CE210528 – PSoC 3 and PSoC 5LP Thermistor Calibration

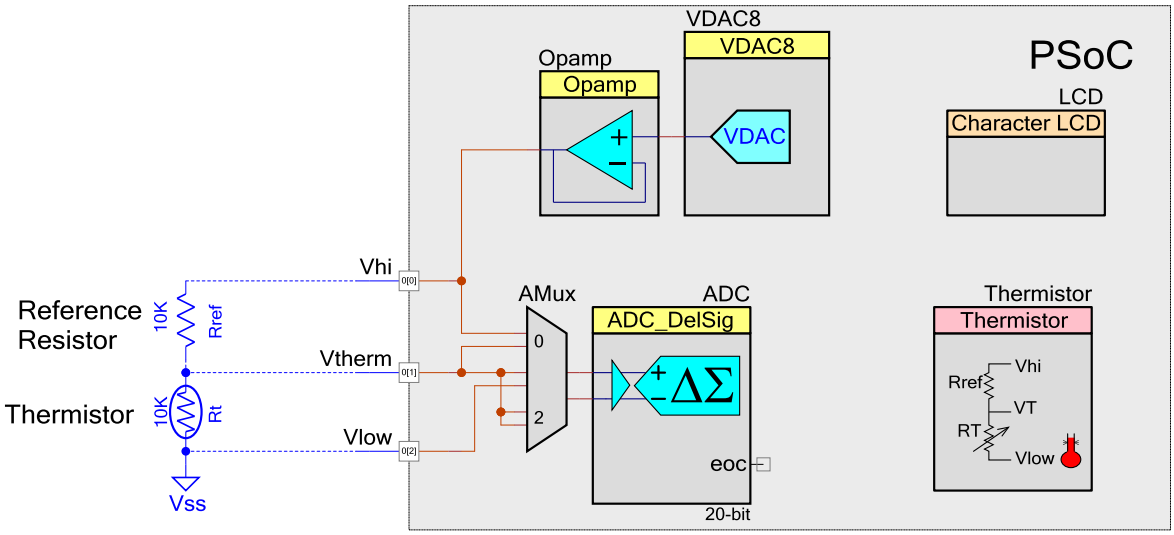
Objective

This example shows how to use a PSoC® 3 or PSoC 5LP to calibrate a thermistor for better temperature measurement accuracy.

Overview

Figure 1 shows a typical circuit for thermistor temperature measurement.

Figure 1. Thermistor Temperature Measurement



The resistance of the thermistor is obtained by the following equation:

$$Rt = Rref * \frac{Vhi - Vtherm}{Vtherm - Vlow}$$

Most applications hard-code Rref to its nominal value, such as 10 kΩ. Any error in Rref results in an error in this calculation, which in turn causes errors in temperature measurement. AN66477 shows that this error can be as high as 0.5 °C. The best way to overcome this error is to determine the actual value of Rref and use that instead of a hard-coded value. This code example demonstrates how that can be done.

Also, the thermistor itself can have errors. This code example demonstrates how to remove any offset errors associated with the thermistor.

Requirements

Tool: PSoC Creator™ 4.2 or newer

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

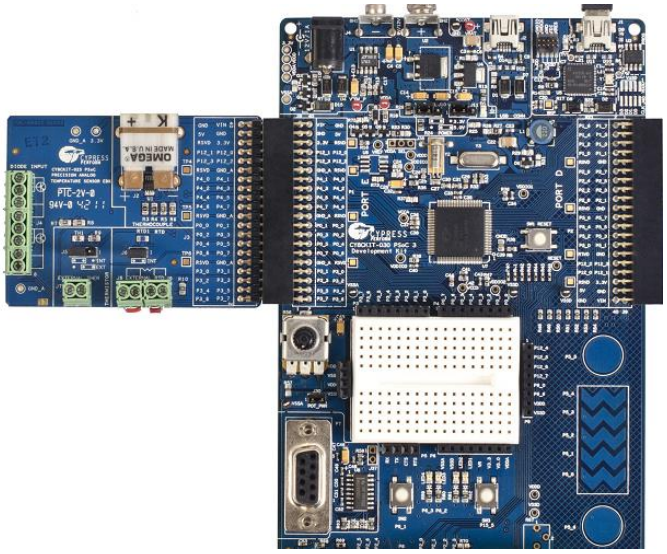
Associated Parts: All PSoC 3 and PSoC 5LP parts

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 into CY8CKIT-050



2. (Optional) Plug an external thermistor into J7 and set J5 to INT on CY8CKIT-025.
3. (Optional) Connect an LCD to the LCD port on CY8CKIT-030 or CY8CKIT-050.
4. Connect P3[7] to TX on the P5 connector and P3[6] to RX on the P5 connector on CY8CKIT-030 or 050. This connects the UART to the DB9 connector.
5. 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 data to a terminal program on a PC.

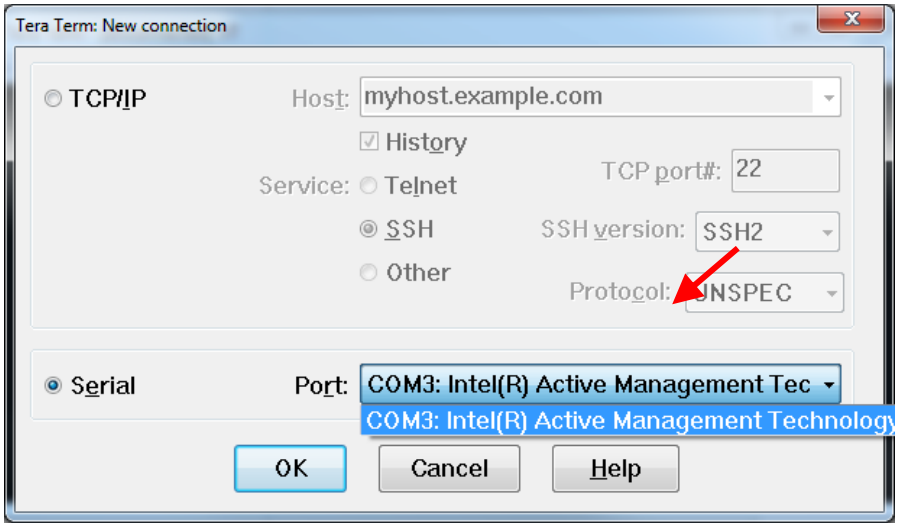
Serial Terminal

The documentation for setting up a terminal emulator for this example uses TeraTerm, but any terminal emulator software that you can configure with the standard UART settings shown in Figure 4 may be used. TeraTerm is open source and downloadable directly from the author's website, https://en.osdn.jp/projects/ttssh2/.

1. Create 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, as Figure 3 shows.

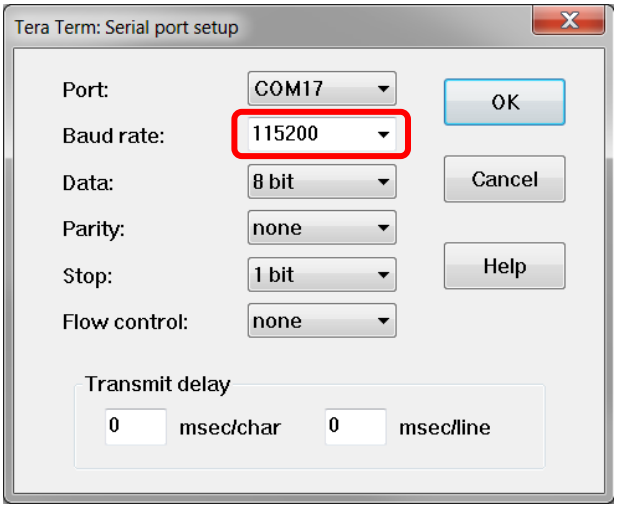
Figure 3. New Connection Creation



2. Set up serial port parameters.

   Open TeraTerm Serial port setup dialog (**Setup** > **Serial port…**). Only the **Baud rate:** should require changing to 115200, but you should also confirm the other settings are as Figure 4 shows.

Figure 4. Terminal Emulator Setup Parameters



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 Creator project and using the Thermistor Calculator component right click on your project click on **Build Settings…** expand the complier by pressing the + button. The name of the compiler will depend on which compiler is used in the design; for Creator 4.2 it is **ARM GCC 5.4-2016-q2-update.** Click on **Linker**, in **Additional Libraries** add m, 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. Consult help documentation for the IDE.

Operation

1. Load the workspace into PSoC Creator by opening `<Install_Directory>\CE210528\CE210528.cywrk`.
2. Build the code example by navigating to **Build** > **Build *CE210528_Thermistor_Calibration*** 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 project to the device by navigating to **Debug** > **Program**.
5. Open a terminal program by following the instructions in the Serial Terminal section.
6. Power the device.
7. Follow the on-screen instructions to calibrate the thermistor.
   a. To calibrate the reference resistor, you must use an external, known resistance. This means removing the thermistor from the circuit.
   b. To calibrate the thermistor, you must bring the thermistor to a known temperature.
   c. To let the main project use the calibration values, you must store these values in EEPROM.

Design

CE210514 shows how basic thermistor temperature measurement is performed using PSoC devices. This code example uses the same basic setup but adds calibration.

Calibration is done on startup. The PSoC device 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 replace the thermistor with a known resistor, ideally around the value of the reference resistor, which is typically 10 kΩ. Next, this resistance must be entered into the terminal program. The PSoC device uses this value to calculate the actual value of Rref. The calculated Rref is stored in EEPROM and used in all subsequent thermistor resistance calculations.

Next, the PSoC device prompts you if you want to calibrate out the temperature offset of the thermistor. To do this, you must do the following:

1. Put the thermistor back into the system.
2. Measure the exact temperature of the thermistor with an accurate temperate sensor.
3. Enter the measured temperature value into the terminal emulator program.

The PSoC device then measures the thermistor and calculates the offset between the entered temperature and the measured temperature. This offset is stored in EEPROM and subtracted from all subsequent temperature measurements.

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

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources and parameter settings 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 Conversion Mode: 0 - Single Sample Resolution(bits): 20 Conversion rate (SPS): 46 Input Rage: +/- 2.048 V (-Input +/- 2 *Vref) Buffer Mode: Level Shift Common Number of Configurations: 1
Voltage DAC (8-bit)	VDAC8	1 ViDAC	Speed: Slow Speed Range: 0 – 4.08 V Value: 1600 mV
Thermistor_Calculator	Thermistor	None	Reference Resistor 10000 Implementation: Equation Temperature Max: 125 °C – 531 Ω Temperature Mid : 25 °C – 10000 Ω Temperature Mid : -40 °C – 195652 Ω
UART	UART	~2 UDBs	Configure: Mode: TX + RX Bits per second: 115200
Character LCD	LCD	7 pins	N/A
EEPROM	EEPROM	EEPROM	N/A

Design-Wide Resources

Table 2 lists pin physical pin assignments for PSoC 3 and PSoC 5LP.

Table 2. Pin Locations for PSoC 3 and PSoC 5LP Devices

Pin Name	PSoC 3/ PSoC 5LP Location
LCD	P2[6:0]
Vhi	P0[0]
Vtherm	P0[1]
Vlow	P0[2]
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		
AN66477	PSoC 3, PSoC 4, and PSoC 5LP Temperature Measurement with a Thermistor	Theory behind thermistor temperature measurement.
Code Examples		
CE210514	PSoC 3, PSoC 4, and PSoC 5LP Thermistor Temperature Measurement	Demonstrates how to measure thermistor temperature.
CE211321	Interfacing the PSoC Analog Coprocessor with a Temperature Sensor	Demonstrates how to implement an analog front end (AFE) for a thermistor using the PSoC Analog Coprocessor.
PSoC Creator Component Datasheets		
Thermistor Calculator	Component Datasheet for Thermistor Calculator Component	
ADC DelSig	Component Datasheet for ADC DelSig Component.	
UART	Component datasheet for UDB based UART Component	
LCD	Component datasheet for LCD Component	
VDAC8	Component datasheet for VDAC8 Component	
Opamp	Component datasheet for Opamp Component	
EEPROM	Component datasheet for EEPROM Component	
Device Documentation		
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals	
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals	
Development Kit (DVK) Documentation		
CY8CKIT-025 PSoC Precision Analog Temperature Sensor Expansion Board		
PSoC 3 and PSoC 5LP Kits		

Document History

Document Title: CE210528 - PSoC 3 and PSoC 5LP Thermistor Calibration
Document Number: 002-10528

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5077566	TDU	01/08/2016	New spec
*A	6076854	TDU	02/18/2018	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.