# BMA180 Data sheet

BMA180 is a digital, triaxial acceleration sensor developed by Bosch Sensortec.

## Table

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order code(s)</td>
<td>0 273 141 053</td>
</tr>
<tr>
<td>Package type</td>
<td>12-pin LGA</td>
</tr>
<tr>
<td>Data sheet version</td>
<td>2.1</td>
</tr>
<tr>
<td>Document release date</td>
<td>10 December 2009</td>
</tr>
<tr>
<td>Document number</td>
<td>BST-BMA180-DS000-03</td>
</tr>
<tr>
<td>Notes</td>
<td>Specifications are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product’s appearance. Proprietary information — not for publication.</td>
</tr>
</tbody>
</table>
BMA180
Triaxial, ultra high performance digital accelerometer with switchable g-ranges and bandwidths and integrated thermometer

**Key information**
- Three-axis accelerometer with integrated temperature sensor
- Ultra high performance g-sensor (ultra-low noise, ultra-high accuracy) with 14 bit ADC operation
- Digital Interfaces: 4-wire SPI, I²C, interrupt pin
- High feature set with customer programmable g-ranges, filters, interrupts, power modes, enhanced features, in-field calibration possibility for customers and self-test capability
- Standard SMD package: LGA package, 3 x 3 mm² footprint, 0.9 mm height
- 256 bit EEPROM for calibration data and customer data
- Low power: typically 650 μA current even in 14 bit operation mode
- Very low-voltage operation: +1.62 V ... +3.6 V for VDD, +1.2 V ... +3.6 V for VDDIO
- Temperature range: -40 °C ... +85°C
- RoHS and halogen free compliant
- No external components needed besides 1 standard blocking capacitor for power supply
- Process based on automotive-proven Bosch Silicon Surface Micromachining

**Key performance (all typical values)**
- Resolution/noise:
  - Ultra low noise: 150 μg/(Hz)½ in low-noise mode
  - 0.25 mg ADC-resolution in 2g-mode
- Offset:
  - Offset at room temperature (+25 °C): 4 mg (incl. fine-offset tuning)
  - Very small Temperature Coefficient of Offset (TCO): 0.25 mg/K
- Sensitivity:
  - Very small sensitivity tolerances @ +25°C: ±1,5 %
  - Very small Temperature Coefficient of Sensitivity (TCS): ±0,01 %/K
Key features
- Customer programmable g-ranges (1 g, 1.5 g, 2 g, 3 g, 4 g, 8 g, 16 g)

- Customer programmable integrated digital filters (no external components):
  - 8 low-pass filters: 10, 20, 40, 75, 150, 300, 600, 1200 Hz (no dig. Filter)
  - 1 high-pass-filter: 1 Hz
  - 1 band-pass-filter: 0.2 – 300 Hz

- Customer programmable interrupt features:
  - wake-up (power management)
  - low-g detection
  - high-g-detection
  - tap sensing functionality
  - slope detection (any motion)

- Customer programmable power modes:
  - 2 main standard modes: low noise and low power mode (+ 2 intermediate modes)
  - sleep mode
  - wake-up mode

- Customer calibration possibility for:
  - Offset
  - Sensitivity
  - Temperature coefficient of offset (TCO)
  - Temperature coefficient of sensitivity (TCS)

- Enhanced features (customer programmable)
  - Switch for by-passing internal band-gap -> very low voltage operation with full performance (by using a high-performance external band-gap)
  - Sample skipping (reduction of µC load in interrupt driven applications by reduction of new-data interrupts)
  - 14 or 12 bit ADC-conversion (switch-able) for read-out acceleration
  - New-data interrupt to provide “synch-signal” to microcontroller
  - 2 selectable I²C addresses
  - Offset regulation (each channel) with subsequent interrupt generation
    - fine regulation (accuracy typically ±4 mg in 2g-mode)
    - in-field re-calibration possibility after regulation
  - Self-test capability
Typical applications

Tilt, motion and vibration sensing in
- Navigation devices (INS/Dead Reckoning)
- Robotics
- Gesture Recognition
- Pointing Devices
- E-Compass
- Cell phones
- Handhelds
- Computer peripherals
- Man-machine interfaces
- Virtual reality
- Gaming devices
- Digital cameras and digital camcorders
- High accuracy tilt sensing (level meter)

General description

BMA180 is a digital ultra-high-performance tri-axial low-g acceleration sensor for consumer market applications. It allows measurements of static as well as dynamic accelerations with very high accuracy. Due to its three perpendicular axes it gives the absolute orientation in a gravity field. As all other Bosch inertial sensors, it is a two-chip arrangement (here in a plastic package). An ASIC evaluates the output of a three-channel micromechanical acceleration-sensing element that works according to the differential capacitance principle. The underlying micromachining process has proven its capability in much more than 100 million Bosch accelerometers and gyroscopes so far.

The BMA180 provides a digital full 14 bit output signal via a 4-wire SPI or I²C interface. With an appropriate command the full measurement range can be chosen between 1 g and 16 g. A second-order Butterworth filter with switchable pole-frequencies between 10 Hz and 600 Hz is included to provide pre-conditioning of the measured acceleration signal. Typical noise level and quantization lead – in 2 g-mode - to a resolution of typically 0.5 mg and a typical accuracy of below 0.25° in an inclination sensing application, respectively. The current consumption is typically 650 µA at a supply voltage of 2.4 V in standard mode. Furthermore, the sensor can be switched into a very low-power mode where it informs the host system about an acceleration change via an interrupt pin. This feature can be used to wake-up the host system from a sleep mode.

The sensor also features self-test capability. It is activated via SPI/I²C command, which results in a physical deflection of the seismic mass in the sensing element due to an electrostatic force. Thus, it provides full testing of the complete signal evaluation path including the micro-machined sensor structure and the evaluation ASIC.

The sensor is available in a standard SMD LGA package with a footprint of 3x3 mm² and a height of 0.9 mm.
Table of Contents

1 SPECIFICATION........................................................................................................................................ 8

2 ABSOLUTE MAXIMUM RATINGS............................................................................................................. 12

3 BLOCK DIAGRAM ................................................................................................................................. 13

4 OPERATION MODES.................................................................................................................................. 14
  4.1 STANDARD OPERATIONAL MODES ............................................................................................ 14
  4.2 SLEEP MODE .............................................................................................................................. 14
    4.2.1 General information ............................................................................................................ 14
    4.2.2 Current consumption using duty cycling ............................................................................. 15
  4.3 WAKE-UP MODE .......................................................................................................................... 15
    4.3.1 General information .......................................................................................................... 15
    4.3.2 Current consumption in wake-up mode ............................................................................. 16

5 DATA CONVERSION................................................................................................................................... 18
  5.1 ACCELERATION DATA .................................................................................................................... 18
  5.2 TEMPERATURE MEASUREMENT ................................................................................................. 18

6 INTERNAL LOGIC FUNCTIONS............................................................................................................. 19
  6.1 LOW-G INTERRUPT LOGIC .......................................................................................................... 19
  6.2 HIGH-G LOGIC ............................................................................................................................ 20
  6.3 SLOPE DETECTION (OR ANY MOTION DETECTION) ...................................................................... 20
  6.4 TAP SENSING ............................................................................................................................... 20
  6.5 ALERT MODE ............................................................................................................................... 20

7 GLOBAL MEMORY MAP....................................................................................................................... 21
  7.1 GLOBAL MEMORY MAPPING: GENERAL INFORMATION .................................................. 22
  7.2 REGISTERS .................................................................................................................................... 23
  7.3 PROGRAMMING OF THE CALIBRATION PARAMETERS ............................................................ 23
  7.4 REGISTER ARITHMETIC ............................................................................................................... 24
  7.5 EEPROM..................................................................................................................................... 24
    7.5.1 General information .............................................................................................................. 24
    7.5.2 EEPROM Reading ............................................................................................................. 24
    7.5.3 EEPROM Writing .............................................................................................................. 24
    7.5.4 EEPROM protection ........................................................................................................... 25
    7.5.5 EEPROM content upon delivery (after production test) .................................................... 25
    7.5.6 ee_w_flag - EEPROM-written flag .................................................................................. 26
    7.5.7 EEPROM Endurance .......................................................................................................... 26
  7.6 IMAGE ............................................................................................................................................ 26
    7.6.1 Image writing ..................................................................................................................... 26
    7.6.2 Image reading ..................................................................................................................... 26
  7.7 GENERAL FUNCTIONAL SETTINGS ............................................................................................... 27
    7.7.1 range ..................................................................................................................................... 27
    7.7.2 bw ....................................................................................................................................... 27
    7.7.3 mode_config ....................................................................................................................... 28
    7.7.4 readout_12bit ...................................................................................................................... 29
    7.7.5 smp_skip (sample skipping) ............................................................................................... 29
    7.7.6 shadow_dis ......................................................................................................................... 30
7.11.3 low_th_int, high_th_int, slope_int_s, tapsens_int .................................................. 49
7.11.4 low_th_s, high_th_s, slope_s, tapsens_s, offset_st_s .................................................. 50
7.11.5 offset_st_s .................................................................................................................. 50
7.11.6 x_first_int, y_first_int, z_first_int ............................................................................ 50
7.11.7 Status bits for acceleration or slope sign .................................................................. 50
7.11.8 ee_write ..................................................................................................................... 50
7.12 DATA REGISTERS ......................................................................................................... 50
7.12.1 temp ........................................................................................................................... 50
7.12.2 acc_x, acc_y, acc_z ...................................................................................................... 50
7.12.3 al_version<3:0>, ml_version<3:0>, chip_id<2:0> ......................................................... 52
8 I²C AND SPI-INTERFACES ................................................................................................. 53
8.1 SPECIFICATION OF INTERFACE PARAMETERS .......................................................... 53
8.2 INTERFACE SELECTION ................................................................................................. 53
8.3 I²C INTERFACE ............................................................................................................... 54
8.3.1 I²C Timings .................................................................................................................... 54
8.3.2 Start and stop conditions ............................................................................................ 55
8.3.3 Bit transfer ................................................................................................................... 55
8.3.4 Acknowledge........................................................................................................................ 56
8.3.5 I²C protocol........................................................................................................................... 56
8.4 SPI INTERFACE (4-WIRE)........................................................................................................ 58
  8.4.1 SPI protocol ...................................................................................................................... 58
  8.4.2 SPI Timings....................................................................................................................... 59
9 PINNING ........................................................................................................................................ 60
  9.1 PIN CONFIGURATION (TOP VIEW)...................................................................................... 60
  9.2 PINNING: ELECTRICAL CONNECTIONS IN CASE OF SPI OR I²C OR NO µC).................... 60
  9.3 EXTERNAL COMPONENT CONNECTION DIAGRAM .......................................................... 62
10 PACKAGE ..................................................................................................................................... 64
  10.1 OUTLINE DIMENSIONS ...................................................................................................... 64
  10.2 ORIENTATION: POLARITY OF THE ACCELERATION OUTPUT ....................................... 64
  10.3 MARKING............................................................................................................................. 65
    10.3.1 Mass Production Samples ............................................................................................ 65
    10.3.2 Engineering samples ................................................................................................... 65
  10.4 LANDING PATTERN RECOMMENDATIONS ...................................................................... 66
  10.5 MOISTURE SENSITIVITY LEVEL AND SOLDERING ......................................................... 66
  10.6 RoHS COMPLIANCE............................................................................................................ 67
  10.7 HALOGEN CONTENT .......................................................................................................... 67
  10.8 NOTE ON INTERNAL PACKAGE STRUCTURES .................................................................... 67
  10.9 TAPE AND REEL .................................................................................................................. 67
  10.10 HANDLING INSTRUCTION ............................................................................................... 67
  10.11 FURTHER HANDLING, SOLDERING AND MOUNTING INSTRUCTIONS .......................... 67
11 LEGAL DISCLAIMER .................................................................................................................... 68
  11.1 ENGINEERING SAMPLES .................................................................................................... 68
  11.2 PRODUCT USE .................................................................................................................. 68
  11.3 APPLICATION EXAMPLES AND HINTS ........................................................................... 68
  11.4 LIMITING VALUES.............................................................................................................. 68
12 DOCUMENT HISTORY AND MODIFICATION .............................................................................. 69
1 Specification

Unless otherwise stated, given minimum, typical, maximum values are corresponding values over lifetime (incl. MSL1 preconditioning – 168h 85% rh, 85°C – before soldering) and full performance temperature/voltage range in the standard operation mode. Min/Max-values represent 3-sigma values. Typical values are not guaranteed.

In the present specification different LSB-values are mentioned; following values are valid:
- 1 LSB\textsubscript{ADC} ≈ 0.25 mg in 2 g range; it scales with range (e. g.: 1g-range -> 1 LSB\textsubscript{ADC} is 0.125 mg)
- 1 LSB\textsubscript{TEMP} = 0.5°C.

Table 1: Operating conditions (unless otherwise specified)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature</td>
<td>T\textsubscript{op}</td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>Temperature range for EEPROM writes.</td>
<td>T\textsubscript{ee_w}</td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>Supply voltage (internal bandgap)</td>
<td>VDD</td>
<td>2.0</td>
<td>2.4</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td>Supply voltage for digital interface (VDD\textsubscript{IO} \leq VDD; VDD should not be applied after VDD\textsubscript{IO})</td>
<td>VDD\textsubscript{IO}</td>
<td>1.2</td>
<td>2.4</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td>Allowed external regulated voltage, if internal band-gap is by-passed (dis_reg=1)</td>
<td>VDD\textsubscript{ext}\textsuperscript{1}</td>
<td>1.62</td>
<td>1.8</td>
<td>2.0</td>
<td>V</td>
</tr>
</tbody>
</table>

\textsuperscript{1} Internal regulators are by-passed by setting bit dis\_reg to “1b”. Customer has to apply stabilized voltage (PSRR\_DC > 60dB) to obtain good performance.
## Table 2: Specification

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITION</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operating Range</strong></td>
<td>gFS1g</td>
<td>Switchable via serial interface</td>
<td>±1.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS1.5g</td>
<td></td>
<td>±1.5</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS2g</td>
<td></td>
<td>±2.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS3g</td>
<td></td>
<td>±3.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS4g</td>
<td></td>
<td>±4.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS8g</td>
<td></td>
<td>±8.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>gFS16g</td>
<td></td>
<td>±16.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td><strong>Acceleration Ranges</strong></td>
<td>gFS16g</td>
<td>Switchable via serial interface</td>
<td>±16.0</td>
<td></td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal band-gap:</td>
<td>2.0</td>
<td>2.4</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full perform. in T-range: -40°C .. +85 °C; disreg=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>External band-gap:</td>
<td>1.62</td>
<td>1.8</td>
<td>2.0</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full perform. in T-range: -40°C .. +85 °C; disreg=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Supply Voltage</strong></td>
<td>VDD</td>
<td>Supply Voltage for digital interfaces VDDIO</td>
<td>VDDIO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDDIO ≤ VDD (timing: VDD should not be applied after VDDIO)</td>
<td>1.2</td>
<td>1.8</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supply Current in “Low power” mode I_DD</td>
<td>I_DD</td>
<td>650</td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supply Current in “Low-Noise” mode I_DD_LN</td>
<td>I_DD_LN</td>
<td>975</td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supply Current in Sleep mode I_DD_SL</td>
<td>I_DD_SL</td>
<td>Sleep mode/no serial interface transfer (T=25°C)</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operating Temperature T_OP</td>
<td>T_OP</td>
<td>-40</td>
<td>+85</td>
<td>°C</td>
</tr>
</tbody>
</table>
### Output Signal

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC resolution</td>
<td></td>
<td>14 bit mode</td>
<td>14</td>
<td>12</td>
<td>bit</td>
<td>bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12 bit mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity (calibration at factory in 2g-range)</td>
<td>$S_{1g}$</td>
<td>g-range: +/-1.0 g</td>
<td>8192</td>
<td>±2.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{1.5g}$</td>
<td>g-range: +/-1.5 g</td>
<td>5460</td>
<td>±2.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{2g}$</td>
<td>g-range: +/-2.0 g</td>
<td>4096</td>
<td>±1.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{3g}$</td>
<td>g-range: +/-3.0 g</td>
<td>2730</td>
<td>±2.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{4g}$</td>
<td>g-range: +/-4.0 g</td>
<td>2048</td>
<td>±2.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{8g}$</td>
<td>g-range: +/-8.0 g</td>
<td>1024</td>
<td>±2.5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{16g}$</td>
<td>g-range: +/-16 g</td>
<td>512</td>
<td>±3.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature Coefficient of Sensitivity</td>
<td>TCS</td>
<td>$T = 25 , ^\circ C, \quad V_{DD} = 2.4 , V$</td>
<td>±0.01</td>
<td></td>
<td>%/K</td>
<td></td>
</tr>
<tr>
<td>Zero-g Offset $^{2}$ exfactory</td>
<td>Off</td>
<td>$T = 25 , ^\circ C, \quad V_{DD} = 2.4 , V$, $g_{FS2g}$</td>
<td>-45</td>
<td>±15</td>
<td>+45</td>
<td>mg</td>
</tr>
<tr>
<td>Zero-g Offset $^{3}$ (fine tuning)</td>
<td>Off_fine</td>
<td>$T = 25 , ^\circ C, \quad V_{DD} = 2.4 , V$, $g_{FS2g}$</td>
<td>-20</td>
<td>±5</td>
<td>+20</td>
<td>mg</td>
</tr>
<tr>
<td>Zero-g Offset Temperature Drift</td>
<td>TCO</td>
<td>$T = -40^\circ C - 85^\circ C, \quad V_{DD} = 2.4 , V$, $g_{FS2g}$</td>
<td>±0.75</td>
<td></td>
<td>mg/K</td>
<td></td>
</tr>
<tr>
<td>DC Power supply rejection ratio</td>
<td>PSRR_DC</td>
<td>in ±2g range, $V_{DD} = 2.4 , V$ (dis_reg = 0)</td>
<td>10</td>
<td></td>
<td>LSB_ADC/V</td>
<td></td>
</tr>
<tr>
<td>AC Power supply rejection ratio</td>
<td>PSRR_AC</td>
<td>With 0.4V peak to peak AC signal on $V_{DD}$ at internal clock frequencies (dis_reg=0)</td>
<td>140</td>
<td></td>
<td>LSB_ADC/V</td>
<td></td>
</tr>
<tr>
<td>Noise density low power mode</td>
<td>N_LP</td>
<td>Low power mode @ 1200 Hz, 2g</td>
<td>200</td>
<td></td>
<td>µg/(Hz)^{\frac{1}{2}}</td>
<td></td>
</tr>
<tr>
<td>Noise density low noise mode</td>
<td>N_LN</td>
<td>Low noise mode @ 1200 Hz, 2g</td>
<td>150</td>
<td></td>
<td>µg/(Hz)^{\frac{1}{2}}</td>
<td></td>
</tr>
</tbody>
</table>

$^{2}$ Exfactory: no offset tuning, no soldering

$^{3}$ Over lifetime, after offset-fine-tuning in 0g-position
### BMA180 Data Sheet

**Digital, triaxial acceleration sensor**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>BW_10</td>
<td></td>
<td>10</td>
<td>20</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_20</td>
<td></td>
<td>20</td>
<td>40</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_40</td>
<td></td>
<td>40</td>
<td>75</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_75</td>
<td></td>
<td>75</td>
<td>150</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_150</td>
<td></td>
<td>150</td>
<td>300</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_300</td>
<td></td>
<td>300</td>
<td>600</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_600</td>
<td></td>
<td>600</td>
<td>1200</td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_1200</td>
<td></td>
<td>1200</td>
<td></td>
<td></td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_HP</td>
<td>High pass</td>
<td>1</td>
<td></td>
<td>0.2</td>
<td>Hz</td>
</tr>
<tr>
<td></td>
<td>BW_BP</td>
<td>Band pass</td>
<td>300</td>
<td></td>
<td>600</td>
<td>Hz</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>1g</td>
<td>best fit straight line</td>
<td>±0.15</td>
<td></td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%FS</td>
</tr>
<tr>
<td></td>
<td>2g</td>
<td></td>
<td></td>
<td>±0.25</td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3g</td>
<td></td>
<td></td>
<td></td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4g</td>
<td></td>
<td></td>
<td></td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8g</td>
<td></td>
<td></td>
<td>±0.75</td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16g</td>
<td></td>
<td></td>
<td></td>
<td>%FS</td>
<td></td>
</tr>
<tr>
<td>Temperature sensor bandwidth</td>
<td>BW_temp</td>
<td></td>
<td>275</td>
<td></td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>Temperature sensor sensitivity</td>
<td>Temp_sens</td>
<td>Low noise mode</td>
<td>0.5</td>
<td>K/LSBTEMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature sensor offset</td>
<td>Temp_off</td>
<td>Low noise mode</td>
<td>-5</td>
<td>5</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>Data output rate</td>
<td>Rate_out_LN</td>
<td>Low noise mode</td>
<td>2400</td>
<td>Hz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data output rate</td>
<td>Rate_out_LP</td>
<td>Low power mode</td>
<td>1200</td>
<td>Hz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wake-up time</td>
<td>Tw_up</td>
<td>BW = 1200 Hz</td>
<td>1.5</td>
<td>ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start-up time</td>
<td>Tst_up</td>
<td>BW = 1200 Hz</td>
<td>2.5</td>
<td>ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start-up time from sleep mode</td>
<td>Tst_sm</td>
<td>BW = 1200 Hz</td>
<td>2</td>
<td>ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EEPROM write duration</td>
<td>Tee_w</td>
<td>For next EEPROM write</td>
<td>10</td>
<td>ms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

4 In low power mode, BW is divided by 2 (e.g. 10Hz → 5Hz)
5 Delay between power on (VDD from 0 to min. = 1.8 V) and end of first conversion
2 Absolute Maximum Ratings

Stresses above absolute maximum ratings may cause permanent damage to the device. Exceeding the specified characteristics may affect device reliability or cause malfunction.

Table 3: Maximum ratings specified for the BMA180

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MIN</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage</td>
<td>$V_{DD}$ and $V_{DDIO}$</td>
<td>-0.3</td>
<td>4.25</td>
<td>V</td>
</tr>
<tr>
<td>Voltage at any digital pad</td>
<td>$V_{pad_dig}$</td>
<td>$V_{SS}$-0.3</td>
<td>$V_{DDIO}$ +0.3</td>
<td>V</td>
</tr>
<tr>
<td>Storage Temperature range</td>
<td></td>
<td>-50</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Junction temperature</td>
<td>$T_j$</td>
<td></td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>EEPROM write cycles</td>
<td>Same Byte</td>
<td>1000</td>
<td></td>
<td>Cycles</td>
</tr>
<tr>
<td>EEPROM retention times</td>
<td>(both conditions are non cumulative, each represents max. time)</td>
<td>At 55°C, after 1000 cycles</td>
<td>10</td>
<td>Years</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At 85°C, after 1000 cycles</td>
<td>2</td>
<td>Years</td>
</tr>
<tr>
<td>Mechanical Shock</td>
<td>Duration ≤ 200μs</td>
<td>10000</td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>Duration ≤ 1.0ms</td>
<td>3000</td>
<td></td>
<td>g</td>
</tr>
<tr>
<td></td>
<td>Free fall onto hard surfaces</td>
<td>1.8</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ESD</td>
<td>HBM</td>
<td>2000</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>CDM</td>
<td>500</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>MM</td>
<td>200</td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>
3 Block Diagram

The Block diagram shows

- the micromechanical g-sensor elements (measurement of acceleration in x-, y- and z-direction),
- the temperature sensor,
- the front-End-circuit including preamplifiers and analogue pre-filtering circuitry,
- a multiplexer,
- the 14 bit ADC,
- the digital part of the circuitry (responsible for offset regulation, calibration, digital filtering, power regulation)
- the interrupt generation and
- the interface circuitry (I²C and 4-wire SPI)

Other blocks like Power-On Reset, Clock generator, internal band-gap, etc. are not shown.
4 Operation modes

BMA180 is able to work in different operation modes:

- **4 Standard modes** low noise and low power modes
- **Sleep mode** device is “sleeping”; power consumption is at minimum
- **Wake up mode** device is “sleeping” for a certain time, waking up for a certain time, falling back to sleep, etc. This mode is an intermediate mode to the standard modes and sleep mode.

All modes mentioned above are shortly described below. A detailed description of the configuration of these modes is given in 7.7.3.

4.1 Standard operational modes

In standard operational mode the sensor IC can be addressed via digital interface. Data and status registers can be read out and control registers and EEPROM values can be read and changed. In parallel to standard operation the user has the option to activate several internal logic paths and set criteria to trigger the interrupt pin.

BMA180 is providing 4 different sub-modes in standard operation mode (see also 7.7.3).

- low power mode
- low noise mode
- 2 intermediate modes (ultra low noise mode = low noise mode with smaller bandwidth and low noise mode with lower power). In those 2 modes overall sensor specification is limited. For further details ask your Bosch Sensortec representative.

BMA180 is designed to enable low current consumption of typically 650 μA in low power mode, by providing at the same time a very high resolution. In the 3 low-noise modes, current is higher, but resolution is better than in low power mode.

4.2 Sleep mode

4.2.1 General information

Sleep mode is activated by setting a special control bit. In sleep mode reduced communication to the sensor IC is possible. The recommended command to switch back to operational mode is the wake-up call or resetting the sleep mode bit to 0.

Sleep mode could be used,

a) if sensor is only part-time used. In this case μC is deactivating and reactivating BMA180 according to its usage (duty cycling = switching between sleep and standard mode).

b) In small bandwidth applications, sleep mode could be used to save a significant amount of power by frequently changing between sleep and standard mode (see next section).
4.2.2 Current consumption using duty cycling

For most of the applications a sensor does not have to stay permanently in standard mode. This allows a significant reduction of current consumption by switching periodically between sleep and standard mode.

Following 2 examples are giving rough indications about current consumption by duty cycling (depending on power mode and selected bandwidth). Examples are with respect to low frequency applications.

Formulas:
- **average current** = average of sleep mode current and standard mode current
- **measurement time** = time from sleep to standard mode + settling time of filter + read-out time + time from standard to sleep mode

**Example 1:**
- sleep mode current = 1 μA, sleep time is 200 ms, selected filter is 150 Hz
- current mode is low power mode (→ bw = bw\_selected/2)
- start-up time from sleep is 2.5 ms
- read-out time is roughly 1 ms (ADC conversion time is 0.417 ms in low noise mode → approx. 1 ms in low power mode)
  - bw is 75 Hz effective → settling time is 6 * 1/75 sec = 6/75 sec = 80 ms
  - overall measurement time = 2.5 ms + 80 ms + 2.5 ms + 0.5 ms = 85.5 ms
  - current = (200 ms * 1 μA + 85.5 ms * 650 μA) / 285.5 ms = 195 μA.
  - result: approx. 70 % decrease in supply current and power consumption

**Example 2:**
- sleep mode current = 1 μA, sleep time is 500 ms, selected filter is 1200 Hz
- current mode is low-noise mode
- start-up time from sleep is 2.5 ms
- read-out time is roughly 0.5 ms (ADC conversion time is 0.417 ms)
  - bw is 1200 Hz → settling time is 6 * 1/1200 sec = 6/1200 sec = 5 ms
  - overall measurement time = 2.5 ms + 5 ms + 2.5 ms + 0.5 ms = 10.5 ms
  - current = (500 ms * 1 μA + 10.5 ms * 950 μA) / 510.5 ms = 20.5 μA.
  - result: approx. 97 % decrease in supply current and power consumption

**Remark:**
In case of a soft-reset, it is recommended to do this reset after having switched from sleep to operational mode. In this case the total typical wake-up and reset time at maximum bandwidth is much smaller than in case the soft-reset is activated during sleep mode.

4.3 Wake-up mode

4.3.1 General information

In general BMA180 is attributed to low power applications and can contribute to the system power management.

- Current consumption 650 μA operational (low power mode)
- Current consumption < 1 μA in sleep mode
- Wake-up time < 2 ms and
- Start-up time < 3.5 ms
- New data ready indicator to reduce unnecessary interface communication
- Sample skipping in combination with new data interrupt to reduce interface traffic
- Wake-up mode to trigger a system wake-up (interrupt output to master) in case of motion
- Low current consumption in wake-up mode

The BMA180 provides the possibility to wake up a system master when specific acceleration values are detected. Therefore the BMA180 stays in an ultra low power mode and periodically evaluates the acceleration data. If acceleration is above a certain threshold (e. g. high-g threshold) an interrupt can be generated, which triggers the system master. The wake-up mode is used for ultra-low power applications where accelerations can be an initiator to change the activity mode of the system.

### 4.3.2 Current consumption in wake-up mode

For estimating the typical self wake-up current the following formula can be applied:

\[
i_{\text{self\_wake\_up}} = \frac{i_{\text{DD}} \cdot t_{\text{active}} + i_{\text{DDsm}} \cdot \text{wake-up\_pause}}{t_{\text{active}} + \text{wake-up\_pause}}
\]

Where  
\[
t_{\text{active}} = 2\text{ms} + 0.417\text{ms} \cdot (2400 / \text{bandwidth}) + 0.417\text{ms} \cdot (1200 / \text{bandwidth}) \cdot n
\]

With the following parameters:
- \(i_{\text{DD}}\) Current in standard mode
- \(i_{\text{DDsm}}\) Current in sleep mode
- \(\text{wake-up\_pause}\) Setting of wake-up pause
- \(n\) number of data points in any-motion logic (n=0 for high-g threshold and low-g threshold interrupt, n=3 for any-motion logic)
- \(\text{bandwidth}\) Setting of bandwidth: 10 - 1200 Hz

Thus, the relevant parameters for power consumption in self-wake up mode are:
- current consumption in standard mode
- current consumption in sleep mode
- self-wake up pause duration
- bandwidth (e. g. length of digital filter to be filled for one data point)
- interrupt criteria (determines the duration of standard operation)
- high-g and low-g criteria (e. g. acquisition of one data point)
- any-motion criterion (e. g. four data points)

The following graph shows typical average current consumption during the wake-up mode of the BMA180. The power consumption in wake-up mode is dependent on the duration of the interrupt algorithm (number of data acquisitions) and the bandwidth (for more details on setting of the bandwidth please refer to chapter 7.7.2)
**BMA180**

Digital, triaxial acceleration sensor

---

**Figure 2:** Typical average current consumption in wake-up mode using high-g or low-g interrupt; here typical values in low power mode

Same graph is shown for low noise mode

**Figure 3:** Average current consumption in self wake-up mode using high-g or low-g interrupt; here typical values in low-noise mode
5 Data conversion

5.1 Acceleration data

Acceleration data are converted by a 14 bit ADC. The description of the digital signal is "2’s complement". The 14 bit data are available as LSB (at lower register address) and MSB. It is possible to read out MSB only (8 bit) or LSB and MSB together (16 bits with 14 data bits and 1 data ready bit). In second case LSB- and MSB-data are closely linked to avoid unintentional LSB/MSB mixing when read out and data conversion overlap accidentally (7.12.2).

The acceleration data is filtered by a 1-pole analogue filter at 1.2 kHz (low noise and low power mode). Additionally, all data can be processed by digital filtering (2-pole filter) to reduce noise level (10 Hz – 600 Hz) and to filter out undesired frequencies. The transfer function of the mechanical element is designed to avoid resonance effects at frequencies below the bandwidth of the ASIC.

The availability of new data can be checked in two ways:

- Bit 0 from the LSB data registers is an indicator whether data has already been read out or the data is new (new-data bits, see 7.12.2)
- The interrupt pin can be configured to indicate new data availability. The synchronization of data acquisition and data read out enables the customer to avoid unnecessary interface traffic in order to reduce the system power consumption and the crosstalk between interface communication and data conversion.

```
417 μs for bw = 1200 Hz
```

Figure 4: Explanation of data ready interrupt: For a bandwidth of e.g. 1.2 kHz the data refresh cycle takes 417 μs to update all data registers. After the final conversion of z-axis the INT pad will be set high. New data can be read out via interface (recommendation: read out within 20 μs after interrupt is high during the conversion of the next temperature value). The interrupt resets automatically after read out.

5.2 Temperature measurement

Temperature data are converted to an 8 bit data register. The temperature output range can be adapted to customer’s requirements by offset correction.
6 Internal logic functions

The sensor IC can inform the host system about specific conditions (e.g., new data ready flag or acceleration thresholds passed) by setting an interrupt pin high even if interface communication is not taking place. This feature can be used for instance as “low-g indicator”, “wake-up” or “data ready flag”.

The interrupt performance can be programmed by means of control bits. Thus the criteria to identify a special event can be tailored to a customer’s application and the sensor IC output can be defined specifically.

6.1 Low-g interrupt logic

For low-g detection the absolute value of the acceleration data of all axes are investigated (global criteria). A low-g situation is likely to occur when all axes fall below the low threshold value (e.g., in free fall situations). The interrupt pin will be raised high if the threshold is passed for a minimum duration. The duration time can be programmed.

![Figure 5: Schematic behaviour in case of low-g detection](image)

The function “Low-g Interrupt” can be switched on/off by a control bit which is located within the image of the non-volatile memory. Thus this functionality can be stored as default setting of the sensor IC (EEPROM) but can also rapidly be changed within the image.

The reset of the low-g interrupt can be accomplished by means of a master reset of the interrupt flag (latched interrupt) or the reset can be triggered by the acceleration signal itself (validation of a programmable “hysteresis”).

Further details concerning low-g and other interrupts, see 7.8
6.2 **High-g logic**
For indicating high-g events an upper threshold can be programmed. This logic can also be activated by a control bit. Threshold, duration and reset behaviour can be programmed. The high-g and low-g criteria can be logically combined with an <OR>.

6.3 **Slope detection (or any motion detection)**
The “any motion algorithm” can be used to detect changes of the acceleration. Thus it provides a relative evaluation of the acceleration signals. The criterion is kind of a gradient threshold of the acceleration over time. Thus one can distinguish between fast events with strong inertial dynamic (e. g. shock), instant changes of force balance (e. g. drop, tumbling) and even slight changes (e. g. touch of a mobile device).

Due to a high bandwidth and a fast responding MEMS device the BMA180 is capable to detect shock situations up to 1200 Hz and 16 g. The “any motion interrupt” or a high-g criterion setting can be used to give a shock alert. The phase shift between start of mechanical shock and interrupt output is defined by the mechanical transfer function of the chassis and internal mounting interfaces (e. g. PDA shell) and the data output rate of the sensor IC.

6.4 **Tap sensing**
Tap sensing feature is closely related to slope detection/any motion feature. Tap sensing is the generation of an interrupt, if 2 slope detection events are detected within a shorten time. Tap sensing is also known as double click. It is widely used in laptops to open software applications via double click on a touch pad (on system level).

6.5 **Alert Mode**
Using the BMA180 it is possible to combine the “any motion criterion” with low-g and high-g interrupt logic to improve the reaction time for e. g. free-fall identification. If alert mode is set, low-g or motion-counters are counting down earlier than without alert mode.

More information about alert mode is given in a separate application note.
# Global Memory Map

The memory map shows all externally accessible registers needed to operate BMA180.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Address (hexadecimal)</th>
<th>bit7</th>
<th>bit6</th>
<th>bit5</th>
<th>bit4</th>
<th>bit3</th>
<th>bit2</th>
<th>bit1</th>
<th>bit0</th>
<th>Reset state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bosch reserved</td>
<td>00h</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>Bosch reserved</td>
<td>9Ch</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>Bosch reserved</td>
<td>0Ch</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>Bosch reserved</td>
<td>5Ch</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz</td>
<td>50h</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz</td>
<td>51h</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>CE_unfz</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz_x</td>
<td>52h</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>CE_unfz_x</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz_y</td>
<td>53h</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>CE_unfz_y</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz_z</td>
<td>54h</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>CE_unfz_z</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_unfz_latest</td>
<td>55h</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>CE_unfz_latest</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault</td>
<td>56h</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>CE_fault</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault preceded</td>
<td>57h</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>CE_fault preceded</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault_threshold</td>
<td>58h</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>CE_fault_threshold</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault_preceded_latch</td>
<td>59h</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>CE_fault_preceded_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault_trigger</td>
<td>5Ah</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>CE_fault_trigger</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault detect</td>
<td>5Bh</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>CE_fault_detect</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_fault detect_timer</td>
<td>5Ch</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>CE_fault_detect_timer</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_low</td>
<td>5Dh</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>CE_low</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_high</td>
<td>5Eh</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>CE_high</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_high_fault</td>
<td>5Fh</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>CE_high_fault</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_high_fault_latch</td>
<td>60h</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>CE_high_fault_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_high_fault_timer</td>
<td>61h</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>CE_high_fault_timer</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_pad</td>
<td>62h</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>CE_pad</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_pad_latch</td>
<td>63h</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>CE_pad_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ce_pad_timer</td>
<td>64h</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>CE_pad_timer</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad</td>
<td>65h</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>EE_pad</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_latch</td>
<td>66h</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>EE_pad_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_timer</td>
<td>67h</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>EE_pad_timer</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device</td>
<td>68h</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>EE_pad_device</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device_latch</td>
<td>69h</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>EE_pad_device_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device_timer</td>
<td>6Ah</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>EE_pad_device_timer</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device_adc</td>
<td>6Bh</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>EE_pad_device_adc</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device_adc_latch</td>
<td>6Ch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>EE_pad_device_adc_latch</td>
<td>Bosch reserved</td>
</tr>
<tr>
<td>ee_pad_device_adc_timer</td>
<td>6Dh</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>EE_pad_device_adc_timer</td>
<td>Bosch reserved</td>
</tr>
</tbody>
</table>

The left columns inform about the memory addresses. The colours of the bits indicate whether they are read-only, write-only or read- and writable. The non EEPROM part of the memory is volatile so that the writable content has to be re-written after each power-on. The extended address space greater than 3Ch in image resister (or 5Ch in EEPROM area), is not shown. These registers are exclusively used for...
Bosch factory testing and trimming. Also some other bits within the global memory map are reserved and should not be used.

7.1 Global Memory Mapping: general information
The global memory map of BMA180 provides three levels of access:

<table>
<thead>
<tr>
<th>Memory Region</th>
<th>Content</th>
<th>Access Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational Registers</td>
<td>Data registers, control registers, status registers, interrupt settings</td>
<td>Direct access via serial interface</td>
</tr>
<tr>
<td>Default Setting Registers</td>
<td>Default values for operational registers, acceleration and temperature trimming values</td>
<td>Access blocked by default; Access enabled by setting control bit in operational registers via serial interface</td>
</tr>
<tr>
<td>Bosch Sensortec Reserved Registers</td>
<td>Internal trimming registers</td>
<td>Protected</td>
</tr>
</tbody>
</table>

The memory of BMA180 is realized in diverse physical architectures. Basically BMA180 uses volatile memory registers to operate. The volatile part of the memory can be changed and read quickly. Part of the volatile memory ("image") is a copy of the non-volatile memory (EEPROM).

The EEPROM can be used to set default values for the operation of the sensor. EEPROM is indirect write only. The EEPROM register values are copied to the image registers after power on or soft reset. The download of EEPROM bytes to image registers is also done when the content of the EEPROM byte has been changed by a write command. After every write command EEPROM has to be reset by soft-reset.

All operational and default setting registers are accessible through serial interface with a standard protocol:

<table>
<thead>
<tr>
<th>Type of Register</th>
<th>Function of Register</th>
<th>Command</th>
<th>Volatile / non-volatile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Registers</td>
<td>− Chip identification, chip version</td>
<td>Read</td>
<td>non-volatile</td>
</tr>
<tr>
<td></td>
<td>− Acceleration data, temperature</td>
<td>Read</td>
<td>volatile</td>
</tr>
<tr>
<td>Control Registers</td>
<td>− Activating self test, soft reset, switch</td>
<td>Read /</td>
<td>Volatile</td>
</tr>
<tr>
<td></td>
<td>to sleep mode etc.</td>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>Status Registers</td>
<td>− Interrupt status and self test status</td>
<td>Read</td>
<td>volatile</td>
</tr>
<tr>
<td></td>
<td>− Customer reserved status bytes</td>
<td>Read /</td>
<td>volatile</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>Setting Register</td>
<td>− Functional settings (range, bandwidth, mode, etc.)</td>
<td>Read /</td>
<td>volatile</td>
</tr>
<tr>
<td></td>
<td>− Interrupt settings</td>
<td>Write</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>EEPROM</td>
<td>− Default settings of functional and</td>
<td>Write</td>
<td>non-volatile</td>
</tr>
<tr>
<td></td>
<td>interrupt settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>− Trimming values</td>
<td>Write</td>
<td>non-volatile</td>
</tr>
<tr>
<td></td>
<td>− Customer reserved data storage</td>
<td>Write</td>
<td>non-volatile</td>
</tr>
<tr>
<td></td>
<td>− Bosch Sensortec Reserved Memory</td>
<td></td>
<td>non-volatile</td>
</tr>
</tbody>
</table>
- The global memory mapping contains EEPROM and latches.
- All EEPROM registers are duplicated into the corresponding image registers.
- Writing to unused bits has no effect on the IC; reading unused bits leads to undefined level.
- Image registers are used to download the EEPROM content to be able to act on IC functions. Registers 20h to 3Fh thus directly correspond to EEPROM bytes 40h to 5Fh.

### 7.2 Registers
There are 5 types of registers in the sensor – test, control, image, status and data registers. All registers are 8-bit. Image and control registers are accessible in read/write mode by the user.

- Test registers are reserved for Bosch, they are not described here. They are not accessible to the customer.
- Control registers are used to set-up the functional mode of IC. See next paragraphs for detailed description of each bit. Few bits are one-shot control bits.
- Status registers contain useful information about the alert/interrupt modes and to know if new acceleration data is available since latest read-out.
- Image registers contain EEPROM values and are downloaded after release of POR, soft-reset or when the `update_image` command is send to BMA180. Writing to these registers has no effect on EEPROM content. Image registers can directly be accessed to trim the device without using any EEPROM write procedure in case of several iterations during calibration. Image registers can also be used to overwrite BMA180 settings defined in the EEPROM memory. It is possible to come back to EEPROM memory settings at anytime by writing `update_image` control bit to 1.
- Data registers contain the 3 acceleration values, the temperature value and information about the chip (see 7.12.3)

### 7.3 Programming of the Calibration Parameters
The full-sensor functionality and precision is provided by trimming the sensor on wafer level and on sensor level during End-of-Line testing. In order to achieve highest precision (e. g. offset accuracy) even after soldering the device onto a PCB, the user can recalibrate the trimming correction values after mounting.
7.4 Register Arithmetic
The following arithmetic is used for memory registers.

<table>
<thead>
<tr>
<th>Register</th>
<th>Format</th>
<th>Bit width</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFSETX</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>GAINX</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>TCox</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>TCS</td>
<td>offset binary</td>
<td>4</td>
</tr>
<tr>
<td>AX</td>
<td>Y</td>
<td>Z (acceleration values)</td>
</tr>
<tr>
<td>Temp.</td>
<td>2's complement</td>
<td>8</td>
</tr>
<tr>
<td>THRESHOLD (TH or TH_X</td>
<td>Y</td>
<td>Z)</td>
</tr>
<tr>
<td>HYSTERESIS (HY or HY_X</td>
<td>Y</td>
<td>Z)</td>
</tr>
</tbody>
</table>

7.5 EEPROM
7.5.1 General information
The embedded EEPROM memory is used to trim analogue parameters and to set-up the interrupt function; it is organized in 16 words of 16 bits (each word contains 2x8 bits).

Each EEPROM data has a corresponding image which is used to latch EEPROM data. Image content act on analog part, it is also used as buffer to read and write to EEPROM. EEPROM data are downloaded into image registers after each of the following events:
- Power On Reset
- Reset command sent through interface (soft reset)
- Control bit update_image set to ‘1’.

7.5.2 EEPROM Reading
No direct EEPROM reading is implemented; result of reading addresses 40h to 5Fh returns content of addresses 20h to 3Fh.

For Reading the EEPROM registers it is possible to download the EEPROM registers in to the image registers by setting update_image=1 and read out the corresponding image register.

7.5.3 EEPROM Writing
Writing to EEPROM is locked by default to prevent mal-function. To unlock writing in the image registers of the non-protected area, set ee_w to ‘1’.

As EEPROM reading, EEPROM writing is also an indirect procedure. Data from corresponding image registers are written to EEPROM after sending write transaction to addresses 40h to 5Fh.
As EEPROM word is 16-bit (and writing is done in parallel), transaction with even address writes to this address (A) and one address above (A+1). The transaction with odd address is ignored. Data of writing transition is ignored (SPI) or can be omitted (I2C).

**Example:**
SPI writing to address 50h starts writing operation (register 30h to EEPROM 50h, register 31h to EEPROM 51h).

EEPROM write operation shouldn’t occur when an *update_image* is ongoing (EEPROM is in read mode at this moment). This means EEPROM write is forbidden also during 10 ms after power ON reset, after *soft_reset* and after *update_image* is written to 1. EEPROM write is also forbidden during sleep mode and for 10 ms after removal of sleep mode.

EEPROM write operation could render ADC conversion results unusable, thus a soft-reset after EEPROM write is necessary.

### 7.5.4 EEPROM protection
The EEPROM bytes between addresses 5Ch to 5Fh are protected in write mode because it contains Bosch Sensortec proprietary information and settings. It also contains the *ee_w_flag* which can be used to detect if any EEPROM write sequence occurred after final test (i.e. if the customer wrote anything into the non-protected EEPROM). Customer is not able to write the *ee_w_flag*.

Also the image registers corresponding to the locked EEPROM area are locked by the EEPROM lock mechanism in order to avoid mal-functions on the overall system.

### 7.5.5 EEPROM content upon delivery (after production test)

1. **CRC**: CRC code may be used to verify whether a calibration was done after Bosch production test
2. **CD1, CD2**: content may differ for each IC, since these bytes can be used by customer to store any data in the non-volatile memory. Its content does not influence the ASIC functionality.
3. **Analog trimming bits (addresses 5Bh to 5Eh)**: Content may differ for each IC.
4. **Calibration data**: These data and its default values are summarized in the following table.
### Register name and default value

<table>
<thead>
<tr>
<th>Register name</th>
<th>default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset_x, offset_y, offset_z, offset_t</td>
<td>calibrated value</td>
</tr>
<tr>
<td>gain_x, gain_y, gain_z, gain_t</td>
<td>calibrated value</td>
</tr>
<tr>
<td>tco_x, tco_y, tco_z</td>
<td>calibrated value</td>
</tr>
<tr>
<td>tcs</td>
<td>calibrated value</td>
</tr>
<tr>
<td>bw</td>
<td>0100b</td>
</tr>
<tr>
<td>range</td>
<td>010b</td>
</tr>
<tr>
<td>wake_up_dur</td>
<td>10b</td>
</tr>
<tr>
<td>slope_dur, mot_cd_r, ff_cd_r, offset_finetuning</td>
<td>01b</td>
</tr>
<tr>
<td>mode_config</td>
<td>00b</td>
</tr>
<tr>
<td>tapsens_dur</td>
<td>100b</td>
</tr>
<tr>
<td>adv_int</td>
<td>1b</td>
</tr>
<tr>
<td>high_th</td>
<td>01010000b</td>
</tr>
<tr>
<td>low_th</td>
<td>00010111b</td>
</tr>
<tr>
<td>high_dur</td>
<td>0110010b</td>
</tr>
<tr>
<td>low_dur</td>
<td>1010000b</td>
</tr>
<tr>
<td>high_int_<em>, low_int_</em>, tapsens_int_<em>, slope_int_</em></td>
<td>1b</td>
</tr>
</tbody>
</table>

All bits, which are not-mentioned in above table are set by default to 0b.

#### 7.5.6 ee_w_flag - EEPROM-written flag

This EEPROM bit is set to ‘1’ as soon as the first EEPROM write to addresses 40h to 5Bh occurs. Any write operation to the non-protected area results in updating of internal registers and ee_w_flag will automatically be set to 1; the user is not able to write this flag back to “0”, since it is placed in the EEPROM protected area.

Remark: please do no mix ee_w_flag with ee_w bit.

#### 7.5.7 EEPROM Endurance

An EEPROM is inherently limited to a maximum number of write cycles. If more cycles are performed, failures can occur which affect the functionality of the sensor. In case of the BMA180 the specified numbers of write cycles is 1000. This maximum number of write cycles should not be exceeded in any application in order to prevent possible failures.

#### 7.6 Image

#### 7.6.1 Image writing

Writing to Image is locked by default to prevent mal-function. To unlock writing, use same command as for EEPROM writing -> set ee_w to ‘1’.

#### 7.6.2 Image reading

Direct reading of the image is possible: no unlock procedure has to be performed.
7.7 General functional settings

7.7.1 range

These 3 bits are used to select the full scale acceleration range (further included are the ADC-resolutions).

<table>
<thead>
<tr>
<th>range&lt;2:0&gt;</th>
<th>Full scale acceleration range [+- g]</th>
<th>ADC resolution [mg/LSB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1</td>
<td>0.13</td>
</tr>
<tr>
<td>001</td>
<td>1.5</td>
<td>0.19</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
<td>0.38</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
<td>0.50</td>
</tr>
<tr>
<td>101</td>
<td>8</td>
<td>0.99</td>
</tr>
<tr>
<td>110</td>
<td>16</td>
<td>1.98</td>
</tr>
<tr>
<td>111</td>
<td>Not authorised code</td>
<td>Not authorised code</td>
</tr>
</tbody>
</table>

Directly after changing the full scale range it takes approximately $1/(2*\text{bandwidth})$ before filters (see next section) are providing correct data.

7.7.2 bw

A 1-pole analogue filter defines the maximum bandwidth in the front-end-circuitry to 1.2 kHz. In order to further increase signal-to-noise-ratio, digital filters can be activated to reduce the bandwidth down to 10 Hz. The digital filters are second order filters. Selection of the filters could be done by using the 4 bits below (first 8 filters are low-pass filters).

<table>
<thead>
<tr>
<th>bw&lt;3:0&gt;</th>
<th>Selected bandwidth (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>10</td>
</tr>
<tr>
<td>0001</td>
<td>20</td>
</tr>
<tr>
<td>0010</td>
<td>40</td>
</tr>
<tr>
<td>0011</td>
<td>75</td>
</tr>
<tr>
<td>0100</td>
<td>150</td>
</tr>
<tr>
<td>0101</td>
<td>300</td>
</tr>
<tr>
<td>0110</td>
<td>600</td>
</tr>
<tr>
<td>0111</td>
<td>1200</td>
</tr>
<tr>
<td>1000</td>
<td>high-pass: 1 Hz</td>
</tr>
<tr>
<td>1001</td>
<td>band-pass: 0.2 Hz .. 300 Hz</td>
</tr>
<tr>
<td>1010 to 1111</td>
<td>not authorized codes</td>
</tr>
</tbody>
</table>

Interrupts might be disabled and re-enabled for each bw change (due to the risk of wrong generated interrupt).

If bw value is written successively with 2 different values, a minimum delay of 10 $\mu$s between the 2 write sequences must be respected to guarantee the latest value will be set correctly. This is valid for image register. EEPROM access time is anyway much slower and this does not affect ASIC function. bw setting is expected to be changed at very low rates.

At wake-up from sleep mode to standard operation, the bandwidth is set to its maximum value and then reduced to bandwidth setting as soon as enough ADC samples are available to fill the whole digital filter.
7.7.3 mode_config

BMA180 has four different working sub-modes in standard mode. By setting the mode_config bits the sub-mode is configured as described in the following table.

<table>
<thead>
<tr>
<th>mode_config &lt;1:0&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Low noise mode -&gt; highest current, low noise, full bandwidth (1200 Hz)</td>
</tr>
<tr>
<td>01</td>
<td>Ultra low noise mode -&gt; highest current, lowest noise, reduced bandwidth (300 Hz)</td>
</tr>
<tr>
<td>10</td>
<td>Low noise mode with reduced power, reduced bandwidth (150 Hz), output data rate = 1200 samples/sec.</td>
</tr>
<tr>
<td>11</td>
<td>Low power mode -&gt; BW is decreased by factor 2, lowest power, noise higher than in low noise modes, output data rate = 1200 samples/sec</td>
</tr>
</tbody>
</table>

00b and 11b are the main configuration modes. 10b and 01b are considered as intermediate configuration modes (intermediate in terms of either noise level or current consumption).

Table below informs about relation between mode_config, bandwidth, current and noise and allows choice of optimal settings to optimize necessary performance for the intended applications (all values are typical values).

<table>
<thead>
<tr>
<th>mode config</th>
<th>Explanation</th>
<th>bw for bw-setting 10 Hz</th>
<th>bw for bw-setting 1200 Hz (1g, 1.5g, 2g mode)</th>
<th>bw for bw-setting 1200 Hz (3g, 4g )</th>
<th>bw for bw-setting is 1200 Hz (8g, 16g)</th>
<th>typical current (incl. filtering) [μA]</th>
<th>typical noise density [μg/rt(Hz) in 2g-mode]</th>
<th>Interrupt &quot;timings&quot; (e. g. tap sensing duration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>low noise mode</td>
<td>10</td>
<td>1200</td>
<td>1200</td>
<td>1200</td>
<td>1025</td>
<td>150</td>
<td>Standard</td>
</tr>
<tr>
<td>01</td>
<td>ultra-low noise mode (low noise mode with reduced noise and bw)</td>
<td>10</td>
<td>472</td>
<td>944</td>
<td>1200</td>
<td>1025</td>
<td>150</td>
<td>Standard</td>
</tr>
<tr>
<td>10</td>
<td>low noise mode with lower power, lower noise and reduced bw</td>
<td>5</td>
<td>236</td>
<td>472</td>
<td>600</td>
<td>900</td>
<td>200</td>
<td>doubled to standard</td>
</tr>
<tr>
<td>11</td>
<td>low power mode</td>
<td>5</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>650</td>
<td>200</td>
<td>doubled to standard</td>
</tr>
</tbody>
</table>

Important remarks:
- When bw is decreased by 2 for mode 11b, all timings used by the digital functions and the system clock frequency remain related to the bw.

- Any change of the mode_config bits results in transient behaviour of the measured acceleration values. The length of the transient response depends on the selected bandwidth. Also some spurious interrupts might be generated as a result of the mode_config change.
The length of the transient response corresponds to an appropriate number of acceleration samples to be acquired. This number of samples is depending on the chosen filter bandwidth and is shown below.

<table>
<thead>
<tr>
<th>Bandwidth (Hz)</th>
<th>120</th>
<th>600</th>
<th>300</th>
<th>150</th>
<th>75</th>
<th>40</th>
<th>20</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr. of samples</td>
<td>0</td>
<td>6</td>
<td>9</td>
<td>18</td>
<td>35</td>
<td>64</td>
<td>127</td>
<td>253</td>
</tr>
</tbody>
</table>

- The sensor is calibrated for `mode_config = "00"`. By changing to other modes, the offset is changed too, thus subsequently an offset correction has to be performed. This could be done by offset fine-tuning the device during in-line calibration.

- It is highly recommended not to change `mode_config` within an application. In this case, offset fine-tuning is still working, but it is not ensured, that the device is properly placed during calibration. If the device is e. g. used in ultra-low noise mode, bandwidth is limited and highest current consumption could be measured. In order to save current, a frequent switching from operation to sleep mode and back could be used (as already described in section 4.2). This is no problem for most of the applications. Furthermore a “switching” error in offset might occur, if modes are changed, thus offset finetuning might be necessary if `mode-config` is changed within an application.

- After a write of `mode_config` bits in EEPROM, a soft reset is mandatory.

### 7.7.4 readout_12bit

For the acceleration read-out code, this bit allows switching from 14 bit (default mode, readout_12bit = '0') to 12 bit (readout_12bit = '1'). In this case, the two last LSB are set by default to '0'. This might be useful if all other devices in a system just use 12 bit.

### 7.7.5 smp_skip (sample skipping)

Intension of this bit is to minimize MCU load, especially in case of very low BW. This bit is only useful if new_data_int = 1.

- When `smp_skip` is set to ‘0’, interrupt is generated at 1/Tupdate.

- When `smp_skip` is set to ‘1’, interrupt is generated depending on bandwidth, it is twice the selected bandwidth. For example, if `bw = 0110b (600 Hz)`, interrupt is generated at the half of the frequency of the sampling rate, thus at 1200 Hz. For `bw = 10 Hz`, interrupt is generated at 20 Hz. In low power mode, `bw = 5 Hz` and interrupt is generated at 10 Hz.

Additional advantage of using this bit in combination with new_data_int = 1 is noise optimization. If customer is not using new_data_int and is collecting data with “small” data rate, effect of MCU load minimization is similar, but noise might be higher, since data is not collected synchronously to availability of new data (increase of noise due to interface traffic).
7.7.6  shadow_dis
BMA180 provides the possibility to block the update of the data MSB while data LSB are read out. This avoids a potential mixing of LSB and MSB of successive conversion cycles. When this bit is at 1, the shadowing procedure for MSB is not realized and MSB only reading is possible.

7.7.7  dis_reg
When this bit is at ‘1’, the internal regulators are disabled and are by-passed. This allows ultra low voltage operation with highly stabilized external power supply. In this case PSRR of the external power supply is determining the PSRR of the whole device.
**Attention:** if dis_reg = ‘1’, voltage must not exceed 2 V (see specification part). It is highly recommended to take special care on this in the hw/sw-design of the device.

7.7.8  wake_up
This bit makes BMA180 automatically switching from sleep mode to standard mode after the delay defined by wake_up_dur (see next section). The ASIC is also able to switch from standard to sleep mode; an interrupt condition must be defined and the IC will go to sleep mode as soon as all required computations have been performed.

When the IC goes from sleep to standard mode, it starts acceleration acquisition and performs the interrupt verification. If a latched interrupt is generated, this will wake-up the microprocessor, the IC will wait for a reset_int command. If non-latched interrupt is generated, the device waits in the standard mode till the interrupt condition disappears. If no interrupt is generated, the IC goes to sleep mode for 20 to 2560ms. BMA180 cannot go back to sleep mode if reset_int is not issued after a latched interrupt.

After setting wake_up to ‘1’, the device goes to the sleep mode and cycle sleep/wake-up/sleep is started.

The IC wakes-up for a minimum duration which depends on the number of required valid acceleration data to determine if an interrupt should be generated.

For example, if bw = 0111, low_int = 1 and low_dur = 31d, the IC will need time to acquire a minimum number of acceleration data: the IC needs low_dur = 31d = 31*5*Tupdate = 64.6 ms to determine if the acceleration is under low_th. Under this example condition, the minimum wake-up time is 64.6 ms. For smaller wake-up times, low_dur has to be decreased significantly.

To activate wake_up bit in EEPROM, the following procedure is necessary:
1) set register dis_wake_up to “1” (wake-up mode is masked)
2) set image register wake_up to “1”
3) write register wake_up to EEPROM -> dummy write to address 0x54.
4) set register dis_wake_up back to “0” (go to wake-up mode)

To disable wake_up bit in EEPROM, the following procedure is necessary:
5) set register dis_wake_up to “1” (wake-up mode is masked)
6) set image register wake_up to “0”
7) write register wake_up to EEPROM -> dummy write to address 0x54.
8) set register dis_wake_up back to “0” (optional)
7.7.9  *wake_up_dur*
These bits define the sleep mode duration between each automatic wake-up (timing below is valid for low-noise mode, in low power mode, sleep mode duration is doubled).

<table>
<thead>
<tr>
<th>wake_up_dur&lt;1:0&gt;</th>
<th>Sleep mode duration (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>20</td>
</tr>
<tr>
<td>01</td>
<td>80</td>
</tr>
<tr>
<td>10</td>
<td>320</td>
</tr>
<tr>
<td>11</td>
<td>2560</td>
</tr>
</tbody>
</table>

7.7.10  *slope_alert*
If this bit is at 1, the slope_th_criteria will turn BMA180 in an alert mode. This bit can be masked by *adv_int*, the value of this bit is ignored when *adv_int* = 0 (in other words: if slope_alert is used, adv_int has to be set to ‘1’).

More explanations to slope alert is given in a separate application note (under construction).

7.7.11  *dis_i2c* – disable I²C
This bit could be used to disable the I²C mode. Per default, both interfaces are usable (*dis_i2c* = “0”), thus automatic switching from SPI to I²C when CSB at high is enabled. For disabling I²C, *dis_i2c* has to be set to “1”. If SPI-interface is used, it is highly recommended to set dis_i2c to ‘1’ to avoid mal-function.

7.7.12  *CRC* - Checksum bits
CRC are checksum bits used to verify the integrity of the trimming codes. This is used for the check of the EEPROM content during the packaging procedure. Furthermore it could be checked, whether a trimming procedure at customer side has been performed.

7.7.13  *ee_cd1, ee_cd2* - Customer data
These 2 bytes can be used by customers to store any data in the non-volatile memory. Its content does not influence the ASIC functionality. At Bosch production site these registers are used to store inter-mediate data, the content of which may differ from sensor to sensor. This is of no importance for the proper functionality of the sensor.

7.8  Interrupt Settings
The sensor is providing 6 different types of user programmable interrupts. When any interrupt condition is valid, the INT pad goes to 1. If many interrupts are activated at the same time, INT goes high when at least one of the interrupt criteria exists. Interrupts can be chosen as latched (interrupt reset by μC necessary) or non latched (interrupt disappears as soon as interrupt condition disappears)

Interrupts generations may be disturbed by EEPROM, image or control bits changes because some of these bits influence the interrupt calculation. As a consequence, no write sequence should occur when microprocessor is triggered by INT or the interrupt should be disabled on the microprocessor side when write sequences must be operated.
Interrupt criteria are using digital code coming from digital filter output, as a consequence all thresholds are scaled with full scale selection (depends on range control bits). Timings used for high acceleration and low acceleration de-bouncing are absolute values (1 LSB of high_dur and low_dur registers corresponds to 5*Tupdate (=2.085ms), timing accuracy is proportional to oscillator accuracy = +/-10%), thus it does not depend on selected bandwidth. Timings used for slope interrupt and slope alert detection are proportional to bandwidth settings.

All interrupt criteria are combined and drive the INT pad with an OR condition.

All interrupts are temporarily blocked after the wake-up, system reset or filtering bandwidth change until there are enough samples to evaluate the interrupt condition for the first time (1 sample for threshold, 4 samples for slope and tap sensing). There is a dependence on the filtering settings of the selected interrupt:

- if low_filt = 0, interrupt condition is evaluated from the non-filtered data and might be enable virtually immediately after the power-up.
- If low_filt = 1, interrupt condition is evaluated from the filtered data and shall be enabled as soon as the transient response of the filter disappears.

New data interrupt is enabled only if samples are available for reading.

7.8.1 adv_int
This bit is used to disable 3 advanced interrupt control bits: slope_alert, slope_int and st_damp. If adv_int = 0, writing these advanced interrupt control bits to 1 has no effect on IC functions (these bits are ignored). This feature is used to avoid IC mal-function when the above mentioned advanced interrupt features shouldn’t be used but these bits are written to 1 by error.

<table>
<thead>
<tr>
<th>adv_int</th>
<th>Advanced interrupt control bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Can’t be activated (writing them to 1 has no effect)</td>
</tr>
<tr>
<td>1</td>
<td>Functions of the bits are enabled</td>
</tr>
</tbody>
</table>

7.8.2 new_data_int
If this bit is set to 1, an interrupt will be generated when all three axes acceleration values are new, i.e., BMA180 updated all acceleration values after latest serial read-out. Interrupt generated from new data detection is a latched one; microcontroller has to write reset_INT at 1 after interrupt has been detected high. This interrupt is also reset by any acceleration byte read procedure (read access to address 02h to 07h).

New data interrupt always occurs at the end of the Z-axis value update in the output register (2.4 kHz rate, if smp_skipping = ‘0’, at 2*bw rate, if smp_skipping = ‘1’). Following figure shows two examples of X-axis read out and the corresponding interrupt generation.

Explanation of new data interrupt (please refer to section 5.1 for more details):

```
T X Y Z T X Y Z T X Y Z T X Y Z
```

X-axis value read out
New data interrupt

```
X Y Z T X Y Z T X Y Z
```

X-axis value read out
New data interrupt
left side  - read out command of x-axis prior to next x-axis conversion
          → new data interrupt after completion of current conversion cycle
          after z-axis conversion

right side - read out of x-axis send after x-axis conversion
          → new data interrupt at the end of next period when x axis has been updated

Note: When using the I²C interface for data transfer, the data read out phase can be longer than 417 µs (depending on I²C clock frequency and the amount of data transmitted). Starting a new data read out sequence may lead to a situation where the new_data_int may not be cleared right in time. This must be considered and taken care of properly.

7.8.3  lat_int
When this bit is at 1, interrupts are latched: the INT pad stays high until microprocessor detects it and writes reset_int control bit to 1.

When this bit is at 0, interrupts are non-latched: interrupts are set and reset directly by BMA180 (e. g. interrupt condition disappears -> interrupt pin is reset to 0).

Following interrupts are influenced by lat_int:
- high    high-g interrupt (high-g detection)
- low     low-g interrupt (low-g or free-fall detection)
- slope   slope or any-motion detection
- tap-sensing  double tap detection

7.8.4  Low-g interrupt
7.8.4.1  General Explanation
Functionality is as follows: the sensor is measuring acceleration and comparing the measured value with a predefined value. If acceleration is below this value and long enough, a low-g interrupt is generated. If acceleration is above, no interrupt occurs. Sign of the acceleration is also considered, thus absolute value is checked and compared to a given value.

Due to different devices (cell-phone, PND, lap-top, etc.) with different internal mechanical constructions and placements of the sensor on a Printed-Circuit-Board (PBC), the sensor is providing different parameters, the configuration of which is enabling device manufacturers to optimize low-g detection.

7.8.4.2  Low-g interrupt configuration parameters and settings
The following configuration parameters/settings are provided (all unsigned integer)

- low_int: This bit enables the low_th_criteria to generate an interrupt.
- low_th: defining low-g threshold value
• low_hy: defining associated low threshold hysteresis to prevent permanent interrupt generation in case acceleration signal is too close to threshold value

• low_int_x: defining if low-g event on x-axis should generate low-g interrupt

• low_int_y: defining if low-g event on y-axis should generate low-g interrupt

• low_int_z: defining if low-g event on z-axis should generate low-g interrupt

• low_filt: evaluation if interrupt generation is done with filtered (low_filt = “1”) or unfiltered (low_filt = “0”) acceleration signal.

• low_dur: low threshold duration

• ff_cd_r: low_g counter_down_register are used for debouncing low-g criteria.

Remarks:  
- The thresholds codes are compared with the 8 MSB bits of acceleration value (in absolute value), the low threshold level can thus be selected anywhere in the full scale range.

- The sign of the acceleration, which has initiated the interrupt signal, is stored in the flag bit low_sign_int_* (see 7.11.77.11), only if the corresponding enable bit low_int_* is set.

7.8.4.3 Low-g interrupt: algorithm

In figure 6 an example is given to explain functionality of the configuration settings and the algorithm behind the calculation of the interrupt generation.

![Figure 6: example of low-g detection debouncing with use of low_th, low_hy, low_dur and ff_cd_r settings](image)
When acceleration signal is passing low_th value, low_th_criteria becomes active and counter \texttt{ff_cd_r} is incremented by 1 \texttt{N_COUNT}. \texttt{(Ncount = 1 LSB/(5xUpdate) = 1 LSB/2.085ms in low noise modes)}. Depending on \texttt{ff_cd_r} register value, the counter could also be reset or count down when low_th_criteria is false:

<table>
<thead>
<tr>
<th>\texttt{ff_cd_r}&lt;1:0&gt;</th>
<th>Low-g interrupt counter status when low_th_criteria is false</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>reset</td>
</tr>
<tr>
<td>01</td>
<td>Count down by 1 \texttt{N_COUNT}</td>
</tr>
<tr>
<td>10</td>
<td>Count down by 2 \texttt{N_COUNT}</td>
</tr>
<tr>
<td>11</td>
<td>Count down by 3 \texttt{N_COUNT}</td>
</tr>
</tbody>
</table>

When the low acceleration interrupt counter value equals low_dur \texttt{(low_dur <>0)}, an interrupt is generated. If low_dur =0, an interrupt is generated as soon as the appropriate criteria is fulfilled.

The low_th_criteria is set with an AND condition, thus in an application acceleration signals of all 3 axis must be long enough below a certain threshold.

If latch_INT=0, the interrupt is not a latched interrupt (as in figure 6) and then it is reset as soon as low_th_criteria becomes false. When interrupt occurs, the interrupt counter is reset.

Remark: After completion of the offset regulation/storage process, offset fine-tuning bits have to be reset to “00” to re-enable the low-interrupt function (details see 7.9.3)

### 7.8.5 High-g interrupt

#### 7.8.5.1 General Explanation

BMA180 is providing a possibility to detect high-g events of a device. Functionality is basically as follows: the sensor is measuring acceleration and comparing the measured value with a certain predefined value. If acceleration is above this value long enough, a high-g interrupt is generated. If acceleration is below, no interrupt occurs. Sign of the acceleration is also considered, thus absolute value is checked and compared to a given value.

Due to different devices (cell-phone, PND, lap-top, etc.) with different internal mechanical constructions and placements of the sensor on a Printed-Circuit-Board (PBC), the sensor is providing different parameters, the configuration of which is enabling device manufacturers to optimize high-g detection.

#### 7.8.5.2 High-g interrupt configuration parameters and settings

The following configuration parameters/settings are provided (all unsigned integer)

- \texttt{high_int}: This bit enables the high_th_criteria to generate an interrupt.
- \texttt{high_th}: defining high-g threshold value
- \texttt{high_hy}: defining associated high threshold hysteresis to prevent permanent interrupt generation in case acceleration signal is too close to threshold value
- \texttt{high_int_x}: defining if high-g event on x-axis should generate high-g interrupt
• **high_int_y:** defining if high-g event on y-axis should generate high-g interrupt

• **high_int_z:** defining if high-g event on z-axis should generate high-g interrupt

• **high_filt:** evaluation if interrupt generation is done with filtered (high_filt = “1”) or unfiltered (high_filt = “0”) acceleration signal.

• **high_dur:** high threshold duration

• **mot_cd_r:** motion_counter_down_register are used for debouncing high-g criteria.

Remarks:
- The thresholds codes are compared with the 8 MSB bits of acceleration value (in absolute value), the high threshold level can thus be selected anywhere in the full scale range.
- The sign of the acceleration, which initiated the interrupt signal, is stored in the flag bit `high_sign_int_*`, only if the corresponding enable bit `high_int_*` is set.

### 7.8.5.3 high-g interrupt: algorithm

The example in figure 6 for low-g interrupt can easily be transferred to the high-g detection.

When acceleration signal is passing high_th value, high_th_criteria becomes active and counter mot_cd_r is incremented by 1 \( N_{COUNT} \) (\( N_{count} = 1 \text{ LSB}/(5 \times T_{update}) = 1 \text{ LSB}/2.085 \text{ms} \)). Depending on mot_cd_r register value, the counter could also be reset or count down when high_th_criteria is false:

<table>
<thead>
<tr>
<th>mot_cd_r&lt;1:0&gt;</th>
<th>High acceleration interrupt counter status when high_th_criteria is false</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>reset</td>
</tr>
<tr>
<td>01</td>
<td>Count down by 1 ( N_{COUNT} )</td>
</tr>
<tr>
<td>10</td>
<td>Count down by 2 ( N_{COUNT} )</td>
</tr>
<tr>
<td>11</td>
<td>Count down by 3 ( N_{COUNT} )</td>
</tr>
</tbody>
</table>

When the high acceleration interrupt counter value equals `high_dur` (\( high_dur <>0 \)), an interrupt is generated. If `high_dur =0`, an interrupt is generated as soon as the appropriate criteria is fulfilled.

The high_th_criteria is set with an OR condition on the three axis to be used as a high-g detection, thus acceleration signals of minimum 1 axis must be long enough above a certain threshold.

If latch_INT=0, the interrupt is not a latched interrupt (as in figure 6) and then it is reset as soon as High_thresh criteria becomes false. When an interrupt occurs, the interrupt counter is reset.
7.8.6  Slope interrupt (any motion interrupt)

7.8.6.1  General Explanation
BMA180 is providing a possibility to detect slope/any motion events of a device (e.g. tumbling). Functionality is basically as follows (see figure below): the sensor is measuring successive accelerations, the data of which is stored internally. Slope d(acc_*)/dt is determined and compared to a preconfigured any motion threshold. Interrupt or slope alert can be generated when absolute value of measured slope is higher than the programmed threshold for long enough duration. If slope is below any-motion-thresh-hold, Interrupt is reset. Slope interrupt is performed, if at least 1 axis is leading to an interrupt (OR-connection).

Due to different devices (cell-phone, PND, lap-top, etc.) with different internal mechanical constructions and placements of the sensor on a Printed-Circuit-Board (PBC), the sensor is providing different parameters, the configuration of which is enabling device manufacturers to optimize slope detection.

![Figure 7: any motion detection with interrupt generation (schematic view)](image)

7.8.6.2  Slope interrupt configuration parameters and settings
The following configuration parameters/settings are provided (all unsigned integer)

- **slope_int**: This bit enables the slope_th_criteria to generate an interrupt. It cannot be turned on simultaneously with slope_alert. This bit can be masked by adv_int (value of slope_int is ignored, if adv_int = 0).

- **slope_th**: defining slope threshold value, LSB size corresponds to 15.6 mg for +/-2g range and scales with range selection.
- **slope_int_x:** defining if any motion event on x-axis should generate slope interrupt
- **slope_int_y:** defining if any motion event on y-axis should generate slope interrupt
- **slope_int_z:** defining if any motion event on z-axis should generate slope interrupt
- **slope_filt:** evaluation if interrupt generation is done with filtered (slope_filt = “1”) or unfiltered (slope_filt = “0”) acceleration signal.

If slope_filt = 1, the signal is filtered and the slope condition depends on bw settings.

If slope_filt = 0, the signal is unfiltered and the slope condition depends only on the maximum bandwidth.

- **slope_dur:** slope_dur determines interrupt duration before generating interrupt
- **slope_alert:** if this bit is set, the sensor is turned into alert mode. This bit can be masked by adv_int (value of slope_int is ignored, if adv_int = 0).

### 7.8.6.3 slope interrupt: algorithm

An example of slope detection (here bw = 0111b, slope_dur = 01b, slope_int = 1) is shown below. At a certain time, a high slope is detected and INT is set to “1” (high slope has to stay for 3 consecutive data points, here until time t0). If slope is decreased, after a certain time low slope is detected (again after 3 consecutive data points) and INT is reset at time t1 to “0”.

![Image of acceleration slope detection](image)

**Figure 8:** acceleration slope detection, example with slope_dur = 01b, 3 consecutive slope criteria must be detected.
slope_dur is used to filter the slope detection and also to determine minimum interrupt duration because the reset condition is also filtered. The minimum interrupt duration is \( \text{slope\_dur} \times n \times \text{Tupdate} \).

<table>
<thead>
<tr>
<th>slope_dur&lt;1:0&gt;</th>
<th>Number of required consecutive conditions to set or reset the slope_th_criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
</tr>
</tbody>
</table>

slope_th_criteria can be used to generate a slope interrupt or to put BMA180 in alert mode; this is selected by slope_int and slope_alert settings. These 2 modes can not be turned ON simultaneously.

The sign of the last slope, which has initiated the interrupt or alert signal, is stored in the flag bit slope_sign_int_*, only if the corresponding enable bit slope_int_ is set. slope_sign_int_ is 2’s complement coded (0 = positive slope; 1 = negative one).

Slope interrupt is performed, if at least 1 axis is leading to an interrupt (OR-connection). Slope criterion is determined from digital filter output and depends on bandwidth settings: for example for slope_dur = 01b and bandwidth=0111b (1.2 kHz), \( 2^\text{bandwidth} = 2.4 \text{ksamples/s} \) leads to reaction for interrupt activation of \( 3 \times 417 \mu s = 1.25 \text{ms} \) and a minimum slope interrupt duration of \( 3 \times 417 \mu s = 1.25 \text{ms} \).

If lower bandwidth is selected

i) the digitally filtered values (lower noise) are taken for the verification of the any motion criterion and

ii) the time scale to evaluate the criterion is stretched. Thus adjusting the bandwidth, the slope threshold, the slope duration as well as the full scale range enables to tailor the sensitivity of the slope algorithm.

### 7.8.7 Tap sensing

#### 7.8.7.1 General Explanation

BMA180 is providing a possibility to detect 2 consecutive slope events of a device and may generate an interrupt.

#### 7.8.7.2 Tap sensing interrupt configuration parameters and settings

The following configuration parameters/settings are provided (all unsigned integer)

- \( \text{tapsens\_int} \): This bit enables the tapsens_criteria to generate an interrupt
- \( \text{tapsens\_th} \): defines the threshold level of the tip-shock.
- \( \text{tapsens\_int\_x} \): defining if tap sensing event on x-axis should generate interrupt
- `tapsens_int_y`: defining if tap sensing event on y-axis should generate interrupt
- `tapsens_int_z`: defining if tap sensing event on z-axis should generate interrupt
- `tapsens_filt`: evaluation if interrupt generation is done with filtered (`tapsens_filt = “1”`) or unfiltered (`tapsens_filt = “0”`) acceleration signal.

  If `tapsens_filt = 1`, the signal is filtered and the slope condition(s) depend on `bw` settings. Thus, \( n = \frac{1200}{BW} \)

  If `tapsens_filt = 0`, the signal is unfiltered and the slope condition(s) depend only on the maximum bandwidth. Thus, \( n = 1 \).

- `tapsens_dur`: `tapsens_dur` (threshold duration) defines the maximum delay between 2 acceleration slope detections. The values of `tapsens_dur` are defined below

<table>
<thead>
<tr>
<th><code>tapsens_dur&lt;2:0&gt;</code></th>
<th>Mode duration</th>
<th>Mode duration [ms] [(low noise mode)]</th>
<th>Mode duration [ms] [(low power mode)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>120*Tupdate</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>001</td>
<td>180*Tupdate</td>
<td>75</td>
<td>37,5</td>
</tr>
<tr>
<td>010</td>
<td>240*Tupdate</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>011</td>
<td>360*Tupdate</td>
<td>150</td>
<td>75</td>
</tr>
<tr>
<td>100</td>
<td>600*Tupdate</td>
<td>250</td>
<td>125</td>
</tr>
<tr>
<td>101</td>
<td>1200*Tupdate</td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td>110</td>
<td>1800*Tupdate</td>
<td>750</td>
<td>375</td>
</tr>
<tr>
<td>111</td>
<td>2400*Tupdate</td>
<td>1000</td>
<td>500</td>
</tr>
</tbody>
</table>

- `tapsens_shock`: if slope is detected within `tapsens_shock`, no interrupt is generated

### 7.8.7.3 Tap sensing interrupt: algorithm

An acceleration slope is detected when the criterion `tapsens_th_criteria` is set.

The tap sensing feature is using two acceleration signal slope detections. The first slope detection sets the status bit `first_tapsens_s` to ‘1’. An interrupt signal is generated only if new slope detection comes after `tapsens_shock = (120*Tupdate) = 50 ms and before `tapsens_dur`.

`first_tapsens` is reset when at least one of the following conditions is true:

- tap sensing feature is disabled during the processing of tap sensing sequence.
- `tapsens_dur` period is passing.
- tap sensing interrupt occurs.

The sign of the slope, which has initiated the interrupt signal, is stored in the flag bit `tapsens_sign_int_*`, only if the corresponding enable bit `tapsens_int_*` is set.
Tap sensing function is defined with an OR condition. For example, if all axes are selected (tapsens_int_x= tapsens_int_y= tapsens_int_z='1'), the procedure could start with a pulse on the X-axis and finish with a pulse on the Z-axis.

The procedure starts always with the first detected pulse.

![Diagram of tap sensing detection with use of tapsens_th, tapsens_dur]

**Figure 9:** example of tap sensing detection with use of tapsens_th, tapsens_dur

### 7.9 Performance Settings

As Performance Settings Offset, Gain (sensitivity), TCO and TCS are considered.

#### 7.9.1 Gain trimming (sensitivity trimming)

Gains of the sensor (temperature and acceleration for x-, y- and z-axis) are calibrated at production line. Corresponding bit widths are shown below:

- **gain_t**: Gain trimming for temperature (5 bits).
- **gain_z**: Gain trimming for Z axis (7 bits).
- **gain_y**: Gain trimming for Y axis (7 bits).
- **gain_x**: Gain trimming for X axis (7 bits).

The above codes are “offset binary” coded. For instance for gain_x trimming code 1000000 is the middle code for no trimming and codes for most negative correction to most positive one are: 0000000, 0000001... 0111111, 1000000, 1000001... 1111110, 1111111.
Attention:
Customer is able to recalibrate sensitivity. If this is done including EEPROM writing, the initial calibration values are lost, thus care has been taken.

7.9.2 Offset trimming
Offsets of the sensor (temperature and acceleration for x-, y- and z-axis) are calibrated at production line. Corresponding bit widths are shown below:

\[
\begin{align*}
offset_{t}: & \quad \text{Offset trimming for temperature (7 bits).} \\
offset_{z}: & \quad \text{Offset trimming for Z axis (12 bits).} \\
offset_{y}: & \quad \text{Offset trimming for Y axis (12 bits).} \\
offset_{x}: & \quad \text{Offset trimming for X axis (12 bits).}
\end{align*}
\]

The above codes are “offset binary” coded. For instance for offset_z trimming of z-axis, code 100000000000 is the middle code for no trimming and codes for most negative correction to most positive one are: 000000000000, 000000000001... 011111111111, 100000000000, 100000000001... 111111111110, 111111111111.

Attention:
Customer is able to recalibrate sensitivity. If this is done including EEPROM writing, the initial calibration values are lost, thus care has been taken.

7.9.3 Offset_finetuning
7.9.3.1 General Explanation
The sensor is providing a possibility to regulate offsets down to very small values. This can be done for each axis separately. Remaining offsets are typically below 5 mg, the value of which are depending on the fine-tuning mode (see below).

If z-axis signal should remain at +1g equivalent (e. g. 4096 LSB at 1g in 2g-mode), offset tuning must be done in 0g-position of z-channel (e. g. by turning sensor). Other method to optimize offset in +1g position of z-channel is to recalculate offset without using fine-tuning for z-channel but offset_z register (in 2g-mode 1 bit change in offset_z is equivalent to approx. 27 LSB_ADC = approx. 7 mg step-size). Thus offset can be appropriately readjusted. Of course in this position sensitivity error is compensated as additional “offset error”.

7.9.3.2 Configuration parameters and settings for offset fine-tuning
Offset fine-tuning method is controlled via a 2-bit register, called offset_finetuning, the regulation procedure itself is enabled by 3 bits called en_offset_*. The definition of the offset_finetuning register is the following:

<table>
<thead>
<tr>
<th>offset_finetuning&lt;1:0&gt;</th>
<th>Offset regulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>no action</td>
</tr>
<tr>
<td>01 (default)</td>
<td>fine calibration</td>
</tr>
<tr>
<td>10</td>
<td>coarse calibration</td>
</tr>
<tr>
<td>11</td>
<td>full calibration</td>
</tr>
</tbody>
</table>
Setting these two bits enables offset regulation and disables automatically low interrupt function (\textit{low\_int} = ‘0’).

The offset cancellation function has two sub-functions:

- **Coarse calibration:** This one is built-in to correct up to +/-1g, in addition to the standard offset correction of the sensor (see section before). E. g. using this calibration method could eliminate the 1 g offset of the z-axis in applications where optimum full scale measurements are necessary – e. g. high accurate tilt measurements with high resolution in 1 g mode, where all 3 acceleration signals are used. Coarse calibration is done via DAC inside the sensor, thus final remaining offset is same as after standard calibration (see table 2).

- **Fine calibration:** the fine calibration is done in the digital part of the sensor and allows reaching an offset correction with a step size of 1 LSB\textsubscript{adc} and a range of +/-64 LSB\textsubscript{adc}, that means 7 bits per channel, called \textit{fine\_offset\_*} bits ("2’s complement" coded). The following table defines the correspondence between the \textit{fine\_offset\_*} bits and the low interrupt bits (in offset-tuning mode the \textit{fine\_offset\_*} bits are stored in the below mentioned \textit{low\_*} bits).

<table>
<thead>
<tr>
<th>\textit{fine_offset_*} bits</th>
<th>Correspondance to \textit{low int} registers</th>
<th>Register</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{fine_offset_x}</td>
<td>low_th</td>
<td>29h</td>
<td>LSB to ‘0’</td>
</tr>
<tr>
<td>\textit{fine_offset_y}</td>
<td>low_dur</td>
<td>26h</td>
<td></td>
</tr>
<tr>
<td>\textit{fine_offset_z}</td>
<td>low_int_x_y_z_filt_hy&lt;2&gt;_hy&lt;3&gt;_hy&lt;4&gt;</td>
<td>25h_23h</td>
<td></td>
</tr>
</tbody>
</table>

- The full calibration is the combination of these both calibrations.

### 7.9.3.3 Offset fine-tuning algorithm

The procedure to run the offset calibration is the following:

1. Set \textit{offset\_finetuning} to a value different to ‘00’. The low interrupt function is consequently disabled if the \textit{offset\_finetuning} bit 0 is set to ‘1’.

2. Set the appropriate \textit{en\_offset\_*} bit(s) to ‘1’ corresponding to the chosen axis.

Remarks:

- If more than one bit is set to ‘1’, only one of the selected axis will be tuned.
- If any of the bits is set to ‘1’ during the offset regulation is in progress, the corresponding bit will be set to ‘1’ but the regulation currently in progress is not disturbed.
- As soon as the regulation is done, all bits are reset to ‘0’ by the sensor itself.
Once the procedure (step 1+2, see above) is completed, the offset calibration sequence starts:

a) Coarse calibration is performed and new offset codes are stored in the appropriate offset_* image register, corresponding to the chosen axis in step 2 (see above). This calibration is performed only if offset_finetuning = ‘10’ or ‘11’.

b) Fine calibration is performed using internal averaging to achieve best accuracy (noise reduction). In fact, the result of the average corresponds to the fine_offset code (offset value), which is stored in the appropriate fine_offset_* image register, corresponding to the chosen axis in step 2 (see above). If an error occurs, the code of this register is either ‘0111111’ in the case of acceleration value is positive or ‘1000000’ otherwise. This calibration is performed only if offset_finetuning = ‘01’ or ‘11’.

c) Status bit offset_st_s is set and a pulse interrupt signal of 1*Tupdate length is generated, indicating that the offset cancellation sequence is finished. This bit can be used as an indication signal to the μC for finalization of the offset regulation procedure. Subsequently the EEPROM writing may occur.

Remarks:
- EEPROM writing has to be performed by the user.

- After completion of the offset regulation/storage process, offset fine-tuning bits have to be reset to "00" to re-enable the low-interrupt function. There are different cases to be distinguished, if low-g interrupt should be used and offset should be kept as regulated after EEPROM writing (and thus before resetting offset fine-tuning).
  - Coarse calibration (via DAC): after EEPROM writing offset_finetuning can be reset to ‘00’. Course offset is remaining as calibrated and low-g interrupt settings can be changed to use low-g interrupt without influencing remaining offset.
  - Fine tuning (changing of ADC output and not via DAC): after EEPROM writing fine_offset_* image registers are written to the corresponding low-g-interrupt registers. If offset_finetuning is reset to ‘00’, low-g interrupts can be used, but offset is not “fine tuned” any more (fine_offset_* registers are overridden by low-g interrupt settings in image register; cancellation is done in digital part using the low-g interrupt image registers and not EEPROM registers). If coarse calibration has been performed before, sensor stays at least within the coarse calibration values. Thus working of offset fine-tuning and low-g interrupt functionality at the same time is not possible.
  - If sensor was fine-tuned and low-g interrupt functionality is not necessary any more, very small offset could be achieved by update_image procedure (in this case EEPROM content is copied into image register and thus fine_offset_* registers are copied into image registers, if offset-finetuning is ’11’ or ’01’).

- Calibration depends on bw. By changing the bw, an offset of 1 or 2 LSBs may be induced.
• Precondition for a minimized offset is the optimum position of the end consumer device (including the sensor) with respect to the g-axis orientation. Any angular mismatch between the sensor package and gravity vector (“0” degree for z-axis and “+/−90 degree for x- and y-axis) are leading to offset errors, which are related to a position mismatch during calibration. They are not due to a bad offset of the sensor itself.

• In order to regulate all offsets, set offset fine-tuning e. g. to “11” and then sequentially enable en_offset_* bits. To optimize waiting time, offset_st_s could be checked. Another solution is a frequent check using a software counter in the µC. Third possibility is waiting for a certain time before enabling offset regulation of another axis.

7.9.4 tc0_x, tc0_y and tco_z
These 18 bits (6 bits each axis) are used to realize the temperature compensation of the offset on each axis. This compensation is directly done in the digital part.

TCO-correction for each channel could be min. -1.6 mg/K (all TCO-bits “0”) to compensate a sensor with maximum TCO, 0 mg/K (TCO-code at middle code = 32) for no TCO-compensation and max. +1.6 mg/K (TCO-code = 64) for compensating a sensor with min TCO. Step-size is 0.05 mg/K, thus all intermediate steps could be calculated easily.

7.9.5 tco_range
By setting this bit to ‘1’, TCO range changes from +/-1.6 mg/K to +/-6.4 mg/K, TCO step size changes from +/-0.05 mg/K to +/-0.20 mg/K.

7.9.6 tcs, tcs_only_z
The 4 tcs-bits are used to provide a temperature compensation of the sensitivity for all axes (trimming range: -4% … +3.5% for the whole temperature range). All 3 g-axes are compensated identically, if tcs_only_z = 0 (this bit is hidden in Bosch reserved area of EEPROM). In this case a different trimming for a different axis is not possible. If tcs_z_only = ‘1’, tcs is only influencing z-axis. Setting of tcs_z_only is done in Bosch production line. Typically tcs_z_only is set to ‘1’, thus only TCS of z-axis can be recalibrated.

The compensation is done with respect to room temperature (approx. 25°C). The devices are pre-trimmed in production line, but if lowest TCS is necessary, trimming after soldering might optimize TCS. This trimming could be performed in-line by device manufacturer.
The following table informs about the signal correction by using the tcs-bits (correction corresponds to a full temperature range correction from -40 °C up to +85 °C. As a consequence a correction of approx. +/-2 % with respect to room temperature (+25 °C) is possible.

<table>
<thead>
<tr>
<th>tcs&lt;3:0&gt;</th>
<th>Full temp (-40°C to +85°C) correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-4.0%</td>
</tr>
<tr>
<td>1</td>
<td>-3.5%</td>
</tr>
<tr>
<td>2</td>
<td>-3.0%</td>
</tr>
<tr>
<td>3</td>
<td>-2.5%</td>
</tr>
<tr>
<td>4</td>
<td>-2.0%</td>
</tr>
<tr>
<td>5</td>
<td>-1.5%</td>
</tr>
<tr>
<td>6</td>
<td>-1.0%</td>
</tr>
<tr>
<td>7</td>
<td>-0.5%</td>
</tr>
<tr>
<td>8</td>
<td>0.0%</td>
</tr>
<tr>
<td>9</td>
<td>0.5%</td>
</tr>
<tr>
<td>10</td>
<td>1.0%</td>
</tr>
<tr>
<td>11</td>
<td>1.5%</td>
</tr>
<tr>
<td>12</td>
<td>2.0%</td>
</tr>
<tr>
<td>13</td>
<td>2.5%</td>
</tr>
<tr>
<td>14</td>
<td>3.0%</td>
</tr>
<tr>
<td>15</td>
<td>3.5%</td>
</tr>
</tbody>
</table>

Example:
Measured sensitivity is changed by -1 % from +25 °C to +85 °C. This is equivalent to -2 % for full temperature range. Thus a correction of +2 % for full temperature range is necessary. This can be achieved by choosing tcs <3:0> = “12” or 1100b.

7.10 Control registers description
All single control bits are active at 1, a truth table is provided for commands coded on more than 1 bit.

7.10.1 reset_int
This is a one-shot control bit. The behaviour of reset_int is as follows:
   a) it is accepted if the appropriate interrupt is latched and generated. In this case, reset_int event resets the interrupt state to not generate.
   b) it is ignored if the appropriate interrupt is not latched or if this one is latched but not generated.

7.10.2 update_image
When this bit is set to 1b, an image update procedure is started: all EEPROM content is copied to image registers and the bit update_image is turned to 0 when the procedure is finished.

No write or read to image registers and no EEPROM write is allowed during the update from EEPROM.

An automatic update image procedure also occurs:
   a) after Power On reset
   b) after soft_reset is issued via the serial interface.
7.10.3 ee_w
This bit must first be written to 1 to be able to write anything into image registers (20h .. 3Bh).

I2C acknowledgement procedure for protected/non-protected area:
   a) I2C slave address: if correct, BMA180 sets acknowledge.
   b) I2C register address (I2C write): BMA180 sets acknowledge for both unprotected and protected registers.
   c) I2C write data (I2C write): BMA180 sets acknowledge for both unprotected and protected registers; no write is done for protected register.
   d) I2C read data (I2C read): acknowledge is set by a master; no error detection is possible.

After power on reset or soft reset, ee_w = 0.

7.10.4 st1
This self test bit does not generate any electrostatic force in the MEMS but is used to verify, whether the digital part is working correctly and that microprocessor is able to react to the interrupts. Basically a 0 g acceleration is emulated, and the user can detect the whole logic path for interrupt, including the PCB path integrity. The low_th interrupt register must be set by user so that st1 generates a low threshold interrupt (e. g. low_th -> 0.4g, low_dur = 0 ms).

7.10.5 st0
The self-test command uses electrostatic forces to move the MEMS common electrode. Self-test can be used only with highest bandwidth setting so whatever is the setting defined by user, the internal mode corresponds to \(bw = 0111\) if \(st0 = 1\). No acceleration change shall occur during self-test procedure and no fine offset compensation is performed during the self-test.

As soon as \(st0\) is set, the self-test sequence starts and an acceleration of typically about +/-0.2g for each channel is emulated (this acceleration is summed with the real acceleration, as long as the sum result stays inside the full scale range). The internal procedure (deflection, measurement, etc.) takes in total less than 10 ms, thus data acquisition must be fast.

A schematic view of this procedure is shown below.
After end of self test procedure, $st0$ is written by sensor itself to 0 ($st0$ stays at 1 as long as the self test procedure is running, thus this bit can be read-out to detect if self-test is finished).

Self test response has to be determined as follows:

- measurement of $x$-, $y$- and $z$-accelerations ($x_{nom}$, $y_{nom}$, $z_{nom}$)
- initiate self test via setting $st0$ to “1”
- measurement of $out_x$, $out_y$ and $out_z$ for in total approx. 5 ms; store all measured samples (in low noise mode: approx. 12 samples for each channel) in $\mu$C.
- check if $st0$ is “0”. If not, increase measurement time by 2 ms and repeat measurement.
- calculate maximum and minimum values of measured samples and determine positive and negative self test responses as below
  - $self_pos_x = |x_{max} - x_{nom}|$
  - $self_pos_y = |y_{max} - y_{nom}|$
  - $self_pos_z = |z_{max} - z_{nom}|$
  - $self_neg_x = |x_{nom} - x_{min}|$
  - $self_neg_y = |y_{nom} - y_{min}|$
  - $self_neg_z = |z_{nom} - z_{min}|$
- self test is o. k., if
  - $\max(self_pos_x, self_neg_x) > 200$ LSB
  - $\max(self_pos_y, self_neg_y) > 200$ LSB
  - $\max(self_pos_z, self_neg_z) > 200$ LSB

A soft-reset is recommended after each self-test sequence.

The above self test responses are just roughly indicating mechanical functionality of the sensor element due to large variants in the self test response amplitudes from sensor to sensor. A much better assessment is possible, if “vector-sum” of all accelerations is used to determine functionality of the sensor. The exact procedure how to do this is given in a separate application note. This is available by Bosch Sensortec. Please ask your representative.

**7.10.6 soft_reset**

BMA180 is reset each time the value B6h is written to this byte. The effect is identical to power-on reset. Control, status and image registers are reset to values stored in the EEPROM. After soft_reset or power-on reset BMA180 comes up in standard mode or wake-up mode. It is not possible to boot BMA180 to sleep mode.
No serial transaction should occur within 10 μs after soft_reset command.

7.10.7 sleep
This bit turns the sensor IC in sleep mode, no acceleration measurements can be performed any more, but control and image registers are not cleared.

When BMA180 is in sleep mode no operation can be performed without waking-up the sensor IC by setting sleep=0 or soft_reset. As a consequence all write and read operations are forbidden when the sensor IC is in sleep mode except command used to wake up the device or soft_reset command.

After sleep mode removal, it takes 1ms to obtain stable acceleration values (>99% data integrity). User must wait for 10ms before first EEPROM write. For the same reason, BMA180 must not be turned in sleep mode when any update_image, self_test or EEPROM write procedure is on going.

Attention: This bit should not be set to “1”, when wake-up mode is enabled.

7.10.8 dis_wake_up
When dis_wake_up = 1, wake-up mode is disabled in order to avoid the fact that the ASIC may enter into sleep mode before EEPROM writing is initiated.

7.10.9 en_offset_x, en_offset_y, en_offset_z
These one-shot control bits enable the offset regulation for the corresponding axis. To regulate all axis, it is necessary to enable the bits sequentially.

7.11 Status register
7.11.1 first_tap_sensing
This status bit is set when a first tap sensing shock has been detected. This bit is reset when at least one of the following conditions is true:
- tap sensing feature is disabled during the processing of tap sensing sequence.
- tapsens_dur period is passed.
- tap sensing interrupt occurs.

7.11.2 Slope alert
Slope alert is a feature, which is described in a separate application note (under construction).

7.11.3 low_th_int, high_th_int, slope_int_s, tapsens_int
These latched status bits are set when the corresponding criteria have been issued. When several interrupt modes are enabled, these bits can be used by microprocessor to detect which criteria generated the interrupt.
If interrupts are not latched, status bits *_int are the same as the corresponding bits *_s, which are defined in 7.11.4.
Disabling of an interrupt (e. g. setting low_th_int to ‘0’) shall not reset active latched interrupt status bit (e. g. low_th_int remains at ‘1’ until a reset is performed by setting reset_int to ‘1’). Changing to interrupt mode to non-latched (setting lat_int to ‘0’) shall immediately reset all latched interrupt status bits.

7.11.4 low_th_s, high_th_s, slope_s, tapsens_s, offset_st_s
These status bits are set when the corresponding criteria have been issued; they are automatically reset by BMA180 when the criteria disappear or if the corresponding interrupt is a latched one and user issues reset_int.

7.11.5 offset_st_s
This status bit is set either at the end of offset regulation’s sequence or at the end of each data acquisition’s phase of the selftest; it is automatically reset by BMA180 after 1*Tupdate.

7.11.6 x_first_int, y_first_int, z_first_int
These latched status bits can be used by microprocessor to detect on which axis any interrupt occurs first after either system reset or reset_int event.

7.11.7 Status bits for acceleration or slope sign
These latched status bits can be used by μC to know the sign of the acceleration or the slope which has initiated an interrupt or alert signal (‘0’ for a positive sign, ‘1’ for a negative one).

If the INT pad shall be asserted, sign bits are updated. The bits corresponding to the disabled axes are set to ‘0’, the other ones are set to the corresponding sign. If the whole interrupt has been disabled, all the appropriate sign bits are set to ‘0’. If just one axis is disabled/enabled, the appropriate sign bit is not touched; it is updated as soon as there is a generated interrupt.

Latched status registers can only be reset by power-on reset or soft-reset.

7.11.8 ee_write
This bit is set to ‘1’ if EEPROM writing is in progress. Any writing transaction sent if ee_write = ‘1’ is ignored.

7.12 Data registers
7.12.1 temp
A thermometer is embedded in BMA180, temperature resolution is 0.5 K/LSB_TEMP. Code 80h stands for lowest temperature which is centered around -40°C and typical code for 25°C is 00000010 in 2’s complement. Offset and gain are trimmable like the acceleration axes, thus temperature offset could be adjusted to achieve a range between -40°C and 87.5°C by changing the offset_t register (typical value 88d)

7.12.2 acc_x, acc_y, acc_z
Acceleration values are stored in these registers to be read out through serial interface. The description of the digital signals acc_x, acc_y and acc_z is “2’s complement”, based on 14 bits. The 2 LSB are fixed to 0 if readout_12bit is set to ‘1’.
From negative to positive accelerations, the following sequence for the ±2g measurement range can be observed (all other g-ranges correspondingly):

-2.00000 g : 10 0000 0000 0000  
-1.99975 g : 10 0000 0000 0001  
...  
-0.00025 g : 11 1111 1111 1111  
0.00000 g : 00 0000 0000 0000  
+0.00025 g : 00 0000 0000 0001  
...  
+1.99950 g : 01 1111 1111 1110  
+1.99975 g : 01 1111 1111 1111

Data is periodically updated with values from the digital filter output. LSB acceleration bytes must be read first. After an acceleration LSB byte read access, the corresponding MSB byte update can optionally be blocked until it is also accessed for read (in fact, MSB content is copied in a shadow register and MSB access is re-directed to this copy which is not affected by updates). Thus, MSB overflow can be avoided.

It is not possible to read-out only MSB bytes if shadow_dis=0, an LSB byte must first be read out (if not, shadow register is never updated and MSB value is always identical to first read value). To be able to read out only MSB byte, shadow_dis must be written to 1.

new_data_* flags at bit position 0 of acc_x_lsb, acc_y_lsb and acc_z_lsb can also be used to detect if acceleration values have already been read out.

If systematic acceleration values read out is planned, new data interrupt should be used. Each time all temperature + 3 axes values have been updated, INT goes high and microcontroller must read out data. With this method, microcontroller accesses are synchronized with internal BMA180 updates (see picture below).

Figure 10: ADC conversion sequence and synchronization with read-out of acceleration values
Synchronization of read-out sequences with internal ADC conversions has 2 goals:
1. it enables a constant phase shift between acceleration value and its digital corresponding value read by microprocessor.
2. noise due to SPI activity perturbation is always generated during the less critical conversion of temperature value. Each ADC conversion takes typically ¼*Tupdate, thus, this is the maximum delay advised to read out acceleration data with lowest noise as possible.

**Remark:**
Without using new-data interrupt, sensor is still in spec, using it is mainly optimizing it.

When acceleration read-out is synchronized by the new_data interrupt feature, each Tupdate only 1 read sequence occurs. Indeed, 4 channels (temperature + 3 axes) are updated between each new data interrupt.

Noise perturbations due to serial interface pad switching should be avoided. This is especially true when many slave ICs are connected on same serial data and clock pins. Much noise could be fed into BMA180 when other slaves are accessed. Thus, to be able to achieve low noise level, no activity on SCK, SDI and SDO should occur excepted to read out acceleration like explained on above chapter.

`new_data_x`, `new_data_y`, `new_data_z` bits are flags which are turned at 1 when acceleration registers have been updated. Reading acceleration data MSB or LSB registers turns the flags at 0. The flag value can be read by microprocessor.

If first SPI transaction is a acc_(_x, y, or z)_LSB byte read, the corresponding MSB byte will always be 0x00 in case of shadow_dis=0. Next read will be correct. To avoid this false first reading, any other SPI read or write sequence should be performed after power on and before first acc_(_x,y, or z)_lsb byte read.

### 7.12.3 al_version<3:0>, ml_version<3:0>, chip_id<2:0>

`al_version<3:0>` and `ml_version<3:0>` are used to identify the chip revision.

`chip_id<2:0>` is used by customer to be able to distinguish BMA180 from other chips which would have same serial interface. This code is fixed to 011b.
8  I²C and SPI-Interfaces

8.1  Specification of Interface parameters

<table>
<thead>
<tr>
<th>Interface parameters</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input – low level</td>
<td>Vil_si</td>
<td>VDDIO=1.2V to 3.6V</td>
<td>0.3*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Input – high level</td>
<td>Vih_si</td>
<td>VDDIO=1.2V to 3.6V</td>
<td>0.7*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Output – low level</td>
<td>Vol_SDI</td>
<td>VDDIO=1.2V, iol=3 mA</td>
<td>0.2*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Output – low level for 1.2V</td>
<td>Vol_SDI_1.2</td>
<td>VDDIO=1.2V, iol=3 mA</td>
<td>0.23*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Output – high level</td>
<td>Voh_SDO</td>
<td>Vddio=1.62V, ioh=2mA</td>
<td>0.8*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Output – high level for 1.2V</td>
<td>Voh_SDO_1.2</td>
<td>Vddio=1.2V, ioh=2mA</td>
<td>0.62*</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Pull-up resistor</td>
<td>Rpull_up</td>
<td>Internal pull-up resistance to VDDIO</td>
<td>70</td>
<td>120</td>
<td>190</td>
<td>kOhm</td>
</tr>
<tr>
<td>Pull-down resistor</td>
<td>Rpull_down</td>
<td>Internal pull-down resistance to VSS1</td>
<td>12</td>
<td>20</td>
<td>32</td>
<td>kOhm</td>
</tr>
<tr>
<td>I²C bus load capacitor</td>
<td>Cb</td>
<td>On SDI and SCK</td>
<td>400</td>
<td></td>
<td></td>
<td>pF</td>
</tr>
</tbody>
</table>

8.2  Interface selection

2 interface protocols are possible:

- When CSB=1, I²C functionality is enabled within state-machine.
- When CSB=0, 4-wire SPI is enabled.
8.3 I²C Interface

8.3.1 I²C Timings

The I²C slave interface is compliant with Philips I²C Specification version 2.1 (January 2000). Only internal hold time is typically 200 ns. All modes (standard, fast, high speed) are supported. SDI and SCK pins are not pure open-drain (they are diodes to VDDIO).

![I²C timing diagram](image)

**Figure 11: I²C timing diagram**

The BMA180 I²C slave address is coded on 7 bits. The first 6 bits are defined by the Sensor itself, they are fixed. The last bit (LSB) is fixed by the value on SDO used as a digital input. Thus by default the I²C address is either 40h for SDO-connection to VSS or 41h for SDO-connection to VDDIO.

The I²C bus uses the 2 wires SCK (Serial Clock) and SDI (Serial Data Input). CSB is connected to internal pull-up and must not be connected to ground (GND). SDI is bi-directional with pull-down open drain: it must be externally connected to VDDIO via a pull up resistor.

**Table 1: I²C bus terminology**

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter</td>
<td>the IC which sends data to the bus</td>
</tr>
<tr>
<td>Receiver</td>
<td>the IC which receives data from the bus</td>
</tr>
<tr>
<td>Master</td>
<td>the IC which initiates a transfer, generates clock signals and terminates the transfer (microcontroller in final application or tester during calibration procedure)</td>
</tr>
<tr>
<td>Slave</td>
<td>the IC addressed by a master (BMA180)</td>
</tr>
</tbody>
</table>
8.3.2 Start and stop conditions

Data transfer begins by a falling edge on SDI when SCK is at high level, this is the start condition (S), initiated by the \( \text{I}^2\text{C} \) bus master. It is stopped with a rising edge on SDI when SCK is at high level; this is the stop condition (P).

![Figure 12: \( \text{I}^2\text{C} \) start and stop conditions](image)

8.3.3 Bit transfer

One data bit is transferred during each SCK pulse. The data on the SDI line must remain stable during the high period of SCK pulses, as any changes at this time would be interpreted as start or stop conditions. Data is transferred with MSB first.

![Figure 13: \( \text{I}^2\text{C} \), 1 bit transfer](image)
8.3.4 Acknowledge

After a start condition, data bits are transferred to BMA180. Each byte is followed by an acknowledge bit: the transmitter lets the SDI line high (no pull down) and generates a high SCK pulse; if transfer concerns the BMA180 slave receiver and has performed correctly, it generates a low SDI level (pull down activated). After acknowledge, BMA180 lets SDI line free, enabling the transmitter to continue transfer or to generate a stop condition.

![Diagram of SDI line acknowledgement](image)

**Figure 14:** acknowledgement on SDI line

8.3.5 I²C protocol

After a start condition, the slave address + RW bit must be send. If the slave address does not match with BMA180 one, there is no acknowledgement and the following data transfer will not affect the chip. If the slave address corresponds to BMA180 one, it will acknowledge (pull SDI down during 9th clock pulse) and data transfer is enabled. The 8th bit RW sets the chip in read or write mode, RW=1 for reading, RW=0 for writing.

After slave address and RW bit, the master sends 1 control byte:
- the 7-bit register address
- 1 dummy bit

When IC is accessed in write mode, sequences of 2 bytes (= 1 control byte to define which address will be written and 1 data byte to fill it) must be send:

The following transactions are supported:
Single byte read, Single byte write, Multiple byte read, Multiple byte write.
Transactions are described in the following figures.
Abbreviations:

S Start
P Stop
ACKS Acknowledge by slave
ACKM Acknowledge by master
NACKM Not acknowledge by master

**Figure 15:** I²C multiple write

To be able to access registers in read mode, first address should first be send in write mode. Then a stop and a start conditions are issued and data bytes are transferred with automatic address increment:

**Figure 16:** I²C multiple read

Figure 16 shows a typical I²C transfer to read out acceleration data. Address register is first written to BMA180, the RW=0 (lowest acceleration data located to address 02h). I²C transfer is stopped and restarted with RW=1, address are automatically incremented; 2 bytes are sequentially read out.
8.4 SPI Interface (4-wire)

8.4.1 SPI protocol

The SPI interface has a polarity = 1 and SPI phase = 0.

CSB is active low. Data on SDI is latched by BMA180 at SCK rising edge and SDO is changed at SCK falling edge. Communication starts when CSB goes to low and stops when CSB goes to high; during these transitions on CSB, SCK must be high. When CSB=1, no SDI change is allowed when SCK=1 (to avoid any wrong start or stop condition for I²C interface).

![Diagram](image17.png)

**Figure 17:** 4-wire SPI sequence

When write is required, sequences of 2 bytes are required: 1 control byte to define the address to be written and the data byte:

![Table](image18.png)

**Figure 18:** SPI multiple write

When read is required, the sequence consists in 1 control byte to define first address to be read followed by data bytes. Addresses are automatically incremented.
8.4.2 SPI Timings

<table>
<thead>
<tr>
<th>4-wire SPI timings</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI clock input frequency</td>
<td>Fspi_4 slow</td>
<td>VDDIO &gt; 1.6V</td>
<td>10</td>
<td>MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fspi_4 slow</td>
<td>VDDIO &lt; 1.6V</td>
<td>7.5</td>
<td>MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCK low pulse</td>
<td>Tlow_sck_4</td>
<td></td>
<td>20</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>SCK high pulse</td>
<td>Thigh_sck_4</td>
<td></td>
<td>20</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>SDI setup time</td>
<td>Tsetup_sdi_4</td>
<td></td>
<td>20</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>SDI hold time</td>
<td>Thold_sdi_4</td>
<td></td>
<td>20</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>SDO output delay</td>
<td>Tdelay_sdo_4</td>
<td>25 pF load</td>
<td>30</td>
<td>ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSB setup time</td>
<td>Tsetup_csb_4</td>
<td></td>
<td>20</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>CSB hold time</td>
<td>Thold_csb_4</td>
<td></td>
<td>40</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

Figure 20: SPI timing
9 Pinning

9.1 Pin Configuration (top view)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Digital Analog</th>
<th>Description</th>
<th>SPI</th>
<th>I²C</th>
<th>no µC</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>reserved</td>
<td>Digital</td>
<td>Do Not Connect (internally connected to VDD)</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>02</td>
<td>V_DD</td>
<td>Power</td>
<td>Supply Voltage</td>
<td>V_DD</td>
<td>V_DD</td>
<td>V_DD</td>
</tr>
<tr>
<td>03</td>
<td>V(SS)</td>
<td>GND</td>
<td>Ground Connection</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>04</td>
<td>INT</td>
<td>Digital out</td>
<td>Interrupt PIN, active high</td>
<td>INT/NC</td>
<td>INT/NC</td>
<td>INT</td>
</tr>
<tr>
<td>05</td>
<td>CSB</td>
<td>Digital in</td>
<td>Chip Select, active low</td>
<td>CSB</td>
<td>V_DDIO</td>
<td>V_DD</td>
</tr>
<tr>
<td>06</td>
<td>reserved</td>
<td>Digital in</td>
<td>Do Not Connect (internally connected to VSS)</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>07</td>
<td>SCK</td>
<td>Digital in</td>
<td>Serial Clock</td>
<td>SCK</td>
<td>SCK</td>
<td>GND</td>
</tr>
<tr>
<td>08</td>
<td>SDO</td>
<td>Digital in/out</td>
<td>SPI output (4 wire) or Set-up of I²C address</td>
<td>SDO</td>
<td>GND or V_DDIO</td>
<td>GND</td>
</tr>
<tr>
<td>09</td>
<td>SDI</td>
<td>Digital in/out</td>
<td>SPI input or I²C serial data</td>
<td>SDI</td>
<td>SDA</td>
<td>GND</td>
</tr>
<tr>
<td>10</td>
<td>V-DDIO</td>
<td>Power</td>
<td>Supply voltage Connection Digital</td>
<td>V-DDIO</td>
<td>V-DDIO</td>
<td>V_DD</td>
</tr>
<tr>
<td>11</td>
<td>reserved</td>
<td>Digital</td>
<td>Do Not Connect (internally connected to VSS)</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>12</td>
<td>reserved</td>
<td>Digital</td>
<td>Do Not Connect (internally connected to VSS)</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
</tbody>
</table>
**Figure 20:** Connection diagram for use with 4-wire SPI interface

![Connection diagram for use with 4-wire SPI interface](image)

**Figure 21:** Connection diagram for use with I²C interface

![Connection diagram for use with I²C interface](image)
Figure 22: Connection diagram for stand alone use without microcontroller

9.3 External component connection diagram

The following external component(s) are recommended to decouple the power source (voltages are examples).
Or (if $V_{DDIO} = V_{DD}$)

- INT
- CSB
- SCK
- MISO
- MOSI

$V_{DDIO}$
$V_{DDA}$
$V_{SS}$

$C_1 = 22 \, \text{nF}$

2.4V
10 Package

10.1 Outline Dimensions
The sensor housing is a standard LGA package. It is compliant with JEDEC Standard MO-220C. Outline dimensions are shown below.

10.2 Orientation: polarity of the acceleration output
If the sensor is accelerated into the indicated directions, the corresponding channels will deliver a positive acceleration signal (dynamic acceleration).

Example: If the sensor is at rest or at uniform motion in a gravity field according to the figure given below, the output signals are:

- ± 0g for the X channel
- ± 0g for the Y channel
- + 1g for the Z channel
The following table lists all corresponding output signals on $Ax$, $Ay$, and $Az$ while the sensor is at rest or at uniform motion in a gravity field under assumption of a top down gravity vector as shown above.

<table>
<thead>
<tr>
<th>Sensor Orientation (gravity vector)</th>
<th>Output Signal $Ax$</th>
<th>Output Signal $Ay$</th>
<th>Output Signal $Az$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0g</td>
<td>+1g</td>
<td>-1g</td>
<td>0g</td>
</tr>
<tr>
<td>+1g</td>
<td>0g</td>
<td>+1g</td>
<td>0g</td>
</tr>
<tr>
<td>-1g</td>
<td>0g</td>
<td>0g</td>
<td>+1g</td>
</tr>
<tr>
<td>0g</td>
<td>0g</td>
<td>0g</td>
<td>-1g</td>
</tr>
</tbody>
</table>

10.3 Marking

10.3.1 Mass Production Samples

<table>
<thead>
<tr>
<th>Sensor Label</th>
<th>Name</th>
<th>Symbol</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product number</td>
<td>053</td>
<td>Coded alphanumerically</td>
</tr>
<tr>
<td></td>
<td>Sub-con ID</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>053</td>
<td>Date code</td>
<td>YWW</td>
<td>Y: year (alpha-numerical 9=2009, A=2010) WW: Calendar week, numerical</td>
</tr>
<tr>
<td>AYWW</td>
<td>Lot counter</td>
<td>CCC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pin 1 identifier</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>

10.3.2 Engineering samples

<table>
<thead>
<tr>
<th>Sensor Label</th>
<th>Name</th>
<th>Symbol</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product name</td>
<td>180 BMA180</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eng. Sample ID</td>
<td>E Engineering samples are marked with an “e”</td>
<td></td>
</tr>
<tr>
<td>180e</td>
<td>Sub-con ID</td>
<td>A Coded alphanumerically</td>
<td></td>
</tr>
<tr>
<td>AYWW</td>
<td>Date code</td>
<td>YWW Y: year, alpha-numerical (9=2009, A=2010) WW: Working week, numerical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lot counter</td>
<td>0Cn e.g. $n = 1 \rightarrow = C1$-Sample</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pin 1 identifier</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>
10.4 Landing pattern recommendations
The following PCB design is recommended in order to minimize solder voids and stress acting on the sensing element. All dimensions are given in mm.

10.5 Moisture Sensitivity Level and Soldering
The moisture sensitivity level of the BMA180 sensors corresponds to JEDEC Level 1, see also


and


The sensor fulfils the lead-free soldering requirements of the above-mentioned IPC/JEDEC standard, i.e. reflow soldering with a peak temperature up to 260°C (see: Handling, soldering & mounting instructions)
10.6 RoHS compliancy
The BMA180 sensor IC meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also


10.7 Halogen content
BMA180 is halogen-free. For more details on the analysis results please contact your Bosch Sensortec representative.

10.8 Note on internal package structures
Within the scope of Bosch Sensortec’s ambition to improve its products and secure the product supply while in mass production. For this purpose Bosch Sensortec might qualify additional sources for the LGA package of the BMA180.

While Bosch Sensortec took care that all of the technical package parameters as described above are 100% identical for both sources, there can be differences in the chemical analysis and internal structural between the different package sources.

However, as secured by the extensive product qualification processes of Bosch Sensortec, this has no impact to the usage or to the quality of the BMA180 product.

10.9 Tape and Reel
(see document: BMA180 Handling, soldering & mounting instructions)

10.10 Handling Instruction
Micromechanical sensors are designed to sense acceleration with high accuracy even at low amplitudes and contain highly sensitive structures inside the sensor element. The MEMS can tolerate mechanical shocks up to several thousand g’s. However, these limits might be exceeded in conditions with extreme shock loads such as e.g. hammer blow on or next to the sensor, dropping of the sensor onto hard surfaces etc.

G-forces beyond the specified limits during transport should be avoided; handling and mounting of the sensors have to be in a defined and qualified installation process.

BMA180 has built-in protections against high electrostatic discharges or electric fields; however, anti-static precautions have to be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the supply voltage range. Unused inputs must always be tied to a defined logic voltage level.

10.11 Further handling, soldering and mounting instructions
Further important information on handling, soldering and mounting is given in a separate document “BMA180 Handling, soldering and mounting instructions”.
11 Legal disclaimer

11.1 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.

11.2 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 mal-function 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.

11.3 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.

11.4 Limiting values
Limiting values given are in accordance with the Absolute Maximum Ratings (Chapter 2). Stress above one or more of the limiting values may cause permanent damage to the device. Operation of the device at these or at any other conditions above is not implied. Exposure to limiting values for extended periods may also affect device reliability.
## 12 Document History and Modification

<table>
<thead>
<tr>
<th>Rev. No</th>
<th>Chapter</th>
<th>Description of Modification/Changes</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.x</td>
<td>All</td>
<td>Initial version</td>
<td>31.01.2009</td>
</tr>
<tr>
<td>1.0</td>
<td>All</td>
<td>Additional information, corrections, etc.</td>
<td>06.03.2009</td>
</tr>
<tr>
<td>2.0</td>
<td>All</td>
<td>Additional information, corrections, etc.</td>
<td>09.08.2009</td>
</tr>
<tr>
<td>2.1</td>
<td>1</td>
<td>Updated specification values</td>
<td>10.12.2009</td>
</tr>
</tbody>
</table>