BMA490L

High-performance longevity acceleration sensor

BMA490L – Data Sheet

Document revision	1.0
Document release date	June 2020
Document number	BST-BMA490L-DS000-01
Sales Part number	0273 017 021
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

BMA490L - Basic Description

BMA490L is a high-performance longevity acceleration sensor with extended availability of up to ten years[1]. It is a 16 bit, digital, triaxial acceleration sensor with intelligent on-chip motion-triggered interrupt features optimized for industrial applications.

Key features:

• Small package size	LGA package (12 pins), footprint 2mm x 2mm, height 0.95 mm
• Digital Interface	SPI (4-wire, 3-wire), I^2C, 2 interrupt pins, V_{DDIO} voltage range: 1.2V to 3.6V
• Programmable functionality	Acceleration ranges ±2g/±4g/±8g/±16g Low-pass filter bandwidths 684Hz -<8Hz up to a max. output data read out of 1.6 kHz
• On-chip FIFO	Integrated FIFO on sensor with 1 kb
• On-chip interrupt features	Any-/No-Motion interrupt
• Ultra-low power	Low current consumption of data acquisition and all integrated features
• (Secondary) Auxilary Interface	Hub for ext. Magnetometer and data synchronization
• ROHS complaint, halogen free	

Typical applications:

- Industrial IoT (IIoT), e.g. predictive maintenance, vibration monitoring
- Logistics, e.g. asset tracking
- Agricultural and industrial robots, e.g. orientation detection, tilt detection
- White goods and home appliances, e.g. vibration monitoring, power management
- Power tools, e.g. power management, device level detection

10 YEARS
LONGEVITY

[1] See longevity disclaimer on the last page of this document.

Index of Contents

1. Specification

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

1.1 Electrical Specification

Table 1: Electrical Parameter specification

Operating Conditions						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Acceleration Range	g_{FS2g}	Selectable via serial digital interface		±2		g
	g_{FS4g}			±4		g
	g_{FS8g}			±8		g
	g_{FS16g}			±16		g
Supply Voltage Internal Domains	V_{DD}		1.62	1.8	3.6	V
Supply Voltage I/O Domain	V_{DDIO}		1.2	1.8	3.6	V
Voltage Input Low Level	V_{IL}	SPI & I²C			$0.3V_{DDIO}$	-
Voltage Input High Level	V_{IH}	SPI & I²C	$0.7V_{DDIO}$			-
Voltage Output Low Level	V_{OL}	V_{DDIO}>=1.62V, I_{OL}<=2mA, SPI			$0.2V_{DDIO}$	-
		V_{DDIO}<1.62V, I_{OL}<=1.5mA, SPI			$0.2V_{DDIO}$	-
Voltage Output High Level	V_{OH}	V_{DDIO}>=1.62V, I_{OH}<=2mA, SPI	$0.8V_{DDIO}$			-
		V_{DDIO}<=1.62V, I_{OH}<=1.5mA, SPI	$0.8V_{DDIO}$			-
Total Supply Current in Performance mode	I_{DD}	Nominal V_{DD} and V_{DDIO}, 25°C, g_{FS4g}		150		µA
Total Supply Current in Suspend Mode	I_{DDsum}	Nominal V_{DD} and V_{DDIO}, 25°C		3.5		µA
Total Supply Current in Low-power Mode	I_{DDlp1}	Nominal V_{DD} and V_{DDIO}, 25°C 50 Hz ODR		14		µA
Power-Up Time	ts_up				1	ms
Non-volatile memory (NVM) write-cycles	n_{NVM}				15	cycles
Operating Temperature	T_A		-40		+85	°C

OUTPUT SIGNAL						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
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 Temperature Drift	TCS			0.005		%/K
Resolution (in ±2g range)				0.06		mg
Zero-g Offset	Off	Nominal V_{DD} and VDD_{IO}, 25°C, g_{FS4g}		20		mg
Zero-g Offset Drift	Off-Dr	Lifetime; overall		10		mg
Zero-g Offset Temperature Drift	TCO			0,25		mg/K
Output Data Rate	ODR_{PERF}	Performance mode	12.5		1600	Hz
Output data rate and BW in Performance mode	$ODR_{12.5}$	3dB cutoff frequency of the accelerometer according to ODR with normal filter mode		5.06		Hz
	ODR_{25}			10.12		Hz
	ODR_{50}			20.25		Hz
	ODR_{100}			40.5		Hz
	ODR_{200}			80		Hz
	ODR_{400}			162 (155 for Z axis)		Hz
	ODR_{800}			324 (262 for Z axis)		Hz
	ODR_{1600}			684 (353 for Z axis)		HZ
Output Data Rate	ODR_{LPM}	Low-power mode	0.78		400	Hz
Nonlinearity	NL	Nominal V_{DD} and VDD_{IO}, 25°C, g_{FS4g}		0.5		%FS
Output Noise Density	n_{dens}	Nominal V_{DD} and VDD_{IO}, 25°C, g_{FS4g}		120		µg/√Hz
Temperature sensor Measurement Range	T_S		-40		+80	°C
Temperature Sensor Slope	dT_S			1		K/LSB
Temperature Sensor Offset	OT_S	at 23°C		1		K
Power Supply Rejection Ratio	PSRR			1		mg/50mV

MECHANICAL CHARACTERISTICS						
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Units**
Cross Axis Sensitivity	S	relative contribution between any two of the three axes		0,7		%
Alignment Error	E_A	relative to package outline		0.5		°

2. Absolute maximum ratings

Table 2: Absolute maximum ratings

Parameter	Condition	Min	Max	Units
Voltage at Supply Pin	V_{DD} Pin	-0.3	4	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
None-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, at any pin	HBM		2	kV
	CDM		500	V
	MM		200	V

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

3. Quick Start Guide

The purpose of this chapter is to help developers who want to start working with the BMA490L by giving you some very basic hands-on application examples to get started.

Note about using the BMA490L:

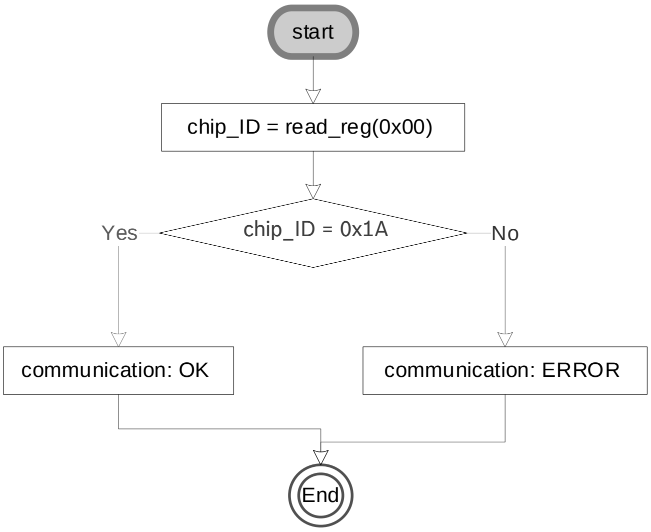
- The communication between application processor and BMA490L will happen either over I2C or SPI interface. For more information about the interfaces, read the related chapter 6 Digital Interfaces.

- Before starting the test, the device has to be properly connected to the master (AP) and powered up. For more information about it, read the related chapter 7 Pin-out and Connection Diagrams.

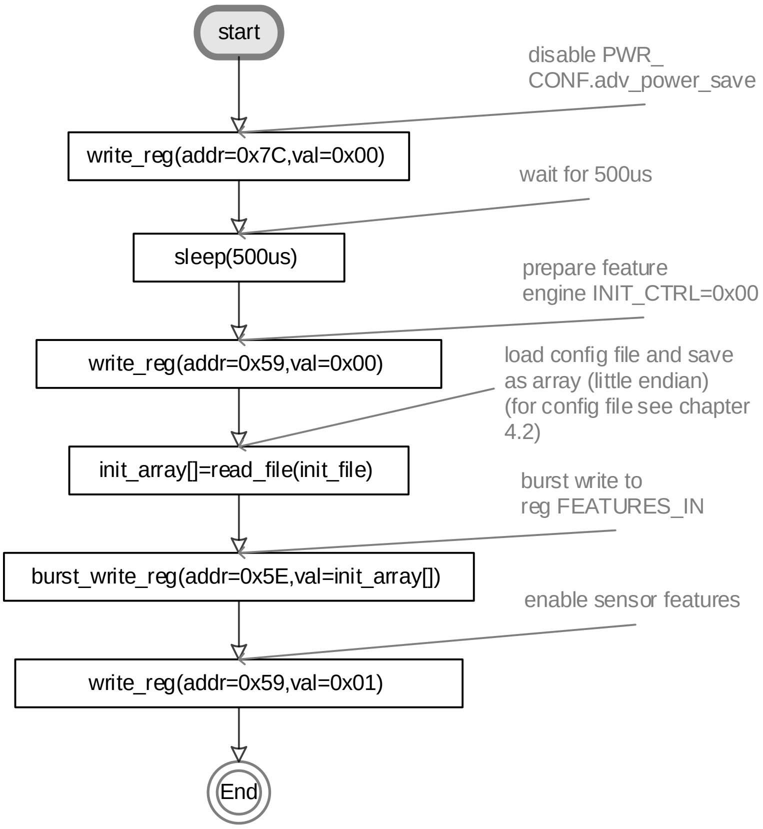
First application setup examples algorithms:

After correct power up by setting the correct voltage to the appropriate external pins, the BMA490L enters automatically into the Power On Reset (POR) sequence. In order to properly make use of the BMA490L, certain steps from host processor side are needed. The most typical operations will be explained in the following application examples in form of flow-diagrams.

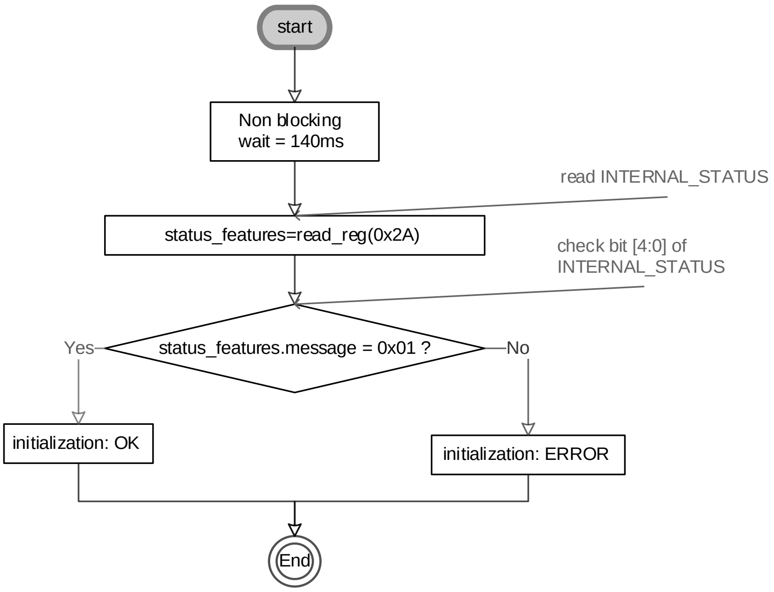
Example 1: ***Testing communication with the BMA490L and initializing feature engine***
a. -reading chip id (checking correct communication)



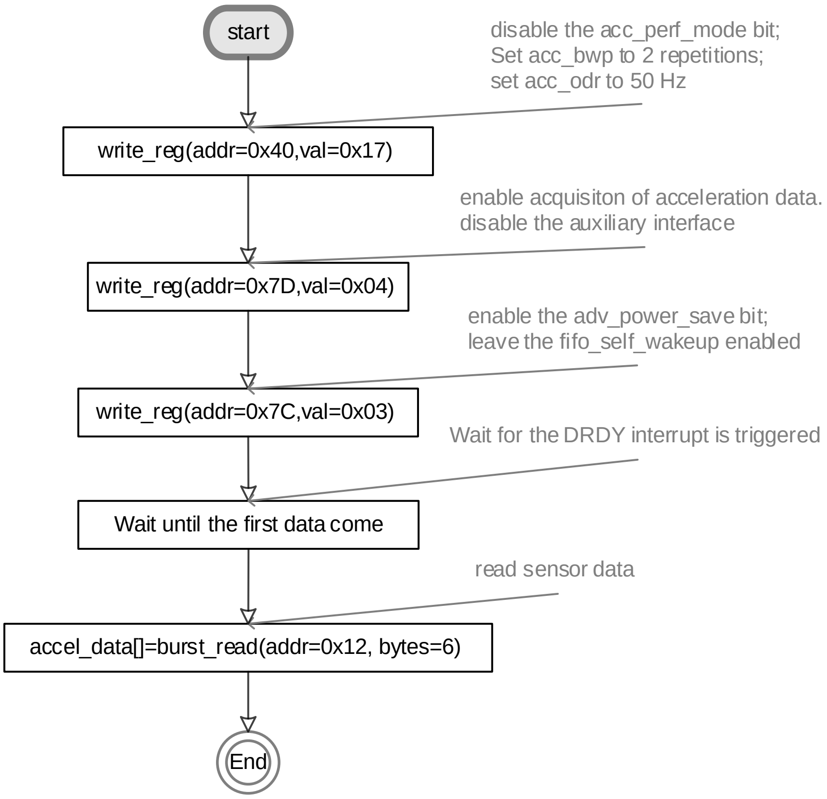
b. -performing initialization sequence (interrupt feature engine)


start
disable PWR_
CONF.adv_power_save
write_reg(addr=0x7C,val=0x00)
wait for 500us
sleep(500us)
prepare feature
engine INIT_CTRL=0x00
write_reg(addr=0x59,val=0x00)
load config file and save
as array (little endian)
(for config file see chapter
4.2)
init_array[]=read_file(init_file)
burst write to
reg FEATURES_IN
burst_write_reg(addr=0x5E,val=init_array[])
enable sensor features
write_reg(addr=0x59,val=0x01)
End

c. -checking the correct status of the interrupt feature engine



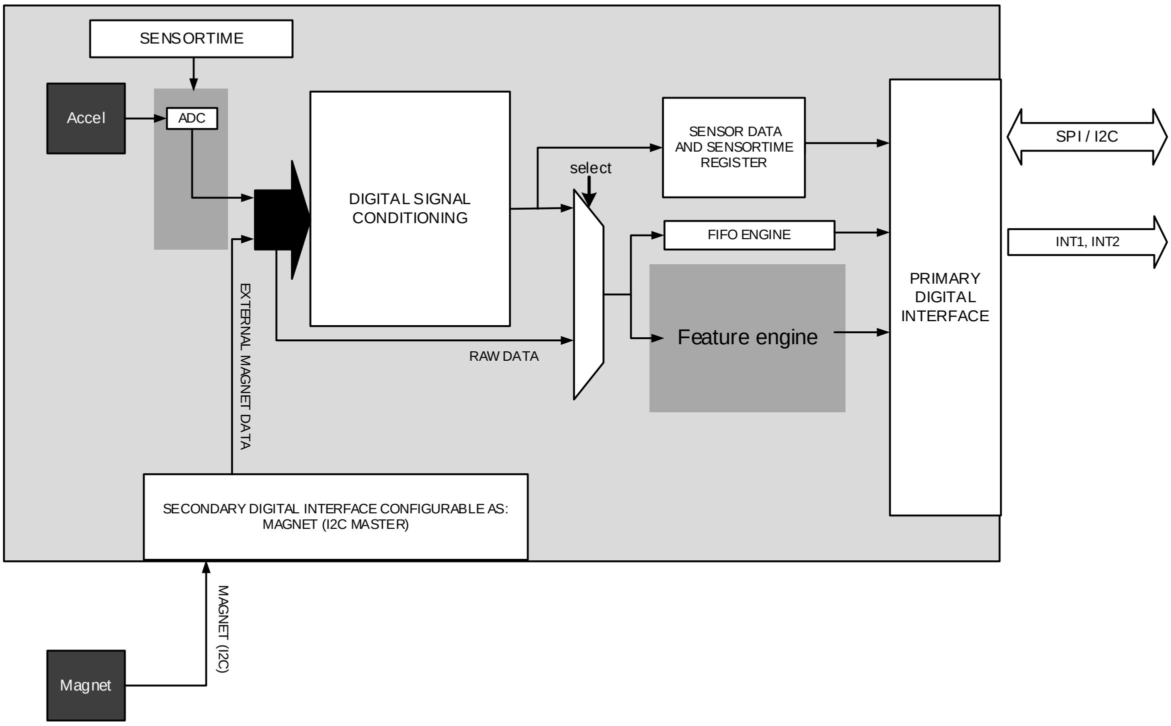
Example 2: ***Reading acceleration data from BMA490L (example: low power mode)***
-setting data processing parameters (power, bandwidth, range) and reading sensor data


start
write_reg(addr=0x40,val=0x17)
disable the acc_perf_mode bit;
Set acc_bwp to 2 repetitions;
set acc_odr to 50 Hz
write_reg(addr=0x7D,val=0x04)
enable acquisiton of acceleration data.
disable the auxiliary interface
write_reg(addr=0x7C,val=0x03)
enable the adv_power_save bit;
leave the fifo_self_wakeup enabled
Wait until the first data come
Wait for the DRDY interrupt is triggered
accel_data[]=burst_read(addr=0x12, bytes=6)
read sensor data
End

Further steps:
The BMA490L has many more 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, many interrupts generation and features like any motion/no motion.

4. Functional Description

4.1 Block Diagram


SENSORTIME
Accel
ADC
DIGITAL SIGNAL CONDITIONING
select
SENSOR DATA AND SENSORTIME REGISTER
FIFO ENGINE
Feature engine
RAW DATA
EXTERNAL MAGNET DATA
PRIMARY DIGITAL INTERFACE
SPI / I2C
INT1, INT2
SECONDARY DIGITAL INTERFACE CONFIGURABLE AS: MAGNET (I2C MASTER)
MAGNET (I2C)
Magnet

4.2 Supply Voltage and Power Management

BMA490L 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.

4.3 Device Initialization

After power up sequence the accelerometer is in suspend mode, device must be initialized through the following procedure. Initialization has to be performed as well after every POR or soft reset.

- Disable advanced power save mode: PWR_CONF.adv_power_save =0b0
- Wait for 450 us. The register SENSORTIME_0 increments every 39.25 µsec and may be used for accurate timing.
- Write INIT_CTRL.init_ctrl=0x00
- Load configuration file
    - Burst write initialization data to Register FEATURES_IN. The configuration file is included in the driver available on the Bosch Sensortec website (www.bosch-sensortec.com) or from your regional support team. Optionally the configuration file can be written to the Register FEATURES_IN in several consecutive burst write access. Every burst write must contain an even number of bytes.
    - Optionally:
      Burst read configuration file from Register FEATURES_IN and check correctness. Check sensor API for details of timing & length.
- Enable sensor features– write 0x01 into register INIT_CTRL.init_ctrl. This operation must not be performed more than once after POR or softreset.
- Wait until Register INTERNAL_STATUS.message contains the value 0b1. This will happen after at most 140-150 msec.

After initialization sequence has been completed, the device is in configuration mode (power mode). Now it is possible to switch to the required power mode and all features are ready to use as described in chapter 4.

4.4 Power Modes

The power state of the BMA490L is controlled through the registers PWR_CONF and PWR_CTRL. The Register PWR_CTRL enables and disables the accelerometer and the auxiliary 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 is controlled in the Registers ACC_CONF.

In all global power configurations both register contents and FIFO contents are retained.

Low Power Mode: This power configuration aggressively reduces power of the device as much as possible. The low power mode configuration is activated through enabling PWR_CONF.adv_power_save=0b1 and disabling ACC_CONF.acc_perf_mode=0b0. In this configuration these externally user visible features may not be available:

- Register writes need an inter-write-delay of at least **1000 us**.
- The sensors log data into the FIFO in performance and low power mode. When the FIFO watermark interrupt is active, the FIFO is accessible for reading in low power mode until a burst read operation on Register FIFO_DATA completes when PWR_CONF.fifo_self_wakeup=0b1. When PWR_CONF.fifo_self_wakeup=0b0, the user needs to disable advanced power save mode (PWR_CONF.adv_power_save=0b0) and wait for 250 µs before reading the FIFO.
- To read out FIFO data w/o a FIFO watermark interrupt, the advanced power save configuration needs to be disabled (PWR_CONF.adv_power_save=0b0)

Table 3: Examples with the optimal power configurations:

Usecase	ACC_CONF. acc_perf_mode	PWR_CONF .adv_power _save	PWR_CTRL.acc_en	Power consumption
Configuration mode	x	0	x	
Suspend (lowest power mode)	x	1	0	suspend power
Performance mode	1	x	1	Accel works in continuous mode
Low power mode	0	1	1	Depends on ACC_CONF

The PWR_CTRL register is used to enable and disable sensors. Per default, all sensors are disabled. Acceleration sensor must be enabled by setting PWR_CTRL.acc_en=0b1.
The auxiliary sensor functionality is supported only when the auxiliary interface is connected for the auxiliary sensor operation. 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:

- 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.
- 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 BMA490L auxiliary sensor interface into manual mode by AUX_IF_CONF.aux_manual_en=0b1 and then communicating with the auxiliary sensor part through the BMA490L registers AUX_RD_ADDR, AUX_WR_ADDR, and AUX_WR_DATA. For details see Chapter 4.9.

Table 4: Current consumption in low power mode

Current Consumption[2] [µA] depending on number of averaged samples in low power mode								
ODR	No Avg	Avg 2	Avg 4	Avg 8	Avg 16	Avg 32	Avg 64	Avg 128
ODR 0.78	3	3	3	4	4	5	7	12
ODR_1.56	3	3	3	4	4	6	10	15
ODR_3.125	4	4	4	6	8	12	21	39
ODR_6.25	4	5	6	8	13	22	40	77
ODR_12.5	6	7	9	14	23	40	77	152
ODR_25	8	11	14	24	43	79	152	152
ODR_50	14	18	27	45	83	152	152	152
ODR_100	22	32	51	87	152	152	152	152
ODR_200	42	60	97	152	152	152	152	152
ODR_400	80	118	152	152	152	152	152	152

4.5 Sensor Data

4.5.1 Acceleration 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).

4.5.2 Filter Settings

The accelerometer digital filter can be configured through the Register ACC_CONF.

Note:

Illegal settings in configuration registers will result in an error code in Register ERR_REG. The content of the data register is undefined, and if the FIFO is used, it may contain no value.

[2] Current consumption based on limited lab measurements. Only for reference.

4.5.3 Accelerometer data processing for performance mode

Performance mode is enabled with ACC_CONF.acc_perf_mode=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 12.5Hz up to 1600Hz.

The filter bandwidth shows a 3db cutoff frequency shown in the following table:

Table 5: 3dB cutoff frequency of the accelerometer according to ODR with normal filter mode

Accelerometer ODR [Hz]	**12.5**	**25**	**50**	**100**	**200**	**400**	**800**	**1600**
3dB Cutoff frequency [Hz]	5.06	10.12	20.25	40.5	80	162 (155 for Z axis)	324 (262 for Z axis)	684 (353 for Z axis)

The noise is also depending on the filter settings and ODR, see table below.

Table 6: Accelerometer noise in mg according to ODR with normal filter mode (range +/- 4g) (based on device measurement)

ODR in Hz	**25**	**50**	**100**	**200**	**400**
RMS-Noise (typ.) [mg]	0.5	0.7	0.9	1.3	1.7

4.5.4 Accelerometer data processing for low power mode

Low power mode can be enabled by PWR_CONF.adv_power_save=0b1 and ACC_CONF.acc_perf_mode=0b0. In this power mode, the accelerometer regularly changes between a suspend power mode phase where no measurement is performed and a performance power mode phase, where data is acquired. The period of the duty cycle for changing between suspend and performance 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 normal mode 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:

averaged samples = $2^{(Val(acc\_bwp))}$
skipped samples = (1600/ODR)-averaged samples

A higher number of averaged samples will result in a lower noise level of the signal, but since the performance power mode phase is increased, the power consumption will also rise.

4.5.5 Data Ready Interrupt

This interrupt fires whenever a new data sample set from accelerometer, or auxiliary sensor is complete. This allows a low latency data readout. In non-latched mode, the interrupt and the flag in Register INT_STATUS_1 are cleared automatically after 1/(3200Hz). If this automatic clearance is unwanted, latched-mode can be used.
In order to enable/use the data ready interrupt map it on the desired interrupt pin via INT_MAP_DATA.

4.5.6 Temperature Sensor

The temperature sensor has 8 bits. The temperature value is defined in Register TEMPERATURE and updated every 1.28 s.

Table 7: The temperature sensor is always on, when the accelerometer sensor is active.

Value	Temperature
0x7F	150 °C
…	…
0x00	23 °C
…	…
0x81	-104 °C
0x80	Invalid

When there is no valid temperature information available (i.e. last measurement before the time defined above), the temperature indicates an invalid value: 0x80.

4.5.7 Sensor Time

The BMA490L 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

Table 8: Bit m in sensor_time with Resolution in [s]

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

Table 9: Bit m in sensor_time with Resolution in [ms]

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

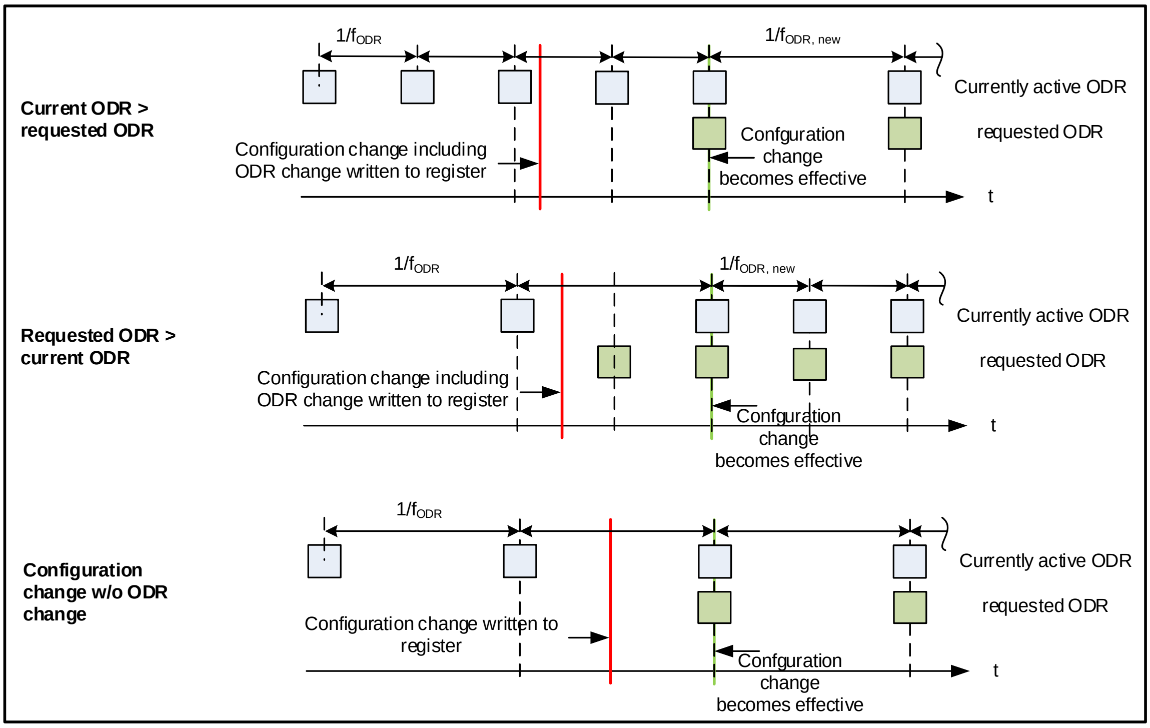
Bit *m* in sensor_time	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Resolution [ms]	5	2.5	1.250	0.625	0.3125	0.156	0.078	0.039
Update rate [Hz]	200	400	800	1600	3200			

The sensortime is synchronized with the data capturing in the data register and the FIFO. Between the data sampling and the data capturing there is a delay which depends on the settings in the Register ACC_CONF. 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 deliver always consistent values, i.e. the value of the register does not change during the burst read.

4.5.8 Configuration Changes

If accelerometer configuration settings in registers ACC_CONF, ACC_RANGE, or AUX_CONF are changed while the accelerometer (PWR_CTRL.acc_en = 0b1) or auxiliary sensor (PWR_CTRL.aux_en = 0b1) is enabled, 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.


Current ODR > requested ODR
1/f_ODR
1/f_ODR, new
Currently active ODR
requested ODR
Configuration change including ODR change written to register
Confguration change becomes effective
t
Requested ODR > current ODR
1/f_ODR
1/f_ODR, new
Currently active ODR
requested ODR
Configuration change including ODR change written to register
Confguration change becomes effective
t
Configuration change w/o ODR change
1/f_ODR
Currently active ODR
requested ODR
Configuration change written to register
Confguration change becomes effective
t

Due to filter settling, some invalid samples can be suppressed in addition after a configuration change.

4.6 **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 depth is 1024 byte and supports the following interrupts:

- FIFO full interrupt
- FIFO watermark interrupt

FIFO is enabled with FIFO_CONFIG_1.fifo_acc_en=0b1 (to enable FIFO for accelerometer data, 0b0= disabled), or set FIFO_CONFIG_1.fifo_aux_en=0b1 (to enable the FIFO for the auxiliary interface (magnetometer), 0b0=disabled).

4.6.1 Frames

The FIFO captures data in frames, which consist of a header and a payload. The FIFO can be configured to skip the header (headerless mode) in which case only payload is stored.

- In header mode (standard configuration) each regular frame consists of a one byte header describing properties of the frame, (which sensors are included in this frame) and the data itself. Beside the regular frames, there are control frames.
- In headerless mode the FIFO contains sampled data only.

Header mode

The header has a 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	Enabled sensors	Tag of INT2 and INT1
0b01	Control	Control opcode	
0b00 and 0b11	Reserved	Na	

fh_parm=0b0000 is invalid for regular mode, a header of 0x80 indicates an uninitialized frame.

In a regular frame, fh_parm frame 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	Reserved	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 data format for data frames is identical to the format defined for the Register (0x0A) DATA_0 to Register (0x17) DATA_13 register. Only frames which contain data of at least one sensor will be written into the FIFO. E.g. fh_parm=0b0101 the data in the frame are shown below. If the read burst length is less than 8 byte, the number of auxiliary sensor data in the frame is reduced to the burst length.

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	ACC_X<7:0> (LSB)	
X=9	ACC_X<15:8> (MSB)	
X=10	ACC_Y<7:0> (LSB)	
X=11	ACC_Y<15:8> (MSB)	
X=12	ACC_Z<7:0> (LSB)	
X=13	ACC_Z<15:8> (MSB)	

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 chapter 4.8). If the burst length is less than 8, BMA490L 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 and accelerometer 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) 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	ACC_X<7:0> (LSB)	
X=9	ACC_X<15:8> (MSB)	
X=10	ACC_Y<7:0> (LSB)	
X=11	ACC_Y<15:8> (MSB)	
X=12	ACC_Z<7:0> (LSB)	
X=13	ACC_Z<15:8> (MSB)	

4.6.2 Conditions and Details

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 un-filtered data are selected in register FIFO_DOWNS.acc_fifo_filt_data for the accelerometer, the sample rate is 1600 Hz. The input data rate to the FIFO can be reduced by selecting a down-sampling factor 2^k in register FIFO_DOWNS.acc_fifo_downs, where k=[0,1..7].

FIFO Overflow

In the case of 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. When FIFO_CONFIG_0.fifo_stop_on_full *=0b0,* the FIFO logic may delete the oldest frames. If header mode is enabled, the skip frame is prepended at the next FIFO readout, when the free FIFO space falls below the maximum size frame.
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.

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 and includes the control frames, with the exception of the sensortime frame.

fh_mode<3:0>	Definition
0x0	Skip Frame
0x1	Sensortime Frame
0x2	Fifo_Input_Config Frame
0x3	Reserved
0x4	Sample Drop Frame
0x5 – 0x7	Reserved

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.

Sensortime Frame (fh_parm=0x1):
The data for the sensortime frame consists content of the Register SENSORTIME_0 to SENSORTIME_2 when the last byte of the last sample frame was read. A 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).

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 one byte of data with the format

Bit	7	6	5	4	3	2	1	0
Content	reserved		aux_ if_ch	aux_ conf_ch	reserved	reserved	acc_ range_ch	acc_ conf_ch

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.
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.

Sample Drop Frame

A sample drop frame has always one byte payload, defined through

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Content	reserved					aux_drop	reserved	acc_drop

Sample drop frame will be inserted after a Fifo_Input_Config frame at the ODR tick at which the sample was dropped and only if no other sensor provides a valid sample at this ODR tick. If another sensor provides valid data, the data of this sensor is just not included and the appropriate header bit of the data frame is not set.

Sample drop frames will be inserted only for transition phases after configuration changes, not for samples dropped between sensor enable and first valid sample. For a detailed description of configuration changes see Section 4.5, Subsection “Configuration Changes”.

FIFO Partial frame reads

When a frame is only partially read through the Register FIFO_DATA 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 may be deleted.

FIFO overreads

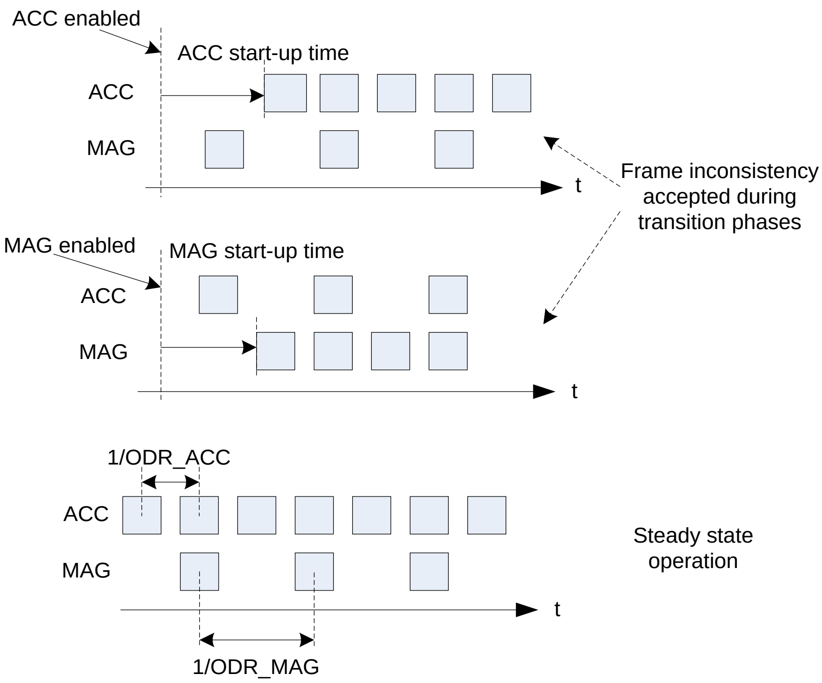
When more data are read from the FIFO than it contains valid data, 0x8000 is returned in headerless mode. While in header mode 0x0080 is returned, where 0x80 indicates an invalid frame.

4.6.3 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 magnetic 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.


ACC enabled
ACC start-up time
ACC
MAG
t
Frame inconsistency accepted during transition phases
MAG enabled
MAG start-up time
ACC
MAG
t
1/ODR_ACC
ACC
MAG
t
1/ODR_MAG
Steady state operation

4.6.4 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/or FIFO_CONFIG_1.fifo_tag_int2_en need to be enabled, as well as INT1_IO_CTRL.input_en and/or 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.

4.6.5 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_0 and FIFO_WTM_1.

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

Both interrupts are suppressed when a read operation on the Register FIFO_DATA is ongoing.
Latched FIFO interrupts will only get cleared, if the status register gets read and the fill level is below the corresponding FIFO interrupt (full or watermark).

4.6.6 FIFO Flush

The user can trigger a FIFO reset by writing the command fifo_flash (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

4.7 Integrated Features set:

4.7.1 Any Motion / No motion detection

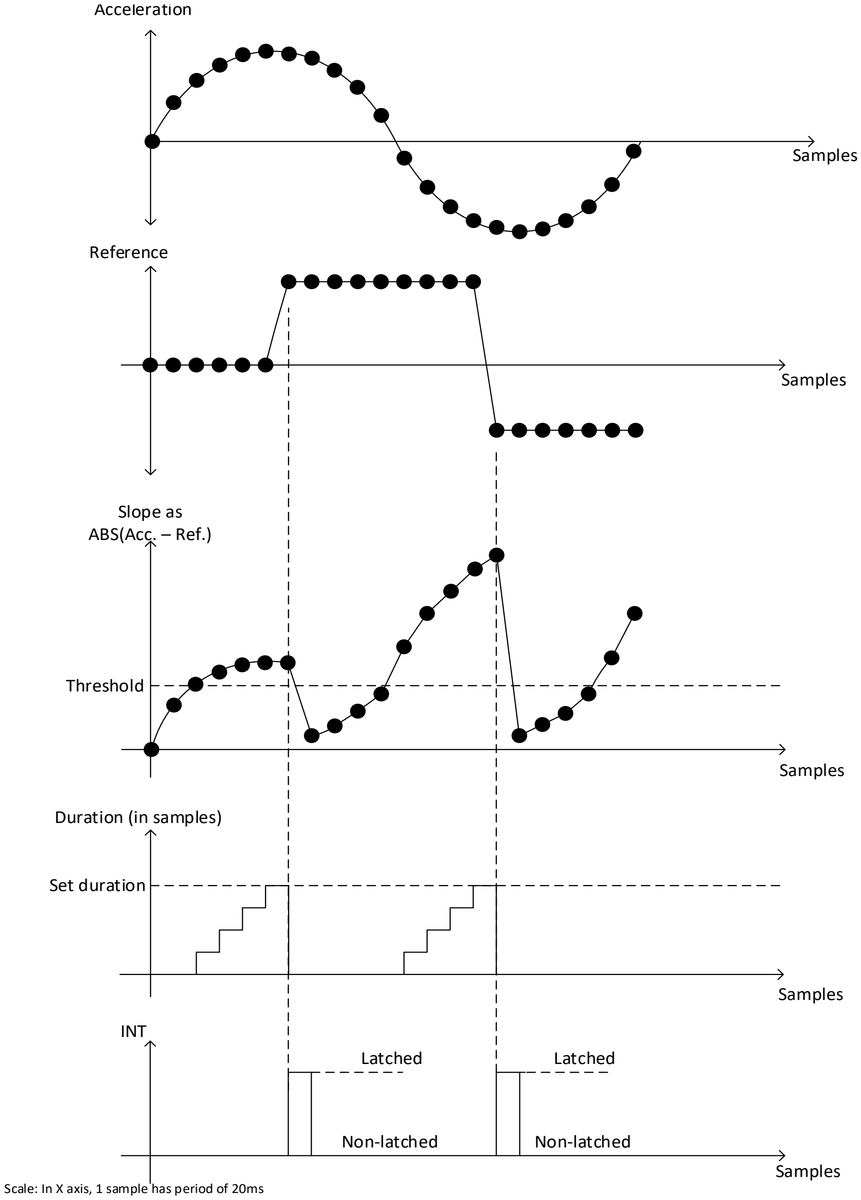
Any-motion detection:

Any-motion detection uses the slope between current input and reference acceleration samples to detect the motion status of the device. Feature can be enabled by setting at least one of the following: FEATURES_IN.any_motion.settings_2.x_en, FEATURES_IN.any_motion.settings_2.y_en and FEATURES_IN.any_motion.settings_2.z_en, respectively for each axis.

Any-motion provides an interrupt when the absolute value of the slope exceeds the configurable FEATURES_IN.any_motion.settings_1.threshold for consecutive FEATURES_IN.any_motion.settings_2.duration samples for at-least one of the enabled sensing axis.

Reference acceleration sample is updated only when an any-motion interrupt is triggered. The interrupt status is reset as soon as the slope falls below the set FEATURES_IN.any_motion.setings_1.threshold value. The signals and timings relevant to the any-motion interrupt functionality are depicted in the figure below:

Signal and timing diagram for any-motion interrupt detection



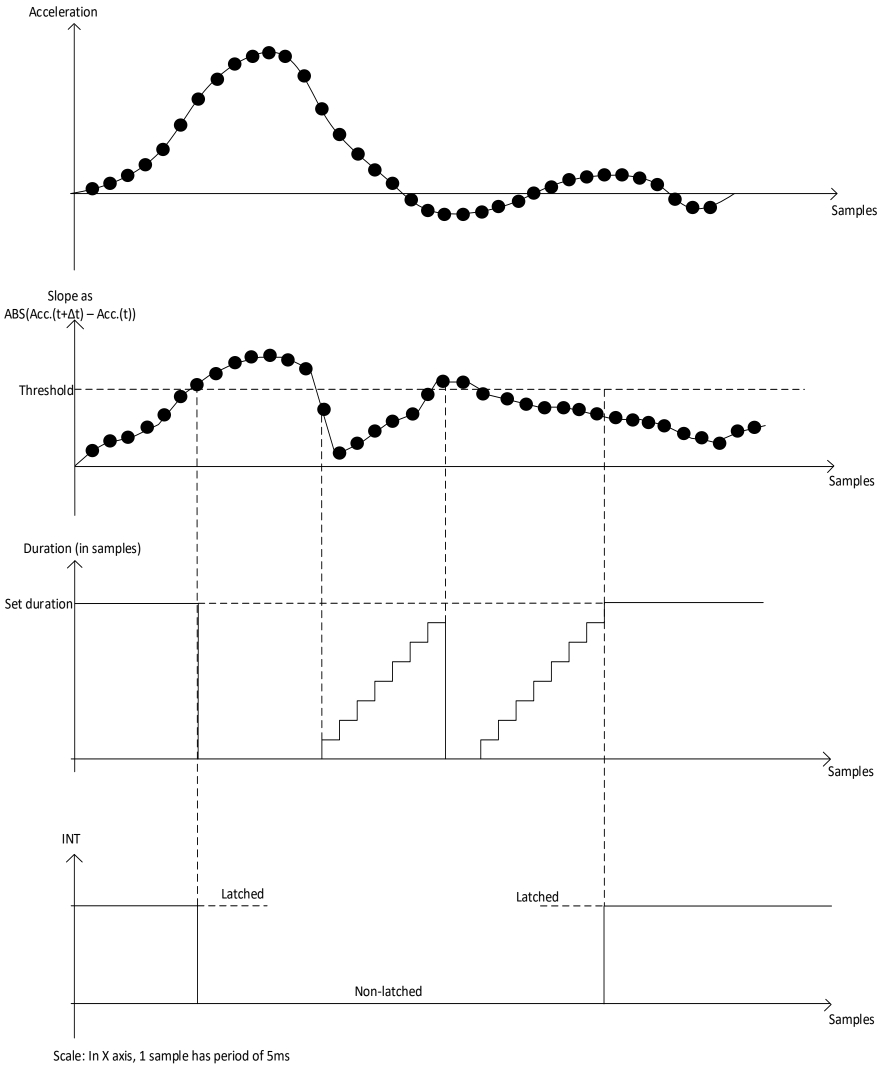
Configuration settings:

1. FEATURES_IN.any_motion.settings_1.threshold – the slope threshold.
2. FEATURES_IN.any_motion.settings_2.duration – the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion.
3. FEATURES_IN.any_motion.settings_2.x_en – indicates if this feature is enabled for x axis
4. FEATURES_IN.any_motion.settings_2.y_en – indicates if this feature is enabled for y axis
5. FEATURES_IN.any_motion.settings_2.z_en –indicates if this feature is enabled for z axis

No Motion Detection:

No-motion detection uses the slope between two consecutive acceleration signal samples to detect static state of the device. Feature can be enabled by setting at least one of the following flags: FEATURES_IN.no_motion.settings_2.x_en, FEATURES_IN.no_motion.settings_2.y_en and FEATURES_IN.no_motion.settings_2.z_en, respectively for each axis.
No-motion interrupt is triggered when the slope on all enabled sensing axis remains smaller than the configurable FEATURES_IN.no_motion.settings_1.threshold for the duration configured by FEATURES_IN.no_motion.settings_2.duration. No-motion interrupt is cleared as soon as the acceleration slope exceeds the set threshold. The signals and timings relevant to the no-motion interrupt functionality are depicted in the figure below.

Signal and timing diagram for no-motion interrupt detection



Register FEATURES_IN.no_motion.settings_2.duration defines the number of consecutive data points for which the slope of enabled axis must be smaller than the threshold for an interrupt to be asserted.

Configuration settings:

1. FEATURES_IN.no_motion.settings_1.threshold – the slope threshold.
2. FEATURES_IN.no_motion.settings_2.duration – the number of consecutive data points for which the threshold condition must be respected, for interrupt assertion.
3. FEATURES_IN.no_motion.settings_2.x_en – indicates if this feature is enabled for x axis
4. FEATURES_IN.no_motion.settings_2.y_en – indicates if this feature is enabled for y axis
5. FEATURES_IN.no_motion.settings_2.z_en –indicates if this feature is enabled for z axis

4.8 **General Interrupt Pin configuration**

Electrical Interrupt Pin Behavior

Both interrupt pins INT1 and INT2 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 the respective FIFO chapter) If both are enabled, the input (e.g. marking FIFO) is driven by the interrupt output.
BMA490L supports edge and level triggered interrupt inputs, this can be configured through INT1_IO_CTRL.edge_ctrl respectively INT2_IO_CTRL.edge_ctrl.

BMA490L supports non-latched and latched interrupts modes for data-ready, FIFO full and FIFO watermark. The mode is selected by INT_LATCH.int_latch. The feature interrupts described in chapter FIFO Interrupts, support only latched mode described below.
In latched mode an asserted interrupt status in INT_STATUS_0 or INT_STATUS_1 and the selected pin are cleared if the corresponding status register is read. If more than one interrupt pin is used in latched mode, all interrupts in INT_STATUS_0 should be mapped to one pin and all interrupts in INT_STATUS_1 should be mapped to the other pin. If just one interrupt pin is used all interrupts may be mapped to this pin. If the activation condition still holds when it is cleared, the interrupt status is asserted again when the interrupt condition holds again.
In the non-latched mode (only for data-ready, FIFO full and FIFO watermark) the interrupt status bit and the selected pin are reset as soon as the activation condition is not valid anymore.

Interrupt Pin Mapping

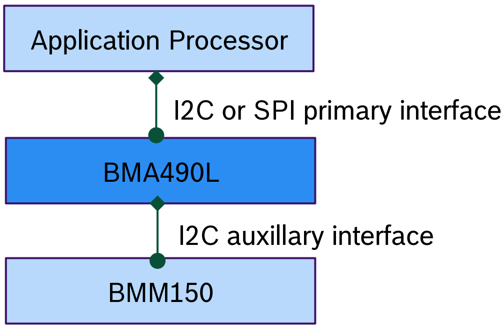
In order, for the Host to react to the features output, they can be mapped to the external pin INT1 or pin INT2, by setting the corresponding bits from the registers INT1_MAP, respectively INT2_MAP.

To disconnect the features outputs to the external pins, the same corresponding bits must be reset, from the registers, INT1_MAP, respectively INT2_MAP.

Once a feature triggered the output pin, the Host can read out the corresponding bit from the register, INT_STATUS_0 (Feature Interrupts) or INT_STATUS_1 (FIFO and data ready).

4.9 Auxiliary Sensor Interface

The auxiliary interface allows to attach one auxiliary sensor (e.g. magnetometer) on dedicated auxiliary sensor interface as shown below.



6 DOF Solution w/ BMA490L and BMM150

4.9.1 Structure and Concept

The BMA490L 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 BMA490L 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 (in setup mode as described in following section).

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 BMA490L FIFO for auxiliary sensor data (BMM150 does not have a FIFO). This is important for monitoring applications.

4.9.2 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 accelerometer sensor itself (this must be done in setup mode).

There are three basis configurations/modes 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 BMA490L 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 BMA490L registers and may be read out from BMA490L 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

The auxiliary sensor interface mode may be enabled by setting bit IF_CONF.if_mode according to the following table.

IF_CONF.if_mode	Result
0	Secondary IF disabled (default)
1	AuxIF enabled

The auxiliary sensor interface operates at 400 kHz. This results in an I2C readout delay of about 250 us for 10 bytes of data.

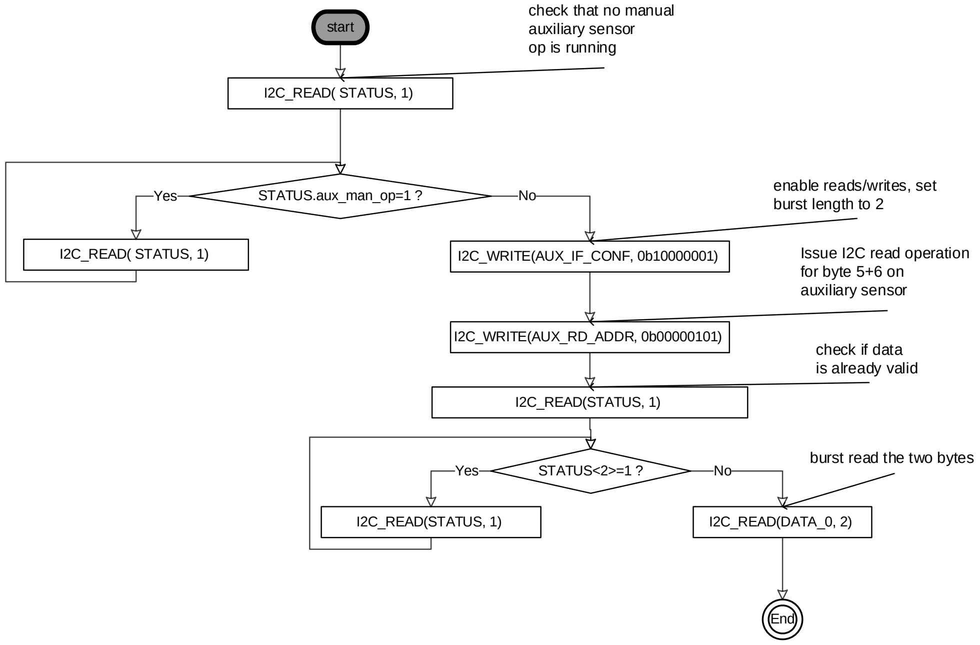
The I2C slave address of the auxiliary sensor is defined in AUX_DEV_ID. i2c_device_addr.

4.9.3 Setup mode (AUX_IF_CONF.aux_manual_en =0b1)

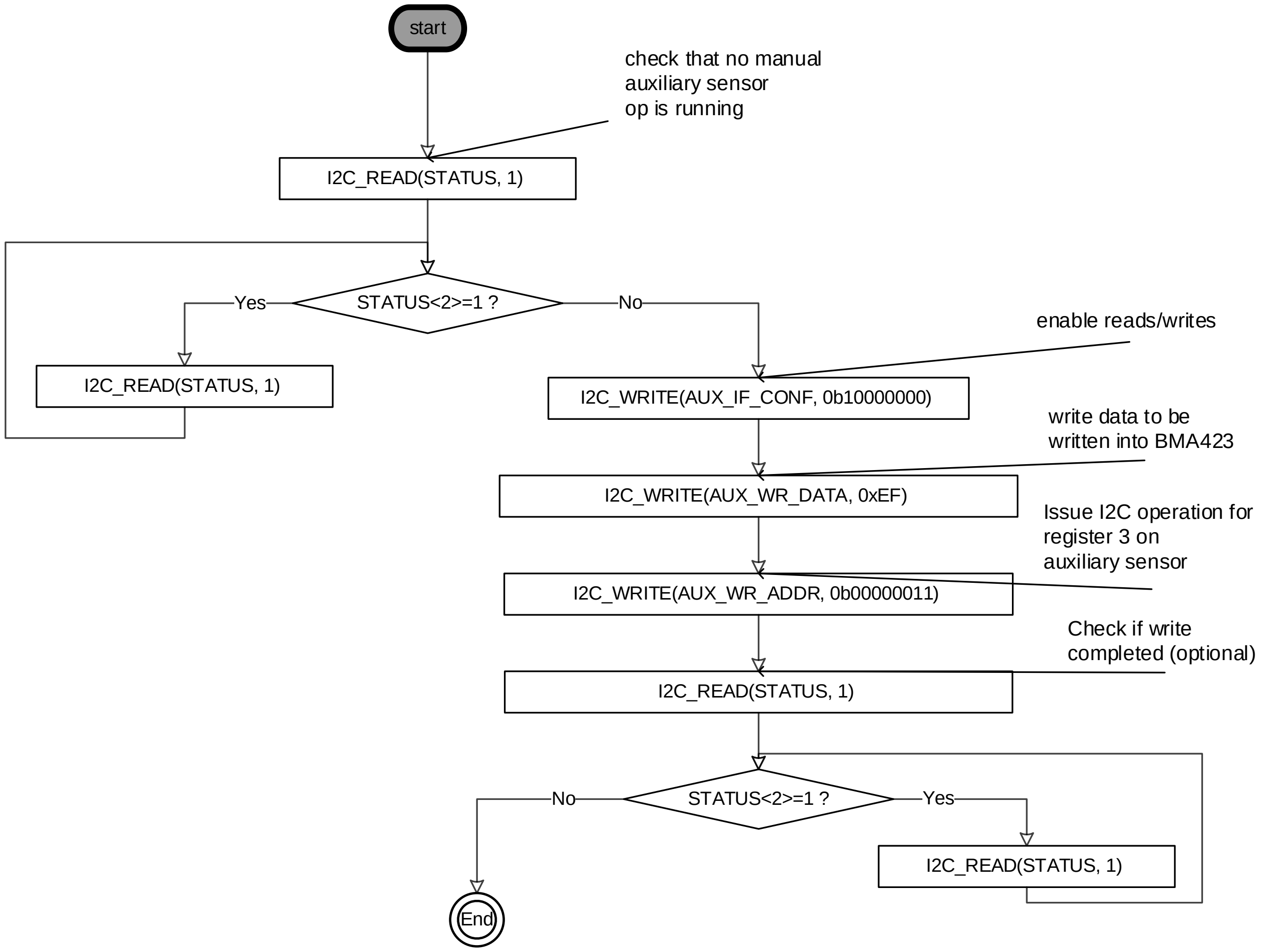
Through the primary interface the auxiliary sensor may be accessed using indirect addressing through the AUX_* registers. AUX_RD_ADDR and AUX_WR_ADDR define the address of the register to read/write in the auxiliary sensor register map and triggers the operation itself, when the auxiliary sensor interface 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 BMA490L 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_man_op 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


start
check that no manual auxiliary sensor op is running
I2C_READ(STATUS, 1)
Yes
STATUS.aux_man_op=1 ?
No
I2C_READ(STATUS, 1)
enable reads/writes, set burst length to 2
I2C_WRITE(AUX_IF_CONF, 0b10000001)
Issue I2C read operation for byte 5+6 on auxiliary sensor
I2C_WRITE(AUX_RD_ADDR, 0b00000101)
check if data is already valid
I2C_READ(STATUS, 1)
Yes
STATUS<2>=1 ?
No
burst read the two bytes
I2C_READ(STATUS, 1)
I2C_READ(DATA_0, 2)
End

Example: Write 0xEF into register 3 of auxiliary sensor



4.9.4 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 data ready status is set in STATUS.drdy_aux, it is typically cleared through reading one of the DATA_0 to DATA_7 registers.
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. 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.if_mode=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.if_mode=0b1) again.

4.9.5 Delay (Time Offset)

BMA490L 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.

4.10 Sensor Self-Test

The BMA490L has a comprehensive self test function for the MEMS element by applying electrostatic forces to the sensor core instead of external accelerations. By actually deflecting the seismic mass, the entire signal path of the sensor can be tested. 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.
Before the self-test is enabled the g-range should be set to 8g. The self-test is activated for all axes by writing ACC_SELF_TEST.acc_self_test_en = 1b1. The self-test is disabled by writing ACC_SELF_TEST.acc_self_test_en = 1b0. It is possible to control the direction of the deflection through bit ACC_SELF_TEST.acc_self_test_sign. The excitation occurs in positive (negative) direction if ACC_SELF_TEST.acc_self_test_sign= 1b1 ('b0). The amplitude of the deflection has to be set low by writing ACC_SELF_TEST.acc_self_test_amp = 1b0. After the self-test is enabled, the user should wait 50ms before interpreting the acceleration data.
In order to ensure a proper interpretation of the self-test signal it is recommended to perform the self-test for both (positive and negative) directions and then to calculate the difference of the resulting acceleration values. 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 for BMA490L.

	x-axis signal	**y-axis signal**	**z-axis signal**
BMA490L	1800 mg	1800 mg	1800 mg

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.

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 low by setting ACC_SELF_TEST.acc_self_test_amp = 1b0
4. Set ACC_CONF.acc_odr=1600Hz, Continuous sampling mode, ACC_CONF.acc_bwp=norm_avg4, ACC_CONF.acc_perf_mode=1b1.
5. Wait for > 2 ms
6. Enable self-test and set positive self-test polarity (ACC_SELF_TEST.acc_self_test_sign= 1b1)
7. Wait for > 50ms
8. Read and store positive acceleration value of each axis from registers DATA_8 to DATA_13
9. Enable self-test and set negative self-test polarity ACC_SELF_TEST.acc_self_test_sign= 1b0)
10. Wait for > 50ms
11. Read and store negative acceleration value of each axis from registers DATA_8 to DATA_13
12. Calculate difference of positive and negative acceleration values and compare against threshold values

4.11 Offset Compensation

BMA490L 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 public 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

4.11.1 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.

4.11.2 Inline Calibration

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.

4.12 **Non-Volatile Memory**

The registers NV_CONF and OFFSET_0 to OFFSET_2 have an NVM backup which are accessible by the user.

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. Write 0b1 to bit NVM_CONF.nvm_prog_en in order to unlock the NVM.
4. Write nvm_prog to the CMD register to trigger the write process.
5. Write 0b0 to bit NVM_CONF.nvm_prog_en in order to lock the NVM, after the write process is completed

Writing to the NVM always renews the entire NVM contents. 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.

4.13 **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

Registers can be read and written in all power configurations with the exception of FEATURES_IN and FIFO_DATA which need PWR_CONF.adv_power_save set to 0b0. The following chapter contains only the general register map, feature related registers are excluded.

5.2 Register Map

read/write	read only	write only	reserved

ID:

Register Address	Register Name	Default Value	7	6	5	4	3	2	1	0
0x7E	CMD	0x00	cmd							
0x7D	PWR_CTRL	0x00	reserved					acc_en	reserved	aux_en
0x7C	PWR_CONF	0x03	reserved						fifo_self_wakeup	adv_power_save
0x7B	-	-	reserved							
...	-	-	reserved							
0x74	-	-	reserved							
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	-	-	reserved							
0x6D	ACC_SELF_TEST	0x00	reserved				acc_self_test_amp	acc_self_test_sign	reserved	acc_self_test_en
0x6C	-	-	reserved							
0x6B	IF_CONF	0x00	reserved			if_mode	reserved			spi3
0x6A	NVM_CONF	0x00	reserved						nvm_prog_en	reserved
0x69	-	-	reserved							
...	-	-	reserved							
0x60	-	-	reserved							
0x5F	INTERNAL_ERROR	0x00	reserved					int_err_2	int_err_1	reserved
0x5E	FEATURES_IN	0x00	features_in							
0x5D	-	-	reserved							
...	-	-	reserved							
0x5A	-	-	reserved							

0x59	INIT_CTRL	0x90	init_ctrl							
0x58	INT_MAP_DATA	0x00	reserved	int2_drdy	int2_fwm	int2_ffull	reserved	int1_drdy	int1_fwm	int1_ffull
0x57	INT2_MAP	0x00	error_int_out	no_motion_out	any_motion_out	reserved				
0x56	INT1_MAP	0x00	error_int_out	no_motion_out	any_motion_out	reserved				
0x55	INT_LATCH	0x00	reserved							int_latch
0x54	INT2_IO_CTRL	0x00	reserved			input_en	output_en	od	lvl	edge_ctrl
0x53	INT1_IO_CTRL	0x00	reserved			input_en	output_en	od	lvl	edge_ctrl
0x52	-	-	reserved							
...	-	-	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	reserved					aux_rd_burst	
0x4B	AUX_DEV_ID	0x20	i2c_device_addr							reserved
0x4A	-	-	reserved							
0x49	FIFO_CONFIG_1	0x10	reserved	fifo_acc_en	fifo_aux_en	fifo_header_en	fifo_tag_int1_en	fifo_tag_int2_en	reserved	
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	0x80	acc_fifo_filt_data	acc_fifo_downs			reserved			
0x44	AUX_CONF	0x46	aux_offset				aux_odr			
0x43	-	-	reserved							
0x42	-	-	reserved							
0x41	ACC_RANGE	0x01	reserved						acc_range	
0x40	ACC_CONF	0xA8	acc_perf_mode	acc_bwp			acc_odr			
0x3F	-	-	reserved							

...	-	-	reserved							
0x2B	-	-	reserved							
0x2A	INTERNAL_STATUS	0x00	reserved	odr_50Hz_error	axes_remap_error	message				
0x29	-	-	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	-	-	reserved							
0x22	TEMPERATURE	0x00	temperature							
0x21	-	-	reserved							
...	-	-	reserved							
0x1E	-	-	reserved							
0x1D	INT_STATUS_1	0x00	acc_drdy_int	reserved	aux_drdy_int	reserved			fwm_int	ffull_int
0x1C	INT_STATUS_0	0x00	error_int_out	no_motion_out	any_motion_out	reserved				
0x1B	EVENT	0x01	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_13	0x00	acc_z_15_8							
0x16	DATA_12	0x00	acc_z_7_0							
0x15	DATA_11	0x00	acc_y_15_8							
0x14	DATA_10	0x00	acc_y_7_0							
0x13	DATA_9	0x00	acc_x_15_8							
0x12	DATA_8	0x00	acc_x_7_0							
0x11	DATA_7	0x00	aux_r_15_8							
0x10	DATA_6	0x00	aux_r_7_0							
0x0F	DATA_5	0x00	aux_z_15_8							
0x0E	DATA_4	0x00	aux_z_7_0							
0x0D	DATA_3	0x00	aux_y_15_8							
0x0C	DATA_2	0x00	aux_y_7_0							
0x0B	DATA_1	0x00	aux_x_15_8							
0x0A	DATA_0	0x00	aux_x_7_0							
0x09	-	-	reserved							
...	-	-	reserved							

0x04	-	-	reserved							
0x03	STATUS	0x10	drdy_acc	reserved	drdy_aux	cmd_rdy	reserved	aux_man_op	reserved	
0x02	ERR_REG	0x00	aux_err	fifo_err	reserved	error_code			cmd_err	fatal_err
0x01	-	-	reserved							
0x00	CHIP_ID	0x1A	chip_id							

FEATURES_IN

Register Address	**Register Name**	**Default Value**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x5E: 0x0B	general_settings.axes_remapping[1]	0x00	reserved							map_z_axis_sign
0x5E: 0x0A	general_settings.axes_remapping[0]	0x88	map_z_axis		map_y_axis_sign	map_y_axis		map_x_axis_sign	map_x_axis	
0x5E: 0x09	general_settings.Reserved[1]	0x00	Reserved							
0x5E: 0x08	general_settings.Reserved[0]	0x00	Reserved							
0x5E: 0x07	no_motion.settings_2[1]	0x00	z_en	y_en	x_en	duration				
0x5E: 0x06	no_motion.settings_2[0]	0x05	duration							
0x5E: 0x05	no_motion.settings_1[1]	0x00	reserved					threshold		
0x5E: 0x04	no_motion.settings_1[0]	0xAA	threshold							
0x5E: 0x03	any_motion.settings_2[1]	0x00	z_en	y_en	x_en	duration				
0x5E: 0x02	any_motion.settings_2[0]	0x05	duration							
0x5E: 0x01	any_motion.settings_1[1]	0x00	reserved					threshold		

0x5E: 0x00	any_motion.settings_1[0]	0xAA	threshold

5.2.1 Register (0x00) CHIP_ID

DESCRIPTION: Chip identification code
RESET: 0x1A
DEFINITION (Go to register map):

Name	**Register (0x00) CHIP_ID**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	1
Content	chip_id			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	1	0	1	0
Content	chip_id			

chip_id: Chip identification code for BMA490L

5.2.2 Register (0x02) ERR_REG

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

Name	**Register (0x02) ERR_REG**			
Bit	7	6	5	4
Read/Write	R	R	n/a	R
Reset Value	0	0	0	0
Content	aux_err	fifo_err	reserved	error_code
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	error_code		cmd_err	fatal_err

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.
cmd_err: Command execution failed.
error_code: Error codes for persistent errors

error_code		
0x00	no_error	no error is reported
0x01	acc_err	error in Register ACC_CONF

fifo_err: Error in FIFO detected: Input data was discarded in stream mode. This flag will be reset when read.
aux_err: Error in I2C-Master detected. This flag will be reset when read.

5.2.3 Register (0x03) STATUS

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

Name	**Register (0x03) STATUS**			
Bit	7	6	5	4
Read/Write	R	n/a	R	R
Reset Value	0	0	0	1
Content	drdy_acc	reserved	drdy_aux	cmd_rdy
Bit	3	2	1	0
Read/Write	n/a	R	n/a	n/a
Reset Value	0	0	0	0
Content	reserved	aux_man_op	reserved	

aux_man_op: ‘1’(‘0’) indicate a (no) manual auxiliary interface operation is ongoing.
cmd_rdy: CMD decoder status. `0´ -> Command in progress `1´ -> Command decoder is ready to accept a new command
drdy_aux: Data ready for auxiliary sensor. It gets reset when one auxiliary DATA register is read out
drdy_acc: Data ready for accelerometer. It gets reset when one accelerometer DATA register is read out

5.2.4 Register (0x0A) DATA_0

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

Name	**Register (0x0A) DATA_0**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_x_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_x_7_0			

5.2.5 Register (0x0B) DATA_1

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

Name	**Register (0x0B) DATA_1**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_x_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_x_15_8			

5.2.6 Register (0x0C) DATA_2

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

Name	**Register (0x0C) DATA_2**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_y_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_y_7_0			

5.2.7 Register (0x0D) DATA_3

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

Name	**Register (0x0D) DATA_3**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_y_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_y_15_8			

5.2.8 Register (0x0E) DATA_4

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

Name	**Register (0x0E) DATA_4**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_z_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_z_7_0			

5.2.9 Register (0x0F) DATA_5

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

Name	**Register (0x0F) DATA_5**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_z_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_z_15_8			

5.2.10 Register (0x10) DATA_6

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

Name	**Register (0x10) DATA_6**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_r_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_r_7_0			

5.2.11 Register (0x11) DATA_7

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

Name	**Register (0x11) DATA_7**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_r_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	aux_r_15_8			

5.2.12 Register (0x12) DATA_8

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

Name	**Register (0x12) DATA_8**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_x_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_x_7_0			

5.2.13 Register (0x13) DATA_9

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

Name	**Register (0x13) DATA_9**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_x_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_x_15_8			

5.2.14 Register (0x14) DATA_10

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

Name	**Register (0x14) DATA_10**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_y_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_y_7_0			

5.2.15 Register (0x15) DATA_11

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

Name	**Register (0x15) DATA_11**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_y_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_y_15_8			

5.2.16 Register (0x16) DATA_12

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

Name	**Register (0x16) DATA_12**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_z_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_z_7_0			

5.2.17 Register (0x17) DATA_13

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

Name	**Register (0x17) DATA_13**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_z_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	acc_z_15_8			

5.2.18 Register (0x18) SENSORTIME_0

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

Name	**Register (0x18) SENSORTIME_0**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_7_0			

sensor_time_7_0: Sensor time <7:0> in units of 39.0625 us.

5.2.19 Register (0x19) SENSORTIME_1

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

Name	**Register (0x19) SENSORTIME_1**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_15_8			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_15_8			

sensor_time_15_8: Sensor time <15:8> in units of 10 ms.

5.2.20 Register (0x1A) SENSORTIME_2

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

Name	**Register (0x1A) SENSORTIME_2**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_23_16			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	sensor_time_23_16			

sensor_time_23_16: Sensor time <23:16> in units of 2.56 s.

5.2.21 Register (0x1B) EVENT

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

Name	**Register (0x1B) EVENT**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	R
Reset Value	0	0	0	1
Content	reserved			por_detected

por_detected: '1' after device power up or softreset. Clear-on-read

5.2.22 Register (0x1C) INT_STATUS_0

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

Name	**Register (0x1C) INT_STATUS_0**			
Bit	7	6	5	4
Read/Write	R	R	R	n/a
Reset Value	0	0	0	0
Content	error_int_out	no_motion_out	any_motion_out	reserved
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			

any_motion_out: Any-motion detection output

no_motion_out: No-motion detection output
error_int_out: Error interrupt output

5.2.23 Register (0x1D) INT_STATUS_1

DESCRIPTION: Interrupt Status. This register will be cleared on read.
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x1D) INT_STATUS_1**			
Bit	7	6	5	4
Read/Write	R	n/a	R	n/a
Reset Value	0	0	0	0
Content	acc_drdy_int	reserved	aux_drdy_int	reserved
Bit	3	2	1	0
Read/Write	n/a	n/a	R	R
Reset Value	0	0	0	0
Content	reserved		fwm_int	ffull_int

ffull_int: FIFO Full Interrupt
fwm_int: FIFO Watermark Interrupt
aux_drdy_int: Auxiliary sensor data ready interrupt
acc_drdy_int: Accelerometer data ready interrupt

5.2.24 Register (0x22) TEMPERATURE

DESCRIPTION: Contains the temperature value of the sensor
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x22) TEMPERATURE**			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	temperature			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	temperature			

temperature: Temperature value in two's complement representation in units of 1 Kelvin: 0x00 corresponds to 23 degree Celsius.

5.2.25 Register (0x24) FIFO_LENGTH_0

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

Name	Register (0x24) FIFO_LENGTH_0			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	fifo_byte_counter_7_0			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	fifo_byte_counter_7_0			

fifo_byte_counter_7_0: Current fill level of FIFO buffer.

5.2.26 Register (0x25) FIFO_LENGTH_1

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

Name	Register (0x25) FIFO_LENGTH_1			
Bit	7	6	5	4
Read/Write	n/a	n/a	R	R
Reset Value	0	0	0	0
Content	reserved		fifo_byte_counter_13_8	
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	fifo_byte_counter_13_8			

fifo_byte_counter_13_8: FIFO byte counter bits 13..8

5.2.27 Register (0x26) FIFO_DATA

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

Name	Register (0x26) FIFO_DATA			
Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	fifo_data			
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	fifo_data			

fifo_data: FIFO read data.

5.2.28 Register (0x2A) INTERNAL_STATUS

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

Name	**Register (0x2A) INTERNAL_STATUS**			
Bit	7	6	5	4
Read/Write	n/a	R	R	R
Reset Value	0	0	0	0
Content	reserved	odr_50Hz_error	axes_remap_error	message
Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	0	0
Content	message			

message: Internal status message

message		
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

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.
odr_50Hz_error: The minimum bandwidth conditions are not respected for the features which require 50 Hz data

5.2.29 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):

Name	**Register (0x40) ACC_CONF**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	1	0	1	0
Content	acc_perf_mode	acc_bwp		
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	1	0	0	0
Content	acc_odr			

acc_odr: ODR in Hz. The output data rate is independent of the power mode setting for the sensor, but not all settings are supported in all power modes.

acc_odr		
0x00	reserved	Reserved
0x01	odr_0p78	25/32
0x02	odr_1p5	25/16
0x03	odr_3p1	25/8
0x04	odr_6p25	25/4
0x05	odr_12p5	25/2
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	Reserved
0x0e	odr_6k4	Reserved
0x0f	odr_12k8	Reserved

acc_bwp: Bandwidth parameter, determines filter configuration (acc_perf_mode=1) and averaging for undersampling mode (acc_perf_mode=0)

acc_bwp		
0x00	osr4_avg1	acc_perf_mode = 1 -> OSR4 mode; acc_perf_mode = 0 -> no averaging
0x01	osr2_avg2	acc_perf_mode = 1 -> OSR2 mode; acc_perf_mode = 0 -> average 2 samples
0x02	norm_avg4	acc_perf_mode = 1 -> normal mode; acc_perf_mode = 0 -> average 4 samples
0x03	cic_avg8	acc_perf_mode = 1 -> Reserved; acc_perf_mode = 0 -> average 8 samples
0x04	res_avg16	acc_perf_mode = 1 -> Reserved; acc_perf_mode = 0 -> average 16 samples
0x05	res_avg32	acc_perf_mode = 1 -> Reserved; acc_perf_mode = 0 -> average 32 samples
0x06	res_avg64	acc_perf_mode = 1 -> Reserved; acc_perf_mode = 0 -> average 64 samples
0x07	res_avg128	acc_perf_mode = 1 -> Reserved; acc_perf_mode = 0 -> average 128 samples

acc_perf_mode: Select accelerometer filter performance mode:

acc_perf_mode		
0x00	cic_avg	averaging mode.
0x01	cont	continuous filter function.

5.2.30 Register (0x41) ACC_RANGE

DESCRIPTION: Selection of the Accelerometer g-range
RESET: 0x01
DEFINITION (Go to register map):

Name	**Register (0x41) ACC_RANGE**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	RW	RW
Reset Value	0	0	0	1
Content	reserved		acc_range	

acc_range: Accelerometer g-range

acc_range		
0x00	range_2g	+/-2g
0x01	range_4g	+/-4g
0x02	range_8g	+/-8g
0x03	range_16g	+/-16g

5.2.31 Register (0x44) AUX_CONF

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

Name	**Register (0x44) AUX_CONF**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	1	0	0
Content	aux_offset			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	1	1	0
Content	aux_odr			

aux_odr: Select the poll rate for the sensor attached to the Auxiliary interface.

aux_odr		
0x00	reserved	Reserved
0x01	odr_0p78	25/32
0x02	odr_1p5	25/16
0x03	odr_3p1	25/8
0x04	odr_6p25	25/4
0x05	odr_12p5	25/2
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	Reserved
0x0d	odr_3k2	Reserved
0x0e	odr_6k4	Reserved
0x0f	odr_12k8	Reserved

aux_offset: 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.

5.2.32 Register (0x45) FIFO_DOWNS

DESCRIPTION: Configure Accelerometer downsampling rates for FIFO
RESET: 0x80
DEFINITION (Go to register map):

Name	**Register (0x45) FIFO_DOWNS**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	1	0	0	0
Content	acc_fifo_filt_data	acc_fifo_downs		
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			

acc_fifo_downs: Downsampling for accelerometer data (2**acc_fifo_downs)
acc_fifo_filt_data: selects filtered or unfiltered Accelerometer data for fifo

acc_fifo_filt_data		
0x00	unfiltered	Unfiltered data
0x01	filtered	Filtered data

5.2.33 Register (0x46) FIFO_WTM_0

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

Name	**Register (0x46) FIFO_WTM_0**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	fifo_water_mark_7_0			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	fifo_water_mark_7_0			

5.2.34 Register (0x47) FIFO_WTM_1

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

Name	**Register (0x47) FIFO_WTM_1**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			fifo_water_mark_12_8
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	1	0
Content	fifo_water_mark_12_8			

5.2.35 Register (0x48) FIFO_CONFIG_0

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

Name	**Register (0x48) FIFO_CONFIG_0**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	RW	RW
Reset Value	0	0	1	0
Content	reserved		fifo_time_en	fifo_stop_on_full

fifo_stop_on_full: Stop writing samples into FIFO when FIFO is full.

fifo_stop_on_full		
0x00	disable	do not stop writing to FIFO when full
0x01	enable	Stop writing into FIFO when full.

fifo_time_en: Return sensortime frame after the last valid data frame.

fifo_time_en		
0x00	disable	do not return sensortime frame
0x01	enable	return sensortime frame

5.2.36 Register (0x49) FIFO_CONFIG_1

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

Name	**Register (0x49) FIFO_CONFIG_1**			
Bit	7	6	5	4
Read/Write	n/a	RW	RW	RW
Reset Value	0	0	0	1
Content	reserved	fifo_acc_en	fifo_aux_en	fifo_header_en
Bit	3	2	1	0
Read/Write	RW	RW	n/a	n/a
Reset Value	0	0	0	0
Content	fifo_tag_int1_en	fifo_tag_int2_en	reserved	

fifo_tag_int2_en: FIFO interrupt 2 tag enable

fifo_tag_int2_en		
0x00	disable	disable tag
0x01	enable	enable tag

fifo_tag_int1_en: FIFO interrupt 1 tag enable

fifo_tag_int1_en		
0x00	disable	disable tag
0x01	enable	enable tag

fifo_header_en: FIFO frame header enable

fifo_header_en		
0x00	disable	no header is stored (output data rate of all enabled sensors need to be identical)
0x01	enable	header is stored

fifo_aux_en: Store Auxiliary data in FIFO (all 3 axes)

fifo_aux_en		
0x00	disable	no Auxiliary data is stored
0x01	enable	Auxiliary data is stored

fifo_acc_en: Store Accelerometer data in FIFO (all 3 axes)

fifo_acc_en		
0x00	disable	no Accelerometer data is stored
0x01	enable	Accelerometer data is stored

5.2.37 Register (0x4B) AUX_DEV_ID

DESCRIPTION: Auxiliary interface slave device id
RESET: 0x20
DEFINITION (Go to register map):

Name	**Register (0x4B) AUX_DEV_ID**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	1	0
Content	i2c_device_addr			
Bit	3	2	1	0
Read/Write	RW	RW	RW	n/a
Reset Value	0	0	0	0
Content	i2c_device_addr			reserved

i2c_device_addr: I2C device address of Auxiliary slave

5.2.38 Register (0x4C) AUX_IF_CONF

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

Name	**Register (0x4C) AUX_IF_CONF**			
Bit	7	6	5	4
Read/Write	RW	n/a	n/a	n/a
Reset Value	1	0	0	0
Content	aux_manual_en	reserved		
Bit	3	2	1	0
Read/Write	n/a	n/a	RW	RW
Reset Value	0	0	1	1
Content	reserved		aux_rd_burst	

aux_rd_burst: Burst data length (1,2,6,8 byte)

aux_rd_burst		
0x00	BL1	Burst length 1
0x01	BL2	Burst length 2
0x02	BL6	Burst length 6
0x03	BL8	Burst length 8

aux_manual_en: Enable auxiliary interface manual mode.

aux_manual_en		
0x00	disable	Data mode
0x01	enable	Setup mode

5.2.39 Register (0x4D) AUX_RD_ADDR

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

Name	**Register (0x4D) AUX_RD_ADDR**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	1	0	0
Content	read_addr			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	1	0
Content	read_addr			

read_addr: Address to read

5.2.40 Register (0x4E) AUX_WR_ADDR

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

Name	**Register (0x4E) AUX_WR_ADDR**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	1	0	0
Content	write_addr			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	1	1	0	0
Content	write_addr			

write_addr: Address to write

5.2.41 Register (0x4F) AUX_WR_DATA

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

Name	**Register (0x4F) AUX_WR_DATA**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	write_data			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	1	0
Content	write_data			

write_data: Data to write

5.2.42 Register (0x53) INT1_IO_CTRL

DESCRIPTION: Configure the electrical behaviour of the interrupt pins
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x53) INT1_IO_CTRL**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			input_en
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	output_en	od	lvl	edge_ctrl

edge_ctrl: Configure trigger condition of INT1 pin (input)

edge_ctrl		
0x00	level_tr	Level
0x01	edge_tr	Edge

lvl: Configure level of INT1 pin

lvl		
0x00	active_low	active low
0x01	active_high	active high

od: Configure behaviour of INT1 pin to open drain.

od		
0x00	push_pull	push-pull
0x01	open_drain	open drain

output_en: Output enable for INT1 pin

output_en		
0x00	off	Output disabled
0x01	on	Output enabled

input_en: Input enable for INT1 pin

input_en		
0x00	off	Input disabled
0x01	on	Input enabled

5.2.43 Register (0x54) INT2_IO_CTRL

DESCRIPTION: Configure the electrical behaviour of the interrupt pins
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x54) INT2_IO_CTRL**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			input_en
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	output_en	od	lvl	edge_ctrl

edge_ctrl: Configure trigger condition of INT2 pin (input)

edge_ctrl		
0x00	level_tr	Level
0x01	edge_tr	Edge

lvl: Configure level of INT2 pin

lvl		
0x00	active_low	active low
0x01	active_high	active high

od: Configure behaviour of INT2 pin to open drain.

od		
0x00	push_pull	push-pull
0x01	open_drain	open drain

output_en: Output enable for INT2 pin

output_en		
0x00	off	Output disabled
0x01	on	Output enabled

input_en: Input enable for INT2 pin

input_en		
0x00	off	Input disabled
0x01	on	Input enabled

5.2.44 Register (0x55) INT_LATCH

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

Name	**Register (0x55) INT_LATCH**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			int_latch

int_latch: Latched/non-latched/temporary interrupt modes

int_latch		
0x00	none	non latched
0x01	permanent	latched

5.2.45 Register (0x56) INT1_MAP

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

Name	**Register (0x56) INT1_MAP**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	n/a
Reset Value	0	0	0	0
Content	error_int_out	no_motion_out	any_motion_out	reserved
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			

any_motion_out:	Any-motion detection output
no_motion_out:	No-motion detection output
error_int_out:	Error interrupt output

5.2.46 Register (0x57) INT2_MAP

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

Name	**Register (0x57) INT2_MAP**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	n/a
Reset Value	0	0	0	0
Content	error_int_out	no_motion_out	any_motion_out	reserved
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			

any_motion_out:	Any-motion detection output
no_motion_out:	No-motion detection output
error_int_out:	Error interrupt output

5.2.47 Register (0x58) INT_MAP_DATA

DESCRIPTION: Interrupt mapping hardware interrupts
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x58) INT_MAP_DATA**			
Bit	7	6	5	4
Read/Write	n/a	RW	RW	RW
Reset Value	0	0	0	0
Content	reserved	int2_drdy	int2_fwm	int2_ffull
Bit	3	2	1	0
Read/Write	n/a	RW	RW	RW
Reset Value	0	0	0	0
Content	reserved	int1_drdy	int1_fwm	int1_ffull

int1_ffull:	FIFO Full interrupt mapped to INT1
int1_fwm:	FIFO Watermark interrupt mapped to INT1
int1_drdy:	Data Ready interrupt mapped to INT1
int2_ffull:	FIFO Full interrupt mapped to INT2
int2_fwm:	FIFO Watermark interrupt mapped to INT2
int2_drdy:	Data Ready interrupt mapped to INT2

5.2.48 Register (0x59) INIT_CTRL

DESCRIPTION: Start initialization
RESET: 0x90
DEFINITION (Go to register map):

Name	Register (0x59) INIT_CTRL			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	1	0	0	1
Content	init_ctrl			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	init_ctrl			

init_ctrl: Start initialization

5.2.49 Register (0x5E) FEATURES_IN

DESCRIPTION: Feature configuration read/write port
RESET: 0x00
DEFINITION (Go to register map):

Name	Register (0x5E) FEATURES_IN			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	features_in			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	features_in			

features_in: Feature configuration read/write data

Address	Bit	Name	Description	Reset	Access
any_motion					
0x5E: 0x00		settings_1	Any-motion detection general configuration flags - part 1	0x00AA	
	10...0	threshold	Slope threshold value for any-motion detection. Range is 0 to 1g. Default value is 0xAA = 83mg.	0xAA	RW
0x5E: 0x02		settings_2	Any-motion detection general configuration flags - part 2	0x0005	
	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	x_en	Enables the feature on a per-axis basis	0x0	RW

	14	y_en	Enables the feature on a per-axis basis	0x0	RW
	15	z_en	Enables the feature on a per-axis basis	0x0	RW
no_motion					
0x5E: 0x04		settings_1	No-motion detection general configuration flags - part 1	0x00AA	
	10...0	threshold	Slope threshold value for no-motion detection. Range is 0 to 1g. Default value is 0xAA = 83mg.	0xAA	RW
0x5E: 0x06		settings_2	No-motion detection general configuration flags - part 2	0x0005	
	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	x_en	Enables the feature on a per-axis basis	0x0	RW
	14	y_en	Enables the feature on a per-axis basis	0x0	RW
	15	z_en	Enables the feature on a per-axis basis	0x0	RW
general_settings					
0x5E: 0x08		Reserved	Reserved	0x0000	
	15...0	Reserved	Reserved	0x0	
0x5E: 0x0A		axes_remapping	Describes axes remapping	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 0x03 reserved Map to x-axis	0x0	RW
	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 inverted 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 inverted 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 inverted Set this bit to invert the z axis	0x0	RW

5.2.50 Register (0x5F) INTERNAL_ERROR

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

Name	**Register (0x5F) INTERNAL_ERROR**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	R	R	n/a
Reset Value	0	0	0	0
Content	reserved	int_err_2	int_err_1	reserved

int_err_1: Internal error flag - long processing time, processing halted
int_err_2: Internal error flag - fatal error, processing halted

5.2.51 Register (0x6A) NVM_CONF

DESCRIPTION: NVM controller mode (Prog/Erase or Read only)
RESET: 0x00
DEFINITION (Go to register map):

Name	**Register (0x6A) NVM_CONF**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	RW	n/a
Reset Value	0	0	0	0
Content	reserved		nvm_prog_en	reserved

nvm_prog_en: Enable NVM programming

nvm_prog_en		
0x00	disable	disable
0x01	enable	enable

5.2.52 Register (0x6B) IF_CONF

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

Name	**Register (0x6B) IF_CONF**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			if_mode
Bit	3	2	1	0
Read/Write	n/a	n/a	n/a	RW
Reset Value	0	0	0	0
Content	reserved			spi3

spi3: Configure SPI Interface Mode for primary interface

spi3		
0x00	spi4	SPI 4-wire mode
0x01	spi3	SPI 3-wire mode

if_mode: Auxiliary interface configuration

if_mode		
0x00	p_auto_s_off	Auxiliary interface:off
0x01	p_auto_s_mag	Auxilary interface:Magnetometer

5.2.53 Register (0x6D) ACC_SELF_TEST

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

Name	**Register (0x6D) ACC_SELF_TEST**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	RW	RW	n/a	RW
Reset Value	0	0	0	0
Content	acc_self_test_amp	acc_self_test_sign	reserved	acc_self_test_en

acc_self_test_en: Enable accelerometer self-test

acc_self_test_en		
0x00	disabled	disabled
0x01	enabled	enabled

acc_self_test_sign: select sign of self-test excitation as

acc_self_test_sign		
0x00	negative	negative
0x01	positive	positive

acc_self_test_amp: select amplitude of the selftest deflection:

acc_self_test_amp		
0x00	low	low
0x01	high	high

5.2.54 Register (0x70) NV_CONF

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

Name	**Register (0x70) NV_CONF**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	acc_off_en	i2c_wdt_en	i2c_wdt_sel	spi_en

spi_en: disable the I2C and enable SPI for the primary interface, when it is in autoconfig mode

spi_en		
0x00	disabled	I2C enabled
0x01	enabled	I2C disabled

i2c_wdt_sel: Select timer period for I2C Watchdog

i2c_wdt_sel		
0x00	wdt_short	I2C watchdog timeout after 1.25 ms
0x01	wdt_long	I2C watchdog timeout after 40 ms

i2c_wdt_en: I2C Watchdog at the SDI pin in I2C interface mode

i2c_wdt_en		
0x00	Disable	Disable I2C watchdog
0x01	Enable	Enable I2C watchdog

acc_off_en: Add the offset defined in the off_acc_[xyz] OFFSET register to filtered and unfiltered Accelerometer data

acc_off_en		
0x00	disabled	Disabled
0x01	enabled	Enabled

5.2.55 Register (0x71) OFFSET_0

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

Name	**Register (0x71) OFFSET_0**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_x			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_x			

off_acc_x: Accelerometer offset compensation (X-axis).

5.2.56 Register (0x72) OFFSET_1

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

Name	**Register (0x72) OFFSET_1**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_y			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_y			

off_acc_y: Accelerometer offset compensation (Y-axis).

5.2.57 Register (0x73) OFFSET_2

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

Name	**Register (0x73) OFFSET_2**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_z			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	off_acc_z			

off_acc_z: Accelerometer offset compensation (Z-axis).

5.2.58 Register (0x7C) PWR_CONF

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

Name	**Register (0x7C) PWR_CONF**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	n/a	RW	RW
Reset Value	0	0	1	1
Content	reserved		fifo_self_wakeup	adv_power_save

adv_power_save		
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.)

fifo_self_wakeup		
0x00	fsw_off	FIFO read disabled in advanced power saving mode.
0x01	fsw_on	FIFO read enabled after interrupt in advanced power saving mode.

5.2.59 Register (0x7D) PWR_CTRL

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

Name	**Register (0x7D) PWR_CTRL**			
Bit	7	6	5	4
Read/Write	n/a	n/a	n/a	n/a
Reset Value	0	0	0	0
Content	reserved			
Bit	3	2	1	0
Read/Write	n/a	RW	n/a	RW
Reset Value	0	0	0	0
Content	reserved	acc_en	reserved	aux_en

aux_en		
0x00	mag_off	Disables the auxiliary sensor.
0x01	mag_on	Enables the auxiliary sensor.

acc_en		
0x00	acc_off	Disables the Accelerometer.
0x01	acc_on	Enables the Accelerometer.

5.2.60 Register (0x7E) CMD

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

Name	**Register (0x7E) CMD**			
Bit	7	6	5	4
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	cmd			
Bit	3	2	1	0
Read/Write	RW	RW	RW	RW
Reset Value	0	0	0	0
Content	cmd			

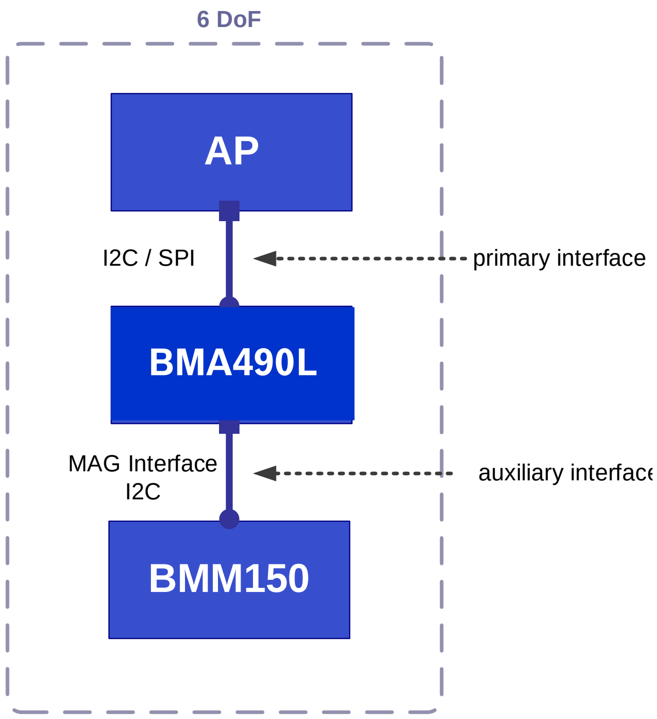
cmd: Available commands (Note: Register will always read as 0x00):

cmd		
0xa0	nvm_prog	Writes the NVM backed registers into NVM
0xb0	fifo_flush	Clears all data in the FIFO, does not change FIFO_CONFIG and FIFO_DOWNS registers
0xb6	softreset	Triggers a reset, all user configuration settings are overwritten with their default state

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, BMA490L supports an auxiliary interface. See picture below.
If the auxiliary interface is enabled, the BMA490L can be connected to an external sensor (e.g. a magnetometer) in order to build a 6-DoF solution. Then the BMA490L 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.



6.2 Primary Interface

By default, the BMA490L operates in I2C mode. The BMA490L 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 3 possible digital interfaces share partly the same pins. The mapping for the primary interface of the BMA490L is given in the following table:

Pin#	Name	I/O Type	Description	Connect to (Primary IF)		
				in SPI4W	in SPI3W	in I2C
1	SDO	Digital I/O	Serial data output in SPI Address select in I²C mode see chapter 7.2	SDO	DNC (float)	GND for default I2C addr.
2	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
5	INT1	Digital I/O	Interrupt output 1 (default) (Input for external FIFO sync) *	INT1 (FIFO sync)	INT1 (FIFO sync)	INT1 (FIFO sync)
6	INT2	Digital I/O	Interrupt output 2 (default) (Input for external FIFO sync) *	INT2 (FIFO sync)	INT2 (FIFO sync)	INT2 (FIFO sync)
10	CSB	Digital in	Chip select for SPI mode	CSB	CSB	V_{DDIO} -
12	SCX	Digital in	SCK for SPI serial clock SCL for I²C serial clock	SCK	SCK	SCL

* INT1 and/or INT2 can also be configured as an input in case the external data synchronization in FIFO is used. See chapter4.6. 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:

Parameter	Symbol	Condition	Min	Typ	Max	Units
Pull-up Resistance, CSB pin	R_{up}	Internal Pull-up Resistance to VDDIO	75	100	125	kΩ
Input Capacitance	C_{in}			5		pF
I²C Bus Load Capacitance (max. drive capability) C_{I2C_Load}		V_{DDIO}>=1.62V			400	pF
		V_{DDIO}<1.62V			120	pF

6.3 Primary Interface I2C/SPI Protocol Selection

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

At reset / power-up, BMA490L is in I2C mode. If CSB is connected to VDDIO during power-up and not changed, the sensor 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 BMA490L 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 in 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.

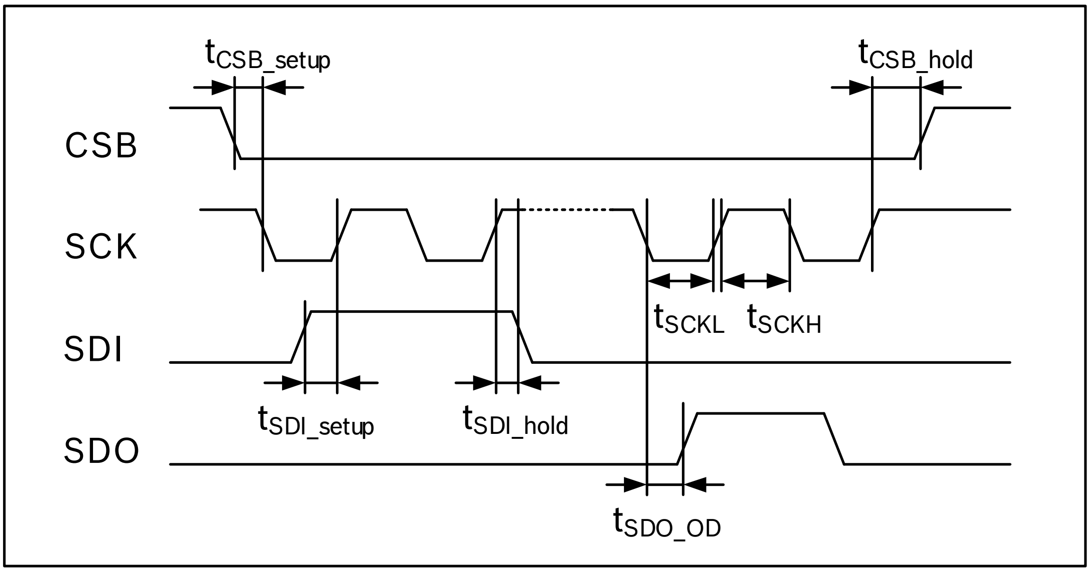
6.4 SPI interface and protocol

The timing specification for SPI of the BMA490L is given in the following table:

SPI timing, valid at $V_{DDIO} \geq 1.71V$

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 between write accesses, suspend mode, low-power mode 1	$t_{IDLE\_wacc\_sum}$		450		µs
Idle time after write and read access, active state	$t_{IDLE\_wr\_act}$		2		µs

The following figure shows the definition of the SPI timings:



SPI timing diagram

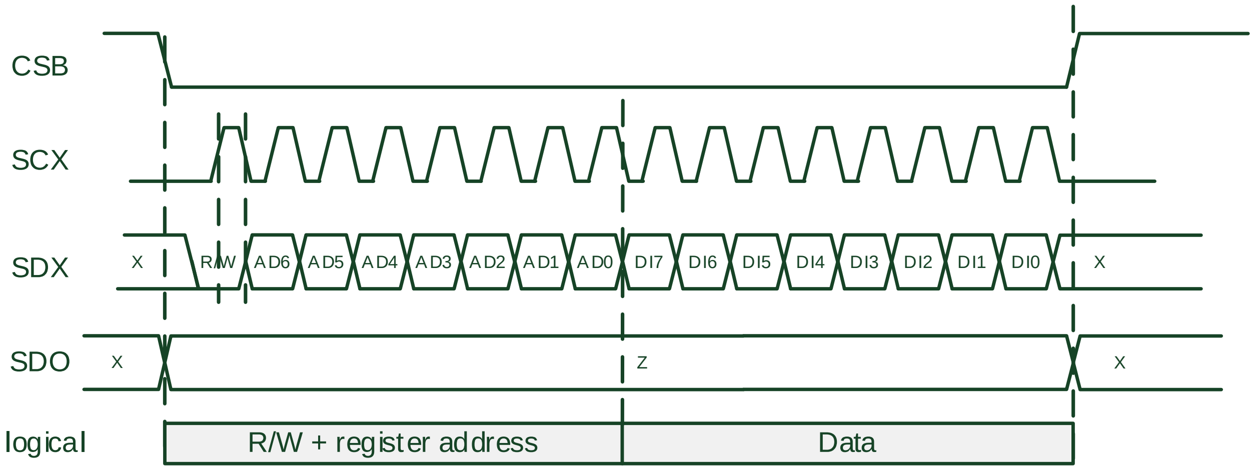
The SPI interface of the BMA490L is compatible with two modes, ´00´ [CPOL = ´0´ and CPHA = ´0´] and ´11´ [CPOL = ´1´ and CPHA = ´1´]. The automatic selection between ´00´ and ´11´ is controlled based on the value of SCK after a falling edge of CSB.

Two configurations of the SPI interface are supported by the BMA490L: 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 SDI 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. The BMA490L also supports multiple-byte read and write operations.

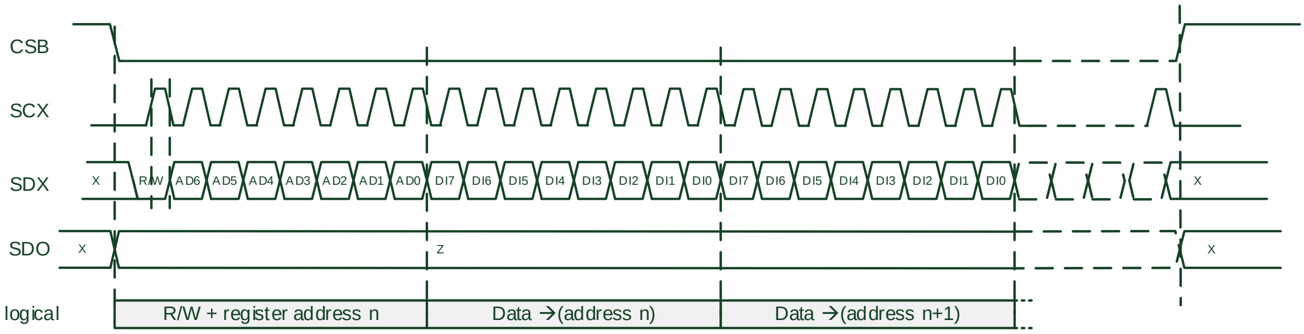
In SPI 4-wire configuration CSB (chip select low active), SCK (serial clock), SDI (serial data input), and SDO (serial data output) pins are used. The communication starts when the CSB is pulled low by the SPI master and stops when CSB is pulled high. SCK is also controlled by SPI master. SDI and SDO 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.



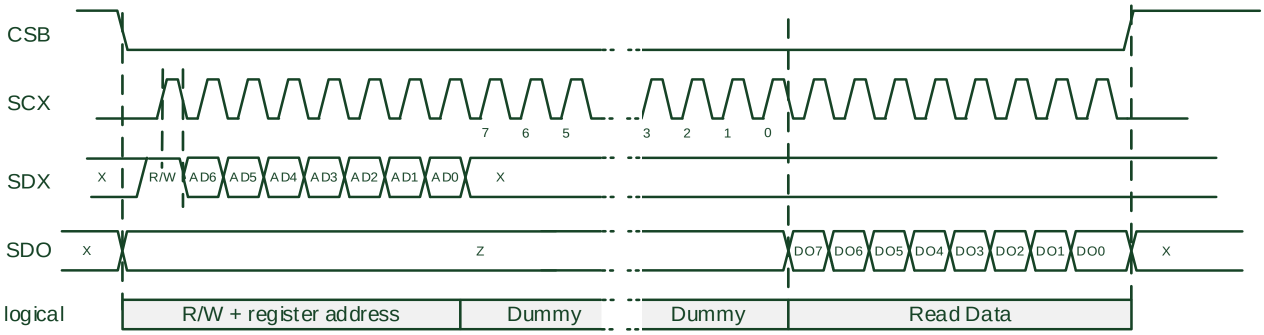
4-wire basic SPI write sequence (mode ´00´)

Multiple write operations are possible by keeping CSB low and continuing the data transfer. Only the first register address has to be written. 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 BMA490L 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.

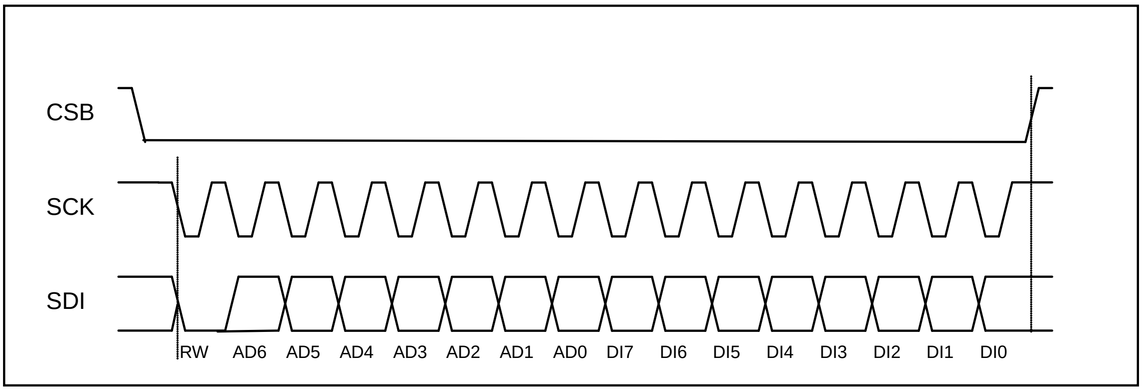
AD6-AD0: Address

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 BMA490L via the SDO line correspond to a dummy byte and the 2nd byte correspond 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), SCK (serial clock), and SDI (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 also 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 wave-form (read or write access) for 3-wire configuration is depicted in the figure below:



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

6.5 Primary I2C Interface

The I²C bus uses SCL (= SCx pin, serial clock) and SDA (= SDx pin, serial data input and output) signal lines. Both lines should connected to V_{DDIO} externally via pull-up resistors so that they are pulled high when the bus is free.

The default I²C address of the device is 0b0011000 (0x18). It is used if the SDO pin is pulled to ´GND´. The alternative address 0b0011001 (0x19) is selected by pulling the SDO pin to ´VDDIO´.

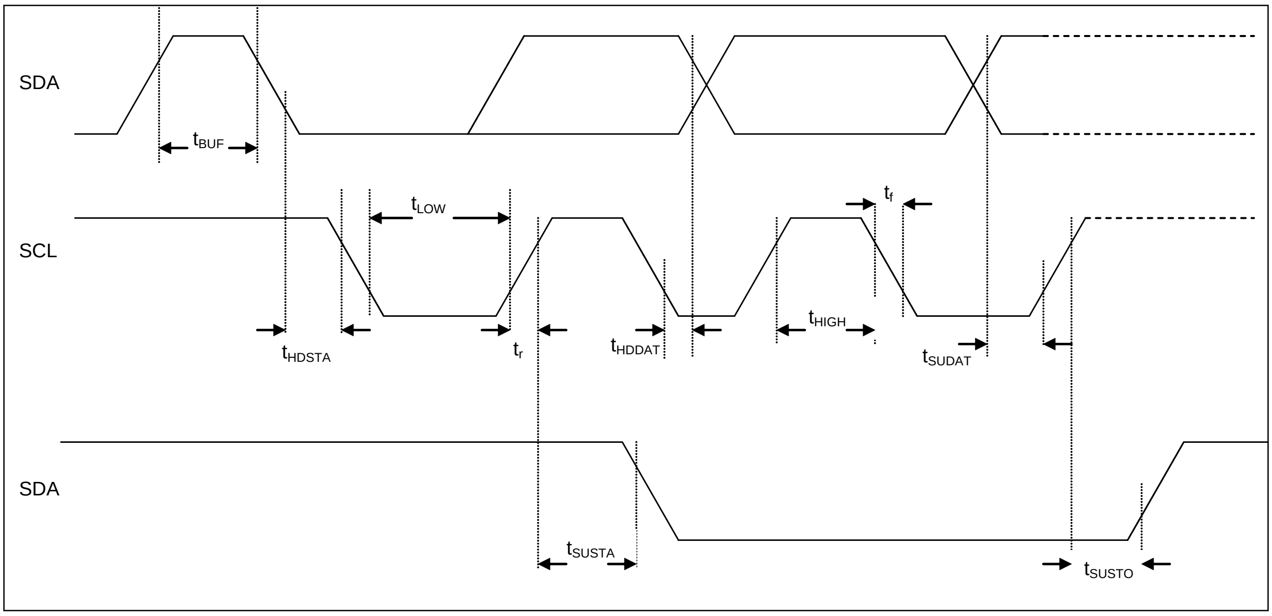
The I²C interface of the BMA490L is compatible with the I²C Specification UM10204 Rev. 03 (19 June 2007), available at http://www.nxp.com. The BMA490L supports **I²C standard mode and fast mode**, only 7-bit address mode is supported. For V_{DDIO} = 1.2V to 1.62 V the guaranteed voltage output levels are slightly relaxed as described in Table 1 of the electrical specification section.

BMA490L also supports an **extended I²C 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 I²C of the BMA490L is given in the following table:

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		
		performance mode	2		
Idle time between write accesses, performance mode, low-power mode	$t_{IDLE\_wacc\_nm}$	low power mode	450		
		performance mode	2		
Idle time between write accesses, suspend mode, low-power mode	$t_{IDLE\_wacc\_sum}$		450		

The figure below shows the definition of the I²C timings given in Table 28:



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	0	0	1	1	0	0	0	0	0	0	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, on special registers with address trap.

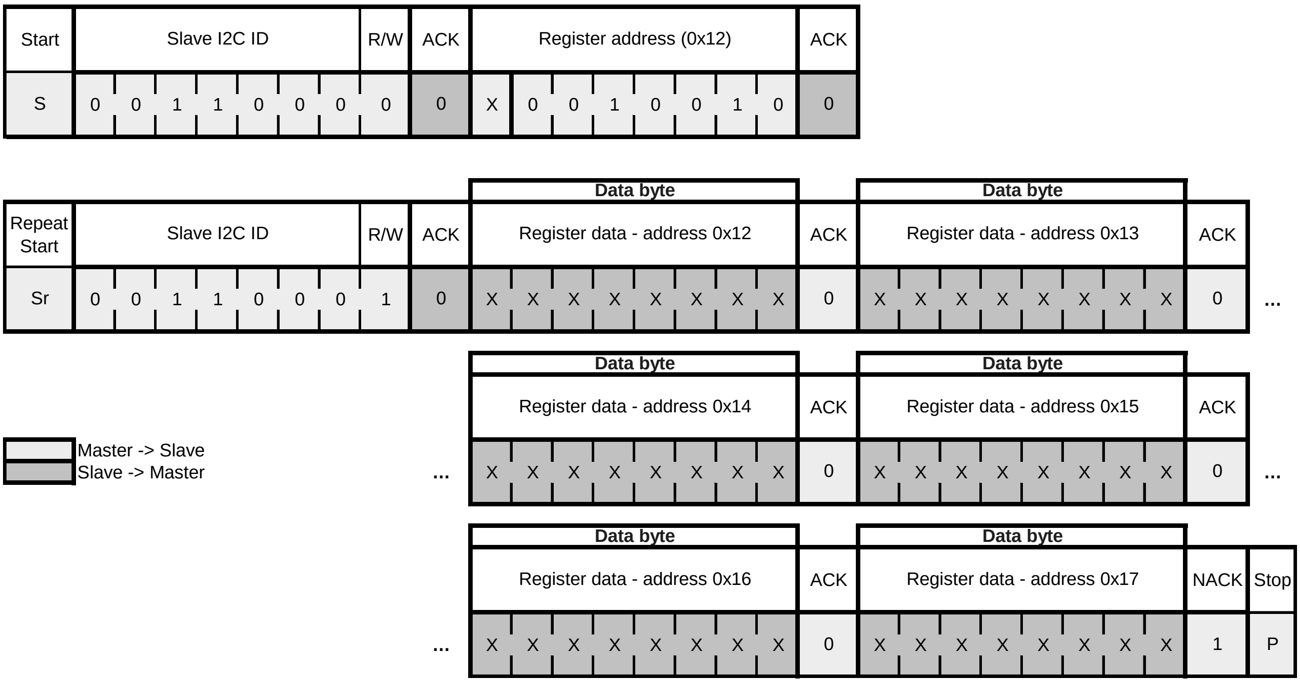
Start	Slave Adress							R/W	ACK	Register address (0x45)								ACK	Register data byte 0 (0x80)								ACK
S	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Register data byte 1 (0x64)								ACK	… …								ACK	Register data byte n (0xXX)								ACK	Stop
0	1	1	0	0	1	0	0	0	.	.	.	.	.	.	.	.	0	x	x	x	x	x	x	x	x	0	P

□ Master -> Slave
▨ Slave -> Master

I²C read access:
I²C read access also 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 (ACKS = 0) to enable further data transfer. A NACKM (after ACKS = 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	0 0 1 1 0 0 0	0	0	X 0 0 1 0 0 1 0	0

				Data byte		Data byte		
Repeat Start	Slave I2C ID	R/W	ACK	Register data - address 0x12	ACK	Register data - address 0x13	ACK	
Sr	0 0 1 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		Data byte		
	Register data - address 0x14	ACK	Register data - address 0x15	ACK	
...	X X X X X X X X	0	X X X X X X X X	0	...

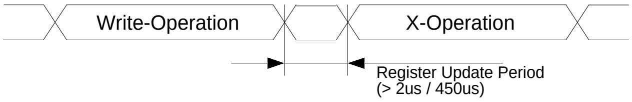
	Data byte		Data byte		
	Register data - address 0x16	ACK	Register data - address 0x17	NACK	Stop
...	X X X X X X X X	0	X X X X X X X X	1	P

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 BMA490L. 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.

6.6 SPI and I²C Access Restrictions

In order to allow for the correct internal synchronization of data written to the BMA490L, 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 performance mode or other modes.

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



Post-Write Access Timing Constraints

6.7 Auxiliary Interface

The BMA490L allows attaching an external sensor (MAG-sensor) to a BMA490L via the auxiliary interface. The connection diagrams for the auxiliary interface are depicted in the chapter 7.3. The timings of the secondary I2C interface are the same as for the primary I2C interface, see chapter 6.5.

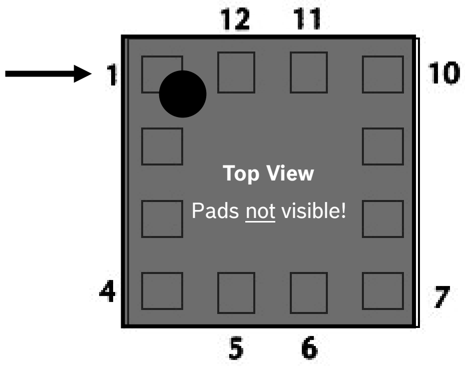
BMA490L acts as a master of the secondary interface, controls the data acquisition of the MAG-sensor (slave of the secondary interface) and presents the data to the application processor (AP) in the user registers of the BMA490L through the primary interface. The internal pull-up resistors of ASCL and ASDA are by default disabled, so it is recommended to added pull-up resistors externally onto the secondary interface for proper I2C communication. Please contact your regional sales representative in case the internal pull-up resistors are necessary to be enabled. No additional I2C master or slave devices must be attached to the magnetometer interfaces.

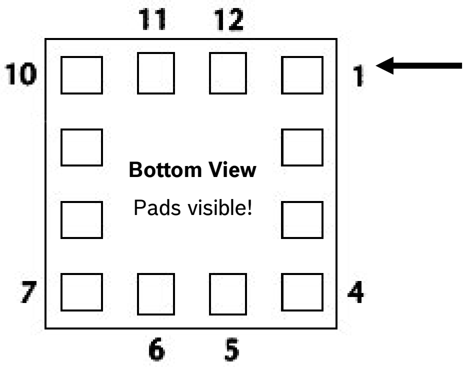
The BMA490L autonomously reads out the sensor data from BMM150 without intervention of the AP and stores the data in its data registers (per default) and FIFO (see Register FIFO_CONFIG_1.fifo_aux_en). The initial setup of the BMM150 after power-on is done through indirect addressing in the BMA490L. From a system perspective the initialization for BMM150 when attached to BMA490L should be possible within 100ms.

More information about the usage of Auxiliary Interface can be found in chapter 4.9.

7. Pin-out and Connection Diagrams

7.1 Pin-out





Pin description

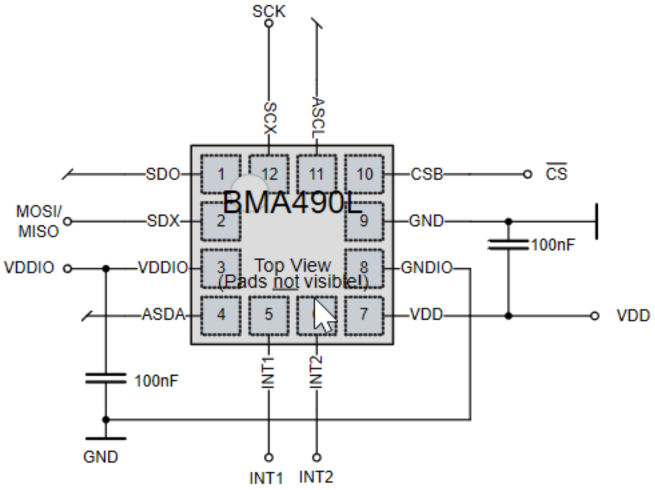
Pin#	Name	I/O Type	Description	Connect to		
				in SPI 4W	**In SPI 3W**	**in I²C**
1	SDO	Digital I/O	Serial data output in SPI Address select in I²C mode see chapter 6.5	SDO	DNC (float)	GND for default I2C addr.
2	SDX	Digital I/O	SDA serial data I/O in I²C SDI serial data input in SPI 4W SDA serial data I/O in SPI 3W	SDI	SDA	SDA
3	VDDIO	Supply	Digital I/O supply voltage (1.2V ... 3.6V)	V_{DDIO}	V_{DDIO}	V_{DDIO}
4	ASDA	Digital I/O	Serial data I/O – Secondary Interface (I²C Master for Magnetometer)	VDDIO/ GNDIO/NC or (ASDA - Secondary interface)	VDDIO/ GNDIO/NC or (ASDA - Secondary interface)	VDDIO/ GNDIO/NC or (ASDA - Secondary interface)
5	INT1	Digital I/O	Interrupt output 1 (default) (Input for external FIFO sync) *	INT1 (FIFO sync)	INT1 (FIFO sync)	INT1 (FIFO sync)
6	INT2	Digital I/O	Interrupt output 2 (default) (Input for external FIFO sync) *	INT2 (FIFO sync)	INT2 (FIFO sync)	INT2 (FIFO sync)
7	VDD	Supply	Power supply for analog & digital domain (1.62V ... 3.6V)	V_{DD}	V_{DD}	V_{DD}
8	GNDIO	Ground	Ground for I/O	GND	GND	GND
9	GND	Ground	Ground for digital & analog	GND	GND	GND
10	CSB	Digital in	Chip select for SPI mode	CSB	CSB	V_{DDIO}
11	ASCL	Digital out	Digital clock (out) – Secondary Interface (I²C Master for Magnetometer)	VDDIO/ GNDIO/NC or (ASCL - Secondary interface)	VDDIO/ GNDIO/ NC or (ASCL - Secondary interface)	VDDIO/ GNDIO/ NC or (ASCL - Secondary interface)
12	SCX	Digital in	SCK for SPI serial clock SCL for I²C serial clock	SCK	SCK	SCL

* INT1 and/or INT2 can also be configured as an input in case the external data synchronization in FIFO is used. See chapter 4.6. If INT1 and/or INT2 are not used, please do not connect them (DNC).

7.2 Connection Diagrams without Auxiliary Interface

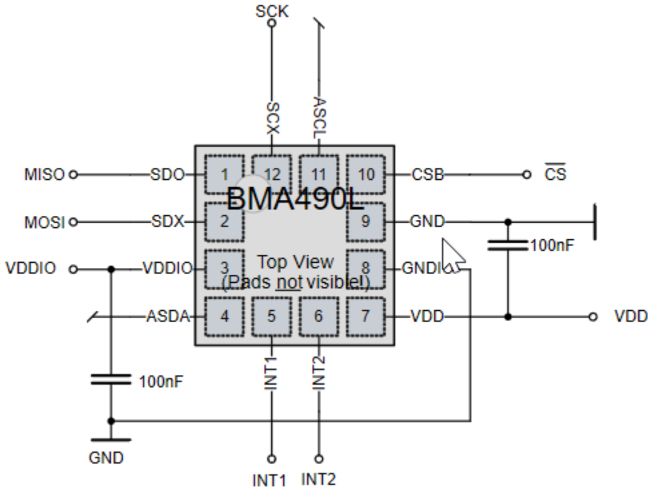
SPI

3-wire



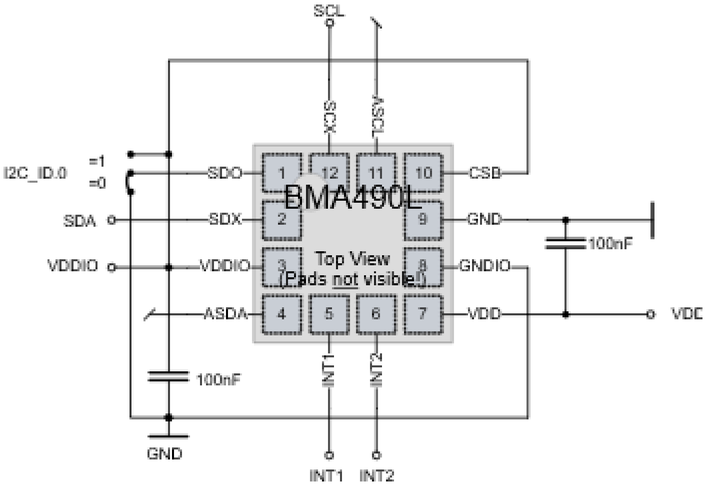
It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD).

4-wire



It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD).

I2C



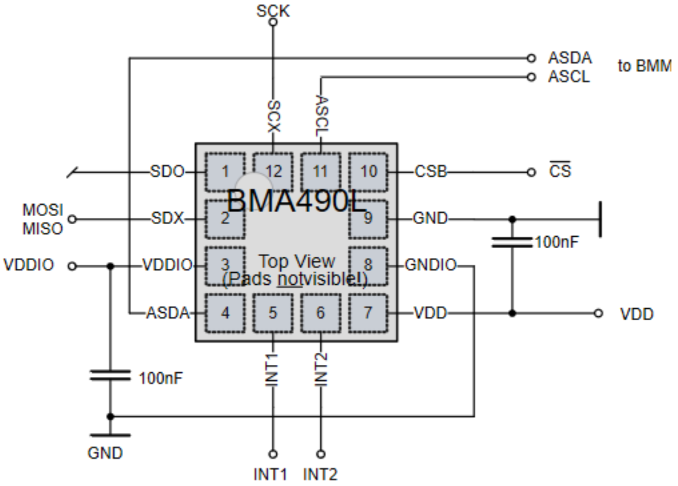
It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD). SDA and SCL should be connected to V_{DDIO} externally via pull-up resistors so that they are pulled high when the bus is free.

7.3 Connection Diagrams with Auxiliary Interface

The internal pull-up resistors of ASCL and ASDA are by default disabled, so it is recommended to added pull-up resistors externally onto the secondary interface for proper I2C communication.

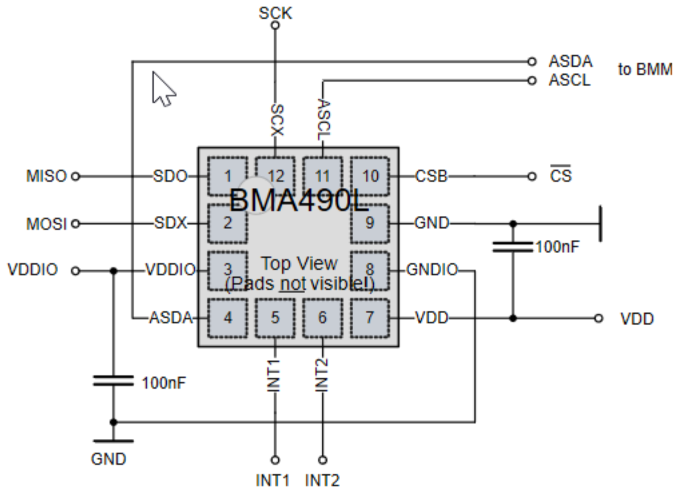
SPI

3-wire



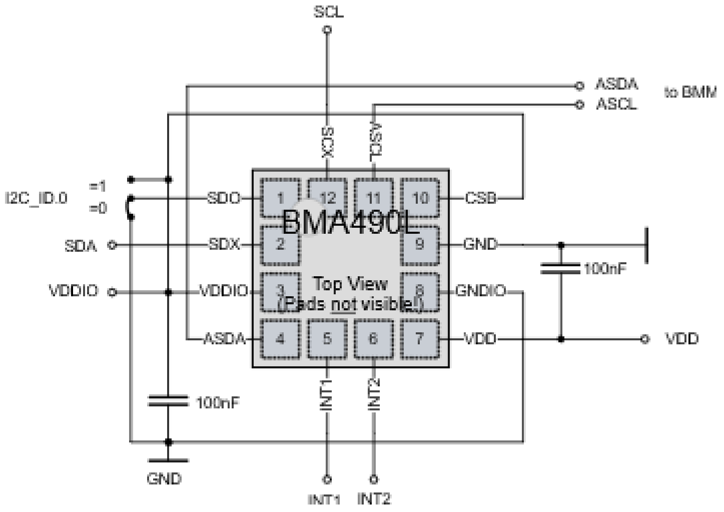
It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD).

4-wire



It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD).

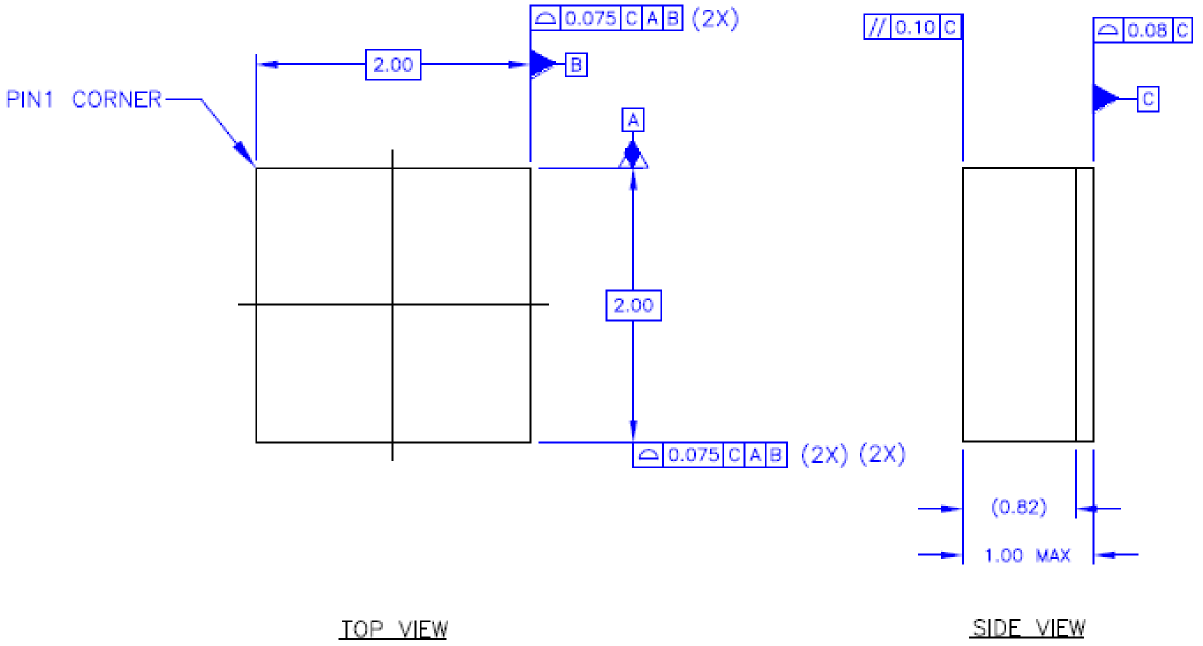
I2C

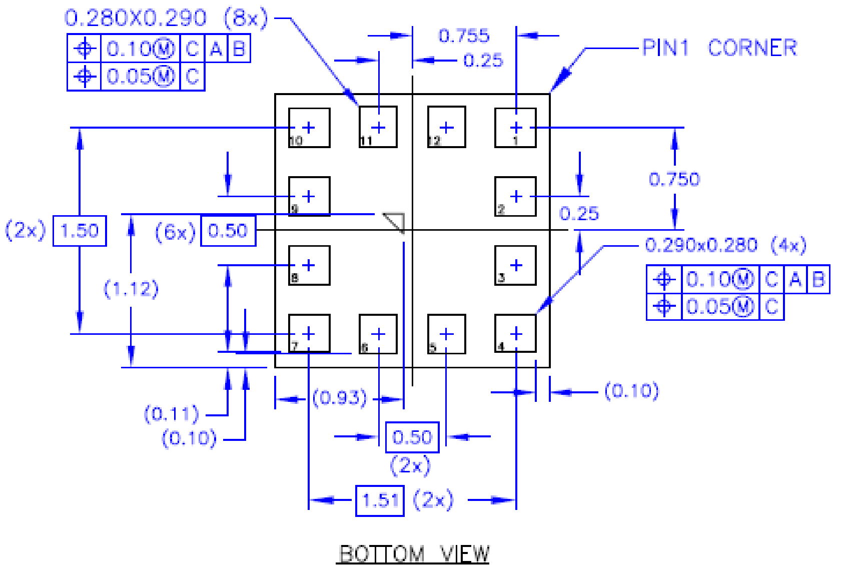


It is recommended to use 100nF decoupling capacitors at pin 3 (VDDIO) and pin 7 (VDD). SDA and SCL should be connected to V_{DDIO} externally via pull-up resistors so that they are pulled high when the bus is free.

8. Package

8.1 Package outline dimensions





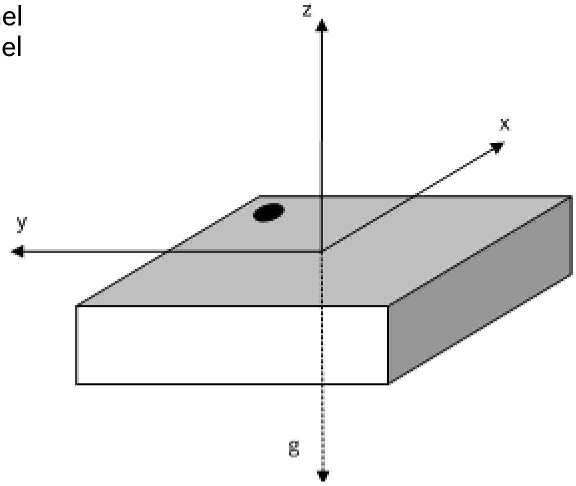
Note that the pin 5, 6, 11, 12 are in same direction (0.280*0.290, 4x), while pin 1, 2, 3, 4, 7, 8, 9, 10 are in same direction (0.290*0.280, 8x).

8.2 Sensing axis orientation

If the sensor is accelerated in the indicated directions, the corresponding channel will deliver a positive acceleration signal (dynamic acceleration). If the sensor is at rest and the force of gravity is acting along the indicated directions, the output of the corresponding channel will be negative (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:

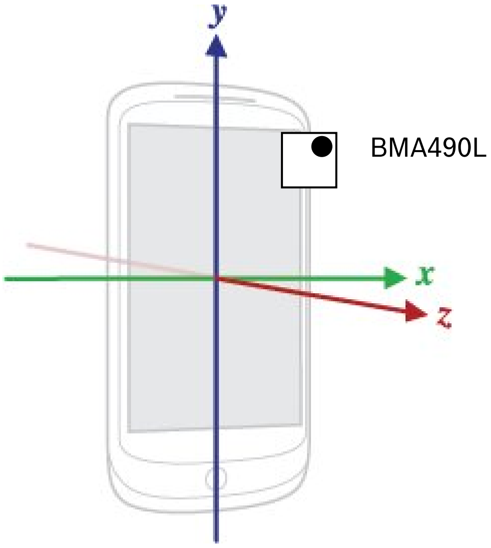
- ± 0g for the X channel
- ± 0g for the Y channel
- + 1g for the Z channel



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.

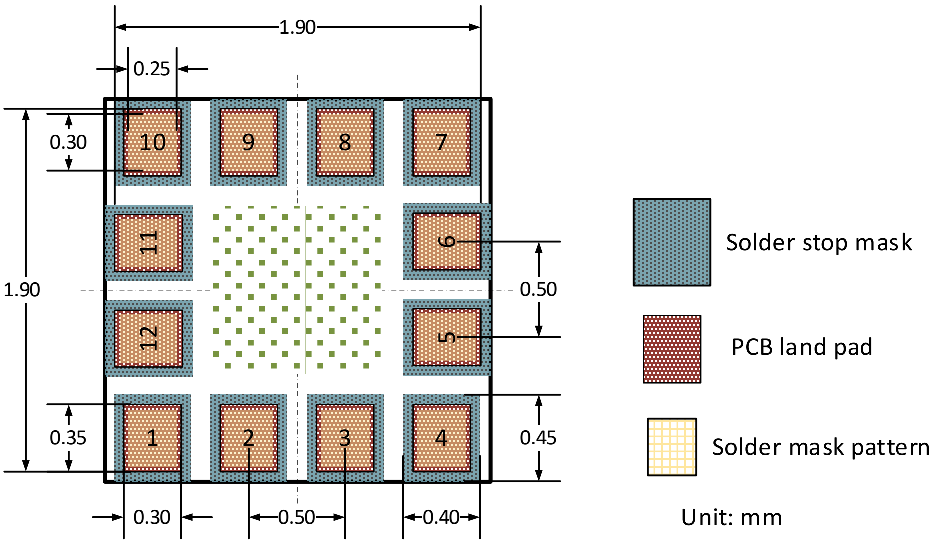
Sensor Orientation (gravity vector

For reference the figure below shows the typical device orientation with an integrated BMA490L.



8.3 Landing pattern recommendation

The recommended landing pattern for the BMA490L on customer's PCB is given in the following figure. It is recommended to avoid any wiring underneath the BMA490L (shaded area).



8.4 Marking

Mass production

Labeling	Name	Symbol	Remark
● ZZ CCC	Internal Code	ZZ	internal
	Counter ID	CCC	3 alphanumeric digits, variable to generate trace-code.
	Pin 1 identifier top side	●	--

Engineering samples

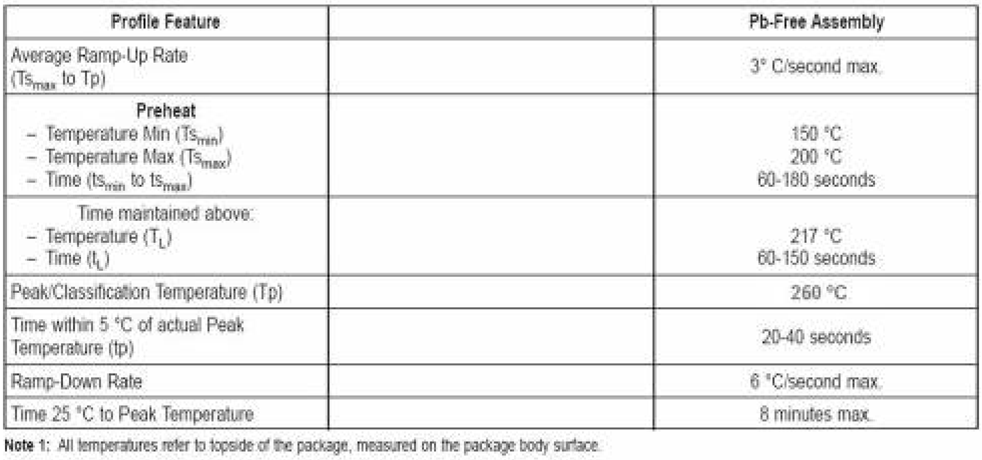
Labeling	Name	Symbol	Remark
● XE NCC	Internal Code	X	internal
	Eng. sample ID	E, N	2 alphanumeric digits, fixed to identify engineering sample, N = “C”
	Sample ID	CC	2 alphanumeric digits, variable to generate trace-code.
	Pin 1 identifier top side	●	--

8.5 Soldering guidelines

The moisture sensitivity level of the BMA490L sensors corresponds to JEDEC Level 1, see also

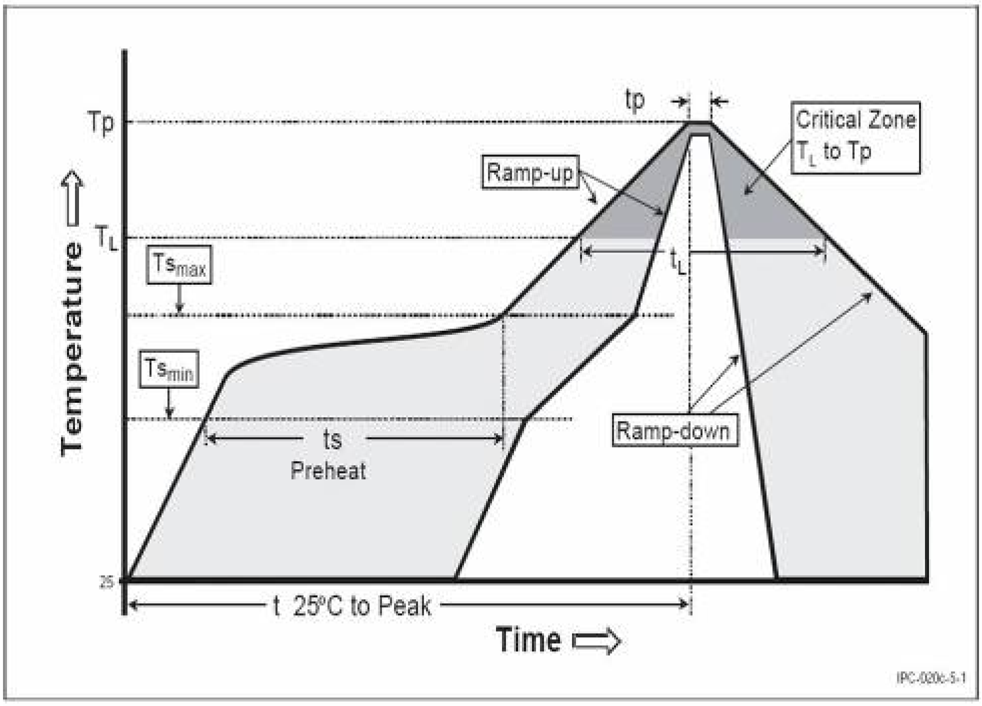
- IPC/JEDEC J-STD-020C "Joint Industry Standard: Moisture/Reflow Sensitivity Classification for non-hermetic Solid State Surface Mount Devices"
- IPC/JEDEC J-STD-033A "Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices"

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



Profile Feature		Pb-Free Assembly
Average Ramp-Up Rate (Ts_{max} to Tp)		3° C/second max.
Preheat – Temperature Min (Ts_{min}) – Temperature Max (Ts_{max}) – Time (ts_{min} to ts_{max})		150 °C 200 °C 60-180 seconds
Time maintained above: – Temperature (T_L) – Time (t_L)		217 °C 60-150 seconds
Peak/Classification Temperature (Tp)		260 °C
Time within 5 °C of actual Peak Temperature (tp)		20-40 seconds
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.



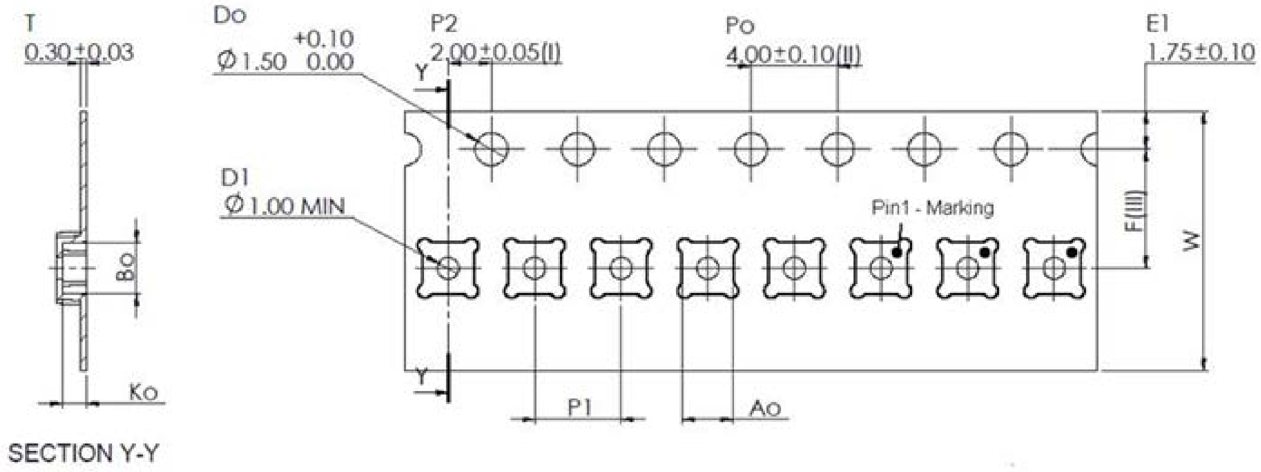
8.6 Handling instructions

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 Tape and Reel specification



Ao	2.35 +/- 0.05
Bo	2.30 +/- 0.05
Ko	1.10 +/- 0.05
F	5.50 +/- 0.05
P1	4.00 +/- 0.10
W	12.00 +0.30 / -0.10

8.8 Environmental safety

The BMA490L sensor meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also:

> *Directive 2015/863 (amending Annex II to Directive 2011/65/EU) of the European Parliament and of the Council on the restriction of the use of certain hazardous substances in electrical and electronic equipment.*

Halogen content

The BMA490L 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 BMA490L.

While Bosch Sensortec took care that all of the technical packages parameters are described above are 100% identical for all sources, there can be differences in the chemical content and the internal structural 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 BMA490L product.

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 may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or safety-critical systems. Safety-critical systems are those for which a malfunction is expected to lead to bodily harm, death or severe property damage. In addition, they shall not be used directly or indirectly for military purposes (including but not limited to nuclear, chemical or biological proliferation of weapons or development of missile technology), nuclear power, deep sea or space applications (including but not limited to satellite technology).

The resale and/or use of Bosch Sensortec products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and to inform Bosch Sensortec without delay of all safety-critical incidents.

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
1.0	All	Public release	06.2020

***Longevity Disclaimer**

Bosch Sensortec strives to maintain the supply of longevity product variants for a period of 10 years (from SOD/product introduction date), including the notification period. During such period, in case of significant volume decrease or manufacturing changes Bosch Sensortec may decide to

(i) replace the product by another (comparable) product
and/or
(ii) change the technology, manufacturing facilities and/or process

Any change will be notified to customers using the standard Bosch Sensortec product/process change policy (PCN)

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

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

Modifications reserved
Preliminary - specifications subject to change without notice
Document number: BST-BMA490L-DS000-01
Revision_1.0_062020