CE210383 – PSoC Temperature Sensing with an RTD

Objective

This example demonstrates how to use a PSoC® 3, PSoC 4, or PSoC 5LP device to measure temperature using a resistance temperature detector (RTD).

Overview

This code example shows how to use a PSoC 3, PSoC 4, or PSoC 5LP device to measure temperature using an RTD. RTDs are sensors commonly used for temperature measurement. PSoC devices contain the necessary resources to do a four-wire RTD measurement; this includes an IDAC, an ADC, and a CPU to perform calculations. For more information on the theory behind RTD measurement, see AN70698.

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, PSoC 4100, PSoC 4100 BLE, PSoC 4100 M, PSoC 4200, PSoC 4200 BLE, PSoC 4200 L, PSoC 4200 M, All PSoC 5LP

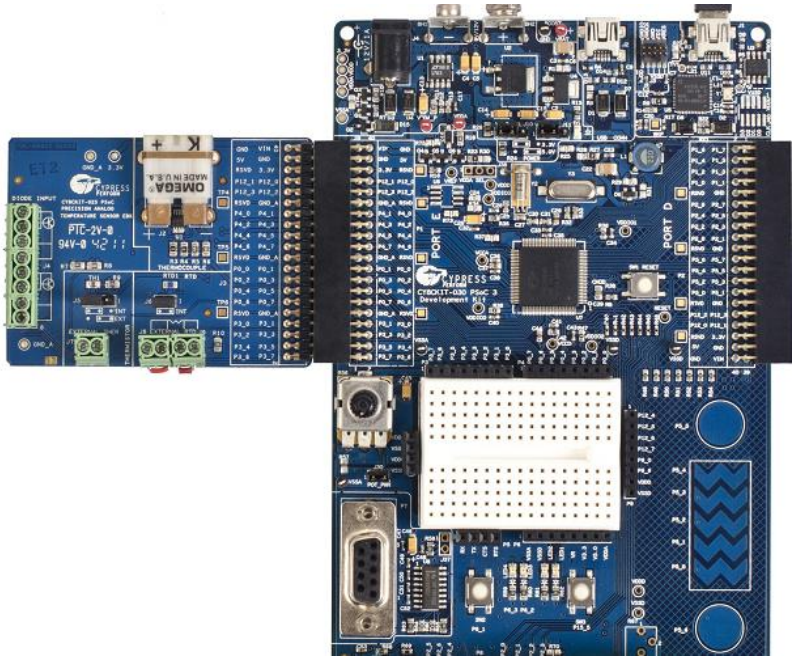
Related Hardware: CY8CKIT-025, CY8CKIT-030, CY8CKIT-050, CY8CKIT-042, CY8CKIT-042-BLE, CY8CKIT-042-BLE-A, CY8CKIT-043, CY8CKIT-044, CY8CKIT-046.

Hardware Setup

For PSoC 3 and PSoC 5LP devices, follow these steps:

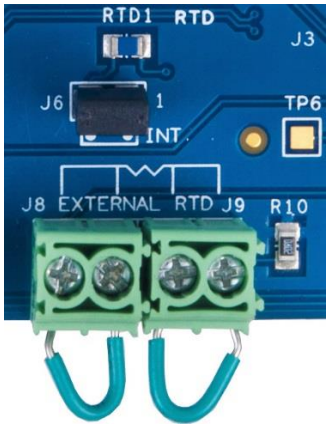
1. Plug CY8CKIT-025 into PORT E of either CY8CKIT-030 or CY8CKIT-050, as Figure 1 shows.

Figure 1. CY8CKIT-025 Plugged into CY8CKIT-050



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

Figure 2. Shorting J8 and J9 on KIT-025



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

For PSoC 4 devices follow these steps:

1. Connect the CY8CKIT-025 to a PSoC 4 kit using fly wires. Connect the fly wires between the CY8CKIT-025 and the PSoC 4 kit as shown below

CY8CKIT-025 Port.Pin	Name of pin in cydwr of CE
P3.1	RTD_Current
P4.0	RTD_Positive
P4.1	RTD_Negative
P3.4	RefRes_Current
P4.4	RefRes_Positive
P4.5	RefRes_Negative

2. On KIT-025, ensure that there are wires shorting the terminals of J8 and J9 as Figure 2 shows.
3. (Optional) Plug an external RTD between J8 and J9, and remove the J6 jumper on CY8CKIT-025.
4. Connect P0.5 on J4 of CY8CKIT-042 to P12.6 on J8 of CY8CKIT-019. This connects the UART.

Software Setup

The code example supports a terminal emulator interface. A UART interface outputs the temperature value 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 4. TeraTerm is open-source and downloadable directly from the author's website: https://en.osdn.jp/projects/ttssh2/.

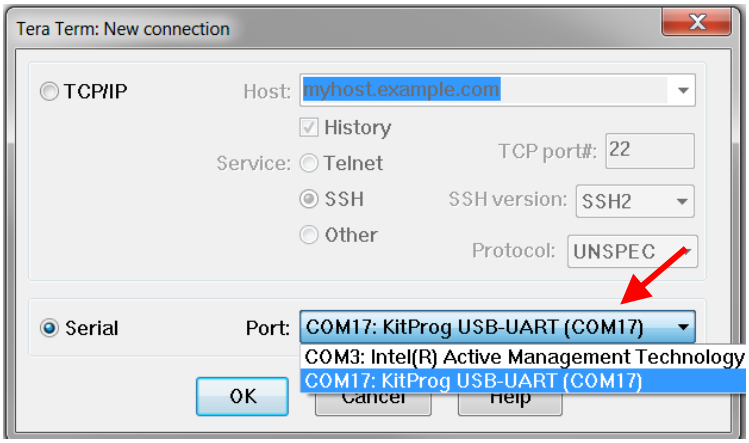
1. Create new connection.

Launch TeraTerm and select **File** > **New connection**. Select **Serial** as the connection type, and do the following:

a. PSoC 4 Devices: Choose the **KitProg USB-UART** communication port (COM), as Figure 3 shows. The actual COM port number will vary between computers and USB ports. If multiple communication ports are listed, it can be helpful to disconnect and reconnect the development kit's USB cable and look for the COM port that disappears and reappears.

b. PSoC 3 or PSoC 5LP devices: Choose the COM port where you connected the DB9 cable.

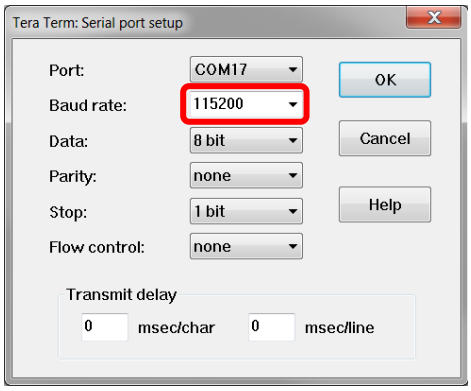
Figure 3. New Connection Creation



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 it is good to also confirm that the other settings are as shown in Figure 4.

Figure 4. Terminal Emulator Setup Parameters



Operation

1. Load the workspace into PSoC Creator by opening `<Install_Directory>\CE210383\CE210383.cywrk`.
2. Select the project you wish to use. One project is for PSoC 4, the other is for PSoC 3/PSoC 5LP. To select the project, right-click on it and select **set as active project**.

Note:

i. PSoC 3/PSoC 5LP project by default uses PSoC 5LP. It can be changed to a PSoC 3 device by right-clicking the project and selecting **device selector**.

ii. The PSoC 4 project by default uses the 4200 device. It can be changed to another PSoC 4 device by right-clicking the project and selecting **device selector**.

3. Build the example project by navigating to **Build** > **Build *<Project Name>*** in PSoC Creator.
4. Connect the device/board to a programmer connected to a PC. On-board KitProg devices are already connected to the programming pins of the on-board device.
5. Program the example to the device by navigating to **Debug** > **Program**.
6. Power the device if not already powered.
7. Open the terminal program following instructions in the Serial Terminal section.
8. Observe temperature display in the terminal program or on the LCD display of the -030 or -050 kit.

Design and Implementation

This example measures temperature using a four-wire RTD that uses a reference resistor, as Figure 5 shows. The RTD and reference resistor are excited through an IDAC. The voltage across the RTD and the reference resistor are measured by the ADC. The two measurements are combined to determine the resistance of the RTD.

Equation 1 $R_{RTD} = \frac{V_{RTD}}{V_{REF}} * R_{Ref}$

Where

V_{RTD} is the voltage measured across the RTD.

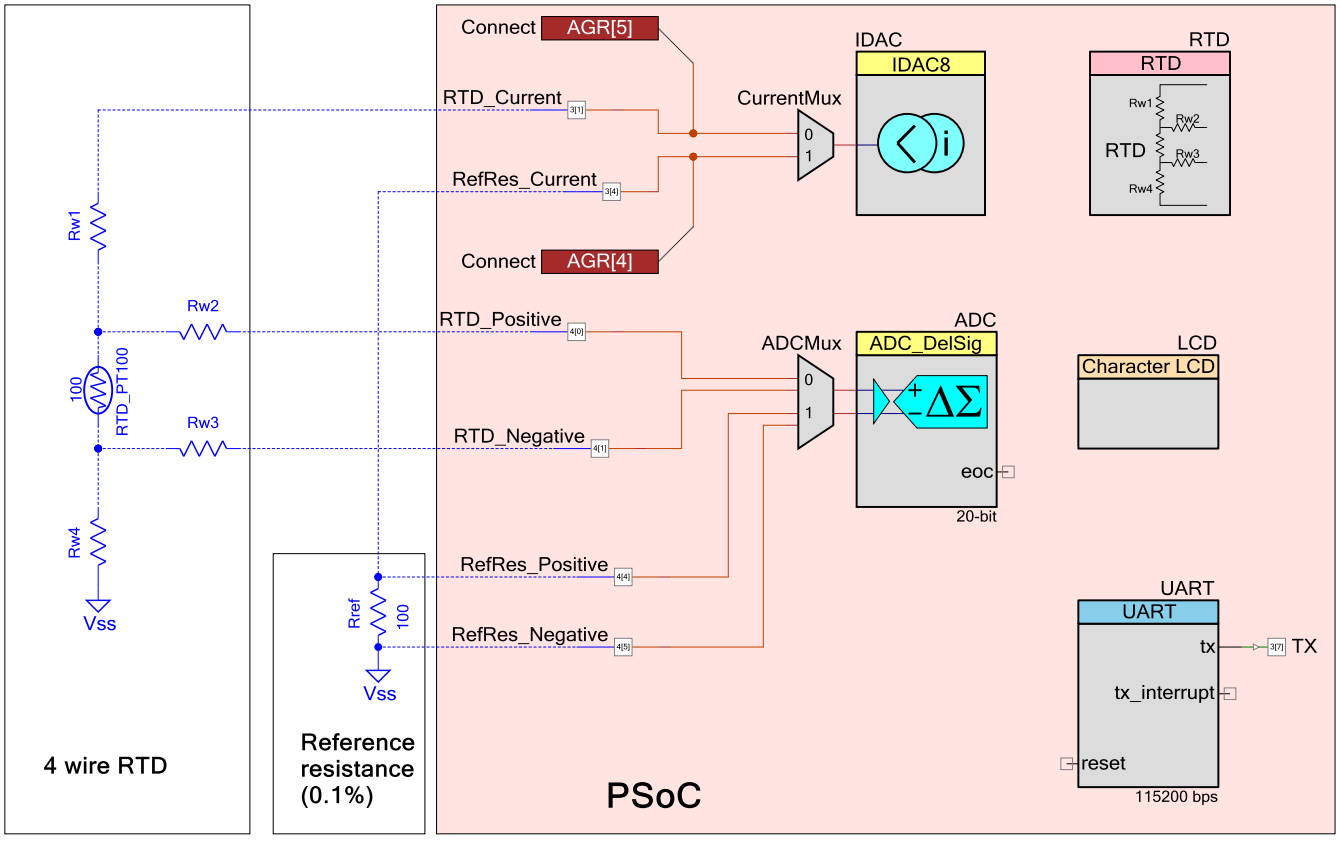
V_{REF} is the voltage measured across the reference resistor.

R_{REF} is the resistance of the reference resistor. For this example, it is 100 Ω.

The resistance is then given to the RTD Calculator Component “RTD”, which calculates the temperature from the resistance value.

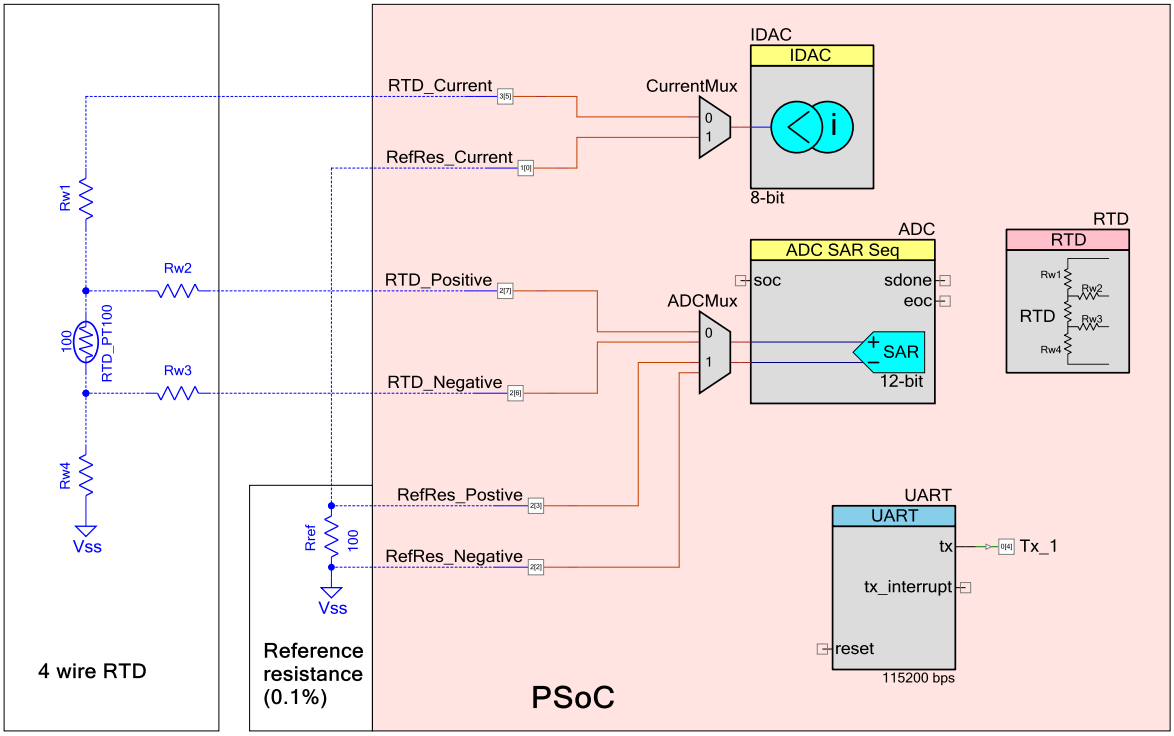
The temperature result is communicated over UART to a PC, or with KIT-030 and KIT-050 it is also displayed on a character LCD.

Figure 5. PSoC 3/PSoC 5LP RTD Measurement



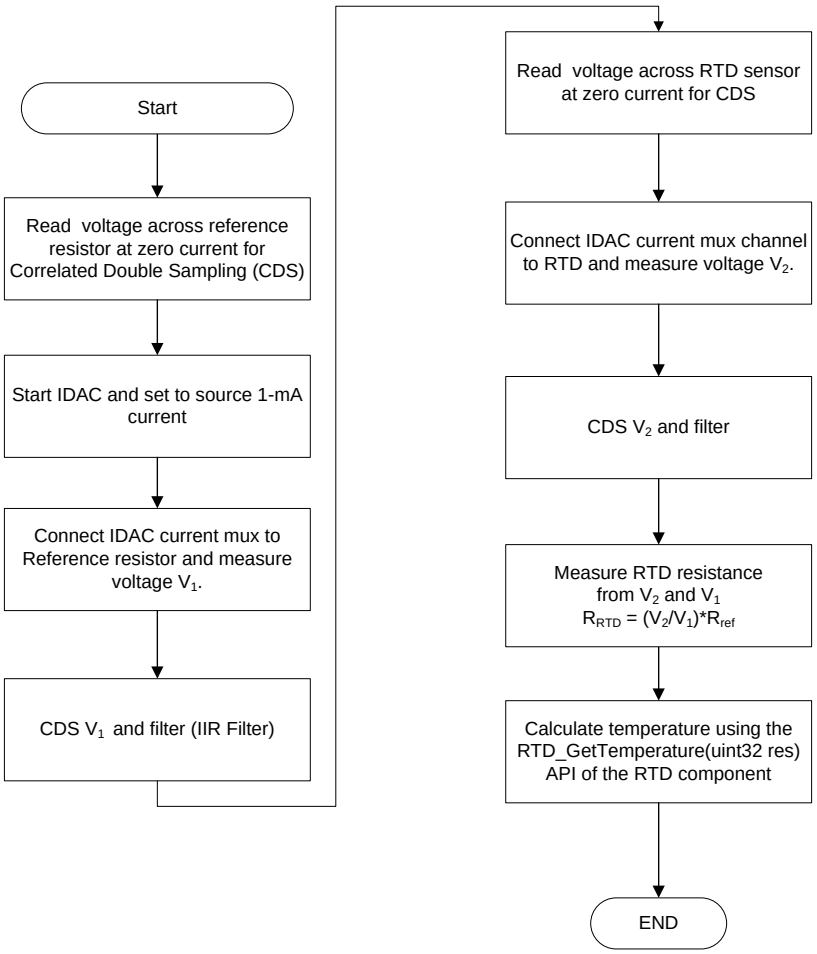
Note that this schematic applies only to PSoC 3 and PSoC 5LP. The schematic for the PSoC 4 project is very similar except for the ADC and IDAC used; in addition, it does not use the LCD. See Figure 6.

Figure 6. PSoC 4 RTD Measurement



The following chart shows the firmware flow for this example.

Figure 7. RTD Measurement Firmware Flow



Design Considerations

Note: The schematic shown in Figure 5 does not match the schematic shown in Figure 5 of AN70698. The reason for this is that this code example uses KIT-025. KIT-025 does not have a dedicated reference resistor—it is shared with a diode temperature sensor. Because the reference resistance is not in series with the RTD, the current through the reference resistance can differ from the current through the RTD. If the currents through the reference resistance and RTD are not equal, you may get a temperature error. Manual analog routing is used to avoid a significant error. In your design, always use a reference resistor in series with the RTD.

Note: KIT-025 is not required for RTD temperature measurement. You can connect your own RTD and reference resistor directly to PSoC device pins. KIT-025 is meant to make the processes easier for demonstration purposes.

If you are designing your own board for RTD measurement, use a reference resistor in series with the RTD, as shown in AN70698.

Components and Settings

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 Conversion Mode: 0 – Single Sample Resolution(bits): 20 Conversion rate (SPS): 166 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	UDBs	Configure: Mode: TX Only Bits per second: 115200
Character LCD	LCD	7 Pins	N/A

Table 2. List of PSoC Creator Components for PSoC 4

Component	Name	Hardware Resources	Non-default Parameter Settings
Sequencing SAR ADC	ADC	1 Sequencing SAR ADC	Channels A clks: 17 Sequenced Channels: 1 Channel 0: AVG checked General Channel sample rate (sps): 3225 Vref Select: Internal 1.024 volts Samples averaged: 16
Current DAC (8-bit) (PSoC 4)	IDAC	1 IDAC	Polarity: Source Resolution: 8-bit Range: 0 – 612 µA Value: 612 µA
RTD_Calculator	RTD	N/A	Temperature Min: -200 Temperature Max: -850 Calculation Error Budget: 0.01
UART	UART	UDBs	Baud Rate (bps): 115200

Design-Wide Resources

Pin locations for PSoC 3, PSoC 5LP and, PSoC 4200

Table 3. Pin locations for PSoC 3, PSoC 5LP, and PSoC 4200

Pin Name	PSoC 3/ PSoC 5LP Pin Location	PSoC 4200 Pin Location
LCD	P2[6:0]	-
RefRes_Current	P3[4]	P1[0]
RefRes_Positive	P4[4]	P2[3]
Ref_Res_Negative	P4[5]	P2[2]
RTD_Current	P3[1]	P3[5]
RTD_Positive	P4[0]	P2[7]
RTD_Negative	P4[1]	P2[6]
TX/ UART: tx	P3[7]	P0[5]
UART: rx	-	
ADC : Bypass	-	P1[7]

Reusing This Example

This example is designed for the CY8CKIT-042, CY8CKIT-050, and CY8CKIT-030. To port the design to a different PSoC device or kit change the target device under the Device Selector and update the pin assignments in the Design Wide Resources Pin settings as needed.

For PSoC 4 designs it is recommended to have the four ADC inputs use the SAR MUX. Check the device datasheet to determine which port is connected to the SAR MUX. Any pin can be used for the IDAC output. UART tx output will vary depending on SCB pin restrictions. Note on all PSoC 4 kits except the CY8CKIT-042 there is a dedicated connection between the UART and the KitProg, this removes the need for a flywire. To determine which pins connect directly between the PSoC 4 device and the KitProg look on the underside of the kit and find the connections printed on the silkscreen, or look in the kit user guide.

Related Documents

Table 4. Related Documents

Application Notes		
AN70698	PSoC - 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
CE210434	PSoC 3 and PSoC 5LP RTD Calibration	Demonstrates how to calibrate out the interchangeability error of an RTD
PSoC Creator Component Datasheets		
RTD Calculator	Component datasheet for RTD Calculator Component	
ADC DelSig	Component datasheet for ADC DelSig Component	
IDAC8	Component datasheet for IDAC8 Component	
UART	Component datasheet for UDB based UART Component	
LCD	Component datasheet for LCD Component	
ADC SAR Seq	Component datasheet for ADC SAR Sequencer Component	
IDAC	Component datasheet for IDAC Component	
UART (SCB)	Component datasheet for SCB based UART Component	
Device Documentation		
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals	
PSoC 4 Datasheets	PSoC 4 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		
PSoC 4 Kits		

Document History

Document Title: CE210383 – PSoC Temperature Sensing with an RTD

Document Number: 002-10383

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5077550	TDU	01/14/2016	New spec
*A	5724160	JSLN	05/02/2017	Added support for PSoC Analog Coprocessor Updated template
*B	6071595	TDU	02/12/2018	Updated template, Removed supported for PSoC Analog Coprocessor.

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.