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

# BMA220
## Digital, triaxial acceleration sensor

## Data sheet

Bosch Sensortec

BOSCH
Invented for life


**BMA220 data sheet**

| | |
|---|---|
| Ordering code | Please contact your Bosch Sensortec representative for the ordering code |
| Package type | 12-pin LGA |
| Document revision | 1.10 |
| Document release date | 13 April 2011 |
| Document number | BST-BMA220-DS003-07 |
| Technical reference code(s) | 0 273 141 102 |
| Notes | Specifications are subject to change without notice.<br>Product photos and pictures are for illustration purposes only and may differ from the real product's appearance. |

# BMA220

## Triaxial ±2g to ±16g acceleration sensor with on-chip motion-triggered interrupt controller

**Key features**

| | |
|---|---|
| • Three-axis accelerometer | |
| • Ultra-Small package | Mold package (LGA 12ld),<br>Footprint 2mm x 2mm, height 0.98mm |
| • Digital interface | SPI (4-wire, 3-wire), I²C, interrupt pin<br>I/O supply voltage range: 1.6V to 3.6V |
| • Programmable functionality | Acceleration ranges ±2g/±4g/±8g/±16g<br>Bandwidth 1kHz . . . 32Hz<br>Self test |
| • On-chip interrupt controller | Motion-triggered interrupt-signal generation for<br>- orientation recognition<br>- any-motion detection<br>- tap/double tap sensing<br>- low-g/high-g detection<br>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
- Tap/double tap sensing
- Menu scrolling
- Gaming
- Drop detection for warranty logging
- Advanced system power management for mobile applications

**General description**

The BMA220 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 interface 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 sensor offers a variable I/O supply voltage range from 1.6V 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 BMA220 senses tilt, motion and shock vibration in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

# CONTENT

## 1 Specification

If not stated otherwise, the given values are maximum values over lifetime and full performance temperature and voltage ranges. Min/max. data represent 3-sigma values.

**Table 1: Operating conditions, output signal and mechanical characteristics**

| Parameter | Symbol | Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| **OPERATING CONDITIONS** | | | | | | |
| Acceleration Range | $g_{FS2g}$ | switchable via serial digital interface | | ±2.0 | | g |
| | $g_{FS4g}$ | | | ±4.0 | | g |
| | $g_{FS8g}$ | | | ±8.0 | | g |
| | $g_{FS16g}$ | | | ±16.0 | | g |
| Supply Voltage Analog Domain | $V_{DDA}$ | | 1.62 | 1.8 | 1.98 | V |
| Supply Voltage Digital Domain | $V_{DDD}$ | | 1.62 | 1.8 | 1.98 | V |
| Supply Voltage I/O Domain | $V_{DDIO}$ | | 1.62 | | 3.6 | V |
| Voltage Input Low Level | $V_{IL\ SPI}$ | SPI | | | $0.1*V_{DDIO}$ | V |
| | $V_{IL\ I2C}$ | I²C | | | $0.3*V_{DDIO}$ | V |
| Voltage Input High Level | $V_{IH\ SPI}$ | SPI | $0.9*V_{DDIO}$ | | | V |
| | $V_{IH\ I2C}$ | I²C | $0.7*V_{DDIO}$ | | | V |
| Voltage Output High Level | $V_{OH}$ | SPI & I²C | | $V_{DDIO}$ | | V |
| Voltage Output Low Level | $V_{OL\ SPI}$ | SPI | | GND | | V |
| | $V_{OL\ I2C}$ | I²C, $R_P \geq 680\ \Omega$ | | | $0.2*V_{DDIO}$ | V |
| Supply Current in Normal Mode | $I_{DD}$ | Nominal $V_{DD}$ supplies at $T_A$=25°C | | 250 | | μA |
| Supply current in Low Power Mode | $I_{DDsl}$ | Nominal $V_{DD}$ supplies $T_A$=25°C, BW = 1kHz sleep dur. > 25ms | | < 10 | | μA |
| Supply Current in Suspend Mode | $I_{DDsd}$ | Nominal $V_{DD}$ supplies at $T_A$=25°C | | | < 1 | μA |
| Wake-Up Time | $t_{w\_up}$ | from sleep/suspend mode @1kHz bw | | | 1.2 | ms |
| Start-Up Time | $t_{s\_up}$ | POR @1kHz bw | | 1.5 | | ms |
| Operating Temperature | $T_A$ | | -40 | | +85 | °C |

| | | | | | | |
|---|---|---|---|---|---|---|
| **OUTPUT SIGNAL** | | | | | | |
| Device resolution | $D_{res}$ | $g_{FS2g}$ | | 62.5 | | mg |
| Sensitivity | $S_{2g}$ | $g_{FS2g}$, $T_A$=25°C | | 16 | | LSB/g |
| | $S_{4g}$ | $g_{FS4g}$,$T_A$=25°C | | 8 | | LSB/g |
| | $S_{8g}$ | $g_{FS8g}$, $T_A$=25°C | | 4 | | LSB/g |
| | $S_{16g}$ | $g_{FS16g}$, $T_A$=25°C | | 2 | | LSB/g |
| Sensitivity Temperature Drift | TCS | -40°C ≤ $T_A$ ≤ +85°C | | ±0.03 | | %/K |
| Zero-g Offset | Off | $T_A$=25°C, $V_{DDA}$=1.8V, $g_{FS2g}$ | | ±95 | | mg |
| Zero-g Offset Temperature Drift | TCO | -40°C ≤ $T_A$ ≤ +85°C, $g_{FS2g}$ | | ±2 | | mg/K |
| Bandwidth | bw | $1^{st}$ order filter, switchable | | 32/64/ 125/250/ 500/1000 | | Hz |
| Nonlinearity | NL | best fit straight line | | ±2 | | %FS |
| Self Test Response | TSTx | depending on sensitivity/ acceleration range | | 0.7g*$S_x$ | | LSB |
| | TSTy | | | 2.0g*$S_y$ | | LSB |
| | TSTz | | | 0.6g*$S_z$ | | LSB |
| Output Noise | $n_{rms}$ | rms, Nominal $V_{DD}$ supplies $T_A$=25°C, BW = 1kHz | | 2 | | mg/√Hz |
| **MECHANICAL CHARACTERISTICS** | | | | | | |
| Cross Axis Sensitivity | $\overline{S}$ | relative contribution between 3 axes | | 2 | | % |
| Alignment Error | $\delta_a$ | relative to package outline | | ±0.5 | | ° |

## 2 Absolute maximum ratings

All voltages below are given with respect to GND.

**Table 2: Absolute maximum ratings**

| Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|
| Voltage at supply pad | $V_{DDD}$ and $V_{DDA}$ | -0.3 | 2.0 | V |
| | $V_{DDIO}$ | -0.3 | 4.25 | V |
| Voltage at any logic pad | Non-supply pad | -0.3 | $V_{DDIO}$+0.3 | V |
| Storage temperature range | rel. humidity <=65% | -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

The following figure describes the functionality of the two basic parts of the sensor module, namely mechanical sensor element and evaluating ASIC.

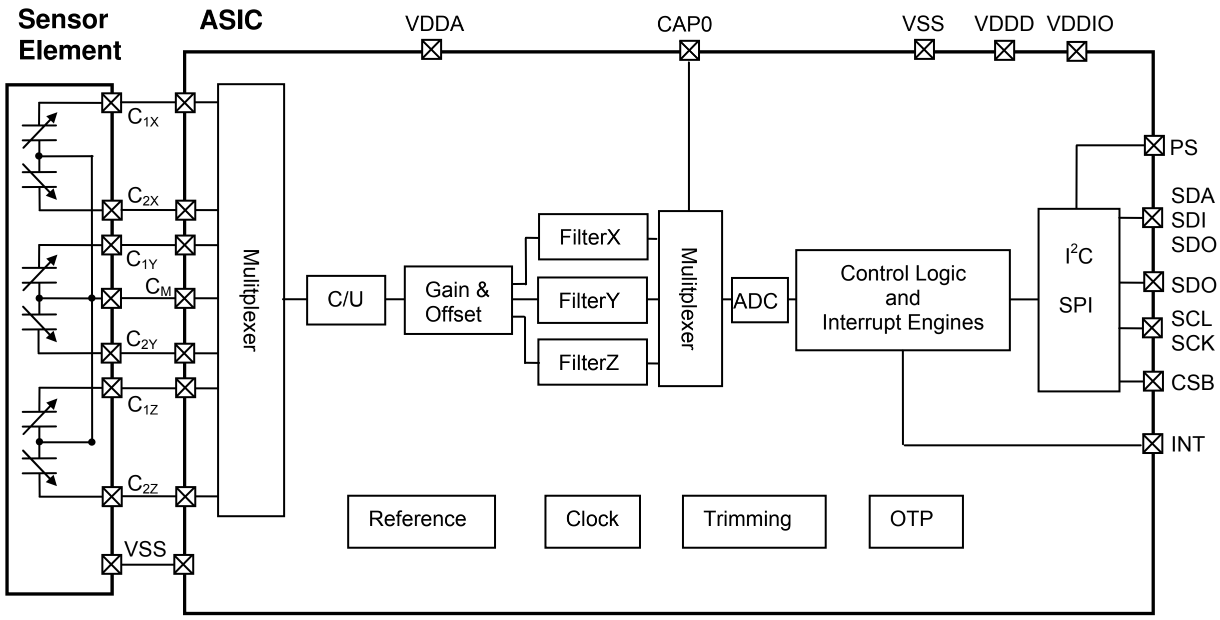



Figure 1: Block diagram BMA220

## 4 Global memory map

The memory map below shows all externally accessible data registers which are needed to operate BMA220.The left columns show the memory addresses. The columns in the middle depict the content of each register bit. The colors of the bits indicate whether they are read-only, write-only or read- and writable. The memory is volatile so that the writable content has to be re-written after each power-on.
The extended address space greater than 0x19 (SPI) / 0x32 ($I^2C$) is not shown. These registers are reserved for further Bosch factory testing and trimming.

| Register Address (I²C) | Register Address (SPI) | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 | default after power-up |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x32 | 0x19 | softreset | | | | | | | | 0x00 |
| 0x30 | 0x18 | suspend | | | | | | | | 0x00 |
| 0x2E | 0x17 | unused | | | | | WDT_TO_en | WDT_TO_sel | SPI3 | 0x00 |
| 0x2C | 0x16 | reserved | | | | | | | | 0x00 |
| 0x2A | 0x15 | reserved | | | | | | | | 0x00 |
| 0x28 | 0x14 | reserved | | | | | | | | 0x00 |
| 0x26 | 0x13 | reserved | | | | | | | | 0x00 |
| 0x24 | 0x12 | reserved | | | | | | | | 0x10 |
| 0x22 | 0x11 | unused | | | sbist_sign | sbist (off,x,y,z) | | range[1:0] | | 0x00 |
| 0x20 | 0x10 | serial_high_bw | unused | | | filt_config[3:0] | | | | 0x00 |
| 0x1E | 0x0F | unused | sleep_en | sleep_dur[2:0] | | | en_x_channel | en_y_channel | en_z_channel | 0x07 |
| 0x1C | 0x0E | reset_int | lat_int[2:0] | | | en_low | en_high_x | en_high_y | en_high_z | 0x00 |
| 0x1A | 0x0D | en_data | en_orient | en_slope_x | en_slope_y | en_slope_z | en_tt_x | en_tt_y | en_tt_z | 0x00 |
| 0x18 | 0x0C | unused | | | tt_int | low_int | high_int | data_int | slope_int | 0x00 |
| 0x16 | 0x0B | orient_int | orient[2:0] | | | int_first_x | int_first_y | int_first_z | int_sign | 0x00 |
| 0x14 | 0x0A | unused | | | tip_en | orient_blocking [1:0] | | tt_samp[1:0] | | 0x08 |
| 0x12 | 0x09 | orient_ex | slope_filt | slope_th[3:0] | | | | slope_dur[1:0] | | 0x45 |
| 0x10 | 0x08 | tt_filt | tt_th[3:0] | | | | tt_dur[2:0] | | | 0xB5 |
| 0xE | 0x07 | low_hy[1:0] | | low_dur[5:0] | | | | | | 0x7F |
| 0xC | 0x06 | low_th[3:0] | | | | high_th[3:0] | | | | 0x4E |
| 0xA | 0x05 | high_hy[1:0] | | high_dur[5:0] | | | | | | 0x7F |
| 0x8 | 0x04 | acc_z<5:0> | | | | | | 0 | 0 | 0x00 |
| 0x6 | 0x03 | acc_y<5:0> | | | | | | 0 | 0 | 0x00 |
| 0x4 | 0x02 | acc_x<5:0> | | | | | | 0 | 0 | 0x00 |
| 0x2 | 0x01 | Revision ID | | | | | | | | 0x00 |
| 0x0 | 0x00 | Chip ID | | | | | | | | 0xDD |

| | |
|---|---|
| (violet) | w/r |
| (yellow) | write only |
| (green) | read only |

Figure 2: Memory map

Note: From SPI → I²C use burst address increment in 0x02h steps.

## 4.1 Control registers

### 4.1.1 3-wire SPI mode selection

The BMA220 supports both 4-wire and 3-wire SPI. The protocols are exactly the same except for the fact that in 3-wire mode, the SDI pin is also used for data output.
The default mode is 4-wire SPI. If 3-wire SPI should be used, the SPI3 bit in register 0x17 (SPI) must be set to ‘1’.

### 4.1.2 Low-power mode configuration

The BMA220 supports a low-power mode. In this low-power mode, the chip wakes up periodically, enables the interrupt controller and goes back to sleep if no interrupt has occurred. The procedure is the following:

1. Wake-up
2. Enable analog front-end and convert acceleration data until the low-pass filters have settled.
3. Enable integrated interrupt controller and evaluate interrupt conditions.
   Once the interrupt conditions have been evaluated and **no** interrupt has occurred, the chip goes back to sleep. If no interrupt is enabled, the acceleration for x-, y- and z-axes are converted once and then the chip goes back to sleep.
4. Sleep for the programmed duration

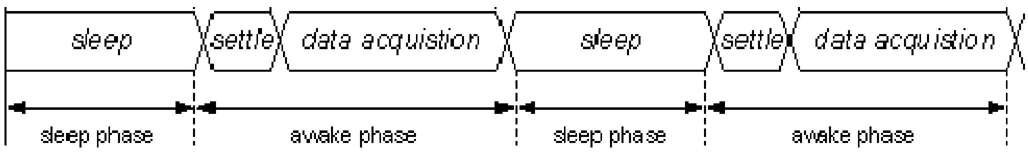



Figure 3: sleep and awake phases

The low-power mode can be enabled by setting the `sleep_en` bit in Reg. 0x0F (SPI) / 0x1E ($I^2C$) and by enabling the data ready interrupt (or any other interrupt, see chapter 5). The sleep duration can be configured via the `sleep_dur` bits in Reg 0x0F (SPI) / 0x1E ($I^2C$).

| sleep_dur setting | Sleep Duration |
|---|---|
| 000 | 2ms |
| 001 | 10ms |
| 010 | 25ms |
| 011 | 50ms |
| 100 | 100ms |
| 101 | 500ms |
| 110 | 1s |
| 111 | 2s |

Table 3: Sleep durations for low-power mode

### 4.1.3 Low-power mode dimensioning

The power saving that can be achieved depends on the programmed sleep duration and the configured bandwidth. Figure 4 explains the power consumption in relation to the different ASIC states (sleep and awake phases).

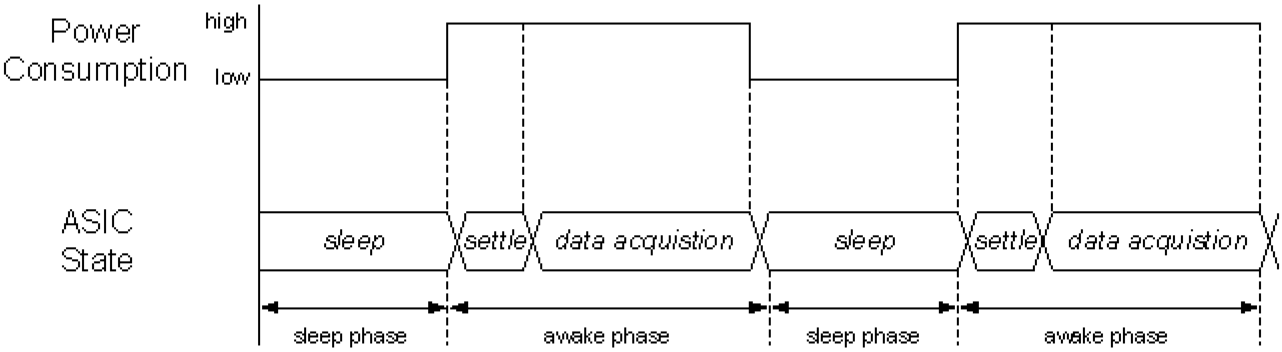



Figure 4: Sleep and awake phase

If a low bandwidth is selected, the time required for filter settling might be the dominating time. Refer to table 2 for the appropriate dimensioning of the attainable current saving.

| Filter cut-off | Settling time |
|---|---|
| 32 Hz | 16690µs |
| 64 Hz | 8690µs |
| 125 Hz | 4690µs |
| 250 Hz | 2690µs |
| 500 Hz | 1690µs |
| 1000 Hz | 1190µs |

Table 4: Approximate awake phase times for 1 data sample

### 4.1.4 Channel activation / de-activation

In order to optimize further power consumption of the BMA220, data evaluation of individual axes can be deactivated. Per default, all three axes are active. If the user wants to disable one or more axes, the appropriate `en_?_channel` bits at address 0x0F (SPI) / 0x1E ($I^2C$) must be set to '0'.

### 4.1.5 Soft-reset

The BMA220 can be put into a soft-reset state by performing a read from the soft-reset address. To bring the chip back into operation, another read must be performed from the same memory address. The reading returns value 0xFF if the system was in soft reset mode; otherwise it returns value 0x00.
Please note that all internal configuration data programmed by the user will be lost.

### 4.1.6 Suspend mode

The BMA220 can be put into a suspend mode e.g. to easily achieve a power consumption below 1µA by performing a read from the suspend address. To bring the chip back into normal mode operation, another read must be performed from the same memory address. The reading returns value 0xFF if the system was in suspend mode, otherwise it returns value 0x00.

Please note, that during suspend, all analog modules expect for power-on-reset will be disabled. Only reads through the serial interface are supported during suspend.

## 4.2 Setting registers

### 4.2.1 Acceleration range and sensitivity setting

The BMA220 has four different range settings for the full scale acceleration range. In dependence of the use case always the lowest full scale range with the maximum resolution should be selected. Please refer to literature to find out, which full scale acceleration range, which sensitivity or which resolution is the ideal one.

This can be configured via the register bits `range`[1:0] at address 0x11 (SPI) / 0x22 (SPI).

The following table shows the range bits with corresponding scale and resolution.

| range[1:0] | Full Scale | Sensitivity | Resolution | Example use case |
|---|---|---|---|---|
| '00' | ±2g | 16 LSB / g | 62.5mg / LSB | Orientation recognition |
| '01' | ±4g | 8 LSB / g | 125mg / LSB | Gaming |
| '10' | ±8g | 4 LSB / g | 0.25g / LSB | |
| '11' | ±16g | 2 LSB / g | 0.5g / LSB | Shock vibration detection |

Table 5: Acceleration resolution

### 4.2.2 Filter and bandwidth configuration

The BMA220 has a digital filter that can be configured by setting the corresponding register bits `filter_config[3:0]` at address 0x10 (SPI) / 0x20 ($I^2C$). For compatibility reasons the settings are defined based on BMA120. To always ensure an ideal cut off frequency of the filter the BMA220 is adjusting the sample rate automatically.

| **filter_config[3:0]** | 0x5 | 0x4 | 0x3 | 0x2 | 0x1 | 0x0 |
|---|---|---|---|---|---|---|
| **digital filter cut-off frequency** | 32Hz | 64Hz | 125Hz | 250Hz | 500Hz | 1kHz |

Table 6: Digital filter configuration

The internal SC-filter has a fix cut-off frequency at 1 KHz. In addition to the internal SC-filter a digital filter is available which is providing a filtered and an unfiltered data stream for all of the 3 axes of acceleration.

If application specific reasons require a bandwidth configuration <32Hz, please contact your Bosch Sensortec representative.

## 4.3 Data registers

### 4.3.1 Acceleration data read-out

The acceleration data can be read-out through addresses 0x02 (SPI) / 0x04 ($I^2C$) through 0x04 (SPI) / 0x08 ($I^2C$). The acceleration data is in 2's complement according to the table below.

An efficient way to read out the acceleration data in I²C or SPI mode is the burst-accesses. During such an access, the BMA220 automatically increments the read address after each byte. By using this kind of access, the data transferred over the I²C bus can be reduced by up to 50%.

| Decimal value | Acceleration (in 2g range mode) |
|---|---|
| + 31 | + 1.94g |
| … | … |
| 0 | 0g |
| … | … |
| -32 | - 2.0g |

Table 7: Acceleration register content

Per default, the bandwidth of the data being read-out is limited by the internal low-pass filters according to the filter configuration. However, it is possible to read-out data only 1st order filtered (1 kHz) even though the internal filters are configured differently.

The reason for this feature is that the interrupt controller may operate on low-bandwidth data while the external master still needs to operate on high-bandwidth data.

Unfiltered (1kHz high-bandwidth) data can be read-out through the serial interface when the `serial_high_bw` bit is set to '1'. Per default, filtered data is read-out through the serial interface.

### 4.3.2 Chip / revision ID

The chip ID and the revision ID can be read-out through addresses 0x00 (SPI & $I^2C$) and 0x01 (SPI) / 0x02 ($I^2C$).

| Chip ID | Revision ID |
|---|---|
| 0xDD | 0x00 |

Table 8: Chip and revision ID

## 4.4 Interrupt control registers

The BMA220 features a programmable interrupt controller that directly supports common mobile applications like tap sensing detection, orientation recognition and any-motion detection.

Supported types of interrupts:

- Any-motion (slope) detection
- Tap/double-tap sensing
- Orientation recognition
- Low-g detection
- High-g detection
- Data-ready interrupt

The configuration and status register bits of all interrupt engines and the exact interrupt functionality are given in chapter 5.

# 5 Interrupt controller

The BMA220 integrates a programmable interrupt controller. It can be configured via SPI/I²C to monitor individual axes (X-, Y- and Z-axis) and check whether certain conditions apply (e.g. the acceleration on one axis exceeds a certain threshold). The interrupt controller of the BMA220 is capable of checking for certain conditions simultaneously.

If at least one of the configured conditions applies, an interrupt (logic '1') is issued through the INT pin of the sensor. More details about the triggering condition (e.g. the type of the interrupt or the axis that triggered the interrupt) are saved in internal status registers and can be read out through the digital interface.

**It is recommended to reset the interrupt controller by setting reset_int to '1' when the interrupt settings has been set or changed.**

## 5.1 Latched vs. non-latched modes

The interrupt controller can be used in two modes

- Latched mode: Once one of the configured interrupt conditions applies, the INT pin is asserted and must be reset by the external master through the digital interface.
- Non-Latched mode: The interrupt controller clears the INT signal once the interrupt condition no longer applies.

The interrupt output can be programmed by *lat_int[2:0]* to be either unlatched ('000') or latched permanently ('111') or have the latch time of 0.25s('001')/0.5s('010')/1s('011')/2s('100')/4s ('101')/8s('110'). The setting of these bits applies to all types of interrupts.

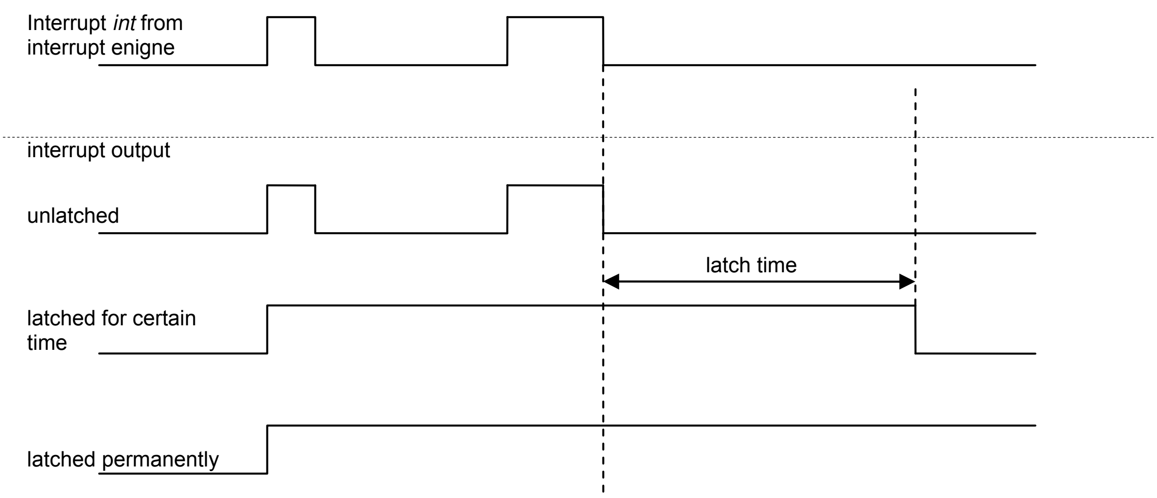



Figure 5: Interrupt output

## 5.2 Supported types of interrupts

The following interrupt modes are provided by the BMA220.

- Any-motion (slope) detection
- Tap/double-tap sensing
- Orientation recognition
- Low-g detection
- High-g detection
- Data-ready interrupt

## 5.3 Power-saving modes

In order to reduce power consumption of the sensor itself, the BMA220 supports a low-power mode in which the ASIC wakes up periodically, checks whether any of the configured interrupt conditions apply and then either goes back to sleep (no interrupt) or stays awake (interrupt).

The BMA220's PMU (power management unit) controls the transitions from the 'awake state' into the 'sleep state' and vice versa.

In normal mode, the interrupt controller is permanently turned on to continuously process the incoming data. In low-power mode, the interrupt controller will be turned on by the PMU once the chip has fully woken up. The time it takes before the sensor can go back to sleep is determined by the active interrupt engines. Once all active engines indicate that no interrupt condition applies, the PMU will switch the sensor back into sleep state (please refer to section 0 for more details).

Furthermore the applied interrupt condition can be used not only to enable the low-power mode of the sensor itself but for the whole system. This enables a dramatically reduced power consumption of the whole system. The result is an extended operation and stand-by time e.g. of mobile devices in an order of magnitude.

### 5.4 Any-motion (slope) detection

The any-motion detection uses the slope between two successive acceleration signals to detect changes in motion. It generates an interrupt when a preset threshold *slope_th* is exceeded. The threshold can be configured between 0 and the maximum acceleration value corresponding to the selected measurement range. The time difference between the successive acceleration signals depends on the bandwidth of the configurable low pass filter and corresponds roughly to 1/(2*bandwidth) ($\Delta t=1/(2*bw)$).

In order to suppress failure signals, the interrupt is only generated if a certain number *slope_dur* of consecutive slope data points is above the slope threshold *slope_th*.

If the same number of data points falls below the threshold, the interrupt is reset.

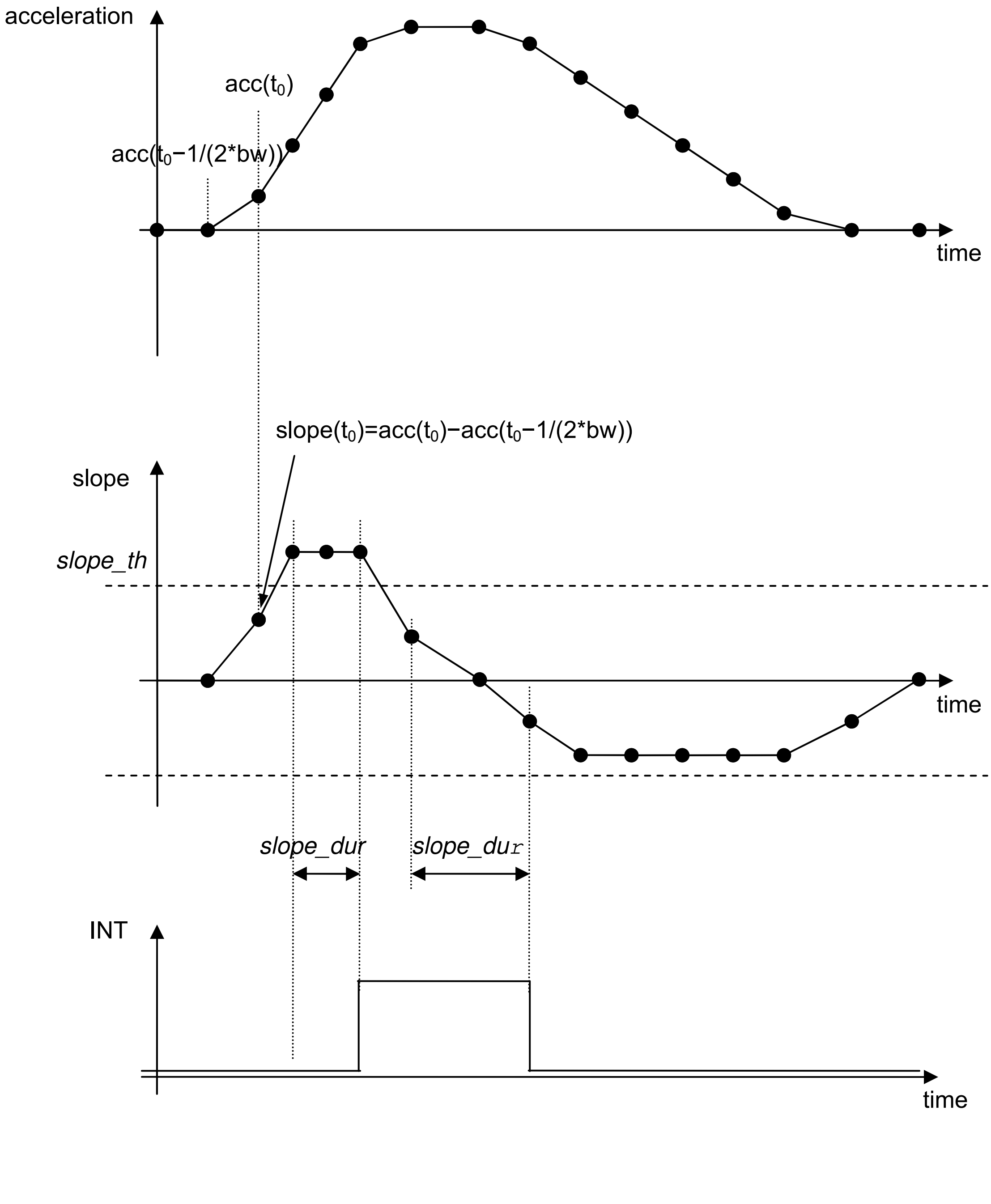
The criteria for any-motion detection are fulfilled and the slope interrupt is generated if any of the enabled channels exceeds the threshold *slope_th* for *slope_dur* consecutive times. As soon as all the enabled channels fall or stay below this threshold for *slope_dur* consecutive times the interrupt is reset unless interrupt signal is latched.

The any-motion interrupt logic sends out the signals of the axis that has triggered the interrupt (*slope_first_x*, *slope_first_y*, *slope_first_z*) and the signal of motion direction (*slope_sign*).

When serial interface is active, any-motion detection logic is enabled if any of the any-motion enable register bits is set. To disable the any-motion interrupt, clear all the axis enable bits.

In the dedicated wake-up mode (0), all three axes are enabled for any-motion detection whether the individual axis enable bits are set or not.

Figure 6: Any-motion (slope) interrupt detection




The following table shows the signals used in any-motion detection. After reset, a default value will be assigned to each register.

| **Name** | **Register Address (SPI) *** | **Description** | **Number of bits** | **Reset-value** |
|---|---|---|---|---|
| en_slope_x<br>en_slope_y<br>en_slope_z | 0x0D.5<br>0x0D.4<br>0x0D.3 | enable slope detection on x-axis<br>enable slope detection on y-axis<br>enable slope detection on z-axis | 3 | "000" |
| slope_th | 0x09[5:2] | define the threshold level of the slope<br>1 LSB threshold is 1 LSB of acc_data | SLOPE_TH_NUM<br>4 | SLOPE_TH_INIT ("0001") |
| slope_dur | 0x09[1:0] | define the number of consecutive slope data points above *slope_th* which are required to set the interrupt<br>("00" = 1,"01" = 2,"10" = 3, "11" = 4) | SLOPE_DUR_NUM<br>2 | SLOPE_DUR_INIT ("01") |
| slope_filt | 0x09.6 | defines whether filtered or unfiltered acceleration data should be used (evaluated)<br>('0'=unfiltered, '1'=filtered) | 1 | '1' |
| slope_int | 0x0C.0 | whether slope interrupt has been triggered | 1 | '0' |
| slope_first_x<br>slope_first_y<br>slope_first_z | | whether x-axis has triggered the interrupt (0=no, 1=yes)<br>whether y-axis has triggered the interrupt (0=no, 1=yes)<br>whether z-axis has triggered the interrupt (0=no, 1=yes) | 3 | "000" |
| slope_sign | | global register bit for all interrupts<br>define the slope sign of the triggering signal (0=positive slope, 1=negative slope) | 1 | '0' |

* For determining the corresponding $I^2C$ register address, please refer to figure 2 in chapter 4 (memory map)

Table 9: Control and status register for any motion detection

## 5.5 Tap sensing

Tap sensing has the same functionality as a common laptop touch-pad. If 2 taps occur within a short time, a pre-defined action will be performed by the system. If time between 2 taps is too long or too short no action happens.

When the serial interface is activated, tap sensing is enabled if any of the tap sensing enable register bits are set (*en_tt_x, en_tt_y, en_tt_z*). To disable the tap sensing interrupt, clear all the axis enable bits.

When the preset threshold *tt_th* is exceeded, a tap-shock is detected. The tap sensing interrupt is generated only when a second tap is detected within a specified period of time.

The slope between two successive acceleration data has to exceed *tt_th* to detect a tap-shock. The time difference between the two successive acceleration values depends on the bandwidth of the low pass filter. It roughly corresponds to 1/(2*bandwidth).

The time delay *tt_dur* between two taps is typically between 12,5ms and 500ms. The threshold is typically between 0.7g and 1.5g in 2g measurement range. Due to different coupling between sensor and device shell (housing) and different measurement ranges of the sensor these parameters are configurable.

The criteria for tap sensing are fulfilled and the interrupt is generated if the second tap occurs after *tap_quiet* and within *tt_dur*. The tap sensing direction is determined by the $1^{st}$ tap. During *tt_quiet* period (30ms) no taps should occur. If a tap occurs during *tap_quiet* period it will be connoted as new tap.

The slope detection interrupt logic stores the direction of the (first) tap-shock in a status register. This register will be locked for *tap_shock*=50ms in order to prevent other slopes to overwrite this information.

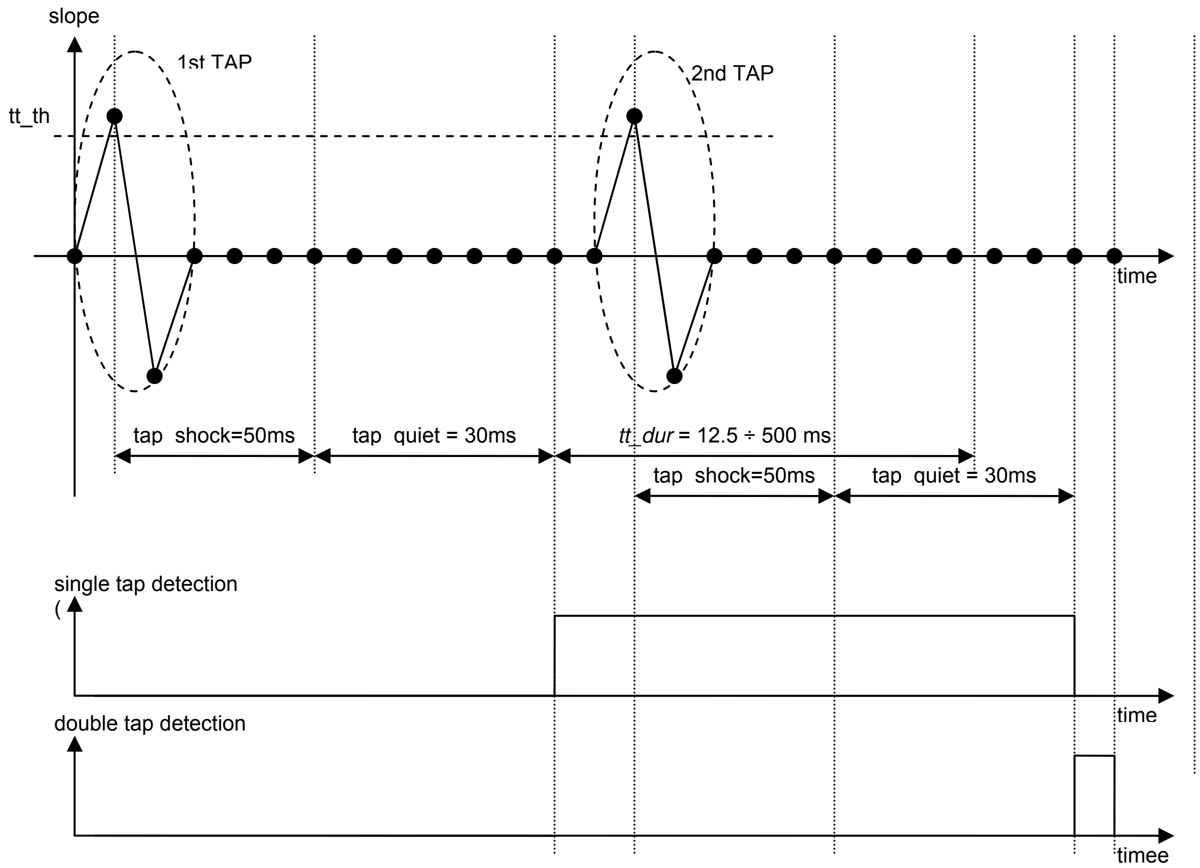
When a tap sensing interrupt is triggered, the signals of the axis that has triggered the interrupt (*tt_first_x*, *tt_first_y*, *tt_first_z*) and the signal of motion direction (*tt_sign*) are stored in the corresponding registers.

The axis on which the biggest slope occurs will trigger the first tap. The second tap will be triggered by any axis (not necessarily same as the first tap).

The register *tap_en* defines whether single tap or double tap shall be detected.

In dedicated tap sensing mode, all three axes are enabled for double tap sensing detection.

Figure 7: Tap sensing interrupt detection




When a tap-sensing interrupt is triggered, the following details can be read from the corresponding registers: the axis that has triggered the interrupt (*tt_first_x*, *tt_first_y*, *tt_first_z*) and the motion direction of the triggering acceleration signal (*tt_sign*).

| Name | Register Address (SPI) * | Description | Number of bits | Reset-value |
|---|---|---|---|---|
| en_tt_x<br>en_tt_y<br>en_tt_z | 0x0D.2<br>0x0D.1<br>0x0D.0 | enable tap sensing detection on x-axis<br>enable tap sensing detection on y-axis<br>enable tap sensing detection on z-axis | 3 | "000" |
| tt_th | 0x08[6:3] | define the threshold level of the tap sensing slope<br>1 LSB is 2*(LSB of acc_data) | TT_TH_NUM<br>4 | TT_TH_INIT<br>"0110" |
| tt_dur | 0x08[2:0] | define the maximum delay of the second tap after the shock suppression (50, 105, 150, 219, 250, 375, 500, 700)ms | TT_DUR_NUM<br>3 | TT_DUR_INIT<br>"101" |
| tt_filt | 0x08.7 | defines whether filtered or unfiltered acceleration data should be used (evaluated) ('0'=unfiltered, '1'=filtered) | 1 | '1' |
| tip_en | 0x0A.4 | whether tap or double-tap shall be detected (0= double tap, 1=tap) | 1 | '0' |
| tt_int | 0x0C.4 | whether tap sensing interrupt has been triggered (0=no, 1=yes) | 1 | '0' |
| *tt_samp* | 0x0A[1:0] | number of data to be sampled after wake-up<br>'00' => 2 data<br>'01' => 4 data<br>'10' => 8 data<br>'11' => 16 data | 2 | '00' |
| tt_first_x<br>tt_first_y<br>tt_first_z | | whether x-axis has triggered the interrupt (0=no, 1=yes)<br>whether y-axis has triggered the interrupt (0=no, 1=yes)<br>whether z-axis has triggered the interrupt (0=no, 1=yes) | 3 | "000" |
| tt_sign | | give the slope sign of the triggering signal (0=positive, 1=negative) | 1 | '0' |

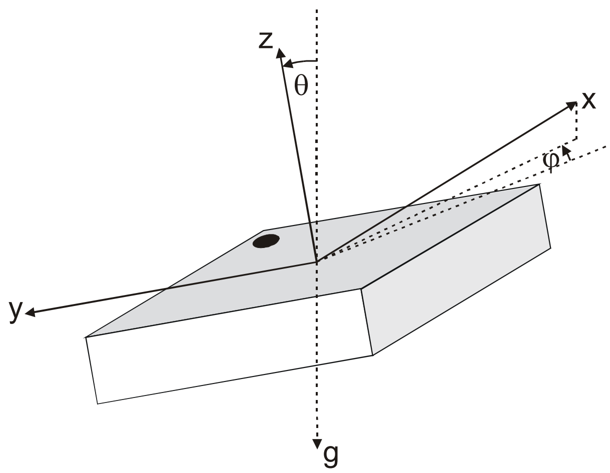
* For determining the corresponding $I^2C$ register address, please refer to figure 2 in chapter 4 (memory map)

Table 10: Control and status register for tap sensing

### 5.6 Orientation recognition

The orientation recognition feature informs on an orientation change of the sensor with respect to the gravitational field vector g. The measured acceleration vector components with respect to the gravitational field look as follows.

Figure 8: Definition of acceleration-vector components




[with respect to graviational field vector g (black dot = pin 1 identifier) ]

acc_x = $1g \cdot \sin\theta \cdot \cos\varphi$
acc_y = $-1g \cdot \sin\theta \cdot \sin\varphi$
acc_z = $1g \cdot \cos\theta$
→ acc_y/acc_x = $-\tan\varphi$

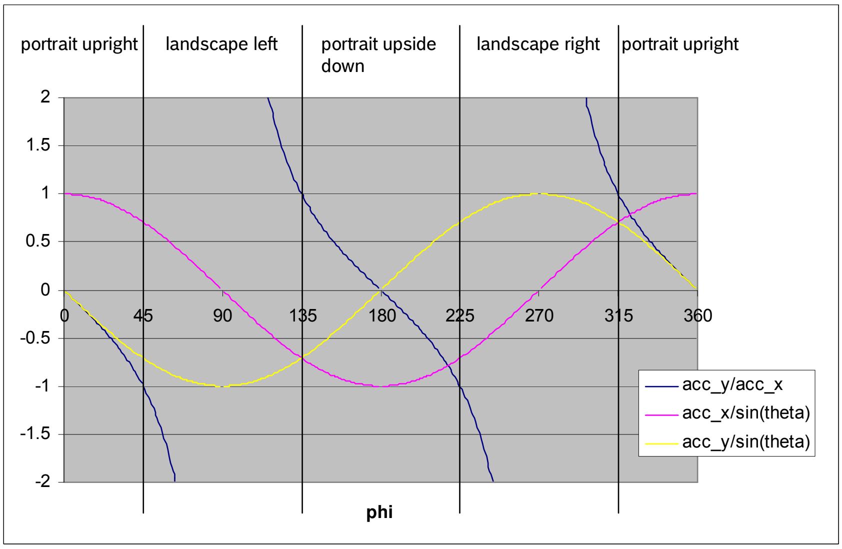
The output register is called *orient* and defined in the following way:

'0xx' upward looking (z>0):

| | |
|---|---|
| '000' portrait upright (315°<φ<45°) | → \|acc_y/acc_x\|<1 && acc_x>0 |
| '001' portrait upside down (135°<φ<225°) | → \|acc_y/acc_x\|<1 && acc_x<0 |
| '010' landscape left (45°<φ<135°) | → \|acc_y/acc_x\|>1 && acc_y<0 |
| '011' landscape right (225°<φ<315°) | → \|acc_y/acc_x\|>1 && acc_y>0 |

'1xx' downward looking (z<0), xx as before

Figure 9: Orientation definition and interrupt thresholds with respect to the angle phi φ




The criteria for portrait/landscape switching is fulfilled and the interrupt is generated when the threshold |acc_y/acc_x|=1 is crossed (i.e. 45°, 135°, 225°, 315°). As soon as the interrupt is set, no new interrupt is generated within the hysteresis level of 0.66<|acc_y/acc_x|<1.66 corresponding to a hysteresis interval of ±13% around the threshold.

It is possible to block the orientation detection depending on certain conditions (no orientation interrupt will be triggered). This orientation interrupt blocking feature is configurable via the *orient_blocking*[1:0] bits in the following manner:

- ‘00’ → interrupt blocking is completely disabled
- ‘01’ → no interrupt is generated, when |z|>0.9g OR |x|+|y| < 0.4g OR when the slopes of the acceleration data exceeds 0.2g (sample-to-sample).
- ‘10’ → no interrupt is generated, when |z|>0.9g OR |x|+|y| < 0.4g OR while the slopes of the acceleration data exceeds 0.3g (sample-to-sample).
- ‘11’ → no interrupt is generated, when |z|>0.9g OR |x|+|y| < 0.4g OR while the slopes of the acceleration data exceeds 0.4g (sample-to-sample).

For all states where interrupt blocking through slope detection is used, the interrupt should be re-enabled after the slope has been below the threshold for 3 times in a row.

For all states where interrupt blocking is enabled, in order to trigger the interrupt, the orientation should remain the same (stable) until the timer runs out (for ~100ms). The timer starts to count when orientation changes between two consecutive samples. If the orientation changes while timer is still counting, the timer is restarted.

The criteria for switching from upward to downward looking fulfilled and the interrupt is generated when the threshold z=0g is crossed. As soon as the interrupt is set, no new interrupt is generated within the hysteresis level of -0.4g<z<0.4g (i.e. ±25° tilt around vertical position).