CE219929 – PSoC 3 and PSoC 5LP Thermocouple Calibration

Objective

This example demonstrates how to use a PSoC 3 or PSoC 5LP device to measure temperature using a thermocouple and how to calibrate the signal chain to achieve 0.1% accuracy with 0.01°C resolution. For a basic project implementing temperature measurement using a thermocouple without calibration and high-end accuracy, see CE219905.

Overview

AN75511 describes the theory of thermocouple operation and measurement. CE219905 describes how to measure a thermocouple using either a PSoC 3 or PSoC 5LP device. This code example shows how to achieve the high-end accuracy described in AN75511. Specifically, this code example shows how to achieve an accuracy of 0.1% with 0.01 °C precision. This accuracy is achieved through calibration. There are two types of calibration – an initial gain offset calibration and a drift calibration.

Requirements

Tool: PSoC Creator™ 4.2 or newer

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

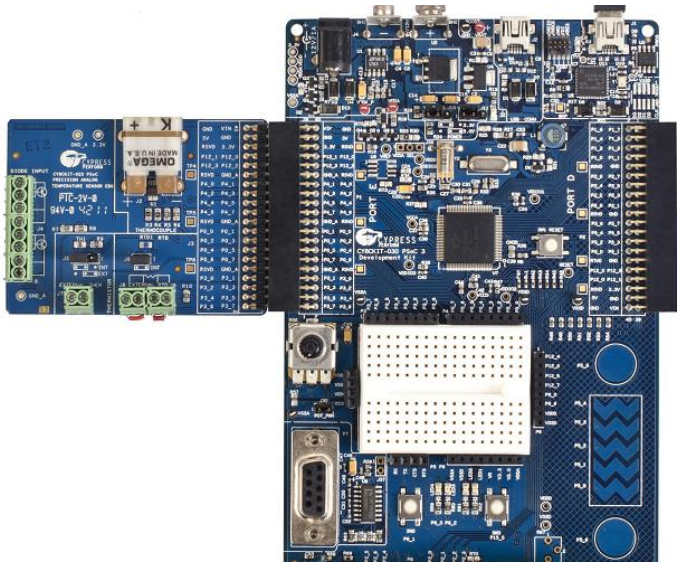
Associated Parts: All PSoC 3 and PSoC 5LP parts

Related Hardware: CY8CKIT-050, CY8CKIT-030, CY8CKIT-025

Hardware Setup

1. Populate an LM4140 (3 ppm/°C) reference on CY8CKIT-030 or CY8CKIT-050 (position U6). Populate R34 (0 Ω), C24 (1 µF) and R37 (0 Ω) on CY8CKIT-030 or CY8CKIT-050. Now, the 1.024-V reference is available on pin 3[2]. It can also be used as an external reference.
2. In the prototyping space, add a resistor divider from pin 3[2] to ground to reduce 1.024 V down to 48.7 mV. Choose R1 = 20 kΩ (0.1%, 5 ppm/°C drift) and R2 = 1 kΩ (0.1%, 5 ppm/°C drift).
3. Connect the 48.7-mV input to pin 0_7.
4. 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



5. Plug the thermocouple provided in CY8CKIT-025 into the thermocouple connector as Figure 2 shows.

Figure 2. Thermocouple Section of the EBK



6. Connect an LCD to the LCD port on CY8CKIT-030 or CY8CKIT-050.
7. Plug a USB cable into a PC and then into the programming USB connector on the kit.

Software Setup

In *main.h*, update the FULL_SCALE define to match your full-scale calibration voltage.

```
/* Calibration input in microvolts */
#define FULL_SCALE                                      44760
```

In *main.h*, adjust the temperature threshold for doing a drift calibration by modifying the following code (this is in 100ths of a degree Celsius).

```
/* Temperature change for which gain correction has to be done in 100ths of a
degree C. */
#define GAIN_CORRECT_THRESHOLD                  500
```

To adjust the IIR filter coefficients go to *removeOffsetNoise.c*. Adjust the initial values for feedforward, and filter_coeff_shift. Also, go to *main.h* and change the value of MAX_FILTER_COEFF_SHIFT. For details, see AN2099.

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 complier 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 Hardware Setup section.
2. In *main.h*, change `#define FULL_SCALE 63777` to match the voltage of your gain calibration in µV.
3. Build the project, and program it into the kit.
4. If you wish to calibrate, press SW2. It first calibrates the offset. Short the terminals of the thermocouple connector together. After this is done, press SW2 again. If you don't wish to calibrate just wait.
5. Next, it will ask for a full scale input (FS). Connect a full scale ~64-mV signal across the thermocouple connector. After this is done, press SW2 again.
6. Next, it will ask if you wish to do drift calibration. Do this only if you have a stable voltage connected to P0.7. If you don't wish to do this, just wait.
7. The LCD displays the thermocouple temperature and cold junction temperature.
8. Press SW2 on the DVK to toggle the cold junction temperature source between the thermistor and the IC.
9. Change the temperature of the thermocouple and observe the measurement change on the LCD.

Design

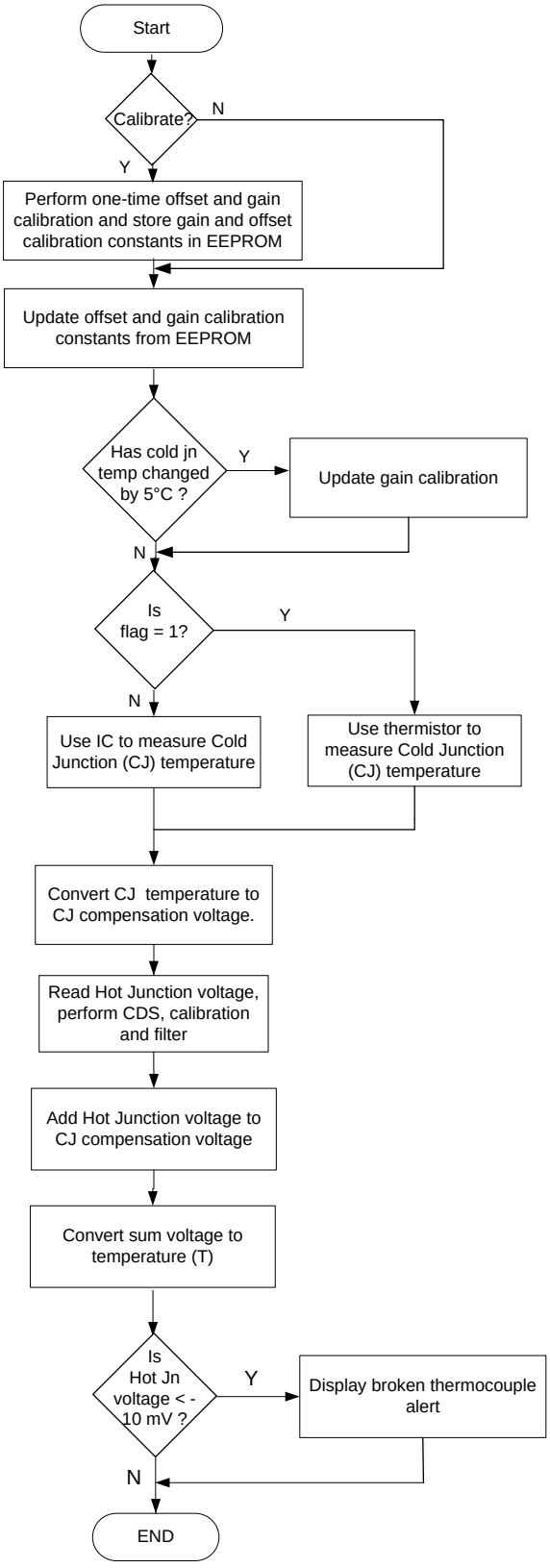
This document describes only how the calibration works. For detailed explanation of how the project works, see CE219905.

To meet the performance of high-end temperature controllers a one-time gain and offset calibration is required. In some cases, a temperature drift calibration is also required. This codes example performs both calibrations.

Figure 3 provides the firmware flowchart of the project. This has the same flow as that of CE219905 with two additional steps:

1. One-time gain and offset calibration
2. Gain drift calibration

Figure 3. CE219929_PSoC3_5_Thermocouple_Calibration Project Flowchart



Gain drift calibration is performed if the ambient temperature changes by 5 °C. The condition for performing temperature calibration can be changed to periodically performing it based on a timer interrupt, or a combination of temperature or time, or always, or whatever is desired.

Gain and Offset Calibration

The gain and offset calibration first asks for 0 V to be applied to the thermocouple input. The best way to do this is to short the two inputs of the thermocouple connector together. Because the negative input of the thermocouple is biased at 15 mV, this can be done with the assurance that the value will not float.

Next, the calibration asks for a full-scale input. Because the ADC is configured for ±64 mV, supply an accurate calibration voltage of less than 64 mV. This calibration voltage should be applied across the thermocouple inputs. Remember that the negative input of the thermocouple is tied to 15 mV.

You also need to change the define `FULL_SCALE` to match the value of your calibration reference. This is defined in *main.h*. If, for example, your calibration voltage is 63.209 mV, then change the define to the following:

```
/* Calibration input in microvolts */
#define FULL_SCALE            63209
```

After you have completed the zero and full-scale calibration, the system asks if you want to do Gain Drift Calibration.

Gain Drift Calibration

The total ADC gain drift (ADC reference + modulator) is characterized at 50 ppm/°C when using an internal reference. The internal reference drift is 20 ppm/°C. You can calibrate out the gain drift using the following procedure:

1. Using a 1.024-V external reference (LM4140), generate a ~50-mV voltage using a low temperature coefficient resistor divider (10 ppm/°C).
2. Measure the generated 50 mV (Vr) immediately after the gain and offset calibration.
3. Store the value of Vr in EEPROM.
4. Whenever the temperature increases by 5 °C, measure the 50-mV reference (Vt) and compute Vr/Vt. The cold junction sensor can be used to measure the internal board temperature. (The condition in Step 4 can be anything such as whenever the temperature increases by 1 °C or at periodic duration based on a timer interval, or always)
5. Multiply all subsequent ADC readings with Vr/Vt.

Calculations

Use 1-kΩ and 20-kΩ resistances to generate the reference voltage of ~50 mV.

Vr = 1.024 * R1/(R1+R2)

= 48.7 mV

For the worst-case tempco error, consider a temperature deviation of ±60 °C from the room temperature.

For a 10 ppm/°C (0.001%) resistor, the worst-case error at 60 °C deviation from the room temperature is 0.06%.

The worst-case voltage reference error happens when the tempco is 10 ppm/°C for R1 and –10 ppm/°C for R2.

At 60 °C, the worst-case reference drift = 0.11%

With 10 ppm/°C resistors, you can achieve a 0.11% gain drift with temperature.

With 5 ppm/°C resistors, you can achieve a 0.057% gain drift with temperature.

Filtering the Thermocouple Output

The firmware IIR filter used has an attenuation factor of 64. From Table 1 in AN2099, you can see that the temperature settling time (0.1%) is 441 cycles. For this code example, the cycle time is about 50 ms, resulting in a temperature settling time of 22 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 22 s.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 1. 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
Emulated EEPROM	Flash

Parameter Settings

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

Table 2. ADC IC Temp Config

Parameter	Change
Resolution	16 → 20
Conversion rate (SPS)	10000 → 180
Buffer Mode	“Rail to Rail” → “Level Shift”

Table 3. ADC Thermocouple Config

Parameter	Change
Conversion Mode	Continuous → Single Sample
Resolution	16 → 20
Conversion rate (SPS)	10000 → 59
Input Range	+/-1.024V → ± 0.064V
Buffer Mode	“Rail to Rail” → “Level Shift”

Table 4. ADC Thermistor Config

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

Figure 4. OpAmp Settings

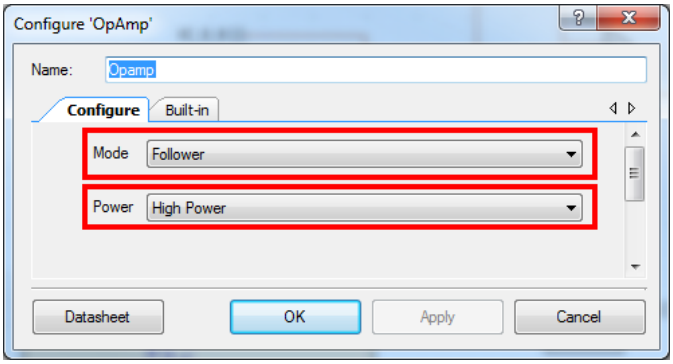
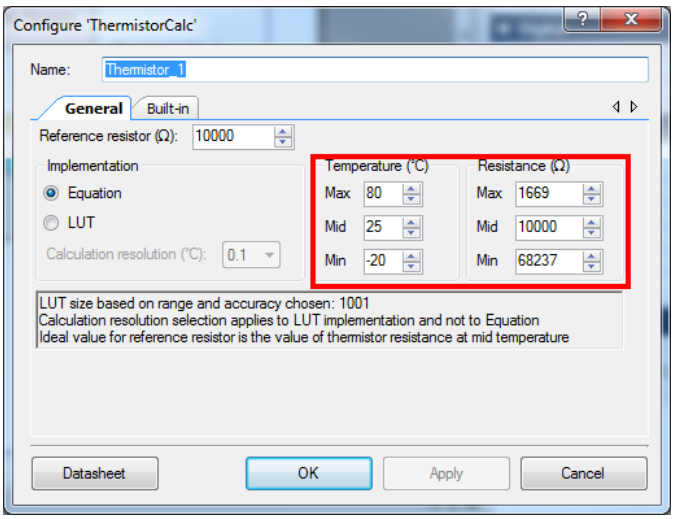


Figure 5. ThermistorCalc Settings



Design-Wide Resources

Pin locations for PSoC 3 and PSoC 5LP devices.

Table 5. Pin Locations for PSoC 3 and PSoC 5LP Devices

Pin Name	PSoC 3/ PSoC 5LP 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]
Calib_In_0_7	P0[7]

Related Documents

Table 6 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Table 6. Related Documents

Code Example		
CE219905	Thermocouple Measurement	Explains how to measure a thermocouple
Application Notes		
AN75511	Temperature Measurement with Thermocouples	Describes how PSoC 3 and PSoC 5LP devices can be used to measure a thermocouple.
AN66444	Correlated Double Sampling to Reduce Offset, Drift, and Low Frequency Noise	Describes how to implement correlated double sampling on a PSoC device.
AN2099	Single Pole Infinite Impulse Response (IIR) Filters	Describes how to implement a software IIR filter.
PSoC Creator Component Datasheets		
Thermocouple Component	Thermocouple Calculator	
Thermistor Component	Thermistor Calculator	
Delta Sigma ADC	Precision ADC component	
Device Documentation		
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals	
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals	
Development Kit (DVK) Documentation		
PSoC 3 and PSoC 5LP Kits		
CY8CKIT-025		

Document History

Document Title: CE219929 - PSoC 3 and PSoC 5LP Thermocouple Calibration
Document Number: 002-19929

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5791122	TDU	03/21/2018	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	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

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

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. Other names and brands may be claimed as property of their respective owners.