Application note

BMP280
Digital pressure sensor

Bosch Sensortec

BOSCH
Invented for life

BMP280: Application note (self-test)

Document revision	1.0
Document release date	29 May 2013
Document number	BST-BMP280-AN000-00
Notes	Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.

Index of Contents

1. Application note objective

This document provides an explanation to the self test code for the Bosch Sensortec BMP280. The code itself refers to the API (Application Programming Interface) of the sensor, which can be obtained from Bosch Sensortec and is also included in this release package.
The included API revision is 1.3. Please ensure that this is the latest version prior to using the self test files.

2. Self test flow

The self test starts by performing a soft reset of the device. After this, Chip-ID and trimming data are read and verified. Then temperature and pressure are measured and compared against customisable plausibility limits. A flow chart is given below.

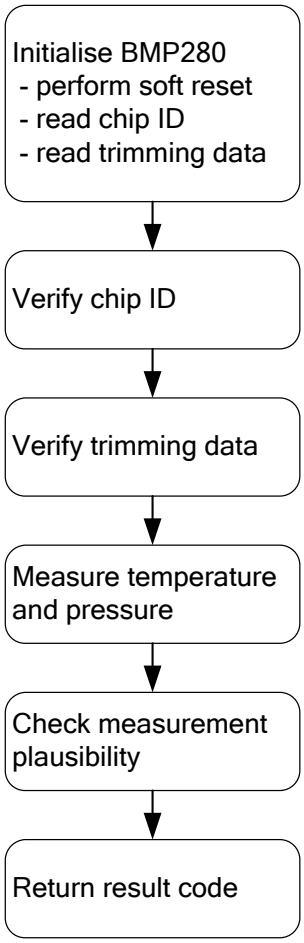

Figure 1: Self test flow chart

3. Function return codes

A list of the possible function return codes can be found below.

0	Sensor OK
10	Communication error or wrong device found
20	Trimming data out of bound
30	Temperature bond wire failure or MEMS defect
31	Pressure bond wire failure or MEMS defect
40	Implausible temperature (default limits: 0…40°C)
41	Implausible pressure (default limits: 900…1100 hPa)

Error testing is done in ascending error code sequence. This means that if e.g. a trimming data error is detected (code 20), the temperature plausibility (code 40) is not checked anymore. Instead, error code 20 is returned and no others tests are performed.

4. Usage

4.1 File and function pointer integration

- Include bmp280.c in your programming environment and add the path to the compiler.
- Include bmp280_selftest.c in your programming environment and add the path to the compiler.
- Modify the lines with read/write function pointer to match your system. Sample functions are given in chapter 6:

```
bmp280.bus_read   = BMP280_I2C_bus_read;  // must be defined by customer
bmp280.bus_write  = BMP280_I2C_bus_write; // must be defined by customer
bmp280.delay_msec = BMP280_delay_msec;    // must be defined by customer
```

- If necessary, adapt the measurement plausibility limits in bmp280_selftest.h. The default limits are 0...40°C for temperature and 900...1100 hPa for pressure measurement.
- If you are using I²C communication with the address 0x77 (SDO pin high), then change the BMP280.h line

```
#define BMP280_I2C_ADDRESS      BMP280_I2C_ADDRESS1
```

into

```
#define BMP280_I2C_ADDRESS      BMP280_I2C_ADDRESS2
```

4.2 Function call

Call the self test function using:

```
unsigned char testresult;
testresult = bmp280_selftest();
```

A test result of 0 indicates no error. The other return codes are detailed in chapter 3.

4.3 Test time and interface requirements

The self test uses a total wait time of 9 milliseconds. Of this, 2 milliseconds are used as wait time for soft reset and 7 milliseconds are used as wait time for conversion. The soft reset is performed in order to erase any possible old settings and could be omitted if the sensor is known to be in an untouched state after power on.

In the self test function, 4 write commands and 6 read commands are issued. In total, 4 bytes are written and 34 bytes are read. Assuming burst read is used, the following time duration can be expected for communication including overhead:

- 6.0 ms for I²C at 100 kHz
- 1.5 ms for I²C at 400 kHz
- 0.5 ms for SPI at 1 MHz

Assuming a 400 kHz I²C interface with burst reads, the total function run time therefore equals 10.5 milliseconds.

5. Function explanation

5.1 Communication test

This function attempts to read the Chip ID. If it is correct, a functioning communication is assumed. Note that the write function functionality is not explicitly tested.

5.2 Trimming data verification

All trimming parameters are tested against their respective bounds. If they exceed these bounds, a memory or programming error has occurred.
For future sensors, the permissible boundaries might change due to process variations. Such a change will be indicated by a marker. The self test code will recognise this marker and skip the data verification part of the code in order to avoid false error messages.
On change of the permissible boundaries, Bosch Sensortec will release a new self test code, which will again perform the trimming data verification considering the new boundaries.

5.3 Bond wire test

A pressure and temperature measurement is performed and uncompensated pressure and temperature values are read out. If the measurement results are clipped to the respective minimum or maximum ADC values, this is usually caused by defective bond wires. However, a defective sensing element could also cause this test to fail.
Please note that some combinations of bond wire or sensing element defects do not result in clipping of the measurement value and will therefore not be detected with this test. These cases can be detected by the plausibility test instead.

5.4 Measurement plausibility test

The pressure and temperature values read out previously are compensated using the read out compensation parameters. The compensated temperature and pressure is compared against plausibility limits set in bmp280_selftest.h, which must be set to match the customer production environment.

6. Sample read, write and delay functions

Below some samples read, write and delay functions are given. These are platform dependant and should only give an idea of how the functions could look.

```
signed char BMP280_I2C_bus_read(unsigned char device_addr, unsigned char
    reg_addr, unsigned char *reg_data, unsigned char cnt)
{
    int iError=0;
    unsigned char array[I2C_BUFFER_LEN];
    unsigned char stringpos;
    array[0] = reg_addr;
    iError = I2C_write_read_string(I2C0, device_addr, array, array, 1, cnt);
    for(stringpos=0;stringpos<cnt;stringpos++)
    {
       *(reg_data + stringpos) = array[stringpos];
    }
    return (signed char)iError;
}

signed char BMP280_I2C_bus_write(unsigned char device_addr, unsigned char
    reg_addr, unsigned char *reg_data, unsigned char cnt)
{
    int iError=0;
    unsigned char array[I2C_BUFFER_LEN];
    unsigned char stringpos;
    array[0] = reg_addr;
    for(stringpos=0;stringpos<cnt;stringpos++)
    {
       array[stringpos+1] = *(reg_data + stringpos);
    }
    iError = I2C_write_string(I2C0, device_addr, array, cnt+1);
    return (signed char)iError;
}

void BMP280_delay_msec(BMP280_U16_t msec) //delay in milliseconds
{
    BMP280_U32_t counter;
    for (counter = 0; counter/2000 < msec; counter++); // 2000 counts = 1 msec
}
```

7. Legal disclaimer

7.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or (e). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

7.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.

The resale and/or use of products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

7.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

8. Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
1.0		Document creation	2013-05-29

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 9
72770 Reutlingen / Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved | Printed in Germany
Specifications subject to change without notice
Document number: BST-BMP280-AN000-00
Revision_1.0_012016