BMI330

Small, versatile 6DoF sensor module

BMI330 Datasheet

Document revision	1.1
Document release date	May 6th 2025
Document number	BST-BMI330-DS000-02
Sales part numbers	0 273 017 061
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.

1 Basic Description

The BMI330 is a highly integrated, low power inertial measurement unit (IMU) that combines precise acceleration and angular rate (gyroscopic) measurement with intelligent on-chip motion-triggered interrupt features. The BMI330 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 16 bit digital, triaxial gyroscope with range configurable to $\pm 125\,^{\circ}/\mathrm{s}$, $\pm 250\,^{\circ}/\mathrm{s}$, $\pm 500\,^{\circ}/\mathrm{s}$, $\pm 1000\,^{\circ}/\mathrm{s}$ and $\pm 2000\,^{\circ}/\mathrm{s}$
- a 16 bit digital temperature sensor for an operating temperature range -40 $^{\circ}\mathrm{C}$ ... +105 $^{\circ}\mathrm{C}$

Key Features

- Compact standard size $2.5 \times 3\,\mathrm{mm}^2$ LGA overmold package, 14 pins, height $0.83\,\mathrm{mm}$
- Primary digital interface with 10 $M\mathrm{Hz}$ SPI slave (4-wire, 3-wire), 12.5 $M\mathrm{Hz}$ I3C and up to 1 $M\mathrm{Hz}$ I²C (Fm+)
- Sample rates (ODR): $0.78125\,\mathrm{Hz}$ ... 6.4 $k\mathrm{Hz}$ (nominal)
- Programmable low-pass filtering
- Wide power supply range: analog VDD 1.71 V ... 3.63 V and VDDIO 1.08 V ... 3.63 V, both independent
- Ultra low current consumption: typ. 790 $\mu\mathrm{A}$ (in full ODR and aliasing free operation)
- Built-in power management unit (PMU) for advanced power management and low power modes
- Rapid startup time: 2.5 $m\mathrm{s}$ for accelerometer and gyroscope (fast start mode)
- $< 1\,\mathrm{ms}$ group delay
- 2 KB on-chip FIFO data buffer for accelerometer, gyroscope, temperature, and sensor timestamps
- Fast offset error compensation for accelerometer and gyroscope
- Gyroscope: fast sensitivity error compensation for the gyroscope (max. sensitivity error < 1)
- Hardware synchronization of accelerometer, gyroscope and temperature ($< 1\,\mu\mathrm{s}$)
- Sensor time stamps for accurate system (host) and sensor (IMU) time synchronization ($< 40\,\mu\mathrm{s}$)
- 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 IMU ultra-low power domain
  - motion detection,
  - step detector
  - plug 'n' play step counter
  - orientation and flat detection
  - single tap, double tap, and triple tap detection
- RoHS compliant, halogen and lead free
- Target Applications:
  - GNSS modules
  - Industrial applications
  - Applications requiring extended temperature range up to 105°C

Table of contents

List of figures

List of tables

2 Specification

This chapter provides the specifications for the BMI330. Minimum values and maximum values are provided for standard distributed quantities as $\pm 3\sigma$. 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 or, if the characteristic is specified with respect to temperature, for the temperature range T_P from −40°C to +105°C.

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 domain	V_{DD}		1.71	1.8	3.63	V
Supply voltage I/O domain	V_{DDIO}		1.08	1.2	3.63	V
Voltage input low level	V_{IL}	SPI, I²C & I3C			$0.3 \cdot V_{DDIO}$	V
Voltage input high level	V_{IH}	SPI, I²C & I3C	$0.7 \cdot V_{DDIO}$			V
Voltage output low level	V_{OL}	SPI			$0.2 \cdot V_{DDIO}$	V
Voltage output high level	V_{OH}	SPI	$0.8 \cdot V_{DDIO}$			V
Current consumption	I_{DD}	A+G suspend mode		15		µA
		A_{only} low power mode, $f_{A,lp}$ = 25Hz		45		
		A_{only} high performance mode, $f_{A,hp}$ = max		145		
		A+G high performance mode, $f_{A,hp}$ = $f_{G,hp}$ = max		790		
		A+G normal mode, $f_{A,nm}$ = $f_{G,nm}$ = max		690		
		A+G low power mode, $f_{A,lp}$ = $f_{G,lp}$ = 25Hz		390		
Power on time	Δt_{PO}	Time from supply "on" to serial I/F operational		1.5		ms
Operating temperature	T_A		-40		+105	°C
Accuracy of the output data rate	Δf_A = Δf_G = Δf_T	Any mode with gyroscope enabled in any mode			1.7	%
	$\Delta f_{A,hp,T}$	full T_P range, A_{only} mode, high performance mode			2	%
	$\Delta f_{G,hp,T}$	full T_P range, combo mode, high performance mode		0.0037		$\frac{\%}{K}$

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	$t_{A,SU}$	suspend to high performance mode $f_{A,\cdot}$ = max		2		ms

Table 3: Performance characteristics of the accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Resolution				16		bit
Sensitivity	S_A	a_{FS} = 2g		16384		$\frac{LSB}{g}$
		a_{FS} = 4g		8192		
		a_{FS} = 8g		4096		
		a_{FS} = 16g		2048		
Sensitivity error	$S_{A,err,8g}$	soldered, over life time		±0.5		%
Sensitivity error change over temperature	$\frac{\Delta S_{A,err,8g}}{\Delta T}$ (TCS_A)	full T_P range, best fit straight line		±0.005		$\frac{\%}{K}$
Zero-g offset	O_A	soldered		±35		mg
	$O_{A,life}$	soldered, over life time		±50		mg
Zero-g offset change over temperature	$\frac{\Delta O_A}{\Delta T}$ (TCO_A)	full T_P range, best fit straight line		±0.3		$\frac{mg}{K}$
Noise density	$n_{A,density}$	High performance mode, range 8g		180		$\frac{\mu g}{\sqrt{Hz}}$
Nonlinearity error	$S_{A,NL}$	best fit straight line, a_{FS} = 2g		0.1		%FS
Output data rate	$f_{A,hp}$, $f_{A,n}$	High performance and normal mode	12.5		6400	Hz
	$f_{A,lpm}$	Low-power mode	0.78125		400	
Bandwidth (BW) in high performance and normal mode	$B_{A=12.5Hz}$	0Hz $\leq f \leq f_{3dB\text{-}cutoff}$ of the accelerometer, $B_A = \frac{1}{2} f_A$		6.2		Hz
	$B_{A=25Hz}$			12.4		
	$B_{A=50Hz}$			24.7		
	$B_{A=100Hz}$			49.4		
	$B_{A=200Hz}$			98.8		
	$B_{A=400Hz}$			198		
	$B_{A=800Hz}$			393		
	$B_{A=1600Hz}$			674		
	$B_{A=3200Hz}$			1181		
	$B_{A=6400Hz}$			1677		

Table 4: Mechanical characteristics of the accelerometer

Parameter	Symbol	Condition	Min	Typ	Max	Units
Cross axis sensitivity: Non-orthogonality	$S_{A,YX}$, $S_{A,ZX}$, $S_{A,ZY}$	Non-orthogonality among axes, evaluated as lower triangular matrix		±0.3		%
Cross axis sensitivity: Alignment error	α_A, β_A, γ_A	Relative to package outline		±0.5		°
Zero-g offset over PCB strain	$O_{A,bending}$	soldered[1]		±0.016		$\frac{mg}{\mu strain}$

The Tables 5, 6 and 7 provide the operating conditions for the gyroscope and the corresponding performance and mechanical characteristics.

Table 5: Operating conditions for the gyroscope

Parameter	Symbol	Condition	Min	Typ	Max	Units
Angular rate range	ω_{FS}	Selectable via serial digital interface		±125		$^{\circ}/s$
				±250		
				±500		
				±1000		
				±2000		
Start-up time	$t_{G,SU}$	suspend to high performance mode, $f_{A,hp}$ = max including filter settling		30		ms

[1] Determined with a sample of 5 devices.

Table 6: Performance characteristics of the gyroscope

Parameter	Symbol	Condition	Min	Typ	Max	Units
Resolution				16		bit
Sensitivity	S_G	ω_{FS} = 2000°/s		16.384		$\frac{LSB}{°/s}$
		ω_{FS} = 1000°/s		32.768		
		ω_{FS} = 500°/s		65.536		
		ω_{FS} = 250°/s		131.072		
		ω_{FS} = 125°/s		262.144		
Sensitivity error	$S_{G,err}$	soldered, over life time, after self-calibration		±0.7		%
	$S_{G,err,SC}$	soldered, over life time, without self-calibration		±3		
Sensitivity error change over temperature	$\frac{\Delta S_{G,err}}{\Delta T}$ (TCS_G)	T_P range, best fit straight line		±0.02		$\frac{\%}{K}$
Zero rate offset	$O_{G,over-life}$	Soldered, over life time		±1		°/s
Zero rate offset change over temperature	$\frac{\Delta O_{G,over-life}}{\Delta T}$ (TCO_G)	Best fit straight line for temperature range from -10°C to +85°C		±0.01		$\frac{°/s}{K}$
		Best fit straight line for temperature range from -40°C to <-10°C and from >+85°C to +105°C		±0.02		
Noise density	$n_{G,density}$	High performance mode		0.007		$\frac{°/s}{\sqrt{Hz}}$
Nonlinearity error	$S_{G,NL}$	best fit straight line,ω_{FS} = 2000°/s		0.15		%
Output Data Rate	$f_{G,hp}$, $f_{G,n}$	High performance and normal mode	12.5		6400	Hz
	$f_{G,lpm}$	Low-power mode	0.78125		400	
Bandwidth in high performance and normal mode	$B_{G=12.5Hz}$	0Hz ≤ f ≤ $f_{3dB-cutoff}$ of the gyroscope, B_G = $\frac{1}{2}f_G$		6.2		Hz
	$B_{G=25Hz}$			12.4		
	$B_{G=50Hz}$			24.7		
	$B_{G=100Hz}$			49.4		
	$B_{G=200Hz}$			98		
	$B_{G=400Hz}$			190		
	$B_{G=800Hz}$			345		
	$B_{G=1600Hz}$			450		
	$B_{G=3200Hz}$			531		
	$B_{G=6400Hz}$			563		

Table 7: Mechanical characteristics of the gyroscope

Parameter	Symbol	Condition	Min	Typ	Max	Units
Cross axis sensitivity: Non-orthogonality	$S_{G,YX}$, $S_{G,ZX}$, $S_{G,ZY}$	Non-orthogonality among axes, evaluated as lower triangular matrix		±0.3		%
Cross axis sensitivity: Alignment error	α_G, β_G, γ_G	Relative to package outline		±0.5		°
Zero-rate offset over PCB strain	$O_{G,bending}$	soldered [2]		±1.1		$\frac{^\circ/s}{mstrain}$
Zero rate offset error, gravitation induced	$O_{G,g}$	Gravitation (1g) parallel to each main axis			0.1	$^\circ/s$

Table 8 provides the temperature sensor related characteristics.

Table 8: Characteristics of the temperature sensor

Parameter	Symbol	Condition	Min	Typ	Max	Units
Resolution				16		bits
Measurement Range	T_S		−41		105	°C
Output at 23°C				0		LSB
Sensitivity	S_T			256		$\frac{LSB}{K}$
Temperature offset	O_T	After soldering		±3	±4	K
Temperature sensitivity error		After soldering, T_P		2	4.5	%
Output Data Rate	$f_{T,G}$, $f_{T,combo}$	Any power operation mode with gyroscope in high performance mode or normal mode			50	Hz
	$f_{T,other}$	Any power operation mode with gyroscope either disabled, in low power mode or drive only enabled			12.5	
	$f_{T,A\text{-}only,lp}$	Accelerometer in low power mode, gyroscope disabled			6.25	

[2] Determined with a sample of 5 devices.

3 Absolute Maximum Ratings

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

Table 9: Absolute maximum ratings

Parameter	Condition	Min	Max	Units
Voltage at Supply Pin	V_{DD} Pin	−0.3	4	V
	V_{DDIO} Pin	−0.3	4	V
Voltage at any Logic Pin	Non-Supply Pin-out	−0.3	V_{DDIO} + 0.3 and < 4	V
Passive Storage Temperature Range	≤ 65%rH	−50	150	°C
OTP Non-Volatile Memory Data Retention	T = +105 °C	10		y
Mechanical Shock	MIL-STD-883K Method 2002.5, Condition E		10000	g
	MIL-STD-883K-2 Method 2002.5, Condition F		20000	g
	Free fall onto hard surfaces 6x 1m & 3x 2m		pass	-
ESD	HBM at any pin JESD22-A114F (class 2)		2000	V
	CDM JESD22-C101E (class C2)		500	

4 Quick Start Guide

The purpose of this section is to provide developers, who want to start working with the device, with some very basic hands-on examples to get started for an application. Before starting the test, the device has to be properly connected to the host and powered up. For more information about it, please find details in Chapters 7 and 8.2.

Notes on the Serial Interface Support The communication between application processor and the device will happen over one of the interfaces I3C, I²C or SPI. Each register read operation includes the following number of inserted dummy bytes before the payload:

- I²C: 2
- I3C: 2
- SPI: 1

For simplicity the dummy bytes are not shown in the examples below. For more information about the interfaces and the protocol selection, see Chapter 7.

The device is configured for suspend mode after Power On Reset (POR) or soft reset. In this device operation mode, reading and writing to the registers is possible without requiring to switch the operation mode of the device.

First Application Setup Example Procedures After proper power up by applying the stable supply voltage to the corresponding device pins, the device enters automatically into the 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:

1. Testing communication and initializing the device

a. Reading the chip identifier to ensure correct communication. The default serial interface configuration is I3C and I²C. One initial dummy read configures it to SPI.

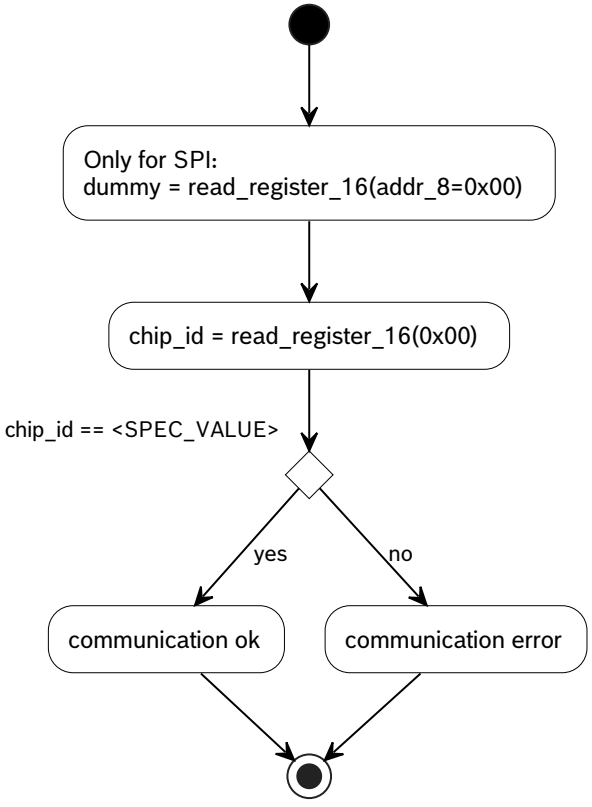


Figure 1: Device communication test

b. Checking the correct initialization status

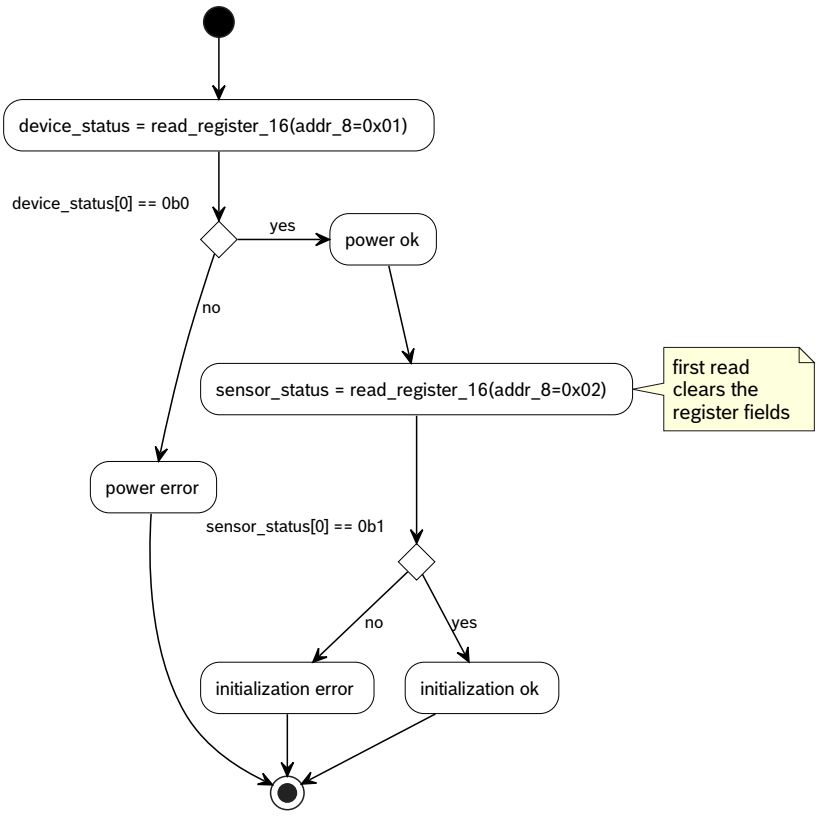


Figure 2: Device initialization status test

2. Configuring the device for I3C communication
Note: for the I3C read operation, two dummy bytes are inserted when reading from registers. For example, in order to read the chip identifier in register 0x00, in total 4 bytes must be read with a burst. The first 2 bytes are dummy bytes, the second two bytes are the chip id register word value where only the least significant byte is valid for the chip id. For more details refer to Section 7.2.4.
About how to issue ENTDAA(0x07) or SETDASA(0x87) command, please refer to MIPI I3C^{SM} Specification. As an example, the SETDASA command shown below in Fig. 3.

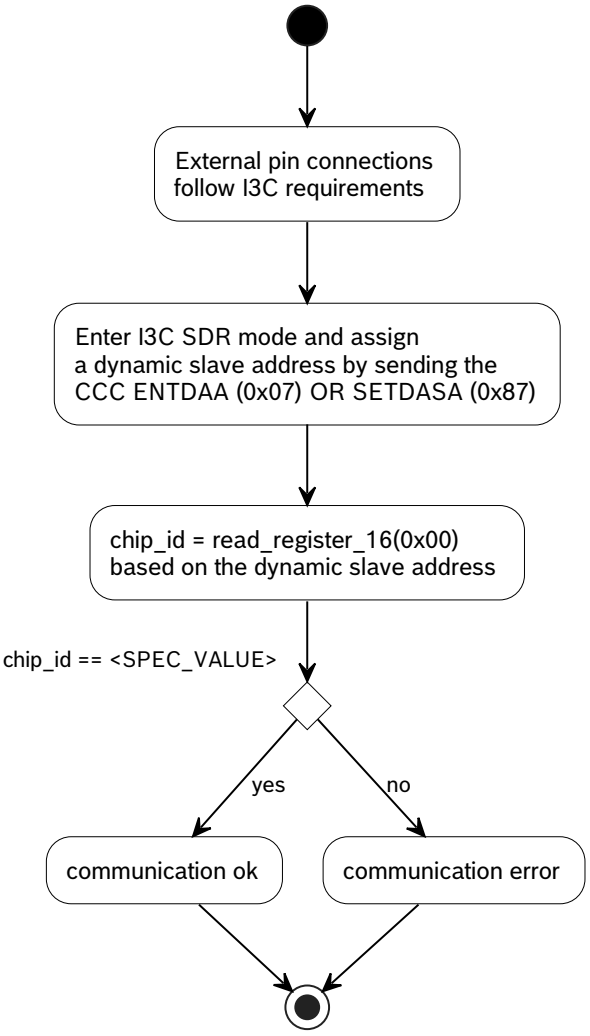


Figure 3: Configure the device for I3C communication

3. Configuring the device for low power mode:
Setting the data processing parameters for power, bandwidth and range followed by reading sensor data

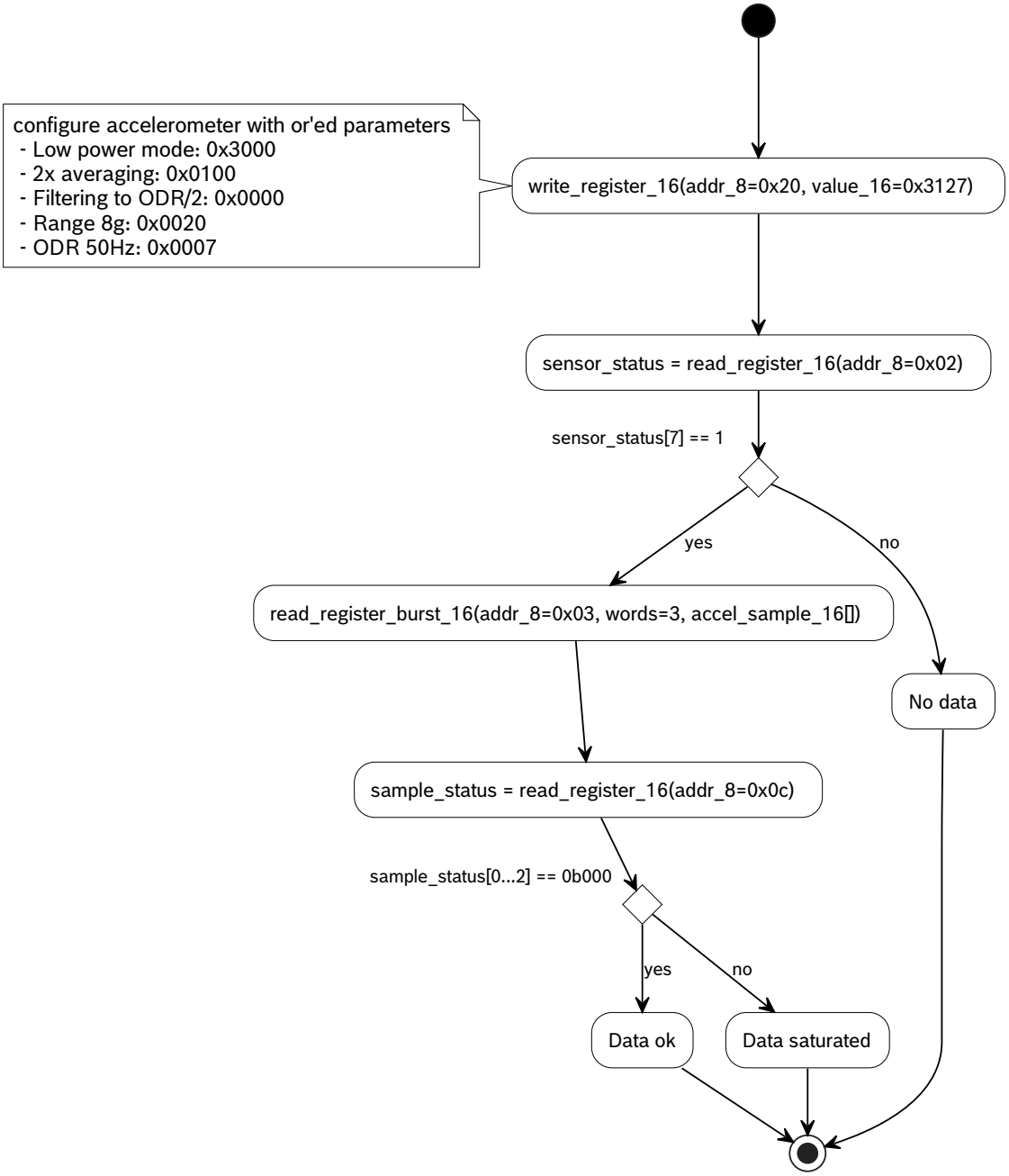


Figure 4: Configure the accelerometer in low power operation mode

4. Configuring the device for normal power mode:
Setting the data processing parameters for power, bandwidth and range followed by reading sensor data

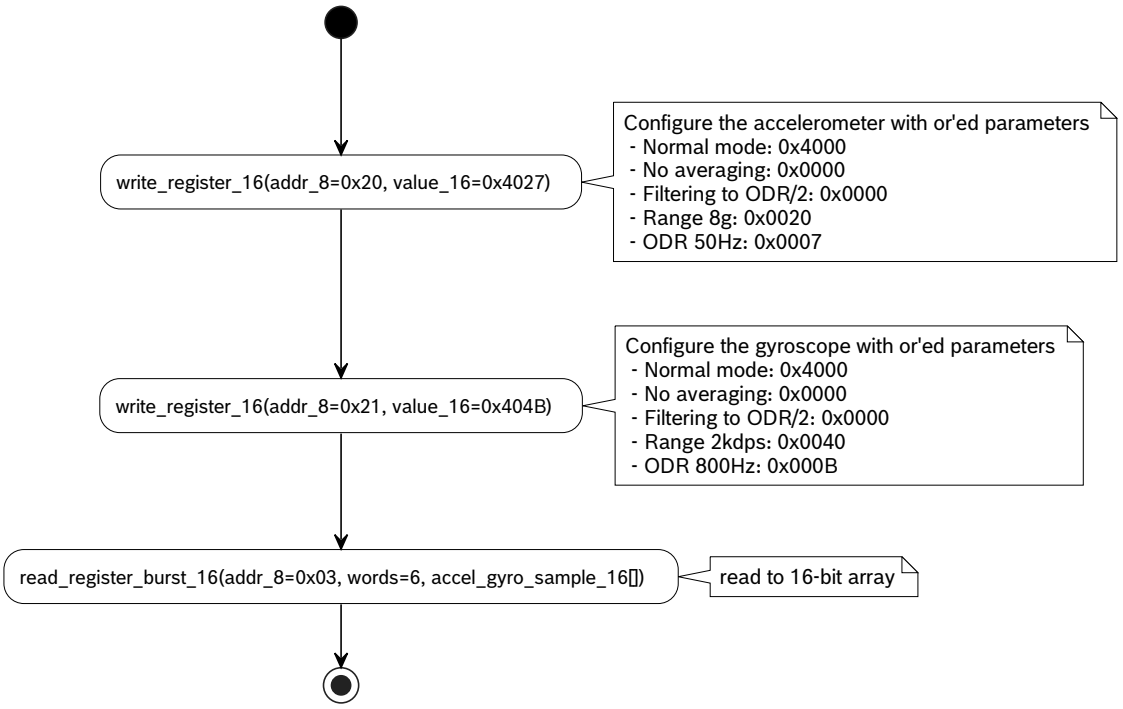


Figure 5: Configure the accelerometer and gyroscope in normal operation mode

5. Configuring the device for performance mode
Setting the data processing parameters for power, bandwidth and range followed by reading sensor data

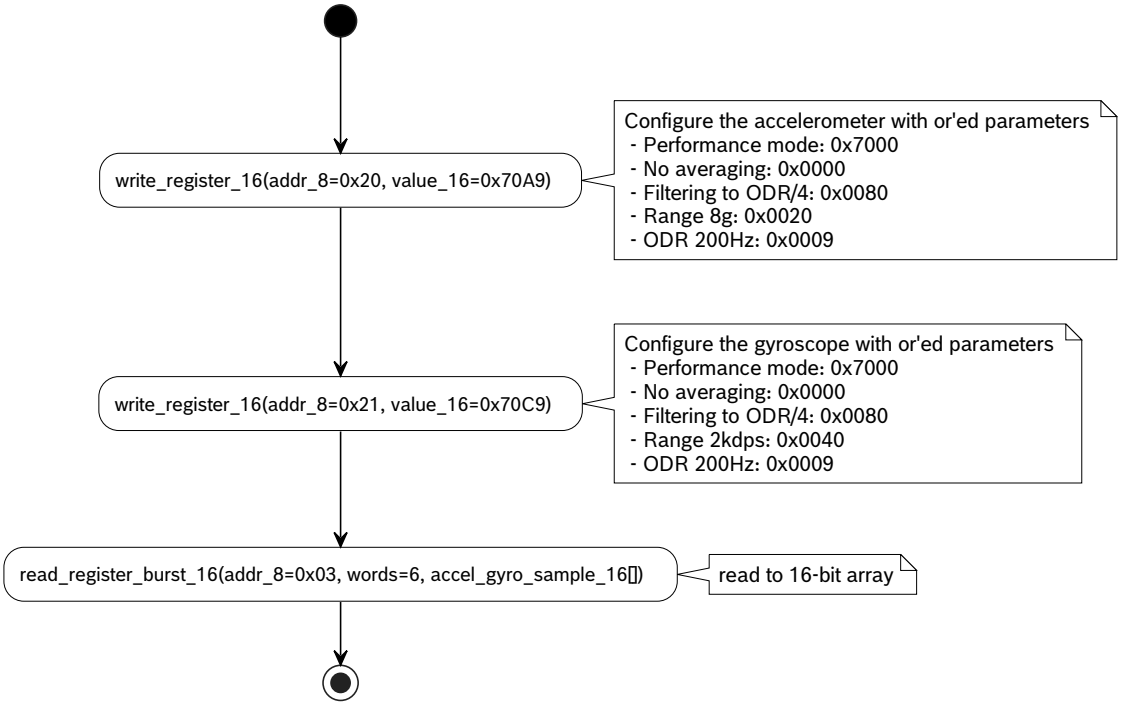


Figure 6: Configure the accelerometer and gyroscope in high performance operation mode

Further Steps: The device has many more capabilities that are described in this document and include FIFO data buffering, power saving modes, synchronization capabilities with the host processor, sample/data synchronization, etc.

Notes on the Data Protocol The host operates the device through the register map with 8-bit addressing to 16-bit data. Data is retrieved from the device by sending one address byte and then reading the required number of dummy bytes followed by two bytes for each register file to be read. If only the least significant byte of an register file is needed, only the first byte has to be read and the second one will be discarded automatically by the device. Data is written to the device by sending one address byte followed two bytes for each register file to be updated. Note: if only one byte is sent, the device will discard the received data and not update the targeted register file. If an odd number of bytes is sent, the device will discard the last received byte and not update the last targeted register file.

For more information about the data protocol and the register map including its special addresses, please refer to the Chapters 6 and 7.

5 Functional Description and Features

This section contains references to the registers of the device. A detailed description of the registers including addresses, bit fields, and values is given in Chapter 6.

5.1 System Configurations

The device includes the two sensors accelerometer and gyroscope. The accelerometer measures the direction and magnitude of the force applied to the sensor. In a free fall scenario, an accelerometer will report a vector of zeros. The gyroscope measures the rotational rate and is reporting a vector zeros when the device is at rest. In addition, the device includes as auxiliary sensor a temperature sensor. All samples are reported synchronously with a unique time value.

5.2 Block Diagram

The Figure 7 details the signal view of the device from analog sensing through analog digital conversion, compensation, filtering and feature computation to the data and the protocol interfaces.

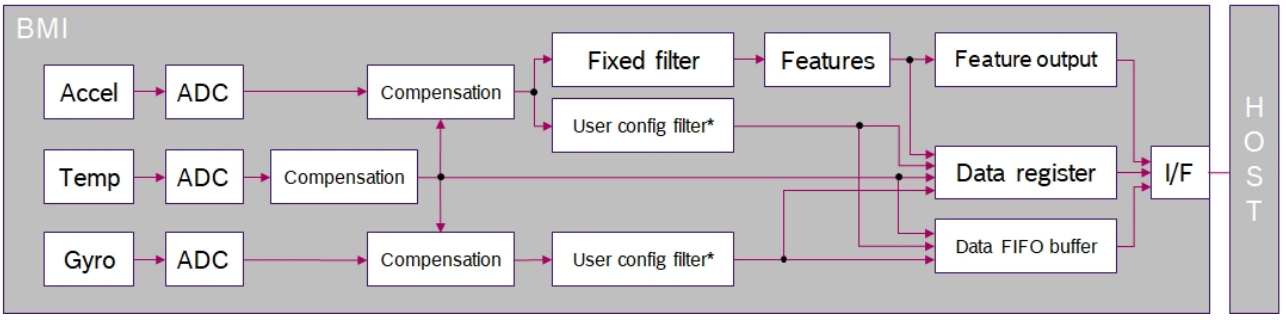


Figure 7: Block Diagram

5.3 Supply Voltage and Power Management

The device has two distinct power supplies:

- the main power supply via the VDD pin with the corresponding GND pin, and
- the digital communication interface driven by a separate power supply via the VDDIO pin with the corresponding GNDIO pin.

There are no limitations with respect to the voltage levels supplied to the VDD and VDDIO pins as long as the absolute minimum and maximum ratings are met, respectively, see Chapter 3 and Section 7.1. Furthermore, the device can be completely switched off, that is $\mathrm{VDD} = 0\,\mathrm{V}$, while keeping the VDDIO supply within the operating range or vice versa. If the VDDIO supply is switched off, all digital interface pins, that is CSB, SDX and SCX, must be kept close to GNDIO potential. The device is reset when the supply voltage provided to at least one of the supply pins VDD or VDDIO drops below the corresponding specified minimum values. No constraints exist for the minimum slew-rate of the voltage applied to the VDD and VDDIO pins.

5.4 Power On Reset and Device Initialization

During Power on Reset (POR), the voltages VDD and VDDIO are ramped to their respective target levels. After the supply voltages reach their target levels, registers are accessible after the start-up time. After every power on reset or soft reset, the device enters the suspend power operation mode. All registers of the device can be read and written without switching the device operation mode.

5.5 Power Operation Modes

The main power operation modes of the device are:

- Suspend mode: lowest possible power consumption, while the device still maintains its configuration. All device registers are accessible at full digital interface speed, see Section 7.1
- Low power mode: motion sensing at lowest possible power consumption
- High performance mode: motion sensing at maximum performance

Table 10 below shows the required configurations for these power operation modes.

Table 10: Power operation modes

		ACC_CONF.acc_mode	GYR_CONF.gyr_mode	Typical current consumption μA
Suspend mode		0b000	0b000	15
Low power mode	Accel only	0b011	0b000	45[3]
	Gyro only	0b000	0b011	355[4]
	IMU	0b011	0b011	390[34]
High performance mode[5]	Accel only	0b111	0b000	145[3]
	Gyro only	0b000	0b111	690[4]
	IMU	0b111	0b111	790[34]
Normal mode[5]	Accel only	0b100	0b000	110[3]
	Gyro only	0b000	0b100	650[4]
	IMU	0b100	0b100	690[34]

The power state of the device is controlled through the configuration of the accelerometer, the gyroscope and the feature engine. The registers `ACC_CONF.acc_mode`, `GYR_CONF.gyr_mode` and `FEATURE_CTRL.engine_en` enable and disable the accelerometer, the gyroscope, the temperature sensor and the feature engine. The registers `ACC_CONF.acc_mode` and `GYR_CONF.gyr_mode` control directly the power state of the sensors. When setting a new configuration, the start-up time of the sensor depends on the prior configuration of the sensor. The performance of the new device configuration does not depend on the prior device configuration.

The sensor characteristics and performance is controlled for the accelerometer via the registers `ACC_CONF.acc_range`, `ACC_CONF.acc_bw` and `ACC_CONF.acc_avg_num` and for the register `GYR_CONF.gyr_range`, `GYR_CONF.gyr_bw` and `GYR_CONF.gyr_avg_num`. This means, that the selected filtering configuration for accelerometer and gyroscope influences the noise power and the latency of the acceleration and angular rate, respectively. In all configurations, register settings as well as FIFO data are retained.

ACC_CONF.acc_mode and GYR_CONF.gyr_mode settings are used to enable and disable the accelerometer, gyroscope and temperature sensing of the device depending on the chosen power mode. After Power on Reset (POR)

[5]The high performance power operation mode is the preferred continuous power operation mode with specification as stated in 2.
[3]Depends on the accelerometer configuration in `ACC_CONF`.
[4]Depends on the gyroscope configuration in `GYR_CONF`.

and soft reset, all sensors of the device are per default disabled. The temperature sensor is automatically enabled and disabled with the either accelerometer or the gyroscope being enabled or disabled. For operation of the device after enabling a sensor, please check before disabling a sensor for the availability of the corresponding sample ready (DRDY) event either by polling the corresponding register or configuring the corresponding interrupt(s). In case the sample ready event is not available after the start-up time for each sensor, the user should check the content of `ERR_REG.fatal_err`. For the gyroscope, the timeout until an error will be reported in `ERR_REG.fatal_err` is max 350 ms. Please note, that a user should check that no error is reported in `ERR_REG.fatal_err` before enabling a sensor.

5.6 Sensor Output

The sensor outputs the three signals acceleration, angular rate and temperature along with the time of the device when the sensors captured the samples. A burst read on the register set of data output from the sensors always provides self contained and therefore consistent values for acceleration, angular rate, temperature and sensor time.

5.6.1 Accelerometer

The three dimensional acceleration data is provided with 16 bits width in two's complement representation. The 16 bits of each axis are available within one 16 bit wide register from `ACC_DATA_X` to `ACC_DATA_Z`. Reading byte-wise the 16 bit acceleration data registers always returns first the least significant byte, then the most significant byte. The default value for each axis is invalid value with 0x8000. Note: the accelerometer axis can be re-mapped affecting the actual content of `ACC_DATA_X`, `ACC_DATA_Y` and `ACC_DATA_Z`. If the device detects an overflow in the signal path, the saturation of acceleration values in the data registers and the FIFO data buffer is reported for each axis individually with `SAT_FLAGS.satf_acc_x`, `SAT_FLAGS.satf_acc_y`, and `SAT_FLAGS.satf_acc_z`.

Accelerometer Power Operation Modes `ACC_CONF.acc_mode` enables the accelerometer with desired power modes or disables the accelerometer. After power-on and soft-reset, the accelerometer is disabled with 0b000. The temperature sensor is automatically enabled and disabled in conjunction the accelerometer with characteristics depending on the accelerometer power mode and, if the gyroscope is enabled, also depending on the gyroscope power mode.

Accelerometer Range Settings The measurement range of the accelerometer can be configured to the ranges $2g$, $4g$, $8g$ and $16g$. After power up and soft reset, the range is per default range $8g$.

Accelerometer Data Processing in High Performance Mode and Normal Mode These two modes can be enabled by setting `ACC_CONF.acc_mode` to 0band self-test 111 for high performance mode and 0b100 for normal mode. The data processing for this mode is configured using `ACC_CONF.acc_bw`. The rate of data available to the host (ODR) can be configured in one of ten different valid ODR configurations going from 12.5 Hz up to 6400 Hz through `ACC_CONF.acc_odr`. Sample rates less than 12.5 Hz are only supported in low power mode. Note: `ACC_CONF.acc_avg_num` has no effect in these modes.

In this power mode, the accelerometer data is output continuously at equidistant points in the time defined by the accelerometer output data rate parameter `ACC_CONF.acc_odr`.

The filtering of the acceleration in these modes is configurable through `ACC_CONF.acc_bw` to either 0b0 for ODR/2 or 0b1 for ODR/4. The filter modes influence the characteristics of the low pass filter applied to the signal, in particular the 3 dB cutoff frequency, the noise suppression, and the group delay. The group delay induced by a filter setting is provided in Table 11. The cut-off frequencies corresponding to the sample rates and filtering are stated in Table 3. The default is 0b0 for ODR/2. Note: the filter settings have no effect when the synchronous timing control mode of I3C is enabled.

Table 11: Accelerometer group delay in high performance mode

Sample rate [Hz]	12.5	25	50	100	200	400	800	1600	3200	6400
Group delay (typ.) [ms]	45.9	23.2	11.8	6.09	3.24	1.82	0.95	0.63	0.47	0.39

Accelerometer Data Processing in Low Power Mode This modes can be enabled by setting `ACC_CONF.acc_mode` to 0b011 for low power mode. The data processing for this mode is configured using `ACC_CONF.acc_avg_num`. The rate of signal available to the host (ODR) can be configured as one of ten different valid sample rate configurations from 0.78 Hz up to 400 Hz through `ACC_CONF.acc_odr`. Sample rates greater than 400 Hz are only supported in normal and high performance mode[5].

In this power mode, the accelerometer alternates between an idle phase, where no measurements are performed, and an active phase, where data is acquired. The data output to the host is the average of all samples acquired during the active phase. The number of averaged samples is configured through `ACC_CONF.acc_avg_num`. A larger number of averaged samples will result in a lower noise level of the signal. Since the active phase is increased, the power consumption will also rise. The period for changing between active and idle mode, also known as duty cycle period, is determined automatically by the value configured through `ACC_CONF.acc_odr` and `ACC_CONF.acc_avg_num`. Valid combinations for these two configurations stated in Table 12. Note: these filter settings have no effect in the time control synchronous mode of I3C is enabled.

Table 12: ASIC internal filter settings for low-power operation mode

ODR, Hz	Internal specification of post-processing by averaging				
	64	32	16	8	4, 2, 1
400	N	N	N	Y	Y
200	N	N	Y	Y	Y
100	N	Y	Y	Y	Y
50	Y	Y	Y	Y	Y
25 ... 0.78	Y	Y	Y	Y	Y

Accelerometer Data Ready Notification The host or another device can be notified about the availability of a new set of sampled data from the accelerometer is available in the registers `ACC_DATA_X` to `ACC_DATA_Z` either

- indirectly by polling the status of the accelerometer data ready in `STATUS.drdy_acc`, or
- directly by an interrupt raised on one of the two (physical) interrupt pins of the in-band-interrupt (IBI) feature of I3C.

Direct notification by an interrupt allows a low latency read of data. To enable the data ready interrupt, please map it via `INT_MAP2.acc_drdy_int` to the desired interrupt interface with 0b01 to the INT1 pin, 0b10 to the INT2 pin or 0b11 to the I3C-IBI. Latching of the interrupt status is configurable via `INT_CONF.int_latch`. In non-latched mode, the interrupt is cleared automatically after $1/6400\,\text{Hz}$. If this automatic clearance is not desired, please configure the latched mode for interrupts, see Section 5.9. In the latched mode, if acknowledgment of interrupts is desired, the flag `INT_STATUS_INT1.int1_acc_drdy`, `INT_STATUS_INT2.int2_acc_drdy` or `INT_STATUS_IBI.ibi_acc_drdy` has to be cleared by reading the status register depending on the mapping of the interrupt to an interrupt interface. The flag `STATUS.drdy_acc` is cleared when any of the registers `ACC_DATA_X` to `ACC_DATA_Z` is read.

Note: a data ready interrupt raised through the INT1 or INT2 pin allows an estimation of the real sample rate when the interrupt data line is also linked to an automatically latched high frequency and high resolution timer on the receiver side of the interrupt.

5.6.2 Gyroscope

The three dimensional angular rate data is provided with 16 bits width in two's complement representation. The 16 bits of each axis are available within one 16 bit wide register from `GYR_DATA_X` to `GYR_DATA_Z`. Reading byte-wise the 16 bit gyroscope data registers always returns first the least significant byte, then the most significant byte. The default value for each axis is invalid value with 0x8000. Note: the gyroscope axis can be re-mapped affecting the actual content of `GYR_DATA_X`, `GYR_DATA_Y` and `GYR_DATA_Z`. If the device detects an overflow in the signal path, the saturation of angular rate values in the data registers and the FIFO data buffer is reported for each axis individually with `SAT_FLAGS.satf_gyr_x`, `SAT_FLAGS.satf_gyr_y`, and `SAT_FLAGS.satf_gyr_z`.

Gyroscope Power Operation Modes GYR_CONF.gyr_mode enables the gyroscope with desired power modes or disables the gyroscope. Per default, the gyroscope is disabled with 0b000. The temperature sensor is automatically enabled and disabled in conjunction with the gyroscope with characteristics depending only the gyroscope power mode. As long as the gyroscope is enabled, the temperature sensor power operation mode is not dependent on the accelerometer power mode, see sub-section 5.6.3.

Gyroscope Range Settings The measurement range of the gyroscope can be configured to the ranges 125 °/s, 250 °/s, 500 °/s, 1000 °/s and 2000 °/s. After power up and soft reset, the default range is 2000 °/s.

Gyroscope Data Processing in High Performance Mode and Normal Mode These two modes can be enabled by setting GYR_CONF.gyr_mode to 0b111 for high performance mode and 0b100 for normal mode[5]. The data processing for this mode is configured using GYR_CONF.gyr_bw. The rate of signal available to the host (ODR) can be configured in one of ten different valid sample rate configurations starting at 12.5 Hz up to 6400 Hz. Sample rates less than 12.5 Hz are only supported in low power mode. Note: GYR_CONF.gyr_avg_num has no effect in these modes.

In this power mode, the gyroscope data is output continuously at equidistant points in the time defined by the gyroscope output data rate parameter GYR_CONF.gyr_odr.

The filtering of the angular rate in these modes is configurable through GYR_CONF.gyr_bw to either 0b0 for ODR/2 or 0b1 for ODR/4. The filter modes influence the characteristics of the low pass filter applied to the signal, in particular the 3 dB cutoff frequency, the noise suppression, and the group delay. The group delay induced by a filter setting is provided in Table 13. The cut-off frequencies corresponding to the sample rates and filtering are stated in Table 6. The default is 0b0 for ODR/2. Note: the filter settings have no effect in the time control synchronous mode of I3C is enabled.

Table 13: Gyroscope group delay in high performance mode

Sample rate [Hz]	12.5	25	50	100	200	400	800	1600	3200	6400
Group delay (typ.) [*ms*]	46.4	23.7	12.3	6.57	3.72	2.30	1.43	1.11	0.95	0.88

Gyroscope Data Processing in Low Power Mode This modes can be enabled by setting GYR_CONF.gyr_mode to 0b011 for low power mode. The data processing for this mode is configured using GYR_CONF.gyr_avg_num. The rate of data available to the host (ODR) can be configured as one of ten different valid ODR configurations from 0.78 Hz up to 400 Hz. Sample rates greater than 400 Hz are only supported in normal and high performance mode[5].

In this power mode, the gyroscope alternates between an idle phase, where no measurements are performed, and an active phase, where data is acquired. The data output to the host is the average of all samples acquired during the active phase. The number of averaged samples is configured through GYR_CONF.gyr_avg_num. A larger number of averaged samples will result in a lower noise level of the signal. Since the active phase is increased, the power consumption will also rise. The period for changing between active and idle mode, also known as duty cycle period, is determined automatically by the value configured through GYR_CONF.gyr_odr and GYR_CONF.gyr_avg_num. Valid combinations for these two configurations are the same as stated for the accelerometer in Table 12 in Subsection 5.6.1. Note: these filter settings have no effect in the time control synchronous mode of I3C is enabled.

Gyroscope Data Ready Notification The host or another device can be notified about the availability of a new set of sampled data from the gyroscope is available in the registers GYR_DATA_X to GYR_DATA_Z either

- indirectly by polling the status of the gyroscope data ready in STATUS.drdy_gyr, or
- directly by an interrupt raised on one of the two (physical) interrupt pins of the in-band-interrupt (IBI) feature of I3C.

Direct notification by an interrupt allows a low latency read of data. To enable the data ready interrupt, please map it via INT_MAP2.gyr_drdy_int to the desired interrupt interface with 0b01 to the INT1 pin, 0b10 to the INT2 pin or 0b11 to the I3C-IBI. Latching of the interrupt status is configurable via INT_CONF.int_latch. In non-latched mode, the interrupt is cleared automatically after $1/6400\,\mathrm{Hz}$. If this automatic clearance is not desired, please configure the latched mode for

interrupts, see Section 5.9. In the latched mode, if acknowledgement of interrupts is desired, the flag `INT_STATUS_INT1.int1_gyr_drdy`, `INT_STATUS_INT2.int2_gyr_drdy` or `INT_STATUS_IBI.ibi_gyr_drdy` has to be cleared by reading the status register depending on the mapping of the interrupt to an interrupt interface. The flag `STATUS.drdy_gyr` is cleared when any of the registers `GYR_DATA_X` to `GYR_DATA_Z` is read.

Note: a data ready interrupt raised through the INT1 or INT2 pin allows an estimation of the real sample rate when the interrupt data line is also linked to an automatically latched high frequency and high resolution timer on the receiver side of the interrupt.

5.6.3 Further Sensor Data

Temperature Sensor The data from the temperature sensor is provided with 16 bits width in two's complement representation in the 16 bit wide register `TEMP_DATA`. Reading byte-wise the 16 bit data registers always returns first the least significant byte, then the most significant byte. The range of the temperature sensor output is from $-41\,^{\circ}\mathrm{C}$ to $+87\,^{\circ}\mathrm{C}$.

The temperature sensor is automatically enabled with enabling either the accelerometer or the gyroscope. It is automatically disabled when both accelerometer and gyroscope are being disabled. The register contains the invalid value 0x8000 until the first temperature measurement is completed. The rate of data available to the host (ODR) depends on the hand on which sensors of the device are enabled and on the other hand to which power mode of a sensor is configured, see Table 14.

Table 14: Temperature sensor sample rate f_T

Operation mode combination	Min	Typ	Max	Unit
Accelerometer in high performance mode or normal mode and gyroscope disabled or in fast start-up mode	12.5		12.5	Hz
Accelerometer in low power mode and gyroscope disabled or in fast start-up mode	$f_{A,\text{low-power}}$		6.25	
Gyroscope in high performance mode or normal mode and accelerometer disabled	50	...	50	
Gyroscope in low power mode and accelerometer disabled	$f_{G,\text{low-power}}$		12.5	
Accelerometer and gyroscope in any of high performance mode or normal mode	50	...	50	
Accelerometer in low power mode and gyroscope in high performance mode or normal mode	50	...	50	
Gyroscope in low power mode and accelerometer in high performance mode or normal mode	12.5		12.5	
Accelerometer in low power mode and gyroscope in low power mode	$\max\{f_{A,\text{low-power}}, f_{G,\text{low-power}}\}$		12.5	

Sensor Time The device provides the time of the sensors synchronized to any sample acquired from the accelerometer, gyroscope and temperature sensor. The value of the timer is provided with 32 bits width in the registers `SENSOR_TIME_1` and `SENSOR_TIME_0`. The most significant word is stored in `SENSOR_TIME_1` and the least significant word in `SENSOR_TIME_0`. The minimum increment of the timer is $39.0625\,\mu s$ per LSB of `SENSOR_TIME_0`. A burst read on the registers always provides self contained and therefore consistent values. The sensor time will wrap around after approximately 11 hours and 39 minutes. Table 15 details the resolution and update rate of the bits in the sensor time register.

Table 15: Resolution and update rate of the sensor time

Bit *m* in SENSOR_TIME_1	31	30	29	28	27	26	25	24
Resolution [s]	res.	res.	20971.52	10485.76	5242.88	2621.44	1310.72	655.36
Sample Rate [Hz]								
Bit *m* in SENSOR_TIME_1	23	22	21	20	19	18	17	16
Resolution [s]	327.68	163.84	81.92	40.96	20.48	10.24	5.12	2.56
Sample Rate [Hz]								
Bit *m* in SENSOR_TIME_0	15	14	13	12	11	10	9	8
Resolution [*ms*]	1280	640	320	160	80	40	20	10
Sample Rate [Hz]	0.78125	1.5625	3.125	6.25	12.5	25	50	100
Bit *m* in SENSOR_TIME_0	7	6	5	4	3	2	1	0
Resolution [μs]	5000	2500	1250	625	312.5	156.25	78.125	39.0625
Sample Rate [Hz]	200	400	800	1600	3200	6400		

5.6.4 Configuration Changes

If the configuration of the device in the registers ACC_CONF or GYR_CONF is altered while the sensors are enabled, the changes are not immediately applied to the sensors. The configuration becomes effective when a sampling event for the currently active ODR coincides with a sampling event for the newly requested ODR on the sensor time sampling grid. In the case where the currently active ODR equals the newly requested ODR, the configuration changes become effective at the next sampling event. This behavior is detailed in the Figure 8.

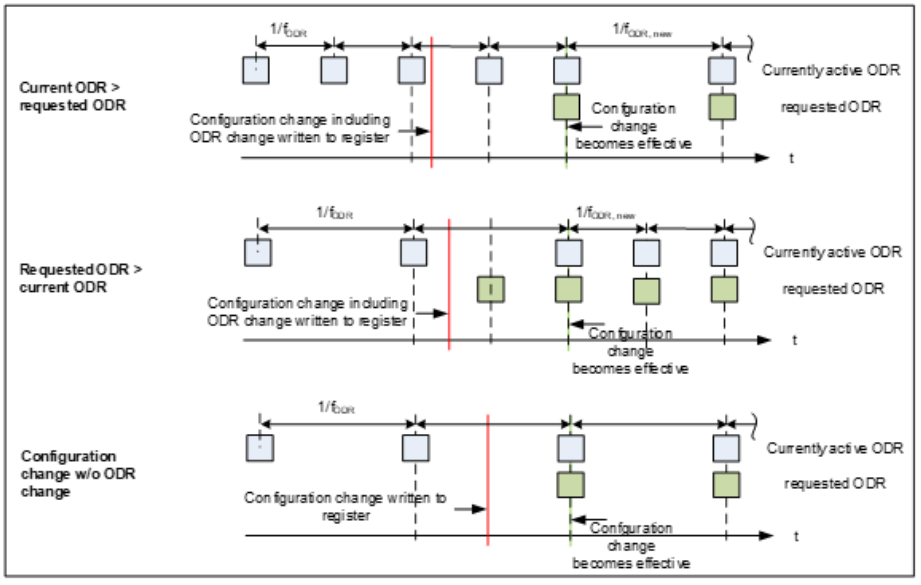


Figure 8: Effectiveness of configuration changes related to sample rate

5.7 FIFO Data Buffering

The FIFO data buffer collects and stores data from selected sensors to minimize transfers between device and host. The device supports the following FIFO operating modes for the FIFO data buffer:

- Streaming mode:
  - Behaviour: overwrite oldest data on buffer full condition

  - Configuration: set `FIFO_CONF.fifo_stop_on_full` to 0b0
- Stop-on-full mode:
  - Behavior: discard newest data on buffer full condition
  - Configuration `FIFO_CONF.fifo_stop_on_full` to 0b1

The FIFO data buffer has a size of 2048 bytes. It can trigger the following interrupts to the host for the conditions of FIFO data buffer filled and FIFO data buffer watermark level reached, see details in section 5.7.3. The FIFO data buffer is enabled to store

- acceleration from the accelerometer with `FIFO_CONF.fifo_acc_en`=0b1,
- angular rate from the gyroscope with `FIFO_CONF.fifo_gyr_en`=0b1,
- temperature with `FIFO_CONF.fifo_temp_en` set to 0b1, and
- sensor time with `FIFO_CONF.fifo_time_en` set to 0b1.

The FIFO data buffer may be used in all power modes of the device to record data. No restrictions for reading the FIFO data buffer from the host apply.

5.7.1 Frames

The data is stored in the FIFO data buffer with frames. The frames have no header. The total width of a buffer frame depends on the signal sources configured for storage in the buffer. When the sources to be stored in the FIFO data buffer are set, the width one frame is constant until the configuration for sources to be stored in the FIFO is changed. Each source adds its own specific width to the total width of a frame. The order of sources for any possible configuration of a buffer frame is fixed, see Table 16. A lower ordering number of a source means that data appears prior in a buffer frame compared to a higher ordering number of a source. Disabled sources do not add any data to the buffer frame. Any change in configuration that have an impact on the width of a buffer frame or the order of sources within a frame causes an instant flush of the FIFO data buffer and a restart of storing signals with the new settings.

Table 16: Order and size of sources to the FIFO data buffer

Order	Source	Size [words]	Description
1	Accelerometer	3	Acceleration with one 16bit word for each axis x, y and z
2	Gyroscope	3	Angular rate with one 16bit word for each axis x, y and z
3	Temperature sensor	1	Temperature as one 16bit word
4	Oscillator	1	Sensor time as one 16bit word

Example: gyroscope, time, and acceleration is enabled, then the order within the buffer frame is (accel:gyro:time)

The format of the fields in a FIFO data frame is stated in Table 17.

Table 17: FIFO Data Description

Frame Field	FIFO Data Description
Accelerometer	Same as configured in `ACC_CONF.acc_range`
Gyroscope	Same as configured in `GYR_CONF.gyr_range`
Temperature	See `TEMP_DATA.temp_data`
Sensortime	Same resolution as sensortime, see 5.6.3

Changes to the configuration of the FIFO data buffer can cause a flush of the buffer, see details in subsection 5.7.4. A change of the configuration of the sensor, e.g. measurement range, ODR, etc, will restart the signal processing in the device causing invalid data until the data path is settled. The invalid data is not inserted into the buffer frames and skipped automatically. Instead, to keep the width of a frame constant, special dummy-data is inserted. This dummy-data can also be used to distinguish frames with old and new settings in place. The dummy data has a fixed signature for each signal source as shown in Table 18.

Table 18: Signature of dummy frames in the FIFO data buffer

Word	Accelerometer	Gyroscope	Temperature
1	0x7f01	0x7f02	0x8000
2...end	0x8000	0x8000	n/a

5.7.2 Conditions and Details

Enabling FIFO Data Buffering The FIFO data buffer has to be enabled before enabling any sensor, that is the accelerometer or the gyroscope. If the FIFO data buffering was not enabled before and is needed in any low power mode, the device must be switched to high performance mode or normal mode to activate then the FIFO data buffer and switch back to the low power mode.

Buffer Frame Reads A buffer frame is deleted from the FIFO data buffer after being fully read through the register `FIFO_DATA`. If a frame is read only partially, it will be repeated completely with the next read of the FIFO data buffer. In the case of a buffer overflow between the first partial read and the second read attempt, the frame is kept only if `FIFO_CONF.fifo_stop_on_full` is set to 0b1.

Buffer Overreads When more data is read from the FIFO buffer than valid data is available, a value of 0x8000 is returned for each word read exceeding the valid data.

Frame Rates The sampling rate of buffer frames in the FIFO buffer is defined by the maximum output data rate of the sensors enabled for storage in the FIFO buffer.

Buffer Overflow In case of an overflow, the FIFO buffer can either stop recording data or overwrite the oldest data. This behavior is controlled by the register `FIFO_CONF.fifo_stop_on_full`. If `FIFO_CONF.fifo_stop_on_full` is set to 0b0, the FIFO logic will delete the oldest frames. If `FIFO_CONF.fifo_stop_on_full` is set to 0b1, the newest frame may be discarded when the remaining free space in the FIFO buffer is less than the maximum size frame. During a read operation from the FIFO buffer by the host, no data at the FIFO buffer tail may be dropped. If the host reads the FIFO buffer with a slower rate than it is filled, it may happen that the sensor needs to drop new data, even if `FIFO_CONF.fifo_stop_on_full` is set to 0b0.

5.7.3 FIFO Buffer Interrupts

The FIFO supports the two interrupts

- buffer full interrupt, and
- watermark level interrupt.

The buffer full interrupt is issued when the FIFO fill level is above the full threshold. The full threshold is reached just before the last two frames are stored in the FIFO data buffer. This interrupt is enabled by setting `INT_MAP2.fifo_full_int` to 0b1.

The watermark level interrupt is issued when the fill level of the FIFO buffer is equal or above a watermark level defined in the register `FIFO_FILL_LEVEL.fifo_fill_level`. Note: if the watermark level is set higher than the full level, the interrupt may be triggered more often than expected. This interrupt is enabled by setting `INT_MAP2.fifo_watermark_int` to 0b1.

The FIFO buffer interrupts can be signalled to host after mapping with `INT_MAP2.fifo_watermark_int` and/or `INT_MAP2.fifo_full_int` them to the desired interrupt signalling channel. Latched FIFO buffer interrupts will only be cleared, if the status register gets read and the fill level is below the corresponding FIFO interrupt.

5.7.4 FIFO Buffer Flush

A flush of the FIFO data buffer can be triggered by the user directly or indirectly by a re-configuration of the device. The user can directly trigger a flush of the FIFO buffer by writing 0b1 to `FIFO_CTRL.fifo_flush`. An indirect, automated flash of the FIFO data buffer is caused by:

- a signal source being enabled or disabled, and
- a signal source to be stored in the FIFO data buffer being added or removed.

Changes of the configuration, that do not cause a flush of the FIFO data buffer, are:

- measurement range,
- sample rate of providing data to the host (ODR), and
- filter configuration.

5.8 Advanced Features

5.8.1 Global Configuration

The advanced features of the device can be activated after enabling the feature engine. The feature engine has to be enabled following the steps also detailed in Fig. 9:

1. disable all sensors except enabling the feature engine directly after power on or soft reset, then
2. first writing 0x012C to `FEATURE_IO2` followed by 0x0001 to `FEATURE_IO_STATUS`, then
3. setting `FEATURE_CTRL.engine_en` to 0b1, and then
4. waiting for the feature engine to be initialized by polling `FEATURE_IO1.error_status` for the value 0b001.

Note: the advanced features may be incorrectly interpreted if the accelerometer and/or gyroscope are enabled before feature engine.

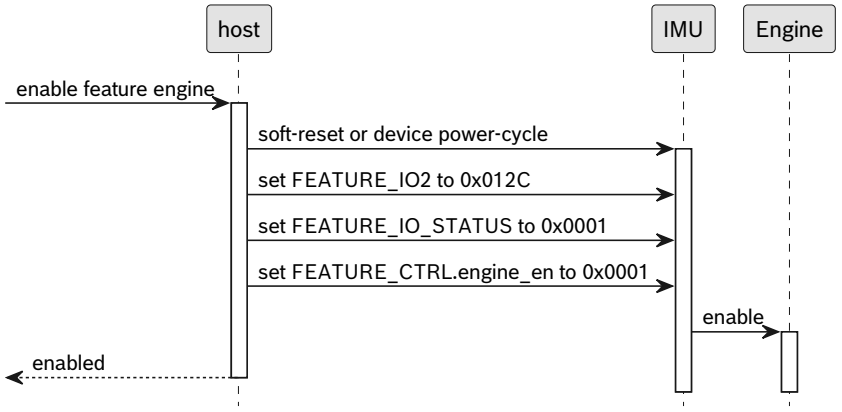


Figure 9: Enable the feature engine

The feature engine may be disabled by setting `FEATURE_CTRL.engine_en` to 0b0, however, a soft reset or power cycle is required to re-enable the feature engine. Configuration, state toggling and the data input/output of the features is possible via

- the register interface through the registers `FEATURE_IO0` to `FEATURE_IO3` and the input/output synchronization register `FEATURE_IO_STATUS` for
  - enabling and disabling of features and
  - data and status output of advanced features, or
- the command interface via the register `CMD` for initiating non-continuously running features, and

- the configuration interface for features through the registers FEATURE_DATA_ADDR, FEATURE_DATA_TX and FEATURE_DATA_STATUS, see Section 6.2.

The configuration of the sensor through FEATURE_IO0 must be executed following this procedure:

- write or modify the configuration to FEATURE_IO0, and subsequently
- activate the configuration by writing 0b1 to FEATURE_IO_STATUS.feature_io_status.

Note: prior to enabling of any of the advanced features, the configuration and enabling of the accelerometer is required. A more enhanced/flexible user interface can be configured by using the function bmi3x0_configure_enhanced_flexibility of the sensor driver API, available in Github.

The advanced features can be optimized depending on the context of the end application. The pre-defined settings in the API are optimized for a mobile context, but can also be switched to wearable or hearable context via the API. The sensor API is available in GitHub.

Table 19 provides the overview on supported sample rates per feature in the low power operation mode of the accelerometer. In the high performance power operation mode, features are supported independent of the configured sample rates of the accelerometer.

<table>
<tr><th>Acceleration sample rate</th><th>Any motion / Motion Detect</th><th>No motion / station-ary detect</th><th>Step counter</th><th>Significant motion</th><th>Tap detector</th><th>Orientation detector</th><th>Flat detector</th><th>Tilt detector</th></tr>
<tr><td>> 200</td><td rowspan="6">Yes</td><td rowspan="6">Yes</td><td rowspan="4">Yes</td><td rowspan="4">Yes</td><td>Yes</td><td rowspan="6">Yes</td><td rowspan="6">Yes</td><td rowspan="6">Yes</td></tr>
<tr><td>200</td><td rowspan="5">No</td></tr>
<tr><td>100</td></tr>
<tr><td>50</td></tr>
<tr><td>25</td><td rowspan="2">No</td><td rowspan="2">No</td></tr>
<tr><td>12.5</td></tr>
</table>

Table 19: Overview of supported sample rates per feature in the low power operation mode

The device adjusts the sample rate dependent parameters of the features to the configured reduced sample rate. Threshold values are not dependent on the sample rate and therefore not modified. All user configurations are retained and not modified until soft-reset or power down. Hence, you do not need to update the configuration when changing the sample rate (ODR) of the accelerometer.

5.8.2 Any-motion Detection and Motion Detect

The feature any-motion detects changes in motion of one axis. It uses the slope between adjacent samples of the acceleration signal to trigger this event. With a different behaviour in interrupt frequency and interrupt hold time, this feature can be configured to the generate an interrupt for motion detect with EXT.GEN_SET_1.event_report_mode set to 0b1. The interrupt is enabled for each axis independently by writing 0b1 to FEATURE_IO0.any_motion_x_en, FEATURE_IO0.any_motion_y_en, and FEATURE_IO0.any_motion_z_en.

The difference between the current acceleration sample and the reference sample gives the slope. The computation of the slope depends on the setting of EXT.ANYMO_1.acc_ref_up. In the default configuration of EXT.ANYMO_1.acc_ref_up being 0b1, the reference sample is the previous acceleration sample. When EXT.ANYMO_1.acc_ref_up is set to 0b0, the reference sample is set to the acceleration sample when the interrupt was generated.

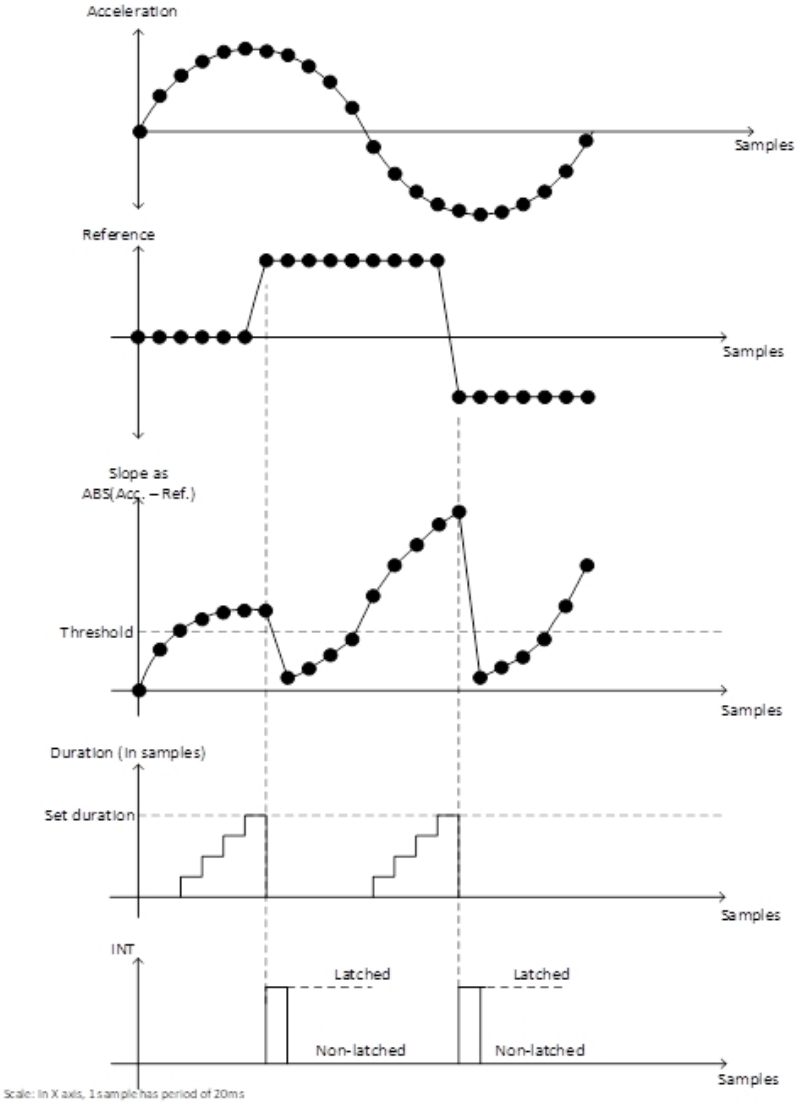


Figure 10: Motion detect / Any-motion detection

The feature generates an interrupt when the absolute value of the slope exceeds the preset `EXT.ANYMO_1.slope_thres` for a certain number of consecutive samples defined by `EXT.ANYMO_3.duration`. The reference sample is updated when the condition is fulfilled. In other words, this means that the reference for the detection is the last state when sensor detected an any-motion event. The generated interrupt will be cleared as soon as the slope value drops below the threshold and after the duration `EXT.ANYMO_3.wait_time` elapsed. The sensitivity of this interrupt to be rethrown when the slope increases again after dropping below the threshold is controlled with `EXT.ANYMO_2.hysteresis`. 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 12.5 Hz to the maximum supported sample rate, see Table 19.

Configuration settings

1. `FEATURE_IO0.any_motion_x_en` – enable the feature for the x-axis
2. `FEATURE_IO0.any_motion_y_en` – enable the feature for the y-axis
3. `FEATURE_IO0.any_motion_z_en` – enable the feature for the z-axis
4. `EXT.ANYMO_1.slope_thres` – the threshold for the slope
5. `EXT.ANYMO_1.acc_ref_up` – update mode of the acceleration reference when an event was detected or always

6. `EXT.ANYMO_2.hysteresis` – hysteresis for the slope of the acceleration
7. `EXT.ANYMO_3.duration` – the number of consecutive samples for which the threshold condition must be respected for interrupt assertion
8. `EXT.ANYMO_3.wait_time` – wait time for clearing the event after the slope is below the configured threshold value `EXT.ANYMO_1.slope_thres`

5.8.3 No-motion Detection and Stationary Detect

The feature no-motion detection evaluates the slope between adjacent samples of the acceleration signal in all selected axes to detect a stationary state. The interrupt is enabled for each axis independently by writing 0b1 to `FEATURE_IO0.no_motion_x_en`, `FEATURE_IO0.no_motion_y_en`, and `FEATURE_IO0.no_motion_z_en`. With a different behaviour in interrupt frequency and interrupt hold time, this feature can be configured to the generate an interrupt for stationary detect with `EXT.GEN_SET_1.event_report_mode` set to 0b1.

The difference between the current acceleration sample and the reference sample gives the slope. An interrupt is generated when the slope for all selected axis remains smaller than the programmable threshold value `EXT.NOMO_1.slope_thres` for the configurable duration value in `EXT.NOMO_3.duration`. 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 12.5 Hz to the maximum supported sample rate, see Table 19.

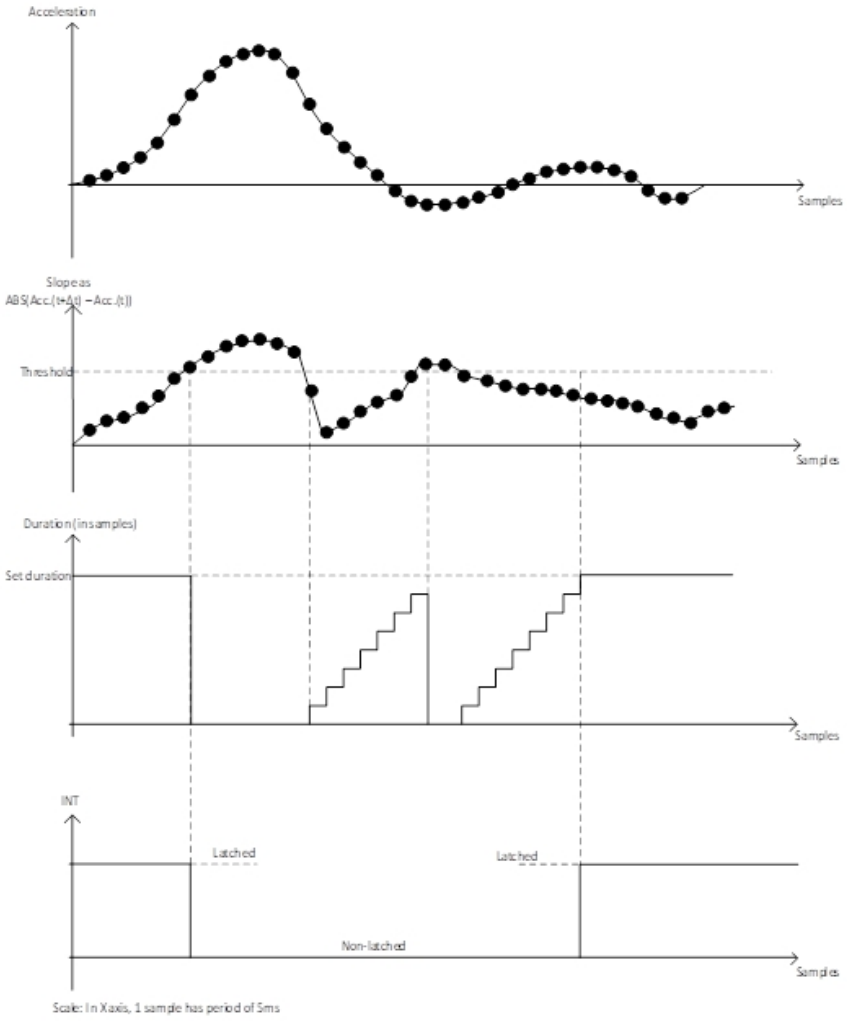


Figure 11: Stationary detect / No-motion detection

Configuration Settings

1. `FEATURE_IO0.no_motion_x_en` – enable the feature for the x-axis
2. `FEATURE_IO0.no_motion_y_en` – enable the feature for the y-axis
3. `FEATURE_IO0.no_motion_z_en` – enable the feature for the z-axis
4. `EXT.NOMO_1.slope_thres` – the threshold for the slope
5. `EXT.NOMO_1.acc_ref_up` – update mode of the acceleration reference when an event was detected or always
6. `EXT.NOMO_2.hysteresis` – hysteresis for the slope of the acceleration
7. `EXT.NOMO_3.duration` – the number of consecutive samples for which the threshold condition must be respected for interrupt assertion
8. `EXT.NOMO_3.wait_time` – wait time for clearing the event after the slope is below the configured threshold value `EXT.NOMO_1.slope_thres`

5.8.4 Significant Motion Detection

The significant motion detection provides the interrupt raised on detection of a significant motion as defined in Android[1]: https://source.android.com/devices/sensors/sensor-types.html#significant_motion.

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 19. It can be enabled by setting `FEATURE_IO0.sig_motion_en` to 0b1.

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. Classification of movement as significant motion or not is based on the analysis of acceleration signal over the time duration configured by `EXT.SIGMO_1.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. The condition is based on the peak-to-peak (P2P) value and mean crossing rate (MCR) of the magnitude of the acceleration signal. The conditions for a significant motion are:

- P2P magnitude is greater than `EXT.SIGMO_2.peak_2_peak_min` and MCR is greater than `EXT.SIGMO_2.mcr_min`, or
- P2P magnitude is greater than `EXT.SIGMO_3.peak_2_peak_max` and MCR is less than `EXT.SIGMO_3.mcr_max`.

The feature can be configured to output only the first detected event and ignoring the following events, also known as one-shot behavior, by setting `EXT.GEN_SET_1.event_report_mode` as 0b1. When this configuration is enabled, internally the block size is set to 5 seconds and changes to the user configurable value of `EXT.SIGMO_1.block_size` are ignored.

Example An example for the behavior of the significant motion detection for a walking scenario is depicted in Fig. 12, where `EXT.SIGMO_1.block_size` is set to 5 seconds (0x00FA).

[1] Android is a trademark of Google LLC.

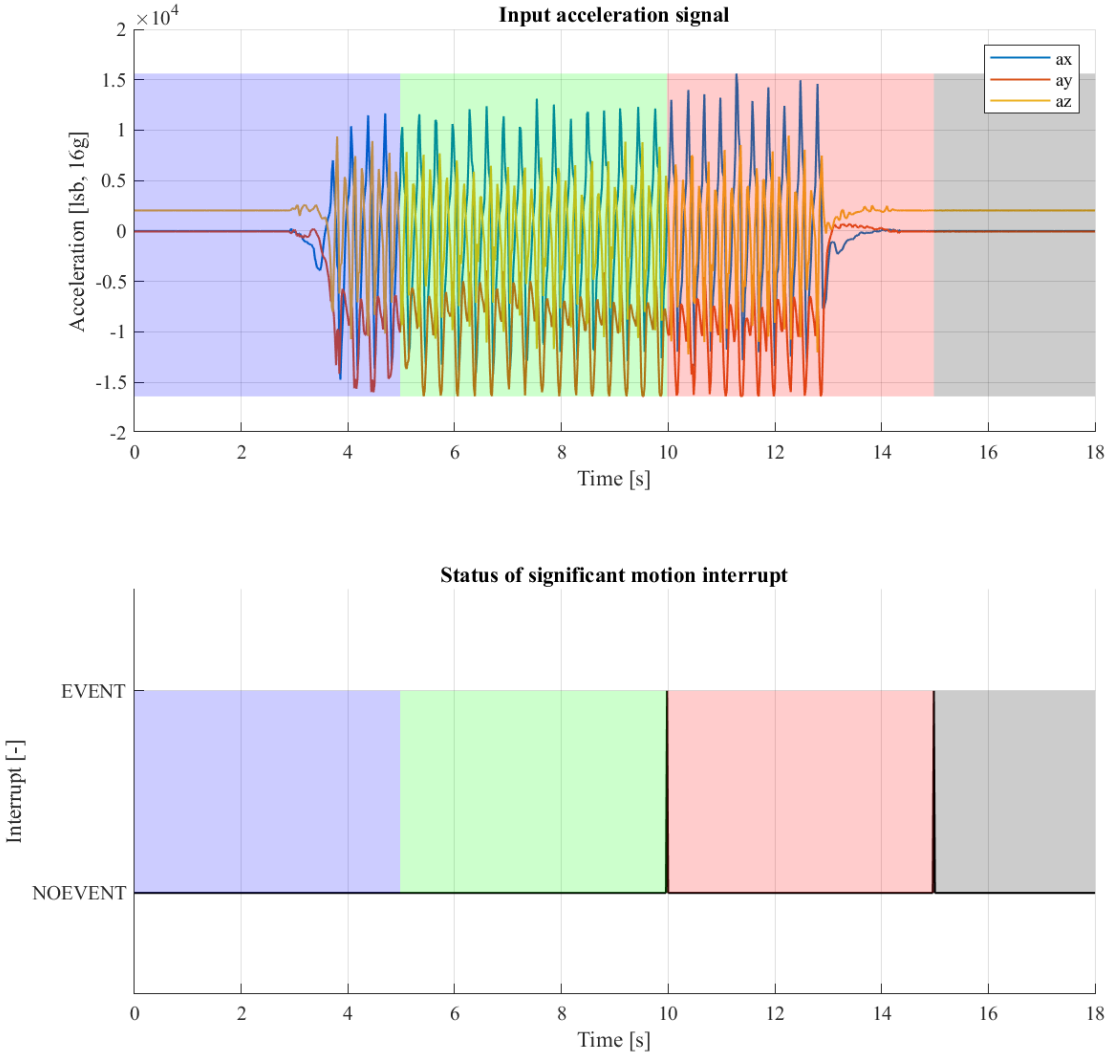


Figure 12: Significant motion interrupt detection behavior for walking use-case

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.
2. Full segment as WALKING.
3. Initial large part of segment as WALKING with remaining being STILL.

Segment 1 encompasses the user movement for less than 50% of `EXT.SIGMO_1.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.

Configuration Settings Significant motion offers configuration parameters for enabling/disabling of feature and to optimize the functional behavior based on use-case. Parameters available are described in Table 20.

Table 20: Configuration parameters of the significant motion

Register/field name	Description	Default value	Range
FEATURE_IO0.sig_motion_en	Enable detection of significant motion status of device	0	0/1 (disable/enable)
EXT.SIGMO_1.block_size	Size of the time segment for significant motion detection	250	0 to 65535
EXT.SIGMO_2.peak_2_peak_min	Minimum threshold for peak to peak value of acceleration magnitude over one second time window	38	0 to 1023
EXT.SIGMO_2.mcr_min	Minimum threshold for mean crossings of acceleration magnitude over one second time window	17	0 to 63
EXT.SIGMO_3.peak_2_peak_max	Maximum threshold for peak to peak value of acceleration magnitude over one second time window	595	0 to 1023
EXT.SIGMO_3.mcr_max	Minimum threshold for mean crossings of acceleration magnitude over one second time window	17	0 to 63

5.8.5 Step Counter and Step Detection

The Step Counter provides the function required for counting of steps as defined for Android[2]: https://source.android.com/devices/sensors/sensor-types.html#step_counter. The Step Detector implements the function required for step counting in Android: https://source.android.com/devices/sensors/sensor-types.html#step_detector. The algorithm for counting of steps is designed for smartphone usecases and optimized on high accuracy, while the algorithm for the detection of steps is optimized for low latency reporting of detection events. 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 19.

Configuration Settings

1. FEATURE_IO0.step_counter_en – indicates if the step counter feature is enabled or not.
2. FEATURE_IO0.step_detector_en – indicates if the step detector feature is enabled or not.
3. FEATURE_IO2.step_counter_out_0 – Step counter value, lower 16bit of the 32bit value
4. FEATURE_IO3.step_counter_out_1 – Step counter value, higher 16bit of the 32bit value
5. EXT.SC_1.watermark_level – the Step Counter will trigger output every time this number of steps are counted. Holds implicitly a 20x factor, so the range is 0 to 1023 (without the implicit factor), with resolution of 20 steps. If 0, the Step Counter watermark is disabled. If the Step Detector is enabled, the watermark interrupt is disabled (as being mutually exclusive).
6. EXT.SC_1.reset_counter – trigger to reset the counted steps. Resets the step count value, if any one of step counter, step detector or activity feature is enabled.

Step Counter The Step Counter accumulates the steps detected by the step detector interrupt, and provides the current value of the 32bit wide step count in the two registers FEATURE_IO2.step_counter_out_0 (lower word) and FEATURE_IO3.step_counter_out_1 (higher word). By setting the flag EXT.SC_1.reset_counter to 0b1, the value of accumulated steps is reset. Afterwards, the value of this flag is automatically reset and counting is restarted. The accumulated step count value can be reset when any of the features Step Counter or Step Detector is enabled.

The watermark option can be useful if the host needs to receive an interrupt every time a certain number of steps occured. When the watermark level is reached, the corresponding interrupt bit is asserted for the mapped interrupt channel, that is

[2] Android is a trademark of Google LLC.

INT_STATUS_INT1.int1_step_counter, INT_STATUS_INT2.int2_step_counter or INT_STATUS_IBI.ibi_step_counter. If EXT.SC_1.watermark_level is set to 10 (holding an implicit factor of 20x), after every interval of 200 steps an interrupt will be raised. As the steps are buffered internally, the output may be triggered between 200 to 210 steps.

Note: in case the watermark interrupt for the step counter is required for the application to be set to a level greater than 65535, an upload of the configuration file is mandatory before enabling the feature engine and executing this feature, see section 5.8.1. The configuration file is provided through the sensor API, available in GitHub, and can be extracted from there. Please refer to the function bmi3x0_configure_enhanced_flexibility of the sensor driver API for a reference implementation.

Step Detector When FEATURE_IO0.step_detector_en is set to 0b1, an interrupt is triggered for every detected step. Every time a new step is detected, the configured corresponding interrupt output is triggered and the corresponding status bit is set. The Step Detector feature is optimized for low latency to ensure for such events a fast reaction by the host . Hence, when a step is detected, it is immediately signaled. Due to this behaviour, there may exist situations when the sum of the detected steps is different than the Step Counter value.

5.8.6 Flat Detection

This interrupt detects a Flat orientation based on the acceleration signal. The interrupt is triggered when the sensor in the device gets close to a horizontal orientation. Note: the interpretation of the sensor orientation as horizontal is dependent on the configured axis re-mapping, see section 5.11. A horizontal orientation is detected by the angle among the orientation of the gravitational force and the axis of the sensor configured as z-axis. The condition for activating the interrupt reads

$$\Theta \cdot a_z \cdot a_z - \Delta a_{\text{hysteresis}} \geq a_x \cdot a_x + a_y \cdot a_y \,. \tag{5.1}$$

The condition for deactivating the interrupt is given by

$$\Theta \cdot a_z \cdot a_z + \Delta a_{\text{hysteresis}} < a_x \cdot a_x + a_y \cdot a_y \,. \tag{5.2}$$

If one of the inequalities does not become true, then the state of interrupt is not changed. The state of the interrupt is actually changed, that is set or reset, when the device remains in one of the conditions for the configured period EXT.FLAT_1.hold_time. 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 12.5 Hz to the maximum supported sample rate, see Table 19.

Theta Angle The threshold angle Θ for detecting a Flat orientation can be configured via EXT.FLAT_1.theta where 1 LSB equals $64 \cdot (\tan(\Theta))^2$. Some important values of EXT.FLAT_1.theta are depicted in Table 21. The value of the hysteresis value EXT.FLAT_2.hysteresis is set according to the following graphic, with values between 0 and 63, which corresponds to hysteresis angle between 0 and 5 degrees. In the following graphic, 4 usual cases are depicted: 0, 1, 2.5 and 5 degrees. The hysteresis is symmetric, used for both going into and out of Flat state. For the default value of 9, the actual interval around the angle of 20 degrees is +/-2.5 degrees; so a 5 degree interval is used for total hysteresis filtering.

Table 21: Register value correspondence to Θ in degrees

EXT.FLAT_1.theta	**0**	**1**	**2**	**5**	**8**	**14**	**22**	**33**	**45**	**63**
Θ/deg	0	7.1	10	15.6	19.5	25.1	30.4	35.7	40	44.8

Hysteresis The threshold angle Θ for Flat detection is associated to a hysteresis to ensure a true flat detection and avoid the oscillation the feature status. The hysteresis value EXT.FLAT_2.hysteresis is used according to Fig. 13. The hysteresis can be configured with values between 0 and 63 that correspond to a hysteresis angle between 0 and 5 degrees. In the following graphs of Fig. 14 and Fig. 15, four typical cases are depicted for the case of the hysteresis configured to 0, 1, 2.5 and 5 degrees. The effect of the hysteresis is symmetric and used for both cases of entering

and lesving the Flat state. For the default hystereis configuration value of 9, the actual interval around the Theta angle Θ = 20 deg is $\pm$2.5 deg. Hence, a total interval for the hysteresis filtering is 5 deg.

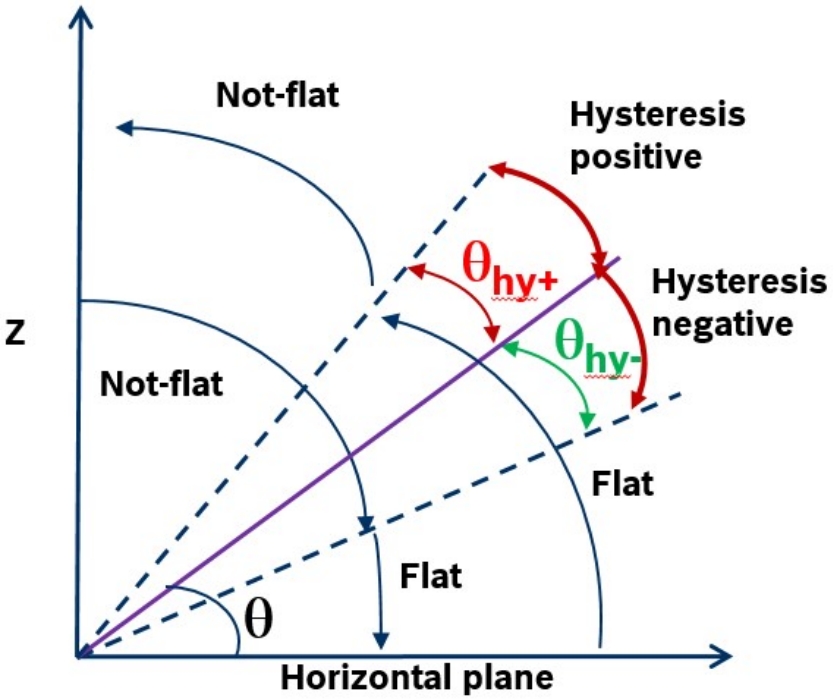


Figure 13: Hysteresis and θ angle

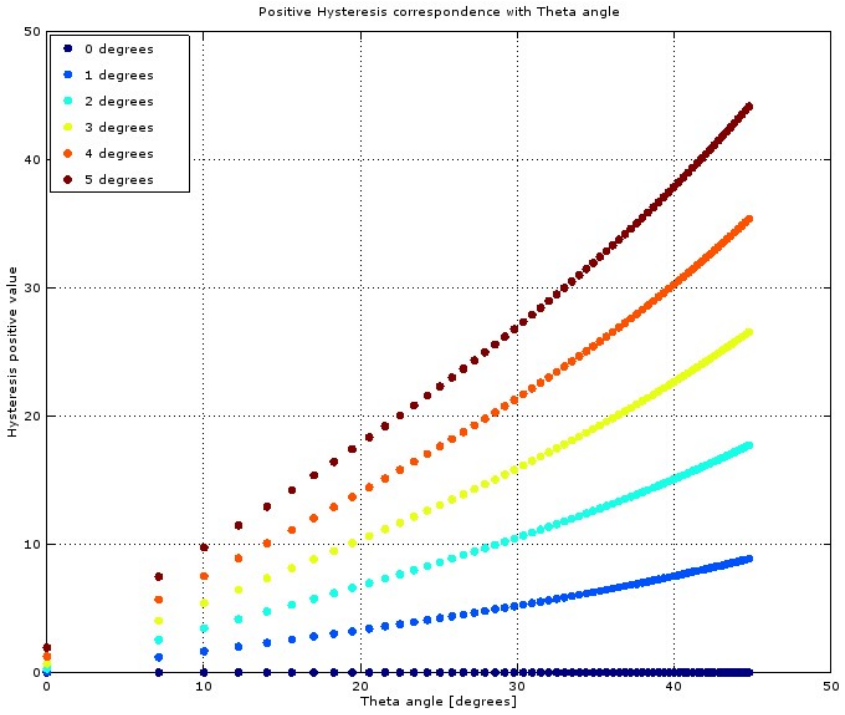


Figure 14: Positive hysteresis correspondence to θ angle

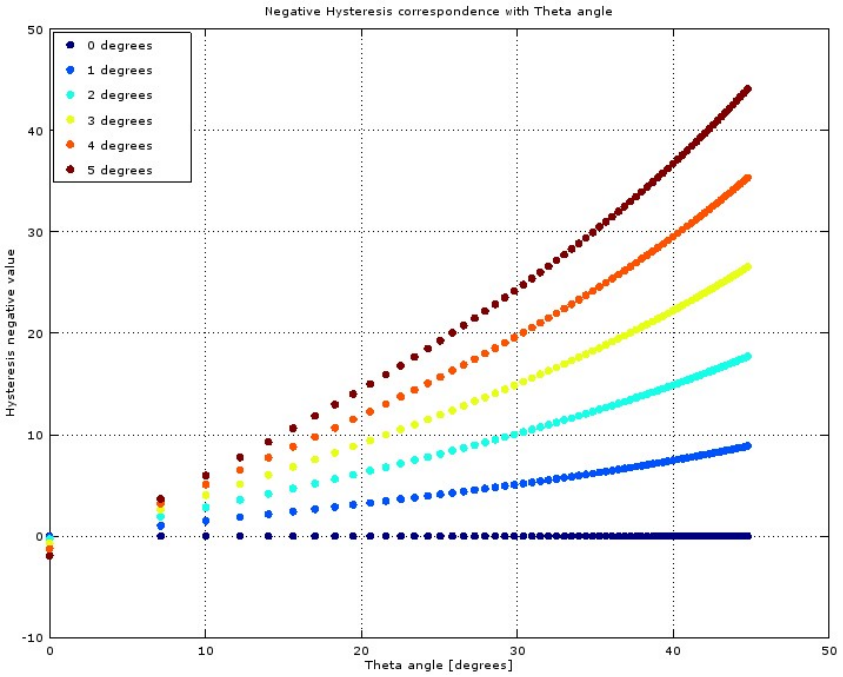


Figure 15: Negative hysteresis correspondence to θ angle

Blocking Mode The Flat detection can be suppressed for the case of a motion with a large magnitude of the acceleration to avoid a change of the Flat detection status. The blocking feature for the Flat detection interrupt is configurable via the `EXT.FLAT_1.blocking` and has the meaning as defined in Table 22.

Table 22: Blocking mode options for flat detection

Blocking mode	Conditions
0b00 and 0b11	Interrupt blocking is disabled
0b01	acceleration of any axis > 1.5 g
0b10	acceleration of any axis > 1.5 g OR slope > `EXT.FLAT_2.slope_thres`

Configuration Settings

1. `FEATURE_IO0.flat_en` – Switch indicating if this feature is enabled or not
2. `EXT.FLAT_1.theta` – Value corresponding to the hysteresis angle Θ for flat detection
3. `EXT.FLAT_1.blocking` – Sets the blocking mode, see Table 22.
4. `EXT.FLAT_1.hold_time` – Duration for which the current state is held before an interrupt can be raised again.
5. `EXT.FLAT_2.slope_thres` – Minimum slope between consecutive acceleration samples to prevent a change of the flat status during large movement
6. `EXT.FLAT_2.hysteresis` – Control the hysteresis to achieve the desired detection sensitivity

5.8.7 Orientation Detection

The Orientation Detection feature informs on an orientation change of the sensor with respect to the Earth gravitational force. There are the orientations face up and face down and orthogonal to them portrait upright, landscape left, portrait downside, and landscape right. The interrupt for face up/face down may be enabled separately by setting `EXT.ORIENT_1.ud_en` to 0b1. 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. 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 12.5 Hz to the maximum supported sample rate, see Table 19.

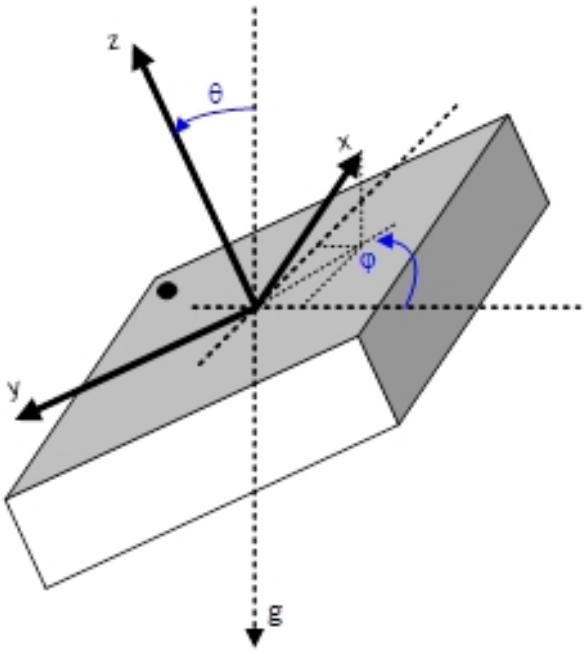


Figure 16: Definition of the default coordinate system with respect to pin 1 marker

The measured acceleration vector components for the default configuration reads as follows:

$$a_x = 1g \cdot \sin\theta \cdot \cos\varphi \tag{5.3}$$

$$a_y = -1g \cdot \sin\theta \cdot \sin\varphi \tag{5.4}$$

$$a_z = 1g \cdot \cos\theta \tag{5.5}$$

$$\frac{a_y}{a_x} = -\tan\varphi \tag{5.6}$$

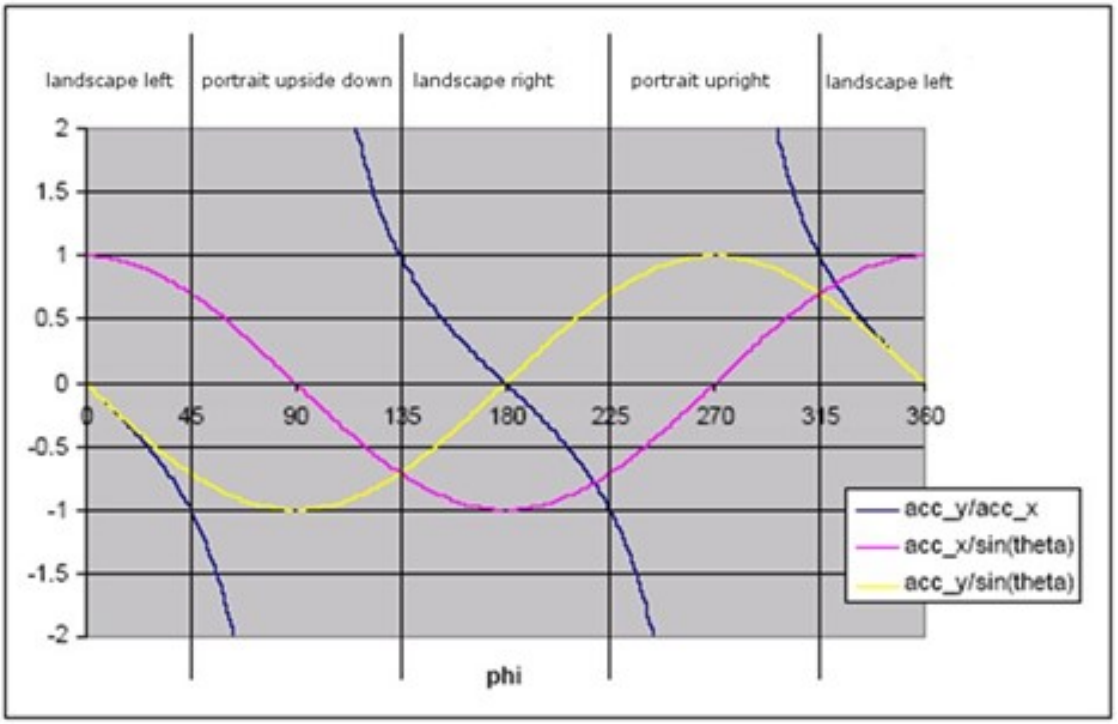


Figure 17: Angle-to-Orientation Mapping

The orientation value is stored in the output register. There are three orientation calculation modes: symmetrical, high-asymmetrical and low-asymmetrical. The mode can be configured through the `EXT.ORIENT_1.mode` as denoted in Table 23.

Orientation mode	**`EXT.ORIENT_1.mode`**
Symmetrical	0b00 and 0b11
High asymmetrical	0b01
Low asymmetrical	0b10

Table 23: Orientation Mode Selection

The output has the meanings depending on the switching mode as stated in the Tables 24, 25 and 26.

Table 24: Symmetrical mode

`FEATURE_EVENT_EXT.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 25: High Asymmetrical Mode

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

Table 26: Low Asymmetrical Mode

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

For upside or downside orientation, the output has to be interpreted according to Table 27.

Table 27: Upside/Downside Definition

FEATURE_EVENT_EXT. orientation_faceup_down	Name	Angle	Condition
0b0	upside	$270° < \varphi < 90°$	$a_z \geq 0$
0b1	downside	$90° < \varphi < 270°$	$a_z < 0$

Both orientation detections, 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 28, 29, and 30. The corresponding hysteresis regions are depicted in the Figures 18, 19, and 20.

Table 28: Hysteresis in the symmetrical mode

FEATURE_EVENT_EXT. 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 29: Hysteresis in the high asymmetrical mode

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

Table 30: Hysteresis in the low asymmetrical mode

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

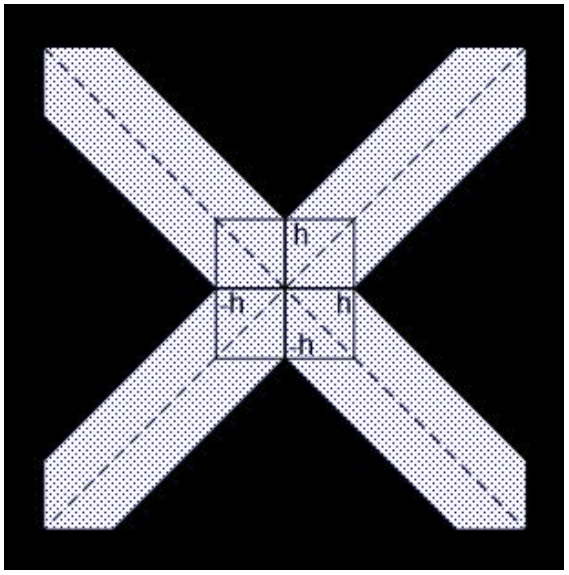


Figure 18: Hysteresis in the symmetrical mode

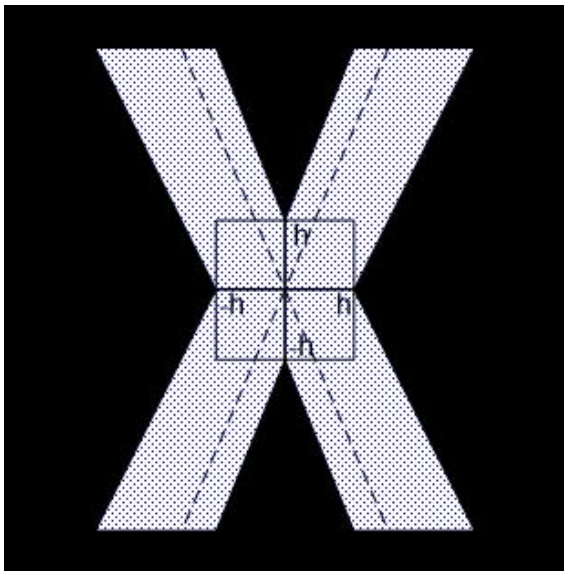


Figure 19: Hysteresis in the high asymmetrical mode

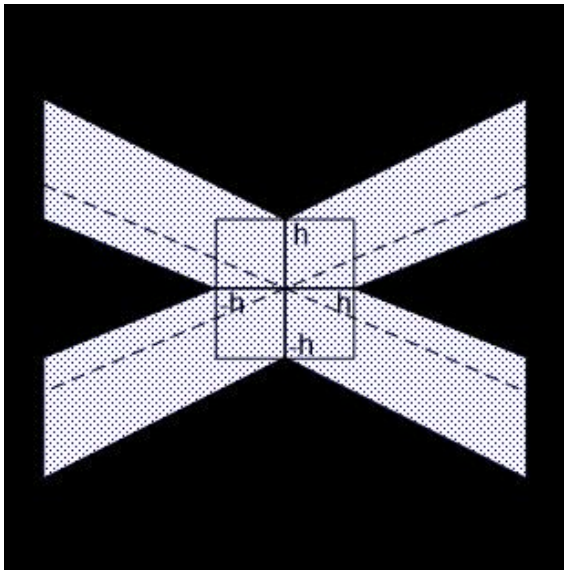


Figure 20: Hysteresis in the low asymmetrical mode

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

Blocking Mode It is be possible to block the notification of an Orientation Detection, that means no orientation change will be done. The orientation interrupt blocking feature is configurable via the `EXT.ORIENT_1.blocking`, and has the meaning defined in Table 31.

Table 31: Blocking mode options for orientation detection

EXT.ORIENT_1.blocking	Conditions
0b00	Interrupt blocking is disabled
0b01	Interrupt blocked if device close to the horizontal position OR acceleration of any axis > 1.5g
0b10	same condition as blocking mode 0b01 OR slope > 0.2g
0b11	same condition as blocking mode 0b10 OR another change within 100ms

Note: in case the 100 ms interrupt blocking is enabled, the interrupt will be triggered only when the device remains in the same (stable) orientation until the timer expires after ~100 ms. The timer starts to run when no change between two consecutive samples are detected. If there are changes while the timer count down is still running, the timer will be restarted.

The flat angle blocking, that is the device is close to the horizontal orientation, is defined by inequality presented in the Flat Detection section. The flat detection used for the blocking mode is based on the 2g acceleration measurement range. If any other measurement range than 2g range is selected for the acceleration, the acceleration samples are saturated before the Flat Detection.

Configuration Settings

1. `FEATURE_IO0.orientation_en` – Indicates if this feature is enabled or not
2. `EXT.ORIENT_1.ud_en` – Face upside/downside enable, in addition to landscape/portrait detection
3. `EXT.ORIENT_1.mode` – Used for setting which of the following modes are being used: symmetrical, high or low asymmetrical
4. `EXT.ORIENT_1.blocking` – Used for setting the blocking mode
5. `EXT.ORIENT_1.theta` – Coded value of the threshold angle with horizontal used in blocking modes; 1 LSB corresponds to $64 \cdot (\tan(\Theta))^2$
6. `EXT.ORIENT_2.hysteresis` – Acceleration hysteresis for Orientation detection

Orientation Output The detected orientation is reported by the three bits:

1. `FEATURE_EVENT_EXT.orientation_faceup_down` – only available if `EXT.ORIENT_1.ud_en` is set to 0b1: then 0b0 means face-up detected while 0b1 means face-down detected
2. `FEATURE_EVENT_EXT.orientation_portrait_landscape` – the two bits tell the orientation as
   a. 0b00: portrait upright
   b. 0b01: landscape left
   c. 0b10: portrait upside down
   d. 0b11: landscape right

5.8.8 Tap Detection

The device allows the detection of different tap gestures. Supported tap gestures include the single-tap, double-tap and triple-tap. Each gesture can be individually enabled and disabled as well as reported. Tap gestures can be enabled or disabled individually by writing 0b1 or 0b0 respectively to dedicated bits `FEATURE_IO0.tap_detector_s_tap_en`, `FEATURE_IO0.tap_detector_d_tap_en` and `FEATURE_IO0.tap_detector_t_tap_en`. When a tap gesture is detected, an event is reported with an interrupt common for single-tap, double-tap and triple-tap. The actual tap gesture reported can be obtained from the register `FEATURE_EVENT_EXT` with the bit encoded field values: `FEATURE_EVENT_EXT.s_tap`, `FEATURE_EVENT_EXT.d_tap`, and `FEATURE_EVENT_EXT.t_tap`. The value of a detected tap gesture is retained in the `FEATURE_EVENT_EXT` register until the next gesture detection.

The dominant sensing axis, along which the tap gesture has to be detected, can be configured using `EXT.TAP_1.axis_sel`. By default, `EXT.TAP_1.axis_sel` is selected to be the z-axis with a value set as 0b10. Depending on the tap impact and direction, it is also possible to detect performed tap gestures not strictly aligned with the selected axis.

A tap event is a subsequent crossing of a threshold within a configured maximum time limit between the crossings. The absolute value of the threshold for tap detection is programmable via `EXT.TAP_2.tap_peak_thres`. The maximum time window between the threshold crossings is defined by the parameter `EXT.TAP_3.max_dur_between_peaks`. To suppress or enable tap detection under noisy operating conditions, the limit on the number of threshold crossings acceptable for a tap can be configured using `EXT.TAP_1.max_peaks_for_tap`.

The classification of a tap gesture as a double-tap or triple-tap gesture is dependent on the time window in which the 2nd or 3rd tap occurs after the first tap. The time boundary is set by `EXT.TAP_2.max_gesture_dur`. In case of a double-tap to be detected, the 2nd tap event have to occur within this limit after the first tap. In case of a triple-tap to be detected, the 2nd and 3rd tap events has to occur within this limit after the first tap.

On detection of a tap gesture, the reporting behavior of the event by the device is determined by the value configured in `EXT.TAP_1.wait_for_timeout`:

- `EXT.TAP_1.wait_for_timeout` set to 0b0: the tap gesture event is reported after `EXT.TAP_3.tap_shock_settling_dur`. For a performed triple-tap gesture, the single-tap, double-tap and triple-tap event, when enabled, are all reported.
- `EXT.TAP_1.wait_for_timeout` set to 0b1: depending on the number of taps detected after `EXT.TAP_2.max_gesture_dur` duration since the first tap and the tap gesture to be detected, the corresponding tap gesture is reported. If the performed tap gesture corresponding to the number of taps detected is disabled or the number of performed taps is greater than three, no event is reported.

Once a tap gesture is reported, the detection of a further gesture is suspended for the “quiet” time requested by the parameter `EXT.TAP_3.quite_time_after_gesture`.

The tap detector offers the selection of a detection mode for a tap event with `EXT.TAP_1.mode`. The detection modes enable a simple selection of sensitivity levels for tap detection while keeping the values of other configuration parameters unchanged. Details are given in the memory map at `EXT.TAP_1.mode`.

The behavior of the tap detector with all supported tap gestures enabled and with `EXT.TAP_1.wait_for_timeout` set to 0b0 is shown in Fig. 21. For every detected tap, the corresponding interrupt is reported if the taps occur within `EXT.TAP_2.max_gesture_dur` since the first tap.

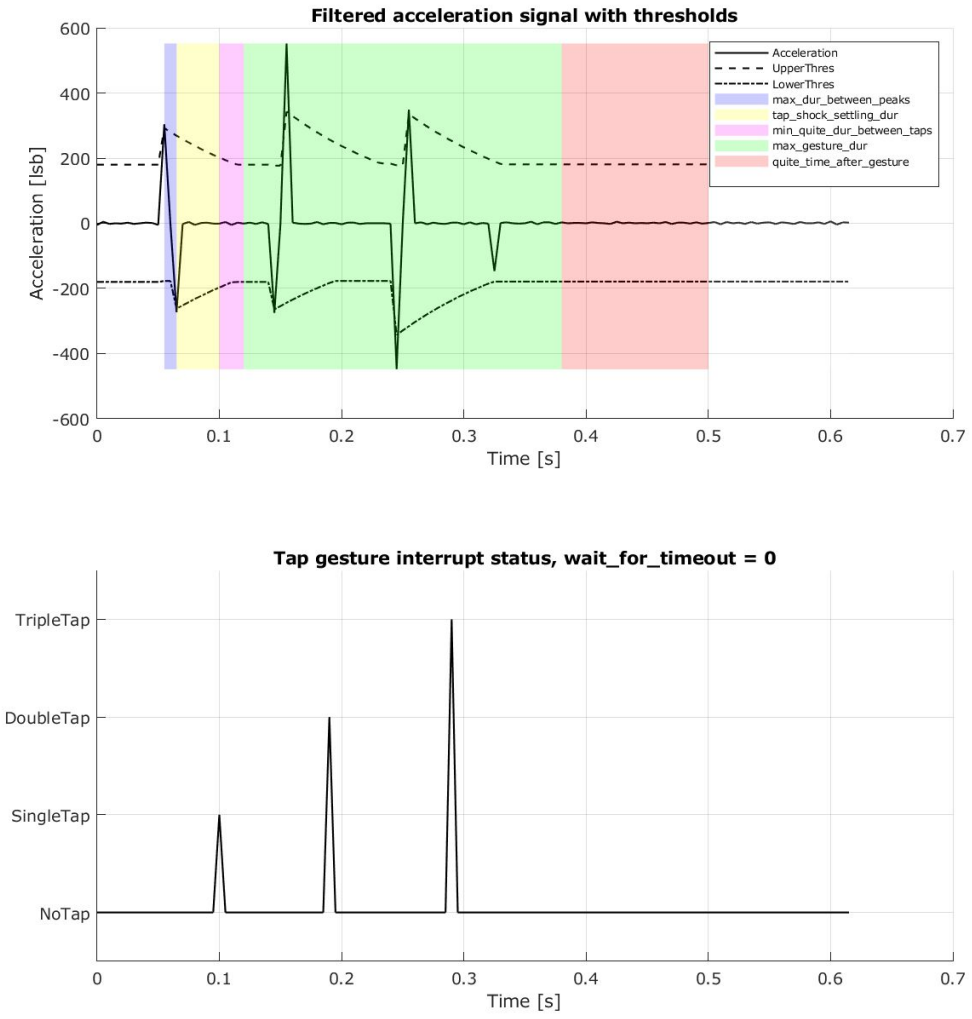


Figure 21: Reporting of tap gesture for `EXT.TAP_1.wait_for_timeout` = 0b0

The tap detector behavior in case of enabling all supported tap gestures and with `EXT.TAP_1.wait_for_timeout` set to 0b1 is depicted in the graphs in Fig. 22. As can be seen there, only the tap detected within `EXT.TAP_2.max_gesture_dur` after the first tap is reported.

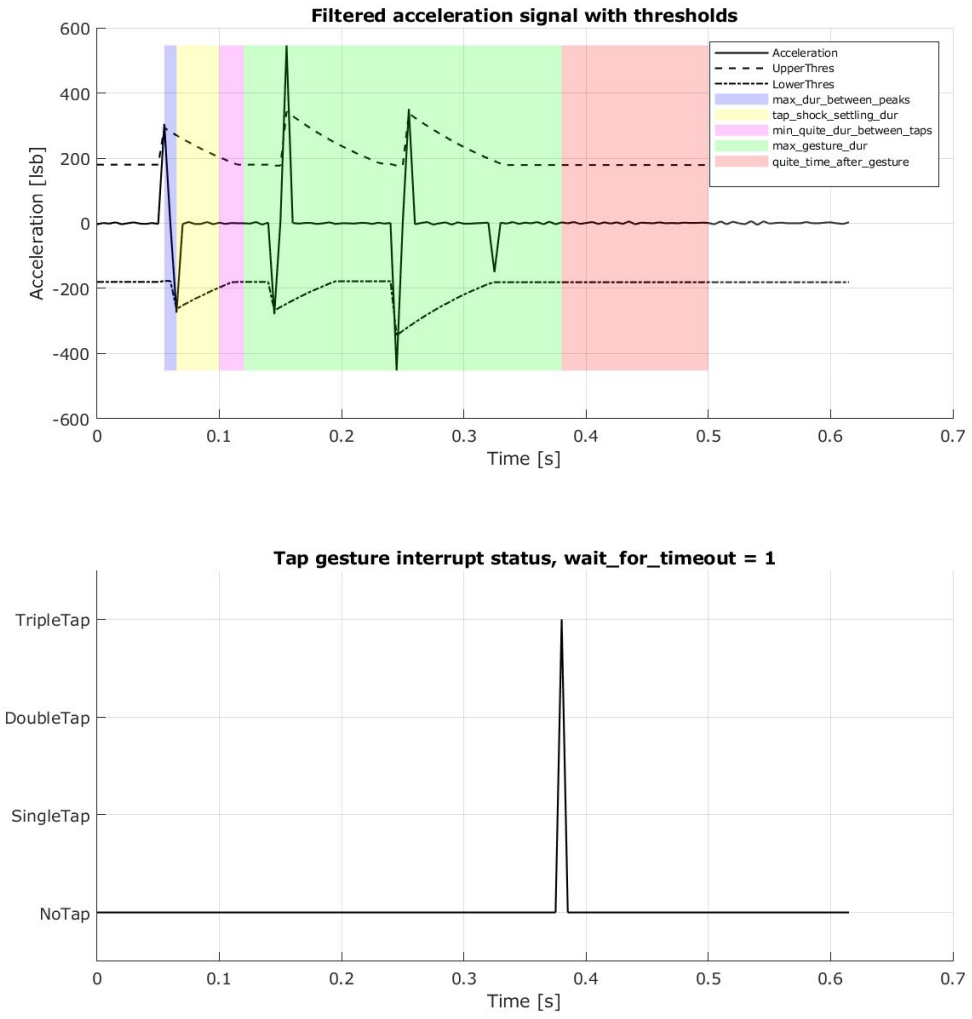


Figure 22: Reporting of tap gesture for `EXT.TAP_1.wait_for_timeout` = 0b1

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 200 Hz to the maximum supported sample rate, see Table 19.

Configuration Settings The configuration parameters for enabling/disabling of feature and adjustment of the behavior to the application and context are summarized in Table 32.

Table 32: Configuration parameters of the tap detector

Register/field name	Description	Default	Range
FEATURE_IO0.tap_detector_s_tap_en	Enable detection of single tap gesture	0	0/1 (dis/en)
FEATURE_IO0.tap_detector_d_tap_en	Enable detection of double tap gesture	0	0/1 (dis/en)
FEATURE_IO0.tap_detector_t_tap_en	Enable detection of triple tap gesture	0	0/1 (dis/en)
EXT.TAP_1.axis_sel	Selection of dominant axis for tap gesture detection	2	0 to 2
EXT.TAP_1.wait_for_timeout	Wait for duration set by max_gesture_dur after first tap for gesture confirmation	1	0/1 (dis/en)
EXT.TAP_1.max_peaks_for_tap	Maximum number of positive and negative threshold crossing for a tap detection	6	0 to 7
EXT.TAP_1.mode	Tap detection mode	1	0 to 2
EXT.TAP_2.tap_peak_thres	Magnitude threshold for the peak of tap event	45	0 to 1023
EXT.TAP_2.max_gesture_dur	Maximum time duration within which 2nd and/or 3rd tap have to performed for gesture classification	16	0 to 63
EXT.TAP_3.max_dur_between_peaks	Maximum duration between positive and negative peaks of a tap	4	0 to 15
EXT.TAP_3.tap_shock_settling_dur	Maximum duration for settling of tap shock	6	0 to 15
EXT.TAP_3.min_quite_dur_between_taps	Minimum quiet time between 2 consecutive taps	8	0 to 15
EXT.TAP_3.quite_time_after_gesture	Quiet time after gesture reporting when gesture detection is disabled	6	0 to 15

5.8.9 Tilt Detection

The function and behavior of the Tilt Detector is derived from the Android[3] specification available via the link https://source.android.com/devices/sensors/sensor-types.html#tilt_detector. A tilt interrupt is reported when the attitude angle of the device changes by a value greater than configured angle threshold. The detection of a tilt of the device can be enabled by writing 0b1 to FEATURE_IO0.tilt_en, otherwise the feature is disabled. 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 19.

The minimum angle of tilt for event detection can be configured using EXT.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 signal vector has to be estimated, is configured via the parameter EXT.TILT_1.segment_size. The lowpass filtering for the continuous estimation of the gravity acceleration vector can be configured with parameter EXT.TILT_2.beta_acc_mean.

The feature can be configured to output only the first detected event and ignoring the following events, also known as one-shot behavior, by setting EXT.GEN_SET_1.event_report_mode as 0b1. When this configuration is enabled, the configuration values of EXT.TILT_1.min_tilt_angle are set to 35 degrees and the EXT.TILT_1.segment_size to 2 seconds. When EXT.GEN_SET_1.event_report_mode is set to 0b1, changes to the values of this user configuration have no impact on the behavior this algorithm.

Example The functional behavior of the tilt detector for default configuration settings is shown in Figure 23.

[3] Android is a trademark of Google LLC.

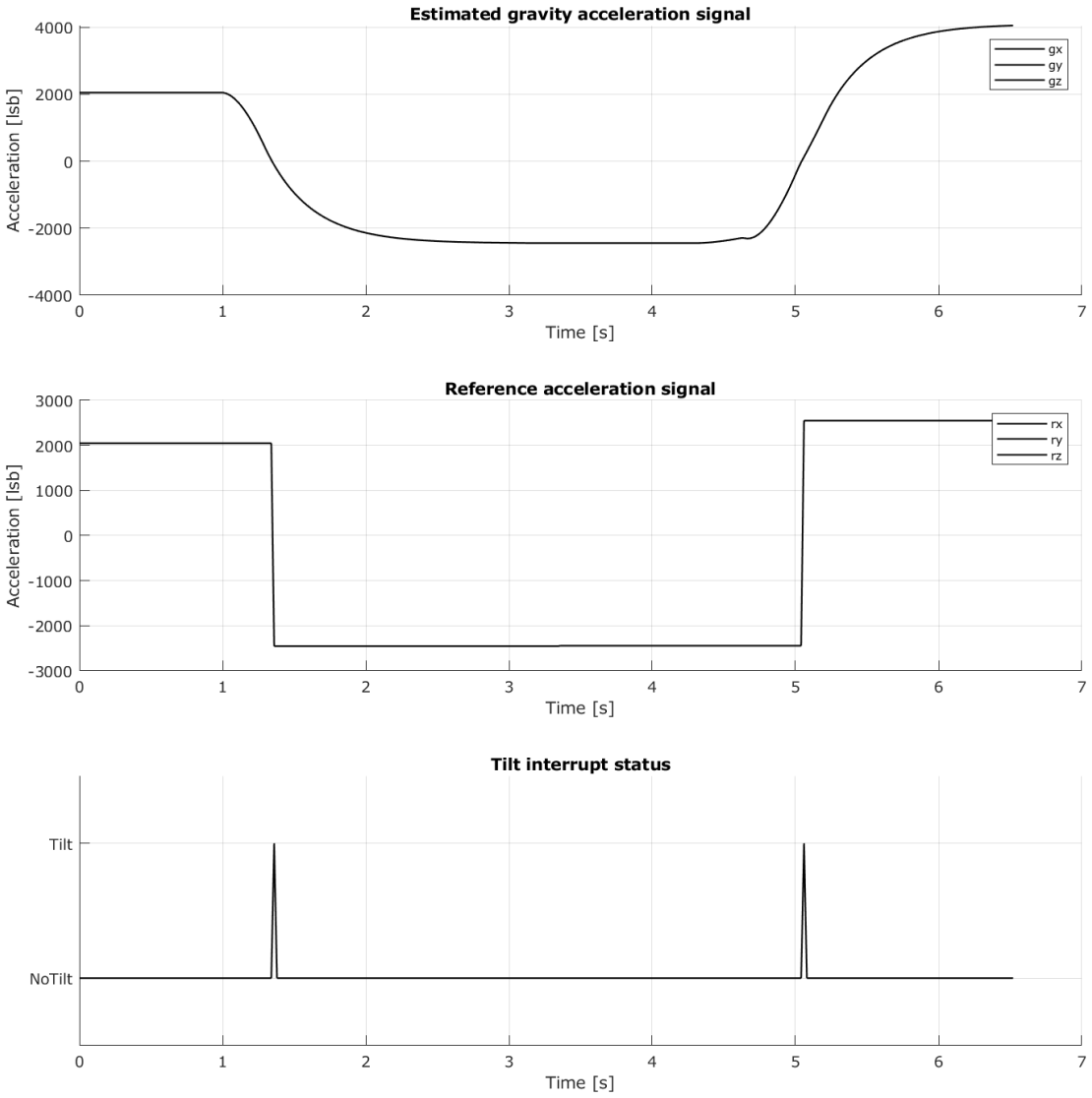


Figure 23: Functional behavior of tilt detection

The tilt interrupt is triggered when the change of the angle is greater than set threshold independent of the direction of movement.

Configuration Settings Configuration parameters for enabling and disabling of feature as well as to adapt the behavior to the application. The available parameters are described in Table 33.

Table 33: Configuration parameters of the tilt detector

Register/field name	Description	Default value	Range
FEATURE_IO0.tilt_en	Enable detection of tilt of the device	0	0/1 (disable/enable)
EXT.TILT_1.segment_size	Time duration for which acceleration vector is averaged for reference vector	100	0 to 255
EXT.TILT_1.min_tilt_angle	Minimum angle by which the device shall be tilted for event detection	210	0 to 255
EXT.TILT_2.beta_acc_mean	Decay coefficient for computation of acceleration vector mean	61545	0 to 65535

5.9 General Interrupt Signalling Configuration

The signalling of interrupts to the host is possible via the I3C feature in-band interrupt (IBI) and two electrical pins. When the protocols SPI and I²C are used, signalling of interrupts to the host is only possible via the INT1 and INT2 pin. IBI can only be used when the device is connected to the host via I3C. When using the protocol I3C, the IBI feature as well as the two interrupt pins can be used at the same time to signal interrupts to the host.

Interrupt Mapping and Acknowledgment The mapping of a feature interrupt to a signaling channel can be configured through the registers INT_MAP1 and INT_MAP2. Each feature is mapped exclusively to one of these signalling channels using the scheme stated in Table 34.

Table 34: Interrupt mapping to signalling channel

Map mode	Behaviour
0b00	Signalling of interrupt disabled
0b01	Signalling of interrupt on INT1 pin
0b10	Signalling of interrupt on INT2 pin
0b11	Signalling of interrupt via I3C IBI

Once an interrupt is notified to the host via a configured signalling channel, the host can determine the source of the interrupt through the status bits of the corresponding signalling interface. For each interrupt signalling channel, a dedicated interrupt status register is present in the register map of the device:

- the INT1 pin has the corresponding status register INT_STATUS_INT1,
- the INT2 pin has the corresponding status register INT_STATUS_INT2, and
- the in-band interrupt feature of I3C has the corresponding status register INT_STATUS_IBI.

It is recommended to separate the interrupts of the feature engine from the basic interrupts (data ready for signals and FIFO buffer interrupts), as the advanced features signaling pattern could be overruled by the basic interrupts.

Electrical Interrupt Pin Behavior The interrupt pins can be enabled via IO_INT_CTRL.int1_output_en and IO_INT_CTRL.int2_output_en, respectively. Note: this is not applicable to the I3C IBI feature. Both interrupt pins INT1 and INT2 can be configured to have the desired

- electrical drive characteristic,
- active level of the interrupt, and

- interrupt latching.

The characteristic of the output driver of the interrupt pins may be configured with bits IO_INT_CTRL.int1_od and IO_INT_CTRL.int2_od. By setting these bits to 0b1, the output driver shows open-drive characteristic, by setting the configuration bits to 0b0, the output driver shows push-pull characteristic.

The interpretation of the active level controlled through the interrupt pin can be configured as either “active-high” or “active-low” via IO_INT_CTRL.int1_lvl and IO_INT_CTRL.int2_lvl, respectively.

The device supports non-latched and latched interrupts modes for data ready, FIFO watermark, FIFO full, error, and the advanced feature interrupts. The mode is selected by INT_CONF.int_latch. Non-latched interrupts are designed for systems using edge triggered interrupts while latched interrupts are designed for systems using level-triggered interrupts. In the latched mode, an asserted interrupt status in INT_STATUS_INT1 and the INT1 pin are reset, if the status register INT_STATUS_INT1 is read. The same applies to an asserted interrupt status in INT_STATUS_INT2 and the INT2 pin, which are reset once the status register INT_STATUS_INT2 is read. If the interrupt activation condition still holds when the interrupt is reset, the interrupt status and pin are asserted again. In the non-latched mode, the configured INT1 or INT2 pin is reset as soon as the activation condition is not valid any more:

- for the feature engine based interrupts, the INT1 and INT2 pins are reset after the hold time selected by the host. In case of the features any-motion, no-motion, flat, orientation, single-tap, double-tap and triple-tap, the INT1/2 pin will be reset and set again if the hold time is configured to a value less than 20 ms even though the condition is true.

The interrupt status bits are active until read by the host.

5.10 Auto-operation mode change

The auto-operation mode change is a built-in feature to support the smart power management of the device. The function provides automatic switching among two sets of operation modes for its accelerometer and gyroscope. The switching is initiated by events of enabled advanced features or by commands sent from the host. In the following, the one set of configurations consists of ACC_CONF and GYR_CONF for the accelerometer and gyroscope and is called user configuration. The other set sensor of configurations consists of ALT_ACC_CONF and ALT_GYR_CONF, and is called alternative configuration.

Switching between the user and alternative configuration is enabled through ALT_CONF.alt_acc_en and ALT_CONF.alt_gyr_en for the accelerometer and gyroscope, respectively. The conditions to switch from the operation mode configured can be configured through EXT.ALT_CONFIG_CHG. By selecting one of the advanced features described in Section 5.8, it can be configure independently

- on the one hand through EXT.ALT_CONFIG_CHG.alt_conf_alt_switch_src_select to switch from the user configuration to the alternative configuration by selecting one of the advanced features with source IDs in Table 35, and
- on the other hand through EXT.ALT_CONFIG_CHG.alt_conf_user_switch_src_select to switch from the alternative configuration to the user configuration by selecting one of the advanced features with source IDs in Table 35.

Table 35 details the selectable advanced features with respect to the index letter. Please be aware, that if advanced features are configured for both transitions, that means for the switch from the user to the alternative configuration and for the switch from the alternative to the user configuration, not the same advanced feature is enabled. If ALT_CONF.alt_rst_conf_write_en is enabled, the configurations of the sensors can be instantly reset to the user configuration by directly writing from the host to either ACC_CONF or GYR_CONF.

Table 35: Advanced feature mapping indices

Feature index letter	Advanced feature name
A	No motion detection
B	Any motion detection
C	Flat detection
D	Orientation detection
E	Step detector
F	Step counter watermark
G	Significant motion detection
H	Tilt detection
I	Tap detection

5.11 Axis Re-mapping and Sign Inversion

The device supports the re-mapping of the axis and inversion of the sign within the data path. This feature of the device ensures consistent usage of the acceleration and angular rate signal in the algorithms on the device as well as in the processing on the host. The re-mapping and sign inversion for the device axis must only be done when the device is not acquiring data nor computing any advanced feature, e.g. after power up or after a soft reset. The re-mapping and sign inversion of the device axis is a volatile configuration and, hence, has to be performed always after power up or soft reset to ensure the same output by the device.

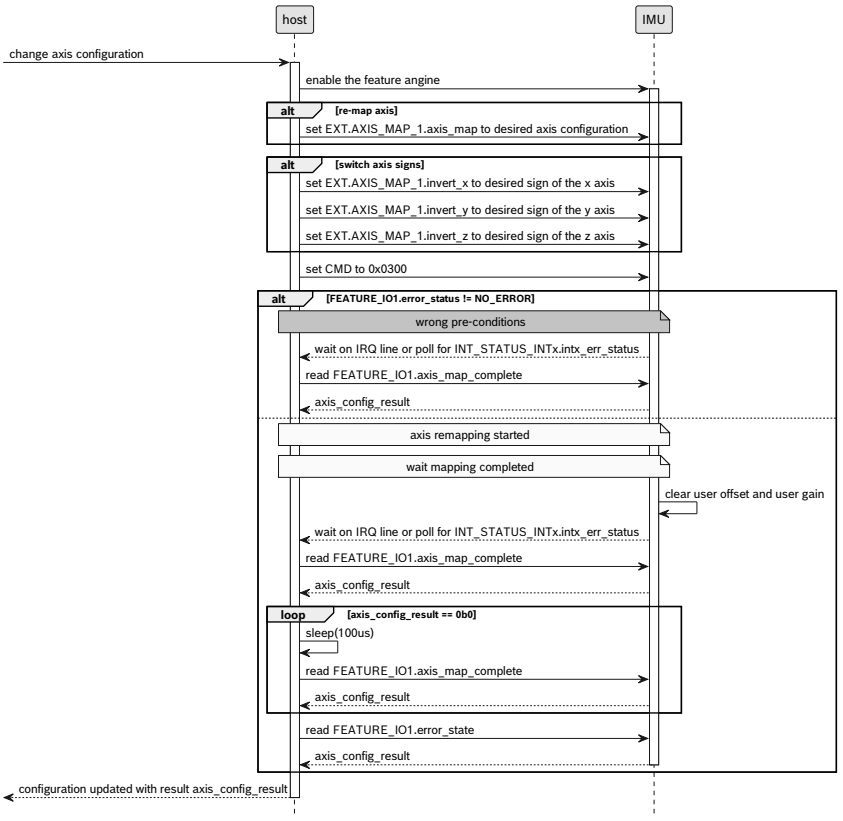


Figure 24: Sequence to perform the axis re-mapping and sign version

The desired configuration for the axis re-mapping and sign inversion has to be written to `EXT.AXIS_MAP_1` before sending the command 0x0300 to the register `CMD`. The mapping of the default axis configuration to the desired axis configuration can be done writing the map code to `EXT.AXIS_MAP_1.axis_map`. The sign inversion of the axis, which can be interpreted

as an axis flip, can be configured for each axis individually by toggling the default value 0b0 to 0b1 in `EXT.AXIS_MAP_1.invert_x`, `EXT.AXIS_MAP_1.invert_y` or `EXT.AXIS_MAP_1.invert_z`, respectively. After the update of this configuration, the desired axis re-mapping and sign inversion can be activated by sending the command 0x0300 to the register `CMD`. The success or failure of the axis re-mapping can be determined by

- monitoring the interrupt err_status via an interrupt pin or I3C IBI, and
- polling the register `FEATURE_IO1` and checking the values in the fields `FEATURE_IO1.axis_map_complete` and `FEATURE_IO1.error_status`.

Any configuration of the device must not be changed until the completion of the axis re-mapping and sign inversion is signalled. This procedure will clear the data path gain and offset values. The gain and offset for accelerometer and gyroscope have to be saved before power down or soft-reset, and they can only be re-applied after power-up and after performing again an axis re-mapping and sign inversion.

Configuration Settings

1. `CMD` – perform the axis re-mapping and sign inversion by sending the code 0x0300
2. `EXT.AXIS_MAP_1.axis_map` – mapping of the default axes configuration to the user configuration
3. `EXT.AXIS_MAP_1.invert_x` – inversion of the x axis in user configuration
4. `EXT.AXIS_MAP_1.invert_y` – inversion of the y axis in user configuration
5. `EXT.AXIS_MAP_1.invert_z` – inversion of the z axis in user configuration

5.12 User offset and sensitivity error update

The compensation of the offset and sensitivity error observed in the field, e.g. due to soldering, is strongly recommended to be applied on the host. Values for compensation have to be determined on the host except for the case of using gyroscope self-calibration provided from within the device. Compensation on the host is recommended because the values must be stored on the host since the device does not provide means to store these within non-volatile memory of the device. In case it is desired to output the calibrated signal of the accelerometer and gyroscope by the device itself, the device provides the user with dedicated means to perform compensation with values provided by the user. To obtain the expected output signal $\mathbf{s}_{\mathrm{cal}}$, where **s** is either the acceleration **a** or the angular rate **w**, the offset and sensitivity error must be estimated on the host according to the model

$$s_{cal,\{x;y;z\}} = g_{\{x;y;z\}} \cdot s_{\mathrm{uncal},\{x;y;z\}} + o_{\{x;y;z\}} \,, \tag{5.7}$$

where $\mathbf{o} = (o_{\mathrm{x}}, o_{\mathrm{y}}, o_{\mathrm{z}})$ and $\mathbf{g} = (g_{\mathrm{x}}, g_{\mathrm{y}}, g_{\mathrm{z}})$ are the user offset and user sensitivity error of the signal $\mathbf{s}_{\mathrm{uncal}}$, respectively. Then, the user sensitivity error and the user offset values can be written by the user directly to registers for the accelerometer and gyroscope. These registers are:

- for the accelerometer:
  - user offset `ACC_DP_OFF_X`, `ACC_DP_OFF_Y`, and `ACC_DP_OFF_Z`
  - user sensitivity error `ACC_DP_DGAIN_X`, `ACC_DP_DGAIN_Y`, and `ACC_DP_DGAIN_Z`
- for the gyroscope:
  - user offset `GYR_DP_OFF_X`, `GYR_DP_OFF_Y`, and `GYR_DP_OFF_Z`
  - user sensitivity error `GYR_DP_DGAIN_X`, `GYR_DP_DGAIN_Y`, and `GYR_DP_DGAIN_Z`

It is strongly recommended to update the registers only when the sensors, that means accelerometer and gyroscope, are disabled to avoid settling of the respective signal, that means either accelerometer or gyroscope, after the values are updated.

Note: the self-calibration for the gyroscope described in section 5.13 can write, depending on the configuration, directly into the registers for compensation of user offset and/or user sensitivity error for the gyroscope. The user must take care of saving these values on the host and either perform compensation on the host or by the device through writing the values for user sensitivity error and user offset registers for the gyroscope after each power-on-reset and soft-reset.

Note: as a prerequisite for self-calibration, gyroscope user offset and user sensitivity error registers must be cleared out before each self-calibration execution.

5.13 Self Calibration (CRT)

The device offers self-calibration for the gyroscope sensitivity error and the gyroscope offset. Self-calibration to reduce the gyroscope sensitivity error is also known as component re-trim (CRT).

Configuration of the Self Calibration The self-calibration sequence can be configured with `EXT.GYR_SC_SELECT.sens_en` and `EXT.GYR_SC_SELECT.offs_en` to calibrate either both gyroscope sensitivity error and gyroscope offset or only one of them. If both gyroscope characteristics are desired to be calibrated, the device is calibrated first for the gyroscope sensitivity error followed by the gyroscope offset. The default configuration of calibrating both gyroscope characteristics can be changed by writing 0b0 to `EXT.GYR_SC_SELECT.sens_en` or `EXT.GYR_SC_SELECT.offs_en`. By default, the results of the self-calibration are written to data path registers `GYR_DP_DGAIN_X`, `GYR_DP_DGAIN_Y` and `GYR_DP_DGAIN_Z` for the sensitivity error and `GYR_DP_OFF_X`, `GYR_DP_OFF_Y` and `GYR_DP_OFF_Z` for the offset. By setting `EXT.GYR_SC_SELECT.apply_corr` to 0b0, the update of the data path registers can be suppressed.

The device monitors the motion that is imposed by the context onto the device. The sensitivity of this motion detection can be configured through `EXT.GYR_MOT_DET.slope`. Note: this configuration of the detection of any motion is also used by the device self test.

Prerequisites for the Self Calibration Before initiating the self-calibration, it should be checked for an on-going self-calibration or self-test by reading `FEATURE_IO1.state`. When the value is 0b00, a self-calibration can be initiated. As long as `FEATURE_IO1.state` is not 0b00, this bit shall be polled by re-reading until the value turns to 0b00. To start a self-calibration, the accelerometer is required to be enabled (already) in high performance mode with a sample rate `ACC_CONF.acc_odr` preferred in the range of 25 Hz up to 200 Hz. The alternative sensor configurations for accelerometer and gyroscope must be disabled by setting `ALT_ACC_CONF.alt_acc_mode` and `ALT_GYR_CONF.alt_gyr_mode` to 0b0, respectively. Then, a self-calibration can be initiated. Note: if the command for a self-calibration is sent once or multiple times while `FEATURE_IO1.state` has the value 0b01, 0b10 and 0b11, the command is ignored.

Execution of the Self Calibration The device shall be kept in a still orientation while the self-calibration is running and shall not be exposed to noise and distortion. The self calibration can be started by writing 0x0101 to the register `CMD`. After initiating the self-calibration, the device shall not be re-configured until the end of self-calibration procedure is reported.

The start of the self-calibration disables immediately the output of gyroscope sensor samples (data). During the execution of the self-calibration, the gyroscope sensor output in the data registers as well as in the FIFO data buffer is invalid, however, accelerometer output is still valid. The output of any previously enabled advanced features of Section 5.8 will still be provided during self-calibration. The duration of the self-calibration for standard settings is approximately 350 ms for the measurement of the re-scaling for the angular rate and 80 ms for the gyroscope offset measurement.

The state of the self-calibration can be determined by checking `FEATURE_IO1.sc_st_complete`. An ongoing self-calibration is reflected by 0b0 while 0b1 is reported when it is completed. An ongoing self-calibration is also reflected in the register field `FEATURE_IO1.state` with the value 0b01. The completion of the self-calibration is also reported in `FEATURE_IO1.error_status` with the value 0x5. The success of the self-calibration is reported in the register field `FEATURE_IO1.gyro_sc_result` with 0b1, a failure with 0b0. Depending on the configuration `EXT.GYR_SC_SELECT.apply_corr` of the self-calibration, the data path registers are updated automatically. The sequence of executing the self-calibration is summarized in Figure 25.

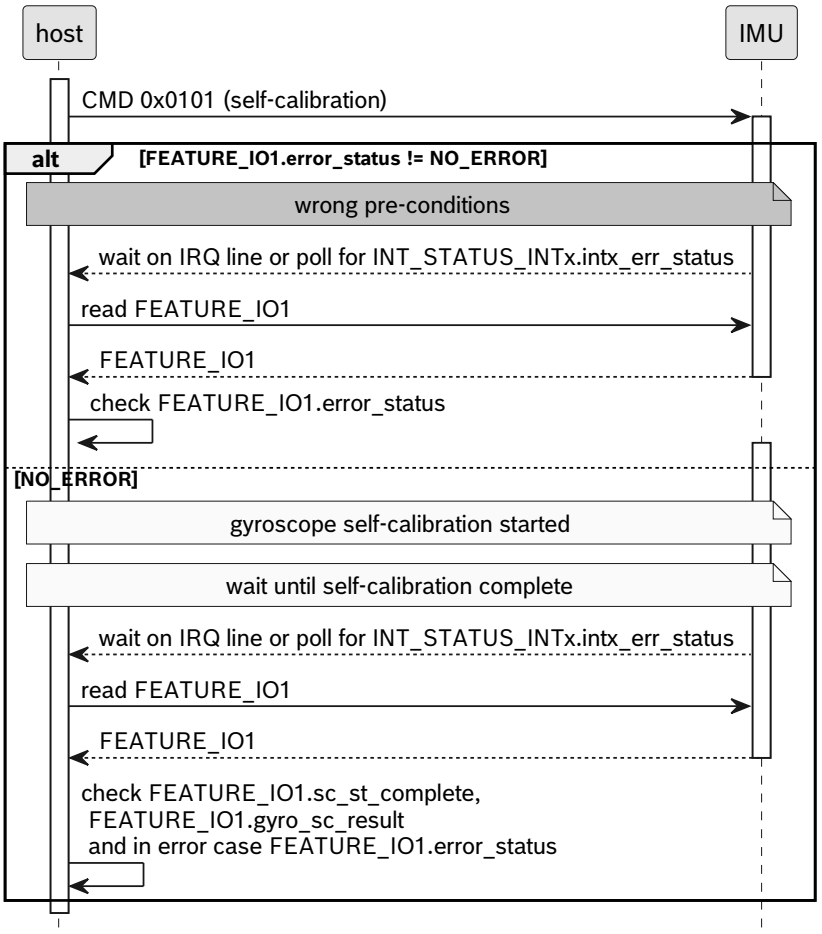


Figure 25: Self-Calibration Start Sequence

Note: the results of the self-calibration in the data path registers should to be saved by the host to a non-volatile storage to allow restoring them after a power down or soft reset of the device.

Abort the Self Calibration A running self-calibration can be aborted by sending the command 0x0200 to `CMD`. Then, `FEATURE_IO1.state` can be checked for any state different than 0b01 and the device reports in `FEATURE_IO1.error_status` the value 0x9. The sequence to abort the self-calibration is detailed in Figure 26.

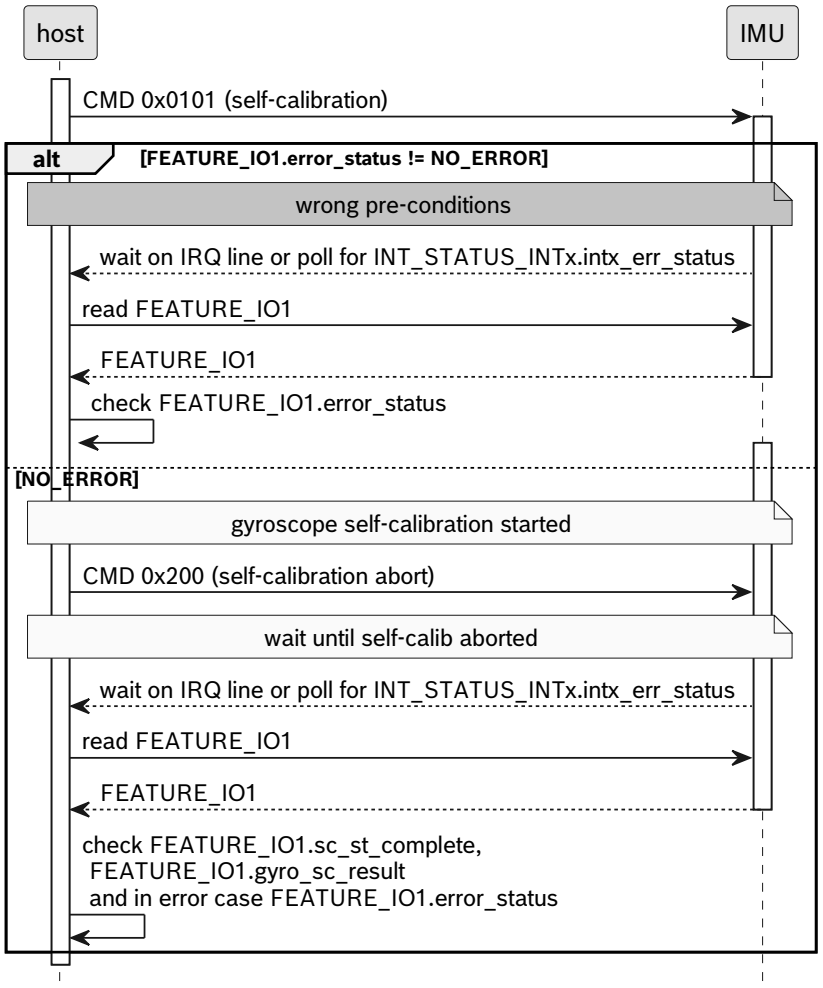


Figure 26: Self-Calibration Abortion Sequence

5.14 Self Test

The self-test of the device checks for a correct function of the accelerometer as well as the gyroscope. The execution of the self-test expects the following prerequisites to be fulfilled:

- for the gyroscope self-test, the accelerometer must be configured to high performance mode at least with a sample rate `ACC_CONF.acc_odr` in the range of 12.5 Hz up to 200 Hz, and
- the alternative sensor configurations for accelerometer and gyroscope must be disabled by setting `ALT_ACC_CONF.alt_acc_mode` and `ALT_GYR_CONF.alt_gyr_mode` to 0b0, respectively.

The self-test for the accelerometer and the gyroscope is initiated by writing the command 0x0100 to the register `CMD`. It is possible to configure the self test to check only the accelerometer or the gyroscope by disabling either the gyroscope with `EXT.ST_SELECT.gyr_st_en` to 0b0 or the accelerometer with `EXT.ST_SELECT.acc_st_en` to 0b0, respectively. Depending on the configuration, the self test takes from 100 ms for the only the accelerometer up to 350 ms for both sensors.

Once a self test is initiated, the output of data of the device to the registers and FIFO data buffer as well as all features are disabled. While the self-test is in progress, the host is not allowed to modify the configuration of the device. The device reports in the register field `FEATURE_IO1.sc_st_complete` with 0b1 when the self-test is completed. The result of the self-test can be determined by evaluating the register field `FEATURE_IO1.st_result`. In case the self-test failed, detailed information about the sensor and axis causing the failure can be obtained by reading from `EXT.ST_RESULT`. After a completed self test, a soft reset or power cycle of the device is not required.

The device monitors the motion the device is exposed to while executing the self-test. The sensitivity of this motion

detection can be configured through `EXT.GYR_MOT_DET.slope`. Note: This configuration of the detection of any motion is also used by the device self calibration.

5.15 I3C Timing Control

The device supports time control modes as defined by the MIPI I3C^{SM} standard. As defined by the I3C specification, the device supports timing control for both synchronous systems and events called synchronous mode as well as for asynchronous systems and events called asynchronous basic mode. The modes can be configured, entered, and left using I3C common command codes (CCCs).

Mode for Asynchronous Systems and Events The device supports the asynchronous basic mode of timing control (asynchronous mode 0) as defined in the MIPI I3C^{SM} specification. For entering the asynchronous mode 0, the host sends the CCC `SETXTIME` with the defining byte 0xDF. For exiting the asynchronous mode 0, the host sends the CCC `SETXTIME` with sub-command 0xFF.

Mode for Synchronous Systems and Events The main idea of the synchronous mode is to configure the sensor for synchronization first and then issue periodically synchronization messages to keep the sensor device synchronized with the host device. The data ready interrupts of accelerometer, gyroscope and temperature sensor are aligned with respect to sample rate and phase. The synchronization process is divided into the two phases configuration and run-time. Note: this mode

- cannot be used together with when the automatic operation mode switch is enabled,
- cannot be enabled while a self-test is ongoing, and
- aborts an on-going self-calibration when this mode is enabled.

Note: if a self-test is in progress while this mode is requested to be enabled, the synchronous timing control will be enabled only after the self-test is finished.

The mode for synchronous systems and events has to be configured for the serial interface with the CCC `SETXTIME`, and the data processing control `FEATURE_IO0.i3c_sync_en`. To have the serial interface entering the synchronous mode, the host sends the CCC `SETXTIME` with any of the sub-command 0x3F, 0x9F, or 0x8F. The argument of the subcommand is ignored, if the device is not in the synchronous mode. To enter the timing control in synchronous mode and to set the time period and the phase value 0xABCD, the host needs to send the two CCCs `SETXTIME` 0x3F|0x9F|0x8F followed by two dummy bytes and then `SETXTIME` 0x3F 0xABCD. Another possibility for the host to configure the synchronous mode is to read or write the synchronous mode parameters T_Ph, TU, and ODR directly via the registers `I3C_TC_SYNC_TPH`, `I3C_TC_SYNC_TU`, and `I3C_TC_SYNC_ODR`, respectively. Before sending I3C synchronization messages, the synchronous mode needs to be enabled for the data processing by setting `FEATURE_IO0.i3c_sync_en` to 0b1. For exiting the synchronous mode, the host sets `FEATURE_IO0.i3c_sync_en` to 0b0 to disable the I3C data processing and sends the CCC `SETXTIME` with sub-command 0xFF to control the serial interface of the device.

Configuration During the configuration phase the parameters sample rate (ODR), time period and phase (T_Ph) and time unit (TU) have to be configured by the host:

ODR The ODR command selects the enforced ODR. The payload of this command consists of one byte. The format of the ODR payload equals the `ACC_CONF.acc_odr` register. The minimum and maximum configurable sample rate for synchronous timing control is in the range of 6.25 Hz up to 800 Hz. If the configuration is selected beyond this range, the default sample rate of 100 Hz is configured. Example: ODR payload = 0x8 means 100 Hz.

T_ph The time period and phase (T_ph) command provides the number of samples that are included between two successive synchronization messages.

TU The TU command configures the resolution ratio of the delay, given by DT command.

During the configuration phase the command ODR, TPH and TU are updated by the CCC `SETXTIME` in I3C mode. The host is allowed to alter `I3C_TC_SYNC_TPH`, `I3C_TC_SYNC_TU`, and `I3C_TC_SYNC_ODR` whenever required at any time. However, the new values take effect only active after sending configuration change command 0x201 for synchronous

timing control to the register CMD. With a ST or DT message, any change of the configuration will become effective for this device.

Before entering the Runtime phase, the sync for sensor feature must be enabled by FEATURE_IO0.i3c_sync_en.

Note: while timing control synchronous is enabled, the configuration of the sample rate of the accelerometer and gyroscope must not be modified through ACC_CONF.acc_odr and GYR_CONF.gyr_odr, respectively. Any change to these values is ignored and will be overwritten by the device, and this is reported in FEATURE_IO1.error_status with a value of 0xF. The configuration of the sample rate has to be done either through the register I3C_TC_SYNC_ODR or the corresponding CCC.

Run Time Operation In the run-time phase, the host issues periodically synchronization messages to the sensor. These messages consist of two inter-related commands. First, the ST command selects a START event as reference for calculating the next synchronization period. Afterwards the DT command is sent by the host validating the ST and providing a delay time information to adjust the START event. If both commands are received, the sensor can update its clock parameters.

ST The sync tick command identifies the transaction START event. It has no payload.

DT The DT command delivers the delay time between the ST message and the correct START moment. It has one byte of payload. The MSB of the payload indicates that the delay time is valid (0) or that this sync procedure is aborted (1). If the delay is valid, the remaining 7 bits indicate the number of resolution steps between ST and the correct START event, see the description of TU.

For a more detailed description of the synchronization protocol see the MIPI I3C^{SM} specification.

ODR Configuration Error The synchronous mode feature will operate in the correct manner only when the user follows the constraints on the supported sample rates. The device checks continuously for matching values of sample rates. If the value for one of the signals acceleration, angular rate or temperature is not within the limits, the i3c_error bit is set. The conditions for return an I3C error for an invalid sample rate (ODR) is detailed in Figure 27.

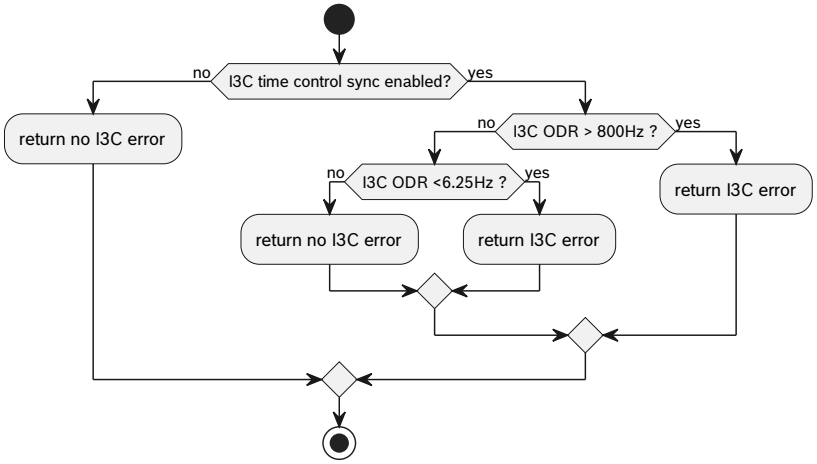


Figure 27: Conditions for I3C Sample Rate (ODR) Configuration Errors

Output Details After enabling the synchronous timing control feature for I3C, the sensor data is available in the FIFO data buffer. Furthermore, the current valid sample set can be obtained via the extended register map from EXT.SYNC_ACC_X to EXT.SYNC_TIME. No valid data will be available in data registers ACC_DATA_X to TEMP_DATA. The format of the data in the FIFO data buffer is the same as in any other mode. That means, sensors that are not enabled will not appear in the FIFO data frame, see Section. 5.7.

5.16 Device Status

The device reports its status through

- the register ERR_REG for hardware errors:
  - Illegal configurations in the registers ACC_CONF and GYR_CONF will result in an error code in the fields ERR_REG.acc_conf_err and ERR_REG.gyr_conf_err, respectively. In this case, the content of the corresponding data register is undefined. For details, please refer to Section 5.6.
  - I3C errors are reported with ERR_REG.i3c_error0 and ERR_REG.i3c_error3. For details, please refer to Section 7.2.4.
- the register STATUS for availability information: for details, please check Section 5.6.
- the register FEATURE_IO_STATUS for advanced feature information
- the register FEATURE_DATA_STATUS for the status of the configuration interface for the advanced features
- the register FEATURE_ENGINE_STATUS for hardware errors of the feature engine

Reserved bits in these registers are for Bosch Sensortec internal purposes only and can be ignored safely.

5.17 Soft Reset

A soft reset can be initiated at any time by writing the command softreset 0xDEAF to the register 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 wherever applicable. To access the serial interface with any of the protocols SPI, I3C or I²C after a soft reset, the same timing constraints apply as for power on, see Chapter 3 for details.

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.

For all data read from registers, content marked as reserved must be ignored. If only the first byte of a register contains non-reserved data, the upper byte is not required to be read.

When performing an update of a register with fields marked as reserved (partial update), a read-modify-write approach has to be followed to write the content of a register. That means, for a write access to a register with reserved content, the whole register must be read, then the desired content must be updated and the content written (back) to the register to avoid overwriting the reserved part with undefined content. This is especially important for the extended register map where many configurations are located.

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 usage of the registers `FEATURE_IO0` to `FEATURE_IO3` together with the register `FEATURE_IO_STATUS` is explained in Section. 5.8. The access to the extended register map through the registers `FEATURE_DATA_ADDR`, `FEATURE_DATA_TX` and `FEATURE_DATA_STATUS` is explained in Section 6.2.

6.1.1 Register Map Overview

Table 36 provides an overview of the register map of the device.

Table 36: 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**
0x00	CHIP_ID	0x0048	reserved								chip_id							
0x01	ERR_REG	0x0000	reserved				i3c_er...	reserved		i3c_er...	reserved	gyr_co...	acc_co...	feat_e...	reserved	feat_e...	reserved	fatal_err
0x02	STATUS	0x0001	reserved								drdy_acc	drdy_gyr	drdy_temp	reserved				por_de...
0x03	ACC_DATA_X	0x8000	acc_x															
0x04	ACC_DATA_Y	0x8000	acc_y															
0x05	ACC_DATA_Z	0x8000	acc_z															
0x06	GYR_DATA_X	0x8000	gyr_x															
0x07	GYR_DATA_Y	0x8000	gyr_y															
0x08	GYR_DATA_Z	0x8000	gyr_z															
0x09	TEMP_DATA	0x8000	temp_data															
0x0A	SENSOR_TIME_0	0x0000	sensor_time_15_0															
0x0B	SENSOR_TIME_1	0x0000	sensor_time_31_16															
0x0C	SAT_FLAGS	0x0000	reserved										satf_g...	satf_g...	satf_g...	satf_a...	satf_a...	satf_a...
0x0D	INT_STATUS_INT1	0x0000	int1_f...	int1_fwm	int1_a...	int1_g...	int1_t...	int1_e...	int1_i3c	int1_tap	int1_tilt	int1_s...	int1_s...	int1_s...	int1_o...	int1_flat	int1_a...	int1_n...
0x0E	INT_STATUS_INT2	0x0000	int2_f...	int2_fwm	int2_a...	int2_g...	int2_t...	int2_e...	int2_i3c	int2_tap	int2_tilt	int2_s...	int2_s...	int2_s...	int2_o...	int2_flat	int2_a...	int2_n...
0x0F	INT_STATUS_IBI	0x0000	ibi_ffull	ibi_fwm	ibi_ac...	ibi_gy...	ibi_te...	ibi_er...	ibi_i3c	ibi_tap	ibi_tilt	ibi_si...	ibi_st...	ibi_st...	ibi_or...	ibi_flat	ibi_an...	ibi_no...
0x10	FEATURE_IO0	0x0000	i3c_sy...	tap_de...	tap_de...	tap_de...	tilt_en	sig_mo...	step_c...	step_d...	orienta...	flat_en	any_mo...	any_mo...	any_mo...	no_mot...	no_mot...	no_mot...
0x11	FEATURE_IO1	0x0000	reserved			state		axis_m...	reserved		sample:.	st_result	gyro_s...	sc_st:.	error_status			
0x12	FEATURE_IO2	0x0000	step_counter_out_0															
0x13	FEATURE_IO3	0x0000	step_counter_out_1															
0x14	FEATURE_IO_STATUS	0x0018	reserved															feature...
0x15	FIFO_FILL_LEVEL	0x0000	reserved				fifo_fill_level											
0x16	FIFO_DATA	0x0000	fifo_data															
...	-	-	reserved															
0x20	ACC_CONF	0x0028	reserved	acc_mode			reserved	acc_avg_num			acc_bw	acc_range			acc_odr			
0x21	GYR_CONF	0x0008	reserved	gyr_mode			reserved	gyr_avg_num			gyr_bw	gyr_range			gyr_odr			
...	-	-	reserved															
0x28	ALT_ACC_CONF	0x3206	reserved	alt_acc_mode			reserved	alt_acc_avg_num			reserved				alt_acc_odr			
0x29	ALT_GYR_CONF	0x1206	reserved	alt_gyr_mode			reserved	alt_gyr_avg_num			reserved				alt_gyr_odr			
0x2A	ALT_CONF	0x0000	reserved							alt_rs...	reserved			alt_gy...	reserved			alt_ac...
0x2B	ALT_STATUS	0x0000	reserved											alt_gy...	reserved			alt_ac...

Table 36: 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**
...	-	-	reserved															
0x35	FIFO_WATERMARK	0x0000	reserved								fifo_watermark							
0x36	FIFO_CONF	0x0000	reserved				fifo_t...	fifo_g...	fifo_a...	fifo_t...	reserved							fifo_s...
0x37	FIFO_CTRL	0x0000	reserved															fifo_f...
0x38	IO_INT_CTRL	0x0000	reserved					int2_o...	int2_od	int2_lvl	reserved					int1_o...	int1_od	int1_lvl
0x39	INT_CONF	0x0000	reserved															int_latch
0x3A	INT_MAP1	0x0000	tilt_out		sig_motion_out		step_counter_out		step_detector_out		orientation_out		flat_out		any_motion_out		no_motion_out	
0x3B	INT_MAP2	0x0000	fifo_full_int		fifo_watermark_int		acc_drdy_int		gyr_drdy_int		temp_drdy_int		err_status		i3c_out		tap_out	
...	-	-	reserved															
0x40	FEATURE_CTRL	0x0000	reserved															engine_en
0x41	FEATURE_DATA_ADDR	0x0000	reserved					data_address										
0x42	FEATURE_DATA_TX	0x0000	data_tx_value															
0x43	FEATURE_DATA_STATUS	0x0000	reserved														data_t...	data_o...
...	-	-	reserved															
0x45	FEATURE_ENGINE_STATUS	0x0000	reserved										watchdo...	disable...	data_t...	reserved	overload	sleep
...	-	-	reserved															
0x47	FEATURE_EVENT_EXT	0x0000	reserved										t_tap	d_tap	s_tap	orienta...	orientation_port...	
...	-	-	reserved															
0x4F	IO_PDN_CTRL	0x0000	reserved															anaio_...
0x50	IO_SPI_IF	0x0000	reserved															spi3_en
0x51	IO_PAD_STRENGTH	0x000A	reserved												if_i2c...	if_drv		
0x52	IO_I2C_IF	0x0000	reserved														watchdo...	watchdo...
0x53	IO_ODR_DEVIATION	0x0000	reserved											odr_deviation				
...	-	-	reserved															
0x60	ACC_DP_OFF_X	0x0000	reserved		acc_dp_off_x													
0x61	ACC_DP_DGAIN_X	0x0000	reserved								acc_dp_dgain_x							
0x62	ACC_DP_OFF_Y	0x0000	reserved		acc_dp_off_y													
0x63	ACC_DP_DGAIN_Y	0x0000	reserved								acc_dp_dgain_y							
0x64	ACC_DP_OFF_Z	0x0000	reserved		acc_dp_off_z													
0x65	ACC_DP_DGAIN_Z	0x0000	reserved								acc_dp_dgain_z							
0x66	GYR_DP_OFF_X	0x0000	reserved						gyr_dp_off_x									
0x67	GYR_DP_DGAIN_X	0x0000	reserved								gyr_dp_dgain_x							
0x68	GYR_DP_OFF_Y	0x0000	reserved						gyr_dp_off_y									
0x69	GYR_DP_DGAIN_Y	0x0000	reserved								gyr_dp_dgain_y							
0x6A	GYR_DP_OFF_Z	0x0000	reserved						gyr_dp_off_z									
0x6B	GYR_DP_DGAIN_Z	0x0000	reserved								gyr_dp_dgain_z							

Table 36: 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**
...	-	-	reserved															
0x70	I3C_TC_SYNC_TPH	0x0000	i3c_tc_sync_tph															
0x71	I3C_TC_SYNC_TU	0x0000	reserved								i3c_tc_sync_tu							
0x72	I3C_TC_SYNC_ODR	0x0000	reserved								i3c_tc_sync_odr							
...	-	-	reserved															
0x7E	CMD	0x0000	cmd															
0x7F	CFG_RES	0x0000	value_two				reserved								value_one			

6.1.2 Register Map Details

Register (0x00) chip_id

Description: Device identification code. Only bits 0 to 7 contain valid information, the contents of bits 8 to 15 must be ignored.

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	R	R	R	R	R	R
Reset Value	0	1	0	0	1	0	0	0
Content	chip_id							

- reserved:write 0x0.
- `CHIP_ID.chip_id`: (bit offset: 0, bit width: 8) Chip identifier

Register (0x01) err_reg

Description: Reports sensor error conditions

Bit	15	14	13	12	11	10	9	8
Read/Write	R	R	R	R	R/W	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				i3c_er...	reserved		i3c_er...

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

- reserved:write 0x0.
- `ERR_REG.fatal_err`: (bit offset: 0, bit width: 1) Fatal Error, chip is not in operational state (Boot-, power-system). This flag will be reset only by power-on-reset or softreset.
- `ERR_REG.feat_eng_ovrld`: (bit offset: 2, bit width: 1) Overload of the feature engine detected. This flag is clear-on-read.
- `ERR_REG.feat_eng_wd`: (bit offset: 4, bit width: 1) Watchdog timer of the feature engine triggered. This flag is clear-on-read.
- `ERR_REG.acc_conf_err`: (bit offset: 5, bit width: 1) Unsupported accelerometer configuration set by user. This flag will be reset when configuration has been corrected.
- `ERR_REG.gyr_conf_err`: (bit offset: 6, bit width: 1) Unsupported gyroscope configuration set by user. This flag will be reset when configuration has been corrected.
- `ERR_REG.i3c_error0`: (bit offset: 8, bit width: 1) SDR parity error or read abort condition (maximum clock stall time for I3C Read Trasfer) occurred. This flag is a clear-on-read type. It is cleared automatically once read. Refer to the MIPI I3C specification chapter 'Master Clock Stalling' for detail info regarding the read abort condition.
- `ERR_REG.i3c_error3`: (bit offset: 11, bit width: 1) S0/S1 error occurred. When S0/S1 error occurs, the slave will recover automatically after 60 us as if we see a HDR-exit pattern on the bus while the flag will persist for notification purpose. This flag is clear-on-read type. It is cleared automatically once read.

Register (0x02) status

Description: Sensor status flags

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	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	1
Content	drdy_acc	drdy_gyr	drdy_temp	reserved				por_de...

- reserved:write 0x0.
- `STATUS.por_detected`: (bit offset: 0, bit width: 1) ’1’ after device power up or soft-reset. This flag is clear-on-read.
- `STATUS.drdy_temp`: (bit offset: 5, bit width: 1) Data ready for Temperature. This flag is clear-on-read.
- `STATUS.drdy_gyr`: (bit offset: 6, bit width: 1) Data ready for Gyroscope. This flag is clear-on-read.
- `STATUS.drdy_acc`: (bit offset: 7, bit width: 1) Data ready for Accelerometer. This flag is clear-on-read.

Register (0x03) acc_data_x

Description: Acceleration sample, x channel in the default axes configuration

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

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							

- `ACC_DATA_X.acc_x`: (bit offset: 0, bit width: 16)

Register (0x04) acc_data_y

Description: Acceleration sample, y channel in the default axes configuration

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

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							

- `ACC_DATA_Y.acc_y`: (bit offset: 0, bit width: 16)

Register (0x05) acc_data_z

Description: Acceleration sample, z channel in the default axes configuration

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

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							

- `ACC_DATA_Z.acc_z`: (bit offset: 0, bit width: 16)

Register (0x06) gyr_data_x

Description: Angular rate sample, x channel in the default axes configuration

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

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	gyr_x							

- `GYR_DATA_X.gyr_x`: (bit offset: 0, bit width: 16)

Register (0x07) gyr_data_y

Description: Angular rate sample, y channel in the default axes configuration

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

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	gyr_y							

- `GYR_DATA_Y.gyr_y`: (bit offset: 0, bit width: 16)

Register (0x08) gyr_data_z

Description: Angular rate sample, z channel in the default axes configuration

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

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	gyr_z							

- `GYR_DATA_Z.gyr_z`: (bit offset: 0, bit width: 16)

Register (0x09) temp_data

Description: Temperature Data. The resolution is 256 LSB/K. 0x0000 -> 23°C 0x8000 -> invalid

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

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: 16) Temperature value. T (°C) := temp_data/256 + 23

Register (0x0A) sensor_time_0

Description: Sensor time, least significant word (15:0)

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	sensor_time_15_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_15_0							

- `SENSOR_TIME_0.sensor_time_15_0`: (bit offset: 0, bit width: 16) Least significant word of the sensor time (15:0) where 1 LSB corresponds to 39.0625 us

Register (0x0B) sensor_time_1

Description: Sensor time, most significant word (31:16)

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	sensor_time_31_16							

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_31_16							

- `SENSOR_TIME_1.sensor_time_31_16`: (bit offset: 0, bit width: 16) Most significant word of the sensor time (31:16)

Register (0x0C) sat_flags

Description: Saturation flags for each sensor and axis

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		satf_g...	satf_g...	satf_g...	satf_a...	satf_a...	satf_a...

- reserved:write 0x0.
- `SAT_FLAGS.satf_acc_x`: (bit offset: 0, bit width: 1) Saturation flag for the accelerometer x axis in the default axes configuration
- `SAT_FLAGS.satf_acc_y`: (bit offset: 1, bit width: 1) Saturation flag for the accelerometer y axis in the default axes configuration
- `SAT_FLAGS.satf_acc_z`: (bit offset: 2, bit width: 1) Saturation flag for the accelerometer z axis in the default axes configuration
- `SAT_FLAGS.satf_gyr_x`: (bit offset: 3, bit width: 1) Saturation flag for the gyroscope x axis in the default axes configuration
- `SAT_FLAGS.satf_gyr_y`: (bit offset: 4, bit width: 1) Saturation flag for the gyroscope y axis in the default axes configuration
- `SAT_FLAGS.satf_gyr_z`: (bit offset: 5, bit width: 1) Saturation flag for the gyroscope z axis in the default axes configuration

Register (0x0D) int_status_int1

Description: INT1 Status Register. This register is clear-on-read.

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	int1_f...	int1_fwm	int1_a...	int1_g...	int1_t...	int1_e...	int1_i3c	int1_tap

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	int1_tilt	int1_s...	int1_s...	int1_s...	int1_o...	int1_flat	int1_a...	int1_n...

- `INT_STATUS_INT1.int1_no_motion`: (bit offset: 0, bit width: 1) No motion detection output
- `INT_STATUS_INT1.int1_any_motion`: (bit offset: 1, bit width: 1) Any motion detection output
- `INT_STATUS_INT1.int1_flat`: (bit offset: 2, bit width: 1) Flat detection output
- `INT_STATUS_INT1.int1_orientation`: (bit offset: 3, bit width: 1) Orientation detection output
- `INT_STATUS_INT1.int1_step_detector`: (bit offset: 4, bit width: 1) Step detector output
- `INT_STATUS_INT1.int1_step_counter`: (bit offset: 5, bit width: 1) Step counter watermark output
- `INT_STATUS_INT1.int1_sig_motion`: (bit offset: 6, bit width: 1) Sigmotion detection output
- `INT_STATUS_INT1.int1_tilt`: (bit offset: 7, bit width: 1) Tilt detection output
- `INT_STATUS_INT1.int1_tap`: (bit offset: 8, bit width: 1) Tap detection output
- `INT_STATUS_INT1.int1_i3c`: (bit offset: 9, bit width: 1) I3C tc sync data ready interrupt
- `INT_STATUS_INT1.int1_err_status`: (bit offset: 10, bit width: 1) Feature engine error or status change
- `INT_STATUS_INT1.int1_temp_drdy`: (bit offset: 11, bit width: 1) Temperature data ready interrupt
- `INT_STATUS_INT1.int1_gyr_drdy`: (bit offset: 12, bit width: 1) Gyroscope data ready interrupt
- `INT_STATUS_INT1.int1_acc_drdy`: (bit offset: 13, bit width: 1) Accelerometer data ready interrupt
- `INT_STATUS_INT1.int1_fwm`: (bit offset: 14, bit width: 1) FIFO watermark interrupt
- `INT_STATUS_INT1.int1_ffull`: (bit offset: 15, bit width: 1) FIFO full interrupt

Register (0x0E) int_status_int2

Description: INT2 Status Register. This register is clear-on-read.

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	int2_f...	int2_fwm	int2_a...	int2_g...	int2_t...	int2_e...	int2_i3c	int2_tap

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	int2_tilt	int2_s...	int2_s...	int2_s...	int2_o...	int2_flat	int2_a...	int2_n...

- `INT_STATUS_INT2.int2_no_motion`: (bit offset: 0, bit width: 1) No motion detection output
- `INT_STATUS_INT2.int2_any_motion`: (bit offset: 1, bit width: 1) Any motion detection output
- `INT_STATUS_INT2.int2_flat`: (bit offset: 2, bit width: 1) Flat detection output
- `INT_STATUS_INT2.int2_orientation`: (bit offset: 3, bit width: 1) Orientation detection output
- `INT_STATUS_INT2.int2_step_detector`: (bit offset: 4, bit width: 1) Step detector output
- `INT_STATUS_INT2.int2_step_counter`: (bit offset: 5, bit width: 1) Step counter watermark output
- `INT_STATUS_INT2.int2_sig_motion`: (bit offset: 6, bit width: 1) Sigmotion detection output
- `INT_STATUS_INT2.int2_tilt`: (bit offset: 7, bit width: 1) Tilt detection output
- `INT_STATUS_INT2.int2_tap`: (bit offset: 8, bit width: 1) Tap detection output
- `INT_STATUS_INT2.int2_i3c`: (bit offset: 9, bit width: 1) I3C tc sync data ready interrupt
- `INT_STATUS_INT2.int2_err_status`: (bit offset: 10, bit width: 1) Feature engine error or status change
- `INT_STATUS_INT2.int2_temp_drdy`: (bit offset: 11, bit width: 1) Temperature data ready interrupt
- `INT_STATUS_INT2.int2_gyr_drdy`: (bit offset: 12, bit width: 1) Gyroscope data ready interrupt
- `INT_STATUS_INT2.int2_acc_drdy`: (bit offset: 13, bit width: 1) Accelerometer data ready interrupt
- `INT_STATUS_INT2.int2_fwm`: (bit offset: 14, bit width: 1) FIFO watermark interrupt
- `INT_STATUS_INT2.int2_ffull`: (bit offset: 15, bit width: 1) FIFO full interrupt

Register (0x0F) int_status_ibi

Description: I3C IBI Status Register. This register is clear-on-read.

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	ibi_ffull	ibi_fwm	ibi_ac...	ibi_gy...	ibi_te...	ibi_er...	ibi_i3c	ibi_tap

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	ibi_tilt	ibi_si...	ibi_st...	ibi_st...	ibi_or...	ibi_flat	ibi_an...	ibi_no...

- `INT_STATUS_IBI.ibi_no_motion`: (bit offset: 0, bit width: 1) No_motion output
- `INT_STATUS_IBI.ibi_any_motion`: (bit offset: 1, bit width: 1) Any_motion output
- `INT_STATUS_IBI.ibi_flat`: (bit offset: 2, bit width: 1) Flat output
- `INT_STATUS_IBI.ibi_orientation`: (bit offset: 3, bit width: 1) Orientation output
- `INT_STATUS_IBI.ibi_step_detector`: (bit offset: 4, bit width: 1) Step detector output
- `INT_STATUS_IBI.ibi_step_counter`: (bit offset: 5, bit width: 1) Step counter watermark output
- `INT_STATUS_IBI.ibi_sig_motion`: (bit offset: 6, bit width: 1) Sigmotion output
- `INT_STATUS_IBI.ibi_tilt`: (bit offset: 7, bit width: 1) Tilt output
- `INT_STATUS_IBI.ibi_tap`: (bit offset: 8, bit width: 1) Tap output
- `INT_STATUS_IBI.ibi_i3c`: (bit offset: 9, bit width: 1) I3C tc sync data ready interrupt
- `INT_STATUS_IBI.ibi_err_status`: (bit offset: 10, bit width: 1) Feature engine error or status change
- `INT_STATUS_IBI.ibi_temp_drdy`: (bit offset: 11, bit width: 1) Temperature data ready interrupt
- `INT_STATUS_IBI.ibi_gyr_drdy`: (bit offset: 12, bit width: 1) Gyroscope data ready interrupt
- `INT_STATUS_IBI.ibi_acc_drdy`: (bit offset: 13, bit width: 1) Accelerometer data ready interrupt
- `INT_STATUS_IBI.ibi_fwm`: (bit offset: 14, bit width: 1) FIFO watermark interrupt
- `INT_STATUS_IBI.ibi_ffull`: (bit offset: 15, bit width: 1) FIFO full interrupt

Register (0x10) feature_io0

Description: Feature engine configuration, before setting/changing an active configuration the register must be cleared (set to 0)

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	i3c_sy...	tap_de...	tap_de...	tap_de...	tilt_en	sig_mo...	step_c...	step_d...

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	orienta...	flat_en	any_mo...	any_mo...	any_mo...	no_mot...	no_mot...	no_mot...

- `FEATURE_IO0.no_motion_x_en`: (bit offset: 0, bit width: 1) Enables no motion feature for X-axis
- `FEATURE_IO0.no_motion_y_en`: (bit offset: 1, bit width: 1) Enables no motion feature for Y-axis
- `FEATURE_IO0.no_motion_z_en`: (bit offset: 2, bit width: 1) Enables no motion feature for Z-axis
- `FEATURE_IO0.any_motion_x_en`: (bit offset: 3, bit width: 1) Enables any motion feature for X-axis
- `FEATURE_IO0.any_motion_y_en`: (bit offset: 4, bit width: 1) Enables any motion feature for Y-axis
- `FEATURE_IO0.any_motion_z_en`: (bit offset: 5, bit width: 1) Enables any motion feature for Z-axis
- `FEATURE_IO0.flat_en`: (bit offset: 6, bit width: 1) Enables flat feature
- `FEATURE_IO0.orientation_en`: (bit offset: 7, bit width: 1) Enables orientation feature
- `FEATURE_IO0.step_detector_en`: (bit offset: 8, bit width: 1) Enables step detector feature
- `FEATURE_IO0.step_counter_en`: (bit offset: 9, bit width: 1) Enables step counter feature
- `FEATURE_IO0.sig_motion_en`: (bit offset: 10, bit width: 1) Enables significant motion feature
- `FEATURE_IO0.tilt_en`: (bit offset: 11, bit width: 1) Enables tilt feature
- `FEATURE_IO0.tap_detector_s_tap_en`: (bit offset: 12, bit width: 1) Enables single tap feature
- `FEATURE_IO0.tap_detector_d_tap_en`: (bit offset: 13, bit width: 1) Enables double tap feature
- `FEATURE_IO0.tap_detector_t_tap_en`: (bit offset: 14, bit width: 1) Enables triple tap feature
- `FEATURE_IO0.i3c_sync_en`: (bit offset: 15, bit width: 1) Enables I3C TC-sync feature.

Register (0x11) feature_io1

Description: Feature engine I/O register 0

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			state		axis_m...	reserved	

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	sample:.	st_result	gyro_s...	sc_st:.	error_status			

- reserved:write 0x0.
- `FEATURE_IO1.error_status`: (bit offset: 0, bit width: 4) Error and status information
  Following values can be set to or read from the field error_status:

Value	Description
0b0000 (0x0)	Feature engine still inactive
0b0001 (0x1)	Feature engine activated
0b0011 (0x3)	Configuration string download failed
0b0101 (0x5)	No error
0b0110 (0x6)	Axis map command was not processed because either a sensor was active or self-calibration or self-test was ongoing
0b1000 (0x8)	I3C TC-sync error because either I3C TC-sync enable request was sent while auto-low-power feature was active or I3C TC-sync configuration command was sent with invalid TPH, TU and ODR values. For later case, invalid configuration parameters TPH, TU and ODR will not be used.
0b1001 (0x9)	Ongoing self-calibration (gyroscope only) or self-test (gyroscope only) was aborted. The command was aborted either due to device movements or due to the abort command (self-calibration only) or due to a request to enable I3C TC-sync feature (self-calibration only).
0b1010 (0xA)	Self-calibration (gyroscope only) command ignored because either self-calibration or self-test or I3C TC-sync was ongoing
0b1011 (0xB)	Self-test (accelerometer and/or gyroscope) command ignored because either self-calibration or self-test or I3C TC-sync was ongoing
0b1100 (0xC)	Self-calibration (gyroscope only) or self-test (accelerometer and/or gyroscope) command was not processed because pre-conditions were not met. Either accelerometer was not configured correctly (self-test and self-calibration gyroscope only) or auto-low-power feature was active.
0b1101 (0xD)	Auto-mode change feature was enabled or illegal sensor configuration change detected in ACC_CONF/GYR_CONF while self-calibration or self-test was ongoing. Self-calibration and self-test results may be inaccurate.
0b1110 (0xE)	I3C TC-sync enable request was sent while self-test (accelerometer and/or gyroscope) was ongoing. I3C TC-sync will be enabled at the end of self-test.
0b1111 (0xF)	Illegal sensor configuration change detected in ACC_CONF/GYR_CONF while I3C TC-sync was active. Sensors are re-configured is to requested I3C TC-sync ODR.

- `FEATURE_IO1.sc_st_complete`: (bit offset: 4, bit width: 1) Self-calibration (gyroscope only) or self-test (accelerometer and/or gyroscope) execution status. 0 indicates that the procedure is ongoing. 1 indicates that the procedure is completed.
- `FEATURE_IO1.gyro_sc_result`: (bit offset: 5, bit width: 1) Gyroscope self-calibration result (1=OK, 0=Not OK). Bit sc_st_complete should be 1 prior to reading this bit.
- `FEATURE_IO1.st_result`: (bit offset: 6, bit width: 1) Accelerometer and/or gyroscope self-test result (1=OK, 0=Not OK). Bit sc_st_complete should be 1 prior to reading this bit.
- `FEATURE_IO1.sample_rate_err`: (bit offset: 7, bit width: 1) Insufficient sample rate for either 50Hz or 200Hz or I3C TC-sync feature
- `FEATURE_IO1.axis_map_complete`: (bit offset: 10, bit width: 1) Axis mapping completed
- `FEATURE_IO1.state`: (bit offset: 11, bit width: 2) Current state of the system
  Following values can be set to or read from the field state:

Value	Description
0b00 (0x0)	System in feature mode
0b01 (0x1)	System is executing self-calibration of gyroscope in feature mode
0b10 (0x2)	System in self-test mode
0b11 (0x3)	System in error mode

Register (0x12) feature_io2

Description: Feature engine I/O register 1

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	step_counter_out_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	step_counter_out_0							

- `FEATURE_IO2.startup_config_0`: (bit offset: 0, bit width: 16) Before feature engine enable: Feature engine start-up configuration
- `FEATURE_IO2.step_counter_out_0`: (bit offset: 0, bit width: 16) After feature engine enable: Step counter value word-0 (low word)

Register (0x13) feature_io3

Description: Feature engine I/O register 2

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	step_counter_out_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	step_counter_out_1							

- `FEATURE_IO3.startup_config_1`: (bit offset: 0, bit width: 16) Before feature engine enable: Feature engine start-up configuration
- `FEATURE_IO3.step_counter_out_1`: (bit offset: 0, bit width: 16) After feature engine enable: Step counter value word-1 (high word)

Register (0x14) feature_io_status

Description: Feature I/O synchronization status and trigger

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							feature...

- reserved:write 0x0.
- `FEATURE_IO_STATUS.feature_io_status`: (bit offset: 0, bit width: 1) On read: data has been written by the feature engine On write: data written by the host shall be sent to the feature engine

Register (0x15) fifo_fill_level

Description: FIFO fill state in words

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					fifo_fill_level		

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							

- reserved:write 0x0.
- `FIFO_FILL_LEVEL.fifo_fill_level`: (bit offset: 0, bit width: 11) Current fill level of FIFO buffer An empty FIFO corresponds to 0x000. The word counter may be reset by reading out all frames from the FIFO buffer or when the FIFO is reset through fifo_flush. The word counter is updated each time a complete frame was read or written.

Register (0x16) fifo_data

Description: FIFO data output register

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	fifo_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	fifo_data							

- `FIFO_DATA.fifo_data`: (bit offset: 0, bit width: 16) FIFO read data (16 bits) Data format depends on the setting of register FIFO_CONF. The FIFO data are organized in frames. The new data flag is preserved. Read burst access must be used, the address will not increment when the read burst reads at the address of FIFO_DATA. When a frame is only partially read out it is retransmitted including the header at the next readout.

Register (0x20) acc_conf

Description: Sets the output data rate, bandwidth, range and the mode of the accelerometer

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

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	1	0	1	0	0	0
Content	acc_bw	acc_range			acc_odr			

- reserved:write 0x0.
- `ACC_CONF.acc_odr`: (bit offset: 0, bit width: 4) Sample rate (ODR) in Hz
  Following values can be set to or read from the field acc_odr:

Value	Description
0b0001 (0x1)	ODR = 0.78125Hz
0b0010 (0x2)	ODR = 1.5625Hz
0b0011 (0x3)	ODR = 3.125Hz
0b0100 (0x4)	ODR = 6.25Hz
0b0101 (0x5)	ODR = 12.5Hz
0b0110 (0x6)	ODR = 25Hz
0b0111 (0x7)	ODR = 50Hz
0b1000 (0x8)	ODR = 100Hz
0b1001 (0x9)	ODR = 200Hz
0b1010 (0xA)	ODR = 400Hz
0b1011 (0xB)	ODR = 800Hz
0b1100 (0xC)	ODR = 1.6kHz
0b1101 (0xD)	ODR = 3.2kHz
0b1110 (0xE)	ODR = 6.4kHz

- `ACC_CONF.acc_range`: (bit offset: 4, bit width: 3) Full scale, Resolution
  Following values can be set to or read from the field acc_range:

Value	Description
0b000 (0x0)	+/-2g, 16.38 LSB/mg
0b001 (0x1)	+/-4g, 8.19 LSB/mg
0b010 (0x2)	+/-8g, 4.10 LSB/mg
0b011 (0x3)	+/-16g, 2.05 LSB/mg

- `ACC_CONF.acc_bw`: (bit offset: 7, bit width: 1) Configure the -3dB cut-off frequency for the accelerometer
  Following values can be set to or read from the field acc_bw:

Value	Description
0b0 (0x0)	BW = acc_odr/2
0b1 (0x1)	BW = acc_odr/4

- `ACC_CONF.acc_avg_num`: (bit offset: 8, bit width: 3) Numbers of samples to be averaged
  Following values can be set to or read from the field acc_avg_num:

Value	Description
0b000 (0x0)	No averaging; pass sample without filtering
0b001 (0x1)	Averaging of 2 samples
0b010 (0x2)	Averaging of 4 samples
0b011 (0x3)	Averaging of 8 samples
0b100 (0x4)	Averaging of 16 samples
0b101 (0x5)	Averaging of 32 samples
0b110 (0x6)	Averaging of 64 samples

- `ACC_CONF.acc_mode`: (bit offset: 12, bit width: 3) Operation modes for the accelerometer
  Following values can be set to or read from the field acc_mode:

Value	Description
0b000 (0x0)	Disables the accelerometer
0b011 (0x3)	Enables the accelerometer with sensing operated in duty-cycling
0b100 (0x4)	Enables the accelerometer in a continuous operation mode with reduced current
0b111 (0x7)	Enables the accelerometer in high performance mode

Register (0x21) gyr_conf

Description: Sets the output data rate, bandwidth, range and the mode of the Gyroscope in the sensor

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

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	gyr_bw	gyr_range			gyr_odr			

- reserved:write 0x0.
- `GYR_CONF.gyr_odr`: (bit offset: 0, bit width: 4) Sample rate (ODR) in Hz
  Following values can be set to or read from the field gyr_odr:

Value	Description
0b0001 (0x1)	ODR = 0.78125Hz
0b0010 (0x2)	ODR = 1.5625Hz
0b0011 (0x3)	ODR = 3.125Hz
0b0100 (0x4)	ODR = 6.25Hz
0b0101 (0x5)	ODR = 12.5Hz
0b0110 (0x6)	ODR = 25Hz
0b0111 (0x7)	ODR = 50Hz
0b1000 (0x8)	ODR = 100Hz
0b1001 (0x9)	ODR = 200Hz
0b1010 (0xA)	ODR = 400Hz
0b1011 (0xB)	ODR = 800Hz
0b1100 (0xC)	ODR = 1.6kHz
0b1101 (0xD)	ODR = 3.2kHz
0b1110 (0xE)	ODR = 6.4kHz

- `GYR_CONF.gyr_range`: (bit offset: 4, bit width: 3) Full scale, Resolution
  Following values can be set to or read from the field gyr_range:

Value	Description
0b000 (0x0)	+/-125°/s, 262.144 LSB/°/s
0b001 (0x1)	+/-250°/s, 131.072 LSB/°/s
0b010 (0x2)	+/-500°/s, 65.536 LSB/°/s
0b011 (0x3)	+/-1000°/s, 32.768 LSB/°/s
0b100 (0x4)	+/-2000°/s, 16.4 LSB/°/s

- `GYR_CONF.gyr_bw`: (bit offset: 7, bit width: 1) Configure the -3dB cut-off frequency for the gyroscope
  Following values can be set to or read from the field gyr_bw:

Value	Description
0b0 (0x0)	BW = gyr_odr/2
0b1 (0x1)	BW = gyr_odr/4

- `GYR_CONF.gyr_avg_num`: (bit offset: 8, bit width: 3) Numbers of samples to be averaged
  Following values can be set to or read from the field gyr_avg_num:

Value	Description
0b000 (0x0)	No averaging; pass sample without filtering
0b001 (0x1)	averaging of 2 samples
0b010 (0x2)	Averaging of 4 samples
0b011 (0x3)	Averaging of 8 samples
0b100 (0x4)	Averaging of 16 samples
0b101 (0x5)	Averaging of 32 samples
0b110 (0x6)	Averaging of 64 samples

- `GYR_CONF.gyr_mode`: (bit offset: 12, bit width: 3) Operation modes for the gyroscope
  Following values can be set to or read from the field gyr_mode:

Value	Description
0b000 (0x0)	Disables the gyroscope
0b001 (0x1)	Disables the gyroscope but keep the gyroscope drive enabled
0b011 (0x3)	Enables the gyroscope with sensing operated in duty-cycling
0b100 (0x4)	Enables the gyroscope in a continuous operation mode with reduced current
0b111 (0x7)	Enables the gyroscope in high performance mode

Register (0x28) alt_acc_conf

Description: Sets the alternative output data rate, bandwidth, range and the mode of the accelerometer

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

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	1	1	0
Content	reserved				alt_acc_odr			

- reserved:write 0x0.
- `ALT_ACC_CONF.alt_acc_odr`: (bit offset: 0, bit width: 4) Sample rate (ODR) in Hz
  Following values can be set to or read from the field alt_acc_odr:

Value	Description
0b0001 (0x1)	ODR = 0.78125Hz
0b0010 (0x2)	ODR = 1.5625Hz
0b0011 (0x3)	ODR = 3.125Hz
0b0100 (0x4)	ODR = 6.25Hz
0b0101 (0x5)	ODR = 12.5Hz
0b0110 (0x6)	ODR = 25Hz
0b0111 (0x7)	ODR = 50Hz
0b1000 (0x8)	ODR = 100Hz
0b1001 (0x9)	ODR = 200Hz
0b1010 (0xA)	ODR = 400Hz
0b1011 (0xB)	ODR = 800Hz
0b1100 (0xC)	ODR = 1.6kHz
0b1101 (0xD)	ODR = 3.2kHz
0b1110 (0xE)	ODR = 6.4kHz

- `ALT_ACC_CONF.alt_acc_avg_num`: (bit offset: 8, bit width: 3) Numbers of samples to be averaged
  Following values can be set to or read from the field alt_acc_avg_num:

Value	Description
0b000 (0x0)	No averaging; pass sample without filtering
0b001 (0x1)	Averaging of 2 samples
0b010 (0x2)	Averaging of 4 samples
0b011 (0x3)	Averaging of 8 samples
0b100 (0x4)	Averaging of 16 samples
0b101 (0x5)	Averaging of 32 samples
0b110 (0x6)	Averaging of 64 samples

- `ALT_ACC_CONF.alt_acc_mode`: (bit offset: 12, bit width: 3) Operation modes for the accelerometer
  Following values can be set to or read from the field alt_acc_mode:

Value	Description
0b000 (0x0)	Disables the accelerometer
0b011 (0x3)	Enables the accelerometer with sensing operated in duty-cycling
0b100 (0x4)	Enables the accelerometer in a continuous operation mode with reduced current
0b111 (0x7)	Enables the accelerometer in high performance mode

Register (0x29) alt_gyr_conf

Description: Sets the alternative output data rate, bandwidth, range and the mode of the gyroscope

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

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	1	1	0
Content	reserved				alt_gyr_odr			

- reserved:write 0x0.
- `ALT_GYR_CONF.alt_gyr_odr`: (bit offset: 0, bit width: 4) Sample rate (ODR) in Hz
  Following values can be set to or read from the field alt_gyr_odr:

Value	Description
0b0001 (0x1)	ODR = 0.78125Hz
0b0010 (0x2)	ODR = 1.5625Hz
0b0011 (0x3)	ODR = 3.125Hz
0b0100 (0x4)	ODR = 6.25Hz
0b0101 (0x5)	ODR = 12.5Hz
0b0110 (0x6)	ODR = 25Hz
0b0111 (0x7)	ODR = 50Hz
0b1000 (0x8)	ODR = 100Hz
0b1001 (0x9)	ODR = 200Hz
0b1010 (0xA)	ODR = 400Hz
0b1011 (0xB)	ODR = 800Hz
0b1100 (0xC)	ODR = 1.6kHz
0b1101 (0xD)	ODR = 3.2kHz
0b1110 (0xE)	ODR = 6.4kHz

- `ALT_GYR_CONF.alt_gyr_avg_num`: (bit offset: 8, bit width: 3) Numbers of samples to be averaged
  Following values can be set to or read from the field alt_gyr_avg_num:

Value	Description
0b000 (0x0)	No averaging; pass sample without filtering
0b001 (0x1)	Averaging of 2 samples
0b010 (0x2)	Averaging of 4 samples
0b011 (0x3)	Averaging of 8 samples
0b100 (0x4)	Averaging of 16 samples
0b101 (0x5)	Averaging of 32 samples
0b110 (0x6)	Averaging of 64 samples

- `ALT_GYR_CONF.alt_gyr_mode`: (bit offset: 12, bit width: 3) Operation modes for the gyroscope
  Following values can be set to or read from the field alt_gyr_mode:

Value	Description
0b000 (0x0)	Disables the gyroscope
0b001 (0x1)	Disables the gyroscope but keep the gyroscope drive enabled
0b011 (0x3)	Enables the gyroscope with sensing operated in duty-cycling
0b100 (0x4)	Enables the gyroscope in a continuous operation mode with reduced current
0b111 (0x7)	Enables the gyroscope in high performance mode

Register (0x2A) alt_conf

Description: Alternate configuration control

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							alt_rs...

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

- reserved:write 0x0.
- `ALT_CONF.alt_acc_en`: (bit offset: 0, bit width: 1) Enables switching possiblility to alternate configuration for accelerometer.
- `ALT_CONF.alt_gyr_en`: (bit offset: 4, bit width: 1) Enables switching possiblility to alternate configuration for gyroscope.
- `ALT_CONF.alt_rst_conf_write_en`: (bit offset: 8, bit width: 1) If enabled, any write to ACC_CONF or GYR_CONF will instanly switch back to associated user configuration.

Register (0x2B) alt_status

Description: Reports the active configuration for the accelerometer and gyroscope

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			alt_gy...	reserved			alt_ac...

- reserved:write 0x0.
- `ALT_STATUS.alt_acc_active`: (bit offset: 0, bit width: 1) accel is using ALT_ACC_CONF if set; ACC_CONF otherwise
- `ALT_STATUS.alt_gyr_active`: (bit offset: 4, bit width: 1) gyro is using ALT_GYR_CONF if set; GYR_CONF otherwise

Register (0x35) fifo_watermark

Description: FIFO watermark level

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

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							

- reserved:write 0x0.
- `FIFO_WATERMARK.fifo_watermark`: (bit offset: 0, bit width: 10) Trigger an interrupt when FIFO contains fifo_watermark words

Register (0x36) fifo_conf

Description: Configuration of the FIFO data buffer behaviour

Bit	15	14	13	12	11	10	9	8
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				fifo_t...	fifo_g...	fifo_a...	fifo_t...

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							fifo_s...

- reserved:write 0x0.
- `FIFO_CONF.fifo_stop_on_full`: (bit offset: 0, bit width: 1) Configure whether to overwrite oldest samples when the FIFO data buffer is full
  Following values can be set to or read from the field fifo_stop_on_full:

Value	Description
0b0 (0x0)	Continue writing to the data buffer by overwriting oldest samples
0b1 (0x1)	Stop writing into the FIFO data buffer when full

- `FIFO_CONF.fifo_time_en`: (bit offset: 8, bit width: 1) Write the sensor time into the FIFO data buffer.
  Following values can be set to or read from the field fifo_time_en:

Value	Description
0b0 (0x0)	No sensortime is written
0b1 (0x1)	Sensortime is written

- `FIFO_CONF.fifo_acc_en`: (bit offset: 9, bit width: 1) Write 3D acceleration samples from the accelerometer into the FIFO data buffer
  Following values can be set to or read from the field fifo_acc_en:

Value	Description
0b0 (0x0)	No accelerometer data is written
0b1 (0x1)	Accelerometer data is written

- `FIFO_CONF.fifo_gyr_en`: (bit offset: 10, bit width: 1) Write 3D angular rate samples from the gyroscope into the FIFO data buffer

Following values can be set to or read from the field fifo_gyr_en:

Value	Description
0b0 (0x0)	No gyroscope data is written
0b1 (0x1)	Gyroscope data is written

- `FIFO_CONF.fifo_temp_en`: (bit offset: 11, bit width: 1) Write temperature samples into the FIFO data buffer
  Following values can be set to or read from the field fifo_temp_en:

Value	Description
0b0 (0x0)	No temperature data is written
0b1 (0x1)	Temperature data is written

Register (0x37) fifo_ctrl

Description: Control of the FIFO data buffer

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	R	R	R	R	R	W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							fifo_f...

- reserved:write 0x0.
- `FIFO_CTRL.fifo_flush`: (bit offset: 0, bit width: 1) Writing 0b1 clears the FIFO data buffer.

Register (0x38) io_int_ctrl

Description: Configures the electrical behavior of the interrupt pins

Bit	15	14	13	12	11	10	9	8
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					int2_o...	int2_od	int2_lvl

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					int1_o...	int1_od	int1_lvl

- reserved:write 0x0.
- `IO_INT_CTRL.int1_lvl`: (bit offset: 0, bit width: 1) Configure level of INT1 pin
  Following values can be set to or read from the field int1_lvl:

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

- `IO_INT_CTRL.int1_od`: (bit offset: 1, bit width: 1) Configure behavior of INT1 pin
  Following values can be set to or read from the field int1_od:

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

- `IO_INT_CTRL.int1_output_en`: (bit offset: 2, bit width: 1) Output enable for INT1 pin
- `IO_INT_CTRL.int2_lvl`: (bit offset: 8, bit width: 1) Configure level of INT2 pin
  Following values can be set to or read from the field int2_lvl:

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

- `IO_INT_CTRL.int2_od`: (bit offset: 9, bit width: 1) Configure behavior of INT2 pin
  Following values can be set to or read from the field int2_od:

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

- `IO_INT_CTRL.int2_output_en`: (bit offset: 10, bit width: 1) Output enable for INT2 pin

Register (0x39) int_conf

Description: Interrupt Configuration Register.

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							int_latch

- reserved:write 0x0.
- `INT_CONF.int_latch`: (bit offset: 0, bit width: 1) Configuration of the interrupt clear behaviour as not latched or permanently latched.
  Following values can be set to or read from the field int_latch:

Value	Description
0b0 (0x0)	non latched
0b1 (0x1)	permanent latched

Register (0x3A) int_map1

Description: Mapping of feature engine interrupts to outputs

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	tilt_out		sig_motion_out		step_counter_out		step_detector_out	

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	orientation_out		flat_out		any_motion_out		no_motion_out	

- `INT_MAP1.no_motion_out`: (bit offset: 0, bit width: 2) Map no motion output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field no_motion_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.any_motion_out`: (bit offset: 2, bit width: 2) Map any motion output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field any_motion_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.flat_out`: (bit offset: 4, bit width: 2) Map flat output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field flat_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.orientation_out`: (bit offset: 6, bit width: 2) Map orientation output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field orientation_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.step_detector_out`: (bit offset: 8, bit width: 2) Map step_detector output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field step_detector_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.step_counter_out`: (bit offset: 10, bit width: 2) Map step_counter watermark output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field step_counter_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.sig_motion_out`: (bit offset: 12, bit width: 2) Map sigmotion output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field sig_motion_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP1.tilt_out`: (bit offset: 14, bit width: 2) Map tilt output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field tilt_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

Register (0x3B) int_map2

Description: Mapping of feature engine interrupts, data ready interrupts for signals and FIFO buffer interrupts to outputs

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	fifo_full_int		fifo_watermark_int		acc_drdy_int		gyr_drdy_int	

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	temp_drdy_int		err_status		i3c_out		tap_out	

- `INT_MAP2.tap_out`: (bit offset: 0, bit width: 2) Map tap output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field tap_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.i3c_out`: (bit offset: 2, bit width: 2) Map i3c output to either INT1 or INT2 or IBI
  Following values can be set to or read from the field i3c_out:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.err_status`: (bit offset: 4, bit width: 2) Map feature engine's error or status change to either INT1 or INT2 or IBI
  Following values can be set to or read from the field err_status:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.temp_drdy_int`: (bit offset: 6, bit width: 2) Map temperature data ready interrupt to either INT1 or INT2 or IBI
  Following values can be set to or read from the field temp_drdy_int:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.gyr_drdy_int`: (bit offset: 8, bit width: 2) Map gyro data ready interrupt to either INT1 or INT2 or IBI
  Following values can be set to or read from the field gyr_drdy_int:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.acc_drdy_int`: (bit offset: 10, bit width: 2) Map accel data ready interrupt to either INT1 or INT2 or IBI
  Following values can be set to or read from the field acc_drdy_int:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.fifo_watermark_int`: (bit offset: 12, bit width: 2) Map FIFO watermark interrupt to either INT1 or INT2 or IBI
  Following values can be set to or read from the field fifo_watermark_int:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

- `INT_MAP2.fifo_full_int`: (bit offset: 14, bit width: 2) Map FIFO full interrupt to either INT1 or INT2 or IBI Following values can be set to or read from the field fifo_full_int:

Value	Description
0b00 (0x0)	Interrupt disabled
0b01 (0x1)	Mapped to INT1
0b10 (0x2)	Mapped to INT2
0b11 (0x3)	Mapped to I3C IBI

Register (0x40) feature_ctrl

Description: Feature engine control register

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							engine_en

- reserved:write 0x0.
- `FEATURE_CTRL.engine_en`: (bit offset: 0, bit width: 1) Enable or disable the feature engine. Note: a soft-reset is required to re-enable the feature engine.

Register (0x41) feature_data_addr

Description: Adress register for feature data: configurations and extended output

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
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					data_address		

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	data_address							

- reserved:write 0x0.
- `FEATURE_DATA_ADDR.data_address`: (bit offset: 0, bit width: 11) Start address for the feature data, that is feature configurations and extended feature output

Register (0x42) feature_data_tx

Description: I/O port for the data values of the feature engine

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	data_tx_value							

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	data_tx_value							

- `FEATURE_DATA_TX.data_tx_value`: (bit offset: 0, bit width: 16) Data port for DMA transfers.

Register (0x43) feature_data_status

Description: Status of the data access to the feature engine

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						data_t...	data_o...

- reserved:write 0x0.
- `FEATURE_DATA_STATUS.data_outofbound_err`: (bit offset: 0, bit width: 1) Too much data read or written to the feature engine. Bit will be reset upon next data transfer to or from the feature engine.
- `FEATURE_DATA_STATUS.data_tx_ready`: (bit offset: 1, bit width: 1) Data is writeable to the feature engine or readable from the feature engine.

Register (0x45) feature_engine_status

Description: Status of the feature engine

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		watchdo...	disable...	data_t...	reserved	overload	sleep

- reserved:write 0x0.
- `FEATURE_ENGINE_STATUS.sleep`: (bit offset: 0, bit width: 1) Feature engine halted or sleeping.
- `FEATURE_ENGINE_STATUS.overload`: (bit offset: 1, bit width: 1) Transfer of data to or from the feature engine is ongoing.
- `FEATURE_ENGINE_STATUS.data_tx_active`: (bit offset: 3, bit width: 1) DMA controller has started DMA and DMA transactions are ongoing
- `FEATURE_ENGINE_STATUS.disabled_by_host`: (bit offset: 4, bit width: 1) Feature engine was disabled by the host. Perform a soft-reset to re-enable the feature engine.
- `FEATURE_ENGINE_STATUS.watchdog_not_ack`: (bit offset: 5, bit width: 1) The feature engine did not acknowledge its internal watchdog in time. Perform a soft-reset to re-enable the feature engine.

Register (0x47) feature_event_ext

Description: Register of extended data on feature events. The register content is valid in combination with an active bit in INT_STATUS_INT1/2

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		t_tap	d_tap	s_tap	orienta...	orientation_portrait_landscape	

- reserved:write 0x0.
- `FEATURE_EVENT_EXT.orientation_portrait_landscape`: (bit offset: 0, bit width: 2) Output value of the orientation detection feature. Value after device initialization is 0b00 i.e. Portrait upright
  Following values can be set to or read from the field orientation_portrait_landscape:

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

- `FEATURE_EVENT_EXT.orientation_faceup_down`: (bit offset: 2, bit width: 1) Output value of face down face up orientation (only if ud_en is enabled). Value after device initialization is 0b0 i.e. Face up
  Following values can be set to or read from the field orientation_faceup_down:

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

- `FEATURE_EVENT_EXT.s_tap`: (bit offset: 3, bit width: 1) Single tap detected
- `FEATURE_EVENT_EXT.d_tap`: (bit offset: 4, bit width: 1) Double tap detected
- `FEATURE_EVENT_EXT.t_tap`: (bit offset: 5, bit width: 1) Triple tap detected

Register (0x4F) io_pdn_ctrl

Description: Pull down behavior control

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							anaio_...

- reserved:write 0x0.
- `IO_PDN_CTRL.anaio_pdn_dis`: (bit offset: 0, bit width: 1) Disable the pull down on the PIN2 and PIN3

Register (0x50) io_spi_if

Description: Configuration register for the SPI interface

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							spi3_en

- reserved:write 0x0.
- `IO_SPI_IF.spi3_en`: (bit offset: 0, bit width: 1) Enable or disable the 3-wire SPI interface. 0b0 enables the default 4-wire configuration, 0b1 enables the 3-wire configuration.

Register (0x51) io_pad_strength

Description: Configuration register for the electrical characteristics of the pads

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	R	R	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	0	1	0
Content	reserved				if_i2c...	if_drv		

- reserved:write 0x0.
- `IO_PAD_STRENGTH.if_drv`: (bit offset: 0, bit width: 3) Generic drive strength control for the output pads: 0:=weakest; 7:=strongest
- `IO_PAD_STRENGTH.if_i2c_boost`: (bit offset: 3, bit width: 1) Enable or disable the drive strength for SDX when interfacing via I2C. 0b1 enables the default increased drive strength, 0b0 does not increase the drive strength.

Register (0x52) io_i2c_if

Description: Configuration register for the I2C interface

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	R	R	R	R	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						watchdo...	watchdo...

- reserved:write 0x0.
- `IO_I2C_IF.watchdog_timer_sel`: (bit offset: 0, bit width: 1) Select the timer period for the I2C watchdog Following values can be set to or read from the field watchdog_timer_sel:

Value	Description
0b0 (0x0)	I2C watchdog timeout after 1.25 ms
0b1 (0x1)	I2C watchdog timeout after 40 ms

- `IO_I2C_IF.watchdog_timer_en`: (bit offset: 1, bit width: 1) Enable or disable the watchdog for the I2C interface. Disable this feature when using I3C or SPI

Register (0x53) io_odr_deviation

Description: ODR Deviation Trim Register (OTP backed) - User mirror register

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	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			odr_deviation				

- reserved:write 0x0.
- `IO_ODR_DEVIATION.odr_deviation`: (bit offset: 0, bit width: 5) ODR clock deviation

Register (0x60) acc_dp_off_x

Description: Data path register for the accelerometer offset of axis x

Bit	15	14	13	12	11	10	9	8
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		acc_dp_off_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	acc_dp_off_x							

- reserved:write 0x0.
- `ACC_DP_OFF_X.acc_dp_off_x`: (bit offset: 0, bit width: 14) Data path register for the temperature independent accelerometer offset of axis x: 1 LSB = 30.52ug; 0x2000 -> invalid

Register (0x61) acc_dp_dgain_x

Description: Data path register for the accelerometer re-scale of axis x

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_dp_dgain_x							

- reserved:write 0x0.
- `ACC_DP_DGAIN_X.acc_dp_dgain_x`: (bit offset: 0, bit width: 8) Data path register for the temperature independent accelerometer re-scale of axis x: covers ±3.125% of sensitivity

Register (0x62) acc_dp_off_y

Description: Data path register for the accelerometer offset of axis y

Bit	15	14	13	12	11	10	9	8
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		acc_dp_off_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	acc_dp_off_y							

- reserved:write 0x0.
- `ACC_DP_OFF_Y.acc_dp_off_y`: (bit offset: 0, bit width: 14) Data path register for the temperature independent accelerometer offset of axis y: 1 LSB = 30.52ug; 0x2000 -> invalid

Register (0x63) acc_dp_dgain_y

Description: Data path register for the accelerometer re-scale of axis y

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_dp_dgain_y							

- reserved:write 0x0.
- `ACC_DP_DGAIN_Y.acc_dp_dgain_y`: (bit offset: 0, bit width: 8) Data path register for the temperature independent accelerometer re-scale of axis y: covers ±3.125% of sensitivity

Register (0x64) acc_dp_off_z

Description: Data path register for the accelerometer offset of axis z

Bit	15	14	13	12	11	10	9	8
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		acc_dp_off_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	acc_dp_off_z							

- reserved:write 0x0.
- `ACC_DP_OFF_Z.acc_dp_off_z`: (bit offset: 0, bit width: 14) Data path register for the temperature independent accelerometer offset of axis z: 1 LSB = 30.52ug; 0x2000 -> invalid

Register (0x65) acc_dp_dgain_z

Description: Data path register for the accelerometer re-scale of axis z

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_dp_dgain_z							

- reserved:write 0x0.
- `ACC_DP_DGAIN_Z.acc_dp_dgain_z`: (bit offset: 0, bit width: 8) Data path register for the temperature independent accelerometer re-scale of axis z: covers ±3.125% of sensitivity

Register (0x66) gyr_dp_off_x

Description: Data path register for the gyroscope offset of axis x

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

- reserved:write 0x0.
- `GYR_DP_OFF_X.gyr_dp_off_x`: (bit offset: 0, bit width: 10) Data path register for the temperature independent gyroscope offset of axis x: 1 LSB = 0.061°/s; 0x200 -> invalid

Register (0x67) gyr_dp_dgain_x

Description: Data path register for the gyroscope re-scale of axis x

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	gyr_dp_dgain_x						

- reserved:write 0x0.
- `GYR_DP_DGAIN_X.gyr_dp_dgain_x`: (bit offset: 0, bit width: 7) Data path register for the temperature independent gyroscope re-scale of axis x: covers ±12.5% of sensitivity

Register (0x68) gyr_dp_off_y

Description: Data path register for the gyroscope offset of axis y

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

- reserved:write 0x0.
- `GYR_DP_OFF_Y.gyr_dp_off_y`: (bit offset: 0, bit width: 10) Data path register for the temperature independent gyroscope offset of axis y: 1 LSB = 0.061°/s; 0x200 -> invalid

Register (0x69) gyr_dp_dgain_y

Description: Data path register for the gyroscope re-scale of axis y

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	gyr_dp_dgain_y						

- reserved:write 0x0.
- `GYR_DP_DGAIN_Y.gyr_dp_dgain_y`: (bit offset: 0, bit width: 7) Data path register for the temperature independent gyroscope re-scale of axis y: covers ±12.5% of sensitivity

Register (0x6A) gyr_dp_off_z

Description: Data path register for the gyroscope offset of axis z

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

- reserved:write 0x0.
- `GYR_DP_OFF_Z.gyr_dp_off_z`: (bit offset: 0, bit width: 10) Data path register for the temperature independent gyroscope offset of axis z: 1 LSB = 0.061°/s; 0x200 -> invalid

Register (0x6B) gyr_dp_dgain_z

Description: Data path register for the gyroscope re-scale of axis z

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	gyr_dp_dgain_z						

- reserved:write 0x0.
- `GYR_DP_DGAIN_Z.gyr_dp_dgain_z`: (bit offset: 0, bit width: 7) Data path register for the temperature independent gyroscope re-scale of axis z: covers ±12.5% of sensitivity

Register (0x70) i3c_tc_sync_tph

Description: I3C Timing Control Sync TPH Register

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	i3c_tc_sync_tph							

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	i3c_tc_sync_tph							

- `I3C_TC_SYNC_TPH.i3c_tc_sync_tph`: (bit offset: 0, bit width: 16) I3C Timing Control Sync TPH Register

Register (0x71) i3c_tc_sync_tu

Description: I3C Timing Control Sync TU Register

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	i3c_tc_sync_tu							

- reserved:write 0x0.
- `I3C_TC_SYNC_TU.i3c_tc_sync_tu`: (bit offset: 0, bit width: 8) I3C Timing Control Sync TU Register

Register (0x72) i3c_tc_sync_odr

Description: I3C Timing Control Sync ODR Register

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	i3c_tc_sync_odr							

- reserved:write 0x0.
- `I3C_TC_SYNC_ODR.i3c_tc_sync_odr`: (bit offset: 0, bit width: 8) I3C Timing Control Sync ODR Register

Register (0x7E) cmd

Description: Command Register

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

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: 16) Available commands (Note: Register will always return 0x00 as read result) Following values can be set to or read from the field cmd:

Value	Description
0x100	Trigger the self-test of the device. Default scope of the self-test is a test of the accelerometer and gyroscope. Notes: an enabled feature engine is required; further settings are possible via the feature engine data interface.
0x101	Trigger the self-calibration of the gyroscope. Notes: an enabled feature engine is required; further settings are possible via the feature engine data interface.
0x200	Abort a running self-calibration of the gyroscope.
0x201	Update the configuration of the I3C timing control synchonronous feature written to all or any of I3C_TC_SYNC_TPH, I3C_TC_SYNC_TU and I3C_TC_SYNC_ODR.
0x300	Axis mapping gets updated
0xDEAF	Triggers a soft reset, that is all user configurations data registers are overwritten with their default state and the feature engine is reset.

Register (0x7F) cfg_res

Description: Reserved configuration

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

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			value_one				

- reserved:write 0x0.
- `CFG_RES.value_one`: (bit offset: 0, bit width: 5) value
- `CFG_RES.value_two`: (bit offset: 14, bit width: 2) value

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 to an address in the extended register map configured in FEATURE_DATA_ADDR by a data exchange transaction. A transaction consists of writing the address to FEATURE_DATA_ADDR and then continuously reading all data from or writing all data to FEATURE_DATA_TX, see Fig. 28. When reading and writing data via this interface, there must be no communication to any other register before the read or write transaction is complete.

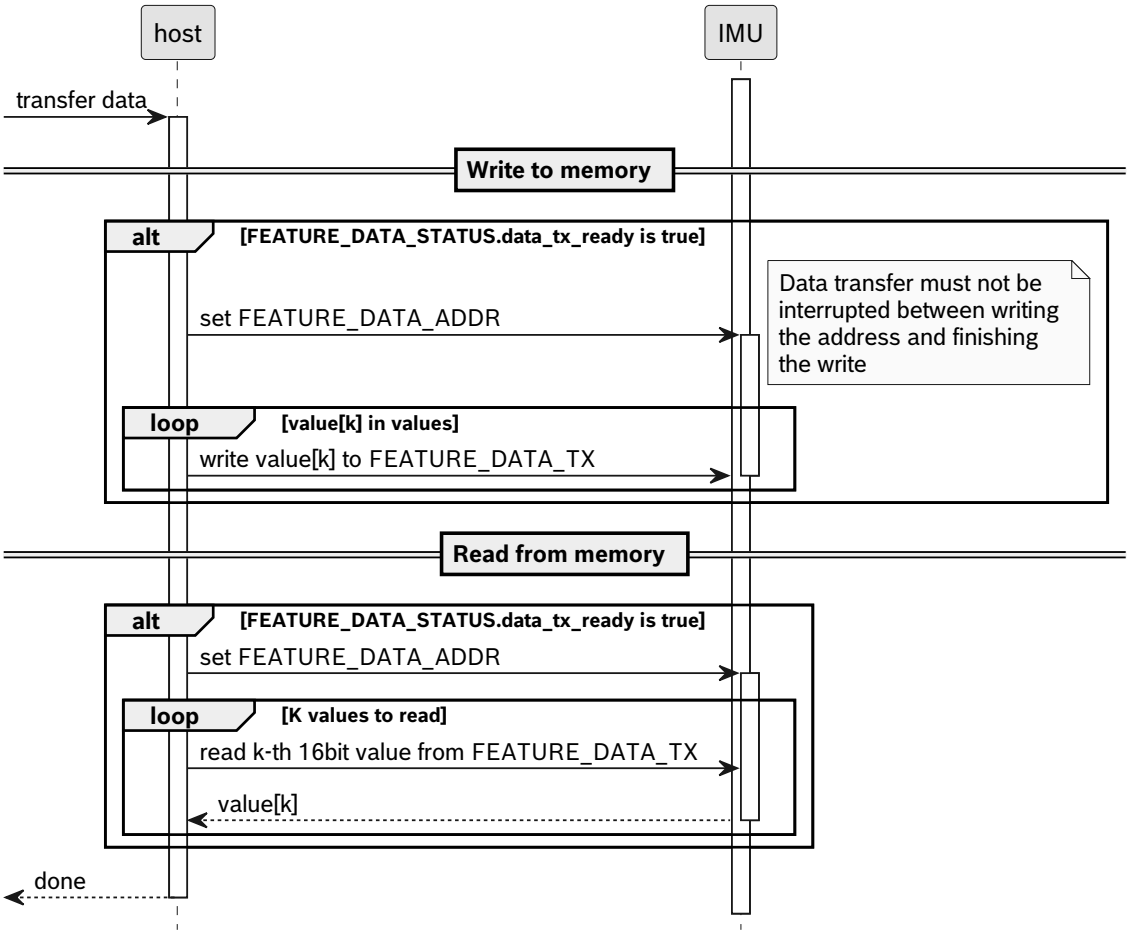


Figure 28: Data transfer to and from the extended register

Otherwise, the written or received data is invalid. The status of the data transfer is reported in the register FEATURE_DATA_STATUS:

- FEATURE_DATA_STATUS.data_tx_ready is set by the device to 0b1, if data can be read from or written to the device
- FEATURE_DATA_STATUS.data_outofbound_err is set by the device to 0b1, if data is tried to be read from or written to outside the specified register address range

6.2.1 Extended Register Map Overview

Table 37 provides an overview of the extended register map of the device.

Table 37: 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**
0x3E	SYNC_TIME	0x0000	time_lsw															
0x3D	SYNC_TEMP	0x0000	temp															
0x3C	SYNC_GYR_Z	0x0000	gyr_z															
0x3B	SYNC_GYR_Y	0x0000	gyr_y															
0x3A	SYNC_GYR_X	0x0000	gyr_x															
0x39	SYNC_ACC_Z	0x0000	acc_z															
0x38	SYNC_ACC_Y	0x0000	acc_y															
0x37	SYNC_ACC_X	0x0000	acc_x															
0x36	I3C_TC	0x0000	reserved															i3c_tc...
0x35	GYR_MOT_DET	0x0000	reserved								slope							
0x34	SC_ST_VALUE12	0x0000	reserved	ref_z							ref_y				ref_x			
0x33	SC_ST_VALUE11	0x0000	value															
0x32	SC_ST_VALUE10	0x0000	value															
0x31	SC_ST_VALUE9	0x0000	value															
0x30	SC_ST_VALUE8	0x0000	value															
0x2F	SC_ST_VALUE7	0x0000	value															
0x2E	SC_ST_VALUE6	0x0000	value															
0x2D	SC_ST_VALUE5	0x0000	value															
0x2C	SC_ST_VALUE4	0x0000	value															
0x2B	SC_ST_VALUE3	0x0000	value															
0x2A	SC_ST_VALUE2	0x0000	value															
0x29	SC_ST_VALUE1	0x0000	value															
0x28	SC_ST_VALUE0	0x0000	value															
0x27	GYR_SC_ST_CONF	0x0B76	reserved	gyr_sc_st_conf...			offs_filtercoeff				sens_filtercoeff				reserved			
0x26	GYR_SC_SELECT	0x0007	reserved													apply_...	offs_en	sens_en
0x25	ST_SELECT	0x0003	reserved														gyr_st...	acc_st...
0x24	ST_RESULT	0x0000	reserved									gyr_dr...	gyr_se...	gyr_se...	gyr_se...	acc_se...	acc_se...	acc_se...
0x23	ALT_CONFIG_CHG	0x0000	reserved								alt_conf_user_switch_src_select				alt_conf_alt_switch_src_select			
0x22	TILT_2	0xF069	beta_acc_mean															
0x21	TILT_1	0xD264	min_tilt_angle								segment_size							
0x20	TAP_3	0x6864	quite_time_after_gesture				min_quite_dur_between_taps				tap_shock_settling_dur				max_dur_between_peaks			

Table 37: 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**
0x1F	TAP_2	0x402D	max_gesture_dur								tap_peak_thres							
0x1E	TAP_1	0x0076	reserved								mode		max_peaks_for_tap			wait_f...		axis_sel
0x1D	ORIENT_2	0x20CD	hysteresis								slope_thres							
0x1C	ORIENT_1	0x2CFC	hold_time								theta				blocking		mode	ud_en
...	-	-	reserved															
0x10	SC_1	0x0000	reserved								reset_...				watermark_level			
0x0F	SIGMO_3	0x4653	mcr_max								peak_2_peak_max							
0x0E	SIGMO_2	0x4426	mcr_min								peak_2_peak_min							
0x0D	SIGMO_1	0x00FA	block_size															
0x0C	FLAT_2	0x09CD	hysteresis								slope_thres							
0x0B	FLAT_1	0x2088	hold_time								blocking		theta					
0x0A	NOMO_3	0x600A	wait_time				duration											
0x09	NOMO_2	0x0002	reserved								hysteresis							
0x08	NOMO_1	0x100A	reserved				acc_re...				slope_thres							
0x07	ANYMO_3	0x600A	wait_time				duration											
0x06	ANYMO_2	0x0002	reserved								hysteresis							
0x05	ANYMO_1	0x100A	reserved				acc_re...				slope_thres							
...	-	-	reserved															
0x03	AXIS_MAP_1	0x0000	reserved										invert_z	invert_y	invert_x	axis_map		
0x02	GEN_SET_1	0x0006	reserved										sw_lock	int_hold_dur				event_...
...	-	-	reserved															

6.2.2 Extended Register Map Details

Register (0x02) gen_set_1

Description: Common register settings

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	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	1	0
Content	reserved		sw_lock	int_hold_dur				event_...

- reserved:write 0x0.
- `GEN_SET_1.event_report_mode`: (bit offset: 0, bit width: 1) Configuration of the reporting mode Following values can be set to or read from the field event_report_mode:

Value	Description
0b0 (0x0)	All detected events are reported.
0b1 (0x1)	Report first detected event, then keep silent.

- `GEN_SET_1.int_hold_dur`: (bit offset: 1, bit width: 4) Interrupt hold time duration. Range 0 to 13. Hold time = 0.625ms * (2 ^interrupt_hold_duration) i.e 0.625ms to 5120ms. If set above 13, then hold time input is considered as 13. Default is 5ms. The hold time is only applicable for interrupts A..I in non-latched mode. Deviation up to +/- 1.25ms + ODR accuracy deviation. If tap detector is active the hold time MUST NOT be 5ms if not applied the enhanced configuration prior via 'bmi3x0_configure_enhanced_flexibility' (sensor driver API)
- `GEN_SET_1.sw_lock`: (bit offset: 5, bit width: 1) Reserved

Register (0x03) axis_map_1

Description: Describes axis map settings

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	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		invert_z	invert_y	invert_x	axis_map		

- reserved:write 0x0.
- `AXIS_MAP_1.axis_map`: (bit offset: 0, bit width: 3) Map accelerometer and gyroscope axis to desired configuration
  Following values can be set to or read from the field axis_map:

Value	Description
0b000 (0x0)	x=x; y=y; z=z;
0b001 (0x1)	x=y; y=x; z=z;
0b010 (0x2)	x=x; y=z; z=y;
0b011 (0x3)	x=z; y=x; z=y;
0b100 (0x4)	x=y; y=z; z=x;
0b101 (0x5)	x=z; y=y; z=x;

- `AXIS_MAP_1.invert_x`: (bit offset: 3, bit width: 1) Invert the x axis of accelerometer and gyroscope
  Following values can be set to or read from the field invert_x:

Value	Description
0b0 (0x0)	Clear this bit to not invert the x axis
0b1 (0x1)	Set this bit to invert the x axis

- `AXIS_MAP_1.invert_y`: (bit offset: 4, bit width: 1) Invert the y axis of accelerometer and gyroscope
  Following values can be set to or read from the field invert_y:

Value	Description
0b0 (0x0)	Clear this bit to not invert the y axis
0b1 (0x1)	Set this bit to invert the y axis

- `AXIS_MAP_1.invert_z`: (bit offset: 5, bit width: 1) Invert the z axis of accelerometer and gyroscope
  Following values can be set to or read from the field invert_z:

Value	Description
0b0 (0x0)	Clear this bit to not invert the z axis
0b1 (0x1)	Set this bit to invert the z axis

Register (0x05) anymo_1

Description: Configuration of acceleration slope threshold and reference update mode

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_re...	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	0	1	0
Content	slope_thres							

- reserved:write 0x0.
- `ANYMO_1.slope_thres`: (bit offset: 0, bit width: 12) Minimum slope of acceleration signal for motion detection The field slope_thres has the following properties:

Property	Value
Bitwidth	12
Sign	unsigned
Unit	g
Scaling	512
Default value	10/512
Range	Min=0.0, Max=7.998046875

- `ANYMO_1.acc_ref_up`: (bit offset: 12, bit width: 1) Mode of the acceleration reference update. Following values can be set to or read from the field acc_ref_up:

Value	Description
0b0 (0x0)	On detection of the event
0b1 (0x1)	On update of acceleration signal

The field acc_ref_up has the following properties:

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

Register (0x06) anymo_2

Description: Configuration for hysteresis of acceleration slope

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

- reserved:write 0x0.
- `ANYMO_2.hysteresis`: (bit offset: 0, bit width: 10) 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
Default value	2/512
Range	Min=0.0, Max=1.998046875

Register (0x07) anymo_3

Description: Configuration of timing related parameters for motion 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							

- `ANYMO_3.duration`: (bit offset: 0, bit width: 13) Minimum duration for which the slope shall be greater than threshold for motion detection The field duration has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50
Default value	0.2
Range	Min=0.0, Max=163.82

- `ANYMO_3.wait_time`: (bit offset: 13, bit width: 3) Wait time for clearing the event after slope is below threshold The field wait_time has the following properties:

Property	**Value**
Bitwidth	3
Sign	unsigned
Unit	s
Scaling	50
Default value	0.06
Range	Min=0.0, Max=0.14

Register (0x08) nomo_1

Description: Configuration of acceleration slope threshold and reference update mode

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_re...	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	0	1	0
Content	slope_thres							

- reserved:write 0x0.
- `NOMO_1.slope_thres`: (bit offset: 0, bit width: 12) Maximum slope of acceleration signal for stationary state detection
  The field slope_thres has the following properties:

Property	**Value**
Bitwidth	12
Sign	unsigned
Unit	g
Scaling	512
Default value	10/512
Range	Min=0.0, Max=7.998046875

- `NOMO_1.acc_ref_up`: (bit offset: 12, bit width: 1) Mode of the acceleration reference update.
  Following values can be set to or read from the field acc_ref_up:

Value	**Description**
0b0 (0x0)	On detection of the event
0b1 (0x1)	On update of acceleration signal

The field acc_ref_up has the following properties:

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

Register (0x09) nomo_2

Description: Configuration for hysteresis of acceleration slope

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

- reserved:write 0x0.
- `NOMO_2.hysteresis`: (bit offset: 0, bit width: 10) 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
Default value	2/512
Range	Min=0.0, Max=1.998046875

Register (0x0A) nomo_3

Description: Configuration of timing related parameters for stationary 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							

- `NOMO_3.duration`: (bit offset: 0, bit width: 13) Minimum duration for which the slope shall be less than threshold for stationary detection
  The field duration has the following properties:

Property	**Value**
Bitwidth	13
Sign	unsigned
Unit	s
Scaling	50
Default value	0.2
Range	Min=0.0, Max=163.82

- `NOMO_3.wait_time`: (bit offset: 13, bit width: 3) Wait time for clearing the event after slope is below threshold The field wait_time has the following properties:

Property	**Value**
Bitwidth	3
Sign	unsigned
Unit	s
Scaling	50
Default value	0.06
Range	Min=0.0, Max=0.14

Register (0x0B) flat_1

Description: Settings for maximum tilt angle for flat state, blocking mode and minimum duration in state for event 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	hold_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	1	0	0	0	1	0	0	0
Content	blocking		theta					

- `FLAT_1.theta`: (bit offset: 0, bit width: 6) 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	20
Range	Min=0, Max=44.8
Interpretation	(tan(x))^2

- `FLAT_1.blocking`: (bit offset: 6, bit width: 2) Blocking mode to prevent change of flat status 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	2
Range	Min=0, Max=3

- `FLAT_1.hold_time`: (bit offset: 8, bit width: 8) Minimum duration the device shall be in flat position for status to be asserted The field hold_time has the following properties:

Property	Value
Bitwidth	8
Sign	unsigned
Unit	s
Scaling	50.0
Default value	0.64
Range	Min=0.0, Max=5.10

Register (0x0C) flat_2

Description: Settings for consecutive sample slope and hysteresis

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	1
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							

- `FLAT_2.slope_thres`: (bit offset: 0, bit width: 8) Minimum slope between consecutive acceleration samples to pervent the change of flat status during large movement The field slope_thres has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	g
Scaling	512
Default value	0.400390625
Range	Min=0.0, Max=0.498046875

- `FLAT_2.hysteresis`: (bit offset: 8, bit width: 8) Angle of hysteresis for flat detection.
  The field hysteresis has the following properties:

Property	**Value**
Bitwidth	8
Sign	unsigned
Unit	degrees
Scaling	512
Default value	2.5
Range	Min=0, Max=44.8
Interpretation	((tand(z)^2) - (tand(z - x)^2))/((tand(z)^2) + 1)

Register (0x0D) sigmo_1

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							

- `SIGMO_1.block_size`: (bit offset: 0, bit width: 16) 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

Register (0x0E) sigmo_2

Description: Configuration of minimum value of peak to peak acceleration magnitude and mean crossing rate

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	1	0	0
Content	mcr_min						peak_2_peak_min	

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	1	0	0	1	1	0
Content	peak_2_peak_min							

- `SIGMO_2.peak_2_peak_min`: (bit offset: 0, bit width: 10) Minimum value of the peak to peak acceleration magnitude
  The field peak_2_peak_min has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512
Default value	38/512
Range	Min=0.0, Max=1.998046875

- `SIGMO_2.mcr_min`: (bit offset: 10, bit width: 6) Minimum number of mean crossing per second in acceleration magnitude
  The field mcr_min has the following properties:

Property	**Value**
Bitwidth	6
Default value	17
Range	Min=0, Max=63

Register (0x0F) sigmo_3

Description: Configuration of maximum value of peak to peak acceleration magnitude and mean crossing rate

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	1	1	0
Content	mcr_max						peak_2_peak_max	

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	1	0	0	1	1
Content	peak_2_peak_max							

- `SIGMO_3.peak_2_peak_max`: (bit offset: 0, bit width: 10) Maximum value of the peak to peak acceleration magnitude
  The field peak_2_peak_max has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512
Default value	595/512
Range	Min=0.0, Max=1.998046875

- `SIGMO_3.mcr_max`: (bit offset: 10, bit width: 6) Maximum number of mean crossing per second in acceleration magnitude
  The field mcr_max has the following properties:

Property	**Value**
Bitwidth	6
Default value	17
Range	Min=0, Max=63

Register (0x10) sc_1

Description: Configuration for step counter watermark and global reset

Bit	15	14	13	12	11	10	9	8
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					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.
- `SC_1.watermark_level`: (bit offset: 0, bit width: 10) An interrupt will be triggered every time the difference in number of steps counted from last event is equal to set value. 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

- `SC_1.reset_counter`: (bit offset: 10, bit width: 1) 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

Register (0x1C) orient_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

- `ORIENT_1.ud_en`: (bit offset: 0, bit width: 1) 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

- `ORIENT_1.mode`: (bit offset: 1, bit width: 2) 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

- `ORIENT_1.blocking`: (bit offset: 3, bit width: 2) 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

- `ORIENT_1.theta`: (bit offset: 5, bit width: 6) 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

- `ORIENT_1.hold_time`: (bit offset: 11, bit width: 5) 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

Register (0x1D) orient_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							

- `ORIENT_2.slope_thres`: (bit offset: 0, bit width: 8) 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

- `ORIENT_2.hysteresis`: (bit offset: 8, bit width: 8) 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

Register (0x1E) tap_1

Description: Selection of the tap detection axis, gesture reporting approach, detection mode

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	1	1	1	0	1	1	0
Content	mode		max_peaks_for_tap			wait_f...	axis_sel	

- reserved:write 0x0.
- `TAP_1.axis_sel`: (bit offset: 0, bit width: 2) Dominant sensing axis of accelerometer along which tap gesture is performed
  Following values can be set to or read from the field axis_sel:

Value	Description
0b00 (0x0)	Use x-axis for tap detection
0b01 (0x1)	Use y-axis for tap detection
0b10 (0x2)	Use z-axis for tap detection

The field axis_sel has the following properties:

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

- `TAP_1.wait_for_timeout`: (bit offset: 2, bit width: 1) Perform gesture confirmation with wait time set by max_gesture_dur
  Following values can be set to or read from the field wait_for_timeout:

Value	Description
0b0 (0x0)	Report the gesture when detected
0b1 (0x1)	Report the gesture after confirmation

The field wait_for_timeout has the following properties:

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

- `TAP_1.max_peaks_for_tap`: (bit offset: 3, bit width: 3) Maximum number of threshold crossing expected around a tap The field max_peaks_for_tap has the following properties:

Property	Value
Bitwidth	3
Default value	6
Range	Min=0, Max=7

- `TAP_1.mode`: (bit offset: 6, bit width: 2) Mode for detection of tap gesture. Default value = Normal. In stable position of device, to improve detection accuracy, sensitive mode can be used. Under noisy scenarios, the false detection can be suppressed with Robust mode
  Following values can be set to or read from the field mode:

Value	Description
0b00 (0x0)	Sensitive detection mode
0b01 (0x1)	Normal detection mode
0b10 (0x2)	Robust detection mode

The field mode has the following properties:

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

Register (0x1F) tap_2

Description: Tap detector setting

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	max_gesture_dur						tap_peak_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	1	0	1	1	0	1
Content	tap_peak_thres							

- `TAP_2.tap_peak_thres`: (bit offset: 0, bit width: 10) Minimum threshold for peak resulting from the tap The field tap_peak_thres has the following properties:

Property	**Value**
Bitwidth	10
Sign	unsigned
Unit	g
Scaling	512
Default value	45/512
Range	Min=0.0, Max=1.998046875

- `TAP_2.max_gesture_dur`: (bit offset: 10, bit width: 6) Maximum duration from first tap within the second and/or third tap is expected to happen The field max_gesture_dur has the following properties:

Property	**Value**
Bitwidth	6
Sign	unsigned
Unit	s
Scaling	25
Default value	0.64
Range	Min=0.0, Max=2.52

Register (0x20) tap_3

Description: Tap detector setting

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	1	0	0	0
Content	quite_time_after_gesture				min_quite_dur_between_taps			

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	tap_shock_settling_dur				max_dur_between_peaks			

- `TAP_3.max_dur_between_peaks`: (bit offset: 0, bit width: 4) Maximum duration between positive and negative peaks to tap The field max_dur_between_peaks has the following properties:

Property	**Value**
Bitwidth	4
Sign	unsigned
Unit	s
Scaling	200
Default value	0.02
Range	Min=0.0, Max=0.075

- `TAP_3.tap_shock_settling_dur`: (bit offset: 4, bit width: 4) Maximum duration for which tap impact is observed The field tap_shock_settling_dur has the following properties:

Property	**Value**
Bitwidth	4
Sign	unsigned
Unit	s
Scaling	200
Default value	0.03
Range	Min=0.0, Max=0.075

- `TAP_3.min_quite_dur_between_taps`: (bit offset: 8, bit width: 4) Mimimum duration between two consecutive tap impact The field min_quite_dur_between_taps has the following properties:

Property	Value
Bitwidth	4
Sign	unsigned
Unit	s
Scaling	200
Default value	0.04
Range	Min=0.0, Max=0.075

- `TAP_3.quite_time_after_gesture`: (bit offset: 12, bit width: 4) Minimum quite duration between two gestures
  The field quite_time_after_gesture has the following properties:

Property	Value
Bitwidth	4
Sign	unsigned
Unit	s
Scaling	25
Default value	0.24
Range	Min=0.0, Max=0.6

Register (0x21) 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) 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) 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)

Register (0x22) 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) 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))

Register (0x23) alt_config_chg

Description: Conditions to switch to alternate or user config of the sensors

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	alt_conf_user_switch_src_select				alt_conf_alt_switch_src_select			

- reserved:write 0x0.
- `ALT_CONFIG_CHG.alt_conf_alt_switch_src_select`: (bit offset: 0, bit width: 4) Selection of int (none = 0, A..I) for switch to alternate configuration mode for both accel and gyro. Invalid values will be ignored and the last valid configuration will remain. Note: Based on the selection in 'alt_conf_alt_switch_src_select' and 'alt_conf_user_switch_src_select' the state transition may happen immediately into the selected mode according to the events detected by the feature engine.
  Following values can be set to or read from the field alt_conf_alt_switch_src_select:

Value	Description
0b0000 (0x0)	None selected
0b0001 (0x1)	Selection of int A for switch of configuration
0b0010 (0x2)	Selection of int B for switch of configuration
0b0011 (0x3)	Selection of int C for switch of configuration
0b0100 (0x4)	Selection of int D for switch of configuration
0b0101 (0x5)	Selection of int E for switch of configuration
0b0110 (0x6)	Selection of int F for switch of configuration
0b0111 (0x7)	Selection of int G for switch of configuration
0b1000 (0x8)	Selection of int H for switch of configuration
0b1001 (0x9)	Selection of int I for switch of configuration

- `ALT_CONFIG_CHG.alt_conf_user_switch_src_select`: (bit offset: 4, bit width: 4) Selection of interrupt (none = 0, A..I) for switch to user configuration mode for both accel and gyro. Invalid values will be ignored and the last valid configuration will remain. Note: Based on the selection in 'alt_conf_alt_switch_src_select' and 'alt_conf_user_switch_src_select' the state transition may happen immediately into the selected mode according to the events detected by the feature engine.
  Following values can be set to or read from the field alt_conf_user_switch_src_select:

Value	Description
0b0000 (0x0)	None selected
0b0001 (0x1)	Selection of int A for switch of configuration
0b0010 (0x2)	Selection of int B for switch of configuration
0b0011 (0x3)	Selection of int C for switch of configuration
0b0100 (0x4)	Selection of int D for switch of configuration
0b0101 (0x5)	Selection of int E for switch of configuration
0b0110 (0x6)	Selection of int F for switch of configuration
0b0111 (0x7)	Selection of int G for switch of configuration
0b1000 (0x8)	Selection of int H for switch of configuration
0b1001 (0x9)	Selection of int I for switch of configuration

Register (0x24) st_result

Description: Self-test (accelerometer and gyroscope) results

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	gyr_dr...	gyr_se...	gyr_se...	gyr_se...	acc_se...	acc_se...	acc_se...

- reserved:write 0x0.
- `ST_RESULT.acc_sens_x_ok`: (bit offset: 0, bit width: 1) Self-test of accelerometer X-axis
- `ST_RESULT.acc_sens_y_ok`: (bit offset: 1, bit width: 1) Self-test of accelerometer Y-axis
- `ST_RESULT.acc_sens_z_ok`: (bit offset: 2, bit width: 1) Self-test of accelerometer Z-axis
- `ST_RESULT.gyr_sens_x_ok`: (bit offset: 3, bit width: 1) Self-test of gyroscope X-axis
- `ST_RESULT.gyr_sens_y_ok`: (bit offset: 4, bit width: 1) Self-test of gyroscope Y-axis
- `ST_RESULT.gyr_sens_z_ok`: (bit offset: 5, bit width: 1) Self-test of gyroscope Z-axis
- `ST_RESULT.gyr_drive_ok`: (bit offset: 6, bit width: 1) Self-test of gyroscope drive

Register (0x25) st_select

Description: Self-test (accelerometer and gyroscope) type selection

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	R	R	R	R	R/W	R/W
Reset Value	0	0	0	0	0	0	1	1
Content	reserved						gyr_st...	acc_st...

- reserved:write 0x0.
- `ST_SELECT.acc_st_en`: (bit offset: 0, bit width: 1) Enable self-test of accelerometer
- `ST_SELECT.gyr_st_en`: (bit offset: 1, bit width: 1) Enable self-test of gyroscope

Register (0x26) gyr_sc_select

Description: Self-calibration (gyroscope only) type selection

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	R	R	R	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	1	1
Content	reserved					apply_...	offs_en	sens_en

- reserved:write 0x0.
- `GYR_SC_SELECT.sens_en`: (bit offset: 0, bit width: 1) Enable gyroscope self-calibration of sensitivity
- `GYR_SC_SELECT.offs_en`: (bit offset: 1, bit width: 1) Enable gyroscope self-calibration of offset
- `GYR_SC_SELECT.apply_corr`: (bit offset: 2, bit width: 1) Apply correction of offset and/or sensitivity error calculated by gyroscope self-calibration feature

Register (0x27) gyr_sc_st_conf

Description: Self-calibration (gyroscope only) and self-test (gyroscope only) configuration and result register.

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

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

- reserved:write 0x0.
- `GYR_SC_ST_CONF.sens_filtercoeff`: (bit offset: 6, bit width: 3) Filter coefficient of low pass filter used during gyroscope sensitivity self-calibration and self-test. Range = 3 to 6. Number of gyroscope samples (@1600Hz) averaged = 2*(2^(sens_filtercoeff+2)).
- `GYR_SC_ST_CONF.offs_filtercoeff`: (bit offset: 9, bit width: 4) Filter coefficient of low pass filter used during gyroscope offset self-calibration. Range = 3 to 11. Number of gyroscope samples (@1600Hz) averaged = 2^(offs_filtercoeff+2).
- `GYR_SC_ST_CONF.gyr_sc_st_conf_res`: (bit offset: 13, bit width: 2) res

Register (0x28) sc_st_value0

Description: res

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	value							

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	value							

- `SC_ST_VALUE0.value`: (bit offset: 0, bit width: 16) Value

Register (0x29) sc_st_value1

Description: res

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	value							

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	value							

- `SC_ST_VALUE1.value`: (bit offset: 0, bit width: 16) Value

Register (0x2A) sc_st_value2

Description: res

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	value							

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	value							

- `SC_ST_VALUE2.value`: (bit offset: 0, bit width: 16) Value

Register (0x2B) sc_st_value3

Description: res

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	value							

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	value							

- `SC_ST_VALUE3.value`: (bit offset: 0, bit width: 16) Value

Register (0x2C) sc_st_value4

Description: res

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	value							

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	value							

- `SC_ST_VALUE4.value`: (bit offset: 0, bit width: 16) Value

Register (0x2D) sc_st_value5

Description: res

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	value							

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	value							

- `SC_ST_VALUE5.value`: (bit offset: 0, bit width: 16) Value

Register (0x2E) sc_st_value6

Description: res

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	value							

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	value							

- `SC_ST_VALUE6.value`: (bit offset: 0, bit width: 16) Value

Register (0x2F) sc_st_value7

Description: res

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	value							

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	value							

- `SC_ST_VALUE7.value`: (bit offset: 0, bit width: 16) Value

Register (0x30) sc_st_value8

Description: res

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	value							

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	value							

- `SC_ST_VALUE8.value`: (bit offset: 0, bit width: 16) Value

Register (0x31) sc_st_value9

Description: res

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	value							

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	value							

- `SC_ST_VALUE9.value`: (bit offset: 0, bit width: 16) Value

Register (0x32) sc_st_value10

Description: res

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	value							

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	value							

- `SC_ST_VALUE10.value`: (bit offset: 0, bit width: 16) Value

Register (0x33) sc_st_value11

Description: res

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	value							

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	value							

- `SC_ST_VALUE11.value`: (bit offset: 0, bit width: 16) Value

Register (0x34) sc_st_value12

Description: res

Bit	15	14	13	12	11	10	9	8
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	ref_z					ref_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_y			ref_x				

- reserved:write 0x0.
- `SC_ST_VALUE12.ref_x`: (bit offset: 0, bit width: 5) Reference value for X-axis
- `SC_ST_VALUE12.ref_y`: (bit offset: 5, bit width: 5) Reference value for Y-axis
- `SC_ST_VALUE12.ref_z`: (bit offset: 10, bit width: 5) Reference value for Z-axis

Register (0x35) gyr_mot_det

Description: Motion detection threshold common for both gyroscope self-calibration and gyroscope self-test

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	R	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			slope				

- reserved:write 0x0.
- `GYR_MOT_DET.slope`: (bit offset: 0, bit width: 5) Maximum acceptable acceleration change between two accelerometer samples. Range = 0 to 31. Value = 10mg + slope*5mg.

Register (0x36) i3c_tc

Description: I3C time control (TC) settings

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	R	R	R	R	R	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							i3c_tc...

- reserved:write 0x0.
- `I3C_TC.i3c_tc_res`: (bit offset: 0, bit width: 1) reserved

Register (0x37) sync_acc_x

Description: Synchronized acceleration sample, x channel in the default axes configuration

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	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	acc_x							

- `SYNC_ACC_X.acc_x`: (bit offset: 0, bit width: 16) Synchronized acceleration sample, x channel in the default axes configuration

Register (0x38) sync_acc_y

Description: Synchronized acceleration sample, y channel in the default axes configuration

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	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	acc_y							

- `SYNC_ACC_Y.acc_y`: (bit offset: 0, bit width: 16) Synchronized acceleration sample, y channel in the default axes configuration

Register (0x39) sync_acc_z

Description: Synchronized acceleration sample, z channel in the default axes configuration

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	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	acc_z							

- `SYNC_ACC_Z.acc_z`: (bit offset: 0, bit width: 16) Synchronized acceleration sample, z channel in the default axes configuration

Register (0x3A) sync_gyr_x

Description: Synchronized angular rate sample, x channel in the default axes configuration

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	gyr_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	gyr_x							

- `SYNC_GYR_X.gyr_x`: (bit offset: 0, bit width: 16) Synchronized angular rate sample, x channel in the default axes configuration

Register (0x3B) sync_gyr_y

Description: Synchronized angular rate sample, y channel in the default axes configuration

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	gyr_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	gyr_y							

- `SYNC_GYR_Y.gyr_y`: (bit offset: 0, bit width: 16) Synchronized angular rate sample, y channel in the default axes configuration

Register (0x3C) sync_gyr_z

Description: Synchronized angular rate sample, z channel in the default axes configuration

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	gyr_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	gyr_z							

- `SYNC_GYR_Z.gyr_z`: (bit offset: 0, bit width: 16) Synchronized angular rate sample, z channel in the default axes configuration

Register (0x3D) sync_temp

Description: description

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	temp							

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	temp							

- `SYNC_TEMP.temp`: (bit offset: 0, bit width: 16) Synchronized temperature sample, only available after using bmi3x0_configure_enhanced_flexibility of sensor driver API

Register (0x3E) sync_time

Description: Synchronized time stamp

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	time_lsw							

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	time_lsw							

- `SYNC_TIME.time_lsw`: (bit offset: 0, bit width: 16) Synchronized time stamp, least significant word

7 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.

7.1 Electrical Specification

By default, the device operates in I²C mode. The interface of the device can be configured to operate in a SPI 4-wire configuration as well. It can also be re-configured by software to work in 3-wire mode instead of 4-wire mode. All three possible digital interfaces share partly the same pins. The mapping for the primary interface of device is given in Table 38.

Table 38: Pin mapping of the digital interface

Pin #	Name	I/O Type	Description	in SPI4W	in SPI3W	in I²C/I3C
1	SDO	Digital I/O	SDO Serial data output in SPI 4W; I²C Address bit-0 select in I²C mode	SDO	DNC	GND for default I²C address
4	INT1	Digital I/O	Interrupt pin 1	INT1	INT1	INT1
9	INT2	Digital I/O	Interrupt pin 2	INT2	INT2	INT2
12	CSB	Digital in	Chip select for SPI mode	CSB	CSB	VDDIO
13	SCx	Digital in	SCK for SPI serial clock ; SCL for I²C/ I3C serial clock	SCK	SCK	SCL
14	SDx	Digital I/O	SDA serial data I/O in I²C/I3C; SDI serial data input in SPI 4W; SDA serial data I/O in SPI 3W	SDI	SDA	SDA

In Table 39, the electrical specifications of the interface pins are given.

Table 39: 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 VDDIO	75	100	140	$k\Omega$
Input Capacitance	C_{in}				5	pF
I²C Bus Load Capacitance (max. drive capability)	$C_{load,I2C}$				400	pF
I3C Bus Load Capacitance (max. drive capability)	$C_{load,I3C}$			10	50	pF

7.2 Digital Interface Protocols

7.2.1 Protocol Selection

The protocol is automatically selected based on the behavior of the signal on the chip select pin CSB after power-up. After soft reset or power-up, the primary interface of the device is in I²C mode. If the CSB is connected to VDDIO during power-up and not changed, the primary interface works in I²C or I3C mode. The possible switches among the modes on the digital interface are summarized in Table 40.

For using I²C and I3C, it is recommended to hard-wire the CSB line to VDDIO. Since power-on-reset is only executed when both VDD and VDDIO are stable, there is no risk of an incorrect protocol detection due to the power-up sequence.

If a rising edge is detected on CSB after power-up, the device interface switches after 200 µs to SPI until a soft reset or until the next power-up occurs. Therefore, a rising edge on CSB is needed before starting the SPI communication. It is recommended to perform a single read from a register, e.g. from `CHIP_ID.chip_id`, before the actual data exchange with the device. Note: the content of the retrieved data will be invalid.

The switch from I²C to I3C follows the MIPI I3C^{SM} specification. Upon power up, the chip stays in I²C mode and once the dedicated Broadcast I3C Address (7'h7E) is seen on the bus, the chip will disable its I²C feature and the interface stays in the I3C mode until a soft reset or the next power-up occurs.

Table 40: Protocol Selection for the Digital interface

protocol switch	to I²C	to I3C	to SPI
from I^2C	n/a	Device ID 7E sent	dummy SPI read
from I3C	power-down or soft-reset	n/a	dummy SPI read
from SPI	power-down or soft-reset	power-down or soft-reset	n/a

7.2.2 Common specifications

The maximum ratings valid for all serial protocols supported by the device are given in Table 41. For SPI, the additional specifications are given in the corresponding sub-section 7.2.3.

Table 41: Serial interface timings

Parameter	Symbol	Condition	Min	Max	Unit
Clock Frequency	f_{I3C}	Max. load on SDI or SDO 30pF, $V_{DDIO} \geq 1.62V$		12.5	MHz
	f_{SPI}			10	MHz
	f_{I3C}, f_{SPI}	$V_{DDIO} < 1.62V$		8	MHz
Idle time after any access in any mode	$t_{IDLE,rd}$		2		μs

7.2.3 SPI Protocol

The dedicated timing specifications for SPI of the device in addition to Table 41 are stated in Table 42. Figure 29 shows the definition of the SPI timings.

Table 42: SPI timings

Parameter	Symbol	Condition	Min	Max	Units
SCK Low Pulse	t_{SCKL}	$V_{DDIO} \geq 1.62V$	45		ns
SCK High Pulse	t_{SCKH}	$V_{DDIO} \geq 1.62V$	45		ns
SCK Low Pulse	t_{SCKL}	$V_{DDIO} < 1.62V$	66		ns
SCK High Pulse	t_{SCKH}	$V_{DDIO} < 1.62V$	66		ns
SDI Setup Time	$t_{SDI,setup}$		20		ns
SDI Hold Time	$t_{SDI,hold}$		20		ns
SDO Output Delay	$t_{SDO,OD}$	Load = 30pF, $V_{DDIO} \geq 1.62V$		30	ns
CSB Setup Time	$t_{CSB,setup}$		40		ns
CSB Hold Time	$t_{CSB,hold}$		40		ns

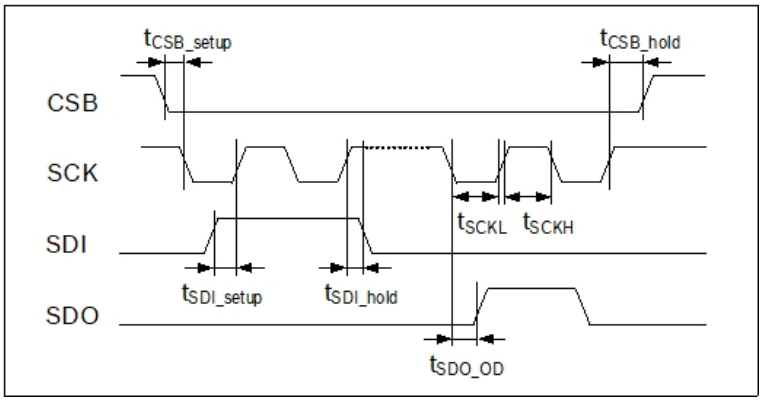


Figure 29: SPI timing diagram

The SPI interface of the device is compatible with two modes:

'00' [CPOL = '0' and CPHA = '0']
'11' [CPOL = '1' and CPHA = '1'].

The automatic selection between '00' and '11' is controlled based on the value of the clock on the SCK pin after a falling edge is detected on the chip select pin CSB.

Two configurations of the SPI interface are supported by the device: 4-wire and 3-wire. The protocol used by both configurations is the same. The device operates in the 4-wire configuration by default. It can be switched to 3-wire configuration by writing 0b1 to `IO_SPI_IF.spi3_en`. In the 3-wire configuration, the pin SDX is used as the common data pin.

SPI 4-wire the pins CSB (chip select low active), SCX (as SCK for serial clock), SDX (as SDI for serial data input), and SDO (serial data output) are used. The communication starts when CSB is pulled low by the SPI master and stops when CSB is pulled high again. SCK is also controlled by SPI master. SDI and SDO are driven at the falling edge of SCK and should be captured at the rising edge of SCK.
Multiple and single write operations are possible by keeping CSB low and continuing the data transfer. Only the address of the first register has to be sent via SDX. Addresses are automatically incremented after each write access as long as CSB stays active low. The principle of multiple write is shown in Figure 30.

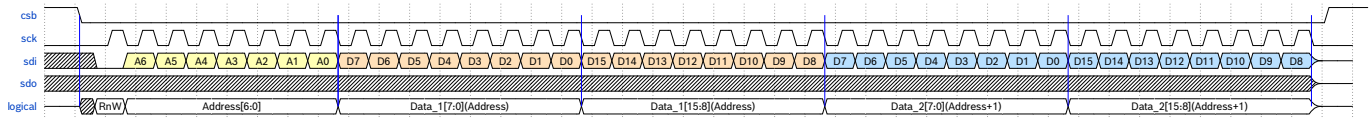


Figure 30: Multiple word write with the SPI protocol (4-wire)

The basic read operation waveform for 4-wire configuration is depicted in Figure 31. Please note, that the first byte received from the device via the SDO line corresponds to a dummy byte and the 2nd byte corresponds to the value read out of the specified register address. That means, for a basic read operation, at least two bytes have to be read and the first byte has to be dropped and all following bytes can be interpreted.

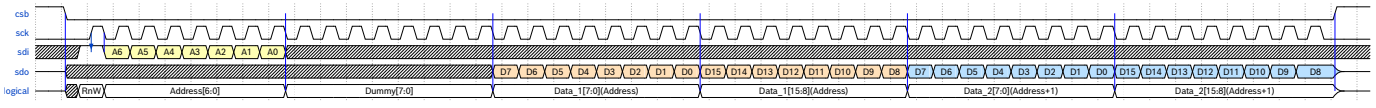


Figure 31: Multiple word read with the SPI protocol (4-wire)

- Address [A6-A0]: address of the first register
- Data *k*[7:0], Data *k*[15:8]: when in write mode, these are the data via the SDI, which will be written into the address.
- Data *k*[7:0], Data *k*[15:8]: when in read mode, these are the data on the SDO, which are read from the address.

Multiple read operations are possible by keeping CSB low and continuing the data transfer. Only the first register address has to be written. Addresses are automatically incremented after each read access as long as CSB stays active low. Please note that the first byte received from the device via the SDO line corresponds to a dummy byte and the 2nd byte corresponds to the value read out of the specified register address. The successive bytes read out correspond to values of incremented register addresses. That means, for a multiple read operation of n bytes, n+1 bytes have to be read, the first has to be dropped and the successive bytes must be interpreted.

SPI 3-wire the pins CSB (chip select low active), SCX (as SCK for serial clock), and SDX (as SDA for serial data input and output) are used. SCK is controlled by the SPI master. While SCK is pulled high, the communication starts when the CSB is pulled low by the SPI master and stops when CSB is pulled high again. SDX is driven (when used as input of the device) at the falling edge of SCK and should be captured, when used as the output of the device, at the rising edge of SCK.

In a 3-wire configuration, the protocol as such is the same as in a 4-wire configuration. The basic operation for read and write access for 3-wire configuration is depicted in Figures 32 and 33, respectively.

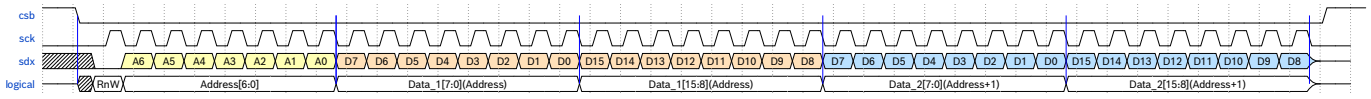


Figure 32: Multiple word write with the SPI protocol (3-wire)

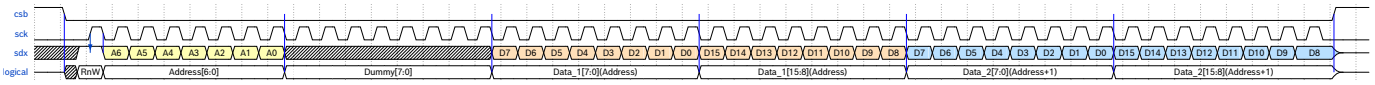


Figure 33: Multiple word read with the SPI protocol (3-wire)

7.2.4 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 asynchronous 0 mode
- Timing control synchronous mode

The I3C bus uses the pin SCX as SCL for serial clock and SDX as SDA for serial data input and output for the signal lines. Both lines are connected to VDDIO externally via pull-up resistors so that they are pulled high when the bus is free.

The I3C interface of the device is compatible with the I3C^{SM} Improved Inter Integrated Circuit Version 1.0 from 23 December 2016, and the Frequently Asked Questions (FAQ) for MIPI I3C^{SM} Version 1.0 from 08 December 2017 available at http://www.mipi.org/specifications/i3c-sensor-specification.

The protocol used on the serial interface for writing data to the device consists of

- write the address byte, and
- then write instantly a sequence of words consisting of two bytes each.

The protocol used on the serial interface for reading data from the device consists of

- write the address byte, and then
- read:
  - two dummy bytes preceding the actual payload, and then instantly
  - the number of desired payload bytes/words.

7.2.4.1 I3C Mode[SM]

The timing specification for I3C can be retrieved from the I3C specification refered to above.

Address Range The master can assign Dynamic Addresses from an allowed set of values to the device. In addition to the I3C Slave Address Restrictions defined in Table 9 in the section 5.1.2.2.5 of the I3C[SM] specification “Improved Inter Integrated Circuit Version 1.0 – 23 December 2016”, the address 7’h7D is reserved for this device as well.

Bus Configuration The I3C protocol uses several identifiers and codes to handle communication between several masters and slaves. For communication with this device, there are defined

- the I3C provisional ID,
- the Device Characteristics Register (DCR),
- the Bus Characteristics Register (BCR), and
- the Mandatory Byte (MDB) for IBIs.

The I3C provisional ID has the value defined in Table 43 where

- Bit 12 is controlled by the level on the SDO pin, that is GND for 0b0 and VDDIO for 0b1
- Bit 13 to 15 are the so-called instance ID and defined as “0b000”.

Table 43: I3C provisional identifier

Byte	**Byte 5**								**Byte 4**								**Byte 3**								**Byte 2**								**Byte 1**								**Byte 0**							
Bit of byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit of word	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
De-scrip-tion	MIPI member ID							IDT	Bosch group ID								Device ID								Instance ID				Reserved															
Bit value	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	SDO	0	0	0	0	0	0	0	0	0	0	0	0
Hex value	0				7				7				0				1				0				4				8				0 or 1				0				0				0			

The value of the Device Characteristics Register (DCR) is fixed to 0xEF. The value of the Bus Characteristics Register (BCR) is fixed to 0x06. This has the following meanings:

- Bit 5 of the BCR is set to 0b0, that is the device supports solely SDR.
- Bit 2 of the BCR is set to 0b1, that means that an accepted IBI is followed by the so-called mandatory byte. For details on the mandatory byte, please refere to the IBI description in the paragraph 7.2.4.1.

The timing requirement is defined in Table 74 “I3C Open Drain Timing Parameters” and Table 75 “I3C Push-Pull Timing Parameters for SDR and HDR-DDR Modes” in the I3C[SM] specification “Improved Inter Integrated Circuit Version 1.0 – 23 December 2016”.

Read and Write Access The SDR read and write access follows the I3C specification. In the following examples are given for write operations with Tables. 44 and 45 for single-word and multi-word writes, respectively, and for read operations with Tables. 46 and 47 for single-word and multi-word reads. For the diagrams/tables, the following color coding is applied:

- transfer from the host to the device is indicated by green coloured normal text
- transfer from the device to the host is indicated by orange coloured text in italics
- acknowledgement from the device to the host:
  - with hand-off is indicated by orange coloured text in italics
  - without hand-off is indicated by brown coloured text in italics
- drives from either host or device are indicated by magenta bold text.

In the diagrams, these abbreviations are used:

S Start
Sr Repeated start
P Stop
ACK Acknowledge
T Transition

For the examples of reading data from the device in Tables. 46 and 47, the stop (P) can be removed and replaced by a repeated start (Sr).

Table 44: I3C Single-Word Write Operation

Broadcast Address			Specified Slave DAA			Register N Address		Register N Data [7:0]		Register N Data [15:8]		
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'0'	*ACK*	Address N	T	Data N low word	T	Data N high word	T	P/Sr

Table 45: I3C Multi-Word Write Operation

Broadcast Address			Specified Slave DAA			Register M Address		Register M Data [7:0]		Register M Data [15:8]		...
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'0'	*ACK*	Address M	T	Data M low word	T	Data M high word	T	...

		...	Register N-1 Data [7:0]		Register N-1 Data [15:8]		Register N Data [7:0]		Register N Data [15:8]			
		...	Data N-1 low word	T	Data N-1 low word	T	Data N low word	T	Data N high word	T	P/Sr	

Table 46: I3C Single-Word Read Operation (Non-Repeated)

Broadcast Address			Specified Slave DAA			Register N Address		
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'0'	*ACK*	Address N	T	P

Broadcast Address			Specified Slave DAA			dummy byte		dummy byte		...
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'1'	*ACK*	*0x00*	**T**	*0x00*	**T**	...

...	Register N Data [7:0]		Register N Data [15:8]		
...	*Data N low word*	**T**	*Data N high word*	**T**	P/Sr

Table 47: I3C Multi-Word Read Operation (Non-Repeated)

Broadcast Address			Specified Slave DAA			Register M Address		
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'0'	*ACK*	Address M	T	P

Broadcast Address			Specified Slave DAA			dummy byte		dummy byte		...
S	7'h7E+'0'	*ACK*	Sr	7'hDAA+'1'	*ACK*	*0x00*	T	*0x00*	T	...

...	Register M Data [7:0]		Register M Data [15:8]		...
...	*Data M low word*	T	*Data M high word*	T	...
...	Register N-1 Data [7:0]		Register N-1 Data [15:8]		...
...	*Data N-1 low word*	T	*Data N-1 high word*	T	...

...	Register N Data [7:0]		Register N Data [15:8]		
...	*Data N low word*	T	*Data N high word*	T	P/Sr

In-Band Interrupt The device supports the IBI feature as defined in the section 5.1.6.2 of the I3C^{SM} specification "Improved Inter Integrated Circuit Version 1.0 – 23 December 2016".

Upon power up, the feature is disabled by default. The IBI feature can be enabled by the Common Command Code (CCC) `ENEC` with the `ENINT` bit set to 0b1, and can be disabled by the CCC `DISEC` with the `DISINT` bit set to 0b1. Please note, that prior to sending the CCC `RSTDAA`, sending of an IBI by the device has to be disabled by sending the direct CCC `DISEC` to the device.

If no START is forthcoming within the Bus Available Condition, then the chip will actively pull down the SDA line to issue a START. In case there is an IBI event, the device will emit its address into the arbitrated address header following a START (but not following a repeated START). The payload of the device specific IBI mandatory byte is defined in Table 48.

Table 48: I3C In-band Interrupt Mandatory Byte Payload

Bit	Purpose	Description
0	Specific Interrupt Identifier Field	FIFO Watermark IRQ or'ed with FIFO full IRQ
1		Sample ready IRQs (DRDY) of acceleration, angular rate and temperature or'ed
2		All feature IRQs or'ed
3 ... 4		0b00
5 ... 6	Interrupt Group Identifier Field	0b00
7		Defined by I3C

List of Common Command Codes Table 49 lists the Common Command Codes (CCC) supported by the device.

Table 49: Supported I3C Common Command Codes

CCC Code	CCC Type	MIPI Requires	CCC name	Description	Supported
0x00	Broadcast	Y	ENEC	enable events command	Y
0x01	Broadcast	Y	DISEC	disable events command	Y
0x02	Broadcast	Y	ENTAS0	enter activity state 0	Y
0x03	Broadcast	N	ENTAS1	enter activity state 1	N
0x04	Broadcast	N	ENTAS2	enter activity state 2	N
0x05	Broadcast	N	ENTAS3	enter activity state 3	N
0x06	Broadcast	Y	RSTDAA	reset dynamic address assignment	Y
0x07	Broadcast	Y	ENTDAA	enter dynamic address assignment	Y
0x08	Broadcast	N	DEFSLVS	define list of slaves	N
0x09	Broadcast	Y	SETMWL	set max write length	N[7]
0x0A	Broadcast	Y	SETMRL	set max read length	N[7]
0x0B	Broadcast	N	ENTTM	enter test mode	N
0x20	Broadcast	N	ENTHDR0	enter HDR mode 0	Y[8]
0x21	Broadcast	N	ENTHDR1	enter HDR mode 1	N
0x22	Broadcast	N	ENTHDR2	enter HDR mode 2	N
0x23	Broadcast	N	ENTHDR3	enter HDR mode 3, reserved by MIPI	N
0x24	Broadcast	N	ENTHDR4	enter HDR mode 4, reserved by MIPI	N
0x25	Broadcast	N	ENTHDR5	enter HDR mode 5, reserved by MIPI	N
0x26	Broadcast	N	ENTHDR6	enter HDR mode 6, reserved by MIPI	N
0x27	Broadcast	N	ENTHDR7	enter HDR mode 7, reserved by MIPI	N
0x28	Broadcast	N	SETXTIME	exchange timing information	Y
0x61 ... 0x7F	Broadcast	N		vendor extension - broadcast CCCs	N
0x80	Direct	Y	ENEC	enable events command	Y
0x81	Direct	Y	DISEC	disable events command	Y
0x82	Direct	Y	ENTAS0	enter activity state 0	Y
0x83	Direct	N	ENTAS1	enter activity state 1	N
0x84	Direct	N	ENTAS2	enter activity state 2	N
0x85	Direct	N	ENTAS3	enter activity state 3	N
0x86	Direct	Y	RSTDAA	reset dynamic address assignment	Y[9]
0x87	Direct	N	SETDASA	set dynamic address assignment from static address	Y
0x88	Direct	Y	SETNEWDA	set new dynamic address	Y
0x89	Direct	Y	SETMWL	set max write length	N[7]
0x8A	Direct	Y	SETMRL	set max read length	N[7]
0x8B	Direct	Y	GETMWL	get max write length	N[7]
0x8C	Direct	Y	GETMRL	get max read length	N[7]
0x8D	Direct	Y	GETPID	get provisional ID	Y
0x8E	Direct	Y	GETBCR	get bus characteristics register	Y
0x8F	Direct	Y	GETDCR	get device characteristics register	Y
0x90	Direct	Y	GETSTATUS	get device status	Y
0x91	Direct	N	GETACCMST	get accept mastership	N
0x93	Direct	N	SETBRGTGT	set bridge targets	N
0x94	Direct	N	GETMXDS	get max data speed	N
0x95	Direct	N	GETHDRCAP	get HDR capability	N
0x98	Direct	N	SETXTIME	exchange timing information	Y
0x99	Direct	N	GETXTIME	get timing information	Y
0xE0 ... 0xFE	Direct	N		vendor extension - direct CCCs	N

Master Clock Stalling The I3C specification defines master clock stalling. This allows the master to stall the clock for at most 100 μs. After this stall time, the slave may release the bus. The device releases the bus after 100 μs if a sensor is enabled in the registers `ACC_CONF` or `GYR_CONF`. In all other cases, the device does not automatically release the bus. In these cases, the watchdog intended for the I²C protocol may be used to detect a master hang situation. If the device drives the SDA line for more than 1.25 or 40 ms, the device resets its interface, if the watchdog is enabled in `IO_I2C_IF.watchdog_timer_en`. For details on the watchdog, see Section 7.2.4.2.

7.2.4.2 I²C Protocol in the I3C Compatibility Mode

The I²C interface of the device is compatible with the legacy support of the I²C of the standard "I3C^{SM} Improved Inter Integrated Circuit Version 1.0" from 23 December 2016, and the "Frequently Asked Questions (FAQ) for MIPI I3C^{SM} Version 1.0" from 08 December 2017 available at http://www.mipi.org/specifications/i3c-sensor-specification. The timing specification and diagrams for I²C can be retrieved from this specification.

The I3C timing requirements table and I3C Legacy Mode Timing diagram on page in Table 50 and Fig 34 on the next page are Copyright 2016 by MIPI Alliance, Inc. and reprinted with their permission[10].

Table 50: I²C timings

		Legacy Mode 400kHz / Fm		Legacy Mode 1MHz / Fm+		
Parameter	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Units**
SCL clock frequency	f_{SCL}	0	0.4	0	1.0	MHz
Setup time for a repeated START	$t_{SU,STA}$	600	–	260	–	ns
Hold time for a (repeated) START	$t_{HD,STA}$	600	–	260	–	ns
SCL clock low period	t_{LOW}	1300	–	500	–	ns
	$t_{DIG,L}$	t_{LOW} + t_{rCL}	–	t_{LOW} + t_{rCL}	–	ns
SCL clock high period	t_{HIGH}	600	–	260	–	ns
	$t_{DIG,H}$	t_{HIGH} + t_{rCL}	–	t_{HIGH} + t_{rCL}	–	ns
Data setup time	$t_{SU,DAT}$	100	–	50	–	ns
Data hold time	$t_{HD,DAT}$	–	–	–	–	ns
SCL signal rise time	t_{rCL}	20	300	–	120	ns
SCL signal fall time	t_{fCL}	$20 \cdot \left(\frac{V_{DDIO}}{5.5V}\right)$	300	$20 \cdot \left(\frac{V_{DDIO}}{5.5V}\right)$	120	ns
SDA signal rise time	t_{rDA}	20	300	–	120	ns
SDA signal fall time	t_{fDA}	$20 \cdot \left(\frac{V_{DDIO}}{5.5V}\right)$	300	$20 \cdot \left(\frac{V_{DDIO}}{5.5V}\right)$	120	ns
Setup time for STOP	$t_{SU,STO}$	600	–	260	–	ns
Bus free time between a STOP condition and a START condition	t_{BUF}	1.3	–	0.5	–	μs
Pulse width of spikes that the spike filter must suppress	t_{SPIKE}	0	50	0	50	μs

[7] These features are optional for the slave, see section Section 5.1.9.3.5 "Set/Get Max Write Length" in the MIPI I3C^{SM} specification "Improved Inter Integrated Circuit Version 1.0 - 23 December 2016".

[8] ENTHDR0 is recognized, but the device does not support HDR operations.

[9] Prior to sending the CCC RSTDAA, sending of an IBI by the device has to be disabled by sending the direct CCC DISEC to the device.

[10] This material may not be disclosed, reproduced or used for any purpose other than as needed to support the use of the products of Bosch Sensortec GmbH. MIPI provides all such material on an AS IS basis, without warranty of any kind.

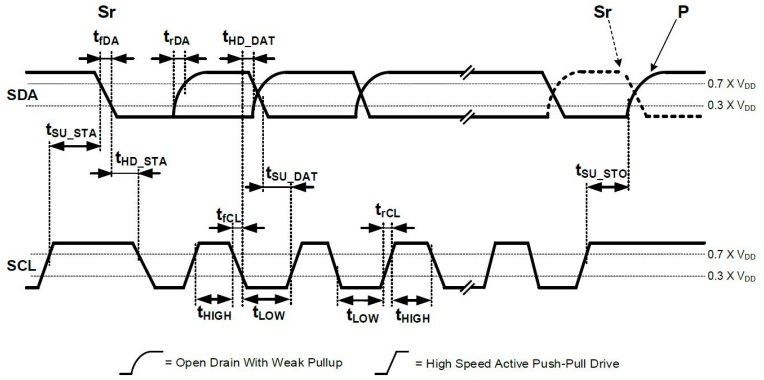


Figure 34: I²C timing diagram

Following the compatible mode defined in the I3C specification, the device supports fast mode (400 kHz fm) and fast mode plus (1 MHz Fm+). Only the 7-bit address mode is supported.

The default I²C legacy address of the device is 7h'0b1101000 (0x68). It is used if the SDO pin is pulled to 'GND'. The alternative address 7h'0b1101001 (0x69) is selected by pulling the SDO pin to 'VDDIO'.

The I²C legacy protocol works as follows:

START: Data transmission on the bus begins with a high to low transition on the SDA line while SCL is held high (start condition (S) indicated by I²C bus master). Once the START signal is transferred by the master, the bus is considered busy.

STOP: Each data transfer should be terminated by a Stop signal (P) generated by master. The STOP condition is a low to high transition on SDA line while SCL is held high.

ACKNOWLEDGE: Each byte of data transferred must be acknowledged. It is indicated by an acknowledge bit sent by the receiver. The transmitter must release the SDA line (no pull down) during the acknowledge pulse while the receiver must then pull the SDA line low so that it remains stable low during the high period of the acknowledge clock cycle.

A START immediately followed by a STOP (without SCL toggling from 'VDDIO' to 'GND') is not supported. If such a combination occurs, the STOP is not recognized by the device. In the following diagrams/tables, these abbreviations are used:

S Start
P Stop
ACK Acknowledge
RW Read / Write

In the diagrams/tables, a transfer from the host to the device is indicated by green coloured normal text while transfer from the device to the host is indicated by orange coloured text in italics.

I²C write access I²C write access can be used to write a data word in one sequence.

The sequence begins with start condition generated by the master, followed by 7 bits slave address and a write bit (RW = 0). The slave sends an acknowledge bit (ACK = 0) and releases the bus. Then the master sends the one byte register address. The slave again acknowledges the transmission and waits for the two times 8 bits of data which shall be written to the specified register address. After the slave has acknowledged the data bytes, the master generates a stop signal and terminates the writing protocol. An example of an I²C write access is stated in Table 51.

Table 51: I²C Single-Word Write Operation

Start	Slave Address	RW	ACK		Register M Address (0x38)	ACK	Register M Data [7:0]	ACK	Register M Data [15:8]	ACK	Stop
S	1 1 0 1 0 0 0	0	*0*	X	0 1 1 1 0 0 0	*0*	1 1 0 1 0 1 0 1	*0*	0 0 1 0 1 0 1 0	*0*	P

Multi-word writes are supported without restriction on normal registers with auto-increment as well as on special registers with address trap. An example of an I²C multi-word write access is stated in Table 52.

Table 52: I²C Multi-Word Write Operation

Start	Slave Address	RW	ACK		Register M Address (0x38)	ACK	Register M Data [7:0]	ACK	Register M Data [15:8]	ACK	⌐
S	1 1 0 1 0 0 0	0	*0*	X	0 1 1 1 0 0 0	*0*	1 1 0 1 0 1 0 1	*0*	0 0 1 0 1 0 1 0	*0*	

⌊	Register N Data [7:0]	ACK	Register N Data [15:8]	ACK	Stop
	1 0 1 0 1 0 1 0	*0*	0 1 0 1 0 1 0 1	*0*	P

I²C read access I²C read access can be used to read one or multiple data bytes or words in one sequence.

A read sequence consists of a one-byte I²C write phase followed by the I²C read phase to read a single byte, a single 16bit word of two bytes or multiple 16bit words. The two parts of the transmission must be separated by a repeated start condition (S). The I²C write phase addresses the slave and sends the register address to be read. After slave acknowledges the transmission, the master generates again a start condition and sends the slave address together with a read bit (RW = 1). Then, the master releases the bus and waits for the data bytes to be read out from slave. After each data byte the master has to generate an acknowledge bit (ACK = 0) to enable further data transfer. A non-ACK by the master (NACKM) (ACK = 1) from the master stops the data being transferred from the slave. The slave releases the bus so that the master can generate a STOP condition and terminate the transmission. The register address is automatically incremented and, therefore, more than one word can be sequentially read out. Once a new data read transmission starts, the start address will be set to the register address specified since the latest I²C write command. By default the start address is set at 0x00. In this way repetitive reads of words from the same starting address are possible.

An example of an I²C single-word read access with Repeated Start is provided in Table 53.

Table 53: I²C Single-Word Read Operation

Start	Slave Address	RW	ACK		Register N Address (0x38)	ACK
S	1 1 0 1 0 0 0	0	*0*	X	0 1 1 1 0 0 0	*0*

Start	Slave Address	RW	ACK	dummy byte	ACK	dummy byte	ACK	⌐
Sr	1 1 0 1 0 0 0	1	*0*	*0 0 0 0 0 0 0 0*	0	*0 0 0 0 0 0 0 0*	0	

⌊	Register N Data [7:0]	ACK	Register N Data [15:8]	ACK	Stop
	x x x x x x x x	0	*x x x x x x x x*	1	P

A further example of an I²C single byte read access with Repeated Start is provided in Table 54.

Table 54: I²C Single Byte Read Operation

Start	Slave Address	RW	ACK		Register N Address (0x38)	ACK
S	1 1 0 1 0 0 0	0	*0*	X	0 1 1 1 0 0 0	*0*

Start	Slave Address	RW	ACK	dummy byte	ACK	dummy byte	ACK	⌐
Sr	1 1 0 1 0 0 0	1	*0*	*0 0 0 0 0 0 0 0*	0	*0 0 0 0 0 0 0 0*	0	

⌊	Register N Data [7:0]	ACK	Stop
	x x x x x x x x	1	P

In order to prevent the I²C slave of the device to lock-up the I²C bus, a watchdog timer (WDT) is implemented. The WDT observes internal I²C signals and resets the I²C interface if the bus is locked-up by the BMI330. The

activity and the timer period of the WDT can be configured through the bits `IO_I2C_IF.watchdog_timer_en` and `IO_I2C_IF.watchdog_timer_sel`.
An example of an I²C read access for multiple words with Repeated Start is given in Table. 55.

Table 55: I²C Multi-Word Read Operation (with Repeated Start)

Start	Slave Address							RW	ACK		Register M Address (0x38)							ACK
S	1	1	0	1	0	0	0	0	*0*	X	0	1	1	1	0	0	0	*0*

Start	Slave Address							RW	ACK	dummy byte									ACK	dummy byte									ACK	⌐
Sr	1	1	0	1	0	0	0	1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	0	0	0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	0	

⌊	Register M Data [7:0]								ACK	Register M Data [15:8]								ACK	...
	x	*x*	*x*	*x*	*x*	*x*	*x*	*x*	0	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	0	...

...	Register N Data [7:0]								ACK	Register N Data [15:8]								ACK	Stop
...	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	0	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	0	P

7.3 Digital communication

Communication Access Restriction In order to allow for the correct internal synchronization of data written to or read from the device, certain access restrictions apply for consecutive write accesses or a write/read sequence through the I3C and I²C interface as well as the SPI. The required waiting period depends on the operation mode of the device: In high-performance and normal mode an interface idle time of at least 2 µs, for suspend mode an idle time of 450 µs is required.
As illustrated in Figure 35, an interface idle time of at least 2 µs is required following an operation when the device operates in any mode.

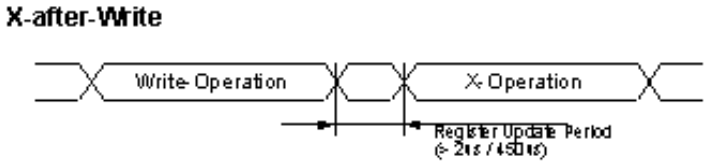


Figure 35: Post-Write Access Timing Constraints

8 Pin Out and Connection Diagrams

8.1 Pin Out

The Figures 36 and 37 shows the pin-out of the device from top and bottom view, respectively.

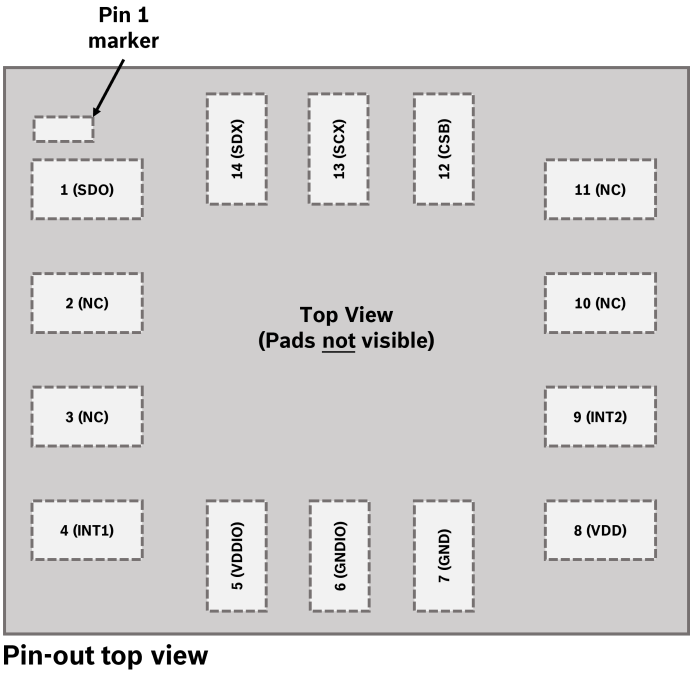


Figure 36: Pin-out: top view

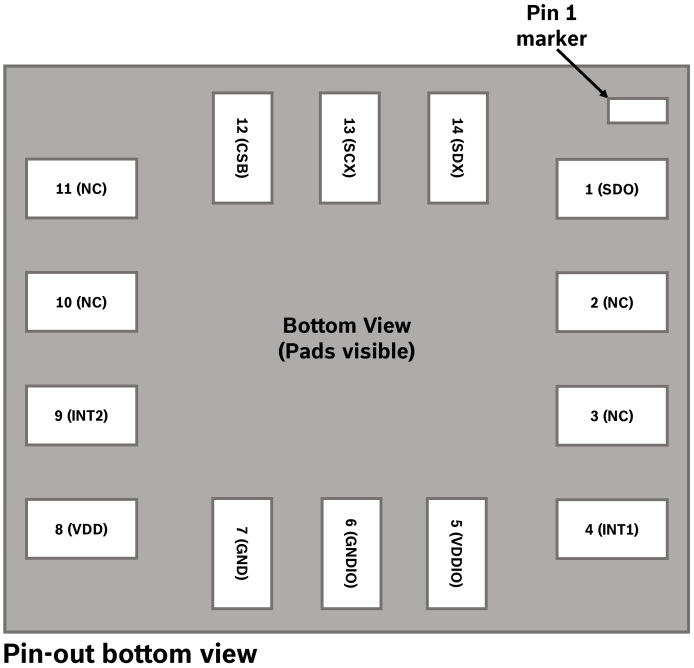


Figure 37: Pin-out: bottom view

The Table 56 details the pin out and the connections of the individual pins of the device.

Table 56: Pin-out and pin connections

Pin #	Name	I/O Type	Description	Connect to		
				in SPI 4w	in SPI 3w	in I²C/I3C
1	SDO	Digital I/O	SDO Serial data output in SPI 4W I²C Address bit-0 select in I²C mode	SDO	DNC	GND for default I²C address
2	NC	Digital I/O		do not connect		
3	NC	Digital I/O		do not connect		
4	INT1	Digital I/O	Interrupt pin 1	INT1	INT1	INT1
5	VDDIO	Supply	Digital I/O supply voltage 1.08V ... 3.63V	VDDIO	VDDIO	VDDIO
6	GNDIO	Ground	Ground for I/O	GNDIO	GNDIO	GNDIO
7	GND	Ground	Ground for digital & analog	GND	GND	GND
8	VDD	Supply	Power supply analog & digital domain 1.71V ... 3.63V	VDD	VDD	VDD
9	INT2	Digital I/O	Interrupt pin 2	INT2	INT2	INT2
10	NC			do not connect		
11	NC			do not connect		
12	CSB	Digital in	Chip select for SPI mode	CSB	CSB	VDDIO
13	SCx	Digital in	SCK for SPI serial clock SCL for I²C serial clock	SCK	SCK	SCL
14	SDx	Digital I/O	SDA serial data I/O in I²C/I3C SDI serial data input in SPI 4-wire SDA serial data I/O in SPI 3-wire	SDI	SDIO	SDA

8.2 Connection Diagrams

It is recommended to use 100nF capacitors for decoupling at pin 5 (VDDIO) and pin 8 (VDD).

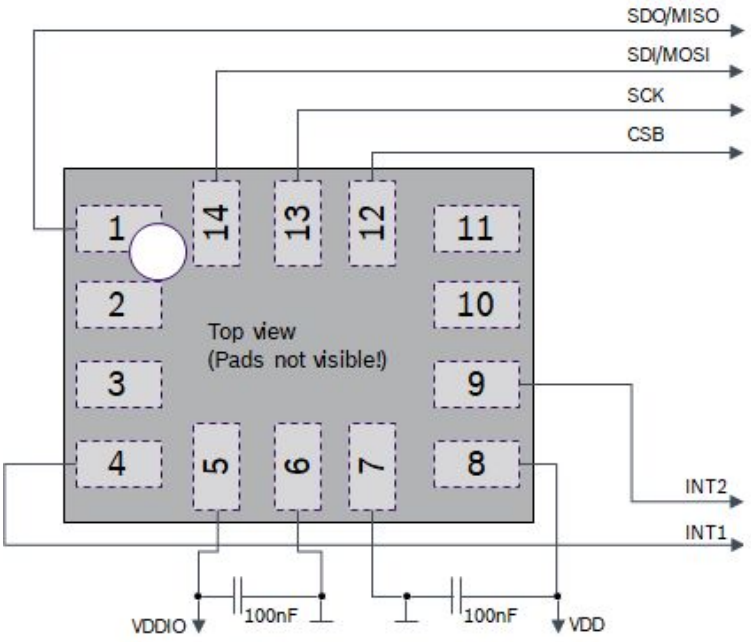


Figure 38: 4-wire SPI connection

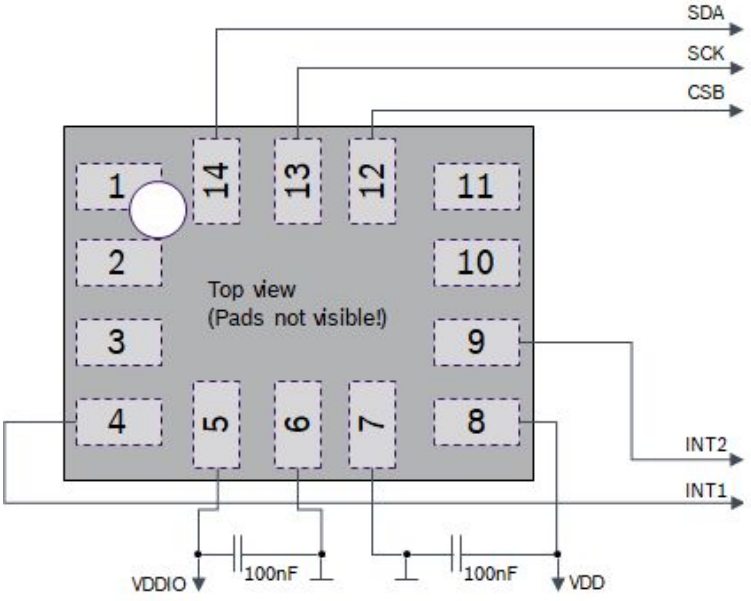


Figure 39: 3-wire SPI connection

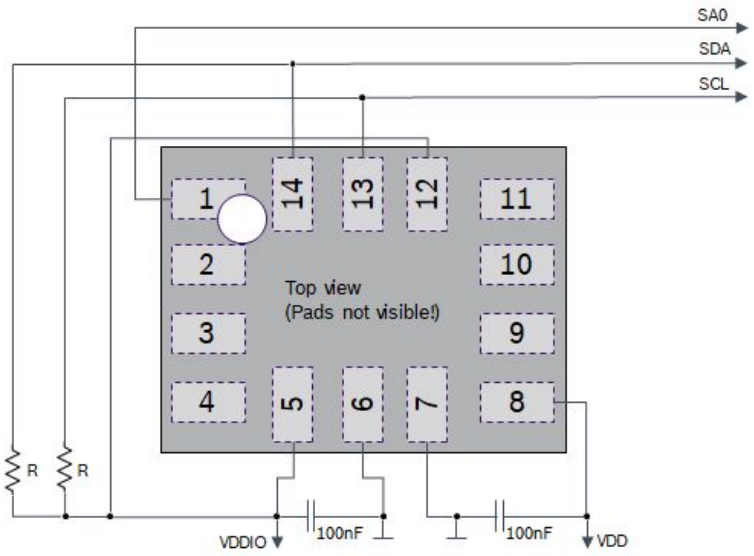


Figure 40: I3C connection

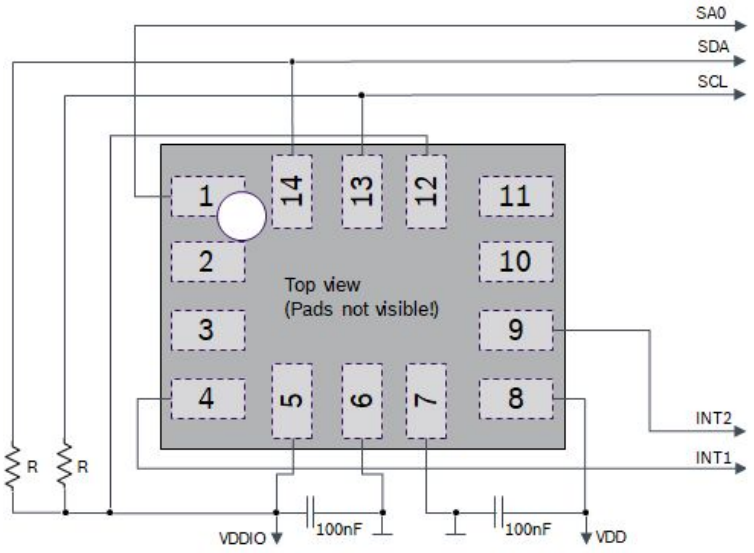


Figure 41: I²C connection

9 Package

9.1 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 and the corresponding gyroscope channel will be “zero” (static acceleration).

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

- $\pm 0g$ for the a_x accelerometer channel and $\pm 0^{\circ}/s$ for the Ω_x gyroscope channel
- $\pm 0g$ for the a_y accelerometer channel and $\pm 0^{\circ}/s$ for the Ω_y gyroscope channel
- $+1g$ for the a_z accelerometer channel and $\pm 0^{\circ}/s$ for the Ω_z gyroscope channel

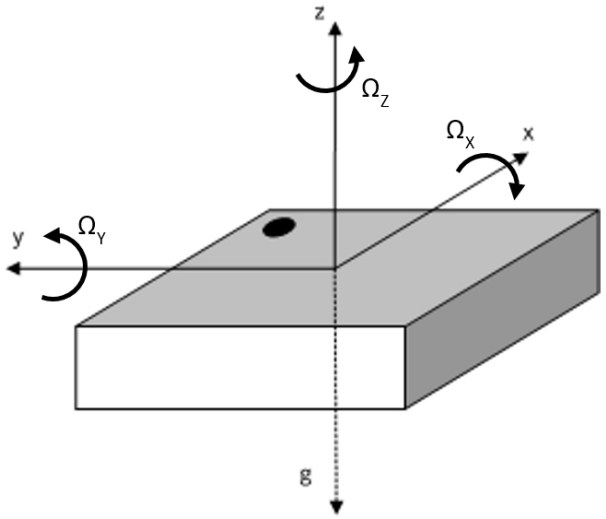
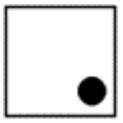
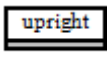
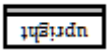


Figure 42: Definition of the sensing axes orientation for the raw device

For reference, Figure 43 below shows the smartphone device orientation with an integrated device.

Table 57: Output value corresponding to device orientation

Sensor Orientation (gravity vector ↓)						
Output Signal X	0 g / 0 LSB	+1 g / +4096 LSB	0 g / 0 LSB	-1 g / -4096 LSB	0 g / 0 LSB	0 g / 0 LSB
Output Signal Y	-1 g / -4096 LSB	0 g / 0 LSB	+1 g / +4096 LSB	0 g / 0 LSB	0 g / 0 LSB	0 g / 0 LSB
Output Signal Z	0 g / 0 LSB	0 g / 0 LSB	0 g / 0 LSB	0 g / 0 LSB	+1 g / +4096 LSB	-1 g / -4096 LSB

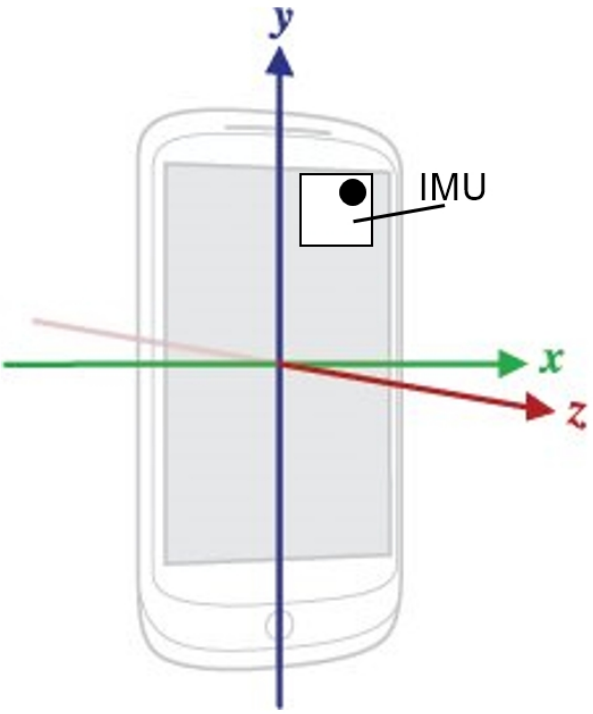


Figure 43: Definition of the sensing axes orientation within a device

Table 57 lists all corresponding output signals on X, Y, and Z while the sensor is at rest or at uniform motion in a gravity field under assumption of a ±8g range setting, a 16 bit resolution, and a top down gravity vector as shown above.

If the sensor axes coordinates do not match the axes coordinates of the platform, then a remapping of the axis is required. For the accelerometer and gyroscope data, the axes remapping may be done by the sensor internal axis remapping feature, see Section 5.11, or may be implemented in the driver of the application processor. To ensure expected function of the advanced features according to the desired coordinate system, the remapping of the axes as described in Section 5.11.

9.2 Dimensions

Figure 44 depicts the view from the top and the side of the package. The bottom view of the package is shown in Figure 45.

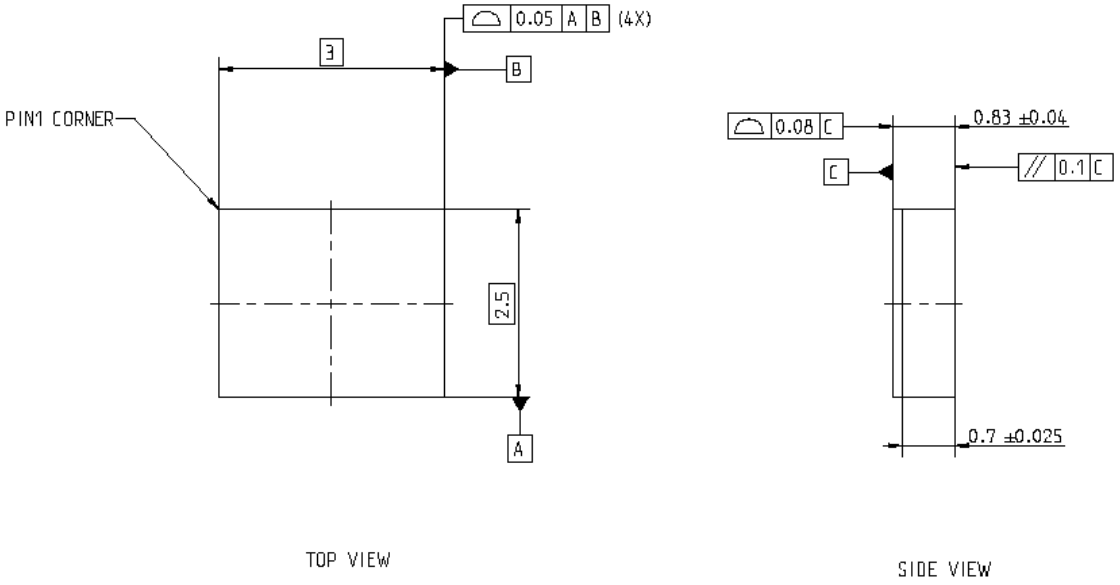


Figure 44: Dimensions from top and side view

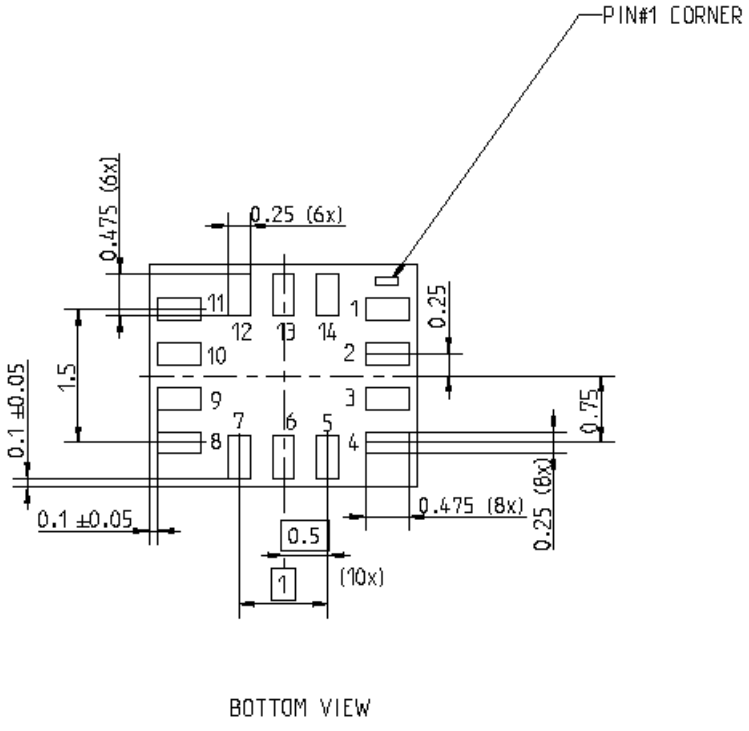


Figure 45: Dimensions from bottom view

Please note, that the Pin 1 marker must not be electrically connected.

9.3 Landing Pattern Recommendation

Figure 46 provides the recommendation for the landing pads to ensure maximum stability of the solder connections. The Pin 1 marker must not be electrically connected. Vias underneath the package have to be avoided.

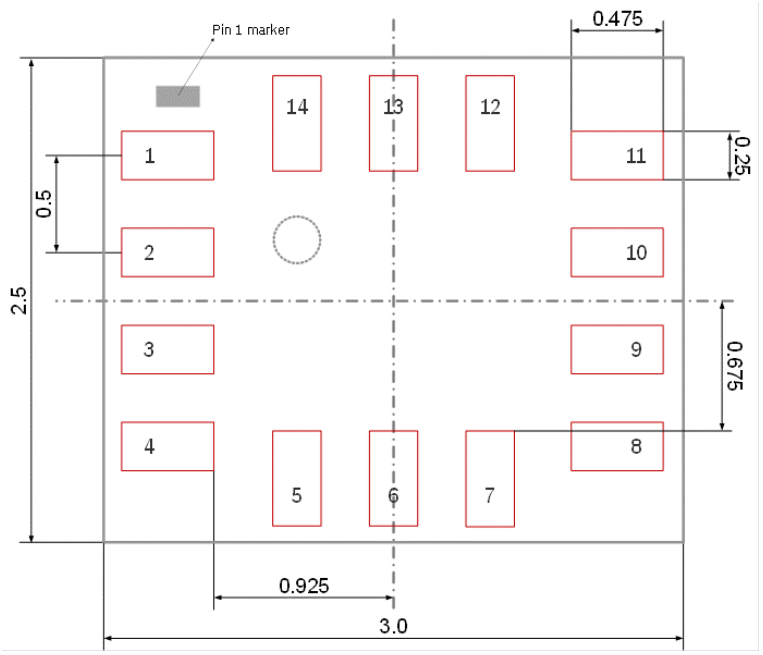


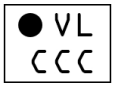
Figure 46: Landing pattern

9.4 Marking

Mass Production

Labeling	Symbol	Name	Remark
	V	Product Identifier	One alphanumeric digit, fixed to "C" to identify the product
	L	Internal Code	1 alphanumeric digit, fixed to “Y”, internal use only
	CCC	Counter ID	Tracing identification by three alphanumeric digits
	•	Pin 1	Identifier on top side

Engineering Samples

Labeling	Symbol	Name	Remark
	V	Product Identifier	One alphanumeric digit, fixed to "C" to identify the product
	L	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

9.5 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-033D “Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive 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.

9.6 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.

For more details on recommended handling, soldering and mounting please refer to the corresponding “Handling, soldering and mounting instructions” document or contact our regional offices, distributors and sales representatives.

9.7 Environmental Safety

The BMI330 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.

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

Internal package structure Within the scope of Bosch Sensortec's ambition to improve its products and secure the mass product supply, Bosch Sensortec qualifies additional sources (e.g. 2nd source) for the package of the device.

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

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

10 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 air or space applications (including but not limited to satellite technology) or supercomputers. If customers would like to use Bosch Sensortec products in supercomputers, customers are obligated to get written permissions from Bosch Sensortec.

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.

11 Document History and Modifications

Table 58: Change log

Rev No	Chapter	Description of modification/changes	Date
1.0	all	Initial release	June 5th, 2024
1.1	2	Added gyro TCO values for extended temperature range	May 6th, 2025
	3	Updated Table 9 ESD section	
		Document number updated to BST-BMI330-DS000-02	

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 9
72770 Reutlingen / Germany

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

Document number: BST-BMI330-DS000-02