AN4825
Application note

Ultra-compact high-performance eCompass module based on the LSM303AGR

Introduction

This document is intended to provide usage information and application hints related to ST's eCompass module.

The LSM303AGR is a 3D digital magnetometer and 3D digital accelerometer system-in-package with a digital I^2C and 3-wire SPI interface standard output, performing at 250 µA in combo high-resolution mode and no more than 60 µA in combo low-power mode. Thanks to the ultra-low noise performance of the magnetometer and to the ultra-low power of the accelerometer, the device combines always-on low-power features with superior sensing precision for an optimal motion experience for the consumer. The device features ultra low-power operational modes that allow advanced power saving and smart sleep-to-wake and return-to-sleep functions.

The device has a magnetic field dynamic range of ±50 gauss and a user-selectable full-scale acceleration range of ±2g/±4g/±8g/±16g.

The LSM303AGR can be configured to generate an interrupt signal for magnetic field detection and to automatically compensate for hard-iron offsets provided from the higher application layer. It can be configured to generate interrupt signals by detecting an independent inertial wakeup/free-fall event as well as by the position of the device itself. Thresholds and timing of the interrupt generator are programmable by the end user on the fly. Automatic programmable sleep-to-wake-up and return-to-sleep functions are also available for enhanced power saving.

The LSM303AGR has an integrated 32-level first in first out (FIFO) buffer allowing the user to store accelerometer data in order to limit intervention by the host processor.

The LSM303AGR is available in a small thin plastic land grid array package (LGA) and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

The ultra-small size and weight of the SMD package make it an ideal choice for handheld portable applications such as smartphones, IoT connected devices, and wearables or any other application where reduced package size and weight are required.

Contents

List of tables

List of figures

1 Pin description

Figure 1. Pin connections

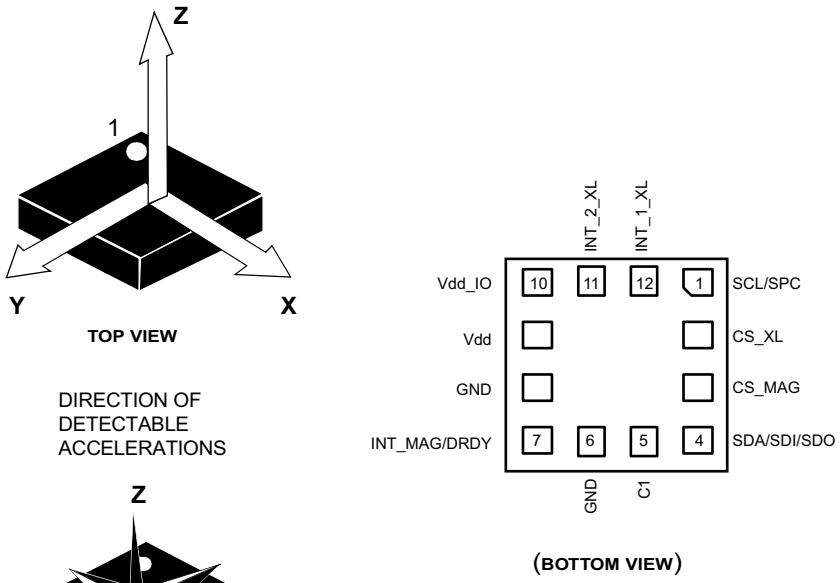


Table 1. Pin description

Pin#	Name	Function	Pin status
1	SCL SPC	I^2C serial clock (SCL) SPI serial port clock (SPC)	Default: input without pull-up
2	CS_XL	Accelerometer: SPI enable I^2C/SPI mode selection 1: SPI idle mode / I^2C communication enabled; 0: SPI communication mode / I^2C disabled	Default: input without pull-up
3	CS_MAG	Magnetometer: SPI enable I^2C/SPI mode selection 1: SPI idle mode / I^2C communication enabled; 0: SPI communication mode / I^2C disabled	Default: input without pull-up
4	SDA SDI SDO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	Default: input without pull-up
5	C1	Capacitor connection (C1 = 220 nF)	
6	GND	0 V supply	
7	INT_MAG/DRDY	Magnetometer interrupt/data-ready signal	High impedance
8	GND	0 V supply	
9	Vdd	Power supply	
10	Vdd_IO	Power supply for I/O pins	
11	INT_2_XL	Accelerometer interrupt 2	Output forced to ground
12	INT_1_XL	Accelerometer interrupt 1	Output forced to ground

2 Registers

Table 2. Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS_REG_AUX_A	07h	-	TOR	-	-	-	TDA	-	-
RESERVED	08h-0Bh								
OUT_TEMP_L_A	0Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUT_TEMP_H_A	0Dh	D15	D14	D13	D12	D11			
INT_COUNTER_REG_A	0Eh	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
WHO_AM_I_A	0Fh	0	0	1	1	0	0	1	1
TEMP_CFG_REG_A	1Fh	TEMP_EN1	TEMP_EN0	0	0	0	0	0	0
CTRL_REG1_A	20h	ODR3	ODR2	ODR1	ODR0	LPen	Zen	Yen	Xen
CTRL_REG2_A	21h	HPM1	HPM0	HPCF2	HPCF1	FDS	HPCLICK	HPIS2	HPIS1
CTRL_REG3_A	22h	I1_CLICK	I1_AOI1	I1_AOI2	I1_DRDY1	I1_DRDY2	I1_WTM	I1_OVERRUN	-
CTRL_REG4_A	23h	BDU	BLE	FS1	FS0	HR	ST1	ST0	SPI_ENABLE
CTRL_REG5_A	24h	BOOT	FIFO_EN	-	-	LIR_INT1	D4D_INT1	LIR_INT2	D4D_INT2
CTRL_REG6_A	25h	I2_CLICKen	I2_INT1	I2_INT2	BOOT_I2	P2_ACT	-	H_LACTIVE	-
REFERENCE /DATACAPTURE_A	26h	Ref7	Ref6	Ref5	Ref4	Ref3	Ref2	Ref1	Ref0
STATUS_REG_A	27h	ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA
OUT_X_L_A	28h	XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0
OUT_X_H_A	29h	XD15	XD14	XD13	XD12	XD11	XD10	XD9	XD8
OUT_Y_L_A	2Ah	YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0
OUT_Y_H_A	2Bh	YD15	YD14	YD13	YD12	YD11	YD10	YD9	YD8
OUT_Z_L_A	2Ch	ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0
OUT_Z_H_A	2Dh	ZD15	ZD14	ZD13	ZD12	ZD11	ZD10	ZD9	ZD8
FIFO_CTRL_REG_A	2E	FM1	FM0	TR	FTH4	FTH3	FTH2	FTH1	FTH0

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_SRC_REG_A	2F	WTM	OVRN_FIFO	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0
INT1_CFG_A	30h	AOI	6D	ZHIE/ ZUPE	ZLIE/ ZDOWNE	YHIE/ YUPE	YLIE/ YDOWNE	XHIE/ XUPE	XLIE/ XDOWNE
INT1_SRC_A	31h	0	IA	ZH	ZL	YH	YL	XH	XL
INT1_THS_A	32h	0	THS6	THS5	THS4	THS3	THS2	THS1	THS0
INT1_DURATION_A	33h	0	D6	D5	D4	D3	D2	D1	D0
INT2_CFG_A	34h	AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
INT2_SRC_A	35h	0	IA	ZH	ZL	YH	YL	XH	XL
INT2_THS_A	36h	0	THS6	THS5	THS4	THS3	THS2	THS1	THS0
INT2_DURATION_A	37h	0	D6	D5	D4	D3	D2	D1	D0
CLICK_CFG_A	38h	-	-	ZD	ZS	YD	YS	XD	XS
CLICK_SRC_A	39h	-	IA	Dclick	Sclick	Sign	Z	Y	X
CLICK_THS_A	3Ah	-	Ths6	Ths5	Ths4	Ths3	Ths2	Ths1	Ths0
TIME_LIMIT_A	3Bh	-	TLI6	TLI5	TLI4	TLI3	TLI2	TLI1	TLI0
TIME_LATENCY_A	3Ch	TLA7	TLA6	TLA5	TLA4	TLA3	TLA2	TLA1	TLA0
TIME_WINDOW_A	3Dh	TW7	TW6	TW5	TW4	TW3	TW2	TW1	TW0
Act_THS_A	3Eh	-	Acth6	Acth5	Acth4	Acth3	Acth2	Acth1	Acth0
Act_DUR_A	3Fh	ActD7	ActD6	ActD5	ActD4	ActD3	ActD2	ActD1	ActD0
RESERVED	40h-44h								
OFFSET_X_REG_L_M	45h	Offset_X_7	Offset_X_6	Offset_X_5	Offset_X_4	Offset_X_3	Offset_X_2	Offset_X_1	Offset_X_0
OFFSET_X_REG_H_M	46h	Offset_X_15	Offset_X_14	Offset_X_13	Offset_X_12	Offset_X_11	Offset_X_10	Offset_X_9	Offset_X_8
OFFSET_Y_REG_L_M	47h	Offset_Y_7	Offset_Y_6	Offset_Y_5	Offset_Y_4	Offset_Y_3	Offset_Y_2	Offset_Y_1	Offset_Y_0
OFFSET_Y_REG_H_M	48h	Offset_Y_15	Offset_Y_14	Offset_Y_13	Offset_Y_12	Offset_Y_11	Offset_Y_10	Offset_Y_9	Offset_Y_8
OFFSET_Z_REG_L_M	49h	Offset_Z_7	Offset_Z_6	Offset_Z_5	Offset_Z_4	Offset_Z_3	Offset_Z_2	Offset_Z_1	Offset_Z_0
OFFSET_Z_REG_H_M	4Ah	Offset_Z_15	Offset_Z_14	Offset_Z_13	Offset_Z_12	Offset_Z_11	Offset_Z_10	Offset_Z_9	Offset_Z_8

Table 2. Registers (continued)

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WHO_AM_I_M	4Fh	0	1	0	0	0	0	0	0
CFG_REG_A_M	60h	COMP_ TEMP_EN	REBOOT	SOFT_RST	LP	ODR1	ODR0	MD1	MD0
CFG_REG_B_M	61h	0	0	0	0	INT_on_ DataOFF	Set_FREQ	OFF_CANC	LPF
CFG_REG_C_M	62h	0	INT_MAG_ PIN	I2C_DIS	BDU	BLE	0	Self_test	INT_MAG
INT_CTRL_REG_M	63h	XIEN	YIEN	ZIEN	0	0	IEA	IEL	IEN
INT_SOURCE_REG_M	64h	P_TH_S_X	P_TH_S_Y	P_TH_S_Z	N_TH_S_X	N_TH_S_Y	N_TH_S_Z	MROI	INT
INT_THS_L_REG_M	65h	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0
INT_THS_H_REG_M	66h	TH15	TH14	TH13	TH12	TH11	TH10	TH9	TH8
STATUS_REG_M	67h	Zyxor	zor	yor	xor	Zyxda	zda	yda	xda
OUTX_L_REG_M	68h	D7	D6	D5	D4	D3	D2	D1	D0
OUTX_H_REG_M	69h	D15	D14	D13	D12	D11	D10	D9	D8
OUTY_L_REG_M	6Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUTY_H_REG_M	6Bh	D15	D14	D13	D12	D11	D10	D9	D8
OUTZ_L_REG_M	6Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUTZ_H_REG_M	6Dh	D15	D14	D13	D12	D11	D10	D9	D8

3 Magnetometer

3.1 Operating modes

The magnetic module provides two power modes: high-resolution (HR) mode and low-power (LP) mode.

After the power supply is applied, the magnetometer is automatically configured in idle mode.

The LP and MD bits of CFG_REG_A_M are used to select the power and operating modes of the device.

The table below summarizes the current consumption of the two power modes with offset cancellation disabled/enabled.

Table 3. Current consumption of operating modes

ODR (Hz)	Current consumption (µA) (CFG_REG_A_M [LP] = 0) high-resolution CFG_REG_B_M [OFF_CANC] = 0	Current consumption (µA) (CFG_REG_A_M [LP] = 1) low-power CFG_REG_B_M [OFF_CANC] = 0	Current consumption (µA) (CFG_REG_A_M [LP] = 0) high-resolution CFG_REG_B_M [OFF_CANC] = 1	Current consumption (µA) (CFG_REG_A_M [LP] = 1) low-power CFG_REG_B_M [OFF_CANC] = 1
10	100	25	120	50
20	200	50	235	100
50	475	125	575	235
100	950	250	1130	460

The operative modes of the device can be set using the MD bits (refer to the table below). In continuous mode the device continuously performs measurements and places the result in the data register. When single measurement mode is selected, the device performs a single measurement, sets DRDY high and returns to idle mode.

Table 4. Operative modes

MD1	MD0	Mode
0	0	Continuous mode
0	1	Single mode
1	0	Idle mode
1	1	Idle mode

3.1.1 Idle mode

When the magnetometer is in idle mode, almost all internal blocks of the device are switched off to minimize power consumption. Digital interfaces (I^2C and SPI) are still active to allow communication with the device. The content of the configuration registers is preserved and the output data registers are not updated, keeping the last data sampled in memory before going into idle mode.

3.1.2 High-resolution mode

In HR mode, the magnetometer circuitry is periodically turned on/off with a duty cycle that is a function of the selected ODR. Data interrupt generation is active.

3.1.3 Low-power mode

In low-power mode, as for high-resolution mode, the magnetometer circuitry is periodically turned on/off with a duty cycle that is a function of the selected ODR. Data interrupt generation is active.

The difference is in the number of samples used to generate each output sample, which is four times less than the number used in high-resolution mode, thus ensuring a lower power consumption.

3.1.4 Single measurement mode

The LSM303AGR offers single measurement mode in both high-resolution and low-power modes.

Single measurement mode is enabled by writing bits MD[1:0] to '01' in CFG_REG_A_M (60h).

In single measurement mode, once the measurement has been performed, the DRDY pin is set to high, data is available in the output register and the LSM303AGR is automatically configured in idle mode by setting the MD[1] bit to '1'.

Single measurement is independent of the programmed ODR but depends on the frequency at which the MD[1:0] bits are written by the microcontroller/application processor. Maximum ODR frequency achievable in single mode measurement is given in the following table.

Table 5. Maximum ODR in single measurement mode (HR and LP modes)

Maximum ODR	Power mode (CFG_REG_A_M[LP])
100 Hz	High resolution (LP = ‘0’)
150 Hz	Low power (LP = ‘1’)

In single measurement mode, for ODR < 10 Hz, current consumption can be calculated with the following formula:

(Current_consumption_10Hz - Current_consumption_in_power_down) / (10 Hz / ODR) + Current_consumption_in_power_down

3.2 Magnetometer low-pass filter

The use of a digital low-pass filter reduces noise. The filter can be enabled by setting the LPF bit in the CFG_REG_B_M register. The table below shows the bandwidth and the reduced noise using the filter.

Table 6. LPF, relative bandwidth and noise

LPF	Bandwidth [Hz]	LP	Noise RMS [mG]
0	ODR / 2	0	4.5
1	ODR / 4	0	3
0	ODR / 2	1	9
1	ODR / 4	1	6

3.3 Reading output data

3.3.1 Startup sequence

Upon power-up of the device, the magnetometer is configured by default in power-down mode.

To turn on the magnetometer and gather magnetic data, it is necessary to select one of the operating modes through the CFG_REG_A_M register.

The following general-purpose sequence can be used to configure the magnetometer:

1. Write CFG_REG_A_M = 00h // Mag = 10 Hz (HR and continuous mode)
2. Write CFG_REG_C_M = 01h // Mag data-ready interrupt enable

Writing 01h in CFG_REG_A_M instead of 00h will set the device to operate in single read mode instead of continuous mode.

3.3.2 Using the status register

The device is provided with a STATUS_REG_M register which should be polled to check when a new set of data is available. The Zyxda bit is set to 1 when a new set of data is available from the magnetometer output.

The reads should be performed as follows:

1. Read STATUS_REG_M
2. If Zyxda = 0, then go to 1
3. Read OUTX_L_REG_M
4. Read OUTX_H_REG_M
5. Read OUTY_L_REG_M
6. Read OUTY_H_REG_M
7. Read OUTZ_L_REG_M
8. Read OUTZ_H_REG_M
9. Data processing
10. Go to 1

If the device is configured in single measurement mode instead of continuous mode, the routine will be stuck at step 1 after one execution, since the device performs a single measurement, sets DRDY high and returns to idle mode. Please note that the MD bits return to idle mode bit values. It is possible to trigger another single read by setting the MD bits to 01h.

3.3.3 Using the data-ready signal

The device can be configured to have one HW signal to determine when a new set of measurement data is available for reading.

The data-ready signal is represented by the Zyxda bit of the STATUS_REG_M register. The signal can be driven to the INT_MAG/DRDY pin by setting to 1 the INT_MAG bit of the CFG_REG_C_M register.

The data-ready signal rises to 1 when a new set of data has been generated and it is available for reading. The signal gets reset when the higher part of one of the channels has been read (69h, 6Bh, 6Dh).

3.3.4 Using the block data update (BDU) feature

If reading the magnetometer data is particularly slow and cannot be synchronized (or it is not required) with either the Zyxda event bit in the STATUS_REG_M register or with the DRDY signal driven to the INT_MAG/DRDY pin, it is strongly recommended to set the BDU (block data update) bit to 1 in the CFG_REG_C_M register.

This feature avoids reading values (most significant and least significant parts of output data) related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent output data produced by the device, but, in case the read of a given pair (i.e. OUTX_H_REG_M and OUTX_L_REG_M, OUTY_H_REG_M and OUTY_L,_REG_M, OUTZ_H_REG_M and OUTZ_L_REG_M) is initiated, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

Note: BDU only guarantees that the LSB part and MSB part have been sampled at the same moment. For example, if the reading speed is too slow, X and Y can be read at T1 and Z sampled at T2.

3.3.5 Understanding output data

The measured magnetic data are sent to the OUTX_H_REG_M, OUTX_L_REG_M, OUTY_H_REG_M, OUTY_L_REG_M, OUTZ_H_REG_M, and OUTZ_L_REG_M registers. These registers contain, respectively, the most significant part and the least significant part of the magnetic signals acting on the X, Y, and Z axes.

The complete output data for the X, Y, Z channels is given by the concatenation OUTX_H_REG_M & OUTX_L_REG_M, OUTY_H_REG_M & OUTY_L_REG_M, OUTZ_H_REG_M & OUTZ_L_REG_M and it is expressed as a two's complement number.

Magnetic data is represented as 16-bit numbers, called LSB. It must be multiplied by the proper sensitivity parameter, M_So = 1.5, in order to obtain the corresponding value in mG.

Example of output data

Hereafter is a simple example of how to use the LSB data and transform it into mG.

Get raw data from the sensor:

```
OUTX_L_REG_M: 21h
OUTX_H_REG_M: 00h
OUTY_L_REG_M: 1Dh
OUTY_H_REG_M: FFh
OUTZ_L_REG_M: CBh
OUTZ_H_REG_M: FEh
```

Do registers concatenation:

```
OUTX_H_REG_M & OUTX_L_REG_M: 0021h
OUTY_H_REG_M & OUTY_L_REG_M: FF1Dh
OUTZ_H & OUTZ_L: FECBh OUTZ_H_REG_M & OUTZ_L_REG_M
```

Calculate signed decimal value (two's complement format):

```
X: +33
Y: -227
Z: -309
```

Apply sensitivity:

```
X: +33 * 1.5 = +49.5 mG
Y: -227 * 1.5 = -340.5 mG
Z: -309 * 1.5 = -463.5 mG
```

3.4 Magnetometer offset cancellation

Offset cancellation is the result of performing a set and reset in the magnetic sensor.

The offset cancellation technique is defined as follows:

$$H_{out} = \frac{H_n - H_{n-1}}{2}$$

where H_n and H_{n-1} are two consecutive magnetic field measurements, one after a set pulse, the other after a reset pulse.

Considering a magnetic offset (H_{off}), the two magnetic field measurements are:

- Set: $H_n = H + H_{off}$
- Reset: $H_{n-1} = -H + H_{off}$

The offset is cancelled according to the offset cancellation technique:

$$H_{out} = \frac{H_n - H_{n-1}}{2} = \frac{2H + H_{off} + (-H_{off})}{2} = H$$

In the device the offset cancellation is enabled by setting the OFF_CANC bit to '1' in CFG_REG_B_M. If the offset cancellation is disabled, a set of the magnetic sensor is performed anyway. The set pulse frequency can be configured by setting the Set_FREQ bit in CFG_REG_B_M. If Set_FREQ is set to '0', the set pulse is released every 63 ODR, otherwise if Set_FREQ is set to '1', the set pulse is released only at power-on after power-down.

If the user performs single reads, in order to enable the offset cancellation, OFF_CANC must be set to ‘1’ also in CFG_REG_B_M. Enabling this bit, the impulse polarity is inverted between a single read and the next one. If this feature is enabled, the user has to remove the offset manually using the formula below:

$$H_{out} = \frac{H_n + H_{n-1}}{2}$$

Offset cancellation using single reads is effective only if the reads are close in time, thus ensuring the offset does not drift between two consecutive reads.

3.5 Magnetometer hard-iron compensation

Hard-iron distortion occurs when a magnetic object is placed near the magnetometer and appears as a permanent bias in the sensor's outputs. The hard-iron correction consists of compensating magnetic data from hard-iron distortion.

The operation is defined as follows:

$$H_{out} = H_{read} - H_{HI}$$

where:

- H_{read} is the generic uncompensated magnetic field data, as read by the sensor;
- H_{HI} is the hard-iron distortion field;
- H_{out} is the compensated magnetic data.

The computation of the hard-iron distortion field should be performed by an external processor. After the computation of the hard iron-distortion field has been performed, the measured magnetic data can be compensated.

The device offers the possibility of storing hard-iron data inside six dedicated registers from 45h to 4Ah.

Each register contains eight bits so that the hard-iron data can be expressed as a 16-bit two's complement number. The OFFSET_axis_REG_H_M registers contain the MSBs of the hard-iron data, while the OFFSET_axis_REG_L_M registers contain the LSBs. Hard-iron data have the same format and weight of the magnetic output data. The hard-iron values stored in dedicated registers are automatically subtracted from the output data.

3.6 Interrupt generation

The device interrupt generation is based on the comparison between data and a programmable threshold. For interrupt-generation purposes, the magnetometer sensor has to be in active operating mode (not in power-down).

In order to enable the interrupt function, the IEN bit in the INT_CTRL_REG_M register must be set to '1'.

The user can select the axis/axes for which the interrupt function is to be enabled. In order to do this, the XIEN, YIEN and ZIEN bits in INT_CTRL_REG_M need to be set properly. The threshold value can be programmed by setting the INT_THS_L_REG_M and INT_THS_H_REG_M registers. The threshold is an absolute value as a 15-bit unsigned number. The threshold has the same sensitivity as magnetic data and is common to all the output values of the three axes: it is considered as an absolute value, but magnetic field measurements on either side of the threshold are calculated as positive or negative values.

The interrupt signals can be driven independently to the INT_MAG/DRDY pin or checked by reading the dedicated source register bits.

The IEA bit in the INT_CTRL_REG_M register must be used to select the polarity of the interrupt pin. If this bit is set to '0' (default value), the interrupt pin is active low and both the bit and pin change from high to low level when the related interrupt condition is verified. Otherwise, if the bit is set to '1' (active high), the interrupt pin is normally at low level and they change from low to high when the interrupt condition is reached.

The IEL bit of INT_SOURCE_REG_M allows applying the latched mode to the interrupt signals. When the IEL bit is set to '1' once the interrupt pin is asserted, it must be reset by reading INT_SOURCE_REG_M. If the IEL bit is set to '0', the interrupt signal is automatically reset when the interrupt condition is no longer verified.

3.6.1 Interrupt pin configuration

The device is provided with one pin that can be activated to generate either data-ready or interrupt signals. The functionality of the pin is selected through specified bits of the CFG_REG_C_M register.

Table 7. CFG_REG_C_M register

b7	b6	b5	b4	b3	b2	b1	b0
0	INT_MAG_PIN	I2C_DIS	BDU	BLE	0	Self_test	INT_MAG

- INT_MAG_PIN: if enabled, the interrupt signal (INT bit in INT_SOURCE_REG_M) is driven on the INT_MAG/DRDY pin;
- INT_MAG: if enabled, the magnetometer DRDY pin is configured as a digital output.

The interrupt control register is used to enable and to configure the interrupt recognition.

Table 8. INT_CTRL_REG_M register

b7	b6	b5	b4	b3	b2	b1	b0
XIEN	YIEN	ZIEN	0	0	IEA	IEL	IEN

- XIEN: enables the interrupt configuration for the X-axis;
- YIEN: enables the interrupt configuration for the Y-axis;
- ZIEN: enables the interrupt configuration for the Z-axis;
- IEA: controls the polarity of the INT bit;
- IEL: controls whether the INT bit is latched or pulsed;
- IEN: enables the interrupt generation

3.6.2 Event status

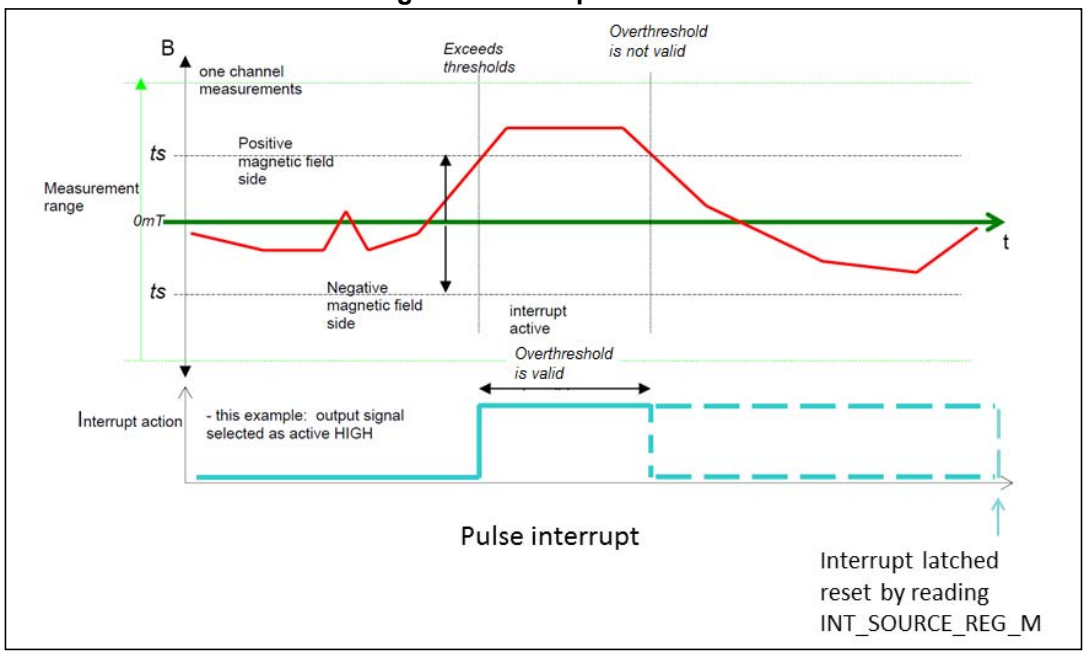
If multiple interrupt signals are routed on the INT_MAG_PIN, the logic level of this pin is the "OR" combination of the selected interrupt signals. In order to know which event has generated the interrupt condition, the application should read the status register INT_SOURCE_REG_M (64h) which also will clear the event.

3.6.3 Threshold interrupt

The magnetometer threshold interrupt can be enabled setting the IEN bit to '1' and enabling the recognition on the specified axes using the XIEN, YIEN and ZIEN bits. The interrupt threshold can be programmed using the INT_THS_L_REG_M and INT_THS_H_REG_M registers. When magnetic data exceeds the positive or the negative threshold, the interrupt signal is generated and the information about the type of interrupt is stored in the INT_SOURCE_REG_M register. In particular, when magnetic data exceeds the positive

threshold, the P_TH_S_axis bit is set to '1', while if data exceeds the negative threshold, the N_TH_S_axis bit is set to '1'. If magnetic data lay between the positive and the negative thresholds, no interrupt signal is released.

Figure 2. Interrupt function



The interrupt event signal can be routed to the INT_MAG/DRDY pin by setting to 1 the corresponding bit of the CFG_REG_C_M register; it can also be checked by reading the INT bit of the INT_SOURCE_REG_M register.

If latch mode is disabled (IEL bit of INT_CTRL_REG_M is set to '0'), the interrupt signal is automatically reset when the overthreshold condition is no longer verified. If latch mode is enabled and the interrupt signal is driven to the interrupt pin, once an overthreshold event has occurred and the interrupt pin is asserted, it must be reset by reading the INT_SOURCE_REG_M register.

Two different approaches for the interrupt function are available:

- Typical: comparison is between magnetic data read by the sensor and the programmable threshold;
- Advanced: comparison is made between magnetic data after hard-iron correction and the programmable threshold.

These approaches are configurable by setting the INT_on_DataOFF bit in CFG_REG_B_M. If INT_on_DataOFF is set to '0' the typical approach is selected, otherwise if it is set to '1', the advanced approach is selected.

A basic SW routine for threshold event recognition is given below.

1. Write 00h in CFG_REG_A_M // Turn on the magnetometer
// ODR = 10 Hz
2. Write 40h in CFG_REG_C_M // Interrupt driven on INT_MAG/DRDY pin
3. Write 80h in INT_THS_L_REG_M // Set a threshold equal to 128 (expressed in LSB)
4. Write E7h in INT_CTRL_REG_M // Enable a latched active-high interrupt on the three axes

The sample code exploits a threshold set to 192 mG (128 LSB * 1.5 mG / LSB) and the event is notified by hardware through the INT_MAG/DRDY pin.

3.7 Magnetometer self-test

The embedded self-test function allows checking device functionality without moving it. When the magnetometer self-test is enabled, a current is forced into a coil inside the device. This current will generate a magnetic field that will produce a variation of the magnetometer output signals. If the output signals change within the amplitude limits, then the sensor is working properly and the parameters of the interface chip are within the defined specifications.

The magnetometer self-test function is off when the Self_test bit of the CFG_REG_C_M register is disabled; setting the Self_test bit enables the self-test.

When the magnetometer self-test is activated, the sensor output level is given by the algebraic sum of the signals produced by the magnetic field acting on the sensor and by the current forced.

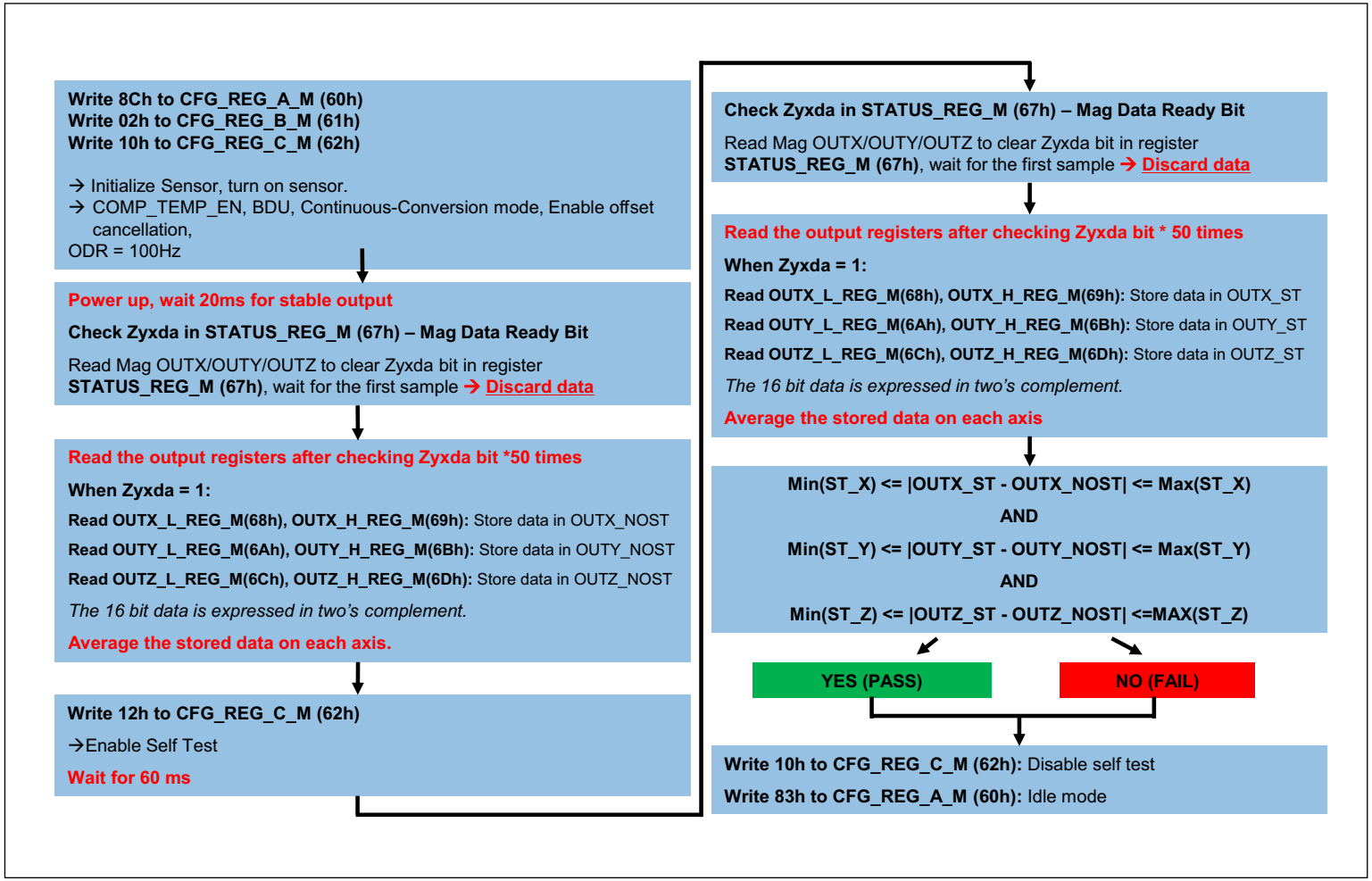
The procedure consists of:

1. enabling the magnetometer;
2. averaging five samples before enabling the self-test;
3. averaging five samples after enabling the self-test;
4. computing the difference in the module for each axis and verifying that it falls in the given range: the min and max value are provided in the datasheet.

The complete magnetometer self-test procedure is indicated in *Figure 3*.

Note: Keep the device still during the self-test procedure.

Figure 3. Magnetometer self-test procedure



4 Accelerometer

4.1 Operating modes

The LSM303AGR provides three different operating modes: high-resolution mode, normal mode, and low-power mode.

After the power supply is applied, the LSM303AGR performs a 5 ms boot procedure to load the trimming parameter. After the boot is completed, the accelerometer is automatically configured in power-down mode.

Referring to the LSM303AGR datasheet, the low-power enable (LPen) bit of CTRL_REG1_A and the HR bit of CTRL_REG4_A are used to select the operating modes (power-down mode, normal mode and low-power mode) and output data rate (ODR[3:0]) (see *Table 9* and *Table 10*).

Table 9. Operating mode selection

Operating mode	CTRL_REG1_A[3] (LPen bit)	CTRL_REG4_A[3] (HR bit)	BW [Hz]	Turn-on time [ms]	So @ ±2 *g* [mg/digit]
Low-power mode (8-bit data output)	1	0	ODR/2	1	16
Normal mode (10-bit data output)	0	0	ODR/2	1.6	4
High-resolution mode (12-bit data output)	0	1	ODR/9	7/ODR	1
Not allowed	1	1	--	--	--

Table 10. Data rate configuration

ODR3	ODR2	ODR1	ODR0	Power mode selection
0	0	0	0	Power-down mode
0	0	0	1	HR / Normal / Low-power mode (1 Hz)
0	0	1	0	HR / Normal / Low-power mode (10 Hz)
0	0	1	1	HR / Normal / Low-power mode (25 Hz)
0	1	0	0	HR / Normal / Low-power mode (50 Hz)
0	1	0	1	HR / Normal / Low-power mode (100 Hz)
0	1	1	0	HR / Normal / Low-power mode (200 Hz)
0	1	1	1	HR/ Normal / Low-power mode (400 Hz)
1	0	0	0	Low-power mode (1.620 kHz)
1	0	0	1	HR/ Normal (1.344 kHz); Low-power mode (5.376 kHz)

Table 11 shows the typical values of the power consumption for the different operating modes.

Table 11. Current consumption of operating modes

ODR [Hz]	Low-power mode (8-bit data output) [µA]	Normal mode (10-bit data output) [µA]	High resolution (12-bit data output) [µA]
1	3.7	3.7	3.7
10	4.4	5.4	5.4
25	5.6	8	8
50	7.7	12.6	12.6
100	11.7	22	22
200	20	40	40
400	36	75	75
1344	--	185	185
1620	102	--	--
5376	186	--	--

4.1.1 Power-down mode

When the device is in power-down mode, almost all internal blocks of the device are switched off to minimize power consumption. Digital interfaces (I^2C and SPI) are still active to allow communication with the device. The content of the configuration registers is preserved and output data registers are not updated, therefore keeping the last data sampled in memory before going into power-down mode.

4.1.2 High-resolution mode

In HR mode, data are generated at the data rate (ODR) selected through the ODR bits and for the axis enabled through the Zen, Yen and Xen bits of CTRL_REG1_A. Data generated for a disabled axis is 00h.

Data Interrupt generation is active and configured through the INT1_CFG_A and INT2_CFG_A registers.

In HR mode, the numbers of samples used to generate the output sample is four times greater than the number used for low-power mode and the bandwidth is ODR/9.

4.1.3 Normal mode

In normal mode, data are generated at the data rate (ODR) selected through the ODR bits and for the axis enabled through the Zen, Yen, and Xen bits of CTRL_REG1_A. Data generated for a disabled axis is 00h.

Data interrupt generation is active and configured through the INT1_CFG_A and INT2_CFG_A registers.

In normal mode, the numbers of samples used to generate the output sample is four times greater than the number used for low-power mode and the bandwidth is ODR/2.

4.1.4 Low-power mode

In low-power mode, data are generated at the data rate (ODR) selected through the ODR bits and for the axis enabled through the Zen, Yen, and Xen bits of CTRL_REG1_A. Data generated for a disabled axis is 00h.

Data interrupt generation is active and configured through the INT1_CFG_A and INT2_CFG_A registers.

In low-power mode, the numbers of samples used to generate the output sample is four times less than the number used for HR and normal mode and the bandwidth is ODR/2.

4.1.5 Switching modes

The turn-on time to transition to another operating mode is given in *Table 12*.

Table 12. Turn-on time for operating mode transition

Operating mode change	Turn-on time [ms]
12-bit mode to 8-bit mode	1/ODR
12-bit mode to 10-bit mode	1/ODR
10-bit mode to 8-bit mode	1/ODR
10-bit mode to 12-bit mode	7/ODR
8-bit mode to 10-bit mode	1/ODR
8-bit mode to 12-bit mode	7/ODR

4.2 Startup sequence

Once the device is powered-up, it automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed, i.e. after approximately 5 milliseconds, the device automatically enters power-down mode. To turn on the device and gather acceleration data, it is necessary to select one of the operating modes through CTRL_REG1_A and to enable at least one of the axes.

The following general-purpose sequence can be used to configure the device:

1. Write to CTRL_REG1_A
2. Write to CTRL_REG2_A
3. Write to CTRL_REG3_A
4. Write to CTRL_REG4_A
5. Write to CTRL_REG5_A
6. Write to CTRL_REG6_A
7. Write to REFERENCE/DATACAPTURE_A
8. Write to INT1_THS_A
9. Write to INT1_DUR_A
10. Write to INT1_CFG_A
11. Write to CTRL_REG5_A

4.2.1 Reading acceleration data

Using the status register

The device is provided with a STATUS_REG_A which should be polled to check when a new set of data is available. The reading procedure should be the following:

1. Read STATUS_REG_A
2. If STATUS_REG_A(3) = 0 then go to 1
3. If STATUS_REG_A(7) = 1 then some data have been overwritten
4. Read OUTX_L_A
5. Read OUTX_H_A
6. Read OUTY_L_A
7. Read OUTY_H_A
8. Read OUTZ_L_A
9. Read OUTZ_H_A
10. Data processing
11. Go to 1

The check performed at step 3 allows understanding whether the reading rate is adequate compared to the data production rate. If one or more acceleration samples have been overwritten by new data, because of an insufficient reading rate, the ZYXOR bit of STATUS_REG_A is set to 1.

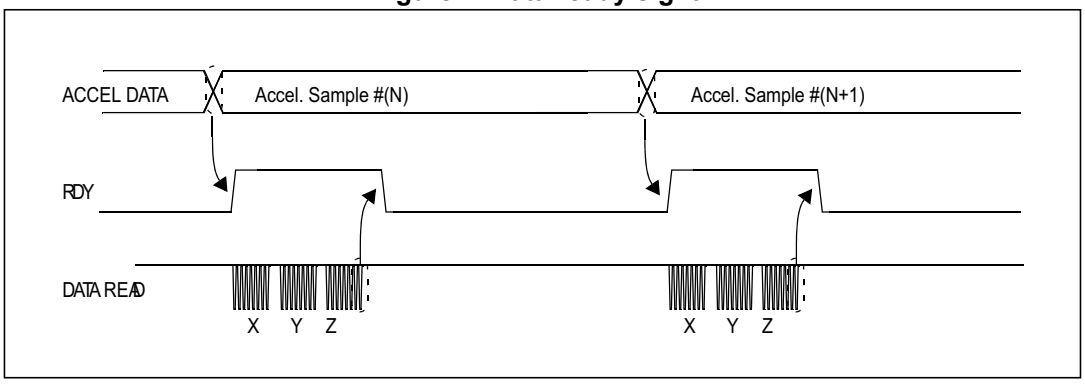
The overrun bits are automatically cleared when all the data present inside the device have been read and new data have not been produced in the meantime.

Using the data-ready (DRY) signal

The device may be configured to have one HW signal to determine when a new set of measurement data is available for reading. This signal is represented by the XYZDA bit of STATUS_REG_A. The signal can be driven to the INT1 pin by setting the I1_DRDY1 bit of CTRL_REG3_A to 1 and its polarity set to active-low or active-high through the H_LACTIVE bit of CTRL_REG6_A.

The data-ready signal rises to 1 when a new set of acceleration data has been generated and is available for reading.The interrupt is reset when the higher part of the data of all the enabled channels has been read (29h, 2Bh, 2Dh).

Figure 4. Data-ready signal



Using the block data update (BDU) feature

If the reading of the acceleration data is particularly slow and cannot be synchronized (or it is not required) with either the XYZDA bit present inside the STATUS_REG_A or with the RDY signal, it is strongly recommended to set the BDU (block data update) bit in CTRL_REG4_A to 1.

This feature avoids the reading of values (most significant and least significant parts of the acceleration data) related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent acceleration data produced by the device, but, in case the read of a given pair (i.e. OUT_X_H_A and OUT_X_L_A, OUT_Y_H_A and OUT-Y_L_A, OUT_Z_H_A and OUT_Z_L_A) is initiated, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

Note: BDU only guarantees that OUT_X(Y, Z)_L_A and OUT_X(X,Z)_H _A have been sampled at the same moment. For example, if the reading speed is too slow, it may read X and Y sampled at T1 and Z sampled at T2.

4.2.2 Understanding acceleration data

The measured acceleration data are sent to the OUT_X_H_A, OUT_X_L_A, OUT_Y_H_A, OUT_Y_L_A, OUT_Z_H_A, and OUT_Z_L_A registers. These registers contain, respectively, the most significant part and the least significant part of the acceleration signals acting on the X, Y, and Z axes.

The complete acceleration data for the X (Y, Z) channel is given by the concatenation OUT_X_H_A & OUT_X_L_A (OUT_Y_H_A & OUT_Y_L_A, OUT_Z_H_A & OUT_Z_L_A) and it is expressed as a 2’s complement number.

Data alignment

Acceleration data are represented as 16-bit numbers and are left-justified.

Big-little endian selection

The LSM303AGR allows swapping the content of the lower and the upper part of the acceleration registers (i.e. OUT_X_H_A with OUT_X_L_A), to be compliant with both little-endian and big-endian data representations.

“Little Endian” means that the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address. (The little end comes first). This mode corresponds to bit BLE in CTRL_REG4_A reset to 0 (default configuration).

On the contrary, “Big Endian” means that the high-order byte of the number is stored in memory at the lowest address, and the low-order byte at the highest address.

Example of acceleration data

Table 13 provides a few basic examples of the data that is read in the data registers when the device is subject to a given acceleration. The values listed in the table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....) and practically show the effect of the BLE bit.

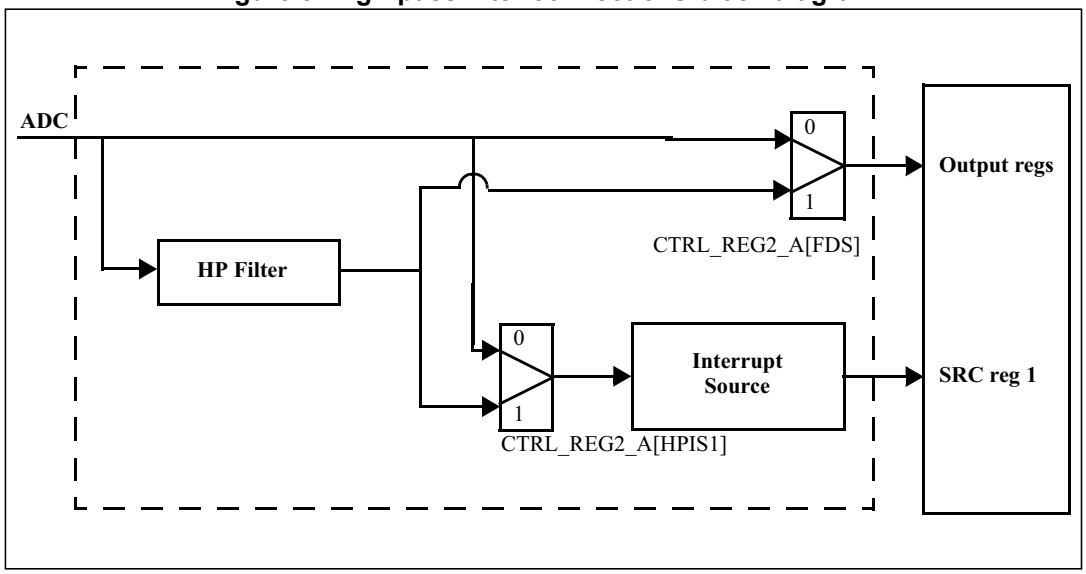
Table 13. Output data registers content vs. acceleration (FS = 2 *g*)

Acceleration values	BLE = 0		BLE = 1	
	Register address			
	28h	29h	28h	29h
0 *g*	00h	00h	00h	00h
350 m*g*	E0h	15h	15h	E0h
1 *g*	00h	04h	04h	00h
-350 m*g*	20h	EAh	EAh	20h
-1 *g*	00h	C0h	C0h	00h

4.3 High-pass filter

The LSM303AGR provides an embedded high-pass filtering capability to easily delete the DC component of the measured acceleration. As shown in *Figure 5*, through the FDS, HPIS2 and HPIS1 bits of the CTRL_REG2_A configuration, it is possible to independently apply the filter on the output data and/or on the interrupts data. This means that it is possible, for example, to get filtered data while the interrupt generation works on unfiltered data.

Figure 5. High-pass filter connections block diagram



4.3.1 Filter configuration

Referring to *Table 14*, two operating modes are possible for the high-pass filter:

Table 14. High-pass filter mode configuration

HPM1	HPM0	High-pass filter mode
0	0	Normal mode (reset by reading the REFERENCE/DATACAPTURE_A register)
0	1	Reference signal for filtering
1	0	Normal mode
1	1	Autoreset on interrupt event

The bandwidth of the high-pass filter depends on the selected ODR and on the settings of HPCF[2:1] bits of CTRL_REG2_A. The high-pass filter cutoff frequencies (f_t) are shown in *Table 15*.

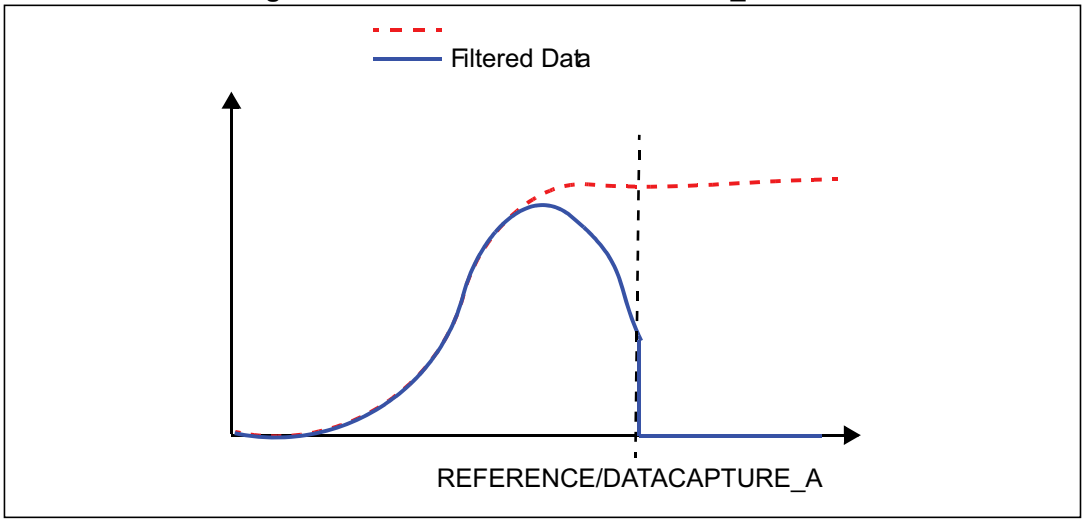
Table 15. Low power mode - high-pass filter cutoff frequency [Hz]

HPC	f_t [Hz] @ 1Hz	f_t [Hz] @ 10Hz	f_t [Hz] @ 25Hz	f_t [Hz] @ 50Hz	f_t [Hz] @ 100Hz	f_t [Hz] @ 200Hz	f_t [Hz] @ 400Hz	f_t [Hz] @ 1.6 kHz	f_t [Hz] @ 5 kHz
00	0.02	0.2	0.5	1	2	4	8	32	100
01	0.008	0.08	0.2	0.5	1	2	4	16	50
10	0.004	0.04	0.1	0.2	0.5	1	2	8	25
11	0.002	0.02	0.05	0.1	0.2	0.5	1	4	12

Normal mode

In this configuration the high-pass filter can be reset by reading the REFERENCE/DATACAPTURE_A register, instantly deleting the DC component of the acceleration.

Figure 6. REFERENCE/DATACAPTURE_A reads



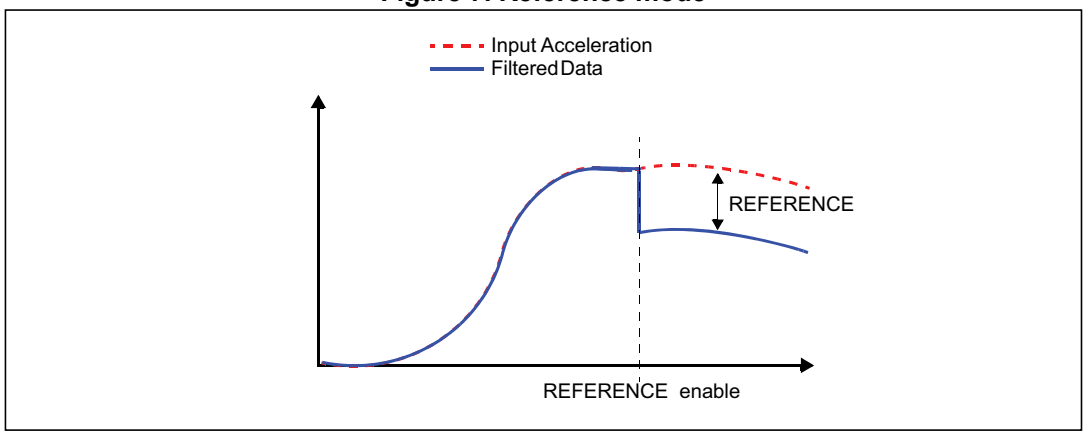
Reference mode

In this configuration the output data is calculated as the difference between the input acceleration and the content of the REFERENCE/DATACAPTURE_A register. This register is in 2's complement representation and the value of 1 LSB of these 7-bit registers depends on the selected full scale (*Table 16*).

Table 16. Reference mode LSB value

Full scale	Reference mode LSB value (mg)
2	~16
4	~31
8	~63

Figure 7. Reference mode

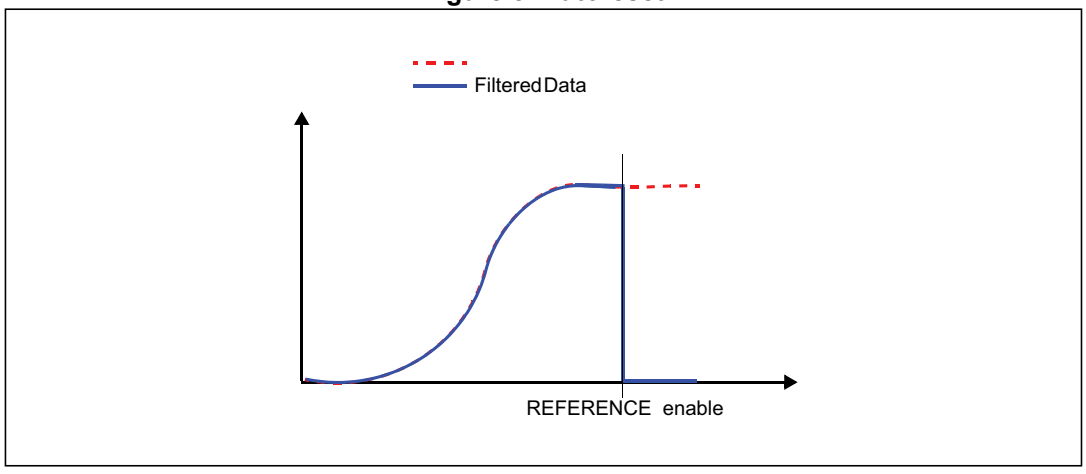


Autoreset

In this configuration the filter is automatically reset when the configured interrupt event occurs. REFERENCE/DATACAPTURE_A is, however, used to set the filter instantaneously.

Note: The XYZ dataset used to reset the filter is the one after the interrupt.

Figure 8. Autoreset



4.4 Interrupt generation

The LSM303AGR interrupt signals can behave as free-fall, wake-up, 6D and 4D orientation detection, and click detection. Those signals can be driven to the two interrupt pins (INT1 and INT2).

4.4.1 Interrupt pin configuration

The device is provided with two pins which can be activated to generate either the data-ready or the interrupt signals. The functionality of the pins is selected through CTRL_REG3_A (22h) and CTRL_REG6_A (25h). Refer to *Table 17* and *Table 18* and to the block diagram given in *Figure 9*.

Table 17. CTRL_REG3_A register

I1_CLICK	I1_AOI1	I1_AOI2	I1_DRDY1	I1_DRDY2	I1_WTM	I1_OVERRUN	-

Table 18. CTRL_REG3 description

I1_CLICK	CLICK interrupt on INT1. Default value 0 (0: disable; 1: enable)
I1_AOI1	AOI1 interrupt on INT1 pin. Default value 0 (0: disable; 1: enable)
I1_AOI2	AOI2 interrupt on INT1 pin. Default value 0 (0: disable; 1: enable)
I1_DRDY1	DRDY1 interrupt on INT1 pin. Default value 0 (0: disable; 1: enable)
I1_DRDY2	DRDY2 interrupt on INT1 pin. Default value 0 (0: disable; 1: enable)
I1_WTM	FIFO watermark interrupt on INT1. Default value 0. (0: disable; 1: enable)
I1_OVERRUN	FIFO overrun interrupt on INT1. Default value 0. (0: disable; 1: enable)

Table 19. CTRL_REG6 register

I2_CLICKen	I2_INT1	I2_INT2	BOOT_I2	P2_ACT	-	H_LACTIVE	-

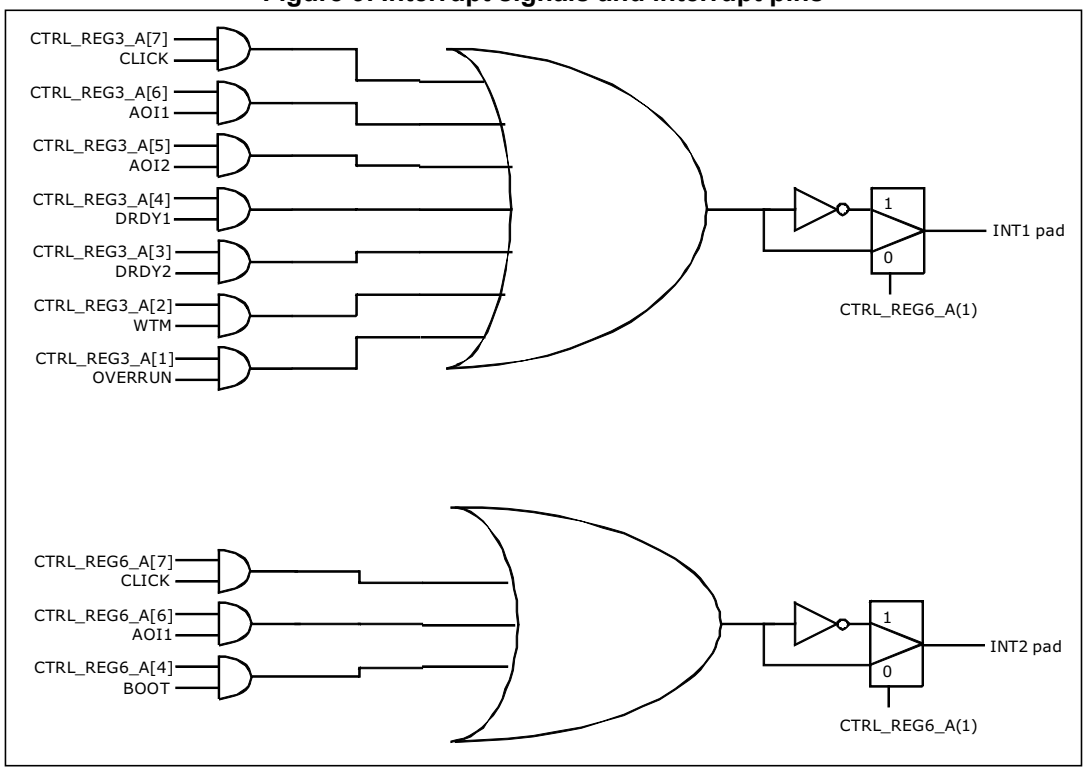
Table 20. CTRL_REG6 register

I2_CLICKen	Click interrupt on INT2 pin. Default value: 0 (0: disable; 1: enable)
I2_INT1	Interrupt 1 function enable on INT2 pin. Default value: 0 (0: disable; 1: enable)
I2_INT2	Interrupt 2 function enable on INT2 pin. Default value: 0 (0: disable; 1: enable)
BOOT_I2	Boot on INT2 pin enable. Default value: 0 (0: disable; 1: enable)

Table 20. CTRL_REG6 register (continued)

P2_ACT	Activity interrupt enable on INT2 pin. Default value: 0 (0: disable; 1: enable)
HL_ACTIVE	Interrupt active. Default value: 0 0: interrupt active high; 1: interrupt active low

Figure 9. Interrupt signals and interrupt pins



4.5 Inertial interrupt

The LSM303AGR can provide two inertial interrupt signals and offers several possibilities to personalize those signals. The registers involved in the interrupt generation behavior are INT1_CFG_A, INT1_THS_A and INT1_DURATION_A.

Table 21. Interrupt mode configuration

AOI	6D	Interrupt mode
0	0	OR combination of interrupt events
0	1	6-direction movement recognition
1	0	AND combination of interrupt events
1	1	6-direction position recognition

Whenever an interrupt condition is verified, the interrupt signal is generated and by reading the INT1_SRC_A register it is possible to understand which condition happened.

4.5.1 Duration

The content of the duration registers sets the minimum duration of the interrupt event to be recognized. Duration steps and maximum values depend on the ODR chosen.

Duration time is measured in N/ODR, where N is the content of the duration register and ODR is 50, 100, 400, 1000 Hz.

Table 22. Duration LSB value in normal mode

ODR (Hz)	Duration LSB value (ms)
1	1000
10	100
25	40
50	20
100	10
200	5
400	2.5
1000	1
1344	0.744
1620	0.617

4.5.2 Threshold

The threshold registers define the reference accelerations used by the interrupt generation circuitry. The value of 1 LSB of these 7-bit registers depends on the selected full scale (*Table 23*).

Table 23. Threshold LSB value

Full scale	Threshold LSB value (mg)
2	~16
4	~31
8	~63
16	~125

4.5.3 Free-fall and wake-up interrupts

The LSM303AGR interrupt signals can behave as free-fall and wake-up. Whenever an interrupt condition is verified, the interrupt signal is generated and by reading the INT1_SRC_A register it is possible to understand which condition happened.

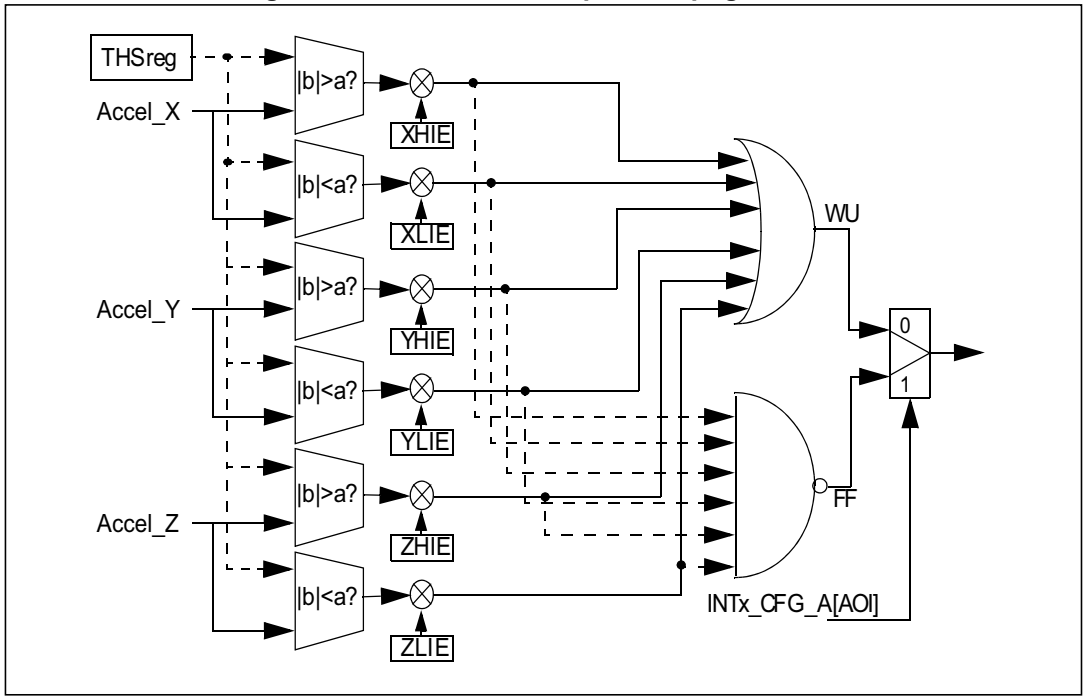
The free-fall signal (FF) and wake-up signal (WU) interrupt generation block is represented in *Figure 10*.

The FF or WU interrupt generation is selected through the AOI bit in the INT1_CFG_A register. If the AOI bit is '0', signals coming from comparators for the axis enabled through the INT1_CFG_A register are put in logical OR. In this case, the interrupt is generated when at least one of the enabled axis exceeds the threshold written in the module in the INT1_THS_A registers. Otherwise, if the AOI bit is '1', signals coming from comparators enter a "NAND" port. In this case an interrupt signal is generated only if all the enabled axes exceed the threshold written in the INT1_THS_A register.

The LIR_INT1 and LIR_INT2 bits of CTRL_REG5_A allow deciding if the interrupt request must be latched or not. If the LIR_INT1 bit is '0' (default value), the interrupt signal goes high when the interrupt condition is satisfied and returns to low immediately if the interrupt condition is no longer verified. Otherwise, if the LIR_INT1 bit is '1', whenever an interrupt condition is applied, the interrupt signal remains high even if the condition returns to a non-interrupt status until a read of the INT1_SRC_A register is performed.

The ZHIE, ZLIE, YHIE, YLIE, XHIE, and HLIE bits of the INT1_CFG_A register allow deciding on which axis the interrupt decision must be performed and on which direction the threshold must be exceeded to generate the interrupt request.

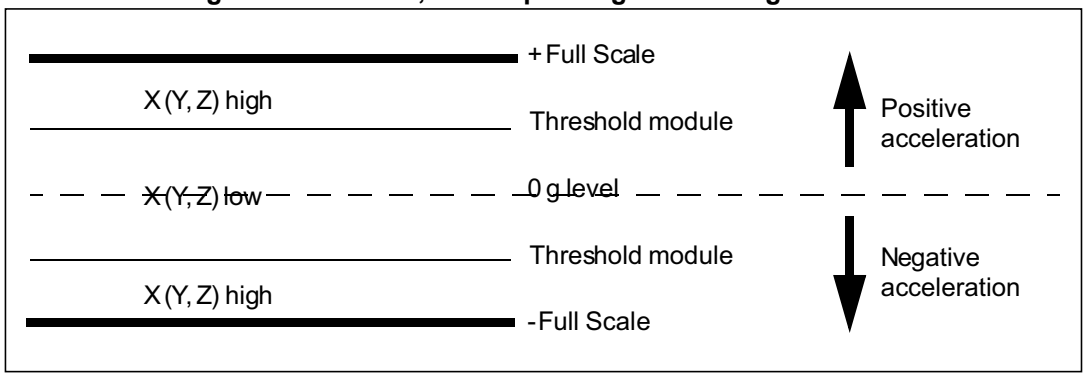
Figure 10. Free-fall, wake-up interrupt generator



The threshold module which is used by the system to detect any free-fall or inertial wake-up event is defined by the INT1_THS_A register. The threshold value is expressed over 7 bits as an unsigned number and is symmetrical around the zero-g level. XH (YH, ZH) is true

when the unsigned acceleration value of the X (Y, Z) channel is higher than INT1_THS_A. Similarly, XL (YL, ZL) low is true when the unsigned acceleration value of the X (Y, Z) channel is lower than INT1_THS_A. Refer to *Figure 11* for more details.

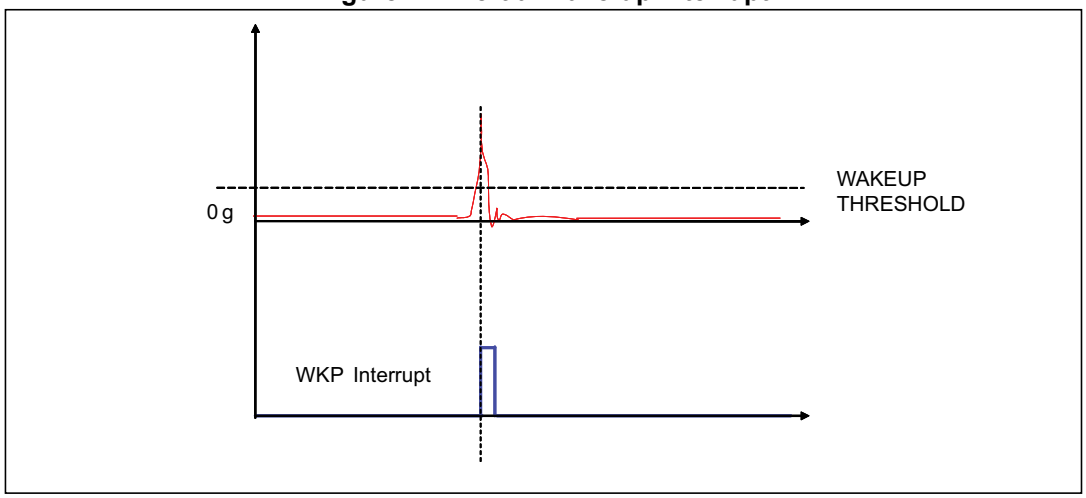
Figure 11. Free-fall, wake-up configuration - high and low



Inertial wake-up

The wake-up interrupt refers to a specific configuration of the Act_THS_A register that allows interrupt generation when the acceleration on the configured axis exceeds a defined threshold (*Figure 12*).

Figure 12. Inertial wake-up interrupt



HP filter bypassed

This paragraph provides a basic algorithm which shows the practical use of the inertial wake-up feature. In particular, with the code below, the device is configured to recognize when the absolute acceleration along either the X or Y axis exceeds a preset threshold (250 m*g* used in the example). The event which triggers the interrupt is latched inside the device and its occurrence is signaled through the use of the INT1 pin.

1	Write A7h into CTRL_REG1_A	// Turn on the sensor and enable X, Y, and Z // ODR = 100 Hz
2	Write 00h into CTRL_REG2_A	// High-pass filter disabled
3	Write 40h into CTRL_REG3_A	// Interrupt driven to INT1 pad
4	Write 00h into CTRL_REG4_A	// FS = 2 *g*
5	Write 08h into CTRL_REG5_A	// Interrupt latched
6	Write10h into INT1_THS_A	// Threshold = 250 m*g*
7	Write 00h into INT1_DURATION_A	// Duration = 0
8	Write 0Ah into INT1_CFG_A	// Enable XH and YH interrupt generation
9	Poll INT1 pad; if INT1=0 then go to 8	// Poll RDY/INT pin waiting for the // wake-up event
10	Read INT1_SRC_A	// Return the event that has triggered the // interrupt
11	(Wake-up event has occurred; insert your code here)	// Event handling
12	Go to 8	

Using the HP filter

The code provided below gives a basic routine which shows the practical use of the inertial wake-up feature performed on high-pass filtered data. In particular the device is configured to recognize when the high-frequency component of the acceleration applied along either the X, Y, or Z axis exceeds a preset threshold (250 m*g* used in the example).
The event which triggers the interrupt is latched inside the device and its occurrence is signaled through the use of the INT1 pin.

1	Write A7h into CTRL_REG1_A	// Turn on the sensor, enable X, Y, and Z // ODR = 100 Hz
2	Write 09h into CTRL_REG2_A	// High-pass filter enabled on data and interrupt1
3	Write 40h into CTRL_REG3_A	// Interrupt driven to INT1 pad
4	Write 00h into CTRL_REG4_A	// FS = 2 *g*
5	Write 08h into CTRL_REG5_A	// Interrupt latched
6	Write 10h into INT1_THS_A	// Threshold = 250 m*g*
7	Write 00h into INT1_DURATION_A	// Duration = 0

8	Read REFERENCE/DATACAPTURE_A	// Dummy read to force the HP filter to
		// current acceleration value
		// (i.e. set reference acceleration/tilt value)

9	Write 2Ah into INT1_CFG_A	// Configure desired wake-up event
10	Poll INT1 pad; if INT1 = 0 then go to 9	// Poll INT1 pin waiting for the // wake-up event
11	(Wake-up event has occurred; insert your code here)	// Event handling
12	Read INT1_SRC_A	// Return the event that has triggered the // interrupt and clear interrupt
13	(Insert your code here)	// Event handling
14	Go to 9	

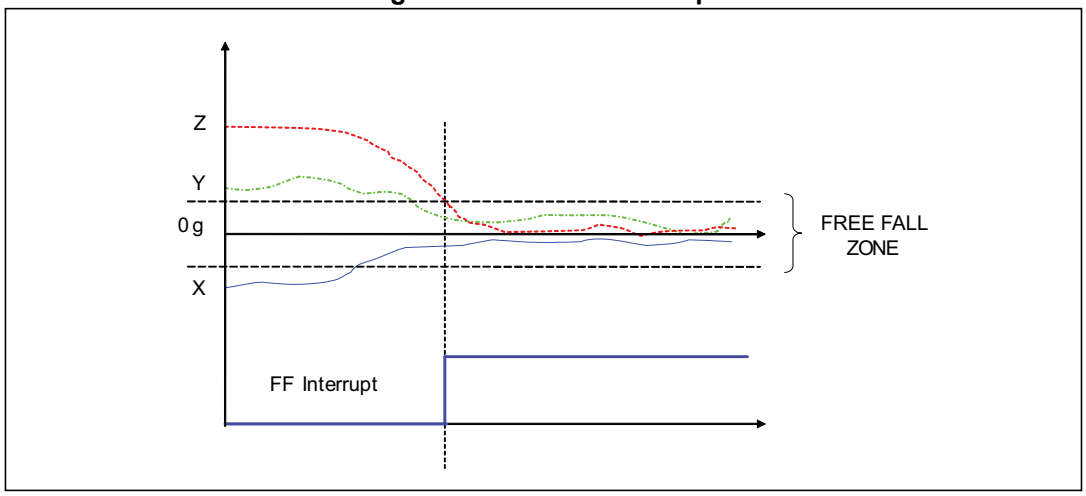
At step 8, a dummy read of the REFERENCE/DATACAPTURE_A register is performed to set the current/reference acceleration/tilt state against which the device performed the threshold comparison.

This read may be performed any time it is required to set the orientation/tilt of the device as a reference state without waiting for the filter to settle.

4.5.4 Free-fall detection

Free-fall detection refers to a specific configuration of INT1_CTRL registers that allows to recognize when the device is free falling: the acceleration measure along all the axes goes to zero. In a real case a “free-fall zone” is defined around the zero-g level where all the accelerations are small enough to generate the interrupt (*Figure 13*).

Figure 13. Free-fall interrupt



This paragraph provides the basics for the use of the free-fall detection feature. In particular, the SW routine that configures the device to detect free-fall events and to signal them is the following:

1	Write A7h into CTRL_REG1_A	// Turn on the sensor, enable X, Y, and Z // ODR = 100 Hz
2	Write 00h into CTRL_REG2_A	// High-pass filter disabled
3	Write 40h into CTRL_REG3_A	// Interrupt driven to INT1 pad
4	Write 00h into CTRL_REG4_A	// FS = 2 *g*
5	Write 08h into CTRL_REG5_A	// Interrupt latched
6	Write 16h into INT1_THS_A	// Set free-fall threshold = 350 m*g*
7	Write 03h into INT1_DURATION_A	// Set minimum event duration
8	Write 95h into INT1_CFG_A	// Configure free-fall recognition
9	Poll INT1 pad; if INT1 = 0 then go to 10	// Poll INT1 pin waiting for the free-fall event
10	(Free-fall event has occurred; insert your code here)	// Event handling
11	Read INT1_SRC_A register	// Clear interrupt request
12	Go to 9	

The code sample exploits a threshold set at 350 m*g* for free-fall recognition and the event is notified by the hardware signal INT1. At step 7, the INT1_DURATION_A register is configured like this to ignore events that are shorter than 3/DR = 3/100 ~= 30 msec in order to avoid false detections.

Once the free-fall event has occurred, a read of the INT1_SRC_A register clears the request and the device is ready to recognize other events.

4.6 6D/4D orientation detection

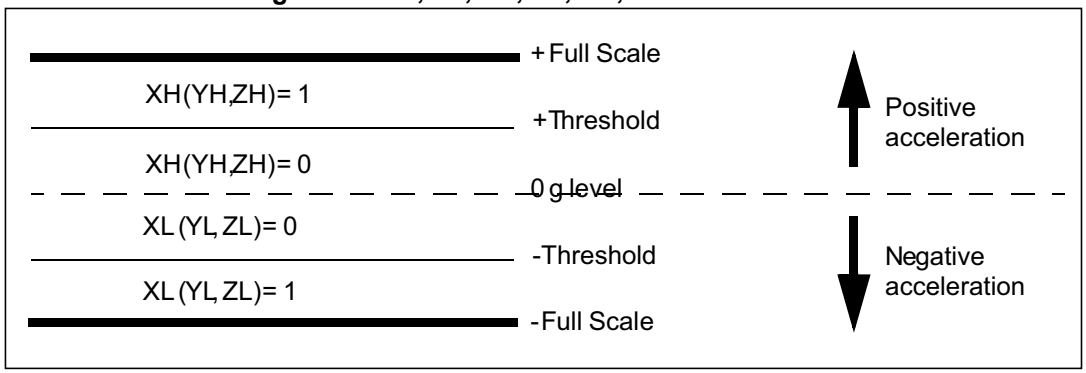
The LSM303AGR provides advanced capability to detect the orientation of the device in space, enabling easy implementation of an energy saving procedure and automatic image rotation for handheld devices.

4.6.1 6D orientation detection

The 6D orientation direction function can be enabled through the AOI and 6D bits of the INT1_CFG_A register. When configured for 6D function, the ZH, ZL, YH, YL, XH, and XL bits of INT1_SRC_A give information about the value of the acceleration generating the interrupt when it is greater than the threshold, and about its sign. In more detail:

- ZH (YH, XH) is 1 when the sensed acceleration is greater than the threshold in the positive direction
- ZL (YL, XL) is 1 when the sensed acceleration is greater than the threshold in the negative direction.

Figure 14. ZH, ZL, YH, YL, XH, and XL behavior



There are two possible configurations for the 6D direction function:

- 6D movement recognition: In this configuration the interrupt is generated when the device moves from a direction (known or unknown) to a different known direction. The interrupt is active only for 1/ODR
- 6D position recognition: In this configuration the interrupt is generated when the device is stable in a known direction. The interrupt is active as long as position is maintained (*Figure 15*, (a) and (b)).

Referring to *Figure 15*, the 6D movement line shows the behavior of the interrupt when the device is configured for 6D movement recognition on the X and Y axis (INT1_CFG_A = 0x4Ah), while the 6D position line shows the behavior of the interrupt when the device is configured for 6D position recognition on the X and Y axis (INT1_CFG_A = 0xCAh). INT1_THS_A is set to 0x21.

Referring to *Figure 16*, the device has been configured for 6D position function on the X, Y, and Z axes. *Table 24* shows the content of the INT1_SRC_A register for each position.

Figure 15. 6D movement vs. 6D position

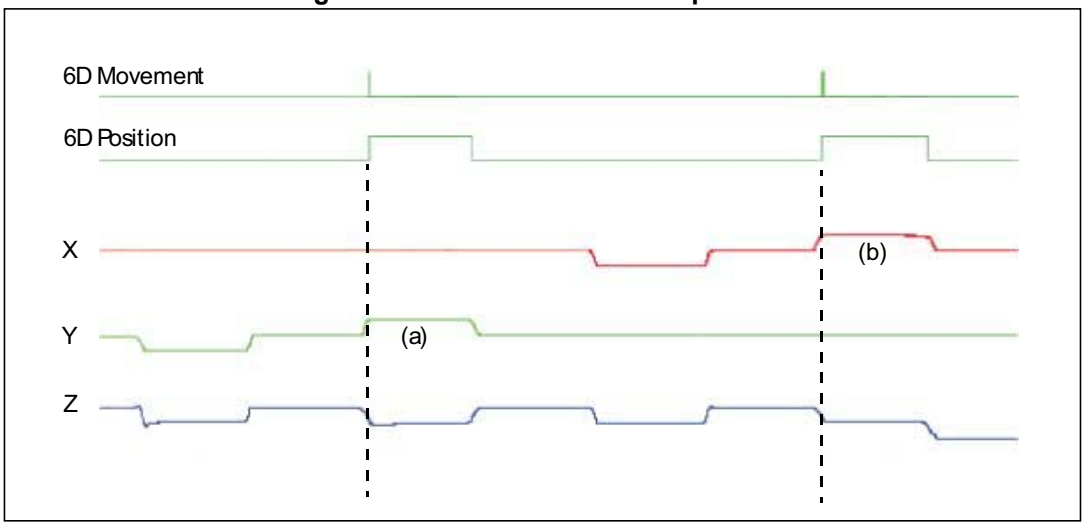


Figure 16. 6D recognized positions

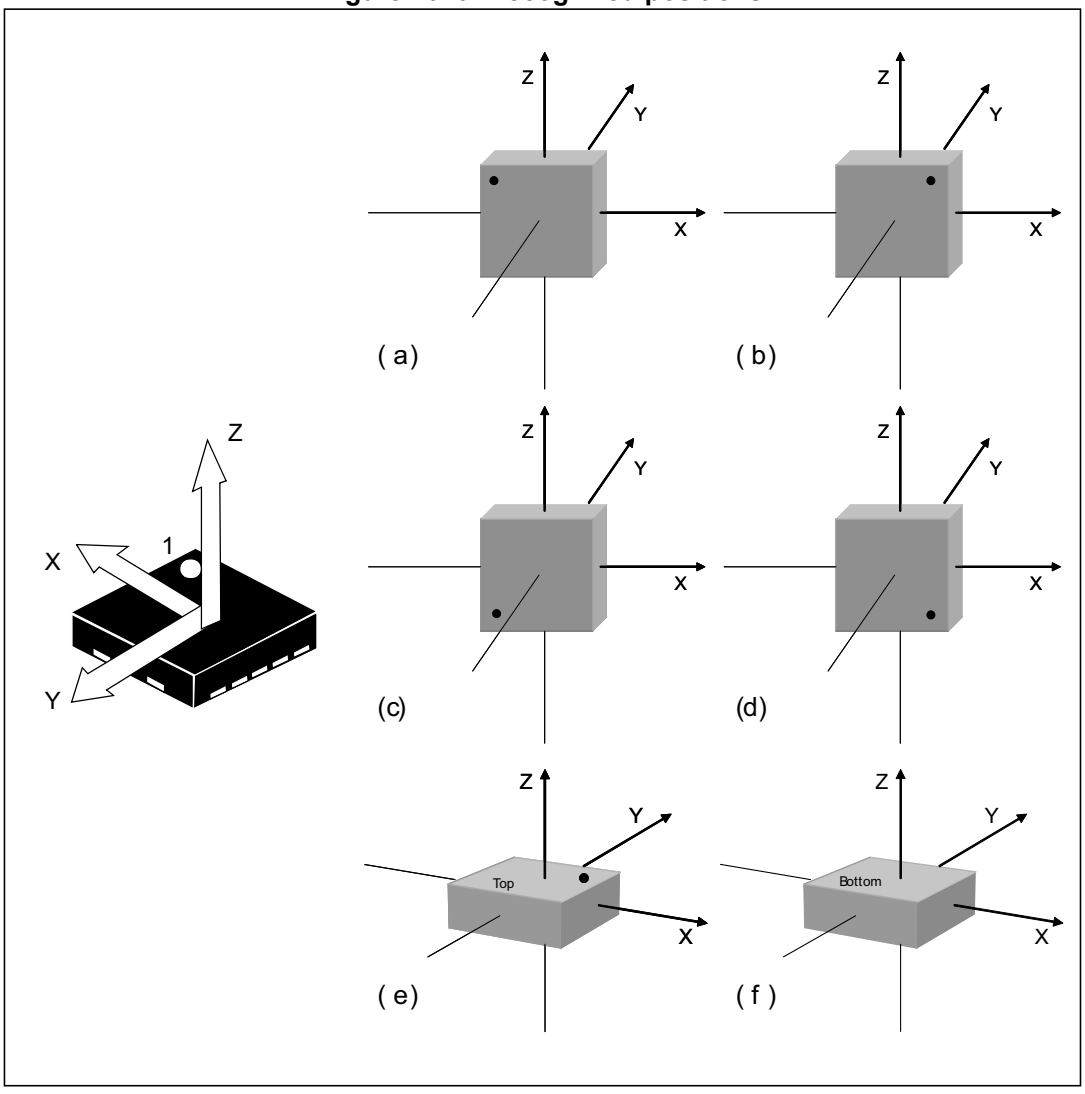


Table 24. INT1_SRC_A register in 6D position

Case	IA	ZH	ZL	YH	YL	XH	XL
(a)	1	0	0	0	1	0	0
(b)	1	0	0	0	0	1	0
(c)	1	0	0	0	0	0	1
(d)	1	0	0	1	0	0	0
(e)	1	1	0	0	0	0	0
(f)	1	0	1	0	0	0	0

4.6.2 4D direction

The 4D direction function is a subset of the 6D function especially defined to be implemented in handheld devices. It can be enabled by setting the D4D_INT1 bit of CTRL_REG5_A to 1 when the 6D bit on INT1_CFG_A is set to 1. In this configuration, the Z-axis position detection is disabled, therefore reducing position recognition to cases (a), (b), (c), and (d) of *Table 24*.

4.7 Click and double-click recognition

The single click and double-click recognition functions featured in the LSM303AGR help to create a man-machine interface with little software loading. The device can be configured to output an interrupt signal on a dedicated pin when tapped in any direction.

If the sensor is exposed to a single input stimulus, it generates an interrupt request on inertial interrupt pin INT1 and/or INT2. A more advanced feature allows the generation of an interrupt request when a double input stimulus with programmable time between the two events is recognized, enabling a mouse button-like function.

This function can be fully programmed by the user in terms of expected amplitude and timing of the stimuli by means of the dedicated set of registers described in *Section 4.7.3*.

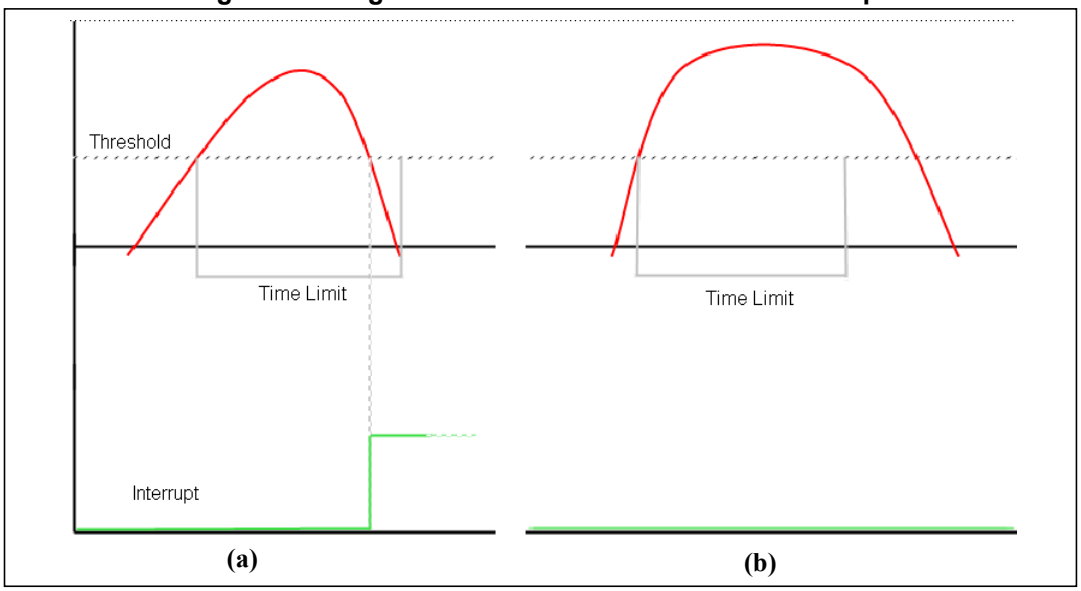
Single and double-click recognition works independently on the selected output data rate.

4.7.1 Single click

If the device is configured for single click event detection, an interrupt is generated when the input acceleration on the selected channel exceeds the programmed threshold, and returns below it within a time window defined by the TIME_LIMIT_A register.

If neither LIR_INT1 nor LIR_INT2 of CTRL_REG_5_A is set, the interrupt is kept high for the duration of the latency window. If either LIR_INT1 or LIR_INT2 is set, the interrupt is kept high until CLICK_SRC_A is read.

Figure 17. Single-click event with non-latched interrupt



In *Figure 17*(a) the click has been recognized, while in *Figure 17*(b) the click has not been recognized because the acceleration falls below the threshold after the Time Limit has expired.

4.7.2 Double click

If the device is configured for double-click event detection, an interrupt is generated when, after a first click, a second click is recognized. The recognition of the second click occurs only if the event satisfies the rules defined by the latency and windows registers.

In particular, after the first click has been recognized, the second click detection procedure is delayed for an interval defined by the latency register. This means that after the first click has been recognized, the second click detection procedure starts only if the input acceleration exceeds the threshold after the latency window but before the window has expired (*Figure 18 (a)*), or if the acceleration is still above the threshold after the latency has expired (*Figure 19 (b)*).

Once the second click detection procedure is initiated, the second click is recognized with the same rule as the first: the acceleration must return below the threshold before the Time Limit has expired.

It is important to appropriately define the latency window to avoid unwanted clicks due to spurious bouncing of the input signal.

Figure 18. Single and double-click recognition

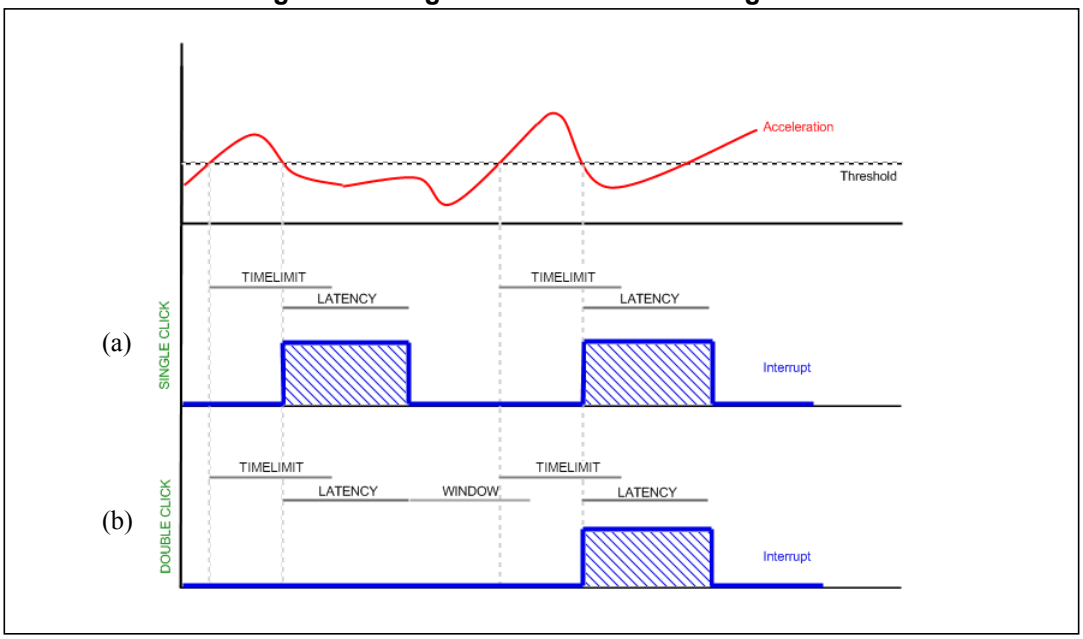
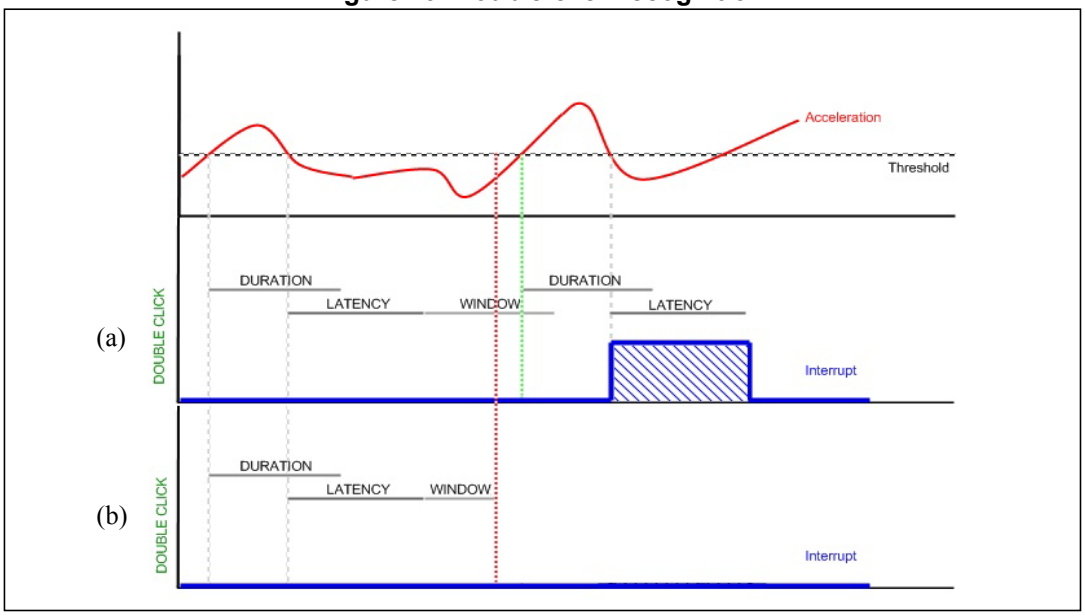


Figure 18 illustrates a single-click event (a) and a double-click event (b). The device is able to distinguish between (a) and (b) by changing the settings of the CLICK_CFG_A register from single to double-click recognition.

Figure 19. Double-click recognition



In *Figure 19* (a) the double click event has been correctly recognized, while in *Figure 19* (b) the interrupt has not been generated because the input acceleration exceeds the threshold after the window interval has expired.

4.7.3 Register description

CLICK_CFG_A (38h)

Table 25. CLICK_CFG_A register

-	-	ZD	ZS	YD	YS	XD	XS

Table 26. CLICK_CFG_A description

ZD	Enable interrupt double-click on Z-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZS	Enable interrupt single-click on Z-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YD	Enable interrupt double-click on Y-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YS	Enable interrupt single-click on Y-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XD	Enable interrupt double-click on X-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XS	Enable interrupt single-click on X-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)

Table 27. Truth table

ZD / YD / XD	ZS / YS / XS	Click output
0	0	0
0	1	Single
1	0	Double
1	1	Single or double

CLICK_SRC_A (39h)

Table 28. CLICK_SRC_A register

-	IA	DClick	SClick	Sign	Z	Y	X

Table 29. CLICK_SRC_A description

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupts have been generated)
DClick	Double-click detection enable. Default value: 0 (0: double-click detection disable, 1: double-click detection enable)
SClick	Single-click detection enable. Default value: 0 (0: single-click detection disable, 1: single-click detection enable)
Sign	Click sign. 0: positive detection, 1: negative detection
Z	Z click detection. Default value: 0 (0: no interrupt, 1: Z high event has occurred)
Y	Y click detection. Default value: 0 (0: no interrupt, 1: Y high event has occurred)
X	X click detection. Default value: 0 (0: no interrupt, 1: X high event has occurred)

CLICK_THS_A (3Ah)

Table 30. CLICK_THS_A register

-	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Table 31. CLICK_THS_A description

THS6-THS0	Click threshold. Default value: 000 0000

1 LSB = full scale/128.

THS6 through THS0 define the threshold which is used by the system to start the click detection procedure. The threshold value is expressed over 6 bits as an unsigned number.

TIME_LIMIT_A (3Bh)

Table 32. TIME_LIMIT_A register

-	TLI6	TLI5	TLI4	TLI3	TLI2	TLI1	TLI0

Table 33. TIME_LIMIT_A register

TLI7-TLI0	Click time limit. Default value: 000 0000

1 LSB = 1/ODR.

TLI7 through TLI0 define the maximum time interval that can elapse between the start of the click detection procedure (the acceleration on the selected channel exceeds the programmed threshold) and when the acceleration falls back below the threshold.

TIME_LATENCY_A (3Ch)

Table 34. TIME_LATENCY_A register

TLA7	TLA6	TLA5	TLA4	TLA3	TLA2	TLA1	TLA0

Table 35. TIME_LATENCY_A description

TLA7-TLA0	Click time latency. Default value: 000 0000

1 LSB = 1/ODR.

TLA7 through TLA0 define the time interval that starts after the first click detection where the click detection procedure is disabled, in cases where the device is configured for double click detection.

TIME_WINDOW_A (3Dh)

Table 36. TIME_WINDOW_A register

TW7	TW6	TW5	TW4	TW3	TW2	TW1	TW0

Table 37. TIME_LATENCY_A description

TW7-TW0	Click time window

1 LSB = 1/ODR.

TW7 through TW0 define the maximum interval of time that can elapse after the end of the latency interval in which the click detection procedure can start, in cases where the device is configured for double-click detection.

CTRL_REG3_A [Interrupt CTRL register] (22h)

Table 38. CTRL_REG3_A register

I1_CLICK	I1_AOI1	I1_AOI2	I1_DRDY1	I1_DRDY2	I1_WTM	I1_OVERRUN	-

Table 39. CTRL_REG3_A description

I1_CLICK	Click interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_AOI1	AOI1 interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_AOI2	AOI2 interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_DRDY1	DRDY1 interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_DRDY2	DRDY2 interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_WTM	FIFO watermark interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)
I1_OVERRUN	FIFO overrun interrupt on INT1 pin. Default value: 0 (0: disable; 1: enable)

4.7.4 Examples

The following figures show the click interrupt generation in different conditions. The illustrations have been captured on a PC running the demonstration kit GUI interface with ODR set to 400 Hz and full scale to 4 *g*. The content of the LSM303AGR registers have been modified via the dedicated panel of the software interface that allows the user to evaluate all the different settings and features of the click embedded function. In the following examples, only the X-axis has been enabled for the click interrupt generation.

Playing with TAP_TimeLimit

Figure 20 shows an acquisition carried out with TAP_TimeLimit = 01h (2.5 ms). With this setting, the single-click recognition window is short and often the acceleration does not fall below the threshold in time.

In *Figure 21* an acquisition done with TAP_TimeLimit = 33h (127 ms) is shown. With this setting the single-click recognition window is longer, and it is easier for the event to be recognized.

Figure 20. Short TimeLimit

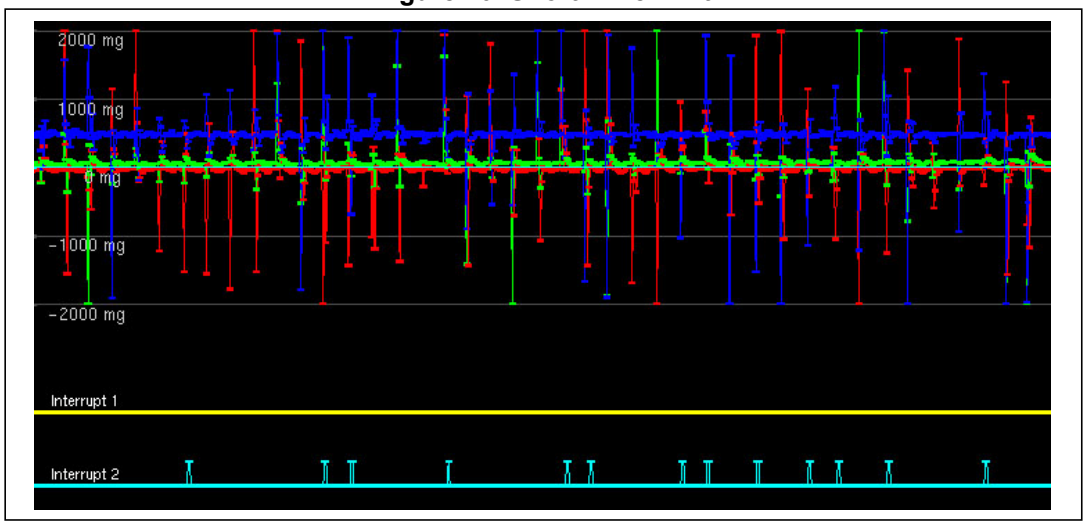
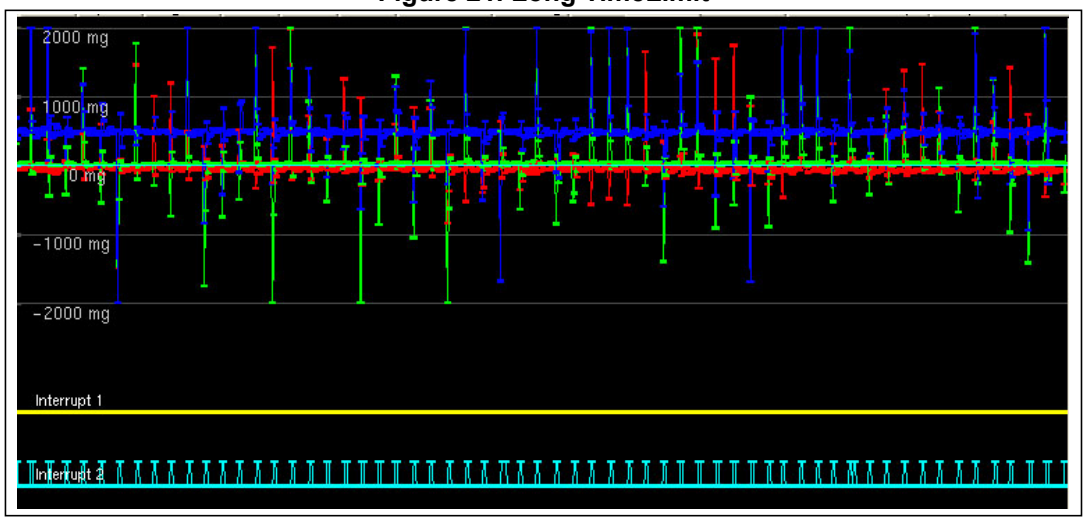


Figure 21. Long TimeLimit



Playing with TAP_Latency

Figure 22 illustrates an acquisition done with TAP_Latency = 15h (52 ms). With this setting the device recognizes nearly every acceleration peak as a click.

In *Figure 23* an acquisition carried out with TAP_Latency = FFh (637 ms) is displayed. With this setting the device recognizes one peak in every two as a click.

Figure 22. Short latency

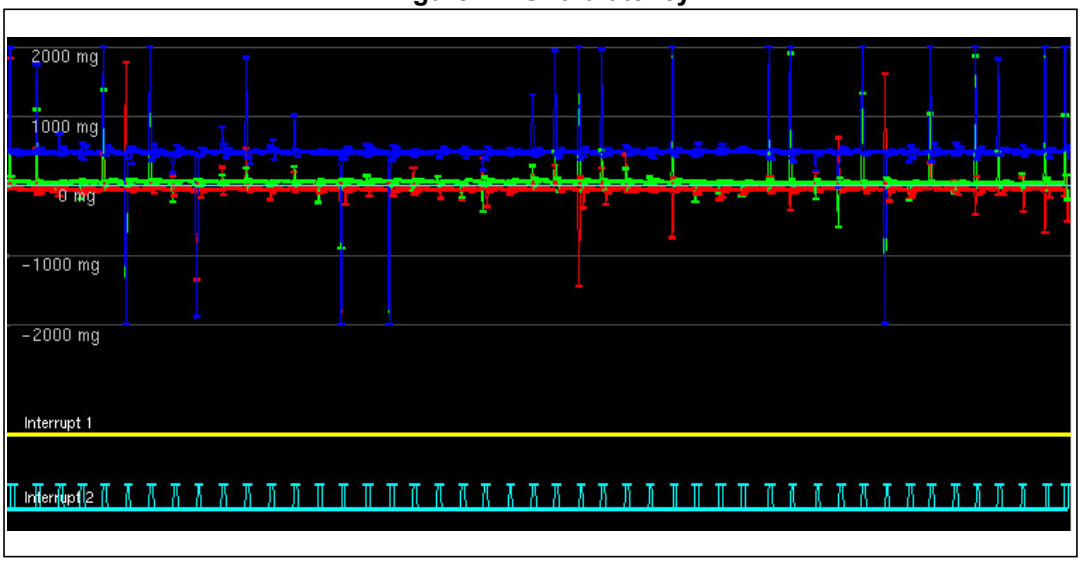
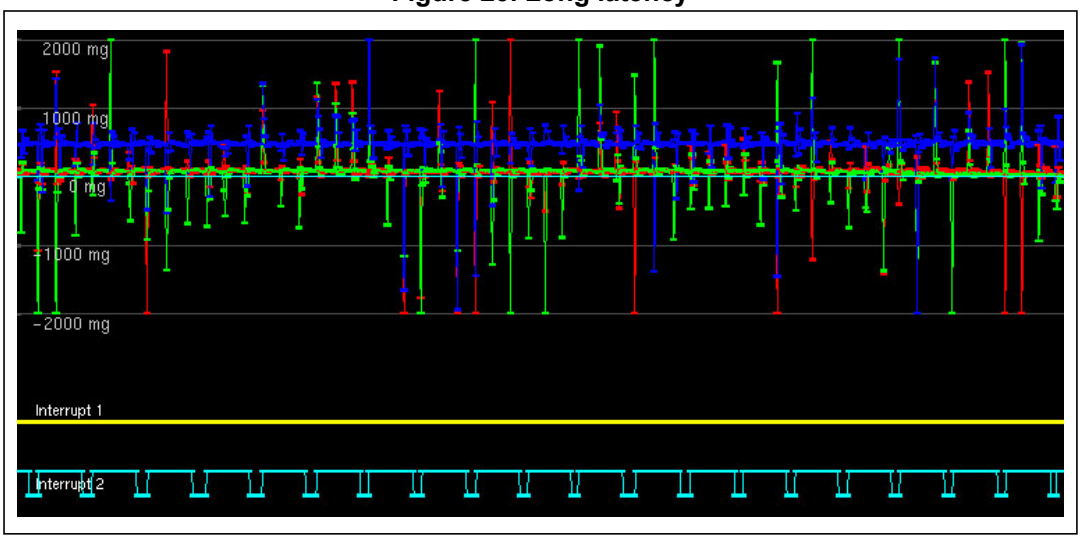


Figure 23. Long latency



Playing with TAP_Window

In cases of double-click recognition, the TAP_Latency + TAP_Window defines the maximum distance between two consecutive clicks to be recognized as a double-click event. By fixing the latency to avoid spurious bouncing of the signal, it is possible to play with the TAP_Window as with the “double-click speed” settings of the mouse properties on the PC.

Figure 24 shows an acquisition done with TAP_Window = 42h (1065 ms). With this setting the two consecutive peaks of acceleration are too far apart and the second one occurs outside of the window.

In *Figure 25* an acquisition carried out with TAP_Window = FFh (637 ms) is shown. With this setting the device correctly generates the double-click interrupt after the second acceleration peak.

Figure 24. Short window

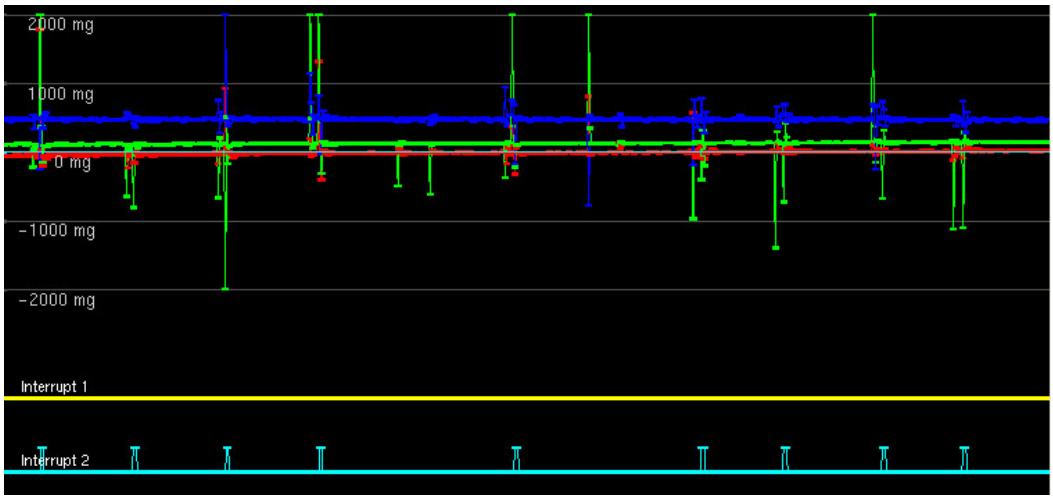
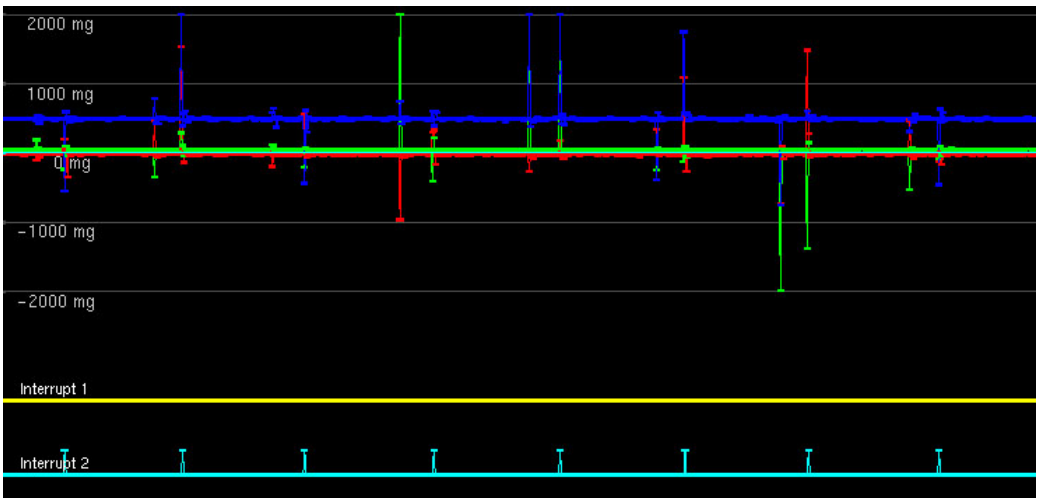


Figure 25. Long window



4.8 First-in first-out (FIFO) buffer

In order to limit intervention by the host processor and facilitate post-processing data for event recognition, LSM303AGR embeds a first-in first-out buffer (FIFO) for each of the three output channels, X, Y, and Z.

FIFO use allows a consistent power saving for the system, it can wake-up only when needed and burst the significant data out from the FIFO.

The FIFO buffer can work according to four different modes that guarantee a high-level of flexibility during application development: Bypass mode, FIFO mode, Stream mode, and Stream-to-FIFO mode.

The programmable watermark level and FIFO overrun event can be enabled to generate dedicated interrupts on the INT1 pin.

4.8.1 FIFO description

The FIFO buffer is able to store up to 32 acceleration samples of 10 bits for each channel; data are stored in the 16-bit 2’s complement left-justified representation.

The data sample set consists of 6 bytes (Xl, Xh, Yl, Yh, Zl, and Zh) and they are released to the FIFO at the selected output data rate (ODR).

The new sample set is placed in the first empty FIFO slot until the buffer is full, therefore, the oldest value is overwritten.

Table 40. FIFO buffer full representation (32nd sample set stored)

Output registers	0x28h	0x29h	0x2Ah	0x2Bh	0x2Ch	0x2Dh
	Xl(0)	Xh(0)	Yl(0)	Yh(0)	Zl(0)	Zh(0)
FIFO index	**FIFO sample set**					
FIFO(0)	Xl(0)	Xh(0)	Yl(0)	Yh(0)	Zl(0)	Zh(0)
FIFO(1)	Xl(1)	Xh(1)	Yl(1)	Yh(1)	Zl(1)	Zh(1)
FIFO(2)	Xl(2)	Xh(2)	Yl(2)	Yh(2)	Zl(2)	Zh(2)
FIFO(3)	Xl(3)	Xh(3)	Yl(3)	Yh(3)	Zl(3)	Zh(3)
...	...	...	...	...	...	...
...	...	...	...	...	...	...
FIFO(30)	Xl(30)	Xh(30)	Yl(30)	Yh(30)	Zl(30)	Zh(30)
FIFO(31)	Xl(31)	Xh(31)	Yl(31)	Yh(31)	Zl(31)	Zh(31)

Table 41. FIFO overrun representation (33rd sample set stored and 1st sample discarded)

Output registers	0x28h	0x29h	0x2Ah	0x2Bh	0x2Ch	0x2Dh
	Xl(1)	Xh(1)	Yl(1)	Yh(1)	Zl(1)	Zh(1)
FIFO index	**FIFO sample set**					
FIFO(0)	Xl(1)	Xh(1)	Yl(1)	Yh(1)	Zl(1)	Zh(1)
FIFO(1)	Xl(2)	Xh(2)	Yl(2)	Yh(2)	Zl(2)	Zh(2)
FIFO(2)	Xl(3)	Xh(3)	Yl(3)	Yh(3)	Zl(3)	Zh(3)
FIFO(3)	Xl(4)	Xh(4)	Yl(4)	Yh(4)	Zl(4)	Zh(4)
...	...	...	...	...	...	...
...	...	...	...	...	...	...
FIFO(30)	Xl(31)	Xh(31)	Yl(31)	Yh(31)	Zl(31)	Zh(31)
FIFO(31)	Xl(32)	Xh(32)	Yl(32)	Yh(32)	Zl(32)	Zh(32)

Table 40 represents the FIFO full status when 32 samples are stored in the buffer while *Table 41* represents the next step when the 33rd sample is inserted into FIFO and the 1st sample is overwritten. The new oldest sample set is made available in the output registers.

When FIFO is enabled and the mode is other than Bypass, the LSM303AGR output registers (28h to 2Dh) always contain the oldest FIFO sample set.

4.8.2 FIFO registers

The FIFO buffer is managed by three different accelerometer registers, two of these allow enabling and configuring FIFO behavior, the third provides information about the buffer status.

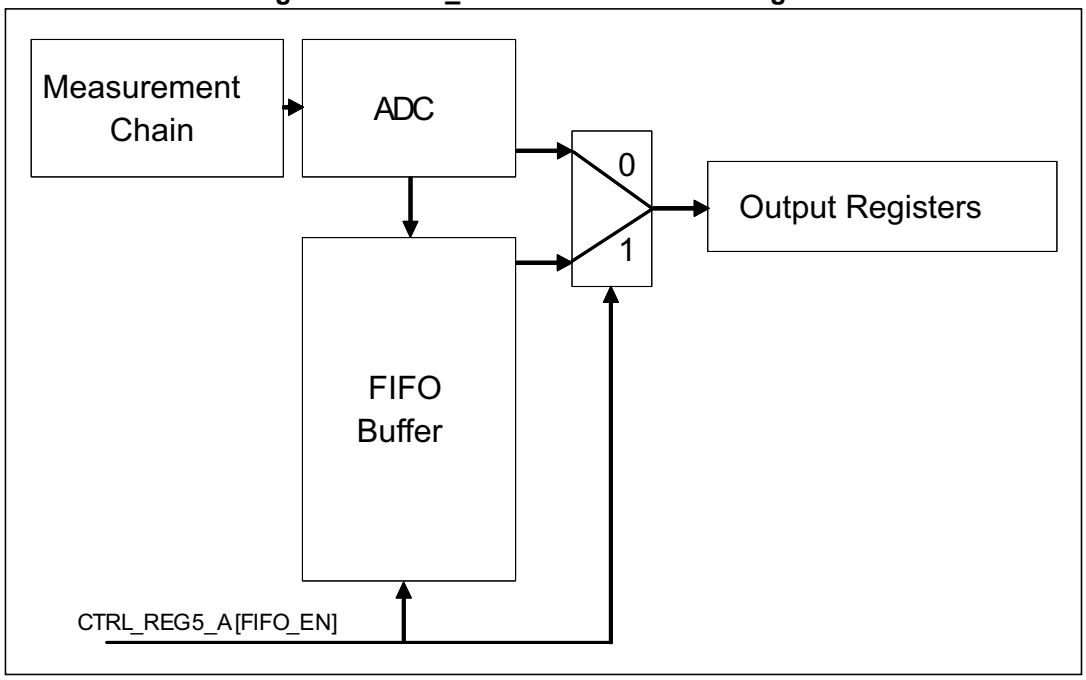
Control register 5 (0x24)

The FIFO_EN bit in CTRL_REG5_A must be set to 1 in order to enable the internal first-in first-out buffer; when this bit is set, the accelerometer output registers (28h to 2Dh) don't contain the current acceleration value but they always contain the oldest value stored in FIFO.

Table 42. FIFO enable bit in CTRL_REG5_A

b7	b6	b5	b4	b3	b2	b1	b0
X	FIFO_EN	X	X	X	X	X	X

Figure 26. FIFO_EN connection block diagram



FIFO control register (0x2E)

This register is dedicated to FIFO mode selection and watermark configuration.

Table 43. FIFO_CTRL_REG_A

b7	b6	b5	b4	b3	b2	b1	b0
FM1	FM0	TR	FTH4	FTH3	FTH2	FTH1	FTH0

The FM[1:0] bits are dedicated to define the FIFO buffer behavior selection:

1. FM[1:0] = (0,0): Bypass mode
2. FM[1:0] = (0,1): FIFO mode
3. FM[1:0] = (1,0): Stream mode
4. FM[1:0] = (1,1): Stream-to-FIFO mode

The trigger used to activate Stream-to-FIFO mode is related to the IA bit value of the selected INT1_SRC_A register and does not depend on the interrupt pin value and polarity. The trigger is generated also if the selected interrupt is not driven to an interrupt pin.

The FTH[4:0] bits are intended to define the watermark level; when FIFO content exceeds this value, the WTM bit is set to “1” in the FIFO source register.

FIFO source register (0x2F)

This register is updated at every ODR and provides information about the FIFO buffer status.

Table 44. FIFO_SRC_REG_A

b7	b6	b5	b4	b3	b2	b1	b0
WTM	OVRN_FIFO	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0

- WTM bit is set high when FIFO content exceeds the watermark level.
- OVRN_FIFO bit is set high when FIFO buffer is full, this means that the FIFO buffer contains 32 unread samples. At the following ODR a new sample set replaces the oldest FIFO value. The OVRN_FIFO bit is reset when the first sample set has been read.
- EMPTY flag is set high when all FIFO samples have been read and FIFO is empty.
- FSS[4:0] field always contains the current number of unread samples stored in the FIFO buffer. When FIFO is enabled, this value increases at ODR frequency until the buffer is full, whereas, it decreases every time that one sample set is retrieved from FIFO.

The content of the register is updated synchronous to the FIFO write and read operation.

Table 45. FIFO_SRC_REG_A behavior assuming FTH[4:0] = 15

WTM	OVRN_FIFO	EMPTY	FSS[4:1]	Unread FIFO samples	Timing
0	0	1	00000	0	t0
0	0	0	00001	1	t0 + 1/ODR
0	0	0	00010	2	t0 + 2/ODR
...	...	...	...	...	...
0	0	0	01111	15	t0 + 15/ODR
1	0	0	10000	16	t0 + 16/ODR
...	...	...	...	...	...
1	0	0	11110	30	t0 + 30/ODR
1	0	0	11111	31	t0 + 31/ODR
1	1	0	11111	32	t0 + 32/ODR

The watermark flag and FIFO overrun event can be enabled to generate a dedicated interrupt on the INT1 pin by configuring CTRL_REG3_A.

Table 46. CTRL_REG3_A (0x22)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	I1_WTM	I1_OVERRUN	X

- I1_WTM bit drives watermark flag (WTM) on the INT1 pin.
- I1_OVERRUN bit drives overrun event (OVRN) on the INT1 pin.

If both bits are set to “1”, the INT1 pin status is the logical OR combination of the two signals.

4.8.3 FIFO modes

The LSM303AGR FIFO buffer can be configured to operate in four different modes selectable by the FM[1:0] field in FIFO_CTRL_REG_A. Available configurations ensure a high-level of flexibility and extend the number of functions usable in application development.

Bypass, FIFO, Stream, and Stream-to-FIFO modes are described in the following paragraphs.

Bypass mode

When Bypass mode is enabled, FIFO is not operational: the buffer content is cleared, output registers (0x28 to 0x2D) are frozen at the last value loaded, and the FIFO buffer remains empty until another mode is selected.

Follow these steps for Bypass mode configuration:

1. Turn on FIFO by setting the FIFO_En bit to “1” in control register 5 (0x24). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate Bypass mode by setting the FM[1:0] field to “00” in the FIFO control register (0x2E). If this mode is enabled, FIFO source register (0x2F) is forced equal to 0x20.

Bypass mode must be used in order to stop and reset the FIFO buffer when a different mode is operating. Note that placing the FIFO buffer into Bypass mode clears the whole buffer content.

FIFO mode

In FIFO mode, the buffer continues filling until full (32 sample sets stored,) then it stops collecting data and the FIFO content remains unchanged until a different mode is selected.

Follow these steps for FIFO mode configuration:

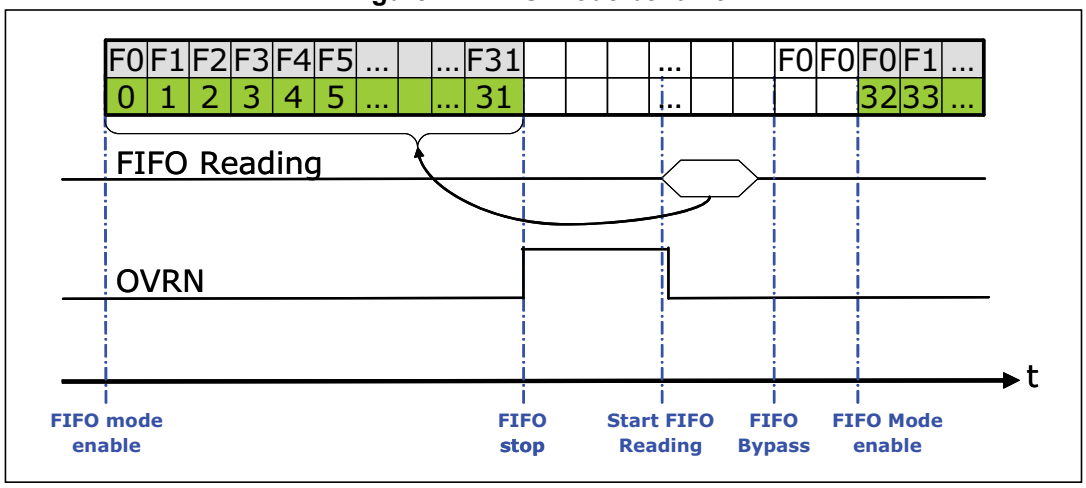
1. Turn on FIFO by setting the FIFO_EN bit to “1” in control register 5 (0x24). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate FIFO mode by setting the FM[1:0] field to “01” in the FIFO control register (0x2E).

By selecting this mode, FIFO starts data collection and source register (0x2F) changes according to the number of samples stored. At the end of the procedure, the source register is set to 0xDF and the OVRN flag generates an interrupt if the I1_OVERRUN bit is selected in control register 3. Data can be retrieved when OVRN_FIFO is set to “1”, performing a 32 sample set read from the output registers, data can be retrieved also on the WTM flag instead of OVRN if the application requires a lower number of samples. Communication speed is not so important in FIFO mode because data collection is stopped and there is no risk of overwriting acquired data. Before restarting FIFO mode, at the end of the reading procedure it is necessary to exit Bypass mode.

The recommended procedure is as follows:

1. Set FIFO_EN = 1: Enable FIFO
2. Set FM[1:0] = (0,1): Enable FIFO mode
3. Wait OVRN or WTM interrupt
4. Read data from the accelerometer output registers
5. Set FM[1:0] = (0,0): Enable Bypass mode
6. Repeat from point 2

Figure 27. FIFO mode behavior



If FIFO mode is enabled, the buffer starts to collect data and fill all the 32 slots (from F0 to F31) at the selected output data rate. When the buffer is full, the OVRN_FIFO bit goes high and data collection is permanently stopped; the user can decide to read FIFO content at any time because it is maintained unchanged until Bypass mode is selected. The reading procedure is composed of a 32 sample set of 6 bytes for a total of 192 bytes and retrieves data starting from the oldest sample stored in FIFO (F0). The OVRN_FIFO bit is reset when the first sample set has been read. The Bypass mode setting resets FIFO and allows the user to enable FIFO mode again.

Stream mode

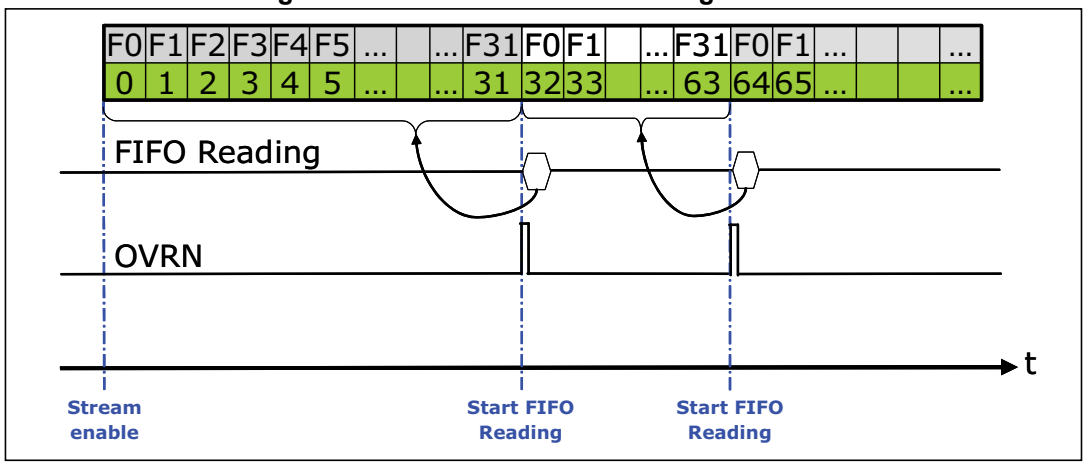
In Stream mode FIFO continues filling, when the buffer is full, the FIFO index restarts from the beginning and older data is replaced by the current. The oldest values continue to be overwritten until a read operation frees FIFO slots. Host processor reading speed is most important in order to free slots faster than new data is made available. FM[1:0] bypass configuration is used to stop this mode.

Follow these steps for FIFO mode configuration:

1. Turn on FIFO by setting the FIFO_EN bit to “1” in control register 5 (0x24). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
2. Activate stream mode by setting the FM[1:0] field to “10” in the FIFO control register (0x2E).

As described, for FIFO mode, data can be retrieved when OVRN_FIFO is set to”1” performing a 32 sample set reading from the output registers, data can be retrieved also on the WTM flag if the application requires a smaller number of samples.

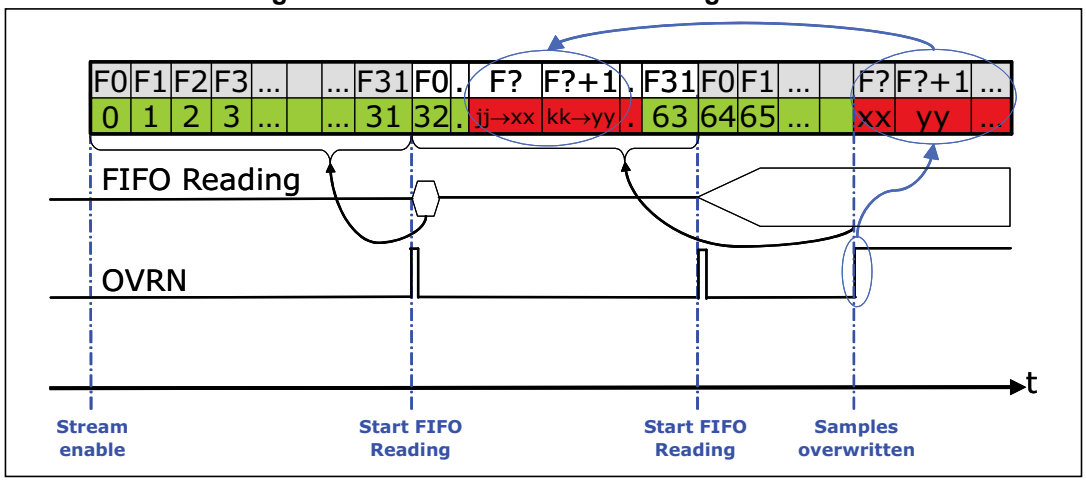
Figure 28. Stream mode fast reading behavior



In Stream mode, the FIFO buffer is continuously filling (from F0 to F31) at the selected output data rate. When the buffer is full the OVRN flag goes high and the recommendation is to read all FIFO samples (192 bytes) faster than 1*ODR, in order to free FIFO slots for the new acceleration samples. This allows avoiding loss of data and limiting intervention by the host processor, thus increasing system efficiency. If the reading procedure is not fast enough, three different cases can be observed:

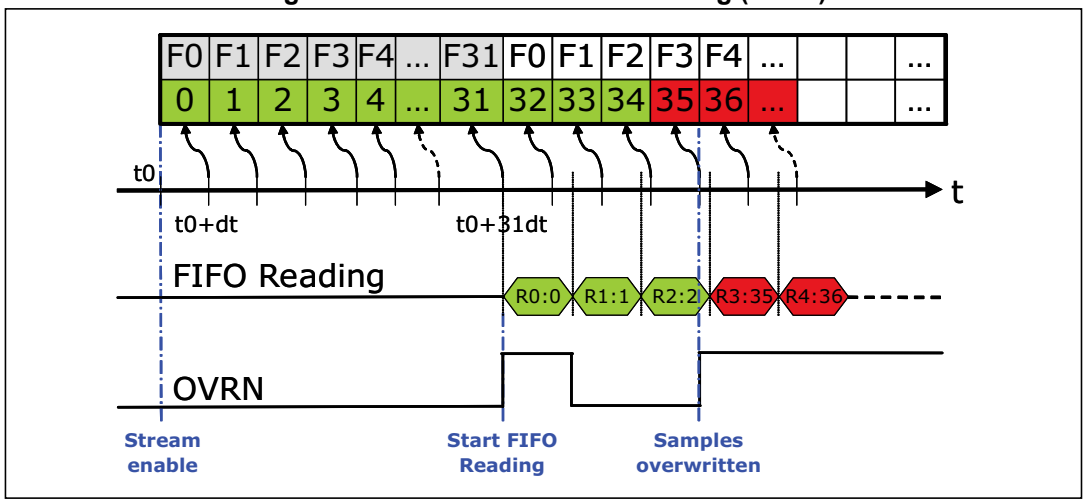
1. FIFO sample set (6 bytes) reading faster than 1*ODR: data are correctly retrieved because a free slot is made available before new data is generated.
2. FIFO sample set (6 bytes) reading synchronous to 1*ODR: data are correctly retrieved because a free slot is made available before new data is generated but FIFO benefits are not exploited. This case is equivalent to reading data on the data ready interrupt and does not limit the intervention by the host processor compared to standard accelerometer reading.
3. FIFO sample set (6 bytes) reading slower than 1*ODR: in this case some data is lost because data recovery is not fast enough to free slots for new acceleration data *Figure 29*. The number of correctly recovered samples is related to the difference between the current ODR and the FIFO sample set reading rate.

Figure 29. Stream mode slow reading behavior



In *Figure 29*, due to slow reading, data from “jj” are not retrieved because they are replaced by the new accelerometer samples generated by the system.

Figure 30. Stream mode slow reading (zoom)



After Stream mode enable, FIFO slots are filled at the end of each ODR time frame. The reading procedure must start as soon as the OVRN flag is set to “1”, data are retrieved from FIFO at the beginning of the reading operation. When a read command is sent to the device, the output registers content is moved to the SPI/I^2C register and the current oldest FIFO value is shifted into the output registers in order to allow the next read operation. In the case of a reading slower than 1*ODR, some data can be retrieved from FIFO after that new sample is inserted into the addressed location. In *Figure 30* the fourth read command starts after the refresh of the F3 index and this generates a disconnect in the data read. The OVRN flag advises the user that this event has taken place. In this example, three correct samples have been read, the number of correctly recovered samples is dependent on the difference between the current ODR and the FIFO sample set reading timeframe.

Stream-to-FIFO mode

This mode is a combination of the Stream and FIFO modes described previously. In Stream-to-FIFO mode, the FIFO buffer starts operating in Stream mode and switches to FIFO mode when the selected interrupt occurs.

Follow these steps for Stream-to-FIFO mode configuration:

1. Configure desired interrupt generator using register INT1_CFG_A (0x30).
2. Set the TR bit in the FIFO control register (0x2E) according to the configured interrupt generator: TR = “0” in order to select interrupt 1, TR = “1” in order to select interrupt 2.
3. Turn on FIFO by setting the FIFO_EN bit to “1” in control register 5 (0x24). After this operation the FIFO buffer is enabled but isn’t collecting data, output registers are frozen to the last samples set loaded.
4. Activate Stream-to-FIFO mode by setting the FM[1:0] field to “11” in the FIFO control register (0x2E).

The interrupt trigger is related to the IA bit in the INT1_SRC_A register and it is generated even if the interrupt signal is not driven to an interrupt pad. Switching modes is performed if both the IA and I1_OVERRUN bits are set high. Stream-to-FIFO mode is sensitive to the trigger level and not to the trigger edge, this means that if Stream-to-FIFO is in FIFO mode and the interrupt condition disappears, the FIFO buffer returns to Stream mode because the IA bit becomes zero. It is recommended to latch the interrupt signal used as the FIFO trigger in order to avoid losing interrupt events. If the selected interrupt is latched, the register INT1_SRC_A must be read to clear the IA bit; after reading, the IA bit takes 2*ODR to go low.

In Stream mode the FIFO buffer continues filling, when the buffer is full, the OVRN_FIFO bit is set high and the next samples overwrite the oldest. When a trigger occurs, two different cases can be observed:

1. If the FIFO buffer is already full (OVRN_FIFO = “1”), it stops collecting data at the first sample after trigger. FIFO content is composed of #30 samples collected before the trigger event, the sample that has generated the interrupt event and one sample after trigger.
2. If FIFO isn’t yet full (initial transient), it continues filling until it is full (OVRN_FIFO = “1”) and then, if the trigger is still present, it stops collecting data.

Figure 31. Stream-to-FIFO mode: interrupt not latched

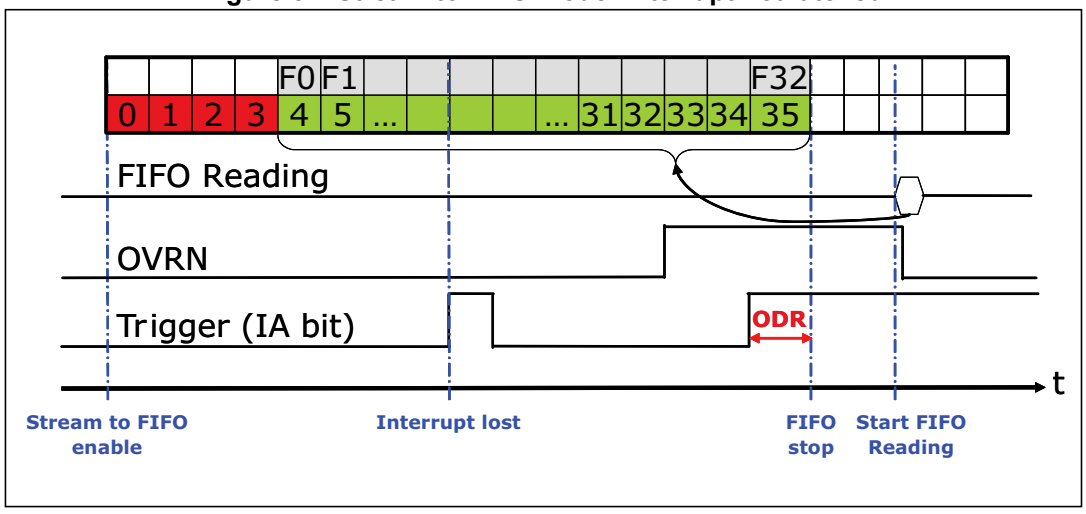
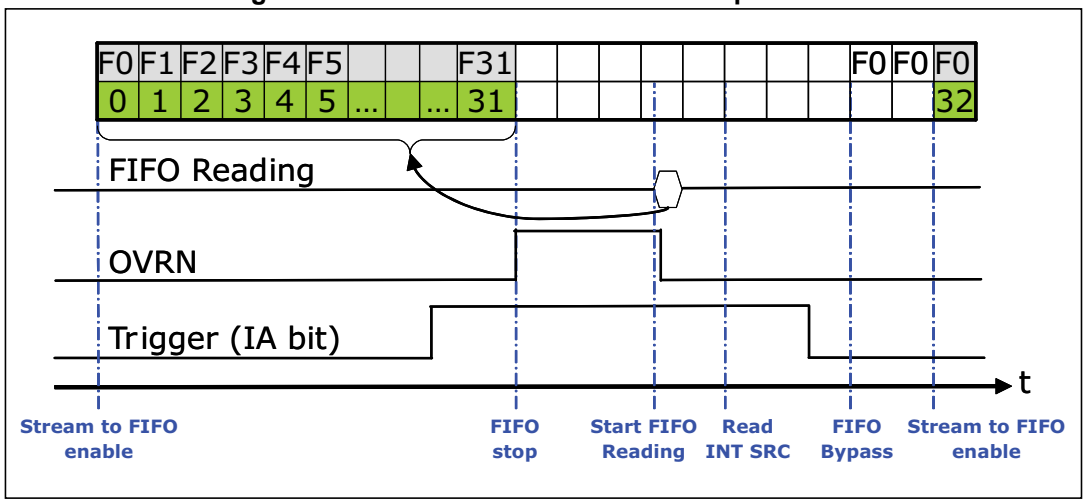


Figure 32. Stream-to-FIFO mode: interrupt latched



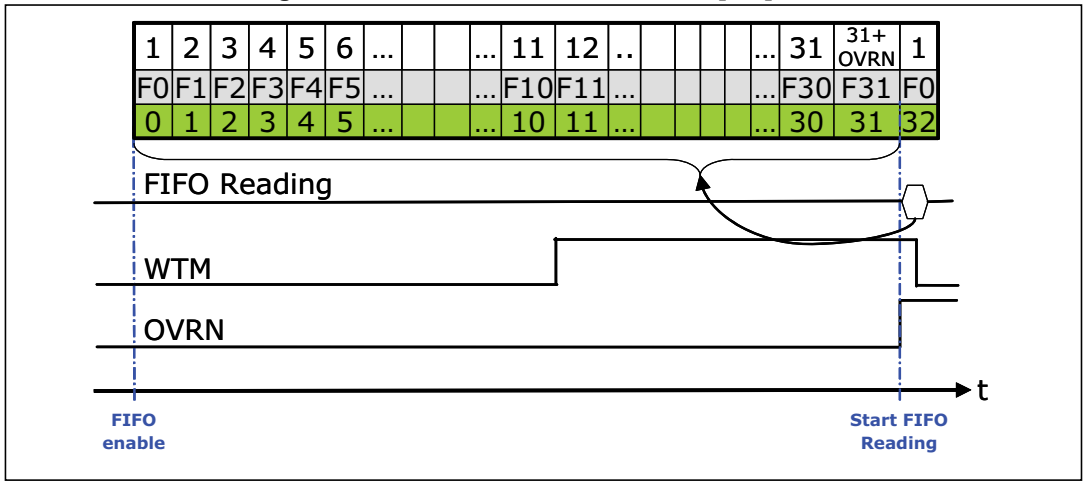
Stream-to-FIFO can be used in order to analyze the samples history that generate an interrupt; the standard operation is to read FIFO content when FIFO mode is triggered and the FIFO buffer is full and stopped.

4.8.4 Watermark

The watermark is a configurable flag that can be used to generate a specific interrupt in order to know when the FIFO buffer contains at least the number of samples defined as the watermark level. The user can select the desired level in a range from 0 to 31 using the FTH[4:0] field in the FIFO control register while the FIFO source register FSS[4:0] always contains the number of samples stored in FIFO.

If FSS[4:0] is greater than FTH[4:0], the WTM bit is set high in the FIFO source register, on the contrary, WTM is driven low when the FSS[4:0] field becomes lower than FTH[4:0]. FSS[4:0] increases by one step at the ODR frequency and decreases by one step every time that a sample set reading is performed by the user.

Figure 33. Watermark behavior - FTH[4:0] = 10



In *Figure 33*, the first row indicates the FSS[4:0] value, the second row indicates the relative FIFO slot and last row shows the incremental FIFO data. Assuming FTH[4:0] = 10, the WTM flag changes from “0” to “1” when the eleventh FIFO slot is filled (F10). *Figure 34* shows that

the WTM flag goes low when the FIFO content is less than FTH[4:0], which means that nine unread sample sets remain in FIFO. The watermark flag (WTM) can be enabled to generate a dedicated interrupt on the INT1 pin by setting the I1_WTM bit high in CTRL_REG3_A.

4.8.5 Retrieving data from FIFO

When FIFO is enabled and the mode is different to Bypass, reading output registers (28h to 2Dh) return the oldest FIFO sample set.

Whenever output registers are read, their content is moved to the SPI/I^2C output buffer. FIFO slots are ideally shifted up one level in order to release room for a new sample reception and output registers load the current oldest value stored in the FIFO buffer.

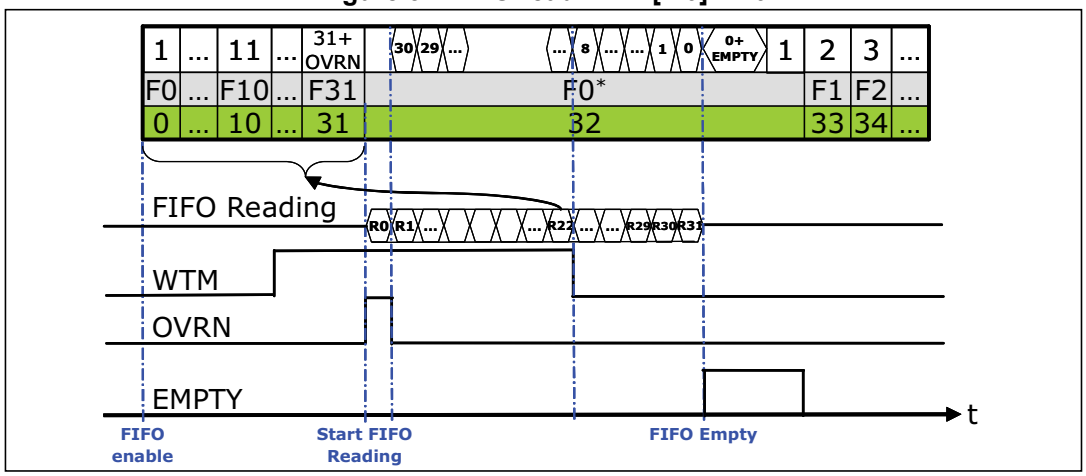
The entire FIFO content is retrieved by performing 32 read operations from the accelerometer output registers, every other reading operation returns the same last value until a new sample set is available in the FIFO buffer.

Data can be retrieved from FIFO using every reading byte combination in order to increase application flexibility (ex: 196 single byte reads, 32 reads of 6 bytes, 1 multiple read of 196 bytes, etc.).

It is recommended to read all FIFO slots in a multiple byte read of 196 bytes (6 output registers by 32 slots) faster than 1*ODR. In order to minimize communication between the master and slave the read address is automatically updated by the device; it rolls back to 0x28 when register 0x2D is reached.

In order to avoid losing data, the right ODR must be selected according to the serial communication rate available. In the case of standard I^2C mode being used (max rate 100 kHz), a single sample set read takes 830 µs while total FIFO download is about 17.57 ms. I^2C speed is slower than SPI and it needs about 29 clock pulses to start communication (Start, Slave Address, Device Address+Write, Restart, Device Address+Read) plus an additional 9 clock pulses for every byte to read. If this recommendation were followed, the complete FIFO read would be performed faster than 1*ODR, which means that using a standard I^2C, the selectable ODR must be lower than 57 Hz. If a fast I^2C mode is used (max rate 400 kHz), the selectable ODR must be lower than 228 Hz.

Figure 34. FIFO read - FTH[4:0] = 10



In *Figure 34* “Rx” indicates a 6-byte reading operation and “F0*” represents a single ODR slot expanded for visibility.

4.9 Temperature sensor

The is supplied with an internal temperature sensor. Temperature data can be enabled by setting the TEMP_EN[1:0] bits to ‘1’ in the *TEMP_CFG_REG_A (1Fh)* register.

To retrieve the temperature sensor data the BDU bit in *CTRL_REG4_A (23h)* must be set to ‘1’.

Both the *OUT_TEMP_L_A (0Ch), OUT_TEMP_H_A (0Dh)* registers must be read.

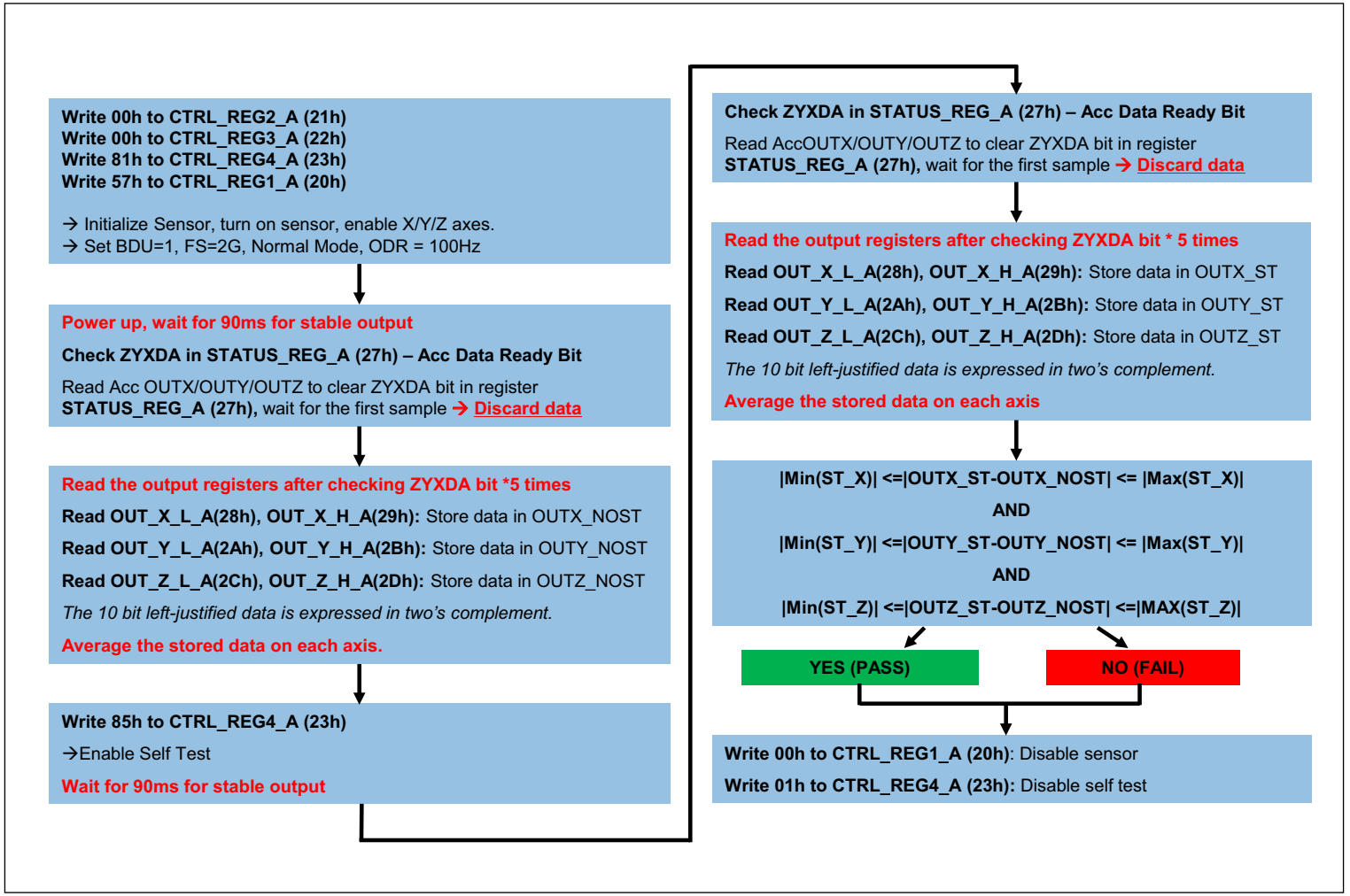
Temperature data is stored inside OUT_TEMP_H as two’s complement data in 8-bit format left-justified.

4.10 Accelerometer self-test

The self-test allows the user to check the sensor functionality without moving it. When the self-test is enabled, an actuation force is applied to the sensor, simulating a definite input acceleration. In this case the sensor outputs will exhibit a change in their DC levels which are related to the selected full scale through the device sensitivity. When the self-test is activated, the device output level is given by the algebraic sum of the signals produced by the acceleration acting on the sensor and by the electrostatic test-force. If the output signals change within the given range (the min and max values are provided in the datasheet), then the sensor is working properly and the parameters of the interface chip are within the defined specifications.

The self-test procedure is described in the following figure.

Figure 35. Accelerometer self-test procedure



5 Revision history

Table 47. Document revision history

Date	Revision	Changes
12-Apr-2016	1	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.