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.

CE223621 – Interfacing PSoC 4 with a Temperature Sensor

Objective

This code example demonstrates how to implement an analog front end (AFE) for a thermistor using PSoC 4100PS.

Overview

This code example demonstrates how to measure the thermistor resistance and calculate temperature. The PSoC® Creator™ Thermistor Calculator Component simplifies the math-intensive resistance-to-temperature conversion. The measured thermistor resistance and calculated temperature are sent over I^2C to a host PC running Cypress's Bridge Control Panel (BCP) software. Refer to the Related Documents section for a list of application notes that discuss AFE implementation for different types of sensors.

Requirements

Tools: PSoC Creator 4.2 or later versions, Bridge Control Panel (part of PSoC Programmer™) 1.18 or later versions

Programming Language: C (Arm® GCC 5.4)

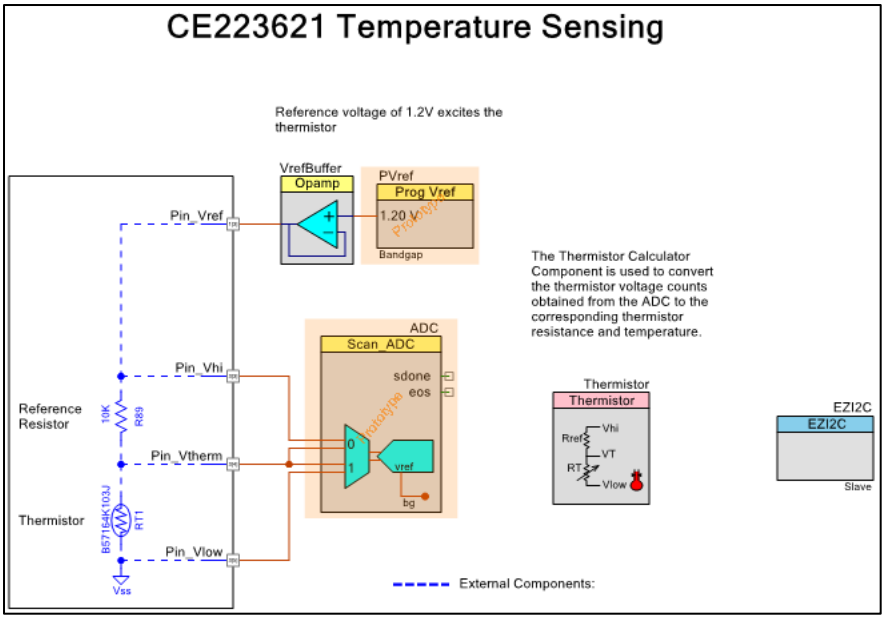
Associated Parts: PSoC 4100PS

Related Hardware: CY8CKIT-147 PSoC 4100PS Prototyping Kit

Design

Figure 1 shows the PSoC Creator schematic for interfacing a thermistor with the PSoC 4100PS.

Figure 1. Temperature Sensing Schematic



A 10-kΩ reference resistor is connected in series with the thermistor (B57164K103J).

The thermistor and R_{ref} are excited using a 1.2 V bandgap voltage as the reference voltage. The bandgap voltage is independent of supply voltage fluctuations and therefore, provides a stable voltage reference. This voltage is generated using the Programmable Reference Component PV_{ref}, and is buffered using an opamp. The same bandgap voltage is used as the ADC reference. This enables measurement in the full-scale range of the ADC and results in an increased resolution of the voltage measurement.

Three voltage signals (V_{therm}, V_{low}, and V_{hi}) from the resistor divider are connected to two differential channels of the Scanning successive approximation register (SAR) ADC Component. The thermistor resistance (R_T) is calculated from the ADC count using the following equation:

$$R_T = R_{ref} \times \left(\frac{V_{therm} - V_{low}}{V_{hi} - V_{therm}}\right)$$

The temperature value is then derived by passing the measured resistance to the Thermistor Calculator Component. The Thermistor Calculator Component uses a lookup table to calculate the temperature with a resolution of 0.1 °C.

The measured thermistor voltage ($V_{therm} - V_{low}$), the thermistor resistance, and the temperature value are sent over I^2C to a host PC.

Design Considerations

This design can be adapted to other thermistor sensors. You may need to change the R_{ref} Value depending on the thermistor characteristics.

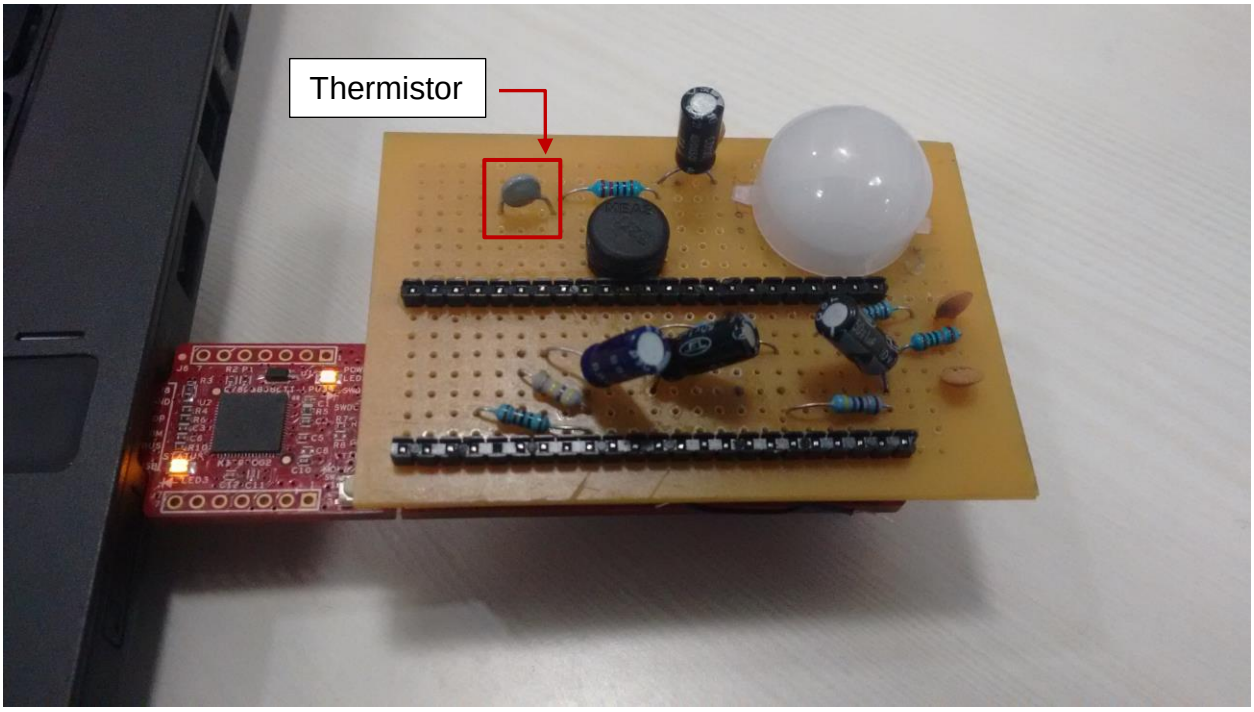
This code example is designed for the PSoC 4100PS Prototyping Kit. The design is easily portable to other kits and PCBs, typically by just changing the sensor or I^2C pin assignments.

Hardware Setup

This code example uses the sensor and reference resistor mounted on a general-purpose PCB (used as customized shield hardware) to make the connection shown in Figure 1.

Stack up the custom shield and CY8CKIT-147 PSoC 4100PS Prototyping Kit and connect it to your computer's USB port as Figure 2 shows.

Figure 2. Hardware Connection



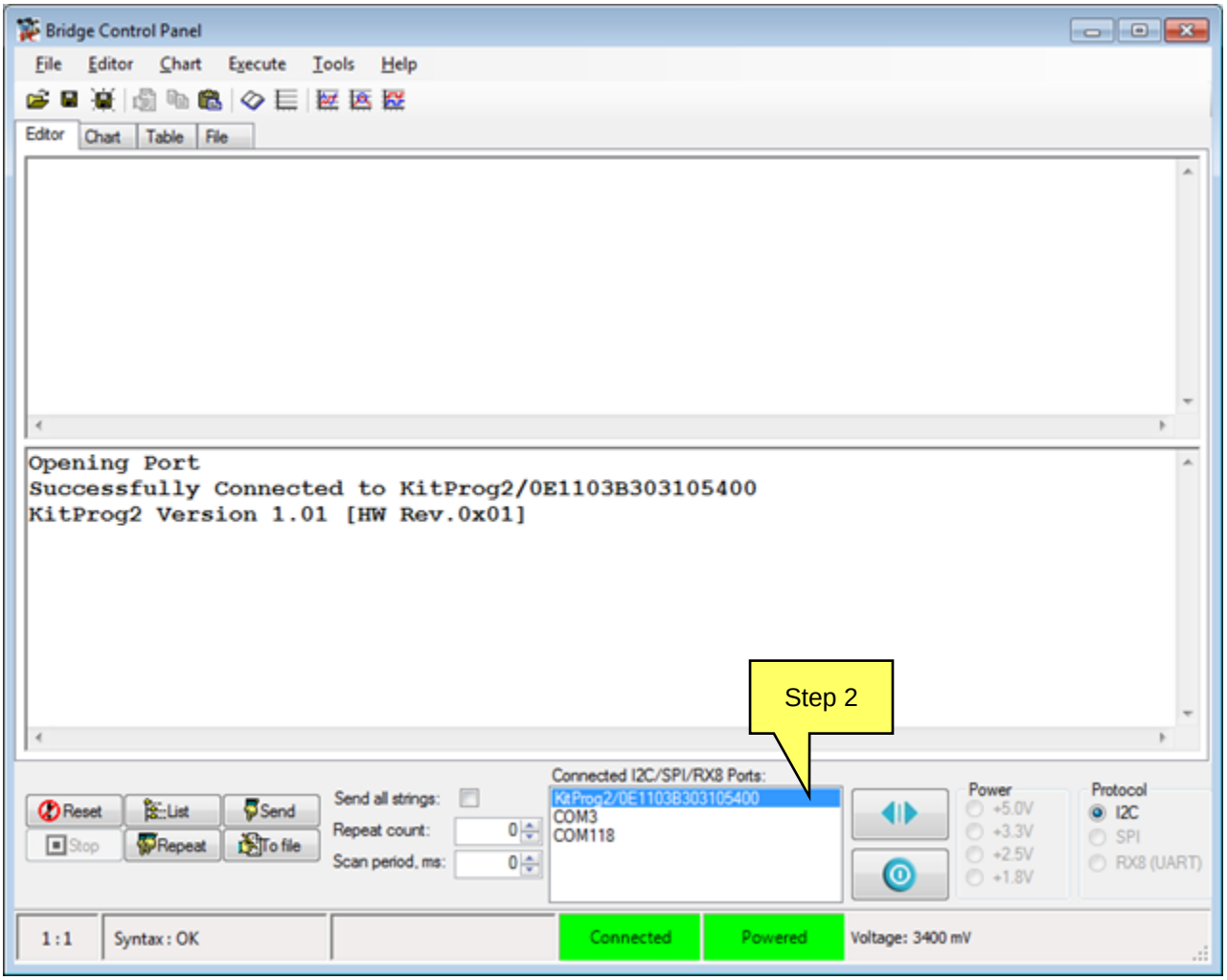
Software Setup

This section describes how to set up the Cypress BCP software to view the sensor data sent over I^2C.

The BCP software is installed automatically as part of the kit software installation. It can also be downloaded as part of PSoC Programmer software. Follow these steps to configure the BCP software:

1. Open the BCP software from: **Start** > **All Programs** > **Cypress** > **Bridge Control Panel <version>** > **Bridge Control Panel <version>.**
2. Under **Connected I2C/SPI/RX8 Ports**, select **KitProg2/<serial number>** as Figure 3 shows. Note that the PSoC 4100PS Prototyping Kit must be connected to the USB port of your computer.

Figure 3. Bridge Control Panel

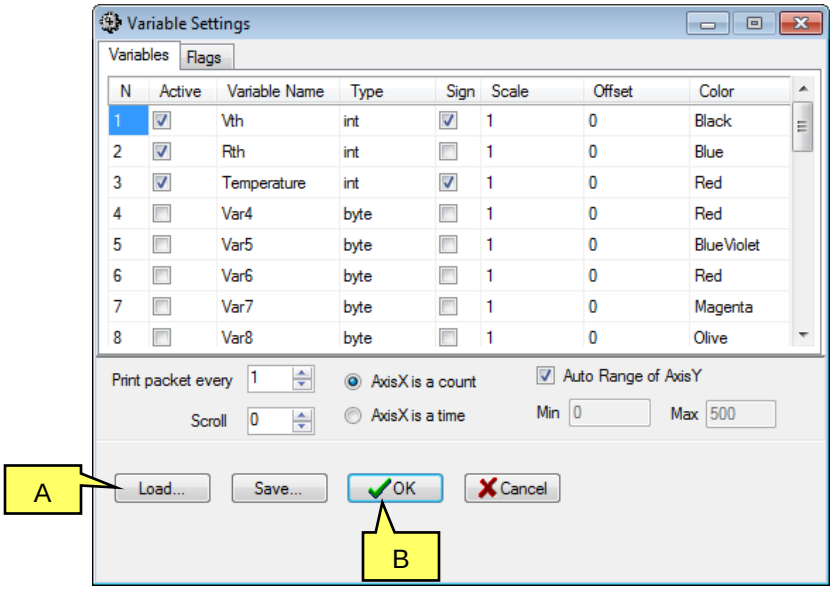


3. Select the menu item **Tools** > **Protocol Configuration**, navigate to the **I2C** tab, and set the **I2C speed** to '100 kHz'. Click **OK**.

4. Select the menu item **Chart** > **Variable Settings**, and
   A. **Load** the *CE223621_Temperature_Sensing.ini* file from the following path: `CE223621_Temperature_Sensing\CE223621_Temperature_Sensing.cydsn\BCP Command\`.
   B. Click **OK** as Figure 4 shows.

   This file includes the variable names, their data type, and their signs to represent the data sent over I^2C.

Figure 4. Variable Settings in Bridge Control Panel Software



The BCP software is now ready for reading and displaying the sensor data. Refer to the Operation section for the testing procedure.

Components

Table 1 lists the PSoC Creator Components used in this example and the hardware resources used by each Component.

Table 1. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
Scanning SAR ADC	ADC	v2.20	SAR ADC
PVref	PVref	v1.0	Programmable Reference Block (PRB)
Opamp	VrefBuffer	v1.20	Continuous Time Block (CTB)
Thermistor Calculator	Thermistor	v1.20	-
EZI2C Slave (SCB mode)	EZI2C	v4.0	Serial Communication Block (SCB)
Analog Pin	Pin_Vhi, Pin_Vtherm, Pin_Vlow, Pin_Vref	v2.20	I/O

Parameter Settings

Table 2 lists the nondefault settings of all the Components used in the design.

Table 2. Component Parameters

Component Instance Name	Settings (Non-Default)
ADC	Free-run scan rate (SPS): 1000
PVref	-
VrefBuffer	Mode: Follower Output: Output to pin Power/Bandwidth: High
Thermistor	Implementation: LUT Temperature > Max: 125 °C Min: -40 °C Temperature and Resistance > Max: 260 Ω Min: 41,9380 Ω
EZI2C	-
Pin_Vhi, Pin_Vtherm, Pin_Vlow, Pin_Vref	External terminal: Enabled

Note: EZI2C pins are embedded within the Component.

Design-Wide Resources

Table 3 lists the physical pins used.

Table 3. Pin Names and Locations

Pin Name	Location
EZI2C: SCL	P3[6]
EZI2C: SDA	P3[7]
Pin_Vhi	P3[3]
Pin_Vlow	P3[5]
Pin_Vref	P1[3]
Pin_Vtherm	P3[4]

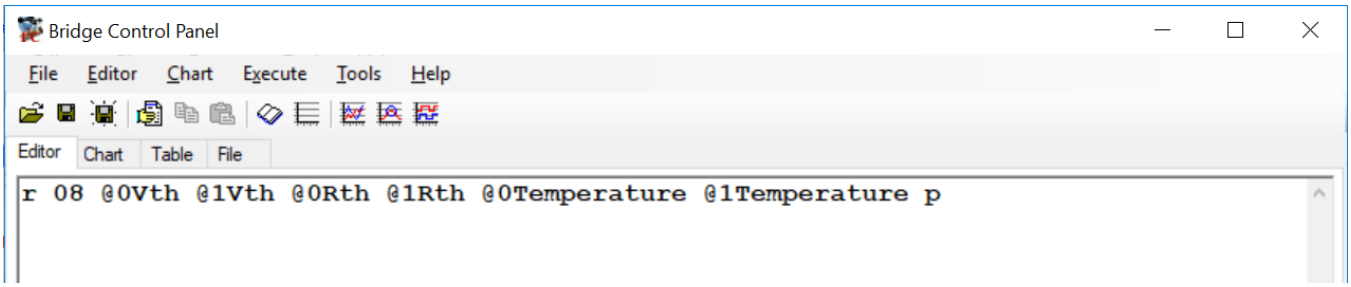
Operation

Follow these steps:

1. Open the project attached with this code example in PSoC Creator.
2. Build the project; select the PSoC Creator menu item **Build** > **Build CE223621_Temperature_Sensing**.
3. Connect the PSoC 4100PS Prototyping Kit to your computer's USB port as described in the Hardware Setup section.
4. Program the PSoC 4100PS device; select **Debug** > **Program**.
5. Configure the BCP software as described in the Software Setup section.
6. Select **File** > **Open File**. Open the *CE223621_Temperature_Sensing.iic* file from the following path:
   `CE223621_Temperature_Sensing\CE223621_Temperature_Sensing.cydsn\BCP Command\`

   This file contains the read command to be executed by the BCP software. The command appears on the BCP software as Figure 5 shows.

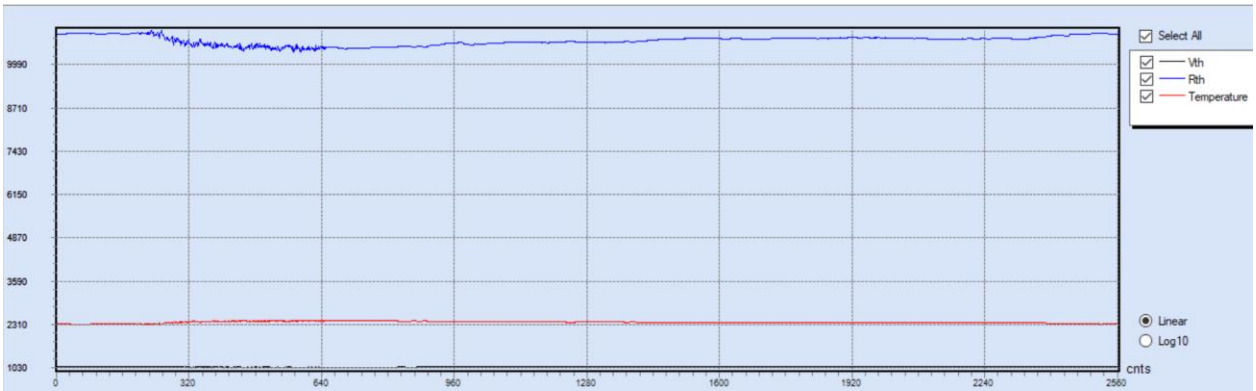
Figure 5. Read Command in the Bridge Control Panel



7. Click the read command on the **Editor** tab. Then click the **Repeat** button to read the sensor data continuously.
8. Go to the **Chart** tab and observe the plot of the three values – thermistor voltage, thermistor resistance, and temperature – that are read from the PSoC 4100PS device. See Figure 6.

   Notice that the temperature value is scaled by 100. For example, if the temperature is 25 °C, then the displayed count is 2500. V_{th} is the ADC count that corresponds to the voltage across the thermistor. R_{th} is the calculated thermistor resistance in ohms.

Figure 6. Temperature Sensor Values Displayed on the Bridge Control Panel Chart



Related Documents

Table 4 lists all relevant application notes, device datasheets, technical reference manuals, Component datasheets, and development kits.

Table 4. Related Documents

Application Notes		
AN79953	Getting Started with PSoC 4	Describes the PSoC 4100PS
AN223616	AFE Implementation Using PSoC 4	Discusses the AFE implementation of different types of sensors
AN66477	PSoC 3, PSoC 4, and PSoC 5LP - Temperature Measurement with a Thermistor	Describes thermistor temperature measurement using PSoC 3, PSoC 4, and PSoC 5LP devices
PSoC Creator Component Datasheets		
Scanning SAR ADC	Supports multiple-channel hardware scan with single-ended and differential input modes	
PVref	Generates configurable voltage references using the internal bandgap voltage or supply voltage V_{DDA}	
Opamp	Supports the voltage follower mode and the Opamp mode with configurable power	
Thermistor Calculator	Supports the Steinhart-Hart equation and LUT method to calculate the temperature	
EZI2C Slave	Simplified I2C slave implementation	
Pins	Supports the connection of hardware resources to physical pins	
Device Documentation		
PSoC 4100PS Datasheet		
PSoC 4100PS Architecture Technical Reference Manual		
PSoC 4100PS Register Technical Reference Manual		
Development Kit (DVK) Documentation		
CY8CKIT-147 PSoC 4100PS Prototyping Kit		

Document History

Document Title: CE223621 - Interfacing PSoC 4 with a Temperature Sensor

Document Number: 002-23621

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6153503	DIMA	07/10/2018	New code example.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturers' 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

PSoC Creator is a trademark of Cypress Semiconductor Corp.

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.