SENSIRION
THE SENSOR COMPANY

SHTxx
Humidity & Temperature
Sensmitter

Application Note
Non-Linearity compensation

1 Introduction

The SHTxx devices show a small non-linearity of the humidity and temperature sensors.
This application note describes various ways to compensate it in the attached microcontroller.

2 Relative Humidity Non Linearity

If the formula given in the datasheet is too complex and therefore too computation intense, the following calculations may provide simplified alternatives.

Type of calculation	Inaccuracy due to non-linearity (10-90%RH) V4 Coefficients	Inaccuracy due to non-linearity (10-90%RH) V3 Coefficients	Complexity of calculation
Simple linearization	± 1.6% RH	± 2.2% RH	Simple (8bit subtract, right shift)
Two segment linearization	± 0.6% RH	± 0.8% RH	Quite simple (8bit multiplication, 16bit add/subtract)
Polynomial 2nd order	± 0.1% RH	± 0.1% RH	Floating point multiplications

New coefficients have been introduced and provide optimized accuracy for V4 sensors along the full measurement range. In this application note the old sets of parameters are marked with an asterisk. They have been proposed in earlier datasheets and are optimized for V3 sensors, but could still be applied to V4 version. V4 sensors can be identified by the alpha-numeric traceability code on the sensor cap instead of the 3 digit numeric code indicating V3 sensors.

2.1 Simple Humidity Linearization

The most basic conversion formula from sensor output to %RH is given in the following.
The coefficients e and f have to be chosen according to table 1 as set of (e, f) or (e*, f*) respectively.

$$RH_{linear} = e + f \bullet SO_{RH} \quad (\%RH)$$

	Version V4		Version V3	
SO_{RH}	**e**	**f**	**e***	**f***
12 bit	1.165	0.0314	0.5	0.0314
8 bit	1.165	0.5	0.5	0.5

Table 1: Coefficients for simple linearization

2.2 Two segment humidity linearization

For improved accuracy with minimal calculation complexity the following calculation is recommended.
The coefficients a and b have to be chosen according to table 3 for 8 bit SO_{RH} readout or table 4 for 12 bit SO_{RH} readout as set of (a, b) or (a*, b*) respectively.

8 bit SO_{RH}

$$RH_{linear} = \frac{(a + b \bullet SO_{RH})}{256} \text{ (\%RH)}$$

Validity Limits	Version V4		Version V3	
	a	b	a*	b*
$0 \le SO_{RH} \le 107$	-230	138	-512	143
$108 \le SO_{RH} \le 255$	1306	122	2893	111

Table 3: Two segment humidity linearization coefficients for 8 bit resolution

12 bit SO_{RH}

$$RH_{linear} = \frac{(a + b \bullet SO_{RH})}{4096} \text{ (\%RH)}$$

Validity Limits	Version V4		Version V3	
	a	b	a*	b*
$0 \le SO_{RH} \le 1712$	-3680	138	-8192	143
$1713 \le SO_{RH} \le 4096$	20896	122	46288	111

Table 4: Two segment humidity linearization coefficients for 12 bit resolution

See Appendix A for a 8 bit SO_{RH} sample code.

2.3 Polynomial 2nd order

This linearization method is according to the datasheet SHT1x and SHT7x. The coefficients c_X have to be chosen according to table 5 as set of c_X or c_X* respectively.

$$RH_{linear} = c_1 + c_2 \bullet SO_{RH} + c_3 \bullet SO_{RH}^2 \text{ (\%RH)}$$

SO_{RH}	Version V4			Version V3		
	c_1	c_2	c_3	c_1*	c_2*	c_3*
12 bit	-2.0468	0.0367	-1.5955E-6	-4.0000	0.0405	-2.8000E-6
8 bit	-2.0468	0.5872	-4.0845E-4	-4.0000	0.6480	-7.2000E-4

Table 5: Coefficients for humidity linearization with polynomial of second order

3 Temperature Non Linearity

Due to the inherent properties of the band gap PTAT (Proportional To Absolute Temperature) temperature sensor, the temperature output signal is not fully linear. The correction formula results in a correction of about -1°C at –40°C and -1 °C at 100°C compared to the linear formula. The parameter setting is the same for version V4 sensors as for version V3 sensors.

$$T = d_1 + d_2 \bullet SO_T + d_3 \bullet (SO_T - g)^2$$

VDD	d_1 [°C]	d_1 [°F]
5V	-40.1	-40.2
4V	-39.8	-39.6
3.5V	-39.7	-39.5
3V	-39.6	-39.3
2.5V	-39.4	-38.9

SO_T	d_2 [°C]	d_2 [°F]	d_3 [°C]	d_3 [°F]	g
14bit	0.01	0.018	-2e-8	-3.6e-8	7000
12bit	0.04	0.072	-3.2e-7	-5.76e-7	1750

Table 6: Coefficients for temperature linearization with polynomial of second order

4 Revision History

Date	Revision	Changes
October 20, 2001	0.9 (Preliminary)	
February 10, 2002	1.0	modified to final coefficients
February 15, 2003	1.1	added Temperature information
Oct. 17, 2003	1.2	Changed download link
May 10, 2004	1.3	Added Temperature non linearity information
October 19, 2004	1.31	Added 16 bit information in chapter 2.2
May 25, 2005	1.32	Changed company address
Oct 3, 2006	1.4	Sensirion Inc. address added
July 22, 2008	1.5	New coefficients for V4, completely revised

The latest version of this document and all application notes can be found at: www.sensirion.com/humidity

Headquarter and Sales Offices

Headquarter
SENSIRION AG
Laubisruetistr. 50
CH-8712 Staefa ZH
Switzerland
Phone: + 41 (0)44 306 40 00
Fax: + 41 (0)44 306 40 30
info@sensirion.com
http://www.sensirion.com/

Sales Office USA:
SENSIRION Inc.
2801 Townsgate Rd., Suite 240
Westlake Village, CA 91361
USA
Phone: 805 409 4900
Fax: 805 435 0467
michael.karst@sensirion.com
http://www.sensirion.com/

Sales Office Korea:
SENSIRION KOREA Co. Ltd.
#1414, Anyang Construction Tower B/D,
1112-1, Bisan-dong, Anyang-city
Gyeonggi-Province
South Korea
Phone: 031 440 9925~27
Fax: 031 440 9927
info@sensirion.co.kr
http://www.sensirion.co.kr

Sales Office Japan:
SENSIRION JAPAN Co. Ltd.
Postal Code: 108-0074
Shinagawa Station Bldg. 7F,
4-23-5, Takanawa, Minato-ku
Tokyo, Japan
Phone: 03 3444 4940
Fax: 03 3444 4939
info@sensirion.co.jp
http://www.sensirion.co.jp

Find your local representative at: http://www.sensirion.com/reps

Appendix A

Sample Code 8 bit SO_{RH}:

```
u16 result;          // 16Bit unsigned for the result
u08 sensor_out;      // 8Bit unsigned for the sensoroutput

sensor_out = readSensor8(); // read 8 bit humidity value from SHTxx

If ( sensor_out <= 107 )
{
   result = mult8Bit( 143, sensor_out );      // result = a * sensor_out
   result < 512 ? result = 512;               // check for underflow
   result = result – 512                      // result = result + b
}
else
{
   result = mult8Bit( 111, sensor_out );      // result = a * sensor_out
   result = result + 2893                     // result = result + b
   result > 25600 ? result = 25600;           // check for overflow (optional)
}

//8 MSB’s are 0-100%RH integers, 8 LSB’s are remainder

result = result >> 8  // result = result / 256
```