SHTxx
Humidity & Temperature
Sensmitter

Application Note
Compensation of RH non-Linearity

1 Introduction

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

2 Revision History

October 20, 2001	C2	URO	Revision 0.9 (Preliminary)
February 10, 2002	C2	URO	Revision 1.0 modified to final coefficients

3 Implementation

If the formula on page 2 of the SHT1x datasheet is to complex and therefore too computation intense, the follow calculations may provide simplified alternatives.

The examples are based on a 8 bit humidity readout. 12 bit readouts can be converted with similar formulas but with a slightly more complex calculation.

Type of calculation	Inaccuracy due to non-linearity (10-90%RH)	Complexity of calculation
linear	± 2.2% RH	Simple (8bit subtract, right shift)
2 * linear	± 0.8% RH	Quite simple (8bit multi, 16bit add/subtract)
Polynomial 2nd order	± 0.1% RH	Floating point multiplications

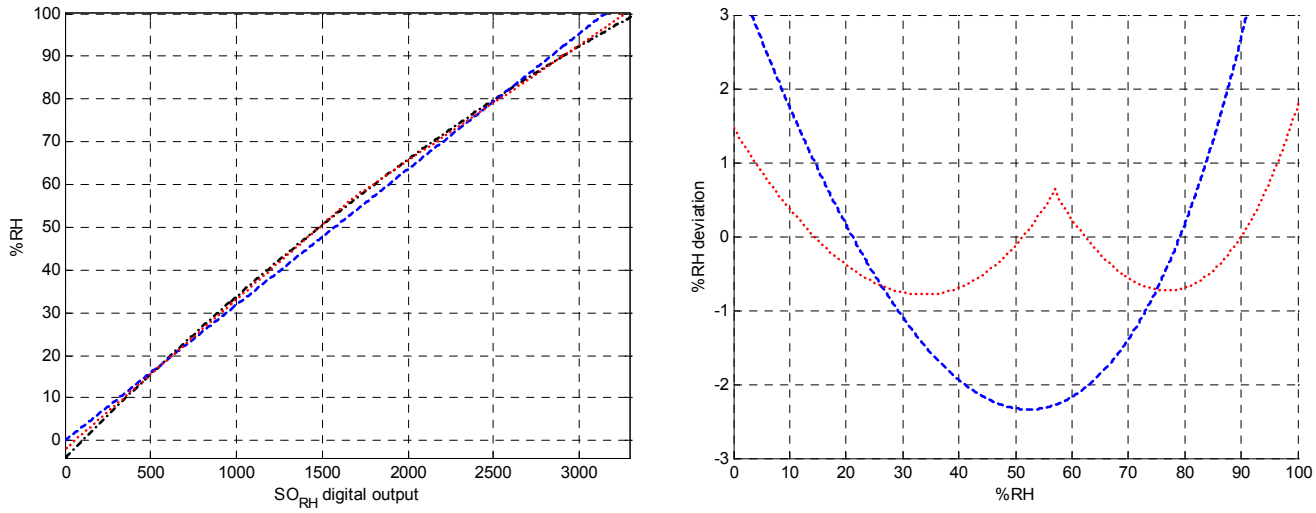

Figure 1 Inaccuracy due to non-linearity, original(from datasheet, black, dash-dotted), linear (blue, dashed) , 2* linear (red, dotted)

3.1 Linear

The most basic conversion formula from sensor output to %RH is:

$$RH_{simple} = c_1 + c_2 \bullet SO_{RH}$$

with $c_1 = 0.5$; $c_2 = 0.5$

3.2 2* linear

For improved accuracy with minimal calculation complexity the following calculation is recommended:
$RH_{real} = (a*SO + b) / 256$
Where SO denotes the 8 bit humidity sensor output signal.

Validity	a	b
0 ≤ SO ≤ 107	143	-512
108 ≤ SO ≤ 255	111	2893

With the above values the calculation can be done with a single 8 bit multiplication followed by a 16bit addition / subtraction.

Sample Code:

```
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
```

3.3 Polynomial 2nd order

Please consult the Datasheet for formula and coefficients.

Headquarters and Sales Office

SENSIRION AG
Eggbühlstr. 14
P.O. Box
CH-8052 Zürich
Switzerland

Phone: + 41 (0)1 306 40 00
Fax: + 41 (0)1 306 40 30
e-mail: info@sensirion.com
http://www.sensirion.com/