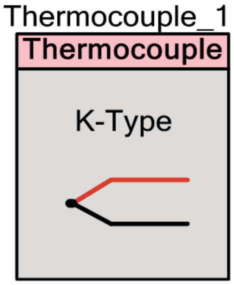
PSoC® Creator™ Component Datasheet

Thermocouple Calculator

1.10



Features

- Supports B, E, J, K, N, R, S, and T Type Thermocouples
- Provides functions for thermo-emf to temperature and temperature to voltage conversions
- Displays Calculation Error Vs. Temperature graph

General Description

In thermocouple temperature measurement, the thermocouple temperature is calculated based on the measured thermo-emf voltage. The voltage to temperature conversion is characterized by the National Institute of Standards and Technology (NIST), and NIST provides tables and polynomial coefficients for thermo-emf to temperature conversion. The NIST tables and polynomial coefficients can be found in the following link:

http://srdata.nist.gov/its90/download/download.html

Thermocouple temperature measurement also involves measuring the thermocouple reference junction temperature and converting it into a voltage. The Thermocouple Calculator component simplifies the thermocouple temperature measurement process by providing APIs for thermo-emf to temperature conversion and vice versa for all thermocouple types mentioned above, using polynomials generated at compile time. The thermocouple component evaluates the polynomial in an efficient way to reduce computation time.

When to Use a Thermocouple Calculator

The component has only one use case. The component provided API is used to convert thermo-emf to temperature and vice versa.

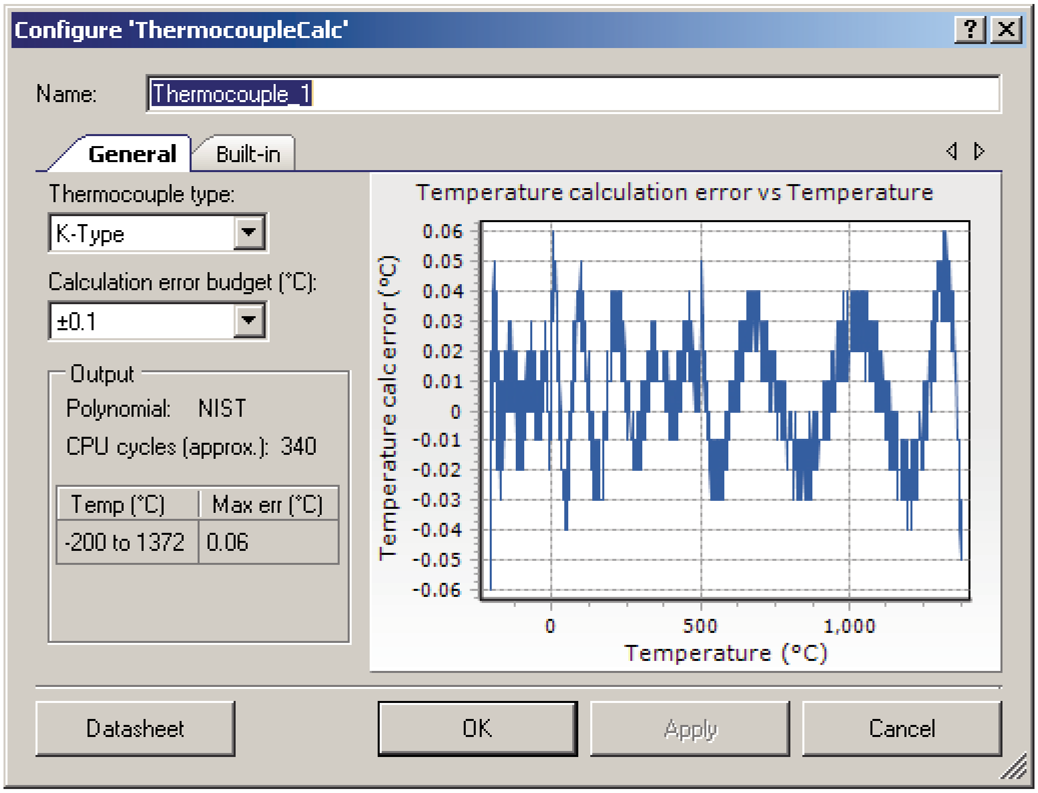
Input/Output Connections

This component is a software component and doesn’t have any input/output connections.

Parameters and Settings

Drag a Thermocouple Calculator component onto your design and double-click it to open the Configure dialog. This dialog has one tab to guide you through the process of setting up the Thermocouple Calculator component.

General Tab



Configure 'ThermocoupleCalc'

Name: Thermocouple_1

Thermocouple type: K-Type

Calculation error budget (°C): ±0.1

Output

Polynomial: NIST

CPU cycles (approx.): 340

Temp (°C)	Max err (°C)
-200 to 1372	0.06

The **General** tab provides the following parameters.

Thermocouple Type

This field allows user selection of the thermocouple type.

Calculation error budget

The user selects the maximum error allowed due to polynomial computation. This is the error only for temperature greater than -200 °C.

Temperature Error Vs Temperature graph

This graph shows the temperature error vs. temperature for the thermocouple type and polynomial.

Polynomial

Displays the order of polynomial that is used for calculations in GetTemperature() API.

Three types of polynomials are used:

- NIST (provided by the National Institute of Standards and Technology)
- 7th order (approximation of NIST polynomial)
- 5th order (approximation of NIST polynomial)

The polynomial is chosen based on the selected error budget. The polynomial with the lowest order which does not exceed the error budget is selected.

CPU cycles

An estimate of the total number of CPU cycles taken for computing the selected polynomial is displayed.

Max error table

The table contains the maximum temperature error for given temperature range for the selected thermocouple type and polynomial.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "Thermocouple_1" to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "Thermocouple".

Function	Description
int32 Thermocouple_GetTemperature(int32 voltage)	Calculates the temperature from thermo-emf in µV
int32 Thermocouple_GetVoltage(int32 temperature)	Calculates the voltage given the temperature in 1/100ths degrees C. Used for calculating the cold junction compensation voltage based on the temperature at the cold junction.

int32 Thermocouple_GetTemperature(int32 voltage)

Description: Calculates the temperature from thermo-emf in µV.

Parameters: Voltage, in µV.

Return Value: Temperature in 1/100ths degrees C

Side Effects: None

int32 Thermocouple_GetVoltage(int32 temperature)

Description: Calculates the voltage given the temperature in 1/100ths degrees C. Used for calculating the cold junction compensation voltage based on the temperature at the cold junction.

Parameters: Temperature in 1/100ths degrees C

Return Value: Thermo-emf voltage in µV.

Side Effects: None

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator components
- specific deviations – deviations that are applicable only for this component

This section provides information on component-specific deviations. Project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment.

The Thermocouple Calculator component has not been verified for MISRA-C:2004 coding guidelines compliance.

Sample Firmware Source Code

PSoC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Functional Description

In 1821, Thomas Seebeck, an Estonian-German physicist discovered that when two dissimilar metals are connected, as shown in Figure 1(a), and one of the junctions is heated, there is a continuous flow of current through the loop. When the loop is broken and the voltage is measured (see Figure 1(b)), the measured voltage is directly related to the temperature difference between the two junctions. This phenomenon where a voltage is produced because of the heating of the junction of two metallic conductors is called thermoelectric effect or Seebeck effect. The junction where heat is applied is called the hot or measurement junction. The other junction is called the cold junction or reference junction, and the voltage developed is called thermo-emf.

Figure 1(a). Thermocouple – Seebeck Effect

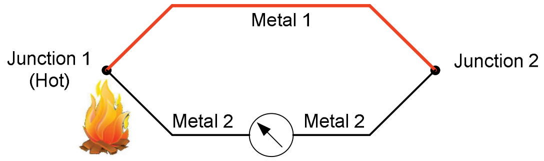
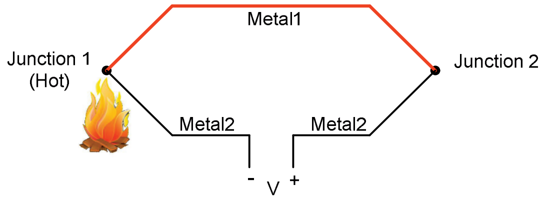


Figure 1(b). Thermocouple –Seebeck Effect



The thermo-emf depends on the following:

- Metals used at the junction
- The temperature difference between the hot and cold junctions
- The absolute value of the cold junction temperature. That is, the thermo-emf produced for hot junction temperature of 100 °C and cold junction temperature of 0 °C will be different from the thermo-emf produced for hot junction temperature of 800 °C and cold junction temperature of 700 °C even though the temperature difference in both cases is 100 °C.

Depending on the types of metals used, thermocouples can be classified into multiple types. The types of thermocouples differ in their temperature range of operation and sensitivities (voltage change per unit change in temperature, V/°C). Two major standards, IEC EN 60584-2 and ASTM E230, govern the thermocouple tolerance. The tolerance specifies the maximum error due to replacing one thermocouple with another of the same type.

Table 1 lists some of the popular thermocouple types, their metal combination, temperature ranges, sensitivities, and their tolerances according to ASTM standard. As shown in Table 1, ASTM establishes two thermocouple tolerance standards, standard and special. Tolerance standards are not defined in the whole temperature range.

The National Institute of Standards and Technology (NIST) provides the thermo-emf versus hot junction temperature data for all thermocouple types for cold junction at 0 °C. Cold junction temperature of 0 °C is chosen as reference because the thermo-emf is 0 V at 0 °C. An ice bath usually provides the 0 °C reference temperature. NIST provides a table as well as polynomial coefficients to convert thermo-emf to temperature and vice versa. Figure 2 shows a K-type thermocouple heated at one junction and maintained at 0 °C in the other junction;

Figure **3** (on the next page) shows the thermo-emf versus hot junction temperature graph for a K-type thermocouple for cold junction at 0 °C. The sensitivity of a K-type thermocouple can be found from the NIST table and is approximately 40 μV/°C for temperatures > -100 °C.

Figure 2. K-type Thermocouple with Cold Junction at 0 °C

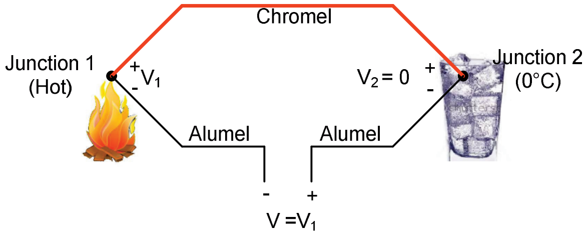
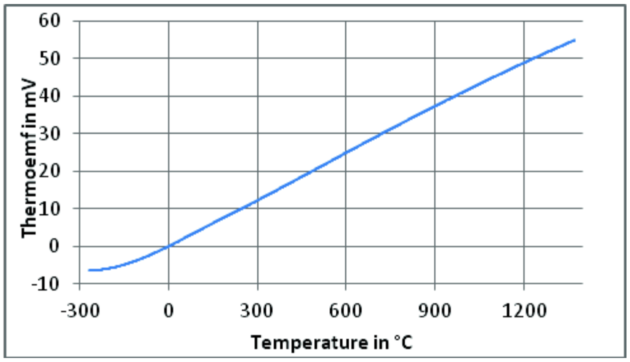


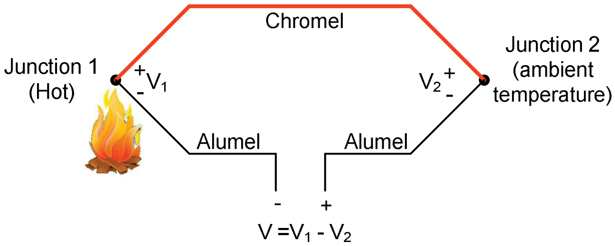
Figure 3. Thermo-emf versus Temperature for K-type Thermocouple (Cold Junction at 0 °C)



Cold Junction Compensation

By measuring the thermo-emf using an ADC, we can easily determine the temperature. But there is one catch; the cold junction has to be maintained at 0 °C to use the NIST tables. It is impractical to provide an ice bath and in most cases the cold junction will be at ambient temperature. If the cold junction temperature is not equal to 0 °C, the cold junction would also develop a thermo-emf, V_2, as shown in Figure 4, reducing the measured voltage, V. To properly measure the hot junction temperature, the cold junction voltage, V_2, has to be added to the final voltage, V.

Figure 4. Cold Junction not at 0 °C



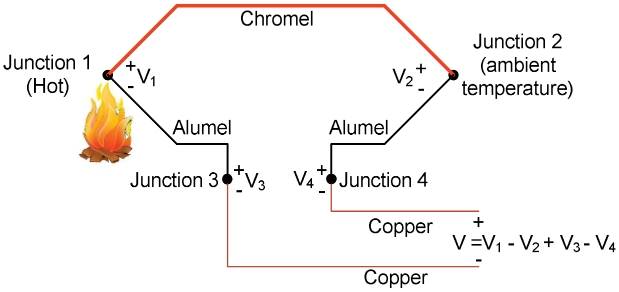
If we find the cold junction temperature, the voltage V_2 can be calculated from the NIST table. Hence in cases where the cold junction is not at 0 °C, the cold junction temperature has to be measured and the thermo-emf corresponding to that temperature has to be added to the thermocouple voltage. This procedure is called cold junction compensation.

A thermistor, RTD, diode, or IC based sensor can be used for cold junction temperature measurement. (Remember that one of these cold junction temperature measurement sensors cannot substitute a thermocouple as they cannot be used for measuring very high temperatures or used in corrosive or rugged environment).

Measuring Thermo-emf

Thermo-emf has to be measured with an ADC by connecting the input leads of the ADC to the thermocouple as shown in Figure 5.

Figure 5. Measuring Thermo-emf



The input (copper) leads of the ADC form two more thermocouples (copper-alumel) adding two more voltages, V_3 and V_4 to the equation. V_3 and V_4 are in opposite directions; they will have the same magnitude as long as both the thermocouples are at the same temperature. Hence, we need to ensure that both the copper-alumel thermocouples are at the same temperature so that the thermo-emf remains unchanged.

Table 1. Thermocouple Types

Thermocouple Type	Metal Content in Positive Leg	Metal Content in Negative Leg	Temp Range (°C)	Sensitivity at 25 °C (µV/°C)	Tolerance (ASTM)		
					Temp Range (°C)	Standard	Special
B	70.4% Platinum (Pt) 29.6% Rhodium (Rh)	93.9% Pt, 6.1% Rh	0 – 1820	0	800 – 1700	0.5%	
E	90% Nickel (Ni) 10% Chromium (Cr)	55% Copper (Cu) 45% Ni	-270 – 1000	61	-200 – 0	1.7 °C or 1%	
					0 - 900	1.7 °C or 0.5%	1°C or 0.4%
J	99.5% Iron (Fe)	55% Cu, 45% Ni	-210 – 1200	52	0 – 750	2.2 °C or 0.75%	1.1 °C or 0.4%
K	90% Ni 10% Cr	95% Ni 5% Various elements	-270 – 1372	41	-200 – 0	2.2 °C or 2%	
					0 – 1250	2.2 °C or 0.75%	1.1 °C or 0.4%
N	84.4% Ni, 14.2% Cr 1.4% Silicon	95.5% Ni, 4.4% Si	-270 – 1300	26	-270 – 0	2.2 °C or 2%	
					0 – 1300	2.2 °C or 0.75%	1.1 °C or 0.4%
R	87% Pt, 13% Rh	100% Pt	-50 – 1768	6	0 - 1450	1.5 °C or 0.25%	0.6 °C or 0.1%
S	90% Pt, 10% Rh	100% Pt	-50 – 1768	6	0 – 1450	1.5 °C or 0.25%	0.6 °C or 0.1%
T	100% Cu	55% Cu, 45% Ni	-270 – 400	41	-200 – 0	1 °C or 1.5%	
					0 – 350	1 °C or 0.75%	0.5 °C or 0.4%

Resources

Component is implemented entirely in firmware. It does not consume any other PSoC resources.

API Memory Usage

The component memory usage varies significantly, depending on the compiler, device, number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements have been done with the associated compiler configured in Release mode with optimization set for Size. For a specific design, the map file generated by the compiler can be analyzed to determine the memory usage.

Configuration	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Thermocouple B-type	710	0	336	0
Thermocouple E-type	730	0	344	0
Thermocouple J-type	702	0	324	0
Thermocouple K-type	730	0	340	0

Configuration	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Thermocouple N-type	899	0	436	0
Thermocouple R-type	911	0	464	0
Thermocouple S-type	895	0	444	0
Thermocouple T-type	706	0	324	0

Performance

The performance of the component depends on the implementation method chosen in the customizer. The measurements below have been gathered using a CPU speed of 24 MHz, with the associated compiler configured in Release mode. These numbers should be treated as approximations and used to determine necessary trade-offs.

GetTemperature API

NIST polynomials

Thermocouple Type	Voltage Range (mV)	Temperature Range (°C)	Polynomial Order	No. of CPU Cycles (PSoC 3)	No. of CPU Cycles (PSoC 5)
B	0.033 to 0.291	100 to 250	7	7900	270
	0.291 to 2.431	250 to 700	8	8500	270
	2.431 to 13.820	700 to 1820	8	8500	270
E	-9.718 to -8.825	-250 to -200	9	9100	310
	-8.825 to 0	-200 to 0	8	8500	310
	0 to 76.373	0 to 1000	8	8500	310
J	-8.095 to 0	-210 to 0	8	8500	270
	0 to 42.919	0 to 760	7	7900	270
	42.919 to 69.553	760 to 1200	5	6700	270
K	-5.891 to 0	-200 to 0	8	8500	310
	0 to 20.644	0 to 500	9	9100	310
	20.644 to 54.886	500 to 1372	6	7300	310
N	-4.313 to -3.990	-250 to -200	9	9100	310
	-3.990 to 0	-200 to 0	9	9100	310

Thermocouple Type	Voltage Range (mV)	Temperature Range (°C)	Polynomial Order	No. of CPU Cycles (PSoC 3)	No. of CPU Cycles (PSoC 5)
	0 to 20.613	0 to 600	7	7900	310
	20.613 to 47.513	600 to 1300	5	6700	310
R	-0.226 to 1.923	-50 to 250	10	9700	340
	1.923 to 13.228	250 to 1200	9	9100	340
	13.228 to 19.739	1200 to 1664.5	5	6700	340
	19.739 to 21.103	1664.5 to 1768.1	4	6100	340
S	-0.235 to 1.874	-50 to 250	9	9100	310
	1.874 to 11.950	250 to 1200	9	9100	310
	11.950 to 17.536	1200 to 1664.5	5	6700	310
	17.536 to 18.693	1664.5 to 1768.1	5	6700	310
T	-6.180 to -5.603	-250 to -200	7	7900	240
	-5.603 to 0	-200 to 0	7	7900	240
	0 to 20.872	0 to 400	6	7300	240

Polynomials of 7th and 5th order

Polynomial Order	No. of CPU Cycles (PSoC 3)	No. of CPU cycles (PSoC 5)
7	7900	240
5	6700	170

GetVoltage API

Thermocouple type	Temperature range (°C)	Polynomial order	No. of CPU cycles (PSoC 3)	No. of CPU cycles (PSoC 5)
B	0 to 100	6	7300	200
E	-20 to 100	8	8500	270
J	-20 to 100	7	7900	240
K	-20 to 100	8	8500	270
N	-20 to 100	8	8500	270
R	-20 to 100	7	7900	240

S	-20 to 100	7	7900	240
T	-20 to 100	8	8500	270

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.10	Added MISRA Compliance section	The component was not verified for MISRA compliance.
	Added Sample Firmware Source Code section	
1.0	Version 1.0 is the first release of the Thermocouple Calculator component	

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 HALFICULAR 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.