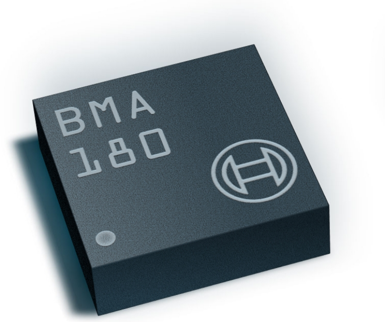
BMA180
Digital, triaxial acceleration sensor

Data sheet

Bosch Sensortec

BOSCH
Invented for life



BMA180 data sheet

Ordering code	Please contact your Bosch Sensortec representative for the ordering code
Package type	12-pin LGA
Data sheet version	2.5
Document release date	07 December 2010
Document number	BST-BMA180-DS000-07
Technical reference code(s)	0 273 141 053
Notes	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.

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 3mm^2 footprint, 0.9mm height
- 256 bit EEPROM for calibration data and customer data
- Low power: typically 650µA current even in 14 bit operation mode
- Very low-voltage operation: +1.62V ... +3.6V for VDD, +1.2V ... +3.6V for VDDIO
- Temperature range: -40°C ... +85°C
- RoHS compliant and halogen free
- 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: 150µg/√Hz in low noise mode
  - 0.25mg ADC-resolution in 2g-mode
- Offset:
  - Offset at room temperature (+25°C): 4mg (incl. fine-offset tuning)
  - Very small Temperature Coefficient of Offset (TCO): 0.25mg/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 ±4mg 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 1g and 16g. A second-order Butterworth filter with switch-able pole-frequencies between 10Hz and 600Hz is included to provide pre-conditioning of the measured acceleration signal. Typical noise level and quantization lead - in 2g-mode - to a resolution of typically 0.5mg and a typical accuracy of below 0.25° in an inclination sensing application, respectively. The current consumption is typically 650µA at a supply voltage of 2.4V 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 mm^2 and a height of 0.9mm.

Table of Contents

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_{ADC} ≈ 0.25 mg in 2 g range; it scales with range (e. g.: 1g-range ->1 LSB_{ADC} is 0.125 mg)

- 1 LSB_{TEMP} = 0.5°C.

Table 1: Operating conditions (unless otherwise specified)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Operating temperature	T_op	-40	25	85	°C
Temperature range for EEPROM writes.	T_ee_w	-40	25	85	°C
Supply voltage (internal bandgap)	V_{DD}	2.0	2.4	3.6	V
Supply voltage for digital interface (VDDIO ≤ VDD; VDD should not be applied after VDDIO)	V_{DDIO}	1.2	2.4	3.6	V
Allowed external regulated voltage, if internal band-gap is by-passed (dis_reg=1)	V_{DD_ext}[1]	1.62	1.8	2.0	V

[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.
Attention: In case of dis_reg='1' continuous operation with VDD>2V may lead to destruction of the sensor.

Table 2: Specification

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
OPERATING RANGE						
Acceleration Ranges	g_{FS1g}	Switchable via serial interface		±1.0		g
	$g_{FS1.5g}$			±1.5		
	g_{FS2g}			±2.0		
	g_{FS3g}			±3.0		
	g_{FS4g}			±4.0		
	g_{FS8g}			±8.0		
	g_{FS16g}			±16.0		
Supply Voltage	V_{DD}	Internal band-gap: Full perform. in T-range: -40°C .. +85 °C; disreg=0	2.0	2.4	3.6	V
		External band-gap: Full perform. in T-range: -40°C .. +85 °C; disreg=1	1.62	1.8	2.0	
Supply Voltage for digital interfaces	V_{DDIO}	$V_{DDIO} \leq V_{DD}$ (timing: V_{DD} should not be applied after V_{DDIO})	1.2	1.8	3.6	V
Supply Current in "Low power" mode	I_{DD}			650		µA
Supply Current in "Low-Noise" mode	I_{DD_LN}			975		µA
Supply Current in Sleep mode	I_{DD_SL}	Sleep mode/no serial interface transfer (T=25°C)		0.5		µA
Operating Temperature	T_OP		-40		+85	°C

OUTPUT SIGNAL						
PARAMETER	**SYMBOL**	**CONDITION**	**MIN**	**TYP**	**MAX**	**UNIT**
ADC resolution		14 bit mode 12 bit mode			14 12	bit bit
Sensitivity (calibration at factory in 2g-range)	S_{1g}	g-range: +/-1.0 g		8192 ±2.0%		LSB_{ADC}/g
	$S_{1.5g}$	g-range: +/-1.5 g		5460 ±2.0%		
	S_{2g}	g-range: +/-2.0 g		4096 ±1.5%		
	S_{3g}	g-range: +/-3.0 g		2730 ±2.0%		
	S_{4g}	g-range: +/-4.0 g		2048 ±2.0%		
	S_{8g}	g-range: +/-8.0 g		1024 ±2.5%		
	S_{16g}	g-range: +/-16 g		512 ±3.0%		
Temperature Coefficient of Sensitivity	TCS_{2g}	T = 25 °C, V_{DD} = 2.4 V 2g range		±0.01		%/K
Temperature Coefficient of Sensitivity	TCS_{4g}	T = 25 °C, V_{DD} = 2.4 V 4g range		±0.01		%/K
Temperature Coefficient of Sensitivity	TCS_{16g}	T = 25 °C, V_{DD} = 2.4 V 16g range		±0.01		%/K
Zero-g Offset[2] ex-factory	Off_ initial	T = 25 °C, V_{DD} = 2.4 V, 2g range		±15		mg
Zero-g Offset[3]	Off_ lifetime	T = 25 °C, V_{DD} = 2.4 V, 2g range		±60		mg
Zero-g Offset[4] (fine tuning)	Off_fine	T = 25 °C, V_{DD} = 2.4 V, 2g range		±5		mg
Zero-g Offset Temperature Drift	TCO_{2g}	T = -40°C - 85°C, V_{DD} = 2.4 V 2g range		±0.5		mg/K
Zero-g Offset Temperature Drift	TCO_{4g}	T = -40°C - 85°C, V_{DD} = 2.4 V 4g range		±0.5		mg/K
Zero-g Offset Temperature Drift	TCO_{16g}	T = -40°C - 85°C, V_{DD} = 2.4 V 16g range		±0.5		mg/K

[2] Ex-factory: no offset tuning, no soldering
[3] Over life-time (incl. MSL1 preconditioning - 168h 85% rH, 85°C - before soldering 3 times), no offset fine tuning.
[4] Over life-time, after offset-fine-tuning in 0g-position

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
DC Power supply rejection ratio	PSRR_DC	in ±2g range, V_{DD} = 2.4 V (dis_reg = 0)		10		LSB_{ADC}/V
AC Power supply rejection ratio	PSRR_AC	With 0.4V peak to peak AC signal on V_{DD} at internal clock frequencies (dis_reg=0)		140		LSB_{ADC}/V
Noise density low power mode	N_LP	Low power mode @ 1200Hz, 2g		200		µg/√Hz
Noise density low noise mode	N_LN	Low noise mode @ 1200Hz, 2g		150		µg/√Hz
Bandwidth[5]	BW_10			10		Hz
	BW_20			20		
	BW_40			40		
	BW_75			75		
	BW_150			150		
	BW_300			300		
	BW_600			600		
	BW_1200			1,200		
	BW_HP	High pass		1		
	BW_BP	Band pass		0.2 - 300		
Nonlinearity	1g 1.5g 2g	best fit straight line		±0.15		%FS
	3g 4g	best fit straight line		±0.25		%FS
	8g 16g	best fit straight line		±0.75		%FS
Temperature sensor bandwidth	BW_temp			275		Hz
Temperature sensor sensitivity	Temp_sens			0.5		K/LSB_{TEMP}
Temperature sensor offset	Temp_off		-5		5	K

[5] In low power mode, BW is divided by 2 (e.g. 10Hz → 5Hz)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
Data output rate	$Rate\_out_{LN}$	Low noise mode		2400		Hz
Data output rate	$Rate\_out_{LP}$	Low power mode		1200		Hz
Wake-up time	Tw_up	BW = 1200 Hz		1.5		ms
Start-up time[6]	Tst_up	BW = 1200 Hz		2.5		ms
Start-up time from sleep mode	Tst_sm	BW = 1200 Hz		2		ms
EEPROM write duration	Tee_w	For next EEPROM write			10	ms
Self-test response		2.4 V, 25 °C	200	700		LSB
MECHANICAL CHARACTERISTICS						
Alignment Error	δ_a	relative to package outline		±1.0		Degree
Cross Axis Sensitivity		relative contribution between 3 axes		1.75		%

[6] Delay between power on (V_{DD} 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 mal-function.

Table 3: Maximum ratings specified for the BMA180

PARAMETER	CONDITION	MIN	MAX	UNITS
Supply Voltage	V_{DD} and V_{DDIO}	-0.3	4.25	V
Voltage at any digital pad	Vpad_dig	V_{SS}-0.3	V_{DDIO} +0.3	V
Storage Temperature range		-50	+150	°C
Junction temperature	Tj		+150	°C
EEPROM write cycles	Same Byte	1,000		Cycles
EEPROM retention times (both conditions are non cumulative, each represents max. time)	At 55°C, after 1000 cycles	10		Years
	At 85°C, after 1000 cycles	2		Years
Mechanical Shock	Duration ≤ 200µs		10,000	g
	Duration ≤ 1.0ms		3,000	g
	Free fall onto hard surfaces		1.8	m
ESD	HBM		2,000	V
	CDM		500	V
	MM		200	V

3 Block Diagram

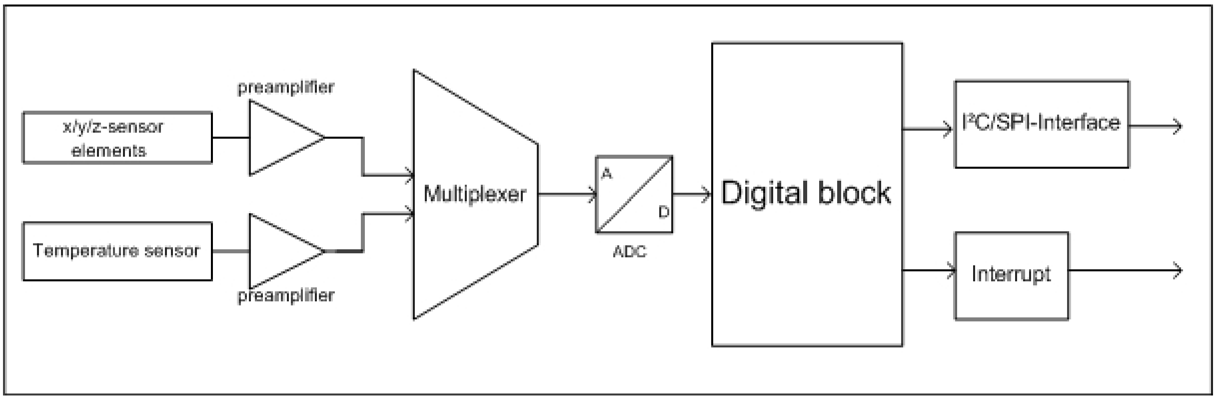


Figure 1: block diagram of BMA180

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.

Formulars:
- 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 tresh-hold) 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_self_wake_up = (i_DD · t_active + i_DDsm · wake-up-pause) / (t_active + wake-up-pause)

Where *t_active = 2ms + 0.417ms · (2400 / bandwidth) + 0.417ms · (1200 / bandwidth) · n*

With the following parameters:

i_DD	Current in standard mode
i_DDsm	Current in sleep mode
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)
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 table 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.

Filters				
Bandwidth [Hz]	**Samples-to-Settle**	**BW<3:0>**	**Type**	**Comments**
10	253	0000	Low-pass	
20	127	0001		
40	64	0010		
75	35	0011		
150	18	0100		
300	9	0101		
600	6	0110		
1200	0	0111		
1	1978	1000	High-pass	
0.2 – 300	8833	1001	Band-pass	

Table of constants	
Tupdate [ms]	0.417
Tstart [ms]	1.27
I_low_noise [µA]	975
I_sleep [µA]	0.5

Low-g & High-g Interrupt					
BW	**Minimum of Samples**	**Mean current consumption [µA] per one wake-up Period for wakeup_dur [ms]**			
		20	**80**	**320**	**2560**
10	254	817.50	555.53	243.54	39.17
20	128	710.16	393.79	141.66	20.47
40	65	569.05	254.12	79.19	10.83
75	36	435.41	164.20	47.16	6.33
150	19	305.59	100.16	27.28	3.67
300	10	207.58	61.95	16.41	2.26
600	7	168.15	48.45	12.73	1.78
1200	1	75.64	20.23	5.29	0.84
1	1979	947.09	884.41	699.32	236.89
0.2 – 300	8834	964.76	949.39	892.51	572.45

Slope & Tap sensing Interrupt					
BW	**Minimum of Samples**	**Mean current consumption [nA] per one wake-up Period for wakeup_dur [ms]**			
		20	**80**	**320**	**2560**
10	614	900.06	739.99	432.44	88.77
20	308	840.44	600.03	279.91	46.97
40	155	744.22	438.26	165.82	24.54
75	84	625.48	302.89	99.00	13.76
150	43	475.22	187.91	55.10	7.42
300	22	332.90	112.19	30.85	4.14
600	13	243.31	75.05	20.06	2.73
1200	4	124.42	34.55	9.02	1.31
1	1982	947.12	884.53	699.62	237.16
0.2 – 300	8846	964.77	949.42	892.61	572.77

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.

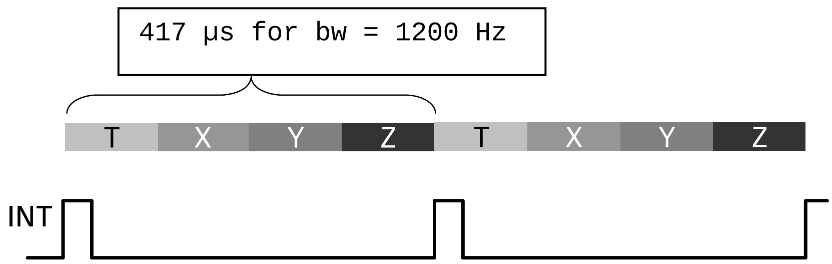


Figure 4: Explanation of data ready interrupt: For a bandwidth of e.g. 1.2 kHz the data refresh cycle takes 417 µs to update all data registers. After the final conversion of z-axis the INT pad will be set high. New data can be read out via interface (recommendation: read out within 20 µs after interrupt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

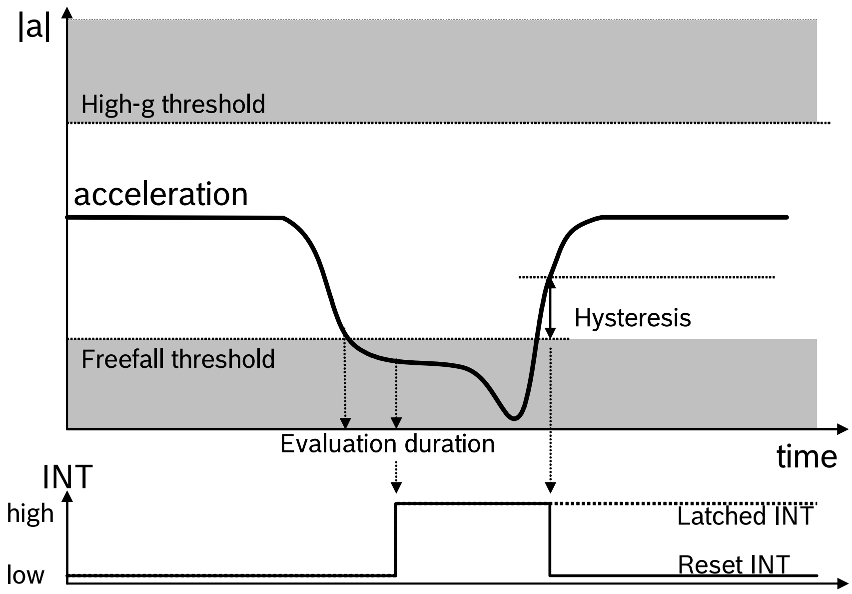


Figure 5: Schematic behaviour in case of low-g detection

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.

7 Global memory map

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

Register Name	Register Address (hexadecimal)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Bosch reserved	60h to 8Fh	Bosch reserved							
Bosch reserved	5Fh	Bosch reserved							
Bosch reserved	5Eh	Bosch reserved				Bosch reserved			
Bosch reserved	5Dh	Bosch reserved		Bosch reserved			Bosch reserved		
Bosch reserved	5Ch	Bosch reserved	Bosch reserved	Bosch reserved	Bosch reserved	Bosch reserved	Bosch reserved	Bosch reserved	
ee_crc	5Bh	crc<7:0>							
ee_offset_z	5Ah	offset_z<11:4> (msb)							
ee_offset_y	59h	offset_y<11:4> (msb)							
ee_offset_x	58h	offset_x<11:4> (msb)							
ee_offset_t	57h	offset_t<6:0> (msb)							readout_12bit
ee_offset_lsb2	56h	offset_z<3:0> (lsb)				offset_y<3:0> (lsb)			
ee_offset_lsb1	55h	offset_x<3:0> (lsb)				range<2:0>			smp_skip
ee_gain_z	54h	gain_z<6:0>							wake-up
ee_gain_y	53h	gain_y<6:0>							shadow_dis
ee_gain_x	52h	gain_x<6:0>							dis_reg
ee_gain_t	51h	gain_t<4:0>					tapsens_dur<2:0>		
ee_tco_z	50h	tco_z<5:0>						mode_config<1:0>	
ee_tco_y	4Fh	tco_y<5:0>						wake_up_dur<1:0>	
ee_tco_x	4Eh	tco_x<5:0>						slope_dur<1:0>	
ee_cd2: customer data	4Dh	cd2<7:0>							
ee_cd1: customer data	4Ch	cd1<7:0>							
ee_slope_th	4Bh	slope_th<7:0>							
ee_high_th	4Ah	high_th<7:0>							
ee_low_th	49h	low_th<7:0>							
ee_tapsens_th	48h	tapsens_th<7:0>							
ee_high_dur	47h	high_dur<6:0>							dis_i2c
ee_low_dur	46h	low_dur<6:0>							tco_range
ee_high_low_info	45h	high_int_x	high_int_y	high_int_z	high_filt	low_int_x	low_int_y	low_int_z	low_filt
ee_slope_tapsens_info	44h	slope_int_x	slope_int_y	slope_int_z	slope_filt	tapsens_int_x	tapsens_int_y	tapsens_int_z	tapsens_filt
ee_hy	43h	high_hy<4:0>					low_hy<4:2>		
ee_ctrl_reg4	42h	low_hy<1:0>		mot_cd_r		ff_cd_r		offset_finetuning<1:0>	
ee_ctrl_reg3	41h	slope_alert	slope_int	high_int	low_int	tapsens_int	adv_int	new_data_int	lat_int
ee_bw_tcs	40h	bw<3:0>				tcs<3:0>			
Bosch reserved	3Fh	Bosch reserved							
Bosch reserved	3Eh	Bosch reserved				Bosch reserved			
Bosch reserved	3Dh	Bosch reserved		Bosch reserved			Bosch reserved		
Bosch reserved	3Ch	Bosch reserved	Bosch reserved		Bosch reserved	Bosch reserved	Bosch reserved	Bosch reserved	
Bosch reserved	3Bh	crc<7:0>							
offset_z	3Ah	offset_z<11:4> (msb)							
offset_y	39h	offset_y<11:4> (msb)							
offset_x	38h	offset_x<11:4> (msb)							
offset_t	37h	offset_t<6:0> (msb)							readout_12bit
offset_lsb2	36h	offset_z<3:0> (lsb)				offset_y<3:0> (lsb)			
offset_lsb1	35h	offset_x<3:0> (lsb)				range<2:0>			smp_skip
gain_z	34h	gain_z<6:0>							wake-up
gain_y	33h	gain_y<6:0>							shadow_dis
gain_x	32h	gain_x<6:0>							dis_reg
gain_t	31h	gain_t<4:0>					tapsens_dur<2:0>		
tco_z	30h	tco_z<5:0>						mode_config<1:0>	
tco_y	2Fh	tco_y<5:0>						wake_up_dur<1:0>	
tco_x	2Eh	tco_x<5:0>						slope_dur<1:0>	
cd2: customer data	2Dh	cd2<7:0>							
cd1: customer data	2Ch	cd1<7:0>							
slope_th	2Bh	slope_th<7:0>							
high_th	2Ah	high_th<7:0>							
low_th	29h	low_th<7:0>							
tapsens_th	28h	tapsens_th<7:0>							
high_dur	27h	high_dur<6:0>							dis_i2c
low_dur	26h	low_dur<6:0>							tco_range
high_low_info	25h	high_int_x	high_int_y	high_int_z	high_filt	low_int_x	low_int_y	low_int_z	low_filt
slope_tapsens_info	24h	slope_int_x	slope_int_y	slope_int_z	slope_filt	tapsens_int_x	tapsens_int_y	tapsens_int_z	tapsens_filt
hy	23h	high_hy<4:0>					low_hy<4:2>		
ctrl_reg4	22h	low_hy<1:0>		mot_cd_r		ff_cd_r		offset_finetuning<1:0>	
ctrl_reg3	21h	slope_alert	slope_int	high_int	low_int	tapsens_int	adv_int	new_data_int	lat_int
bw_tcs	20h	bw<3:0>				tcs<3:0>			
- - -	11h to 1Fh	unused							
reset	10h	soft_reset							
ctrl_reg2	0Fh	unlock_ee<3:0>				reserved			unused
ctrl_reg1	0Eh	en_offset_x	en_offset_y	en_offset_z	unused	reserved			reserved
ctrl_reg0	0Dh	reserved	reset_int	update_image	ee_w	st1	st0	sleep	dis_wake_up
status_reg4	0Ch	high_sign_x_int	high_sign_y_int	high_sign_z_int	apsens_sign_x_in	apsens_sign_y_in	apsens_sign_z_in	unused	
status_reg3	0Bh	high_th_int	low_th_int	slope_int_s	tapsens_int	unused	x_first_int	y_first_int	z_first_int
status_reg2	0Ah	high_th_s	low_th_s	slope_s	tapsens_s	unused	low_sign_x_int	low_sign_y_int	low_sign_z_int
status_reg1	09h	first_tapsens_s	reserved	alert	slope_sign_x_int	slope_sign_y_int	slope_sign_z_int	offset_st_s	ee_write
temp	08h	temp<7:0>							
acc_z_msb	07h	acc_z<13:6> (msb)							
acc_z_lsb	06h	acc_z<5:0> (lsb)						0	new_data_z
acc_y_msb	05h	acc_y<13:6> (msb)							
acc_y_lsb	04h	acc_y<5:0> (lsb)						0	new_data_y
acc_x_msb	03h	acc_x<13:6> (msb)							
acc_x_lsb	02h	acc_x<5:0> (lsb)						0	new_data_x
version	01h								
chip_id	00h	0					chip_id<2:0>		

The left columns inform about the memory addresses. The remaining columns show the content of each register bit. 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. The EEPROM content upon delivery (after production test) is documented in 7.5.5.

7.1 Global memory mapping: general information

The global memory map of BMA180 provides three levels of access:

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

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:

Type of Register	Function of Register	Command	Volatile / non-volatile
Data Registers	– Chip identification – Acceleration data, temperature	Read Read	non-volatile volatile
Control Registers	– Activating self test, soft reset, switch to sleep mode etc.	Read / Write	Volatile
Status Registers	– Interrupt status and self test status – Customer reserved status bytes	Read Read / Write	volatile volatile
Setting Register	– Functional settings (range, bandwidth, mode, etc.) – Interrupt settings	Read / Write Read / Write	volatile volatile
EEPROM	– Default settings of functional and interrupt settings – Trimming values – Customer reserved data storage – Bosch Sensortec Reserved Memory	Write Write Write Write	non-volatile non-volatile non-volatile non-volatile

- 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.

Register	Format	Bit width
$OFFSET_{X\|Y\|Z}$	offset binary	3x12
$GAIN_{X\|Y\|Z}$	offset binary	3x7
$TCO_{X\|Y\|Z}$	offset binary	3x6
TCS	offset binary	4
$A_{X\|Y\|Z}$ (acceleration values)	2's complement	3x14
Temp.	2's complement	8
THRESHOLD (TH or TH_X|Y|Z)	unsigned positive	8
HYSTERESIS (HY or HY_X|Y|Z)	unsigned positive	5

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.

5. *chip_id (address 00h):* 03h

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

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)

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

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

Important remark:

- The sensor is calibrated using 2g-range. After changing the g-range to 8 or 16g, offset-shifts up to 400mg might occur. If using 8 or 16g range an offset correction is recommended. This could be done by offset fine-tuning the device during in-line calibration or other methods during use. For further details please contact your Bosch Sensortec representative.

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).

bw<3:0>	Selected bandwidth (Hz)
0000	10
0001	20
0010	40
0011	75
0100	150
0101	300
0110	600
0111	1200
1000	high-pass: 1 Hz
1001	band-pass: 0.2 Hz .. 300 Hz
1010 to 1111	not authorized codes

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 µs 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.

mode_config <1:0>	**Description**
00	Low noise mode -> highest current, low noise, full bandwidth (1200 Hz)
01	Not recommended
10	Not recommended
11	Low power mode -> BW is decreased by factor 2, lowest power, noise higher than in low noise modes, output data rate = 1200 samples/sec

The 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).

modeconfig	**Explanation**	**bw for bw-setting 10 Hz**	**bw for bw-setting 1200 Hz (1g, 1.5g, 2g)**	**bw for bw-setting 1200Hz (3g,4g)**	**bw for bw-setting is 1200Hz (8g, 16g)**	**typical current (incl. filtering) [µA]**	**typical noise density [µg/√Hz] in 2g-mode**	**Interrupt "timings" (e. g. tap sensing duration)**
00	low noise mode	10	1200	1200	1200	1025	150	Standard
11	low power mode	5	600	600	600	650	200	doubled to standard

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.

Bandwidth	1200	600	300	150	75	40	20	10
No. of samples	0	6	9	18	35	64	127	253

- 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 fine-tuning 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.
Important: Continuous operation with V_{DD} >2V may lead to destruction of the sensor.

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).

wake_up_dur<1:0>	Sleep mode duration (ms)
00	20
01	80
10	320
11	2560

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').

7.7.11 dis_i2c – disable I²C

This bit could be used to disable the I^2C mode. Per default, both interfaces are usable (dis_i2c = "0"), thus automatic switching from SPI to I^2C 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.

Non enabled interrupts must not be set to other duration than 0.

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.

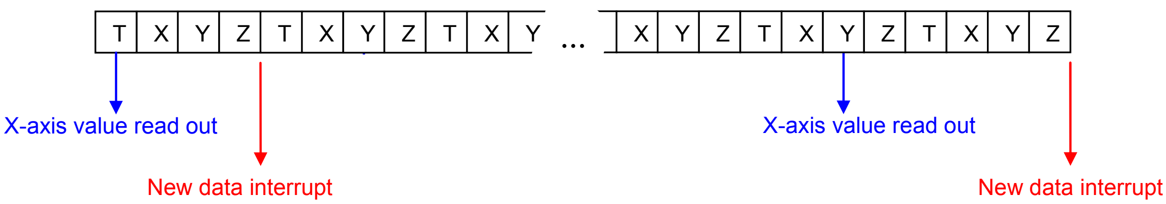
adv_int	Advanced interrupt control bits
0	Can't be activated (writing them to 1 has no effect)
1	Functions of the bits are enabled

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):



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 I^2C interface for data transfer, the data read out phase can be longer than 417 µs (depending on I^2C 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.

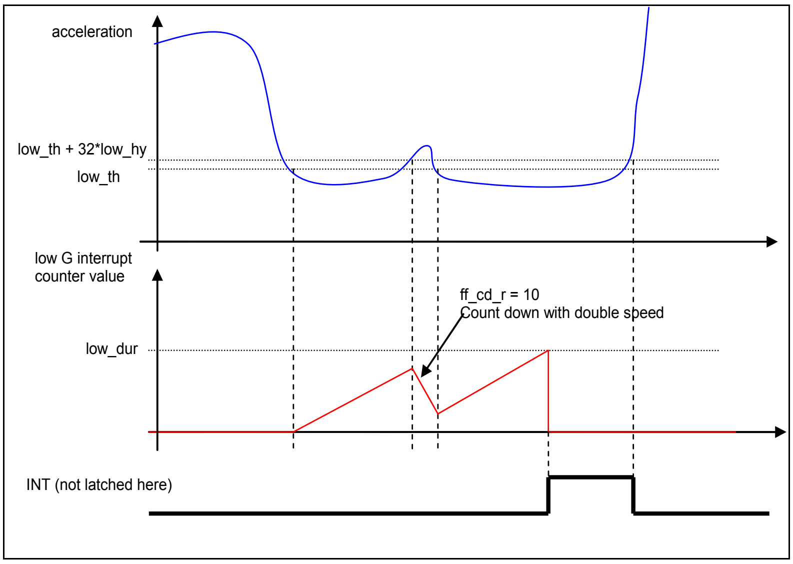


Figure 6: example of low-g detection debouncing with use of low_th, low_hy, low_dur and ff_cd_r settings

When acceleration signal is passing low_th value, low_th_criteria becomes active and counter *ff_cd_r* is incremented by 1 N_{COUNT}. (Ncount = 1 LSB/(5xTupdate) = 1 LSB/2.085ms in low noise modes). Depending on *ff_cd_r* register value, the counter could also be reset or count down when low_th_criteria is false:

ff_cd_r<1:0>	Low-g interrupt counter status when low_th_criteria is false
00	reset
01	Count down by 1 N_{COUNT}
10	Count down by 2 N_{COUNT}
11	Count down by 3 N_{COUNT}

When the low acceleration interrupt counter value equals *low_dur (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 thresh-hold.

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)

- *high_int:* This bit enables the high_th_criteria to generate an interrupt.
- *high_th:* defining high-g threshold value
- *high_hy:* defining associated *high* threshold hysteresis to prevent permanent interrupt generation in case acceleration signal is too close to threshold value
- *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} (Ncount = 1 LSB/(5xTupdate) = 1 LSB/2.085ms). Depen-ding on *mot_cd_r* register value, the counter could also be reset or count down when high_th_criteria is false:

mot_cd_r<1:0>	High acceleration interrupt counter status when high_th_criteria is false
00	reset
01	Count down by 1 N_{COUNT}
10	Count down by 2 N_{COUNT}
11	Count down by 3 N_{COUNT}

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 thresh-hold.

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.

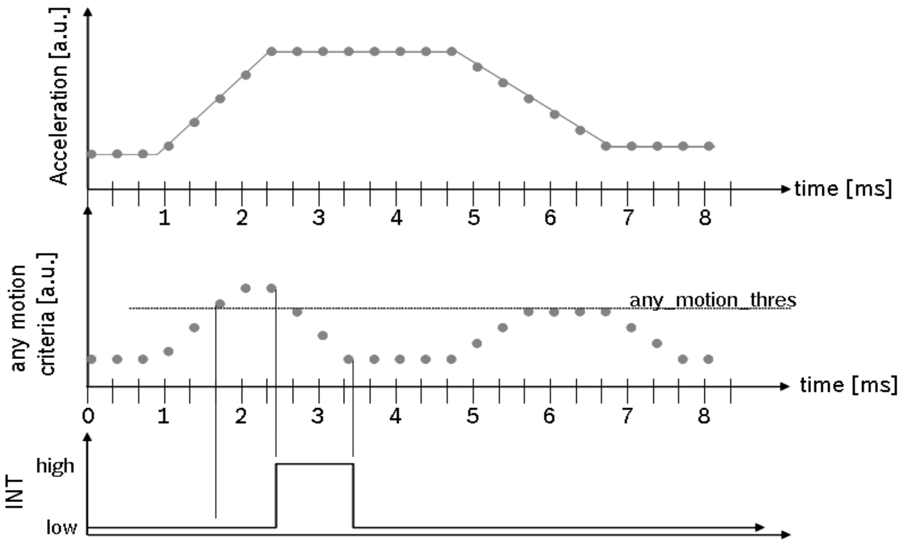


Figure 7: any motion detection with interrupt generation (schematic view)

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".

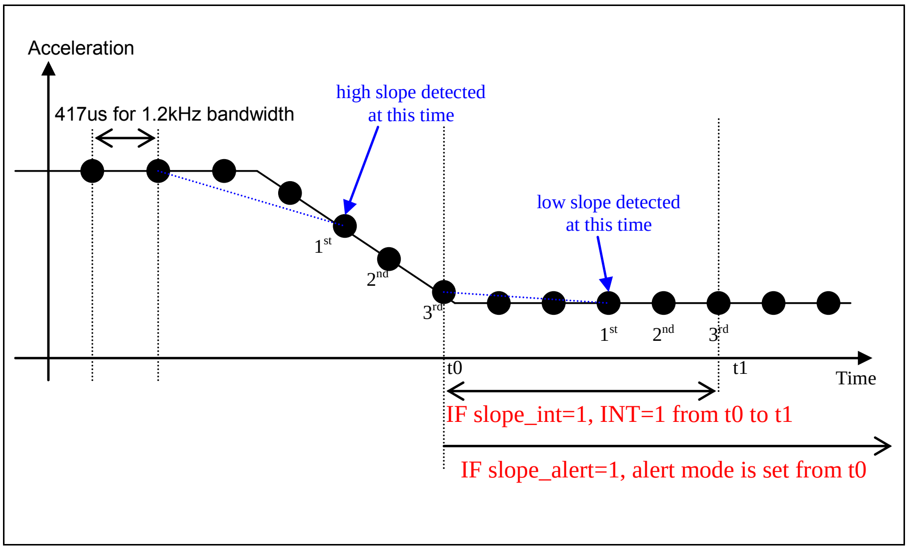


Figure 8: acceleration slope detection, example with slope_dur = 01b, 3 consecutive slope criteria must be detected.

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 *slope_dur**n*Tupdate.

slope_dur<1:0>	Number of required consecutiv conditions to set or reset the *slope_th_criteria*
00	1
01	3
10	5
11	7

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*bandwidth=2.4 ksamples/s leads to reaction for interrupt activation of 3*417 µs = 1.25 ms and a minimum slope interrupt duration of 3*417 µs = 1.25 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)

- *tapsens_int:* This bit enables the tapsens_criteria to generate an interrupt
- *tapsens_th* defines the threshold level of the tip-shock.
- *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 = 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

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

- *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.

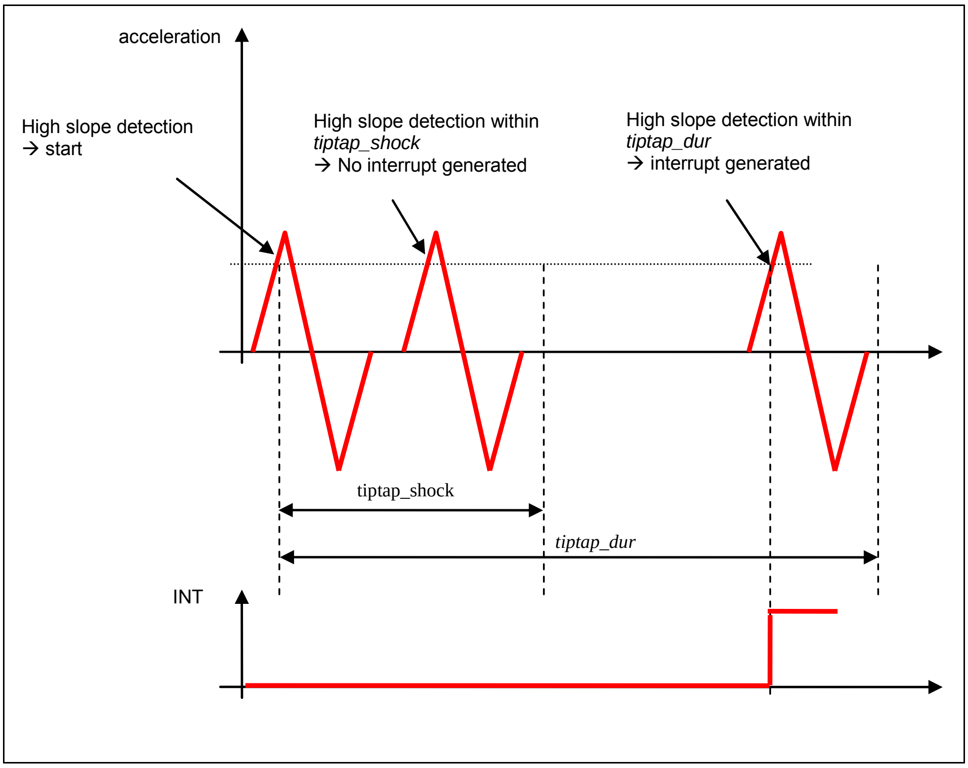


Figure 9: example of tap sensing detection with use of tapsens_th, tapsens_dur

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 to be 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:

offset_t: Offset trimming for temperature (7 bits).
offset_z: Offset trimming for Z axis (12 bits).
offset_y Offset trimming for Y axis (12 bits).
offset_x Offset trimming for X axis (12 bits).

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 to be 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 recalibrate 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:

offset_finetuning<1:0>	Offset regulation
00	no action
01 (default)	fine calibration
10	coarse calibration
11	full calibration

Setting these two bits enables offset regulation and disables automatically low interrupt function (*low_int* = '0').

The offset cancellation function has two sub-functions:

- <u>*Coarse calibration*</u>: 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).

- <u>Fine calibration</u>: 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_{adc} and a range of +/-64 LSB_{adc}, that means 7 bits per channel, called *fine_offset_** bits ("2's complement" coded). The following table defines the correspondence between the fine_offset_* bits and the low interrupt bits (in offset-tuning mode the fine_offset_* bits are stored in the below mentioned low_* bits).

fine_offset_* bits	Correspondance to low int registers	Register	Comments
fine_offset_x	low_th	29h	LSB to '0'
fine_offset_y	low_dur	26h	-
fine_offset_z	low_int_x	25h	bit 6
	low_int_y		bit 5
	low_int_z		bit 4
	low_filt		bit 3
	low_hy<4>	23h	bit 2
	low_hy<3>		bit 1
	low_hy<2>		bit 0

- The <u>full calibration</u> 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 *offset_finetuning* to a value different to '00'. The low interrupt function is consequently disabled if the *offset_finetuning* bit 0 is set to '1'.

2. Set the appropriate *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.

tcs<3:0>	Full temp (-40°C to +85°C) correction
0	-4,0%
1	-3,5%
2	-3,0%
3	-2,5%
4	-2,0%
5	-1,5%
6	-1,0%
7	-0,5%
8	0,0%
9	0,5%
10	1,0%
11	1,5%
12	2,0%
13	2,5%
14	3,0%
15	3,5%

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).

I^2C acknowledgement procedure for protected/non-protected area:

a) I^2C slave address: if correct, BMA180 sets acknowledge.
b) I^2C register address (I^2C write): BMA180 sets acknowledge for both unprotected and protected registers.
c) I^2C write data (I^2C write): BMA180 sets acknowledge for both unprotected and protected registers; no write is done for protected register.
d) I^2C read data (I^2C 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

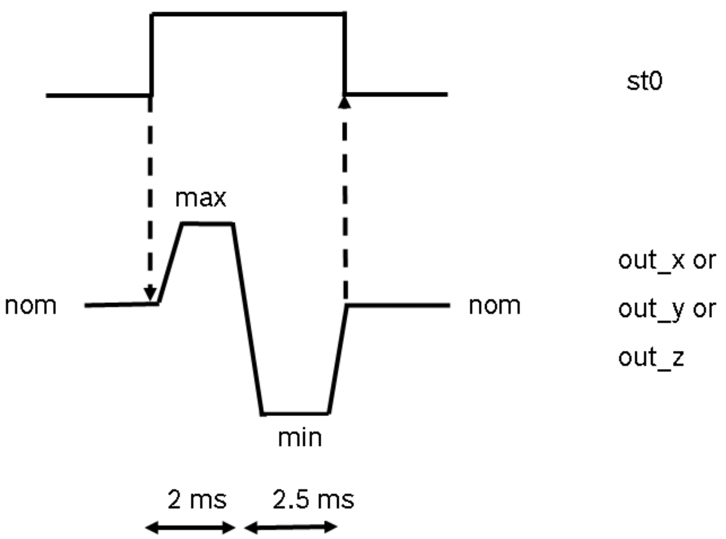
This self test bit does not generate any electrostatic force in the MEMS but is used to verify, whether the digital part is working correctly and that microprocessor is able to react to the interrupts. Basically a 0 g acceleration is emulated, and the user can detect the whole logic path for interrupt, 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 µC.
- 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 = abs(x_max – x_nom)
  - self_pos_y = abs(y_max – y_nom)
  - self_pos_z = abs(z_max – z_nom)
  - self_neg_x = abs(x_nom – x_min)
  - self_neg_y = abs(y_nom – y_min)
  - self_neg_z = abs(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. For more details please contact your Bosch Sensortec 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 the 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, the 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 unlock_ee

unlock_ee register allows access to the forbidden area of the EEPROM. Do not overwrite it.

7.10.10 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

See chapters 6.5 and 7.7.10 for more details.

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 rset 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 acquisation’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 bitsa re 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 the 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
+0.00025g	:	00 0000 0000 0001
		...
+1.99950g	:	01 1111 1111 1110
+1.99975g	:	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). LSB and MSB read-out must not be separated by another read. 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).

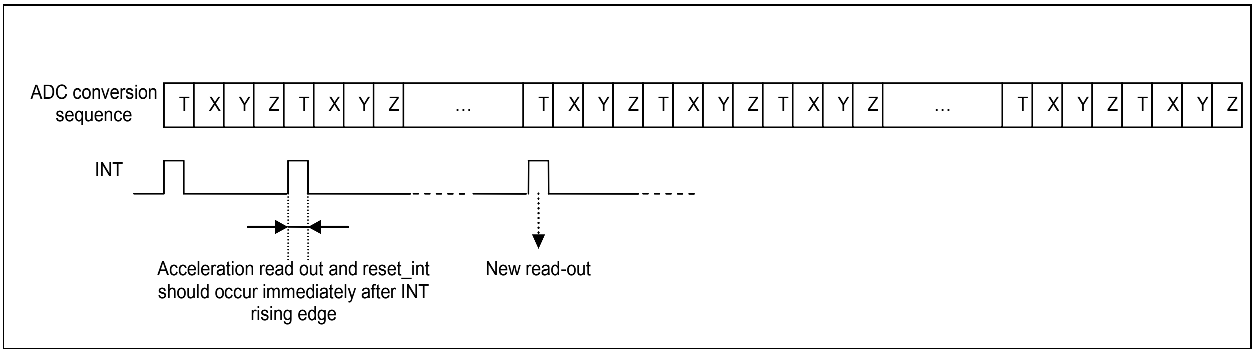


Figure 10: ADC conversion sequence and synchronization with read-out of acceleration values

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 correspond-ding 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 chip_id<2:0>

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 03h.

8 I²C and SPI-interfaces

8.1 Specification of interface parameters

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

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).

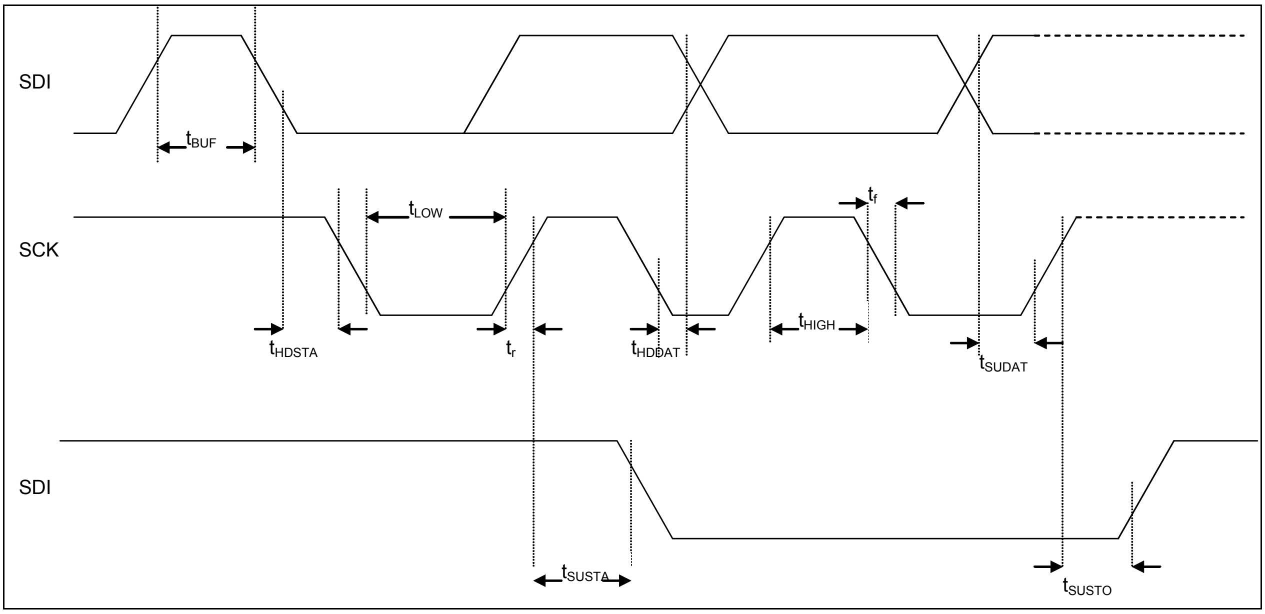


Figure 11: I²C timing diagram

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

Term	Description
Transmitter	the IC which sends data to the bus
Receiver	the IC which receives data from the bus
Master	the IC which initiates a transfer, generates clock signals and terminates the transfer (microcontroller in final application or tester during calibration procedure)
Slave	the IC addressed by a master (BMA180)

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 I^2C bus master. It is stopped with a rising edge on SDI when SCK is at high level; this is the stop condition (P).

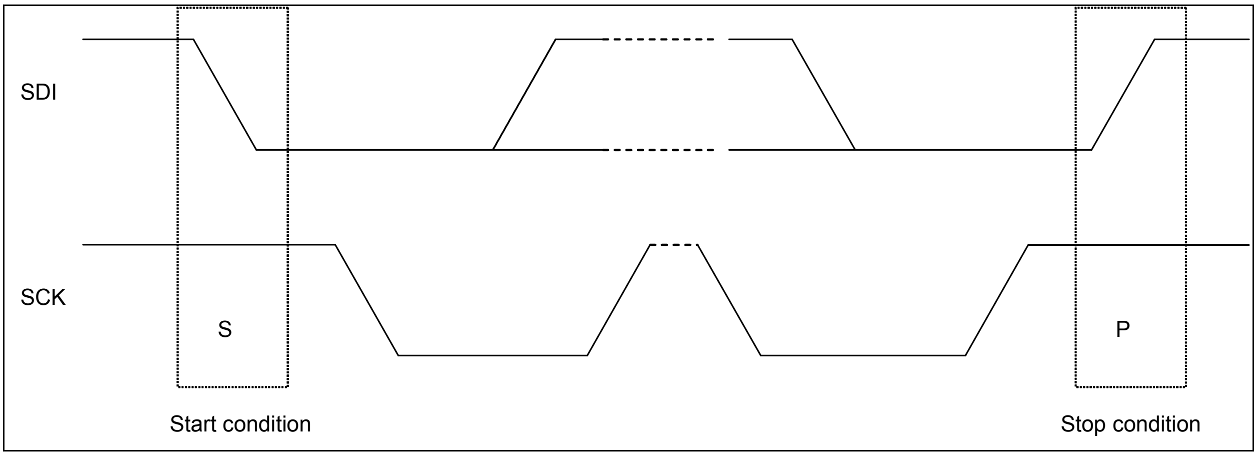


Figure 12: I^2C start and stop conditions

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.

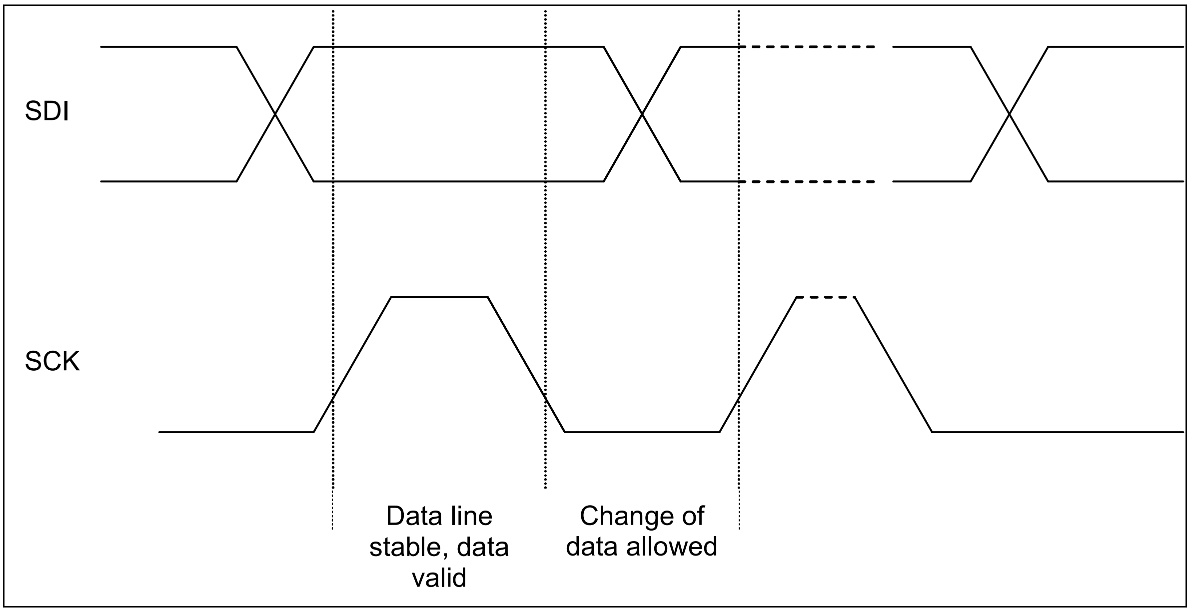


Figure 13: I^2C, 1 bit transfer

8.3.4 Acknowledge

After a start condition, data bits are transferred to BMA180. Each byte is followed by an acknowledge bit: the transmitter let 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 let SDI line free, enabling the transmitter to continue transfer or to generate a stop condition.

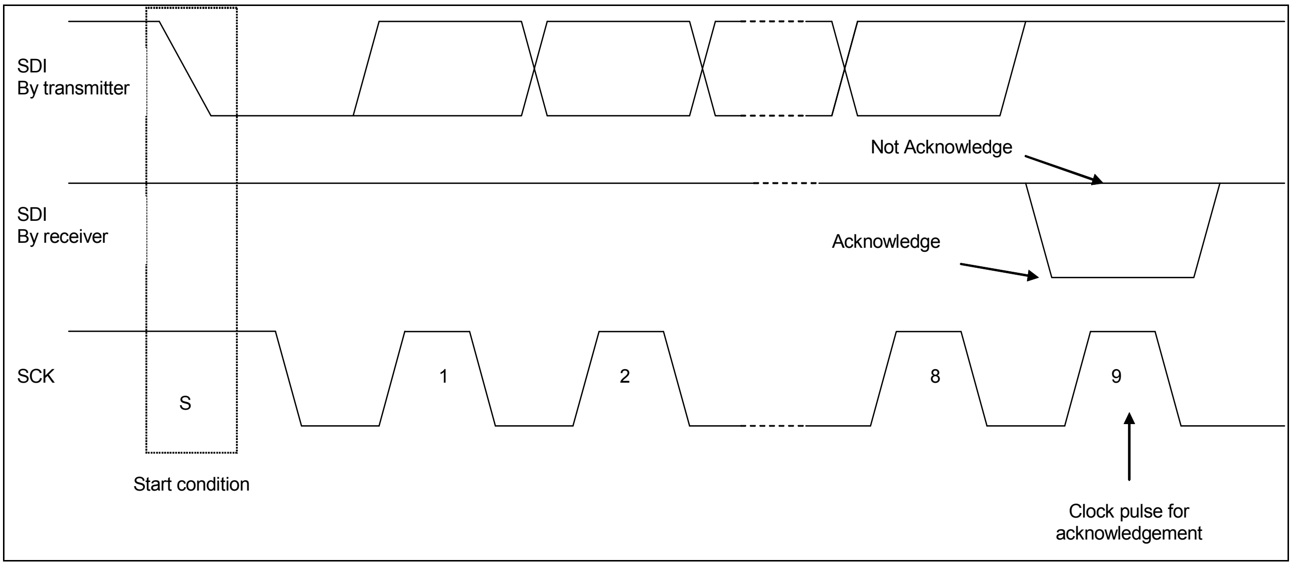


Figure 14: acknowledgement on SDI line

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 9^{th} clock pulse) and data transfer is enabled. The 8^{th} 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

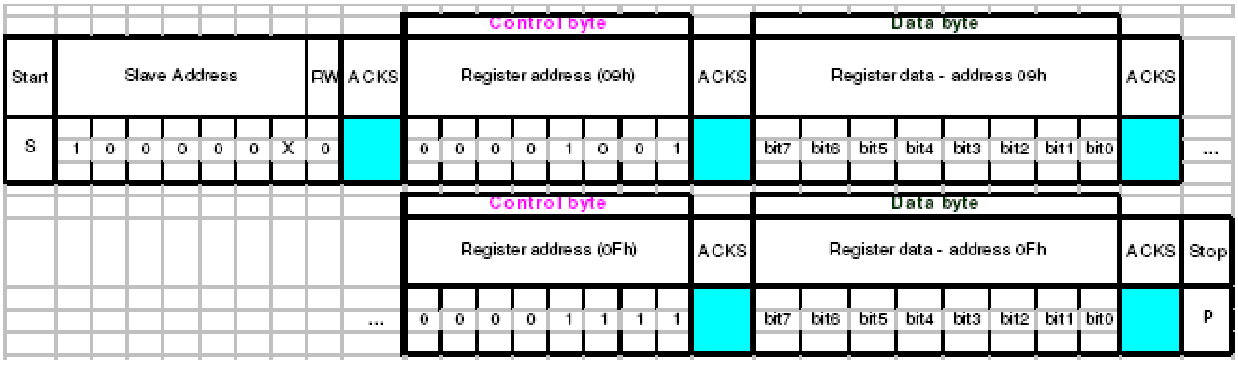


Figure 15: I^2C multiple write

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: I^2C multiple read

Figure 16 shows a typical I^2C transfer to read out acceleration data. Address register is first written to BMA180, the RW=0 (lowest acceleration data located to address 02h). I^2C 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

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 I^{2}C interface).

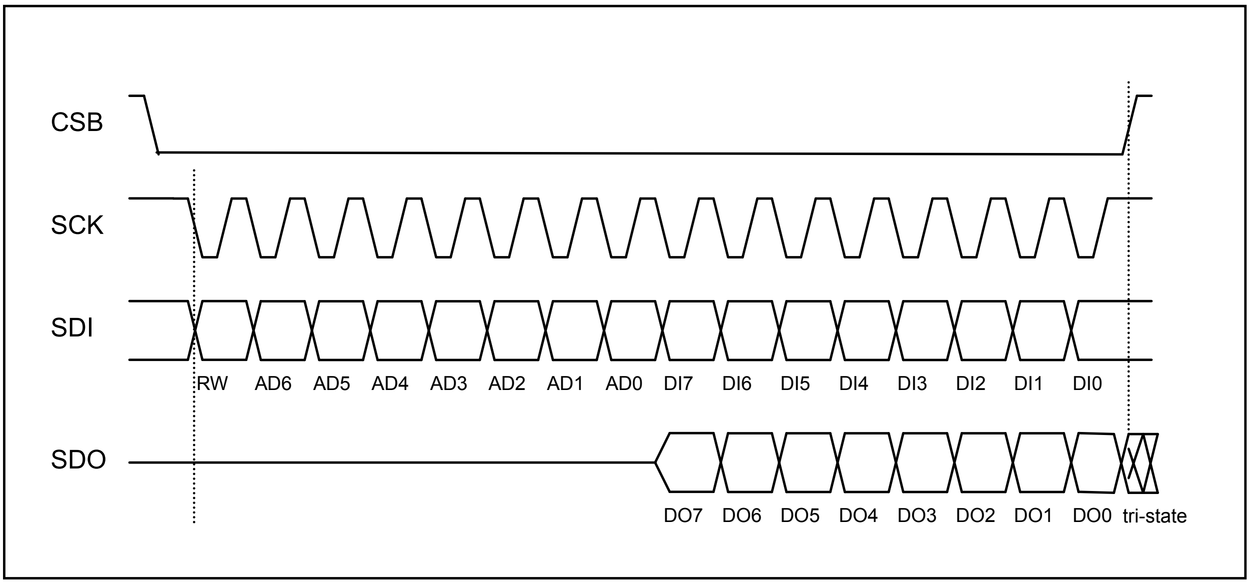


Figure 17: 4-wire SPI sequence

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

	Control byte		Data byte	Control byte		Data byte	
Start	RW	Register adress (16h)	Data register - adress 1Eh	RW	Register adress (0Bh)	Data register - adress 02h	Stop
CSB = 0	0	0 0 1 0 1 1 0	bit7 bit6 bit5 bit4 bit4 bit2 bit1 bit0	0	0 0 0 1 0 1 1	bit7 bit6 bit5 bit4 bit4 bit2 bit1 bit0	CSB = 1

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.

	Control byte		Data byte	Data byte	Data byte	
Start	RW	Register adress (02h)	Data register - adress 02h	Data register - adress 03h	Data register - adress 04h	Stop
CSB = 0	1	0 0 0 0 0 1 0	bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0	bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0	bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0	CSB = 1

Figure 19: SPI multiple read

8.4.2 SPI timings

4-wire SPI timings	Symbol	Condition	Min.	Typ.	Max	Unit
SPI clock input frequency	Fspi_4 Fspi_4_slow	VDDIO > 1.6V VDDIO < 1.6V			10 7.5	MHz MHz
SCK low pulse	Tlow_sck_4		20			ns
SCK high pulse	Thigh_sck_4		20			ns
SDI setup time	Tsetup_sdi_4		20			ns
SDI hold time	Thold_sdi_4		20			ns
SDO output delay	Tdelay_sdo_4	25 pF load			30	ns
CSB setup time	Tsetup_csb_4		20			ns
CSB hold time	Thold_csb_4		40			ns

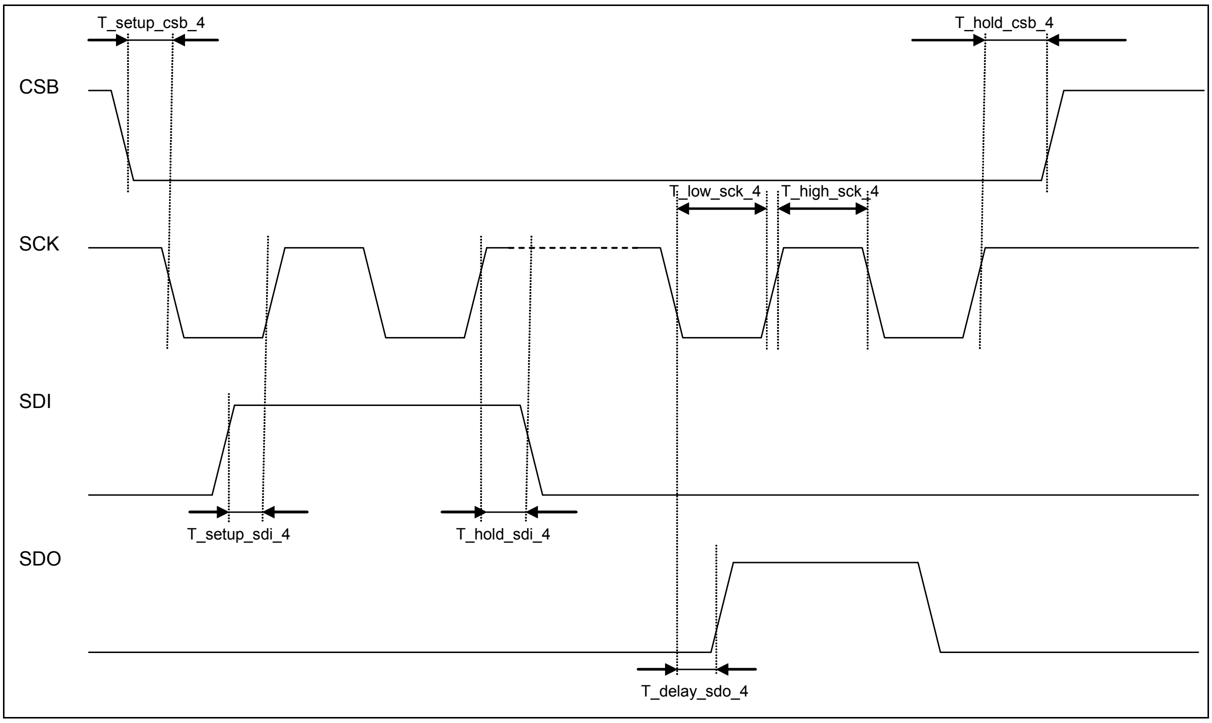
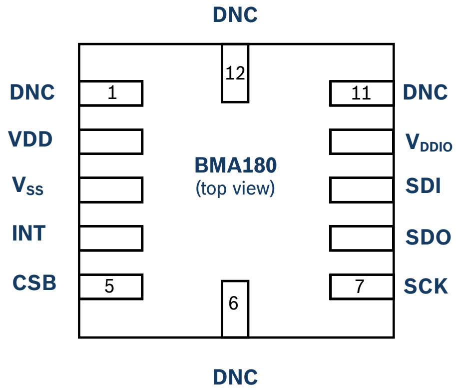


Figure 20: SPI timing

9 Pin-out

9.1 Pin configuration (top view)



9.2 Pin-out: electrical connections in case of SPI or I²C or no µC)

Pin	Name	Digital Analog	Description	SPI	I²C	no µC
01	reserved		Do Not Connect (internally connected to VDD)	DNC	DNC	DNC
02	V_{DD}	Power	Supply Voltage	V_{DD}	V_{DD}	V_{DD}
03	V_{SS}	GND	Ground Connection	GND	GND	GND
04	INT	Digital out	Interrupt PIN, active high	INT/NC	INT/NC	INT
05	CSB	Digital in	Chip Select, active low	CSB	V_{DDIO}	V_{DD}
06	reserved		Do Not Connect (internally connected to VSS)	DNC	DNC	DNC
07	SCK	Digital in	Serial Clock	SCK	SCK	GND
08	SDO	Digital in/ out	SPI output (4 wire) or Set-up of I²C address	SDO	GND or V_{DDIO}	GND
09	SDI	Digital in/out	SPI input or I²C serial data	SDI	SDA	GND
10	V_{DDIO}	Power Digital	Supply voltage Connection Digital	V_{DDIO}	V_{DDIO}	V_{DD}
11	reserved		Do Not Connect (internally connected to VSS)	DNC	DNC	DNC
12	reserved		Do Not Connect (internally connected to VSS)	DNC	DNC	DNC

Figure 20: Connection diagram for use with 4-wire SPI interface

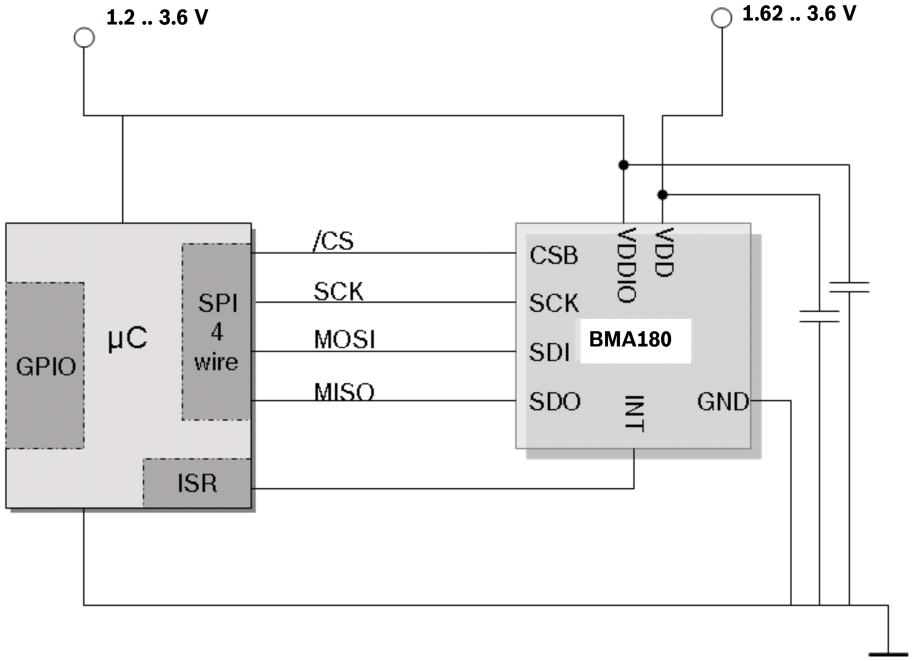


Figure 21: Connection diagram for use with I²C interface

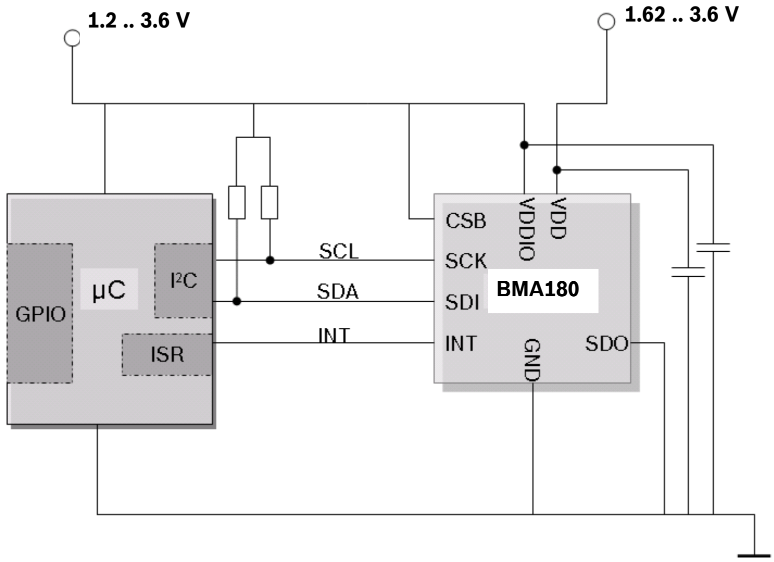
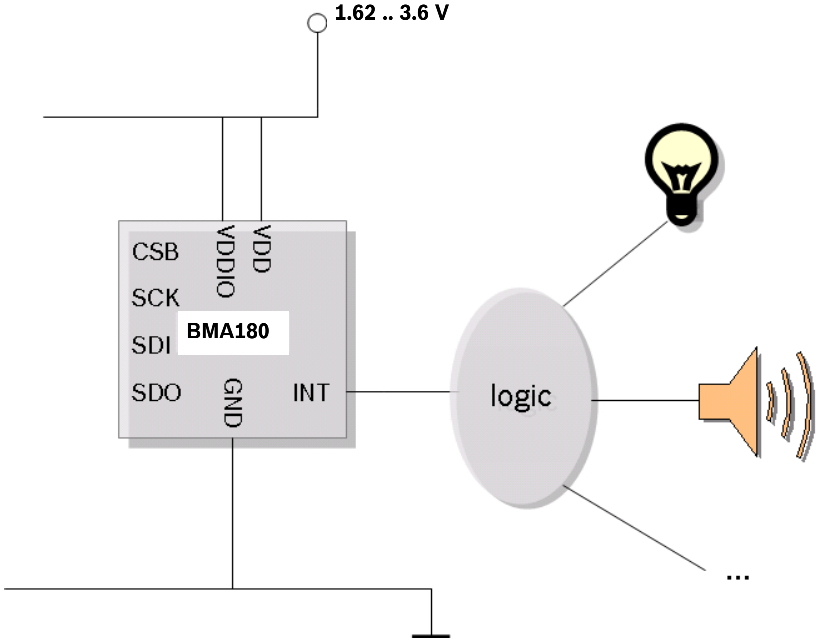
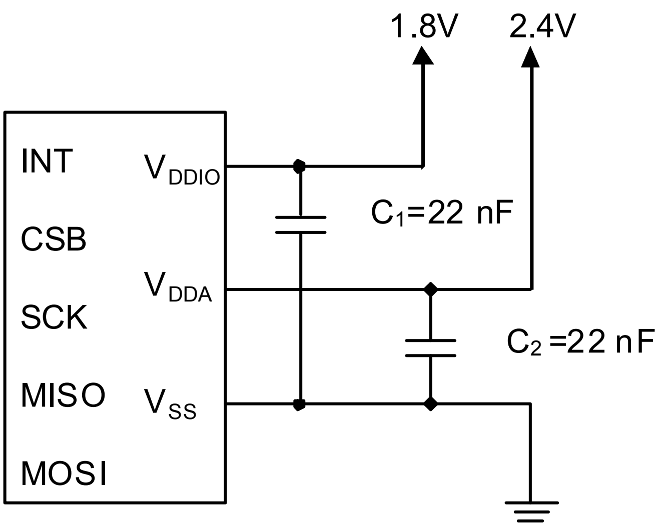


Figure 22: Connection diagram for stand alone use without microcontroller

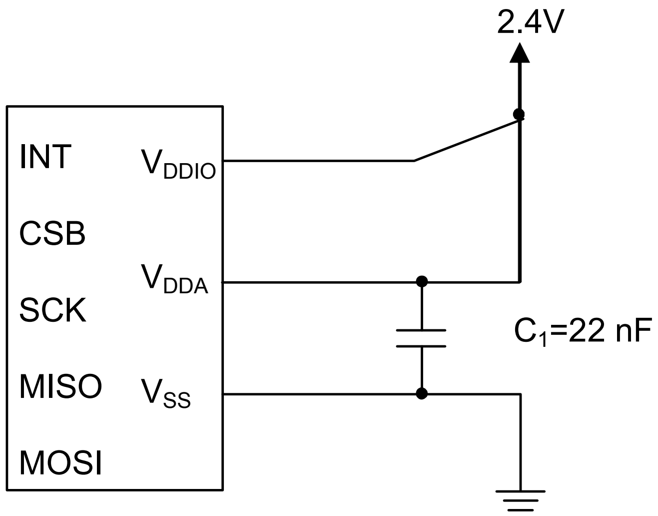


9.3 External component connection diagram

The following external component(s) are recommended to decouple the power source (voltages are examples).



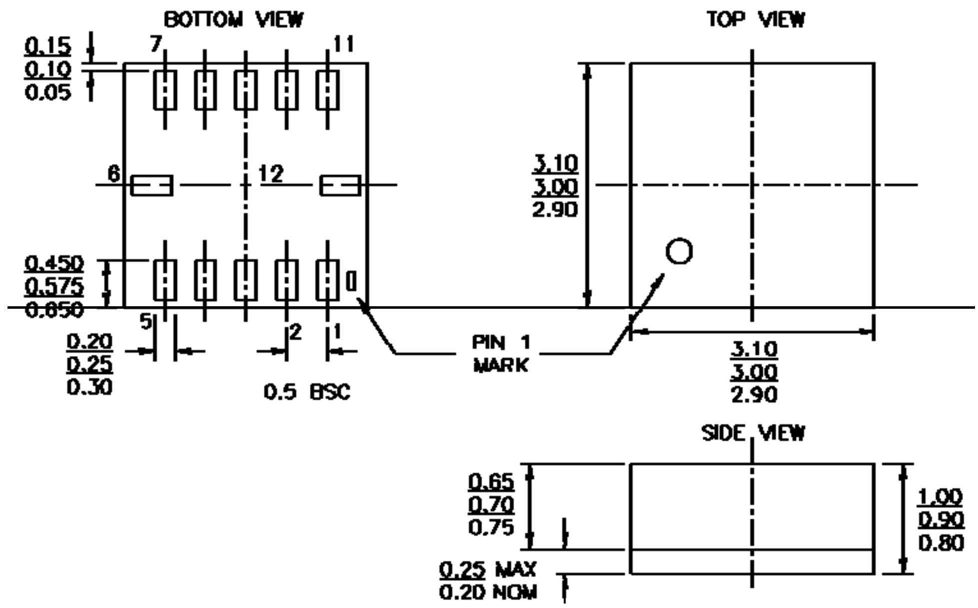
Or (if VDDIO = VDD)



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.

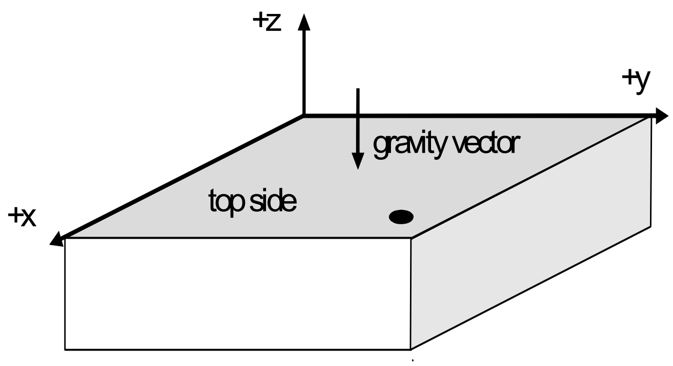


10.2 Orientation: polarity of the acceleration output

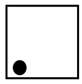
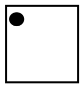
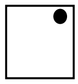
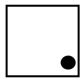
If the sensor is accelerated into the indicated directions, the corresponding channels will deliver a positive acceleration signal (dynamic acceleration).

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.

Sensor Orientation (gravity vector↓)						
Output Signal Ax	0g	+1g	0g	-1g	0g	0g
Output Signal Ay	-1g	0g	+1g	0g	0g	0g
Output Signal Az	0g	0g	0g	0g	+1g	-1g

10.3 Marking

10.3.1 Mass production devices

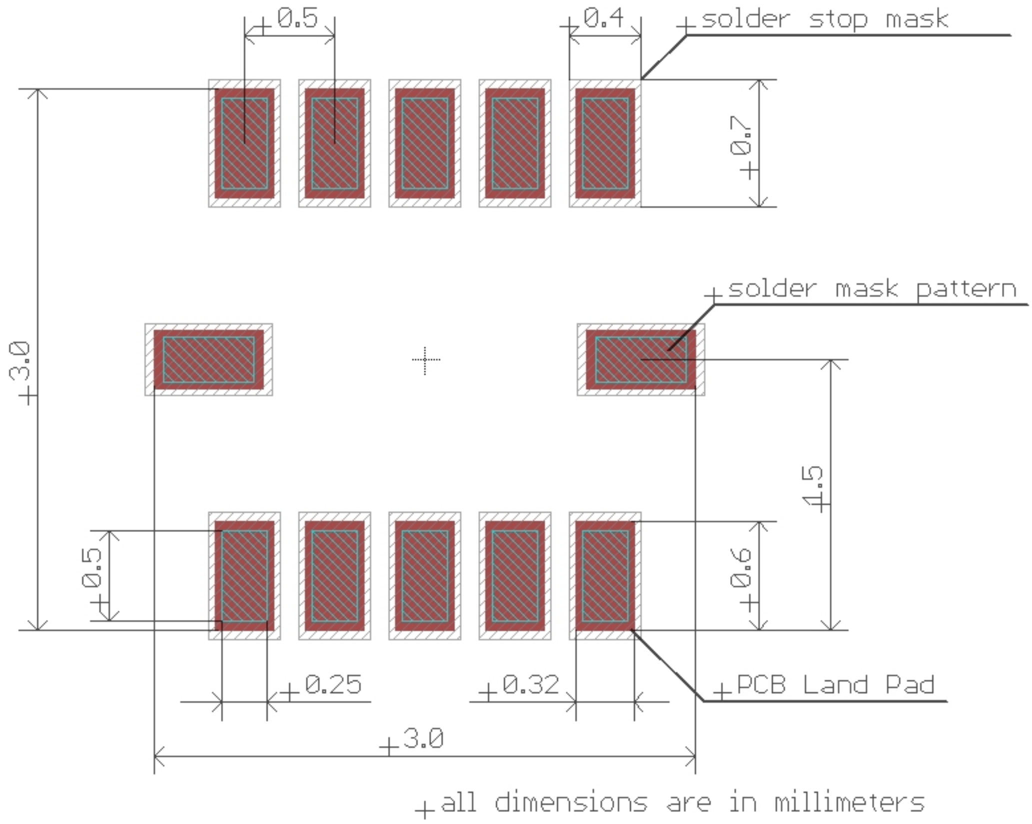
Sensor Label	Name	Symbol	Remark
053 AYWW ● CCC	Product number	053	
	Sub-con ID	A	Coded alphanumerically
	Date code	YWW	Y: year, numerically coded 9 = 2009, 0 = 2010, 1 = 2011, ... WW: working week, numerical
	Lot counter	CCC	
	Pin 1 identifier	•	

10.3.2 Engineering samples

Sensor Label	Name	Symbol	Remark
180e AYWW ● 0C1	Product name	180	BMA180
	Eng. Sample ID	E	Engineering samples are marked with an "e"
	Sub-con ID	A	Coded alphanumerically
	Date code	YWW	Y: year, numerically coded 9 = 2009, 0 = 2010, 1 = 2011, ... WW: working week, numerical
	Lot counter	0Cn	e.g. n = 1 → = C1-Sample
	Pin 1 identifier	•	

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

IPC/JEDEC J-STD-020C "Joint Industry Standard: Moisture/Reflow Sensitivity Classification for non-hermetic Solid State Surface Mount Devices"

and

IPC/JEDEC J-STD-033A "Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices".

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

Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

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

Rev. no	Chapter	Description of modification/changes	Date
0.x	All	Initial version	31.01.2009
1.0	All	Additional information, corrections, etc.	06.03.2009
2.0	All	Additional information, corrections, etc.	09.08.2009
2.1	1	Updated specification values	10.12.2009
2.2	10.3.1, 10.3.2	Update, date code, year coded numerically	26-February-2010
2.3	1	Added important comment on operation conditions	20-May-2010
	1	Update of table 2	
	7.7.7	Added important comment on operation conditions	
2.4	7, 7.5.5	Remove of reset state in global memory map	06 August 2010
	7.12.3	Update chip ID	
2.5	1	Added TCS and TCO for 4g and 16g ranges; replaced g_{FS2g} by "2g range"	07 December 2010
	4.3.2	Updated of the chapter	
	7.7.1	Added remark	
	7.7.3	Updated tables, text	
	6.5, 7.7.10, 7.8, 7.10.5, 7.11.2, 8.4.1	Text updates	
	7.12.2	Remark concerning LSB, MSB reading	

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 8
72770 Reutlingen / Germany

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

Modifications reserved | Printed in Germany
Specifications are subject to change without notice
Document number: BST-BMA180-DS000-07
Version_2.5_122010