the LCD included with the PSoC DVK is attached and the LCD power jumper (J12) is in the ON position.



5. In the CY8CKIT-025 EBK, choose appropriate jumper settings for internal sensors; see the Kit Operation chapter on page 18.

5.2.2.2 *Run Tempsense Example Firmware*

5.2.2.2.1 *CY8CKIT-030 PSoC 3 DVK*

The steps to run the example firmware on CY8CKIT-030 are:

1. Attach a USB cable from the PC to the PSoC 3 DVK program/debug USB port (use J1, the USB connector closest to the corner of the board).
2. Open the Tempsense_030_050 project using PSoC Creator and select **Debug > Program** to program the PSoC.
3. Detach the USB cable and connect hardware according to Hardware Connections on page 33.
4. Attach the USB cable again from the PC to the PSoC 3 DVK program/debug USB port (use J1).
5. Press the **Reset (SW1)** button on the PSoC 3 DVK to run the newly programmed firmware image.
6. On reset, the LCD displays a welcome message and the temperature reading from the thermocouple, which is the first item in the user selectable menu. SW2 can be used to toggle the cold junction compensation sensor between the IC (DS600) and thermistor.
7. Press CapSense button **P5_5** to scroll down the menu and **P5_6** to scroll up the menu.
8. Use SW2 to proceed through the calibration steps; see RTD Calibration on page 47.
9. Use SW3 to return to the previous step during calibration.

5.2.2.2.2 *CY8CKIT-001 PSoC DVK*

The steps to run the example firmware on CY8CKIT-001 are:

1. Apply 12 VDC power to the PSoC DVK.
2. Attach the MiniProg3 device, first to a USB port on the PC and then to the PROG port on the CY8CKIT-009 PSoC 3/CY8CKIT-010 PSoC 5LP processor module.
3. Open the Tempsense_001 project using PSoC Creator and select **Debug > Program** to program the PSoC.
4. Disconnect power to the PSoC DVK and connect hardware according to Hardware Connections on page 33.
5. Power the PSoC DVK.
6. On reset, the LCD displays a welcome message and the temperature reading from the thermocouple, which is the first item in the user selectable menu. SW1 can be used to toggle the cold junction compensation sensor between the IC (DS600) and thermistor

7. Press CapSense button **P0_5** to scroll down the menu and **P0_6** to scroll up the menu.
8. Use SW1 to proceed through calibration steps; see RTD Calibration on page 47.
9. Use SW2 to return to the previous step during calibration.

5.2.2.3 *Testing the Project*

For best results when testing temperature measurements, ensure that you have good thermal contact between the sensor and your reference thermometer. Also, make sure there is minimal air flow, which can result in temperature gradient. Sample test results are given in the following sections.

5.2.2.4 *Expected Performance and Test Results*

This section explains the expected resolution and accuracy of the temperature displayed by all the sensors.

5.2.2.4.1 *Accuracy*

The accuracy of the temperature displayed depends on the sensor and the signal chain.

Sensor

Each sensor has a tolerance, which is available in the datasheet provided by the sensor manufacturer. The tolerance indicates the error due to replacement of one sensor with another of the same part number. The tolerance of each onboard sensor is listed in Table 5-1. These numbers are from the respective sensor datasheets.

Table 5-1. Temperature Sensor Tolerances

Sensor	Tolerance Standard	Maximum Tolerance Error at 25 °C	Maximum Tolerance Error at 100°C
Thermocouple 5SRTC-GG-K-20-36	ASTM special limits of error 1.1 °C or 0.4%, whichever is higher	± 1.1 °C	± 1.1 °C
Thermistor NCP18XH103F03RB	1% at 25 °C 1% B-value tolerance	± 0.4 °C	± 2 °C
RTD PTS080501B100RP100	IEC Class B RTD (0.30 + 0.005T) °C	± 0.43 °C	± 0.8 °C
Diode	–	–	–
IC	–	± 0.5 °C	± 0.5 °C

For diode, the ideality factor is the source of its tolerance. This number is generally not mentioned in the diode datasheet and must be determined experimentally. Diode ideality factor generally lies between 1 and 1.01. An ideality factor difference of 0.004 (using 1.008 instead of 1.004) causes an error of around 2 °C. See AN60590 - PSoC 3 and PSoC 5LP - Temperature Measurement with a Diode for more details on these calculations.

Signal Chain

Signal chain error includes the errors due to the ADC, IDAC, the calibration (reference) resistors and the measurement method. The resources used differ from sensor to sensor.

The main sources of ADC error are its offset, gain error, and INL. Of these, the offset error is cancelled in all cases by correlated double sampling. ADC gain error is eliminated by using a precision reference resistor in thermistor and RTD.

The main sources of IDAC error are its offset, gain error, and INL. The IDAC errors are completely removed by passing the IDAC current through a precise reference resistor and measuring the voltage across it.

The reference resistor tolerance determines the error caused by it. A 0.1% reference resistance is used in both RTD and thermistor temperature measurements.

The error component caused by the signal chain for each sensor is listed in Table 5-2.

Table 5-2. Signal Chain Errors for PSoC 3

Sensor	Signal Chain Error Components	Typ/Max Error at 25 °C	Max Error at 100 °C
Thermocouple	ADC gain error, INL, cold junction IC sensor error (± 0.5 °C)	0.1 / 2.1	2.2
Thermistor	ADC INL, reference resistor tolerance	0 / 0.05	0.6
RTD	ADC INL, reference resistor tolerance	0.1 / 0.5	0.9
Diode	ADC gain error, INL	0.3 / 2.6	3.8

Cypress application notes on each of the four sensors explain the signal chain errors in detail.

5.2.2.4.2 *Resolution (peak to peak)*

The temperature measurement resolution is the minimum temperature increment possible. A 0.1 °C resolution is achieved on all sensors. A software IIR filter (see Sensor Output Filter on page 51) is applied on some sensors to achieve 0.1 °C resolution. The application notes associated with each sensor gives more details on how 0.1 °C resolution is achieved on each sensor. Generally, a small flicker is seen on the LCD temperature display. A small flicker of 0.2 °C on thermocouple readings, 0 °C on IC and thermistor, 0.1 °C on RTD, and 0.3 °C on diode at a very small rate has been observed in the example projects. It can be reduced by additional filtering at the cost of temperature settling time. The IIR filter can be changed in firmware, as shown in Sensor Output Filter on page 51.

5.2.2.4.3 *Test Results*

This section shows the test results on the sample board. Wherever possible, measurement techniques are used to identify the signal chain performance of PSoC.

Thermocouple

Thermocouple input can be simulated by a millivolt source. This eliminates any error due to thermocouple tolerance and allows you to test the signal chain in the whole temperature range without a heat source. A millivolt source is fed to the thermocouple inputs and the cold junction temperature is forced to 0 °C in the example project. The temperature shown by PSoC is noted and the voltage input is measured using an accurate voltmeter; the temperature corresponding to that voltage is calculated using NIST tables. The expected results and actual results are listed in Table 5-3.

Table 5-3. Thermocouple Test Results

Simulated thermo-emf (mV)	Voltage read by multimeter (mV)	Expected temperature (°C)	Obtained Temperature (°C)	Error (°C)
–5	–4.695	–141	–141.1	0.1
–4	–3.666	–103.7	–103.7	0
–3	–2.575	–69.6	–69.5	–0.1
–2	–1.741	–45.9	–45.7	–0.2
–1	–0.654	–16.77	–16.7	–0.1
0	0	0	0	0

Table 5-3. Thermocouple Test Results

Simulated thermo-emf (mV)	Voltage read by multimeter (mV)	Expected temperature (°C)	Obtained Temperature (°C)	Error (°C)
1	0.666	16.7	16.7	0
2	1.754	43.46	43.4	0.1
3	2.58	63.5	63.5	0
4	3.663	89.5	89.4	0.1
5	4.712	114.9	114.9	0
10	9.582	235.9	235.8	0.1
20	19.58	475	475	0
30	28.76	691.2	691.1	0.1
40	39.54	955.75	955.8	–0.1
50	49.816	1227	1226.9	0.1

Thermistor

A thermistor can be simulated by a potentiometer. By varying the potentiometer resistance, you can simulate temperatures in the whole temperature range of the thermistor. A potentiometer resistance is used to provide resistance in the whole thermistor temperature range; the thermistor temperature measured by PSoC is noted. The resistance is also measured by an accurate multimeter and the expected temperature is calculated using the Steinhart-Hart equation. Table 5-4 shows the results.

Table 5-4. Thermistor Test Results

Resistance in ohms (Multimeter Measurement)	Expected Temperature	Measured Temperature (°C)	Signal Chain Error (°C)
99318.0	–27.6	–27.60	0.00
32593.0	–4.0	–4.1	–0.10
12135.0	19.9	19.9	0.00
10000.0	25.0	25	0.00
6951.0	35.0	35	0.00
4230.0	49.5	49.6	0.10
3021.0	60.0	60	0.00
2065.0	72.6	72.7	0.10
1260.0	90.2	90.2	0.00
977.6	99.9	99.9	0.00
800.8	107.8	107.8	0.00
549.5	123.5	123.6	0.10
419.3	135.6	135.5	–0.10
308.3	150.1	150	–0.10
270.0	156.6	156.5	–0.10
189.3	175.1	175	–0.10
122.7	199.7	199.5	–0.20

RTD

An RTD can be simulated by a potentiometer. By varying the potentiometer resistance, you can simulate the temperature in the whole temperature range of the RTD. A potentiometer is used to provide resistances in the whole RTD temperature range; the RTD temperature measured by PSoC is noted.

The resistance is also measured by an accurate multimeter and the expected temperature is calculated using Callender Van Dusen equation. Table 5-5 shows the results .

Table 5-5. RTD Test Results

Resistance in Ohms (Multimeter Measurement)	Expected Temperature (°C)	Measured Temperature (°C)	Temp Measure Error (Signal Chain) (°C)
79.85	–51.1	–51.3	–0.20
100.06	0.2	0.2	0.00
110.76	27.6	27.7	0.10
120.25	52.2	52.1	–0.10
133	85.5	85.3	–0.20
138.5	100.0	100	0.00
150	130.4	130.3	–0.10
194.4	250.8	251	0.20
280.85	499.8	500	0.20
315.7	606.2	606.2	0.00
371.8	787.0	787.3	0.30

Diode

Diode temperature accuracy is tested in a temperature forcing system by forcing a specific temperature and comparing the diode results with those of a standard thermometer. The standard microthermal thermocouple thermometer, with an accuracy of 1.3 °C (including thermocouple probe error), is used. Table 5-6 shows the results.

Table 5-6. Diode Test Results

Temperature Measured using Standard Thermometer	Diode Measured Temperature (with PSoC) (°C)	Diode Temperature Error (°C)
0.8	1.2	0.4
6.2	5.6	–0.6
17	16.1	–0.9
22	21	–1
30	29.3	–0.7
75.3	74.7	–0.6
99.5	99.4	–0.1

DS600 IC Sensor

The IC sensor signal chain accuracy is tested by measuring the voltage output by the IC at various temperatures and calculating the expected temperature and comparing it to the temperature displayed by PSoC. Sample test results at a few temperature points are shown in Table 5-7.

Table 5-7. IC Sensor Test Results

IC Output Voltage (mV)	Expected Temperature (°C)	Observed Temperature (°C)	Error (°C)
640.32	20.4	20.4	0.0
660.34	23.5	23.5	0.0
702.22	30.0	30	0.0
733.73	34.8	34.8	0.0

5.2.2.4.4 Simultaneous Temperature Display of all Six Sensors

The sensor-to-sensor variation in temperature is 0.5 °C to 1.1 °C without calibration. The signal chain can add additional error. Remember the following thermal considerations to understand the temperature difference between the sensors in the simultaneous display of six temperatures.

- The thermocouple measures the air temperature; the RTD and thermistor measure the board temperature; the IC measures the temperature of the exposed pad below it; and diodes are screwed on to the board although they hang in the air.
- Because the thermocouple measures the air temperature, it is about a degree Celsius lower than the board at room temperature. Place the tip of the thermocouple on the board and you will see the thermocouple temperature rise by a degree.
- Place the board in the sun; you will note that the temperature display shown by the onboard sensors slowly rise. In tropical regions, you can see the board temperature rise to greater than 50 °C, while the thermocouple temperature will be around 35 °C to 37 °C.
- The IC sensor has a thermal pad underneath it. Ideally, it is expected to be isothermal with the cold junction. Ensuring an isothermal connection requires providing good thermal contact between the thermocouple cold junction and the exposed IC pad. If the thermal contact is not provided, having the cold junction IC very close to the cold junction terminals is sufficient. The temperature difference between the cold junction terminals and the IC pad will not be more than 0.5 °C. Most thermocouple applications do not demand high accuracy and it is sufficient to have the cold junction IC very close to the cold junction. The IC temperature is about 0.3 °C to 0.5 °C lower than the RTD and thermistor temperature because the exposed pad underneath the IC is connected to ground. This is an offset error. This offset error can be eliminated by performing an offset correction, as shown in IC Calibration on page 50.

Apart from the above thermal considerations, the sensor and bias resistor tolerances should determine the sensor temperature. Typically, RTD and thermistor temperatures are within about 0.3 °C of each other; IC temperature and diode temperature is within about 0.5°C lower than the thermistor temperature; and thermocouple temperature is about 1 °C to 2 °C lower than the thermistor temperature.

5.2.2.4.5 CY8CKIT-001 PSoC DVK Thermistor Temperature

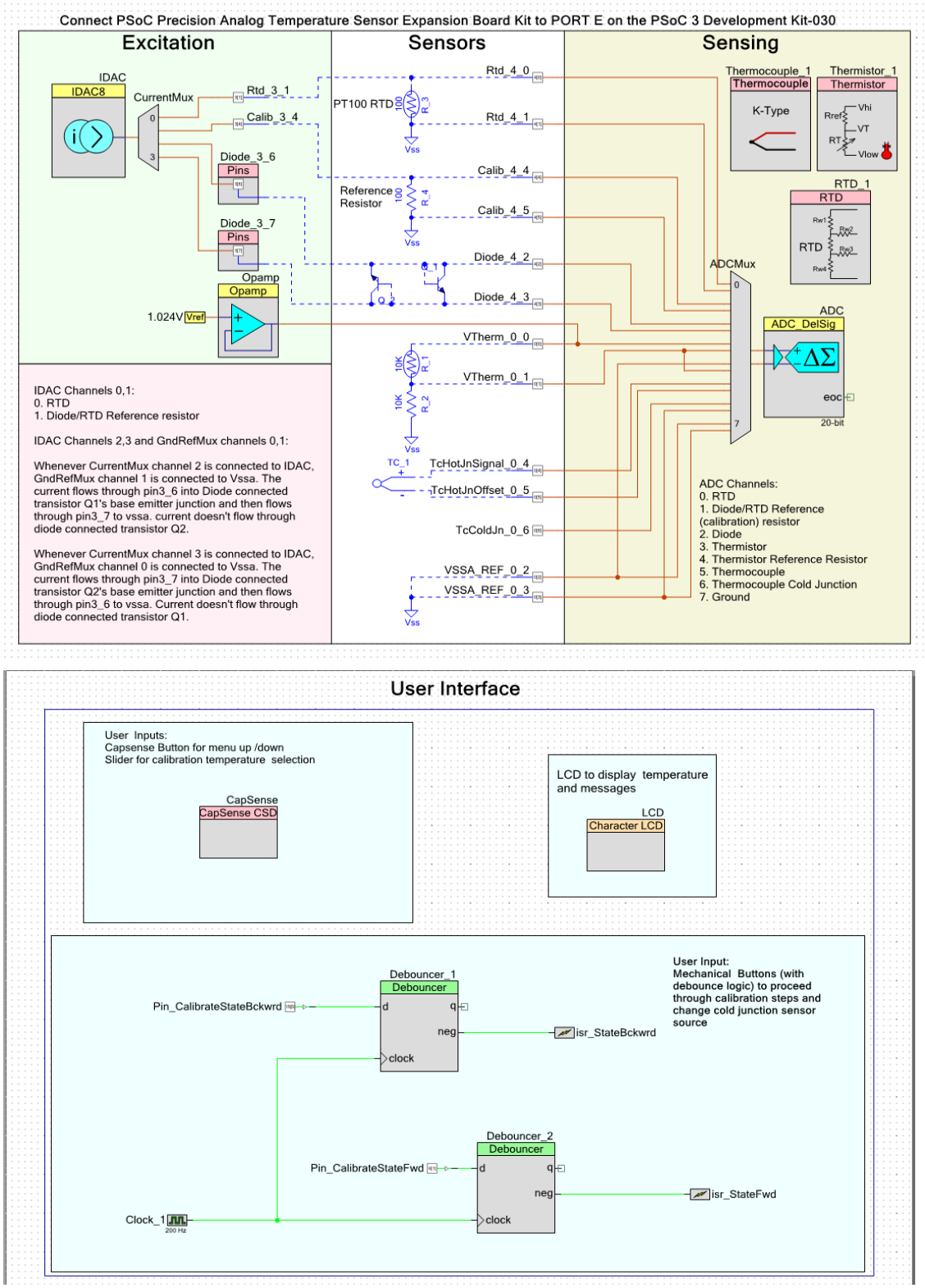
The CY8CKIT-025 connects to the Port A connector on the CY8CKIT-001. Through this connection, the resistor/thermistor network connects to pin 5_0. On PSoC port pins 0_0, 0_1, 3_6, and 3_7 connect directly to the opamp output. Thus, there is no direct connection from an opamp to the thermistor/resistor network, which is connected to pin 5_0. Therefore, an unbuffered VDAC is used to supply the required voltage for the thermistor/resistor network. Due to internal routing this connection has some parasitic resistance, which causes a voltage drop. The ADC measures this voltage drop and compensates. However, thermistor measurement on CY8CKIT-001 may result in higher error up to 0.5 °C. For details on internal analog routing, see AN58827 - PSoC 3 and PSoC 5LP Internal Analog Routing Considerations.

5.2.3 Project Details

5.2.3.1 Project Schematic

Figure 5-5 shows the PSoC Creator schematic of the project. An eight-channel ADC is used to sense the six sensors. The IDAC has four channels to provide excitation to the RTD and the diodes. Whenever IDAC channel 2 is connected, the IDAC output flows through the diode connected transistor Q1, Pin 3_7 and to ground. The ground connection is provided on the pin in firmware. PSoC's pin structure allows easy connection to the ground.

Figure 5-5. PSoC Creator Top Design

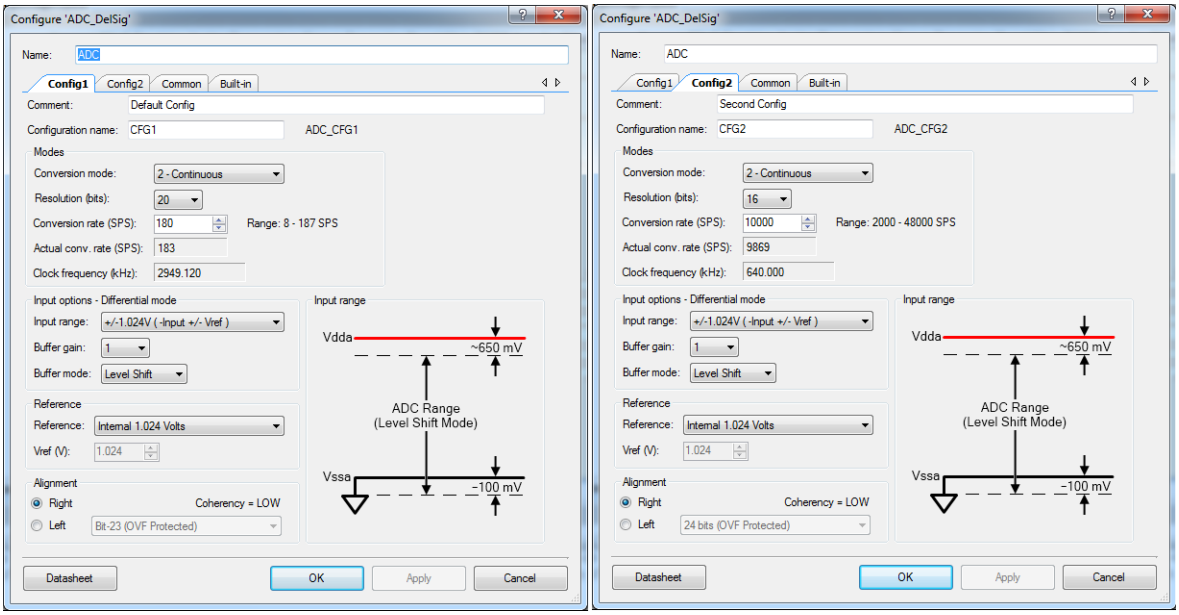


5.2.3.2 Component Configuration

5.2.3.2.1 ADC Configuration Settings

The ADC is configured in 20-bit mode for all sensors except the thermistor, where it is configured in 16-bit mode. Figure 5-6 shows the configuration used.

Figure 5-6. ADC Configuration

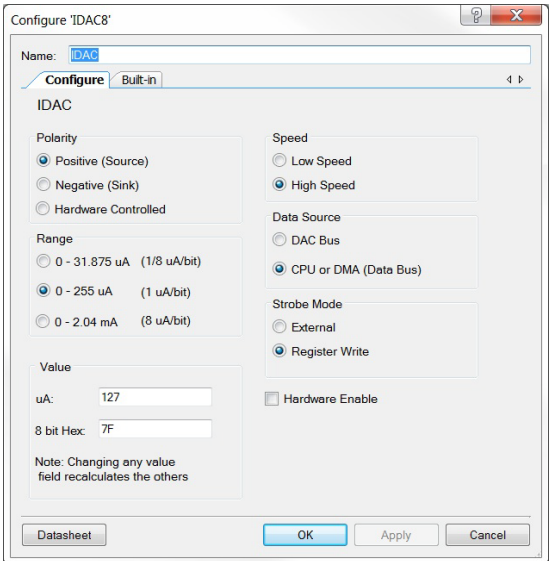


The ADC Buffer Mode is set to **Level Shift** because many of the voltages being read are near VSS; this setting yields better ADC results for signals near VSS.

5.2.3.2.2 IDAC Configuration

The IDAC is configured in the 0 to 255 µA-range. The IDAC is used for RTD and diode temperature measurements. The IDAC data register value is modified in firmware, as required for each of these temperature measurements.

Figure 5-7. IDAC Configuration



5.2.3.3 *Firmware Description and Flowchart*

The temperature calculation flowchart for each sensor and the flowchart for calibrating RTD and diode are explained in this section. A single point calibration is provided for RTD and diode.

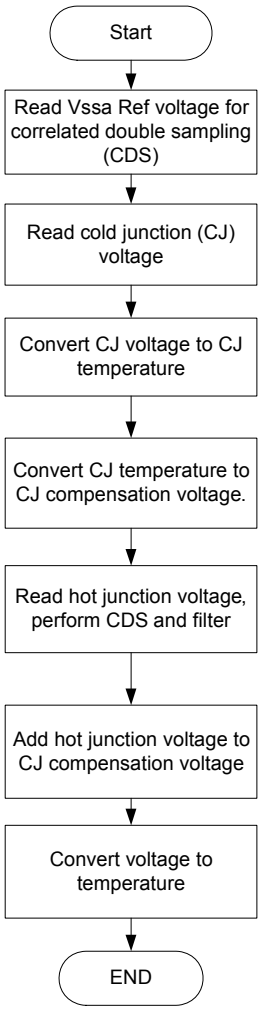
5.2.3.3.1 *Thermocouple*

Figure 5-8 illustrates the steps involved in thermocouple temperature measurement.

This flowchart involves conversion of voltage to temperature and vice versa. The conversion is done using thermocouple calculator component, which in turn uses the polynomial provided by NIST. The mathematical conversion results in an error less than 0.07 °C.

The example project uses a K-type thermocouple. The component supports all thermocouple types and provides application programming interfaces (APIs) for voltage to temperature conversion and vice versa. IC (DS600) or thermistor can be used for cold junction compensation.

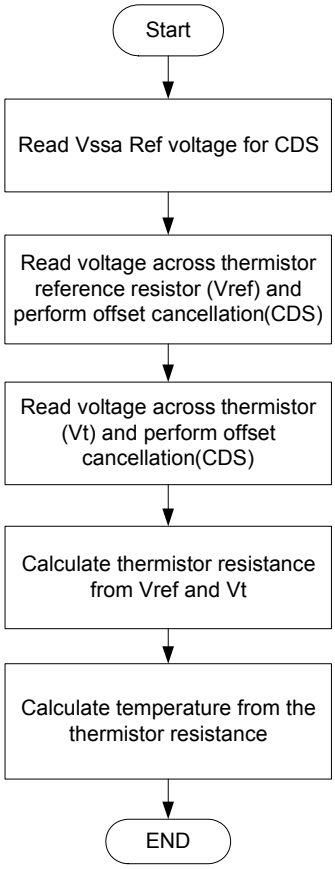
Figure 5-8. Thermocouple Flowchart



5.2.3.3.2 *Thermistor*

Figure 5-9 shows the thermistor temperature calculation flowchart.

Figure 5-9. Thermistor Flowchart

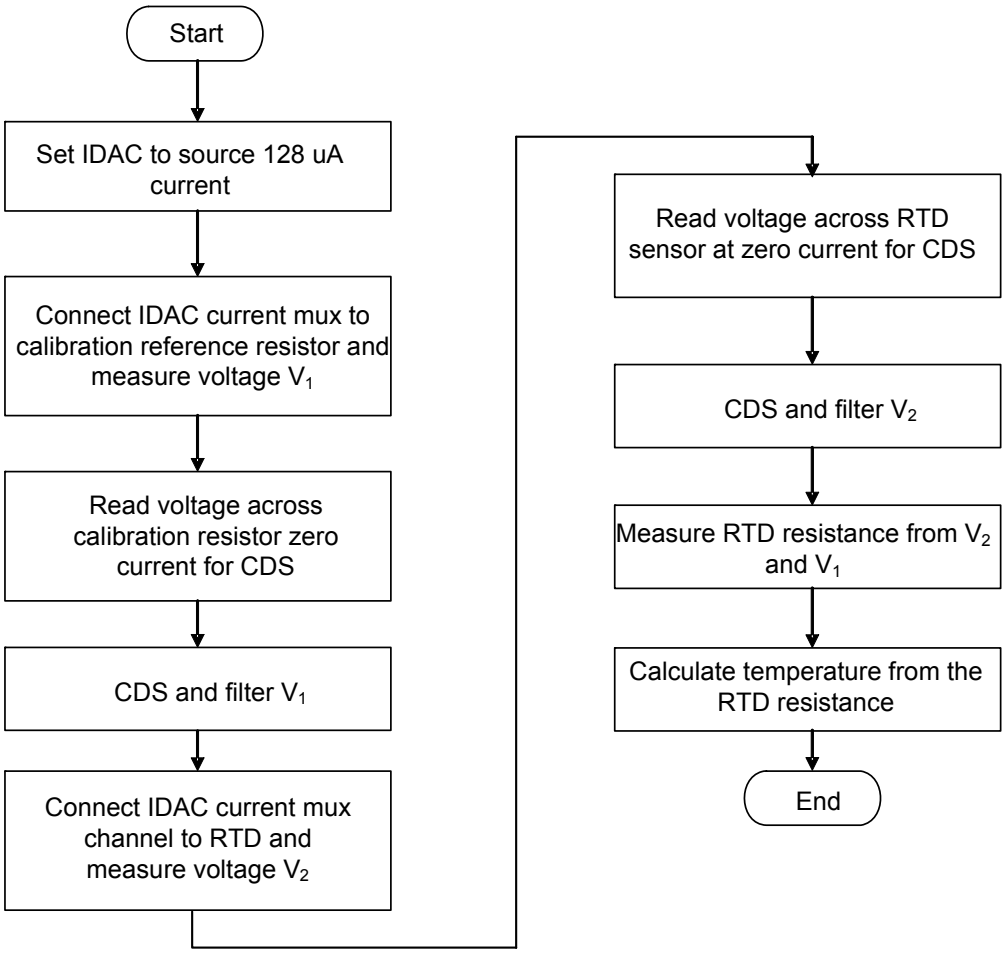


The flowchart involves conversion of thermistor resistance to temperature. This is done using the thermistor calculator component. The component supports both LUT and the Steinhart-Hart equation, and provides an API for converting resistance to temperature.

5.2.3.3.3 RTD

Figure 5-10 shows the RTD temperature calculation flowchart.

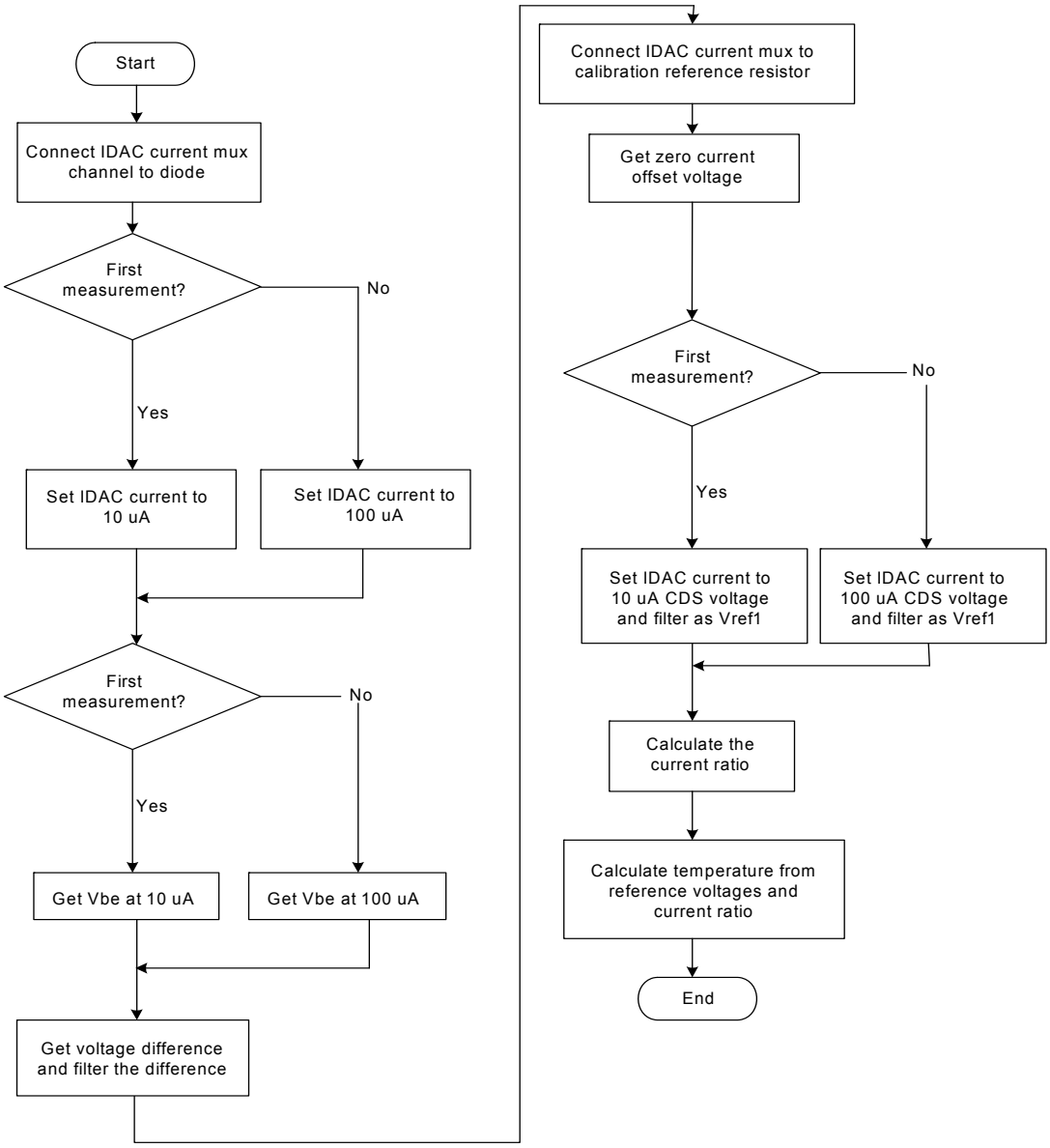
Figure 5-10. RTD Flowchart



The flowchart involves converting RTD resistance to temperature. The RTD calculator component is used for this conversion. The RTD component chooses the correct polynomial order based on your accuracy requirements.

5.2.3.3.4 Diode

Figure 5-11. Diode Flowchart



5.2.3.3.5 RTD Calibration

The RTD provided onboard confirms to IEC-70651 class B standards. A class-B RTD conforms to the following tolerance specifications.

Tolerance = 0.30 + 0.005|t|

This means replacing one RTD with another of the same part number causes a maximum error of 0.3 °C at 0 °C and 0.8 °C at 100 °C. Calibrating the RTD can help avoid this error. AN70698 describes the calibration details.

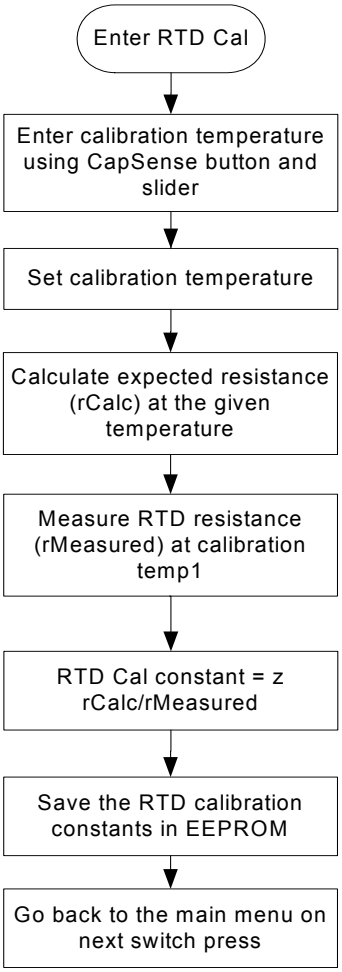
The calibration steps for the RTD are as follows:

1. Scroll down to the **Calibrate RTD** (eighth) item in the menu using CapSense buttons.

2. Proceed through calibration steps using mechanical switch SW2 connected to P6[1] of CY8CKIT-030 or SW1 connected to P1[7] of CY8CKIT-001.
3. The LCD displays "Set Temp:".

   Force the RTD temperature to a known measured value between 0 °C and 100 °C and enter that temperature using the CapSense slider and buttons. Use the CapSense slider for coarse temperature setting and the CapSense button for fine settings (P5_5/P0_5 to increment the temperature in 0.1 °C steps and P5_6/P0_6 to decrement the temperature in 0.1 °C). After the temperature is entered accurately, press the mechanical button **SW2** (CY8CKIT-030) or **SW1** (CY8CKIT-001) to calibrate.
4. The RTD calibration constant is calculated from the measured RTD resistance at the set temperature and the expected temperature value entered by the user. After calibration, the LCD displays the message "Calibrated". Press **SW2** (CY8CKIT-030) or **SW1**(CY8CKIT-001) to go to the main menu temperature display.

Figure 5-12. RTD Calibration



Note: The calibration constant is used inside 'RTDRestoTemp' for calibration. On system reset, the calibration constant is initialized as ‘1’. After calibration, the constant is stored in EEPROM. This value can be used for power-on-reset calibration. This can be implemented by users and is not demonstrated in this project.

5.2.3.3.6 Diode Calibration

Diode temperature measurement follows the equation:

$$T = \frac{q}{\eta k}\left(\frac{V_1 - V_2}{\ln\left(\frac{I_1}{I_2}\right)}\right)$$

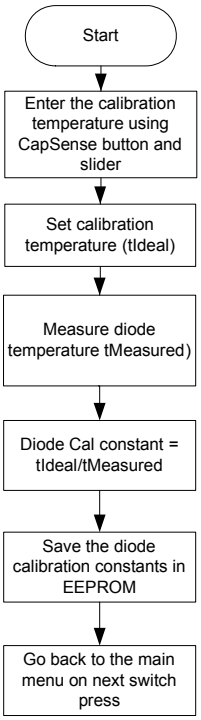
Equation 9

Here η is the ideality factor of the diode, varied from diode to diode. The temperature variation resulting from the ideality factor variation can be calibrated out. As equation 9 shows, variation in η results in a scale error. Performing a scale correction eliminates this error. The procedure to perform scale calibration is as follows:

1. Scroll down to the **Calibrate Diode1** (ninth) or **Calibrate Diode2** (tenth) item in the menu using CapSense buttons. Proceed through the calibration steps using mechanical switch SW2 connected to P6[1] of CY8CKIT-030 or SW1 connected to P1[7] of CY8CKIT-001.
2. The LCD displays “Set Temp:".

   Force the diode1 temperature to a known measured value between 0 °C and 100 °C, and enter that temperature using the CapSense slider and buttons. Use the CapSense slider for coarse temperature setting and the CapSense button for fine settings (P5_5/P0_5 to increment the temperature in 0.1 °C steps and P5_6/P0_6 to decrement the temperature in 0.1 °C). After the temperature is entered accurately, press the switch **SW2** in CY8CKIT-030 or **SW1** in CY8CKIT-001 to calibrate.
3. The diode calibration constant is calculated from the measured temperature and the expected temperature value entered by the user. After calibration, the LCD displays the message "Calibrated". Press **SW2** (CY8CKIT-030) or **SW1** (CY8CKIT-001) to go to the main menu temperature display.

Figure 5-13. Diode Calibration



Note: The calibration constant is used inside diode temperature measurement function for calibration. On system reset, the calibration constant is initialized as ‘1’. After calibration, the constants are stored in EEPROM. These values can be used for power-on-reset calibration. This can be implemented by users and is not demonstrated in this project.

5.2.3.3.7 IC Calibration

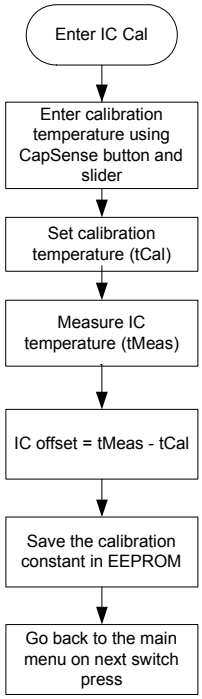
DS600 IC is used for cold junction compensation. Ideally an isothermal connection should be provided between the cold junction sensor and cold junction terminals. If that is not possible, having the cold junction sensor very close to the cold junction terminals is sufficient as it is done on CY8CKIT-025. This may result in a small temperature offset of around 0.5 °C. For high accuracy, a calibration routine, which performs a single point calibration, is provided to calibrate the temperature offset between the cold junction and the IC thermal pads.

The calibration steps for the IC temperature measurement are as follows:

1. Scroll down to the **Calibrate IC** (eleventh) item in the menu using CapSense buttons.
2. Proceed through the calibration steps using mechanical switch SW2 connected to P6[1] of CY8CKIT-030 or SW1 connected to P1[7] of CY8CKIT-001.
3. The LCD displays "Set Temp:".

   Force the IC temperature to a known measured value between 0 °C and 100 °C and enter that temperature using the CapSense slider and buttons. Use the CapSense slider for coarse temperature setting and the CapSense button for fine settings (P5_5/P0_5 to increment the temperature in 0.1 °C steps and P5_6/P0_6 to decrement the temperature in 0.1 °C). After the temperature is entered accurately, press the mechanical button **SW2** (CY8CKIT-030) or **SW1** (CY8CKIT-001) to calibrate.
4. The IC calibration constant (Offset) is calculated as the difference in measured temperature and the expected temperature value entered by the user. After calibration, the LCD displays the message "Calibrated". Press **SW2** (CY8CKIT-030) or **SW1**(CY8CKIT-001) to go back to main menu.

Figure 5-14. IC Calibration



Note: The calibration constant is subtracted from the measured IC temperature inside 'MeasureColdJnSensorTemp' function, for offset calibration. On system reset, the calibration constant is initialized to '0' and is set to the appropriate value after calibration. The calibration constant stored in EEPROM can be used for power-on-reset calibration. This can be implemented by users and is not demonstrated in this project.

5.2.3.3.8 *Noise Reduction and Offset Cancellation*

The accuracy of temperature reading displayed by a sensor depends on the ADC offset. Reducing ADC offset increases the temperature accuracy. Similarly, electronic noise affects the resolution measurement. A lower noise improves resolution. The techniques used in this example project to reduce offset and noise for all sensors are described in the following sections.

Sensor Output Filter

A software IIR filter is applied on the output voltage of the thermocouple, RTD, and diode sensors to eliminate noise. See application note, AN2099 - PSoC 1, PSoC 3, and PSoC 5LP - Single-Pole Infinite Impulse Response (IIR) Filters, for details. The filter (attenuation factor) applied depends on the noise on the output voltage of the sensor. The filter for a sensor is selected in such a way that the temperature resolution of the sensor is 0.1 °C. The function FilterSignal() in the code performs filtering.

Applying a filter increases the temperature settling time. A filter with a low cut-off frequency takes a much larger time for the temperature to settle down. To avoid large settling times when the temperature changes drastically, an algorithm is used where the filter is applied only when required (when the temperature is closer to the final temperature).

The filter attenuation factor can be changed in the *filter.h* file. For instance, to change the filter attenuation factor of the thermocouple from 4 to 5, go to *filter.h* and modify the following line of code.

```
/* Filter coefficient for sensors */
#define TC_FILTER_COEFF_SHIFT                4
```

Note: A higher attenuation factor reduces LCD flicker, but increases the temperature settling time.

Correlated Double Sampling (CDS)

Correlated double sampling is a technique where the offset and low frequency noise is eliminated by subtracting a zero voltage reading from every voltage sample. See application note AN66444 for more details on correlated double sampling. The ADC multiplexer, Channel 7, is used to make zero-voltage measurements for the thermistor and the thermocouple. To make zero-voltage measurements for the RTD and the diodes, pass a zero current through the RTD, diode, or calibration resistor and measure the ADC output across the RTD/diode/calibration resistor.

5.3 Sequenced ADC

This project demonstrates PSoC's ability to sense multiple inputs such as temperature sensor inputs, power rails, and generic analog inputs at a specific sample rate where the 50- or 60-Hz noise is eliminated.

5.3.1 Project Description

This project uses the RTD, IC temperature sensor, and diode temperature sensor from the CY8CKIT-025 DVK. This example also measures a 20 mV full scale input, 100 mV full scale input, 3.3 V rail, and 5 V rail. The 20 mV and 100 mV measurements require external stimulus. You can uses a low noise voltage source or analog output sensors such as a pressure sensor or a load cell to provide 0–20 mV and 0–100 mV inputs. The 5 V and 3.3 V signals can be derived from CY8CKIT-030 or CY8CKIT-001.

The project user interface is controlled by the CapSense buttons and character LCD. When the example project is programmed on CY8CKIT-030, the LCD displays the measurement from the RTD. Pressing CapSense button P5_5 successively displays measurements from the subsequent temperature sensors and voltage inputs. These measurements are in the following order: diode temperature, IC temperature, 3.3 V rail, 5 V rail, 20 mV full scale input, and 100 mV full scale input. P5_6 can be used to cycle through the measurements in reverse order.

In the previous example, readings are only taken when the LCD displays that particular measurement. However, in this project, ADC readings are taken continuously at repeated intervals. This allows filtering of ADC readings to remove specific frequencies of interest, such as 50- or 60-Hz hum. Furthermore, this project configures the delta-sigma ADC for 16-bit resolution instead of the maximum 20-bit resolution. This allows faster ADC conversion times and measurement update rates.

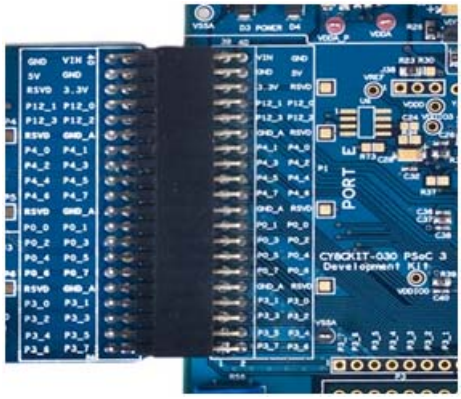
5.3.2 Project Operation

5.3.2.1 Hardware Connections

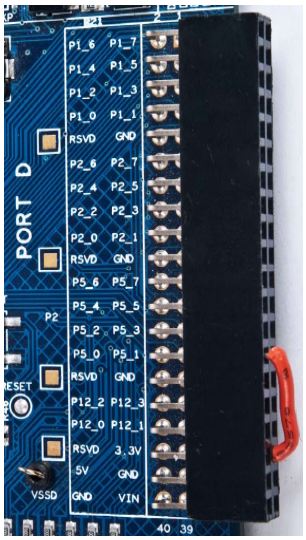
The kit includes example projects for the CY8CKIT-001 DVK, CY8CKIT-030 DVK, and CY8CKIT-050 DVK hardware platforms. The pin mapping for CY8CKIT-030 and CY8CKIT-050 are identical; follow the instructions provided for CY8CKIT-030 PSoC 3 DVK for CY8CKIT-050 PSoC 5LP DVK as well. The main difference between the examples for the CY8CKIT-001 and CY8CKIT-030 platforms is the PSoC pin mapping. The project provided for the CY8CKIT-001 PSoC DVK can be used with both PSoC 3 and PSoC 5LP. See Migrating Projects to use with CY8CKIT-050 and CY8CKIT-010 on page 31 for more details.

5.3.2.1.1 CY8CKIT-030 PSoC DVK

1. Plug the PSoC Precision Analog Temperature Sensor EBK to port E of the CY8CKIT-030 DVK.



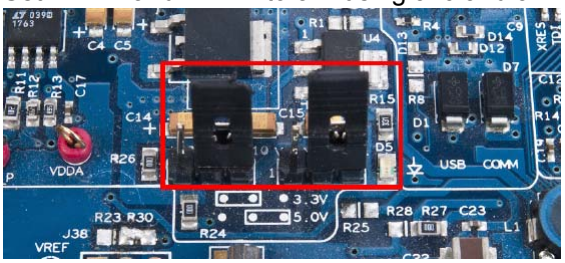
2. Connect 3.3 V on port D to P5_0 on port D



3. The variable resistor (VR) is used for the 5 V input. Turning VR changes the voltage input to PSoC from 0 to VDDA. On the CY8CKIT-030, the VR is directly tied to P6_5. Ensure that the VR is turned to output 5 V. The VR is used to make sure that 5 V is not accidentally driven on the pin if the system is set for 3.3 V.
4. Connect a 20 mV full scale differential signal from a voltage to source P5_2 (+) and P5_3 (–). You can also use an analog output pressure sensor or load cell to give 0-20 mV input.
5. Connect a 100 mV full scale differential signal from a voltage to source P1_6 (+) and P1_7 (–). You can also use an analog output pressure sensor or load cell to give 0-100 mV input.
6. Ensure that the LCD included with the CY8CKIT-030 DVK is attached to the LCD header.

   Note: For testing purpose an external voltage source can be created by using 1.5 V AA battery and resistor divider. Upper arm of the resistor divider can be 15 K resistor while bottom arm can be 1 K potentiometer.

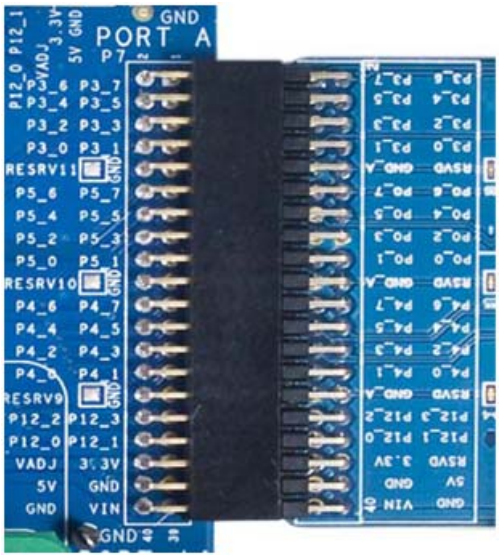
7. Set VDDD and VDDA to 5 V using J10 and J11.



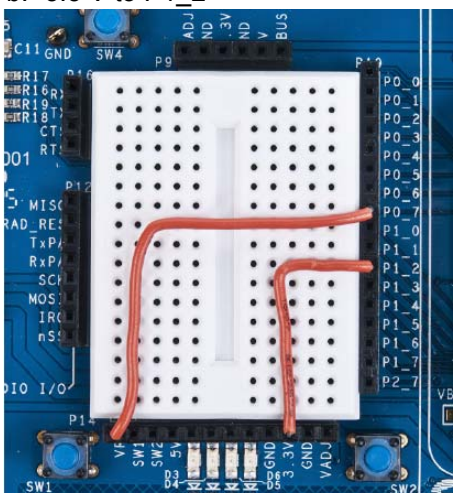
8. In the CY8KIT-025 EBK, choose appropriate jumper settings for internal sensors; see the Kit Operation chapter on page 18.

5.3.2.1.2 CY8CKIT-001 PSoC DVK

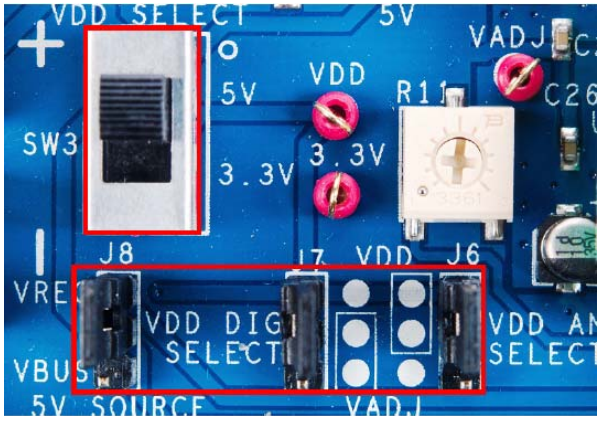
1. Plug the PSoC Precision Analog Temperature Sensor EBK to port A of the CY8CKIT-001 DVK.



2. In the pin header/breadboard area of the CY8CKIT-001 base board, use jumper wires to make the following connections:
   a. VR to P0_7. The VR is used as the 5 V input. Ensure that the VR is turned to output 5 V. The VR makes sure that 5 V is not accidentally driven on the pin if the system is set for 3.3 V.
   b. 3.3 V to P1_2



3. Connect a 20 mV full scale differential signal to P0_0 (+) and P0_1 (–).
4. Connect a 100 mV full scale differential signal to P1_6 (+) and P1_7 (–).
5. Set the system to run at 5 V using SW 3; set J7 VDD DIG and J6 VDD ANLG to VDD = 5 V.



6. Ensure that the LCD included with the CY8CKIT-001 PSoC DVK is attached and the LCD power jumper (J12) is in the ON position.



7. In the CY8CKIT-025 EBK, choose appropriate jumper settings for internal sensors; see the Kit Operation chapter on page 18.

5.3.2.2 *Run Sequenced ADC Example Firmware*

5.3.2.2.1 *CY8CKIT-030 PSoC 3 DVK*

The steps to run the example firmware on CY8CKIT-030 are:

1. Attach a USB cable from the PC to the PSoC 3 DVK program/debug USB port (use J1, the USB connector closest to the corner of the board).
2. Open the SequencedADC_030_050 project using PSoC Creator and select **Debug > Program** to program the PSoC.
3. Detach the USB cable and connect hardware according to Hardware Connections on page 52.
4. Plug at 12 V wall-wart into the J4 barrel jack connector.
5. Press the **Reset** (SW1) button on the PSoC 3 DVK to run the newly programmed firmware image.
6. On reset, the LCD displays the RTD temperature, which is the first item in the user selectable menu.
7. Press CapSense button **P5_5** to scroll up the menu and **P5_6** to scroll down the menu.

Note: In the firmware, voltage displayed on the LCD for the 5-V rail is truncated to two digits after the decimal point. Therefore, when the potentiometer is turned to the **0V** position, if the ADC result is negative due to negative offset error, you will see '–0.00' on the LCD. Increase the number of digits after the decimal point to be displayed on the LCD by increasing the floating point qualifier in the line number 105 of the *display.c* file.

5.3.2.2.2 CY8CKIT-001 PSoC DVK

The steps to run the example firmware on CY8CKIT-001 are:

1. Apply 12 VDC power to the PSoC DVK.
2. Attach the MiniProg3 device, first to a USB port on the PC and then to the PROG port on the CY8CKIT-009 PSoC 3/CY8CKIT-010 PSoC 5LP processor module.
3. Open the SequenceADC_001 project using PSoC Creator and select **Debug > Program** to program the PSoC.
4. Disconnect power to the PSoC DVK and connect hardware according to Hardware Connections on page 33.
5. Power the PSoC DVK.
6. On reset, the LCD displays a welcome message and then the RTD temperature, which is the first item in the user selectable menu.
7. Press CapSense button **P0_5** to scroll up the menu and **P0_6** to scroll down the menu.

Note: The diode temperature in CY8CKIT-001 is slightly lower than the actual temperature. This is because CY8CKIT-001 is a general-purpose board and is not optimized for analog routing.

5.3.2.3 Testing the Project

For best results when testing the temperature measurements, ensure that you have good thermal contact between the sensor (RTD, diode, or IC sensor) and your reference thermometer.

5.3.2.4 Expected Performance and Test Results

This section provides details about the expected performance of the temperature, voltage, and power rail measurements. The output expected from the temperature sensors are similar to those described in the previous example project, except for minor differences, such as use of 16-bit ADC for all measurements. It is recommended to see Expected Performance and Test Results on page 36 before reading this section. The focus of this example is to show multiplexing multiple input sources, that is, voltages, temperatures, and power rails. The section Expected Performance and Test Results on page 36 details the tolerance specification of each temperature sensor and the expected temperature performance from the sensors.

Following are typical expected results for each measurement observed on a sample board.

5.3.2.4.1 Voltage (20 mV) Input

A voltage source is used to provide inputs from –20 mV to +20 mV and the output displayed by PSoC is compared to a multimeter output.

The 20 mV reading should typically be within ± 40 µV of the actual value. The ADC gain error is the main factor driving this error and the PSoC 3 ADC with a maximum gain error of 0.2% keeps the measurement inaccuracy to < 40 µV.

5.3.2.4.2 100 mV Input

A voltage source is used to provide inputs from –100 mV to +100 mV and the output displayed by PSoC is compared to a multimeter output.

The 10 mV reading should typically be within ± 200 µV of the actual value. The ADC gain error is the main factor driving this error and PSoC 3 ADC with a maximum gain error of 0.2 percent keeps the measurement inaccuracy to <100 µV.

5.3.2.4.3 RTD

RTD temperature is simulated with a potentiometer, similar to the testing for the tempsense project. Following are the test results.

Table 5-8. RTD Test Results

Actual Temp(°C)	Measured Temp(°C)	Difference(°C)
76.60	76.64	–0.04
37.14	37.06	0.08
20.44	20.5	–0.06
0.13	0.03	0.10
–19.73	–19.77	0.04
–45.76	–45.82	0.06

5.3.2.4.4 Diode

The diode is typically within ± 2°C. See the description in Expected Performance and Test Results on page 36 to understand more about the expected error and thermal considerations. Table 5-9 provides an example of measured temperature. A standard thermometer with ±1 °C accuracy is used for comparison.

Table 5-9. Diode Test Results

Actual Temp (°C)	Measured Temp (°C)	Difference (°C)
0.8	1.2	0.4
6.2	5.6	–0.6
17	16.1	–0.9
22	21	–1
30	29.3	–0.7
75.3	74.7	–0.6
99.5	99.4	–0.1

5.3.2.4.5 DS600 IC Temperature Sensor

The DS600 typically is within ± 0.5°C. Table 5-10 provides an example of measured temperature. A standard thermometer with ±1 °C accuracy is used for comparison. Note that as mentioned in DS600 IC Sensor on page 39, the IC measures the pad temperature below it and the IC signal chain is accurate, as shown in Table 5-7 on page 39.

Table 5-10. DS600 IC Temperature Sensor Test Results

Actual Temp (°C)	Measured Temp (°C)	Difference (°C)
39.9	39.5	0.42
20.1	19.8	0.26
0.8	0.4	0.36
–19.7	–20.0	0.31
–39.4	–39.8	0.40

5.3.2.4.6 3.3 V Rail

The 3.3 V rail is typically within ± 20 mV. The ADC gain error is the main source of error.

5.3.2.4.7 5 V Rail

The 5 V rail is typically within ± 20 mV. The ADC gain error is the main source of error.

5.3.3 Project Details

The Tempsense project demonstrates a sensing implementation style where the sensor is only sensed when its value is required. The ADC is controlled by firmware and the sensor is only sampled when its value is displayed on the LCD. In many applications, sensors must be read all the time and at precise intervals. The Sequenced ADC project implements ADC scanning in hardware and samples sensors at precise intervals. This ensures that sensor reading is always available for the application.

Another reason for taking ADC readings continuously is the ability to filter certain frequencies such as 50 Hz or 60 Hz hum on each reading. If the ADC readings are taken at consistent intervals, then specific filters can be applied for each reading.

This project reads seven different inputs: a 20 mV full scale voltage, a 100 mV full scale voltage, an RTD temperature sensor, a diode temperature sensor, a DS600 IC temperature sensor, a 3.3 V voltage rail, and a 5 V rail.

Some measurements such as RTD, diode, and IC temperature sensor require multiple readings to generate the final result. For the RTD example, four ADC readings are required for each temperature: RTD, RTD CDS, reference resistor, and reference resistor CDS. Due to this, a total of 15 ADC readings are required to generate the seven measurement values. Table 5-11 shows the ADC readings that make up each measurement.

Table 5-11. ADC Reading

Measurement	ADC Reading
20 mV Input	20 mV Input
RTD	RTD
	RTD CDS
	Reference Resistor
	Reference Resistor CDS
100 mV Input	100 mv Input
Diode	Voltage Across Diode Current 1
	Voltage Across Diode Current 2
	Reference Resistor Current1
	Reference Resistor Current2
	Reference Resistor CDS
DS600 IC Temp Sensor	DS600 Voltage
	DS600 CDS
3.3 V Rail	3.3 V Rail
5 V Rail	5 V Rail

5.3.3.1 Project Schematic

Figure 5-15 shows the PSoC Creator schematic of the Sequenced ADC project. To facilitate ADC readings at consistent intervals, a hardware counter is used to trigger the start of convert (SOC). The SOC is automatically triggered by the hardware 16-bit counter "Scheduler". This counter triggers an ADC reading every 606 µs. The counter interval is selected to support the 15 ADC readings at a repetition rate of 110 Hz per reading. To change this rate, change the ADC_PERIOD defined in *sensors.h*. Note that if this define is changed, the counter period should not run faster than the ADC conversion rate and processing time for the ADC interrupt. The ADC conversion rate of 3861 samples per second takes 259 µs. The processing time takes about 320 µs with the BUS_CLK set to 48 MHz.

Every time the ADC finishes a conversion, it triggers an interrupt, ADC_INT.c. In this interrupt, the IDAC8, CurrentMux, ADCMux, and ADC are changed to prepare the signal chain for the next measurement. Also, the ADC readings are filtered in this ISR. The filtered results are then stored in a ping-pong buffer to be read later.

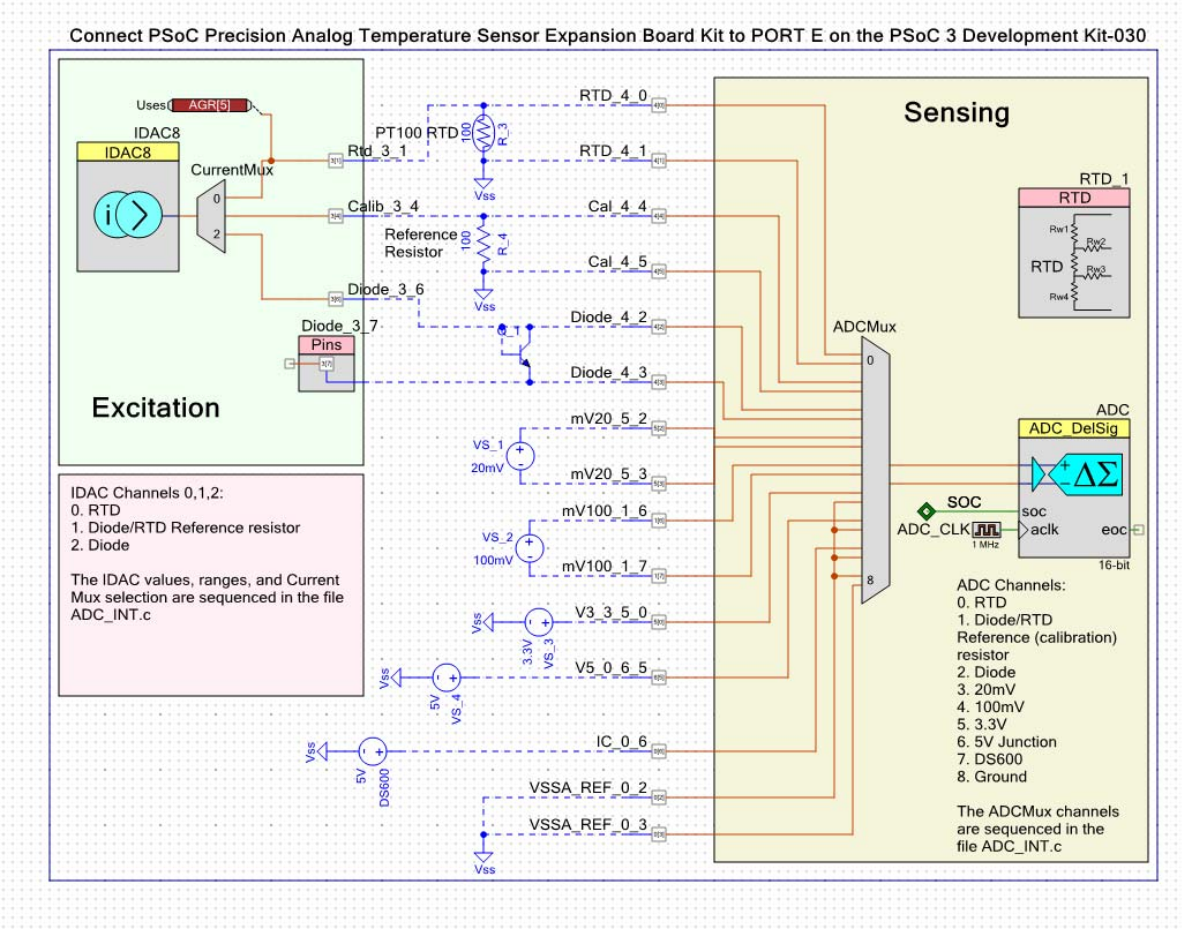
To ensure that the firmware in the ADC ISR executes before the next ADC SOC is triggered by the scheduler, the counter "Slot Counter" is used. This hardware counter counts the number of SOCs triggered by the scheduler. This count is compared against a software counter in the ADC ISR. If the two counters do not match, then the reading is flagged as invalid.

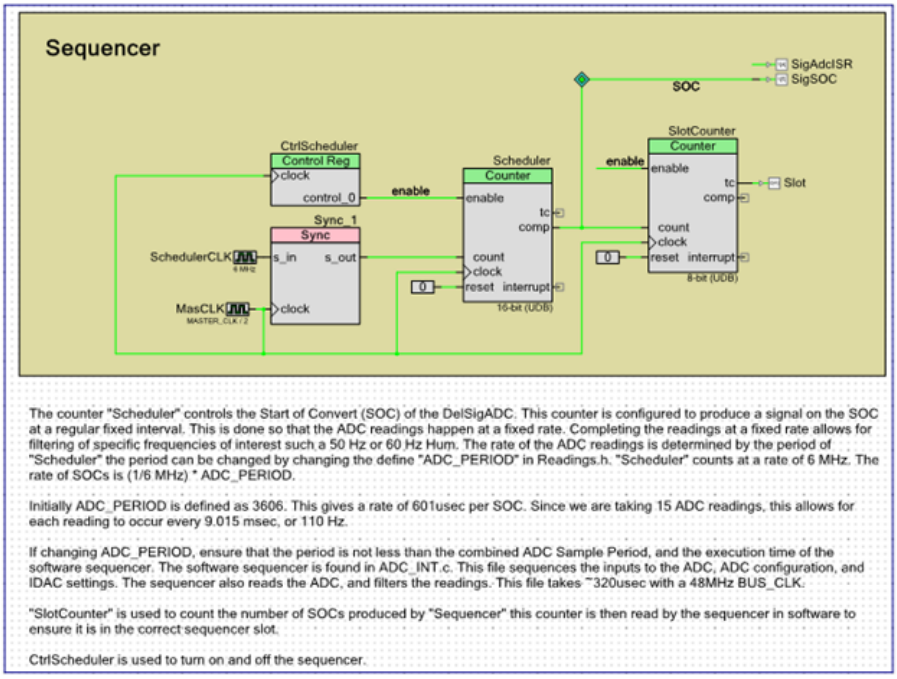
The control register CtrlScheduler is used to control whether the scheduler is running. Setting this control register low disables the scheduler and stops ADC readings.

IDAC8 is used to force current through the RTD, diode, and reference resistor. Diode P3_7 is used to force GND collector of the diode.

Manual routing is used on the RTD to ensure that the load resistance of the IDAC does not cause the value to droop between the reference resistor reading and RTD reading. For more information, refer to AN70698.

Figure 5-15. Schematic for Input Muxes and IDAC





5.3.3.2 *Component Configuration*

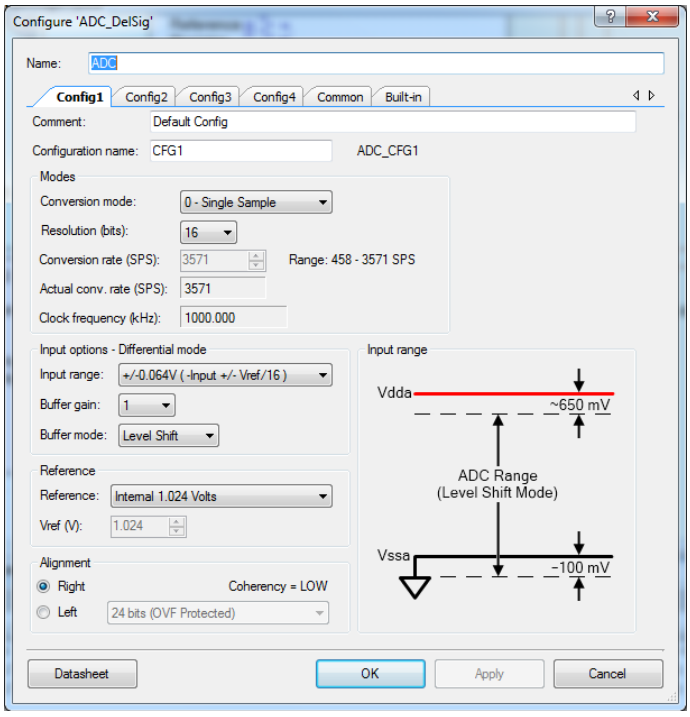
As mentioned earlier, you need to take a total of 15 different ADC readings for the seven measurements. Each of these readings requires different ADC configurations to achieve the maximum dynamic range out of the ADC. For example, the 3.3 V and 5 V rail require the ADC to have a large input voltage range. The same range yields imprecise results for the 20 mV input. To accommodate these different sensors, four different ADC configurations are used. The ADC configurations are switched in the ADC interrupt. Table 5-12 shows the 15 ADC readings and their ADC voltage range.

Table 5-12. ADC Reading and Voltage Range

Measurement	ADC Reading	ADC Input Range
20 mV Input	20 mV Input	± 0.064 V
RTD	RTD	± 0.256 V
	RTD CDS	± 0.256 V
	Reference Resistor	± 0.256 V
	Reference Resistor CDS	± 0.256 V
100 mV Input	100 mv Input	± 0.256 V
Diode	Voltage Across Diode Current 1	± 1.024 V
	Voltage Across Diode Current 2	± 1.024 V
	Reference Resistor Current1	± 0.064 V
	Reference Resistor Current2	± 0.064 V
	Reference Resistor CDS	± 0.064 V
DS600 IC Temp Sensor	DS600 Voltage	± 1.024 V
	DS600 CDS	± 1.024 V
3.3 V Rail	3.3 V Rail	± 6.144 V
5 V Rail	5 V Rail	± 6.144 V

The following images show the different ADC configurations.

Figure 5-16. ± 0.064 V ADC Configuration



Note that Buffer Mode is set to **Level Shift** because many of the voltages being read are near VSS; this setting yields better ADC results for signals near VSS.

The sample rate of 3861 is chosen such that each sensor is read at a repeat rate of 110 Hz.

Figure 5-17. ± 0.256 V ADC Configuration

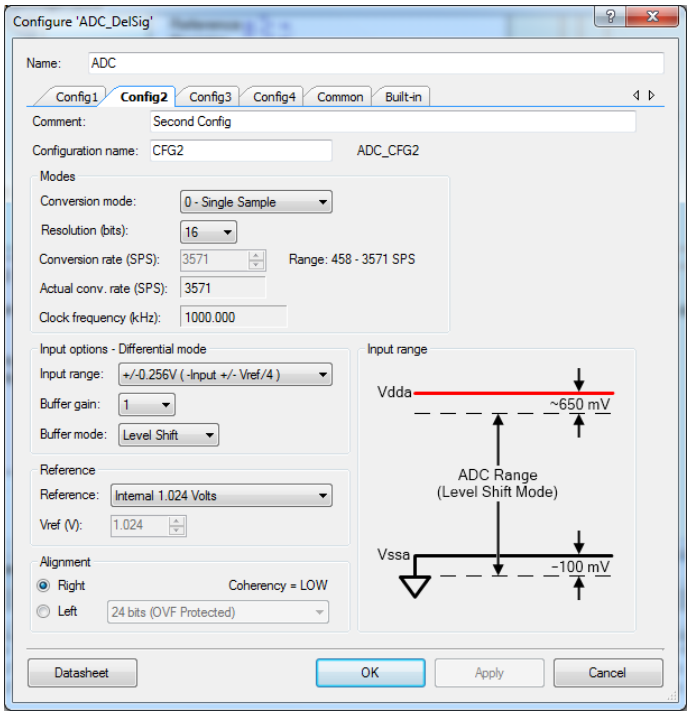


Figure 5-18. ± 1.024 V ADC Configuration

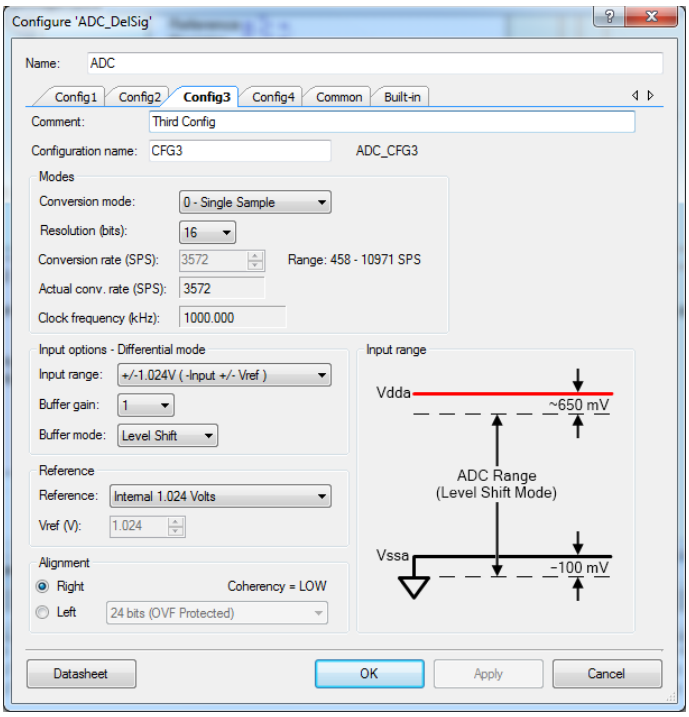
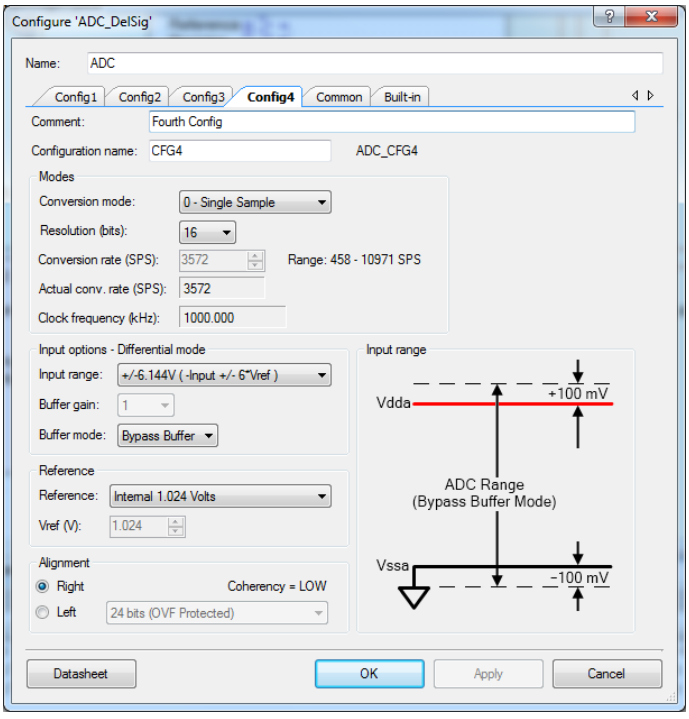


Figure 5-19. ± 6.144 V ADC Configuration



Note that the Buffer Mode is set to **Bypass Buffer** because the voltage being read is near VDD; this setting yields better ADC results for signals near VDD. However, the input impedance of the ADC is lower without the buffer, so the input signal should not load the ADC.

As stated earlier, the IDAC is used to force current through the RTD, diode, and calibration resistor. Table 5-13 lists the IDAC range and values used for each sensor. The IDAC configurations are switched in the ADC ISR.

Table 5-13. IDAC Range and Values

Measurement	ADC Reading	IDAC Range	IDAC Value
RTD	RTD	2 mA	1.024 mA
	RTD CDS	2 mA	1.024 mA
	Reference Resistor	2 mA	1.024 mA
	Reference Resistor CDS	2 mA	0 mA
Diode	Voltage Across Diode Current 1	255 µA	10 µA
	Voltage Across Diode Current 2	255 µA	100 µA
	Reference Resistor Current 1	255 µA	10 µA
	Reference Resistor Current 2	255 µA	100 µA
	Reference Resistor CDS	255 µA	0 µA

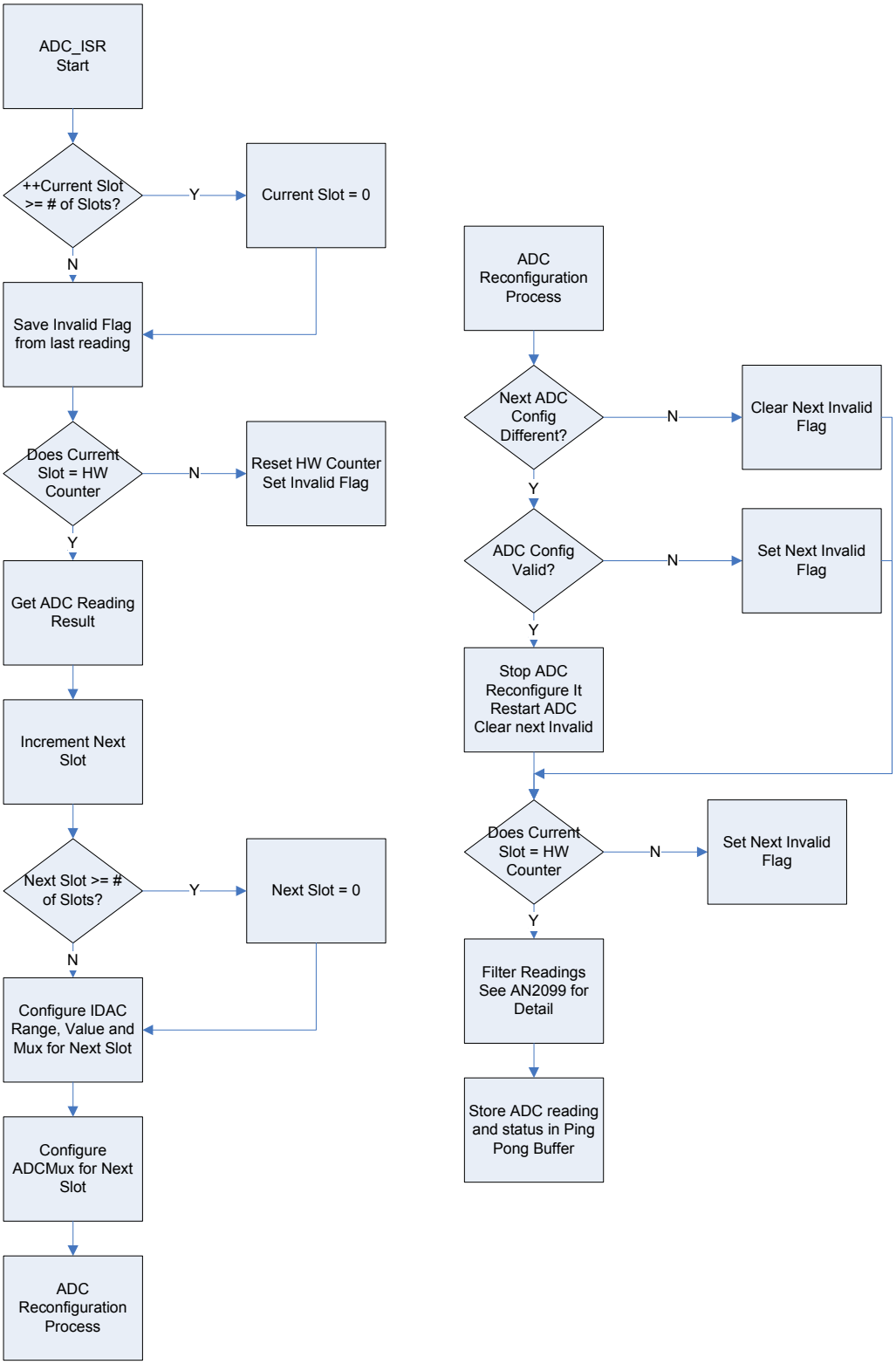
5.3.3.3 *Firmware Description and Flowcharts*

5.3.3.3.1 *ADC ISR Flowchart*

The ADC ISR automatically prepares the signal chain for the next ADC reading. The ISR also filters the current ADC reading and stores the filtered result in a ping pong buffer to be read later.

Figure 5-20 shows the ADC ISR flowchart. The ISR code is found in ADC_INT.c. This interrupt is triggered at the end of every ADC conversion.

Figure 5-20. ADC ISR Flowchart



The SlotCounter is used to count ADC EOCs. This hardware counter is compared with the current slot value in the ADC ISR. If they do not match, then the measurement is flagged as invalid. This is

done after the ADC reconfiguration to ensure that reconfiguration completed before the next ADC SOC is triggered.

5.3.3.3.2 *Fixed Point Math*

This project does not use floating point math when calculating the measurement results. This is done to reduce processor overhead when converting ADC readings into measurement values. Therefore, scaling is used to calculate the final measurement value. For ease of use, the values are converted to float when displayed on the LCD. The following sections describe each measurement and how its readings are converted into measurement values.

20 mV and 100 mV Input

For 20 mV and 100 mV reading, ADC_COUNTS_PER_VOLT is used to convert the ADC counts to volts. Because integer math is used, dividing the ADC counts by this constant results in a precision of only 1 V, which is not adequate to measure a 20 mV or 100 mV full scale input. To fix this, the ADC counts for these readings are scaled up by 10000 and divided by ADC_COUNTS_PER_VOLT. The result is in ten-thousandths of a volt.

3.3 V Rail and 5 V Rail

For the 3.3 V and 5 V reading, ADC_COUNTS_PER_VOLT is used to convert the ADC counts to volts. Because integer math is used, dividing the ADC counts by this constant results in a precision of only 1 V, which is not adequate to measure a 3.3 V or 5 V rail. To fix this, the ADC counts for these readings are scaled up by 1000 and then divided by ADC_COUNTS_PER_VOLT. The result is in thousandths of a volt.

RTD and DS600

To avoid floating point math, the conversion from ADC readings to temperature uses integer math. To preserve precision, the readings are scaled up. Thus the final result of the temperature measurements is in hundredths of a degree C. For more information on the conversion, see the inline code comments in the associated project.

Diode

The diode temperature measurement uses floating point math. This is because the natural log of the current ratio is taken. Not using floating point math with this measurement results in poor accuracy and stability of the diode measurement.

5.3.3.3.3 *Filter*

A software IIR filter is applied on all ADC readings to eliminate noise. See application note AN2099 for details on the software IIR filter. The filter (attenuation factor) applied depends on the noise on the voltage output of the sensor. Applying a filter increases the measurement settling time. A very low cut-off filter requires a much larger time for the measurement to settle down. To avoid large settling times when the input changes drastically, an algorithm is used where the filter is applied only when required (when the measurement is closer to the final measurement).

The filter attenuation factor can be changed in the Init_Buffers routine in *Readings.c*. For example, to change the filter attenuation factor of 3.3 V from 8 to 16, go to *Readings.c* and modify the following line of code.

```
Filter_CoeffBuf[READ_RAIL3V3]=8;
```

to

```
Filter_CoeffBuf[READ_RAIL3V3]=16;
```

Note that a higher attenuation factor reduces LCD flicker, but increases temperature settling time.

5.3.3.3.4 Correlated Double Sampling (CDS)

Correlated double sampling is a technique where the offset and low frequency noise is eliminated by subtracting a zero voltage reading from every voltage sample. See application notes AN2226 and AN66444 for more details on correlated double sampling. The ADCMux, Channel 8, is used to make zero-voltage measurement for the DS600. To make zero-voltage measurement for the RTD and diode, pass a zero current through the RTD, diode, or calibration resistor and measure the ADC output across the RTD/diode/calibration resistor.

5.3.3.3.5 Calibration

For this project, user calibration is not available for the temperature sensors. The 3.3 V and 5 V measurements are adjusted by a calibration factor. This is because the ADC configuration (±6.144 V ADC input range) used to read the 3.3 V and 5 V rail is not factory calibrated. To achieve this calibration factor, a voltage is read with one of the factory calibrated ADC ranges (ADC configuration 1 – ±1.024 V range); that same value is read with the non-factory calibrated range. The ratio of these two readings is then used to adjust the 3.3 V and 5 V measurements. This calibration is done before the main code loop.

5.4 Thermal Management

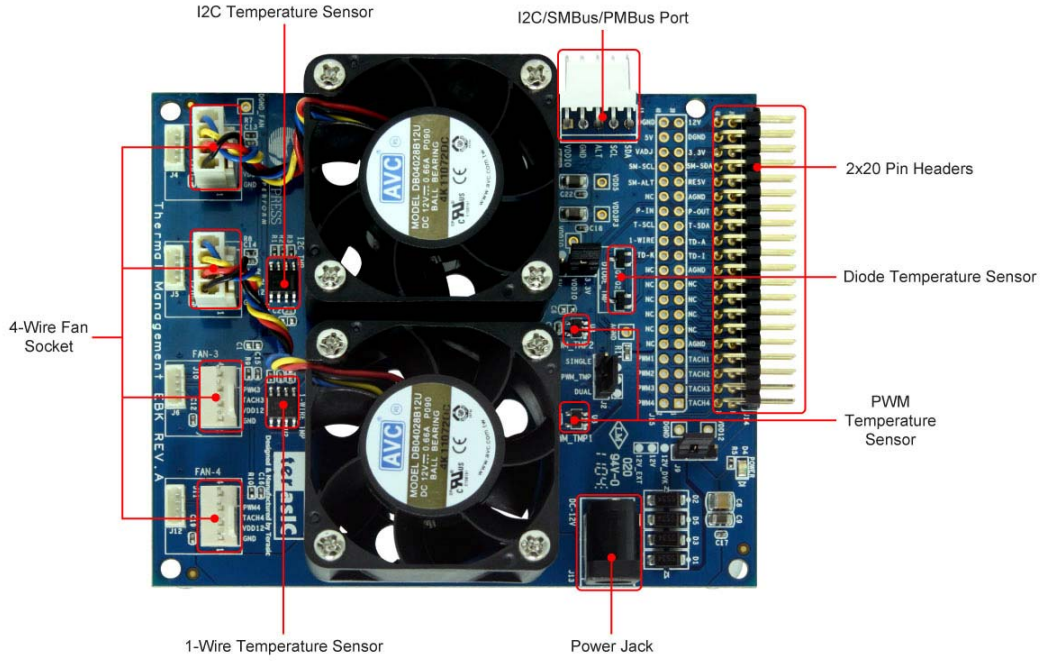
This project demonstrates fan speed control using analog temperature sensors.

5.4.1 Project Description

The thermal management project controls the fan speed based on the temperatures measured by the temperature sensors. This project uses the CY8CKIT-036 Thermal Management (TME) EBK in addition to the CY8CKIT-025 EBK. The example project controls two fans based on their zone temperatures. The concept of a thermal zone and the sensors used in a specific zone are explained in Project Details on page 75. On reset, the LCD displays Zone 1 Summary. Press switch SW2 to cycle through the three menu items:

- Zone 1 Summary: Temperature algorithm used to compute zone temperature, desired fan speed, and actual fan speed
- Zone 2 Summary: Temperature algorithm used to compute zone temperature, desired fan speed, and actual fan speed
- Temperature Sensor Summary: The temperatures of the individual temperature sensors in each zone and the zone temperatures

Figure 5-21. Thermal Management EBK



5.4.1.1 *Thermal Management EBK Description*

The Thermal Management (TME) EBK contains two four-wire, 12 V brushless DC fans with connectors to support an additional two fans for designers who need to prototype with their own specific fan models. Six temperature sensors (four different kinds) are also installed on the kit:

- TMP175 I2C digital temperature sensor
- Two TMP05 PWM output digital temperature sensors
- DS18S20 "One Wire" digital temperature sensor
- Two MMBT3094 temperature diodes.

This combination of hardware elements enables designers to rapidly prototype thermal management solutions in a variety of configurations.

The TME EBK also provides an I2C/SMBus/PMBus compatible header to support systems that require communication with a host controller. This functionality is implemented in a single PSoC 3. The TME EBK routes the input/output signals for thermal management to a PSoC 3 mounted on a development kit platform such as the CY8CKIT-001 or CY8CKIT-030. PSoC 3 is not mounted on the TME EBK.

The CY8CKIT-036 TME EBK has three example projects that demonstrate firmware-based fan control, hardware-based closed-loop fan control, and a thermal management system. A detailed explanation of the TME EBK and the firmware examples that are provided with the kit are available at www.cypress.com/go/CY8CKIT-036.

5.4.1.2 *Using TME EBK with CY8CKIT-025*

You can control the fans in the TME EBK using the temperature sensors in the CY8CKIT-025 EBK. The TME EBK has only one analog temperature sensor, which is the diode-based temperature sensor. To illustrate the analog temperature sensing capabilities of PSoC 3 better, the CY8CKIT-025 EBK is combined with the TME EBK to demonstrate thermal management using different analog temperature sensors such as diode and RTD.

Figure 5-22 shows the CY8CKIT-036 TME EBK and CY8CKIT-025 EBK connected to the CY8CKIT-030 PSoC 3 DVK. The CY8CKIT-025 EBK is connected to port E and the CY8CKIT-036 TME EBK is connected to port D. The same setup and connections can be used when using CY8CKIT-050.

Figure 5-23 shows the CY8CKIT-036 TME EBK and CY8CKIT-025 EBK connected to the CY8CKIT-001 PSoC DVK. The CY8CKIT-025 EBK is connected to port A and the CY8CKIT-036 TME EBK is connected to port B. The same setup and connections can be used when using CY8CKIT-010.

Figure 5-22. TME EBK and CY8CKIT-025 EBK Connected to CY8CKIT-030 PSoC 3 DVK

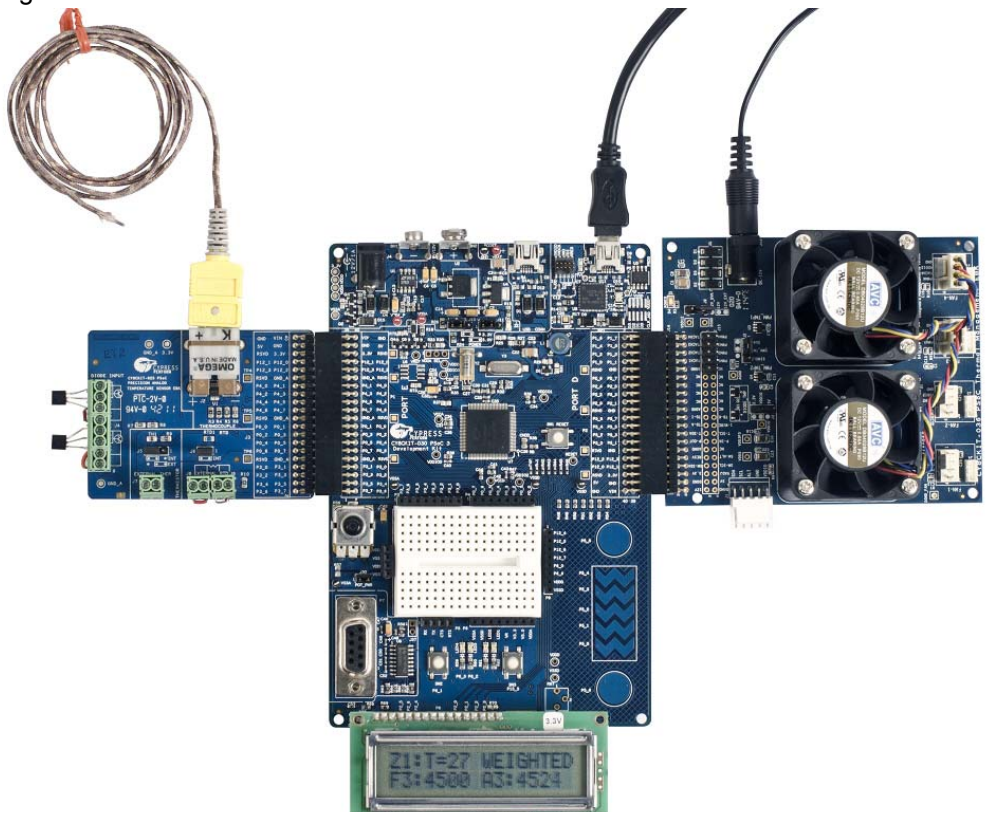
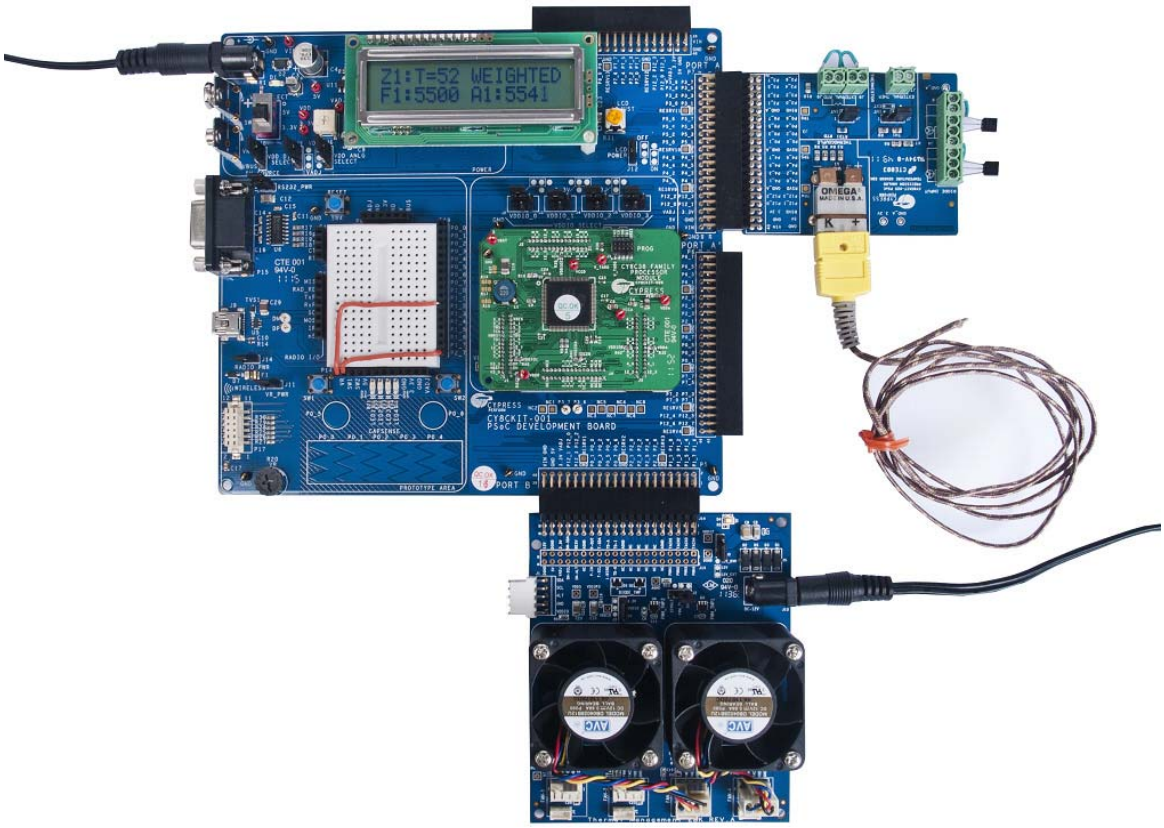


Figure 5-23. TME EBK and CY8CKIT-025 EBK Connected to CY8CKIT-001 PSoC 1 DVK



5.4.2 Project Operation

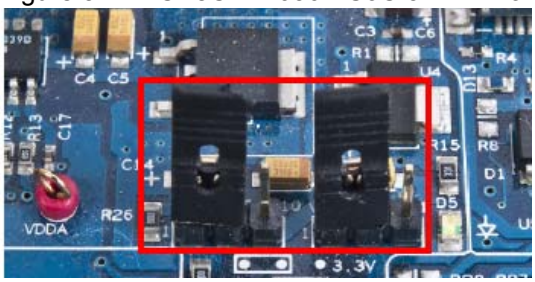
5.4.2.1 Hardware Connections

The CY8CKIT-025 kit includes example projects for the CY8CKIT-001 DVK, CY8CKIT-030 DVK, and CY8CKIT-050 DVK hardware platforms. The pin mapping for CY8CKIT-030 and CY8CKIT-050 are identical; follow the instructions provided for CY8CKIT-030 PSoC 3 DVK for CY8CKIT-050 PSoC 5LP DVK as well except that PSoC 5LP device CY8C5868AXI-LP035 should be selected (Project > Device selector). The main difference between the examples for the CY8CKIT-001 and CY8CKIT-030 platforms is the PSoC pin mapping.

5.4.2.1.1 CY8CKIT-030 PSoC 3 DVK

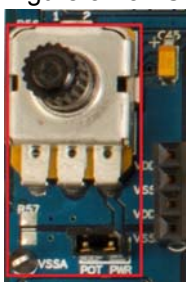
1. No jumper wires are required for the PSoC 3 DVK examples because the buttons and potentiometer are hardwired to GPIOs. Ensure that the LCD included with the PSoC 3 DVK is attached.
2. Set VDDD and VDDA to 3.3 V using J10 and J11.

Figure 5-24. CY8CKIT-030 PSoC 3 DVK Power Jumpers



3. Ensure that POT_PWR is enabled by installing a jumper on J30.

Figure 5-25. CY8CKIT-030 PSoC 3 DVK Potentiometer Power

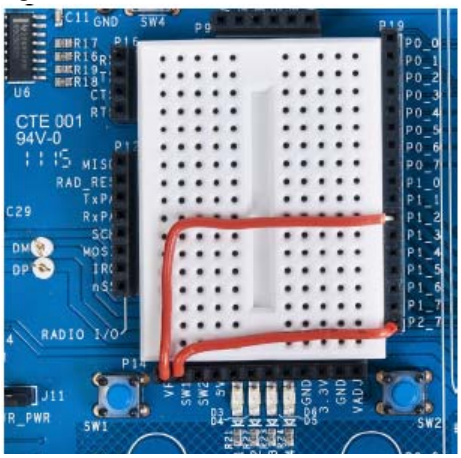


CAUTION: The TME EBK plugged into port D of the CY8CKIT-030 DVK uses the SWD/JTAG programming pins for fan control as well. Due to this pin sharing, the TME EBK should not be attached to the CY8CKIT-030 DVK while programming the PSoC 3 device using MiniProg3 or DVKProg3 programmer. After the programming is complete, remove the programmer from the programming header on the PSoC 3 DVK and then attach the TME EBK to port D. Due to pin sharing, debugging the project is also not possible because it uses the same programming pins.

5.4.2.1.2 CY8CKIT-001 PSoC DVK

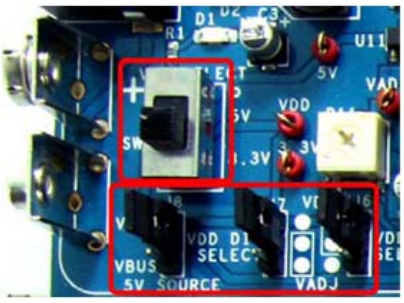
1. In the pin header/breadboard area of the PSoC DVK base board, use jumper wires to make the following connections:
   - VR to P1_2
   - SW1 to P2_7

Figure 5-26. CY8CKIT-001 PSoC DVK Breadboard



2. Set the system to run at 3.3 V using SW3 and set J6 VDD DIG and J7 VDD ANLG to VDD = 3.3 V.

Figure 5-27. CY8CKIT-001 PSoC DVK Power Jumpers



3. Ensure that the LCD included with the PSoC DVK is attached and that the LCD power jumper (J12) is in the ON position.

Figure 5-28. CY8CKIT-001 PSoC DVK LCD Power Jumper



4. Ensure that the VR_PWR jumper (J11) is installed.

Figure 5-29. CY8CKIT-001 PSoC DVK VR_POWER Jumper

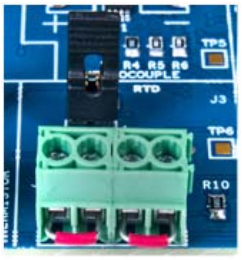


CAUTION: The TME EBK plugged into port B of the CY8CKIT-001 DVK uses the SWD/JTAG programming pins for fan control as well. Due to this pin sharing, the TME EBK should not be attached to the PSoC DVK while programming the PSoC 3 device on the DVK using MiniProg3 programmer. After the programming is complete, remove MiniProg3 from the programming header on the PSoC DVK and then attach the TME EBK to port B. Due to pin sharing, debugging the project is also not possible because it uses the same programming pins.

5.4.2.1.3 CY8CKIT-025 PSoC Precision Analog Temperature Sensor EBK

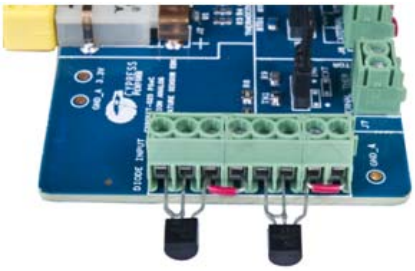
1. Make sure the RTD jumper J6 is on; on jumpers J8 and J9, short 1-2 with wires so that the onboard RTD is used for temperature measurements.

Figure 5-30. CY8CKIT-025 RTD Jumper



2. Connect the diodes, as shown in Figure 5-31.

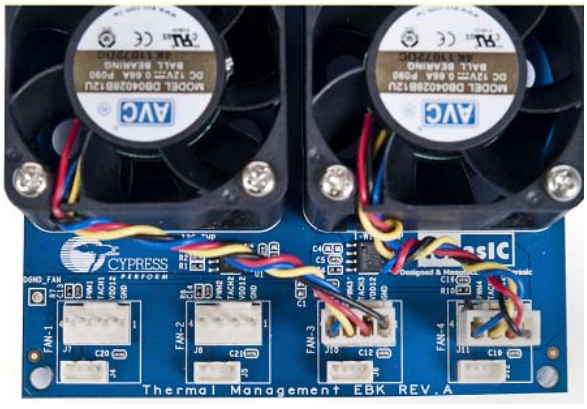
Figure 5-31. CY8CKIT-025 Diodes



5.4.2.1.4 CY8CKIT-036 Thermal Management EBK

1. Connect the fans to headers fan 3 and fan 4, as shown in Figure 5-32. Note that the fans are connected to headers 1 and 2 by default.

Figure 5-32. Thermal Management EBK Fans



2. Configure jumper J9 to power the thermal management EBK using an external power supply, as shown in Figure 5-33.

Figure 5-33. External Power Supply Connected



5.4.2.2 Run Thermal Management System Firmware

5.4.2.2.1 CY8CKIT-030 PSoC 3 DVK

The steps to run the example firmware on CY8CKIT-030 are:

1. Make the hardware connections on the CY8CKIT-030 DVK, as explained in 5.4.2.1.1 CY8CKIT-030 PSoC 3 DVK.
2. Attach a USB cable from the PC to the PSoC 3 DVK program/debug USB port (use J1, the USB connector closest to the corner of the board). Ensure that the TME EBK is not attached to the PSoC 3 DVK before attaching the USB cable for programming.
3. Open the ThermalManagementSystem_030_050 project using PSoC Creator and select **Debug > Program** to program the PSoC.
4. Detach the USB cable; if the DVK is powered by external power supply, disconnect power supply to the DVK as well.
5. Connect the CY8CKIT-025 EBK and TME EBK to the DVK, as shown in Figure 5-22.
6. Power the PSoC 3 DVK and the TME EBK, as shown in Figure 5-22.
7. Press the **Reset** (SW1) button on the PSoC 3 DVK to run the newly programmed firmware image.
8. On reset, the LCD displays a welcome message and starts with the Zone 1 temperature reading and the corresponding temperature calculation algorithm. The desired and actual fan speeds corresponding to that zone are also displayed.
9. Use the push button SW2 to view the different zone characteristics.

5.4.2.2.2 CY8CKIT-001 PSoC DVK

The steps to run the example firmware on CY8CKIT-001 are:

1. Make the hardware connections on the CY8CKIT-001 DVK, as explained in 5.4.2.1.2 CY8CKIT-001 PSoC DVK.
2. Apply 12 VDC power to the PSoC DVK.
3. Attach the MiniProg3 device, first to a USB port on the PC and then to the PROG port on the CY8CKIT-009 PSoC 3/CY8CKIT-010 PSoC 5LP processor module. Ensure that the TME EBK is not attached to the PSoC 3 DVK before attaching the USB cable for programming.

4. Open the ThermalManagementSystem_001 project using PSoC Creator and select **Debug > Program** to program PSoC 3.
5. Detach the MiniProg3 and remove power supply to the PSoC DVK.
6. Connect the CY8CKIT-025 EBK and TME EBK to the PSoC DVK, as shown in Figure 5-23.
7. Power the PSoC DVK and TME EBK, as shown in Figure 5-23.
8. Press the **Reset** (SW4) button on the PSoC 3 DVK to run the newly programmed firmware image.
9. On reset, the LCD displays a welcome message and starts with the Zone 1 temperature reading and the corresponding temperature calculation algorithm. The desired and actual fan speeds corresponding to that zone are also displayed.
10. Use the push button SW1 to view the different zone characteristics.

5.4.2.3 Testing the Project

In this project, the fan temperature is controlled based on the temperature of the corresponding thermal zone. To test the closed loop speed control of Fan 3, vary the potentiometer on the DVK. This simulates the condition of changing the corresponding zone temperature in the project. The speed of Fan 3 is adjusted according to the temperature vs speed graph shown in Figure 5-34. To test the closed loop speed control of Fan 4, touch the RTD sensor on the CY8CKIT-025 EBK. This results in variations in the measured RTD temperature. The speed of Fan 4 is adjusted according to the temperature vs speed graph shown in Figure 5-35.

On reset, the LCD displays the characteristics of Zone 1. These characteristics include the zone temperature, temperature calculation algorithm, desired fan speed, and the actual fan speed. The fan speeds will match the corresponding zone thermal profiles. Press the push button (SW1 on CY8CKIT-001 or SW2 on CY8CKIT-030) once to view the characteristics of Zone 2. Press the push button (SW1 on CY8CKIT-001 or SW2 on CY8CKIT-030) again to view the individual sensor temperatures for each zone and the weighted composite zone temperature. Pressing the button again will repeat this display sequence.

5.4.3 Project Details

This project demonstrates how the temperature sensors in the CY8CKIT-025 EBK combined with the fans on the TME EBK can create a complete thermal management system. The example shows how to combine temperature readings from a number of temperature sensors in a variety of ways and use the composite temperature to set desired fan speeds according to a customizable transfer function.

The thermal management example uses the concept of a "Thermal Zone", which describes how to :

- combine multiple temperature sensor readings to form a composite zone temperature
- map the zone temperature to a fan speed

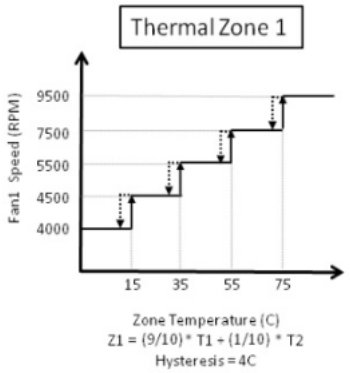
By this definition, each fan is controlled according to its own independent thermal zone. This example has two thermal zones because the TME EBK has only two fans installed. Algorithms currently implemented to combine multiple temperature sensors into a composite zone temperature include: straight average, weighted average, and maximum.

In this example, the weighted method is used on both fans. A zone temperature-to-fan speed transfer function is then definable for each zone. Linear and table driven transfer functions are currently implemented. This example uses the table driven transfer function on both fans; that is, a look-up table maps the composite zone temperature-to-fan speed.

This example is a simulation of a thermal management system. Zone 1, corresponding to Fan 3 in Figure 5-34 combines temperature measurements from two temperature sensors (potentiometer

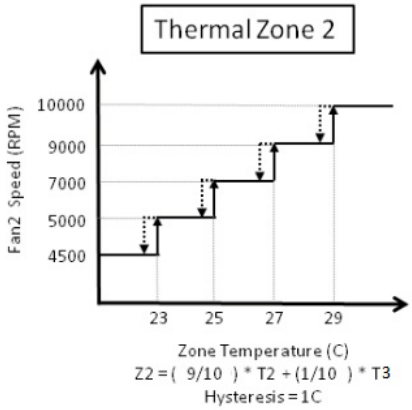
temperature emulator and RTD). The potentiometer temperature emulator allows easy demonstration of fan control over a wide simulated temperature range without the need for an environmental chamber to cycle through temperatures. In Zone 1, the temperature sensors are combined using a weighted average where the potentiometer is given 90 percent weight and the RTD is given 10 percent weight. Adjust the potentiometer (R20 on the CY8CKIT-001 DVK and R56 on the CY8CKIT-030 DVK) to vary the simulated temperature value in the approximate range of 15 °C to 100 °C. The Zone 1 speed transfer function is table driven and follows the profile shown in Figure 5-34.

Figure 5-34. Thermal Management System Project - Zone1 Thermal Profile



Zone 2, corresponding to Fan 4 in Figure 5-35, consists of two temperature sensors and a single fan. The Zone 2 speed transfer function is table driven and is shown in Figure 5-35. Note that the temperature range is very narrow and close to room temperature. This is to allow simple testing at room by touching a temperature sensor with a warm finger to cause a fan speed change. In Zone 2, the temperature sensors are combined using a weighted average where the RTD is given 90 percent of the weight and the diode is given 10 percent weight. In this example, the RTD temperature reading dominates the overall zone temperature calculation.

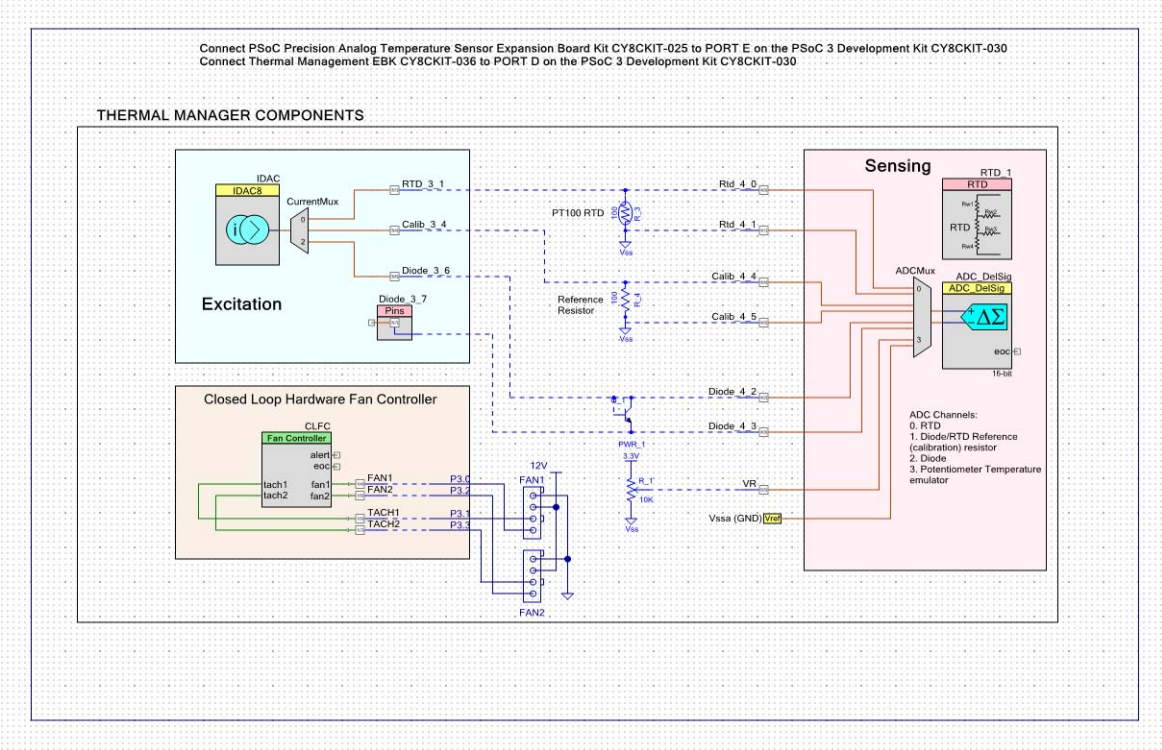
Figure 5-35. Thermal Management System Project - Zone2 Thermal Profile



5.4.3.1 *Project Schematic*

Figure 5-36 shows the PSoC Creator schematic of the project.

Figure 5-36. PSoC Creator Schematic



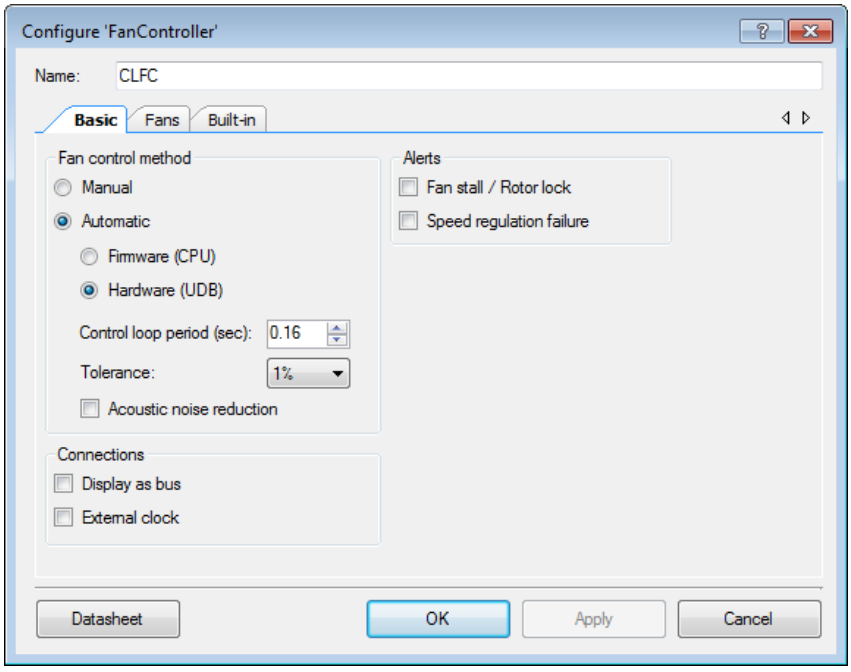
A four-channel ADC is used to measure the voltages across the RTD, diode, RTD/diode current calibration resistor, and potentiometer, which simulates an analog temperature sensor. IDAC is used to provide the excitation currents for the diode, RTD, and calibration resistor. The closed loop fan controller component senses and controls the fan speed.

Application notes AN60590 and AN70698 provide detailed explanations on temperature measurement using diode and RTD, respectively. Application note AN66627 describes the fan controller operation in detail.

5.4.3.2 *Component Configuration*

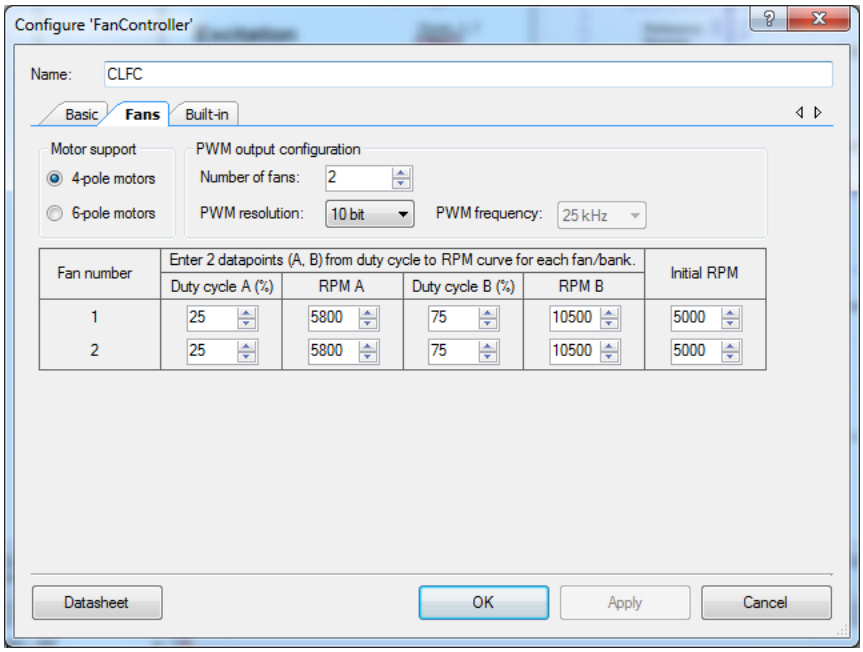
Figure 5-37 and Figure 5-38 show the FanController component configuration.

Figure 5-37. FanController Component Configuration - Basic Tab



The Fan Control Method is chosen as hardware (UDB), which means that the fan speed is controlled using the hardware blocks inside PSoC without any CPU intervention. See the FanController component datasheet in PSoC Creator for details on the other parameters.

Figure 5-38. FanController Component Configuration - Fans Tab



This tab provides the option to choose the number of fans and the speed to RPM mapping of each fan. The individual fan parameters are configured, as shown in Figure 5-38. The configuration is done based on the datasheet specifications of the fans used on the TME EBK.

Figure 5-39 shows the ADC configuration to measure the potentiometer voltage. The ±6.144 V voltage range is chosen because the potentiometer voltage can vary from 0 to VDDA. The Bypass Buffer option is selected for the Buffer Mode parameter to measure voltage near the supply rails.

Figure 5-39. ADC Configuration for Potentiometer Voltage Measurement

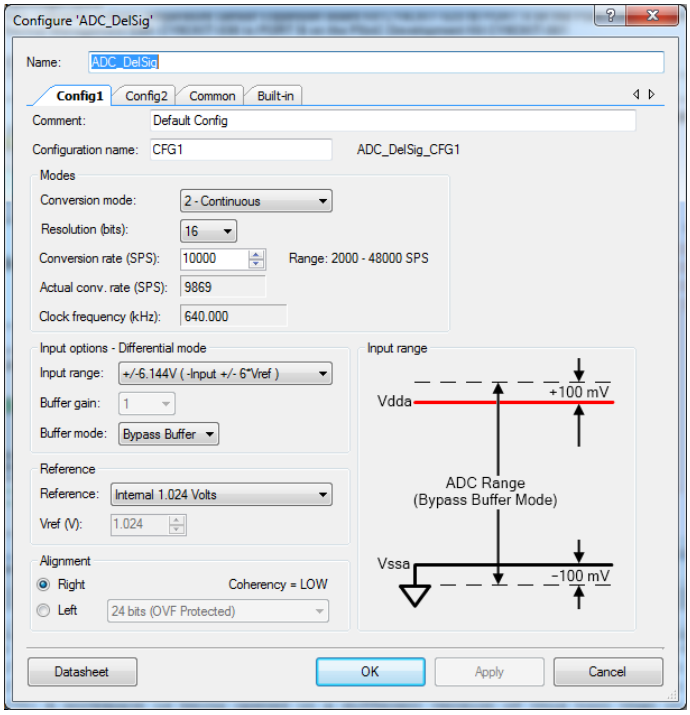


Figure 5-40 shows the ADC configuration to measure voltage across the sensors (diode and RTD). The Resolution parameter is set to 20 bits to perform high-resolution temperature measurements.

Figure 5-40. ADC Configuration for Temperature Sensor Measurement

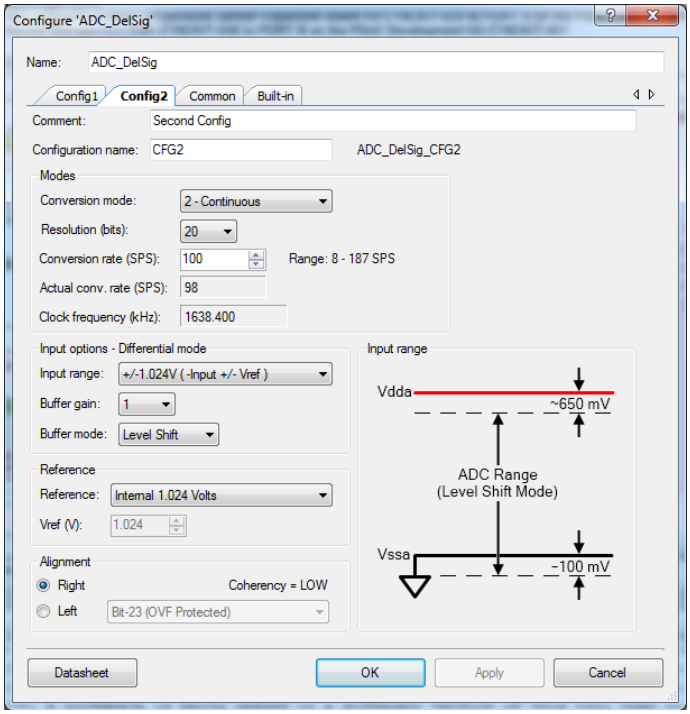
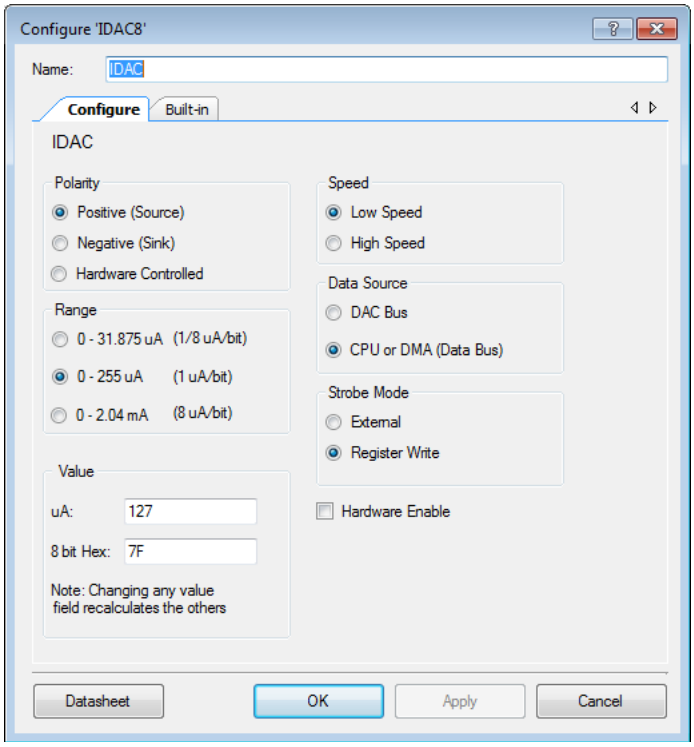


Figure 5-41 shows the IDAC component configuration. The Polarity parameter of the IDAC is set to the current sourcing mode and the Speed parameter is set to **High Speed** to have a faster settling time for the IDAC output.

Figure 5-41. IDAC Configuration



5.4.3.3 *Firmware Description and Flowchart*

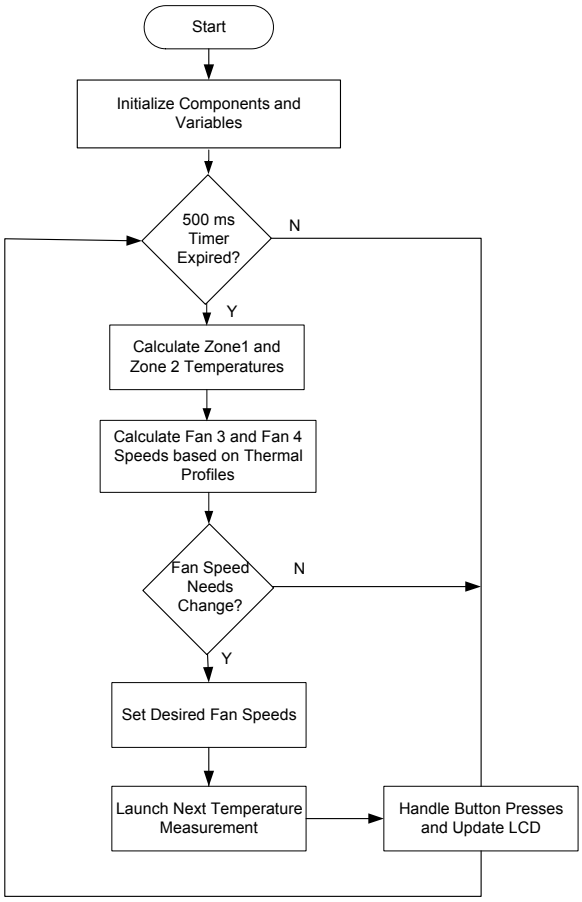
The thermal management system example consists of the main application and thermal manager. The main application is responsible for the user interface and for periodically calling the thermal manager. The application implementation is available in *main.c* and on the Test Application tab of the project schematic. The thermal manager implementation is available in *ThermalManager.c* and on the Thermal Manager tab of the project schematic.

The main application must call ThermalManager_Start() to initialize the thermal manager. Then it must periodically call ServiceThermalManager() to run temperature and speed updates. In this example, this is done every 500 ms but can be changed by modifying
`#define THERMAL_UPDATE_MS_RATE in main.c.`

All the parameters that define the zone composite temperature sensor algorithm and the zone temperature-to-fan speed algorithm are defined in the beginning of *ThermalManager.c*. To modify these settings, refer to *ThermalManager.h* for the relevant keywords.

The following flowchart shows the basic function of the thermal manager along with the APIs in *ThermalManager.c* that implement the main service loop.

Figure 5-42. Thermal Manager Flowchart



In this example, the LCD displays status information about thermal management system across three screens. You can cycle through the status screens by pressing SW1 on the CY8CKIT-001 DVK or SW2 on the CY8CKIT-030 DVK. The three screens are:

- Screen 1 - Zone 1 Summary

  This screen displays the current status of Zone 1. Line 1 displays the zone number, current composite zone temperature, and the zone temperature calculation algorithm used. Line 2 displays the desired fan speed and the actual fan speed for Zone 1.

Figure 5-43. Zone 1 Summary



- Screen 2 - Zone 2 Summary

  This screen displays the current status of Zone 2. Line 1 displays the zone number, current composite zone temperature, and the zone temperature calculation algorithm used. Line 2 displays the desired fan speed and the actual fan speed for Zone 2.

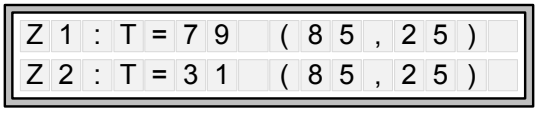
Figure 5-44. Zone 2 Summary



- Screen 3 - Temperature Sensors Summary

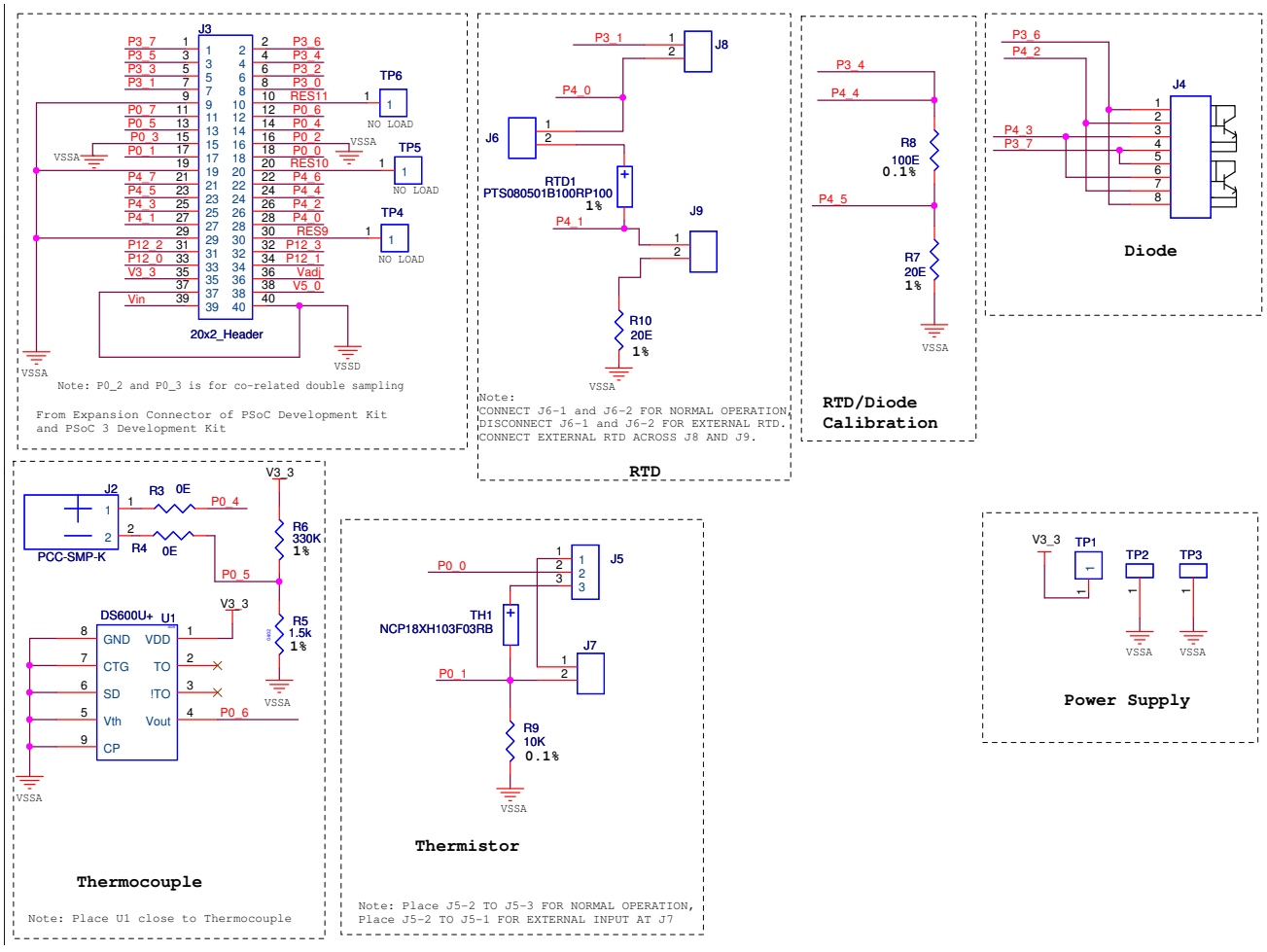
  This screen displays the current temperature sensor readings for all sensors in the system. Line 1 displays the Zone 1 temperature sensor values. The temperature on the extreme left is the zone's composite temperature followed by the temperatures of each contributing sensor. Line 2 displays the same information for Zone 2.

Figure 5-45. Temperature Sensors Summary



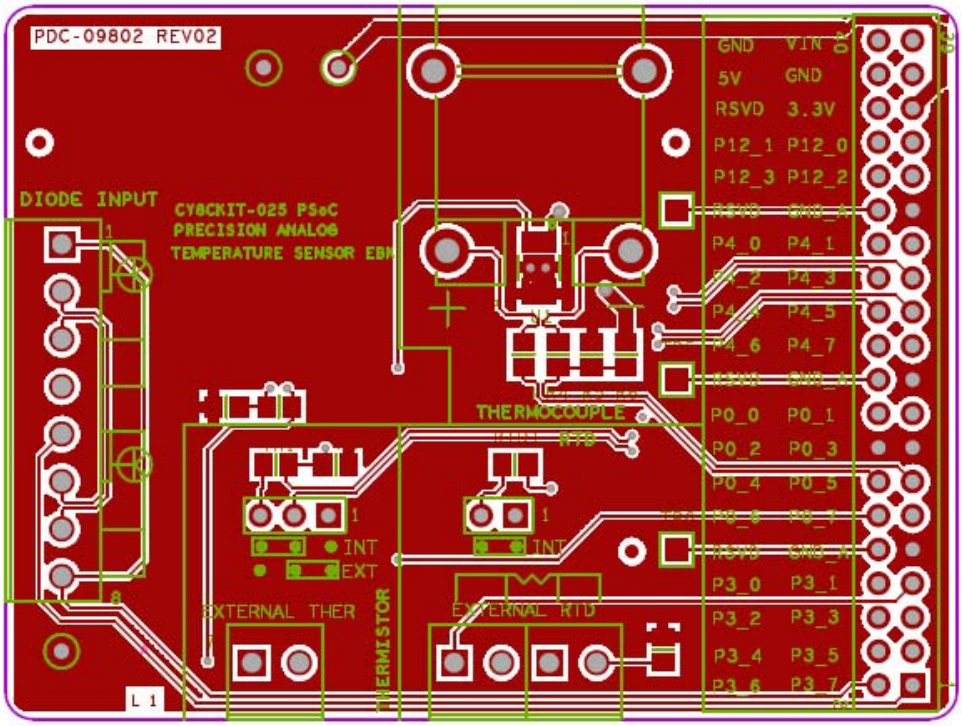
A. Appendix

A.1 Schematic

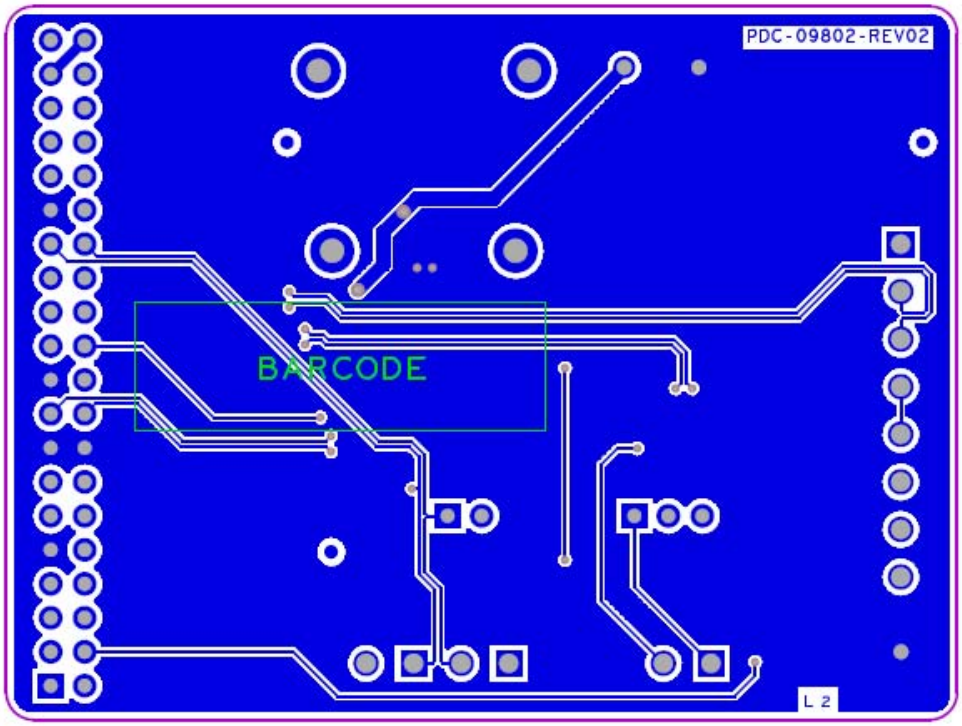


A.2 Board Layout

A.2.1 PDC-09802 Top



A.2.2 PDC-09802 Bottom



A.3 Bill of Materials (BOM)

Item	Qty.	Reference	Value	Description	Manufacturer	Manufacturer Part Number
1				PCB	Cypress	PDC-09802 Rev02
2	1	J3	40 Pin Header	CONN HEADER .100 DUAL R/A 40POS	Protectron Electromech	P9104-40-12-1
3	1	J2	Thermocople Connector	CONN OMEGA Thermocouple Type K	OMEGA	PCC-SMP-K-5
4	1	TH1	THERMISTOR NTC 10K OHM 1%	THERMISTOR 10K OHM NTC 0603	Murata	NCP18XH103F03RB
5	1	R9	10K, 0.1%	RES 10K OHM 1/8W .1% 0805 SMD	Panasonic - ECG	ERA-6AEB103V
6	1	J4	8 PIN Connector	CONN TERM BLOCK T/H 8POS 3.5MM	Phoenix Contact	1984675
7	3	J7, J8, J9	2 Pin Jumper	Terminal Blocks PT 1.5/2-3.5H 2POS HRZ 3.5mm SCREW	Phoenix Contact	1984617
8	2	R7, R10	20E, 1%	RES 20.0 OHM 1/8W 1% 0805 SMD	Vishay/Dale	CRCW080520R0FKEA
9	1	RTD1	TEMP SENSOR RTD	TEMP SENSOR RTD 100 OHM 0805	Vishay/Beyschlag	PTS080501B100RP100
10	1	R8	100E, 0.1%	RES 100 OHM 1/8W 0.1% 0805 SMD	Panasonic - ECG	ERA-6AEB101V
11	1	J6	2p_jumper	CONN HEADER VERT SGL 2POS GOLD	3M	961102-6404-AR
12	2	R3, R4	0E Resistor	RES 0.0 OHM 1/8W 0805 SMD	Panasonic - ECG	ERJ-6GEY0R00V
13	1	U1	DS600U+	IC SENSOR TEMP 8-USOP	Maxim Integrated Products	DS600U+
14	1	R6	330K, 1%	RES 330K OHM 1/8W 1% 0805 SMD	Panasonic - ECG	ERJ-6ENF3303V
15	1	R5	1.5K, 1%	RES 1.50K OHM 1/8W 1% 0805 SMD	Panasonic - ECG	ERJ-6ENF1501V
16	1	J5	3p_jumper	CONN HEADER VERT SGL 3POS GOLD	3M	961103-6404-AR
No Load Components						
17	1	TP1	RED TP	TEST POINT PC MINI .040"D RED	Keystone Electronics	5000
18	2	TP2, TP3	BLACK TP	TEST POINT PC MINI .040"D RED	Keystone Electronics	5001
19	3	TP4, TP5, TP6	PADS	PADS		
Special Jumper Installation Instructions						
20	1	J6	Install jumper across pins 1 and 2	Rectangular Connectors MINI JUMPER GF 13.5 CLOSE TYPE BLACK	Kobiconn	151-8030-E
21	1	J5	Install jumper across pins 3 and 2	Rectangular Connectors MINI JUMPER GF 13.5 CLOSE TYPE BLACK	Kobiconn	151-8030-E
Install as per Assembly drawing						
22	3			BUMPER CLEAR .500X.23" SQUARE	Richco Plastic Co	RBS-3R

A.4 Regulatory Compliance Information

The CY8CKIT-025 PSoC Precision Analog Temperature Sensor Expansion Board has been tested and verified to comply with the following electromagnetic compatibility (EMC) regulations.

- EN 55022:2010 Class A - Emissions
- EN 55024:2010 Class A - Immunity

Revision History

Document Revision History

Document Title: CY8CKIT-025 PSoC® Precision Analog Temperature Sensor Expansion Board Kit Guide Document Number: 001-65791				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
**	3222188	04/11/11	PFZ	Initial version of kit guide.
*A	3455347	11/29/11	SASH	Updated PSoC Creator version.
*B	3561312	03/24/12	RKAD	Updated existing project; added two new projects.
*C	3639690	06/07/12	PFZ	Updates to Additional Resources section. Updated images in the Example Projects chapter.
*D	3737398	09/07/12	PFZ	Document changes to reflect support for PSoC5
*E	3954642	03/28/13	VRNK	Added the Safety section. Updated images in the Code Examples chapter. Minor updates throughout the document.
*F	3988373	04/26/13	VRNK	Updated "CY8CKIT-001 PSoC DVK Thermistor Temperature" on page 40 and "CY8CKIT-030 PSoC DVK" on page 53.
*G	4121210	09/12/13	SASH	Updated images for PSoC Creator ECR. Added section 5.1.1
*H	4335470	04/07/14	RKAD	Updated section 5.2.2.4.5.
*I	5406476	08/17/16	SRDS	Updated PSoC Creator version to PSoC Creator 3.3 CP3; updated related images. Updated PSoC Programmer version to 3.24.2. Added information about floating point variables in "Migrating Projects to use with CY8CKIT-050 and CY8CKIT-010" on page 31. Updated to new template.
*J	5446907	09/23/2016	SRDS / SAGA	Updated Software Installation chapter on page 15: Updated Figure 2-2. Updated Figure 2-4.
*K	5705955	04/21/2017	RKAD	Updated to new template. Completing Sunset Review.