

Crocus Magnetic Sensor

Temperature Compensation

Relevant Crocus Devices

The concepts and examples in this application note are applicable to all of the following Crocus devices:

CTSR206V-IQ2, CTSR209V-IQ2, CTSR212V-IQ2,
 CTSR215V-IQ2, CTSR218V-IQ2, CTSR222V-IQ2

Introduction

Sensor devices like all semiconductor devices show change in performance over temperature. In the case of MLU sensor the undesired change in the output of the sensor (due to a change in the temperature) is usually linear, but it is still a challenge and must be separated from the desired signal if one wants to use the sensor over a wide temperature range.

The Crocus CTSR2xxV series is a family of magnetic sensors designed for sensing low magnetic field. The sensors are very good at sensing very low magnetic field. In fact they are 10 - 15 times more sensitive than other magnetic sensors. However, they show variation of parameters over temperature, thus temperature compensation is required.

There are many ways to minimize the effects of temperature on the output of a magnetic sensor. In this application note, we'll explore some common techniques to minimize the effects of temperature on the Crocus CTSR2xxV magnetic sensors.

Crocus Magnetic Sensor

The Crocus Magnetic Sensor is a four terminal device. The terminals comprise of two input terminals and two output terminals. Figure 1 shows the schematic symbol of the device and shows the four terminals: I_{IN} , I_{INGND} , V_B and V_{BGND} . The two output terminals of the sensor, V_B and V_{BGND} , connect to the sensor output resistor R_{OUT} that changes resistance while in the presence of a magnetic field. I_{IN} and I_{INGND} are the two input terminals that are used to bias the sensor in its linear region of operation. More about this later in this application note.

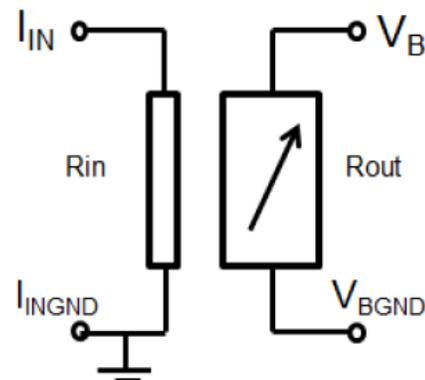


FIGURE 1

The transfer curve of the output resistor R_{OUT} versus the external magnetic field acting on the device can be seen in Figure 2.

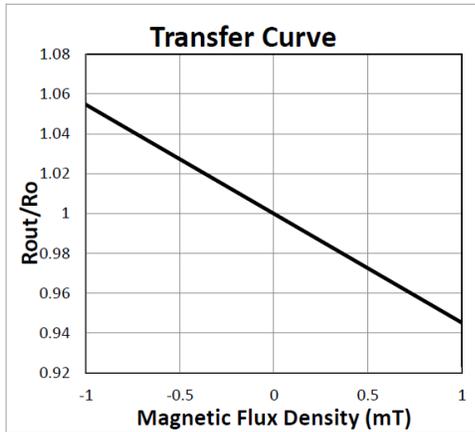


FIGURE 2

The resistance of the output resistor R_{OUT} changes while in the presence of a magnetic field. The same resistance also changes with temperature.

Regardless of the technique used to minimize the effects of the temperature on the output of a sensor, the very first thing that must be thoroughly understood is exactly how the temperature affects the sensor’s output. This information is usually obtained from the datasheet of the device. From the Crocus CTSR218V-IQ2 datasheet, we see that the temperature coefficient is negative. That is to say that the effect of a temperature increase is negative on the sensor’s resistor output R_{OUT} . As the temperature goes up, the output resistance R_{OUT} goes down. Figure 3 shows this relationship graphically. The units on the Y-axis, R_{OUT}/R_{25C} , are a bit cryptic. Let’s look at a quick example to clear up any confusion. Let’s use the CTSR218V-IQ2 datasheet for the example. From the datasheet:

$$R_o = 18K \text{ ohms @ } 25C$$

Let’s say that the temperature changes from 25C to 70C. Based on the graph in Figure 3, what is the final resistance of the sensor?

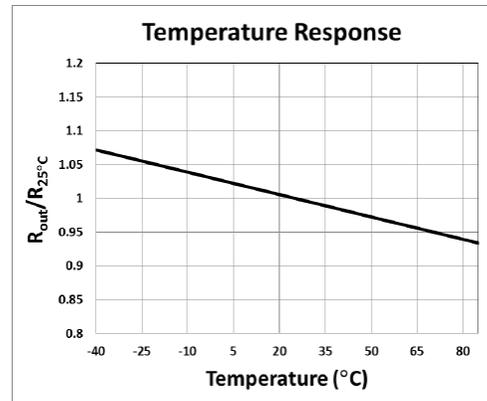


FIGURE 3

We can estimate that the R_{OUT}/R_{25C} is approximately 0.95. So the calculation to find the resistance of the sensor at 70C is very simple:

$$R_{(70C)} = R_{25C} * R_{OUT@70C}/R_{25} \text{ [ohms]}$$

$$= 18000 * 0.95 = 17.1K \text{ [ohms]}$$

Another more accurate way to calculate the resistance at a particular temperature is to use the Temperature Coefficient of Resistance from the datasheet.

Magnetic and Electrical Specifications

Parameter	Description	Min	Typical	Max	Unit
Voltage Supply (V _s)*		1.8	3	4.2	V
R _e	(at zero external field and 10 mA bias)	16	18	20	kΩ
Input Bias (I _{in})			10		mA
R _{in}			50		Ω
Sensitivity			50		T ⁻¹
Linearity Range			(+/-) 1		mT
Linearity Error	sweep (+/-) 1 mT		2.0		% FS
Hysteresis	3 sweeps (+/-) 1 mT		1.0		% FS
Max. Exposed Field			1		T
Operating Frequency		DC		500	MHz
Operating Temperature		-40		85	°C
Temperature Coefficient of Resistance	(at 10 mA input bias)		-0.1		% °C ⁻¹
Temperature Coefficient of Sensitivity			0.35		% °C ⁻¹
Package	QFN16		3x3		mm

*note: this product also works with voltage supply of 1.2V and 5V

FIGURE 4

Figure 4 was taken directly out of the datasheet for the CTSR218V-IQ2. We see that the temperature coefficient is $-0.1\%/C$. So let's quickly calculate the resistance of the sensor, R_{OUT} , at $70C$ using the temperature coefficient.

Total % change = $-0.1\%/C * \text{temp change } [C]$

$$= -0.1\%/C * (70 - 25)[C]$$

$$= -0.1\%/C * (45)[C]$$

$$= -4.5\%$$

Total R_{OUT} decrease = $18000 * 4.5\% = 810[\text{ohms}]$

$$R_{OUT} = 18000 - 810 = 17190 [\text{ohms}]$$

The R_{OUT} calculated using the temperature coefficient is a little different than the results of the calculation using the table in Figure 4. This is because of the rounding error when reading the graph from Fig. 3. But both results are very close. The more accurate way is to use the purely mathematical method with the temperature coefficient.

So now that we understand how the output of the sensor will change with temperature, let's see how we can mitigate the temperature effects.

Half-Bridge Temperature Compensation

One way to easily compensate for the temperature effects on the Crocus sensor is to arrange the sensing circuit in a half-bridge circuit as shown in Figure 5. The current source, I_1 , provides the bias current for the Crocus magnetic Sensor. Please see the application note AN103_GeneralCurrentSensing for more details about how to bias the sensor.

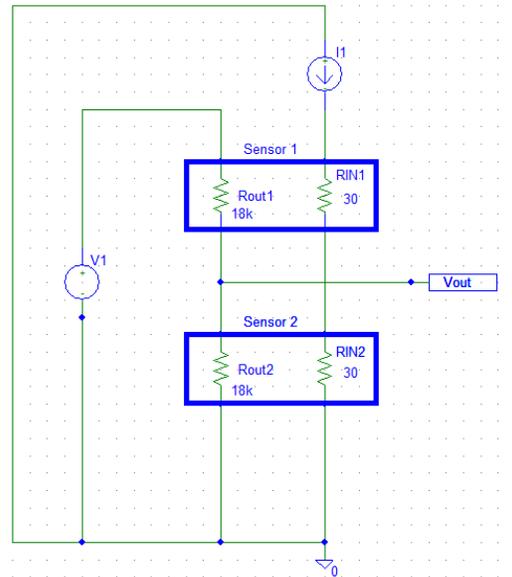


FIGURE 5

The voltage source, V_1 , is the voltage supply of the sensor half-bridge circuit. The theory of operation for the circuit is quite simple and is based on the voltage divider equation:

$$V_{out} = V_1 * (R_{OUT2} / (R_{OUT2} + R_{OUT1}))$$

As the resistance of Sensor 1, R_{OUT1} , increases due to temperature, Sensor 2, R_{OUT2} will also increase since both sensors have the same temperature coefficient as they are made of the same material in the circuit above. V_{out} will not change with the changes in the temperature. The reason for this is that both sensors will either go up or down in resistance with the temperature change. Based on the simple voltage divider equation, the V_{out} will not change if the R_{OUT2} and R_{OUT1} ratio of resistance remains constant.

Temperature Sensitive Resistor (TSR) Compensation

The concept behind using another temperature sensitive resistor (TSR) is to neutralize the change of sensor resistance change with

temperature. It sounds simple, but there are a few hurdles to get over to make it work. The biggest design issue to overcome is to find a temperature sensitive resistor that has the same temperature coefficient as the sensor. Actually we can essentially tune the effects of the temperature coefficient of the TSR to exactly fit our application by just adding a few passive components to the compensation circuit.

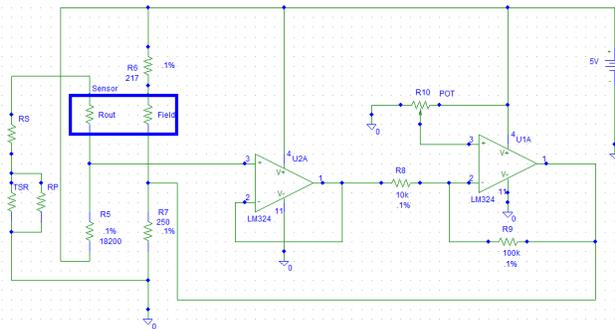


FIGURE 6

The circuit in Figure 6 is a single-sensor circuit with temperature compensation. The temperature compensation is accomplished by the use of a TSR. The particular TSR that we are using here has a positive temperature coefficient equal to 0.14%/C. As mentioned earlier, the TSR must be tuned to match the temperature coefficient of the sensor which is -0.1%/C. This is done by simply adding a series and parallel resistor shown in the Figure 6 as RS and RP respectively. The values of these components will depend on the resistive value of the TSR and its temperature coefficient. To simplify the calculation of RS and RP, we'll first develop an equation of the temperature coefficient of the Crocus sensor. We'll be able use this equation later when we model the circuit in PSpice. Using Excel it's very easy to find the linear equation of a set of any data. The first step is to generate a table of values of the

output resistance R_{OUT} of the sensor at different temperatures.

T [C]	Sensor R_{OUT}
0	18450
5	18360
10	18270
15	18180
20	18090
25	18000
30	17910
35	17820
40	17730
45	17640
50	17550
55	17460
60	17370
65	17280
70	17190
75	17100
80	17010
85	16920

TABLE 1

Table 1 was generated by using the -0.1%/C temperature coefficient of the sensor. The chart in Figure 7 shows a chart of this data. It also shows the trend-line equation generated by Excel. The sensor output resistance R_{OUT} as a function of temperature is:

$$R_{OUT}(T) = -18 * T[C] + 18450 \quad [\text{ohms}]$$

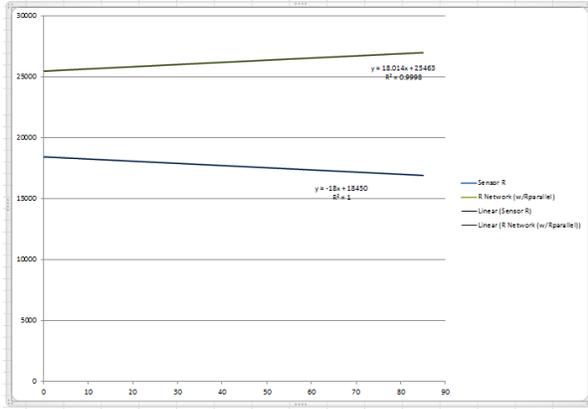


FIGURE 7

Also shown in Figure 7 is the plot of the resistive network comprised of the TSR in parallel with the resistor RP. There is no need for a series resistor in this case because the value of the TSR is high enough in resistive value to match the sensor resistance. Similarly to the way that we generated the plot for the sensor, the resistor network plot was also generated from calculated data using the temperature coefficient of the TSR which is 0.14%/C. The resistance of the TSR is 51000 ohms and the resistance of the parallel resistor is 52700 ohms. The value of the parallel resistor was determined by an iterative process to match the slope of the sensor’s temperature equation but with the opposite sign. As one can observe from the chart in Figure 7, the slopes of the two linear plots are equal but with opposite signs. To prove the effectiveness of the TSR in the circuit to compensate for the temperature effects, we can use the resistor equations to simulate this in PSpice. To recap, here are the two equations:

$$\text{SENSOR: } R_{\text{OUT}}(T) = -18 * T[\text{C}] + 18450 \quad [\text{ohms}]$$

$$\text{TSR: } R(T) = 18 * T[\text{C}] + 25463 \quad [\text{ohms}]$$

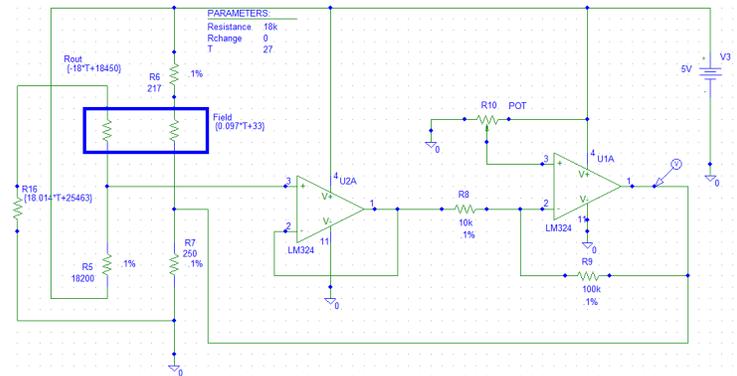


FIGURE 8

Figure 8 shows the temperature equations implemented in the circuit for the resistor values of the Crocus sensor and for the TSR.

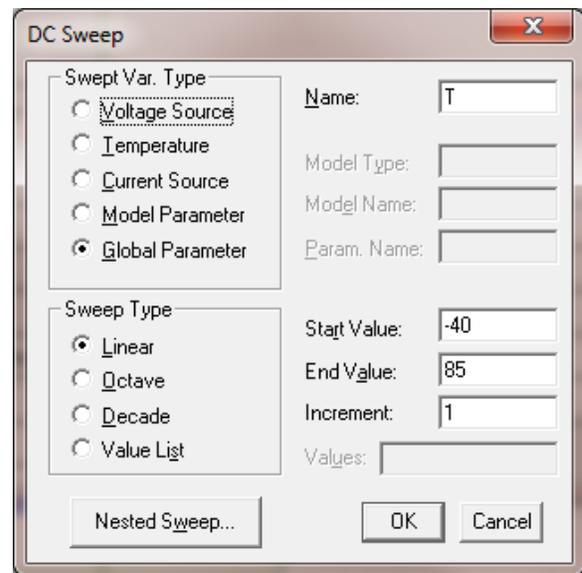


FIGURE 9

To test the effectiveness of the temperature compensation, we’ll sweep the parameter T over the temperature range from -40 C to 85 C.

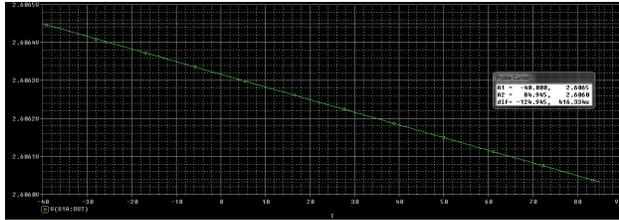


FIGURE 10

Figure 10 is the resulting plot of the output of the circuit in Figure 8 over the temperature -40 C to +85 C. The total change of the output voltage of the circuit is less than 425µV. This is close enough, but it could be tightened up even more by simply changing the parallel resistor value.

Temperature Compensation Using Software

To compensate for the temperature coefficient of the Crocus magnetic sensor in firmware, one simply measures the temperature with a temperature sensor and then subtracts the known effects of the temperature from the output resistance of the sensor. The effects of the temperature on the sensor are known because we have calculated the equation of the sensor output resistance as a function of temperature. Obviously the correction needs to be applied to the resulting analog to digital conversion into the micro, but for the sake ease of discussion here, we'll just apply the correction to the resistance calculation of the sensor. The application of the method to the analog to digital conversion is trivial. Remember that the resistance of the sensor as a function of temperature is:

$$R_{OUT}(T) = -18 * T[C] + 18450 \quad [\text{ohms}]$$

In firmware the effects of temperature can be removed by simply subtracting the error due to the temperature effects:

$$R_{ERROR} = R_{OUT} - (-18 * T + 18450) \quad [\text{ohms}]$$

$$R_{TC} = R_{OUT} + R_{ERROR} \quad [\text{ohms}]$$

T [C]	R _{OUT}	R _{ERROR}	R _{TC}
0	18450	-450	18000
5	18360	-360	18000
10	18270	-270	18000
15	18180	-180	18000
20	18090	-90	18000
25	18000	0	18000
30	17910	90	18000
35	17820	180	18000
40	17730	270	18000
45	17640	360	18000
50	17550	450	18000
55	17460	540	18000
60	17370	630	18000
65	17280	720	18000
70	17190	810	18000
75	17100	900	18000
80	17010	990	18000
85	16920	1080	18000

TABLE 2

Table 2 shows this calculation applied to the resistance values at different temperatures from 0 C to 85 C.

Summary

This application note describes three temperature compensation techniques that can be applied to neutralize the change of output sensor resistance over temperature. The Crocus magnetic sensor is especially well suited for these techniques because the temperature effects are well defined and repeatable.