Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

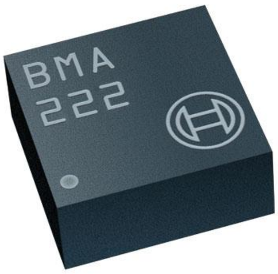
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

BMA222
Digital, triaxial acceleration sensor

Data sheet

Bosch Sensortec

BOSCH
Invented for life



BMA222 Data sheet

Ordering code	Please contact your Bosch Sensortec representative for the ordering code
Package type	12-pin LGA
Document revision	1.15
Document release date	31 May 2012
Document number	BST-BMA222-DS002-05
Technical reference code(s)	0 273 141 120
Notes	Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product’s appearance. Not intended for publishing.

BMA222
Digital, triaxial ±2g to ±16g acceleration sensor with intelligent on-chip motion-triggered interrupt controller

Key features

• Ultra-Small package	LGA package (12 pins), footprint 2mm x 2mm, height 0.95mm
• Digital interface	SPI (4-wire, 3-wire), I²C, 2 interrupt pins V_{DDIO} voltage range: 1.2V to 3.6V
• Programmable functionality	Acceleration ranges ±2g/±4g/±8g/±16g Low-pass filter bandwidths 1kHz - <8Hz
• On-chip interrupt controller	Motion-triggered interrupt-signal generation for - new data - any-motion (slope) detection - tap sensing (single tap / double tap) - orientation recognition - flat detection - low-g/high-g detection Stand-alone capability (no microcontroller needed)
• Ultra-low power ASIC	Low current consumption, short wake-up time, Advanced features for system power management
• RoHS compliant, halogen-free	

Typical applications
- Display profile switching
- Menu scrolling, tap / double tap sensing
- Gaming
- Pedometer / step counting
- Free-fall detection
- E-compass tilt compensation
- Drop detection for warranty logging
- Advanced system power management for mobile applications

General description
The BMA222 is a triaxial, low-g acceleration sensor with digital output for consumer market applications. It allows measurements of acceleration in three perpendicular axes. An evaluation circuitry (ASIC) converts the output of a micromechanical acceleration-sensing structure (MEMS) that works according to the differential capacitance principle.

Package and interfaces of the BMA222 have been defined to match a multitude of hardware requirements. Since the sensor features an ultra-small footprint and a flat package it is ingeniously suited for mobile applications.

The BMA222 offers a variable V_{DDIO} voltage range from 1.2V to 3.6V and can be programmed to optimize functionality, performance and power consumption in customer specific applications. In addition it features an on-chip interrupt controller enabling motion-based applications without use of a microcontroller.

The BMA222 senses tilt, motion and shock vibration in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

Index of Contents

1. Specification

If not stated otherwise, the given values are over lifetime and full performance temperature and voltage ranges, minimum/maximum values are ±3 σ.

Table 1: Parameter Specification

OPERATING CONDITIONS						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Acceleration Range	g_{FS2g}	Selectable via serial digital interface		±2		g
	g_{FS4g}			±4		g
	g_{FS8g}			±8		g
	g_{FS16g}			±16		g
Supply Voltage Internal Domains	V_{DD}		1.62	2.4	3.6	V
Supply Voltage I/O Domain	V_{DDIO}		1.2	2.4	3.6	V
Voltage Input Low Level	V_{IL}	SPI & I²C			$0.3V_{DDIO}$	-
Voltage Input High Level	V_{IH}	SPI & I²C	$0.7V_{DDIO}$			-
Voltage Output Low Level	V_{OL}	V_{DDIO} = 1.62V I_{OL} = 3mA, SPI & I²C			$0.2V_{DDIO}$	-
		V_{DDIO} = 1.2V I_{OL} = 3mA, SPI & I²C			0.23 V_{DDIO}	-
Voltage Output High Level	V_{OH}	V_{DDIO} = 1.62V I_{OL} = 2mA, SPI & I²C	$0.8V_{DDIO}$			-
		V_{DDIO} = 1.2V I_{OL} = 2mA, SPI & I²C	0.62 V_{DDIO}			-
Supply Current in Normal Mode	I_{DD}	Nominal V_{DD} supplies T_A=25°C, bw = 1kHz		139		µA
Supply Current in Low-Power Mode	I_{DDlp}	Nominal V_{DD} supplies T_A=25°C, bw = 1kHz sleep duration ≥ 25ms		7		µA
Supply Current in Suspend Mode	I_{DDsm}	Nominal V_{DD} supplies T_A=25°C		0.5		µA
Wake-Up Time	t_{w_up}	from Low-Power Mode or Suspend Mode, bw = 1kHz		0.8		ms
Start-Up Time	t_{s_up}	POR, bw = 1kHz		2		ms
Operating Temperature	T_A		-40		+85	°C

OUTPUT SIGNAL						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Device Resolution	D_{res}	g_{FS2g}		15.6		mg
Sensitivity	S_{2g}	g_{FS2g}, T_A=25°C		64		LSB/g
	S_{4g}	g_{FS4g}, T_A=25°C		32		LSB/g
	S_{8g}	g_{FS8g}, T_A=25°C		16		LSB/g
	S_{16g}	g_{FS16g}, T_A=25°C		8		LSB/g
Sensitivity Temperature Drift	TCS	g_{FS2g}, -40°C ≤ T_A ≤ +85°C Nominal V_{DD} supplies		±0.02		%/K
Zero-g Offset	Off	g_{FS2g}, T_A=25°C Nominal V_{DD} supplies		±100		mg
Zero-g Offset Temperature Drift	TCO	g_{FS2g}, -40°C ≤ T_A ≤ +85°C Nominal V_{DD} supplies		±1		mg/K
Bandwidth	bw_8	1st order filter, selectable via serial digital interface		8		Hz
	bw_{16}			16		Hz
	bw_{31}			31		Hz
	bw_{63}			63		Hz
	bw_{125}			125		Hz
	bw_{250}			250		Hz
	bw_{500}			500		Hz
	bw_{1000}			1000		Hz
Nonlinearity	NL	best fit straight line		±1		%FS
Output Noise	n_{rms}	g_{FS2g}, T_A=25°C Nominal V_{DD} supplies Normal mode		1		mg/√Hz
Power Supply Rejection Rate	PSRR	T_A=25°C Nominal V_{DD} supplies			20	mg/V
Temperature Sensor Measurement Range	T_S	T_A=25°C Nominal V_{DD} supplies	-40		+87.5	°C
Temperature Sensor Slope	dT_S	T_A=25°C Nominal V_{DD} supplies		0.5		LSB/K
Temperature Sensor Offset	OT_S	T_A=25°C Nominal V_{DD} supplies		±5		K
MECHANICAL CHARACTERISTICS						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Cross Axis Sensitivity	S	relative contribution between any two of the three axes		1		%
Alignment Error	E_A	relative to package outline		±0.5		°

2. Absolute maximum ratings

Table 2: Absolute maximum ratings

Parameter	Condition	Min	Max	Units
Voltage at Supply Pin	V_{DD} Pin	-0.3	4.25	V
	V_{DDIO} Pin	-0.3	4.25	V
Voltage at any Logic Pad	Non-Supply Pin	-0.3	V_{DDIO}+0.3	V
Passive Storage Temp. Range	≤ 65% rel. H.	-50	+150	°C
Mechanical Shock	Duration ≤ 200µs		10,000	g
	Duration ≤ 1.0ms		2,000	g
	Free fall onto hard surfaces		1.8	m
ESD	HBM, at any Pin		2	kV
	CDM		500	V

3. Block diagram

Figure 1 shows the basic building blocks of the BMA222:

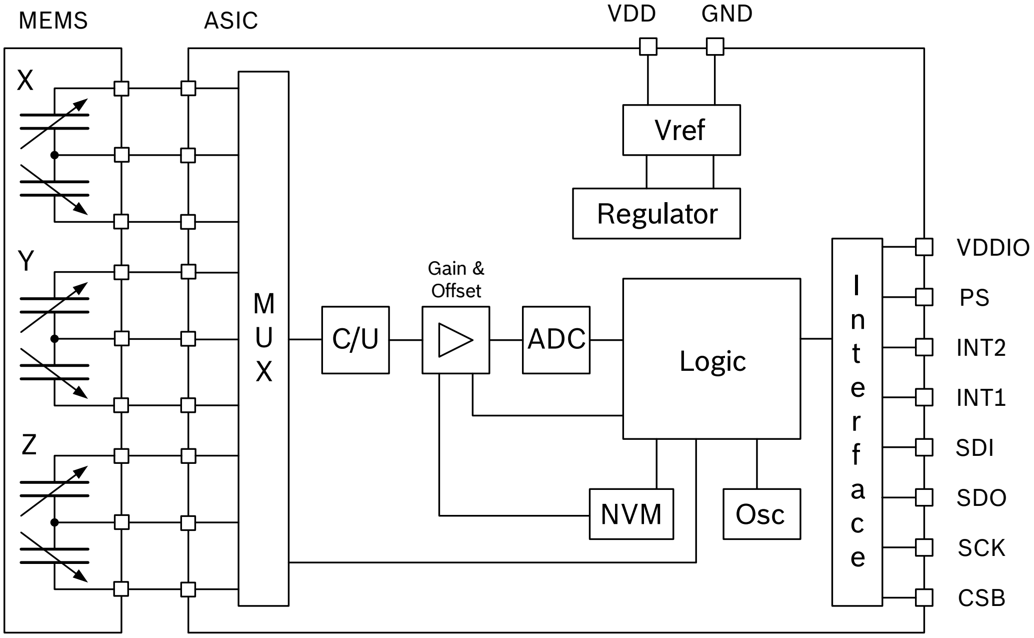


Figure 1: Block diagram of BMA222

4. Functional description

Note: Default values for registers can be found in chapter 5.

4.1 Power management

The BMA222 has two distinct power supply pins:
- V_{DD} is the main power supply for all internal analog and digital functional blocks;
- V_{DDIO} is a separate power supply pin, exclusively used for the supply of the digital interface.

There are no limitations on the voltage levels of both pins relative to each other, as long as each of them lies within its operating range. Furthermore, the device can be completely switched off (V_{DD} = 0V) while keeping the V_{DDIO} supply on (V_{DDIO} > 0V). To switch off the interface supply (V_{DDIO} = 0V) and keep the internal supply on (V_{DD} > 0V) is safe only in normal mode. If the device is in low-power mode or suspend mode while V_{DDIO} = 0V, there is a risk of excess current consumption on the V_{DD} supply (non-destructive).

It is absolutely prohibited to keep any interface at a logical high level when V_{DDIO} is switched off. Such a configuration will permanently damage the device (i.e. if V_{DDIO} = 0 → [SDI & SDO & SCK & CSB] ≠ high).

The device contains a power-on reset (POR) generator. It resets the logic part and the register values after powering-on V_{DD} and V_{DDIO}. There is no limitation on the sequence of switching on both supply voltages. In case the I²C interface shall be used, a direct electrical connection between V_{DDIO} supply and the PS pin is needed in order to ensure reliable protocol selection (see section 4.2 Operational modes).

4.2 Operational modes

Depending on the configuration the BMA222 is able to operate in two different operational modes:

- General mode: The device is acting as a slave on a digital interface (SPI or I²C) and is controlled by the external bus master (e.g. µC). The master gets measurement data and status information from the device through the digital interface. In particular, the master can configure the interrupt controller and read out the interrupt status registers. Moreover, it can freely configure and use the two interrupt pins (INT1, INT2). Several interrupts may be enabled in parallel.

- Dedicated mode: The dedicated mode allows the sensor to be operated as a stand-alone device in a simple µC-less system without abandon of the interrupt functionality. No digital interface is needed and, as a consequence, no measurement data can be read from the device. Instead of the digital interface the internal interrupt engine with its default setting is used. The interrupt status is mapped onto dedicated output pins. One out of three different sub-modes can be chosen: A) orientation recognition, B) tap sensing or C) slope (any-motion) detection. Only one interrupt at a time can be assigned.

The selection of the operational mode is done during start-up or reset by the state of the PS pin. If PS is floating, the dedicated mode is selected. A defined digital state selects the general mode. All pads are in input mode (no output driver active) during the start-up sequence until the operational mode and, in case of the general mode, the interface type is selected. The start-up sequence is run after power-up and after reset.

Figure 2 illustrates the selection of the different operational modes:

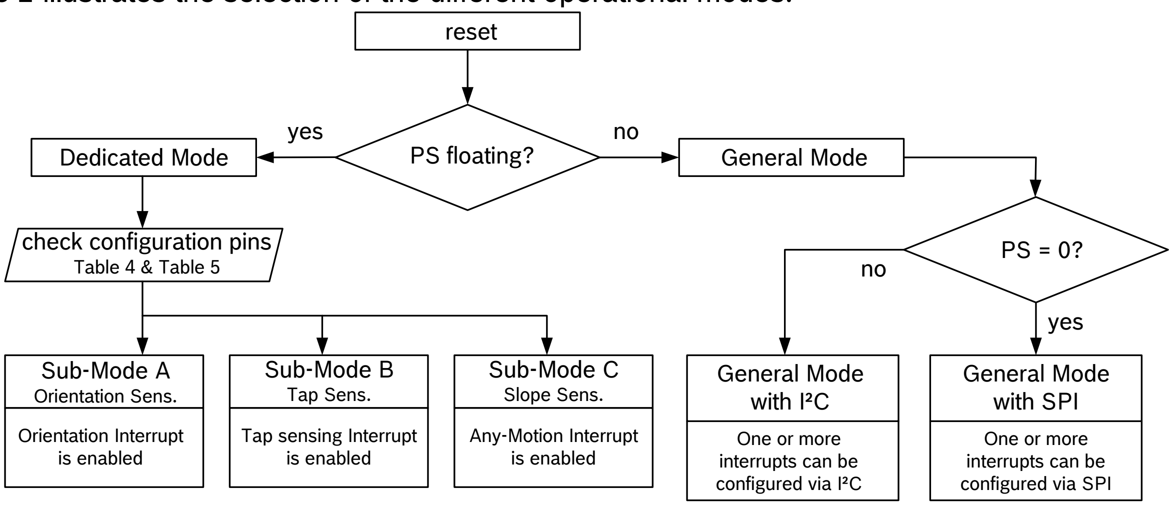


Figure 2: Operational mode selection

4.2.1 General mode

A defined digital state at the PS pin selects the general mode. Its polarity determines the kind of interface to be used:

- PS = GND enables the digital SPI interface
- PS = V_{DDIO} enables the digital I²C interface
- PS = float enables the dedicated mode

4.2.2 Dedicated mode (µC-less or stand-alone mode)

The dedicated mode operates with pre-defined settings of the interrupt engine in order to generate the motion-triggered interrupt-signals, i.e. bandwidth, sleep time, low-power mode, threshold, and hysteresis are use case optimized. Nevertheless some minor configurations can be selected by the user. The dedicated mode is entered if the device is connected according to table 3. During the start-up / power on sequence the PS pin (#11) must float.

Table 3: Entering and operating dedicated mode

VDDIO Pin#3	**NC** Pin#4	**VDD** Pin#7	**GNDIO** Pin#8	**GND** Pin#9	**PS** Pin#11
V_{DDIO}	NC	V_{DD}	GND	GND	**float**

Depending on the configuration of the other device pins according to table 4 the corresponding sub-mode of the dedicated mode is entered. In table 4 and table 5 the unshaded entries represent necessary input values for the corresponding sub-mode selection while the shaded entries represent corresponding output parameters of the events to be detected.

Table 4: Sub-mode selection and specific outputs of the dedicated mode

Sub-Mode	**SDO** Pin#1	**SDx** Pin#2	**INT1** Pin#5	**INT2** Pin#6	**CSB** Pin#10	**SCx** Pin#12
Orientation	output orient1-detect	output orient0-detect	output orient2-detect	output flat-detect	select orient sleep	GND
Tap	output double-detect	output single-detect	GND	select tap type	select tap sleep	V_{DD}
Slope	GND	output motion-detect	V_{DD}	GND	select slope sleep	V_{DD}

Table 5 contains state and description details of the parameters introduced in table 4. Unshaded entries represent input values to be set, shaded entries represent output parameters to be detected.

Table 5: Description of the parameters of table 4

Sub-Mode	Parameter see Table 4	State	Description
Orientation BW = 62.5 Hz	output orient0-detect	low	"upright" for portrait / "left" for landscape
		high	"upside-down" for portrait / "right" for landscape
	output orient1-detect	low	portrait
		high	landscape
	output orient2-detect	low	z-axis upward looking i.e. $\|\theta\| < 90°$ (Fig. 8)
		high	z-axis downward looking i.e. $\|\theta\| > 90°$ (Fig. 8)
	output flat-detect	low	non flat i.e. $\|\theta\| > 19{,}5°$ (Fig. 8)
		high	flat i.e. $\|\theta\| < 19{,}5°$ (Fig. 8)
	select orient sleep	GND	Low-Power mode enabled, sleep time = 100ms
		V_{DD}	Low-Power mode enabled, sleep time = 1s
Tap BW = 1k Hz	output double-detect	low	currently no Double-Tap event
		high	Double-Tap event detected
	output single-detect	low	currently no single-tap event
		high	Single-Tap event detected
	select tap type	GND	Single-Tap detection enabled
		V_{DD}	Double-Tap detection enabled
	select tap sleep	GND	Low-Power Mode disabled
		V_{DD}	Low-Power Mode enabled, sleep time = 10ms
Slope BW = 125 Hz	output motion-detect	low	currently no Any-Motion event
		high	Any-Motion event detected
	select slope sleep	GND	Low-Power mode enabled, sleep time = 50ms
		V_{DD}	Low-Power mode enabled, sleep time = 1s

low = GND, high = V_{DDIO}

For more details, refer to chapter 4.3 Power modes and 4.8 Interrupt Controller

- Orientation recognition sub mode → refer to chapter 4.8.7
- Tap sensing sub mode → refer to chapter 4.8.6
- Any-motion (slope) detection) sub mode → refer to chapter 4.8.5

4.3 Power modes

The BMA222 has three different power modes. Besides normal mode, which represents the fully operational state of the device, there are two special energy saving modes: low-power mode and suspend mode.

The possible transitions between the power modes are illustrated in figure 3:

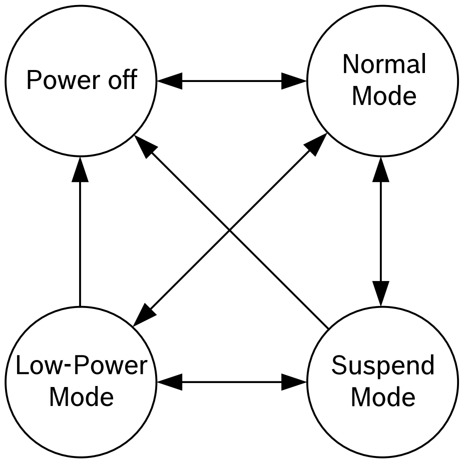


Figure 3: Power mode transition diagram

In normal mode, all parts of the electronic circuit are held powered-up and data acquisition is performed continuously.

In contrast to this, in suspend mode the whole analog part, oscillators included, is powered down. No data acquisition is performed, the only supported operations are reading registers (latest acceleration data are kept) and writing to the *(0x11) suspend* bit or *(0x14) softreset* register. Suspend mode is entered (left) by writing ´1´ (´0´) to the *(0x11) suspend* bit.

In low-power mode, the device is periodically switching between a sleep phase and a wake-up phase. The wake-up phase essentially corresponds to operation in normal mode with complete power-up of the circuitry. During the sleep phase the analog part except the oscillators is powered down. Low-power mode is entered (left) by writing ´1´ (´0´) to the *(0x11) lowpower_en* bit.

During the wake-up phase the number of samples required by any enabled interrupt is processed. If an interrupt is detected, the device stays in the wake-up phase as long as the interrupt condition endures (non-latched interrupt), or until the latch time expires (temporary interrupt), or until the interrupt is reset (latched interrupt). If no interrupt is detected, the device enters the sleep phase.

The duration of the sleep phase is set by the *(0x11) sleep_dur* bits as shown in the following table:

Table 6: Sleep phase duration settings

(0x11) sleep_dur	**Sleep Phase Duration t_{sleep}**
0000b	0.5ms
0001b	0.5ms
0010b	0.5ms
0011b	0.5ms
0100b	0.5ms
0101b	0.5ms
0110b	1ms
0111b	2ms
1000b	4ms
1001b	6ms
1010b	10ms
1011b	25ms
1100b	50ms
1101b	100ms
1110b	500ms
1111b	1s

The current consumption of the BMA222 can be calculated according to this formula:

$$I_{DDlp} \approx \frac{t_{sleep} \cdot I_{DDsm} + t_{active} \cdot I_{DD}}{t_{sleep} + t_{active}}.$$

When making an estimation about the length of the wake-up phase t_{active}, the wake-up time, t_{w_up}, has to be considered. Therefore, $t_{active} = t_{ut} + t_{w\_up}$, where t_{ut} is given in table 8. During the wake-up phase all analog modules are held powered-up, while during the sleep phase most analog modules are powered down. As a consequence, a wake-up time of less than 1ms (typ. value 0.8ms) is needed to settle the analog modules in order to get reliable acceleration data.

Table 7 gives an overview of the resulting average supply currents I_{DDlpe} for the different sleep phase durations and a selected bandwidth of 1000Hz, assuming no interrupt is active and thus only one sample per wake-up phase is taken:

Table 7: Average current consumption in low-power mode

Sleep phase duration	Average current consumption
0.5ms	100.5 µA
1ms	78.8 µA
2ms	55.0 µA
4ms	34.5 µA
6ms	25.2 µA
10ms	16.4 µA
25ms	7.4 µA
50ms	4.0 µA
100ms	2.3 µA
500ms	0.9 µA
1s	0.7 µA

4.4 Sensor data

4.4.1 Acceleration data

The width of acceleration data is 8 bits given in two´s complement representation. The 8 bits for each axis are given in registers (*0x03) acc_x, (0x05) acc_y* and*, (0x07) acc_z*.

The corresponding new data flags are given as *(0x02) new_data_x, (0x04) new_data_y* and *(0x06) new_data_z*. The remaining bits of these registers are fixed to 0.

The new data flags *(0x02) new_data_x, (0x04) new_data_y* and *(0x06) new_data_z* are set if the corresponding acceleration data registers (*0x03) acc_x, (0x05) acc_y* or*, (0x07) acc_z* have internally been updated. They are reset if the corresponding acceleration data registers have been read.

Two different streams of acceleration data are available, unfiltered and filtered. The unfiltered data is sampled with 2kHz. The sampling rate of the filtered data depends on the selected filter bandwidth; it is twice the bandwidth. Which kind of data is stored in the acceleration data registers depends on bit *(0x13) data_high_bw*. If *(0x13) data_high_bw* is ´0´ (´1´), then filtered (unfiltered) data is stored in the registers. Both data streams are separately offset-compensated. Both kinds of data can be processed by the interrupt controller.

The bandwidth of filtered acceleration data is determined by setting the *(0x10) bw* bit as followed:

Table 8: Bandwidth configuration

bw	Bandwidth	Update Time t_{ut}
00xxx	*)	-
01000	7.81Hz	64ms
01001	15.63Hz	32ms
01010	31.25Hz	16ms
01011	62.5Hz	8ms
01100	125Hz	4ms
01101	250Hz	2ms
01110	500Hz	1ms
01111	1000Hz	0.5ms
1xxxx	*)	-

*) Note: Settings 00xxx result in a bandwidth of 7.81 Hz; settings 1xxxx result in a bandwidth of 1000 Hz. It is recommended to actively use the range from ´01000b´ to ´01111b´ only in order to be compatible with future products.

The BMA222 supports four different acceleration measurement ranges. A measurement range is selected by setting the *(0x0F) range* bits as follows:

Table 9: Range selection

Range	Acceleration measurement range	Resolution
0011	±2g	15.6mg/LSB
0101	±4g	31.3mg/LSB
1000	±8g	62.5mg/LSB
1100	±16g	125mg/LSB
others	reserved	-

4.4.2 Temperature data

The width of temperature data is 8 bits given in two´s complement representation. Temperature values are available in the *(0x08) temp* register.

The slope of the temperature sensor is 0.5K/LSB, its center temperature is 24°C [*(0x08) temp* = 0x00]. Therefore, the typical temperature measurement range is -40°C up to 87.5°C.

4.5 Self-test

This feature permits to check the sensor functionality by applying electrostatic forces to the sensor core instead of external accelerations. By actually deflecting the seismic mass, the entire signal path of the sensor can be tested. Activating the self-test results in a static offset of the acceleration data; any external acceleration or gravitational force applied to the sensor during active self-test will be observed in the output as a superposition of both acceleration and self-test signal.

The self-test is activated individually for each axis by writing the proper value to the *(0x32) self_test_axis* bits (´01b´ for x-axis, ´10b´ for y-axis, ´11b´ for z-axis, ´00b´ to deactivate self-test). It is possible to control the direction of the deflection through bit *(0x32) self_test_sign*. The excitation occurs in positive (negative) direction if *(0x32) self_test_sign* = ´0b´ (´1b´).

In order to ensure a proper interpretation of the self-test signal it is recommended to perform the self-test for both (positive and negative) directions and then to calculate the difference of the resulting acceleration values. Table 10 shows the minimum differences for each axis. The actually measured signal differences can be significantly larger.

Table 10: Self-test difference values

	x-axis signal	y-axis signal	z-axis signal
resulting minimum difference signal	+0.8 g	+0.8 g	+0.4 g

It is recommended to perform a reset of the device after self-test. If the reset cannot be performed, the following sequence must be kept to prevent unwanted interrupt generation: disable interrupts, change parameters of interrupts, wait for at least 600 μs, enable desired interrupts.

4.6 Offset compensation

Offsets in measured signals can have several causes but they are always unwanted and disturbing in many cases. Therefore, the BMA222 offers an advanced set of four digital offset compensation methods which are closely matched to each other. These are slow, fast, and manual compensation, and inline calibration.

The compensation is performed for unfiltered and filtered data independently. It is done by adding a compensation value to the acceleration data coming from the ADC. The result of this computation is saturated if necessary to prevent any overflow errors (the smallest or biggest possible value is set, depending on the sign). However, the public registers used to read and write compensation values have only a width of 8 bits.

An overview of the offset compensation principle is given in figure 4:

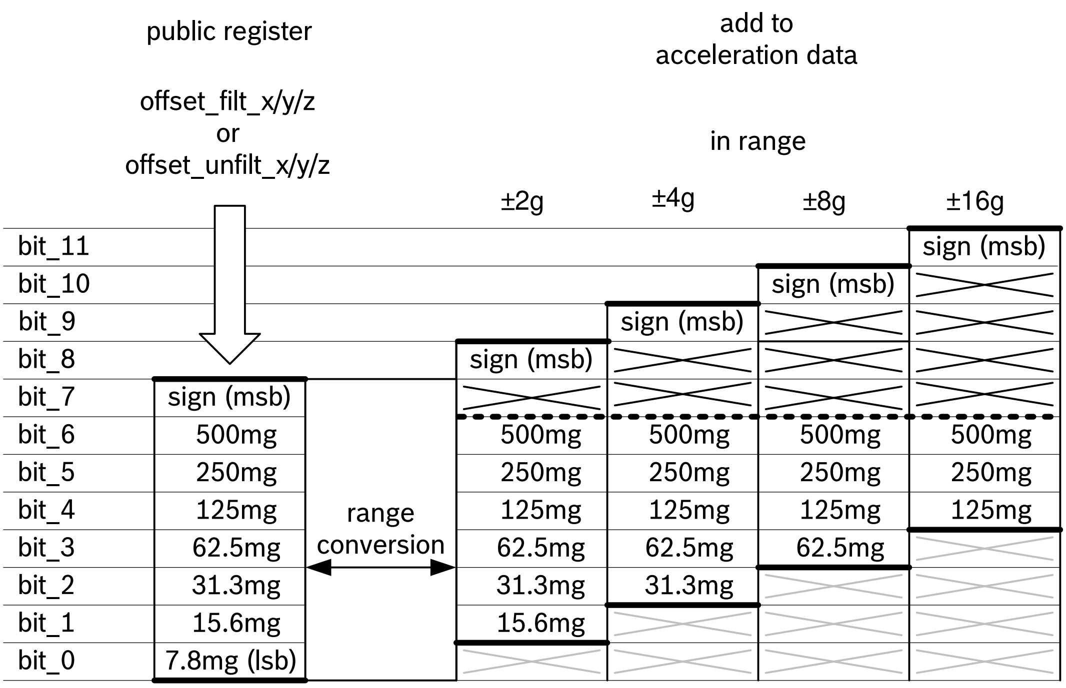


Figure 4: Principle of offset compensation

In dependence to the measurement range which has been set, the compensation value, which has been written into the public register will correct the data output according to figure 4.

e.g. ±2g range:
public register = 00000001b → add to acceleration data = ±0mg = ±0LSB
public register = 00000010b → add to acceleration data = +15.6mg = +1LSB
public register = 00000101b → add to acceleration data = +31.3mg = +2LSB

The public registers are image registers of EEPROM registers. With each image update (see section 4.7 Non-volatile memory" for details) the contents of the non-volatile EEPROM registers is written to the public registers. At any time the public register can be over-written by the user. After changing the contents of the public registers by either an image update or manually, all values are stored in the corresponding internal registers. In the opposite direction, if the value of an internal register changes due to the computation performed by a compensation algorithm, it is stored in the public register.

For slow and fast offset compensation, the compensation target can be chosen by setting the bits *(0x37) offset_target_x*, *(0x37) offset_target_y*, and *(0x37) offset_target_z* according to table 11:

Table 11: Offset target settings

(0x37) offset_target_x/y/z	**Target value**
00b	0g
01b	+1g
10b	-1g
11b	0g

By writing ´1´ to the *(0x36) offset_reset* bit, all offset compensation registers are reset to zero.

4.6.1 Slow compensation

Slow compensation is a quasi-continuous process which regulates the acceleration value of each axis towards the target value by comparing the current value with the target and adding or subtracting a fixed value depending on the comparison.

The algorithm in detail: If an acceleration value is larger (smaller) than the target value *(0x37) offset_target_x/y/z* for a number of samples (given by the parameter Offset Period see table 11), the internal offset compensation value *(0x38, 0x039, 0x3A) offset_filt_x/y/z* or *(0x3B, 0x03C, 0x3D) offset_unfilt_x/y/z* is decremented (incremented) by 4 LSB.

The public registers *(0x38, 0x039, 0x3A) offset_filt_x/y/z* and *(0x3B, 0x03C, 0x3D) offset_unfilt_x/y/z* are not used for the computations but they are updated with the contents of the internal registers (using saturation if necessary) and can be read by the user.

The compensation period offset_period is set by the *(0x37) cut_off* bit as represented in table 12:

Table 12: Compensation period settings

(0x37) cut_off	**Offset Period**
0b	8
1b	16

The slow compensation can be enabled (disabled) for each axis independently by setting the bits *(0x36) hp_x_en, hp_y_en, hp_z_en* to ´1´ (´0´), respectively.

Slow compensation should not be used in combination with low-power mode. In low-power mode the conditions (availability of necessary data) for proper function of slow compensation are not fulfilled.

4.6.2 Fast compensation

Fast compensation is a one-shot process by which the compensation value is set in such a way that when added to the raw acceleration, the resulting acceleration value of each axis equals the target value.

The algorithm in detail: An average of 16 consecutive acceleration values is computed and the difference between target value and computed value is written to *(0x38, 0x39, 0x3A) offset_filt_x/y/z* or *(0x3B, 0x3C, 0x3D) offset_unfilt_x/y/z* The public registers *(0x38, 0x39, 0x3A) offset_filt_x/y/z* and *(0x3B, 0x3C, 0x3D) offset_unfilt_x/y/z* are updated with the contents of the internal registers (using saturation if necessary) and can be read by the user.

Fast compensation is triggered for each axis individually by setting the *(0x36) cal_trigger* bits as shown in table 13:

Table 13: Fast compensation axis selection

(0x36) cal_trigger	**Selected Axis**
00b	none
01b	x
10b	y
11b	z

The register *(0x36) cal_trigger* keeps its non-zero value while the fast compensation procedure is running. Slow compensation is blocked as long as fast compensation endures. Bit *(0x36) cal_rdy* is ´0´ when *(0x36) cal_trigger* is not ´00´.

Fast compensation should not be used in combination with low-power mode. In low-power mode the conditions (availability of necessary data) for proper function of fast compensation are not fulfilled.

4.6.3 Manual compensation

As explained above, the contents of the public compensation registers *(0x38, 0x39, 0x3A) offset_filt_x/y/z* and *(0x3B, 0x3C, 0x3D) offset_unfilt_x/y/z* can be set manually via the digital interface. It is recommended to write into these registers immediately after a new data interrupt in order not to disturb running offset computations.

Writing to the offset compensation registers is not allowed if slow compensation is enabled or if the fast compensation procedure is running.

4.6.4 Inline calibration

For a given application, it is often desirable to calibrate the offset once and to store the compensation values permanently. This can be achieved by using one of the aforementioned offset compensation methods to determine the proper compensation values and then storing these values permanently in the non-volatile memory (EEPROM). See section 4.7 Non-volatile memory for details of the storing procedure.
Each time the device is reset, the compensation values are loaded from the non-volatile memory into the image registers and used for offset compensation until they are possibly overwritten using one of the other compensation methods.

4.7 Non-volatile memory

The entire memory of the BMA222 consists of three different kinds of registers: hard-wired, volatile, and non-volatile. Non-volatile memory is implemented as EEPROM. Part of it can be both read and written by the user. Access to non-volatile memory is only possible through (volatile) image registers.

Altogether, there are eight registers (bytes) of EEPROM which are accessible by the customer. The addresses of the image registers range from 0x38 to 0x3F. While the addresses up to 0x3D are used for offset compensation (see 4.6 Offset Compensation), addresses 0x3E and 0x3F are general purpose registers not linked to any sensor-specific functionality.

The content of the EEPROM is loaded to the image registers after a reset (either POR or softreset) or after a user request which is performed by writing ´1´ to bit *(0x33) nvm_load*. As long as the image update is not yet complete, bit *(0x33) nvm_load* is ´1´, otherwise it is ´0´.

The image registers can be read and written like any other register.

Writing to the EEPROM is a three-step procedure:

1. Write the new contents to the image registers.
2. Write ´1´ to bit *(0x33) nvm_prog_mode* in order to unlock the EEPROM.
3. Write ´1´ to bit *(0x33) nvm_prog_trig* and keep ´1´ in bit *(0x33) nvm_prog_mode* in order to trigger the write process.

Writing to the EEPROM always renews the entire EEPROM contents. It is possible to check the write status by reading bit *(0x33) nvm_rdy*. While *(0x33) nvm_rdy* = ´0´, the write process is still enduring; if *(0x33) nvm_rdy* = ´1´, then writing is completed. As long as the write process is ongoing, no power mode change and no image update is allowed. It is forbidden to write to the EEPROM while the image update is running, in low-power mode, and in suspend mode.

4.8 Interrupt controller

Seven interrupt engines are integrated in the BMA222. Each interrupt can be independently enabled and configured. If the condition of an enabled interrupt is fulfilled, the corresponding status bit is set to ´1´ and the selected interrupt pin is activated. There are two interrupt pins, INT1 and INT2; interrupts can be freely mapped to any of these pins. The pin state is a logic ´or´ combination of all mapped interrupts.

The interrupt status registers are updated together with writing new data into the acceleration data registers. If an interrupt is disabled, all active status bits and pins are immediately reset.

All time constants are based upon the typical frequency of the internal oscillator. This is reflected by the bandwidths (bw) as specified in table 1.

4.8.1 General features

An interrupt is cleared depending on the selected interrupt mode, which is common to all interrupts. There are three different interrupt modes: non-latched, latched, and temporary. The mode is selected by the *(0x21) latch_int* bits according to table 14

Table 14: Interrupt mode selection

(0x21) latch_int	**Interrupt mode**
0000b	non-latched
0001b	temporary, 250ms
0010b	temporary, 500ms
0011b	temporary, 1s
0100b	temporary, 2s
0101b	temporary, 4s
0110b	temporary, 8s
0111b	latched
1000b	non-latched
1001b	temporary, 500µs
1010b	temporary, 500µs
1011b	temporary, 1ms
1100b	temporary, 12.5ms
1101b	temporary, 25ms
1110b	temporary, 50ms
1111b	latched

An interrupt is generated if its activation condition is met. It can not be cleared as long as the activation condition is fulfilled. In the non-latched mode the interrupt status bit and the selected pin (the contribution to the ´or´ condition for INT1 and/or INT2) are cleared as soon as the activation condition is no more valid. Exceptions to this behaviour are the new data, orientation, and flat interrupts, which are automatically reset after a fixed time.

In the latched mode an asserted interrupt status and the selected pin are cleared by writing ´1´ to bit *(0x21) reset_int*. If the activation condition still holds when it is cleared, the interrupt status is asserted again with the next change of the acceleration registers.

In the temporary mode an asserted interrupt and selected pin are cleared after a defined period of time. The behaviour of the different interrupt modes is shown graphically in figure 5:

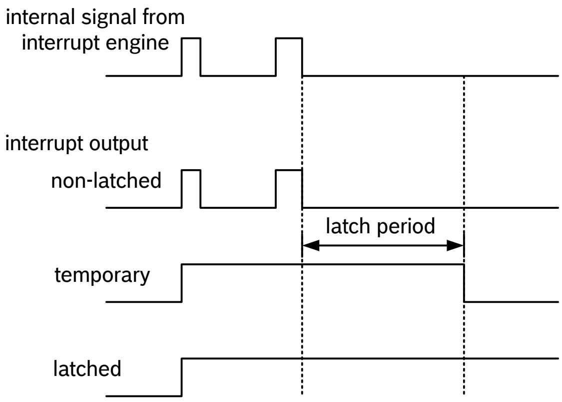


Figure 5: Interrupt modes

Several interrupt engines can use either unfiltered or filtered acceleration data as their input. For these interrupts, the source can be selected with the respective *(0x1E) int_src_...* bits, in details these are *(0x1E) int_src_data*, *(0x1E) int_src_tap*, *(0x1E) int_src_slope*, *(0x1E) int_src_high*, and *(0x1E) int_src_low*. Setting the respective bits to ´0´ (´1´) selects filtered (unfiltered) data as input. For the other interrupts, orientation recognition and flat detection, such a selection is not possible. They always use filtered input data.

It is strongly recommended to set interrupt parameters prior to enabling the interrupt. Changing parameters of an already enabled interrupt may cause unwanted interrupt generation and generation of a false interrupt history. A safe way to change parameters of an enabled interrupt is to keep the following sequence: disable the desired interrupt, change parameters, wait for at least 600 μs, enable the desired interrupt.

4.8.2 Mapping (inttype to INT Pin#)

The mapping of interrupts to the interrupt pins #05 or #06 is done by registers *(0x19)* to *(0x1B)*. Setting *(0x19) int1_*"inttyp" to ´1´ (´0´) maps (unmaps) "inttyp" to pin #5 (INT1), correspondingly setting *(0x1B) int2_*"inttyp" to ´1´ (´0´) maps (unmaps) "inttyp" to pin #6 (INT2).

Note: "inttyp" to be replaced with the precise notation, given in the memory map in chapter 5.

Example: For flat interrupt (int1_flat): Setting *(0x19) int1_flat* to ´1´ maps *int1_flat* to pin #5 (INT1).

4.8.3 Electrical behaviour (INT pin# to open-drive or push-pull)

Both interrupt pins can be configured to show desired electrical behaviour. The ´active´ level of each pin is determined by the *(0x20) int1_lvl* and *(0x20) int2_lvl* bits.

If *(0x20) int1_lvl* = ´1´ (´0´) / *(0x20) int2_lvl* = ´1´ (´0´), then pin #05 (INT1) / pin #06 (INT2) is active ´1´ (´0´). In addition to that, also the electric type of the interrupt pins can be selected. By setting bits *(0x20) int1_od* / *(0x20) int2_od* to ´0´, the interrupt pin output type gets push-pull, by setting the configuration bits to ´1´, the output type gets open-drive.

Remark: Due to their use for sub-mode selection in dedicated mode, the states of both INT pins are not defined during the first 2 ms after power-up.

4.8.4 New data interrupt

This interrupt serves for synchronous reading of acceleration data. It is generated after storing a new value of z-axis acceleration data in the data register. The interrupt is cleared automatically when the next cycle of data acquisition starts. The interrupt status is ´0´ for at least 50μs.

The interrupt mode of the new data interrupt is fixed to non-latched.

It is enabled (disabled) by writing ´1´ (´0´) to bit *(0x17) data_en*. The interrupt status is stored in bit *(0x0A) data_int*.

4.8.5 Any-motion (slope) detection

Any-motion detection uses the slope between successive acceleration signals to detect changes in motion. An interrupt is generated when the slope (absolute value of acceleration difference) exceeds a preset threshold. It is cleared as soon as the slope falls below the threshold. The principle is made clear in figure 6.

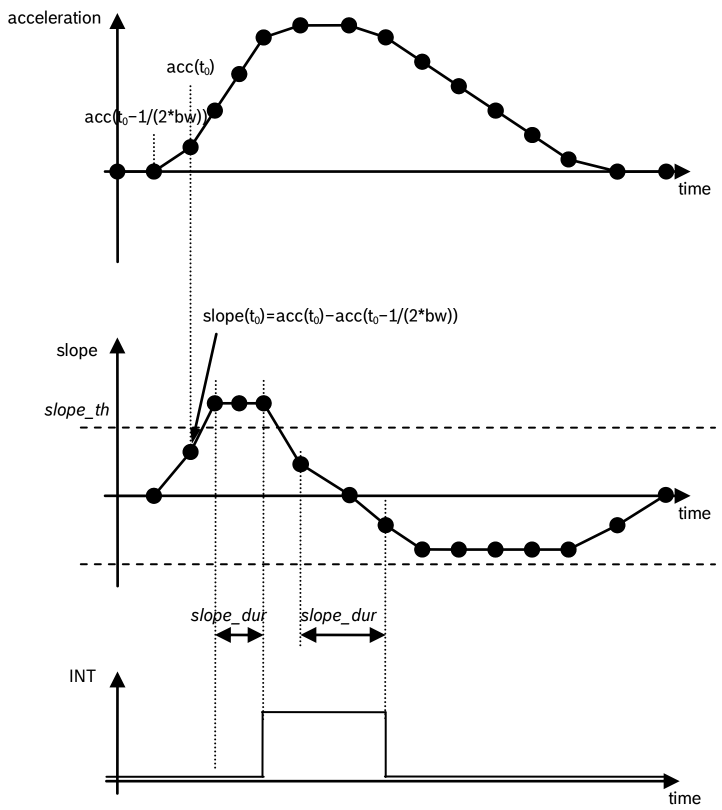


Figure 6: Principle of any-motion detection

The threshold is set with the value of register *(0x28) slope_th*. 1 LSB of *(0x28) slope_th* corresponds to 1 LSB of acceleration data. Therefore, an increment of *(0x28) slope_th* is 15.6 mg in 2g-range (31.3 mg in 4g-range, 62.5 mg in 8g-range and 125 mg in 16g-range). And the maximum value is 996 mg in 2g-range (1.99g in 4g-range, 3.98g in 8g-range and 7.97g in 16g-range).

The time difference between the successive acceleration signals depends on the selected bandwidth and equates to 1/(2*bandwidth) ($\Delta t=1/(2*bw)$). In order to suppress failure signals, the interrupt is only generated (cleared) if a certain number *N* of consecutive slope data points is larger (smaller) than the slope threshold given by *(0x28) slope_th*. This number is set by the *(0x27) slope_dur* bits. It is *N = (0x27) slope_dur* + 1 for *(0x27)*.

Example: *(0x27) slope_dur* = 00b, ..., 11b = 1decimal, ..., 4decimal