CE210434 – PSoC 3 and PSoC 5LP RTD Calibration

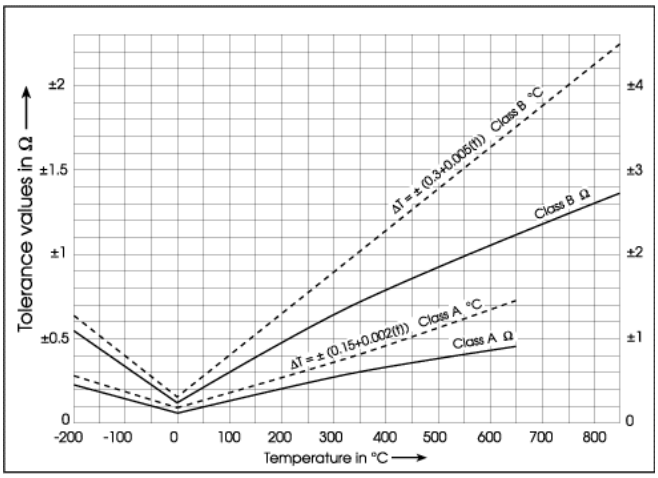
Objective

This example demonstrates how to use PSoC® 3 or PSoC 5LP to calibrate a resistance temperature device (RTD).

Overview

AN70698 describes how the interchangeability error of an RTD can cause several degrees of temperature error when measuring RTD temperature. Figure 1 shows the error versus temperature for two types of RTDs.

Figure 1. Temperature Error due to RTD Interchangeability Error



Note that the units on the right side of the figure are in degrees celsius (°C).

This error is unacceptable in systems that require high accuracy. To overcome this error, calibration is required. This code example demonstrates how the RTD error can be calibrated out. For a basic example on how to measure RTD temperature see CE210383

Requirements

Tool: PSoC Creator™ 4.2 or newer

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

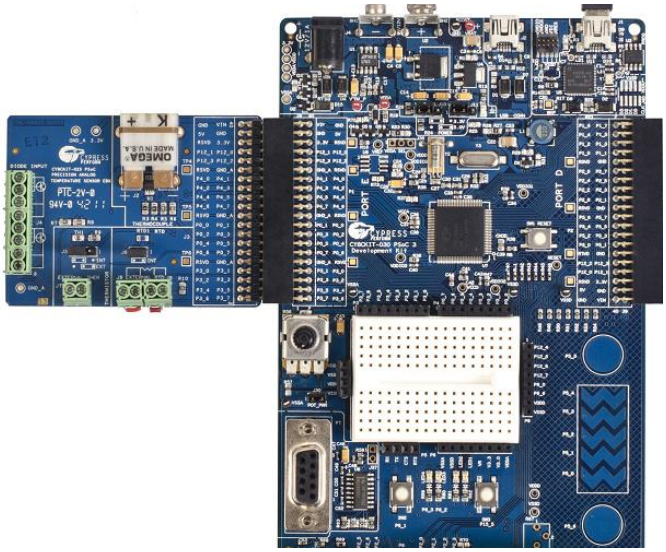
Associated Parts: All PSoC 3 and PSoC 5LP parts

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

Hardware Setup

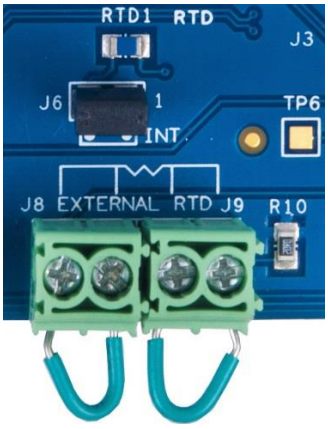
1. Plug CY8CKIT-025 into PORT E of either CY8CKIT-030 or CY8CKIT-050, as Figure 2 shows.

Figure 2. CY8CKIT-025 Plugged In to CY8CKIT-050



2. On KIT-025, ensure that there are wires shorting the terminals of J8 and J9, as Figure 3 shows.

Figure 3. Shorting J8 and J9 on KIT-025



3. (Optional) Plug an external RTD into J8 and J9, remove the J6 jumper and wires in J8 and J9, and CY8CKIT-025.
4. Connect an LCD to the LCD Port on CY8CKIT-030 or CY8CKIT-050.
5. Connect P3[7] to TX on the P5 connector, and P3[6] to RX on the P5 connector on CY8CKIT-030 or 050 to connect the UART to the DB9 cable.

   Connect a DB9 cable from CY8CKIT-030 or CY8CKIT-050 to a PC.

Software Setup

This example firmware supports a terminal emulator interface. A UART interface outputs the temperature to a terminal program on a PC.

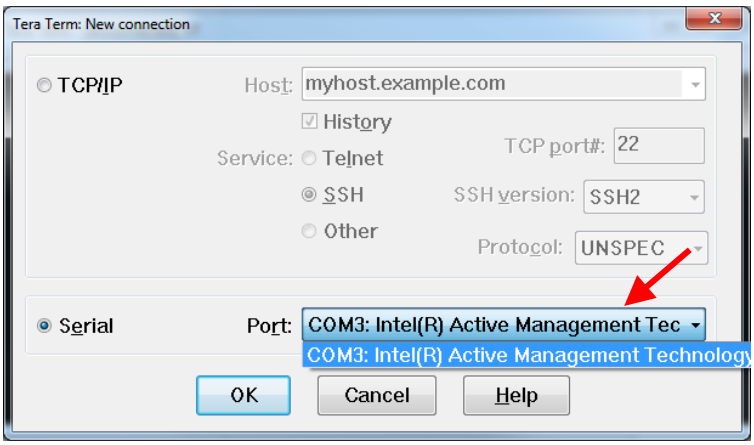
Serial Terminal

This document demonstrates using TeraTerm but any terminal emulator software may be used that is configurable to the standard UART settings shown in Figure 5. TeraTerm is open-source and downloadable directly from the author's website: https://en.osdn.jp/projects/ttssh2/.

1. Create a new connection.

   Launch TeraTerm and select **File** > **New connection**. Select **Serial** as the connection type and choose the COM port where you connected the DB9 cable.

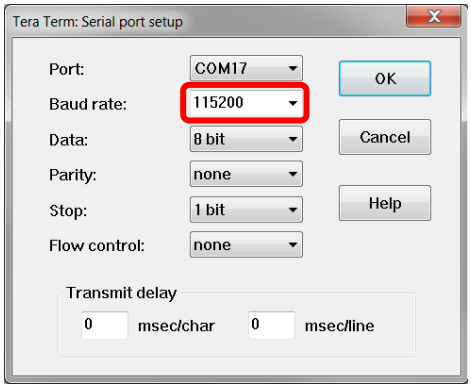
Figure 4. New Connection Creation



2. Setup serial port parameters.

   Open TeraTerm Serial port setup dialog (**Setup** > **Serial port…**). Only the **Baud rate:** should require changing to 115200, but it is good to also confirm the other settings are as shown in Figure 5.

Figure 5. Terminal Emulator Setup Parameters



Operation

1. Load the workspace into PSoC Creator by opening `<Install_Directory>\CE210434\CE210434.cywrk`.
2. Build the code example by navigating to **Build** > **Build CE210434_PSoC3_5_RTD_Calibraiton in PSoC Creator.**
3. Connect the device/board to a programmer connected to a PC. On-board KitProgs are already connected to the programming pins of the on-board device.
4. Program the example to the device by navigating to **Debug** > **Program**.
5. Open the terminal program by following the instructions in the Serial Terminal section.
6. Power the device.
7. Follow the on-screen instructions to calibrate the RTD.

Design

CE210383 describes and demonstrates how a basic RTD temperature measurement is performed on PSoC. This code example uses the same basic setup but adds calibration.

Calibration is done on startup. The PSoC sends a message over UART to a terminal emulator program on a PC. First, it prompts you to confirm if you want to calibrate. To calibrate, you must bring the RTD to a known temperature, and enter this temperature into the terminal program. The PSoC then calculates the RTD resistance for this temperature.

The PSoC then measures the RTD and calculates a ratio between the measured RTD and the value that you entered. This ratio is then used to scale all future RTD results.

Note that this project works only for PSoC 3 and PSoC 5LP. In PSoC 4, the primary cause of error is integral nonlinearity (INL), which cannot be calibrated out; thus calibration is not applied.

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 for PSoC 3 or PSoC 5LP

Component	Name	Hardware Resources	Non-default Parameter Settings
ADC_DelSig	ADC	1 DelSig ADC Block	Config1 Resolution(bits): 20 Conversion rate (SPS): 61 Input Rage: +/- 0.512 V (-Input +/- Vref/2) Buffer Mode: Level Shift Common Number of Configurations: 1
Current DAC (8-bit)	IDAC	1 ViDAC	Polarity: Source Speed: High Speed Range: 0 – 2.04 mA Value: 1000 µA
RTD_Calculator	RTD	N/A	Temperature Min: -200 Temperature Max: -850 Calculation Error Budget: 0.01
UART	UART	~2 UDBs	Configure: Mode: TX + RX Bits per second: 115200
Character LCD	LCD	7 Pins	N/A

Design-Wide Resources

Pin locations for PSoC 3 and PSoC 5LP MCUs.

Table 2. Pin locations for PSoC 3, PSoC 5LP MCUs

Pin Name	PSoC 3/ PSoC 5LP MCU Location
LCD	P2[6:0]
RefRes_Current	P3[4]
RefRes_Positive	P4[4]
RefRes_Negative	P4[5]
RTD_Current	P3[1]
RTD_Positive	P4[0]
RTD_Negative	P4[1]
TX	P3[7]
RX	P3[6]

Related Documents

Table 3 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 3. Related Documents

Application Notes		
AN70698	PSoC 3/ PSoC 4/ PSoC 5LP temperature measurement with an RTD	Theory behind RTD temperature measurement.
Code Examples		
CE210435	PSoC 3 and PSoC 5LP Broken RTD Reconfiguration	Demonstrates how to detect a broken wire on an RTD connection and reconfigure the Analog front end to keep measuring the RTD
CE210383	PSoC 3, PSoC 4, and PSoC 5LP Temperature Sensing with an RTD	Demonstrates basic RTD temperature measurement
PSoC Creator Component Datasheets		
RTD Calculator	Component datasheet for RTD Calculator Component	
ADC DelSig	Component datasheet for ADC DelSig Component.	
Emulated EEPROM	Component datasheet for Emulated EEPROM Component	

Document History

Document Title: CE210434 – PSoC 3 and PSoC 5LP RTD Calibration
Document Number: 002-10434

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5077566	TDU	01/08/2016	New spec
*A	6071629	TDU	02/14/2018	Updated code example to new template

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, 2016-2018. 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.