BMI270

6-axis, smart, low-power Inertial Measurement Unit for high-performance applications

BMI270 – Data sheet

Document revision	1.0
Document release date	May 2019
Document number	BST-BMI270-DS000-02
Technical reference code	0 273 017 008
Notes	Data and descriptions in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product appearance

Basic Description

BMI270

The device 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. BMI270 is an Ultra-low power IMU optimized for wearable applications.

The device integrates:
- 16-bit digital, triaxial accelerometer with ±2g/±4g/±8g/±16g range
- 16-bit digital, triaxial gyroscope with ±125dps/±250dps/±500dps/±1000dps/±2000dps range

Key features

- Compact standard size LGA mold package, 14 pins, footprint 2.5x3.0mm² height 0.83mm
- Primary digital interface with 10 MHz slave SPI (4-wire, 3-wire) and up to 1MHz I²C (Fm+)
- Output data rates (ODR): 25 Hz ... 6.4 kHz (gyroscope) and 0.78 Hz ... 1.6 kHz (accelerometer)
- Programmable low-pass filter (accelerometer I gyroscope): bandwidth ca. 5 I 3 ... 680 I 890 Hz
- Wide power supply range: Analog VDD 1.71V ... 3.6V and independent VDDIO 1.2V…3.6V
- Ultra-low current consumption: typ. 685 µA (in full ODR and aliasing-free operation)
- Performance mode for gyroscope to minimize noise level: typ. < 8 mdps /√Hz.
- Built-in power management unit (PMU) for advanced power management and low power modes
- Rapid startup time: 2 ms for gyroscope (in fast start mode) and 2 ms for accelerometer

- Freely configurable secondary digital interface
  - 400 kHz I²C (Fm) master interface hub for 1 I2C AUX sensor (e.g. ext. magnetometer, pressure)
    - data synchronized to IMU
  - 10 MHz slave SPI (4-wire, 3-wire) for high speed, calibration free OIS / Dual OIS (SPI) applications
    - up to 6.4 kHz ODR, control register access and max. 676 µs group delay

- 2 KB on-chip FIFO buffer for accelerometer, gyroscope, timestamps, and AUX sensor data

- Fast offset error compensation for accelerometer and gyroscope
- Fast sensitivity error compensation for gyroscope (CRT, max. sensitivity error < 1%)

- HW synchronization of accelerometer, gyroscope, and AUX sensor (< 1 µs)
- Sensortime stamps for accurate system (host) and sensor (IMU) time synchronization (<40 µs)
- 2 independent programmable I/O pins for interrupt and synchronization events
- RoHS compliant, halogen and lead free

    - BMI270 Features
      - Significant motion/Any motion/Motion detect/No motion/Stationary detect/Wrist wear wakeup/Wrist worn step counter and detector/Activity change recognition/Push arm up/down/Wrist jiggle/Flick in /out

Typical Applications

- Wearables
- Hearables
- Smart clothing
- Augmented / virtual reality (AR/VR)

Target Devices

- Fitness trackers, wristbands, smart watches
- Earbuds, ankle bands, neck bands
- Smart clothes
- Augmented and virtual reality glasses and controllers

Table of contents

1. Specification 12

2. Absolute maximum ratings 18

3. Quick Start Guide 19

Note about using the device 19

4. Functional Description and Features 25

4.1. System Configurations 25

4.2. Block Diagram 26

4.3. Supply Voltage and Power Management 27

4.4. Power-On-Reset (POR) and Device Initialization 27

4.5. Power Modes 28

4.6. Sensor Data 30

Accelerometer Data 30

Accelerometer Filter Settings 30

Accelerometer Filter Modes 30

Accelerometer Data Processing in Normal and Performance Mode 30

Accelerometer Data Processing in Low Power Mode 31

Accelerometer Data Ready Interrupt 31

Gyroscope Data 31

Gyroscope Filter Settings 31

Gyroscope Filter Modes 31

Gyroscope Data Post-Processing 32

Gyroscope Data Processing in Normal and Performance Mode 32

Gyroscope Data Processing in Low Power Mode 33

Gyroscope Data Ready Interrupt 33

Temperature Sensor 34

Sensor Time 35

Configuration Changes 36

4.7. FIFO 37
  Frames 37
  Conditions and Details 43
  FIFO data synchronization 44
  FIFO synchronization with external interrupts 45
  FIFO Interrupts 45
  FIFO Reset 45
  FIFO in Low Power Mode 45

4.8. Advanced Features 46
  Global Configuration 46
  Anymotion Detection 49
  Nomotion Detection 51
  Significant Motion Detection 53
  Activity and Activity Change Recognition 54
  Wrist Wear Wakeup 55
  Wrist Wear Navigation Gesture Detector 57
  Step counter / detector (Wrist-worn) 60

4.9. General Interrupt Pin Configuration 61
  Electrical Interrupt Pin Behavior 61
  Interrupt Pin Mapping 61

4.10. Auxiliary Sensor Interface 62
  Structure and Concept 62
  Interface Control 62
  Interface Configuration 63
  Setup mode (AUX_IF_CONF.aux_manual_en =0b1) 64
  Data mode (AUX_IF_CONF.aux_manual_en=0) 66
  Delay (Time Offset) 66

4.11. OIS Interface 67
  OIS Register Map 68
  Register (0x0C..0x17) OIS_DATA_0..11 69
  Register (0x40) OIS_CRTL 69

4.12. Sensor Self-Test 70
Accelerometer 70
Gyroscope 71
4.13. Offset Compensation 72
Accelerometer 72
Gyroscope 72
4.14. Sensitivity Error Compensation 74
Accelerometer 74
Gyroscope 74
4.15. Non-Volatile Memory 75
4.16. Error Reporting 76
4.17. Soft-Reset 76

5. Register Description..... 77

5.1. General Remarks 77
5.2. Register Map 78
Register (0x00) CHIP_ID 83
Register (0x02) ERR_REG 83
Register (0x03) STATUS 84
Register (0x04) DATA_0 84
Register (0x05) DATA_1 84
Register (0x06) DATA_2 85
Register (0x07) DATA_3 85
Register (0x08) DATA_4 85
Register (0x09) DATA_5 85
Register (0x0A) DATA_6 86
Register (0x0B) DATA_7 86
Register (0x0C) DATA_8 86
Register (0x0D) DATA_9 86
Register (0x0E) DATA_10 87
Register (0x0F) DATA_11 87
Register (0x10) DATA_12 87

Register (0x11) DATA_13 87
Register (0x12) DATA_14 88
Register (0x13) DATA_15 88
Register (0x14) DATA_16 88
Register (0x15) DATA_17 88
Register (0x16) DATA_18 89
Register (0x17) DATA_19 89
Register (0x18) SENSORTIME_0 89
Register (0x19) SENSORTIME_1 89
Register (0x1A) SENSORTIME_2 90
Register (0x1B) EVENT 90
Register (0x1C) INT_STATUS_0 91
Register (0x1D) INT_STATUS_1 91
Register (0x1E) SC_OUT_0 91
Register (0x1F) SC_OUT_1 92
Register (0x20) WR_GEST_ACT 92
Register (0x21) INTERNAL_STATUS 93
Register (0x22) TEMPERATURE_0 94
Register (0x23) TEMPERATURE_1 94
Register (0x24) FIFO_LENGTH_0 95
Register (0x25) FIFO_LENGTH_1 95
Register (0x26) FIFO_DATA 95
Register (0x2F) FEAT_PAGE 96
Register (0x30) FEATURES[16] 96
Register (0x40) ACC_CONF 113
Register (0x41) ACC_RANGE 114
Register (0x42) GYR_CONF 115
Register (0x43) GYR_RANGE 116
Register (0x44) AUX_CONF 117
Register (0x45) FIFO_DOWNS 118
Register (0x46) FIFO_WTM_0 118
Register (0x47) FIFO_WTM_1 119

Register (0x48) FIFO_CONFIG_0 119

Register (0x49) FIFO_CONFIG_1 120

Register (0x4A) SATURATION 121

Register (0x4B) AUX_DEV_ID 121

Register (0x4C) AUX_IF_CONF 122

Register (0x4D) AUX_RD_ADDR 122

Register (0x4E) AUX_WR_ADDR 123

Register (0x4F) AUX_WR_DATA 123

Register (0x52) ERR_REG_MSK 123

Register (0x53) INT1_IO_CTRL 124

Register (0x54) INT2_IO_CTRL 125

Register (0x55) INT_LATCH 125

Register (0x56) INT1_MAP_FEAT 126

Register (0x57) INT2_MAP_FEAT 126

Register (0x58) INT_MAP_DATA 127

Register (0x59) INIT_CTRL 127

Register (0x5B) INIT_ADDR_0 127

Register (0x5C) INIT_ADDR_1 128

Register (0x5E) INIT_DATA 128

Register (0x5F) INTERNAL_ERROR 128

Register (0x68) AUX_IF_TRIM 129

Register (0x69) GYR_CRT_CONF 129

Register (0x6A) NVM_CONF 130

Register (0x6B) IF_CONF 130

Register (0x6C) DRV 130

Register (0x6D) ACC_SELF_TEST 131

Register (0x6E) GYR_SELF_TEST_AXES 131

Register (0x70) NV_CONF 132

Register (0x71) OFFSET_0 132

Register (0x72) OFFSET_1 133

Register (0x73) OFFSET_2 133

Register (0x74) OFFSET_3 133

Register (0x75) OFFSET_4 133

Register (0x76) OFFSET_5 134

Register (0x77) OFFSET_6 134

Register (0x7C) PWR_CONF 135

Register (0x7D) PWR_CTRL 136

Register (0x7E) CMD 136

6. Digital Interfaces 137

6.1. Interfaces 137

6.2. Primary Interface 138

6.3. Primary Interface Digital Protocol Selection 139

6.4. Primary Interface SPI 139

6.5. Primary Interface I²C 144

6.6. Secondary Interface 148

Auxiliary Interface 148

OIS Interface 149

7. Pinout and Connection Diagram 150

7.1. Pin-out 150

7.2. Connection Diagrams without Secondary Interface 153

Primary: 3wire SPI Secondary: None 153

Primary: 4wire SPI Secondary: None 153

7.3. Connection Diagrams with I2C Auxiliary Interface 155

Primary: 3wire SPI Secondary: Auxiliary interface I2C (e.g. BMM150 sensor) 155

Primary: 4wire SPI Secondary: Auxiliary interface I2C (e.g. BMM150 sensor) 155

Primary: I2C Secondary: Auxiliary interface I2C (e.g. BMM150 sensor) 156

7.4. Connection Diagrams with OIS Interface 157

Primary: 3wire SPI Secondary: 4 wire SPI for OIS interface 157

Primary: 3wire SPI Secondary: 3 wire SPI for OIS interface 157

Primary: 4wire SPI Secondary: 4 wire SPI for OIS interface 158

Primary: 4wire SPI Secondary: 3 wire SPI for OIS interface 158

Primary: I2C Secondary: 4 wire SPI for OIS interface 159

Primary: I2C Secondary: 3 wire SPI for OIS interface 159

8. Package............160

8.1. Package outline dimensions 160

8.2. Sensing axis orientation 161

8.3. Landing pattern recommendation 163

8.4. Marking 164

Mass production 164

Engineering samples 164

8.5. Soldering guidelines 165

8.6. Handling instructions 166

8.7. Environmental safety 167

Halogen content 167

Internal package structure 167

9. Legal disclaimer............168

9.1. Engineering samples 168

9.2. Product use 168

9.3. Application examples and hints 168

10. Document history and modification 169

Table 1: Basic electrical parameter specifications 12
Table 2: Accelerometer characteristics 13
Table 3: Gyroscope Characteristics 15
Table 4: Electrical characteristics temperature sensor 17
Table 5: Absolute maximum ratings 18
Table 6: Power Modes 28
Table 7: Cutoff freq. of the accelerometer according to ODR, Normal & Performance Mode 30
Table 8: Accelerometer noise according to ODR, Normal & Performance Mode, +/- 8g range 30
Table 9: Accelerometer group delay according to ODR, Normal & Performance Mode 30
Table 10: Cutoff frequency of the gyroscope according to ODR, Normal & Performance Mode 32
Table 11: Gyroscope noise according to ODR, Normal Mode, +/- 2000 dps range 32
Table 12: Gyroscope noise according to ODR, Performance Mode, +/- 2000 dps range 32
Table 13: Gyroscope group delay according to ODR, Normal and Performance Modes 32
Table 14: Positive use-case scenarios for the wrist wear wakeup gestures 56
Table 15: Step counter Configuration 60
Table 16: OIS Register Map 68
Table 17: Mapping for primary interface 138
Table 18: Electrical specifications of the interface pins 138
Table 19: Timing specifications for SPI 139
Table 20: Timing specification for I²C of the device 144
Table 21: Mapping of the device pins for secondary interface 148
Table 22: Pin-out and pin connections 151

Figure 1: Any-motion detection 50
Figure 2: No-motion detection 52
Figure 3: Device co-ordinate system assumed for gesture detection 55
Figure 4: Flick-in/out movement 57
Figure 5: Push arm-down/pivot up 58
Figure 6: Wrist Jiggle 58
Figure 7: 9-DOF Solution w/ magnetometer (AUX) connected to the 2nd I/F 62
Figure 8: OIS interface 67
Figure 9: Digital Interfaces 137
Figure 10: SPI timing diagram 140
Figure 11: I²C timing diagram 145

1. Specification

Unless stated otherwise, the given values are over lifetime, operating temperature and voltage ranges. Minimum/maximum values are ±3σ.

Table 1: Basic electrical parameter specifications

OPERATING CONDITIONS						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Supply Voltage Internal Domains	V_{DD}		1.71	1.8	3.6	V
Supply Voltage I/O Domain	V_{DDIO}		1.2	1.8	3.6	V
Voltage Input Low Level	VIL	SPI & I^2C			0.3VDDIO	-
Voltage Input High Level	VIH	SPI & I^2C	0.7VDDIO			-
Voltage Output Low Level	VOL	VDDIO>=1.62V, IOL<=2mA, SPI			0.2VDDIO	-
		VDDIO<1.62V, IOL<=1.5mA, SPI			0.2VDDIO	
Voltage Output High Level	VOH	VDDIO>=1.62V, IOH<=2mA, SPI	0.8VDDIO			-
		VDDIO<=1.62V, IOH<=1.5mA, SPI	0.8VDDIO			
Current consumption	I_{DD}	A+G Performance Mode VDD= 1.8 V, T_A=25°C, ODR_{max}		970		µA
		A+G Normal Mode VDD= 1.8 V, T_A=25°C, ODR_{max}		685		
		A+G Low Power Mode VDD= 1.8 V, T_A=25°C, ODR_{25Hz}		420		
		A_{only} Normal Mode VDD= 1.8 V, T_A=25°C, ODR_{max}		200		
		A_{only} Low Power Mode VDD= 1.8 V, T_A=25°C, ODR_{25Hz}		10		
		A+G Suspend mode, VDD= 1.8 V, T_A=25°C		3.5		
		Advanced features VDD= 1.8 V, TA=25°C, depends on enabled feature set		3		

Power on time	t_{PO}	Time from supply "on" to SPI or I2C I/F operational			0.5	ms
Non-volatile memory (NVM) write-cycles	nNVM	Using nvm_prog cmd			14	cycles
Operating Temperature	T_A		-40		+85	°C

Table 2: Accelerometer characteristics

OPERATING CONDITIONS ACCELEROMETER						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Acceleration Range	g_{FS2g}	Selectable via serial digital interface		±2		g
	g_{FS4g}			±4		g
	g_{FS8g}			±8		g
	g_{FS16g}			±16		g
Start-up time	$t_{A,SU}$	suspend to normal mode VDD= 1.8 V, T_A=25°C, ODR_{max}		2		ms

OUTPUT SIGNAL ACCELEROMETER						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Resolution				16		bit
Sensitivity	S_{2g}	g_{FS2g}, T_A=25°C		16384		LSB/g
	S_{4g}	g_{FS4g}, T_A=25°C		8192		LSB/g
	S_{8g}	g_{FS8g}, T_A=25°C		4096		LSB/g
	S_{16g}	g_{FS16g}, T_A=25°C		2048		LSB/g
Sensitivity Error	$S_{A\_err\_8g}$	T_A=25°C, nominal V_{DD} soldered, over life time		±0.4		%
Sensitivity Temperature Drift	TCS_A	full T_A range, nominal V_{DD} best fit straight line		0.004		%/K
Sensitivity Supply Volt. Drift	$S_{A,VDD}$	T_A=25°C, full V_{DD} range soldered, over life time		0.0001		%/V
Zero-g Offset	$Off_{A,life}$	T_A=25°C, nominal V_{DD} soldered, over life time		±20		mg
Zero-g Offset Temperature Drift	TCO_A	full T_A range, nominal V_{DD} best fit straight line		±0.25		mg/K
Zero-g Offset Supply Volt. Drift	$Off_{A,VDD}$	T_A=25°C, full V_{DD} range soldered, over life time		<0.5		mg/V
Power supply rejection ratio	Off_PSRR_A	100Hz – 1 MHz sine wave, 50mV		< 8		mg/50mV

Output Noise	$n_{A,nd}$	Normal mode T_A = 25°C, nominal V_{DD}, range = 8g		0.16		mg/√Hz
	$n_{A,rms}$	Normal mode T_A=25°C, nominal V_{DD}, BW = 80 Hz ODR = 200 Hz range = 8g		1.51		mg-rms
Nonlinearity	NL_A	T_A=25°C, nominal V_{DD}, best fit straight line g_{FS2g}		0.5		%FS
Output Data Rate	$ODR_{A,n}$	Normal mode	12.5		1600	Hz
	$ODR_{A,lpm}$	Low-power mode	0.78		400	
ODR Accuracy	$OAcy_{A,n}$	Normal mode, variation part to part, T_A=25°C, nominal V_{DD}, Accel only operation		1		%
		Normal mode, variation part to part, T_A=25°C, nominal V_{DD}, IMU operation		1.7		
	$OAcy_{A,n,T}$	Normal mode, variation full T_A range, same part nominal V_{DD}, Accel only operation		0.03		%/K
		Normal mode, variation full T_A range, same part nominal V_{DD}, IMU operation		0.0037		
Bandwidth (BW) in normal mode	$ODR_{A,12.5}$	3dB cutoff frequency of the accelerometer T_A=25°C, nominal V_{DD}, Filter setting [acc_bwp] = 0x02		5.5		Hz
	$ODR_{A,25}$			11		
	$ODR_{A,50}$			22		
	$ODR_{A,100}$			44		
	$ODR_{A,200}$			89		
	$ODR_{A,400}$			178		
	$ODR_{A,800}$			343		
	$ODR_{A,1600}$			740		

MECHANICAL CHARACTERISTICS ACCELEROMETER						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Cross Axis Sensitivity	SX_A	Relative contribution between any two of the three axes		1		%
Alignment Error	E_A	Relative to package outline		0.5		°
Zero-g offset over PCB strain	$Off_{A,PCB}$	T_A=25°C, nominal V_{DD} soldered, ∅ 5 parts		±0.010		mg/µstrain

Table 3: Gyroscope Characteristics

OPERATING CONDITIONS GYROSCOPE						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Range	R_{FS125}	Selectable via serial digital interface		125		dps
	R_{FS250}			250		dps
	R_{FS500}			500		dps
	R_{FS1000}			1,000		dps
	R_{FS2000}			2,000		dps
Start-up time	$t_{G,SU}$	suspend to normal mode repeated, VDD = 1.8 V, T_A=25°C, ODR_{max}		45		ms
	$t_{G,FSU}$	fast start mode VDD = 1.8 V, T_A=25°C, ODR_{max}		2		

OUTPUT SIGNAL GYROSCOPE						
Resolution				16		bit
Sensitivity	R_{FS2000}	Ta=25°C		16.384		LSB/dps
	R_{FS1000}	Ta=25°C		32.768		LSB/dps
	R_{FS500}	Ta=25°C		65.536		LSB/dps
	R_{FS250}	Ta=25°C		131.072		LSB/dps
	R_{FS125}	Ta=25°C		262.144		LSB/dps
Sensitivity Error	S_{G_err}	T_A=25°C, nominal V_{DD} soldered, over life time		±2		%
	$S_{G\_err\_CRT}$	T_A=25°C, nominal V_{DD} soldered, after CRT		±0.4		
Sensitivity Temperature Drift	TCS_G	full T_A range, nominal V_{DD} best fit straight line		0.02		%/K
Sensitivity Supply Volt. Drift	$S_{G,VDD}$	T_A=25°C, full V_{DD} range soldered, over life time		0.0005		%/V

Zero-rate offset	$\Omega_{,oL}$	T_A=25°C, nominal V_{DD} soldered, over life time		±0.5		dps
Zero-rate offset change over temperature	TCO_G	Nominal V_{DD} supplies best fit straight line		±0.015		dps / K
Zero-rate offset Supply Volt. Drift	$Off_{G,VDD}$	T_A=25°C, full V_{DD} range soldered, over life time		0.02		dps / V
Power supply rejection ratio	Off_PS RR_G	100 Hz – 1 MHz sine wave, 50mV		0.04		dps / 50mV
Output Noise	$n_{G,nd}$	Performance mode T_A=25°C, nominal V_{DD}		0.007		dps /√Hz
		Normal mode T_A=25°C, nominal V_{DD}		0.010		dps /√Hz
	$n_{G,rms}$	Performance mode T_A=25°C, nominal V_{DD}, BW = 74.6 Hz ODR = 200 Hz		0.07		dps-rms
		Normal mode T_A=25°C, nominal V_{DD}, BW = 74.6 Hz ODR = 200 Hz		0.09		dps-rms
Nonlinearity	NL_G	T_A=25°C, nominal V_{DD}, best fit straight line R_{FS250}, R_{FS2000}		0.01		% FS
Output Data Rate	$ODR_{G,n,hp}$	Normal and performance mode	25		6400	Hz
	$ODR_{G,lpm}$	Low-power mode	25		100	
ODR Accuracy	$OAcy_{G,n}$	Normal and performance mode, T_A=25°C, nominal V_{DD}		1.7		%
	$OAcy_{G,n,T}$	Normal mode, full T_A range, same part, nominal V_{DD}		0.0037		%/K
Bandwidth (BW) in normal and performance mode	$ODR_{G,25}$	3dB cutoff frequency of the accelerometer, T_A=25°C, nominal V_{DD} Filter setting [gyr_bwp] = 0x02		11		Hz
	$ODR_{G,50}$			20		
	$ODR_{G,100}$			39		
	$ODR_{G,200}$			77		
	$ODR_{G,400}$			152		
	$ODR_{G,800}$			300		
	$ODR_{G,1600}$			557		

	$ODR_{G,3200}$			751		
	$ODR_{G,6400}$			712		

MECHANICAL CHARACTERISTICS GYROSCOPE						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Cross Axis Sensitivity	SX_G	Relative contribution between any two of the three axes [1]		0.2		%
Alignment Error	E_A	Relative to package outline		0.5		°
Zero-rate offset over PCB strain	$Off_{G, PCB}$	T_A=25°C, nominal V_{DD} soldered, ∅ 5 parts		±1.5		mdps / µstrain
g-Sensitivity		Sensitivity to static acceleration stimuli in all three axis			0.1	dps / g

Table 4: Electrical characteristics temperature sensor

OPERATING CONDITIONS AND OUTPUT SIGNAL OF TEMPERATURE SENSOR						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
ADC Resolution				16		bits
Temperature Sensor Measurement Range	T_S		-41		87	°C
Output at 23 °C				0		LSB
Sensitivity	S_T			512		LSB/K
Output Data Rate Temperature Sensor	$ODR_{T,G}$	Normal mode and performance mode, T_A=25°C, nominal V_{DD}, Gyroscope on			100	Hz
	ODR_T	All other modes incl. low power mode			0.78	
ODR Accuracy Temperature Sensor	$OAcy_{T,G}$	Normal mode and performance mode, T_A=25°C, nominal V_{DD}, Gyroscope on		<1.5	2	%
	$OAcy_T$	All other modes incl. low power mode		1.5		

---

[1] For details see Section 4.6

2. Absolute maximum ratings

Stress above these limits may cause damage to the device. Exceeding the specified electrical limits may affect the device reliability or cause malfunction.

Table 5: 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	-0.3	V_{DDIO}+0.3, <4	V
Passive Storage Temp. Range	≤ 65% rel. H.	-50	+150	°C
Non-volatile memory (NVM) Data Retention	T = 85°C, after 15 cycles	10		y
Mechanical Shock	Duration ≤ 200µs		10,000	g
	Duration ≤ 1.0ms		2,000	g
	Free fall onto hard surfaces		1.8	m
ESD according JESD47	HBM at any pin		2	kV
	CDM		500	V
	MM JESD22A115C		35	V

3. Quick Start Guide

The purpose of this section is to help developers who want to start working with the device by giving some basic hands-on application examples to get started.

Note about using the device

The communication between application processor and the device will happen either over I2C or SPI interface. For more information about the interfaces, see Section 6.

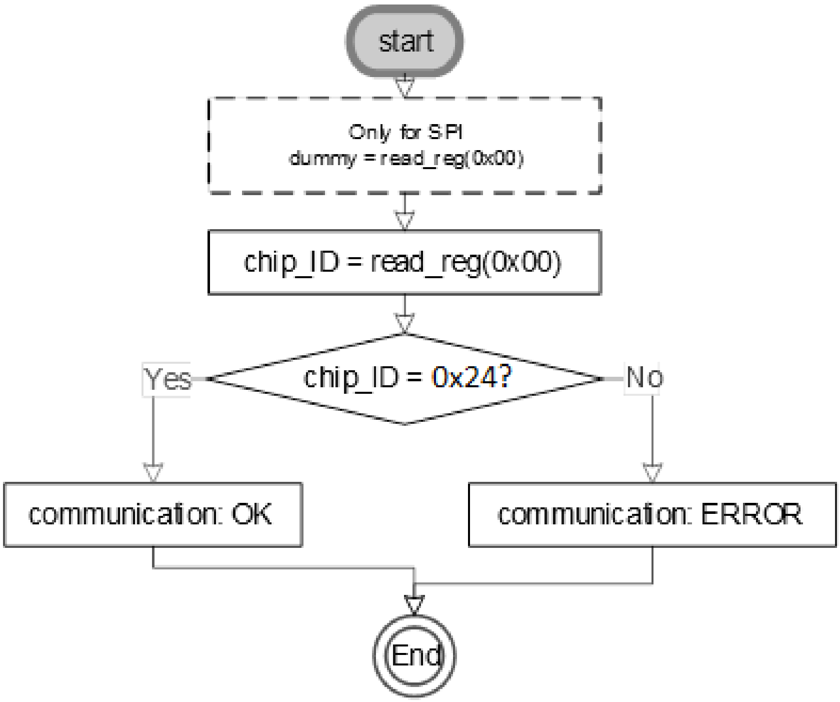
Before starting the test, the device has to be properly connected to the master (AP) and powered up. For more information, read related Section 7.

First application setup examples algorithms:

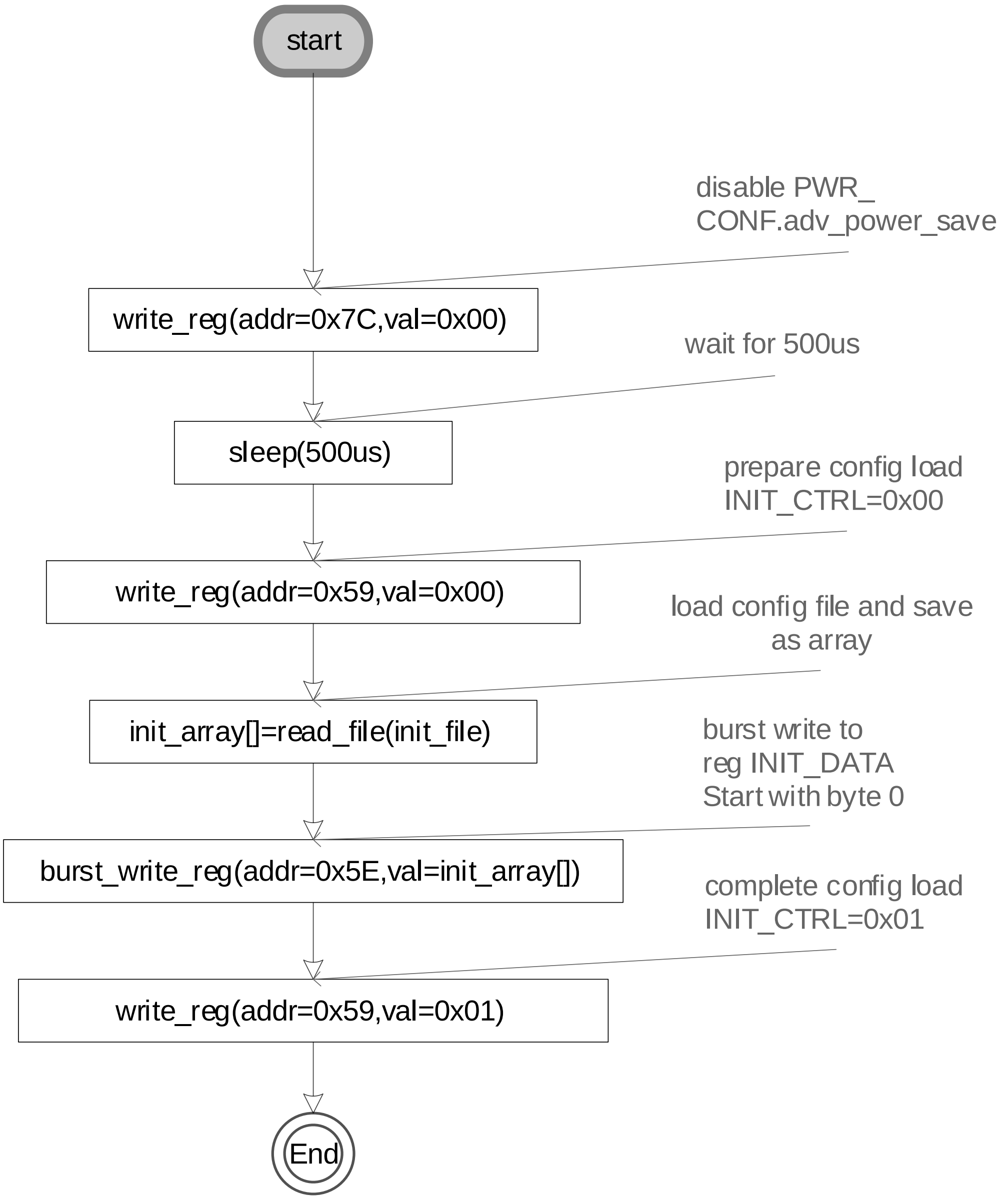
After correct power up by setting the correct voltage to the appropriate external pins, the device enters automatically into the Power On Reset (POR) sequence. In order to properly use the device, certain steps from host processor front are needed. The most typical operations will be explained in the following application examples in form of flow diagrams.

1. ***Testing communication and initializing the device***

   a. Reading chip id CHIP_ID (0x24) (checking correct communication). The interface is coming up configured for I2C, the initial dummy read configures it to SPI.)



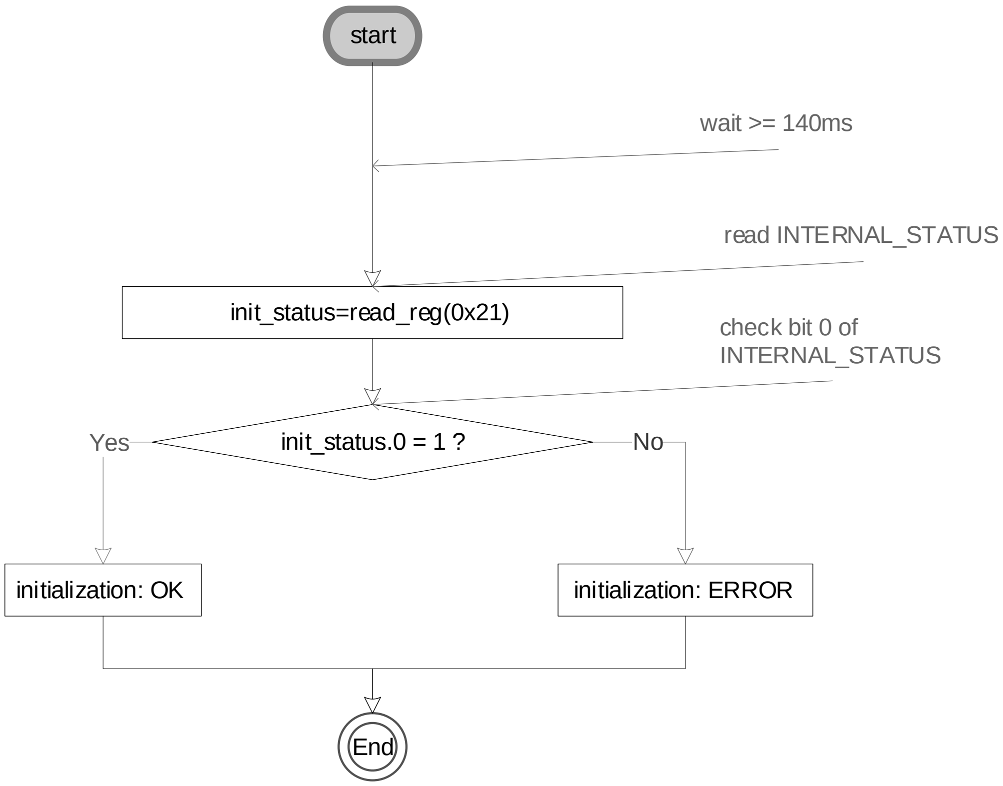
b. Performing initialization sequence[2]



---

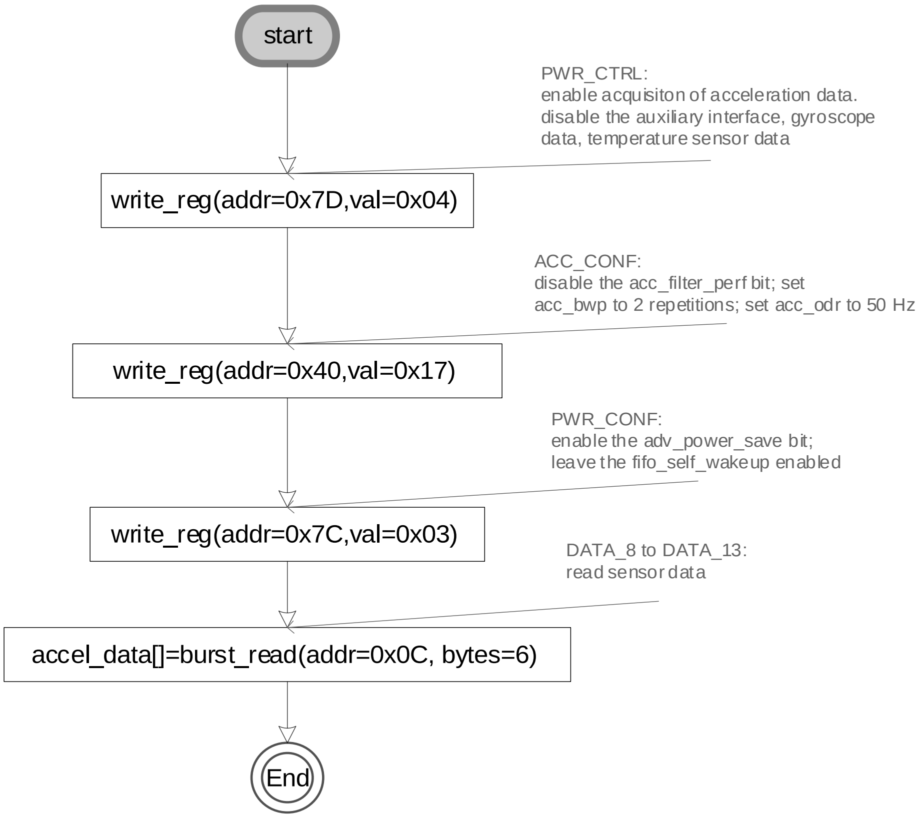
[2] The config file BMI270_main.tbin is provided by BST regional support team as part of the APIs and drivers.

c. Checking the correct initialization status



2. *Configuring the device for low power mode*

Setting data processing parameters (power, bandwidth, range) and reading sensor data



PWR_CTRL:
enable acquisiton of acceleration data.
disable the auxiliary interface, gyroscope data, temperature sensor data

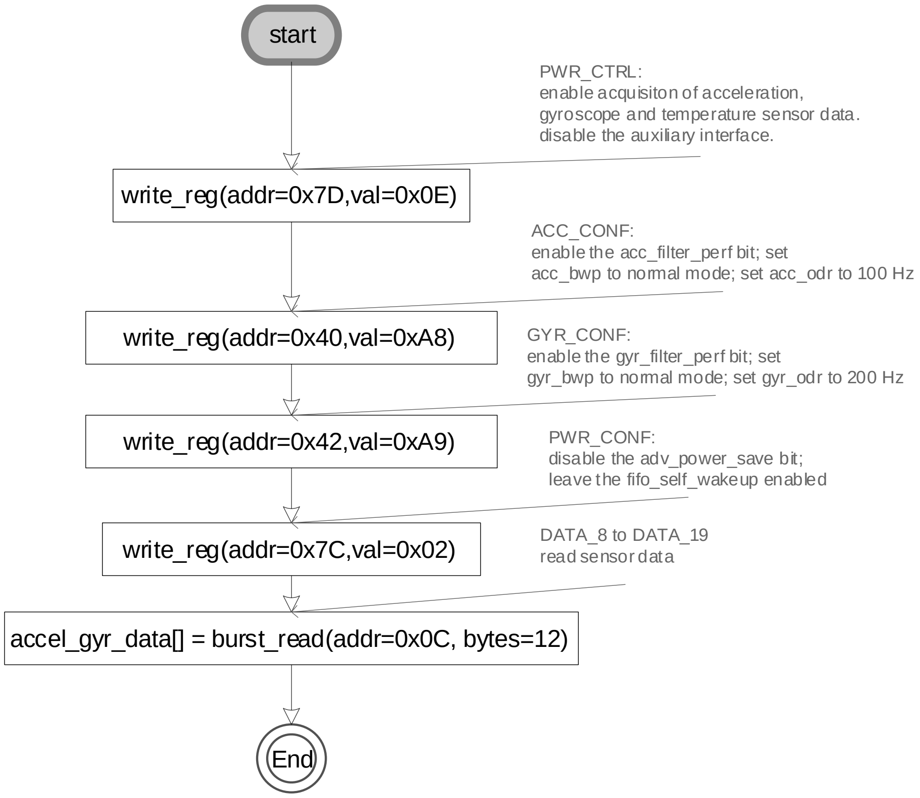
ACC_CONF:
disable the acc_filter_perf bit; set acc_bwp to 2 repetitions; set acc_odr to 50 Hz

PWR_CONF:
enable the adv_power_save bit;
leave the fifo_self_wakeup enabled

DATA_8 to DATA_13:
read sensor data

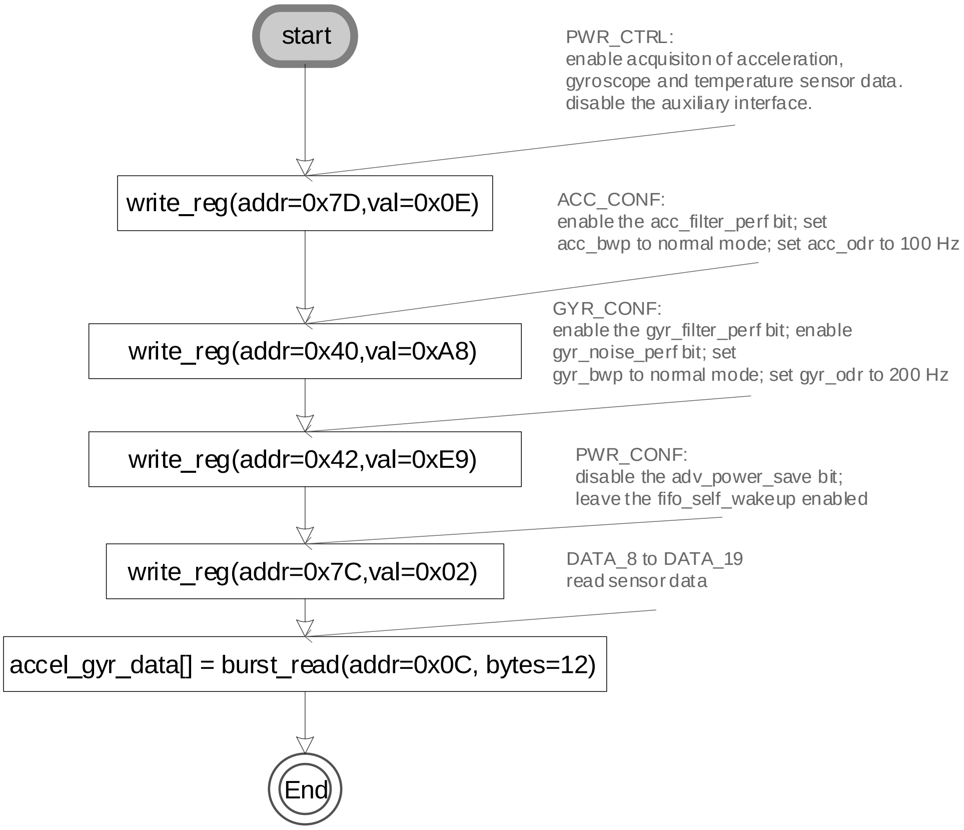
3. Configuring the device for normal power mode

Setting data processing parameters (power, bandwidth, range) and reading sensor data



4. Configuring the device for performance mode

Setting data processing parameters (power, bandwidth, range) and reading sensor data



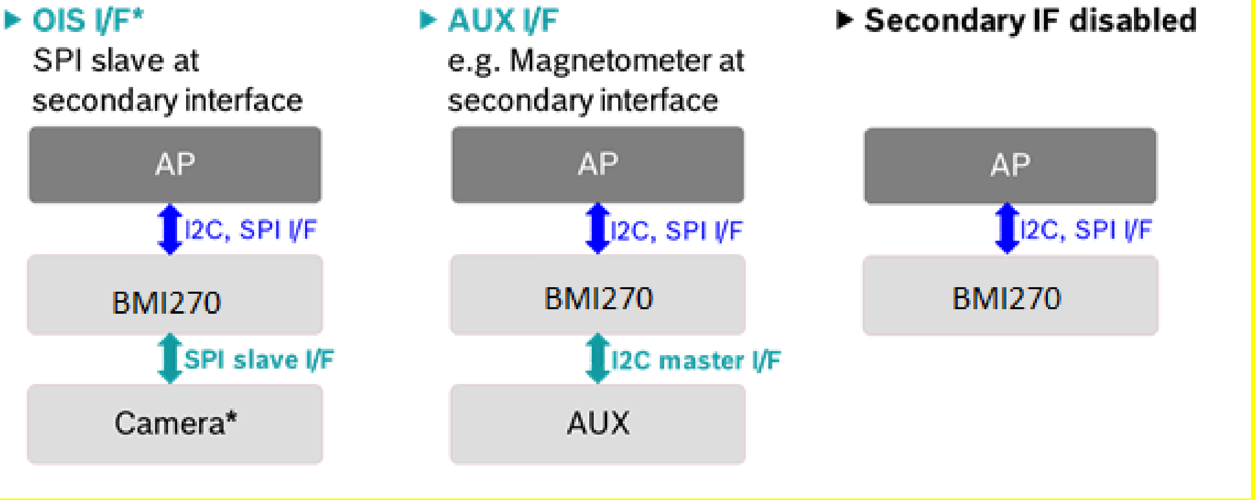
Further steps:
The device has additional capabilities that are described in this document and include FIFO, power saving modes, synchronization capabilities with host processor, data synchronization and integration with third party sensors (see Section 4).

4. 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 Section 5.

4.1. System Configurations

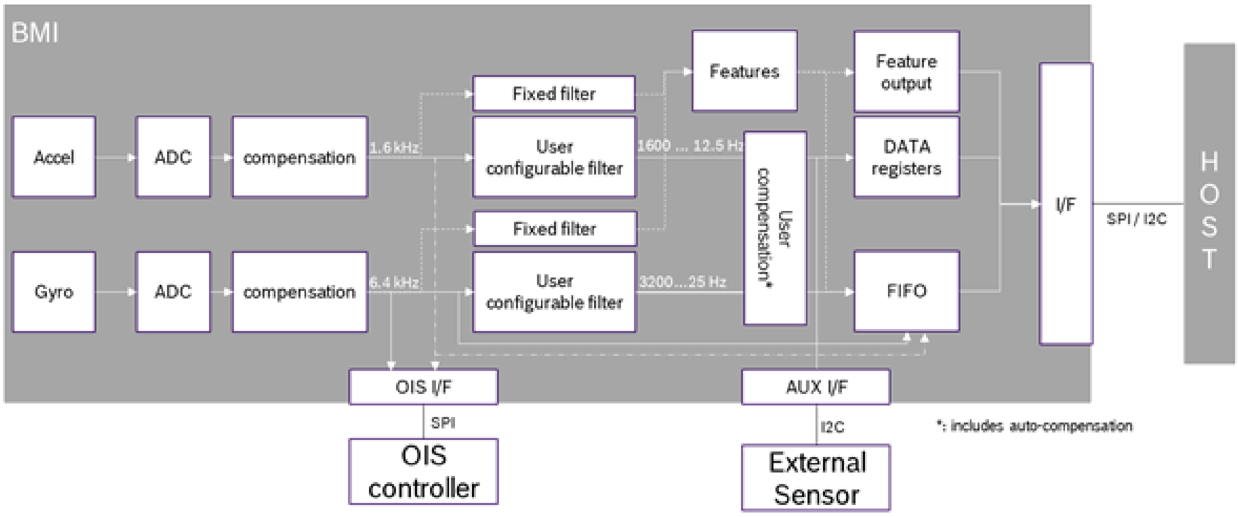
The device has 14 external I/F pins and supports the SPI and I2C protocols on its primary interface to the host system. The device supports on its secondary interface (I2C master) an auxiliary sensor configuration (e.g. a magnetometer) configuration or an external OIS interface. Both configurations work independent of the configuration (SPI/I2C) of the primary interface. If the secondary I/F is configured as AUX I/F, the sensor data of the IMU and the AUX sensor are synchronized (see Section 4.10).



The device includes two sensors, an accelerometer and a gyroscope. The accelerometer measures the direction and magnitude of the force applied to the sensor, reporting zero in a free fall scenario. The gyroscope measures the rotation rate, reporting zero at rest.

4.2. Block Diagram

BMI270



4.3. Supply Voltage and Power Management

The device has two distinct power supply pins:

- VDD is the main power supply.
- VDDIO is a separate power supply pin used for supplying power for the interface including the auxiliary interface.

There are no limitations with respect to the voltage level applied to the VDD and VDDIO pins, as long as it lies within the respective operating range. Furthermore, the device can be completely switched off (VDD= 0V) while keeping the VDDIO supply within operating range or vice versa. However, if the VDDIO supply is switched off, all interface pins (CSB, SDX, SCX) must be kept close to GNDIO potential. The device is reset when the supply voltage applied to at least one supply pin VDD or VDDIO falls below the specified minimum values. No constraints exist for the minimum slew-rate of the voltage applied to the VDD and VDDIO pins.

4.4. Power-On-Reset (POR) and Device Initialization

During POR the voltages VDD/VDDIO are ramped to their respective target values. After reaching the target supply voltages, all registers are accessible after a delay of 450µs.

After every POR or soft-reset the IMU remains in suspend mode. To get ready for operation the device must be initialized through the following procedure:

- *Disable advanced power save mode:* PWR_CONF.adv_power_save *=0b0*
- *Wait for 450 us (or 12 LSB of* SENSORTIME_0*)*
- Write INIT_CTRL.init_ctrl*=0x00 – to prepare* config load
- Upload configuration file
    - Burst write 8 kB initialization data to Register INIT_DATA (start with byte 0 of initialization data)[3]. This requires ca. 6.6 ms at 10 MHz SPI I/F frequency. The configuration file BMI270_main.tbin is included in the driver available on the Bosch Sensortec website (www.bosch-sensortec.com) or from your regional support team.
    - Optionally: Burst read configuration file from Register INIT_DATA and check correctness by comparing it to the data written to the register in the previous step.
- Write INIT_CTRL.init_ctrl=0x01 – to complete config load.
  Note: This operation **must not** be performed more than once after POR or soft-reset.
- *Wait until Register* INTERNAL_STATUS.message *contains the value 0b0001. This will happen after at most 150 msec.*

After the initialization sequence is completed, the power mode of the device is automatically set to “Configuration mode” (refer to Section 3.2) Now it is possible to switch to other power modes and the device is ready for operation as required and described in the following sections.

---

[3] If the maximum burst write length of the host is less than 8 kB the initialization data can be written in smaller chunks. Between two write operations the Registers INIT_ADDR_0 and INIT_ADDR_1 need to be incremented by the length of the first chunk write operation in bytes/2.

4.5. **Power Modes**

The main power modes of the device are:

- Suspend mode: Lowest possible power consumption, while still maintaining its configuration
- Configuration mode: All IMU features accessible at full interface speed
- Low power mode: Motion sensing at lowest possible power consumption
- Normal mode: Aliasing free motion sensing at maximum ODR
- Performance mode: Motion sensing at maximum sensor performance

The table below shows the required configurations for these power modes

Table 6: Power Modes

		PWR_CTRL.acc_en	PWR_CTRL.gyr_en	ACC_CONF.acc_filter_perf	GYR_CONF.gyr_filter_perf	GYR_CONF.gyr_noise_perf	PWR_CONF.adv_power_save	Typ. current consumption * depends on ACC_CONF ** depends on GYR_CONF
Suspend (lowest power mode)		0	0	X	X	X	1	6 µA
Configuration mode		X	X	X	X	X	0	50 µA
Low power mode	Accel only	1	0	0	X	X	1	Down to 6 µA*
	Gyro only	0	1	X	0	0		Down to 400 µA**
	IMU	1	1	0	0	0		Down to 420 µA* **
Normal mode[4]	Accel only	1	0	1	X	X	X	200 µA
	Gyro only	0	1	X	1	0		650 µA
	IMU	1	1	1	1	0		685 µA
Performance mode[1]	Accel only	1	0	1	X	X	X	200 µA
	Gyro only	0	1	X	1	1		900 µA
	IMU	1	1	1	1	1		970 µA

The power state of the IMU is controlled through the registers PWR_CTRL and PWR_CONF. The Register PWR_CTRL enables and disables the accelerometer, the gyroscope, the auxiliary sensor, and the temperature sensor. The Register PWR_CONF controls which power state the sensors enter if they are enabled or disabled in the Register PWR_CTRL. The power state impacts the behavior of the sensor with respect to start-up time, available functions, etc. but not the sensor data quality. The sensor data quality (e.g. the noise performance and/or the filter characteristics) is controlled in the Register ACC_CONF and Register GYR_CONF. In all global power configurations both register contents and FIFO contents are retained.

---

[4] Accelerometer does not differ in normal and performance mode.

Section 4.5 shows how to configure the device for the most relevant power modes. But any other combination of the shown register settings is allowed. These registers are described as follows:

PWR_CTRL: used to enable and disable sensors (accelerometer, gyroscope, auxiliary, and temperature). Per default, all sensors are disabled.

PWR_CTRL.acc_en: enable or disable the accelerometer in all power modes.

PWR_CTRL.gyr_en: enable or disable the gyroscope in all power modes.

ACC_CONF.acc_filter_perf: enable or disable aliasing[5] free acceleration sensing.

GYR_CONF.gyr_filter_perf: enable or disable aliasing² free yaw rate sensing.

GYR_CONF.gyr_noise_perf: enable or disable low noise mode for precision yaw rate sensing.

PWR_CONF.adv_power_save: enable or disable the advanced power save configuration. If the device is configured for accelerometer only operation and ACC_CONF.acc_filter_perf=0b0 or all sensors are disabled, there is a potential for additional (maximal) power saving. If the configuration is set by PWR_CONF.adv_power_save=0b1, the devices internally reduces the power consumption always to a minimum without compromising data quality defined by the performance parameters set above at the expense of these restrictions which apply:

- Register writes need an inter-write-delay of at least 1 ms.
- The sensors log data into the FIFO in all power modes. The user needs to disable advanced power save mode (PWR_CONF.adv_power_save=0b0), respect the timing constraints in Sections 6.4 and 6.5 before reading the FIFO.

If PWR_CONF.adv_power_save=0b0 the device is accessible without the restrictions of the advanced power save configuration after 1 ms.

---

[5] An effect that causes different signals to become indistinguishable when sampled with low ODR's.

4.6. Sensor Data

Accelerometer Data

The width of acceleration data is 16 bits given in two´s complement representation in the registers DATA_8 to DATA_13. The 16 bits for each axis are split into an MSB upper part and an LSB lower part. Reading the acceleration data registers shall always start with the LSB part. In order to ensure the integrity of the acceleration data, the content of an MSB register is locked by reading the corresponding LSB register (shadowing procedure).

Accelerometer Filter Settings

The accelerometer digital filter is configured through the Register ACC_CONF.

Accelerometer Filter Modes

The accelerometer filter modes influence the low pass filter characteristics, in particular the 3dB cutoff frequency, noise, and group delay. The accelerometer filter mode is configured through ACC_CONF.acc_bwp. This datasheet describes the device in normal filter mode configuration for ACC_CONF.acc_bwp=0x02.

Accelerometer Data Processing in Normal and Performance Mode

The data processing for this mode is configured using ACC_CONF.acc_filter_perf=0b1. In this power mode, the accelerometer data is sampled at equidistant points in the time, defined by the accelerometer output data rate parameter ACC_CONF.acc_odr. The output data rate can be configured in one of eight different valid ODR configurations going from 12.5 Hz up to 1600Hz.

The characteristics of the implemented low pass filter are described in the following tables:

Table 7: Cutoff freq. of the accelerometer according to ODR, Normal & Performance Mode

Accelerometer ODR [Hz]	12.5	25	50	100	200	400	800	1600
3dB Cutoff frequency [Hz] normal filter mode ACC_CONF.acc_bwp=0x02	5.5	11	22	44	89	178	343	740

Table 8: Accelerometer noise according to ODR, Normal & Performance Mode, +/- 8g range

ODR in Hz	12.5	25	50	100	200	400	800	1600
RMS-Noise (typ.) [mg] normal filter mode ACC_CONF.acc_bwp=0x02	0.38	0.53	0.75	1.06	1.51	2.13	2.96	4.35

Table 9: Accelerometer group delay according to ODR, Normal & Performance Mode

ODR in Hz	12.5	25	50	100	200	400	800	1600
Group Delay (typ.) [ms] normal filter mode ACC_CONF.acc_bwp=0x02	80	40	20.5	10.5	5.4	2	1.3	0.6

Accelerometer Data Processing in Low Power Mode

Low power mode can be enabled by PWR_CONF.adv_power_save=0b1 and ACC_CONF.acc_filter_perf=0b0. In this power mode, the accelerometer regularly changes between an idle phase where no measurement is performed and an active phase, where data is acquired. The period of the duty cycle for changing between active and idle mode will be determined by the output data rate (ACC_CONF.acc_odr). The output data rate can be configured in one of 10 different valid ODR configurations going from 0.78Hz up to 400Hz.

The samples acquired during the active phase will be averaged and the result will be the output data. The number of averaged samples can be determined by the parameter ACC_CONF.acc_bwp through the following formula:

$$\text{averaged samples} = 2^{(\text{Val(acc\_bwp)})}$$

$$\text{skipped samples} = (1600/\text{ODR})\text{-averaged samples}$$

A higher 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.

Accelerometer Data Ready Interrupt

This interrupt fires whenever a new data sample set from accelerometer is available in Registers DATA_8 to DATA_13. This allows a low latency data readout. In non-latched mode, the interrupt are cleared automatically after 1/(6400Hz). If this automatic clearance is unwanted, please use latched mode (see Section 4.9). The flag INT_STATUS_1.acc_drdy_int is cleared when the register INT_STATUS_1 is read. The flag STATUS.drdy_acc is cleared when any of the Registers DATA_8 to DATA_13 is read.

To enable the data ready interrupt please map it on the desired INT pin via INT_MAP_DATA.

Gyroscope Data

The width of gyroscope data is 16 bits given in two´s complement representation in the registers DATA_14 to DATA_19. The 16 bits for each axis are split into an MSB upper part and an LSB lower part. Reading the gyroscope data registers shall always start with the LSB part. In order to ensure the integrity of the gyroscope data, the content of an MSB register is locked by reading the corresponding LSB register (shadowing procedure).

Gyroscope Filter Settings

The gyroscope digital filter can be configured through the Register GYR_CONF.

Gyroscope Filter Modes

The gyroscope filter modes influence the low pass filter characteristics, in particular the 3dB cutoff frequency, noise, and group delay. The gyroscope filter mode is configured through GYR_CONF.gyr_bwp. This datasheet describes the device in normal filter mode configuration for GYR_CONF.gyr_bwp=0x02.

Gyroscope Data Post-Processing

To further optimize the gyroscope performance the following data post-processing step is recommended:

Rate$_x$ = DATA_15<<8+DATA_14 - GYR_CAS.factor_zx * (DATA_19<<8+DATA_18) / 2^9

Rate$_y$ =DATA_17 <<8+DATA_16

Rate$_z$ =DATA_19<<8+DATA_18

Gyroscope Data Processing in Normal and Performance Mode

The data processing for these modes is configured using GYR_CONF.gyr_filter_perf=0b1. In these power modes, the gyroscope data is sampled at equidistant points in the time, defined by the gyroscope output data rate parameter GYR_CONF.gyr_odr. The output data rate can be configured in one of eight different valid ODR configurations going from 25 Hz up to 3.2 kHz. For 6.4 kHz operation use FIFO data readout described in section 4.7.

The characteristics of the implemented low pass filter are described in the following tables:

Table 10: Cutoff frequency of the gyroscope according to ODR, Normal & Performance Mode

Gyroscope ODR [Hz]	25	50	100	200	400	800	1.6 k	3.2 k	6.4 k
3dB Cutoff frequency [Hz] normal filter mode GYR_CONF.gyr_bwp=0x02	11	20	39	77	152	300	557	751	712

Table 11: Gyroscope noise according to ODR, Normal Mode, +/- 2000 dps range

ODR in Hz	25	50	100	200	400	800	1.6 k	3.2 k	6.4 k
RMS-Noise (typ.) [mdps] normal filter mode GYR_CONF.gyr_bwp=0x02	31,0	43,9	62,0	87,7	124	176	248	431	500

Table 12: Gyroscope noise according to ODR, Performance Mode, +/- 2000 dps range

ODR in Hz	25	50	100	200	400	800	1.6 k	3.2 k	6.4 k
RMS-Noise (typ.) [mdps] normal filter mode GYR_CONF.gyr_bwp=0x02	21,7	30,7	43,4	61,4	86,9	123	174	302	350

Table 13: Gyroscope group delay according to ODR, Normal and Performance Modes

ODR in Hz	25	50	100	200	400	800	1.6 k	3.2 k	6.4 k
Group Delay (typ.) [ms] normal filter mode GYR_CONF.gyr_bwp=0x02	40	20.5	10.8	5.97	3.55	2.34	0.97	0.82	0.68

Gyroscope Data Processing in Low Power Mode

Low power mode can be enabled by PWR_CONF.adv_power_save=0b1 and GYR_CONF.gyr_filter_perf=0b0. In this power mode, the gyroscope regularly changes between an idle phase where no measurement is performed and an active phase, where data is acquired. The period of the duty cycle for changing between active and idle mode will be determined by the output data rate (GYR_CONF.gyr_odr). The output data rate can be configured in one of 3 different valid ODR configurations 25Hz, 50Hz, and 100Hz.
Four samples are acquired during the active phase and will be averaged and the result will be the output data.

Gyroscope Data Ready Interrupt

This interrupt fires whenever a new data sample set from the gyroscope is available in Registers DATA_14 to DATA_19. This allows a low latency data readout. In non-latched mode, the interrupt are cleared automatically after 1/(6400Hz). If this automatic clearance is unwanted, please use latched mode (see Section 4.9). The flag INT_STATUS_1.gyr_drdy_int is cleared when the register INT_STATUS_1 is read. The flag STATUS.drdy_gyr is cleared when any of the Registers DATA_14 to DATA_19 is read.

To enable the data ready interrupt please map it on the desired INT pin via INT_MAP_DATA.

Temperature Sensor

The temperature sensor has 16 bits defined as:

Value	Temperature
0x7FFF	87 – 1/2^9 °C
...	...
0x0000	23 °C
...	...
0x8001	-(41-1/2^9) °C
0x8000	Invalid

The measured temperature is accessible via the Registers TEMPERATURE_0 and TEMPERATURE_1. After enabling the temperature sensor, the register contains the invalid value 0x8000 until the first temperature measurement is completed.

If the gyroscope is enabled (i.e. PWR_CTRL.gyr_en=0b01) the temperature sensor is automatically enabled at an update rate of 100 Hz +/-12%.

If the gyroscope is disabled (i.e. PWR_CTRL.gyr_en=0b00, e.g. for accelerometer only operation) the temperature sensor must be manually enabled or disabled using PWR_CTRL.temp_en. Disabling the temperature sensor reduces the overall current consumption of the device by approximately 1.8 µA in average. If the temperature sensor is enabled it updates the results aligned with bit 7 of the Register SENSORTIME_1 at an update rate of 0.78 Hz.

Sensor Time

The device supports the concept of sensortime. Its core element is a free running counter with a width of 24 bits. It increments with a resolution of 39.0625us. The user can access the current state of the counter by reading registers SENSORTIME_0 to SENSORTIME_2.
All sensor events e.g. updates of data registers are synchronous to this sensor time register as defined in the table below. With every update of the data register or the FIFO, a bit *m* in the registers SENSORTIME_0 to SENSORTIME_2 toggles where *m* depends on the output data rate for the data register and the output data rate and the FIFO downsampling rate for the FIFO. The table below shows which bit toggles for which update rate of data register and FIFO

Bit m in sensor_time	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
Update rate [Hz]	0.0031	0.0061	0.012	0.024	0.049	0.10	0.20	0.39

Bit *m* in sensor_time	15	14	13	12	11	10	9	8
Resolution [ms]	1280	640	320	160	80	40	20	10
Update rate [Hz]	0.78	1.56	3.125	6.25	12.5	25	50	100

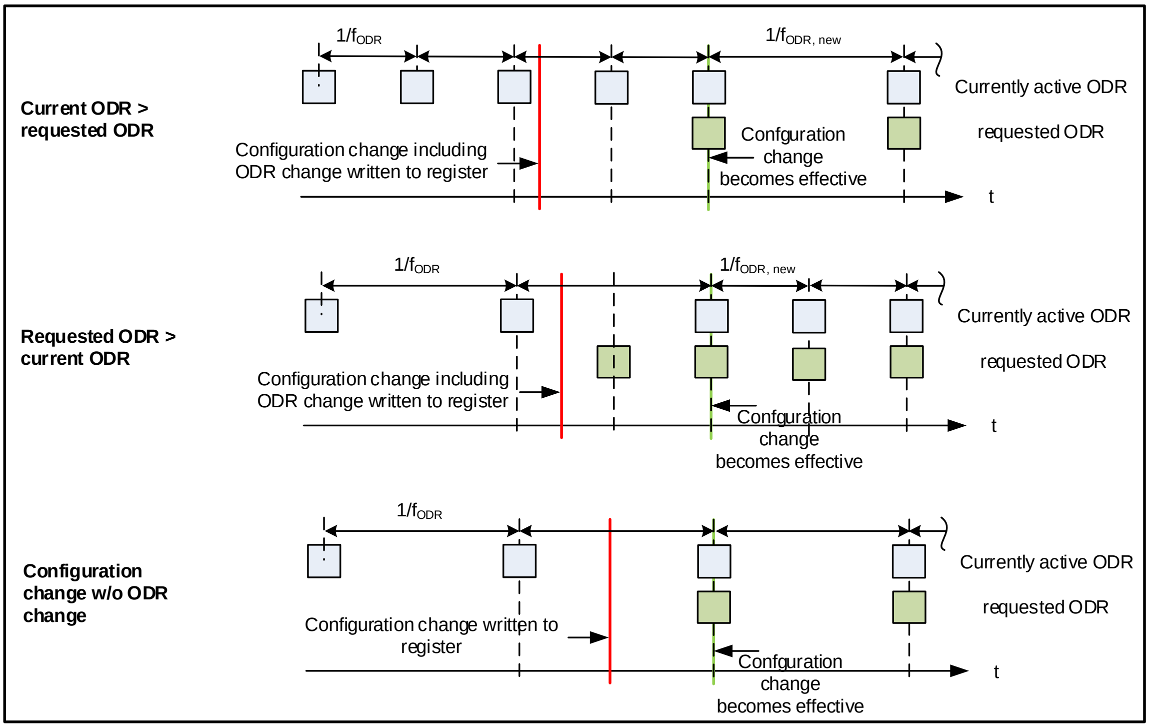
Bit *m* in sensor_time	7	6	5	4	3	2	1	0
Resolution [ms]	5	2.5	1.250	0.625	0.3125	0.156	0.078	0.039
Update rate [Hz]	200	400	800	1600	3200			

The sensortime is synchronized with the data capturing in the data register and the FIFO. The sensortime supports multiple seconds of sample counting and a sub-microsecond resolution, see Register SENSORTIME_0 for details.

Burst reads on the registers SENSORTIME_0 to SENSORTIME_2 always deliver consistent values, i.e. the value of the register does not change during the burst read.

Configuration Changes

If device configuration settings in registers ACC_CONF, ACC_RANGE, GYR_CONF, GYR_RANGE, or AUX_CONF are changed while the sensors are enabled (accelerometer PWR_CTRL.acc_en = 0b1, gyroscope PWR_CTRL.gyr_en or auxiliary sensor PWR_CTRL.aux_en = 0b1), the configuration changes are not immediately applied. The configuration changes become effective if a sampling event for the currently active ODR coincides with a sampling event for the newly requested ODR on the sensortime 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. See also following figure.



4.7. FIFO

The device supports the following FIFO operating modes:

- Streaming mode: overwrites oldest data on FIFO full condition
- FIFO mode: discards newest data on FIFO full condition

The FIFO size is 2048 byte and supports the following interrupts:

- FIFO full interrupt
- FIFO watermark interrupt

FIFO is enabled for accelerometer data with FIFO_CONFIG_1.fifo_acc_en=0b1, for gyroscope data with FIFO_CONFIG_1.fifo_gyr_en=0b1, and auxiliary interface (e.g. magnetometer) data with FIFO_CONFIG_1.fifo_aux_en=0b1 (0b0=disabled).

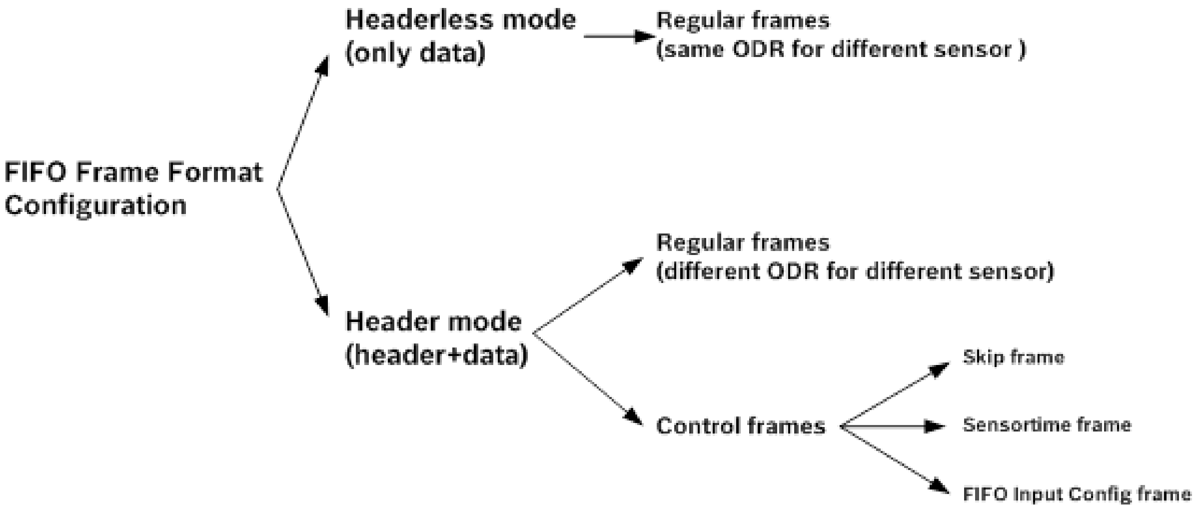
The FIFO may be used in all power modes of the device.

Frames

The FIFO captures data in frames, which consist in header mode of a header and a payload data, in headerless mode only payload is stored.

In header mode (standard configuration) each regular frame consists of a one byte header describing properties of the frame (e.g. which sensors are included in this frame) and the payload data itself. Beside the regular frames, which contain the sensor data, there are control frames, which contain metadata (e.g. sensortime).

An overview of the possible frame types is show below



Header mode

The header has a fixed length of 8 bit and the following format:

Bit	7	6	5	4	3	2	1	0
Content	fh_mode<1:0>		fh_parm<3:0>				fh_ext<1:0>	

These *fh_mode* and *fh_parm* and *fh_ext* fields are defined below

fh_mode<1:0>	Definition	fh_parm <3:0>	fh_ext<1:0>
0b10	Regular frame	Enabled sensors	Tag of INT2 and INT1
0b01	Control frame	Control opcode	
0b00 and 0b11	Reserved	N/A	

fh_parm=0b0000 is invalid for regular mode, a header of 0x80 indicates an uninitialized frame, which is reported if the fifo read operations reads more data, than contained in the fifo. An uninitialized frame contains one byte of payload 0x00.

In a regular frame, fh_parm parameter defines which sensors are included in the data part of the frame. The format is

Name	fh_parm<3:0>			
Bit	3	2	1	0
Content	Reserved	FIFO_aux_data	FIFO_gyr_data	FIFO_acc_data

When FIFO_<sensor x>_data is 0b1 (0b0) data for sensor x is included (not included) in the data part of the frame.

The fh_ext<1:0> field are used for external tagging.

The order of the data in the FIFO data frame (see following table) differs from the order defined for the Registers DATA_0 to DATA_19.

A valid regular frame, contains data of at least one sensor (accelerometer, gyroscope, or auxillary sensor). Only valid frames will be written into the FIFO. E.g. fh_parm=0b0111 in the header of a frame will result in the data layout shown below.

DATA[X]	Acronym	
X=0	AUX_0	copy of register Val(AUX_RD_ADDR) in auxiliary sensor register map
X=1	AUX_1	copy of register Val(AUX_RD_ADDR)+1 in auxiliary sensor register map
X=2	AUX_2	copy of register Val(AUX_RD_ADDR)+2 in auxiliary sensor register map
X=3	AUX_3	copy of register Val(AUX_RD_ADDR)+3 in auxiliary sensor register map
X=4	AUX_4	copy of register Val(AUX_RD_ADDR)+4 in auxiliary sensor register map
X=5	AUX_5	copy of register Val(AUX_RD_ADDR)+5 in auxiliary sensor register map
X=6	AUX_6	copy of register Val(AUX_RD_ADDR)+6 in auxiliary sensor register map
X=7	AUX_7	copy of register Val(AUX_RD_ADDR)+7 in auxiliary sensor register map
X=8	GYR_X<7:0> (LSB)	
X=9	GYR_X<15:8> (MSB)	
X=10	GYR_Y<7:0> (LSB)	
X=11	GYR_Y<15:8> (MSB)	
X=12	GYR_Z<7:0> (LSB)	
X=13	GYR_Z<15:8> (MSB)	
X=14	ACC_X<7:0> (LSB)	
X=15	ACC_X<15:8> (MSB)	
X=16	ACC_Y<7:0> (LSB)	
X=17	ACC_Y<15:8> (MSB)	
X=18	ACC_Z<7:0> (LSB)	
X=19	ACC_Z<15:8> (MSB)	

The length of the auxillary sensor data block in a FIFO frame depends on the configued burst read length of the auxillary interface in Register AUX_IF_CONF.aux_rd_burst:
If the read burst length for the auxillary sensor is configured to less than 8 byte, the number of data bytes in the regular FIFO frame is reduced accordingly. I.e. in the above example, the gyro data would start before Byte 8.

Control frames

Control frames are only supported in header mode. There are a number of control frames defined through the *fh_parm* field. These are shown in below.
A skip frame indicates the number of skipped frames after a FIFO overrun occurred. A sensortime frame contains the sensortime when the last sampled frame stored in the FIFO is read. A FIFO input config frames indicates a change in sensor configuration which affects the sensor data.
The FIFO fill level is contained in registers FIFO_LENGTH_1.fifo_byte_counter_13_8 and FIFO_LENGTH_0.fifo_byte_counter_7_0. The fifo fill level includes the space needed for the regular and the control frames, with the exception of the sensortime frame.

fh_mode<3:0>	Definition	Number of
0x0	Skip Frame	1 byte payload
0x1	Sensortime Frame	3 bytes payload
0x2	Fifo_Input_Config Frame	4 bytes payload
0x3 – 0x7	Reserved	

1.1.1.1.1.1 Skip Frame (fh_parm=0x0):

In the case of FIFO overflows, a skip_frame is prepended to the FIFO content, when read out next time. The data for the frame consists of one byte and contains the number of skipped frames. When more than 0xFF frames have been skipped, 0xFF is returned. A skip frame is expected always as first frame in a FIFO read burst. A skip frame does not consume memory in the FIFO.

1.1.1.1.1.2 Sensortime Frame (fh_parm=0x1):

The data for the sensortime frame is a copy of the Register SENSORTIME_0 to SENSORTIME_2 when the last byte of the last sample frame was read. One sensortime frame is always expected as last frame in the FIFO. A sensortime frame is only sent if the FIFO becomes empty during the burst read. A sensortime frame does not consume memory in the FIFO. Sensortime frames are enabled (disabled) by setting FIFO_CONFIG_0.fifo_time_en to 0b1 (0b0).

1.1.1.1.1.3 Fifo_Input_Config Frame (fh_parm=0x2):

Whenever the filter configuration of the FIFO input data sources changes, a FIFO input config frame is inserted into the FIFO, before the configuration change becomes active. E.g. when the bandwidth for the accelerometer filter is changed in Register ACC_CONF, a FIFO input config frame is inserted before the first frame with accelerometer data with the new bandwidth configuration. The FIFO input config frame contains four byte of data with the format

Bit	7	6	5	4	3	2	1	0
Byte 0	reserved	reserved	aux_ if_ch	aux_ conf_ch	gyr_ range_ch	gyr_ conf_ch	acc_ range_ch	acc_ conf_ch
Byte 1	Sensortime_0 for next frame (may be drop frame)							
Byte 2	Sensortime_1 for next frame (may be drop frame)							
Byte 3	Sensortime_2 for next frame (may be drop frame)							

aux_if_ch	**A write to Register AUX_IF_CONF, AUX_RD_ADDR, or AUX_WR_ADDR becomes active.**
aux_conf_ch	A write to Register AUX_CONF becomes active.
gyr_range_ch	A write to Register GYR_RANGE becomes active.
gyr_conf_ch	A write to Register GYR_CONF or gyr_FIFO_filt_data or gyr_FIFO_downsampling in Register FIFO_DOWNS becomes active.
acc_range_ch	A write to Register ACC_RANGE becomes active.
acc_conf_ch	A write to Register ACC_CONF or acc_FIFO_filt_data or acc_FIFO_downsampling in Register FIFO_DOWNS becomes active.

Headerless mode

When the data rates of all enabled sensor elements are identical, the FIFO header may be disabled in FIFO_CONFIG_1.fifo_header_en.

The headerless mode supports only regular frames. To be able to distinguish frames from each other, all frames must have the same size. For this reason, any change in configuration that have an impact to frame size or order of data within a frame will cause an instant flush of FIFO, restarting capturing of data with the new settings.

If the auxiliary sensor interface is enabled, the number of auxiliary sensor bytes in a FIFO frame is always AUX_IF_CONF.aux_rd_burst bytes (see section 4.10). If the burst length is less than 8, the device will pad the values read form the auxiliary sensor. E.g. if AUX_IF_CONF.aux_rd_burst=0b01 (2 Bytes), a frame with auxiliary sensor, accelerometer, and gyroscope data will look like

DATA[X]	Acronym	
X=0	AUX_0	copy of register Val(AUX_RD_ADDR.read_addr) in auxiliary sensor register map
X=1	AUX_1	copy of register Val(AUX_RD_ADDR.read_addr+1) in auxiliary sensor register map
X=2	Padding byte	Undefined value
X=3	Padding byte	Undefined value
X=4	Padding byte	Undefined value
X=5	Padding byte	Undefined value
X=6	Padding byte	Undefined value
X=7	Padding byte	Undefined value
X=8	GYR_X<7:0> (LSB)	
X=9	GYR_X<15:8> (MSB)	
X=10	GYR_Y<7:0> (LSB)	
X=11	GYR_Y<15:8> (MSB)	
X=12	GYR_Z<7:0> (LSB)	
X=13	GYR_Z<15:8> (MSB)	
X=14	ACC_X<7:0> (LSB)	
X=15	ACC_X<15:8> (MSB)	
X=16	ACC_Y<7:0> (LSB)	
X=17	ACC_Y<15:8> (MSB)	
X=18	ACC_Z<7:0> (LSB)	
X=19	ACC_Z<15:8> (MSB)	

Conditions and Details

FIFO frame reads

If a frame is fully read through the Register FIFO_DATA, it gets deleted from the FIFO of the device. If a frame is only partially read it will be repeated completely with the next access both in headerless and in header mode. In headermode, this includes the header. In the case of a FIFO overflow between the first partial read and the second read attempt, the frame is kept only if FIFO_CONFIG_0.fifo_stop_on_full *=0b1*.

FIFO overreads

When more data are read from the FIFO than it contains valid data, 0x8000 is returned in headerless mode. In header mode 0x80 indicates an invalid frame.

Frame rates

The frame sampling rate of the FIFO is defined by the maximum output data rate of the sensors enabled for FIFO sampling. The FIFO sampling configuration is set in register FIFO_CONFIG_0 to FIFO_CONFIG_1. It is possible to select filtered or pre-filtered data as an input to the FIFO. If pre-filtered data is selected in register FIFO_DOWNS.acc_fifo_filt_data for the accelerometer, the sample rate is 1600 Hz. If pre-filtered data is selected in register FIFO_DOWNS.gyr_fifo_filt_data for the gyroscope, the sample rate is 6400 Hz. The input data rate to the FIFO can be reduced by selecting a down-sampling factor 2^k in registers FIFO_DOWNS.acc_fifo_downs or FIFO_DOWNS.gyr_fifo_downs where k={0..7}.

FIFO overflow

In the case of an overflow the FIFO can either stop recording data or overwrite the oldest data. The behavior is controlled by Register FIFO_CONFIG_0.fifo_stop_on_full. If FIFO_CONFIG_0.fifo_stop_on_full *=0b0,* the FIFO logic may delete the oldest frames. If header mode is enabled and the free FIFO space falls below the maximum size frame, the skip frame is the prepended at the next FIFO readout.

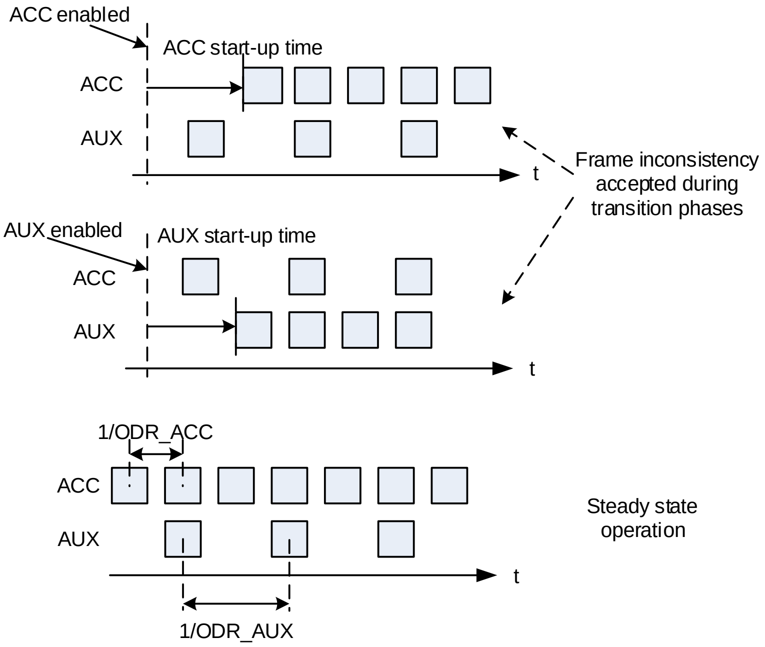
If FIFO_CONFIG_0.fifo_stop_on_full =0b1, the newest frame may be discarded, if the free FIFO space falls below the maximum size frame. If header mode is enabled, a skip frame is prepended at the next FIFO readout (which is **not** the position where the frame(s) have been discarded).

During a FIFO read operation of the host, no data at the FIFO tail may be dropped. If the host reads the FIFO with a slower rate than it is filled, it may happen that the sensor needs to drop new data, even when FIFO_CONFIG_0.fifo_stop_on_full =0b0. These events are recorded in the Register ERR_REG.fifo_err.

FIFO data synchronization

All sensor data are sampled with respect to a common ODR time grid. Even if a different ODR is selected for the acceleration and the auxillary sensor the data remains synchronized:
If a frame contains a sample from a sensor element with ODR x, then it must contain also samples of all sensor elements with an ODR y>=x. This applies for steady state operation. In transition phases, it is more important not to lose data, therefore exceptions are possible if the sensor elements with ODR y>=x do not have data, e.g. due to a sensor configuration change.

FIFO Data Synchronization Scheme in the following figure illustrates the steady state and transient operating conditions.



FIFO synchronization with external interrupts

External interrupts may be synchronized into the FIFO data. For this operation mode the FIFO_CONFIG_1.fifo_tag_int1_en and FIFO_CONFIG_1.fifo_tag_int2_en need to be enabled, as well as INT1_IO_CTRL.input_en and INT2_IO_CTRL.input_en. The fh_ext field in FIFO header will then be set according to the signal at the INT1/INT2 inputs. The minium active level of the external signal is 10 ns.

FIFO Interrupts

The FIFO supports two interrupts, a FIFO full interrupt and a watermark interrupt:

- The FIFO 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.
- The FIFO watermark is issued when the FIFO fill level is equal or above a watermark defined in Register FIFO_WTM_1.fifo_water_mark_12_8.

In order to enable/use the FIFO full or watermark interrupts, map them on the desired interrupt pin via INT_MAP_DATA.

Latched FIFO interrupts will only be cleared, if the status register gets read and the fill level is below the corresponding FIFO interrupt (full or watermark).

FIFO Reset

The user can trigger a FIFO reset by writing the command *fifo_flush* (0xB0) in CMD.
Automatic resets are only performed in the following cases:

- A sensor is enabled or disabled in headerless mode
- A transition between headerless and headermode or vice versa has occurred.
- Size of auxiliary sensor data in a frame changed in header or headerless mode

FIFO in Low Power Mode

In the low power mode the device supports FIFO usage. The data storage into the FIFO is identical to the normal and performance mode, for the readout the description below applies.

- If PWR_CONF.fifo_self_wakeup=0b0 the advanced power save configuration needs to be disabled (PWR_CONF.adv_power_save=0b0) before reading out FIFO data.
- If PWR_CONF.fifo_self_wakeup=0b1 and the FIFO watermark or FIFO full interrupt is triggered, the restrition for PWR_CONF.adv_power_save=0b1 (see Section 4.5) do not apply as long as a single burst read on Register FIFO_DATA completes. This may be used to read the complete FIFO with one single burst read without leaving low power mode. Without a FIFO watermark interrupt or full interrupt, the advanced power save configuration needs to be disabled (PWR_CONF.adv_power_save=0b0) before reading out FIFO data.

4.8. Advanced Features

Global Configuration

The configuration of the interrupt feature engine is described in the Registers FEATURES. These registers are partitioned into several pages, the page valid for the next read or write to the Registers FEATURES is selected by the Register FEAT_PAGE.page. Writes to a FEATURES register must be 16-bit word oriented, i.e. writes should start at an even address (2m) and the last byte written should be at an odd address (2n+1), where 0x30<=2m<=2n<0x3F. If the write start address is less than 0x30 the write may start at any address (see example 4 below), if the end address is greater than 0x3F, it may stop at any address (see example 5 below).

- For register writes which stop at an even SPI address (2n), the data at the odd SPI address (2n+1) are undefined (see Example 2, 3 below)
- For writes which start at an odd SPI address (2m+1), the data at the even address (2m) are undefined. (see Example 3 below)

Ex. 1) Write 4 bytes starting at address 0x30

0x30	**Valid Data**
0x31	Valid Data
0x32	Valid Data
0x33	Valid Data

Ex. 2) Write 3 bytes starting at address 0x30

0x30	**Valid Data**
0x31	Valid Data
0x32	Valid Data
0x33	Undefined

Ex. 3) Write 2 bytes starting at address 0x31

0x30	**Undefined**
0x31	Valid Data
0x32	Valid Data
0x33	Undefined

Ex. 4) Write 9 bytes starting at address 0x29

0x29	**Valid Data**
0x2A	Valid Data
⋮	⋮
0x2E	Valid Data
0x2F	Valid Data
0x30	Valid Data
0x31	Valid Data

Ex. 5) Write 5 bytes starting at address 0x3E

0x3E	**Valid Data**
0x3F	Valid Data
0x40	Valid Data
0x41	Valid Data
0x42	Valid Data
⋮	⋮

Make sure the sensor is initialized properly before the feature configuration is performed (see description in section 4.4.)

Some features generate interrupts. INT1_MAP_FEAT and INT2_MAP_FEAT configure these features. INT_STATUS_0 reports the interrupt source.

In order to minimize the power consumption or to enable always-on motion sensing, all advanced features (algorithms) rely on accelerometer data samples.

Minimum Bandwidth Settings

If the filter performance of the accelerometer is configured to high performance (ACC_CONF.acc_filter_perf is 0b1), the features operate at highest performance independent of the ODR and the bandwidth set by the host.

If the filter performance of the accelerometer is configured to ultra low power (ACC_CONF.acc_filter_perf is 0b0), the feature performance is depending on the ODR and the bandwidth set by the host:

1. The ODR must be set to minimum 50 Hz

If the device configuration does not meet the minimum requirements, the corresponding flag in the Register INTERNAL_STATUS *is set, if one of the advanced features is enabled. In this case the features are still evaluated, the same number of samples are evaluated, but they are sampled at tha lower rate.*

Error Interrupts

The device supports an error interrupt, which triggers if the device cannot be recovered without a softrest or a POR. This error interrupt is enabled through INT_MAP_DATA. The interrupt status is available in INT_STATUS_1.err_int. After restarting the device a device reinitialization must be done.

Axis remapping for interrupt features

If the coordinate system of the end device differs from the sensor coordinate system described in Section 8.2 the sensor axis must be remapped to use the orientation dependent features (e.g. orientation interrupt, flat interrupt) properly.

Axis remapping register allows the host to freely map individual axis to the coordinate system of the used platform. Individual axis can be mapped to any other defined axis. The sign value of the axis can be also configured. For example x axis can be mapped to –x axis, +y axis, - y axis, +z axis or –z axis. Similarly, other axes also have their own combinations.

Invalid remappings are signaled through the register INTERNAL_STATUS.axes_remap_error if an advanced feature is enabled.

Note:
The axis remapping applies only to the data fetched into the features. The DATA_0 to DATA_13 registers and FIFO are not affected and should be remapped accordingly on the driver level.

Configuration settings:

1. GEN_SET_1.map_x_axis – describes which axis shall be mapped to x axis.
2. GEN_SET_1.map_x_axis_sign – describes whether the mapped axis shall be inverted or not to be inverted.
3. GEN_SET_1.map_y_axis – describes which axis shall be mapped to y axis.
4. GEN_SET_1.map_y_axis_sign – describes whether the mapped axis shall be inverted or not to be inverted.
5. GEN_SET_1.map_z_axis – describes which axis shall be mapped to z axis.
6. GEN_SET_1.map_z_axis_sign – describes whether the mapped axis shall be inverted or not to be inverted.

Anymotion Detection

The anymotion detection uses the slope between two acceleration signals to detect changes in motion. The interrupt is configured by setting enable flag ANYMO_2.enable along with at least one of the following flags: ANYMO_1.select_x, ANYMO_1.select_y and ANYMO_1.select_z respectively for each axis.

It generates an interrupt when the absolute value of the slope (the difference between two accelerations) exceeds the preset ANYMO_2.threshold for a certain number of consecutive data points ANYMO_1.duration.

The slope (difference) is being computed between the current acceleration sample and the reference sample. The reference sample is updated while the anymotion is detected; basically this means the reference is the last state when sensor detected Anymotion.

The interrupt generated will be reset as soon as the slope value falls below the threshold.

Configuration settings

1. ANYMO_2.enable – enable the feature.
2. ANYMO_1.duration – the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion.
3. ANYMO_2.threshold – the slope threshold.
4. ANYMO_1.select_x – select the feature for x axis
5. ANYMO_1.select_y – select the feature for y axis
6. ANYMO_1.select_z – select the feature for z axis

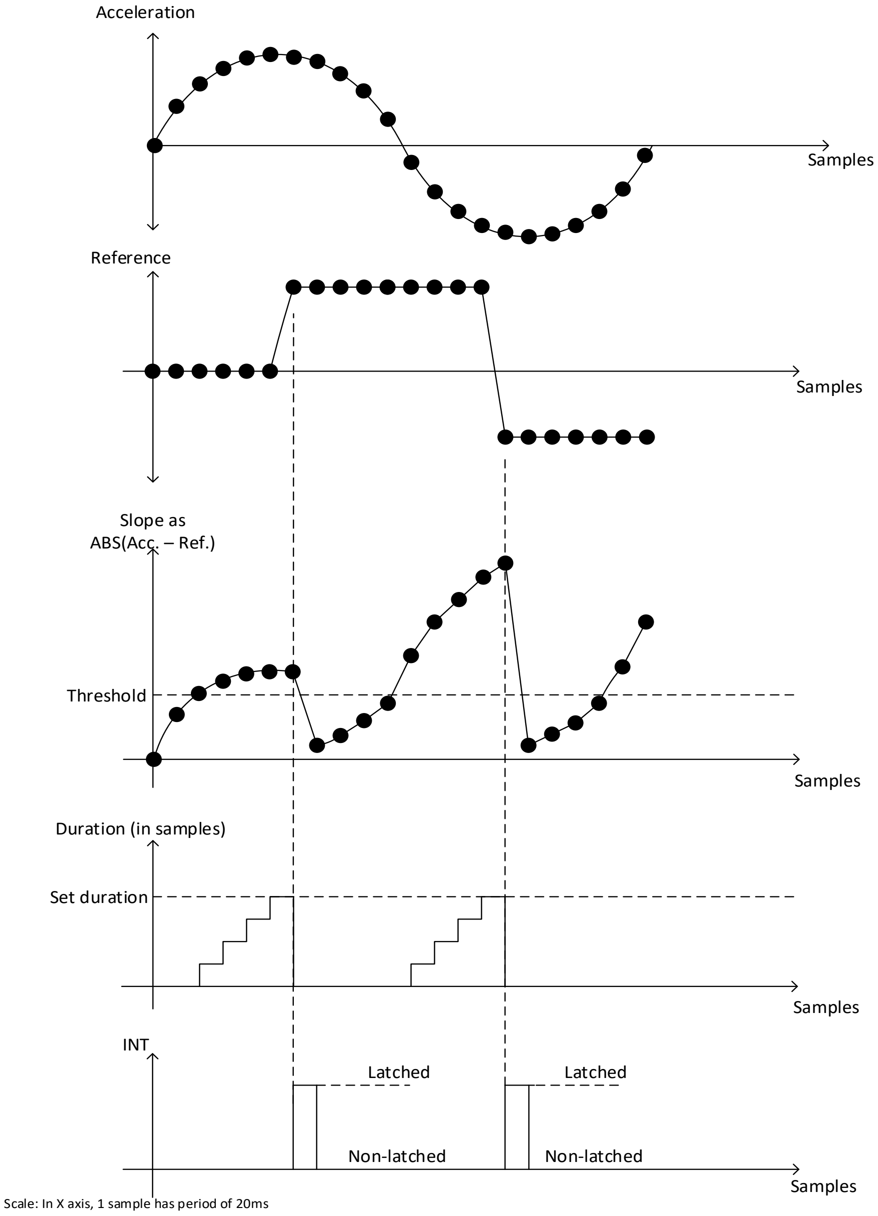


Figure 1: Any-motion detection

Nomotion Detection

The interrupt is configured by setting enable flag NOMO_2.enable along with at least one of the following flags: NOMO_1.select_x, NOMO_1.select_y, and NOMO_1.select_z respectively for each axis.

Nomotion Detection interrupt is generated when the slope on all selected axis remains smaller than a programmable NOMO_2.threshold for a programmable time. The signals and timings relevant to the nomotion interrupt functionality are depicted in the figure below.

Register NOMO_1.duration defines the number of consecutive slope data points of the selected axis which must exceed the threshold for an interrupt to be asserted.

Configuration settings

1. NOMO_2.enable – enable the feature.
2. NOMO_1.duration – the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion.
3. NOMO_2.threshold – the slope threshold.
4. NOMO_1.select_x – select the feature for x axis
5. NOMO_1.select_y – select the feature for y axis
6. NOMO_1.select_z – select the feature for z axis

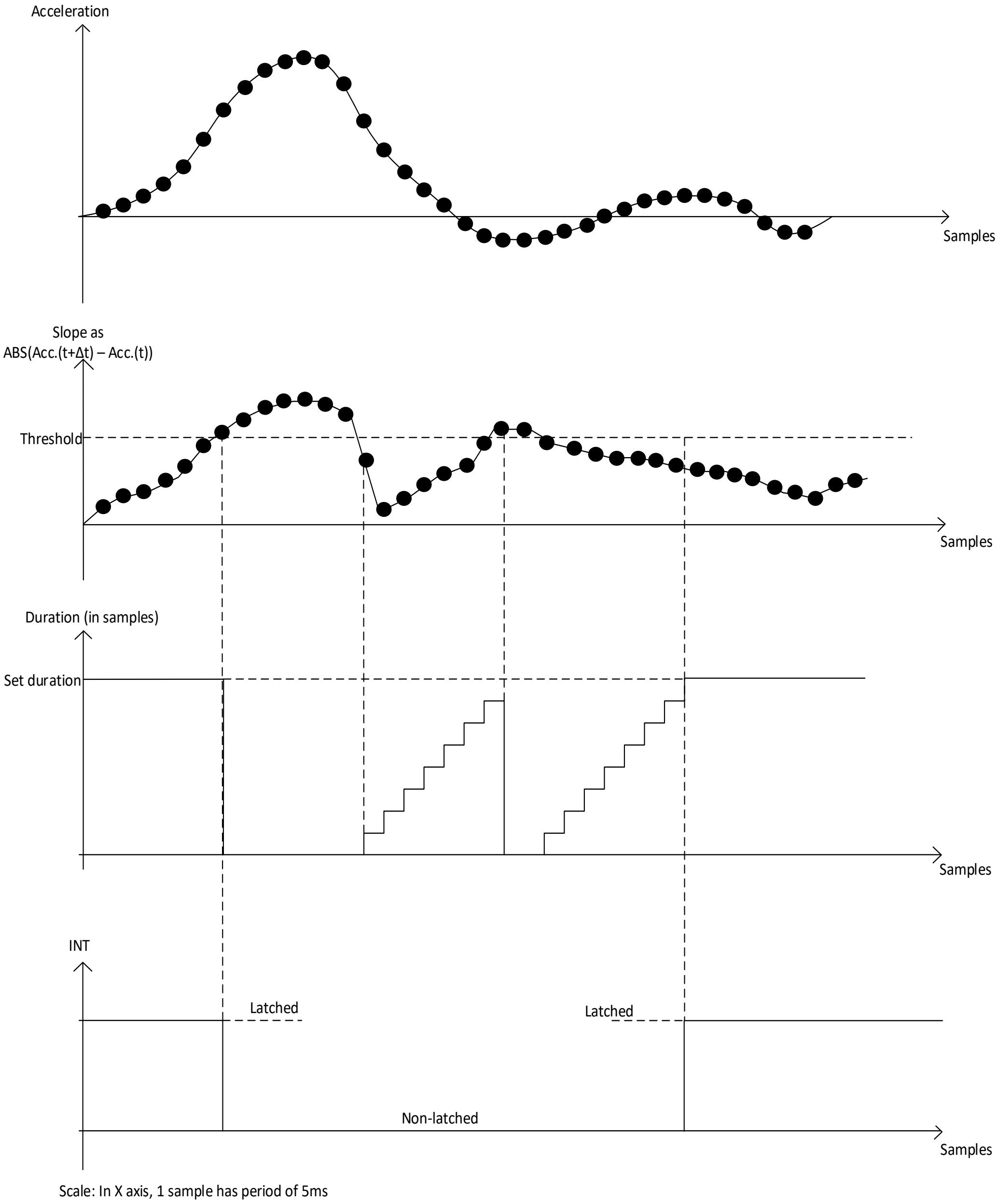


Figure 2: No-motion detection

Significant Motion Detection

The significant motion interrupt implements the interrupt required for motion detection in Android 4.3 and greater (https://source.android.com/devices/sensors/sensor-types.html#significant_motion)

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.

Configuration settings

1. SIGMO_2.enable – indicates if this feature is enabled or not.
2. SIGMO_1.block_size – Defines the duration after which the significant motion interrupt is triggered. It is expressed in 50 Hz samples (20 ms). Default value is 0xFA=5sec.

Activity and Activity Change Recognition

The device can detect simple user activities (unknown, still, walking, running) and can send an interrupt if those are changed, e.g. from walking to running or vice versus. The interrupt is shared with step detector/step counter watermark interrupts and can be configured independently of all other interrupts to any of the interrupt lines.

1. The device reports changes for following activity changes by an interrupt
    1. Still - 0
    2. Walking - 1
    3. Running -2
    4. Unknown – 3
2. Activity interrupt will be triggered only when there is change in status
3. ACT_OUT.act_out reports the activity status

During power on, activity will be unknown (0x03) and the device receives an activity change interrupt once activity is enabled, and a new activity detected. When activity is disabled, status will be changed to unknown.

Configuration settings

1. SC_26.en_activity indicates if the activity feature is enabled or not

Wrist Wear Wakeup

Wrist wear wakeup feature is designed to detect any natural way of user moving the hand to see the watch dial when wearing a classical wrist watch. The feature is intended to be used as wakeup gesture (i.e. for triggering screen-on or screen-off) in wrist wearable devices.
This feature has dependency on the device orientation in the user system. Implementation of the feature to detect gesture assumes that the sensor co-ordinate frame is aligned with the device/system co-ordinate frame. The assumed default device/system co-ordinate frame is depicted below. Please refer to section for details about axis remapping

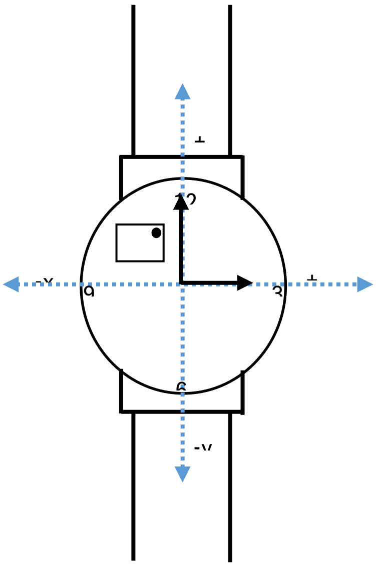


Figure 3: Device co-ordinate system assumed for gesture detection

The feature can distinguish if the device is in one of the following two positions:

- Focus position: In this position, the arm in-front of the body and the user should be able to comfortably look at the watch dial.
- Non-focus position: In this position, the user is not able to look at the watch dial.

WR_WAKEUP_3.min_angle_nonfocus and WR_WAKEUP_2.min_angle_focus can be used to adjust the angle change needed to detect a wrist wear wakeup gesture. WR_WAKEUP_4.max_tilt_lr, WR_WAKEUP_5.max_tilt_ll, WR_WAKEUP_6.max_tilt_pd and WR_WAKEUP_7.max_tilt_pu can be used to define the maximum tilt angle within which device will still remain in focus position.

Table 14: Positive use-case scenarios for the wrist wear wakeup gestures

Environment	Scenario	Device initial position	User movement
Outdoor / Indoor / train / bus	Walking	Arm swinging / hand in pocket	Lifts and brings the arm in-front of the body to be able to comfortably look at the watch dial
Outdoor / Indoor	Walking / Running / Jogging	Arm swinging	Lift and bring the arm in-front of the body to be able to comfortably look at the watch dial
Outdoor / Indoor / train / bus	Sitting / Standing	Arm is down on side of the body / hand in pocket	Lifts and brings the arm in-front of the body to be able to comfortably look at the watch dial
		Arm is in-front of the body	Rolls the wrist towards the user to look at the watch dial
Indoor / train	Working with computer	Arm on table or arm rest	Rolls the wrist towards the user to look at the watch dial

Configuration Settings

- WR_WAKEUP_1.enable - Enables the feature.
- WR_WAKEUP_2.min_angle_focus – Cosine of minimum expected attitude change of the device within 1 second time window when moving within focus position.
- WR_WAKEUP_3.min_angle_nonfocus – Cosine of minimum expected attitude change of the device within 1 second time window when moving from non-focus to focus position.
- WR_WAKEUP_4.max_tilt_lr – Sine of the maximum allowed downward tilt angle in landscape right direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device).
- WR_WAKEUP_5.max_tilt_ll – Sine of the maximum allowed downward tilt angle in landscape left direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device).
- WR_WAKEUP_6.max_tilt_pd – Sine of the maximum allowed backward tilt angle in portrait down direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device).
- WR_WAKEUP_7.max_tilt_pu – Sine of the maximum allowed forward tilt angle in portrait up direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device).

Wrist Wear Navigation Gesture Detector

BMI270 is designed for Wear OS by Google™[6] and features wrist gestures such as flick in/out, push arm down/pivot up, wrist jiggle/shake that help navigate the smartwatch. (https://support.google.com/wearos/answer/6312406?hl=en)

Flick in/out

Flick-in movement

Flick-out movement

Figure 4: Flick-in/out movement

For flick-in detection, the user must slowly turn the wrist away from the body (i.e. roll-out shown with a light-grey arrow in) and then quickly bring it back (i.e. roll-in shown with a darker-black arrow in to its original position.
For flick-out detection, the user must quickly turn the wrist away from the body (i.e. roll-out shown with a darker-black arrow in above picture) and then slowly bring it back (i.e. roll-in shown with a light-grey arrow in above picture) to its original position.

The speed of the roll-out and roll-in movements determine if the user performed a flick-in or a flick-out movement. WR_GEST_3.min_flick_samples can be used to control the time difference between the roll-in and roll-out movement. WR_GEST_2.min_flick_peak can be used to adjust the amount to tilt needed on the device to detect a flick gesture.

---

[6] Wear OS by Google is a trademark of Google LLC

Push arm down/Pivot up

Push-arm-down movement

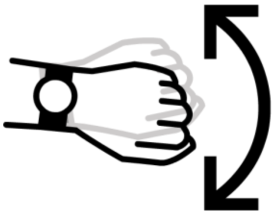
Pivot-up movement

Figure 5: Push arm-down/pivot up

For push-arm down detection, the user should hold the arm in front of the body and quickly push down and then bring it back normally to the original position.

For pivot-up detection, the user should hold the arm in front of the body and quickly pivot up and then bring it back normally to the original position.

Wrist Jiggle / Shake



Shake or jiggle movement

Figure 6: Wrist Jiggle

For a jiggle detection, the user must shake the hand quickly.

The device will detect the above mentioned gestures only when the user completes the movement within the duration defined by WR_GEST_4.max_duration.

Once feature is disabled, output will hold the previous value.
This feature has dependency on the device orientation in the user system. Please refer to section for details about axis remapping

Configuration settings

- WR_GEST_1.enable - Enables the feature.
- WR_GEST_1.wearable_arm – Configures the device in left (0) or right (1) arm. By default, the wearable device is assumed to be in left arm i.e. default value is 0."
- WR_GEST_2.min_flick_peak – Sine of the minimum tilt angle in portrait down direction of the device when wrist is rolled away (roll-out) from user.
- WR_GEST_3.min_flick_samples – Value of minimum time difference between wrist's roll-out and roll-in movement during flick gesture.
- WR_GEST_4.max_duration – Maximum time within which gesture movement has to be completed.

Output details

- WR_GEST_OUT.wr_gest_out – 3-bits indicate type of gestures detected:

Gestures	Value
No gesture	0
Push arm down	1
Pivot up	2
Wrist shake/jiggle	3
Flick in	4
Flick out	5

Step counter / detector (Wrist-worn)

The wrist worn step counter/detector in BMI270 is optimized for wearable applications including smartwatches/bands/fitness trackers, among others. The stepcounter algorithm is optimized for high accuracy in wrist use-case applications, while Step Detector is optimized for low latency.

Table 15: Step counter Configuration

Configuration	
Parameters	**Wrist**
SC_1.param_1	301
SC_2.param_2	31700
SC_3.param_3	315
SC_4.param_4	31451
SC_5.param_5 (STEP_BUFFER_SIZE)	4
SC_6.param_6	31551
SC_7.param_7	27853
SC_8.param_8	1219
SC_9.param_9	2437
SC_10.param_10	1219
SC_11.param_11	-6420
SC_12.param_12	17932
SC_13.param_13	1
SC_14.param_14	39
SC_15.param_15	25
SC_16.param_16	150
SC_17.param_17	160
SC_18.param_18	1
SC_19.param_19	12
SC_20.param_20	15600
SC_21.param_21	256
SC_22.param_22	1
SC_23.param_23	3
SC_24.param_24	1
SC_25.param_25	14

1. SC_26.watermark_level – watermark level; the step counter will trigger output every time specific number of steps are counted
2. SC_26.reset_counter – flag to reset the counted steps. Step count value can be reset only when any one of features mentioned in this register is enabled.
3. SC_26.en_counter – indicates if the Step Counter feature is enabled or not.
4. SC_26.en_detector – indicates if the Step Detector feature is enabled or not.
5. SC_26.en_activity – indicates if the activity feature is enabled or not
6. SC_1.param_1 to SC_25.param_25 – there are 25 parameters, which can customize the sensitivity of the Step Counter and Detector.

4.9. General Interrupt Pin Configuration

Electrical Interrupt Pin Behavior

Both interrupt pins PIN1 and PIN2 can be configured to show the desired electrical behavior. Interrupt pins can be enabled in INT1_IO_CTRL.output_en respectively INT2_IO_CTRL.output_en. The characteristic of the output driver of the interrupt pins may be configured with bits INT1_IO_CTRL.od and INT2_IO_CTRL.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 electrical behavior of the Interrupt pins, whenever an interrupt is triggered, can be configured as either "active-high" or "active-low" via INT1_IO_CTRL.lvl respectively INT2_IO_CTRL.lvl.

Both interrupt pins can be configured as input pins via INT1_IO_CTRL.input_en respectively INT2_IO_CTRL.input_en. This is necessary when FIFO tag feature is used (see Section 4.7 "FIFO synchronization with external interrupts"). If both are enabled, the input (e.g. marking FIFO) is driven by the interrupt output.
The device supports edge and level triggered interrupt inputs, this can be configured through FIFO_CONFIG1.fifo_tag_int1_en and FIFO_CONFIG1.fifo_tag_int2_en.

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_LATCH.int_latch.

In latched mode an asserted interrupt status in INT_STATUS_0 (advanced feature interrupts) or INT_STATUS_1 (data ready, FIFO and error interrupts) and the selected pin are reset if the corresponding status register is read. If the interrupt activation condition still holds when the interrupt is reset, the interrupt status and pin are asserted again. If more than one interrupt pin is used in latched mode, all interrupts in INT_STATUS_0 should be mapped to one interrupt pin and all interrupts in INT_STATUS_1 should be mapped to the other interrupt pin. If just one interrupt pin is used all interrupts may be mapped to this interrupt pin.

In the non-latched mode the selected pin are reset as soon as the activation condition is not valid anymore. The interrupt status bits are active until read by the host.

Interrupt Pin Mapping

The data ready, FIFO watermark, FIFO full, error, and the advanced feature interrupts are mapped to the external INT1 or INT2 pins by setting the corresponding bits in the Registers INT_MAP_DATA, INT1_MAP_FEAT and INT2_MAP_FEAT. To unmap these interrupts, the corresponding bits must be reset.

Once an interrupt triggered the output pin, the host can derive the source of the interrupt of the corresponding status bit in the Register: INT_STATUS_0 and INT_STATUS_1.

4.10. Auxiliary Sensor Interface

The auxiliary interface allows to attach one auxiliary sensor (AUX, e.g. magnetometer) on the secondary interface of the device as shown in the figure below.

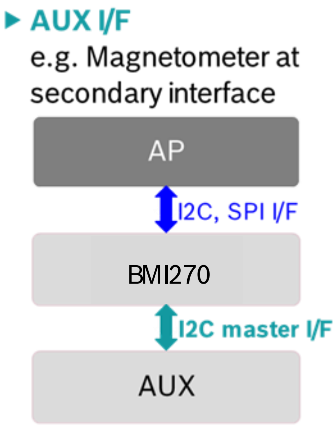


Figure 7: 9-DOF Solution w/ magnetometer (AUX) connected to the 2nd I/F

Structure and Concept

The device controls the data acquisition of the auxiliary sensor and presents the data to the application processor through the primary I2C or SPI interface. No other I2C master or slave devices must be attached to the auxiliary sensor interface.

The device autonomously reads the sensor data from a compatible auxiliary sensor without intervention of the application processor and stores the data in its data registers and FIFO. The initial setup of the auxiliary sensor after power-on is done through indirect addressing, see Section 4.10 for details.

The main benefits of the auxiliary sensor interface are

- Synchronization of sensor data of auxiliary sensor and accelerometer. This results in an improved sensor data fusion quality.
- Usage of the device FIFO for auxiliary sensor data (BMM150 does not have a FIFO). This is important for monitoring applications.

Interface Control

The auxiliary sensor functionality is supported only if an AUX sensor is connected according to Section 7.3 and the auxiliary interface is configured for the auxiliary sensor operation by PWR_CTRL.aux_en=0b1. If the auxiliary interface is not used for auxiliary sensor operation, then the auxiliary sensor interface must remain disabled by setting PWR_CTRL.aux_en=0b0 (default).

To change the power mode of the auxiliary sensor, both the power mode of the auxiliary interface and the auxiliary sensor part needs to be changed, e.g. to set the auxiliary sensor to suspend mode:

- The auxiliary sensor part itself must be put into suspend mode by writing the respective configuration bits of the auxiliary sensor part. The power mode of the auxiliary sensor part is controlled by setting the device auxiliary sensor interface into manual mode by AUX_IF_CONF.aux_manual_en=0b1 and then communicating with the auxiliary sensor part through the device registers AUX_RD_ADDR, AUX_WR_ADDR, and AUX_WR_DATA. For details see Section 4.10

- Set the auxiliary sensor interface to suspend in Register PWR_CTRL.aux_en=0b0. Changing the auxiliary sensor interface power mode to suspend does not imply any mode change in the auxiliary sensor.

Interface Configuration

The configuration registers that control the auxiliary sensor interface operation, are only affecting the interface to the auxiliary sensor, not the configuration of the sensor itself (this must be done in setup mode).

There are three basis configurations of the auxiliary sensor interface:

- No auxiliary sensor access
- Setup mode: Auxiliary sensor access in manual mode
- Data mode: Auxiliary sensor access through hardware readout loop.

The setup of the auxiliary sensor itself must be done through the primary interface using indirect addressing in setup mode. When collecting sensor data, the device autonomously triggers the measurement of the auxiliary sensor using the auxiliary sensor forced mode and the data readout from the auxiliary sensor (data mode).

In setup mode, the auxiliary sensor may be configured and trim data may be read out from the auxiliary sensor. In the data mode the auxiliary sensor data are continuously copied into the device's registers and may be read out from the device directly over the primary interface. For a BMM150 magnetometer, these are the auxiliary sensor data itself and Hall resistance, temperature is not required. The table below shows how to configure these three modes using the registers PWR_CONF, PWR_CTRL, and AUX_IF_CONF.aux_manual_en.

Mode	AUX_IF_CONF.aux_manual_en	PWR_CONF.adv_power_save	PWR_CTRL.aux_en
No auxiliary sensor access	1	1	0
Setup mode	1	0	0
Data mode	0	X	1

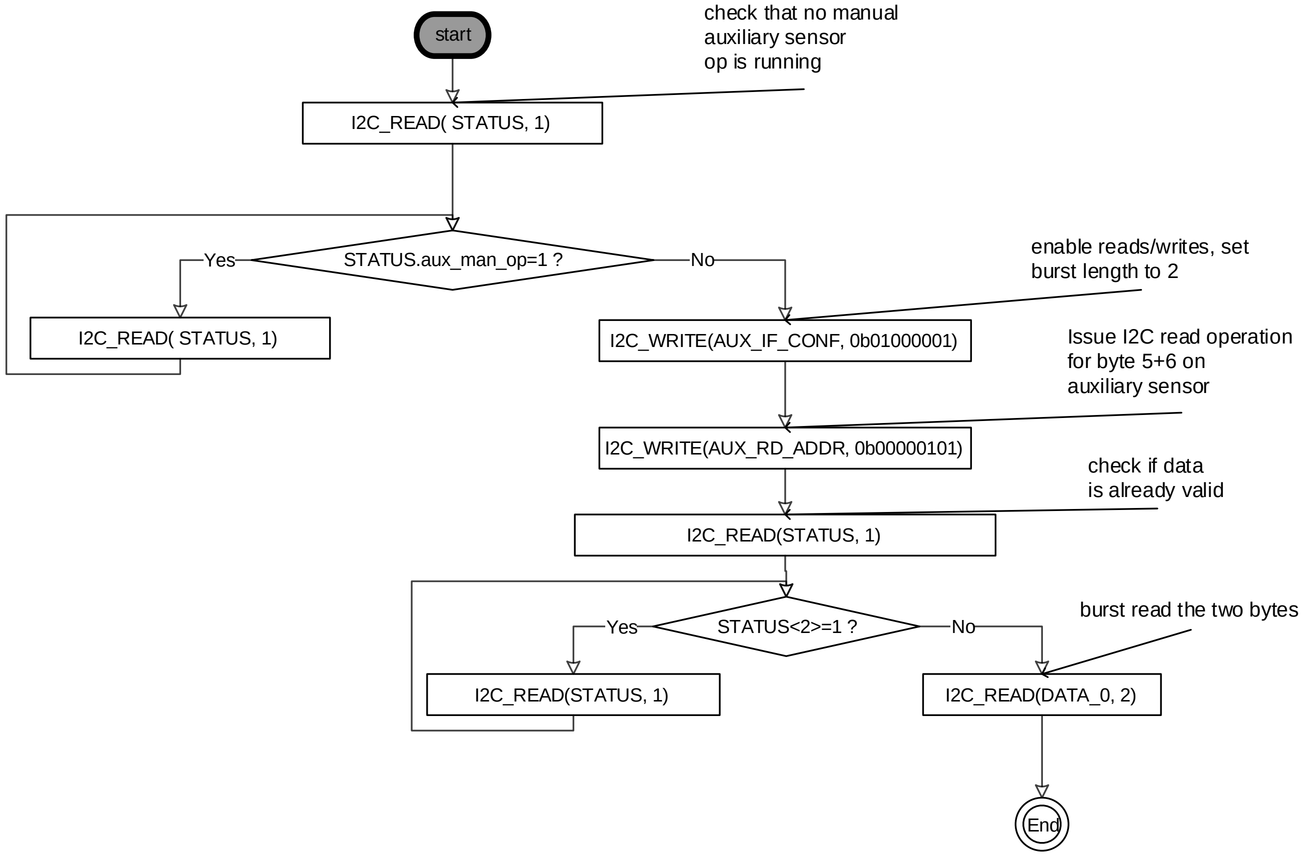
IF_CONF.aux_en enables (disables) the auxiliary sensor interface. The auxiliary sensor interface operates at 400 kHz. This results in an I2C readout delay of about 250 us for 10 bytes of data.

Setup mode (AUX_IF_CONF.aux_manual_en =0b1)

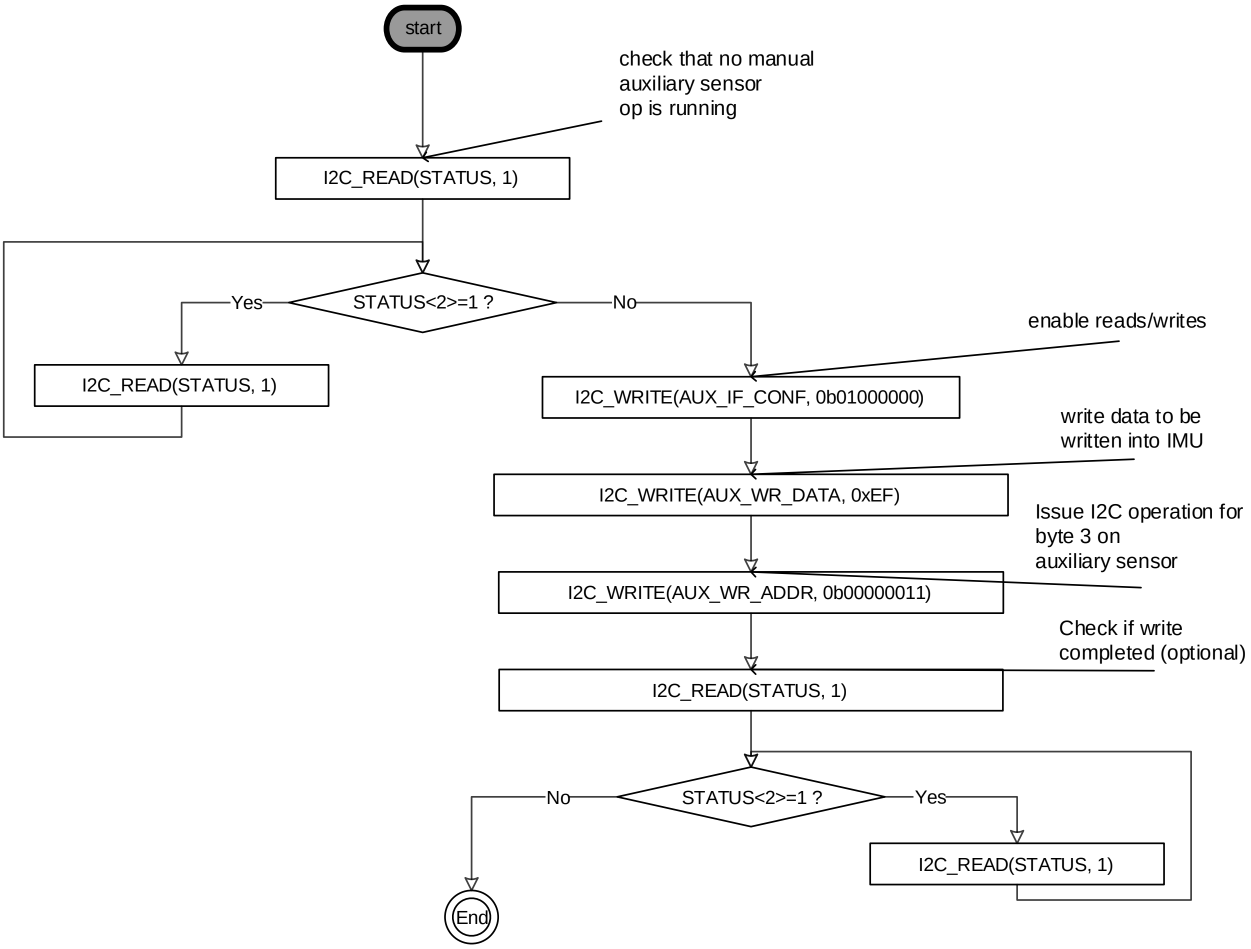
Through the primary interface the auxiliary sensor may be accessed using indirect addressing through the AUX_* registers. AUX_RD_ADDR and AUX_WR_ADDR define the address of the register to read/write in the auxiliary sensor register map and triggers the operation itself, when the auxiliary sensor interface is enabled through PWR_CTRL.aux_en.
For reads, the number of data bytes defined in AUX_IF_CONF.aux_rd_burst are read from the auxiliary sensor and written into the device Register DATA_0 to DATA_7. For writes only single bytes are written, independent of the settings in AUX_IF_CONF.aux_rd_burst. The data for the I2C write to auxiliary sensor must be stored in AUX_WR_DATA before the auxiliary sensor register address is written into AUX_WR_ADDR.
When a read or write operation is triggered by writing to AUX_RD_ADDR and AUX_WR_ADDR, STATUS.aux_busy is set and it is reset when the operation is completed. For reads the DATA_0 to DATA_7 contains the read data, for writes AUX_WR_DATA may be overwritten again.

Configuration phase of the auxiliary sensor.

Example: Read bytes 5 and 6 of auxiliary sensor



Example: Write 0xEF into byte 3 of auxiliary sensor



Data mode (AUX_IF_CONF.aux_manual_en=0)

AUX_RD_ADDR.read_addr defines the address of the data register from which to read the number of data bytes configured in AUX_IF_CONF.aux_rd_burst from AUX_0… AUX_7 data of the auxiliary sensor. These data are stored in the DATA_0 up to DATA_7 register. the device uses bit 0 of the DATA_6 register to determine the data ready status.

The data ready interrupt fires whenever a new data sample set from the AUX sensor is available in Registers DATA_0 to DATA_7. This allows a low latency data readout. In non-latched mode, the interrupt are cleared automatically after 1/(6400Hz). If this automatic clearance is unwanted, please use latched mode (see Section 4.9). The flag INT_STATUS_1.aux_drdy_int is cleared when the register INT_STATUS_1 is read. The flag STATUS.drdy_aux is cleared when the Registers DATA_0 to DATA_7 are read.

To enable the data ready interrupt please map it on the desired INT pin via INT_MAP_DATA.

AUX_WR_ADDR.write_addr defines the register address of auxiliary sensor to start a measurement in forced mode in the auxiliary sensor register map. During read and write operations STATUS.aux_busy is set and it is reset, when the operation is completed. The delay (time offset) between triggering an auxiliary sensor measurement and reading the measurement data is specified in AUX_CONF.aux_offset. Reading of the data is done in a single I2C read operation with a burst length specified in AUX_IF_CONF.aux_rd_burst. For BMM150 AUX_IF_CONF.aux_rd_burst should be set to 0b11, i.e. 8 bytes. If AUX_IF_CONF.aux_rd_burst is set to a value lower than 8 bytes, the remaining auxiliary sensor data in the Register DATA_0 to DATA_7 and the FIFO are undefined.

It is recommended to disable the auxiliary sensor interface (IF_CONF.aux_en=0b0) before setting up AUX_RD_ADDR.read_addr and AUX_WR_ADDR.write_addr for the data mode. This does not put the auxiliary sensor itself into suspend mode but avoids gathering unwanted data during this phase. Afterwards the auxiliary sensor interface can be enabled (IF_CONF.aux_en =0b1) again.

Delay (Time Offset)

The device supports starting the measurement of the sensor at the auxiliary sensor interface between 2.5 and 37.5 ms before the Register DATA are updated. This offset is defined in AUX_CONF.aux_offset. If set to 0b0, the measurement is done right after the last Register DATA update, therefore this measurement will be included in the next register DATA update.

4.11. OIS Interface

The device includes a secondary interface (see Section 6.6 for further details). This may be configured as a dedicated OIS interface. The OIS interface supports phone architectures which share a IMU for a regular host interface (HMI, activity recognition and gesture recognition, PDR) and for optical image stabilization (OIS) at the same time. The OIS interface is a second SPI slave interface, see Section 7.4 for detailed connection diagrams.

The OIS controller has access to low latency accelerometer and gyroscope data through the OIS interface. This is independent of the settings on the host interface e.g. any settings in the Registers ACC_CONF and GYR_CONF will not influence the OIS interface, it remains always in the minimum group delay configuration. With the exception of GYR_CONF.gyr_noise_perf which trades power and noise performance globally for both interfaces, i.e. noise may be reduced w/o compromising group delay.

The use-case for this data is to stabilize photo and video images by real time motion compensation of the camera lenses.

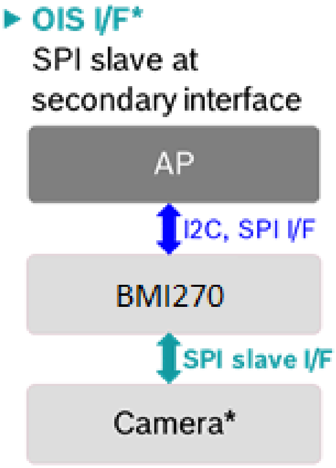


Figure 8: OIS interface

*) supported by the marked leading providers of OIS controllers.

By default, the OIS interface is in disabled state. If the system design requires an OIS interface, the host enables it by IF_CONF.ois_en=0b1. If the OIS interface is enabled the sensors may be controlled both throught the host controller and the OIS controller. The OIS controller has access to a dedicated OIS register map.

Table 16: OIS Register Map

OIS Register Map

read/write	read only	write only	reserved

Id: #ai260aa#flow#dig#reg-ipxact#BAI260_AddrMap_Shell.xml,v 1.14 2018-02-02 09:23:38+01 rio1rt Exp

Register Address	**Register Name**	**Default Value**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x7F			reserved							
...			reserved							
0x41			reserved							
0x40	OIS_CRTL	0x00	acc_en	gyr_en	reserved					
0x3F		0x00	reserved							
...		0x01	reserved							
0x18		0x00	reserved							
0x17	OIS_DATA_11	0x00	gyr_z_15_8							
0x16	OIS_DATA_10	0x00	gyr_z_7_0							
0x15	OIS_DATA_9	0x00	gyr_y_15_8							
0x14	OIS_DATA_8	0x00	gyr_y_7_0							
0x13	OIS_DATA_7	0x00	gyr_x_15_8							
0x12	OIS_DATA_6	0x00	gyr_x_7_0							
0x11	OIS_DATA_5	0x00	acc_z_15_8							
0x10	OIS_DATA_4	0x00	acc_z_7_0							
0x0F	OIS_DATA_3	0x00	acc_y_15_8							
0x0E	OIS_DATA_2	0x00	acc_y_7_0							
0x0D	OIS_DATA_1	0x00	acc_x_15_8							
0x0C	OIS_DATA_0	0x00	acc_x_7_0							
0x0B		0x00	reserved							
...		-	reserved							
0x00		0x27	reserved							

Register (0x0C..0x17) OIS_DATA_0..11

DESCRIPTION:These 12 registers publish accelerometer and gyroscope data. The register map layout is identical to the host register map (see Section **Error! Reference source not found.**, Registers HYPERLINK \l "DATA_8" DATA_8 .. DATA_19). The registers of both register maps are separate instances and thus their content is not a simple copy of each other. The OIS registers provide output data of a separate low latency datapath.

RESET: 0x00

DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0C..0x17		OIS_DATA_0..11		0x00	
	7...0		acc_x_7_0 to gyr_z_15_8	0x0	R

Register (0x40) OIS_CRTL

DESCRIPTION: controls the IMU through the OIS interface

RESET: 0x00

DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1E		OIS_CRTL		0x00	
	7	acc_en	Enables accelerometer via OIS interface in registers OIS_DATA_0 till OIS_DATA_5 with minimum group delay @ 1.6KHz ODR	0x0	RW
	6	gyr_en	Enables gyroscope via OIS interface in registers OIS_DATA_6 till OIS_DATA_11 with minimum group delay @ 6.4KHz ODR	0x0	RW

4.12. Sensor Self-Test

Accelerometer

Activating the self-test results in a static offset of the acceleration data. Any external acceleration or gravitational force applied to the sensor during active self-test will be observed in the output as a superposition of both acceleration and self-test signal.

The recommended self test procedure is as follows:

1. Enable accelerometer with register PWR_CTRL.acc_en=1b1.
2. Set ±8g range in register ACC_RANGE.acc_range
3. Set self test amplitude to high by setting ACC_SELF_TEST.acc_self_test_amp = 1b1
4. Set ACC_CONF.acc_odr=1600Hz, Continuous sampling mode, ACC_CONF.acc_bwp=norm_avg4, ACC_CONF.acc_filter_perf=1b1.
5. Wait for > 2 ms
6. Set positive self-test polarity (ACC_SELF_TEST.acc_self_test_sign= 1b1)
7. Enable self-test ACC_SELF_TEST.acc_self_test_en = 1b1
8. Wait for > 50ms
9. Read and store positive acceleration value of each axis from registers DATA_8 to DATA_13
10. Set negative self-test polarity ACC_SELF_TEST.acc_self_test_sign= 1b0)
11. Enable self-test ACC_SELF_TEST.acc_self_test_en = 1b1
12. Wait for > 50ms
13. Read and store negative acceleration value of each axis from registers DATA_8 to DATA_13
14. Calculate difference of positive and negative acceleration values and compare against minimum difference signal values defined in the table below
15. Disable self-test ACC_SELF_TEST.acc_self_test_en = 1b0

The table below shows the minimum differences for each axis in order for the self test to pass. The actually measured signal differences can be significantly larger.

Self-test: Resulting minimum difference signal

	x-axis signal	y-axis signal	z-axis signal
Accelerometer	> +16g	< -15g	> +10g

It is recommended to perform a reset of the device after a self-test has been performed. If the reset cannot be performed, the following sequence must be kept to prevent unwanted interrupt generation: disable interrupts, change parameters of interrupts, wait for at least 50ms, and enable desired interrupts.

Gyroscope

The gyroscope self-test consists of independent parts, a drive test, a sense test, and a datapath test.

To perform a gyroscope self-test

1. Issue a soft-reset (see Section 4.17) or a power-on reset (POR) (see Section 4.4)
2. Initialize device (see Section 4.4)
3. Disable APS PWR_CONF.adv_power_save=0b0 and wait for 450us
4. Enable accelerometer PWR_CTRL.acc_en=0b1
5. Ensure that the device is at rest during self-test execution
6. Send *g_trigger* command using the register CMD
7. Self-test is complete, after the device sets GYR_SELF_TEST_AXES.gyr_st_axes_done=0b1
8. GYR_GAIN_STATUS.g_trig_status reports a succesful self-test or execution errors
9. The test passed if all axes report the status "ok" by GYR_SELF_TEST_AXES.gyr_axis_[xyz]_ok=0b1.

During the gyroscope self-test described above and at every gyroscope startup, i.e.:

- PWR_CTRL.gyr_en: 0b0->0b1 and PWR_CONF.fup_en==0b0
  or
- PWR_CONF.fup_en: 0b0->0b1 and PWR_CTRL.gyr_en==0b0

a drive test is automatically performed and if it fails it is reported through ERR_REG.fatal_err latest after 320 ms.

4.13. Offset Compensation

Accelerometer

The device offers manual compensation as well as inline calibration.

Offset compensation is performed with pre-filtered data, and the offset is then applied to both, pre-filtered and filtered data. If necessary the result of this computation is saturated to prevent any overflow errors (the smallest or biggest possible value is set, depending on the sign).
The offset compensation Registers OFFSET_0 to OFFSET_2 are images of the corresponding registers in the NVM. With each image update the contents of the NVM registers are written to the public registers. The public registers can be overwritten by the user at any time.
The offset compensation registers have a width of 8 bit using two's complement notation. The offset resolution (LSB) is 3.9 mg and the offset range is +- 0.5 g. Both are independent of the range setting.
Offset compensation needs to be enabled through NV_CONF.acc_off_en = 0b1

Manual Offset Compensation
The contents of the public compensation Register OFFSET_0 to OFFSET_2 may be set manually via the digital interface. After modifying the Register OFFSET_0 to OFFSET_2 the next data sample is not valid.

Offset compensation needs to be enabled through NV_CONF.acc_off_en.

Fast Offset Compensation FOC (Semi-Automatic Offset Compensation)
For certain applications, it is often desirable to calibrate the offset once and to store the compensation values permanently. This can be achieved by using manual offset compensation to determine the proper compensation values and then storing these values permanently in the NVM.

Each time the device is reset, the compensation values are loaded from the non-volatile memory into the image registers and used for offset compensation.

Gyroscope

Manual Offset Compensation
The contents of the compensation Register OFFSET_3 to OFFSET_6 may be set manually via the digital interface. After modifying the Register OFFSET_3 to OFFSET_6 the next data sample is not valid.

Offset compensation needs to be enabled through OFFSET_6.gyr_off_en.

Fast Offset Compensation FOC (Semi-Automatic Offset Compensation)
For certain applications, it is often desirable to calibrate the offset once and to store the compensation values permanently. This can be achieved by using manual offset compensation to determine the proper compensation values and then storing these values permanently in the NVM.

Each time the device is reset, the compensation values are loaded from the non-volatile memory into the image registers and used for offset compensation.

In-use Offset Compensation IOC (Full-Automatic Offset Compensation)

MEMS devices typically show offset drifts due to thermomechanical stress effects within the application, the use-case or over lifetime. To compensate such potential drifts the device offers an in-use offset compensation (IOC), which operates fully autonomeous without any necessary host interaction and in parallel to the normal device operation.

The host can choose to use either the built-in full automatic IOC feature to compensate the gyroscope offset in registers OFFSET_3 ... OFFSET_6 or control these registers manually. This is controlled by the Register GEN_SET_1.gyr_self_off.

The device will update the gyroscope offset registers automatically if all of the following conditions are met (host should not update the registers OFFSET_3 ... OFFSET_6 when this feature is enabled):

- Bit GEN_SET_1.gyr_self_off is 1
- Bit OFFSET_6.gyr_off_en is 1
- Accelerometer is enabled from either primary or OIS interface
- Gyroscope is enabled from primary interface
- Gyroscope is disabled from OIS interface

If any one of the above conditions are not met, then the feature is disabled. In this case, host can update the gyroscope offset registers. The recommended way to disable this feature is to clear the GEN_SET_1.gyr_self_off bit to 0.

4.14. **Sensitivity Error Compensation**

Accelerometer

The device supports an ultra low sensitivity (gain) compensation already by design. Refer to Section 1.

Gyroscope

The device supports sensitivity (gain) compensation (e.g. to compensate for a soldering drift). This can be done either manually by rotating the device and comparing against a known reference or can be done motionless using *Component Retrim*.

Manual SENS Error Compensation

The device supports correcting the sensitivity difference, with respect to the reference system using manual SENS error compensation. Assuming the offset of the gyro is compensated, gain compensation is enabled (OFFSET6.gyr_gain_en=0b1) and the gyroscope reports $\underline{\omega}_m$, whereas the reference system reports $\underline{\omega}_r$, the host must supply the rate ratios $\omega_{r[x-z]}/\omega_{m[x-z]}$ in the Registers GYR_GAIN_UPD_[1-3].

The encoding is given by

a. Bit width = 11 bits (FxP representation is 1.10)
b. Resolution = 0.0009765 = 2^{-10} (i.e. 0.09765% i.e. <0.1%)
c. Range = 0.75 .. 1.25 (i.e. 1 +- 25%)

e.g. 1+25% = 1.25 is represented as

b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	0	0	0	0	0	0	0	0

e.g. 1+0.09765% = 1.0009765 is represented as

b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	0	0	0	1

e.g. 1-0.09765% = 0.9990234 is represented as

b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	1	1	1	1	1	1	1

e.g. 1-25% = 0.75 is represented as

b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	0	0	0	0	0	0	0	0

The host must enable the update operation via Register GYR_GAIN_UPD_3.enable =0b1 and disable the gyroscope (PWR_CTRL.gyr_en=0b0) before triggering the manual SENS error compensation operation via issuing a command *usr_gain* to the Register CMD. After GYR_GAIN_UPD_3.enable bit gets cleared, the operation is completed, and the host can reenable the gyroscope (PWR_CTRL.gyr_en=0b1)

In case the compensation reaches the compensation range limit, the device reports this through Register GYR_GAIN_STATUS.

This compensated sensitivity value may be retained over power cycles by storing it in NVM, see Section 4.15 for details.

Component ReTrimming Feature CRT (Fast, motionless SENS Error Compensation)

For motionless SENS error compensation (CRT) the following flow needs to be executed:

1. Issue a soft-reset (see Section 4.17) or a power-on reset (POR) (see Section 4.4)
2. Initialize device (see Section 4.4)
3. Disable APS PWR_CONF.adv_power_save=0b0 and wait for 450us
4. Enable accelerometer PWR_CTRL.acc_en=0b1
5. Ensure that the device is at rest during CRT execution
6. Set GYR_CRT_CONF.crt_running=0b1
7. Set G_TRIG_1.select=1
8. Set G_TRIG_1.block=0
9. Send *g_trigger* command using the register CMD
10. CRT is complete, after the device sets GYR_CRT_CONF.crt_running=0b0
11. GYR_GAIN_STATUS.g_trig_status reports a succesful CRT run or execution errors
12. Optionally, the new gyroscope gain values can be programmed to NVM. See Section 4.15 for details about NVM programming.
13. The new gain values are applied automatically at the next start of the gyroscope.

If the device detects motion during the CRT flow, the operation is aborted and the gain remains unchanged. If CRT is abort, Register GYR_GAIN_STATUS.g_trig_status will be set to 0x03.

CRT may run in the full operating temperature range. We recommend to run CRT at the operating temperature of the device. The sensitivity error is typically minimal at the temperature CRT was performed at.

We recommend performing CRT according the description above for one-time CRT calibration. Both one-time and repeated CRT is supported by the device. For a description how to perform repeated CRT during device operation without resetting and re-initializing the device, see application note “Component ReTrimming for the BMI260 Family” BST-BMI260-AN002-00.

4.15. Non-Volatile Memory

The registers NV_CONF, OFFSET_0 to OFFSET_6, AUX_IF_TRIM, and DRV have an NVM backup which are accessible by the user. In addition, the registers for the sensitivity error compensation for the gyroscope are included in the NVM backup (see Section 4.14).

The content of the NVM is loaded to the image registers after a reset (either POR or softreset). As long as the image update is in progress, STATUS.cmd_rdy is 0b0, otherwise it is 0b1.

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

Writing to the NVM is a 4-step procedure:

1. Set PWR_CONF.adv_power_save = 0b0
2. Write the new contents to the image registers.
3. Prepare NVM write by setting GEN_SET_1.nvm_prog_prep =0b1
4. Wait 40 ms
5. Write 0b1 to bit NVM_CONF.nvm_prog_en in order to unlock the NVM.
6. Write *prog_nvm* to the CMD register to trigger the write process.

Writing to the NVM always renews the entire NVM contents and is limited in write cycles. It is possible to check the write status by reading *STATUS.cmd_rdy*. While *STATUS.cmd_rdy* = 0b0, the write process is still in progress; when *STATUS.cmd_rdy* = 0b1, writing is completed. An NVM write cycle can only be initiated, if PWR_CONF.adv_power_save = 0b0.
Until boot phase is finished (after POR or softreset), the serial interface is not operational. The NVM shadow registers must not be accessed during an ongoing NVM command (initiated through the Register CMD). In all other cases, register can be read or written.
As long as an NVM read (during sensor boot and soft reset) or an NVM write is ongoing, writes to sensor registers are discarded, reads return the Register STATUS independent of the read address.

4.16. **Error Reporting**

Device errors during operation are reported through the registers ERR_REG (hardware errors), EVENT (POR and invalid configuration events), INTERNAL_STATUS (initialization and invalid configuration), and INTERNAL_ERROR (unexpected behavior). Reserved bits in the error registers are for Bosch Sensortec internal purposes and can be ignored savely.

The register ERR_REG_MSK controls which bits in Register ERR_REG trigger an interrupt. Register INT_MAP_DATA.err_int1 and INT_MAP_DATA.err_int2 defines on which interrupt pin, the error interrupt is mapped.

Illegal settings in configuration registers ACC_CONF and GYR_CONF will result in an error code in Register EVENT. The content of the data register is undefined.

Sensor Self-Test errors are covered in Section 4.12.

4.17. **Soft-Reset**

A softreset can be initiated at any time by writing the command *softreset (0xB6)* to 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 (setting stored in the NVM) wherever applicable. This command is functional in all operation modes but must not be performed while NVM writing operation is in progress.

5. Register Description

5.1. General Remarks

This section contains register definitions. REG[x]<y> denotes bit y in byte x in register REG. Val(Name) is the value contained in the register interpreted as non-negative binary number. When writing to reserved bits, ‘0’ should be written if not stated different.

For most of the registers auto address increment applies for, with the exception of the registers below, which trap the address:

- FIFO_DATA
- INIT_DATA

Register read from a burst read must remain consistent. In order to ensure this, when a read starts in one register of a group, the registers in this group are shadowed:

- STATUS, DATA, SENSORTIME_X, TEMPERATURE_X, SC_OUT_X, FIFO_LENGTH_X
- FEATURES

5.2. **Register Map**

read/write	read only	write only	reserved

Register Address	**Register Name**	**Default Value**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x7E	CMD	0x00	cmd							
0x7D	PWR_CTRL	0x00	reserved				temp_en	acc_en	gyr_en	aux_en
0x7C	PWR_CONF	0x03	reserved					fup_en	fifo_self_wake_up	adv_power_save
0x7B	-	-	reserved							
...	-	-	reserved							
0x78	-	-	reserved							
0x77	OFFSET_6	0x00	gyr_gain_en	gyr_off_en	gyr_usr_off_z_9_8		gyr_usr_off_y_9_8		gyr_usr_off_x_9_8	
0x76	OFFSET_5	0x00	gyr_usr_off_z_7_0							
0x75	OFFSET_4	0x00	gyr_usr_off_y_7_0							
0x74	OFFSET_3	0x00	gyr_usr_off_x_7_0							
0x73	OFFSET_2	0x00	off_acc_z							
0x72	OFFSET_1	0x00	off_acc_y							
0x71	OFFSET_0	0x00	off_acc_x							
0x70	NV_CONF	0x00	reserved				acc_off_en	i2c_wdt_en	i2c_wdt_sel	spi_en
0x6F	-	-	reserved							
0x6E	GYR_SELF_TEST_AXES	0x00	reserved				gyr_axis_z_ok	gyr_axis_y_ok	gyr_axis_x_ok	gyr_st_axes_done
0x6D	ACC_SELF_TEST	0x00	reserved				acc_self_test_amp	acc_self_test_sign	reserved	acc_self_test_en
0x6C	DRV	0xFF	io_pad_i2c_b2	io_pad_drv2			io_pad_i2c_b1	io_pad_drv1		
0x6B	IF_CONF	0x00	reserved		aux_en	ois_en	reserved		spi3_ois	spi3
0x6A	NVM_CONF	0x00	reserved						nvm_prog_en	reserved

0x69	GYR_CRT_CONF	0x00	reserved				rdy_for_dl	crt_running	reserved	
0x68	AUX_IF_TRIM	0x01	reserved					spare3	asda_pupsel	
0x67	-	-	reserved							
...	-	-	reserved							
0x60	-	-	reserved							
0x5F	INTERNAL_ERROR	0x00	reserved			feat_eng_disabled	reserved	int_err_2	int_err_1	reserved
0x5E	INIT_DATA	0x00	data							
0x5D	-	-	reserved							
0x5C	INIT_ADDR_1	0x00	base_11_4							
0x5B	INIT_ADDR_0	0x00	reserved				base_0_3			
0x5A	-	-	reserved							
0x59	INIT_CTRL	0x00	init_ctrl							
0x58	INT_MAP_DATA	0x00	err_int2	drdy_int2	fwm_int2	ffull_int2	err_int1	drdy_int1	fwm_int1	ffull_int1
0x57	INT2_MAP_FEAT	0x00	reserved	any_motion_out	no_motion_out	wrist_gesture_out	wrist_wear_wakeup_out	activity_out	step_counter_out	sig_motion_out
0x56	INT1_MAP_FEAT	0x00	reserved	any_motion_out	no_motion_out	wrist_gesture_out	wrist_wear_wakeup_out	activity_out	step_counter_out	sig_motion_out
0x55	INT_LATCH	0x00	reserved							int_latch
0x54	INT2_IO_CTRL	0x00	reserved			input_en	output_en	od	lvl	reserved
0x53	INT1_IO_CTRL	0x00	reserved			input_en	output_en	od	lvl	reserved
0x52	ERR_REG_MSK	0x00	aux_err	fifo_err	reserved	internal_err				fatal_err
0x51	-	-	reserved							
0x50	-	-	reserved							
0x4F	AUX_WR_DATA	0x02	write_data							
0x4E	AUX_WR_ADDR	0x4C	write_addr							
0x4D	AUX_RD_ADDR	0x42	read_addr							
0x4C	AUX_IF_CONF	0x83	aux_manual_en	aux_fcu_write_en	reserved		man_rd_burst		aux_rd_burst	
0x4B	AUX_DEV_ID	0x20	i2c_device_addr							reserved
0x4A	SATURATION	0x00	reserved		gyr_z	gyr_y	gyr_x	acc_z	acc_y	acc_x

0x49	FIFO_CONFIG_1	0x10	fifo_gyr_en	fifo_acc_en	fifo_aux_en	fifo_header_en	fifo_tag_int2_en		fifo_tag_int1_en	
0x48	FIFO_CONFIG_0	0x02	reserved						fifo_time_en	fifo_stop_on_full
0x47	FIFO_WTM_1	0x02	reserved			fifo_water_mark_12_8				
0x46	FIFO_WTM_0	0x00	fifo_water_mark_7_0							
0x45	FIFO_DOWNS	0x88	acc_fifo_filt_data	acc_fifo_downs			gyr_fifo_filt_data	gyr_fifo_downs		
0x44	AUX_CONF	0x46	aux_offset				aux_odr			
0x43	GYR_RANGE	0x00	reserved				ois_range	gyr_range		
0x42	GYR_CONF	0xA9	gyr_filter_perf	gyr_noise_perf	gyr_bwp		gyr_odr			
0x41	ACC_RANGE	0x02	reserved						acc_range	
0x40	ACC_CONF	0xA8	acc_filter_perf	acc_bwp			acc_odr			
0x3F	FEATURES[15]	0x00	features_in_out							
...	...	-								
0x30	FEATURES[0]	0x00								
0x2F	FEAT_PAGE	0x00	reserved					page		
0x2E	-	-	reserved							
...	-	-	reserved							
0x27	-	-	reserved							
0x26	FIFO_DATA	0x00	fifo_data							
0x25	FIFO_LENGTH_1	0x00	reserved		fifo_byte_counter_13_8					
0x24	FIFO_LENGTH_0	0x00	fifo_byte_counter_7_0							
0x23	TEMPERATURE_1	0x80	tmp_data_15_8							
0x22	TEMPERATURE_0	0x00	tmp_data_7_0							
0x21	INTERNAL_STATUS	0x00	Reserved	odr_50hz_error	axes_remap_error	Reserved	message			
0x20	WR_GEST_ACT	0x00	reserved			act_out		wr_gest_out		
0x1F	SC_OUT_1	0x00	byte_1							
0x1E	SC_OUT_0	0x00	byte_0							

0x1D	INT_STATUS_1	0x00	acc_drdy_int	gyr_drdy_int	aux_drdy_int	reserved		err_int	fwm_int	ffull_int
0x1C	INT_STATUS_0	0x00	reserved	any_motion_out	no_motion_out	wrist_gesture_out	wrist_wear_wakeup_out	activity_out	step_counter_out	sig_motion_out
0x1B	EVENT	0x01	reserved			error_code			reserved	por_detected
0x1A	SENSORTIME_2	0x00	sensor_time_23_16							
0x19	SENSORTIME_1	0x00	sensor_time_15_8							
0x18	SENSORTIME_0	0x00	sensor_time_7_0							
0x17	DATA_19	0x00	gyr_z_15_8							
0x16	DATA_18	0x00	gyr_z_7_0							
0x15	DATA_17	0x00	gyr_y_15_8							
0x14	DATA_16	0x00	gyr_y_7_0							
0x13	DATA_15	0x00	gyr_x_15_8							
0x12	DATA_14	0x00	gyr_x_7_0							
0x11	DATA_13	0x00	acc_z_15_8							
0x10	DATA_12	0x00	acc_z_7_0							
0x0F	DATA_11	0x00	acc_y_15_8							
0x0E	DATA_10	0x00	acc_y_7_0							
0x0D	DATA_9	0x00	acc_x_15_8							
0x0C	DATA_8	0x00	acc_x_7_0							
0x0B	DATA_7	0x00	aux_r_15_8							
0x0A	DATA_6	0x00	aux_r_7_0							
0x09	DATA_5	0x00	aux_z_15_8							
0x08	DATA_4	0x00	aux_z_7_0							
0x07	DATA_3	0x00	aux_y_15_8							
0x06	DATA_2	0x00	aux_y_7_0							
0x05	DATA_1	0x00	aux_x_15_8							

0x04	DATA_0	0x00	aux_x_7_0							
0x03	STATUS	0x10	drdy_acc	drdy_gyr	drdy_aux	cmd_rdy	reserved	aux_busy	reserved	
0x02	ERR_REG	0x00	aux_err	fifo_err	reserved	internal_err				fatal_err
0x01	-	-	reserved							
0x00	CHIP_ID	0x24	chip_id							

FEATURES Pages

Register Address	Register Name	Page 0	Page 1	Page 2	Page 3
0x30	FEATURES[0,1]	SC_OUT_0_1	Reserved	NOMO_1	SC_1
0x32	FEATURES[2,3]	SC_OUT_2_3	G_TRIG_1	NOMO_2	SC_2
0x34	FEATURES[4,5]	ACT_OUT	GEN_SET_1	SIGMO_1	SC_3
0x36	FEATURES[6,7]	WR_GEST_OUT	GYR_GAIN_UPD_1	Reserved	SC_4
0x38	FEATURES[8,9]	GYR_GAIN_STATUS	GYR_GAIN_UPD_2	Reserved	SC_5
0x3A	FEATURES[10,11]	Reserved	GYR_GAIN_UPD_3	Reserved	SC_6
0x3C	FEATURES[12,13]	GYR_CAS	ANYMO_1	Reserved	SC_7
0x3E	FEATURES[14,15]	Reserved	ANYMO_2	SIGMO_2	SC_8

FEATURES Pages

Register Address	Register Name	Page 4	Page 5	Page 6	Page 7
0x30	FEATURES[0,1]	SC_9	SC_17	SC_25	WR_WAKEUP_1
0x32	FEATURES[2,3]	SC_10	SC_18	SC_26	WR_WAKEUP_2
0x34	FEATURES[4,5]	SC_11	SC_19	SC_27	WR_WAKEUP_3
0x36	FEATURES[6,7]	SC_12	SC_20	WR_GEST_1	WR_WAKEUP_4
0x38	FEATURES[8,9]	SC_13	SC_21	WR_GEST_2	WR_WAKEUP_5
0x3A	FEATURES[10,11]	SC_14	SC_22	WR_GEST_3	WR_WAKEUP_6
0x3C	FEATURES[12,13]	SC_15	SC_23	WR_GEST_4	WR_WAKEUP_7
0x3E	FEATURES[14,15]	SC_16	SC_24	Reserved	Reserved

Register (0x00) CHIP_ID

DESCRIPTION: Chip identification code
RESET: 0x24
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x00		CHIP_ID		0x24	
	7...0	chip_id	Chip identification code	0x24	R

Register (0x02) ERR_REG

DESCRIPTION: Reports sensor error conditions
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x02		ERR_REG		0x00	
	0	fatal_err	Fatal Error, chip is not in operational state (Boot-, power-system). This flag will be reset only by power-on-reset or softreset.	0x0	R
	4...1	internal_err	Internal error, please contact your Bosch Sensortec regional support team.	0x0	R
	6	fifo_err	Error when a frame is read in streaming mode (so skipping is not possible) and fifo is overfilled (with virtual and/or regular frames). This flag will be reset when read.	0x0	R
	7	aux_err	Error in I2C-Master detected. This flag will be reset when read.	0x0	R

Register (0x03) STATUS

DESCRIPTION: Sensor status flags
RESET: 0x10
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x03		STATUS		0x10	
	2	aux_busy	‘0’(‘1’) indicate a (no) Auxiliary sensor interface operation is ongoing triggered via AUX_RD_ADDR, AUX_WR_ADDR or from FCU.	0x0	R
	4	cmd_rdy	CMD decoder status. `0´ -> Command in progress `1´ -> Command decoder is ready to accept a new command	0x1	R
	5	drdy_aux	Data ready for Auxiliary sensor. It gets reset, when one Auxiliary sensor DATA register is read out	0x0	R
	6	drdy_gyr	Data ready for Gyroscope. It gets reset, when one Gyroscope DATA register is read out	0x0	R
	7	drdy_acc	Data ready for Accelerometer. It gets reset, when one Accelerometer DATA register is read out	0x0	R

Register (0x04) DATA_0

DESCRIPTION: AUX_X(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x04		DATA_0		0x00	
	7...0	aux_x_7_0	copy of register Val(AUX_IF[1]) in Auxiliary sensor register map.	0x0	R

Register (0x05) DATA_1

DESCRIPTION: AUX_X(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x05		DATA_1		0x00	
	7...0	aux_x_15_8	copy of register Val(AUX_IF[1])+1 in Auxiliary sensor register map	0x0	R

Register (0x06) DATA_2

DESCRIPTION: AUX_Y(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x06		DATA_2		0x00	
	7...0	aux_y_7_0	copy of register Val(AUX_IF[1])+2 in Auxiliary sensor register map	0x0	R

Register (0x07) DATA_3

DESCRIPTION: AUX_Y(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x07		DATA_3		0x00	
	7...0	aux_y_15_8	copy of register Val(AUX_IF[1])+3 in Auxiliary sensor register map	0x0	R

Register (0x08) DATA_4

DESCRIPTION: AUX_Z(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x08		DATA_4		0x00	
	7...0	aux_z_7_0	copy of register Val(AUX_IF[1])+4 in Auxiliary sensor register map	0x0	R

Register (0x09) DATA_5

DESCRIPTION: AUX_Z(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x09		DATA_5		0x00	
	7...0	aux_z_15_8	copy of register Val(AUX_IF[1])+5 in Auxiliary sensor register map	0x0	R

Register (0x0A) DATA_6

DESCRIPTION: AUX_R(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0A		DATA_6		0x00	
	7...0	aux_r_7_0	copy of register Val(AUX_IF[1])+6 in Auxiliary sensor register map	0x0	R

Register (0x0B) DATA_7

DESCRIPTION: AUX_R(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0B		DATA_7		0x00	
	7...0	aux_r_15_8	copy of register Val(AUX_IF[1])+7 in Auxiliary sensor register map	0x0	R

Register (0x0C) DATA_8

DESCRIPTION: ACC_X(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0C		DATA_8		0x00	
	7...0	acc_x_7_0		0x0	R

Register (0x0D) DATA_9

DESCRIPTION: ACC_X(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0D		DATA_9		0x00	
	7...0	acc_x_15_8		0x0	R

Register (0x0E) DATA_10

DESCRIPTION: ACC_Y(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0E		DATA_10		0x00	
	7...0	acc_y_7_0		0x0	R

Register (0x0F) DATA_11

DESCRIPTION: ACC_Y(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x0F		DATA_11		0x00	
	7...0	acc_y_15_8		0x0	R

Register (0x10) DATA_12

DESCRIPTION: ACC_Z(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x10		DATA_12		0x00	
	7...0	acc_z_7_0		0x0	R

Register (0x11) DATA_13

DESCRIPTION: ACC_Z(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x11		DATA_13		0x00	
	7...0	acc_z_15_8		0x0	R

Register (0x12) DATA_14

DESCRIPTION: GYR_X(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x12		DATA_14		0x00	
	7...0	gyr_x_7_0		0x0	R

Register (0x13) DATA_15

DESCRIPTION: GYR_X(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x13		DATA_15		0x00	
	7...0	gyr_x_15_8		0x0	R

Register (0x14) DATA_16

DESCRIPTION: GYR_Y(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x14		DATA_16		0x00	
	7...0	gyr_y_7_0		0x0	R

Register (0x15) DATA_17

DESCRIPTION: GYR_Y(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x15		DATA_17		0x00	
	7...0	gyr_y_15_8		0x0	R

Register (0x16) DATA_18

DESCRIPTION: GYR_Z(LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x16		DATA_18		0x00	
	7...0	gyr_z_7_0		0x0	R

Register (0x17) DATA_19

DESCRIPTION: GYR_Z(MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x17		DATA_19		0x00	
	7...0	gyr_z_15_8		0x0	R

Register (0x18) SENSORTIME_0

DESCRIPTION: Sensor time <7:0>
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x18		SENSORTIME_0		0x00	
	7...0	sensor_time_7_0	Sensor time <7:0>	0x0	R

Register (0x19) SENSORTIME_1

DESCRIPTION: Sensor time <15:8>
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x19		SENSORTIME_1		0x00	
	7...0	sensor_time_15_8	Sensor time <15:8>.	0x0	R

Register (0x1A) SENSORTIME_2

DESCRIPTION: Sensor time <23:16>
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1A		SENSORTIME_2		0x00	
	7...0	sensor_time_23_16	Sensor time <23:16> The sensor time is a 24 bit counter available in suspend, low power, and normal mode. The value of the SENSORTIME register is shadowed, when it is read in a burst read with the data register at the beginning of the operation and the shadowed value is returned. When the fifo is read the register is shadowed, whenever a new frame is read. The resolution of the sensor_time is 39.06125 us, and it is synchrounous to ODR. The register wraps if it reaches 0xFFFFFF.	0x0	R

Register (0x1B) EVENT

DESCRIPTION: Sensor event flags
RESET: 0x01
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1B		EVENT		0x01	
	0	por_detected	'1' after device power up or softreset, '0' after status read.	0x1	R
	4...2	error_code	Error codes for persistent errors **Value** **Name** **Description** 0x00 no_error no error is reported 0x01 acc_err error in Register ACC_CONF 0x02 gyr_err error in Register GYR_CONF 0x03 acc_and_gyr_err error in Registers ACC_GYR & GYR_CONF	0x0	R

Register (0x1C) INT_STATUS_0

DESCRIPTION: Interrupt/Feature Status. Will be cleared on read.
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1C		INT_STATUS_0		0x00	
	0	sig_motion_out	Sigmotion output.	0x0	R
	1	step_counter_out	Step-counter watermark or Step-detector output	0x0	R
	2	activity_out	Step activity output	0x0	R
	3	wrist_wear_wakeup_out	Wrist wear wakeup output	0x0	R
	4	wrist_gesture_out	Wrist gesture output	0x0	R
	5	no_motion_out	No motion detection output	0x0	R
	6	any_motion_out	Any motion detection output	0x0	R
	7	reserved	Reserved	0x0	R

Register (0x1D) INT_STATUS_1

DESCRIPTION: Interrupt Status 1
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1D		INT_STATUS_1		0x00	
	0	ffull_int	FIFO Full Interrupt	0x0	R
	1	fwm_int	FIFO Watermark Interrupt	0x0	R
	2	err_int	ERROR Interrupt	0x0	R
	5	aux_drdy_int	Auxiliary Data Ready Interrupt	0x0	R
	6	gyr_drdy_int	Gyroscope Data Ready Interrupt	0x0	R
	7	acc_drdy_int	Accelerometer Data Ready Interrupt	0x0	R

Register (0x1E) SC_OUT_0

DESCRIPTION: Step counting value byte-0
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1E		SC_OUT_0		0x00	
	7...0	byte_0	Step counting value byte-0 (least significant byte)	0x0	R

Register (0x1F) SC_OUT_1

DESCRIPTION: Step counting value byte-1
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x1F		SC_OUT_1		0x00	
	7…0	byte_1	Step counting value byte-1	0x0	R

Register (0x20) WR_GEST_ACT

DESCRIPTION: Wrist gesture and activity detection output
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x20		WR_GEST_ACT		0x00	
	2…0	wr_gest_out	Output value of the wrist gesture detection feature. Value after device initialization is 0b00 i.e. unknown gesture **Value** **Name** **Description** 0x00 unknown_gesture Unknown gesture 0x01 push_arm_down Push arm down gesture 0x02 pivot_up Pivot up gesture 0x03 wrist_shake_jiggle Wrist shake/jiggle gesture 0x04 flick_in Arm flick in gesture 0x05 flick_out Arm flick out gesture	0x0	R
	4…3	act_out	Output value of the activity detection feature. Value after device initialization is 0b11 i.e. unknown activity **Value** **Name** **Description** 0x00 still User stationary 0x01 walking User walking 0x02 running User running 0x03 unknown Unknown state	0x0	R

Register (0x21) INTERNAL_STATUS

DESCRIPTION: Error bits and message indicating internal status
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x21		INTERNAL_STATUS		0x00	
	3...0	message	Internal Status Message **Value** **Name** **Description** 0x00 not_init ASIC is not initialized 0x01 init_ok ASIC initialized 0x02 init_err Initialization error 0x03 drv_err Invalid driver 0x04 sns_stop Sensor stopped 0x05 nvm_error Internal error while accessing NVM 0x06 start_up_error Internal error while accessing NVM and Initialization error 0x07 compat_error Compatibility error	0x0	R
	4	Reserved	Reserved	0x0	R
	5	axes_remap_error	Incorrect axes remapping. X,Y,Z axes must be mapped to exclusively separate axes i.e. they cannot be mapped to same axes.	0x0	R
	6	odr_50hz_error	The minimum bandwidth conditions are not respected for the features which require 50 Hz data.	0x0	R
	7	Reserved	The minimum bandwidth conditions are not respected for the features which require 200 Hz data.	0x0	R

Register (0x22) TEMPERATURE_0

DESCRIPTION: Temperature LSB; The temperature is disabled when all sensors are in suspend. The output word of the 16-bit temperature sensor is valid if the Gyroscope is in normal mode, i.e. gyr_pmu_status=1. The resolution is 1/2^9 K/LSB. The absolute accuracy of the temperature is in the order of:

0x7FFF -> 87-1/2^9 °C
0x0000 -> 23°C
0x8001 -> -41+1/2^9 °C
0x8000 -> invalid

If the Gyroscope is in normal mode (see register PMU_STATUS), the temperature is updated every 10 ms (+-12%), if the gyroscope is in standby mode or fast-power up mode, the temperature is updated ever 1.28 s aligned with bit 15 of the register SENSORTIME.

RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x22		TEMPERATURE_0		0x00	
	7...0	tmp_data_7_0	Temperature value.	0x0	R

Register (0x23) TEMPERATURE_1

DESCRIPTION: Contains the MSBs of temperature sensor value
RESET: 0x80
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x23		TEMPERATURE_1		0x80	
	7...0	tmp_data_15_8	Temperature LSBs.	0x80	R

Register (0x24) FIFO_LENGTH_0

DESCRIPTION: FIFO byte count register (LSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x24		FIFO_LENGTH_0		0x00	
	7...0	fifo_byte_counter_7_0	Current fill level of FIFO buffer This includes the skip frame for a full fifo. An empty FIFO corresponds to 0x000. The byte counter may be reset by reading out all frames from the FIFO buffer or when the FIFO is reset through the register CMD. The byte counter is updated each time a complete frame was read or written.	0x0	R

Register (0x25) FIFO_LENGTH_1

DESCRIPTION: FIFO byte count register (MSB)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x25		FIFO_LENGTH_1		0x00	
	5...0	fifo_byte_counter_13_8	FIFO byte counter bits 13..8	0x0	R

Register (0x26) FIFO_DATA

DESCRIPTION: FIFO data output register
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x26		FIFO_DATA		0x00	
	7...0	fifo_data	FIFO read data (8 bits) Data format depends on the setting of register FIFO_CONFIG. 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.	0x0	R

Register (0x2F) FEAT_PAGE

DESCRIPTION: Page number for feature configuration and output registers
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x2F		FEAT_PAGE		0x00	
	2...0	page	Map 16 feature registers to one of the 8 feature pages	0x0	RW

Register (0x30) FEATURES[16]

DESCRIPTION: Input registers for feature configuration. Output registers for feature results.
RESET: 0x00
DEFINITION (Go to register map):

Page 0

Address	Bit	Name	Description	Reset	Access
step_counter_output					
0x30		SC_OUT_0_1	Describes lower word of step counter	0x0000	
	7...0	byte_0	Value of step counter byte 0	0x0	R
	15...8	byte_1	Value of step counter byte 1	0x0	R
0x32		SC_OUT_2_3	Describes higher word of step counter	0x0000	
	7...0	byte_2	Value of step counter byte 2	0x0	R
	15...8	byte_3	Value of step counter byte 3	0x0	R
activity_output					
0x34		ACT_OUT	Describes activity output	0x0000	
	1...0	act_out	Output value of the activity detection feature. Value after device initialization is 0b11 i.e. unknown activity **Value** **Name** **Description** 0x00 still User stationary 0x01 walking User walking 0x02 running User running 0x03 unknown Unknown state	0x0	R

wrist_gesture_output					
0x36		WR_GEST_OUT	Describes wrist gesture output	0x0000	
	2…0	wr_gest_out	Output value of the wrist gesture detection feature. Value after device initialization is 0b00 i.e. unknown gesture **Value** **Name** **Description** 0x00 unknown_gesture Unknown gesture 0x01 push_arm_down Push arms down gesture 0x02 pivot_up Pivot up gesture 0x03 wrist_shake_jiggle Wrist shake/jiggle gesture 0x04 flick_in Arm flick in gesture 0x05 flick_out Arm flick out gesture	0x0	R
gyr_gain_status					
0x38		GYR_GAIN_STATUS	Describes the saturation status for the gyroscope gain update and G_TRIGGER command status	0x0000	
	0	sat_x	This bit will be 1 if the updated gain results to saturated value based on the ratio provided for x axis, otherwise it will be 0	0x0	R
	1	sat_y	This bit will be 1 if the updated gain results to saturated value based on the ratio provided for y axis, otherwise it will be 0	0x0	R
	2	sat_z	This bit will be 1 if the updated gain results to saturated value based on the ratio provided for z axis, otherwise it will be 0	0x0	R
	5…3	g_trig_status	Status of gyroscope trigger G_TRIGGER command. These bits are updated at the end of feature execution. **Value** **Name** **Description** 0x00 no_err Command is valid. Selected feature has been executed and output of feature	0x0	R

			has been updated. 0x01 precon_err Command is aborted. Pre-condition to start the feature was not completed. 0x02 dl_err Command is aborted. Unsuccessful download of 2kB configuration stream. 0x03 abort_err Command is aborted either by host via the block bit or due to motion detection.		
Reserved					
0x3A		Reserved	Reserved	0x0000	
	15...0	Reserved	Reserved	0x0	R
gyr_postproc					
0x3C		GYR_CAS	Register for gyroscope data post processing	0x0000	
	6...0	factor_zx	Factor to further optimize the gyroscope performance	0x0	R
Reserved					
0x3E		Reserved	Reserved	0x0000	
	8	Reserved	Reserved	0x0	R
	9	Reserved	Reserved	0x0	R
	10	Reserved	Reserved	0x0	R
	11	Reserved	Reserved	0x0	R
	12	Reserved	Reserved	0x0	R
	13	Reserved	Reserved	0x0	R
	14	Reserved	Reserved	0x0	R
	15	Reserved	Reserved	0x0	R

Page 1

Address	Bit	Name	Description	Reset	Access
general_settings					
0x30		Reserved	Reserved	0x0000	
	15...0	Reserved	Reserved	0x0	R
0x32		G_TRIG_1	Configuration for features triggered by G_TRIGGER command.	0x0000	
	7...0	max_burst_len	Maximum burst-write length in 16-bits words to download 2kB configuration stream of G_TRIGGER feature. Range is 0 to 255. E.g. value = 20 means that maximum burst-write length is set to 20 words or 40 bytes.	0x0	RW
	8	select	Select feature that should be executed **Value** **Name** **Description** 0x00 gyr_bist Gyroscope built-in self-test will be executed 0x01 crt CRT will be executed	0x0	RW
	9	block	Block feature with next G_TRIGGER command **Value** **Name** **Description** 0x00 unblock Do not block further G_TRIGGER commands 0x01 block With the next G_TRIGGER command, the ongoing selected feature will be aborted OR if a feature is not ongoing then the G_TRIGGER command will be ignored	0x0	RW
0x34		GEN_SET_1	Describes configuration of general features	0x0088	
	1...0	map_x_axis	Map the x axis to desired axis **Value** **Name** **Description** 0x00 x_axis Map to x-axis 0x01 y_axis Map to y-axis 0x02 z_axis Map to z-axis	0x0	RW

			0x03 reserved Map to x-axis		
	2	map_x_axis_sign	Map the x axis sign to the desired one. **Value** **Name** **Description** 0x00 not_invert Clear this bit to not invert the x axis 0x01 invert Set this bit to invert the x axis	0x0	RW
	4...3	map_y_axis	Map the y axis to desired axis **Value** **Name** **Description** 0x00 x_axis Map to x-axis 0x01 y_axis Map to y-axis 0x02 z_axis Map to z-axis 0x03 reserved Map to y-axis	0x1	RW
	5	map_y_axis_sign	Map the y axis sign to the desired one **Value** **Name** **Description** 0x00 not_invert Clear this bit to not invert the y axis 0x01 invert Set this bit to invert the y axis	0x0	RW
	7...6	map_z_axis	Map the z axis to desired axis **Value** **Name** **Description** 0x00 x_axis Map to x-axis 0x01 y_axis Map to y-axis 0x02 z_axis Map to z-axis 0x03 reserved Map to z-axis	0x2	RW
	8	map_z_axis_sign	Map the z axis sign to the desired one **Value** **Name** **Description** 0x00 not_invert Clear this bit to not invert the z axis 0x01 invert Set this bit to invert the z axis	0x0	RW
	9	gyr_self_off	Describes the self offset correction behavior **Value** **Name** **Description** 0x00 disable Disable self offset correction. Host should update the gyroscope offset register. 0x01 enable Enable self offset correction. Gyroscope offset register will be	0x0	RW

			updated by the device. Host should not update the gyroscope offset registers.		
	10	nvm_prog_prep	Prepares the system for NVM programming	0x0	RW
gyr_gain_update					
0x36		GYR_GAIN_UPD_1	ωrx/ωmx for which the gain needs to be updated.	0x0000	
	10…0	ratio_x	gain update value for x-axis. Fixed point representation is Q(1,10) with range from 1±0.25. For example, value of 0.75 shall be represented in 11bits as 0x300 and 1.25 shall be represented in 11bits as 0x500	0x0	RW
0x38		GYR_GAIN_UPD_2	ωry/ωmy for which the gain needs to be updated.	0x0000	
	10…0	ratio_y	gain update value for y-axis. Fixed point representation is Q(1,10) with range from 1±0.25. For example, value of 0.75 shall be represented in 11bits as 0x300 and 1.25 shall be represented in 11bits as 0x500	0x0	RW
0x3A		GYR_GAIN_UPD_3	ωrz/ωmz for which the gain needs to be updated.	0x0000	
	10…0	ratio_z	gain update value for z-axis. Fixed point representation is Q(1,10) with range from 1±0.25. For example, value of 0.75 shall be represented in 11bits as 0x300 and 1.25 shall be represented in 11bits as 0x500	0x0	RW
	11	enable	Enable the gyroscope gain update by writing a value 1 to it. Once the gain update is completed, the device will clear the bit.	0x0	RW
any_motion					
0x3C		ANYMO_1	Any-motion detection general configuration flags - part 1	0xE005	
	12…0	duration	Defines the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion. It is expressed in 50 Hz samples (20 ms). Range is 0 to 163sec. Default value is 5=100ms.	0x5	RW
	13	select_x	Selects the feature on a per-axis basis	0x1	RW

	14	select_y	Selects the feature on a per-axis basis	0x1	RW
	15	select_z	Selects the feature on a per-axis basis	0x1	RW
0x3E		ANYMO_2	Any-motion detection general configuration flags - part 2	0x38AA	
	10...0	threshold	Slope threshold value for any-motion detection. Range is 0 to 1g. Default value is 0xAA = 83mg.	0xAA	RW
	14...11	out_conf	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value** | **Name** | **Description** 0x00 | disable | Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 | BIT_0 | Output assigned to bit-0 0x02 | BIT_1 | Output assigned to bit-1 0x03 | BIT_2 | Output assigned to bit-2 0x04 | BIT_3 | Output assigned to bit-3 0x05 | BIT_4 | Output assigned to bit-4 0x06 | BIT_5 | Output assigned to bit-5 0x07 | BIT_6 | Output assigned to bit-6 0x08 | BIT_7 | Output assigned to bit-7	0x7	RW
	15	enable	Enables the feature	0x0	RW

Page 2

Address	Bit	Name	Description	Reset	Access
no_motion					
0x30		NOMO_1	No-motion detection general configuration flags - part 1	0xE005	
	12…0	duration	Defines the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion. It is expressed in 50 Hz samples (20 ms). Range is 0 to 163sec. Default value is 5=100ms.	0x5	RW
	13	select_x	Selects the feature on a per-axis basis	0x1	RW
	14	select_y	Selects the feature on a per-axis basis	0x1	RW
	15	select_z	Selects the feature on a per-axis basis	0x1	RW
0x32		NOMO_2	No-motion detection general configuration flags - part 2	0x3090	
	10…0	threshold	Slope threshold value for no-motion detection. Range is 0 to 1g. Default value is 0x90 = 70mg.	0x90	RW
	14…11	out_conf	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value** **Name** **Description** 0x00 disable Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 BIT_0 Output assigned to bit-0 0x02 BIT_1 Output assigned to bit-1 0x03 BIT_2 Output assigned to bit-2 0x04 BIT_3 Output assigned to bit-3 0x05 BIT_4 Output assigned to bit-4 0x06 BIT_5 Output assigned to bit-5 0x07 BIT_6 Output assigned to bit-6 0x08 BIT_7 Output assigned to bit-7	0x6	RW
	15	enable	Enables the feature	0x0	RW
sig_motion					
0x34		SIGMO_1	Block size	0x00FA	
	15…0	block_size	Defines the duration after which the significant motion interrupt is triggered. It is expressed in 50 Hz samples (20 ms). Default value is 0xFA=5sec.	0xFA	RW
0x36		Reserved	Reserved	0x0096	
	15…0	Reserved	Reserved	0x96	RW
0x38		Reserved	Reserved	0x094B	

<table>
<tr><td></td><td>15…0</td><td>Reserved</td><td>Reserved</td><td>0x94B</td><td>RW</td></tr>
<tr><td>0x3A</td><td></td><td>Reserved</td><td>Reserved</td><td>0x0011</td><td></td></tr>
<tr><td></td><td>15…0</td><td>Reserved</td><td>Reserved</td><td>0x11</td><td>RW</td></tr>
<tr><td>0x3C</td><td></td><td>Reserved</td><td>Reserved</td><td>0x0011</td><td></td></tr>
<tr><td></td><td>15…0</td><td>Reserved</td><td>Reserved</td><td>0x11</td><td>RW</td></tr>
<tr><td>0x3E</td><td></td><td>SIGMO_2</td><td>Significant motion setting</td><td>0x0002</td><td></td></tr>
<tr><td></td><td>0</td><td>enable</td><td>Enables the feature</td><td>0x0</td><td>RW</td></tr>
<tr><td></td><td>4…1</td><td>out_conf</td><td>Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0x00</td><td>disable</td><td>Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT</td></tr>
<tr><td>0x01</td><td>BIT_0</td><td>Output assigned to bit-0</td></tr>
<tr><td>0x02</td><td>BIT_1</td><td>Output assigned to bit-1</td></tr>
<tr><td>0x03</td><td>BIT_2</td><td>Output assigned to bit-2</td></tr>
<tr><td>0x04</td><td>BIT_3</td><td>Output assigned to bit-3</td></tr>
<tr><td>0x05</td><td>BIT_4</td><td>Output assigned to bit-4</td></tr>
<tr><td>0x06</td><td>BIT_5</td><td>Output assigned to bit-5</td></tr>
<tr><td>0x07</td><td>BIT_6</td><td>Output assigned to bit-6</td></tr>
<tr><td>0x08</td><td>BIT_7</td><td>Output assigned to bit-7</td></tr>
</table></td><td>0x1</td><td>RW</td></tr>
</table>

Page 3

Address	Bit	Name	Description	Reset	Access
step_counter_1					
0x30		SC_1	Step Counter setting	0x012D	
	15...0	param_1	Step Counter param 1	0x12D	RW
0x32		SC_2	Step Counter setting	0x7BD4	
	15...0	param_2	Step Counter param 2	0x7BD4	RW
0x34		SC_3	Step Counter setting	0x013B	
	15...0	param_3	Step Counter param 3	0x13B	RW
0x36		SC_4	Step Counter setting	0x7ADB	
	15...0	param_4	Step Counter param 4	0x7ADB	RW
0x38		SC_5	Step Counter setting	0x0004	
	15...0	param_5	Step Counter param 5	0x4	RW
0x3A		SC_6	Step Counter setting	0x7B3F	
	15...0	param_6	Step Counter param 6	0x7B3F	RW
0x3C		SC_7	Step Counter setting	0x6CCD	
	15...0	param_7	Step Counter param 7	0x6CCD	RW
0x3E		SC_8	Step Counter setting	0x04C3	
	15...0	param_8	Step Counter param 8	0x4C3	RW

Page 4

Address	Bit	Name	Description	Reset	Access
step_counter_2					
0x30		SC_9	Step Counter setting	0x0985	
	15...0	param_9	Step Counter param 9	0x985	RW
0x32		SC_10	Step Counter setting	0x04C3	
	15...0	param_10	Step Counter param 10	0x4C3	RW
0x34		SC_11	Step Counter setting	0xE6EC	
	15...0	param_11	Step Counter param 11	0xE6EC	RW
0x36		SC_12	Step Counter setting	0x460C	
	15...0	param_12	Step Counter param 12	0x460C	RW
0x38		SC_13	Step Counter setting	0x0001	
	15...0	param_13	Step Counter param 13	0x1	RW
0x3A		SC_14	Step Counter setting	0x0027	
	15...0	param_14	Step Counter param 14	0x27	RW
0x3C		SC_15	Step Counter setting	0x0019	
	15...0	param_15	Step Counter param 15	0x19	RW
0x3E		SC_16	Step Counter setting	0x0096	
	15...0	param_16	Step Counter param 16	0x96	RW

Page 5

Address	Bit	Name	Description	Reset	Access
step_counter_3					
0x30		SC_17	Step Counter setting	0x00A0	
	15...0	param_17	Step Counter param 17	0xA0	RW
0x32		SC_18	Step Counter setting	0x0001	
	15...0	param_18	Step Counter param 18	0x1	RW
0x34		SC_19	Step Counter setting	0x000C	
	15...0	param_19	Step Counter param 19	0xC	RW
0x36		SC_20	Step Counter setting	0x3CF0	
	15...0	param_20	Step Counter param 20	0x3CF0	RW
0x38		SC_21	Step Counter setting	0x0100	
	15...0	param_21	Step Counter param 21	0x100	RW
0x3A		SC_22	Step Counter setting	0x0001	
	15...0	param_22	Step Counter param 22	0x1	RW
0x3C		SC_23	Step Counter setting	0x0003	
	15...0	param_23	Step Counter param 23	0x3	RW
0x3E		SC_24	Step Counter setting	0x0001	
	15...0	param_24	Step Counter param 24	0x1	RW

Page 6

Address	Bit	Name	Description	Reset	Access
step_counter_4					
0x30		SC_25	Step Counter setting	0x000E	
	15...0	param_25	Step Counter param 25	0xE	RW
0x32		SC_26	Step Counter and Step Detector Settings	0x0000	
	9...0	watermark_level	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 20460, with resolution of 20 steps. If 0, the output is disabled.	0x0	RW
	10	reset_counter	Step count value can be reset only when any one of features mentioned in this register is enabled.	0x0	RW
	11	en_detector	Enables the Step Detector.	0x0	RW
	12	en_counter	Enables the Step Counter.	0x0	RW
	13	en_activity	Enables the activity detection(Running, Walking, Stationary, Unknown)	0x0	RW

0x34		SC_27	Step Counter and Step Detector Settings	0x0032	
	3...0	out_conf_step_detector	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value** **Name** **Description** 0x00 disable Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 BIT_0 Output assigned to bit-0 0x02 BIT_1 Output assigned to bit-1 0x03 BIT_2 Output assigned to bit-2 0x04 BIT_3 Output assigned to bit-3 0x05 BIT_4 Output assigned to bit-4 0x06 BIT_5 Output assigned to bit-5 0x07 BIT_6 Output assigned to bit-6 0x08 BIT_7 Output assigned to bit-7	0x2	RW
	7...4	out_conf_activity	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value** **Name** **Description** 0x00 disable Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 BIT_0 Output assigned to bit-0 0x02 BIT_1 Output assigned to bit-1 0x03 BIT_2 Output assigned to bit-2	0x3	RW

			0x04 BIT_3 Output assigned to bit-3 0x05 BIT_4 Output assigned to bit-4 0x06 BIT_5 Output assigned to bit-5 0x07 BIT_6 Output assigned to bit-6 0x08 BIT_7 Output assigned to bit-7		
wrist_gesture					
0x36		WR_GEST_1	Wrist gesture configuration settings	0x0005	
	3...0	out_conf	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value Name Description** 0x00 disable Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 BIT_0 Output assigned to bit-0 0x02 BIT_1 Output assigned to bit-1 0x03 BIT_2 Output assigned to bit-2 0x04 BIT_3 Output assigned to bit-3 0x05 BIT_4 Output assigned to bit-4 0x06 BIT_5 Output assigned to bit-5 0x07 BIT_6 Output assigned to bit-6 0x08 BIT_7 Output assigned to bit-7	0x5	RW
	4	wearable_arm	Device in left (0) or right (1) arm. By default, the wearable device is assumed to be in left arm i.e. default value is 0.	0x0	RW
	5	enable	Enables the feature	0x0	RW
0x38		WR_GEST_2	Wrist gesture setting	0x06EE	

	15...0	min_flick_peak	Sine of the minimum tilt angle in portrait down direction of the device when wrist is rolled away (roll-out) from user. The configuration parameter is scaled by 2048 i.e. 2048 * sin(angle). Range is 1448 to 1774. Default value is 1774.	0x6EE	RW
0x3A		WR_GEST_3	Wrist gesture setting	0x0004	
	15...0	min_flick_samples	Value of minimum time difference between wrist roll-out and roll-in movement during flick gesture. Range is 3 to 5 samples at 50Hz (i.e. 0.06 to 0.1 seconds). Default value is 4 (i.e. 0.08 seconds).	0x4	RW
0x3C		WR_GEST_4	Wrist gesture setting	0x00C8	
	15...0	max_duration	Maximum time within which gesture movement has to be completed. Range is 150 to 250 samples at 50Hz (i.e. 3 to 5 seconds). Defualt value is 200 (i.e. 4 seconds).	0xC8	RW
Reserved					
0x3E		Reserved	Reserved	0x0000	
	15...0	Reserved	Reserved	0x0	RW

Page 7

Address	Bit	Name	Description	Reset	Access
wrist_wear_wakeup					
0x30		WR_WAKEUP_1	Wrist wear wakeup configuration settings	0x0004	
	3...0	out_conf	Enable bits for enabling output into the register status bits and, if desired, onto the interrupt pin **Value** **Name** **Description** 0x00 disable Output of feature not assigned to any interrupt bits 0..7 of INT_STATUS_0 and INT1/2_MAP_FEAT 0x01 BIT_0 Output assigned to bit-0 0x02 BIT_1 Output assigned to bit-1 0x03 BIT_2 Output assigned to bit-2 0x04 BIT_3 Output assigned to bit-3 0x05 BIT_4 Output assigned to bit-4 0x06 BIT_5 Output assigned to bit-5 0x07 BIT_6 Output assigned to bit-6 0x08 BIT_7 Output assigned to bit-7	0x4	RW
	4	enable	Enables the feature	0x0	RW
0x32		WR_WAKEUP_2	Wrist wear wakeup setting	0x05A8	
	15...0	min_angle_focus	Cosine of minimum expected attitude change of the device within 1 second time window when moving within focus position. The parameter is scaled by 2048 i.e. 2048 * cos(angle). Range is 1024 to 1774. Default is 1448.	0x5A8	RW
0x34		WR_WAKEUP_3	Wrist wear wakeup setting	0x06EE	
	15...0	min_angle_nonfocus	Cosine of minimum expected attitude change of the device within 1 second	0x6EE	RW

			time window when moving from non-focus to focus position. The parameter is scaled by 2048 i.e. 2048 * cos(angle). Range is 1448 to 1856. Default value is 1774.		
0x36		WR_WAKEUP_4	Wrist wear wakeup setting	0x0400	
	15...0	max_tilt_lr	Sine of the maximum allowed downward tilt angle in landscape right direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device). The configuration parameter is scaled by 2048 i.e. 2048 * sin(angle). Range is 700 to 1024. Default value is 1024.	0x400	RW
0x38		WR_WAKEUP_5	Wrist wear wakeup setting	0x02BC	
	15...0	max_tilt_ll	Sine of the maximum allowed downward tilt angle in landscape left direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device). The configuration parameter is scaled by 2048 i.e. 2048 * sin(angle). Range is 700 to 1024. Default value is 700.	0x2BC	RW
0x3A		WR_WAKEUP_6	Wrist wear wakeup setting	0x00B3	
	15...0	max_tilt_pd	Sine of the maximum allowed backward tilt angle in portrait down direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device). The configuration parameter is scaled by 2048 i.e. 2048 * sin(angle). Range is 0 to179. Default value is 179.	0xB3	RW
0x3C		WR_WAKEUP_7	Wrist wear wakeup setting	0x0785	
	15...0	max_tilt_pu	Sine of the maximum allowed forward tilt angle in portrait up direction of the device, when it is in focus position (i.e. user is able to comfortably look at the dial of wear device). The configuration parameter is scaled by 2048 i.e. 2048 * sin(angle). Range is 1774 to 1978. Default value is 1925.	0x785	RW

Reserved					
0x3E		Reserved	Reserved	0x0000	
	15… 0	Reserved	Reserved	0x0	RW

Register (0x40) ACC_CONF

DESCRIPTION: Sets the output data rate, the bandwidth, and the read mode of the acceleration sensor
RESET: 0xA8
DEFINITION (Go to register map):

<table>
<tr><th>Address</th><th>Bit</th><th>Name</th><th>Description</th><th>Reset</th><th>Access</th></tr>
<tr><td>0x40</td><td></td><td>ACC_CONF</td><td></td><td>0xA8</td><td></td></tr>
<tr><td></td><td>3…0</td><td>acc_odr</td><td>ODR in Hz. The output data rate is independent of the power mode setting for the sensor
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0x00</td><td>reserved</td><td>Reserved</td></tr>
<tr><td>0x01</td><td>odr_0p78</td><td>25/32</td></tr>
<tr><td>0x02</td><td>odr_1p5</td><td>25/16</td></tr>
<tr><td>0x03</td><td>odr_3p1</td><td>25/8</td></tr>
<tr><td>0x04</td><td>odr_6p25</td><td>25/4</td></tr>
<tr><td>0x05</td><td>odr_12p5</td><td>25/2</td></tr>
<tr><td>0x06</td><td>odr_25</td><td>25</td></tr>
<tr><td>0x07</td><td>odr_50</td><td>50</td></tr>
<tr><td>0x08</td><td>odr_100</td><td>100</td></tr>
<tr><td>0x09</td><td>odr_200</td><td>200</td></tr>
<tr><td>0x0a</td><td>odr_400</td><td>400</td></tr>
<tr><td>0x0b</td><td>odr_800</td><td>800</td></tr>
<tr><td>0x0c</td><td>odr_1k6</td><td>1600</td></tr>
<tr><td>0x0d</td><td>odr_3k2</td><td>Reserved</td></tr>
<tr><td>0x0e</td><td>odr_6k4</td><td>Reserved</td></tr>
<tr><td>0x0f</td><td>odr_12k8</td><td>Reserved</td></tr>
</table></td><td>0x8</td><td>RW</td></tr>
<tr><td></td><td>6…4</td><td>acc_bwp</td><td>Bandwidth parameter determines filter configuration (acc_filt_perf=1) and averaging for undersampling mode (acc_filt_perf=0)
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0x00</td><td>osr4_avg1</td><td>acc_filt_perf = 1 -> OSR4 mode;
acc_filt_perf = 0 -> no averaging</td></tr>
<tr><td>0x01</td><td>osr2_avg2</td><td>acc_filt_perf = 1 -> OSR2 mode;
acc_filt_perf = 0 -> average 2 samples</td></tr>
<tr><td>0x02</td><td>norm_avg4</td><td>acc_filt_perf = 1 -> normal mode;
acc_filt_perf = 0 -> average 4 samples</td></tr>
<tr><td>0x03</td><td>cic_avg8</td><td>acc_filt_perf = 1 -> CIC mode;
acc_filt_perf = 0 -> average 8 samples</td></tr>
</table></td><td>0x2</td><td>RW</td></tr>
</table>

			0x04 res_avg16 acc_filt_perf = 1 -> Reserved; acc_filt_perf = 0 -> average 16 samples 0x05 res_avg32 acc_filt_perf = 1 -> Reserved; acc_filt_perf = 0 -> average 32 samples 0x06 res_avg64 acc_filt_perf = 1 -> Reserved; acc_filt_perf = 0 -> average 64 samples 0x07 res_avg128 acc_filt_perf = 1 -> Reserved; acc_filt_perf = 0 -> average 128 samples		
	7	acc_filter_perf	Select accelerometer filter performance mode: **Value** **Name** **Description** 0x00 ulp power optimized 0x01 hp performance opt.	0x1	RW

Register (0x41) ACC_RANGE

DESCRIPTION: Selection of the Accelerometer g-range
RESET: 0x02
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x41		ACC_RANGE		0x02	
	1...0	acc_range	Accelerometer g-range **Value** **Name** **Description** 0x00 range_2g +/-2g 0x01 range_4g +/-4g 0x02 range_8g +/-8g 0x03 range_16g +/-16g	0x2	RW

Register (0x42) GYR_CONF

DESCRIPTION: Sets the output data rate and the bandwidth of the Gyroscope in the sensor
RESET: 0xA9
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x42		GYR_CONF		0xA9	
	3...0	gyr_odr	ODR in Hz **Value** **Name** **Description** 0x00 reserved Reserved 0x01 odr_0p78 Reserved 0x02 odr_1p5 Reserved 0x03 odr_3p1 Reserved 0x04 odr_6p25 Reserved 0x05 odr_12p5 Reserved 0x06 odr_25 25 0x07 odr_50 50 0x08 odr_100 100 0x09 odr_200 200 0x0a odr_400 400 0x0b odr_800 800 0x0c odr_1k6 1600 0x0d odr_3k2 3200 0x0e odr_6k4 Reserved 0x0f odr_12k8 Reserved	0x9	RW
	5...4	gyr_bwp	The Gyroscope bandwidth coefficient defines the 3 dB cutoff frequency of the low pass filter for the sensor data **Value** **Name** **Description** 0x00 osr4 OSR4 mode 0x01 osr2 OSR2 mode 0x02 norm normal mode 0x03 res reserved	0x2	RW
	6	gyr_noise_perf	Select noise performance: **Value** **Name** **Description** 0x00 ulp power optimized 0x01 hp performance opt.	0x0	RW
	7	gyr_filter_perf	Select gyroscope filter performance mode: **Value** **Name** **Description** 0x00 ulp power optimized 0x01 hp performance opt.	0x1	RW

Register (0x43) GYR_RANGE

DESCRIPTION: Defines the Gyroscope angular rate measurement range
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x43		GYR_RANGE		0x00	
	2...0	gyr_range	Full scale, Resolution **Value** **Name** **Description** 0x00 range_2000 +/-2000dps, 16.4 LSB/dps 0x01 range_1000 +/-1000dps, 32.8 LSB/dps 0x02 range_500 +/-500dps, 65.6 LSB/dps 0x03 range_250 +/-250dps, 131.2 LSB/dps 0x04 range_125 +/-125dps, 262.4 LSB/dps	0x0	RW
	3	ois_range	Full scale, Resolution **Value** **Name** **Description** 0x00 range_250 +/-250dps, 131.2 LSB/dps 0x01 range_2000 +/-2000dps, 16.4 LSB/dps	0x0	RW

Register (0x44) AUX_CONF

DESCRIPTION: Sets the output data rate of the Auxiliary sensor interface
RESET: 0x46
DEFINITION (Go to register map):

<table>
<tr><th>Address</th><th>Bit</th><th>Name</th><th>Description</th><th>Reset</th><th>Access</th></tr>
<tr><td>0x44</td><td></td><td>AUX_CONF</td><td></td><td>0x46</td><td></td></tr>
<tr><td></td><td>3...0</td><td>aux_odr</td><td>Define the poll rate for the magnetometer attached to the Auxiliary sensor interface. This is independent of the power mode setting for the sensor. The output data rate in Hz. In addition to setting the poll rate, it is required to configure the Auxiliary sensor properly using the MAG_IF register.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0x00</td><td>reserved</td><td>Reserved</td></tr>
<tr><td>0x01</td><td>odr_0p78</td><td>25/32</td></tr>
<tr><td>0x02</td><td>odr_1p5</td><td>25/16</td></tr>
<tr><td>0x03</td><td>odr_3p1</td><td>25/8</td></tr>
<tr><td>0x04</td><td>odr_6p25</td><td>25/4</td></tr>
<tr><td>0x05</td><td>odr_12p5</td><td>25/2</td></tr>
<tr><td>0x06</td><td>odr_25</td><td>25</td></tr>
<tr><td>0x07</td><td>odr_50</td><td>50</td></tr>
<tr><td>0x08</td><td>odr_100</td><td>100</td></tr>
<tr><td>0x09</td><td>odr_200</td><td>200</td></tr>
<tr><td>0x0a</td><td>odr_400</td><td>400</td></tr>
<tr><td>0x0b</td><td>odr_800</td><td>800</td></tr>
<tr><td>0x0c</td><td>odr_1k6</td><td>Reserved</td></tr>
<tr><td>0x0d</td><td>odr_3k2</td><td>Reserved</td></tr>
<tr><td>0x0e</td><td>odr_6k4</td><td>Reserved</td></tr>
<tr><td>0x0f</td><td>odr_12k8</td><td>Reserved</td></tr>
</table></td><td>0x6</td><td>RW</td></tr>
<tr><td></td><td>7...4</td><td>aux_offset</td><td>trigger-readout offset in units of 2.5 ms. If set to zero, the offset is maximum, i.e. after readout a trigger is issued immediately.</td><td>0x4</td><td>RW</td></tr>
</table>

Register (0x45) FIFO_DOWNS

DESCRIPTION: Configure Gyroscope and Accelerometer downsampling rates for FIFO
RESET: 0x88
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x45		FIFO_DOWNS		0x88	
	2...0	gyr_fifo_downs	Downsampling for Gyroscope (2**downs_gyro)	0x0	RW
	3	gyr_fifo_filt_data	selects filtered or unfiltered Gyroscope data for fifo **Value** **Name** **Description** 0x00 unfiltered Unfiltered data 0x01 filtered Filtered data	0x1	RW
	6...4	acc_fifo_downs	Downsampling for Accelerometer (2**downs_accel)	0x0	RW
	7	acc_fifo_filt_data	selects filtered or unfiltered Accelerometer data for fifo **Value** **Name** **Description** 0x00 unfiltered Unfiltered data 0x01 filtered Filtered data	0x1	RW

Register (0x46) FIFO_WTM_0

DESCRIPTION: FIFO Watermark level LSB
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x46		FIFO_WTM_0		0x00	
	7...0	fifo_water_mark_7_0	Trigger an interrupt when FIFO contains fifo_water_mark_7_0+fifo_water_mark_12_8*256 bytes	0x0	RW

Register (0x47) FIFO_WTM_1

DESCRIPTION: FIFO Watermark level MSB and frame content configuration
RESET: 0x02
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x47		FIFO_WTM_1		0x02	
	4...0	fifo_water_mark_12_8	Trigger an interrupt when FIFO contains fifo_water_mark_7_0+fifo_water_mark_12_8*256 bytes	0x2	RW

Register (0x48) FIFO_CONFIG_0

DESCRIPTION: FIFO frame content configuration
RESET: 0x02
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x48		FIFO_CONFIG_0		0x02	
	0	fifo_stop_on_full	Stop writing samples into FIFO when FIFO is full. **Value** **Name** **Description** 0x00 disable do not stop writing to FIFO when full 0x01 enable Stop writing into FIFO when full.	0x0	RW
	1	fifo_time_en	Return sensortime frame after the last valid data frame. **Value** **Name** **Description** 0x00 disable do not return sensortime frame 0x01 enable return sensortime frame	0x1	RW

Register (0x49) FIFO_CONFIG_1

DESCRIPTION: FIFO frame content configuration
RESET: 0x10
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x49		FIFO_CONFIG_1		0x10	
	1…0	fifo_tag_int1_en	FIFO interrupt 1 tag enable **Value** **Name** **Description** 0x00 int_edge enable tag on rising edge of int pin 0x01 int_level enable tag on level value of int pin 0x02 acc_sat enable tag on saturation of accelerometer data 0x03 gyr_sat enable tag on saturation of gyroscope data	0x0	RW
	3…2	fifo_tag_int2_en	FIFO interrupt 2 tag enable **Value** **Name** **Description** 0x00 int_edge enable tag on rising edge of int pin 0x01 int_level enable tag on level value of int pin 0x02 acc_sat enable tag on saturation of accelerometer data 0x03 gyr_sat enable tag on saturation of gyroscope data	0x0	RW
	4	fifo_header_en	FIFO frame header enable **Value** **Name** **Description** 0x00 disable no header is stored (output data rate of all enabled sensors need to be identical) 0x01 enable header is stored	0x1	RW
	5	fifo_aux_en	Store Auxiliary sensor data in FIFO (all 3 axes) **Value** **Name** **Description** 0x00 disable no Auxiliary sensor data is stored 0x01 enable Auxiliary sensor data is stored	0x0	RW
	6	fifo_acc_en	Store Accelerometer data in FIFO (all 3 axes) **Value** **Name** **Description**	0x0	RW

			0x00 disable no Accelerometer data is stored 0x01 enable Accelerometer data is stored		
	7	fifo_gyr_en	Store Gyroscope data in FIFO (all 3 axes) **Value Name Description** 0x00 disable no Gyroscope data is stored 0x01 enable Gyroscope data is stored	0x0	RW

Register (0x4A) SATURATION

DESCRIPTION: Contains the information if one of the raw data samples used to generate current filtered data sample has been saturated (reached 0x8001 or 0x7FFF).
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4A		SATURATION		0x00	
	0	acc_x	ACC X-axis raw data saturation flag.	0x0	R
	1	acc_y	ACC Y-axis raw data saturation flag.	0x0	R
	2	acc_z	ACC Z-axis raw data saturation flag.	0x0	R
	3	gyr_x	GYR X-axis raw data saturation flag.	0x0	R
	4	gyr_y	GYR Y-axis raw data saturation flag.	0x0	R
	5	gyr_z	GYR Z-axis raw data saturation flag.	0x0	R

Register (0x4B) AUX_DEV_ID

DESCRIPTION: Auxiliary interface device_id
RESET: 0x20
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4B		AUX_DEV_ID		0x20	
	7...1	i2c_device_addr	I2C device address of Auxiliary sensor	0x10	RW

Register (0x4C) AUX_IF_CONF

DESCRIPTION: Auxiliary interface configuration register
RESET: 0x83
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4C		AUX_IF_CONF		0x83	
	1...0	aux_rd_burst	Burst data length (1,2,6,8 byte) **Value Name Description** 0x00 BL1 Burst length 1 0x01 BL2 Burst length 2 0x02 BL6 Burst length 6 0x03 BL8 Burst length 8	0x3	RW
	3...2	man_rd_burst	Manual burst data length (1,2,6,8 byte) **Value Name Description** 0x00 BL1 Burst length 1 0x01 BL2 Burst length 2 0x02 BL6 Burst length 6 0x03 BL8 Burst length 8	0x0	RW
	6	aux_fcu_write_en	enables FCU write command on AUX IF for auxiliary sensors that need a trigger.	0x0	RW
	7	aux_manual_en	switches auxiliary interface between automatic and manual mode. In manual mode all read and write operations on auxiliary interface must be triggered manually; in automatic mode (aux_manual_en = "0") FCU triggers read and write operations periodically (as programmed by user).	0x1	RW

Register (0x4D) AUX_RD_ADDR

DESCRIPTION: Auxiliary interface read address
RESET: 0x42
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4D		AUX_RD_ADDR		0x42	
	7...0	read_addr	Address to read. In manual mode it triggers the read operation.	0x42	RW

Register (0x4E) AUX_WR_ADDR

DESCRIPTION: Auxiliary interface write address
RESET: 0x4C
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4E		AUX_WR_ADDR		0x4C	
	7...0	write_addr	Address to write. In manual mode it triggers the write operation.	0x4C	RW

Register (0x4F) AUX_WR_DATA

DESCRIPTION: Auxiliary interface write data
RESET: 0x02
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x4F		AUX_WR_DATA		0x02	
	7...0	write_data	Data to write	0x2	RW

Register (0x52) ERR_REG_MSK

DESCRIPTION: Defines which error flag will trigger the error interrupt once enabled
'1' - use to generate the error interrupt
'0' - do not use to generate error interrupt
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x52		ERR_REG_MSK		0x00	
	0	fatal_err	Use fatal error to generate the error interrupt.	0x0	RW
	4...1	internal_err	Use internal error to generate the error interrupt	0x0	RW
	6	fifo_err	Use fifo error to generate the error interrupt.	0x0	RW
	7	aux_err	Use aux interface error to generate the error interrupt.	0x0	RW

Register (0x53) INT1_IO_CTRL

DESCRIPTION: Configure the electrical behavior of the interrupt pin INT1
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x53		INT1_IO_CTRL		0x00	
	1	lvl	Configure level of INT1 pin **Value** **Name** **Description** 0x00 active_low active low 0x01 active_high active high	0x0	RW
	2	od	Configure behaviour of INT1 pin **Value** **Name** **Description** 0x00 push_pull push-pull 0x01 open_drain open drain	0x0	RW
	3	output_en	Output enable for INT1 pin **Value** **Name** **Description** 0x00 off Output disabled 0x01 on Output enabled	0x0	RW
	4	input_en	Input enable for INT1 pin **Value** **Name** **Description** 0x00 off Input disabled 0x01 on Input enabled	0x0	RW

Register (0x54) INT2_IO_CTRL

DESCRIPTION: Configure the electrical behavior of the interrupt pin INT2
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x54		INT2_IO_CTRL		0x00	
	1	lvl	Configure level of INT2 pin **Value** **Name** **Description** 0x00 active_low active low 0x01 active_high active high	0x0	RW
	2	od	Configure behaviour of INT2 pin **Value** **Name** **Description** 0x00 push_pull push-pull 0x01 open_drain open drain	0x0	RW
	3	output_en	Output enable for INT2 pin **Value** **Name** **Description** 0x00 off Output disabled 0x01 on Output enabled	0x0	RW
	4	input_en	Input enable for INT2 pin **Value** **Name** **Description** 0x00 off Input disabled 0x01 on Input enabled	0x0	RW

Register (0x55) INT_LATCH

DESCRIPTION: Configure interrupt modes
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x55		INT_LATCH		0x00	
	0	int_latch	Latched/non-latched interrupt modes **Value** **Name** **Description** 0x00 none non latched 0x01 permanent permanent latched	0x0	RW

Register (0x56) INT1_MAP_FEAT

DESCRIPTION: Interrupt/Feature mapping on INT1
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x56		INT1_MAP_FEAT		0x00	
	0	sig_motion_out	Sigmotion output.	0x0	RW
	1	step_counter_out	Step-counter watermark or Step-detector output	0x0	RW
	2	activity_out	Step activity output	0x0	RW
	3	wrist_wear_wakeup_out	Wrist wear wakeup output	0x0	RW
	4	wrist_gesture_out	Wrist gesture output	0x0	RW
	5	no_motion_out	No motion detection output	0x0	RW
	6	any_motion_out	Any motion detection output	0x0	RW
	7	reserved	Reserved	0x0	RW

Register (0x57) INT2_MAP_FEAT

DESCRIPTION: Interrupt/Feature mapping on INT2
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x57		INT2_MAP_FEAT		0x00	
	0	sig_motion_out	Sigmotion output	0x0	RW
	1	step_counter_out	Step-counter watermark or Step-detector output	0x0	RW
	2	activity_out	Step activity output	0x0	RW
	3	wrist_wear_wakeup_out	Wrist wear wakeup output	0x0	RW
	4	wrist_gesture_out	Wrist gesture output	0x0	RW
	5	no_motion_out	No motion detection output	0x0	RW
	6	any_motion_out	Any motion detection output	0x0	RW
	7	reserved	Reserved	0x0	RW

Register (0x58) INT_MAP_DATA

DESCRIPTION: Data Interrupt mapping for both INT pins
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x58		INT_MAP_DATA		0x00	
	0	ffull_int1	FIFO Full interrupt mapped to INT1	0x0	RW
	1	fwm_int1	FIFO Watermark interrupt mapped to INT1	0x0	RW
	2	drdy_int1	Data Ready interrupt mapped to INT1	0x0	RW
	3	err_int1	Error interrupt mapped to INT1	0x0	RW
	4	ffull_int2	FIFO Full interrupt mapped to INT2	0x0	RW
	5	fwm_int2	FIFO Watermark interrupt mapped to INT2	0x0	RW
	6	drdy_int2	Data Ready interrupt mapped to INT2	0x0	RW
	7	err_int2	Error interrupt mapped to INT2	0x0	RW

Register (0x59) INIT_CTRL

DESCRIPTION: Start initialization
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x59		INIT_CTRL		0x00	
	7...0	init_ctrl	Start initialization	0x0	RW

Register (0x5B) INIT_ADDR_0

DESCRIPTION: Base address of the initialization data. Increment by burst write length in bytes/2 after each burst write operation. Please ignore, if your host supports to load the initialization data in a single 8kB burst write operation.
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x5B		INIT_ADDR_0		0x00	
	3...0	base_0_3	Bits 0 to 3 of the base address for initialization data.	0x0	RW

Register (0x5C) INIT_ADDR_1

DESCRIPTION: Base address of the initialization data. Increment by burst write length in bytes/2 after each burst write operation. Please ignore, if your host supports to load the initialization data in a single 8kB burst write operation.
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x5C		INIT_ADDR_1		0x00	
	7...0	base_11_4	Bits 4 to 11 of the base address for initialization data.	0x0	RW

Register (0x5E) INIT_DATA

DESCRIPTION: Initialization register
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x5E		INIT_DATA		0x00	
	7...0	data	Register for initialization data	0x0	RW

Register (0x5F) INTERNAL_ERROR

DESCRIPTION: Internal error flags. Value of all reserved bits should be ignored.
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x5F		INTERNAL_ERROR		0x00	
	1	int_err_1	Internal error flag - long processing time, processing halted	0x0	R
	2	int_err_2	Internal error flag - fatal error, processing halted	0x0	R
	4	feat_eng_disabled	Feature engine has been disabled by host during sensor operation	0x0	R

Register (0x68) AUX_IF_TRIM

DESCRIPTION: Auxiliary interface trim register (NVM backed)
RESET: 0x01
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x68		AUX_IF_TRIM		0x01	
	1...0	asda_pupsel	Pullup configuration for ASDA **Value Name Description** 0x00 pup_res_off Pullup off 0x01 pup_res_40k Pullup 40k 0x02 pup_res_10k Pullup 10k 0x03 pup_res_2k Pullup 2k	0x1	RW
	2	spare3	(Spare NVM bits.)	0x0	RW

Register (0x69) GYR_CRT_CONF

DESCRIPTION: Component Retrimming for Gyroscope
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x69		GYR_CRT_CONF		0x00	
	2	crt_running	Indicates that CRT is currently running. If CRT completed, check CRT_STATUS register for the completion status **Value Name Description** 0x00 disabled disabled 0x01 enabled enabled	0x0	RW
	3	rdy_for_dl	pacemaker bit for downloading the CRT data **Value Name Description** 0x00 ongoing ongoing or not started 0x01 complete complete	0x0	R

Register (0x6A) NVM_CONF

DESCRIPTION: NVM Configuration
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x6A		NVM_CONF		0x00	
	1	nvm_prog_en	Enable NVM programming. **Value** **Name** **Description** 0x00 disable disable 0x01 enable enable	0x0	RW

Register (0x6B) IF_CONF

DESCRIPTION: Serial interface settings
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x6B		IF_CONF		0x00	
	0	spi3	Configure SPI Interface Mode for primary interface **Value** **Name** **Description** 0x00 spi4 SPI 4-wire mode 0x01 spi3 SPI 3-wire mode	0x0	RW
	1	spi3_ois	Configure SPI Interface Mode for OIS interface (if enabled) **Value** **Name** **Description** 0x00 spi4 SPI 4-wire mode 0x01 spi3 SPI 3-wire mode	0x0	RW
	4	ois_en	Interface configuration - OIS enable bit. It has lower priority than aux_en.	0x0	RW
	5	aux_en	Interface configuration - AUX enable bit. It has higher priority than ois_en.	0x0	RW

Register (0x6C) DRV

DESCRIPTION: Drive strength control register (NVM backed)
RESET: 0xFF
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x6C		DRV		0xFF	
	2...0	io_pad_drv1	Output pad drive strength setting.	0x7	RW
	3	io_pad_i2c_b1	Output pad drive strength setting.	0x1	RW
	6...4	io_pad_drv2	Output pad drive strength setting.	0x7	RW
	7	io_pad_i2c_b2	Output pad drive strength setting.	0x1	RW

Register (0x6D) ACC_SELF_TEST

DESCRIPTION: Settings for the accelerometer self-test configuration and trigger
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x6D		ACC_SELF_TEST		0x00	
	0	acc_self_test_en	Enable accelerometer self-test **Value** **Name** **Description** 0x00 disabled disabled 0x01 enabled enabled	0x0	RW
	2	acc_self_test_sign	select sign of self-test excitation as **Value** **Name** **Description** 0x00 negative negative 0x01 positive positive	0x0	RW
	3	acc_self_test_amp	select amplitude of the selftest deflection: **Value** **Name** **Description** 0x00 low low 0x01 high high	0x0	RW

Register (0x6E) GYR_SELF_TEST_AXES

DESCRIPTION: Settings for the gyroscope AXES self-test configuration and trigger
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x6E		GYR_SELF_TEST_AXES		0x00	
	0	gyr_st_axes_done	STATUS: functional test of detection channels finished.	0x0	R
	1	gyr_axis_x_ok	status of gyro X-axis self test	0x0	R
	2	gyr_axis_y_ok	status of gyro Y-axis self test	0x0	R
	3	gyr_axis_z_ok	status of gyro Z-axis self test	0x0	R

Register (0x70) NV_CONF

DESCRIPTION: NVM backed configuration bits.
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x70		NV_CONF		0x00	
	0	spi_en	disable the I2C and enable SPI for the primary interface, when it is in autoconfig mode **Value** **Name** **Description** 0x00 disabled I2C enabled 0x01 enabled I2C disabled	0x0	RW
	1	i2c_wdt_sel	Select timer period for I2C Watchdog **Value** **Name** **Description** 0x00 short I2C watchdog timeout after 1.25 ms 0x01 long I2C watchdog timeout after 40 ms	0x0	RW
	2	i2c_wdt_en	I2C Watchdog at the SDI pin in I2C interface mode **Value** **Name** **Description** 0x00 Disable Disable I2C watchdog 0x01 Enable Enable I2C watchdog	0x0	RW
	3	acc_off_en	Add the offset defined in the off_acc_[xyz] OFFSET register to filtered and unfiltered Accelerometer data **Value** **Name** **Description** 0x00 disabled Disabled 0x01 enabled Enabled	0x0	RW

Register (0x71) OFFSET_0

DESCRIPTION: Offset compensation for Accelerometer X-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x71		OFFSET_0		0x00	
	7…0	off_acc_x	Accelerometer offset compensation (X-axis).	0x0	RW

Register (0x72) OFFSET_1

DESCRIPTION: Offset compensation for Accelerometer Y-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x72		OFFSET_1		0x00	
	7...0	off_acc_y	Accelerometer offset compensation (Y-axis).	0x0	RW

Register (0x73) OFFSET_2

DESCRIPTION: Offset compensation for Accelerometer Z-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x73		OFFSET_2		0x00	
	7...0	off_acc_z	Accelerometer offset compensation (Z-axis).	0x0	RW

Register (0x74) OFFSET_3

DESCRIPTION: Offset compensation for Gyroscope X-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x74		OFFSET_3		0x00	
	7...0	gyr_usr_off_x_7_0	Gyroscope offset compensation (X-axis).	0x0	RW

Register (0x75) OFFSET_4

DESCRIPTION: Offset compensation for Gyroscope Y-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x75		OFFSET_4		0x00	
	7...0	gyr_usr_off_y_7_0	Gyroscope offset compensation (Y-axis).	0x0	RW

Register (0x76) OFFSET_5

DESCRIPTION: Offset compensation for Gyroscope Z-axis
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x76		OFFSET_5		0x00	
	7...0	gyr_usr_off_z_7_0	Gyroscope offset compensation (Z-axis).	0x0	RW

Register (0x77) OFFSET_6

DESCRIPTION: Offset compensation (MSBs gyroscope, enables)
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x77		OFFSET_6		0x00	
	1...0	gyr_usr_off_x_9_8	Gyroscope offset compensation (X-axis).	0x0	RW
	3...2	gyr_usr_off_y_9_8	Gyroscope offset compensation (Y-axis).	0x0	RW
	5...4	gyr_usr_off_z_9_8	Gyroscope offset compensation (Z-axis).	0x0	RW
	6	gyr_off_en	Add the offset defined in the gyr_usr_off_[xyz] OFFSET register to filtered and unfiltered Gyroscope data **Value** **Name** **Description** 0x00 disabled Disabled 0x01 enabled Enabled	0x0	RW
	7	gyr_gain_en	Compensate the gain defined in the gyr_usr_gain_[xyz] GAIN register to filtered and unfiltered Gyroscope data **Value** **Name** **Description** 0x00 disabled Disabled 0x01 enabled Enabled	0x0	RW

Register (0x7C) PWR_CONF

DESCRIPTION: Power mode configuration register
RESET: 0x03
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x7C		PWR_CONF		0x03	
	0	adv_power_save	Advanced power save disabled (fast clk always enabled). **Value** **Name** **Description** 0x00 aps_off Advanced power save disabled (fast clk always enabled). 0x01 aps_on Advanced power mode enabled (slow clk is active when no measurement is ongoing.)	0x1	RW
	1	fifo_self_wake_up	FIFO read disabled in low power mode **Value** **Name** **Description** 0x00 fsw_off FIFO read disabled in low power mode 0x01 fsw_on FIFO read enabled in low power mode after FIFO interrupt is fired	0x1	RW
	2	fup_en	Fast power up enable **Value** **Name** **Description** 0x00 fup_off Fast power up disabled 0x01 fup_on Fast power up enabled	0x0	RW

Register (0x7D) PWR_CTRL

DESCRIPTION: Power mode control register
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x7D		PWR_CTRL		0x00	
	0	aux_en	**Value** **Name** **Description** 0x00 aux_off Disables the Auxiliary sensor. 0x01 aux_on Enables the Auxiliary sensor.	0x0	RW
	1	gyr_en	**Value** **Name** **Description** 0x00 gyr_off Disables the Gyroscope. 0x01 gyr_on Enables the Gyroscope.	0x0	RW
	2	acc_en	**Value** **Name** **Description** 0x00 acc_off Disables the Accelerometer. 0x01 acc_on Enables the Accelerometer.	0x0	RW
	3	temp_en	**Value** **Name** **Description** 0x00 temp_off Disables the Temperature sensor. 0x01 temp_on Enables the Temperature sensor.	0x0	RW

Register (0x7E) CMD

DESCRIPTION: Command Register
RESET: 0x00
DEFINITION (Go to register map):

Address	Bit	Name	Description	Reset	Access
0x7E		CMD		0x00	
	7...0	cmd	Available commands (Note: Register will always return 0x00 as read result): **Value** **Name** **Description** 0x02 g_trigger Trigger special gyro operations. 0x03 usr_gain Starts user defined command 0. 0xa0 nvm_prog Writes the NVM backed registers into NVM 0xb0 fifo_flush Clears FIFO content 0xb6 softreset Triggers a reset, all user configuration settings are overwritten with their default state	0x0	W

6. Digital Interfaces

6.1 . Interfaces

Beside the standard primary interface (I2C and SPI configurable), where sensor acts as a slave to the application processor the IMU device supports a secondary interface. The secondary interface can be configured as either auxiliary interface (I2C master) or OIS interface (SPI slave). See picture below. Both secondary configurations work independent of the primary interface configuration, i.e. I2C or SPI between the device and application processor.

If the secondary interface is configured as an auxiliary interface, the device can be connected to an external sensor (e.g. a magnetometer) in order to build a 9-DoF solution. Then the device will act as a master to the external sensor, reading the sensor data automatically and providing it to the application processor via the primary interface.

Alternatively, the secondary interface can be used as OIS interface to connect to an external OIS control unit. The OIS control unit acts as a master and device as slave.

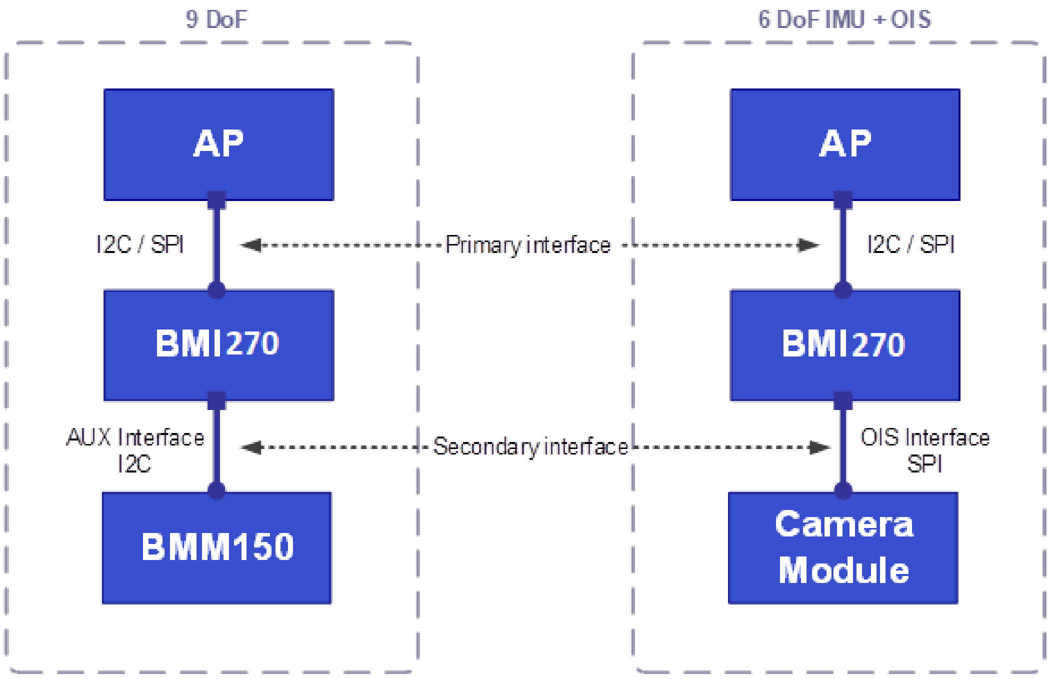


Figure 9: Digital Interfaces

6.2. Primary Interface

By default, the device operates in I2C mode. The device interface can also be configured to operate in a SPI 4-wire configuration. 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 the following table:

Table 17: Mapping for primary interface

PIN#	NAME	I/O TYPE	DESCRIPTION	CONNECT TO PRIMARY INTERFACE		
				IN SPI4W	IN SPI3W	IN I2C
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 addr.
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 (preferred) or DNC
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 SDI serial data input in SPI 4W SDA serial data I/O in SPI 3W	SDI	SDA	SDA

* INT1 and/or INT2 can also be configured as an input in case the external data synchronization in FIFO is used. If INT1 and/or INT2 are not used, please do not connect them (DNC).

The following table shows the electrical specifications of the interface pins:
Note: Values are target values and will be verified during characterization.

Table 18: Electrical specifications of the interface pins

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Pull-up Resistance, CSB pin	R_{up}	Internal Pull-up Resistance to VDDIO	75	100	140	kΩ
Input Capacitance	C_{in}				5	pF
I²C Bus Load Capacitance (max. drive capability)	C_{I2C_Load}				400	pF

6.3. Primary Interface Digital Protocol Selection

The protocol is automatically selected based on the chip select CSB pin behavior after power-up.

After reset / power-up, device's primary interface is in I2C mode. If CSB is connected to VDDIO during power-up and not changed, the primary interface works in I2C mode. For using I2C, it is recommended to hard-wire the CSB line to VDDIO. Since power-on-reset is only executed when, both VDD and VDDIO are established, there is no risk of incorrect protocol detection due to power-up sequence.

If CSB sees a rising edge after power-up, the device interface switches to SPI until a reset or the next power-up occurs. Therefore, a CSB rising edge is needed before starting the SPI communication. Hence, it is recommended to perform a SPI single read of register CHIP_ID (the obtained value will be invalid) before the actual communication start, in order to use the SPI interface.
If toggling of the CSB bit is not possible without data communication, there is an addition the spi_en bit in register NV_CONF, which can be used to permanently set the primary interface to SPI without the need to toggle the CSB pin at every power-up or reset.

6.4. Primary Interface SPI

The timing specification for SPI of the device is given in the following table:
Note: values are target values and will be verified during characterization.

SPI timing, valid at $V_{DDIO} \geq 1.62V$

Table 19: Timing specifications for SPI

PARAMETER	SYMBOL	CONDITION	MIN	MAX	UNITS
Clock Frequency	f_{SPI}	Max. Load on SDI or SDO = 30pF, $V_{DDIO} \geq 1.62$ V		10	MHz
		$V_{DDIO} < 1.62V$		7	MHz
SCK Low Pulse	t_{SCKL}	$V_{DDIO}>=1.62V$	45		ns
SCK High Pulse	t_{SCKH}	$V_{DDIO}>=1.62V$	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
Idle time after write and read access in normal mode or fast startup mode	$t_{IDLE\_wr\_act}$		2		µs
Idle time between write accesses in suspend mode, low-power mode	$t_{IDLE\_wacc\_sum}$		450		µs

The following figure shows the definition of the SPI timings:

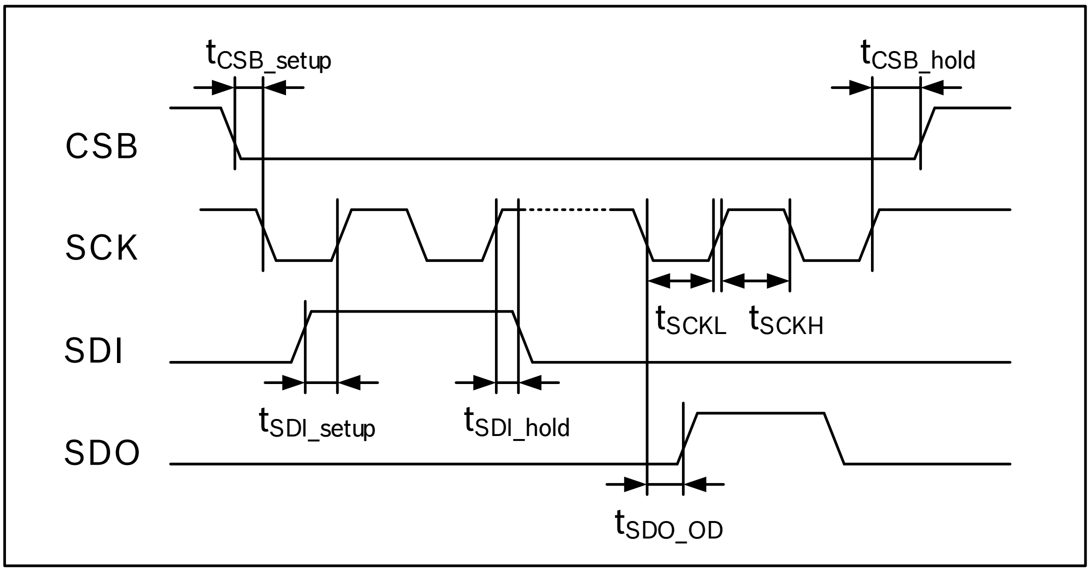


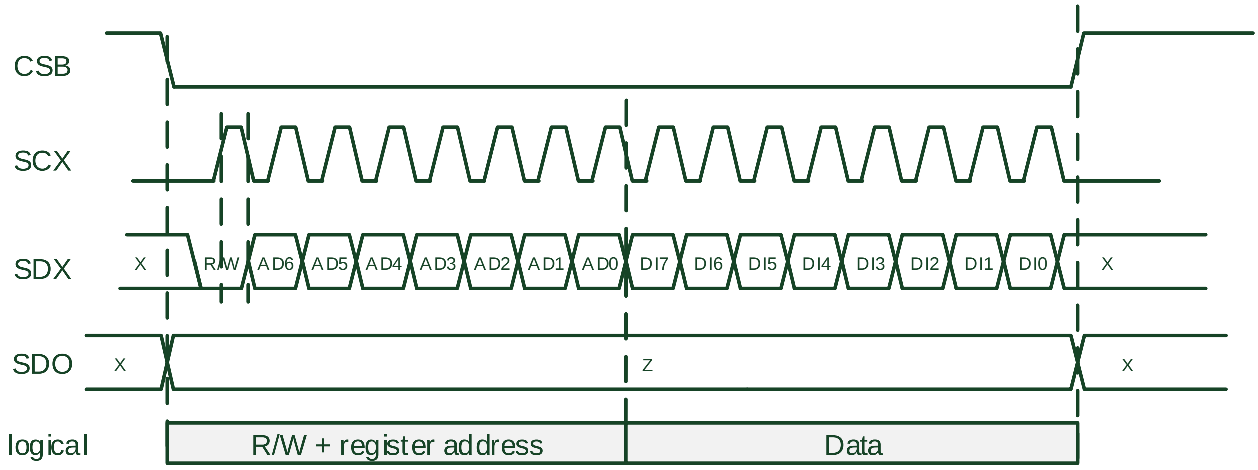
Figure 10: SPI timing diagram

The SPI interface of the device is compatible with two modes, ´00´ [CPOL = ´0´ and CPHA = ´0´] and ´11´ [CPOL = ´1´ and CPHA = ´1´]. The automatic selection between ´00´ and ´11´ is controlled based on the value of SCK after a falling edge of CSB.

Two configurations of the SPI interface are supported by device: 4-wire and 3-wire. The same protocol is used by both configurations. The device operates in 4-wire configuration by default. It can be switched to 3-wire configuration by writing IF_CONF.spi3 = 0b1. Pin SDX is used as the common data pin in 3-wire configuration.
For single byte read as well as write operations, 16-bit protocols are used. device also supports multiple-byte read and write operations.

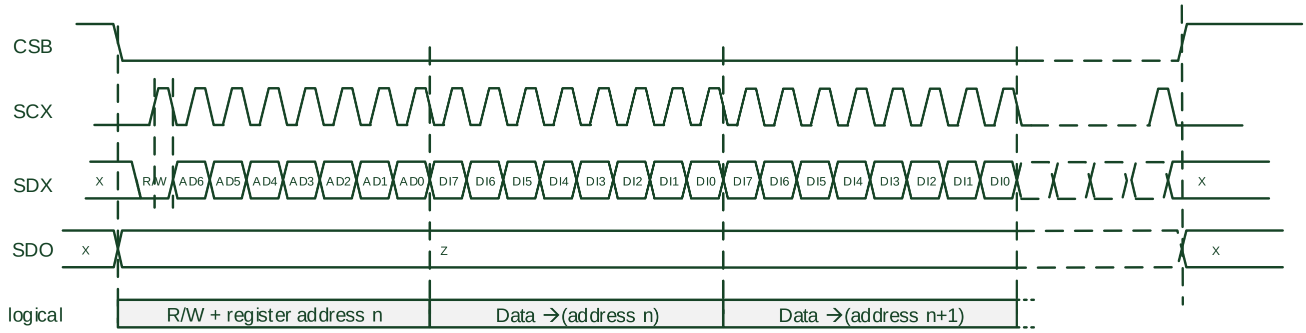
In SPI 4-wire configuration CSB (chip select low active), SCX (as SCK for serial clock), SDX (as SDI for serial data input), and SDO (serial data output) pins are used. The communication starts when the CSB is pulled low by the SPI master and stops when CSB is pulled high. SCK is also controlled by SPI master. SDI and SDO are driven at the falling edge of SCK and should be captured at the rising edge of SCK.

The basic write operation waveform for 4-wire configuration is depicted in the following figure. During the entire write cycle SDO remains in high-impedance state.



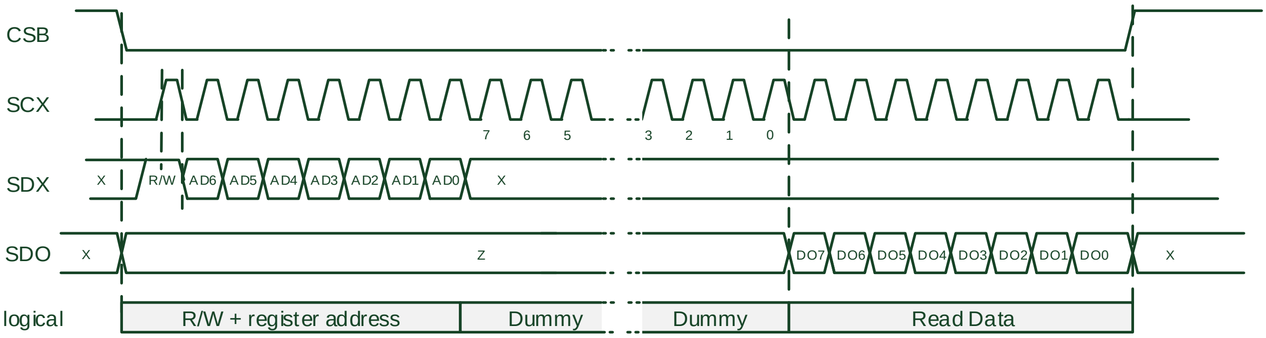
4-wire basic SPI write sequence (mode ´00´)

Multiple write operations are possible by keeping CSB low and continuing the data transfer. Only the first register's address has to be placed in 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 below:



SPI multiple write

The basic read operation waveform for 4-wire configuration is depicted in the figure below. Please note that the first byte received from the device via the SDO line correspond to a dummy byte and the 2nd byte correspond to the value read out of the specified register address. That means, for a basic read operation two bytes have to be read and the first has to be dropped and the second byte must be interpreted.



4-wire basic SPI read sequence (mode ´00´)

The data bits are used as follows:

R/W: Read/Write bit. When 0, the data SDI is written into the chip. When 1, the data SDO from the chip is read.

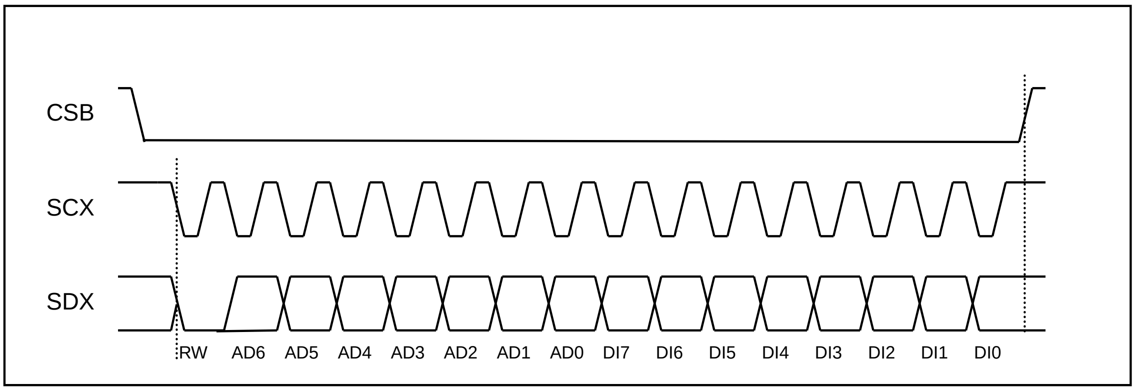
AD6-AD0: Register address

DI7-DI0: When in write mode, these are the data SDI, which will be written into the address.
DO7-DO0: When in read mode, these are the data SDO, which are read from the address.

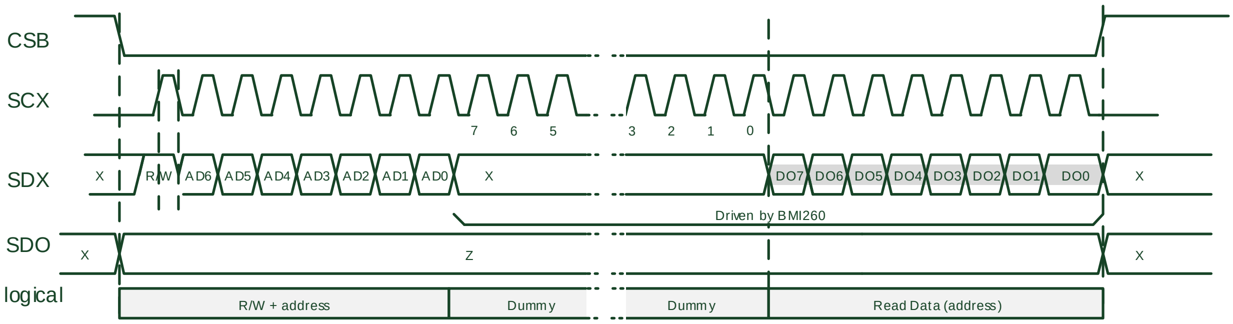
Multiple read operations are possible by keeping CSB low and continuing the data transfer. Only the first register address has to be written. Addresses are automatically incremented after each read access as long as CSB stays active low. Please note that the first byte received from the device via the 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.

In SPI 3-wire configuration CSB (chip select low active), SCX (as SCK for serial clock), and SDX (as SDA for serial data input and output) pins are used. While SCK is high, the communication starts when the CSB is pulled low by the SPI master and stops when CSB is pulled high. SCK is controlled by SPI master. SDI 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.

The protocol as such is the same in 3-wire configuration as it is in 4-wire configuration. The basic operation for read and write access for 3-wire configuration is depicted in the figure below:



3-wire basic SPI write sequence (mode ´11´)



3-wire basic SPI read sequence (mode ´00´)

6.5. Primary Interface I^2C

The I^2C bus uses SCX (as SCL for serial clock) and SDX (as SDA for serial data input and output) signal lines. Both lines are connected to V_{DDIO} externally via pull-up resistors so that they are pulled high when the bus is free.

The default I^2C address of the device is 0b1101000 (0x68). It is used if the SDO pin is pulled to ´GND´. The alternative address 0b1101001 (0x69) is selected by pulling the SDO pin to ´VDDIO´.

The I^2C interface of device is compatible with the I^2C Specification UM10204 Rev. 03 (19 June 2007), available at http://www.nxp.com. The device supports **I^2C standard mode (100kHz), fast mode (400kHz) and fast mode plus (1000kHz)**. Only 7-bit address mode is supported.

The device supports **fast mode plus I^2C mode** that allows using clock frequencies up to 1 MHz. In this mode all timings of the fast mode apply and it additionally supports clock frequencies up to 1MHz.

The timing specification for I^2C of the device is given in the following table:
Note: values are target values and will be verified during characterization.

Table 20: Timing specification for I^2C of the device

Parameter	Symbol	Condition	Min	Max	Units
Clock Frequency	f_{SCL}			1000	kHz
SCL Low Period	t_{LOW}		1.3		µs
SCL High Period	t_{HIGH}		0.6		
SDA Setup Time	t_{SUDAT}		0.1		
SDA Hold Time	t_{HDDAT}		0.0		
Setup Time for a repeated Start Condition	t_{SUSTA}		0.6		
Hold Time for a Start Condition	t_{HDSTA}		0.6		
Setup Time for a Stop Condition	t_{SUSTO}		0.6		
Time before a new Transmission can start	t_{BUF}	low power mode	400		
		normal mode	1.3		
Idle time between write accesses in normal mode, fast startup mode	$t_{IDLE\_wacc\_nm}$		2		
Idle time between write accesses in suspend mode, low-power mode	$t_{IDLE\_wacc\_sum}$		450		

The figure below shows the definition of the I²C timings given in the above table:

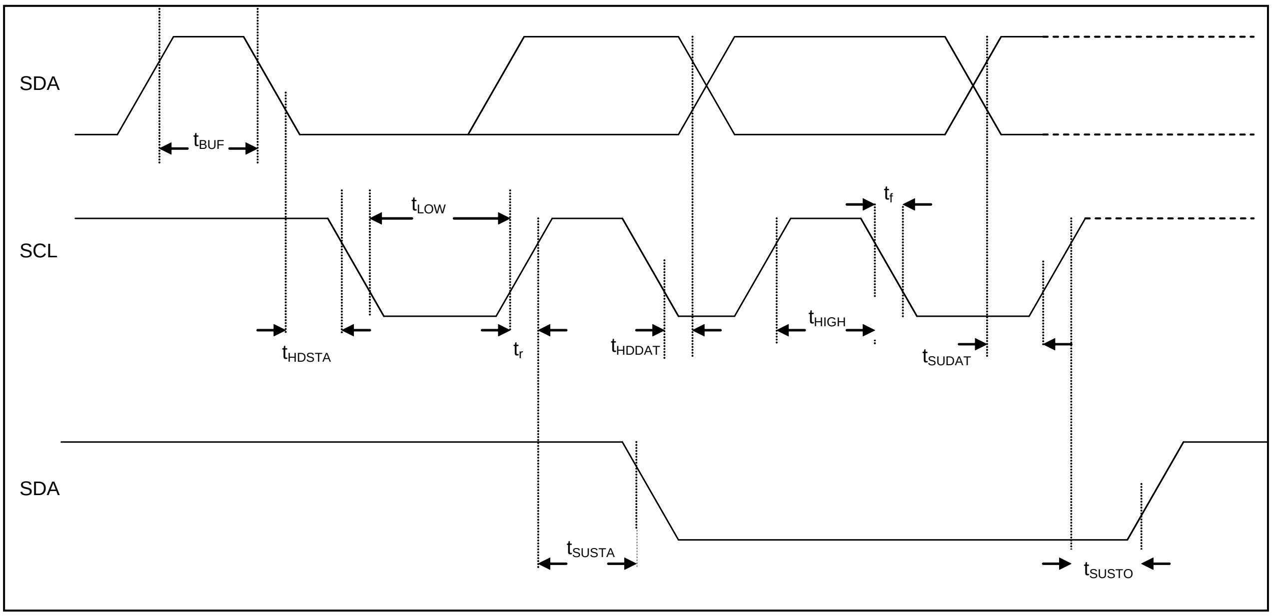


Figure 11: I²C timing diagram

The I²C 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.

ACKS: 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.

In the following diagrams these abbreviations are used:

S	Start
P	Stop
ACKS	Acknowledge by slave
ACKM	Acknowledge by master
NACKM	Not acknowledge by master
RW	Read / Write

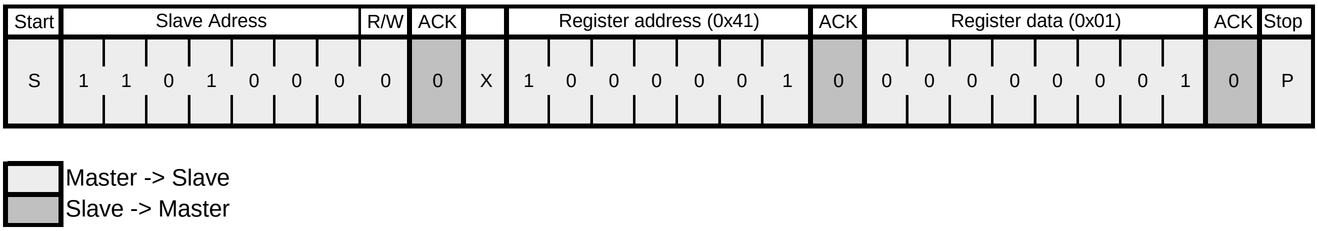
A START immediately followed by a STOP (without SCL toggling from ´VDDIO´ to ´GND´) is not supported. If such a combination occurs, the STOP is not recognized by the device.

I²C write access:

I²C write access can be used to write a data byte 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 (ACKS = 0) and releases the bus. Then the master sends the one byte register address. The slave again acknowledges the transmission and waits for the 8 bits of data which shall be written to the specified register address. After the slave acknowledges the data byte, the master generates a stop signal and terminates the writing protocol.

Example of an I²C write access:



Start	Slave Adress							R/W	ACK		Register address (0x41)							ACK	Register data (0x01)								ACK	Stop
S	1	1	0	1	0	0	0	0	0	X	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	P

▢ Master -> Slave
▨ Slave -> Master

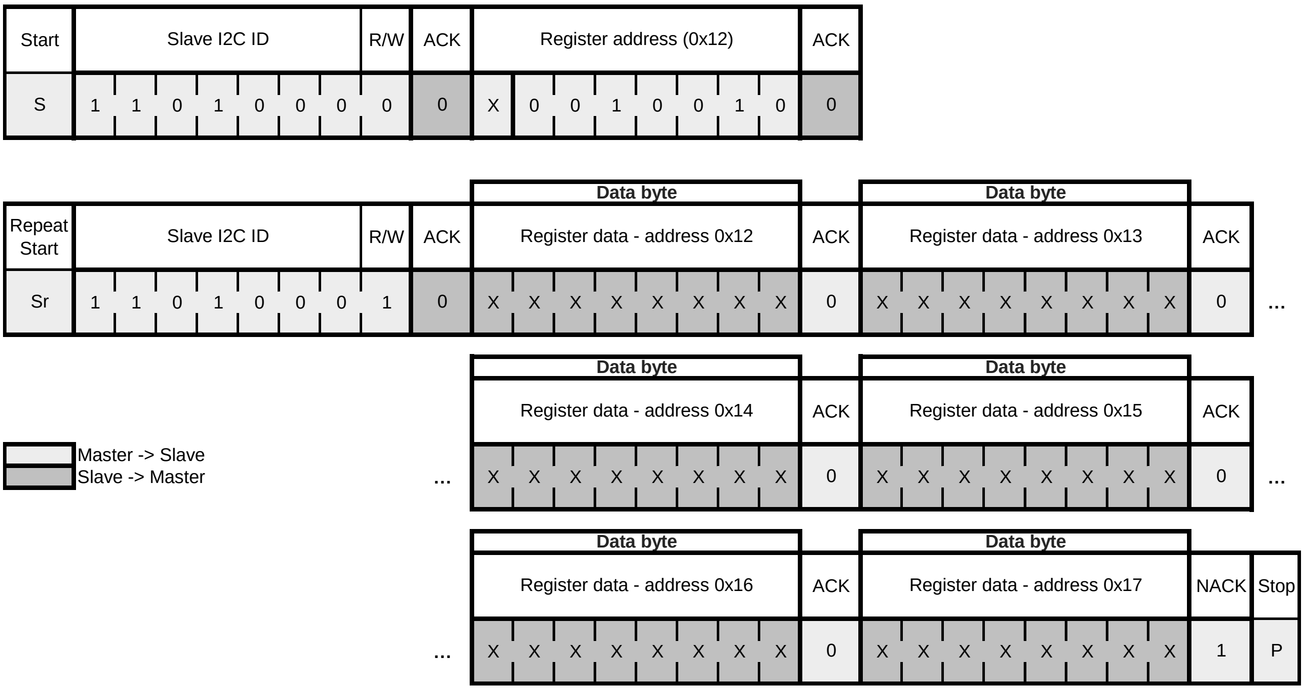
I²C write

Multi-byte writes are supported without restriction on normal registers with auto-increment as well as on special registers with address trap.

I²C read access:

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

A read sequence consists of a one-byte I²C write phase followed by the I²C read phase. 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 (ACKM = 0) to enable further data transfer. A NACKM (ACKM = 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 byte 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 multi-bytes reads from the same starting address are possible.



Start	Slave I2C ID							R/W	ACK	Register address (0x12)								ACK
S	1	1	0	1	0	0	0	0	0	X	0	0	1	0	0	1	0	0

Repeat Start	Slave I2C ID							R/W	ACK	Data byte Register data - address 0x12								ACK	Data byte Register data - address 0x13								ACK	
Sr	1	1	0	1	0	0	0	1	0	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	0	...

	Data byte Register data - address 0x14								ACK	Data byte Register data - address 0x15								ACK	
...	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	0	...

	Data byte Register data - address 0x16								ACK	Data byte Register data - address 0x17								NACK	Stop
...	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	1	P

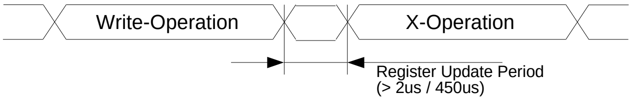
☐ Master -> Slave
▨ Slave -> Master

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 device. The activity and the timer period of the WDT can be configured through the bits NV_CONF.i2c_wdt_en and NV_CONF.i2c_wdt_sel.

SPI and I²C Access Restrictions

In order to allow for the correct internal synchronization of data written to the device, certain access restrictions apply for consecutive write accesses or a write/read sequence through the SPI as well as I2C interface. The required waiting period depends on whether the device is operating in normal mode or other modes.

As illustrated in the figure below, an interface idle time of at least 2µs is required following a write operation when the device operates in normal mode. In suspend mode an interface idle time of least 450µs is required.



Post-Write Access Timing Constraints

6.6. Secondary Interface

The secondary interface can be used in **either** of the following two configurations:

- Auxiliary interface (I2C master) for connecting an external sensor:
  In this case, the secondary interface is used as a two-wire I2C interface (ASDX and ASCX pins) where an external sensor like a magnetometer can be connected as a slave to the device. Typical application is connecting a Bosch Sensortec geomagnetic sensor like BMM150.
- OIS interface (SPI slave) for connecting to OIS control unit
  In this case, the secondary interface is used as an SPI interface where an external controller can be connected as a master to the device. External controller can be an OIS control unit.

The mapping of the device pins for secondary interface usage is given in following table:

Table 21: Mapping of the device pins for secondary interface

PIN #	**NAME**	**I/O TYPE**	**DESCRIPTION**	**CONNECTION TO SECONDARY INTERFACE**		
				OIS SPI 4W	**OIS SPI 3W**	**AUXILIARY I2C**
2	ASDX	Digital I/O	Aux interface / OIS interface	SDI	SDA	SDA
3	ASCX	Digital I/O	Aux interface / OIS interface	SCK	SCK	SCL
10	OCSB	Digital in	OIS interface	CSB	CSB	DNC
11	OSDO	Digital out	OIS interface	SDO	DNC	DNC

Auxiliary Interface

The device allows attaching an external sensor (e.g. magnetometer) to the secondary interface. The connection diagrams for the auxiliary interface are depicted in the section 7.3. The timings of the secondary I2C interface are the same as for the primary I2C interface, see section 6.5.

The device acts as a master of the secondary interface, controls the data acquisition of the external sensor (slave of the secondary interface) and presents the data to the application processor (AP) in the user registers of the device through the primary interface. No external pull-up resistors need to be connected, since an internal pull-up register can be configured through AUX_IF_TRIM.asda_pupsel. No additional I2C master or slave devices can be attached to the magnetometer interfaces.

The device autonomously reads out the sensor data from the external sensor without intervention of the application processor and stores the data in its data registers (per default) and FIFO (see Register FIFO_CONFIG_1.fifo_aux_en). The initial setup of the external sensor after power-on is done through indirect addressing in the device.

For more information about the usage of auxiliary interface see Section 4.10.

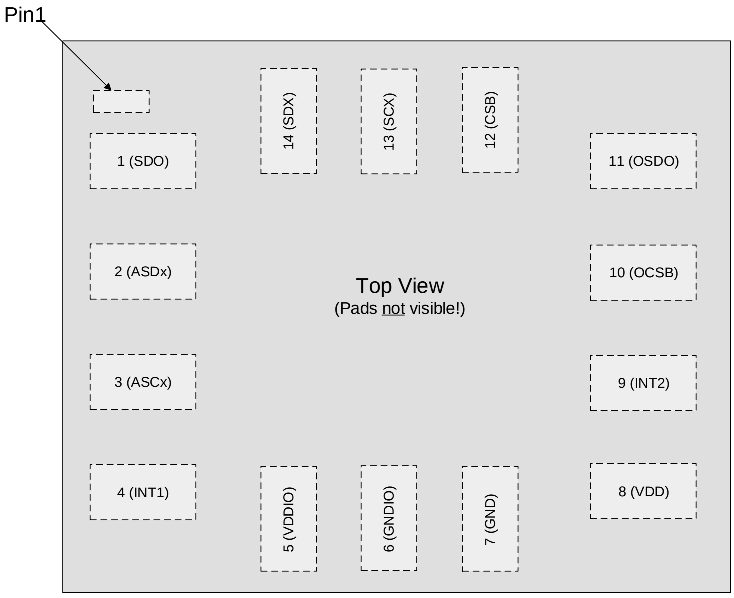
OIS Interface

The device can support optical image stabilization (OIS) applications with the secondary interface. The interface is used for direct access to pre-filtered gyroscope and accelerometer data with minimum latency. Pre-filter gyroscope data is available at ODR of 6.4kHz and accelerometer data with ODR 1.6kHz. OIS SPI interface supports 3wire and 4wire modes. The timing of OIS SPI interface is same as for the primary SPI interface described in section 6.4.

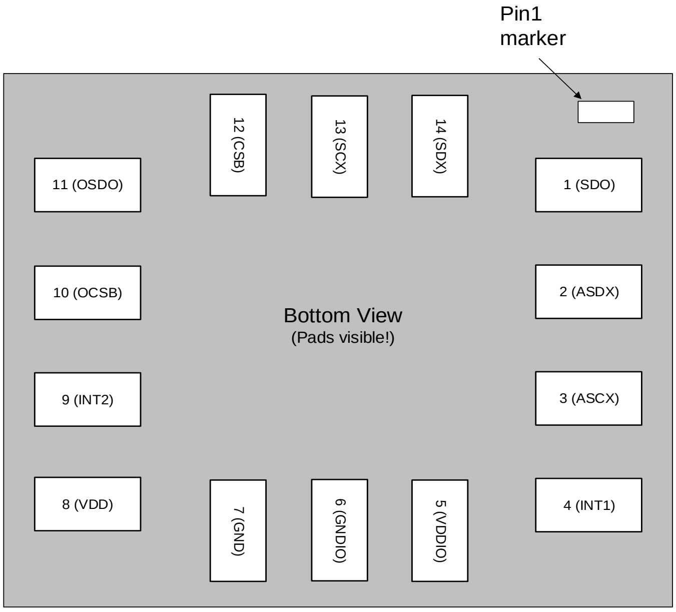
For more information about the usage of the OIS interface see Section 4.11.

7. Pinout and Connection Diagram

7.1. Pin-out



Pin-out top view



Pin-out bottom view

Table 22: Pin-out and pin connections

PIN#	**NAME**	**I/O TYPE**	**INTERFACE**	**DESCRIPTION**	**CONNECT TO**		
					IN SPI 4W	**IN SPI 3W**	**IN I2C**
1	SDO	Digital I/O	Primary	SDO Serial data output in SPI 4W I2C Address bit-0 select in I2C mode	SDO	DNC	GND for default I2C address
2	ASDx	Digital I/O	Secondary	Aux interface / OIS interface**	VDDIO or DNC or Aux SDA or OIS SDI	VDDIO or DNC or Aux SDA or OIS SDI	VDDIO or DNC or Aux SDA or OIS SDI
3	ASCx	Digital I/O	Secondary	Aux interface / OIS interface**	VDDIO or DNC or Aux SCL or OIS SCK	VDDIO or DNC or Aux SCL or OIS SCK	VDDIO or DNC or Aux SCL or OIS SCK
4	INT1	Digital I/O	-	Interrupt pin 1*	INT1	INT1	INT1
5	VDDIO	Supply	-	Digital I/O supply voltage (1.2 ... 3.6V)	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 – 3.6V)	VDD	VDD	VDD
9	INT2	Digital I/O	-	Interrupt pin 2 *	INT2	INT2	INT2
10	OCSB	Digital in	Secondary	OIS interface	DNC*** or OIS CSB	DNC*** or OIS CSB	DNC*** or OIS CSB
11	OSDO	Digital out	Secondary	OIS interface	DNC*** or OIS SDO	DNC*** or OIS SDO	DNC*** or OIS SDO
12	CSB	Digital in	Primary	Chip select for SPI mode	CSB	CSB	VDDIO(preferred) or DNC

13	SCx	Digital in	Primary	SCK for SPI serial clock SCL for I²C serial clock	SCK	SCK	SCL
14	SDx	Digital I/O	Primary	SDA serial data I/O in I2C SDI serial data input in SPI 4W SDA serial data I/O in SPI 3W	SDI	SDIO	SDA

*) If INT1 and/or INT2 are not used, please do not connect them (DNC). INT1 and/or INT2 can also be configured as input in case the external data synchronization of FIFO is used.

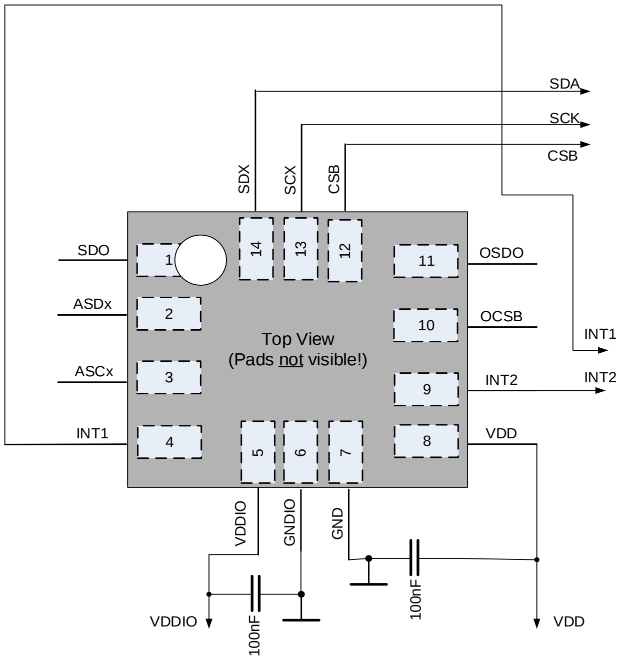
**) If secondary interface is unused, ASDX and ASCX can be connected to VDDIO or left unconnected. Do not connect to GND.

***) Can be tied to GND only if register IF_CONF.ois_en = 0

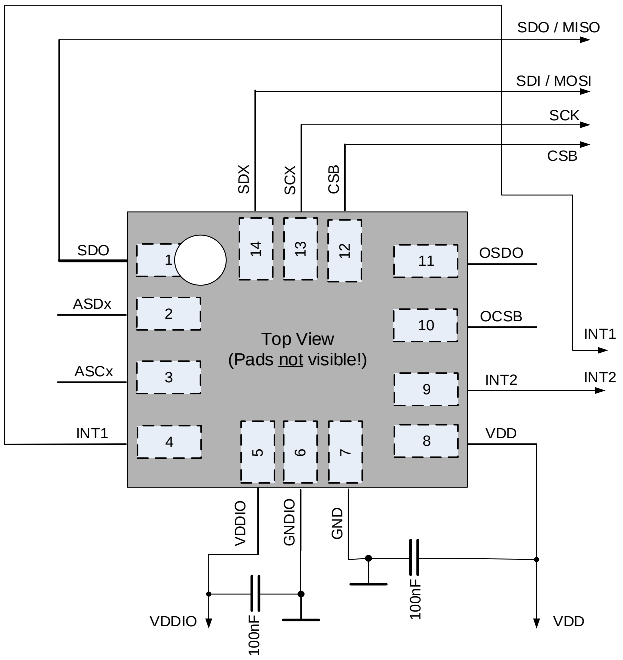
7.2. Connection Diagrams without Secondary Interface

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

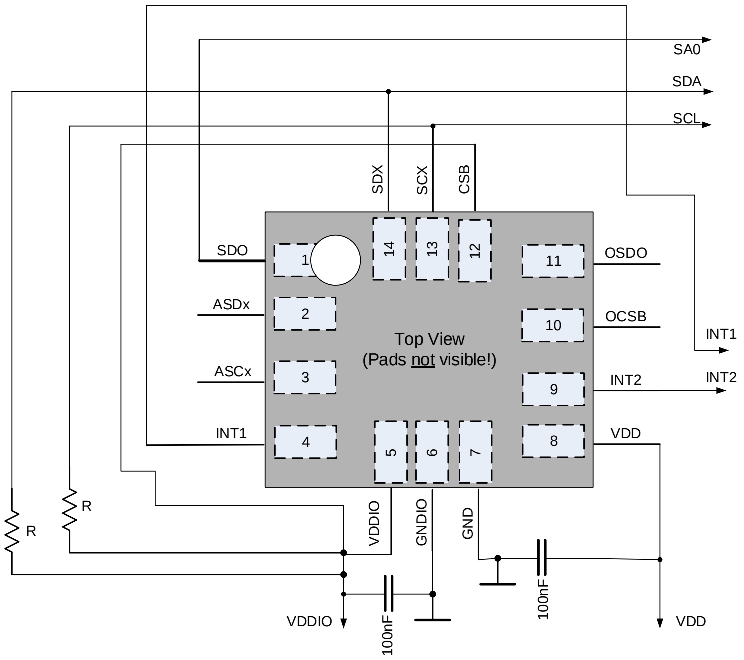
Primary: 3wire SPI Secondary: None



Primary: 4wire SPI Secondary: None



Primary: I2C Secondary: None

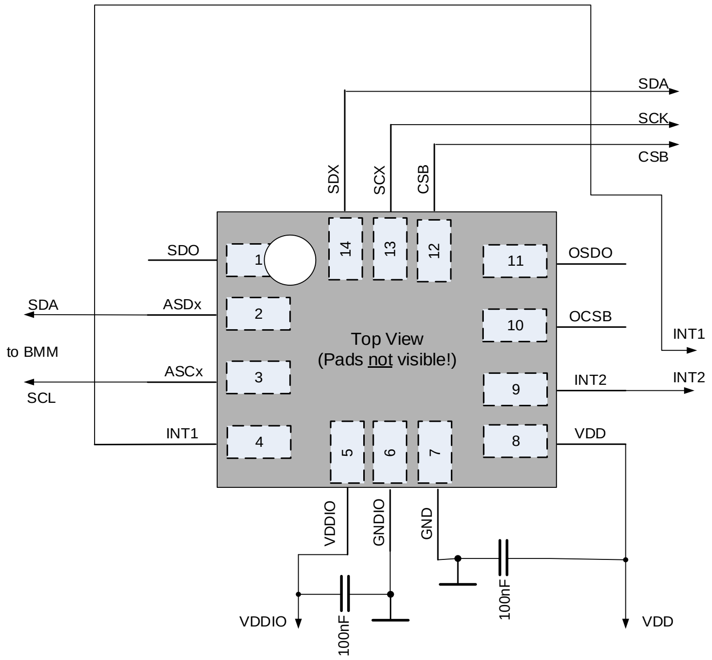


Here SA0 = I2C slave address bit-0 select

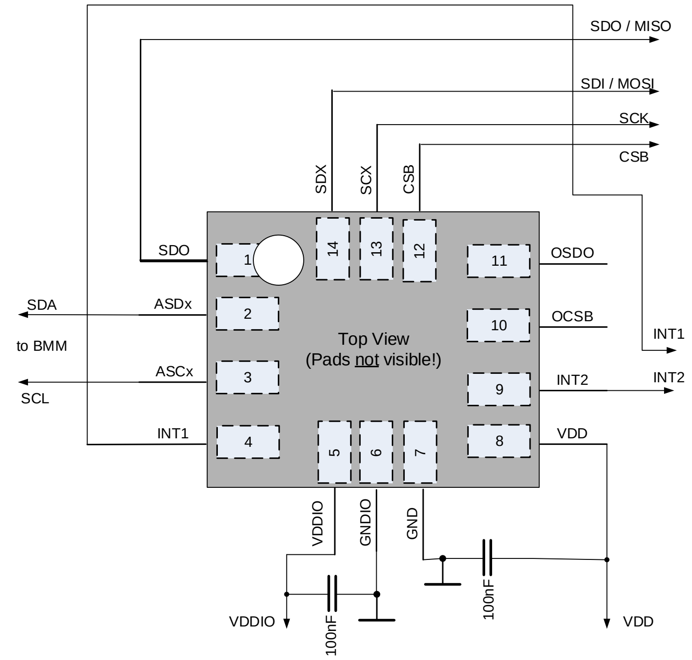
7.3. Connection Diagrams with I2C Auxiliary Interface

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

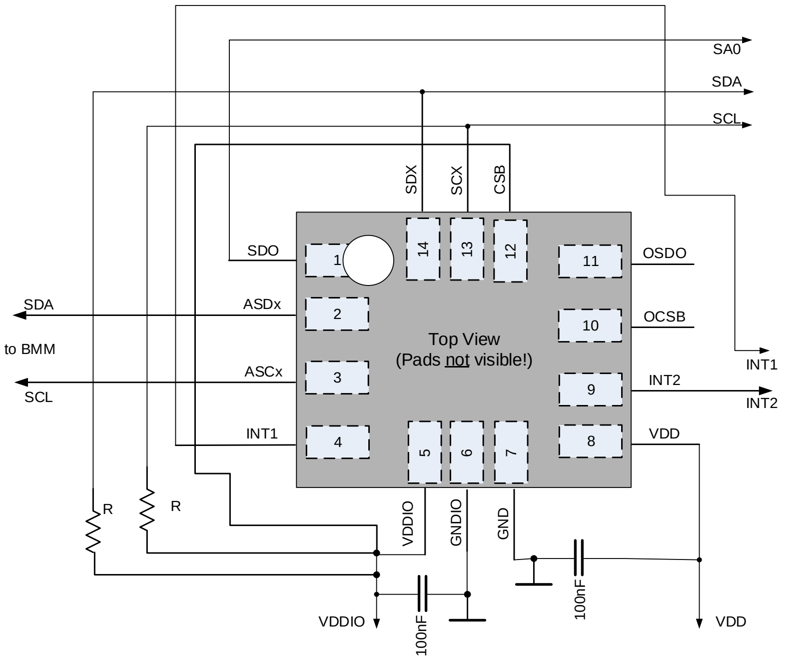
Primary: 3wire SPI Secondary: Auxiliary interface I2C (e.g. BMM150 sensor)



Primary: 4wire SPI Secondary: Auxiliary interface I2C (e.g. BMM150 sensor)



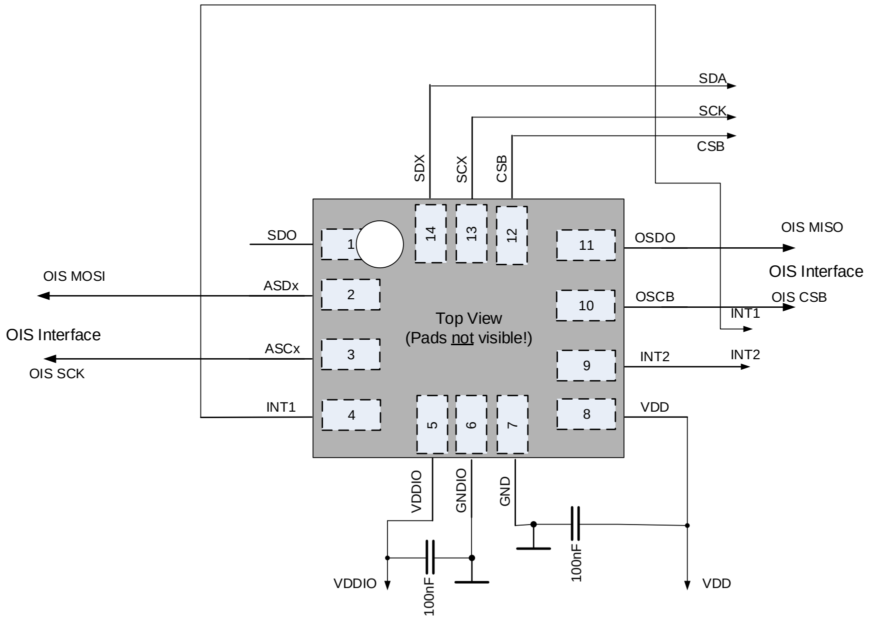
Primary: I2C Secondary: Auxiliary interface I2C (e.g. BMM150 sensor)



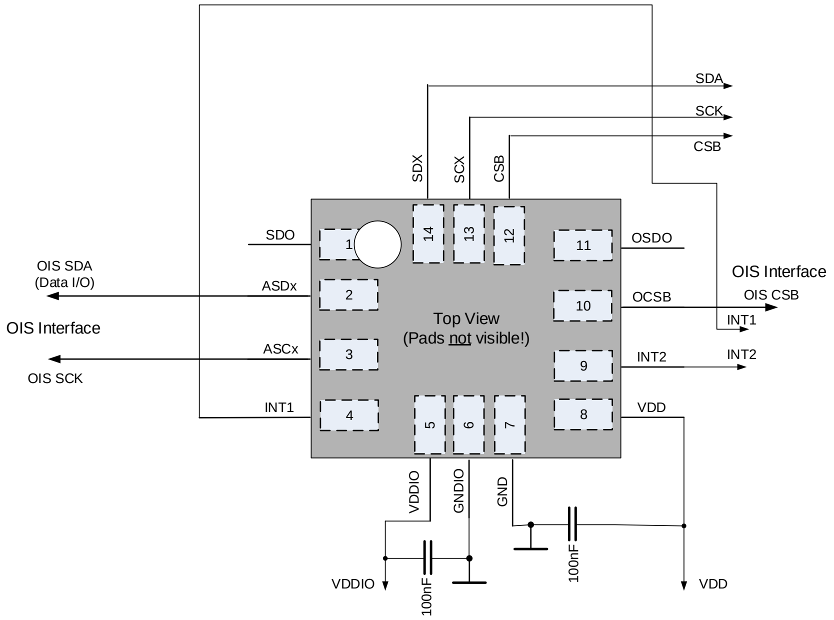
7.4. Connection Diagrams with OIS Interface

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

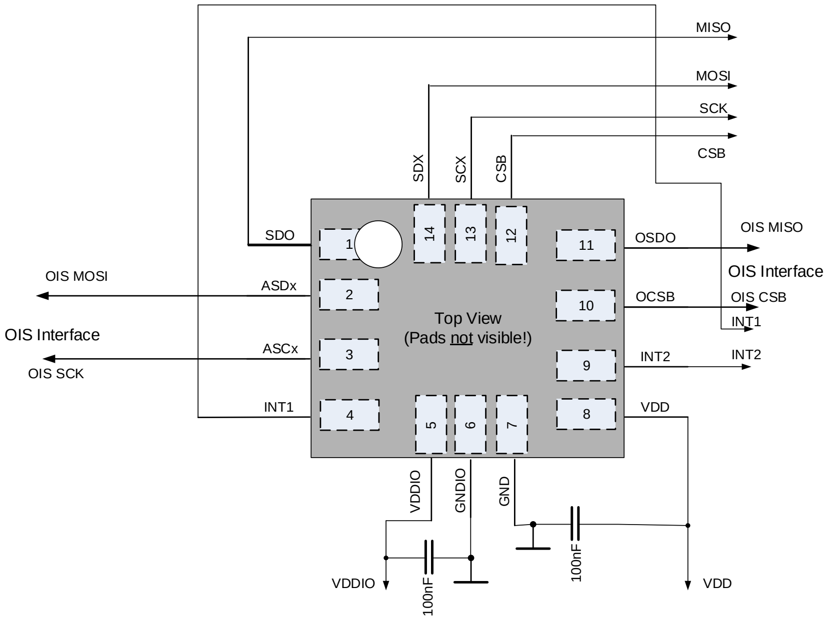
Primary: 3wire SPI Secondary: 4 wire SPI for OIS interface



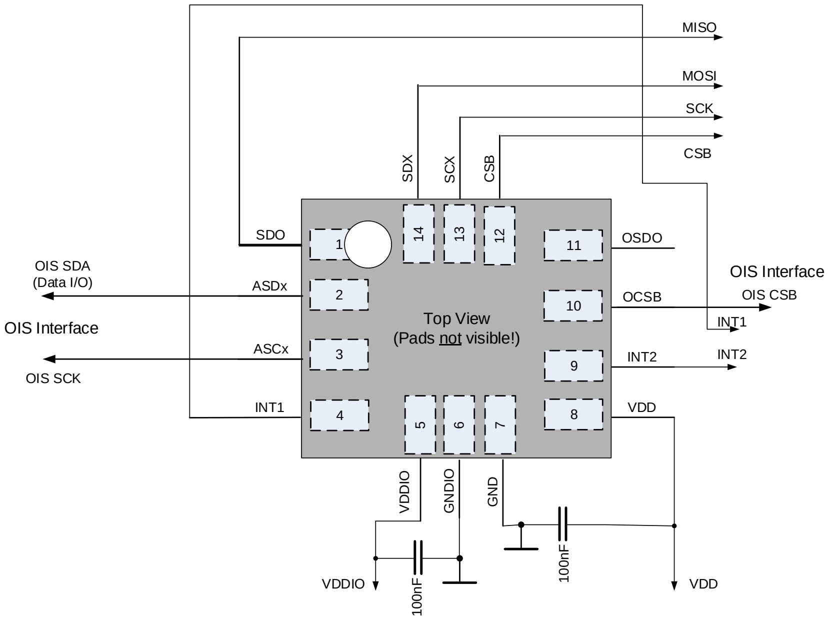
Primary: 3wire SPI Secondary: 3 wire SPI for OIS interface



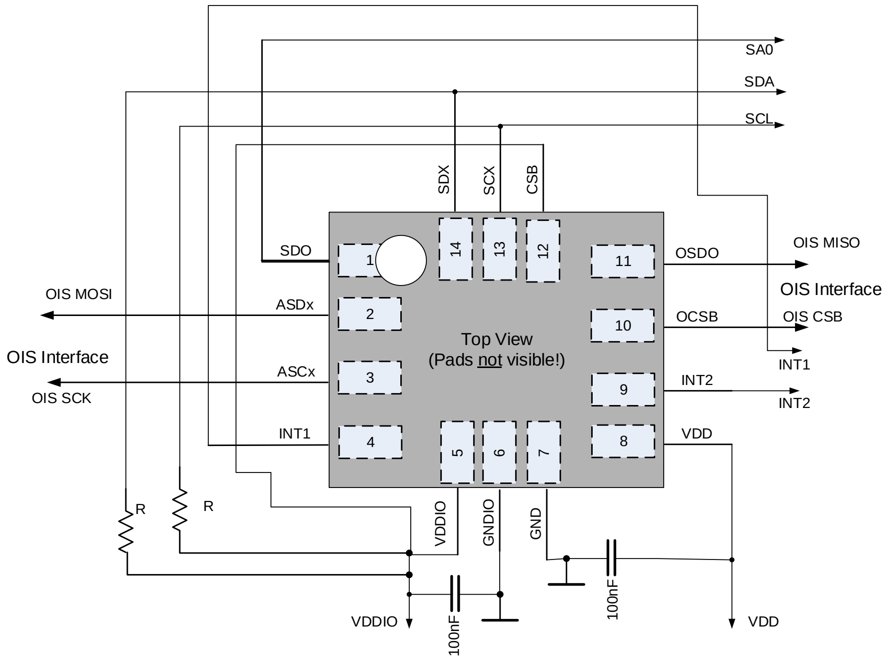
Primary: 4wire SPI Secondary: 4 wire SPI for OIS interface



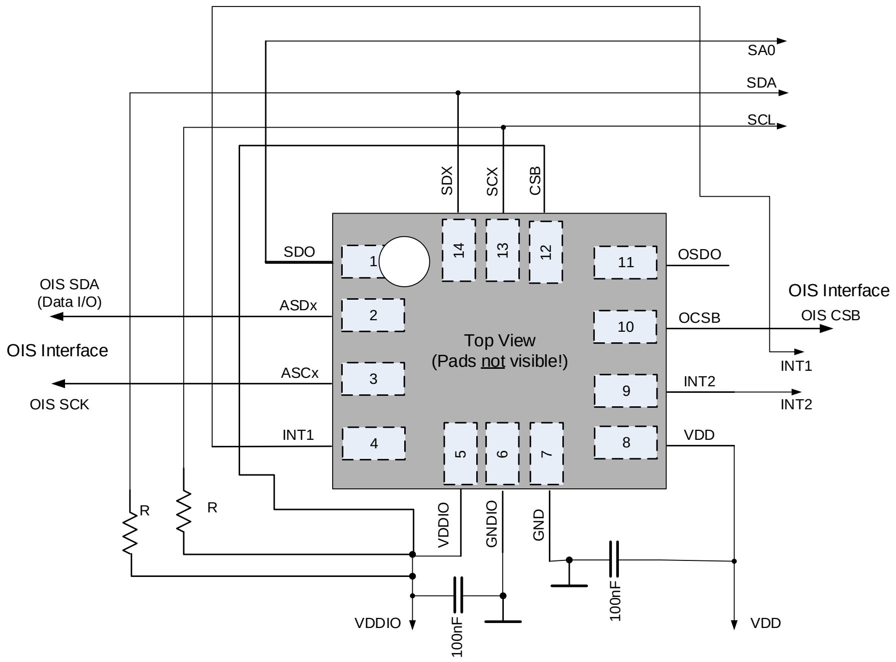
Primary: 4wire SPI Secondary: 3 wire SPI for OIS interface



Primary: I2C Secondary: 4 wire SPI for OIS interface



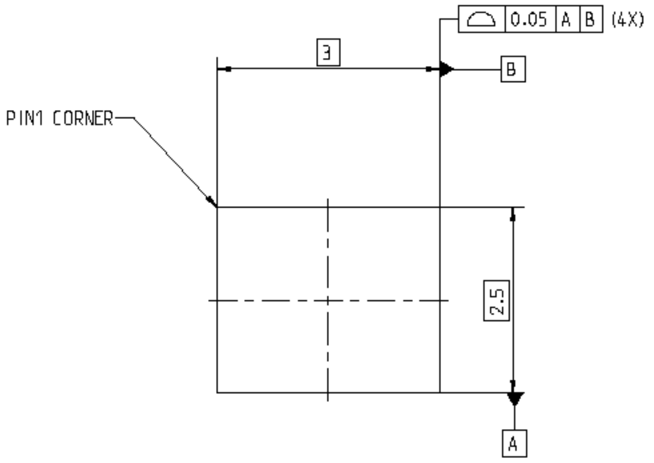
Primary: I2C Secondary: 3 wire SPI for OIS interface



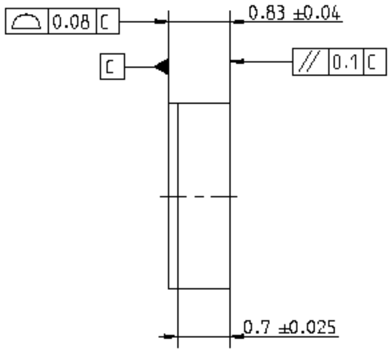
I2C primary and 3 wire SPI for OIS interface

8. Package

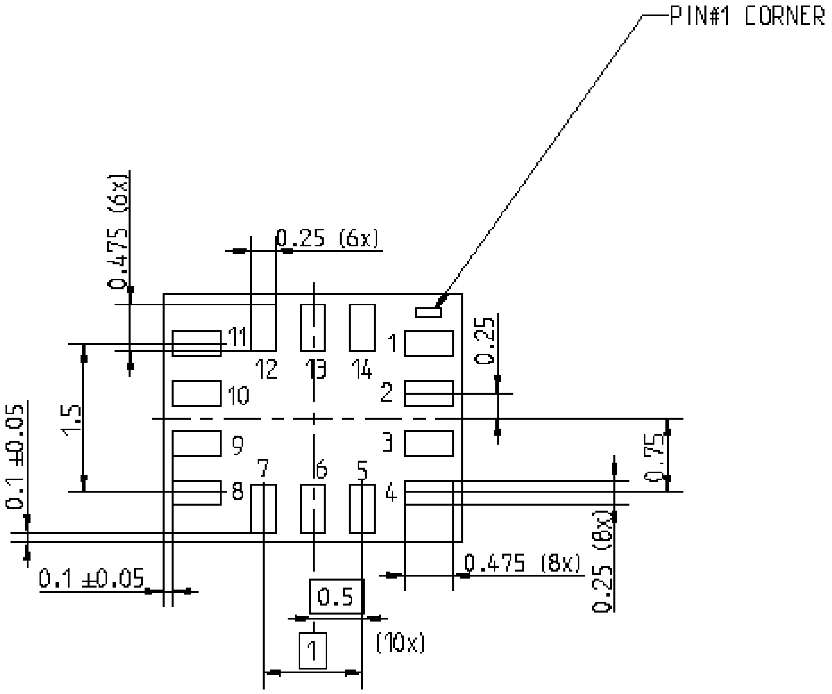
8.1 . Package outline dimensions



TOP VIEW



SIDE VIEW



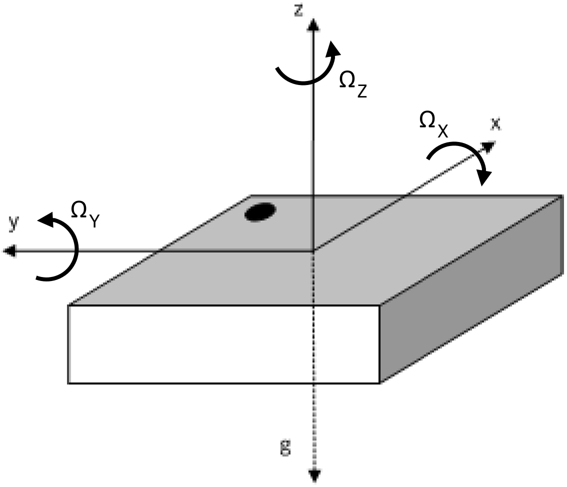
BOTTOM VIEW

8.2. Sensing axis orientation

If the sensor is accelerated and/or rotated in the indicated directions, the corresponding channels of the device will deliver a positive acceleration and/or yaw rate signal (dynamic acceleration). If the sensor is at rest without any rotation and the force of gravity is acting contrary to the indicated directions, the output of the corresponding acceleration channel will be positive and the corresponding gyroscope channel will be "zero" (static acceleration).

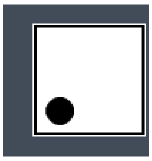
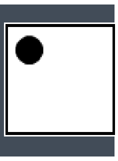
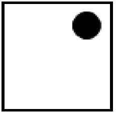
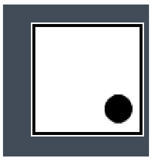
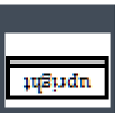
Example: If the sensor is at rest or at uniform motion in a gravity field according to the figure given below, the output signals are:

- ± 0g for the X ACC channel and ± 0°/sec for the Ω_X GYR channel
- ± 0g for the Y ACC channel and ± 0°/sec for the Ω_Y GYR channel
- + 1g for the Z ACC channel and ± 0°/sec for the Ω_Z GYR channel



Definition of sensing axes orientation

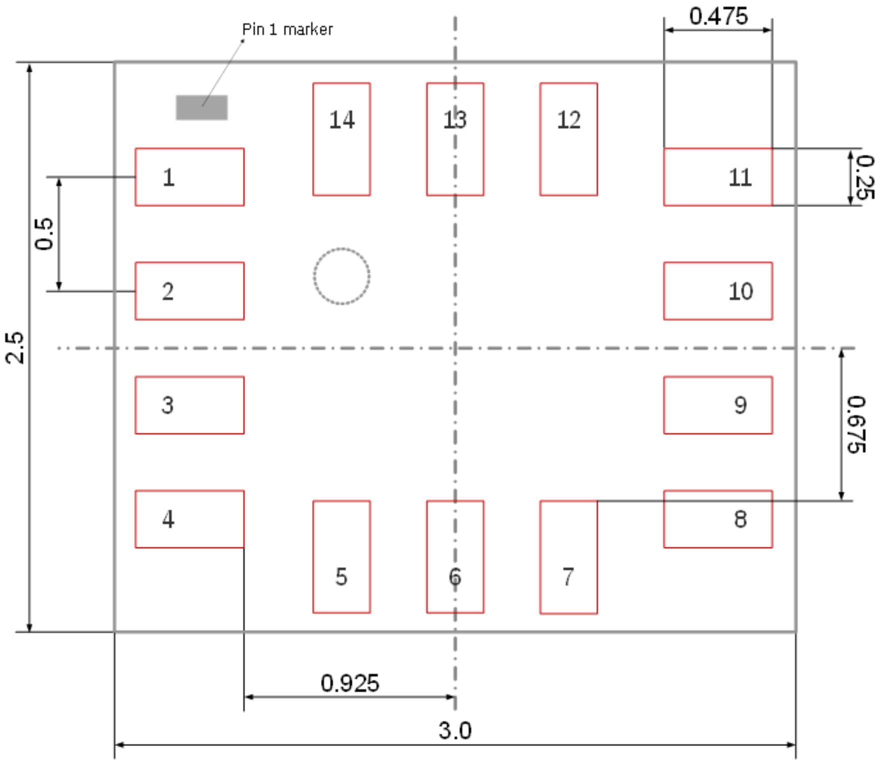
The following table lists all corresponding output signals on X, Y, and Z while the sensor is at rest or at uniform motion in a gravity field under assumption of a ±4g range setting, a 16 bit resolution, and a top down gravity vector as shown above.

Sensor Orientation (gravity vector↓)						
Output Signal X	0g/0 LSB	1g/8192 LSB	0g/0 LSB	-1g/-8192 LSB	0g/0 LSB	0g/0 LSB
Output Signal Y	-1g/-8192 LSB	0g/0 LSB	1g/8192 LSB	0g/0 LSB	0g/0 LSB	0g/0 LSB
Output Signal Z	0g/0 LSB	0g/0 LSB	0g/0 LSB	0g/0 LSB	1g/8192 LSB	-1g/-8192 LSB

If the sensor axes coordinates do not match the platforms axes coordinates, then axis remapping is required. For the accelerometer and gyroscope data, the axes remapping needs to be implemented in the applications processor's driver. For the interrupt features to work properly, axes remapping information must be written to configuration registers of the device. Axes remapping is supported via most interrupt features in a configuration registers, which applies to the feature algorithms.

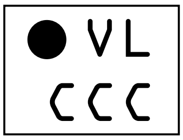
8.3. Landing pattern recommendation

The following landing pad recommendation is given for maximum stability of the solder connections.

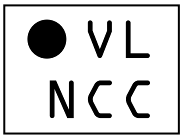


8.4. Marking

Mass production

Labeling	Name	Symbol	Remark
	Internal Code	L	1 alphanumeric digit, fixed, L ≠ “E” L = “P” or “L”, internal use
	Product Identifier	V	1 alphanumeric digit, fixed, V = “5” to identify BMI2xy product family
	Counter ID	CCC	3 alphanumeric digits, variable to generate trace-code.
	Pin 1 identifier top side	●	--

Engineering samples

Labeling	Name	Symbol	Remark
	Eng. sample ID	L, N	2 alphanumeric digit, fixed, L = “E” to identify engineering sample, N = “L” or “C”
	Product Identifier	V	1 alphanumeric digit, fixed, V = “P” or “L” to identify BMI2xy product family
	Counter ID	CC	2 alphanumeric digits, variable Internal revision ID
	Pin 1 identifier top side	●	--

8.5. Soldering guidelines

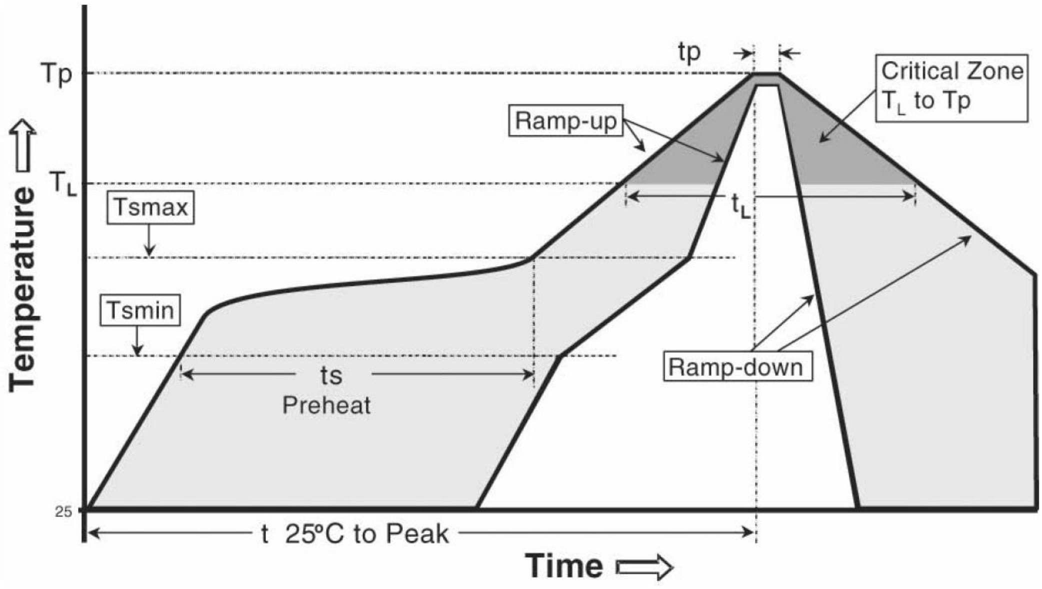
The moisture sensitivity level of the device corresponds to JEDEC Level 1, see also

- IPC/JEDEC J-STD-020C "Joint Industry Standard: Moisture/Reflow Sensitivity Classification for non-hermetic Solid State Surface Mount Devices"
- IPC/JEDEC J-STD-033A "Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices"

The sensor fulfils the lead-free soldering requirements of the above-mentioned IPC/JEDEC standard, i.e. reflow soldering with a peak temperature T_p up to 260°C.

Profile Feature	**Pb-Free Assembly**
Average ramp-up rate (Tsmax to Tp)	3° C/second max.
Preheat - Temperature Min (Ts_{min}) - Temperature Max (Ts_{max}) - Time (Ts_{min} to Ts_{max}) (ts)	 150 °C 200 °C 60-180 seconds
Time maintained above: - Temperature (T_L) - Time (t_L)	 217 °C 60-150 seconds
Peak Temperature (Tp)	See Table 4.2
Time within 5°C of actual Peak Temperature $(tp)^2$	20-40 seconds 260 °C
Ramp-down Rate	6 °C/second max.
Time 25°C to Peak Temperature	8 minutes max.

Note 1: All temperatures refer to topside of the package, measured on the package body surface.
Note 2: Time within 5 °C of actual peak temperature (tp) specified for the reflow profiles is a "supplier" minimum and "user" maximum.



8.6. 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's. However, these limits might be exceeded in conditions with extreme shock loads such as e.g. hammer blow on or next to the sensor, dropping of the sensor onto hard surfaces etc.

We recommend to avoid g-forces beyond the specified limits during transport, handling and mounting of the sensors in a defined and qualified installation process.

This device has built-in protections against high electrostatic discharges or electric fields (e.g. 2kV HBM); however, anti-static precautions should be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the supply voltage range. Unused inputs must always be tied to a defined logic voltage level.

8.7. Environmental safety

The device meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also:

> *Directive 2002/95/EC of the European Parliament and of the Council of 8 September 2011 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 LGA 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.

9. Legal disclaimer

9.1. Engineering samples

Engineering Samples are marked with an asterisk (*) or (e). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

9.2. Product use

Bosch Sensortec products are developed for the consumer goods industry. They are not designed or approved for use in military applications, life-support appliances, safety-critical automotive applications and devices or systems where malfunctions of these products can reasonably be expected to result in personal injury. They may only be used within the parameters of this product data sheet.

The resale and/or use of products are at the Purchaser’s own risk and the Purchaser’s own responsibility.

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 inform Bosch Sensortec without delay of any security relevant incidents.

9.3. Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

10. Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
0.9	-	Preliminary version	29-Feb-2019
1.0	6.5	Typo	17-May-2019

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

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

Document number: BST-BMI270-DS000-02
Revision_1.0_052019