

## Objective

This example demonstrates how to use a PSoC® 3 or PSoC 5LP device to measure temperature using a thermocouple. This example does not explain thermocouple calibration; for that, see [CE219929](#).

## Overview

Thermocouples are sensors commonly used to measure temperature in applications that measure very high temperatures, or require a sensor that is rugged. A thermocouple produces a small change in voltage per change in temperature ( $40 \mu\text{V}/^\circ\text{C}$ ). This requires a precise and accurate measurement system. The Delta Sigma ADC on PSoC 3 and PSoC 5LP devices has the required precision and accuracy to measure temperature using a thermocouple with no need for external amplifiers or ADCs. For full details on the theory of thermocouple measurement, see [AN75511](#).

This code example demonstrates how to configure a PSoC 3 or PSoC 5LP device to accurately measure temperature using a thermocouple. The measured temperature is displayed on an LCD on either CY8CKIT-030 or CY8CKIT-050. This code example allows the use of either an IC temperature sensor or thermistor for cold junction compensation. CY8CKIT-025 is required to run this example.

## Requirements

**Tool:** PSoC Creator™ 4.2 or newer

**Programming Language:** C (Arm® GCC 5.4.1, Arm MDK 5.22, DP8051 Keil 9.51)

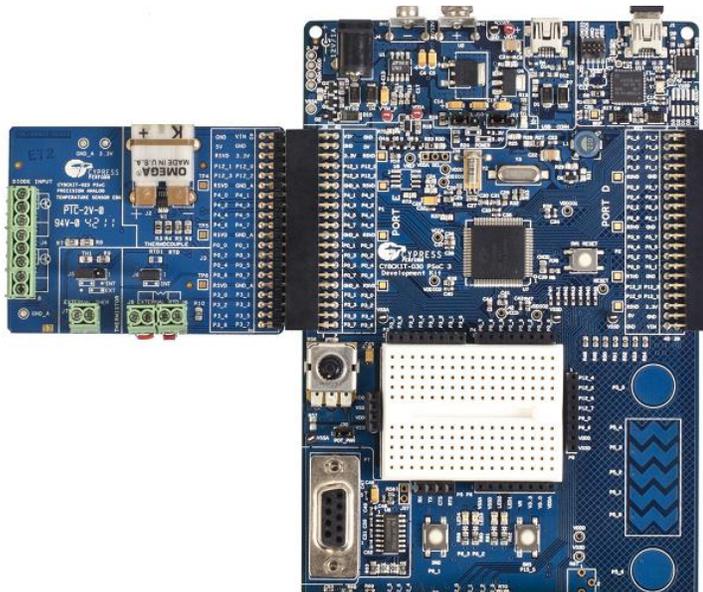
**Associated Parts:** All PSoC 3 and PSoC 5LP parts

**Related Hardware:** [CY8CKIT-050](#), [CY8CKIT-030](#), [CY8CKIT-025](#)

## Hardware Setup

1. Plug [CY8CKIT-025](#) into PORT E of either [CY8CKIT-030](#) or [CY8CKIT-050](#), as [Figure 1](#) shows.

Figure 1. CY8CKIT-025 Plugged into CY8CKIT-030



2. Plug the thermocouple provided in [CY8CKIT-025](#) into the thermocouple connector as [Figure 2](#) shows.

Figure 2. Thermocouple Section of the EBK



3. Connect an LCD to the LCD port on [CY8CKIT-030](#) or [CY8CKIT-050](#).
4. Plug a USB cable into a PC and then into the programming USB connector on the kit.

## Software Setup

To modify the filter characteristics of the IIR filter go to *main.h* and adjust:

```
/* Filter coefficient for sensors */
#define MAX_FILTER_COEFF_SHIFT 8
#define TC_FILTER_COEFF_SHIFT 6

/* Filter Feed Forward Term. It is set equal to 640 ADC counts or 2°C.
 * 256 (2^MAX_FILTER_COEFF_SHIFT) is to make the comparison fair */
#define FILTER_FEEDFORWARD ((int32)640 * 256)
```

**Important:** If you create this project from an empty project, go to the System tab and change the heap size from 0x80 to 2048. This is needed for the `sprintf` function.

**Important:** If you create this project from an empty project, right-click on the project and select build setting. Go to **Linker > command line** and add “-u\_printf\_float” without quotes.

## Math Library

The Thermistor Calculator component used in this design requires the math library to be linked into the design. That step is already done for you in the attached example.

If you are creating a new PSoC Creator project and using the Thermistor Calculator Component, right-click on your project, and then select **Build Settings...** Expand the compiler group by pressing the **+** button. The name of the compiler will depend on the compiler used in the design; for PSoC Creator 4.2, it is **ARM GCC 5.4-2016-q2-update**. Click on **Linker**, in **Additional Libraries** add m, and then click **Apply**.

When using third-party IDEs, the math library will need to be added as well. The steps to do this are specific to each IDE and not described in this document. See your IDE's Help.

## Operation

1. Complete the steps in the previous sections.
2. Build either the mid-end or low-end project, and program it into the kit.  
The LCD displays thermocouple temperature and cold junction temperature.
3. Press SW2 on the DVK to toggle the cold junction temperature source between the thermistor and the IC.
4. Change the temperature of the thermocouple and observe the measurement change on the LCD.

## Design

This code example includes two projects: *CE219005\_PSoC3\_5\_Thermocouple\_Mid\_End* and *CE219005\_PSoC3\_5\_Thermocouple\_Low\_End*. These two projects correspond to the mid-end and low-end segments described in [AN75511](#). The mid-end provides a resolution of 0.1 °C. The low-end has a resolution of 1 °C. The projects are nearly identical except for the configuration of the ADC and software filtering applied.

Figure 3 shows the thermocouple measurement circuit (PSoC Creator schematic). The circuit has a five-channel ADC, the Thermistor Component, and a character LCD. The five ADC channels and their purposes are listed in Table 1.

Table 1. Five ADC Channels

Channel	Connection	Measurement
0	Thermocouple	Thermocouple Voltage (Thermo-emf)
1	IC voltage output	Cold junction temperature
2	Thermistor voltage	Cold junction temperature
3	Thermistor ref	Cold junction temperature
4	Short	Offset

As shown in Table 1, a thermistor or an IC can be used for measuring cold junction temperature. A switch is used in the project to select between the two cold junction sensors.

According to the National Institute of Standards and Technology (NIST) thermocouple tables, at  $-270\text{ }^{\circ}\text{C}$ , the thermocouple gives an output voltage of  $-6.458\text{ mV}$ . Because PSoC devices are single-supply, the voltage on the input pins must be kept above  $0\text{ V}$ . To accomplish this, external resistors are added to the negative input of the thermocouple to add a small  $15\text{-mV}$  bias. These resistors have been populated on CY8CKIT-025.

Figure 3(a). Thermocouple Measurement Circuit

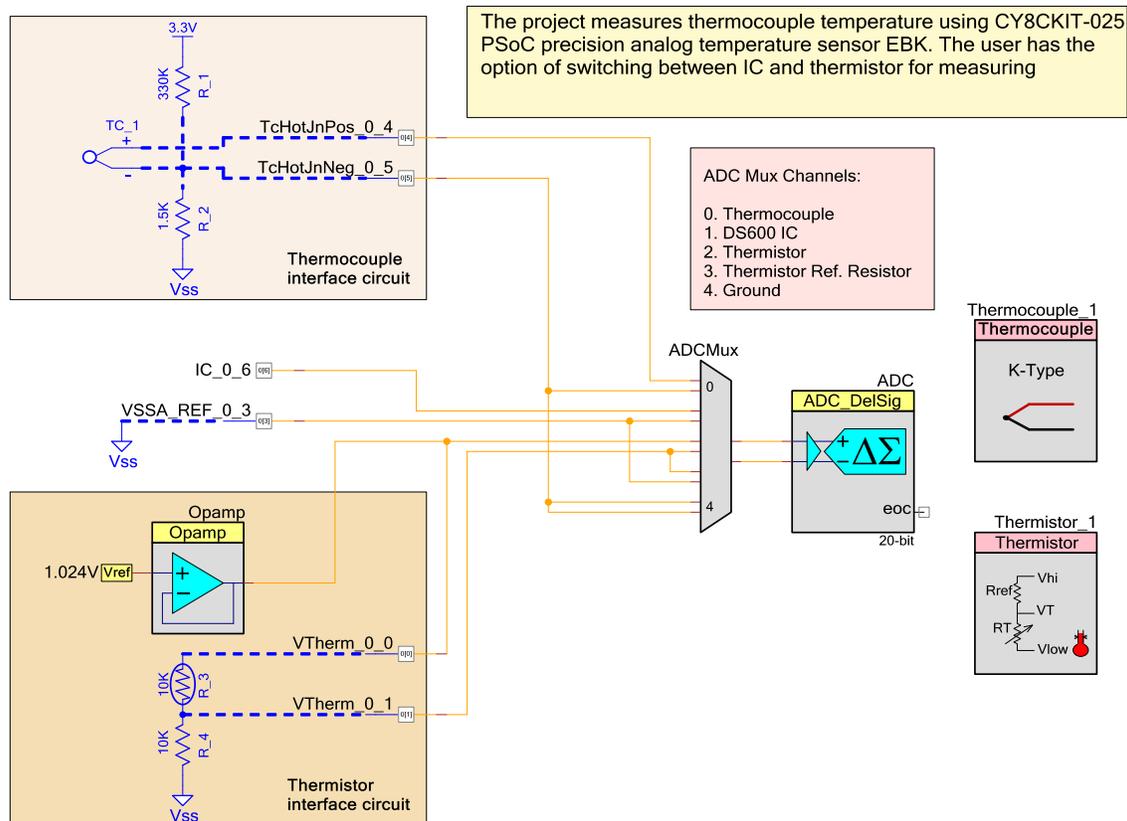
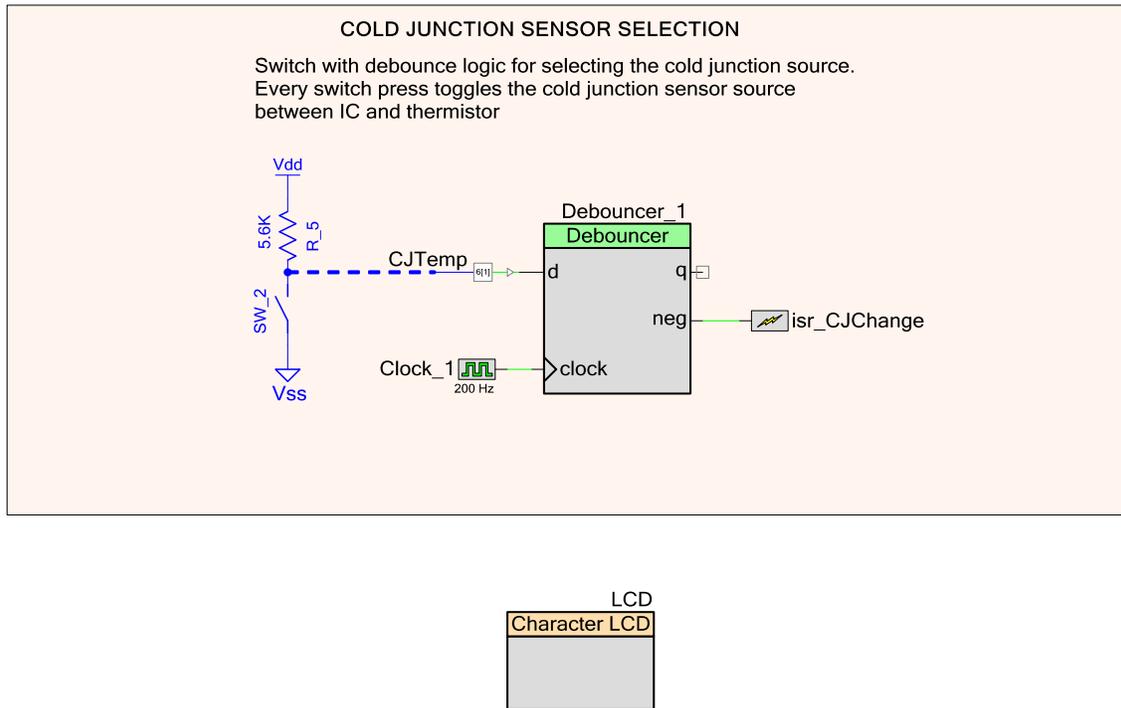


Figure 3(b). Thermocouple Measurement Circuit



### ADC Configurations

One useful feature of the Delta Sigma ADC is that it can be reconfigured during runtime to different configurations. For the mid-end project, three configurations are used:

1. IC\_Config – Used to measure temperature using an IC temperature sensor
2. TC\_Config – Used to measure temperature using a thermocouple
3. Therm\_Config – Used to measure temperature using a thermistor

For the low-end project, two configurations are used:

1. IC\_Therm\_Config – Used to measure temperature using an IC temperature sensor and a thermistor
2. TC\_Config – Used to measure temperature using a thermocouple

### Offset Cancellation

A K-type thermocouple has a typical sensitivity of approximately  $40 \mu\text{V}/^\circ\text{C}$ . A  $40\text{-}\mu\text{V}$  offset results in a  $1^\circ\text{C}$  temperature error. Therefore, eliminating the offset is important. Offset cancellation is done by correlated double sampling (CDS). In this technique, the offset is measured and subtracted after every voltage reading.

CDS not only removes the offset drift and reduces the low-frequency noise, but also reduces the ADC sample rate by 50 percent. Offset can be measured in a number of ways. See [AN66444 - PSoC® 3 and PSoC 5LP Correlated Double Sampling](#) for details on different ways to measure offset. Offset in this case is measured using ADC channel four (see [Figure 3](#)).

### Broken Thermocouple

If the thermocouple wire breaks, the small negative voltage applied to the negative terminal of the thermocouple connector (see [Figure 3 \(a\)](#)) takes the ADC reading to a large negative value. Checking the ADC for a large negative value ( $< -10 \text{ mV}$ ) detects a broken thermocouple connection. This code example detects a broken thermocouple connection and displays a broken alert on the LCD if the thermocouple is broken.

### Filtering the Thermocouple Output

The thermocouple output is filtered using a software infinite impulse response (IIR) filter to reduce the noise and improve the noise-free temperature resolution. See [AN2099](#) for details on the filter.

For the mid-end project, the firmware IIR filter used has an attenuation factor of 32. From Table 1 in AN2099, the temperature settling time (0.1%) would be 219 cycles. For the project, the cycle time is about 50 ms, resulting in a temperature settling time of 10 s.

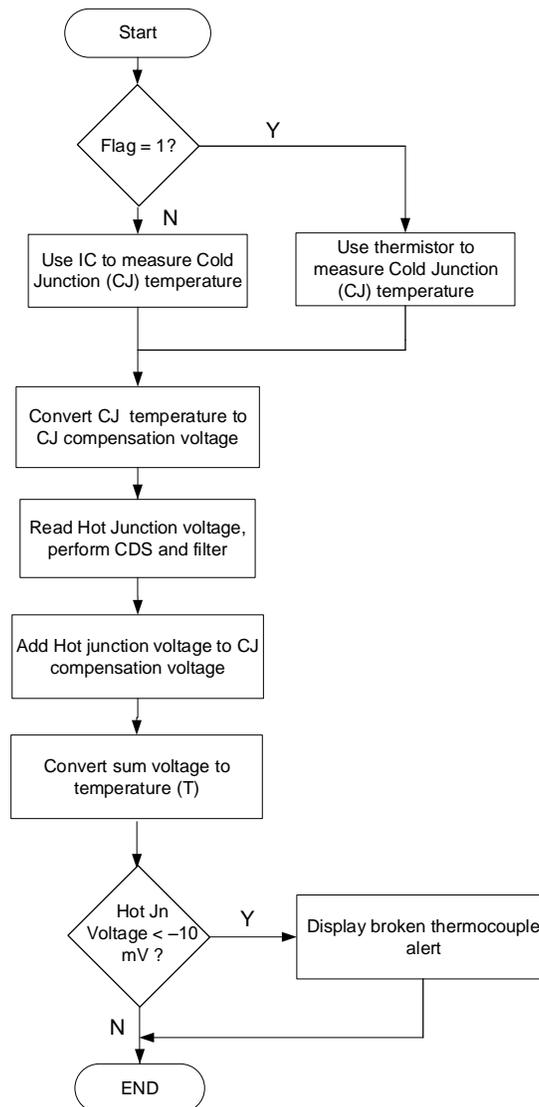
The IIR filter has a feed forward term that ensures that the temperature settles to within 2 °C in 50 ms. That is, if the source temperature changes from 50 °C to 150 °C, the temperature shown by PSoC will reach 148 °C in 50 ms and 149.9 °C in 10 s.

### Firmware Flow

An interrupt is triggered when the switch is pressed (see Figure 3 (b)). A flag is toggled in the interrupt service routine (ISR), changing the cold junction temperature source between the IC and the thermistor.

APIs generated by the thermocouple Component are used for converting thermocouple voltage to temperature and vice versa.

Figure 4. Firmware Flow



## Design Considerations

CY8CKIT-025 provides a connector for a K-type thermocouple. The cold junction sensor, DS600 IC, is placed very near to the cold junction terminals. Silkscreen marked 'U1' shows the position of the IC on the board. Ideally, an isothermal connection should be provided between the cold junction sensor and the cold junction terminals as described in AN75511. This requires a material with very good thermal conductivity providing a thermal connection between the thermal pad of the IC and the cold junction terminals. In cases where this is not possible, place the cold junction sensor very near to the cold junction terminals as it has been done in CY8CKIT-025. The temperature difference between the cold junction terminals and the IC is expected to be approximately 0.5 °C in this case.

Note that the thermistor, RTD, or diode on CY8CKIT-025 can also be used for cold junction temperature measurement. The included project gives the option of using a thermistor for cold junction compensation. Similarly, the RTD or diode can be used for cold junction temperature measurement. Cypress application notes AN66477 - *Temperature Measurement with a Thermistor*, AN70698 - *Temperature Measurement with an RTD*, and AN60590 - *Diode Temperature Measurement* explain thermistor, RTD, and diode temperature measurements with PSoC devices in detail.

However, these temperature sensors (RTD, thermistor, and diode) are farther from the cold junction than the IC and the temperature difference between the cold junction and these sensors will be higher if there is air flow. With no significant air flow, the temperature difference between the cold junction terminals and the other sensors is expected to be less than 1 °C.

## Components

Table 2 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 2. List of PSoC Creator Components

Component	Hardware Resources
Delta Sigma ADC	1 Delta Sigma ADC
Opamp	1 Opamp
Thermocouple	Firmware
Thermistor	Firmware
AMux	Analog Routing
Debouncer	1 UDB
Character LCD	7 Pins

## Parameter Settings

The following tables list the Component parameters that have been changed from the default configuration.

Table 3. Mid End ADC IC Temp Config

Parameter	Change
Conversion rate (SPS)	10000 → 2000
Buffer Mode	"Rail to Rail" → "Level Shift"

Table 4. Mid End ADC Thermocouple Config

Parameter	Change
Resolution	16 → 20
Conversion rate (SPS)	10000 → 60
Input Range	±1.024 V → ± 0.064 V
Buffer Mode	"Rail to Rail" → "Level Shift"

Table 5. Mid-End ADC Thermistor Config

Parameter	Change
Buffer Mode	“Rail to Rail” → “Level Shift”

Table 6. Low-End ADC IC and Thermistor Config

Parameter	Change
Resolution	16 → 12
Buffer Mode	“Rail to Rail” → “Level Shift”

Table 7. Low End ADC Thermocouple Config

Parameter	Change
Resolution	16 → 12
Input Range	$\pm 1.024V \rightarrow \pm 0.064V$
Buffer Mode	“Rail to Rail” → “Level Shift”

Figure 5. Opamp Settings

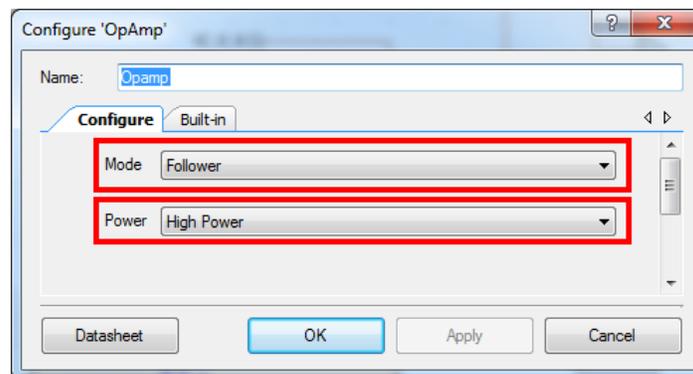
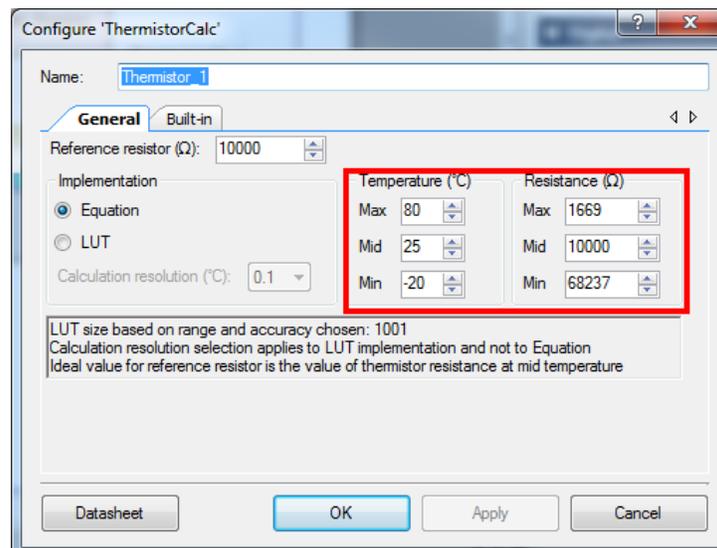


Figure 6. Thermistor Settings



## Design-Wide Resources

Table 8. Pin locations for PSoC 3 and PSoC 5LP Devices

Pin Name	PSoC 3/ PSoC 5LP Device Pin Location
LCD	P2[6:0]
CJTemp	P6[1]
IC	P0[6]
TcHotJnNeg	P0[5]
TcHotJnPos	P0[4]
VSSA_REF	P0[3]
VThermPos	P0[0]
VThermNeg	P0[1]

## Related Documents

Code Example		
<a href="#">CE219929</a>	PSoC 3 and PSoC 5LP Thermocouple Calibration	Explains how to calibrate a thermocouple measurement system to achieve 0.1° C accuracy
Application Notes		
<a href="#">AN75511</a>	Temperature Measurement with Thermocouples	Describes how PSoC 3 and PSoC 5LP devices can be used to measure a thermocouple.
<a href="#">AN66444</a>	Correlated Double Sampling to Reduce Offset, Drift, and Low Frequency Noise	Describes how to implement correlated double sampling on a PSoC device.
<a href="#">AN2099</a>	Single Pole Infinite Impulse Response (IIR) Filters	Describes how to implement a software IIR filter.
PSoC Creator Component Datasheets		
<a href="#">Thermocouple Component</a>	Thermocouple Calculator	
<a href="#">Thermistor Component</a>	Thermistor Calculator	
<a href="#">Delta Sigma ADC</a>	Precision ADC component	
Device Documentation		
<a href="#">PSoC 3 Datasheets</a>	<a href="#">PSoC 3 Technical Reference Manuals</a>	
<a href="#">PSoC 5LP Datasheets</a>	<a href="#">PSoC 5LP Technical Reference Manuals</a>	
Development Kit (DVK) Documentation		
<a href="#">PSoC 3 and PSoC 5LP Kits</a>		
<a href="#">CY8CKIT-025</a>		

## Document History

Document Title: CE219905 - PSoC 3 and PSoC 5LP: Temperature Sensing with a Thermocouple  
Document Number: 002-19905

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5791140	TDU	07/10/2018	New spec

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

### Products

Arm® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
Microcontrollers	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Power Management ICs	<a href="http://cypress.com/pmic">cypress.com/pmic</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless Connectivity	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

### Cypress Developer Community

[Community Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

### Technical Support

[cypress.com/support](http://cypress.com/support)

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2018. 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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. 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, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.