BMA530

Advanced, ultra-small, triaxial low-g accelerometer with digital interfaces

BMA530 Datasheet

Document revision	1.0
Document release date	May 14th 2024
Document number	BST-BMA530-DS000-03
Sales part numbers	0 273 017 042
Notes	Data and descriptions 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 appearance.

Basic Description

The BMA530 is an advanced,ultra-small, triaxial low-g accelerometer with digital interfaces. The sensor is suitable for low-power and demanding consumer electronics applications. The BMA530 integrates

- a 16 bit digital, triaxial accelerometer with range configurable to $\pm 2\,\mathrm{g}$, $\pm 4\,\mathrm{g}$, $\pm 8\,\mathrm{g}$, $\pm 16\,\mathrm{g}$
- a 8 bit digital temperature sensor for an operating temperature range $-40\,^{\circ}\mathrm{C}$ ... $85\,^{\circ}\mathrm{C}$

Key Features

- Compact size $1.2 \times 0.8\,\mathrm{mm}^2$ Wafer Level Chip Scale Package (WLCSP), 6 pins, height $0.55\,\mathrm{mm}$
- Primary digital interface with 10 $M\mathrm{Hz}$ slave SPI (4-wire, 3-wire), 12.5 $M\mathrm{Hz}$ I3C and up to 1 $M\mathrm{Hz}$ I²C (Fm+)
- Sample rates (output data rates ODR): $1.5625\,\mathrm{Hz} \ldots$ 6.4 $k\mathrm{Hz}$ (nominal)
- Programmable low-pass filtering
- Wide power supply range: analog VDD $1.62\,\mathrm{V}$ ... $3.63\,\mathrm{V}$
- Ultra low current consumption: typ. 125μA (in full ODR and aliasing free operation)
- Built-in power management unit (PMU) for advanced power management and low power modes
- Power on time: 1.8ms for communication readiness
- up to 1 KB on-chip FIFO buffer for accelerometer, temperature sensor and sensor time stamps
- Fast offset error compensation for accelerometer
- Sensor time stamps for accurate system (host) and sensor time synchronization
- Two independent programmable I/O pins for interrupt and synchronization events
- On-chip interrupt engine and integrated smart features for always-on applications (e.g., activity, action, and gesture recognition) using the sensor ultra-low power domain:
  - motion detection
  - step detector
  - plug ’n’ play step counter
  - orientation, flat, tilt
  - generic interrupts (three parallel instances of a highly configurable flexible interrupt)
- RoHS compliant, halogen and lead free

Table of contents

List of figures

List of tables

1 Specification

This chapter provides the specifications for the BMA530. Minimum values and maximum values are provided for standard distributed quantities as $\mu \pm 3\sigma$, typical values as $\mu \pm \sigma$. Unless stated otherwise, the specifications provide the characteristics for a nominal supply voltage of V_{DD} = V_{DDIO} = 1.8V either at an ambient temperature of T_{A} = 25°C. This definition for minimum (Min), maximum (Max) and typical (Typ) values is also used throughout the other following chapters. Table 1 provides the electrical characteristics for the device.

Table 1: Basic electrical parameter specification

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply voltage core (and I/O) domain	V_{DD} = V_{DDIO}		1.62	1.8	3.63	V
Voltage input low level	V_{IL}	SPI, I²C & I3C			$0.3 \cdot V_{\mathrm{DDIO}}$	V
Voltage input high level	V_{IH}	SPI, I²C & I3C	$0.7 \cdot V_{\mathrm{DDIO}}$			V
Voltage output low level	V_{IL}	SPI			$0.2 \cdot V_{\mathrm{DDIO}}$	V
Voltage output high level	V_{IH}	SPI	$0.8 \cdot V_{\mathrm{DDIO}}$			V
Current consumption	I_{DD}	Suspend mode		4.75		μA
		Low power mode, $f_{\mathrm{A,lp}}$ = 100Hz		18		
		High performance mode, $f_{\mathrm{A,\cdot}}$ = max		125		
Power on time	Δt_{PO}	Time from supply "on" to serial I/F operational (and stable register access)		1.8		ms
Operating temperature	T_{A}		-40		+85	°C
Accuracy of the output data rate (accelerometer and temperature sensor)	Δf_{A} = Δf_{T}	Any mode enabled @ T_{A} = 25°C			3.0	%

The Tables 2, 3 and 4 provide the operating conditions for the accelerometer and the related performance and mechanical characteristics.

Table 2: Operating conditions for the accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Acceleration range	a_{FS}	Selectable via serial digital interface		±2		g
				±4		
				±8		
				±16		
Start-up time - Time to valid data	$t_{\mathrm{A,SU}}$	From suspend mode to first data sample (in high performance mode $f_{\mathrm{A,\cdot}}$ = 1600Hz)		3.15		ms

Table 3: Performance characteristics of the accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Resolution				16		bit
Sensitivity	$S_{A,2g}$	a_{FS} = 2g		16384		$\frac{LSB}{g}$
	$S_{A,4g}$	a_{FS} = 4g		8192		
	$S_{A,8g}$	a_{FS} = 8g		4096		
	$S_{A,16g}$	a_{FS} = 16g		2048		
Sensitivity error	$S_{A,err,8g}$	Soldered, over life time			1	%
Sensitivity error temperature drift	TCS	Full T_A range, best fit straight line		0.005		$\frac{\%}{K}$
Zero-g offset	O_A	Soldered		$\pm$50		mg
	$O_{A,life}$	Soldered, over life time		$\pm$75		mg
Zero-g offset temperature drift	TCO	Full T_A range, best fit straight line		$\pm$0.5		$\frac{mg}{K}$
Noise density	$n_{A,density}$	High performance mode, range 8g		120		$\frac{\mu g}{\sqrt{Hz}}$
Nonlinearity error	$S_{A,NL}$	Best fit straight line, a_{FS} = 2g		0.2		%FS
Output data rate (ODR)	$f_{A,hpm}$, $f_{A,n}$	High performance mode	12.5		6400	Hz
	$f_{A,lpm}$	Low-power mode	1.5625		400	
Bandwidth (BW) in high performance	$B_{A=12.5Hz}$	$0Hz \leq f \leq f_{3dB\text{-}cutoff}$ of the accelerometer, $B_A = \frac{1}{2} f_A$ [Hz]		6.3		Hz
	$B_{A=25Hz}$			12.5		
	$B_{A=50Hz}$			25		
	$B_{A=100Hz}$			50		
	$B_{A=200Hz}$			100		
	$B_{A=400Hz}$			200		
	$B_{A=800Hz}$			400		
	$B_{A=1600Hz}$			800		
	$B_{A=3200Hz}$			850		
	$B_{A=6400Hz}$			1675		

Table 4: Mechanical characteristics of the accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Cross axis sensitivity	$S_{A,X}$	Relative contribution between any two of the three axes		0.3		%
Alignment error	$\Delta\xi_A$	Relative to package outline		0		°

Table 5 provides the temperature sensor related characteristics.

Table 5: Characteristics of the temperature sensor

Parameter	Symbol	Condition	Min	Typ	Max	Units
Resolution				8		bits
Measurement Range	T_S		-41		87	$^\circ C$
Output at 23°C				0		LSB
Sensitivity	S_T			1		$\frac{LSB}{K}$
Temperature offset	O_T	After soldering @ T_A = 25°C		±1.5		K
Temperature sensitivity error		After soldering, T_P			±18	%
Output Data Rate	$f_{T,LPM}$	Accelerometer in low power mode		$f_{A,lpm}$		Hz
	$f_{T,HPM}$	Accelerometer in high performance mode	1.5625		200	

2 Absolute Maximum Ratings

Important: Stress above limits stated in Table 6 may cause damage to the device. Exceeding the specified limits may affect the reliability of the device or can cause malfunction.

Table 6: Absolute maximum ratings

Parameter	Condition	Min	Max	Units
Voltage at Supply Pin	V_{DD} = V_{DDIO} Pin	-0.3	4	V
Voltage at any Logic Pin	Non-Supply Pin-out	-0.3	VDD + 0.3 and < 4	V
Passive Storage Temperature Range	$\leq$ 65	-50	150	°C
OTP Non-Volatile Memory Data Retention	$T \leq 85°\text{C}$	10		a
Mechanical Shock	Duration $\leq$ 200μs		20000	g
ESD	HBM at any pin		2000	V
	CDM		500	V

3 Quick Start Guide

The purpose of this chapter is to help developers to start working with the device by giving basic hands-on application examples. Before starting, the device has to be properly connected to the host and powered up.

Notes on the Serial Interface Support

The communication between host processor and the device happens over one of the interfaces: I²C, I3C or SPI (4-wire and 3-wire). Each register read operation includes the following number of inserted dummy bytes before the payload:

- I²C: 0
- I3C: 1
- SPI: 1

For simplicity, the dummy bytes are not shown in the examples within this chapter. For more information about the interfaces and the protocol selection, please see Chapter 5. After power on reset or soft reset, the device is automatically configured in high performance mode.

First Application Setup Example Procedures

After the proper power-up by applying the stable supply voltage to the corresponding device pins, the device enters automatically into the Power On Reset (POR) sequence. To ensure proper use of the device, certain configuration steps from the host are a mandatory prerequisite. The most typical operations will be explained in the following application examples by flow diagrams:

- Test communication and initialize the device in I²C or SPI 4-wire

Read the `CHIP_ID.chip_id` to ensure the correct communication. Before reading the `CHIP_ID.chip_id`, one initial transaction is required, while the returning value is invalid. This initial transaction determines the serial interface in either I²C or SPI 4-wire for later communication. Notably, the initial transaction through the I²C interface is not acknowledged by the sensor (NACK). Also, it is recommended to read the health status of the sensor to ensure the proper power-on.

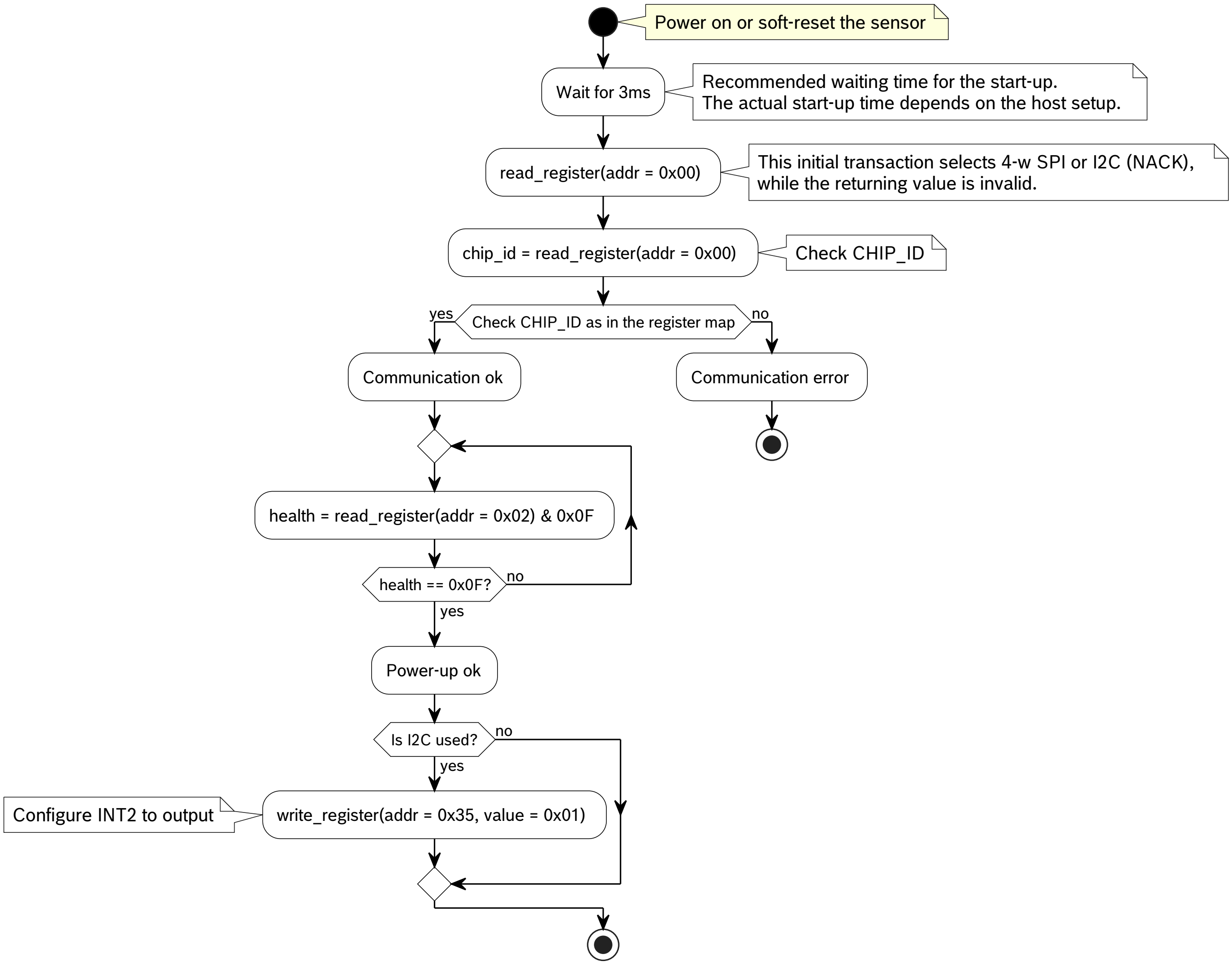

Figure 1: Device communication test

- Enable the SPI 3-wire interface:

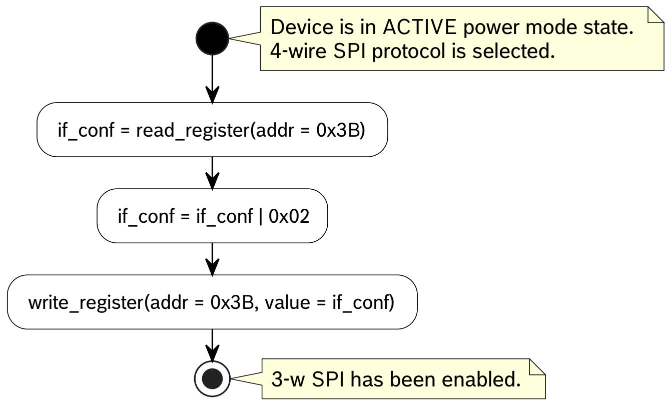

Figure 2: Configure the SPI 3-wire interface

- Enable the I3C interface:

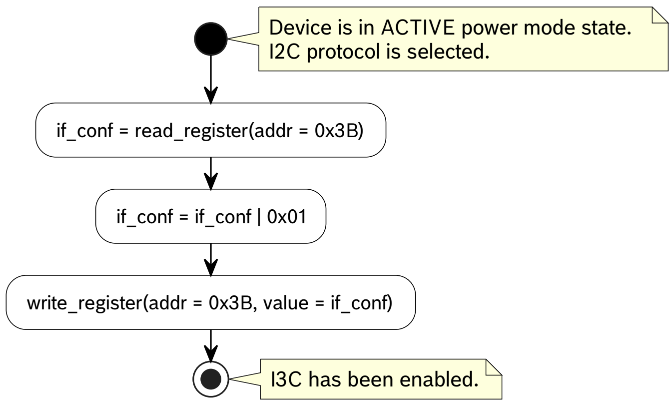

Figure 3: Configure the I3C interface

- Configure the power mode state:

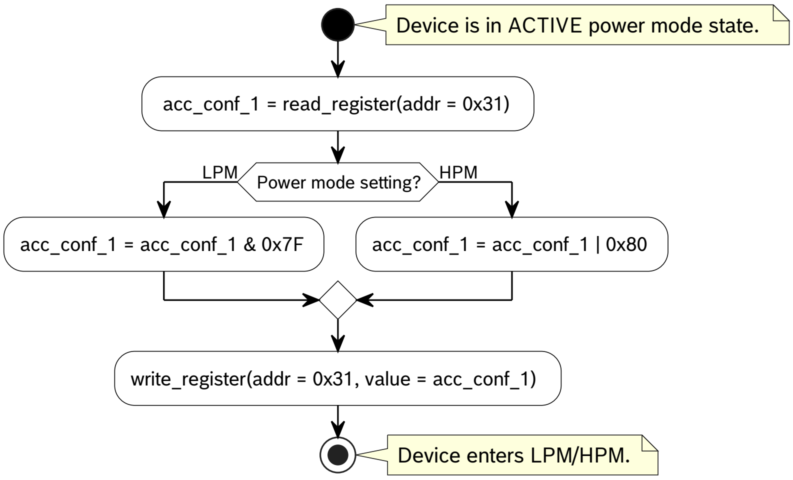

Figure 4: Configure the device power mode

- Configure the device in suspend mode:

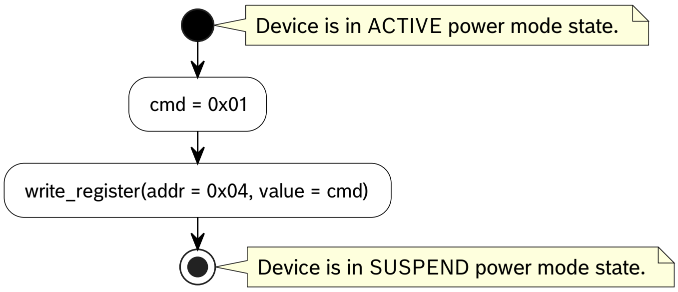

Figure 5: Configure the device suspend mode

- Set the sensor parameters followed by reading the sensor data:

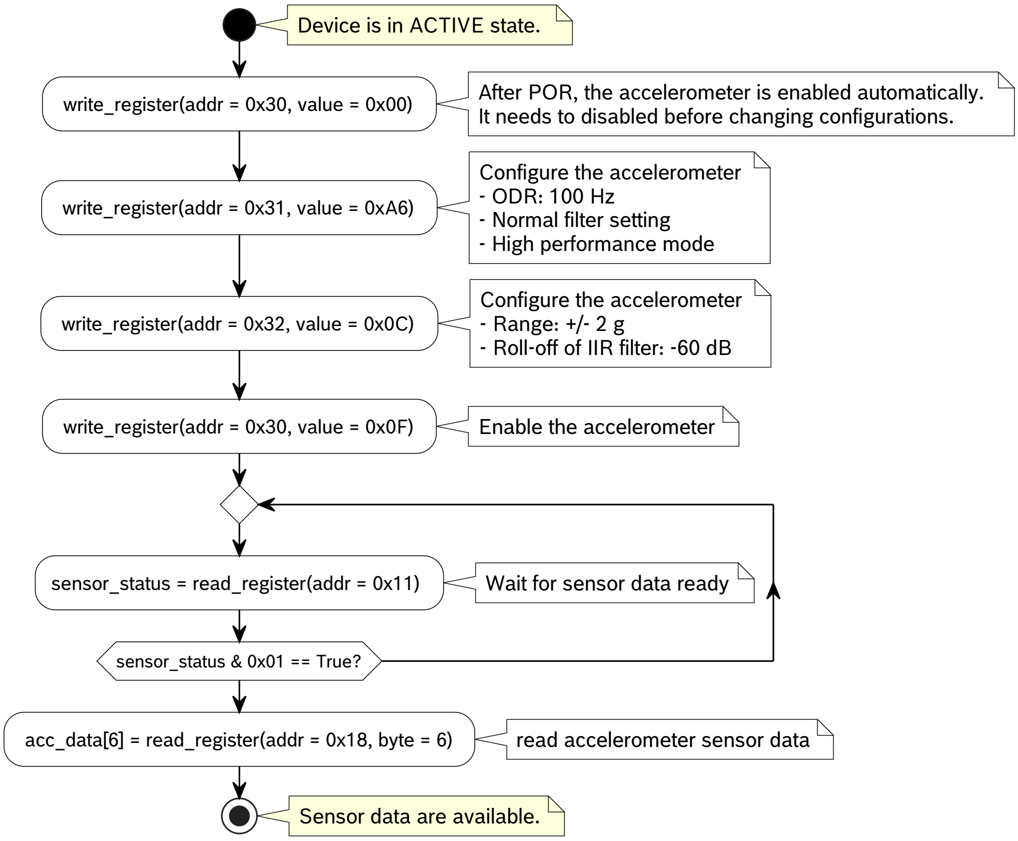

Figure 6: Configure the sensor parameters and read sensor data

- Map the data ready hardware interrupt:

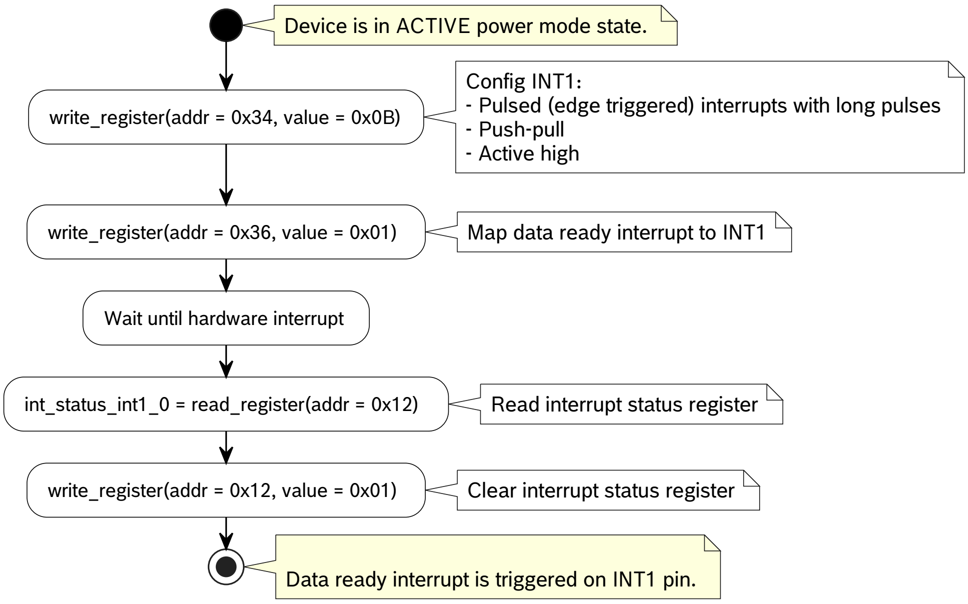

Figure 7: Mapping hardware interrupt

- Change the FIFO size:

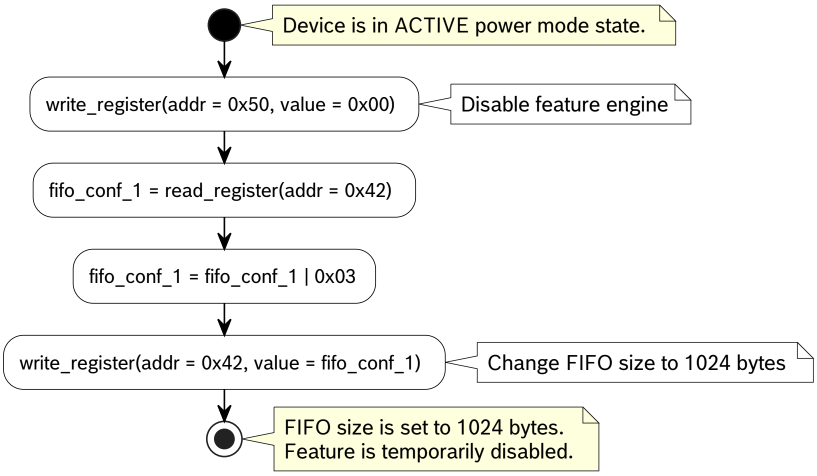

Figure 8: Change FIFO size

- Read the registers in the extended register map:

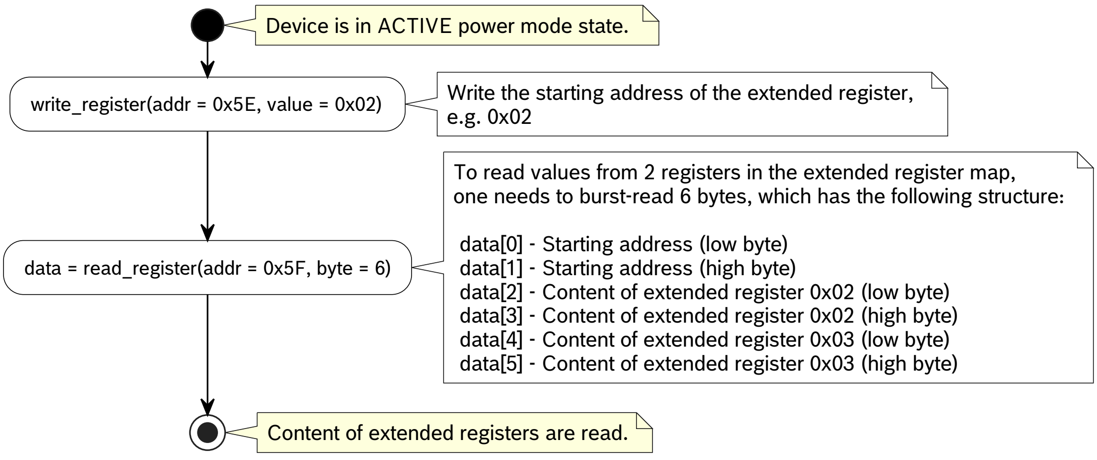

Figure 9: Read registers in the extended register map

- Write the registers in the extended register map:

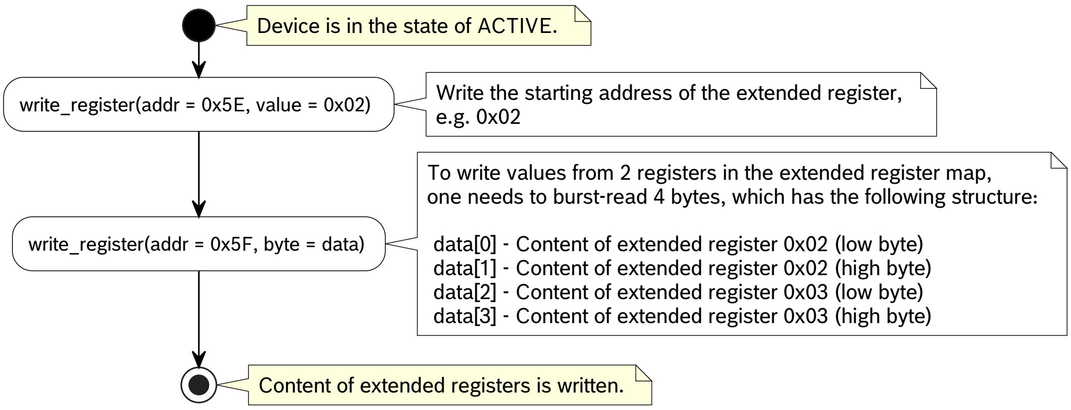

Figure 10: Write registers in the extended register map

- Enable advanced feature, e.g., generic interrupt 1:

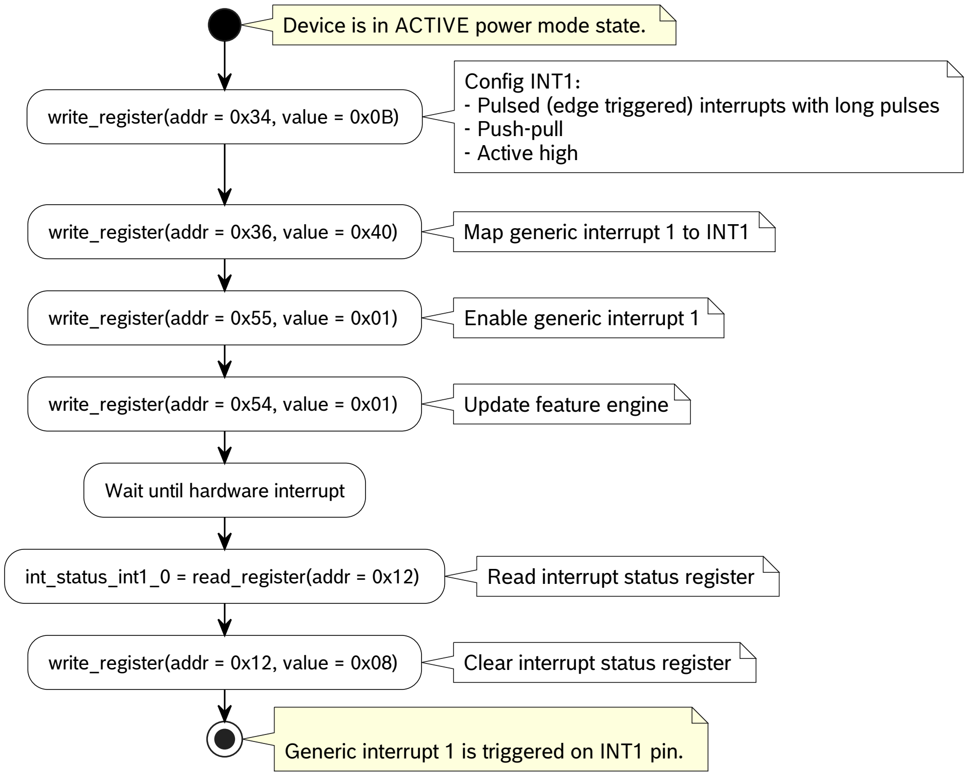

Figure 11: Enable generic interrupt 1

- Enable the Fast Offset Compensation (FOC) feature on Z-axis in combination with INT1. For a complete FOC procedure, it is recommended to perform the feature on all axes as explained in detail in chapter 4.9.9.

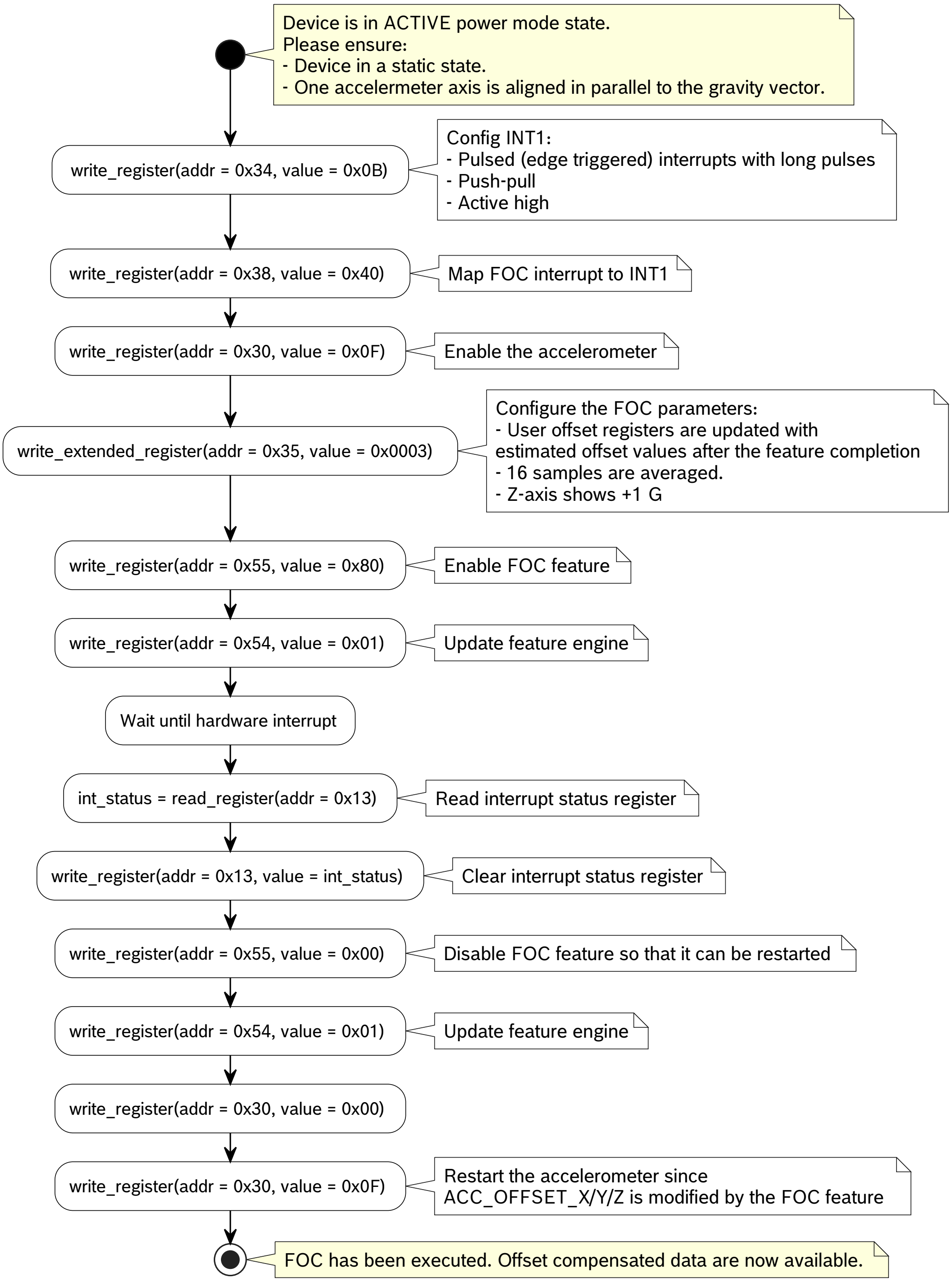

Figure 12: Enable FOC feature in combination with INT1

4 Functional Description and Advanced Features

4.1 Power Mode States

The BMA530 supports ACTIVE and SUSPEND power mode states, which can be switched in `CMD_SUSPEND`. After power on or soft-reset, the default power mode state of BMA530 is HPM in ACTIVE. Conclusively, the switch process is illustrated in Figure 13 .

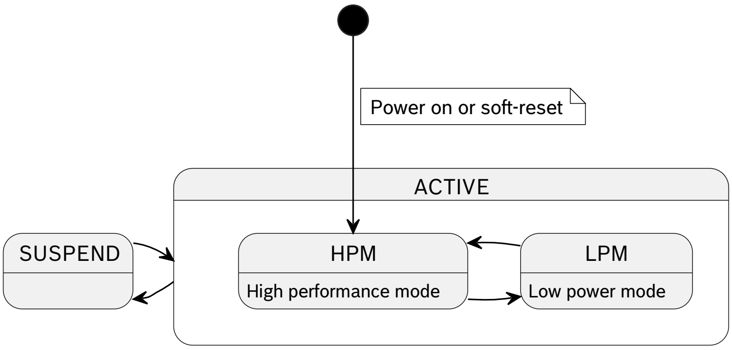

Figure 13: Sensor power mode state diagram

4.1.1 ACTIVE State

In ACTIVE state, the accelerometer is active and there is no restriction in accessing the register map. The device can enter two further performance modes, namely low power mode (LPM) and high performance mode (HPM), and switch between them while in the ACTIVE state. The main difference between the two modes is the data sampling behavior of the acceleration signal.

High Performance Mode (HPM)

In HPM, data is sampled continuously and fed to the filter that is configured by the host.

Low Power Mode (LPM)

In LPM, only the necessary number of data is sampled for the average purpose, so that one can optimize the power consumption. However, since the acceleration signal is undersampled, the duty-cycling mode is prone to aliasing effects.

4.1.2 SUSPEND State

In the SUSPEND state, the accelerometer is inactive and the internal oscillator is also shut down. In this mode, the register content prior to entering this power mode will be retained. Also, the host is limited to access the `CHIP_ID`, `CMD_SUSPEND` and `CMD.cmd` registers. Notably, executing soft-reset is possible in the SUSPEND mode.

Please also note that, once in the SUSPEND state, both INT1 and INT2 pins are configured in high-impedance state. To prevent the error interrupt detection by the host due to signal cross-talk, it is suggested to pull-up or pull-down the interrupt pins from the host side.

4.1.3 Power on Time

The power on time of the BMA530 is typically 1.8ms (see also specification table 1). The power on time describes the time between powering the device ($V_{DD} \geq$1.62V) and the interface beeing ready to respond with stable register access. During the powering phase, the device will not be able to respond to any command sent on the serial interface.

Note that the power on time may vary if the ramp of VDD between 0V and 1.62V, which is controlled by the host, takes a longer time.

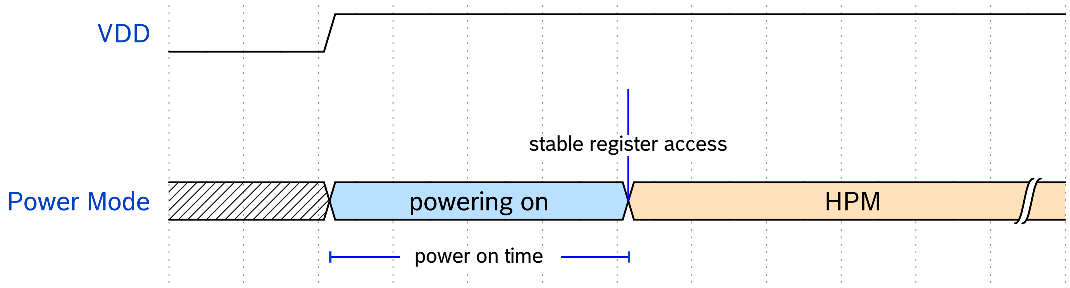

Figure 14: Power on Time

The power on time does not include the processing of the first acceleration value. See chapter 4.2.9 for more information about the time for the first valid data.

4.2 Accelerometer

4.2.1 Accelerometer Data

The three-dimensional acceleration data are provided with 16 bits width in two's complement representation, which are available in registers from `ACC_DATA_0` to `ACC_DATA_5`. The 16 bits acceleration data for each axis contain a high byte and a low byte. To ensure the data integrity, the content in `ACC_DATA_0` to `ACC_DATA_5` must be read in a single burst read.

The output acceleration data are in LSB unit. They can be converted to a g unit using the following formula:

$$ACC\ [g] = 2\text{'}s(ACC_{High}\ [LSB] \ll 8 + ACC_{Low}\ [LSB]) \times \frac{a_{FS}}{2^{15}}, \tag{4.1}$$

where a_{FS} is the acceleration range. The selection of the acceleration range leads to different sensitivity, as concluded in table 7; ACC_{High} and ACC_{Low} are the high byte and low byte of the acceleration data, respectively; "2's()" is the calculation of two's complement.

Table 7: Sensitivity under different acceleration ranges

Acceleration range	Sensitivity (typical value)	
2 g	1 g = 16384 LSB	1 LSB = 61.035 µg
4 g	1 g = 8192 LSB	1 LSB = 122.070 µg
8 g	1 g = 4096 LSB	1 LSB = 244.141 µg
16 g	1 g = 2048 LSB	1 LSB = 488.281 µg

For example, if a_{FS}= 8 g is selected, the acceleration data of 0x7FFF represents 8 g, while 0x8001 represents -8 g. Please note that, the the acceleration data of 0x8000 represents an invalid value, which occurs e.g., when the host sets an invalid configuration, or when the acceleration data is not yet ready after the power-on or during the configuration change.

4.2.2 Accelerometer Data Processing

The acceleration signals and the temperature data are processed according to the configured settings from corresponding registers (`ACC_CONF_0` - `ACC_CONF_2` and `TEMP_CONF`). Additionally, the acceleration signals of the device can be compensated through the registers from `ACC_OFFSET_0` to `ACC_OFFSET_5`. Please note that values in these registers are not persistent and must be written each time after the power-up or reset of the device.

4.2.3 Accelerometer Configuration

The host can configure the accelerometer via registers from `ACC_CONF_0` - `ACC_CONF_2`. In detail:

- `ACC_CONF_0` is used to enable or disable the accelerometer. Please note that, in BMA530, the accelerometer is enabled by default after power on, since the default power mode state is HPM.
- `ACC_CONF_1` is used to select the output data rate (ODR), the bandwidth parameter (BWP) for the filter configuration and the performance mode.
- `ACC_CONF_2` is used to select the dynamic range, the filter roll-off, the measurement preference and clear mechanism of the acceleration data ready interrupt.
- `CONFIG_STATUS.acc_conf_err` indicates invalid or valid accelerometer configurations.

4.2.4 Accelerometer Performance Mode

The device has two performance modes, namely low power mode (LPM) and high performance mode (HPM). The mode switching is controlled by the register field `ACC_CONF_1.power_mode`. In LPM mode the overall power consumption depends strongly on the chosen ODR and the amount of averaged samples. Typical values can be seen in table 8.

Table 8: Current Consumption (typical values) depending on ODR and number of averaged samples in LPM at V_{DD} = 1.8 V, T_A = 25°C.

	current consumption - typical (µA)		
ODR(Hz)	**No Avg**	**Avg 2**	**Avg 4**
1.5625	7.1	7.2*	7.4*
12.5	8.3	8.9*	10.1*
25	9.8*	11.1*	13.7*
50	12.4	14.9*	19.9*
100	18.0	19.7	23.3
200	28.9	38.9*	58.9*
400	51.1	71.2*	111.4*

(*) estimated values.

4.2.5 Accelerometer Effective Bandwidth

The effective bandwidth of the accelerometer depends on the selection of ODR and BWP.

4.2.6 Accelerometer Change Configuration

Before the host changes the accelerometer configuration, it is recommended to disable the accelerometer first. The host can again enable the accelerometer after the configuration is finished. Any change of the accelerometer configuration is applied immediately.

After the configuration change, the host needs to wait for a certain time until the first valid sample is available. This waiting time depends on the changed configuration and the timing of the change. Please note that, in HPM, all samples after the first valid sample are given at the expected ODR. In LPM, it can sometimes occur that the time interval between the first and second samples is not as expected. This is meant for a quick data delivery in the LPM mode. The host can skip the first and second samples if an accurate ODR is necessary.

4.2.7 Accelerometer Self-Test

The BMA530 has a comprehensive self-test function for the MEMS element 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. The activation of 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 and deactivated for all axes via `ACC_SELF_TEST.self_test`. It is also possible to control the direction of the deflection through `ACC_SELF_TEST.self_test_sign`. The excitation occurs in positive (negative) direction if `ACC_SELF_TEST.self_test_sign` = 0b1 (0b0).

In below, the recommended procedure to use the self-test is given:

1. Disable all advanced features and interrupts, if any of them are enabled.
2. Activate the self-test
   a. Disable the accelerometer in `ACC_CONF_0.sensor_ctrl`
   b. Apply the following configurations:
      - `ACC_CONF_1.acc_odr` = 10
      - `ACC_CONF_1.acc_bwp` = 2
      - `ACC_CONF_1.power_mode` = 1
      - `ACC_CONF_2.acc_range` = 2
      - `ACC_CONF_2.acc_iir_ro` = 1
      - `ACC_CONF_2.noise_mode` = 0
      - `ACC_CONF_2.acc_drdy_int_auto_clear` = 0
   c. Enable the accelerometer in `ACC_CONF_0.sensor_ctrl`
   d. Wait for at least 10 ms
   e. Enable self-test and set the negative self-test polarity by setting
      - `ACC_SELF_TEST.self_test_sign` = 0
      - `ACC_SELF_TEST.self_test` = 1
   f. Wait for at least 10 ms
   g. Read and store valid data of each axis from registers `ACC_DATA_0` to `ACC_DATA_5`. Please check in `SENSOR_STATUS.acc_data_rdy` before reading, if the valid data is ready.
   h. Enable self-test and set the positive self-test polarity by setting
      - `ACC_SELF_TEST.self_test_sign` = 1
      - `ACC_SELF_TEST.self_test` = 1
   i. Wait for at least 10 ms
   j. Read and store valid data of each axis from registers `ACC_DATA_0` to `ACC_DATA_5`. Please check in `SENSOR_STATUS.acc_data_rdy` before reading, if the valid data is ready.
   k. Check self-test results:
      i. Convert values from steps 2g and 2j for each axis. Please note that those values are signed values, so the host has to apply the two's complement calculation to the raw data.
      ii. Calculate the difference between the values from step 2g and 2j
      iii. Compare the difference against the minimum threshold values in Table 9. To pass the self-test, the measured difference has to exceed the minimum threshold value.

Table 9: Minimum threshold value of self-test

x-axis (LSB)	y-axis (LSB)	z-axis (LSB)
17500	17500	8000

   l. Disable the self-test by setting
      - `ACC_SELF_TEST.self_test` = 0
3. It is recommended to perform a soft-reset of the device after the self-test. Please note that, after the soft-reset, all user configuration settings are overwritten with their default state. If the soft-reset cannot be performed, the following sequence is required to reset the signal path:
   a. Disable the accelerometer in `ACC_CONF_0.sensor_ctrl`
   b. Wait for at least 50 ms
   c. Enable the accelerometer in `ACC_CONF_0.sensor_ctrl`

4. Now the host can apply user configuration to the accelerometer and again enable the advanced features and interrupts.

4.2.8 Accelerometer Data Ready Interrupt

This interrupt fires whenever a new data sample set from accelerometer. This allows a low latency data readout.

4.2.9 Accelerometer Startup Time (Time to valid data)

The accelerometer startup time of the BMA530 is typically 3.15ms (see also specification table 2). The accelerometer startup time describes the time between leaving the suspend mode and the availability of the first valid acceleration data, if the selected power mode is HPM and the selected ODR is 1600Hz. The BMA530 can indicate this time point with an interrupt, if the data ready trigger is selected. See chapter 4.5 for more details. During the powering phase the device will not be able to respond to any command sent on the serial interface.

Please note, that the accelerometer startup time may vary, if a different power mode is selected or a different ODR.

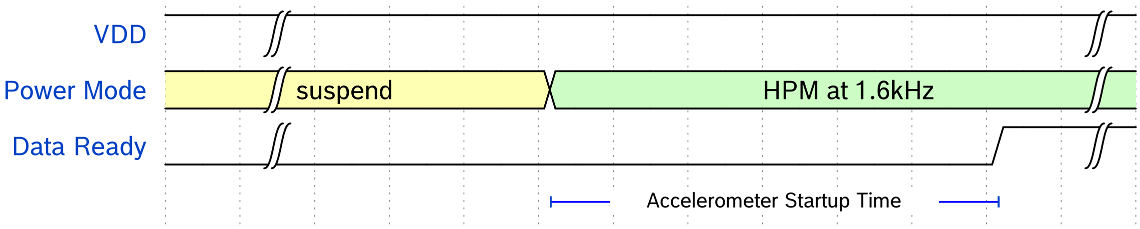

Figure 15: Accelerometer Startup Time

The accelerometer startup time is also related to the power on time, but not overlapping. For more information on the power on time, see chapter 4.1.3. The following picture shows both times in an example illustration:

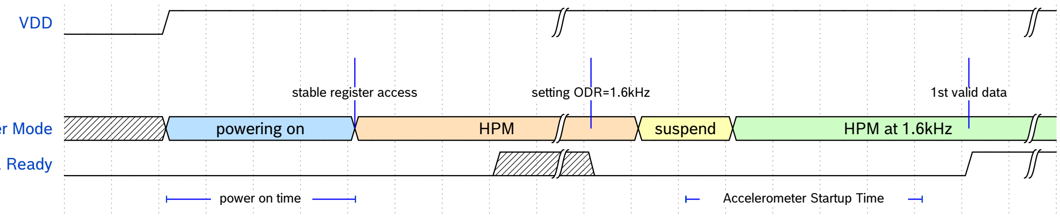

Figure 16: Power on Time and Accelerometer Startup Time

4.2.10 Accelerometer Offset Compensation

The BMA530 offers manual compensation. The offset compensation is effective for data in `ACC_DATA_0` - `ACC_DATA_5` and FIFO, and signals for the advanced features. If necessary the result of this computation is saturated to prevent any overflow errors (the smallest or biggest possible value is set, depending on the sign).

The offset compensation uses of the registers `ACC_OFFSET_0` to `ACC_OFFSET_5`, providing a compensation value for each accelerometer axis x, y, z, respectively. The contents of the compensation register `ACC_OFFSET_0` to `ACC_OFFSET_5` may be set manually via the digital interface. It is recommended to restart the accelerometer after writing new values to the compensation register or write to the compensation register, while the accelerometer is disabled (see register `ACC_CONF_0`).

To disable the offset compensation, a value of 0x0 has to be written to all the compensation register `ACC_OFFSET_0` to `ACC_OFFSET_5`.

The offset compensation registers have a width of 9 bit using two's-complement binary notation. The offset resolution is 0.98 mg (1024 LSB/g) with an offset range of +/-0.25 g. Please note that the resolution of the offset register is independent of the range setting (see register `ACC_CONF_2.acc_range`). The compensation offset values are not persistent and must be written each time after power-up or reset of the device.

The BMA530 offers also the “Fast Offset Compensation” (FOC) feature, which is described in chapter 4.9.9.

4.3 Sensor time

The device supports the concept of sensor time. Its core element is a free running counter with a width of 24 bits. It runs at the frequency of 3.2 kHz, while the time resolution is 312.5 µs. The host can access the current state of the counter by reading registers from `SENSOR_TIME_0` to `SENSOR_TIME_2`. The sensor time counter is synchronized with the data capturing event in the register from `ACC_DATA_0` to `ACC_DATA_5` and FIFO.

Please note that a burst read on register from `SENSOR_TIME_0` to `SENSOR_TIME_2` delivers always consistent values. Once the device enters the SUSPEND power mode state, the sensor time counter stops.

4.4 Temperature Sensor

The BMA530 provides a temperature sensor, sensing the internal temperature of the device. The temperature sensor is always on, when the accelerometer sensor is active.

The temperature sensor has 8 bits, the data can be read from register field `TEMP_DATA.temp_data`. The data register output is of the unit K. A data value of 0x0 means 23°C. The sensor can be configured via the register `TEMP_CONF`: The output data rate for the temperature sensor can be set in the field `TEMP_CONF.temp_rate`

When there is no valid temperature information available, the temperature indicates an invalid value (0x80) and the register field `SENSOR_STATUS.temperature_rdy` shows a 0x0.

4.5 Interrupt Pin Configuration

The BMA530 has two external pins to provide the status of feature events. For certain digital interface settings, these pins are not available for this interrupt behavior but used by the digital interface. In I²C and I3C mode the two external pins are available for the feature events. In SPI 3-Wire mode, one pin is still available and in SPI 4-Wire mode no external pin is available to provide feature events. See table 43 and chapter 5 for more details.

4.5.1 Electrical Interrupt Pin Behavior

The electrical behavior of interrupt pins INT1 and INT2 can be configured in the register `INT1_CONF` and `INT2_CONF`, respectively.

4.5.1.1 Output Mode

In the register fields `INT1_CONF.mode` and `INT2_CONF.mode`, the output on the pins can be enabled/disabled, and the output mode can be configured between latch, short pulses and long pulses mode. Please note that, if the output pin is disabled, the interrupt status will not be updated.

- In the latch mode, the interrupt output is active when the status bit of any mapped interrupt source is set. It will remain active until cleared.
- In the short and long pulses mode, the interrupt output is active when the status bit of any mapped interrupt source is set. Then, after a certain pulse duration, the the interrupt output becomes automatically inactive, while the corresponding status bit of any mapped interrupt source remains uncleared. In other words, the host needs to clear the interrupt status bit if necessary. Table 10 provides the typical pulse duration in the short and long pulses mode.

Table 10: Pulse duration in the short and long pulses mode

	Short pulses mode	Long pulses mode
Typ. pulse duration	625 ns	10 us

Especially, in addition to the common output mode setting, the auto clear mechanism of the data ready interrupt can be configured in `ACC_CONF_2.acc_drdy_int_auto_clear`. When this option is enabled, the status flag of acc_drdy_int is cleared automatically after the half of the ODR duration. This saves the need for the host to clear each data ready interrupt status. Please note that, it is recommended to enable the auto clear mechanism in latch mode only, since the auto clear mechanism can be considered as an extension of the pulses mode.

4.5.1.2 Output Characteristics

The characteristic of the output driver of the interrupt pins may be configured with fields `INT1_CONF.od` and `INT2_CONF.od`. By setting these bits to 0b1, the output driver shows open-drain characteristic. By setting the configuration bits to 0b0, the output driver shows push-pull characteristic. The electrical behavior of the interrupt pins, whenever an interrupt is triggered, can be configured as either “active-high” or “active-low” via `INT1_CONF.lvl` respectively `INT2_CONF.lvl`.

Please note the high impedance state of interrupt pins when the BMA530 is in the SUSPEND state, as already mentioned in chapter 4.1.2.

4.5.2 Interrupt Pin Mapping

In order for the host to react to the features output, they can be mapped to the external pin INT1 or pin INT2, by setting the corresponding bits from the registers `INT_MAP_0`, `INT_MAP_1` and `INT_MAP_3`. To disconnect the features outputs to the external pins, the same corresponding bits must be reset, from those registers. Once a feature triggers the output pin, the host can read out the corresponding bit from the register `INT_STATUS_INT1_0`, `INT_STATUS_INT1_1`,`INT_STATUS_INT2_0` or `INT_STATUS_INT2_1`.

Besides to the two external pins, the interrupts can also be mapped to the I3C in band interrupts (IBI), if the BMA530 is in I3C mode. In this case, the status can be handled in the register `INT_STATUS_I3C_0` and `INT_STATUS_I3C_1`.

An interrupt source can be mapped also to several destinations (INT1, INT2, I3C IBI) in parallel. In this case, the status of the interrupt has to be read and cleared for each single destination separately in the registers `INT_STATUS_INT1_0`, `INT_STATUS_INT1_1`,`INT_STATUS_INT2_0`,`INT_STATUS_INT2_1`,`INT_STATUS_I3C_0` and `INT_STATUS_I3C_1`.

4.5.3 Clear Interrupt Status

In BMA530, the interrupt status is cleared upon writing 1’b1 to the corresponding interrupt status bit.

4.5.4 Interrupt Behavior Example

For a better understanding of the interrupt pin behavior of BMA530, the following examples under various configurations are provided. For simplicity, the “INT pin” represents both INT1 and INT2 pins, and the electrical behavior of the interrupt pins is configured as “active-high”.

Latch Mode Figure 17 shows the timing diagram when the latch mode is configured. FEATURE_A and FEATURE_B can represent any interrupt source, and are mapped to the same INT pin target. Detailed explanations of the timing diagram are provided in below:

- When the interrupt event “1” of FEATURE_A comes, the corresponding FEATURE_A_int_status is set immediately high. FEATURE_A_int_status is cleared after the host executes the clear operation at “2”.
- When the interrupt event “3” of FEATURE_B comes, the corresponding FEATURE_B_int_status is set immediately high. While the interrupt event “3” of FEATURE_B remains active, a clear operation such as “4” will fail to clear the FEATURE_B_int_status. This describes the case of e.g., FIFO full interrupt. FEATURE_B_int_status can be cleared at “5”, when FEATURE_B is no more active.
- The INT pin is set to high as long as one of the interrupt status is active.

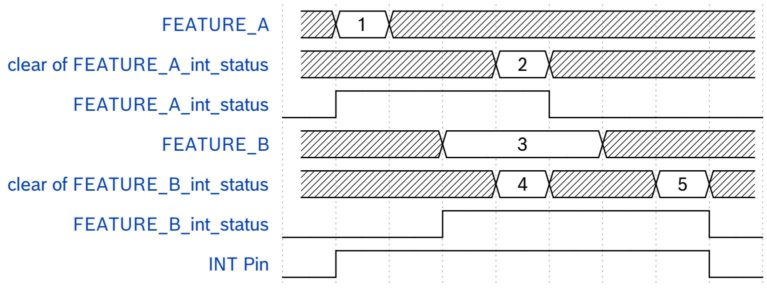

Figure 17: Interrupt output in latch mode

Pulses Mode Figure 18 shows the timing diagram when the pulses mode is configured. FEATURE_A and FEATURE_B can represent any interrupt source, and are mapped to the same INT pin target. Detailed explanations of the timing diagram are provided in below:

- When the interrupt event "1" of FEATURE_A comes, the corresponding FEATURE_A_int_status is set immediately high, so is the INT pin. Since the pulses mode is configured, the INT pin turns to low after the duration defined in table 10. Then, when another interrupt event "2" of FEATURE_A comes, a pulse signal is again generated on the INT pin without the FEATURE_A_int_status to be cleared. At the end, FEATURE_A_int_status is cleared after the host executes the clear operation at "3".
- When the interrupt event "4" of FEATURE_B comes, the corresponding FEATURE_B_int_status is set immediately high, so is the INT pin. Since the pulses mode is configured, the INT pin turns to low after the duration defined in table 10. While the interrupt event "4" of FEATURE_B remains active, a clear operation such as "5" will fail to clear the FEATURE_B_int_status. This describes the case of e.g., FIFO full interrupt. FEATURE_B_int_status can be cleared at "6", when the FEATURE_B is no more active.

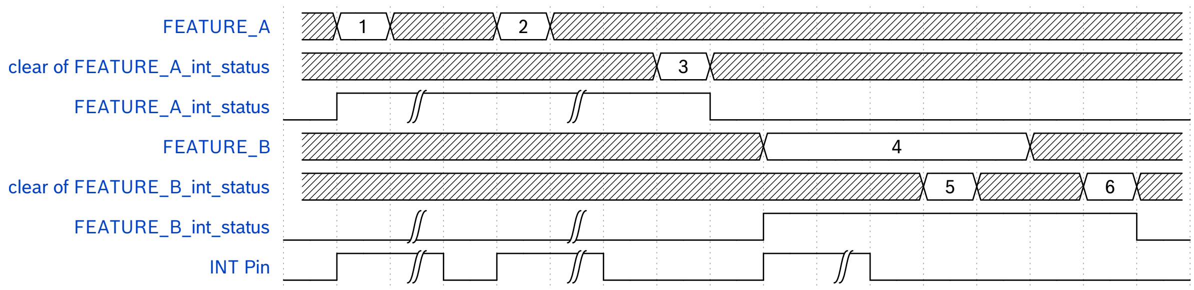

Figure 18: Interrupt output in pulses mode

Auto Clear Mechanism of the Data Ready Interrupt Figure 19 shows the timing diagram when the auto clear mechanism of the data ready interrupt is enabled. Both the acc_drdy_int and INT pin is cleared automatically after the half of the ODR duration.

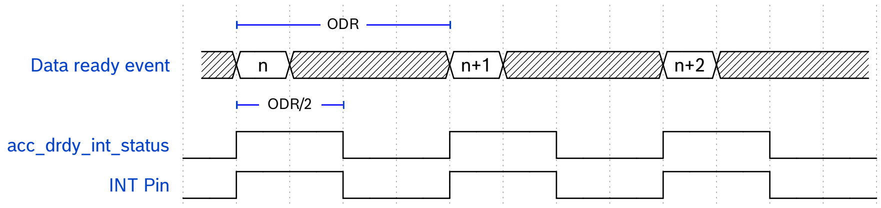

Figure 19: Interrupt output when auto clear mechanism of the data ready interrupt is enabled

4.6 FIFO

The BMA530 provides a first-in first-out (FIFO) data buffer for the accelerometer data as well as optionally the sensor time . The size of this FIFO is configurable with a maximum of 1024 bytes.

4.6.1 FIFO Configuration

The FIFO can be configured by the user registers `FIFO_CONF_0` and `FIFO_CONF_1`.

4.6.1.1 Enabling FIFO

The register `FIFO_CONF_0` is used to enable or disable the complete FIFO functionality or to sample only individual axis.

4.6.1.2 FIFO Compression

The field `FIFO_CONF_0.fifo_compression` can be configured to store only 8 bit of acceleration data. If the compression is enabled, only the high byte of the acceleration data is stored, e.g., `ACC_DATA_1.acc_x_15_8`.

4.6.1.3 Sensor Time in FIFO

The field `FIFO_CONF_1.fifo_sensor_time` can be configured to disable the sensor time, to send dedicated sensor time frame, or to append sensor time to each frame. For more information of the dedicated sensor time frame, please refer to section 4.6.2.4.

4.6.1.4 FIFO Stop-on-full Mode

The FIFO stop-on-full mode can be configured in the field `FIFO_CONF_1.fifo_stop_on_full`, and the full level is defined as the FIFO size minus two times the payload size.

4.6.1.5 FIFO Size

The size of this FIFO is configurable with a maximum of 1024 bytes. The default is 512 bytes, which allows the feature engine to work in parallel to the FIFO. The FIFO size can be configured with the help of the register field `FIFO_CONF_1.fifo_size`. Since FIFO and the feature engine share a common memory, the size configuration is locked when the feature engine is enabled. Then this register is controlled by the feature engine and the value might change depending on the chosen features with their possible configuration.

The register field `FIFO_CONF_1.fifo_size` can only be changed, once the feature engine is disabled. With the feature engine disabled, a FIFO size up to 1024 bytes is possible, while with the feature engine enabled, a FIFO size of 512 bytes or less is possible. In some few configurations the FIFO size might be limited to 256 bytes (for example when 200 Hz filter data for a generic interrupt is used, there is a limitation on the FIFO size. Please refer to 4.9.1.2 for more information).

4.6.2 FIFO Frames

4.6.2.1 FIFO Header

The FIFO header has the following structure in below:

Table 11: FIFO header structure

Legend			Read-only		Read/Write		Write-only		Reserved	
Index	**Name**	**Value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x0	HEADER	0x80	const_1	frame_type		compr_en	acc_z_en	acc_y_en	acc_x_en	acc_t_en

Please note that the FIFO header is not stored in the memory while generated during the FIFO read-out process. This needs to be considered cautiously, when the host calculates the correct number of data frames in the FIFO.

Field frame_type Field frame_type encodes the type of payload data:

- 2'b00: empty frame
- 2'b10: acceleration data frame
- 2'b01: dedicated sensor time frame

Field acc_x/y/z_en Field acc_x/y/z_en displays the selection of accelerometer axis in the FIFO. If the acc_x/y/z_en field of the header equals 1'b1, the corresponding axis data is contained in the FIFO frame. Otherwise, the axis data is not part of the FIFO frame. The order of the payload bytes is x, y and z.

Field compr_en Field compr_en displays the enabling state of FIFO data compression. If data compression is enabled (`FIFO_CONF_0.fifo_compression` = 1'b1), each enabled axis contributes one byte (the MSB) to the payload. Otherwise, each enabled axis contributes two bytes to the payload.

Field acc_t_en Field acc_t_en displays the enabling state of sensor time in each FIFO frame.The host can choose the way to display sensor time by configuring `FIFO_CONF_1.fifo_sensor_time`, which can change the header format. This is explained in 4.6.2.3 and 4.6.2.4.

4.6.2.2 Empty Frame

An empty frame has no payload and only consists of a single byte as header.

Table 12: FIFO empty frame

Legend			Read-only		Read/Write		Write-only		Reserved	
Index	**Name**	**Value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x0	HEADER	0x80	const_1	frame_type		compr_en	acc_z_en	acc_y_en	acc_x_en	acc_t_en

4.6.2.3 Acceleration Data Frame

The number of payload bytes depends on the configuration of `FIFO_CONF_0` and `FIFO_CONF_1`. If the FIFO is configured in such a way that the payload would be 0 (i.e. no axis enabled, no sensor time), no data will be stored in the FIFO memory and only empty frames will be read.

The minimum data frame size is 2 byte, when only a single axis and data compression is enabled.

If `FIFO_CONF_1.fifo_sensor_time` is configured as 2'b10, the sensor time is appended to each data frame. Please note that the sensor time data in each data frame occupies the FIFO space.

For an acceleration data frame the maximum frame size looks as follows:

Table 13: FIFO sensor data frame

Legend			Read-only		Read/Write		Write-only		Reserved	
Index	**Name**	**Value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x0	HEADER	0xCF	const_1	frame_type		compr_en	acc_z_en	acc_y_en	acc_x_en	acc_t_en
0x1	PAYLOAD_0	0x00	acc_x_7_0							
0x2	PAYLOAD_1	0x00	acc_x_15_8							
0x3	PAYLOAD_2	0x00	acc_y_7_0							
0x4	PAYLOAD_3	0x00	acc_y_15_8							
0x5	PAYLOAD_4	0x00	acc_z_7_0							
0x6	PAYLOAD_5	0x00	acc_z_15_8							
0x7	PAYLOAD_6	0x00	sensor_time_7_0							
0x8	PAYLOAD_7	0x00	sensor_time_15_8							
0x9	PAYLOAD_8	0x00	sensor_time_23_16							

4.6.2.4 Dedicated Sensor Time Frame

If `FIFO_CONF_1.fifo_sensor_time` is configured as 2'b01, the FIFO sends a dedicated sensor time frame when the FIFO runs empty during a read burst. It has the following format:

Table 14: FIFO sensor time frame

Legend			Read-only		Read/Write		Write-only		Reserved	
Index	**Name**	**Value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x0	HEADER	0xA1	const_1	frame_type		compr_en	acc_z_en	acc_y_en	acc_x_en	acc_t_en
0x1	PAYLOAD_0	0x00	sensor_time_7_0							
0x2	PAYLOAD_1	0x00	sensor_time_15_8							
0x3	PAYLOAD_2	0x00	sensor_time_23_16							

The dedicated sensor time frame will be transmitted after the last sensor data frame has been read. This means that the dedicated sensor time frame is always preceded by sensor data frames and followed by empty frames. No sensor time frame will be transmitted if a read burst starts when the FIFO is already empty. The content of the dedicated sensor time frame is sampled when the header byte of the sensor time is read.

Please note that, the dedicated sensor time frame is not stored in the FIFO memory.

4.6.2.5 FIFO Frame Read Out

It is recommended to burst read the FIFO frame to ensure the proper read out of dedicated sensor time frame, if it is enabled. Also, once FIFO frames are read out, it will be discarded in the FIFO memory, while the unread ones remains there. Therefore, it is strongly recommended to read out all available frames once the FIFO content is ready. Otherwise, the remaining unread data will corrupt the next run of the FIFO.

Read Out in Stop-on-full Mode Please pay special attention to the FIFO read out process when the FIFO stop-on-full is enabled. In this case, the FIFO stops buffering data once the full condition is met. Then, the host can perform the burst-read operation to read out data. However, during the burst read process, once old samples are read out, they are discarded from the FIFO. Therefore, the FIFO full condition becomes no more valid, thus allowing new samples to be buffered in the FIFO.

This design helps to ensure the data continuity. However, this also leads to the situation that the host needs to read out more frames than what the FIFO can store in a full condition. At higher ODR, there are more additional frames. If the dedicated sensor time frame is enabled, this situation needs to be especially considered because the dedicated sensor time frame will be only transmitted after the last sensor data frame, as mentioned in section 4.6.2.4.

If the data continuity is not important, the host can disable the accelerometer before burst-read the FIFO. In this way, no additional frames during the read out process are required.

Read Out Sensor Time in FIFO The BMA530 provides various methods to read out acceleration data in association with sensor time data.

- The host can enable the sensor time data in each frame as described in section 4.6.2.3. In this way, each frame is labeled with a sensor time.
- The host can enable the dedicated sensor time frame as described in section 4.6.2.4. In this way, the BMA530 provides the sensor time at the moment when the host finishes reading the FIFO.
- Without enabling the dedicated sensor time frame, the host can perform a burst read starting from `SENSOR_TIME_0`. In this way, the BMA530 provides the sensor time at the moment when the host starts to read the FIFO.

4.6.3 FIFO Interrupts

The BMA530 offers two kinds of interrupt events, which can be mapped to the interrupt pins like any other interrupt sources. General information about interrupt pin configuration is described in chapter 4.5. Once the FIFO is enabled, the interrupt can be mapped directly to any destination. They do not have to be enabled separately. Please note, that for both interrupts, only the acceleration and sensor time data counts. The headers are not stored in the FIFO memory and not considered when determining the FIFO fill level.

4.6.3.1 FIFO Watermark Interrupt

The FIFO watermark interrupt will be asserted as long as the fill level is equal to or larger than the watermark level. Before the FIFO has reached the watermark level, any attempts to manually clear the FIFO watermark interrupt status will fail.

The watermark level can be set in the registers `FIFO_WM_1.fifo_watermark_level_10_8` and `FIFO_WM_0.fifo_watermark_level_` The unit is bytes. If the level is set to a higher value than the FIFO size, the watermark interrupt will never be triggered.

4.6.3.2 FIFO Full Interrupt

The full level is defined as the FIFO size minus two times the payload size, as already mentioned in section 4.6.1.4. The FIFO full interrupt will be asserted as long as the fill level is equal to or larger than the full level. While the FIFO remains at the full level, any attempts to manually clear the FIFO full interrupt status will fail.

4.6.4 FIFO Reset/Flush

The FIFO may be explicitly flushed by writing `FIFO_CTRL.fifo_rst` = 1'b1. The FIFO is flushed automatically, if the FIFO configuration registers `FIFO_CONF_0` and `FIFO_CONF_1` are written or the device wakes up from suspend power state. Also, the FIFO is flushed automatically, if the accelerometer configuration is changed and the accelerometer signal path is reset. Conclusively, any writing operation to the register `ACC_CONF_0`, `ACC_CONF_1` or `ACC_CONF_2` will trigger the flushing.

4.7 Soft-Reset

In order to reset the BMA530 without removing the supply voltage, the offers a Soft-Reset. A Soft-Reset can be initiated at any time by writing the command softreset (0xB6) to register `CMD.cmd`. The softreset performs a fundamental reset to the device which is largely equivalent to a power cycle. Following a delay, all user configuration settings are overwritten with their default state. This command is functional in all operation modes.

4.8 Sensor Health Status

The register field `HEALTH_STATUS.sensor_health_status` indicates the internal health state of the device. A value of 0x0F indicates a good internal health state. The reserved bits in the same register `HEALTH_STATUS` should be ignored. Any other values in the field `HEALTH_STATUS.sensor_health_status` indicate an internal error. If the value remains on error state after reset and the external supply is stable and in correct range, the device should be checked.

4.9 Advanced Features

4.9.1 General Configuration

4.9.1.1 Enable and Disable Advanced Feature

To enable/disable the advanced features of the device, please follow the step in below:

1. Set/clear the corresponding feature enable bit. E.g., for the generic interrupt 1, write `FEAT_ENG_GPR_0.gen_int1_en`= 1/0.
2. Set `FEAT_ENG_GPR_CTRL.update_gprs`= 1.

Please note that the second step needs to be performed every time when the corresponding feature enable bit is changed, so that the change becomes effective.

4.9.1.2 Data Path of Advanced Features

Depending on the advanced features, the datapath that feeds acceleration data to the feature engine is constructed in different ways.

Data Path of Generic Interrupt In the generic interrupt, the host can choose one of the following data sources via `FEAT_ENG_GPR_1.gen_int1_data_src`, `FEAT_ENG_GPR_1.gen_int2_data_src`, `FEAT_ENG_GPR_1.gen_int3_data_src`:

- 50 Hz filter data: the acceleration data that is directly down-sampled from the temperature compensated raw data. The ODR of this data source is locked at 50 Hz. This setting is the default and recommended value in case the generic interrupts are configured to the motion detection feature behavior (like Any-Motion Detector, see chapters 4.9.3 or 4.9.4).
- 200 Hz filter data: the acceleration data that is directly down-sampled from the temperature compensated raw data. The ODR of this data source is locked at 200 Hz.
- User filter data: the acceleration data that is available in registers `ACC_DATA_0` - `ACC_DATA_5`, which is configurable for the host via `ACC_CONF_0` - `ACC_CONF_2`.

Please note that, when different data path sources for the advanced features are used, there are limitations on the accelerometer ODR in LPM. Please refer to section 4.9.1.3.

Since the data source for the advanced detector features(see chapter 4.9.1.2) is 50 Hz filter data, the default and recommended configuration is 50 Hz filter data for the generic interrupt, too. Please note that, when 200 Hz filter data (or user filter data) is used, there is a limitation on the usage of FIFO, because there is more memory needed for the data processing. In this case, only a FIFO size of 256 bytes is available. The user needs to activate another advanced feature with 50 Hz data path in parallel so that the FIFO size is automatically set to the 256-byte-size and all data paths are working properly. This additional enabled feature does not have to be mapped to any physical interrupt pin.

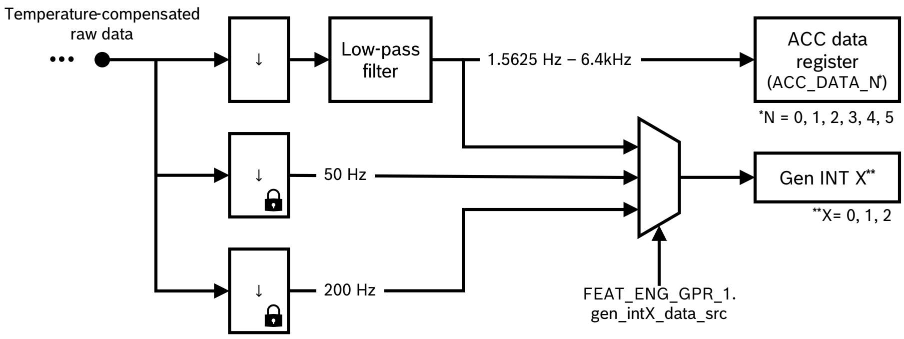

Figure 20: Data Path of generic interrupt

Data Path of Android Features The Android features, including:

- step counters/step detector
- significant motion detection
- orientation detection
- tilt detection,

have the following data source:

- 50 Hz filter data: the acceleration data that is directly down-sampled from the temperature compensated raw data. The ODR of this data source is locked at 50 Hz.

The user filter data, which is available in `ACC_DATA_0` - `ACC_DATA_5`, is configurable via `ACC_CONF_0` - `ACC_CONF_2`.

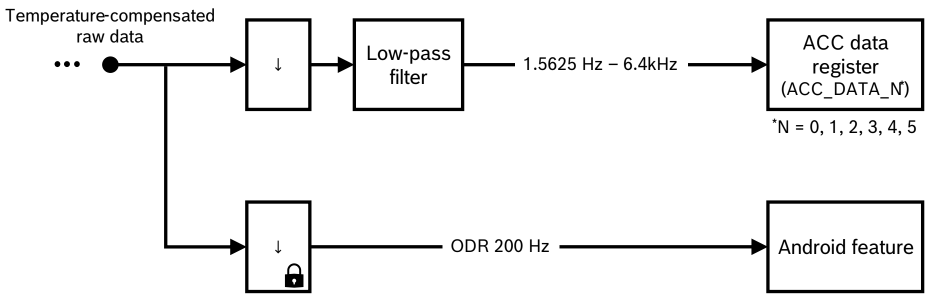

Figure 21: Data path of Android feature

Axis remapping of datapath The host can remap the accelerometer axes in `GENERAL_SETTINGS_0.feat_axis_ex` and change their polarity in `GENERAL_SETTINGS_0.feat_x_inv`, `GENERAL_SETTINGS_0.feat_y_inv` and `GENERAL_SETTINGS_0.feat_z_inv`, as illustrated in figure 22 . Please note that the axis remapping feature does not influence the values in `ACC_DATA_0` to `ACC_DATA_5`. It affects only the datapath that is fed to advanced features.

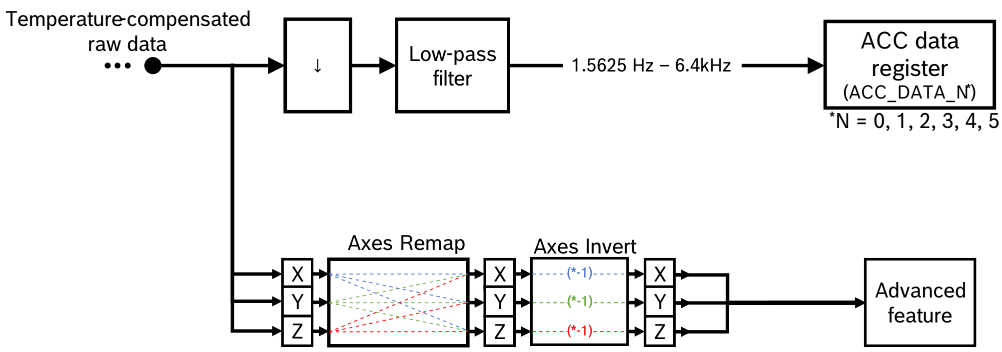

Figure 22: Axis remapping of datapath

4.9.1.3 Supported ODR in Advanced Features

The advanced features support various accelerometer ODRs.

For the following advanced features:

- generic interrupts
- any-motion/motion detector
- no-motion/stationary detector

in HPM, all available accelerometer ODRs are supported. In LPM, depending on the selection of the data path, the generic interrupts support only limited ODRs. Table 15 lists the supported ODRs.

Table 15: Overview of supported ODRs in generic interrupts, any-motion/motion detector and no-motion/stationary detector

Power Mode State	Data path[1]	Supported accelerometer ODR
HPM	`FEAT_ENG_GPR_1.gen_int1_data_src = 1`	All available ODRs
	`FEAT_ENG_GPR_1.gen_int1_data_src = 2`	
	`FEAT_ENG_GPR_1.gen_int1_data_src = 3`	
LPM	`FEAT_ENG_GPR_1.gen_int1_data_src = 1`	$\geq$ 50 Hz
	`FEAT_ENG_GPR_1.gen_int1_data_src = 2`	$\geq$ 200 Hz
	`FEAT_ENG_GPR_1.gen_int1_data_src = 3`	6.25 Hz $\geq$ ODR $\geq$ 400

[1] The generic interrupt 1 is taken as an example. The rule is also applied to the generic interrupt 2 and 3.

For the following advanced features:

- step counters/step detector
- significant motion detection
- orientation detection
- tilt detection

in HPM, all available accelerometer ODRs are supported. In LPM, these advanced features only support ODR $\geq$ 50 Hz. Table 16 lists the supported ODRs.

Table 16: Supported ODRs in step counters/step detector, significant motion detection, orientation detection and tilt detection.

Power Mode State	ODR
HPM	All available ODRs
LPM	$\geq$ 50 Hz

4.9.1.4 Invalid Feature Configuration

If the host sets an invalid ODR that is not mentioned in Table 15 and Table 16 , the device will automatically set `FEAT_ENG_GPR_5.feat_conf_err` = 1 as an error signal. The detailed information about the features with invalid configuration can be looked up in `FEAT_CONF_ERR`.

4.9.1.5 Android Compliance

Some advanced features in BMA530 can be configured to be Android compliant, which includes:

- any-motion detector,
- no-motion detector,
- significant motion detection (by default android compliant) and
- tilt detection (by default android compliant)

The Android compliance for the motion detectors can be enabled or disabled by setting `GENERAL_SETTINGS_0.android_comp` = 1 or 0. Once enabled, the user setting of certain feature parameters will be ignored (but not modified), while the internal setting in compliance with the Android requirement will be used.

4.9.1.6 Enable Advanced Features in Parallel

All advanced features can be enabled and work in parallel.

It is recommended to deactivate all other features, when using the Fast Offset Compensation (FOC), described in chapter 4.9.9.

4.9.2 Generic Interrupt

The generic interrupt feature of the device is designed to detect device's movements (activity) or device's static state (in-activity).

4.9.2.1 General Functional Behavior of Generic Interrupt

The functional behavior of the generic interrupt is presented in Figure 23. Generally, the change in the acceleration signal with respect to a defined reference value is monitored by the feature. Then, this change in the acceleration signal is compared against a configurable threshold value. Additionally, to avoid unwanted rapid interrupts, a hysteresis value can also be configured. Finally, to trigger the generic interrupt, the acceleration change must be either greater or lower than the configured threshold for a configured duration. The interrupt can be again cleared, when the condition remains false for a configured wait time.

The device provides the generic interrupts 1, 2 and 3, which can be enabled and disabled respectively via:

- `FEAT_ENG_GPR_0.gen_int1_en`
- `FEAT_ENG_GPR_0.gen_int2_en`

- FEAT_ENG_GPR_0.gen_int3_en

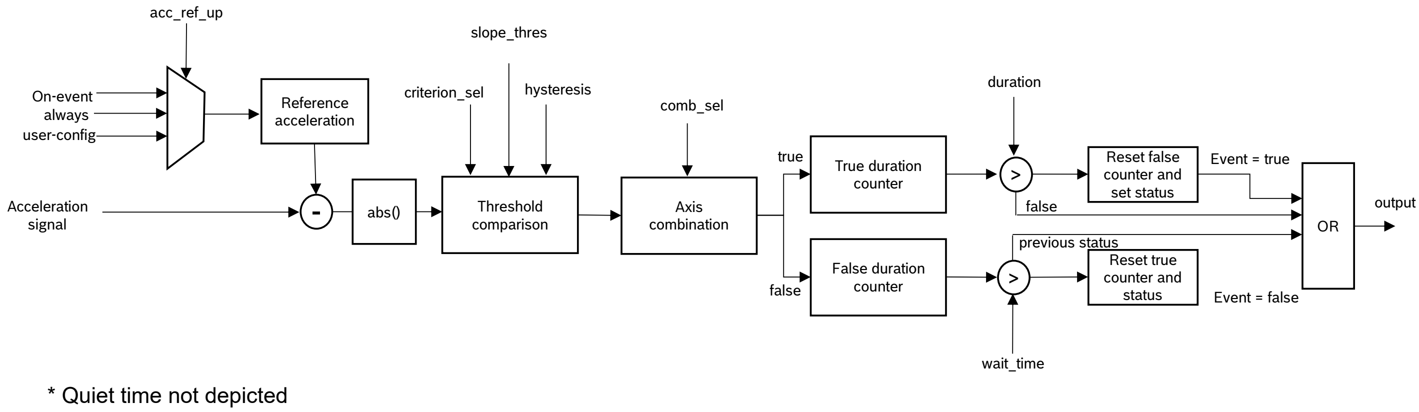

Figure 23: Functional behavior of generic interrupt

All generic interrupts have the same implementation. Therefore, in the following description, only the generic interrupt 1 is used as an example.

1. GENERIC_INTERRUPT1_1.comb_sel selects the combination logic of the chosen axis:
   - GENERIC_INTERRUPT1_1.comb_sel = 0: combination of axes is set to logic OR.
   - GENERIC_INTERRUPT1_1.comb_sel = 1: combination of axes is set to logic AND.
2. GENERIC_INTERRUPT1_2.criterion_sel selects, whether the interrupt is triggered based on activity or in-activity:
   - GENERIC_INTERRUPT1_2.criterion_sel = 0: interrupt triggers based on in-activity
   - GENERIC_INTERRUPT1_2.criterion_sel = 1: interrupt triggers based on activity
3. GENERIC_INTERRUPT1_2.acc_ref_up selects the approach, with which the acceleration reference signal is updated:
   - GENERIC_INTERRUPT1_2.acc_ref_up = 0: on-event. The reference acceleration is updated whenever there is an event (interrupt) being triggered by the feature.
   - GENERIC_INTERRUPT1_2.acc_ref_up = 1: always. An update of the reference acceleration is done with each new available acceleration sample.
   - GENERIC_INTERRUPT1_2.acc_ref_up = 2: manual. The reference acceleration is set manually by the host in:
     - GENERIC_INTERRUPT1_5.ref_acc_x
     - GENERIC_INTERRUPT1_6.ref_acc_y
     - GENERIC_INTERRUPT1_7.ref_acc_z
4. GENERIC_INTERRUPT1_1.axis_sel selects the acceleration axes used for the generic interrupt feature:
   - GENERIC_INTERRUPT1_1.axis_sel = 1: x-axis is used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 2: y-axis is used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 3: x-axis and y-axis are used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 4: z-axis is used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 5: x-axis and z-axis are used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 6: y-axis and z-axis are used for the evaluation.
   - GENERIC_INTERRUPT1_1.axis_sel = 7: x-axis, y-axis and z-axis are used for the evaluation.
5. GENERIC_INTERRUPT1_1.slope_thres configures threshold value that is compared to the change in the acceleration signal.
6. GENERIC_INTERRUPT1_2.hysteresis configures the hysteresis value to avoid unwanted rapid interrupts.
7. GENERIC_INTERRUPT1_3.duration configures the duration parameter, in which the condition needs to remain true, so that the interrupt can be triggered.
8. GENERIC_INTERRUPT1_3.wait_time configures the wait time parameter, in which the condition needs to remain false, so that the interrupt can be cleared.

9. `GENERIC_INTERRUPT1_4.quiet_time` configures the quiet time behavior of the generic interrupt, which is not depicted in figure 23. It defines the minimum quiet time between two consecutive interrupt detection. This means that, after an interrupt was triggered, no new interrupt will be triggered before the configured quiet time is expired.

4.9.3 Any-Motion Detector

The any-motion detector triggers an interrupt, when the slope between adjacent acceleration samples exceeds a threshold for a duration. It is realized using the generic interrupt. For that purpose, the following parameters have fixed values:

1. The axis combination selection is configured as logic OR (e.g. `GENERIC_INTERRUPT1_1.comb_sel` = 0)
2. The criterion is configured as activity (e.g. `GENERIC_INTERRUPT1_2.criterion_sel` = 1)
3. The acceleration reference update is configured as always (e.g. `GENERIC_INTERRUPT1_2.acc_ref_up` = 1).

By configuring the remaining parameters in the generic interrupt, the behavior of the any-motion detector can be influenced. In detail:

4. The axis selection (e.g. `GENERIC_INTERRUPT1_1.axis_sel`) defines, which axis or combination of axis is used to for the evaluation. (Change to single axis only if a the use case really demands it, in general the any motion detector works with all axis.)
5. The slope threshold (e.g. `GENERIC_INTERRUPT1_1.slope_thres`) influences the sensitivity of the detection.
6. The hysteresis (e.g. `GENERIC_INTERRUPT1_2.hysteresis`) influences the sensitivity of the detection.
7. The duration (e.g. `GENERIC_INTERRUPT1_3.duration`) defines how long a motion beyond the threshold needs to be present before triggering an interrupt.
8. The wait time (e.g. `GENERIC_INTERRUPT1_3.wait_time`) defines, after an any-motion interrupt, how long a motion below the threshold needs to be present, before an interrupt is again cleared.
9. The quiet time (e.g. `GENERIC_INTERRUPT1_4.quiet_time`) defines the time of the no-interrupt state after an interrupt is triggered.

By default, the generic interrupt 1 is already configured as an any-motion detector. If necessary, other generic interrupts can also be configured as a an any-motion detector by the host.

Android compliance If the Android compliance bit `GENERAL_SETTINGS_0.android_comp` is enabled, the generic interrupt will be configured as "motion detect" as defined by Android, if `GENERIC_INTERRUPT2_2.criterion_sel` = 1. Also, the following parameters will be ignored, while their internal values will be used. The generic interrupt 1 is taken as an example:

- `GENERIC_INTERRUPT1_1.comb_sel` = 0
- `GENERIC_INTERRUPT1_2.acc_ref_up` = 1
- `GENERIC_INTERRUPT1_3.duration` is set to 5 s.

4.9.4 No-Motion Detector

The no-motion detector triggers an interrupt, when the slope between adjacent acceleration samples remains below a threshold for a duration. It is realized using the generic interrupt. For that purpose, the following parameters have fixed values:

1. The axis combination selection is configured as logic AND (e.g. `GENERIC_INTERRUPT2_1.comb_sel` = 1).
2. The criterion is configured as in-activity (e.g. `GENERIC_INTERRUPT2_2.criterion_sel` = 0).
3. The acceleration reference update is configured as always (e.g. `GENERIC_INTERRUPT2_2.acc_ref_up` = 1).

By configuring the remaining parameters in the generic interrupt, the behavior of the any-motion detector can be influenced. In detail:

4. The axis selection (e.g. `GENERIC_INTERRUPT2_1.axis_sel`) defines, which axis or combination of axis is used to for the evaluation. (Change to single axis only if a the use case really demands it, in general the no motion detector works with all axis.)
5. The slope threshold (e.g. `GENERIC_INTERRUPT2_1.slope_thres`) influences the sensitivity of the detection.

6. The hysteresis (e.g. GENERIC_INTERRUPT2_2.hysteresis) influences the sensitivity of the detection.
7. The duration (e.g. GENERIC_INTERRUPT2_3.duration) defines how long a motion blow the threshold needs to be present before triggering an interrupt.
8. The wait time (e.g. GENERIC_INTERRUPT2_3.wait_time) defines, after an no-motion interrupt, how long a motion beyond the threshold needs to be present, before an interrupt is again cleared.
9. The quiet time (e.g. GENERIC_INTERRUPT2_4.quiet_time) defines the time of the no-interrupt state after an interrupt is triggered.

By default, the generic interrupt 2 is already configured as a no-motion detector. If necessary, other generic interrupts can also be configured as a no-motion detector by the host.

Android compliance If the Android compliance bit GENERAL_SETTINGS_0.android_comp is enabled, the generic interrupt will be configured as “stationary detect” as defined by Android, if GENERIC_INTERRUPT2_2.criterion_sel = 0. Also, the following parameters will be ignored, while their internal values will be used. The generic interrupt 2 is taken as an example:

- GENERIC_INTERRUPT2_1.comb_sel = 1
- GENERIC_INTERRUPT2_2.acc_ref_up = 1
- GENERIC_INTERRUPT2_3.duration is set to 5 s.

4.9.5 Step Counter and Step Detector

The step counter provides the function required for counting of steps as defined by Android[2], [3]. The step detector implements the function required for step detection in Android[4]. The algorithm for counting of steps is designed for smartphone, wearable and hearable use-cases and optimized for a high accuracy, while the algorithm for the detection of steps is optimized for low latency. Each event can be enabled independently. This feature is supported in the high performance power operation mode for all sample rates and in the low power operation mode for the sample rates from 50 Hz to the maximum supported sample rate, see Table 15.

Enable and disable

1. FEAT_ENG_GPR_0.step_en – common enable bit for both step counter and step detector.
2. STEP_COUNTER.sc_en– enable or disable the step counter.
3. STEP_COUNTER.sd_en – enable or disable the step detector.

Output The output of the step counter is stored in a 24-bit value consisting of the following registers:

1. FEAT_ENG_GPR_1.step_cnt_out_0 (low byte)
2. FEAT_ENG_GPR_2.step_cnt_out_1 (middle byte)
3. FEAT_ENG_GPR_3.step_cnt_out_2 (high byte)

Also, the events of the step counter watermark and step detector are reported in interrupt status registers, if the interrupt source is mapped. Both events are reported individually. For example, if mapped to INT1, the interrupt status is available in:

- INT_STATUS_INT1_0.step_cnt_int_status
- INT_STATUS_INT1_0.step_det_int_status

Step Counter The step counter accumulates the detected steps, and provides a 24-bit counting value. By setting STEP_COUNTER.reset_counter to 0b1, the value of accumulated steps is reset. Afterwards, STEP_COUNTER.reset_counter

[2] Android is a trademark of Google LLC.
[3] https://source.android.com/devices/sensors/sensor-types.html#step_counter
[4] https://source.android.com/devices/sensors/sensor-types.html#step_detector

is automatically reset and counting is restarted. The accumulated step count value can be reset when either the step counter or step detector is enabled.

The watermark option can be useful if the host needs to receive an interrupt when a certain number of steps occurred, which can be defined in `STEP_COUNTER.watermark_level`. Please note that the value in this field holds implicitly a 20x scaling factor, which means that the watermark function counts the steps with a resolution of 20 steps. If this field is set to 0, the step counter watermark is disabled. Once the watermark level is reached, the corresponding interrupt bit, e.g. `INT_STATUS_INT1_0.step_cnt_int_status`, is asserted if mapped. As the steps are buffered internally, the output may be triggered between 200 to 210 steps.

Step Detector The step detector triggers an interrupt for every detected step. Every time when a new step is detected, the configured corresponding interrupt output is triggered and the status bit is set. The step detector feature is optimized for a low latency to ensure a fast host reaction. Hence, when a step is detected, it is immediately signaled. Due to this behavior, there may exist situations when the sum of the detected steps is different than the step counter value.

4.9.6 Significant Motion Detection

The significant motion detection provides the interrupt raised by the detection of a significant motion as defined by Android[5]. This feature is supported in the high performance power operation mode for all sample rates and in the low power operation mode for the sample rates from 50 Hz to the maximum supported sample rate, see Table 15.

Enable and disable It can be enabled and disabled via `FEAT_ENG_GPR_0.sig_mo_en`.

Configuration In Android, a significant motion is a motion due to a change in the user location. Examples of such significant motions are walking or biking, sitting in a moving car, coach or train, etc. Examples of situations that does typically not trigger significant motion include phone in pocket and person is stationary or phone is at rest on a table which is in normal office use. Upon detection of a user movement classified as significant according to the aforementioned examples, an interrupt is triggered indicating the probable change of an user location. The classification of movement as significant motion or not is based on the analysis of acceleration signal over the time duration configured by `SIG_MOTION.block_size`. Time segments are assumed to be non-overlapping. If the significant motion condition is evaluated as true for greater than 50% of the configured duration, an interrupt is reported.

Example An example for the behavior of the significant motion detection for a walking scenario is depicted in figure 24, where `SIG_MOTION.block_size` is set to 5 seconds (0x00FA).

[5] https://source.android.com/devices/sensors/sensor-types.html#significant_motion

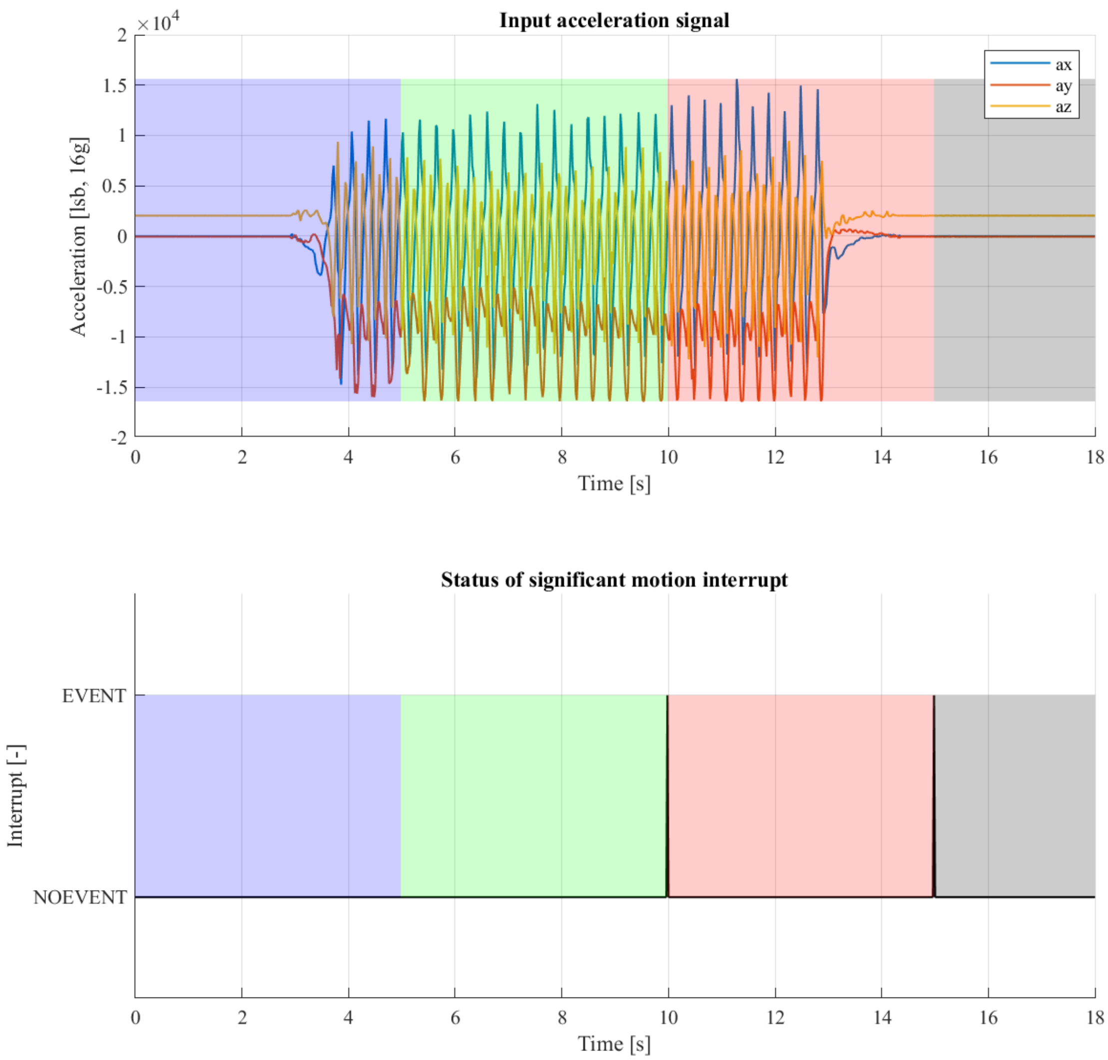

Figure 24: Significant motion interrupt detection behavior for walking use-case

This example contains 3 scenarios of motion within the configured block size time interval:

1. Large initial part of segment as STILL with small part of segment as WALKING. (Blue segment 1 in figure 24)
2. Full segment as WALKING. (Green segment 2 in figure 24)
3. Initial large part of segment as WALKING with remaining being STILL. (Red segment 3 in figure 24)

The segment 1 (blue) encompasses the user movement for less than 50% of `SIG_MOTION.block_size`, hence no interrupt is reported. In contrast to that, segments 2 and 3 include the user movement for greater than 50% for which the interrupts are reported at end of the segment.

4.9.7 Orientation Detection

The orientation detection feature informs an orientation change of the device with respect to the earth gravitational force. The orientation types comprise two orthogonal aspects, namely

- face up
- face down

and

- portrait upright
- landscape left

- portrait downside
- landscape right

The orientation detection can be enabled and disabled via `FEAT_ENG_GPR_0.orient_en`. The sensor orientation is defined by the angles φ and θ. φ is the rotation around the stationary z-axis and θ is the rotation around the stationary y-axis before the φ rotation. The measured acceleration vector components has the following relationship with φ and θ:

$$a_x = 1g \cdot \sin\theta \cdot \cos\varphi \quad (4.2)$$

$$a_y = -1g \cdot \sin\theta \cdot \sin\varphi \quad (4.3)$$

$$a_z = 1g \cdot \cos\theta \quad (4.4)$$

$$\frac{a_y}{a_x} = -\tan\varphi \quad (4.5)$$

Their definitions of φ and θ is visualized in figure figure 25, while their mapping to the orientation is presented in figure 26.

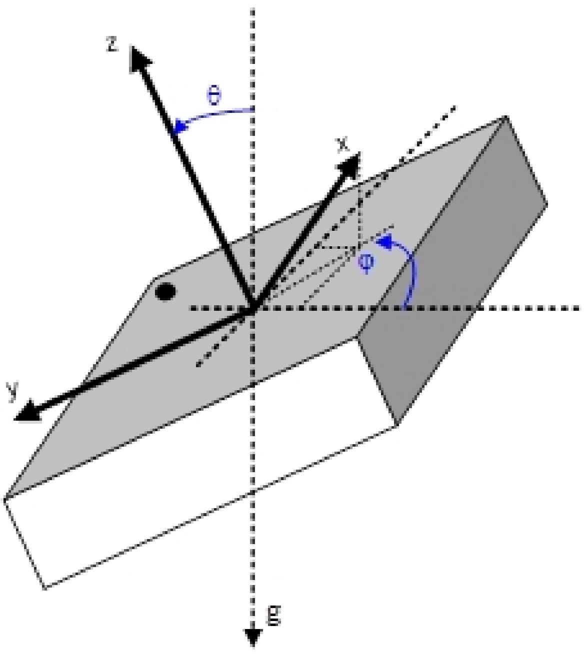

Figure 25: Definition of the default coordinate system with respect to pin 1 marker

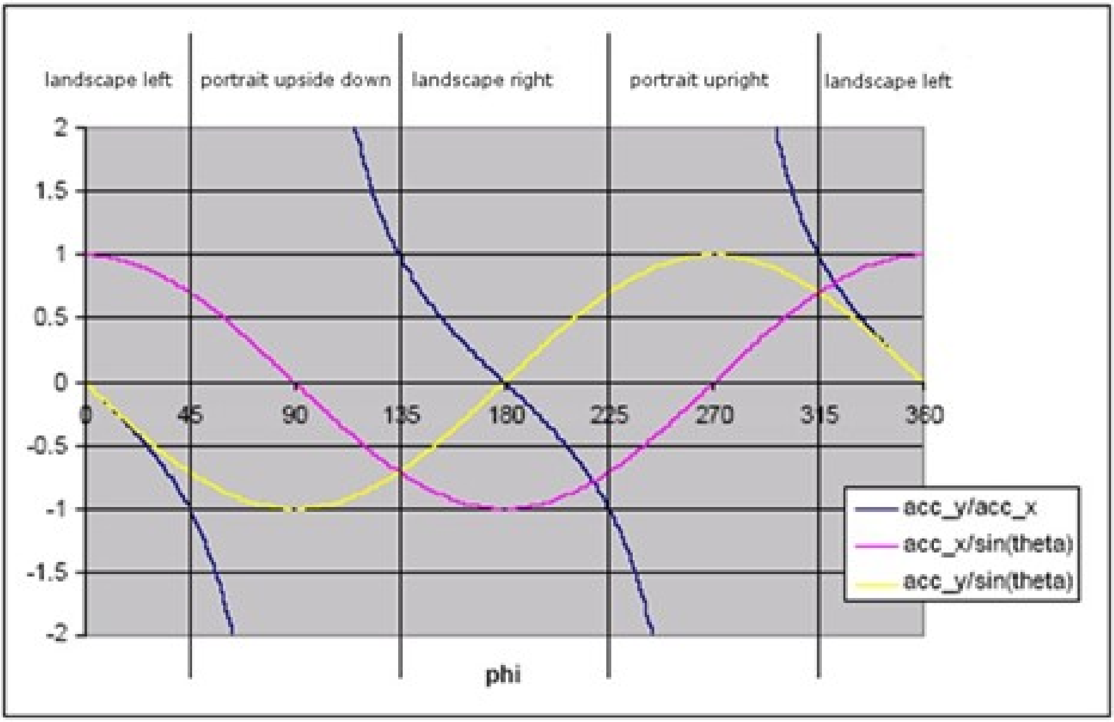

Figure 26: Angle-to-orientation mapping

Orientation Calculation Mode There are three orientation calculation modes: symmetrical, high-asymmetrical and low-asymmetrical. The mode can be configured through the `ORIENTATION_1.mode` as denoted in Table 17.

Orientation mode	`ORIENTATION_1.mode`
Symmetrical	0b00
High asymmetrical	0b01
Low asymmetrical	0b10

Table 17: Orientation mode selection

Output Register Depending on the calculation mode, values in the output register have different meanings as stated in the Tables 18, 19 and 20.

Table 18: Symmetrical mode

`FEAT_ENG_GPR_5.orientation_portrait_landscape`	Name	Angle	Condition
0b01	landscape left	$315° < \varphi < 45°$	$\left\lvert \frac{a_y}{a_x} \right\rvert < 1$ and $a_x \geq 0$
0b11	landscape right	$135° < \varphi < 225°$	$\left\lvert \frac{a_y}{a_x} \right\rvert < 1$ and $a_x < 0$
0b10	portrait upside down	$45° < \varphi < 135°$	$\left\lvert \frac{a_y}{a_x} \right\rvert \geq 1$ and $a_y < 0$
0b00	portrait upright	$225° < \varphi < 315°$	$\left\lvert \frac{a_y}{a_x} \right\rvert \geq 1$ and $a_y \geq 0$

Table 19: High Asymmetrical Mode

FEAT_ENG_GPR_5. orientation_portrait_landscape	**Name**	**Angle**	**Condition**
0b01	landscape left	$297° < \varphi < 63°$	$\left\lvert\frac{a_y}{a_x}\right\rvert < 2$ and $a_x \geq 0$
0b11	landscape right	$117° < \varphi < 243°$	$\left\lvert\frac{a_y}{a_x}\right\rvert < 2$ and $a_x < 0$
0b10	portrait upside down	$63° < \varphi < 117°$	$\left\lvert\frac{a_y}{a_x}\right\rvert \geq 2$ and $a_y < 0$
0b00	portrait upright	$243° < \varphi < 297°$	$\left\lvert\frac{a_y}{a_x}\right\rvert \geq 2$ and $a_y \geq 0$

Table 20: Low Asymmetrical Mode

FEAT_ENG_GPR_5. orientation_portrait_landscape	**Name**	**Angle**	**Condition**
0b01	landscape left	$333° < \varphi < 27°$	$\left\lvert\frac{a_y}{a_x}\right\rvert < 0.5$ and $a_x \geq 0$
0b11	landscape right	$153° < \varphi < 207°$	$\left\lvert\frac{a_y}{a_x}\right\rvert < 0.5$ and $a_x < 0$
0b10	portrait upside down	$27° < \varphi < 153°$	$\left\lvert\frac{a_y}{a_x}\right\rvert \geq 0.5$ and $a_y < 0$
0b00	portrait upright	$207° < \varphi < 333°$	$\left\lvert\frac{a_y}{a_x}\right\rvert \geq 0.5$ and $a_y \geq 0$

For upside or downside orientation, the output value in `FEAT_ENG_GPR_5.orientation_face_up_down` is interpreted according to Table 21.

Table 21: Upside/Downside Definition

FEAT_ENG_GPR_5. orientation_face_up_down	**Name**	**Angle**	**Condition**
0b0	upside	$270° < \varphi < 90°$	$a_z \geq 0$
0b1	downside	$90° < \varphi < 270°$	$a_z < 0$

Hysteresis Both kinds of orientation detection, namely the portrait/landscape and upside/downside detection, use a hysteresis to avoid frequent interrupts due to the non-stable states of an assumed orientation, e.g. by hand tremor or noisy environments. The hysteresis for orientation detection except portrait upside and portrait downside is configurable and applies to all conditions as detailed in Tables 22, 23, and 24. The corresponding hysteresis regions are depicted in the Figures 27, 28, and 29.

Table 22: Hysteresis in the symmetrical mode

FEAT_ENG_GPR_5. orientation_portrait_landscape	**Name**	**Angle**	**Condition**
0b01	landscape left	$315° + \varphi_h < \varphi < 45° - \varphi_h$	$\lvert a_y \rvert < \lvert a_x \rvert - h$ and $a_x \geq 0$
0b11	landscape right	$135° + \varphi_h < \varphi < 225° - \varphi_h$	$\lvert a_y \rvert < \lvert a_x \rvert - h$ and $a_x < 0$
0b10	portrait upside down	$45° + \varphi_h < \varphi < 135° - \varphi_h$	$\lvert a_y \rvert > \lvert a_x \rvert + h$ and $a_y < 0$
0b00	portrait upright	$225° + \varphi_h < \varphi < 315° - \varphi_h$	$\lvert a_y \rvert > \lvert a_x \rvert + h$ and $a_y \geq 0$

Table 23: Hysteresis in the high asymmetrical mode

`FEAT_ENG_GPR_5.orientation_portrait_landscape`	**Name**	**Angle**	**Condition**
0b01	landscape left	$297° + \varphi_h < \varphi < 63° - \varphi_h$	$\|a_y\| < 2 \cdot (\|a_x\| - h)$ and $a_x \geq 0$
0b11	landscape right	$117° + \varphi_h < \varphi < 243° - \varphi_h$	$\|a_y\| < 2 \cdot (\|a_x\| - h)$ and $a_x < 0$
0b10	portrait upside down	$63° + \varphi_h < \varphi < 117° - \varphi_h$	$\|a_y\| > 2 \cdot \|a_x\| + h$ and $a_y < 0$
0b00	portrait upright	$243° + \varphi_h < \varphi < 297° - \varphi_h$	$\|a_y\| > 2 \cdot \|a_x\| + h$ and $a_y \geq 0$

Table 24: Hysteresis in the low asymmetrical mode

`FEAT_ENG_GPR_5.orientation_portrait_landscape`	**Name**	**Angle**	**Condition**
0b01	landscape left	$333° + \varphi_h < \varphi < 27° - \varphi_h$	$\|a_y\| < 0.5 \cdot (\|a_x\| - h)$ and $a_x \geq 0$
0b11	landscape right	$153° + \varphi_h < \varphi < 207° - \varphi_h$	$\|a_y\| < 0.5 \cdot (\|a_x\| - h)$ and $a_x < 0$
0b10	portrait upside down	$27° + \varphi_h < \varphi < 153° - \varphi_h$	$\|a_y\| > 0.5 \cdot \|a_x\| + h$ and $a_y < 0$
0b00	portrait upright	$207° + \varphi_h < \varphi < 333° - \varphi_h$	$\|a_y\| > 0.5 \cdot \|a_x\| + h$ and $a_y \geq 0$

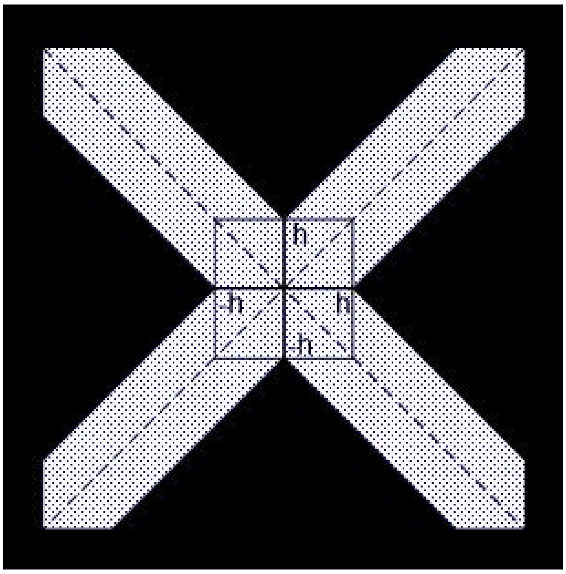

Figure 27: Hysteresis in the symmetrical mode

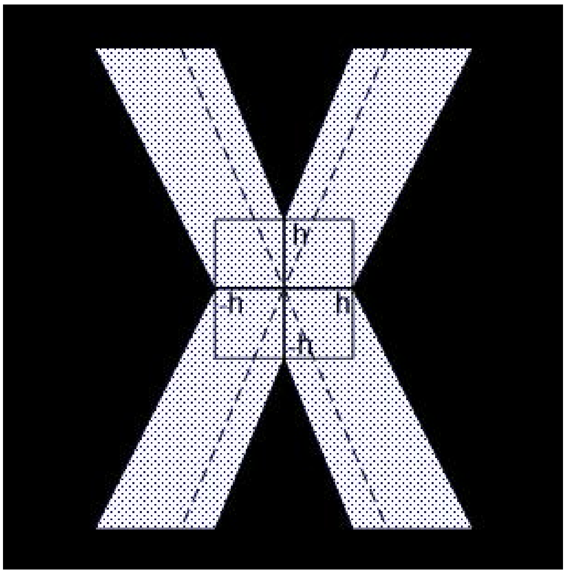

Figure 28: Hysteresis in the high asymmetrical mode

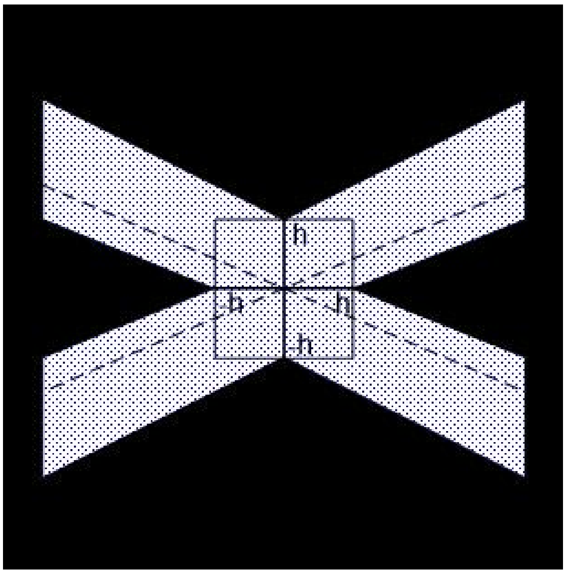

Figure 29: Hysteresis in the low asymmetrical mode

The hysteresis for detection of portrait upside and portrait downside is fixed to 11.5° which is corresponds to approximately 200 mg.

4.9.8 Tilt Detection

The feature of the tilt detection is derived from the Android[6]. A tilt interrupt is triggered when the attitude angle of the device changes by a value greater than configured angle threshold.

Enable and disable The tilt detection can be enabled and disabled via `FEAT_ENG_GPR_0.tilt_en`.

Configuration The minimum angle of tilt for event detection can be configured using `TILT_1.min_tilt_angle`. The value for the threshold for tilt angle set is computed as $256 \cdot \cos\theta$. The time interval, in which the gravity acceleration

[6] https://source.android.com/devices/sensors/sensor-types.html#tilt_detector

signal vector has to be estimated, is configured via the parameter `TILT_1.segment_size`. The low-pass filtering for the continuous estimation of the gravity acceleration vector can be configured with parameter `TILT_2.beta_acc_mean`.

Example The functional behavior of the tilt detector for default configuration settings is shown in figure 30.

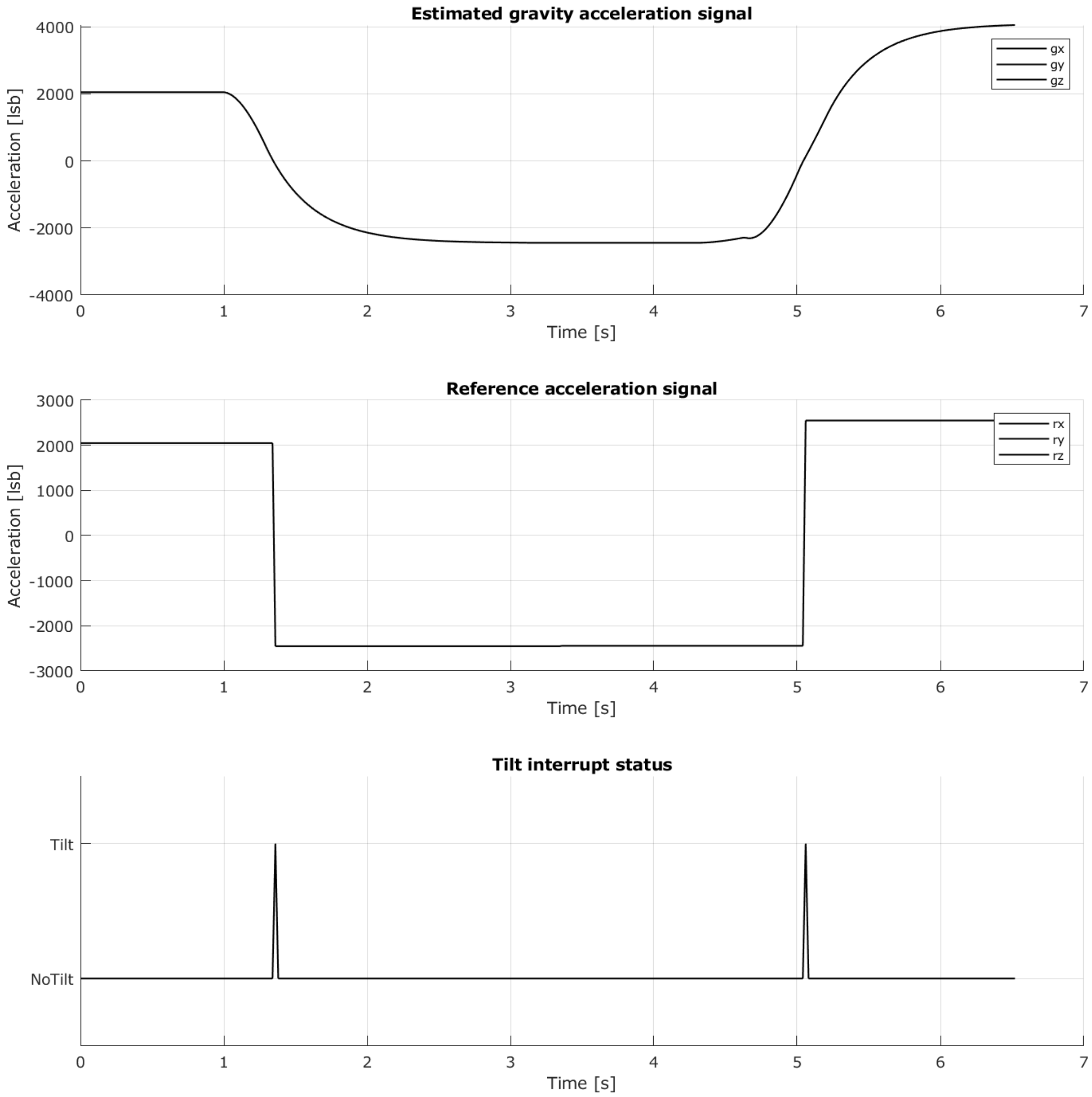

Figure 30: Functional behavior of tilt detection

Android compliance Tile detection feature can be configured to work in compliance with Android requirements by setting `GENERAL_SETTINGS_0.android_comp` as 0b1.

4.9.9 Fast Offset Compensation (FOC)

The BMA530 offers the advanced feature “Fast Offset Compensation” (FOC). In principle, the FOC uses the same registers as the manual compensation as described in chapter 4.2.10, but offers an easier seeking of the dedicated compensation values.

Per-requisites There are per-requisites to use the FOC feature:

- It is recommended to place the sensor in a stable and noiseless environment.

- Additionally, one of the accelerometer axes must be aligned in parallel to the gravity vector during the compensation process.
- It is also recommended to perform FOC for all axes, in which the axis is aligned in parallel to the gravity vector respectively.
-

Configuration To configure the FOC, the following parameters are available:

- `FOC_3.foc_apply_corr`: this option decides, if the feature updates the `ACC_OFFSET_0` - `ACC_OFFSET_5` with estimated offset values after feature completion automatically.
- `FOC_3.foc_filter_coeff`: number of 200 Hz accelerometer samples that are averaged to estimate the offset.
- `FOC_3.foc_axis_1g`: alignment information of the accelerometer axis to the gravity vector. Please note that BMA530 does not warn the user if the device is not static or an axis is not parallel to the gravitational vector.

Execution While the FOC feature is being executed, `FEAT_ENG_GP_FLAGS.foc_running` is set to 0b1. It will be cleared at the end of the compensation. Then, the FOC interrupts is raised, if it is mapped to any destination. Checking the interrupt is the recommended way to get the notification of FOC progress. After the FOC process is completed, it is recommended that the host disables the FOC feature in `FEAT_ENG_GPR_0.acc_foc_en`, so that the feature can be restarted again. Also, please note that, if `FOC_3.foc_apply_corr` is enabled, it is recommended to restart the accelerometer after the FOC completion, as was also suggested for the manual compensation in chapter 4.2.10.

As an example, Chapter 3 provides the recommend execution flow to perform FOC on one axis in combination with INT 1.

5 Digital Interfaces

The device provides one serial interface to the host. It acts as a slave to the host. The serial interface is configurable to the interface protocols SPI, I3C and I²C. Please note that, in the following chapter, only VDD is used to notate the power supply of the device, since VDD = VDDIO. The communication between host processor and the device happens over one of the interfaces: I²C, I3C or SPI (4-wire and 3-wire). Each register read operation includes the following number of inserted dummy bytes before the payload:

- I²C: 0
- I3C: 1
- SPI: 1

5.1 Electrical Specification

By default, the device operates in I²C mode or SPI 4-wire. The interface of the device can be configured to operate in a I3C or SPI 3-wire configuration as well. All digital interfaces share partly the same pins. The mapping for the primary interface of device is given in Table 25. The full pin mapping can be found in table 43.

Table 25: Pin mapping of the digital interface

Pin #	Name	I/O Type	Description	in SPI 4-wire	in SPI 3-wire	in I²C/I3C
2	INT1	Digital I/O	Interrupt pin 1 (or Serial Data)	SDO/MISO	INT1	INT1
3	INT2	Digital I/O	Interrupt pin 2 (or Chip Select for SPI)	CSB	CSB	INT2*
4	SDA	Digital I/O	Serial Data	SDI/MOSI	SDX	SDA
5	SCL	Digital I/O	Serial Clock	SCK	SCK	SCL

* Since pin 3 is used as CSB in SPI mode, it should not be driven low; see the chapter 7.2.1 for more details.

In Table 26, the electrical specifications of the interface pins are given.

Table 26: Electrical specification of the digital interface

Parameter	Symbol	Condition	Min	Typ	Max	Units
Pull-up resistance, CSB pin	R_{up}	Internal Pull-up Resistance to VDD	75	100	125	$k\Omega$
Input capacitance	C_{in}			5		$p\mathrm{F}$
I²C bus load capacitance (max. drive capability)	$C_{\mathrm{load,I2C}}$				400	$p\mathrm{F}$
I3C bus load capacitance (max. drive capability)	$C_{\mathrm{load,I3C}}$			10	50	$p\mathrm{F}$

5.2 Digital Interface Protocols

5.2.1 Protocol Selection

5.2.1.1 Automatic Protocol Selection of I²C or 4-wire SPI

After the power on or soft-reset, the sensor automatically selects protocol after the host sends an initial transaction, while the returning value is invalid. This initial transaction determines the serial interface in either I²C or 4-wire SPI for the later communication.

Additionally, if I²C is selected as the communication protocol, there are certain limitations on the electrical connections on pin 3, especially during the power-up when INT 2 is configured as input. In detail:

- When I²C/I3C is used, configure pin 3 to be output.
- When the output characteristics of pin 3 is disabled (or not yet enabled), please do not connect pin 3 to the ground.
- When the output characteristics of pin 3 is disabled (or not yet enabled), please do not connect pin 3 to a GPIO pin configured in the pull-down state.

Please find illustrations of the connection diagrams in chapter 7.2.1.

In practice, when I²C is selected as the primary protocol, it is strongly suggested to configure INT 2 as output via `INT2_CONF.mode`. This helps to prevent an unexpected erroneous detection of SPI and therefore improve the stability of I²C communication.

5.2.1.2 Protocol Selection of I3C

The host can switch to I3C from I²C protocol. Before accessing the registers of the device via I3C private transfers, the I3C must be enabled in the following way:

- The host has to set register `IF_CONF_1.if_i3c_en` = 1 via I²C write.
- The host has to apply the DAA (Dynamic Address Assignment) procedure via I3C CCC (Common Command Code) sequences according to the BCR (Bus Characteristics), the PID (Provisioned ID) and the static I²C slave address known by the host.
- After a successful DAA, the host can access registers via I3C private read and write.

5.2.1.3 Protocol Selection of 3-wire SPI

The device supports both 4-wire and 3-wire SPI interfaces. The device operates in the 4-wire configuration by default. It can be switched to 3-wire configuration by setting register `IF_CONF_1.if_spi3_en` = 1. In the 3-wire configuration, the pin SDX is used as the common data input and output pin. Notably, although the change of SPI interface configuration is executed immediately, the SPI 3-wire configuration is effective only at the first read operation following the change to SPI 3-wire configuration and vice versa.

5.2.2 SPI Protocol

The SPI interface of the device encompasses two orthogonal aspects, namely 3-wire or 4-wire interface and mode 0 or mode 3 configuration. The signaling conventions applicable to the supported SPI modes are defined in Table 27.

Table 27: SPI mode 0 and mode 3 configuration

SPI mode	Description
0	CPOL = 0 and CPHA = 0
3	CPOL = 1 and CPHA = 1

Specifically:

- 3-wire and 4-wire configurations: SPI 3-wire mode can be configured through bit `IF_CONF_1.if_spi3_en`, as described in the previous chapter.
- Mode 0 and mode 3 configurations: The selection between SPI mode 0 and 3 is performed automatically by detecting the value of the SCK signal at the first falling edge of the CSB signal.

The following chapters describe the protocol properties for each SPI configuration and mode.

5.2.2.1 SPI Timing specification

The timing specifications are stated in Table 28 for the SPI interface of the device. Additionally, figures 31 to 33 show the definition of the SPI timings. Here, the 4-wire SPI protocol with mode 0 is presented as an example.

Table 28: SPI interface timing specifications

Parameter	Comment	Symbol	Min	Typ	Max	Units
CSB lead time		T_setup_csb	40			ns
CSB lag time		T_hold_csb	40			ns
SDI setup time		T_setup_sdi	20			ns
SDI hold time		T_hold_sdi	20			ns
SDO output delay time	Load = 30pF	T_delay_sdo,nv			30	ns
SDO release delay time	Load = 30pF	T_release_sdo,nv			30	ns
SDO drive delay time	Load = 30pF	T_drive_sdo,nv			30	ns
SCX frequency		F_sck,nv			10	MHz
SCX pulse high time		T_high_scx,nv	45			ns
SCX pulse low time		T_low_scx,nv	45			ns
Idle time after write access	Active state	T_wr_idle_act,spi	2			µs
	Suspend state	T_wr_idle_susp,spi	450			µs
Idle time after read access		T_rd_idle,spi	2			µs

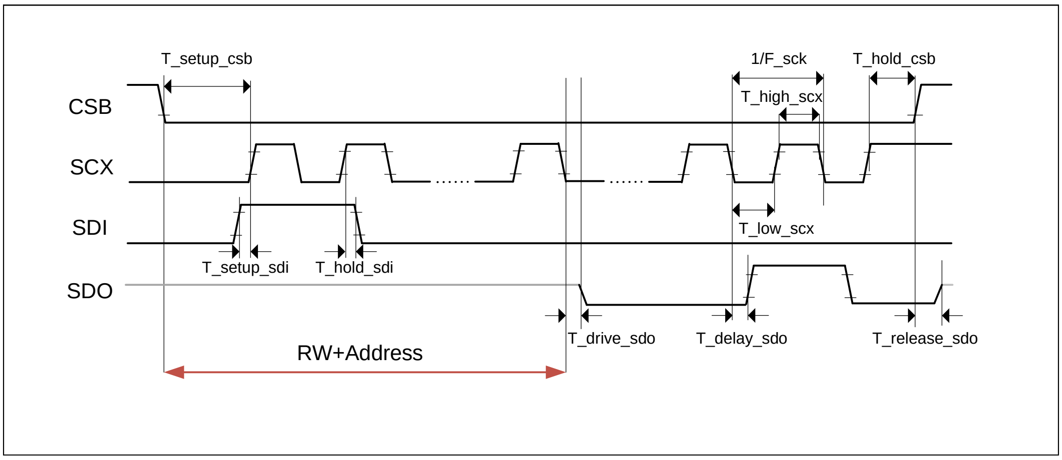

Figure 31: SPI timing diagram

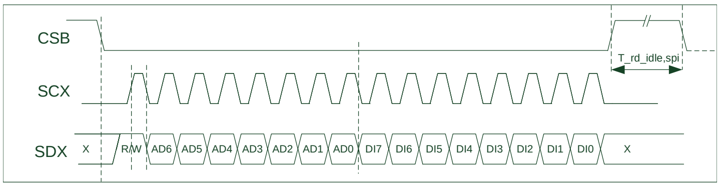

Figure 32: SPI idle read timing

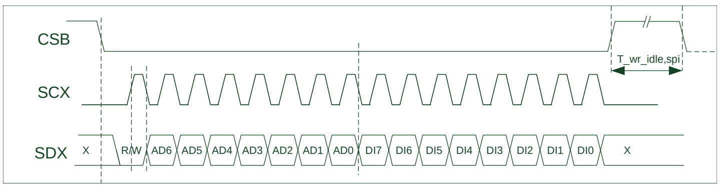

Figure 33: SPI idle write timing

5.2.2.2 4-wire SPI

The 4-wire SPI interface is based on the following pins:

- CSB (chip select low active)
- SCX (serial clock)
- SDI (serial data input)
- SDO (serial data output)

The communication starts (stops), when the CSB is pulled low (high) by the host. The SDX input receiver is enabled (disabled), when the CSB is pulled low (high) by the host. In figures 34 to 38, the basic operation waveform is presented with respect to the 4-wire SPI. The 4-wire SPI mode 3 and mode 0 configurations are equivalent in terms of multiple-byte write, single byte read and multiple-byte read operations using the respective SCX signaling properties. Hence these modes are omitted here.

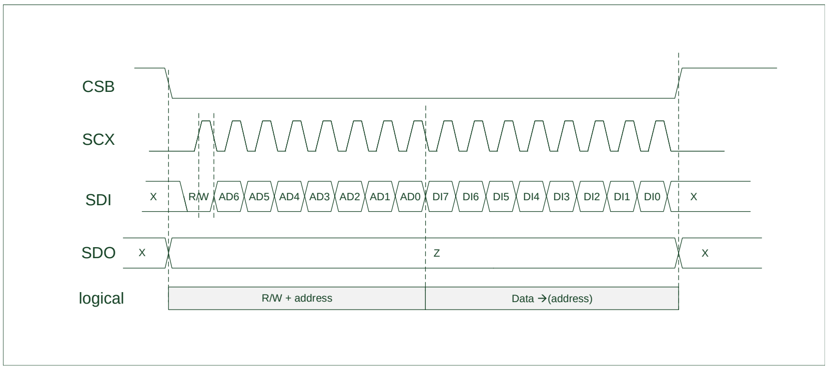

Figure 34: Single-byte write operation of 4-wire SPI with mode 0

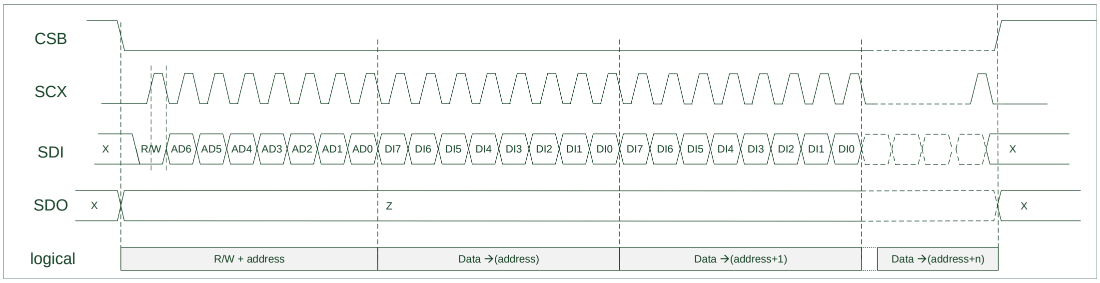

Figure 35: Multiple-byte write operation of 4-wire SPI with mode 0

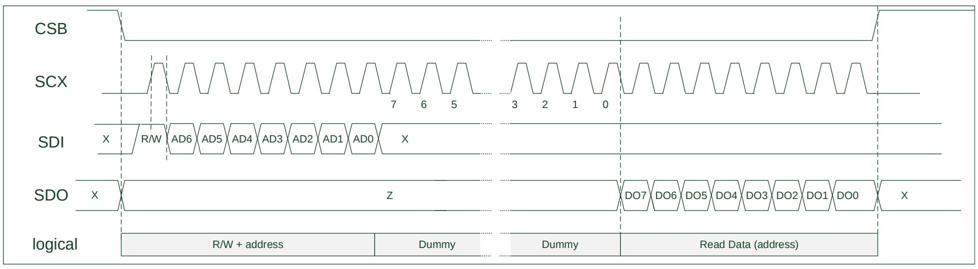

Figure 36: Single-byte read operation of 4-wire SPI with mode 0

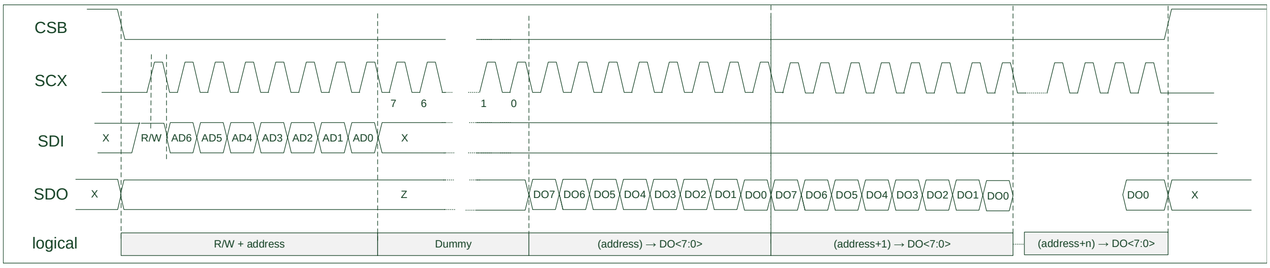

Figure 37: Multiple-byte read operation of 4-wire SPI with mode 0

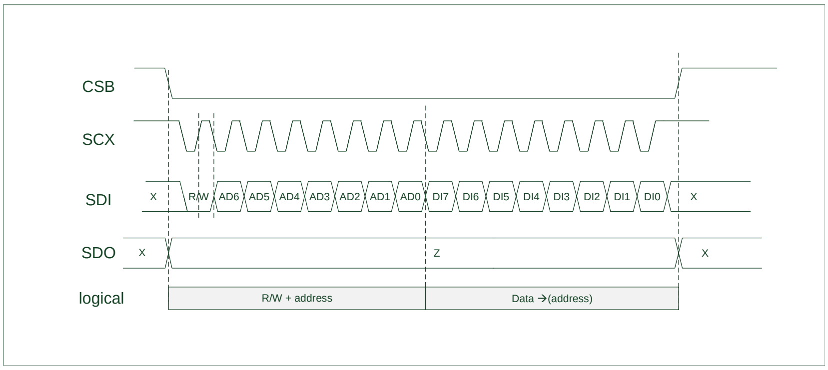

Figure 38: Single-byte write operation of 4-wire SPI with mode 3

5.2.2.3 3-wire SPI

The 3-wire SPI interface is based on the following pins:

- CSB (chip select low active)
- SCX (serial clock)
- SDX (serial data input and output)

The 3-wire SPI interface mode uses the SDX pin for both data input and output. The write command for the 3-wire SPI is identical to the 4-wire SPI write command. When a read command is performed, output data appear at the SDX pin once the last address bit AD0 has been latched. Output data are synchronized at falling edge of SCX. Both input and output data shall be captured at rising edge of SCX. The SDX input receiver is enabled when the CSB is pulled low by the host, and disabled when CSB is pulled high (write access) or output data is driven (read access). In figure 39, the basic operation waveform is presented with respect to the 3-wire SPI, where the single-byte read operation with mode is given as an example.

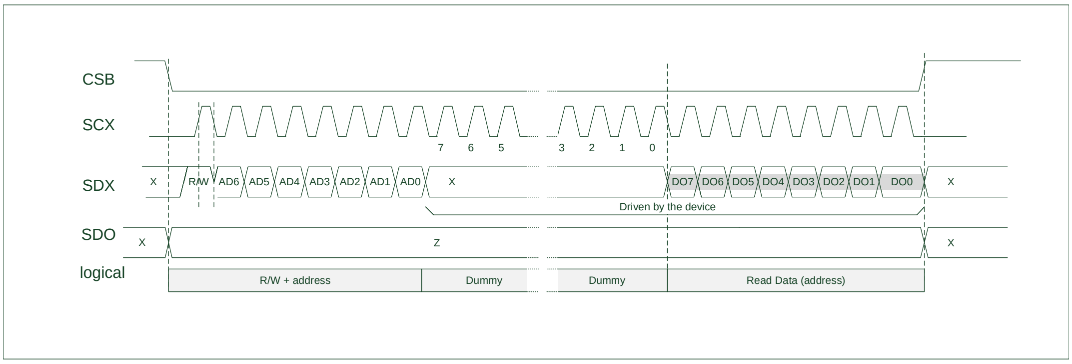

Figure 39: Single-byte read operation of 3-wire SPI with mode 0

5.2.3 I²C Protocol

The device supports the following I²C modes:

- Normal mode (100 kHz)
- Fast mode (400 kHz)
- Fast mode plus (Fm+) (1 MHz)

The default 7 bits I²C address is 0x18.

I²C Timing specification The I²C timing specification of the device is given in Table 29, figure 40 and figure 41.

Table 29: I²C timing requirements (standard mode, fast mode snf fast mode plus)

Parameter	Comment	Symbol	Min	Typ	Max	Unit
SCL frequency		F_scl	0		1000	kHz
Fall time		T_F	0		300	ns
Rise time [1]	Load = 400 pF	T_R	20		300	ns
SCL low period		T_LOW	0.5			µs
SCL high period		T_HIGH	0.26			µs
Hold time (repeated start condition)		T_HD, STA	0.26			µs
Set-Up time (repeated start condition)		T_SU, STA	0.26			µs
Data hold time, data written to slave		T_HD, DAT, slv	0			µs
Data hold time, data written to host		T_VD, DAT	120		450	ns
Data set-up time		T_SU, DAT	50			ns
Set-up time stop condition		T_SU, STO	0.26			µs
Bus free time		T_BUF	0.5			µs
Spike suppression		T_SP	50			ns
Noise margin at low input level		V_{nL}		0.1 * VDD		V
Noise margin at high input level		V_{nH}		0.2 * VDD		V
Idle time after write access	Suspend state	T_wr_idle_susp, I²C	450			µs

[-1] Determined by the external pull-up resistor.
[0] Determined by the external pull-up resistor.
[1] Determined by the external pull-up resistor.

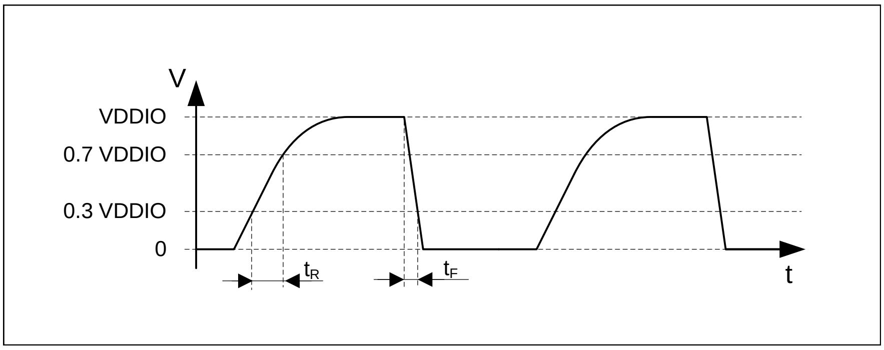

Figure 40: Definition of rise- and fall-time of I²C interface signals

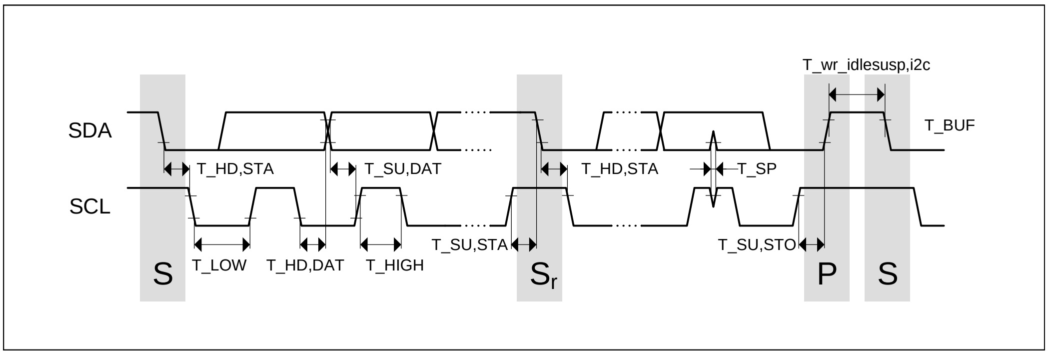

Figure 41: I²C timing diagram

5.2.3.1 I²C Write Operation

Figure 42 depicts the I²C write telegram for a single-byte write operation. The telegram begins with a start condition generated by the host, followed by 7 bits slave address and a write bit (R/W = 0). The slave sends an acknowledge bit (ACK = 0) and releases the bus. Subsequently, the host is expected to send the one-byte register address. Please note that only the first 7 bits (right aligned) are the valid address bits, while the MSB is ignored.The slave shall again acknowledge the transmission and wait for the 8 bits data which shall be written to the specified register address. After the slave acknowledges the data byte, the host generates a stop signal and terminates the writing protocol.

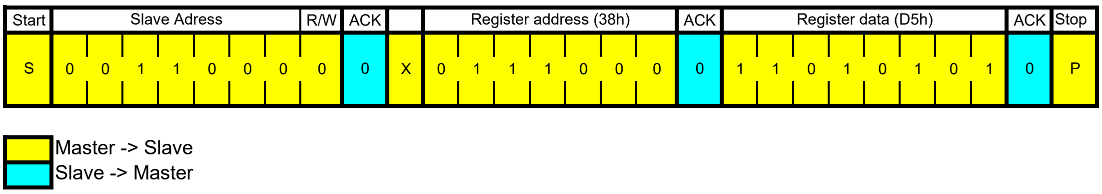

Figure 42: Single-byte write operation of I²C

The device also supports multi-byte write operation when operating in I²C mode. The multi-byte write telegram is depicted in Fig. . The telegram begins with a start condition generated by the host, followed by 7bits slave address and a write bit (R/W = 0). The slave sends an acknowledge bit (ACK = 0) and releases the bus. Subsequently, the host sends the one-byte register address. Agian, please note that only the first 7 bits (right aligned) are the valid address bits, while the MSB is ignored. The slave shall again acknowledge the transmission and wait for several 8-bit wide data words. The first data word is written to the specified register address. The register address pointer is automatically incremented for each data word (please see chapter 5.2.5 for details). Each received data word is written to the register referenced by the current register address pointer. The slave acknowledges each data byte. When no more data words need to be written, the host generates a stop signal and terminates the writing protocol.

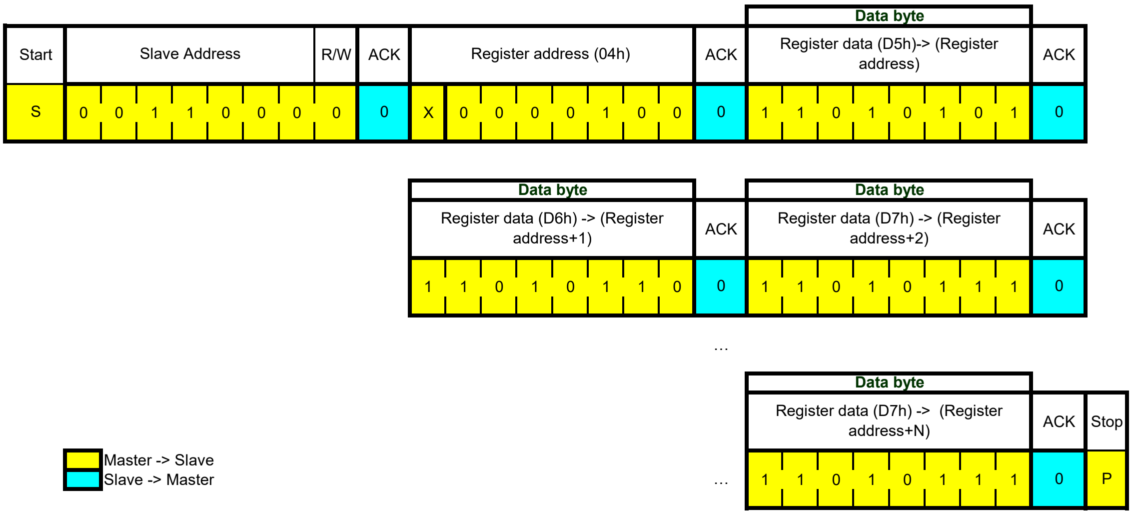

Figure 43: Multiple-byte write operation of I²C

5.2.3.2 I²C Read Operation

The I²C read operation supports multiple bytes reading. A read command consists of a 1-byte I²C write phase followed by I²C read phase. The two I²C transmissions must be separated by a repeated start condition (Sr). The I²C write phase addresses the slave and sends the register address to be read. After the slave acknowledges the transmission, the host is expected to generate a start condition and then to send the slave address together with a read bit (R/W = 1). Then the host releases the bus and waits for the data bytes to be read out from slave. After each data byte, the host has to generate an acknowledge bit (ACK = 0) to enable further data transfer. A NACK (ACK = 1) from the host stops the data transferring from slave. The slave releases the bus so that the host can generate a STOP condition and terminate the transmission.

Figure 44: Multiple-byte read operation of I²C

Multiple-byte read transmissions within one read command are also possible. Once a new read transmission starts, the start address is set to the register address specified in the latest I²C write command. In this way, repetitive multi-bytes reads from the same starting address are possible. The default start address is 0x00.

Figure 45: Multiple read transmissions of I²C from the same start address

5.2.4 I3C Protocol

The device supports the I3C protocol with the features:

- I3C single data rate (SDR) mode with up to 12.5 MHz data rate
- I²C compatibility
- In-Band Interrupt (IBI)
- Timing control (TC) synchronous mode
- Timing control asynchronous (TC Async) 0 and 1 mode

To select the I3C protocol, please refer to chapter 5.2.1.2.

The I3C interface of the device is compatible with the Specification for I3C® Improved Inter Integrated Circuit Version 1.1.1 – 11 June 2021 (hereinafter called "MIPI specification") available at `http://www.mipi.org/specifications/i3c-sensor-specification`.

5.2.4.1 Bus Identifier

The I3C protocol provides the following identifiers in Table 30 for the I3C communication:

Table 30: I3C Bus Identifier

Bus identifier	Value
I3C Provisioned ID (PID) [48:0]	0x077000011001
Device Characteristics Register (DCR) [8:0]	0x41[1]
Bus Characteristics Register (BCR) [8:0]	0x27[2, 3, 4]

[1]: This register describes the I3C compliant Device type as defined in MIPI specification.
[2]: BCR[5] is set to 0b1, meaning that device supports solely SDR.
[3]: BCR[2] is set to 0b1, meaning that an accepted IBI is followed by the mandatory byte (MDB).
[4]: BCR[0] is set to 0b1 according to MIPI specification, Table 123. However, the device supports the typical clock frequency of 12.5 MHz. See Table 40.

5.2.4.2 In-Band Interrupt (IBI)

The IBI is enabled by default after power-up and the Dynamic Address Assignment (DAA). If DAA is reset via RSTDAA CCC, IBI will be disabled automatically. As long as no new dynamic address has been assigned, IBI will stay disabled and must be enabled explicitly by ENEC CCC in contrast to the situation after power-up.

IBI Payload Format The IBI payload is defined in Table 31 and Table 32 in dependency of TC.

Table 31: IBI Payload Format

TC configuration	Byte No.	Name	Bit	Description
No TC	1	MDB	[7:5]	Interrupt Group Identifier fixed to 0x0
			[4:0]	Interrupt status bits provided by I3C interrupts status[1]
	2	Additional Byte	[7:0]	Interrupt status bits provided by I3C interrupts status[1]
TC Async 0	1	MDB	[7:5]	Interrupt Group Identifier fixed to 0x4
			[4:0]	Interrupt status bits provided by I3C interrupts status[1]
	2	T_C1_LSByte	[7:0]	LSByte of Target C1 counter value
	3	T_C1_MSByte	[7:0]	MSByte of Target C1 counter value
	4	T_C2	[7:0]	Target C2 counter value
	5	Additional Byte	[7:0]	Interrupt status bits provided by I3C interrupts status
TC Async 1	1	MDB	[7:5]	Interrupt Group Identifier fixed to 0x4
			[4:0]	Interrupt status bits provided by I3C interrupts status[1]
	2	T_C1_LSByte	[7:0]	LSByte of Target C1 counter value
	3	T_C1_MSByte	[7:0]	MSByte of Target C1 counter value
	4	T_C2	[7:0]	Target C2 counter value
	5	T_AME	[7:0]	Target AME counter value
	6	Additional Byte	[7:0]	Interrupt status bits provided by I3C interrupts status

[1]: Please refer to Table 32 for the mapping of I3C interrupts status bits to IBI payload.

Table 32: Mapping of I3C interrupts status bits to IBI payload

Name	Bit	I3C interrupts status bit
MDB	[0]	`INT_STATUS_I3C_0.acc_drdy_int_status`
	[1]	`INT_STATUS_I3C_0.fifo_wm_int_status`
	[2]	`INT_STATUS_I3C_0.fifo_full_int_status`
	[3]	`INT_STATUS_I3C_0.gen_int1_int_status`
	[4]	`INT_STATUS_I3C_0.gen_int2_int_status`
Additional Byte	[0]	`INT_STATUS_I3C_0.gen_int3_int_status`
	[1]	`INT_STATUS_I3C_0.acc_foc_int_status`
	[2]	`INT_STATUS_I3C_0.stap_int_status`
	[3]	`INT_STATUS_I3C_1.dtap_int_status`
	[4]	`INT_STATUS_I3C_1.ttap_int_status`
	[5]	`INT_STATUS_I3C_1.vad_int_status`
	[6]	`INT_STATUS_I3C_1.self_wake_up_int_status`
	[7]	`INT_STATUS_I3C_1.feat_eng_err_int_status`

IBI Payload Abortion When a I3C controller reads the IBI payload from BMA530, it is controller's responsibility to end the message. The controller can either determine the number of payload byte through Table 31 and Table 32, or by using GETMRL.

5.2.4.3 Common Command Code (CCC)

Support for Defining Bytes Defining Bytes formats for the following CCCs are not supported and will always respond with format 1:

- GETSTATUS with format 1 (defined by GETCAP3[4] = 0b0) returning 2 bytes
- GETCAPS with format 1 (defined by GETCAP3[3] = 0b0) returning 4 bytes
- GETMXDS with format 1: Bytes MaxWr and MaxRd will be returned.

GETCAPS GETCAPS (Get Optional Feature Capabilities) format 1 returns the following four bytes in Table 33.

Table 33: GETCAPS returning value

Byte No.	Byte	Bit	Name	Description
1	GETCAP1	[7:0]	High Data Rate (HDR) Mode	No HDR supported
2	GETCAP2	[7:6] = 0b0	Defining Byte Support	Not supported
		[5:4] = 0b0	Group Address Capabilities	Not supported
		[5:4] = 0x1	I3C 1.x Specification Version	Minor version number of the MIPI I3C Specification
3	GETCAP3	[7] = 0b0	Reserved	
		[6] = 0b0	Pending Read Notification	No Pending Read Notification for IBI
		[5] = 0b0	HDR-BT CRC32 Support	Not supported
		[4:3] = 0b0	Defining Byte Support	No Defining Byte support for GETCAPS and GETSTATUS
		[2:1] = 0b0	Device to Device Transfer	Not supported
		[0] = 0b0	Multi-Lane for Speed Support	Not supported
4	GETCAP4	[7:0]=0x00	Reserved	

GETSTATUS GETSTATUS (Get Device Status) format 1 returns the following two bytes in Table 34.

Table 34: GETSTATUS returning value

Bits	Name	Description
[15:8] = 0x00	Reserved	
[7:6] = 0b00	Activity mode	Not used.
[5]	Protocol Error	If 1'b1: The device detected a protocol error since the last status read.The device might or might not be able to check for such errors. Note that this value self-clears upon every successful completion of a host read of the device's status.
[4] = 0b0	Reserved	
[3:0]	Pending Interrupt	Contains the interrupt number of any pending interrupt, or 0 if no interrupts are pending. This encoding allows for up to 15 numbered interrupts. If more than one interrupt is set, then the highest priority interrupt shall be returned.

GETMXDS The GETMXDS (Get Max Data Speed) format 1 returns the following five bytes in Table 35.

Table 35: GETMXDS returning value

Byte No.	Byte	Bit	Name	Description
1	MaxWr	[7:4]	Reserved	
		[3] = 0b0	Defining Byte Support	No Defining Byte format supported
		[2:0] = 0b000	Maximum Sustained Data Rate for non-CCC Messages sent by Controller Device to Target Device	f_{SCL} Max
2	MaxRd	[6] = 0b0	Write-to-Read Permits Stop Between	STOP would cancel the Read
		[5:3] = 0b111	Clock to Data Turnaround Time (t_{SCO})	t_{SCO} is > 12 ns
		[2:0] = 0b000	Maximum Sustained Data Rate for non-CCC Messages sent by Target Device to Controller Device	fSCL Max
3	maxRdTurn	[7:0]=0x00	Maximum Read Turnaround Time	fSCL Max
4		[15:8]=0x00		
5		[23:16]=0x00		

GETMWL The GETMWL (Get Max Write Length) returns the following two bytes in Table 36. Please note that SETMWL (Set Max Write Length) is not supported.

Table 36: GETMWL returning value

Byte No.	Byte	Bit	Name	Description
1	MWL MSB	[8:0] = 0xFF	MSByte of Max Write length	No limit
2	MWL LSB	[8:0] = 0xFF	LSByte of Max Write length	

GETMRL The GETMRL (Get Max Read Length) returns the following three bytes in 36. Please note that SETMRL is not supported.

Table 37: GETMRL returning value

Byte No.	Byte	Bit	Name	Description
1	MWL MSB	[8:0] = 0xFF	MSByte of Max Write length	No limit
2	MWL LSB	[8:0] = 0xFF	LSByte of Max Write length	
3	IBI PL	[8:0]	IBI Payload Size	**Values:** 0x02: When No TC is activated. IBI payload size is 3 bytes, including MDB + 1 additional byte. 0x05: When TC Async 0 is activated. IBI payload size is 5 bytes, including MDB + 4 additional byte. 0x06: When TC Async 1 is activated. IBI payload size is 6 bytes, including MDB + 5 additional byte.

GETXTIME The GETXTIME (Get Exchange Timing Support Information) returns the following four bytes in Table 36.

Table 38: GETXTIME returning value

Byte No.	Name	Description
1	Supported Modes Byte	TC Async 0 and TC Async 1 are supported.
2	State Byte	TC Async 0 and TC Async 1 enabling state is shown. Overflow bit is related to Async Mode 0 and 1.
3	Frequency Byte	Fixed to 0x0D: 6.5 MHz.
4	Inaccuracy Byte	Fixed to 0x14: 2%.

The overflow bit in Byte No. 2 will be set in case of any TC Async counter overflow, and will be cleared automatically in the following way:

- TC Async 0: a new IBI is requested by hardware interrupt.
- TC Async 1: an I3C start has been found after a drdy event.

Note that the overflow bit will not be cleared by the GETXTIME.

5.2.4.4 I3C Timing Specification

The device supports I3C single data rate (SDR) mode according to the MIPI specification. The I3C timing specification of the device is given in Table 39 and Table 40.

Table 39: Open drain timing parameters of I3C

Parameter	Diagram in MIPI specification	Symbol	Min	Max	Unit
SCL clock low period	Figure 233 Figure 234	t_{LOW_OD}	200		ns
		$t_{DIG\_OD\_L}$	$t_{LOW\_ODmin} + t_{fDA\_ODmin}$		ns
SCL clock high period (for Broadcast Address)		t_{HIGH_INIT}	200		ns
SCL clock high period (for Mixed Bus)	Figure 230	t_{HIGH}		41	ns
		t_{DIG_H}		tHIGH + tCF	ns
SCL clock high period (for Pure Bus)	Figure 230	t_{HIGH}	24		ns
		t_{DIG_H}	32		ns
SDA fall time	Figure 233	t_{fDA_OD}		12	ns
SDA setup time	Figure 233	t_{SU_OD}	3		ns
Clock after Start	Figure 233	t_{CAS}	38.4e-9	ENTAS0: 1e-6	s
				ENTAS1: 100e-6	s
				ENTAS2: 2e-3	s
				ENTAS3: 50e-3	s
Clock before Stop	Figure 233	t_{CBP}	$t_{CASmin}/2$		ns
Bus available condition		t_{AVAIL}	1		µs
Bus idle condition		t_{IDLE}	200		µs

Table 40: Push-pull timing parameters for SDR of I3C

Parameter	Diagram in MIPI specification	Symbol	Min	Typ	Max	Unit
SCL clock frequency		f_{SCL}	0.01	12.5	12.9	MHz
SCL clock low period	Figure 230	t_{LOW}	24			ns
		t_{DIG_L}	32			ns
SCL clock high period (for Mixed Bus)	Figure 230	t_{HIGH_MIXED}	24			ns
		$t_{DIG\_H\_MIXED}$	32		45	ns
SCL clock high period (for Pure Bus)	Figure 230	t_{HIGH}	24			ns
		t_{DIG_H}	32			ns
Clock-in to data-out	Figure 236	t_{SCO}			12	ns
SCL clock rise time	Figure 230	t_{CR}			150e06 * $1/f_{SCL}$	ns
SCL clock fall time	Figure 230	t_{CF}			150e06 * $1/f_{SCL}$	ns
SDA signal hold time	Figure 235	t_{HD_PP}	$t_{CR/F}$ + 3			ns
SDA signal setup time	Figure 235 and Figure 236	t_{SU_PP}	3			ns
Clock after SR	Figure 239	t_{CASr}	$t_{CASmin}/2$			ns
Clock before SR	Figure 239	t_{CABr}	$t_{CASmin}/2$			ns
Capacitive bus load		C_B			50	pF

5.2.4.5 I3C Private Write Operation

The I3C write operation supports single-byte as well as multi-byte (burst) writing. figure 46 depicts the I3C write transfer for single-byte write operation. The transfer begins with a start condition generated by the host, followed by 7-bit I3C dynamic slave address and a write bit (R/W = 0). Then, the slave sends an acknowledge bit (ACK = 0) and releases the bus. Subsequently, the host is expected to send the register address (only the first 7-bit (right aligned) are the valid address bits, the MSB shall be ignored). Compared to I²C, the slave will not acknowledge the data bytes after the transmission. Instead, the I3C master is transmitting a parity bit during the T-bit phase. The next 8-bit data shall be written to the specified register address. After the final T-bit, the host generates a stop signal and terminates the writing protocol.

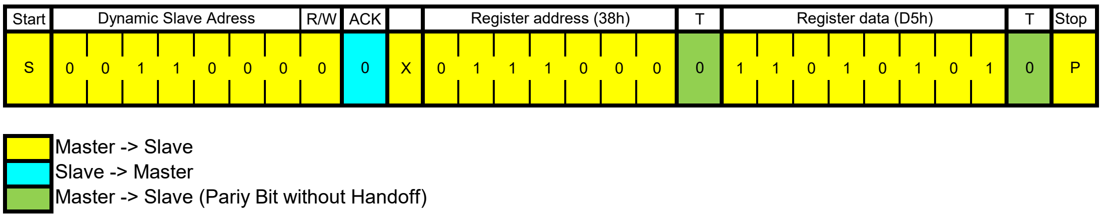

Figure 46: Single-byte write protocol of I3C

The device also supports multi-byte (burst) write operation in I3C mode. The multi-byte write telegram is depicted in Fig. 47. The telegram begins with a start condition generated by the host, followed by 7-bit dynamic slave address and a write bit (R/W = 0). The slave sends an acknowledge bit (ACK = 0) and releases the bus. Subsequently the host sends the one byte register address (only the first 7-bit (right aligned) are the valid address bits, the MSB shall be ignored). The I3C master is transmitting a parity bit during the T-bit phase. The first data word is written to the specified register address. The register address pointer is automatically incremented for each data word. Each received data word is written to the register referenced by the current register address pointer. When no more data words need to be written, after the final T-bit, the host generates a stop signal and terminates the writing protocol.

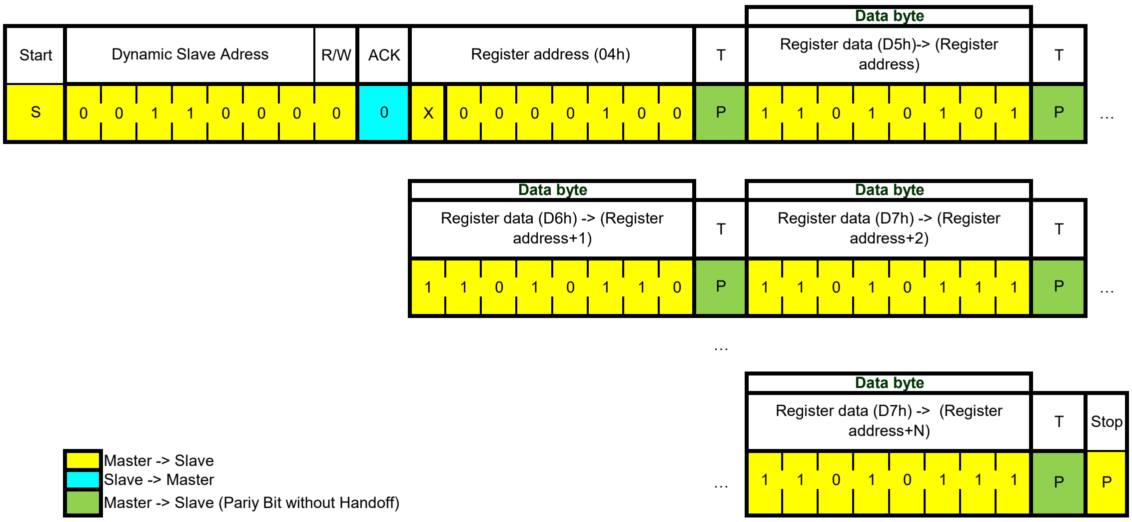

Figure 47: Multi-Byte write protocol of I3C

5.2.4.6 I3C Private Read Operation

The I3C read operation supports single-byte as well as multi-byte (burst) reading. **Please note that, burst accesses are not permitted when sensor is in suspend mode.** A read command consists of a 1-byte I3C write phase followed by an I3C read phase. The two I3C transmissions must be separated by a repeated start condition (Sr) as shown in figure 48 or a stop followed by start condition (P followed by S) as shown in figure 49. The I3C write phase addresses the slave and sends the register address to be read. After the slave acknowledges the transmission, the host is expected to generate a start condition and then to send the dynamic slave address together with a read bit (R/W = 1). Then, the host releases the bus and waits for the data bytes to be read out from slave. After each data byte, the slave can continue the burst by driving the T-Bit high until the rising edge of SCL and release its driver right after SCL rising edged the to give the master the possibility to create a STOP or RESTART condition to terminate the transmission. If both the slave and master are keeping the T-Bit high, the burst will continue. The register address is automatically incremented and more than one byte can be sequentially read out (please refer to chapter 5.2.5 for address handling). Once a new data read transmission starts, the start address is set to the register address specified in the latest I3C write command. By default, the start address is set at 8h'00. In this way, repetitive multi-bytes reads from the same starting address are possible, as shown in figure 50.

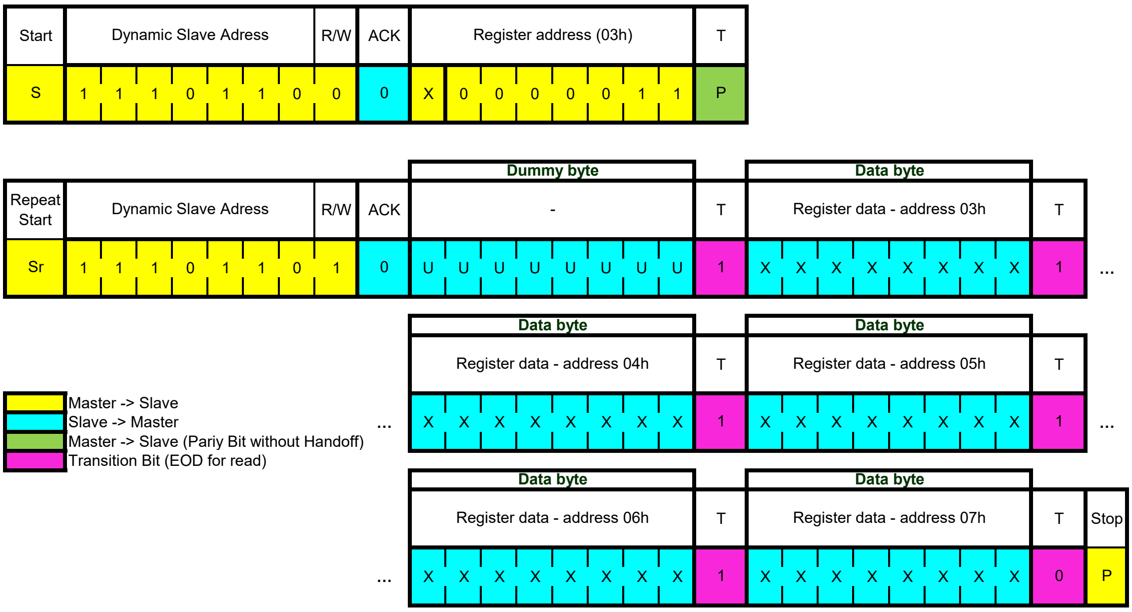

Figure 48: Multi-byte read protocol of I3C with repeated start

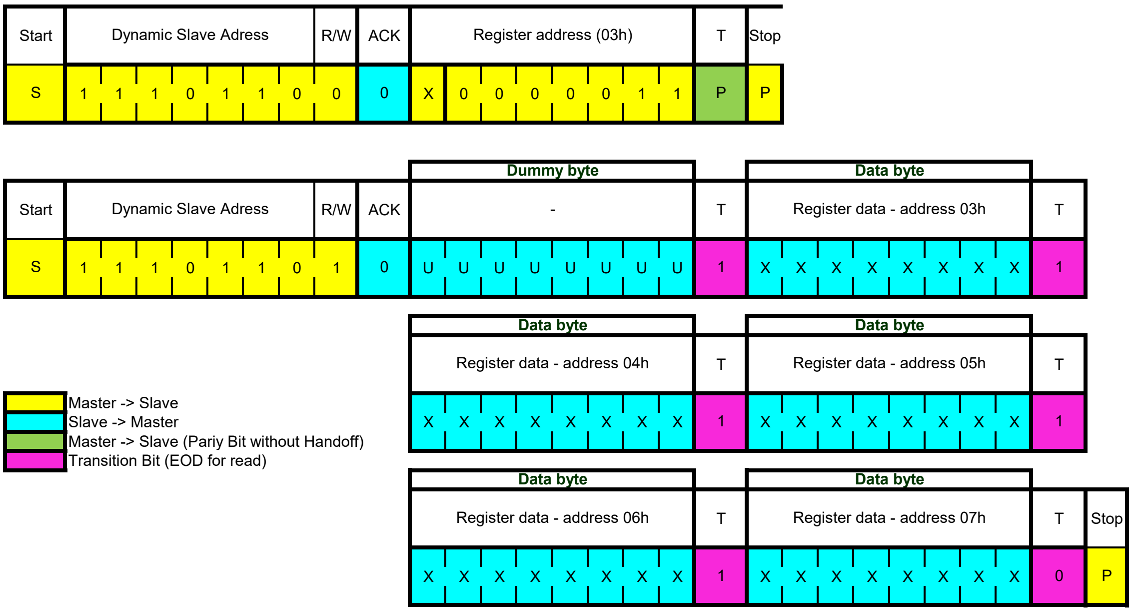

Figure 49: Multi-byte read protocol of I3C with stop-start

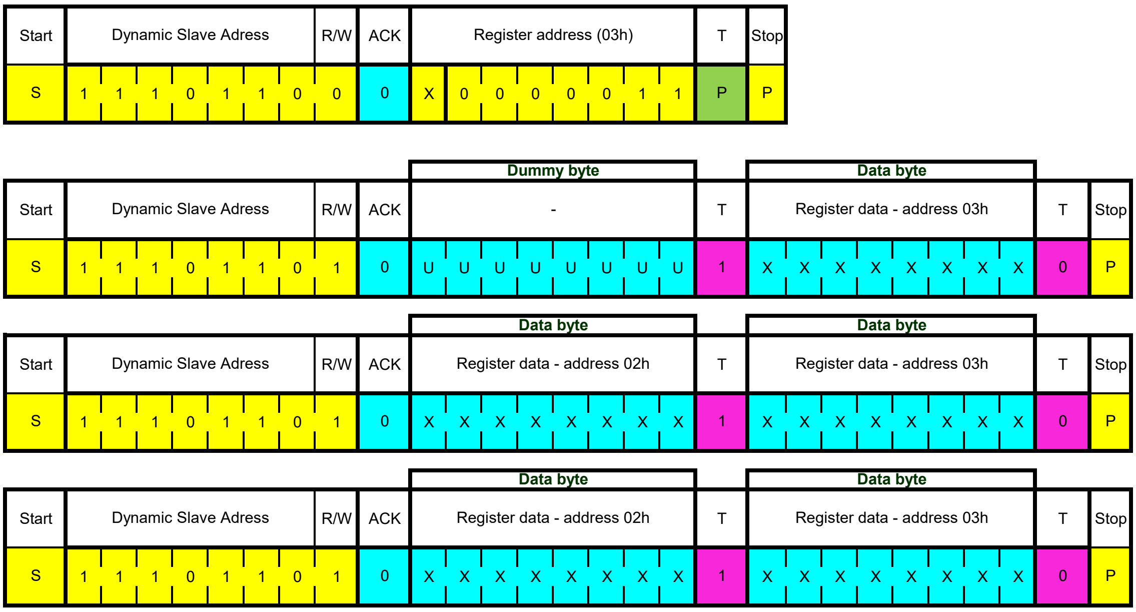

Figure 50: Multi-byte read protocol of I3C from the same start address with stop-start

5.2.5 Automatic Address Increment for Burst Access

Whenever a host accesses the device with a burst transfer, the register address is incremented upon each data byte, independent of the access type (read or write). When the highest address (0x7F) is reached, it wraps around to the start address (0x00). On dedicated addresses, the auto-incrementing is stalled, and all succeeding bytes are transferred with this stalled address. Auto-incrementing is stalled, even when the address is not the first one within a burst access. These stalling addresses are:

- `FIFO_DATA_OUT` (read access only)
- `FEATURE_DATA_TX` (write and read access)

6 Memory Map

The device can be operated for all standard features directly through registers. The registers are described in the register map in Section 6.1. The configuration and extended outputs of the advanced features provided by the feature engine can be accessed through the extended register map. The layout of the extended registers is described in Section 6.2.

In case a bit field of a register is marked as “reserved”, the value read from it cannot be assumed to be “0x0” in every case . To reserved bit fields only “0x0” should be written to.

6.1 Register Map Description

The description of the register map is split into the overview of the register map and a detailed description for each register. The access to the extended register map through the registers `FEATURE_DATA_ADDR` and `FEATURE_DATA_TX` is explained in Section 6.2.

6.1.1 Register Map Overview

The Table 41 provides an overview of the register map of the device.

Table 41: Register map overview

Legend			Read-only		Read/Write		Write-only		Reserved	
Addr	**Name**	**Reset value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x00	CHIP_ID	0xC2	chip_id							
...	-	-	reserved							
0x02	HEALTH_STATUS	0x00	reserved				sensor_health_status			
...	-	-	reserved							
0x04	CMD_SUSPEND	0x00	reserved							suspend
...	-	-	reserved							
0x10	CONFIG_STATUS	0x00	reserved						acc_co...	feat_e...
0x11	SENSOR_STATUS	0x04	reserved					sensor:.	tempera...	acc_da...
0x12	INT_STATUS_INT1_0	0x00	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...
0x13	INT_STATUS_INT1_1	0x00	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...
0x14	INT_STATUS_INT2_0	0x00	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...
0x15	INT_STATUS_INT2_1	0x00	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...
0x16	INT_STATUS_I3C_0	0x00	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...
0x17	INT_STATUS_I3C_1	0x00	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...
0x18	ACC_DATA_0	0x00	acc_x_7_0							
0x19	ACC_DATA_1	0x80	acc_x_15_8							
0x1A	ACC_DATA_2	0x00	acc_y_7_0							
0x1B	ACC_DATA_3	0x80	acc_y_15_8							
0x1C	ACC_DATA_4	0x00	acc_z_7_0							
0x1D	ACC_DATA_5	0x80	acc_z_15_8							
0x1E	TEMP_DATA	0x00	temp_data							
0x1F	SENSOR_TIME_0	0x00	sensor_time_7_0							
0x20	SENSOR_TIME_1	0x00	sensor_time_15_8							
0x21	SENSOR_TIME_2	0x00	sensor_time_23_16							
0x22	FIFO_LEVEL_0	0x00	fifo_fill_level_7_0							
0x23	FIFO_LEVEL_1	0x00	reserved					fifo_fill_level_10_8		
0x24	FIFO_DATA_OUT	0x80	fifo_data_out							
...	-	-	reserved							
0x30	ACC_CONF_0	0x0F	reserved				sensor_ctrl			
0x31	ACC_CONF_1	0xA7	power_...	acc_bwp			acc_odr			
0x32	ACC_CONF_2	0x0E	acc_dr...	reserved		noise_...	acc_iir_ro		acc_range	

Table 41: Register map overview *(continued)*

Legend			Read-only		Read/Write		Write-only		Reserved	
Addr	**Name**	**Reset value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x33	TEMP_CONF	0x00	reserved					temp_rate		
0x34	INT1_CONF	0x00	reserved				lvl	od	mode	
0x35	INT2_CONF	0x00	reserved				lvl	od	mode	
0x36	INT_MAP_0	0x00	gen_int1_int_map		fifo_full_int_map		fifo_wm_int_map		acc_drdy_int_map	
0x37	INT_MAP_1	0x00	step_cnt_int_map		step_det_int_map		gen_int3_int_map		gen_int2_int_map	
0x38	INT_MAP_2	0x00	acc_foc_int_map		orient_int_map		tilt_int_map		sig_mo_int_map	
0x39	INT_MAP_3	0x00	reserved						feat_eng_err_i...	
0x3A	IF_CONF_0	0x18	reserved	if_i2c_slv_addr						
0x3B	IF_CONF_1	0x38	reserved	if_i2c...	if_pad_drv			if_csb...	if_spi...	if_i3c...
...	-	-	reserved							
0x40	FIFO_CTRL	0x00	reserved						fifo_f...	fifo_rst
0x41	FIFO_CONF_0	0x0E	reserved			fifo_c...	fifo_a...	fifo_a...	fifo_a...	fifo_cfg
0x42	FIFO_CONF_1	0x06	reserved			fifo_s...	fifo_sensor_time		fifo_size	
0x43	FIFO_WM_0	0x00	fifo_watermark_level_7_0							
0x44	FIFO_WM_1	0x04	reserved					fifo_watermark_level_10_8		
...	-	-	reserved							
0x50	FEAT_ENG_CONF	0x01	reserved							feat_e...
0x51	FEAT_ENG_STATUS	0x00	reserved				feat_e...	host_g...	feat_e...	feat_e...
0x52	FEAT_ENG_GP_FLAGS	0x00	reserved				fifo_s...	foc_ru...	feat_init_stat	
0x53	FEAT_ENG_GPR_CONF	0x00	reserved	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...
0x54	FEAT_ENG_GPR_CTRL	0x00	reserved						unlock:.	update:.
0x55	FEAT_ENG_GPR_0	0x00	acc_fo...	orient_en	tilt_en	sig_mo...	step_en	gen_in...	gen_in...	gen_in...
0x56	FEAT_ENG_GPR_1	0x00	reserved		gen_int3_data_src		gen_int2_data_src		gen_int1_data_src	
0x57	FEAT_ENG_GPR_2	0x00	step_cnt_out_0							
0x58	FEAT_ENG_GPR_3	0x00	step_cnt_out_1							
0x59	FEAT_ENG_GPR_4	0x00	step_cnt_out_2							
0x5A	FEAT_ENG_GPR_5	0x00	gen_in...	gen_in...	gen_in...	activ_stat		orienta...	orientation_result	
...	-	-	reserved							
0x5E	FEATURE_DATA_ADDR	0x00	reserved	feature_data_addr						
0x5F	FEATURE_DATA_TX	0x00	feature_data							
...	-	-	reserved							
0x70	ACC_OFFSET_0	0x00	acc_doff_x_7_0							
0x71	ACC_OFFSET_1	0x00	reserved							acc_do...
0x72	ACC_OFFSET_2	0x00	acc_doff_y_7_0							
0x73	ACC_OFFSET_3	0x00	reserved							acc_do...

Table 41: Register map overview *(continued)*

Legend			Read-only		Read/Write		Write-only		Reserved	
Addr	**Name**	**Reset value**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x74	ACC_OFFSET_4	0x00	acc_doff_z_7_0							
0x75	ACC_OFFSET_5	0x00	reserved							acc_do...
0x76	ACC_SELF_TEST	0x00	reserved						self_t...	self_test
...	-	-	reserved							
0x7E	CMD	0x00	cmd							
...	-	-	reserved							

6.1.2 Register Map Details

Register (0x00) **CHIP_ID**

Description: The product chip_id. This register can be used to identify the product and perform a first simple communication test while reading out the chip id after boot up.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	1	1	0	0	0	0	1	0
Content	chip_id							

- `CHIP_ID.chip_id`: (bit offset: 0, bit width: 8, access: read-only) Chip ID: a constant number to identify the product. Following values can be read from the field chip_id:

Value	**Description**
0xC2	product identifier for BMA530

Use this link to go back to the overview table: `CHIP_ID`.

Register (0x02) **HEALTH_STATUS**

Description: This register contains internal health status information

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				sensor_health_status			

- reserved: write 0x0.
- `HEALTH_STATUS.sensor_health_status`: (bit offset: 0, bit width: 4, access: read-only) The value 0xF indicate a good internal health state.

Use this link to go back to the overview table: `HEALTH_STATUS`.

Register (0x04) **CMD_SUSPEND**

Description: Command register to activate suspend mode.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							suspend

- reserved: write 0x0.
- `CMD_SUSPEND.suspend`: (bit offset: 0, bit width: 1, access: read-write) Write ’1’ to activate suspend mode. The register content prior to entering this power mode will NOT be lost.
  Following values can be set to or read from the field suspend:

Value	**Description**
0b0 (0x0)	Suspend mode is disabled. Sensor in normal operation mode.
0b1 (0x1)	Suspend mode is enabled. Only Register CHIP_ID and this register are accessible.

Use this link to go back to the overview table: `CMD_SUSPEND`.

Register (0x10) **CONFIG_STATUS**

Description: Global error flags

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						acc_co...	feat_e...

- reserved: write 0x0.
- `CONFIG_STATUS.feat_eng_err`: (bit offset: 0, bit width: 1, access: read-write) Set by feature engine in case of feature engine error condition. Needs to be reseted by the host. For more details there are further status register in the feature engine section and inside the DMA region.
  Following values can be set to or read from the field feat_eng_err:

Value	Description
0b0 (0x0)	feature engine is okay
0b1 (0x1)	feature engine indicates error

- `CONFIG_STATUS.acc_conf_err`: (bit offset: 1, bit width: 1, access: read-only) This flag is set if the ACC configuration in ACC_CONF_0, ACC_CONF_1, and ACC_CONF_2 is an invalid combination.
  Following values can be read from the field acc_conf_err:

Value	Description
0b0 (0x0)	sensor configuration okay
0b1 (0x1)	sensor configuration invalid

Use this link to go back to the overview table: `CONFIG_STATUS`.

Register (0x11) **SENSOR_STATUS**

Description: Global status flags

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R/W	R/W
Reset Value	0	0	0	0	0	1	0	0
Content	reserved					sensor:.	tempera...	acc_da...

- reserved: write 0x0.
- `SENSOR_STATUS.acc_data_rdy`: (bit offset: 0, bit width: 1, access: read-write) Set when new ACC data is available. This flag can be cleared by writing ’1’ to it.
- `SENSOR_STATUS.temperature_rdy`: (bit offset: 1, bit width: 1, access: read-write) Set when new temperature data is available. This flag can be cleared by writing ’1’ to it.
- `SENSOR_STATUS.sensor_rdy`: (bit offset: 2, bit width: 1, access: read-only) Sensor is ready for operation.

Use this link to go back to the overview table: `SENSOR_STATUS`.

Register (0x12) **INT_STATUS_INT1_0**

Description: INT1 interrupt status register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...

- `INT_STATUS_INT1_0.acc_drdy_int_status`: (bit offset: 0, bit width: 1, access: read-write) Accelerometer data ready interrupt status.
- `INT_STATUS_INT1_0.fifo_wm_int_status`: (bit offset: 1, bit width: 1, access: read-write) FIFO watermark interrupt status.
- `INT_STATUS_INT1_0.fifo_full_int_status`: (bit offset: 2, bit width: 1, access: read-write) FIFO full interrupt status.
- `INT_STATUS_INT1_0.gen_int1_int_status`: (bit offset: 3, bit width: 1, access: read-write) Generic interrupt 1 interrupt status.
- `INT_STATUS_INT1_0.gen_int2_int_status`: (bit offset: 4, bit width: 1, access: read-write) Generic interrupt 2 interrupt status.
- `INT_STATUS_INT1_0.gen_int3_int_status`: (bit offset: 5, bit width: 1, access: read-write) Generic interrupt 3 interrupt status.
- `INT_STATUS_INT1_0.step_det_int_status`: (bit offset: 6, bit width: 1, access: read-write) Step detection interrupt status.
- `INT_STATUS_INT1_0.step_cnt_int_status`: (bit offset: 7, bit width: 1, access: read-write) Step counter watermark interrupt status.

Use this link to go back to the overview table: `INT_STATUS_INT1_0`.

Register (0x13) **INT_STATUS_INT1_1**

Description: INT1 interrupt status register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...

- reserved: write 0x0.
- `INT_STATUS_INT1_1.sig_mo_int_status`: (bit offset: 0, bit width: 1, access: read-write) Significant motion detection interrupt status.
- `INT_STATUS_INT1_1.tilt_int_status`: (bit offset: 1, bit width: 1, access: read-write) Tilt detection interrupt status.
- `INT_STATUS_INT1_1.orient_int_status`: (bit offset: 2, bit width: 1, access: read-write) Orientation detection status.
- `INT_STATUS_INT1_1.acc_foc_int_status`: (bit offset: 3, bit width: 1, access: read-write) Accelerometer FOC completion status.
- `INT_STATUS_INT1_1.feat_eng_err_int_status`: (bit offset: 4, bit width: 1, access: read-write) MCU error interrupt status.

Use this link to go back to the overview table: `INT_STATUS_INT1_1`.

Register (0x14) **INT_STATUS_INT2_0**

Description: INT2 interrupt status register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...

- `INT_STATUS_INT2_0.acc_drdy_int_status`: (bit offset: 0, bit width: 1, access: read-write) Accelerometer data ready interrupt status.
- `INT_STATUS_INT2_0.fifo_wm_int_status`: (bit offset: 1, bit width: 1, access: read-write) FIFO watermark interrupt status.
- `INT_STATUS_INT2_0.fifo_full_int_status`: (bit offset: 2, bit width: 1, access: read-write) FIFO full interrupt status.
- `INT_STATUS_INT2_0.gen_int1_int_status`: (bit offset: 3, bit width: 1, access: read-write) Generic interrupt 1 interrupt status.
- `INT_STATUS_INT2_0.gen_int2_int_status`: (bit offset: 4, bit width: 1, access: read-write) Generic interrupt 2 interrupt status.
- `INT_STATUS_INT2_0.gen_int3_int_status`: (bit offset: 5, bit width: 1, access: read-write) Generic interrupt 3 interrupt status.
- `INT_STATUS_INT2_0.step_det_int_status`: (bit offset: 6, bit width: 1, access: read-write) Step detection interrupt status.
- `INT_STATUS_INT2_0.step_cnt_int_status`: (bit offset: 7, bit width: 1, access: read-write) Step counter watermark interrupt status.

Use this link to go back to the overview table: `INT_STATUS_INT2_0`.

Register (0x15) **INT_STATUS_INT2_1**

Description: INT2 interrupt status register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...

- reserved: write 0x0.
- `INT_STATUS_INT2_1.sig_mo_int_status`: (bit offset: 0, bit width: 1, access: read-write) Significant motion detection interrupt status.
- `INT_STATUS_INT2_1.tilt_int_status`: (bit offset: 1, bit width: 1, access: read-write) Tilt detection interrupt status.
- `INT_STATUS_INT2_1.orient_int_status`: (bit offset: 2, bit width: 1, access: read-write) Orientation detection status.
- `INT_STATUS_INT2_1.acc_foc_int_status`: (bit offset: 3, bit width: 1, access: read-write) Accelerometer FOC completion status.
- `INT_STATUS_INT2_1.feat_eng_err_int_status`: (bit offset: 4, bit width: 1, access: read-write) MCU error interrupt status.

Use this link to go back to the overview table: `INT_STATUS_INT2_1`.

Register (0x16) **INT_STATUS_I3C_0**

Description: I3C interrupt status register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	step_c...	step_d...	gen_in...	gen_in...	gen_in...	fifo_f...	fifo_w...	acc_dr...

- `INT_STATUS_I3C_0.acc_drdy_int_status`: (bit offset: 0, bit width: 1, access: read-write) Accelerometer data ready interrupt status.
- `INT_STATUS_I3C_0.fifo_wm_int_status`: (bit offset: 1, bit width: 1, access: read-write) FIFO watermark interrupt status.
- `INT_STATUS_I3C_0.fifo_full_int_status`: (bit offset: 2, bit width: 1, access: read-write) FIFO full interrupt status.
- `INT_STATUS_I3C_0.gen_int1_int_status`: (bit offset: 3, bit width: 1, access: read-write) Generic interrupt 1 interrupt status.
- `INT_STATUS_I3C_0.gen_int2_int_status`: (bit offset: 4, bit width: 1, access: read-write) Generic interrupt 2 interrupt status.
- `INT_STATUS_I3C_0.gen_int3_int_status`: (bit offset: 5, bit width: 1, access: read-write) Generic interrupt 3 interrupt status.
- `INT_STATUS_I3C_0.step_det_int_status`: (bit offset: 6, bit width: 1, access: read-write) Step detection interrupt status.
- `INT_STATUS_I3C_0.step_cnt_int_status`: (bit offset: 7, bit width: 1, access: read-write) Step counter watermark interrupt status.

Use this link to go back to the overview table: `INT_STATUS_I3C_0`.

Register (0x17) **INT_STATUS_I3C_1**

Description: I3C interrupt status register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			feat_e...	acc_fo...	orient:.	tilt_i...	sig_mo...

- reserved: write 0x0.
- `INT_STATUS_I3C_1.sig_mo_int_status`: (bit offset: 0, bit width: 1, access: read-write) Significant motion detection interrupt status.
- `INT_STATUS_I3C_1.tilt_int_status`: (bit offset: 1, bit width: 1, access: read-write) Tilt detection interrupt status.
- `INT_STATUS_I3C_1.orient_int_status`: (bit offset: 2, bit width: 1, access: read-write) Orientation detection status.
- `INT_STATUS_I3C_1.acc_foc_int_status`: (bit offset: 3, bit width: 1, access: read-write) Accelerometer FOC completion status.
- `INT_STATUS_I3C_1.feat_eng_err_int_status`: (bit offset: 4, bit width: 1, access: read-write) MCU error interrupt status.

Use this link to go back to the overview table: `INT_STATUS_I3C_1`.

Register (0x18) **ACC_DATA_0**

Description: ACC data register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	acc_x_7_0							

- `ACC_DATA_0.acc_x_7_0`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for x-axis. (LSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_0`.

Register (0x19) **ACC_DATA_1**

Description: ACC data register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	1	0	0	0	0	0	0	0
Content	acc_x_15_8							

- `ACC_DATA_1.acc_x_15_8`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for x-axis. (MSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_1`.

Register (0x1A) **ACC_DATA_2**

Description: ACC data register 2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	acc_y_7_0							

- `ACC_DATA_2.acc_y_7_0`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for y-axis. (LSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_2`.

Register (0x1B) **ACC_DATA_3**

Description: ACC data register 3

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	1	0	0	0	0	0	0	0
Content	acc_y_15_8							

- `ACC_DATA_3.acc_y_15_8`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for y-axis. (MSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_3`.

Register (0x1C) **ACC_DATA_4**

Description: ACC data register 4

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	acc_z_7_0							

- `ACC_DATA_4.acc_z_7_0`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for z-axis. (LSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_4`.

Register (0x1D) **ACC_DATA_5**

Description: ACC data register 5

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	1	0	0	0	0	0	0	0
Content	acc_z_15_8							

- `ACC_DATA_5.acc_z_15_8`: (bit offset: 0, bit width: 8, access: read-only) Accelerometer data for z-axis. (MSB). The full 16bit range cover the selected g-range. (e.g. 8G-range: 1LSB = 16/65536=0.244 mg).

Use this link to go back to the overview table: `ACC_DATA_5`.

Register (0x1E) **TEMP_DATA**

Description: Temperature data register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	temp_data							

- `TEMP_DATA.temp_data`: (bit offset: 0, bit width: 8, access: read-only) Calculated temperature. Resolution: 1 K/LSB. The value 0 represents 23degC.

Use this link to go back to the overview table: `TEMP_DATA`.

Register (0x1F) **SENSOR_TIME_0**

Description: Sensor time register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	sensor_time_7_0							

- `SENSOR_TIME_0.sensor_time_7_0`: (bit offset: 0, bit width: 8, access: read-only) Sensor time in units 1 LSB = 312.5us.

Use this link to go back to the overview table: `SENSOR_TIME_0`.

Register (0x20) **SENSOR_TIME_1**

Description: Sensor time register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	sensor_time_15_8							

- `SENSOR_TIME_1.sensor_time_15_8`: (bit offset: 0, bit width: 8, access: read-only) Sensor time in units 1 LSB = 312.5us.

Use this link to go back to the overview table: `SENSOR_TIME_1`.

Register (0x21) **SENSOR_TIME_2**

Description: Sensor time register 2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	sensor_time_23_16							

- `SENSOR_TIME_2.sensor_time_23_16`: (bit offset: 0, bit width: 8, access: read-only) Sensor time in units 1 LSB = 312.5us.

Use this link to go back to the overview table: `SENSOR_TIME_2`.

Register (0x22) **FIFO_LEVEL_0**

Description: FIFO fill level register (LSB)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	fifo_fill_level_7_0							

- `FIFO_LEVEL_0.fifo_fill_level_7_0`: (bit offset: 0, bit width: 8, access: read-only) The fill level of the fifo only reflects the stored data. The frame header are not stored and not part of the FIFO fill level. To read out complete FIFO, the best way is to read as long as valid frames are read. LSB of the FIFO fill level. Should be read before the MSB register. .

Use this link to go back to the overview table: `FIFO_LEVEL_0`.

Register (0x23) **FIFO_LEVEL_1**

Description: FIFO fill level register (MSB)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved					fifo_fill_level_10_8		

- reserved: write 0x0.
- `FIFO_LEVEL_1.fifo_fill_level_10_8`: (bit offset: 0, bit width: 3, access: read-only) MSB of the FIFO fill level. Should be read after the LSB register.

Use this link to go back to the overview table: `FIFO_LEVEL_1`.

Register (0x24) **FIFO_DATA_OUT**

Description: FIFO data register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	1	0	0	0	0	0	0	0
Content	fifo_data_out							

- `FIFO_DATA_OUT.fifo_data_out`: (bit offset: 0, bit width: 8, access: read-only) Output of the FIFO. During burst reads on this address the address increment stops and the FIFO can be read out with help of the burst read. The type of data stored in the FIFO depends on configuration stored in FIFO_CONF_* registers.

Use this link to go back to the overview table: `FIFO_DATA_OUT`.

Register (0x30) **ACC_CONF_0**

Description: Accelerometer configuration register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1
Content	reserved				sensor_ctrl			

- reserved: write 0x0.
- `ACC_CONF_0.sensor_ctrl`: (bit offset: 0, bit width: 4, access: read-write) This bit enables/disables the accelerometer and the temperature sensor.
  Following values can be set to or read from the field sensor_ctrl:

Value	**Description**
0b0000 (0x0)	The accelerometer and the temperature sensor are disabled.
0b1111 (0xF)	The accelerometer and the temperature sensor are enabled.
0b1110 (0xE)	A wrong configuration was found: The accelerometer and the temperature sensor are disabled.

Use this link to go back to the overview table: `ACC_CONF_0`.

Register (0x31) **ACC_CONF_1**

Description: Accelerometer configuration register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	0	1	0	0	1	1	1
Content	power_...	acc_bwp			acc_odr			

- `ACC_CONF_1.acc_odr`: (bit offset: 0, bit width: 4, access: read-write) The ODR (Output Data Rate) in Hz. Not all settings are available in all power modes.
  Following values can be set to or read from the field acc_odr:

Value	**Description**
0b0000 (0x0)	1.5625 Hz. Only available in duty cycling mode (LPM).
0b0001 (0x1)	3.125 Hz. Only available in duty cycling mode (LPM).
0b0010 (0x2)	6.25 Hz.Only available in duty cycling mode (LPM).
0b0011 (0x3)	12.5 Hz.
0b0100 (0x4)	25 Hz.
0b0101 (0x5)	50 Hz.
0b0110 (0x6)	100 Hz.
0b0111 (0x7)	200 Hz.
0b1000 (0x8)	400 Hz.
0b1001 (0x9)	800 Hz. Only available in continuous mode (HPM).
0b1010 (0xA)	1.6 kHz. Only available in continuous mode (HPM).
0b1011 (0xB)	3.2 kHz. Only available in continuous mode (HPM).
0b1100 (0xC)	6.4 kHz. Only available in continuous mode (HPM).

- `ACC_CONF_1.acc_bwp`: (bit offset: 4, bit width: 3, access: read-write) Accelerometer bandwith parameter. This parameter determines the filter configuration. The different settings have a different impact depending on the setting of the power_mode bit. The name of the settings are therefore (HPM-setting)_(LPM-setting). (e.g. norm_avg4 means norm mode for HPM and avg4 for LPM).
  Following values can be set to or read from the field acc_bwp:

Value	**Description**
0b000 (0x0)	HPM –> OSR4 mode; LPM –> no averaging.
0b001 (0x1)	HPM –> OSR2 mode; LPM –> average 2 samples.
0b010 (0x2)	HPM –> normal mode; LPM –> average 4 samples.
0b011 (0x3)	HPM –> CIC mode; LPM –> average 8 samples.
0b100 (0x4)	HPM –> reserved; LPM –> average 16 samples.
0b101 (0x5)	HPM –> reserved; LPM –> average 32 samples.
0b110 (0x6)	HPM –> reserved; LPM –> average 64 samples.
0b111 (0x7)	HPM –> reserved; LPM –> reserved.

- `ACC_CONF_1.power_mode`: (bit offset: 7, bit width: 1, access: read-write) With this config bit, it is possible to set the basic measurement power mode. There are two possible settings:LPM (Low Power Mode) with duty cycling or HPM (High Performance Mode) with continous measurement. This setting has an influence on the signal path and the filter settings, too.
  Following values can be set to or read from the field power_mode:

Value	Description
0b0 (0x0)	LPM: Low power mode (Duty Cycling mode)
0b1 (0x1)	HPM: High Performance Mode (Continuous mode)

Use this link to go back to the overview table: `ACC_CONF_1`.

Register (0x32) **ACC_CONF_2**

Description: Accelerometer configuration register 2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	0
Content	acc_dr...	reserved		noise_...	acc_iir_ro		acc_range	

- reserved: write 0x0.
- `ACC_CONF_2.acc_range`: (bit offset: 0, bit width: 2, access: read-write) The measurement range of the accelerometer. This setting has influence on the scaling of the ACC_DATA registers.
  Following values can be set to or read from the field acc_range:

Value	**Description**
0b00 (0x0)	measurement range: +/-2g.
0b01 (0x1)	measurement range: +/-4g.
0b10 (0x2)	measurement range: +/-8g.
0b11 (0x3)	measurement range: +/-16g.

- `ACC_CONF_2.acc_iir_ro`: (bit offset: 2, bit width: 2, access: read-write) Select roll-off of IIR filter in continuous mode.
  Following values can be set to or read from the field acc_iir_ro:

Value	**Description**
0b00 (0x0)	reserved
0b01 (0x1)	-20dB roll-off
0b10 (0x2)	-40dB roll-off
0b11 (0x3)	-60dB roll-off

- `ACC_CONF_2.noise_mode`: (bit offset: 4, bit width: 1, access: read-write) Select the performance mode of the sensor. The choice is between high performance with lower noise or reduce the power consumption but with an increased noise level. The default is the high performance (lower noise). Changing this setting from default migh also influence the sensor behaviour like offset. This configuration should only be used in HPM. .
  Following values can be set to or read from the field noise_mode:

Value	**Description**
0b0 (0x0)	Default config. Lower noise level.
0b1 (0x1)	Lower power consumption. Higher noise level. This setting should only be used in HPM mode!

- `ACC_CONF_2.acc_drdy_int_auto_clear`: (bit offset: 7, bit width: 1, access: read-write) Configuration bit to enable/disable the auto clear mechanism of the data ready interrupt. If enabled, a clock like with freq=odr can be

enabled on the external interrupt pin.
Following values can be set to or read from the field acc_drdy_int_auto_clear:

Value	Description
0b0 (0x0)	The status flag of acc_drdy_int is not cleared automatically.
0b1 (0x1)	The status flag of acc_drdy_int is cleared automatically after approximately 1/(2*ODR).

Use this link to go back to the overview table: `ACC_CONF_2`.

Register (0x33) **TEMP_CONF**

Description: Temperature Sensor configuration register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved					temp_rate		

- reserved: write 0x0.
- `TEMP_CONF.temp_rate`: (bit offset: 0, bit width: 3, access: read-write) Select rate in Hz at which the temperature is sampled.
  Following values can be set to or read from the field temp_rate:

Value	**Description**
0b000 (0x0)	Sample temperature at 1.5625 Hz.
0b001 (0x1)	Sample temperature at 3.125 Hz.
0b010 (0x2)	Sample temperature at 6.25 Hz.
0b011 (0x3)	Sample temperature at 12.5 Hz.
0b100 (0x4)	Sample temperature at 25 Hz.
0b101 (0x5)	Sample temperature at 50 Hz.
0b110 (0x6)	Sample temperature at 100 Hz.
0b111 (0x7)	Sample temperature at 200 Hz.

Use this link to go back to the overview table: `TEMP_CONF`.

Register (0x34) **INT1_CONF**

Description: Configuration register for INT1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				lvl	od	mode	

- reserved: write 0x0.
- `INT1_CONF.mode`: (bit offset: 0, bit width: 2, access: read-write) Output enable for INT1 pin.
  Following values can be set to or read from the field mode:

Value	**Description**
0b00 (0x0)	Output disabled.
0b01 (0x1)	Latched (level triggered) interrupts.
0b10 (0x2)	Pulsed (edge triggered) interrupts with short pulses.
0b11 (0x3)	Pulsed (edge triggered) interrupts with long pulses.

- `INT1_CONF.od`: (bit offset: 2, bit width: 1, access: read-write) Configure behaviour of INT1 pin to open drain.
  Following values can be set to or read from the field od:

Value	**Description**
0b0 (0x0)	push-pull
0b1 (0x1)	open drain

- `INT1_CONF.lvl`: (bit offset: 3, bit width: 1, access: read-write) Configure level of INT1 pin.
  Following values can be set to or read from the field lvl:

Value	**Description**
0b0 (0x0)	active low
0b1 (0x1)	active high

Use this link to go back to the overview table: `INT1_CONF`.

Register (0x35) **INT2_CONF**

Description: Configuration register for INT2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				lvl	od	mode	

- reserved: write 0x0.
- `INT2_CONF.mode`: (bit offset: 0, bit width: 2, access: read-write) Mode for INT2 pin.
  Following values can be set to or read from the field mode:

Value	**Description**
0b00 (0x0)	Output disabled.
0b01 (0x1)	Latched (level triggered) interrupts.
0b10 (0x2)	Pulsed (edge triggered) interrupts with short pulses.
0b11 (0x3)	Pulsed (edge triggered) interrupts with long pulses.

- `INT2_CONF.od`: (bit offset: 2, bit width: 1, access: read-write) Configure behaviour of INT2 pin to open drain.
  Following values can be set to or read from the field od:

Value	**Description**
0b0 (0x0)	push-pull
0b1 (0x1)	open drain

- `INT2_CONF.lvl`: (bit offset: 3, bit width: 1, access: read-write) Configure level of INT2 pin.
  Following values can be set to or read from the field lvl:

Value	**Description**
0b0 (0x0)	active low
0b1 (0x1)	active high

Use this link to go back to the overview table: `INT2_CONF`.

Register (0x36) **INT_MAP_0**

Description: Interrupt mapping register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	gen_int1_int_map		fifo_full_int_map		fifo_wm_int_map		acc_drdy_int_map	

- `INT_MAP_0.acc_drdy_int_map`: (bit offset: 0, bit width: 2, access: read-write) Data ready interrupt mapping. Following values can be set to or read from the field acc_drdy_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_0.fifo_wm_int_map`: (bit offset: 2, bit width: 2, access: read-write) FIFO watermark interrupt mapping. Following values can be set to or read from the field fifo_wm_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_0.fifo_full_int_map`: (bit offset: 4, bit width: 2, access: read-write) FIFO full interrupt mapping. Following values can be set to or read from the field fifo_full_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_0.gen_int1_int_map`: (bit offset: 6, bit width: 2, access: read-write) Generic interrupt 1 interrupt mapping. Following values can be set to or read from the field gen_int1_int_map:

Value	Description
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

Use this link to go back to the overview table: `INT_MAP_0`.

Register (0x37) **INT_MAP_1**

Description: Interrupt mapping register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	step_cnt_int_map		step_det_int_map		gen_int3_int_map		gen_int2_int_map	

- `INT_MAP_1.gen_int2_int_map`: (bit offset: 0, bit width: 2, access: read-write) Generic interrupt 2 interrupt mapping. Following values can be set to or read from the field gen_int2_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_1.gen_int3_int_map`: (bit offset: 2, bit width: 2, access: read-write) Generic interrupt 3 interrupt mapping. Following values can be set to or read from the field gen_int3_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_1.step_det_int_map`: (bit offset: 4, bit width: 2, access: read-write) Step detection interrupt mapping. Following values can be set to or read from the field step_det_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_1.step_cnt_int_map`: (bit offset: 6, bit width: 2, access: read-write) Step counter watermark interrupt mapping.
  Following values can be set to or read from the field step_cnt_int_map:

Value	Description
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

Use this link to go back to the overview table: `INT_MAP_1`.

Register (0x38) **INT_MAP_2**

Description: Interrupt mapping register 2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_foc_int_map		orient_int_map		tilt_int_map		sig_mo_int_map	

- `INT_MAP_2.sig_mo_int_map`: (bit offset: 0, bit width: 2, access: read-write) Significant motion detection interrupt mapping.
  Following values can be set to or read from the field sig_mo_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_2.tilt_int_map`: (bit offset: 2, bit width: 2, access: read-write) Tilt detection interrupt mapping.
  Following values can be set to or read from the field tilt_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_2.orient_int_map`: (bit offset: 4, bit width: 2, access: read-write) Orientation detection interrupt mapping.
  Following values can be set to or read from the field orient_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

- `INT_MAP_2.acc_foc_int_map`: (bit offset: 6, bit width: 2, access: read-write) Accelerometer FOC completion interrupt mapping.
  Following values can be set to or read from the field acc_foc_int_map:

Value	Description
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

Use this link to go back to the overview table: `INT_MAP_2`.

Register (0x39) **INT_MAP_3**

Description: Interrupt mapping register 3

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						feat_eng_err_i...	

- reserved: write 0x0.
- `INT_MAP_3.feat_eng_err_int_map`: (bit offset: 0, bit width: 2, access: read-write) MCU error interrupt mapping. Following values can be set to or read from the field feat_eng_err_int_map:

Value	**Description**
0b00 (0x0)	Interrupt is not mapped to any destination.
0b01 (0x1)	Interrupt is mapped to INT1 pin.
0b10 (0x2)	Interrupt is mapped to INT2 pin.
0b11 (0x3)	Interrupt is mapped to I3C in-band interrupts.

Use this link to go back to the overview table: `INT_MAP_3`.

Register (0x3A) **IF_CONF_0**

Description: Serial interface settings

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	1	1	0	0	0
Content	reserved	if_i2c_slv_addr						

- reserved: write 0x0.
- `IF_CONF_0.if_i2c_slv_addr`: (bit offset: 0, bit width: 7, access: read-only) I2C slave address of this device. Following values can be read from the field if_i2c_slv_addr:

Value	**Description**
0x18	the default i2c slave address of this device

Use this link to go back to the overview table: `IF_CONF_0`.

Register (0x3B) **IF_CONF_1**

Description: Serial interface settings

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	1	1	0	0	0
Content	reserved	if_i2c...	if_pad_drv			if_csb...	if_spi...	if_i3c...

- reserved: write 0x0.
- `IF_CONF_1.if_i3c_cfg`: (bit offset: 0, bit width: 1, access: read-write) Configuration of I3C mode.
  Following values can be set to or read from the field if_i3c_cfg:

Value	**Description**
0b0 (0x0)	The I3C mode is disabled.
0b1 (0x1)	The I3C mode is enabled.

- `IF_CONF_1.if_spi3_cfg`: (bit offset: 1, bit width: 1, access: read-write) Configuration of SPI3 mode(SPI 3 wire protocol).
  Following values can be set to or read from the field if_spi3_cfg:

Value	**Description**
0b0 (0x0)	The SPI3 mode is disabled.
0b1 (0x1)	The SPI3 mode is enabled.

- `IF_CONF_1.if_csb_pullup`: (bit offset: 2, bit width: 1, access: read-write) Configuration of CSB pullup in SPI mode.
  Following values can be set to or read from the field if_csb_pullup:

Value	**Description**
0b0 (0x0)	The pullup is disabled.
0b1 (0x1)	The pullup is enabled.

- `IF_CONF_1.if_pad_drv`: (bit offset: 3, bit width: 3, access: read-write) Pad drive strength in I2C mode.
- `IF_CONF_1.if_i2c_drv_sel`: (bit offset: 6, bit width: 1, access: read-write) select drive strength in I2C mode.
  Following values can be set to or read from the field if_i2c_drv_sel:

Value	**Description**
0b0 (0x0)	use maximum pad drive strength
0b1 (0x1)	use drive strength settings of if_pad_drv

Use this link to go back to the overview table: `IF_CONF_1`.

Register (0x40) **FIFO_CTRL**

Description: FIFO control register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	W	W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						fifo_f...	fifo_rst

- reserved: write 0x0.
- `FIFO_CTRL.fifo_rst`: (bit offset: 0, bit width: 1, access: write-only) FIFO reset trigger. Writing '1' to this field synchronously resets the FIFO.
- `FIFO_CTRL.fifo_frame_sync`: (bit offset: 1, bit width: 1, access: write-only) FIFO frame synchronization trigger. Writing '1' to this field tells the FIFO that another frame is about to be written to FIFO_DATA_IN.

Use this link to go back to the overview table: `FIFO_CTRL`.

Register (0x41) **FIFO_CONF_0**

Description: FIFO configuration register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	0
Content	reserved			fifo_c...	fifo_a...	fifo_a...	fifo_a...	fifo_cfg

- reserved: write 0x0.
- `FIFO_CONF_0.fifo_cfg`: (bit offset: 0, bit width: 1, access: read-write) Enable bit for the FIFO. Cannot be set to 1 if fifo_size equals 0.
  Following values can be set to or read from the field fifo_cfg:

Value	**Description**
0b0 (0x0)	The FIFO is disabled.
0b1 (0x1)	The FIFO is enabled.

- `FIFO_CONF_0.fifo_acc_x`: (bit offset: 1, bit width: 1, access: read-write) Configuration bit to enable the storage of the x-axis acceleration data in the FIFO.
  Following values can be set to or read from the field fifo_acc_x:

Value	**Description**
0b0 (0x0)	The FIFO x-axis acceleration channel is disabled.
0b1 (0x1)	The FIFO x-axis acceleration channel is disabled.

- `FIFO_CONF_0.fifo_acc_y`: (bit offset: 2, bit width: 1, access: read-write) Configuration bit to enable the storage of the y-axis acceleration data in the FIFO.
  Following values can be set to or read from the field fifo_acc_y:

Value	**Description**
0b0 (0x0)	The FIFO y-axis acceleration channel is disabled.
0b1 (0x1)	The FIFO y-axis acceleration channel is disabled.

- `FIFO_CONF_0.fifo_acc_z`: (bit offset: 3, bit width: 1, access: read-write) Configuration bit to enable the storage of the z-axis acceleration data in the FIFO.
  Following values can be set to or read from the field fifo_acc_z:

Value	**Description**
0b0 (0x0)	The FIFO z-axis acceleration channel is disabled.
0b1 (0x1)	The FIFO z-axis acceleration channel is disabled.

- `FIFO_CONF_0.fifo_compression`: (bit offset: 4, bit width: 1, access: read-write) Enable bit for FIFO data compression.
  Following values can be set to or read from the field fifo_compression:

Value	Description
0b0 (0x0)	compression disbaled. full 16bit acceleration data.
0b1 (0x1)	compression enabled. 8bit compressed acceleration data.

Use this link to go back to the overview table: `FIFO_CONF_0`.

Register (0x42) **FIFO_CONF_1**

Description: FIFO configuration register 1

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	1	0
Content	reserved			fifo_s...	fifo_sensor_time		fifo_size	

- reserved: write 0x0.
- `FIFO_CONF_1.fifo_size`: (bit offset: 0, bit width: 2, access: read-write) FIFO size. Since FIFO and feature engine share a common RAM, the size for the FIFO share has to be adjusted. Cannot be changed if locked by the feature engine. In Order to change this value, first disable the feature engine. If the feature engine is turned on again, a minimum share migh be needed and this setting might be changed by the feature engine.
  Following values can be set to or read from the field fifo_size:

Value	Description
0b00 (0x0)	The FIFO has a size of 0 bytes. The feature engine ('feat_eng') has a RAM size of 1024 bytes. This setting forces fifo_en to 0.
0b01 (0x1)	The FIFO has a size of 256 bytes. The feature engine ('feat_eng') has a RAM size of 768 bytes.
0b10 (0x2)	The FIFO has a size of 512 bytes. The feature engine ('feat_eng') has a RAM size of 512 bytes.
0b11 (0x3)	The FIFO has a size of 1024 bytes. The feature engine ('feat_eng') has a RAM size of 0 bytes. This setting forces feature engine ('feat_eng')_en to 0.

- `FIFO_CONF_1.fifo_sensor_time`: (bit offset: 2, bit width: 2, access: read-write) FIFO sensor time configuration.
  Following values can be set to or read from the field fifo_sensor_time:

Value	Description
0b00 (0x0)	The FIFO does not transmit the sensor time.
0b01 (0x1)	The FIFO sends a dedicated sensor time frame when the FIFO runs empty during a read burst.
0b10 (0x2)	The FIFO has appends the sensor time to each frame.

- `FIFO_CONF_1.fifo_stop_on_full`: (bit offset: 4, bit width: 1, access: read-write) If set, the FIFO stops storing new data if it is full. Otherwise the oldest frame is dropped in order to make room for a new frame.
  Following values can be set to or read from the field fifo_stop_on_full:

Value	Description
0b0 (0x0)	feature disbaled.The FIFO will be filled continously with new data, old data will be dropped.
0b1 (0x1)	feature enabled. The FIFO will stop, when it is full.

Use this link to go back to the overview table: FIFO_CONF_1.

Register (0x43) **FIFO_WM_0**

Description: FIFO watermark level register (LSB)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	fifo_watermark_level_7_0							

- `FIFO_WM_0.fifo_watermark_level_7_0`: (bit offset: 0, bit width: 8, access: read-write) LSB of the FIFO watermark level.

Use this link to go back to the overview table: `FIFO_WM_0`.

Register (0x44) **FIFO_WM_1**

Description: FIFO watermark level register (MSB)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	0	0
Content	reserved					fifo_watermark_level_10_8		

- reserved: write 0x0.
- `FIFO_WM_1.fifo_watermark_level_10_8`: (bit offset: 0, bit width: 3, access: read-write) LSB of the FIFO watermark level.

Use this link to go back to the overview table: `FIFO_WM_1`.

Register (0x50) **FEAT_ENG_CONF**

Description: feature engine ('feat_eng') configuration register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	1
Content	reserved							feat_e...

- reserved: write 0x0.
- `FEAT_ENG_CONF.feat_eng_ctrl`: (bit offset: 0, bit width: 1, access: read-write) An enable/disable switch for the feature engine. The feature engine is internally reseted, once the engine is disabled and the enabled again.
  Following values can be set to or read from the field feat_eng_ctrl:

Value	Description
0b0 (0x0)	the feature engine is disabled (and reset)
0b1 (0x1)	the feature engine is enabled.

Use this link to go back to the overview table: `FEAT_ENG_CONF`.

Register (0x51) **FEAT_ENG_STATUS**

Description: feature engine ('feat_eng') status register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				feat_e...	host_g...	feat_e...	feat_e...

- reserved: write 0x0.
- `FEAT_ENG_STATUS.feat_eng_halted`: (bit offset: 0, bit width: 1, access: read-only) When this field equals 1, the feature engine is currently halted. This means that the "halt" instruction has been executed and that the processor waits for a wakeup trigger.
- `FEAT_ENG_STATUS.feat_eng_running`: (bit offset: 1, bit width: 1, access: read-only) When this field equals 1, the feature engine is currently executing code.
- `FEAT_ENG_STATUS.host_gpr_update_pending`: (bit offset: 2, bit width: 1, access: read-only) This field reads 1'b1 as long as an update of the host-owned GPRs is pending. .
- `FEAT_ENG_STATUS.feat_eng_gpr_update_pending`: (bit offset: 3, bit width: 1, access: read-only) This field reads 1'b1 as long as an update of the feature engine-owned GPRs is pending. .

Use this link to go back to the overview table: `FEAT_ENG_STATUS`.

Register (0x52) **FEAT_ENG_GP_FLAGS**

Description: feature engine ('feat_eng') general purpose flags

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				fifo_s...	foc_ru...	feat_init_stat	

- reserved: write 0x0.
- `FEAT_ENG_GP_FLAGS.feat_init_stat`: (bit offset: 0, bit width: 2, access: read-only) Feature engine initialization status .
  Following values can be read from the field feat_init_stat:

Value	**Description**
0b00 (0x0)	Feature engine is not initialized
0b01 (0x1)	Feature engine is initialized
0b10 (0x2)	Reserved
0b11 (0x3)	Reserved

- `FEAT_ENG_GP_FLAGS.foc_running`: (bit offset: 2, bit width: 1, access: read-only) Bit is set to '1' if fast-offset compensation feature is being executed. Bit is cleared to '0' at the end of feature compensation. User should not change the accelerometer configuration while the feature is running. .
- `FEAT_ENG_GP_FLAGS.fifo_size_changed`: (bit offset: 3, bit width: 1, access: read-only) Bit is set when FIFO size is changed by feature engine. Bit is cleared, when default FIFO size (512bytes) is set .

Use this link to go back to the overview table: `FEAT_ENG_GP_FLAGS`.

Register (0x53) **FEAT_ENG_GPR_CONF**

Description: feature engine ('feat_eng') general purpose register configuration register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...	feat_e...

- reserved: write 0x0.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_0_dir: (bit offset: 0, bit width: 1, access: read-write) host Direction for GP register 0 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_1_dir: (bit offset: 1, bit width: 1, access: read-write) host Direction for GP register 1 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_2_dir: (bit offset: 2, bit width: 1, access: read-write) host Direction for GP register 2 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_3_dir: (bit offset: 3, bit width: 1, access: read-write) host direction for GP register 3 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_4_dir: (bit offset: 4, bit width: 1, access: read-write) host direction for GP register 4 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_5_dir: (bit offset: 5, bit width: 1, access: read-write) host direction for GP register 5 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.
- FEAT_ENG_GPR_CONF.feat_eng_gpr_6_dir: (bit offset: 6, bit width: 1, access: read-write) host direction for GP register 6 ('0': feature engine has write access, '1': host has write access). This field is only writeable by the feature engine.

Use this link to go back to the overview table: FEAT_ENG_GPR_CONF.

Register (0x54) **FEAT_ENG_GPR_CTRL**

Description: feature engine ('feat_eng') general purpose register control register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	W	W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						unlock:.	update:.

- reserved: write 0x0.
- FEAT_ENG_GPR_CTRL.update_gprs: (bit offset: 0, bit width: 1, access: write-only) If the host writes 1'b1 to this field, it requests that the host-owned first stage registers are copied to the host-owned second stage registers. If the feature engine writes 1'b1 to this field, it requests that the feature engine-owned first stage registers are copied to the feature engine-owned second stage registers. .
- FEAT_ENG_GPR_CTRL.unlock_gprs: (bit offset: 1, bit width: 1, access: write-only) If the host writes 1'b1 to this field, it releases the lock of the feature engine-owned GPRs and thus allows for an update of the feature engine-owned second stage registers. If the feature engine writes 1'b1 to this field, it releases the lock of the host-owned GPRs and thus allows for an update of the host-owned second stage registers. .

Use this link to go back to the overview table: FEAT_ENG_GPR_CTRL.

Register (0x55) **FEAT_ENG_GPR_0**

Description: feature engine ('feat_eng') general purpose register 0

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_fo...	orient_en	tilt_en	sig_mo...	step_en	gen_in...	gen_in...	gen_in...

- `FEAT_ENG_GPR_0.gen_int1_en`: (bit offset: 0, bit width: 1, access: read-write) Enables generic interrupt 1 feature.
- `FEAT_ENG_GPR_0.gen_int2_en`: (bit offset: 1, bit width: 1, access: read-write) Enables generic interrupt 2 feature.
- `FEAT_ENG_GPR_0.gen_int3_en`: (bit offset: 2, bit width: 1, access: read-write) Enables generic interrupt 3 feature.
- `FEAT_ENG_GPR_0.step_en`: (bit offset: 3, bit width: 1, access: read-write) Enables step counter and/or step detection features.
- `FEAT_ENG_GPR_0.sig_mo_en`: (bit offset: 4, bit width: 1, access: read-write) Enables significant motion detection feature.
- `FEAT_ENG_GPR_0.tilt_en`: (bit offset: 5, bit width: 1, access: read-write) Enables tilt detection feature.
- `FEAT_ENG_GPR_0.orient_en`: (bit offset: 6, bit width: 1, access: read-write) Enables orientation detection feature.
- `FEAT_ENG_GPR_0.acc_foc_en`: (bit offset: 7, bit width: 1, access: read-write) Enables accelerometer fast-offset compensation.

Use this link to go back to the overview table: `FEAT_ENG_GPR_0`.

Register (0x56) **FEAT_ENG_GPR_1**

Description: feature engine ('feat_eng') general purpose register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		gen_int3_data_src		gen_int2_data_src		gen_int1_data_src	

- reserved: write 0x0.
- `FEAT_ENG_GPR_1.gen_int1_data_src`: (bit offset: 0, bit width: 2, access: read-write) Data source selection for gen_int1 feature.
  Following values can be set to or read from the field gen_int1_data_src:

Value	**Description**
0b00 (0x0)	Uses 50Hz filter data
0b01 (0x1)	Uses 200Hz filter data
0b10 (0x2)	Uses user filter data
0b11 (0x3)	Uses 50Hz filter data. Same as data_src_1

- `FEAT_ENG_GPR_1.gen_int2_data_src`: (bit offset: 2, bit width: 2, access: read-write) Data source selection for gen_int2 feature.
  Following values can be set to or read from the field gen_int2_data_src:

Value	**Description**
0b00 (0x0)	Uses 50Hz filter data
0b01 (0x1)	Uses 200Hz filter data
0b10 (0x2)	Uses user filter data
0b11 (0x3)	Uses 50Hz filter data. Same as data_src_1

- `FEAT_ENG_GPR_1.gen_int3_data_src`: (bit offset: 4, bit width: 2, access: read-write) Data source selection for gen_int3 feature.
  Following values can be set to or read from the field gen_int3_data_src:

Value	**Description**
0b00 (0x0)	Uses 50Hz filter data
0b01 (0x1)	Uses 200Hz filter data
0b10 (0x2)	Uses user filter data
0b11 (0x3)	Uses 50Hz filter data. Same as data_src_1

Use this link to go back to the overview table: `FEAT_ENG_GPR_1`.

Register (0x57) **FEAT_ENG_GPR_2**

Description: feature engine ('feat_eng') general purpose register 2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	step_cnt_out_0							

- `FEAT_ENG_GPR_2.step_cnt_out_0`: (bit offset: 0, bit width: 8, access: read-only) Step counter value byte-0.

Use this link to go back to the overview table: `FEAT_ENG_GPR_2`.

Register (0x58) **FEAT_ENG_GPR_3**

Description: feature engine ('feat_eng') general purpose register 3

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	step_cnt_out_1							

- `FEAT_ENG_GPR_3.step_cnt_out_1`: (bit offset: 0, bit width: 8, access: read-only) Step counter value byte-1.

Use this link to go back to the overview table: `FEAT_ENG_GPR_3`.

Register (0x59) **FEAT_ENG_GPR_4**

Description: feature engine ('feat_eng') general purpose register 4

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	step_cnt_out_2							

- `FEAT_ENG_GPR_4.step_cnt_out_2`: (bit offset: 0, bit width: 8, access: read-only) Step counter value byte-2.

Use this link to go back to the overview table: `FEAT_ENG_GPR_4`.

Register (0x5A) **FEAT_ENG_GPR_5**

Description: feature engine ('feat_eng') general purpose register 5

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	gen_in...	gen_in...	gen_in...	activ_stat		orienta...	orientation_result	

- `FEAT_ENG_GPR_5.orientation_result`: (bit offset: 0, bit width: 2, access: read-only) Output of orientation detection feature. Device orientation can be either portrait or landscape. Value after device initialization is 0b00 i.e. portrait up.
  Following values can be read from the field orientation_result:

Value	**Description**
0b00 (0x0)	Portrait up orientation
0b01 (0x1)	Landscape left orientation
0b10 (0x2)	Portrait down orientation
0b11 (0x3)	Landscape right orientation

- `FEAT_ENG_GPR_5.orientation_face_up_down`: (bit offset: 2, bit width: 1, access: read-only) Output of orientation detection feature. Output is only valid if "ud_en" is enabled. Device orientation can be either face up or face down. Value after device initialization is 0b0 i.e. face up.
  Following values can be read from the field orientation_face_up_down:

Value	**Description**
0b0 (0x0)	Face up orientation
0b1 (0x1)	Face down orientation

- `FEAT_ENG_GPR_5.activ_stat`: (bit offset: 3, bit width: 2, access: read-only) Status of user activity reported by step counter.
  Following values can be read from the field activ_stat:

Value	**Description**
0b00 (0x0)	After device reset or while step counter is disabled
0b01 (0x1)	User is stationary
0b10 (0x2)	User is walking
0b11 (0x3)	User is running

- `FEAT_ENG_GPR_5.gen_int1_stat`: (bit offset: 5, bit width: 1, access: read-only) Status of generic interrupt 1 motion detection.
- `FEAT_ENG_GPR_5.gen_int2_stat`: (bit offset: 6, bit width: 1, access: read-only) Status of generic interrupt 2 motion detection.

- `FEAT_ENG_GPR_5.gen_int3_stat`: (bit offset: 7, bit width: 1, access: read-only) Status of generic interrupt 3 motion detection.

Use this link to go back to the overview table: `FEAT_ENG_GPR_5`.

Register (0x5E) **FEATURE_DATA_ADDR**

Description: feature engine ('feat_eng') feature data start address

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved	feature_data_addr						

- reserved: write 0x0.
- `FEATURE_DATA_ADDR.feature_data_addr`: (bit offset: 0, bit width: 7, access: read-write) Feature data address. For the address values see the extended memory map.
  Following values can be set to or read from the field feature_data_addr:

Value	Description
0x2	address(0x2) to access register: FEAT_CONF_ERR
0x3	address(0x3) to access register: GENERAL_SETTINGS_0
0x4	address(0x4) to access register: GENERIC_INTERRUPT1_1
0x5	address(0x5) to access register: GENERIC_INTERRUPT1_2
0x6	address(0x6) to access register: GENERIC_INTERRUPT1_3
0x7	address(0x7) to access register: GENERIC_INTERRUPT1_4
0x8	address(0x8) to access register: GENERIC_INTERRUPT1_5
0x9	address(0x9) to access register: GENERIC_INTERRUPT1_6
0xA	address(0xa) to access register: GENERIC_INTERRUPT1_7
0xB	address(0xb) to access register: GENERIC_INTERRUPT2_1
0xC	address(0xc) to access register: GENERIC_INTERRUPT2_2
0xD	address(0xd) to access register: GENERIC_INTERRUPT2_3
0xE	address(0xe) to access register: GENERIC_INTERRUPT2_4
0xF	address(0xf) to access register: GENERIC_INTERRUPT2_5
0x10	address(0x10) to access register: GENERIC_INTERRUPT2_6
0x11	address(0x11) to access register: GENERIC_INTERRUPT2_7
0x12	address(0x12) to access register: GENERIC_INTERRUPT3_1
0x13	address(0x13) to access register: GENERIC_INTERRUPT3_2
0x14	address(0x14) to access register: GENERIC_INTERRUPT3_3
0x15	address(0x15) to access register: GENERIC_INTERRUPT3_4
0x16	address(0x16) to access register: GENERIC_INTERRUPT3_5
0x17	address(0x17) to access register: GENERIC_INTERRUPT3_6
0x18	address(0x18) to access register: GENERIC_INTERRUPT3_7
0x19	address(0x19) to access register: STEP_COUNTER
0x2B	address(0x2b) to access register: SIG_MOTION
0x2E	address(0x2e) to access register: TILT_1
0x2F	address(0x2f) to access register: TILT_2
0x30	address(0x30) to access register: ORIENTATION_1
0x31	address(0x31) to access register: ORIENTATION_2
0x32	address(0x32) to access register: FOC_0
0x33	address(0x33) to access register: FOC_1
0x34	address(0x34) to access register: FOC_2
0x35	address(0x35) to access register: FOC_3

Use this link to go back to the overview table: `FEATURE_DATA_ADDR`.

Register (0x5F) **FEATURE_DATA_TX**

Description: feature engine ('feat_eng') feature data

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	feature_data							

- `FEATURE_DATA_TX.feature_data`: (bit offset: 0, bit width: 8, access: read-write) The data port associated with feature_data_addr. During burst read/write operations on this address the address increment stops and the burst operation can be used to read/write multiple feature_data words. See the extendend memory map for details.

Use this link to go back to the overview table: `FEATURE_DATA_TX`.

Register (0x70) **ACC_OFFSET_0**

Description: Offset compensation value (x-axis) LSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_doff_x_7_0							

- `ACC_OFFSET_0.acc_doff_x_7_0`: (bit offset: 0, bit width: 8, access: read-write) Accelerometer compensation value for x-axis. 9 bit signed, resolution is 0.98 mg/LSB. Range is [-0.25 g .. 0.25 g]. The resolution of the compensation value is independent of the range setting. To disable the offset compensation, a value of 0x0 has to be written to this field.The compensation offset values are not persistent and must be written each time after power-up or reset of the device.

Use this link to go back to the overview table: `ACC_OFFSET_0`.

Register (0x71) **ACC_OFFSET_1**

Description: Offset compensation value (x-axis) MSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							acc_do...

- reserved: write 0x0.
- `ACC_OFFSET_1.acc_doff_x_8`: (bit offset: 0, bit width: 1, access: read-write) Highest bit of the acc_doff_x field. See details at description of previous register field acc_doff_x_7_0.

Use this link to go back to the overview table: `ACC_OFFSET_1`.

Register (0x72) **ACC_OFFSET_2**

Description: Offset compensation value (y-axis) LSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_doff_y_7_0							

- `ACC_OFFSET_2.acc_doff_y_7_0`: (bit offset: 0, bit width: 8, access: read-write) Accelerometer compensation value for y-axis. 9 bit signed, resolution is 0.98 mg/LSB. Range is [-0.25 g .. 0.25 g]. The resolution of the compensation value is independent of the range setting. To disable the offset compensation, a value of 0x0 has to be written to this field.The compensation offset values are not persistent and must be written each time after power-up or reset of the device.

Use this link to go back to the overview table: `ACC_OFFSET_2`.

Register (0x73) **ACC_OFFSET_3**

Description: Offset compensation value (y-axis) MSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							acc_do...

- reserved: write 0x0.
- `ACC_OFFSET_3.acc_doff_y_8`: (bit offset: 0, bit width: 1, access: read-write) Highest bit of the acc_doff_y field. See details at description of previous register field acc_doff_y_7_0.

Use this link to go back to the overview table: `ACC_OFFSET_3`.

Register (0x74) **ACC_OFFSET_4**

Description: Offset compensation value (z-axis) LSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_doff_z_7_0							

- `ACC_OFFSET_4.acc_doff_z_7_0`: (bit offset: 0, bit width: 8, access: read-write) Accelerometer compensation value for z-axis. 9 bit signed, resolution is 0.98 mg/LSB. Range is [-0.25 g .. 0.25 g]. The resolution of the compensation value is independent of the range setting. To disable the offset compensation, a value of 0x0 has to be written to this field.The compensation offset values are not persistent and must be written each time after power-up or reset of the device.

Use this link to go back to the overview table: `ACC_OFFSET_4`.

Register (0x75) **ACC_OFFSET_5**

Description: Offset compensation value (z-axis) MSB

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							acc_do...

- reserved: write 0x0.
- `ACC_OFFSET_5.acc_doff_z_8`: (bit offset: 0, bit width: 1, access: read-write) Highest bit of the acc_doff_z field. See details at description of previous register field acc_doff_z_7_0.

Use this link to go back to the overview table: `ACC_OFFSET_5`.

Register (0x76) **ACC_SELF_TEST**

Description: Select NORMAL/SELF_TEST mode and test data. If you write to this register, the ACC data path is reset.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R	R	R	R	R	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						self_t...	self_test

- reserved: write 0x0.
- `ACC_SELF_TEST.self_test`: (bit offset: 0, bit width: 1, access: read-write) Enable flag for the self test mode. Following values can be set to or read from the field self_test:

Value	**Description**
0b0 (0x0)	normal operation mode
0b1 (0x1)	built-in test excitation mode

- `ACC_SELF_TEST.self_test_sign`: (bit offset: 1, bit width: 1, access: read-write) Select sign of self test excitation. Following values can be set to or read from the field self_test_sign:

Value	**Description**
0b0 (0x0)	negative
0b1 (0x1)	positive

Use this link to go back to the overview table: `ACC_SELF_TEST`.

Register (0x7E) **CMD**

Description: Command Register

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	W	W	W	W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0
Content	cmd							

- `CMD.cmd`: (bit offset: 0, bit width: 8, access: write-only) Available commands (Note: Register will always read as 0x00):.
  Following values can be set to the field cmd:

Value	Description
0x0	reserved. No command.
0xB6	Triggers a reset, all user configuration settings are overwritten with their default state. If this register is set using I2C, an ACK will NOT be transmitted to the host

Use this link to go back to the overview table: `CMD`.

6.2 Extended Register Map Description

The extended configuration and input/output of the feature engine has to be done through the feature engine data interface. The data can be read from or written through `FEATURE_DATA_TX.feature_data` to an address in the extended register map configured in `FEATURE_DATA_ADDR.feature_data_addr` by a data exchange transaction. For more details on how to access the extended register map, please refer to the example no3 in the quick start guide section chapter 3.

6.2.1 Extended Register Map Overview

The Table 42 provides an overview of the extended register map of the device.

Table 42: Extended register map overview

Legend			Read-only				Read/Write				Write-only				Reserved			
Addr	**Name**	**Reset value**	**bit15**	**bit14**	**bit13**	**bit12**	**bit11**	**bit10**	**bit9**	**bit8**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
...	-	-	reserved															
0x02	FEAT_CONF_ERR	0x0000	reserved								acc_fo...	orient:.	tilt_c...	sig_mo...	step_c...	gen_in...	gen_in...	gen_in...
0x03	GENERAL_SETTINGS_0	0x0000	reserved									feat_z...	feat_y...	feat_x...	feat_axis_ex			android...
0x04	GENERIC_INTERRUPT1_1	0xE00F	axis_sel			comb_sel	slope_thres											
0x05	GENERIC_INTERRUPT1_2	0x0C03	reserved			acc_ref_up		criteri...	hysteresis									
0x06	GENERIC_INTERRUPT1_3	0x600A	wait_time			duration												
0x07	GENERIC_INTERRUPT1_4	0x0040	reserved			quiet_time												
0x08	GENERIC_INTERRUPT1_5	0x0000	ref_acc_x															
0x09	GENERIC_INTERRUPT1_6	0x0000	ref_acc_y															
0x0A	GENERIC_INTERRUPT1_7	0x0800	ref_acc_z															
0x0B	GENERIC_INTERRUPT2_1	0xF00F	axis_sel			comb_sel	slope_thres											
0x0C	GENERIC_INTERRUPT2_2	0x0803	reserved			acc_ref_up		criteri...	hysteresis									
0x0D	GENERIC_INTERRUPT2_3	0x600A	wait_time			duration												
0x0E	GENERIC_INTERRUPT2_4	0x0040	reserved			quiet_time												
0x0F	GENERIC_INTERRUPT2_5	0x0000	ref_acc_x															
0x10	GENERIC_INTERRUPT2_6	0x0000	ref_acc_y															
0x11	GENERIC_INTERRUPT2_7	0x0800	ref_acc_z															
0x12	GENERIC_INTERRUPT3_1	0xF082	axis_sel			comb_sel	slope_thres											
0x13	GENERIC_INTERRUPT3_2	0x1008	reserved			acc_ref_up		criteri...	hysteresis									
0x14	GENERIC_INTERRUPT3_3	0x4003	wait_time			duration												
0x15	GENERIC_INTERRUPT3_4	0x0040	reserved			quiet_time												
0x16	GENERIC_INTERRUPT3_5	0x0000	ref_acc_x															
0x17	GENERIC_INTERRUPT3_6	0x0000	ref_acc_y															
0x18	GENERIC_INTERRUPT3_7	0x0000	ref_acc_z															
0x19	STEP_COUNTER	0x1800	reserved			sc_en	sd_en	reset_...	watermark_level									
...	-	-	reserved															
0x2B	SIG_MOTION	0x00FA	block_size															
...	-	-	reserved															
0x2E	TILT_1	0xD264	min_tilt_angle								segment_size							
0x2F	TILT_2	0xF069	beta_acc_mean															
0x30	ORIENTATION_1	0x2CFC	hold_time					theta						blocking		mode		ud_en

Table 42: Extended register map overview *(continued)*

Legend			Read-only				Read/Write				Write-only				Reserved			
Addr	**Name**	**Reset value**	**bit15**	**bit14**	**bit13**	**bit12**	**bit11**	**bit10**	**bit9**	**bit8**	**bit7**	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
0x31	`ORIENTATION_2`	0x20CD	hysteresis								slope_thres							
0x32	`FOC_0`	0x0000	reserved							foc_of...								
0x33	`FOC_1`	0x0000	reserved							foc_of...								
0x34	`FOC_2`	0x0000	reserved							foc_of...								
0x35	`FOC_3`	0x0000	reserved									foc_axis_1g			foc_filter_coeff			foc_ap...

6.2.2 Extended Register Map Details

Register (0x02) **FEAT_CONF_ERR**

Description: Bits reflects the error status of accel config for features

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	acc_fo...	orient:.	tilt_c...	sig_mo...	step_c...	gen_in...	gen_in...	gen_in...

- reserved: write 0x0.
- `FEAT_CONF_ERR.gen_int1_conf_err`: (bit offset: 0, bit width: 1, access: read-write) Internal filter cannot produce enough samples for generic interrupt 1 feature .
  Following values can be set to or read from the field gen_int1_conf_err:

Value	**Description**
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.gen_int2_conf_err`: (bit offset: 1, bit width: 1, access: read-write) Internal filter cannot produce enough samples for generic interrupt 2 feature.
  Following values can be set to or read from the field gen_int2_conf_err:

Value	**Description**
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.gen_int3_conf_err`: (bit offset: 2, bit width: 1, access: read-write) Internal filter cannot produce enough samples for generic interrupt 3 feature.
  Following values can be set to or read from the field gen_int3_conf_err:

Value	**Description**
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.step_conf_err`: (bit offset: 3, bit width: 1, access: read-write) Internal filter cannot produce enough samples for step counter and/or step detection features.
  Following values can be set to or read from the field step_conf_err:

Value	Description
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.sig_mo_conf_err`: (bit offset: 4, bit width: 1, access: read-write) Internal filter cannot produce enough samples for significant motion detection feature.
  Following values can be set to or read from the field sig_mo_conf_err:

Value	Description
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.tilt_conf_err`: (bit offset: 5, bit width: 1, access: read-write) Internal filter cannot produce enough samples for tilt detection feature.
  Following values can be set to or read from the field tilt_conf_err:

Value	Description
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.orient_conf_err`: (bit offset: 6, bit width: 1, access: read-write) Internal filter cannot produce enough samples for orientation detection feature.
  Following values can be set to or read from the field orient_conf_err:

Value	Description
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

- `FEAT_CONF_ERR.acc_foc_conf_err`: (bit offset: 7, bit width: 1, access: read-write) Internal filter cannot produce enough samples for accelerometer fast-offset compensation.
  Following values can be set to or read from the field acc_foc_conf_err:

Value	Description
0b0 (0x0)	no error
0b1 (0x1)	feature configuration is invalid

Use this link to go back to the overview table: `FEAT_CONF_ERR`.

Register (0x03) **GENERAL_SETTINGS_0**

Description: Configuration parameters common across all features

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved	feat_z...	feat_y...	feat_x...	feat_axis_ex			android...

- reserved: write 0x0.
- `GENERAL_SETTINGS_0.android_comp`: (bit offset: 0, bit width: 1, access: read-write) Enable Android compatibility mode. Few features operate with Android specific configurations if this bit is set.
  Following values can be set to or read from the field android_comp:

Value	**Description**
0b0 (0x0)	Host configuration
0b1 (0x1)	Android configuration

- `GENERAL_SETTINGS_0.feat_axis_ex`: (bit offset: 1, bit width: 3, access: read-write) Axes exchange scheme that is applied in host software.
  Following values can be set to or read from the field feat_axis_ex:

Value	**Description**
0b000 (0x0)	X(feat)=x(datapath),Y(feat)=y(datapath),Z(feat)=z(datapath)
0b001 (0x1)	X(feat)=y(datapath),Y(feat)=x(datapath),Z(feat)=z(datapath)
0b010 (0x2)	X(feat)=x(datapath),Y(feat)=z(datapath),Z(feat)=y(datapath)
0b011 (0x3)	X(feat)=z(datapath),Y(feat)=x(datapath),Z(feat)=y(datapath)
0b100 (0x4)	X(feat)=y(datapath),Y(feat)=z(datapath),Z(feat)=x(datapath)
0b101 (0x5)	X(feat)=z(datapath),Y(feat)=y(datapath),Z(feat)=x(datapath)
0b110 (0x6)	Same as default_0
0b111 (0x7)	Same as default_0

- `GENERAL_SETTINGS_0.feat_x_inv`: (bit offset: 4, bit width: 1, access: read-write) Invert polarity of X-axis data after axis exchange.
  Following values can be set to or read from the field feat_x_inv:

Value	Description
0b0 (0x0)	X(feat) remains unchanged
0b1 (0x1)	X(feat) = -X(feat)

- `GENERAL_SETTINGS_0.feat_y_inv`: (bit offset: 5, bit width: 1, access: read-write) Invert polarity of Y-axis data after axis exchange.
  Following values can be set to or read from the field feat_y_inv:

Value	Description
0b0 (0x0)	Y(feat) remains unchanged
0b1 (0x1)	Y(feat) = -Y(feat)

- `GENERAL_SETTINGS_0.feat_z_inv`: (bit offset: 6, bit width: 1, access: read-write) Invert polarity of Z-axis data after axis exchange.
  Following values can be set to or read from the field feat_z_inv:

Value	Description
0b0 (0x0)	Z(feat) remains unchanged
0b1 (0x1)	Z(feat) = -Z(feat)

Use this link to go back to the overview table: `GENERAL_SETTINGS_0`.

Register (0x04) **GENERIC_INTERRUPT1_1**

Description: Configuration of acceleration slope threshold, axis enabling and evaluation condition between axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	0	0	0	0	0
Content	axis_sel			comb_sel	slope_thres			

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1
Content	slope_thres							

- `GENERIC_INTERRUPT1_1.slope_thres`: (bit offset: 0, bit width: 12, access: read-write) Minimum/maximum slope of acceleration signal for interrupt detection based on selected motion criterion. . The field slope_thres has the following properties:

Property	**Value**
Bitwidth	12
Sign	unsigned
Unit	g
Scaling	512.0
Default value	15/512
Range	Min=0.0, Max=7.998046875

- `GENERIC_INTERRUPT1_1.comb_sel`: (bit offset: 12, bit width: 1, access: read-write) Logical evaluation condition between enabled axis status.
  Following values can be set to or read from the field comb_sel:

Value	**Description**
0b0 (0x0)	One of the enabled axis should meet the set condition
0b1 (0x1)	All of the enabled axis should meet the set condition

The field comb_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	0
Range	Min=0, Max=1

- `GENERIC_INTERRUPT1_1.axis_sel`: (bit offset: 13, bit width: 3, access: read-write) Enabling of axis for generic

interrupt detection. The field axis_sel has the following properties:

Property	Value
Bitwidth	3
Default value	7
Range	Min=0, Max=7

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_1`.

Register (0x05) **GENERIC_INTERRUPT1_2**

Description: Configuration for hysteresis of acceleration slope, motion criterion selection and reference update mode selection.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	0	0
Content	reserved			acc_ref_up		criteri...	hysteresis	

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	1	1
Content	hysteresis							

- reserved: write 0x0.
- `GENERIC_INTERRUPT1_2.hysteresis`: (bit offset: 0, bit width: 10, access: read-write) Hysteresis for the slope of the acceleration signal. The field hysteresis has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512.0
Default value	3/512
Range	Min=0.0, Max=1.998046875

- `GENERIC_INTERRUPT1_2.criterion_sel`: (bit offset: 10, bit width: 1, access: read-write) Selection of motion criterion .
  Following values can be set to or read from the field criterion_sel:

Value	**Description**
0b0 (0x0)	Evaluate the condition for stationary state of the device
0b1 (0x1)	Evaluate the condition for motion state of the device

The field criterion_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	1
Range	Min=0, Max=1

- `GENERIC_INTERRUPT1_2.acc_ref_up`: (bit offset: 11, bit width: 2, access: read-write) Mode of the acceleration reference update. .
  Following values can be set to or read from the field acc_ref_up:

Value	Description
0b00 (0x0)	On detection of the event
0b01 (0x1)	On update of acceleration signal
0b10 (0x2)	Manually update by host

The field acc_ref_up has the following properties:

Property	Value
Bitwidth	2
Default value	1
Range	Min=0, Max=2

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_2`.

Register (0x06) **GENERIC_INTERRUPT1_3**

Description: Configuration of timing related parameters for generic interrupt detection

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	1	0	0	0	0	0
Content	wait_time			duration				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	1	0
Content	duration							

- `GENERIC_INTERRUPT1_3.duration`: (bit offset: 0, bit width: 13, access: read-write) Minimum duration for which the selected criterion is true for interrupt detection. . The field duration has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.2
Range	Min=0.0, Max=163.82

- `GENERIC_INTERRUPT1_3.wait_time`: (bit offset: 13, bit width: 3, access: read-write) Wait time for clearing the event after condition evaluates false. The field wait_time has the following properties:

Property	**Value**
Bitwidth	3
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.06
Range	Min=0.0, Max=0.14

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_3`.

Register (0x07) **GENERIC_INTERRUPT1_4**

Description: Configuration for quiet time between interrupts

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			quiet_time				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	0	0	0	0	0	0
Content	quiet_time							

- reserved: write 0x0.
- `GENERIC_INTERRUPT1_4.quiet_time`: (bit offset: 0, bit width: 13, access: read-write) Quiet time after an interrupt where no additional interrupts are detected . The field quiet_time has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	1.28
Range	Min=0.0, Max=163.82

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_4`.

Register (0x08) **GENERIC_INTERRUPT1_5**

Description: Manually set acceleration signal reference for x-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

- `GENERIC_INTERRUPT1_5.ref_acc_x`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for x-axis . The field ref_acc_x has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_5`.

Register (0x09) **GENERIC_INTERRUPT1_6**

Description: Manually set acceleration signal reference for y-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

- `GENERIC_INTERRUPT1_6.ref_acc_y`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for y-axis . The field ref_acc_y has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_6`.

Register (0x0A) **GENERIC_INTERRUPT1_7**

Description: Manually set acceleration signal reference for z-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	0	0
Content	ref_acc_z							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_z							

- `GENERIC_INTERRUPT1_7.ref_acc_z`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for z-axis . The field ref_acc_z has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	1.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT1_7`.

Register (0x0B) **GENERIC_INTERRUPT2_1**

Description: Configuration of acceleration slope threshold, axis enabling and evaluation condition between axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	0	0	0	0
Content	axis_sel			comb_sel	slope_thres			

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1
Content	slope_thres							

- `GENERIC_INTERRUPT2_1.slope_thres`: (bit offset: 0, bit width: 12, access: read-write) Minimum/maximum slope of acceleration signal for interrupt detection based on selected motion criterion. . The field slope_thres has the following properties:

Property	**Value**
Bitwidth	12
Sign	unsigned
Unit	g
Scaling	512.0
Default value	15/512
Range	Min=0.0, Max=7.998046875

- `GENERIC_INTERRUPT2_1.comb_sel`: (bit offset: 12, bit width: 1, access: read-write) Logical evaluation condition between enabled axis status.
  Following values can be set to or read from the field comb_sel:

Value	**Description**
0b0 (0x0)	One of the enabled axis should meet the set condition
0b1 (0x1)	All of the enabled axis should meet the set condition

The field comb_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	1
Range	Min=0, Max=1

- `GENERIC_INTERRUPT2_1.axis_sel`: (bit offset: 13, bit width: 3, access: read-write) Enabling of axis for generic

interrupt detection. The field axis_sel has the following properties:

Property	Value
Bitwidth	3
Default value	7
Range	Min=0, Max=7

Use this link to go back to the overview table: GENERIC_INTERRUPT2_1.

Register (0x0C) **GENERIC_INTERRUPT2_2**

Description: Configuration for hysteresis of acceleration slope, motion criterion selection and reference update mode selection.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	0	0
Content	reserved			acc_ref_up		criteri...	hysteresis	

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	1	1
Content	hysteresis							

- reserved: write 0x0.
- `GENERIC_INTERRUPT2_2.hysteresis`: (bit offset: 0, bit width: 10, access: read-write) Hysteresis for the slope of the acceleration signal. The field hysteresis has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512.0
Default value	3/512
Range	Min=0.0, Max=1.998046875

- `GENERIC_INTERRUPT2_2.criterion_sel`: (bit offset: 10, bit width: 1, access: read-write) Selection of motion criterion
  .
  Following values can be set to or read from the field criterion_sel:

Value	**Description**
0b0 (0x0)	Evaluate the condition for stationary state of the device
0b1 (0x1)	Evaluate the condition for motion state of the device

The field criterion_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	0
Range	Min=0, Max=1

- GENERIC_INTERRUPT2_2.acc_ref_up: (bit offset: 11, bit width: 2, access: read-write) Mode of the acceleration reference update. .
  Following values can be set to or read from the field acc_ref_up:

Value	Description
0b00 (0x0)	On detection of the event
0b01 (0x1)	On update of acceleration signal
0b10 (0x2)	Manually update by host

The field acc_ref_up has the following properties:

Property	Value
Bitwidth	2
Default value	1
Range	Min=0, Max=2

Use this link to go back to the overview table: GENERIC_INTERRUPT2_2.

Register (0x0D) **GENERIC_INTERRUPT2_3**

Description: Configuration of timing related parameters for generic interrupt detection

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	1	0	0	0	0	0
Content	wait_time			duration				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	1	0
Content	duration							

- `GENERIC_INTERRUPT2_3.duration`: (bit offset: 0, bit width: 13, access: read-write) Minimum duration for which the selected criterion is true for interrupt detection. . The field duration has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.2
Range	Min=0.0, Max=163.82

- `GENERIC_INTERRUPT2_3.wait_time`: (bit offset: 13, bit width: 3, access: read-write) Wait time for clearing the event after condition evaluates false. The field wait_time has the following properties:

Property	**Value**
Bitwidth	3
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.06
Range	Min=0.0, Max=0.14

Use this link to go back to the overview table: `GENERIC_INTERRUPT2_3`.

Register (0x0E) **GENERIC_INTERRUPT2_4**

Description: Configuration for quiet time between interrupts

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			quiet_time				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	0	0	0	0	0	0
Content	quiet_time							

- reserved: write 0x0.
- `GENERIC_INTERRUPT2_4.quiet_time`: (bit offset: 0, bit width: 13, access: read-write) Quiet time after an interrupt where no additional interrupts are detected . The field quiet_time has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	1.28
Range	Min=0.0, Max=163.82

Use this link to go back to the overview table: `GENERIC_INTERRUPT2_4`.

Register (0x0F) **GENERIC_INTERRUPT2_5**

Description: Manually set acceleration signal reference for x-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

- `GENERIC_INTERRUPT2_5.ref_acc_x`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for x-axis . The field ref_acc_x has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT2_5`.

Register (0x10) **GENERIC_INTERRUPT2_6**

Description: Manually set acceleration signal reference for y-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

- `GENERIC_INTERRUPT2_6.ref_acc_y`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for y-axis . The field ref_acc_y has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT2_6`.

Register (0x11) **GENERIC_INTERRUPT2_7**

Description: Manually set acceleration signal reference for z-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	0	0
Content	ref_acc_z							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_z							

- `GENERIC_INTERRUPT2_7.ref_acc_z`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for z-axis . The field ref_acc_z has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	1.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT2_7`.

Register (0x12) **GENERIC_INTERRUPT3_1**

Description: Configuration of acceleration slope threshold, axis enabling and evaluation condition between axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	0	0	0	0
Content	axis_sel			comb_sel	slope_thres			

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	0	0	0	0	0	1	0
Content	slope_thres							

- `GENERIC_INTERRUPT3_1.slope_thres`: (bit offset: 0, bit width: 12, access: read-write) Minimum/maximum slope of acceleration signal for interrupt detection based on selected motion criterion. . The field slope_thres has the following properties:

Property	**Value**
Bitwidth	12
Sign	unsigned
Unit	g
Scaling	512.0
Default value	130/512
Range	Min=0.0, Max=7.998046875

- `GENERIC_INTERRUPT3_1.comb_sel`: (bit offset: 12, bit width: 1, access: read-write) Logical evaluation condition between enabled axis status.
  Following values can be set to or read from the field comb_sel:

Value	**Description**
0b0 (0x0)	One of the enabled axis should meet the set condition
0b1 (0x1)	All of the enabled axis should meet the set condition

The field comb_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	1
Range	Min=0, Max=1

- `GENERIC_INTERRUPT3_1.axis_sel`: (bit offset: 13, bit width: 3, access: read-write) Enabling of axis for generic

interrupt detection. The field axis_sel has the following properties:

Property	**Value**
Bitwidth	3
Default value	7
Range	Min=0, Max=7

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_1`.

Register (0x13) **GENERIC_INTERRUPT3_2**

Description: Configuration for hysteresis of acceleration slope, motion criterion selection and reference update mode selection.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0
Content	reserved			acc_ref_up		criteri...	hysteresis	

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	0	0
Content	hysteresis							

- reserved: write 0x0.
- `GENERIC_INTERRUPT3_2.hysteresis`: (bit offset: 0, bit width: 10, access: read-write) Hysteresis for the slope of the acceleration signal. The field hysteresis has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512.0
Default value	8/512
Range	Min=0.0, Max=1.998046875

- `GENERIC_INTERRUPT3_2.criterion_sel`: (bit offset: 10, bit width: 1, access: read-write) Selection of motion criterion
  .
  Following values can be set to or read from the field criterion_sel:

Value	**Description**
0b0 (0x0)	Evaluate the condition for stationary state of the device
0b1 (0x1)	Evaluate the condition for motion state of the device

The field criterion_sel has the following properties:

Property	**Value**
Bitwidth	1
Default value	0
Range	Min=0, Max=1

- `GENERIC_INTERRUPT3_2.acc_ref_up`: (bit offset: 11, bit width: 2, access: read-write) Mode of the acceleration reference update. .
  Following values can be set to or read from the field acc_ref_up:

Value	Description
0b00 (0x0)	On detection of the event
0b01 (0x1)	On update of acceleration signal
0b10 (0x2)	Manually update by host

The field acc_ref_up has the following properties:

Property	Value
Bitwidth	2
Default value	2
Range	Min=0, Max=2

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_2`.

Register (0x14) **GENERIC_INTERRUPT3_3**

Description: Configuration of timing related parameters for generic interrupt detection

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	0	0	0	0	0	0
Content	wait_time			duration				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	1	1
Content	duration							

- `GENERIC_INTERRUPT3_3.duration`: (bit offset: 0, bit width: 13, access: read-write) Minimum duration for which the selected criterion is true for interrupt detection. . The field duration has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.06
Range	Min=0.0, Max=163.82

- `GENERIC_INTERRUPT3_3.wait_time`: (bit offset: 13, bit width: 3, access: read-write) Wait time for clearing the event after condition evaluates false. The field wait_time has the following properties:

Property	**Value**
Bitwidth	3
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.04
Range	Min=0.0, Max=0.14

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_3`.

Register (0x15) **GENERIC_INTERRUPT3_4**

Description: Configuration for quiet time between interrupts

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			quiet_time				

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	0	0	0	0	0	0
Content	quiet_time							

- reserved: write 0x0.
- `GENERIC_INTERRUPT3_4.quiet_time`: (bit offset: 0, bit width: 13, access: read-write) Quiet time after an interrupt where no additional interrupts are detected . The field quiet_time has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50.0
Default value	1.28
Range	Min=0.0, Max=163.82

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_4`.

Register (0x16) **GENERIC_INTERRUPT3_5**

Description: Manually set acceleration signal reference for x-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_x							

- `GENERIC_INTERRUPT3_5.ref_acc_x`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for x-axis . The field ref_acc_x has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_5`.

Register (0x17) **GENERIC_INTERRUPT3_6**

Description: Manually set acceleration signal reference for y-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_y							

- `GENERIC_INTERRUPT3_6.ref_acc_y`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for y-axis . The field ref_acc_y has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_6`.

Register (0x18) **GENERIC_INTERRUPT3_7**

Description: Manually set acceleration signal reference for z-axis

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_z							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	ref_acc_z							

- `GENERIC_INTERRUPT3_7.ref_acc_z`: (bit offset: 0, bit width: 16, access: read-write) Reference acceleration signal for z-axis . The field ref_acc_z has the following properties:

Property	**Value**
Bitwidth	16
Sign	signed
Unit	g
Scaling	2048.0
Default value	0.0
Range	Min=-16.0, Max=15.99951171875

Use this link to go back to the overview table: `GENERIC_INTERRUPT3_7`.

Register (0x19) **STEP_COUNTER**

Description: Configuration for step counter watermark and global reset

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	1	0	0	0
Content	reserved			sc_en	sd_en	reset_...	watermark_level	

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	watermark_level							

- reserved: write 0x0.
- `STEP_COUNTER.watermark_level`: (bit offset: 0, bit width: 10, access: read-write) An interrupt will be triggered every time the difference in number of steps counted from last event is equal to (set value * 20). If 0, the interrupt is disabled . The field watermark_level has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Scaling	20
Default value	0
Range	Min=0, Max=1023

- `STEP_COUNTER.reset_counter`: (bit offset: 10, bit width: 1, access: read-write) Reset the accumulated step count value . The field reset_counter has the following properties:

Property	**Value**
Bitwidth	1
Sign	unsigned
Default value	0
Range	Min=0, Max=1

- `STEP_COUNTER.sd_en`: (bit offset: 11, bit width: 1, access: read-write) Enable step detector.
- `STEP_COUNTER.sc_en`: (bit offset: 12, bit width: 1, access: read-write) Enable step counter.

Use this link to go back to the overview table: `STEP_COUNTER`.

Register (0x2B) **SIG_MOTION**

Description: Size of the segment for detection of significant motion

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	block_size							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	0	1	0
Content	block_size							

- `SIG_MOTION.block_size`: (bit offset: 0, bit width: 16, access: read-write) Size of the segment for detection of significant motion of the device . The field block_size has the following properties:

Property	**Value**
Bitwidth	16
Sign	unsigned
Unit	s
Scaling	50
Default value	5.0
Range	Min=0.0, Max=1310.7

Use this link to go back to the overview table: `SIG_MOTION`.

Register (0x2E) **TILT_1**

Description: Configuration for averaging duration of reference vector and minimum tilt angle

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	0	1	0	0	1	0
Content	min_tilt_angle							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	1	0	0	1	0	0
Content	segment_size							

- `TILT_1.segment_size`: (bit offset: 0, bit width: 8, access: read-write) Duration for which the acceleration vector is averaged to be reference vector . The field segment_size has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	s
Scaling	50
Default value	2.0
Range	Min=0.0, Max=5.10

- `TILT_1.min_tilt_angle`: (bit offset: 8, bit width: 8, access: read-write) Minimum angle by which the device shall be tilted for event detection . The field min_tilt_angle has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	degrees
Scaling	256
Default value	35.0
Range	Min=0.0, Max=90.0
Interpretation	cos(x)

Use this link to go back to the overview table: `TILT_1`.

Register (0x2F) **TILT_2**

Description: Configuration for averaging of acceleration vector

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	0	0	0	0
Content	beta_acc_mean							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	1	0	1	0	0	1
Content	beta_acc_mean							

- `TILT_2.beta_acc_mean`: (bit offset: 0, bit width: 16, access: read-write) Exponential smoothing coefficient for computing low-pass mean of acceleration vector . The field beta_acc_mean has the following properties:

Property	**Value**
Bitwidth	16
Sign	unsigned
Scaling	65536
Default value	2.0
Range	Min=0.0, Max=5.1
Interpretation	exp(2*pi/(50*x))

Use this link to go back to the overview table: `TILT_2`.

Register (0x30) **ORIENTATION_1**

Description: Orientation general configuration flags

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	0	1	1	0	0
Content	hold_time					theta		

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	0	0
Content	theta			blocking		mode		ud_en

- `ORIENTATION_1.ud_en`: (bit offset: 0, bit width: 1, access: read-write) Selection of upside down orientation detection. Following values can be set to or read from the field ud_en:

Value	Description
0b0 (0x0)	Disable detection of upside-down position
0b1 (0x1)	Enable detection of upside-down position

The field ud_en has the following properties:

Property	Value
Bitwidth	1
Default value	0
Range	Min=0, Max=1

- `ORIENTATION_1.mode`: (bit offset: 1, bit width: 2, access: read-write) Selection of mode for orientation spread in the detection plane.
  Following values can be set to or read from the field mode:

Value	Description
0b00 (0x0)	Symmetrical spread of area for portrait and landscape orientations
0b01 (0x1)	Area of landscape is more compared to portrait orientation
0b10 (0x2)	Area of portrait is more compared to landscape orientation

The field mode has the following properties:

Property	Value
Bitwidth	2
Default value	2
Range	Min=0, Max=2

- `ORIENTATION_1.blocking`: (bit offset: 3, bit width: 2, access: read-write) Blocking allows to prevent change of orientation during large movement of device .
  Following values can be set to or read from the field blocking:

Value	Description
0b00 (0x0)	Blocking is disabled
0b01 (0x1)	Block if acceleration on any axis is greater than 1.5g
0b10 (0x2)	Block if acceleration on any axis is greater than 1.5g or slope is greater than half of slope_thres
0b11 (0x3)	Block if acceleration on any axis is greater than 1.5g or slope is greater than slope_thres

The field blocking has the following properties:

Property	Value
Bitwidth	2
Default value	3
Range	Min=0, Max=3

- `ORIENTATION_1.theta`: (bit offset: 5, bit width: 6, access: read-write) Maximum allowed tilt angle for device to be in flat state. The field theta has the following properties:

Property	Value
Bitwidth	6
Sign	unsigned
Unit	degrees
Scaling	64.0
Default value	37.9764794968186
Range	Min=0, Max=44.77442373390876
Interpretation	(tan(x))^2

- `ORIENTATION_1.hold_time`: (bit offset: 11, bit width: 5, access: read-write) Minimum duration the device shall be in new orientation for change detection. The field hold_time has the following properties:

Property	Value
Bitwidth	5
Sign	unsigned
Unit	s
Scaling	50
Default value	0.1
Range	Min=0.0, Max=0.62

Use this link to go back to the overview table: `ORIENTATION_1`.

Register (0x31) **ORIENTATION_2**

Description: Settings for acceleration slope and hysteresis in orientation detection

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	1	0	0	0	0	0
Content	hysteresis							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	0	0	1	1	0	1
Content	slope_thres							

- `ORIENTATION_2.slope_thres`: (bit offset: 0, bit width: 8, access: read-write) Minimum slope between consecutive acceleration samples to prevent the change of orientation during large movement. The field slope_thres has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	g
Scaling	512
Default value	205/512
Range	Min=0.0, Max=0.498046875

- `ORIENTATION_2.hysteresis`: (bit offset: 8, bit width: 8, access: read-write) Hysteresis of acceleration for orientation change detection. The field hysteresis has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	g
Scaling	512
Default value	32/512
Range	Min=0.0, Max=0.498046875

Use this link to go back to the overview table: `ORIENTATION_2`.

Register (0x32) **FOC_0**

Description: Accelerometer fast-offset compensation feature

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							foc_of...

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	foc_off_x							

- reserved: write 0x0.
- `FOC_0.foc_off_x`: (bit offset: 0, bit width: 9, access: read-write) Offset estimated for accelerometer X-axis using fast-offset compensation feature. Value has same range and resolution as the user offset registers. . The field foc_off_x has the following properties:

Property	**Value**
Bitwidth	9
Sign	signed
Scaling	1024
Default value	0
Range	Min=-256, Max=255

Use this link to go back to the overview table: `FOC_0`.

Register (0x33) **FOC_1**

Description: Accelerometer fast-offset compensation feature

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							foc_of...

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	foc_off_y							

- reserved: write 0x0.
- `FOC_1.foc_off_y`: (bit offset: 0, bit width: 9, access: read-write) Offset estimated for accelerometer Y-axis using fast-offset compensation feature. Value has same range and resolution as the user offset registers. . The field foc_off_y has the following properties:

Property	**Value**
Bitwidth	9
Sign	signed
Scaling	1024
Default value	0
Range	Min=-256, Max=255

Use this link to go back to the overview table: `FOC_1`.

Register (0x34) **FOC_2**

Description: Accelerometer fast-offset compensation feature

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							foc_of...

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	foc_off_z							

- reserved: write 0x0.
- `FOC_2.foc_off_z`: (bit offset: 0, bit width: 9, access: read-write) Offset estimated for accelerometer Z-axis using fast-offset compensation feature. Value has same range and resolution as the user offset registers. . The field foc_off_z has the following properties:

Property	**Value**
Bitwidth	9
Sign	signed
Scaling	1024
Default value	0
Range	Min=-256, Max=255

Use this link to go back to the overview table: `FOC_2`.

Register (0x35) **FOC_3**

Description: Accelerometer fast-offset compensation feature

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved	foc_axis_1g			foc_filter_coeff			foc_ap...

- reserved: write 0x0.
- `FOC_3.foc_apply_corr`: (bit offset: 0, bit width: 1, access: read-write) Update user offset registers with estimated offset values after feature completion.
- `FOC_3.foc_filter_coeff`: (bit offset: 1, bit width: 3, access: read-write) Number of 200Hz accelerometer samples that are averaged to estimate the offset. Number of samples = 2^(foc_filter_coeff + 3). Default value of field is 0 which means 8 samples will be averaged. .
- `FOC_3.foc_axis_1g`: (bit offset: 4, bit width: 3, access: read-write) Fast-offset compensation must be executed only when the device is still and one axis is parallel to the gravitation vector. This axis can be either aligned with gravitational vector or in the opposite direction of the gravitational vector. Device will not warn the user if the device is not static or an axis is not parallel to the gravitational vector. .
  Following values can be set to or read from the field foc_axis_1g:

Value	**Description**
0b000 (0x0)	Z-axis shows 1G
0b001 (0x1)	Z-axis shows -1G
0b010 (0x2)	Y-axis shows 1G
0b011 (0x3)	Y-axis shows -1G
0b100 (0x4)	X-axis shows 1G
0b101 (0x5)	X-axis shows -1G
0b110 (0x6)	Same as x_plus_1g
0b111 (0x7)	Same as x_minus_1g

Use this link to go back to the overview table: `FOC_3`.

7 Pin Out and Connection Diagrams

7.1 Pin Out

The figures 51 and 52 shows the pin-out of the device from top and bottom view, respectively.

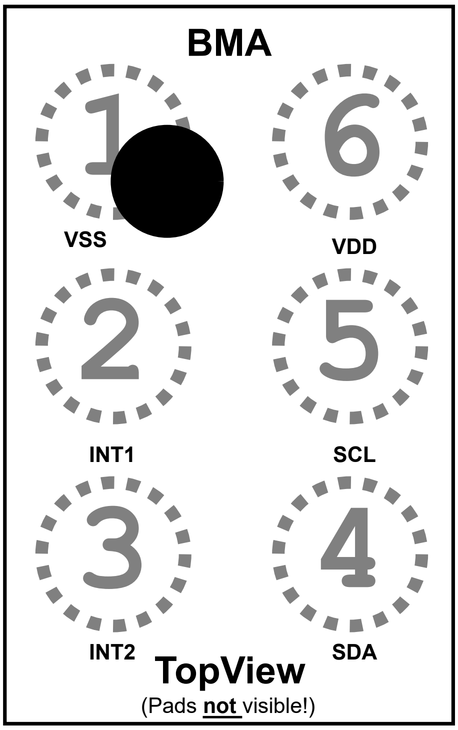

Figure 51: Pin-out: top view

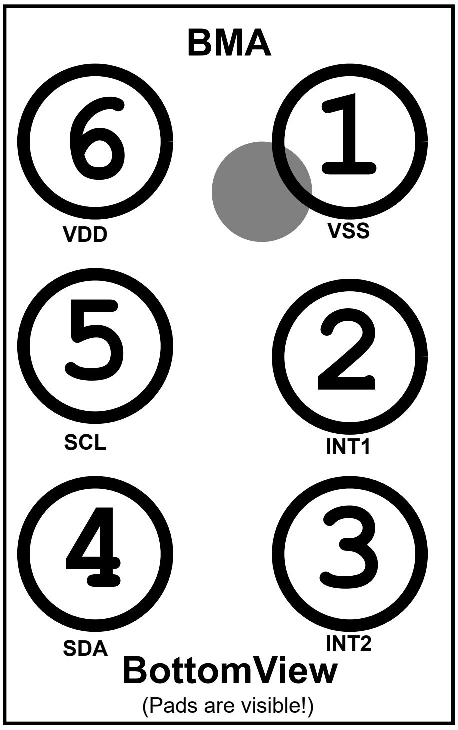

Figure 52: Pin-out: bottom view

The table 43 details the pin out and the connections of the individual pins of the device.

Table 43: Pin-out and pin connections

				Connect to		
Pin #	**Name**	**I/O Type**	**Description**	**in SPI 4-wire**	**in SPI 3-Wire**	**in I²C/I3C**
1	VSS	Ground	Ground (VSS=GND=GNDIO)	GND	GND	GND
2	INT1	Digital I/O	Interrupt pin 1 (or Serial Data)	SDO/MISO	INT1	INT1
3	INT2	Digital I/O	Interrupt pin 2 (or Chip Select for SPI)	CSB	CSB	INT2* (or VDD, if unused)
4	SDA	Digital I/O	Serial Data	SDI/MOSI	SDX	SDA
5	SCL	Digital I/O	Serial Clock	SCK	SCK	SCL
6	VDD	Supply	Power supply analog & digital domain and digital I/O 1.62V ... 3.63V (VDD=VDDIO)	VDD (= VDDIO)	VDD (= VDDIO)	VDD (= VDDIO)

*Do not drive INT2 low during startup, see the following chapter 7.2.1 for more details.

7.2 Connection Diagrams

7.2.1 Connection Diagrams with I²C and I3C

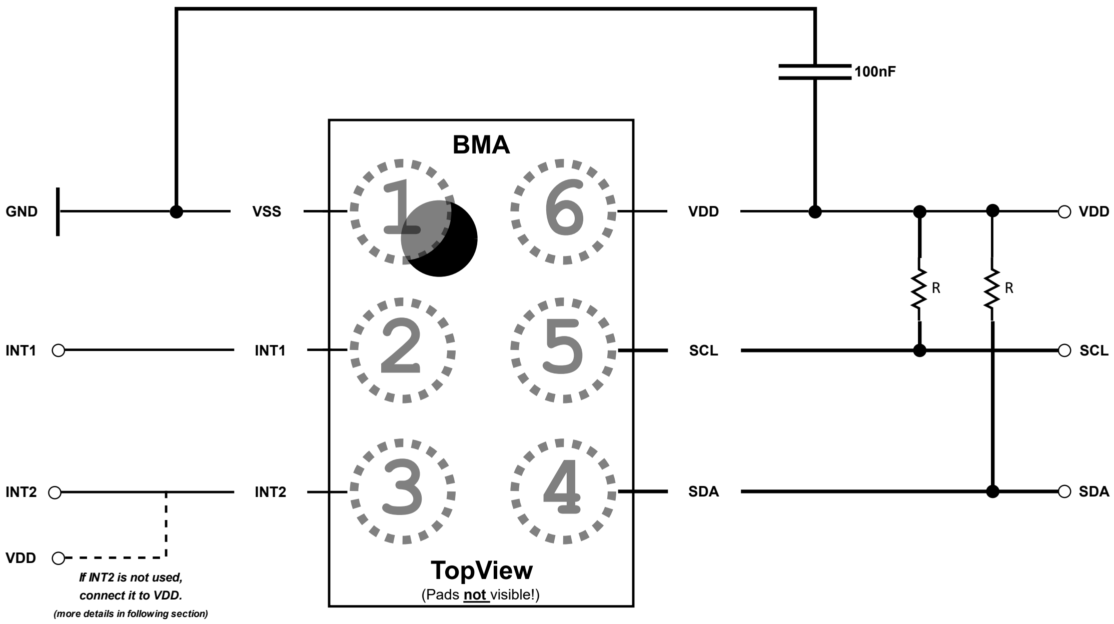

Figure 53: Connection Diagram with I²C and I3C

It is recommended to use 100nF decoupling capacitors at pin 6 (VDD).

Please note for the I²C/I3C mode, as already mentioned in chapter 5.2.1.1:

- Configure the pin 3 to be an output.
- When the output characteristics of pin 3 is disabled (or not yet enabled), please do not connect pin 3 to ground, as shown in figure 54.
- If the pin 3 is not used, connected it to VDD.

When the output characteristics of pin 3 is disabled

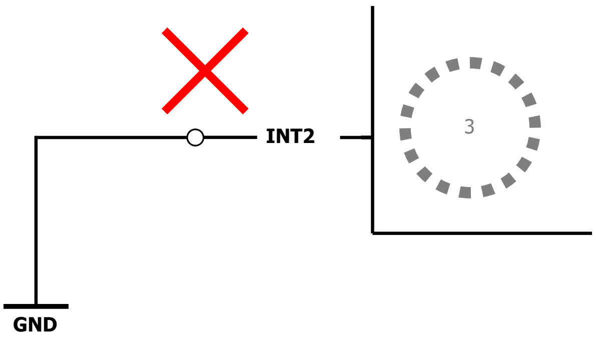

Figure 54: Connection that is not allowed in I²C and I3C before the pin 3 is configured to the output characteristics

- When the output characteristics of pin 3 is disabled (or not yet enabled), please do not connect pin 3 to a GPIO pin configured in the pull-down state, as shown in figure 55. Please consider this behavior especially during the startup of the device, since pin 3 is used in SPI Mode as CSB and therefore a connection to ground will trigger the SPI mode.

When the output characteristics of pin 3 is disabled

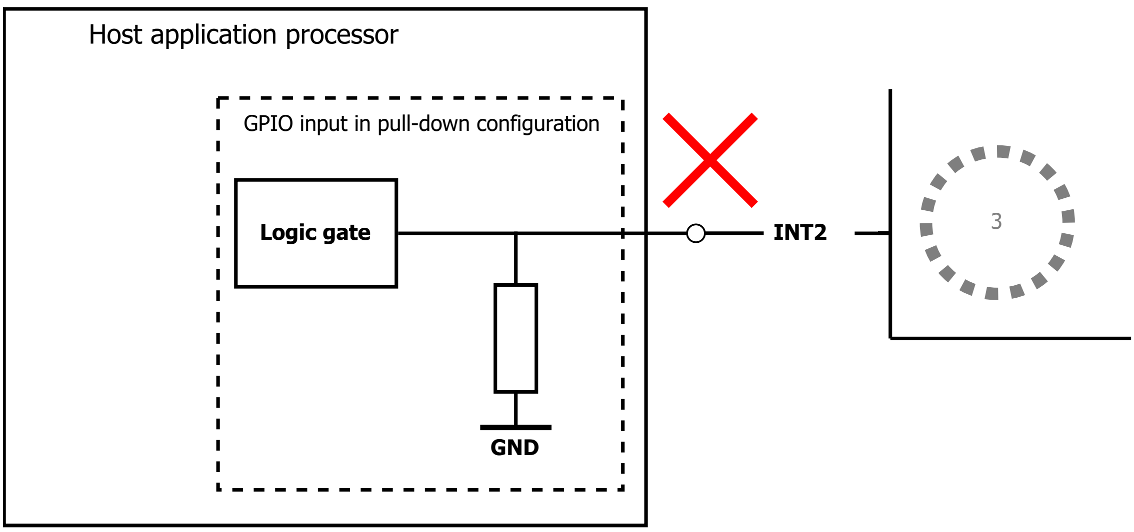

Figure 55: Connection that is not allowed in I²C and I3C before the pin 3 is configured to the output characteristics

7.2.2 Connection Diagrams with SPI (3-Wire)

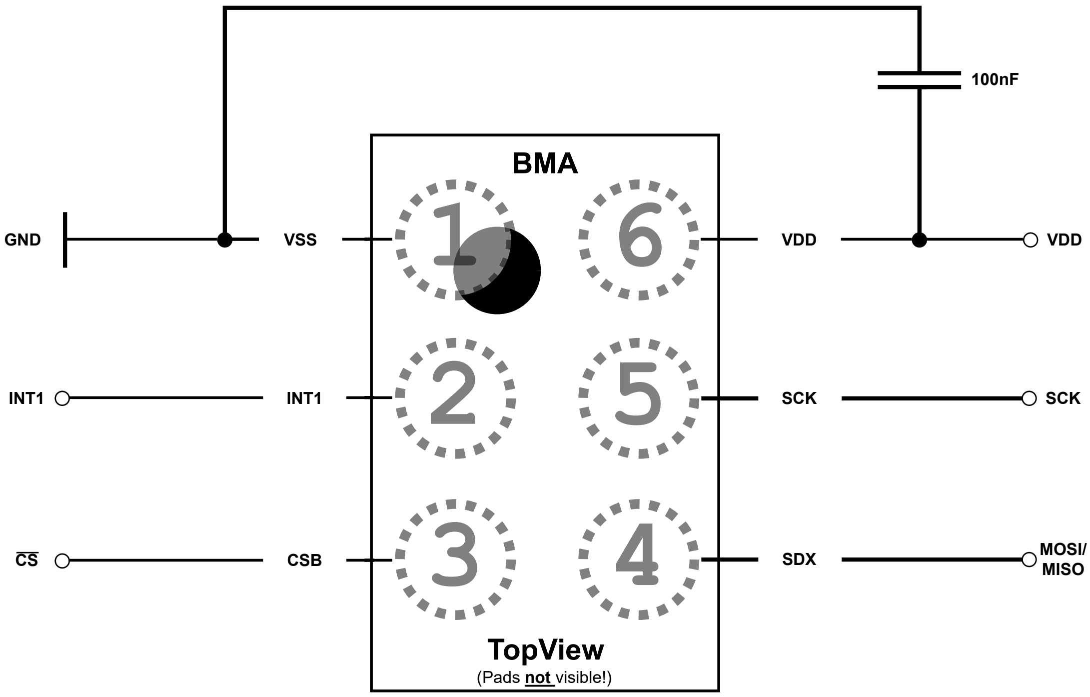

Figure 56: Connection Diagram with SPI 3-Wire

It is recommended to use 100nF decoupling capacitors at pin 6 (VDD).

7.2.3 Connection Diagrams with SPI (4-Wire)

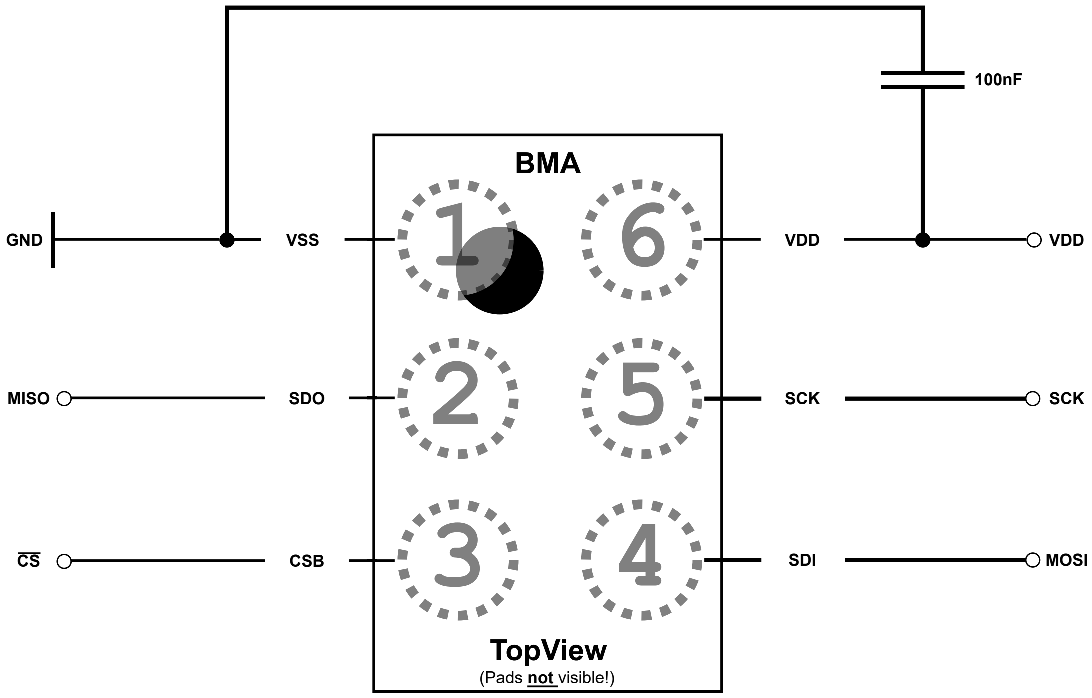

Figure 57: Connection Diagram with SPI 4-Wire

It is recommended to use 100nF decoupling capacitors at pin 6 (VDD).

8 Package

8.1 Dimensions

The BMA530 has a very compact Wafer Level Chip Scale Package (WLCSP). Figures 58, 59 and 60 show the dimensions of the package, the unit for all dimension specifications in the figures is mm.

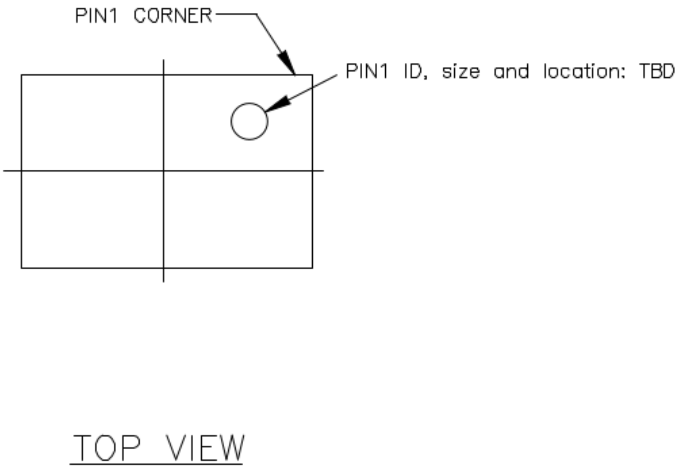

Figure 58: Dimensions from top (in mm)

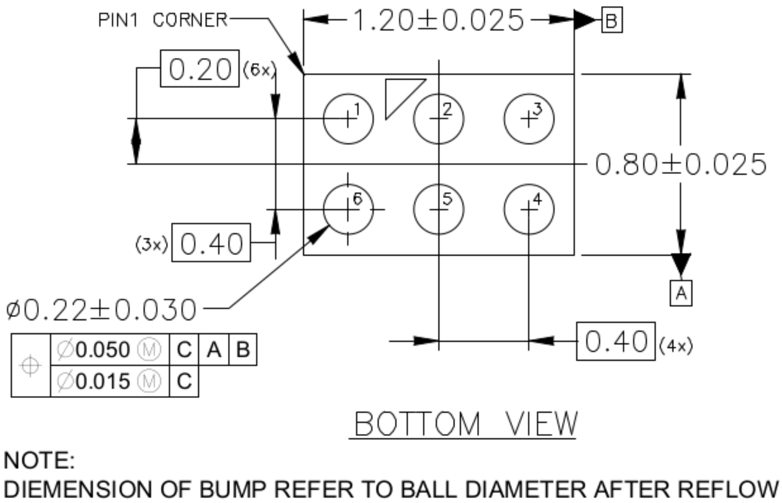

Figure 59: Dimensions from bottom view (in mm)

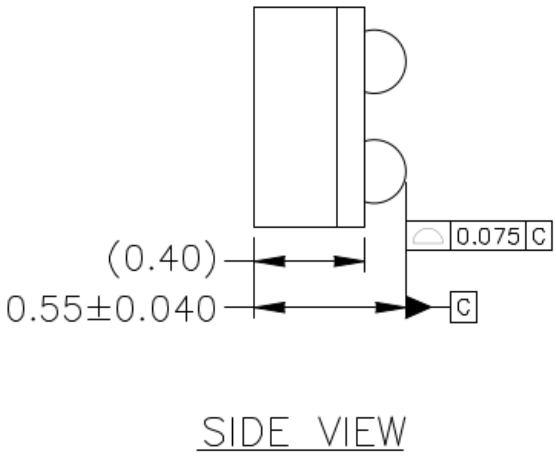

Figure 60: Dimensions from side view (in mm)

8.2 Sensing Axis Orientation

If the sensor is accelerated and/or rotated in the indicated directions, the corresponding channels of the device will deliver a positive acceleration and/or yaw rate signal (dynamic acceleration). If the sensor is at rest without any rotation and the force of gravity is acting contrary to the indicated directions, the output of the corresponding acceleration channel will be positive.

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:

- $\pm 0g$ for the x accelerometer channel
- $\pm 0g$ for the y accelerometer channel
- $+1g$ for the z accelerometer channel

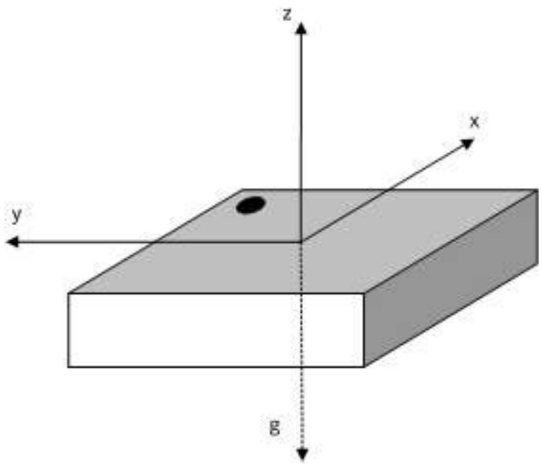

Figure 61: Definition of the sensing axes orientation for the raw device

For reference, Figure 62 below shows the smartphone device orientation with an integrated device.

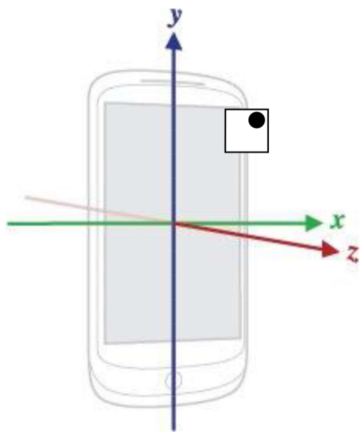

Figure 62: Definition of the sensing axes orientation within a device

8.3 Landing Pattern Recommendation

Figure 63 provides the recommendation for the landing pad to ensure maximum stability of the solder connections.

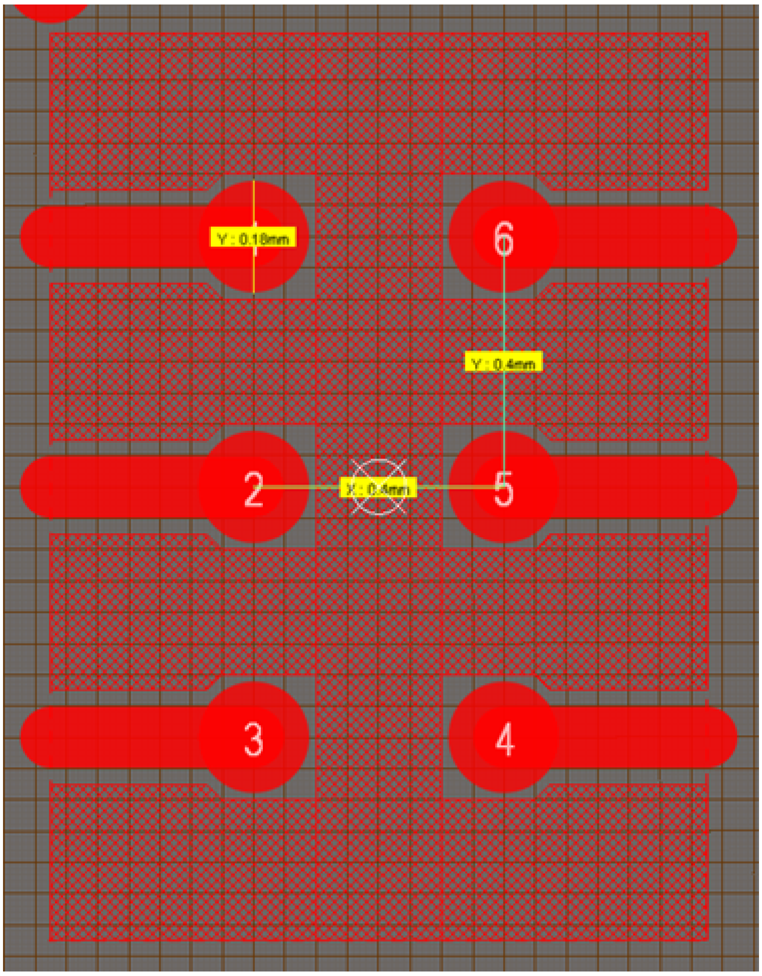

Figure 63: Landing pattern recommendation

IPC recommends shrinking of PCB-pads to nominal ball diameter:

- PCB-pad size = 0.18 mm
- Nominal ball diameter = 0.21 mm

8.4 Marking

Mass Production

Table 44: Marking – Mass Production

Labeling	Symbol	Name	Remark
ACCC ⊕ 6CCCCC	A	Product Identifier	One alphanumeric digit, fixed to "A" to identify the product
	6	Internal Code	1 alphanumeric digit, fixed to "6", internal use only
	CCC...	Counter ID	Tracing identification by eight alphanumeric digits
	⊙	Pin 1	Identifier on top side

Engineering Samples

Table 45: Marking – Engineering Samples

Labeling	Symbol	Name	Remark
ECCC ⊕ ACCCCC	A	Product Identifier	One alphanumeric digit, fixed to "A" to identify the product
	E	Internal Code	1 alphanumeric digit, fixed to "E", internal use only
	CCC...	Counter ID	Tracing identification by three alphanumeric digits
	⊙	Pin 1	Identifier on top side

8.5 Tape and Reel Information

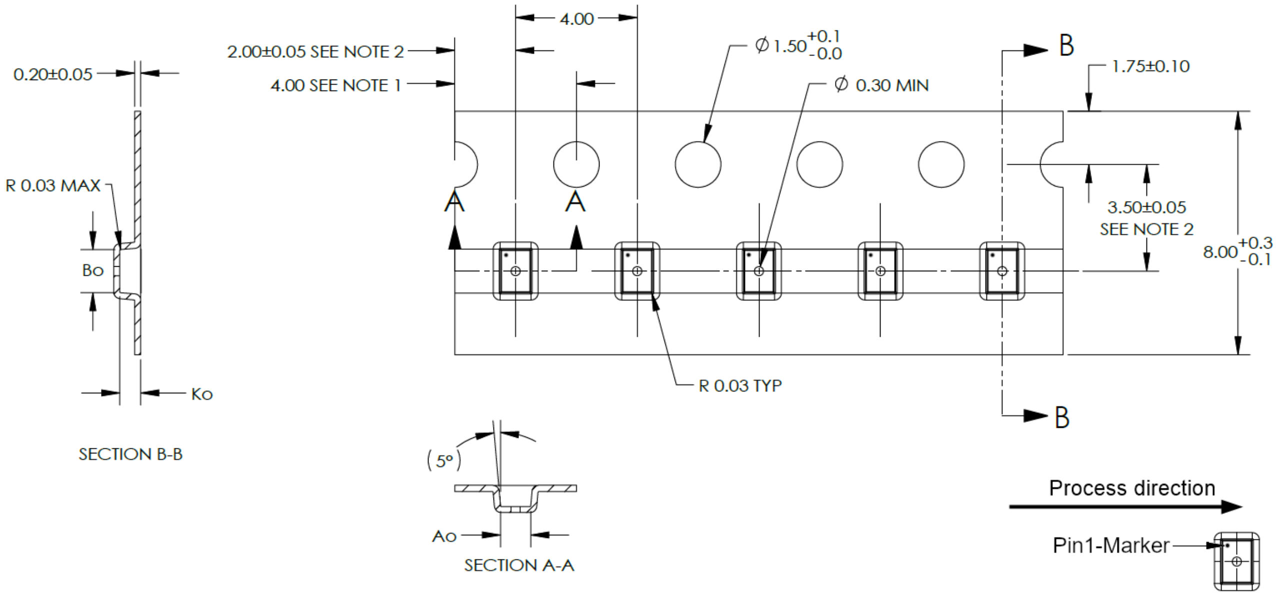

Figure 64: Tape and Reel information

8.6 Soldering Guidelines

The moisture sensitivity level of the device corresponds to JEDEC Level 1, see also

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

Both documents are available on the JEDEC website `https://www.jedec.org/`.

The sensor fulfills the lead-free soldering requirements of the above-mentioned IPC/JEDEC standard, that means reflow soldering with a peak temperature T_p up to $260°C$.

8.7 Handling Instructions

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 sensor can tolerate mechanical shocks up to several thousand g. 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.

We recommend to avoid accelerations beyond the specified limits during transport, handling and mounting of the sensors in a defined and qualified installation process.

This device has built-in protections against high electrostatic discharges or electric fields (e.g., 2kV HBM); however, anti-static precautions should 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.

8.8 Environmental Safety

The BMA530 WLCSP sensor meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also:

RoHS – Directive 2011/65/EU and its amendments, including the amendment 2015/863/EU on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

The BMA530 is halogen-free. For more details on the corresponding analysis results, please contact your Bosch Sensortec representative.

Corresponding chemical analysis certificates are available as separate documents from Bosch Sensortec.

9 Legal Disclaimer

i. Engineering samples

Engineering Samples are marked with an asterisk (*), (E) 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.

ii. 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 safety-critical systems. Safety-critical systems are those for which a malfunction is expected to lead to bodily harm, death or severe property damage. In addition, they shall not be used directly or indirectly for military purposes (including but not limited to nuclear, chemical or biological proliferation of weapons or development of missile technology), nuclear power, deep sea or space applications (including but not limited to satellite technology).

Bosch Sensortec products are released on the basis of the legal and normative requirements relevant to the Bosch Sensortec product for use in the following geographical target market: BE, BG, DK, DE, EE, FI, FR, GR, IE, IT, HR, LV, LT, LU, MT, NL, AT, PL, PT, RO, SE, SK, SI, ES, CZ, HU, CY, US, CN, JP, KR, TW. If you need further information or have further requirements, please contact your local sales contact.

The resale and/or use of Bosch Sensortec 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 accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and to inform Bosch Sensortec without delay of all safety-critical incidents.

iii. 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.

10 Document History and Modifications

Table 46: Change log

Rev No	Chapter	Description of modification/changes	Date
1.0	all	public release	May 14th 2024

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

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

Modifications reserved
Document number: BST-BMA530-DS000-03