AN75511

PSoC® 3 / PSoC 5 – Temperature Measurement with Thermocouples

Author: Praveen Sekar
Associated Project: Yes
Associated Part Family: CY8C34xx, CY8C36xx, CY8C38xx, CY8C55xx
Software Version: PSoC® Creator™
Related Application Notes: None

If you have a question, or need help with this application note, contact the author at pfz@cypress.com

Abstract

AN75511 explains how to make precise temperature measurements using a thermocouple and a PSoC® 3 or PSoC 5. The application note also gives a thermocouple component, which simplifies thermo-emf to temperature conversion. The project associated with the application note provides a code example built using the component, to measure thermocouple temperature on the CY8CKIT-025 – PSoC Precision Analog Temperature Sensor Expansion Board Kit (EBK).

Contents

Introduction 1
Thermocouples – Theory of Operation.....1
Cold Junction Compensation.....3
Measuring Thermo-emf 3
Practical Thermocouple Measurements 4
Thermocouple Temperature Measurement with PSoC 4
Hardware Used - CY8CKIT-025 EBK.....4
PSoC Creator Schematic Description 5
Thermocouple - Voltage to Temperature Conversion...7
Firmware Flow.....9
Performance Measures 9
Conclusion 12
Appendix 13

Introduction

Modern temperature measurements typically use one of four temperature sensors as front end: thermocouple, thermistor, diode, or resistance temperature detector (RTD). The selection of a sensor depends primarily on the cost, accuracy, and temperature range of the sensor. Thermocouples have the largest temperature measurement range and are the most rugged of all sensors making them the first choice for use in industrial and corrosive environment.

Thermocouples – Theory of Operation

Thermoelectric Effect

In 1821, 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

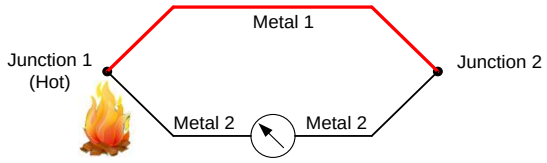
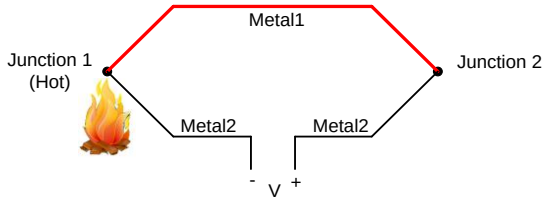


Figure 1(b). Thermocouple –Seebeck Effect



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.

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

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%

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 and Figure 3 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

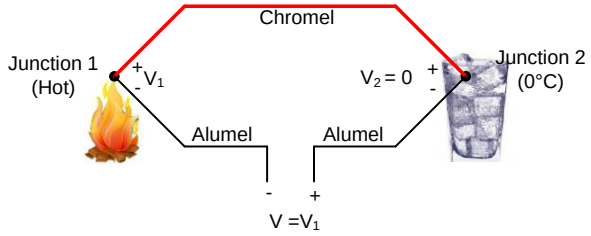
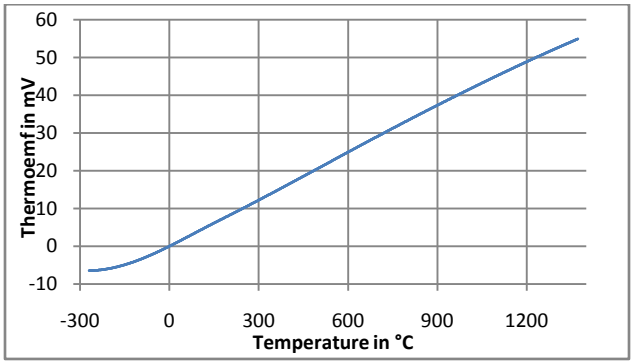


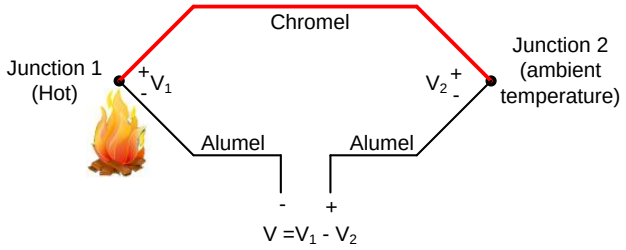
Figure 3. Thermo-emf versus Temperature for K-type Thermocouple (Cold Junction at 0 °C)



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



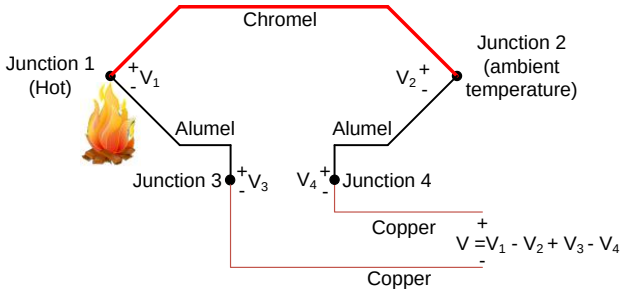
If we find the cold junction temperature, the voltage V_2 can be calculated from the NIST table. Hence in cases where the cold junction is not at 0 °C, the cold junction temperature has to be measured and the thermo-emf corresponding to that temperature has to be added to the thermocouple voltage. This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

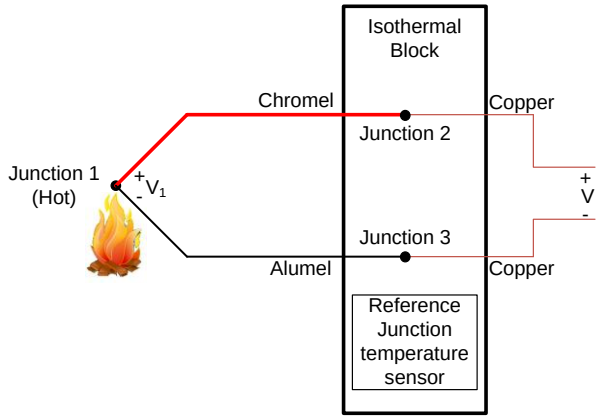


The input (copper) leads of the ADC form two more thermocouples (copper-alumel) adding two more voltages, V_3 and V_4 to the equation. V_3 and V_4 are in opposite directions and they will have the same magnitude as long as both the thermocouples are at the same temperature. Hence we need to ensure that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

Practical Thermocouple Measurements

In practical thermocouple measurements, the two metals are joined at the junction and the junctions at the open-ends (junctions 2 and 3) form the cold junction as shown in Figure 6. It can be shown that the circuits in Figure 5 and Figure 6 are equivalent and the voltage V measured by circuits in the figures 5 and 6 are equal as long as the temperatures of junction 3 and 4 are equal in figure 5 and temperatures of junctions 2 and 3 are equal in figure 6.

Figure 6. Practical Thermocouple



Measuring temperature using a practical thermocouple involves the following steps:

1. Measure thermocouple voltage (V_{TC})
2. Measure cold/reference junction temperature (T_{ref})
3. Convert the cold junction temperature to compensation voltage (V_{ref})
4. Add the cold junction compensation voltage to the thermocouple voltage ($V = V_{TC} + V_{ref}$)
5. Convert the voltage to temperature.

Thermocouple Temperature Measurement with PSoC

The application note has an example project that showcases thermocouple temperature measurement using PSoC. The project displays thermocouple temperature on an LCD. This section describes the project in detail -

Hardware Used - CY8CKIT-025 EBK

CY8CKIT-025 PSoC precision analog temperature sensor EBK is used in the example project. The PSoC Precision Analog Temperature Sensor EBK provides four temperature sensors—thermocouple, thermistor, RTD, and diode. In addition, interface slots let you plug in your own thermocouple, thermistor, RTD, and diode. The EBK can be connected to the CY8CKIT-001 PSoC Development Kit (DVK) or CY8CKIT-030 PSoC 3 Development Kit. Figure 7 shows the kit. For more details on the kit, go to www.cypress.com/go/Cy8CKIT-025.

Figure 7. PSoC Precision Analog Temperature Sensor EBK

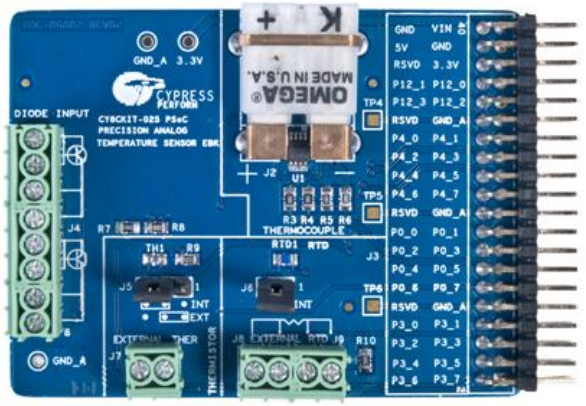


Figure 8. Thermocouple Section of the EBK



Figure 8 shows the thermocouple portion of the kit. A K-type thermocouple connector is provided. The cold junction sensor, DS600 IC, is placed very close to the cold junction terminals. Silk marked U1 shows the position of IC on the board. Ideally an isothermal connection should be provided between the cold junction sensor and the cold junction terminals as shown in Figure 6. This requires a material with very good thermal conductivity providing a thermal connection between the thermal pad of the IC and the cold junction terminals. In cases where this is not possible, we can have the cold junction sensor placed very close to the cold junction terminals as it has been done in CY8CKIT-025. The temperature difference between the cold junction terminals and the IC is expected to be around 0.5 °C in this case.

Note that the thermistor, RTD, or diode on board can also be used for cold junction temperature measurement. The kit project gives you the option of using thermistor for cold junction compensation. Similarly you can also use RTD or diode for cold junction temperature measurement. Cypress application notes 'AN66477 - Temperature measurement with thermistor', 'AN70698 - Temperature measurement with RTD' and 'AN60590 - Diode temperature mesurement' explains thermistor, RTD, and diode temperature measurements with PSoC in detail.

But these temperature sensors (RTD, Thermistor, and IC) are farther from the cold junction than the IC and the temperature difference between the cold junction and these

sensors will be higher if there is airflow. With no significant airflow, the temperature difference between the cold junction terminals and the other sensors is expected to be less than 1 °C.

PSoC Creator Schematic Description

Figure 9 shows the thermocouple measurement circuit (PSoC Creator schematic). The circuit has a five channel ADC, the Thermocouple Component described in page 7, and a character LCD. The five ADC channels and their purpose are listed in Table 2.

Table 2. Five ADC Channels

Channel	Connection	Measurement
0	Thermocouple	Thermo-emf
1	IC voltage output	Cold junction temperature
2	Thermistor voltage	Cold junction temperature
3	Thermistor ref	Cold junction temperature
4	Ground	Offset

As shown in Table 2, both a thermistor and IC are used for measuring cold junction temperature. A switch is used in the project to select between the two cold junction sensors. Make sure the cold junction sensor is isothermal with the cold junction. Additional resistors are used around thermocouple to add a small 15 mV bias to the negative lead of the thermocouple. This has been done to ensure that the voltage at both ADC terminals is positive at the most negative temperature. For example, at -270 °C the thermocouple gives an output voltage of -6.458 mV and by having the 15 mV bias voltage, we ensure PSoC pins see a positive voltage.

Offset Cancellation

K-type thermocouple has a sensitivity of around 40 µV/°C. Even a 40 µV offset results in 1 °C temperature error. Hence it is important to eliminate offset. Offset cancellation is done by correlated double sampling (CDS). CDS is a technique where the offset is measured and subtracted after every voltage reading. CDS also removes offset, offset drift and reduces low frequency noise but also reduces the ADC sample rate by 50 percent. Offset can be measured in a number of ways. See AN66444 - PSoC® 3 and PSoC 5 Correlated Double Sampling for details on different ways to measure offset. Offset in this case is measured using ADC channel 2 (The three ADC channels are 0, 1, and 2).

Filtering the Thermocouple Output

The thermocouple output is filtered using a software IIR filter to reduce the noise and improve the noise free temperature resolution. The Temperature Resolution section in page 9 explains the ADC configuration and the filter used in the project in detail.

Figure 9(a). Thermocouple Measurement Circuit

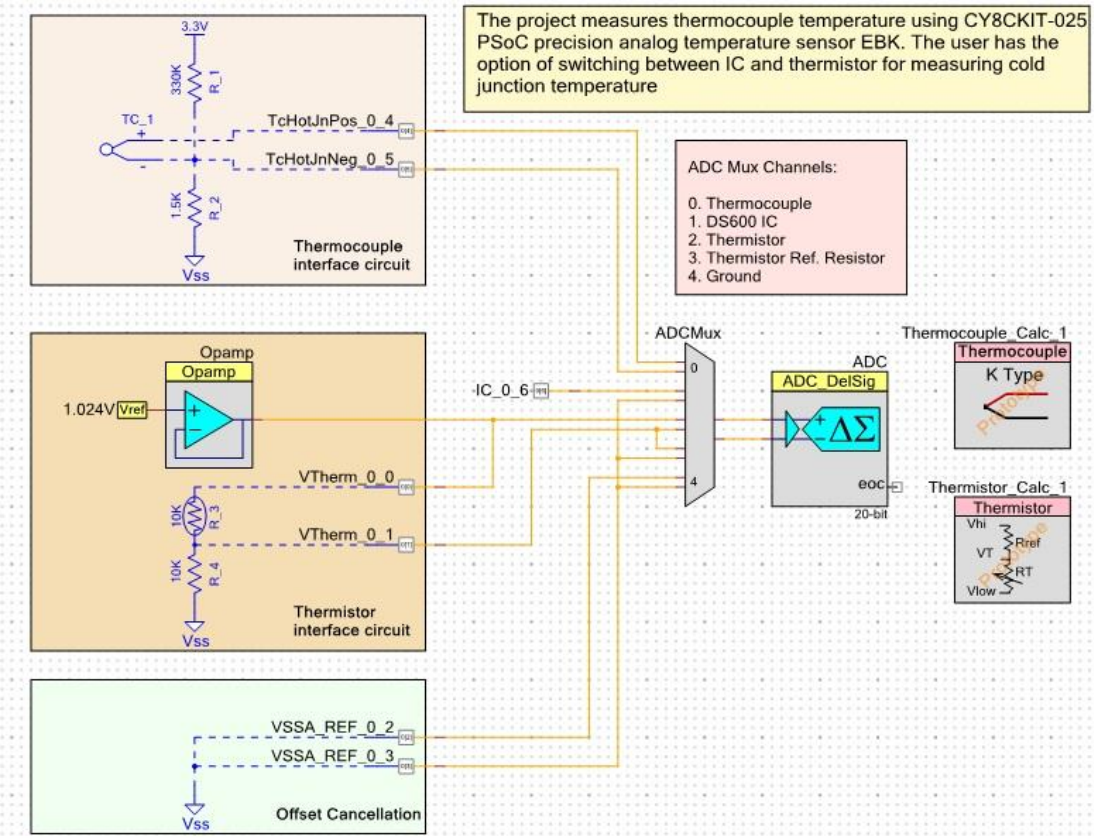
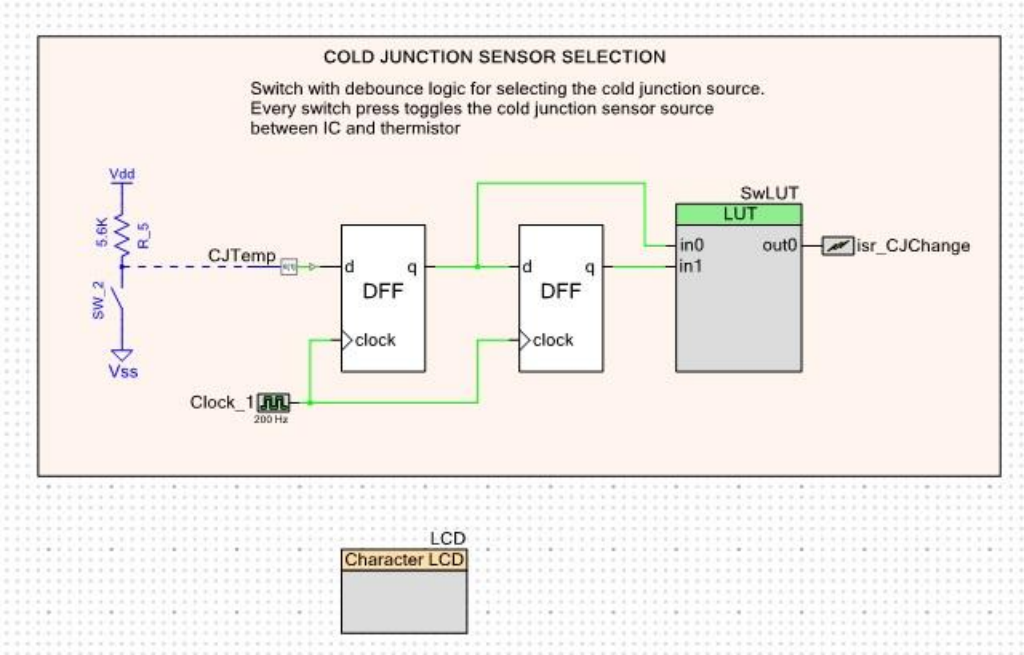


Figure 9(b). Thermocouple Measurement Circuit



Thermocouple measurement circuit (Figure 9) is used to measure thermocouple voltage (V_{TC}) and cold junction temperature (T_{ref}). The APIs provided by the thermocouple component are used to obtain cold junction compensation voltage (V_{ref}) and the actual temperature. The details of the thermocouple component are discussed in the following section.

Thermocouple - Voltage to Temperature Conversion

The steps to compute the thermocouple temperature involves conversion of cold junction temperature to equivalent cold junction compensation voltage and converting the thermo-emf to temperature.

NIST provides both polynomial coefficients and tables for voltage to temperature conversion and vice versa.

LUT versus Polynomial

For the same accuracy, using lookup table (LUT) is both memory and computation intensive compared to using polynomials. Table 3 shows a comparison between the accuracy and number of CPU cycles (apprx) taken for computing temperature using polynomial (ninth order) and a look up table (80 element) for a K-Type thermocouple.

Table 3. Accuracy versus No. of CPU Cycles

	Accuracy	No of CPU cycles
80 element LUT + piecewise linear approximation	<0.2 °C	8000
Polynomial	<0.07 °C	5000

One polynomial does not fit the whole temperature range. Multiple polynomials should be used in different temperature ranges to get good accuracy. A PSoC Creator component is provided to you for simplifying these conversions.

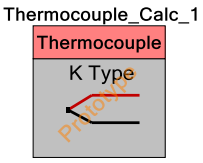
Thermocouple Component

The thermocouple component provided with this application note simplifies voltage to temperature conversion and vice versa by providing the two APIs given below.

```
float Thermocouple_1_mVoltToTemp(float voltage)
float Thermocouple_1_ColdJunctionTempTomVolt(float
temperature)
```

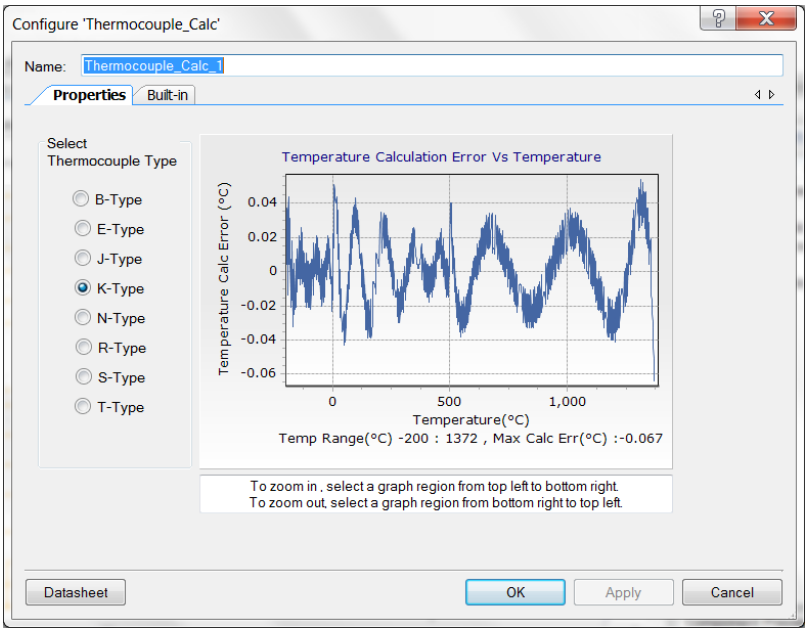
The thermocouple component symbol is shown in Figure 10.

Figure 10. Thermocouple Component Symbol



The component configuration dialog, used to enter the thermocouple type and user parameters is shown in Figure 11.

Figure 11. Thermocouple Component Configuration Dialog



Choose the thermocouple type and the configuration dialog will display a graph showing temperature calculation error versus temperature. This is the error due to polynomial approximation and floating point computation. The maximum calculation error in the whole temperature range is also displayed. Figure 11 shows that the maximum temperature calculation error caused by the component for K-type thermocouple is -0.067 °C.

On building the project, the component generates the following APIs. It is assumed that the thermocouple is named Thermocouple_1.

```
float Thermocouple_1_mVoltToTemp(float
voltage)
float Thermocouple_1_ColdJunctionTempTomVolt
(float temperature)
```

After measuring the cold junction temperature, use the `Thermocouple_1_ColdJunctionTempTomVolt()` API to obtain the cold junction compensation voltage. This API takes cold junction temperature as input and returns cold junction compensation voltage.

Once you have the final thermo-emf (V_{TC} + V_{ref}), use the `Thermocouple_1_mVoltToTemp()` API to obtain temperature. This API takes thermo-emf as input and returns temperature.

The following code snippet shows how the two APIs are used:

```
void main()
{
float coldJnTemp, tcColdJnmVolt,
tcHotJnmVolt, tcmVolt, tcTemp ;
    /* Measure cold junction temperature.
The function MeasureColdJnSensorTemp()
    returns cold junction temperature */
    coldJnTemp = MeasureColdJnSensorTemp();

    /* ColdJunctionTempTomVolt() API is used
to convert temp to millivolts */

tcColdJnmVolt = Thermocouple_Calc_1_ColdJunc
tionTempTomVolt(coldJnTemp);

    /* FindHotJnmVolt() API finds the hot
junction voltage in millivolts */
    tcHotJnmVolt = FindHotJnmVolt();

    /* Add cold junction compensation
voltage to hot junction voltage */
    tcmVolt = tcColdJnmVolt + tcHotJnmVolt;

    /* mVoltToTemp() API is used for
converting thermo emf to temperature */

tcTemp = Thermocouple_Calc_1_mVoltToTemp(tcm
Volt);
}
```

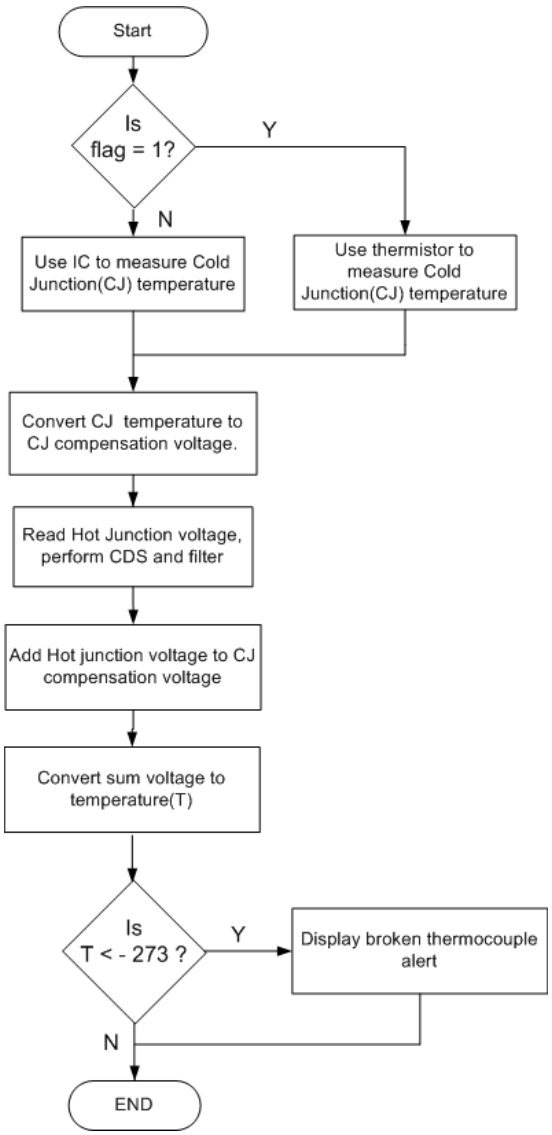
The datasheet associated with the component gives more details on the thermocouple component implementation, the number of CPU cycles taken by the APIs and the order of the polynomial used in different temperature ranges.

Temperature error depends on several factors apart from voltage to temperature conversion error. The thermocouple customizer shows only the error due to voltage to temperature conversion and does not take the other errors into account. If your required accuracy is 2 °C, make sure the thermo-emf to temperature conversion error is less than one-tenth of the total error budget to accommodate other errors. The other errors are discussed in Temperature Accuracy section.

The thermocouple component is not a standard component as a part of PSoC Creator. For use in the PSoC Creator project, the component should be added as a dependency to the project. See Appendix for details on how to add the component to your project.

Firmware Flow

The firmware flow is given in the flow chart below. An interrupt is triggered when a switch is pressed (see Figure 9). A flag is toggled in the interrupt service routine (ISR) changing the cold junction temperature source between IC and thermistor. The APIs generated by the thermocouple component is used for converting thermocouple voltage to temperature and vice versa.



Broken Thermocouple

If the thermocouple wire breaks, the small negative bias we apply to the negative terminal of the thermocouple connector (see Figure 9) takes the ADC voltage and hence the thermocouple temperature to a large negative value. Checking the thermocouple output for a large negative value (<-273 °C) helps us to detect a broken thermocouple connection. The project associated with the application note detects a broken thermocouple connection and displays a broken alert in the LCD if the thermocouple is broken.

Testing the Project

1. Plug the CY8CKIT-025 PSoC precision analog temperature sensor EBK to PORT E of PSoC 3 Development Kit (DVK).
2. Build the attached project and program PSoC device
3. The LCD displays thermocouple temperature and cold junction temperature
4. Press SW2 on the DVK to toggle the cold junction temperature source between thermistor and IC

Multiple Thermocouples

You can use multiple thermocouples in your design. The number of thermocouples that you can use is limited only by the number of GPIOs (input/output terminals) available in PSoC.

If you are using multiple thermocouples of the same type, one thermocouple component will be sufficient and the APIs generated by that component can be reused for all the thermocouples.

If you are using multiple thermocouples in your project and have more than one thermocouple type, one component per type should be used in your project. For instance, if your project has three K-type thermocouples, two J-type thermocouples, and one T-type thermocouple, you need to use three thermocouple components, one for each type (J, K, and T).

Performance Measures

Thermocouple resolution and accuracy, and test results of the measurement are discussed in the following section. A temperature test has also been performed to validate the design and the test results are provided at the end of this section.

Temperature Resolution

The temperature resolution depends on the ADC resolution and the temperature range.

Let us calculate the resolution required for measuring -270 °C to 1370 °C with a 0.1 °C resolution using a K-type thermocouple. The first step is to translate the temperature resolution to voltage resolution. Temperature range -270 °C to 1370 °C corresponds to voltage range -6.45 mV to 54.88 mV. Temperature resolution of 0.1 °C corresponds to voltage resolution of 4 µV (sensitivity = 40 µV/°C).

Ideally the 20-bit PSoC 3 or PSoC 5 delta sigma ADC in ±1.024 V range should give 1.95 µV resolution, but the effective resolution of any high resolution convertor will be lower than the actual resolution. Configuring the ADC in ±1.024 V with internal reference, we get an Effective Number of Bits (ENoB) of 18-bits (voltage resolution of 7.8 µV). Because the thermocouple output voltage range is -6.45 mV to 54.88 mV, we can use the ±0.064 V range of the ADC. This range has a better noise performance compared

to ±1.024 V mode. (See Delta-sigma ADC RMS Noise in Counts versus Input Range and Sample Rate table under delta sigma ADC specifications in PSoC 3 Datasheet).

Note that the ENOB mentioned here assumes that board has been laid out properly. AN57821 – PSoC 3 and PSoC 5 Mixed Signal Circuit Board Layout Considerations discusses the board design considerations for achieving a better ENoB.

A small software based IIR filter is added to the code in the attached project to achieve ENoB of 19-bits. See AN2099 – Single Pole IIR filter for a detailed description of software based IIR filter.

If the temperature resolution required is greater than or equal to 1 °C, then a 16-bit ADC would suffice for the application. A 16-bit ADC provides a much higher throughput.

Temperature Accuracy

Thermocouple temperature accuracy can be calculated by calculating the effect of the individual errors that occur during measurement and conversion. To understand the different errors consider the equation used, thermocouple component API, to obtain the final temperature:

T_h = `Thermocouple_1_mVoltToTemp(`V_{TC}+V_{ref}`)`

Equation 1

where T_h is the thermocouple temperature; V_{TC} is the thermocouple voltage measured; V_{ref} is the cold junction compensation voltage; `Thermocouple_1_mVoltToTemp()` function performs the voltage to temperature conversion.

V_{ref} = `Thermocouple_1_ColdJunctionTempTomVolt(`T_{ref}`)`

Equation 2

where T_{ref} is the cold junction temperature.

A temperature error can result from one of the factors:

1. Measured thermocouple voltage, V_{TC}.
2. Cold junction compensation voltage, V_{ref}
3. Voltage to temperature conversion

Each of the factors is discussed in detail in the following section:

Thermocouple Voltage Measurement Error

Thermocouple voltage measurement error is primarily due to ADC offset error, gain error, and INL error.

Offset Error

The ADC offset error leads to incorrect thermocouple voltage measurement. Offset cancellation is done by correlated double sampling (CDS) as explained in Offset Cancellation in page 5.

Gain Error

The ADC gain error also leads to incorrect thermocouple voltage measurement. PSoC 3 or PSoC 5 delta sigma ADC is factory calibrated for gain error of ±0.2% in a subset of ADC configurations. This 0.2% gain error also includes the ADC reference error. ±0.064 V range (the ADC range used in the project) in 20-bit mode is among the calibrated ranges. AN68403 – Analog signal chain calibration lists the factory calibrated ADC configurations and explains how to perform calibration on uncalibrated ADC ranges.

0.2% gain error results in 0.2% error in the measured voltage. The error due to 0.2% gain error at various temperatures is shown in Table 4.

If the ambient temperature (temperature of the PSoC 3 device) is different from 25 °C, ADC gain drift causes additional error. PSoC 3 or PSoC 5 delta sigma ADC has a gain drift of 50 ppm/°C. It will be 2000 ppm or 0.2% for an ambient temperature of 65 °C or -15 °C. Table 4 again shows the error due to 0.2% gain drift.

Table 4. Temperature Error Caused by Gain Error / Drift

Temperature (°C)	Error due to 0.2% Gain Error / Gain Drift (°C)
-250	-3
-100	0.2
0	0
100	0.2
250	0.5
500	0.95
1000	1.9
1300	3

ADC INL

The INL of an ADC at any point is the difference between the ideal ADC count and the actual ADC count at that point after gain and offset corrections have been done. The datasheet specifies the maximum INL of all points across Process, Voltage, and Temperature (PVT). PSoC 3 ADC has an INL of +/-32 LSb in ±1.024 V mode; 32LSb corresponds to 64 µV for 20-bit resolution and ±1.024 V range. The analog value 64 µV remains the same in ±0.064 V range. This error of 64 µV corresponds to temperature error of 1.5 °C (for temperatures > -100 °C).

Note that these are worst-case errors. We have used the worst-case INL across PVT and used it for error calculation. This is a pessimistic approach and is done to indicate the worst-case limits due to INL. Practically, the error due to INL will be much lower. For a typical INL of around 4 LSb, the temperature error due to INL will be <0.2 °C (for temperatures > -100 °C).

Cold Junction Compensation Voltage Error

Error in cold junction compensation voltage is due to the error in the measured cold junction temperature or error in temperature to voltage conversion. The

`Thermocouple_1_ColdJunctionTempTomVolt()` function ensures almost zero error because of temperature to voltage conversion.

The cold junction temperature error depends on the sensor used for cold junction compensation. A 1 °C error in cold junction temperature causes approximately the same error in the measured hot junction temperature for temperatures > -100 °C (1 ± 0.2 °C).

Voltage to Temperature Conversion Error

Voltage to temperature conversion error is due to the polynomial approximation error or the LUT approximation error. `Thermocouple_1_mVoltToTemp` () API ensures that this error is less than 0.05 °C in most cases.

All K-type thermocouples do not follow the NIST thermo-emf versus temperature data accurately. Thermocouple tolerances provided by two major standards IEC EN 60584-2 and ASTM E230 are given in Table 1.

Table 5 gives the temperature error due to various components at 500 °C and 100 °C. As seen from the table, thermocouple tolerance is the biggest source of error.

Table 5. List of all Possible Errors in Thermocouple Temperature Measurement at 500 °C and 100 °C

Error Source	Maximum Error value at 500°C (K-Type)	Maximum Error Value at 100 °C (K-type)
Offset Error/drift	0 °C	0 °C
Gain Error	0.95 °C	0.2 °C
Gain drift (Ambient temp = 25 °C)	0 °C	0 °C
Gain drift (Ambient temp = 65 °C)	0.95 °C	0.2 °C
ADC INL*	1.5 °C	1.5 °C
Error due to cold junction temperature error	Same as cold junction temperature error	Same as cold junction temperature error
Error due to Thermocouple tolerance (Special)	2 °C	1.1 °C
Voltage to temperature conversion error	0.05 °C	0.05 °C

Note(*) The ADC INL error indicates the worst case limit. The actual temperature error will be much lower than 1.5 °C depending on the INL at that point. For a typical INL of around 4 LSb, the temperature error due to INL will be <0.2 °C.

Temperature Test

A temperature test has performed on the attached project at different temperatures and the temperature measured by thermocouple is compared to the temperature displayed by a standard thermometer (accuracy ±0.5 °C). A precision temperature forcing system (air flow) is used to set various temperatures. The accuracy results are shown in Table 6. Column 1 shows the temperature set on the temperature forcing system, column 2 shows the temperature displayed by the standard thermometer, column 3 shows the thermocouple temperature, and column 4 shows the temperature error.

Table 6. Accuracy Test

Temperature Source (°C)	Standard Thermometer (°C)	Thermocouple (°C)	Temperature Error (°C)
-40	-39.5	-39.7	0.2
-20	-19.5	-19.5	0
0	-0.2	-0.7	0.5
25	24.8	25.2	-0.4
40	39.4	39.6	-0.2
60	59.1	59	0.1
80	78.9	78.7	0.2
100	98.8	98.2	0.6
120	118.8	118.2	0.6

As seen from Table 6, the temperature error is <1 °C in -40 °C to 120 °C

Thermocouple Signal Chain Test

Temperature test includes the temperature error caused by the thermocouple tolerance too. As shown in Table 5, the thermocouple tolerance is the biggest error of all. We can test the accuracy of the signal chain by feeding a millivolt source input to the thermocouple connectors and noting down the resultant thermocouple temperature shown by PSoC. The mV source voltage is then measured with a multimeter and the expected temperature is calculated from the measured voltage using NIST tables. The cold junction temperature is forced to 0 °C while performing this test. Table 7 shows the signal chain accuracy results for a sample board.

Table 7. Signal Chain Accuracy Results

Simulated Thermo-emf (mV)	Expected Temperature (°C)	Obtained Temperature (°C)	Error (°C)
-4.695	-141	-141.1	0.1
-3.666	-103.7	-103.7	0
-2.575	-69.6	-69.5	-0.1
-1.741	-45.9	-45.7	-0.2
-0.654	-16.8	-16.7	-0.1
0	0	0	0
0.666	16.7	16.7	0
1.754	43.5	43.4	0.1
2.58	63.5	63.5	0
3.663	89.5	89.4	0.1
4.712	114.9	114.9	0
9.582	235.9	235.8	0.1
19.58	475	4.9	0.1
28.76	691.2	691.0	0.2
39.54	955.7	955.4	0.3
49.816	1227	1226.6	0.4

Conclusion

Thermocouples are the sensors of choice in industrial environment and for measuring temperatures >850 °C. Thermocouples require high-resolution ADC (or low noise amplifiers along with a low resolution ADC) and require another temperature sensor for measuring cold junction temperature. PSoC 3 or PSoC 5 delta sigma ADC and the thermocouple component make it easy to measure thermocouple temperature

About the Author

Name:	Praveen Sekar
Title:	Applications Engineer
Background:	Praveen holds a Bachelors degree in Electronics and Communication from the College of Engineering, Guindy, Chennai. He focuses on analog modules in PSoC.
Contact:	pfz@cypress.com

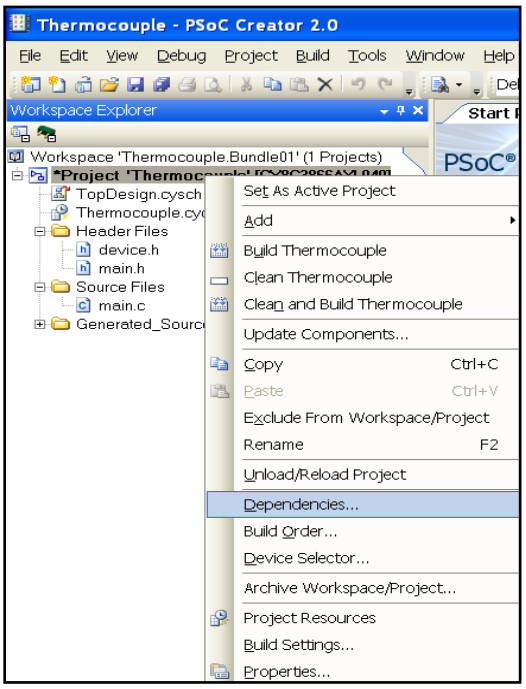
Appendix

How to add thermocouple component to your project:

The Thermocouple_Calc component can be found in the *Thermocouple_Calc.cylib* folder in the attached project. If you want to use thermocouple calculation component in the project following steps need to be followed:

1. Add the thermocouple component as a dependency to your project. For that right click on the project, go to dependencies (See Figure 12). Doing this opens the dependencies dialog.

   Figure 12. Opening Dependencies Dialog



2. In the Dependencies Dialog (See Figure 13(a)), Go to User dependencies section and select the New Entry (folder symbol) and browse to the component folder Thermocouple_Calc.cylib and select Thermocuple_Calc.cyprj (See Figure 13(b)).

Figure 13(a). Dependencies Dialog

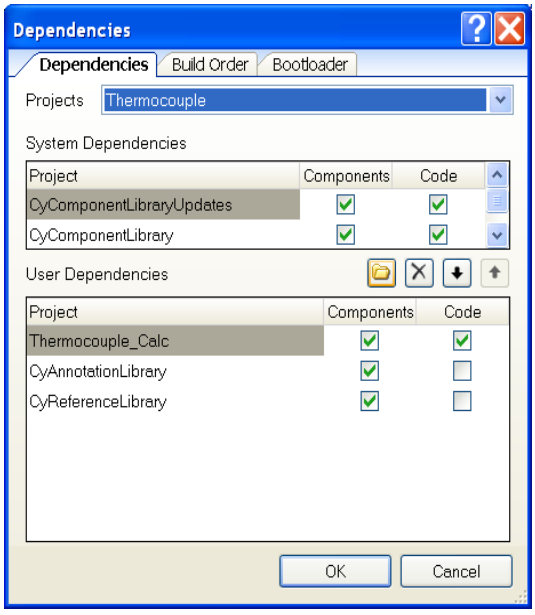
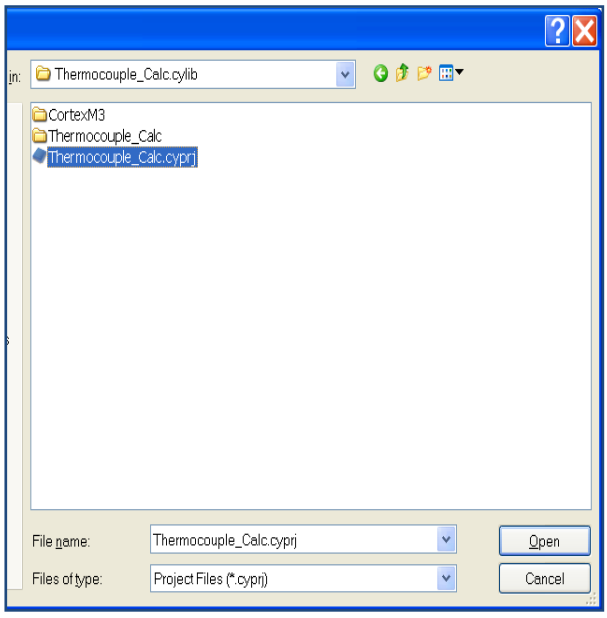
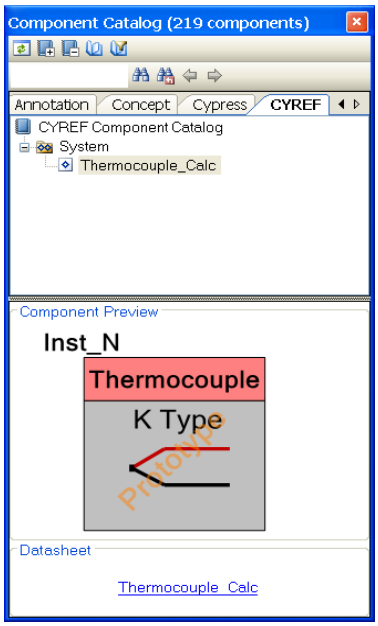


Figure 13(b). Selecting Thermocouple_Calc.cyprj



3. Now the component will appear in the CYREF category (see Figure 14) in the component catalog:

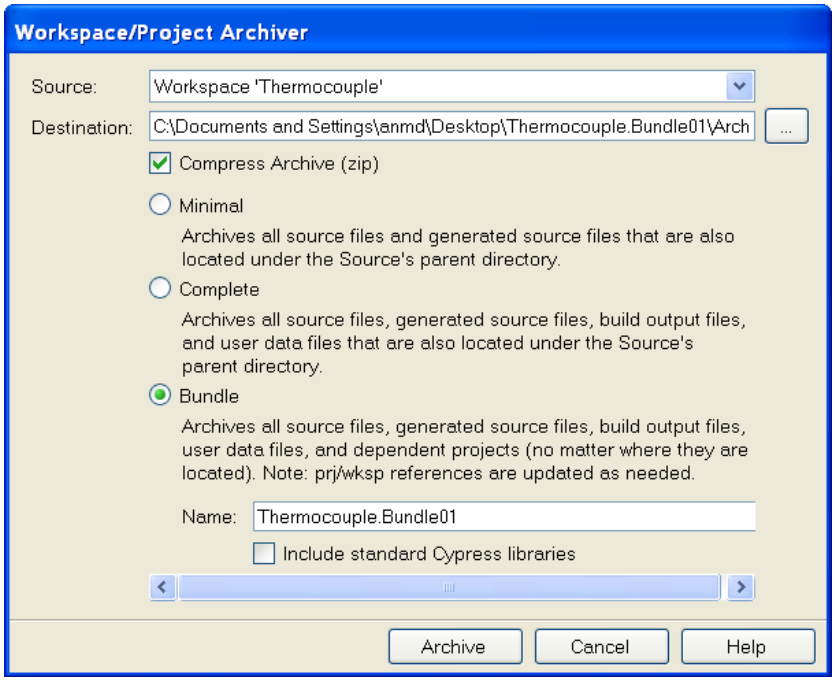
Figure 14. Thermocouple Component in the Component Catalog



4. Drag and drop the component to use it in your design

5. To archive the project, use bundle option (available in Project -> Archive Workspace/Project… menu item) provided by the PSoC Creator, so that it bundles the custom component along with other project files. The Workspace/Project Archiver dialog is shown in Figure 15.

Figure 15. Workspace/Project Archiver Dialog



Document History

Document Title: PSoC® 3 / PSoC 5 – Temperature Measurement with Thermocouples – AN75511

Document Number: 001-75511

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3571217	PFZ	04/03/2012	New application note

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 cypress.com/go/plc
Memory	cypress.com/go/memory
Optical Navigation Sensors	cypress.com/go/ons
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 5

Cypress Developer Community

Community | Forums | Blogs | Video | Training

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip" and PSoC Designer are trademarks of Cypress Semiconductor Corp. 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. 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.