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.

AN78920

PSoC® 1 Temperature Measurement Using Diode

Author: Dineshbabu M/Prem Sai V
Associated Project: Yes
Associated Part Family: CY8C28xxx Only
Software Version: PSoC® Designer™ 5.4
Related Application Notes: AN60590

AN78920 explains the diode-based temperature measurement using PSoC® 1 – CY8C28xxx family. The temperature is measured based on the principle of a diode's forward bias current dependence on temperature.

Contents

1 Introduction 1
2 PSoC Resources 2
2.1 PSoC Designer 2
2.2 Code Examples 3
2.3 Technical Support 4
3 The Diode Equation 5
4 Measuring the Temperature 5
5 Measuring Diode Temperature Using CY8C28xxx 6
6 Block Diagram 7
6.1 Functionality 7
6.2 Interconnect View 8
6.3 Flow Chart 9
7 Hardware Connection 10
7.1 Transistor Selection 11
8 Selection of the IDAC Calibration Resistor 11
9 Error Budget Analysis 12
9.1 Ideality Factor of the Transistor Diode 12
9.2 IDAC Current Ratio 12
9.3 ADC Error 13
9.4 Summary of Error Sources 13
10 Test Results 13
11 CY8CKIT-036 for Diode Temperature Measurement 15
12 Summary 15
Worldwide Sales and Design Support 17

1 Introduction

PSoC 1 – CY8C28xxx family has on-chip 8-bit IDAC, and a 14-bit Delta Sigma ADC, which enable accurate and high-resolution temperature measurements using an external diode-connected transistor. The example projects attached with this application note work with CY8CKIT-036 – PSoC Thermal management EBK.

There are various sensors available for measuring temperature such as Thermistor, Thermocouple, resistance temperature detectors (RTD). Choosing a sensor or method to employ for measuring the temperature depends on factors such as the accuracy requirement, the temperature range to be measured, and the cost of the temperature sensor. The diode based temperature measurement is an easy, accurate, and also relatively low-cost method for measuring the temperature. With on-chip current DACs and a 14-bit Delta Sigma ADC, PSoC 1 – CY8C28xxx family enables simple and accurate temperature measurement using just an external diode and a calibration resistor. With 14-bit Delta Sigma ADC, it is possible to achieve resolution of 1 °C, and an accuracy of ±2 °C in temperature measurement.

Diode based temperature measurement is typically used in one of the following two ways:

1. Most CPU processors and some Application Specific Integrated Circuits (ASIC's) provide access to thermal diode in their architecture to measure the temperature of the processor core. This temperature measurement is used for thermal management functions such as fan control to cool the processor core. PSoC 1 – CY8C28xxx family can be used to interface with those thermal diodes to measure core temperature, and perform system management functions such as fan speed control.

2. It is also possible to use a general purpose transistor (such as 2N3904) for temperature measurement. The reason to opt for a general purpose transistor in this case would be the extremely cheap transistor cost along with less stringent accuracy requirements (±3 °C). To perform temperature measurement, the transistor must be configured as a diode by shorting the collector and base terminals of the transistor. Note that a general purpose diode does not give as accurate results as a general purpose transistor (configured as a diode) while measuring the temperature. This is due to the fact that the variations due to manufacturing processes are lesser in transistors compared to diodes.

2 PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. In this document, PSoC refers to the PSoC 1 family of devices. To learn more about PSoC 1, refer to the application note AN75320 - *Getting Started with PSoC 1*.

The following is an abbreviated list for PSoC 1:

- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Designer includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 1 device family.
- **Application Notes and Code Examples:** Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the internal architecture of the PSoC 1 devices.
- **Development Kits:**
  - CY3215A-DK In-Circuit Emulation Lite Development Kit includes an in-circuit emulator (ICE). While the ICE-Cube is primarily used to debug PSoC 1 devices, it can also program PSoC 1 devices using ISSP.
  - CY3210-PSOCEVAL1 Kit enables you to evaluate and experiment Cypress's PSoC 1 programmable system-on-chip design methodology and architecture.
  - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg1 and MiniProg3 devices provide an interface for flash programming.

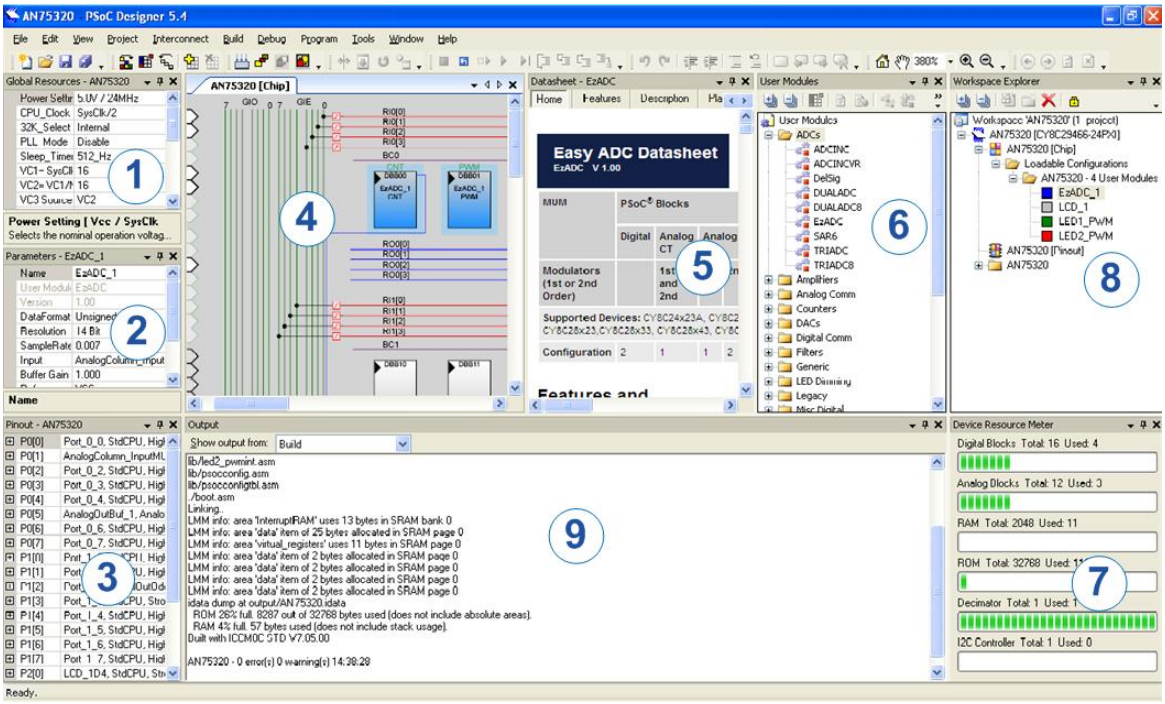
2.1 PSoC Designer

PSoC Designer is a free Windows-based Integrated Design Environment (IDE). Develop your applications using a library of pre-characterized analog and digital peripherals in a drag-and-drop design environment. Then, customize your design leveraging the dynamically generated API libraries of code. Figure 1 shows PSoC Designer windows. **Note:** This is not the default view.

1. **Global Resources –** all device hardware settings.
2. **Parameters –** the parameters of the currently selected User Modules.
3. **Pinout –** information related to device pins.
4. **Chip-Level Editor –** a diagram of the resources available on the selected chip.
5. **Datasheet –** the datasheet for the currently selected UM
6. **User Modules –** all available User Modules for the selected device.
7. **Device Resource Meter –** device resource usage for the current project configuration.
8. **Workspace –** a tree level diagram of files associated with the project.
9. **Output –** output from project build and debug operations.

Note: For detailed information on PSoC Designer, go to **PSoC® Designer > Help > Documentation > Designer Specific Documents > IDE User Guide**.

Figure 1. PSoC Designer Layout



2.2 Code Examples

The following webpage lists the PSoC Designer based Code Examples. These Code Examples can speed up your design process by starting you off with a complete design, instead of a blank page and also show how PSoC Designer User modules can be used for various applications.

http://www.cypress.com/go/PSoC1Code Examples

To access the Code Examples integrated with PSoC Designer, follow the path **Start Page > Design Catalog > Launch Example Browser** as shown in Figure 2.

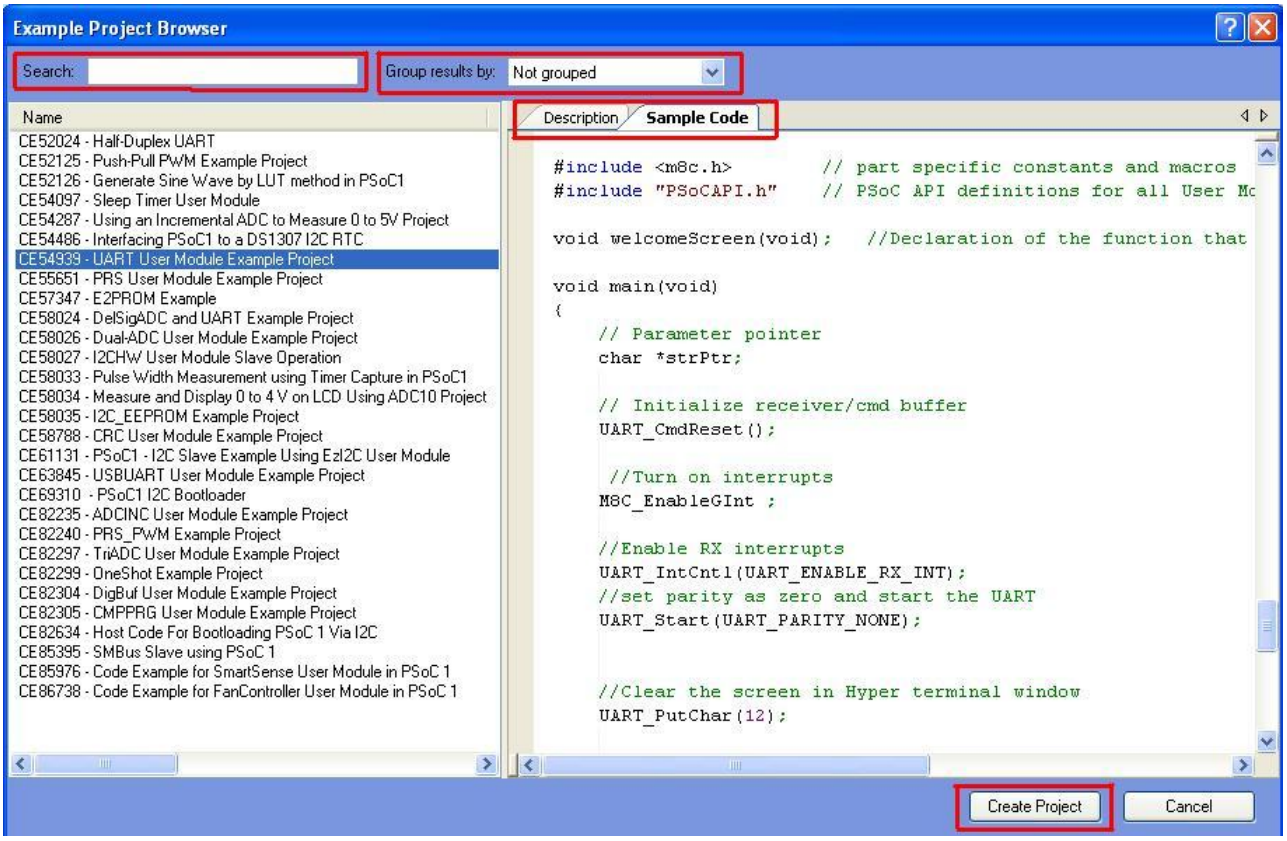
Figure 2. Code Examples in PSoC Designer



In the Example Projects Browser shown in Figure 3, you have the following options.

- Keyword search to filter the projects.
- Listing the projects based on Category.
- Review the datasheet for the selection (on the Description tab).
- Review the code example for the selection. You can copy and paste code from this window to your project, which can help speed up code development, or
- Create a new project (and a new workspace if needed) based on the selection. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 3. Code Example Projects, with Sample Codes



2.3 Technical Support

If you have any questions, our technical support team is happy to assist you. You can create a support request on the Cypress Technical Support page.

You can also use the following support resources if you need quick assistance.

- Self-help
- Local Sales Office Locations

3 The Diode Equation

The current I through a forward biased diode is given by the equation,

$$I = I_s \exp(V / \eta V_T) \qquad I = I_s e^{\frac{V}{\eta V_T}} \qquad \text{Equation 1}$$

Where,

V – The diode forward voltage drop

I_s – The reverse saturation current

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

V_T – The thermal voltage given by,

$$V_T = KT / q \qquad \text{Equation 2}$$

Where,

k – Boltzmann's constant (1.38 x 10^{-23} J/K) ($1.38 \times 10^{-23} joules/kelvin$)

T– The absolute temperature in Kelvin

q– The magnitude of electronic charge (1.602 x 10^{-19} C)

The temperature-dependent factors in Equation 1 are I_S and V_T. The reverse saturation current I_S typically doubles for every 5 °C rise in temperature. I_S depends on the physical properties of the diode. V_T is directly proportional to temperature.

4 Measuring the Temperature

The technique for measuring the temperature is based on applying two different known currents to flow through the diode, and measuring the diode voltage in each case.

For two different currents I_1 and I_2, such that $I_2 = NI_1$

$$I_1 = I_s \exp(V_1 / \eta V_T)$$

$$I_2 = I_s \exp(V_2 / \eta V_T)$$

$$\frac{I_2}{I_1} = N = \exp((V_2 - V_1) / \eta V_T) \; I_2 = NI_1 \qquad \text{Equation 3}$$

Taking natural logarithm on both sides,

$$\ln(N) = (V_2 - V_1) / \eta V_T \qquad \text{Equation 4}$$

Using Equation 2, the temperature (T) in Kelvin is given by,

$$T(in Kelvin) = (V_2 - V_1) * \frac{q}{\ln(N) * K * \eta}$$

$$T(in Kelvin) = c * \Delta V$$

$$T(in \;\; ^0C) = (c * \Delta V) - 273 \qquad \text{Equation 5}$$

Where,

$\Delta V = (V_2 - V_1)$ – The difference in diode forward voltage drop for two different currents.

'c' – A constant given by,

$$c = \frac{(q)}{k*\eta*\ln(N)} \quad \text{Equation 6}$$

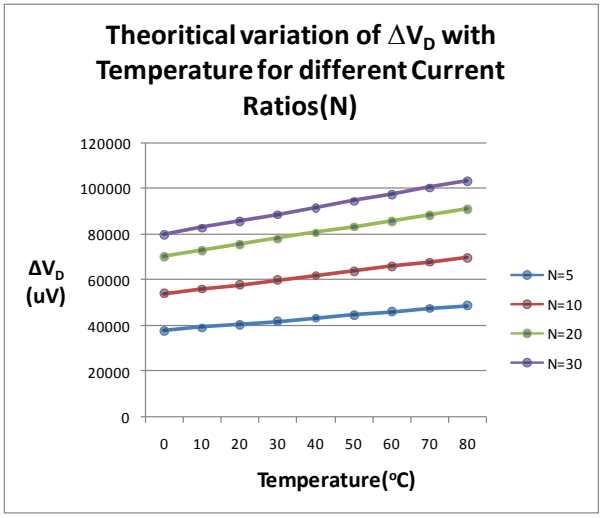
For $'\eta' = 1, 'N' = 20$ we get '**c**' ≈ 3875 Kelvin/Volt.

$$T(in^{\circ}C) = (\Delta V * 3875) - 273 \quad \text{Equation 7}$$

A 1 °C or 1-K difference in temperature translates into a change in ΔV by a factor (1/c), which for the above considered values translates to around 258 µV.

For increasing values of current ratio 'N', there would be greater change in ΔV with respect to temperature, as seen the Figure 4.

Figure 4. Theoretical Variation of ΔV_D with Temperature for Different Current Ratios (N)

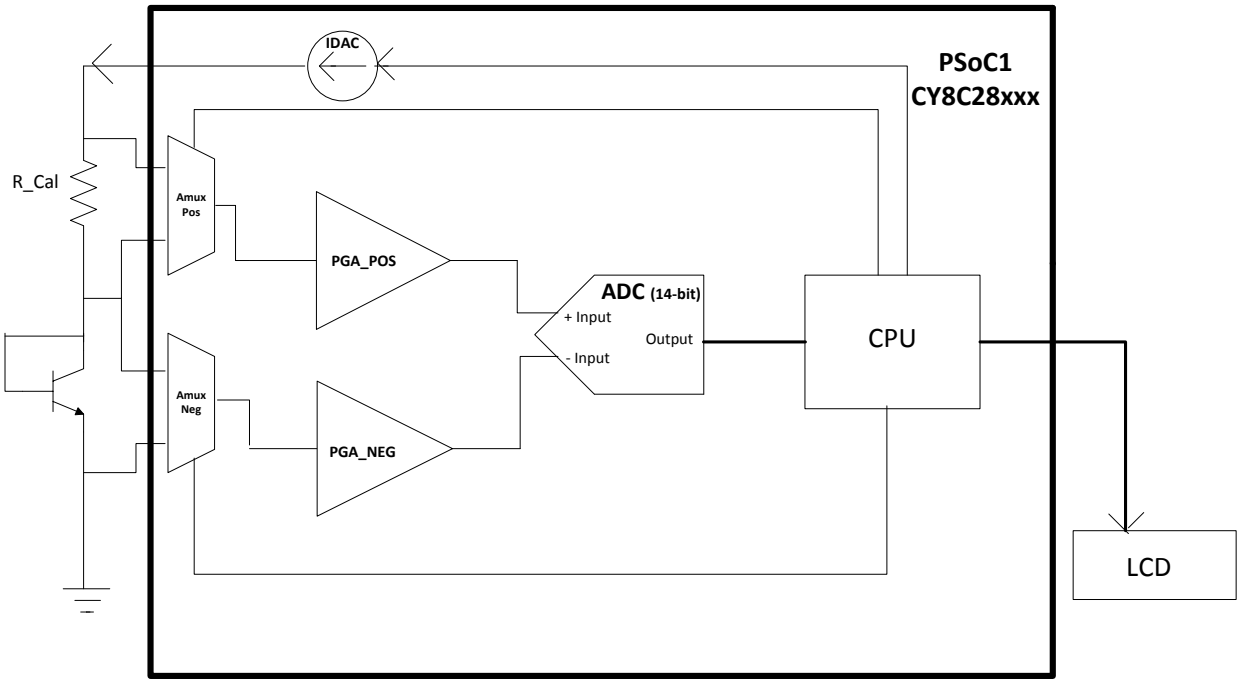


5 Measuring Diode Temperature Using CY8C28xxx

CY8C28xxx family has powerful analog architecture that enables the accurate measurement of Diode temperature. The implementation in CY8C28xxx is based on Equation 5. Figure 5 shows the block diagram for this implementation showing the external diode connected transistor and also the external calibration resistor. This calibration resistor is used to accurately calculate the IDAC current ratio 'N' used in Equation 6.

6 Block Diagram

Figure 5. Block Diagram

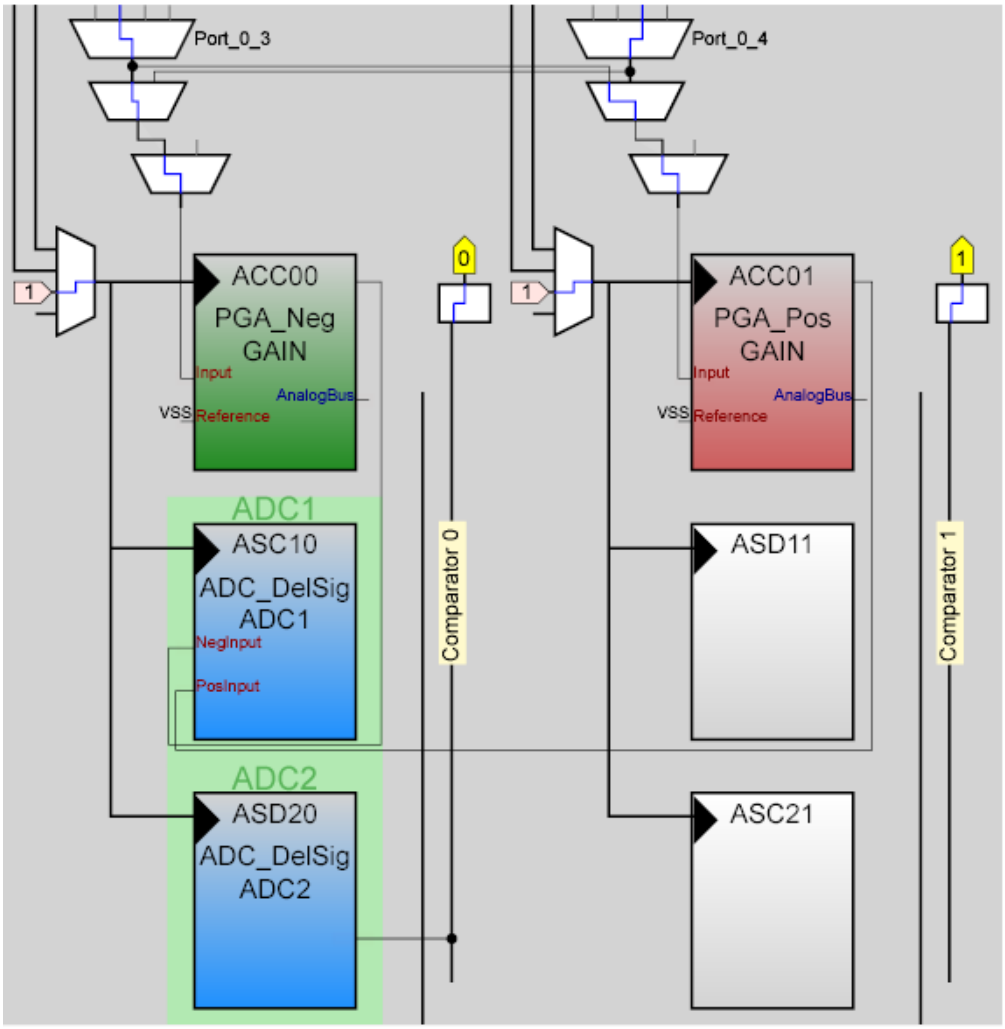


6.1 Functionality

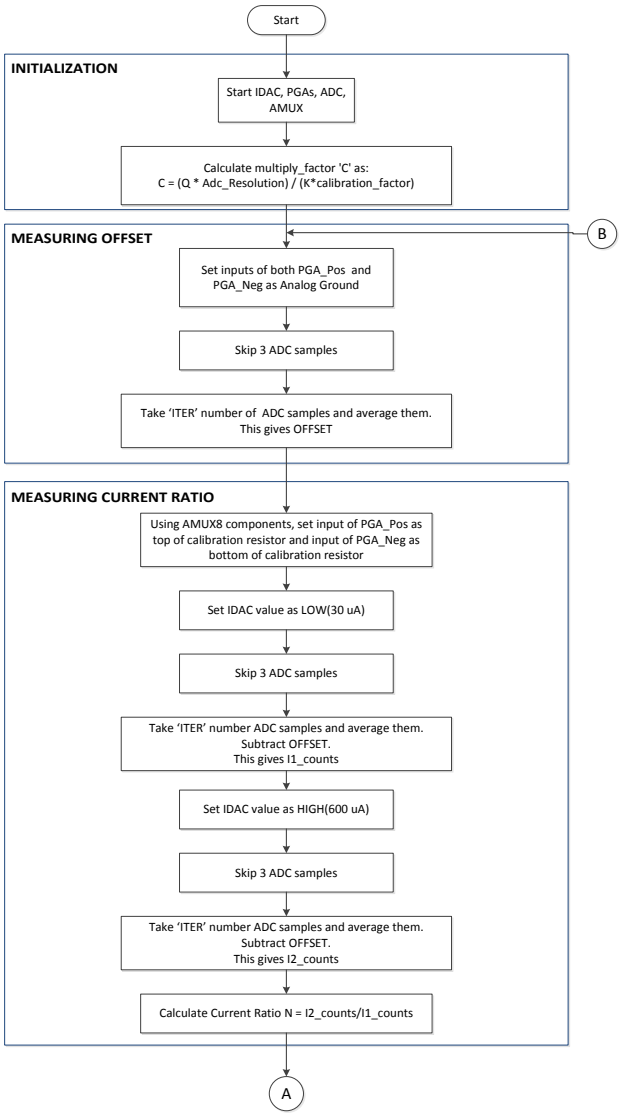
- Current DAC (IDAC) forces two known currents I1, I2 through the transistor diode.
- The corresponding base-emitter voltages V1, V2 are measured using Delta Sigma ADC.
- Equation 5 is then used to calculate the temperature in firmware. To ensure that the IDAC current ratio 'N' (N = I2/I1) in Equation 5 is computed accurately, a calibration resistor is connected at the output of IDAC in series with the transistor diode. The ratio of voltages across the calibration resistor gives the IDAC current ratio. This calibration removes the error due to IDAC offset, non-linearity in temperature measurement. By using 14-bit Delta Sigma ADC, the current ratio can be calculated accurately.
- The AMUX8 component in PSoC Designer is used to multiplex the input pins to perform ADC Offset calculation, Current ratio calculation and Temperature calculation.
- The ADC offset calculation is performed by connecting the two ADC inputs to AGND and measuring the ADC output. This offset correction is used while doing IDAC calibration to measure the current ratio accurately.

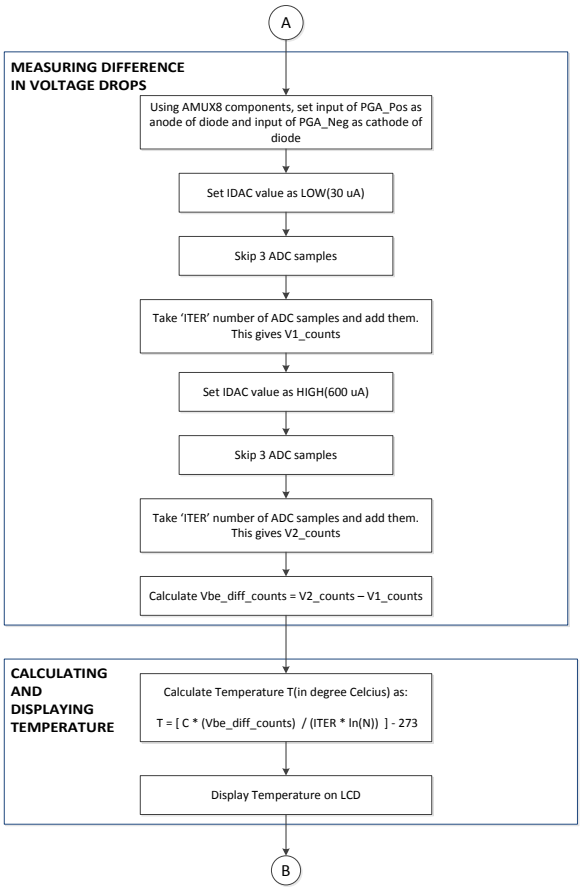
6.2 Interconnect View

Figure 6. PSoC Designer Interconnect View



6.3 Flow Chart





7 Hardware Connection

The following schematic shows the connection between the pins of PSoC (that is placed on CY8CKIT-001) to the transistor present on CY8CKIT-036 EBK. The calibration resistor is an external resistor. The CY8CKIT-025 EBK that has on-board calibration resistor can also be used with this application note.

Figure 7. Connection Between CY8CKIT-001 and CY8CKIT-036 EBK

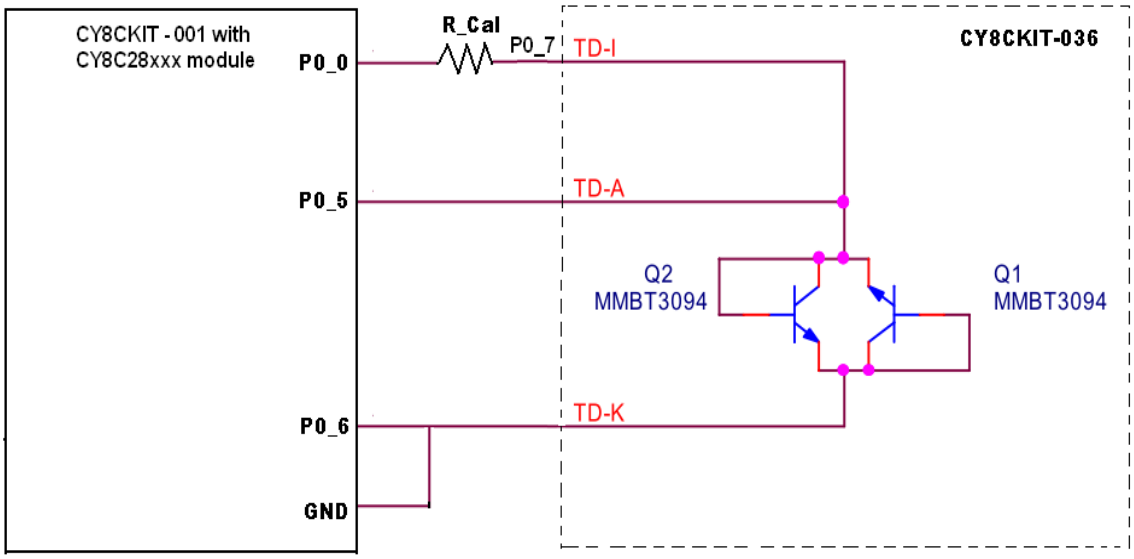
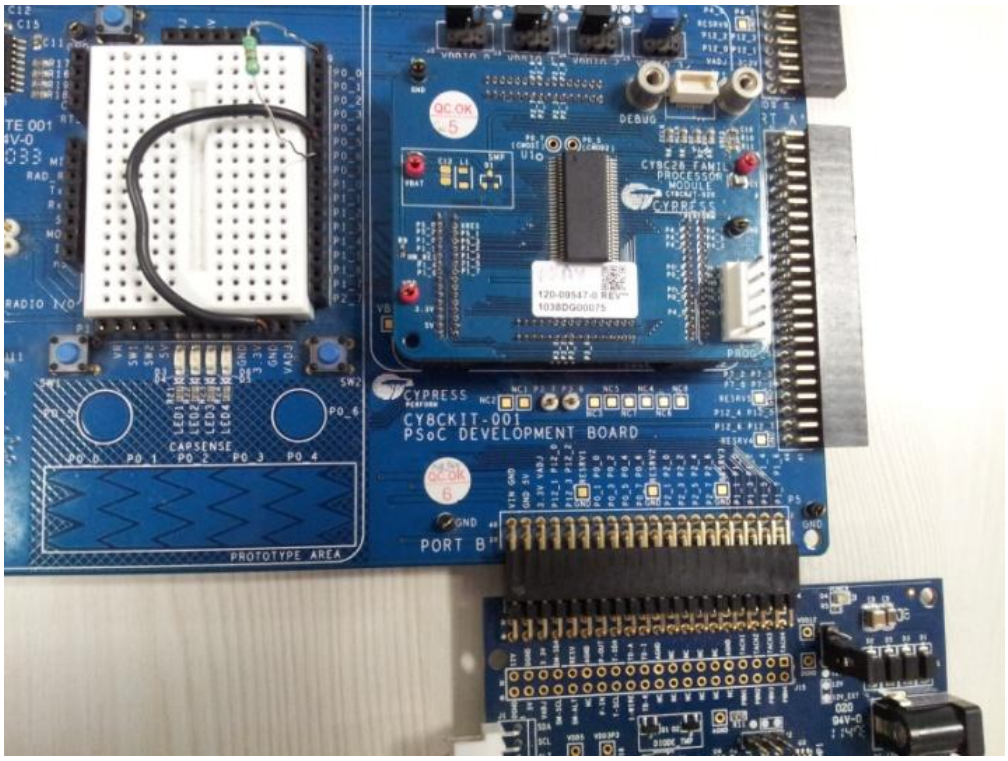


Figure 8. Snapshot of Hardware Setup



7.1 Transistor Selection

The selection of right transistor is very important for accurate temperature measurement. The CY8CKIT-036 uses the MMBT3904 transistor from Fairchild Semiconductors for temperature measurement. The following factor should be considered for choosing the transistor that gives the best results for temperature measurement.

7.1.1 Ideality Factor

Ideality factor (η) of diode is involved in temperature measurement as shown in Equation 5 and Equation 6. The datasheets of the general purpose transistors do not provide ideality factor value in their datasheets. But a correct measurement of the ideality factor is required for accurate temperature measurement. The procedure to calculate the ideality factor is given as follows:

1. Assuming an ideality factor $\eta_{assumed}$ (for example: 1), measure the diode temperature ($T_{measured}$ in Kelvin) using CY8C28xxx.
2. Measure the ambient temperature 'T_{actual}' (in Kelvin) using an accurate temperature measurement source. In lab testing, the MicroTherma2 temperature measurement system was used to find the ambient temperature accurately. Since the self-heating of diode is negligible, both the diode temperature and ambient temperature will be almost the same.

   Using the assumed ideality factor($\eta_{assumed}$), and the two temperature parameters ($T_{measured}$, T_{actual}), the correct ideality factor can be calculated as given by below equation.

$$IdealityFactor, \eta_{correct} = Ideality\ Factor, \eta_{correct} = \frac{T_{measured}}{T_{actual}} * \eta_{assumed}$$

Based on above equation, the ideality factor of the MMBT3904 transistor used in this application note is 1.043619.

The error in temperature measurement due to a wrong calculation of ideality factor is given by,

$$\Delta T = T_{actual} - T_{measured}$$

$$\Delta T = T_{actual}(inKelvin) * (1 - \eta_{correct} / \eta_{assumed}) \qquad \text{Equation 8}$$

T_{actual} is the expected temperature for correct ideality factor $\eta_{correct}$, and $T_{measured}$ is the measured temperature for assumed ideality factor$\eta_assumed$. It can be inferred from Equation 8 that the error due to wrong ideality factor increases with increasing temperature. A 1 percent error in ideality factor $\eta_{assumed} = 1.01 * \eta_{correct}$ would cause a measurement error of 3.7 °C or 3.7 K at temperature of 373 K (100 °C).

8 Selection of the IDAC Calibration Resistor

An external resistor in series with the diode is used for calibrating IDAC so as to calculate the IDAC current ratio accurately. This resistor need not be highly accurate as the absolute value of the resistance does not matter when taking ratio of voltages.

From Equation 5,

Current ratio, $N = I_2/I_1 = (V_2 * R)/(V_1 * R) = V_2/V_1$

The two limiting factors while choosing the resistor are:

1. This minimum value of calibration resistor must be such that the voltage-drop across the calibration resistor when passing the minimum current is greater than the ADC's resolution.
2. The maximum value of the calibration resistance is determined by two independent factors:
   a. The IDAC compliance voltage: The compliance voltage of the IDAC is (V_{DD} – 1), where V_{DD} is supply voltage.

      $$R_A < (V_{DD} - 2)/(I_{max})$$

      A difference of 1 V is also present along with IDAC compliance voltage of (V_{DDA} – 1), which is to account for maximum diode forward voltage drop. I_{max} is the maximum current output of IDAC. For example, V_{DD} =5 V, I_{max} = 600 µA results in maximum resistance of ~5 kΩ.

b. The voltage drop across the calibration resistor when passing the maximum current must be lesser than the ADC's range in the positive side (RefHi-AGND)

$$R_B < (RefHi - AGND) / (I_{max})$$

The maximum value of calibration resistor is the lesser of R_A and R_B, i.e.,

$$R_{max} = Min(R_A, R_B)$$

An optimized resistance value of 450 Ω, which satisfies the above conditions, is used in this application note.

9 Error Budget Analysis

This section discusses the different factors that affect the accuracy of diode temperature measurement. This includes the error caused by the PSoC resources (PGA, ADC, and IDAC), and the external transistor diode. The analysis in this section is based on the following equation:

$$T\ (in\ Kelvin) = (V_2 - V_1) * \frac{q}{\ln(N) * k * \eta}$$ Equation 9

9.1 Ideality Factor of the Transistor Diode

The following equation gives the error due to the ideality factor:

$$\Delta T = T_{ideal} - T_{measured}$$

$$\Delta T = T_{ideal} * (1 - \eta_{correct} / \eta_{assumed})$$ Equation 10

T_{ideal} is the expected temperature in Kelvin for the correct ideality factor $\eta_{correct}$, and $T_{measured}$ is the measured temperature in Kelvin for the assumed ideality factor $\eta_{assumed}$. It can be inferred from Equation 10 that the error due to a wrong ideality factor increases with the increasing temperature. A 0.1 percent error in the ideality factor ($\eta_{assumed} = 1.001 * \eta_{correct}$) would cause a measurement error of 0.36 °C at a temperature of 85 °C.

9.2 IDAC Current Ratio

The final temperature measurement accuracy is highly dependent on the excitation current ratio I2/I1. Therefore, you need to calibrate the IDAC output current before the actual measurement. The basic theory is to measure the voltage drop on R_{cal} at I1 and I2 and use the actual N = I2/I1 temperature calculation in Equation 9.

The error is mainly due to the ADC and the unity gain PGAs present in the input path.

Error due to INL:

Rcal = 450 Ω

At I1 = 30 µA, V_{Rcal} (voltage drop across the calibration resistor) is 13.5 mV.

At I2 = 600 µA, V_{Rcal} = 270 mV.

The 8-LSB INL (max. limit) of ADC at a 12-bit resolution, which means a maximum 5-mV voltage measurement error.

$$Error_{I1} = \frac{13.5\ mV + 5\ mV}{13.5 mV} - 1 = 37\%$$

$$Error_{I2} = \frac{270 m\ V + 5\ mV}{270 m\ V} - 1 = 1.8\%$$

Then, the total equivalent gain error caused by the ADC INL in the excitation current ratio calibration is +/–38.8 percent.

Error due to the PGA gain: 0.5 percent

Therefore, the I2/I1 total error is about 39.3 percent.

The gain error of the ADC does not affect the current ratio calibration because it affects both I1 and I2 measurement and will be canceled when you divide I1 by I2.

In this case, the final temperature error caused by the current ratio calibration is:

$$Error_{Cal} = 1 - \frac{In20}{In(20 \times 1.093)} = 0.099 = 9.9\%$$

A 39.3% error in the current ratio would cause a measurement error of 0.35 °C at a temperature of 85 °C.

9.3 ADC Error

The ADC has three sources of error: offset error, gain error, and ADC nonlinearity. The offset calibration process in firmware takes care of the ADC offset.

9.3.1 Gain Error

Considering a 1.3% gain error, the temperature error caused by ADC in 85 °C is about 4.7 °C.

9.4 Summary of Error Sources

Table 1 summarizes the temperature error due to various error sources. In the table, all error sources except the ideality factor are due to the PSoC signal chain. The errors due to the PSoC signal chain are for worst-case conditions. The Test Results section provides the practically observed results of the temperature measurement.

Table 1. Temperature Measurement Error Sources

Error Source	Error at 85 °C	Comments
IDAC Current Ratio	0.35 °C	INL error calculation is the worst case; in practice, the error should be much smaller.
ADC Gain Error	4.7 °C	Use internal reference.
ADC Gain Drift	–	–
Ideality Factor (Special)	0.36 °C	For a 0.1 % error in ideality factor at 85 °C. This error is due to the transistor itself, not to the PSoC signal chain.

10 Test Results

Temperature measurement in the range of 0-80 °C was made using the hardware setup shown in Figure 7 and Figure 8. Thermonics instrument was used to force different temperatures onto the transistor on the CY8CKIT-036. The actual temperature of the transistor was measured using an instrument called MicroTherma2, which was used as the reference while making temperature measurements. The MicroTherma2 is an accurate, thermocouple (Type-K) based temperature measurement instrument.

A calibration resistor of 450 Ω was used and the correct ideality factor was found to be 1.043619.

Figure 9 shows the observed variation of ΔVD with respect to temperature. This variation is seen to be almost linear and similar to the theoretical 'Temperature versus ΔVD' plot shown in Figure 4 for N=20.

Figure 9. Measured ΔV_D Using PSoC 1

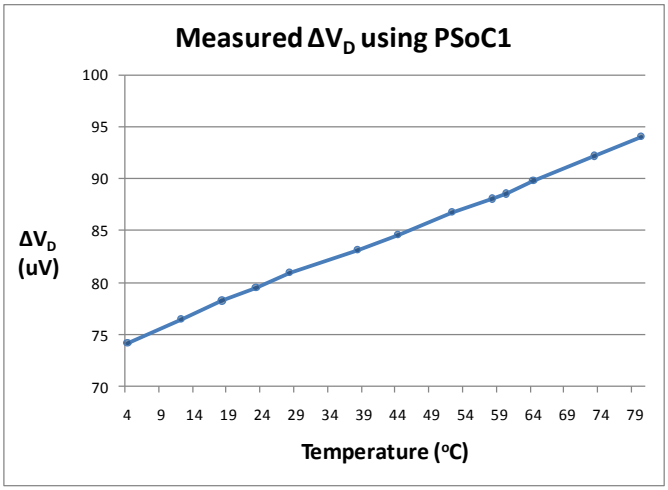
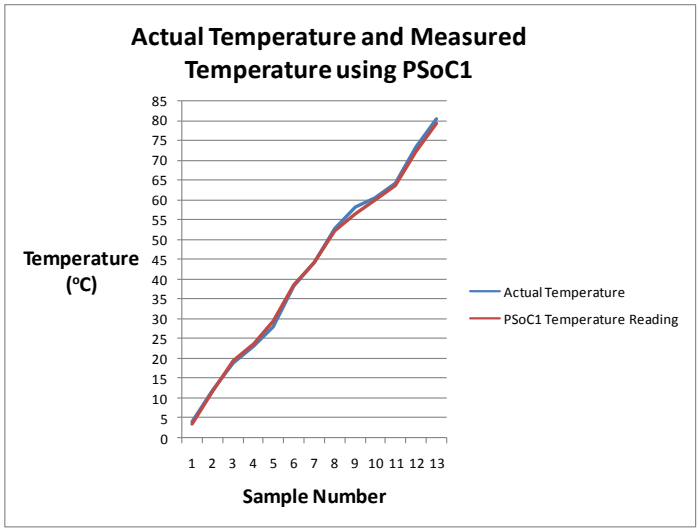


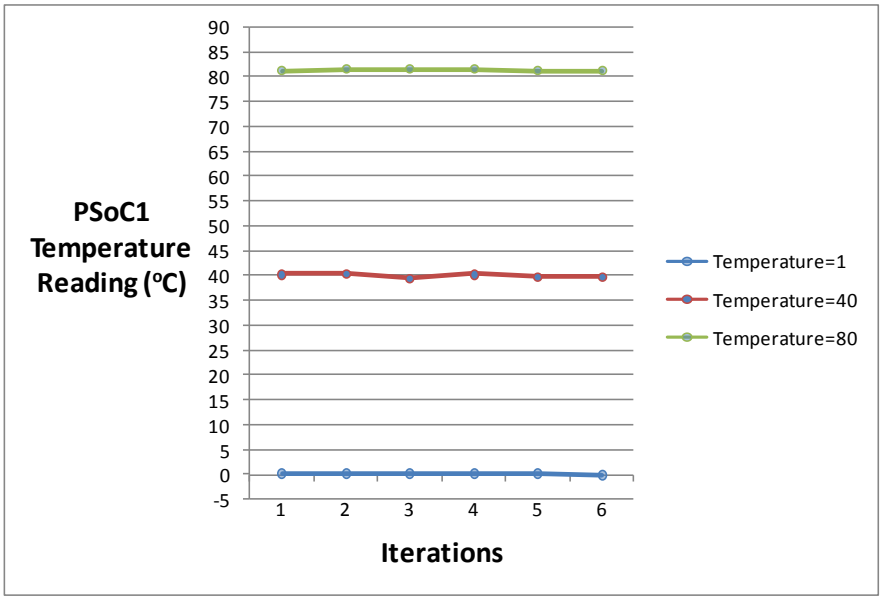
Figure 10 shows a comparison between the actual temperature and the temperature measured by PSoC 1. An accuracy of ±2 °C and resolution of less than 1 °C was observed over the range of 0-80 °C.

Figure 10. Actual Temperature and Measured Temperature using PSoC 1



To verify the repeatability of the results, temperatures of 1, 40 and 80 °C were forced multiple times onto the transistor and the PSoC1's readings were observed. The Figure 11 shows a plot of the multiple readings at these 3 temperatures.

Figure 11. Repeatability Test Results



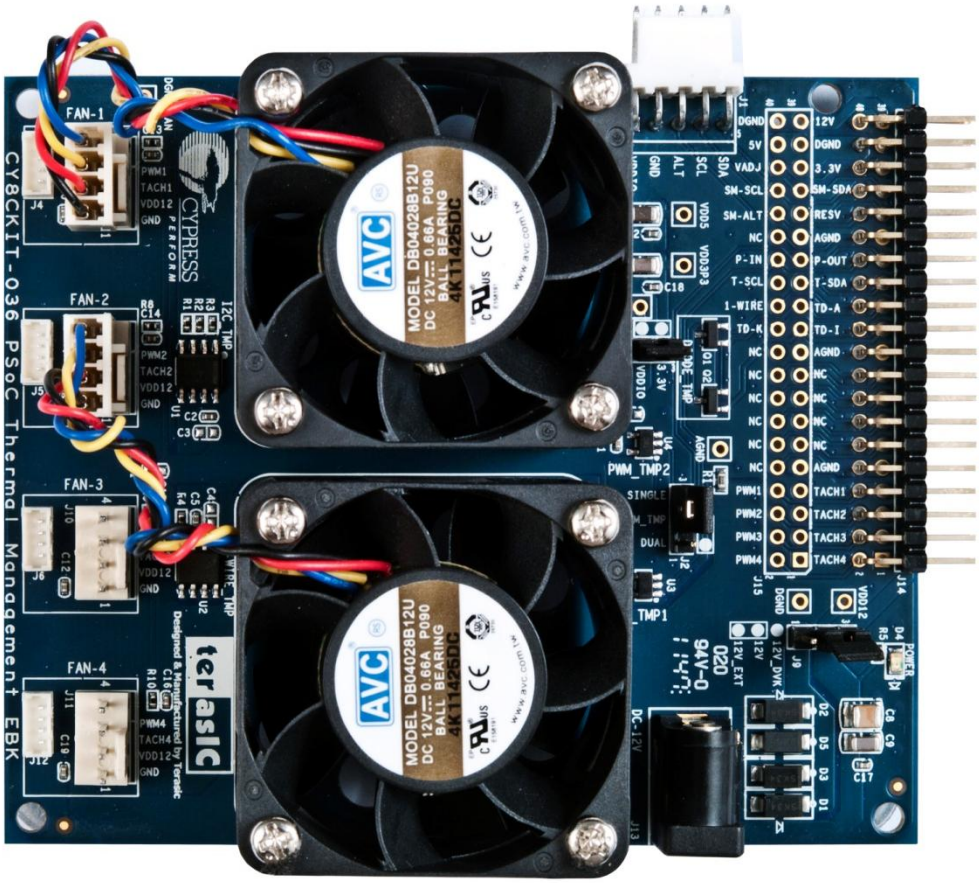
11 CY8CKIT-036 for Diode Temperature Measurement

CY8CKIT-036 is intended to provide a demonstration and development platform for developing system thermal management coprocessor solutions with compelling example projects that demonstrate a variety of modes:

- Temperature monitoring
- Open-loop and closed-loop fan control
- Thermal zone management: the relationship between temperatures and cooling functions
- Algorithms to detect thermal and cooling failures or warnings

This kit comes with two MMBT3904 (SOT-23 package type) transistor diodes connected in anti-parallel fashion. This kit has been used for the example project of this application note. More details about the kit can be found in the Kit User Guide.

Figure 12. CY8CKIT-036 PSoC Thermal Management Expansion Board Kit



12 Summary

This application note explains how the analog features in CY8C28xxx family of devices enable accurate temperature measurement using general purpose transistor or thermal diodes.

Document History

Document Title: AN78920 – PSoC® 1 Temperature Measurement Using Diode

Document Number: 001-78920

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3632571	DIMA	05/31/2012	New application note
*A	3691380	PRKU	07/24/2012	Document revised to improve clarity in images.
*B	4404906	SREH	06/11/2014	Sunset Review. Updated PSoC Designer version to 5.4.
*C	4762918	DIMA	05/12/2015	Added Error Budget Analysis. Updated project to PSoC Designer 5.4 SP1. Sunset Review. Updated 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

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2012-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.
Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
Use may be limited by and subject to the applicable Cypress software license agreement.