AN5127
Application note

ISM303DAC: high-performance, low-power, compact 3D accelerometer and 3D magnetometer module

Introduction

This document is intended to provide usage information and application hints related to ST's ISM303DAC device.

The ISM303DAC is an ultra-low-power high-performance system-in-package that combines a 3D high-performance digital linear accelerometer and magnetometer. The best-in-class heading accuracy, the high full-scale and data rate, and ST's 10-year longevity commitment make the ISM303DAC sensor especially suitable for industrial solutions.

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

The ISM303DAC can be configured to generate an interrupt signal for magnetic field detection and to automatically compensate for hard-iron offsets.

It can be also configured to generate interrupt signals by using hardware recognition of free-fall events, 6D orientation, tap and double-tap sensing, activity or inactivity, and wake-up events.

The integrated 256-level first-in first-out (FIFO) buffer allows the user to store accelerometer data in order to limit intervention by the host processor.

The ISM303DAC 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.

Contents

List of tables

List of figures

1 Pin description

Figure 1. Pin connections

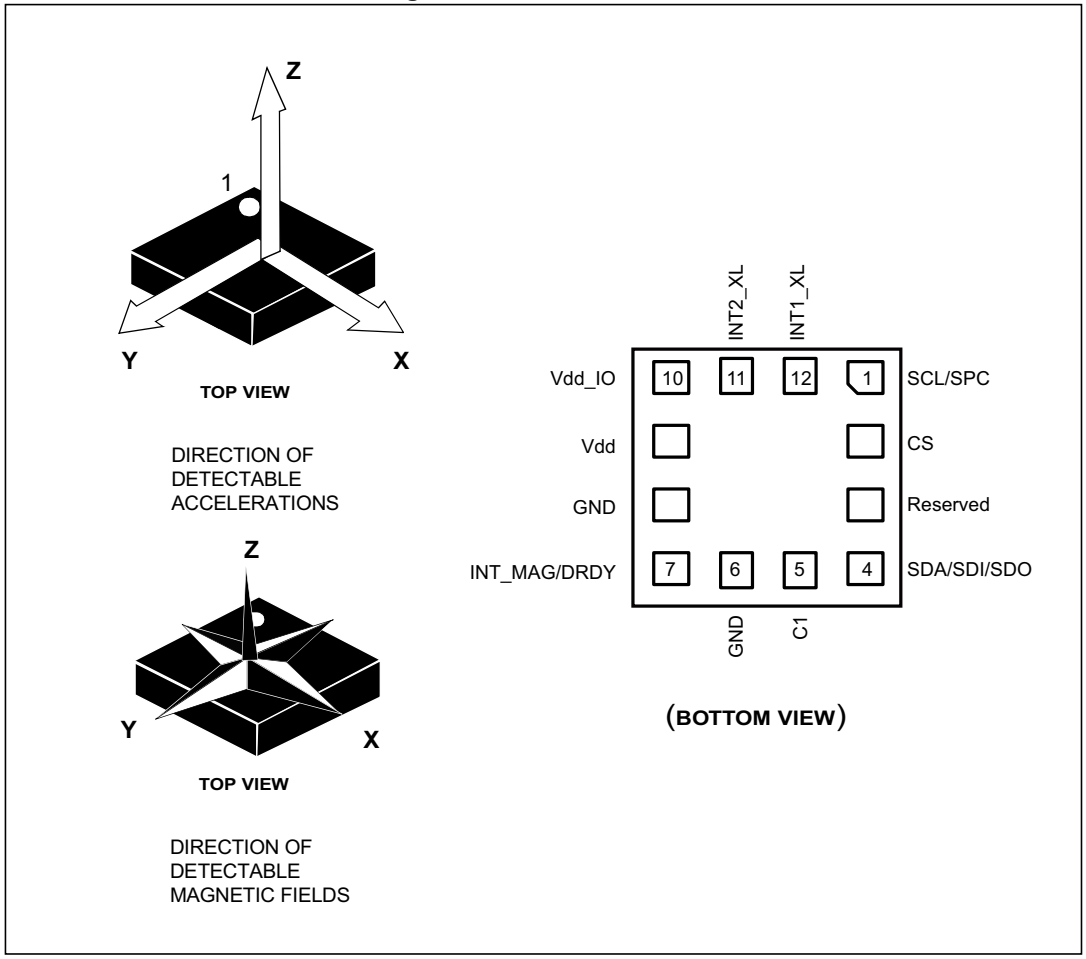


Table 1. Pin description

Pin #	Name	Function	Pin status
1	SCL SPC	I²C serial clock (SCL) SPI serial port clock (SPC)	Default: input without pull-up
2	CS	I²C/SPI mode selection 1: SPI idle mode / I²C communication enabled; 0: SPI communication mode / I²C disabled	Default: input without pull-up
3	Reserved	Connect to GND	
4	SDA SDI SDO	I^{2}C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	Default: (SDA) input without pull-up
5	C1	Capacitor connection (C1 = 220 nF)	External capacitor, voltage forced by the device
6	GND	0 V supply	
7	INT_MAG/ DRDY	Magnetometer interrupt/data-ready signal	Default: output high impedance
8	GND	0 V supply	
9	Vdd	Power supply	
10	Vdd_IO	Power supply for I/O pins	
11	INT2_XL	Accelerometer interrupt pin 2	Default: push-pull output forced to GND
12	INT1_XL	Accelerometer interrupt pin 1	Default: push-pull output forced to GND

1.1 INT_MAG/DRDY pin configuration

The INT_MAG/DRDY pin can be configured to have one HW signal to determine when a new set of measurement data is available for reading or when an interrupt event occurs.

The Zyxda data-ready signal in the STATUS_REG_M register can be driven to the INT_MAG/DRDY pin by setting the INT_MAG bit in the CFG_REG_C_M register to 1 (see *Section 6.3: Using the data-ready signal* for further details).

The INT signal in the INT_SOURCE_REG_M register can be driven to the INT_MAG/DRDY pin by setting the INT_MAG_PIN bit in CFG_REG_C_M to 1 (see *Section 10: Magnetometer interrupt generation* for further details).

Note: Both the INT_MAG *and* INT_MAG_PIN *bits in CFG_REG_C_M configure the INT_MAG/DRDY pin as a digital output, but only one signal (INT or DRDY) can be routed on the INT_MAG/DRDY pin. If both bits are asserted, only the INT signal is routed as shown in following table.*

The table below summarizes the INT_MAG/DRDY pin status and functionality in different configurations of the INT_MAG and INT_MAG_PIN bits of the CFG_REG_C_M register.

Table 2. INT_MAG/DRDY pin configuration

INT_MAG	INT_MAG_PIN	INT_MAG/DRDY pin status	INT_MAG/DRDY pin function
0	0	High impedance output	None
1	0	Push-pull output	DRDY (Zyxda) signal routed
0	1	Push-pull output	INT signal routed
1	1	Push-pull output	INT signal routed

2 Registers

Table 3. Registers

Register name	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Module_8bit_A	0Ch	Module_7	Module_6	Module_5	Module_4	Module_3	Module_2	Module_1	Module_0
WHO_AM_I_A	0Fh	0	1	0	0	0	0	1	1
CTRL1_A	20h	ODR3	ODR2	ODR1	ODR0	FS1	FS0	HF_ODR	BDU
CTRL2_A	21h	BOOT	SOFT_ RESET	0	0	FDS_SLOPE	IF_ADD_INC	I2C_ DISABLE	SPI_ENABLE
CTRL3_A	22h	ST2	ST1	TAP_X_EN	TAP_Y_EN	TAP_Z_EN	LIR	H_LACTIVE	PP_OD
CTRL4_A	23h	0	INT1_S_TAP	INT1_WU	INT1_FF	INT1_TAP	INT1_6D	INT1_FTH	INT1_DRDY
CTRL5_A	24h	DRDY _PULSED	INT2_BOOT	INT2_ON _INT1	0	0	0	INT2_FTH	INT2_DRDY
FIFO_CTRL_A	25h	FMODE2	FMODE1	FMODE0	0	MODULE_ TO_FIFO	0	0	IF_CS_PU _DIS
OUT_T_A	26h	TEMP7	TEMP6	TEMP5	TEMP4	TEMP3	TEMP2	TEMP1	TEMP0
STATUS_A	27h	FIFO_THS	WU_IA	SLEEP _STATE	DOUBLE _TAP	SINGLE _TAP	6D_IA	FF_IA	DRDY
OUTX_L_A	28h	X_L7	X_L6	X_L5	X_L4	X_L3	X_L2	0	0
OUTX_H_A	29h	X_H7	X_H6	X_H5	X_H4	X_H3	X_H2	X_H1	X_H0
OUTY_L_A	2Ah	Y_L7	Y_L6	Y_L5	Y_L4	Y_L3	Y_L2	0	0
OUTY_H_A	2Bh	Y_H7	Y_H6	Y_H5	Y_H4	Y_H3	Y_H2	Y_H1	Y_H0
OUTZ_L_A	2Ch	Z_L7	Z_L6	Z_L5	Z_L4	Z_L3	Z_L2	0	0
OUTZ_H_A	2Dh	Z_H7	ZH_6	Z_H5	ZH_4	Z_H3	Z_H2	Z_H1	Z_H0
FIFO_THS_A	2Eh	FTH7	FTH6	FTH5	FTH4	FTH3	FTH2	FTH1	FTH0
FIFO_SRC_A	2Fh	FTH	FIFO_OVR	DIFF8	0	0	0	0	0
FIFO_SAMPLES_A	30h	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
TAP_6D_THS_A	31h	4D_EN	6D_THS1	6D_THS0	TAP_THS4	TAP_THS3	TAP_THS2	TAP_THS1	TAP_THS0

Table 3. Registers

Register name	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INT_DUR_A	32h	LAT3	LAT2	LAT1	LAT0	QUIET1	QUIET0	SHOCK1	SHOCK0
WAKE_UP_THS_A	33h	SINGLE_ DOUBLE_TAP	SLEEP_ON	WU_THS5	WU_THS4	WU_THS3	WU_THS2	WU_THS1	WU_THS0
WAKE_UP_DUR_A	34h	FF_DUR5	WU_DUR1	WU_DUR0	INT1_FSS7	SLEEP_ DUR3	SLEEP_ DUR2	SLEEP_ DUR1	SLEEP_ DUR0
FREE_FALL_A	35h	FF_DUR4	FF_DUR3	FF_DUR2	FF_DUR1	FF_DUR0	FF_THS2	FF_THS1	FF_THS0
STATUS_DUP_A	36h	OVR	WU_IA	SLEEP _STATE	DOUBLE _TAP	SINGLE_ TAP	6D_IA	FF_IA	DRDY
WAKE_UP_SRC_A	37h	0	0	FF_IA	SLEEP_ STATE_IA	WU_IA	X_WU	Y_WU	Z_WU
TAP_SRC_A	38h	0	TAP_IA	SINGLE _TAP	DOUBLE _TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP
6D_SRC_A	39h	0	6D_IA	ZH	ZL	YH	YL	XH	XL
FUNC_SRC_A	3Eh	0	0	0	0	0	0	MODULE_ READY	0
FUNC_CTRL_A	3Fh	0	0	MODULE_ ON	0	0	0	0	0
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
WHO_AM_I_M	4Fh	0	1	0	0	0	0	0	0

Table 3. Registers

Register name	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CFG_REG_A_M	60h	COMP_ TEMP_EN(1)	REBOOT	SOFT_RST	LP	ODR1	ODR0	MD1	MD0
CFG_REG_B_M	61h	0	0	0	OFF_CANC_ ONE_SHOT	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

1. For proper operation of the magnetometer sensor, the COMP_TEMP_EN bit in the CFG_REG_A_M register must be set to 1 by the user.

3 Magnetometer operating modes

The ISM303DAC magnetometer provides three operating modes:

- Idle mode;
- Continuous mode;
- Single mode.

After the power supply is applied, the magnetometer performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed, the magnetometer is automatically configured in Idle mode.

In order to avoid potential conflicts, during the power-on sequence it is recommended to set the lines connected to the device IO pins to high-impedance state on the host side. Furthermore, to guarantee proper power-off of the device it is recommended to maintain the duration of the Vdd line to GND for at least 100 µs.

The operating modes of the magnetometer can be set through the MD[1:0] bits of CFG_REG_A_M as shown in table below.

Table 4. Operative modes

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

In all three operating modes, the typical value of the magnetic dynamic range is 50 gauss which applies when the magnetic field is fully aligned to one of the sensitive axes. In presence of a stray field in the cross-axis direction, the magnetic dynamic range can decrease down to 25 gauss (in the worst case).

3.1 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 (I²C 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.

3.2 Continuous mode

Continuous mode can be enabled by writing the MD[1:0] bits to 00 in CFG_REG_A_M register.

In Continuous mode the device continuously performs measurements and places the result in the output data registers. Either High-Resolution or Low-Power mode can be selected by configuring the LP bit in CFG_REG_A_M (please refer to *Section 4: Magnetometer power modes*).

In Continuous mode the output data rate can be selected using the ODR[1:0] bits of CFG_REG_A_M register as shown in table below.

Table 5. Output data rate configuration

ODR1	ODR0	ODR (Hz)
0	0	10 (default)
0	1	20
1	0	50
1	1	100

3.3 Single mode

The ISM303DAC magnetometer offers Single mode in both High-Resolution and Low-Power modes (please refer to *Section 4: Magnetometer power modes*).

Single mode configuration allows performing a single acquisition upon request; the acquisition is triggered by writing the MD[1:0] bits to 01 in the CFG_REG_A_M register. Once the measurement has been performed, the Zyxda, zda, yda, xda bits of the STATUS_REG_M register are asserted, data are available in the output registers and the magnetometer is automatically configured in Idle mode by setting the MD[1:0] bits to 11.

Single mode is independent of the programmed ODR: it 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 is given in the following table and strictly depends on the power mode selected (please refer to *Section 4: Magnetometer power modes*).

Table 6. Maximum ODR in single mode

Power mode (LP bit of CFG_REG_A_M register)	Maximum ODR [Hz]
High-Resolution (LP = 0)	100
Low-Power (LP = 1)	150

In Single mode, the typical time needed for the generation of the new data corresponds to the turn-on time indicated in *Table 8: Operating mode and turn-on time*.

4 Magnetometer power modes

The ISM303DAC magnetometer provides two power modes, operating with the device configured either in Continuous or Single mode:

- High-Resolution mode;
- Low-Power mode.

The power mode can be selected by configuring the LP bit of CFG_REG_A_M register: if the LP bit is asserted, the device operates in Low-Power mode, otherwise it operates in High-Resolution mode (default configuration).

In both Low-Power mode and High-Resolution mode, the magnetometer circuitry is periodically turned on/off with a duty cycle that is a function of the selected ODR and data interrupt generation is active.

The difference distinguishing the two modes is in the number of samples used to generate each output sample, which is four times less in Low-Power mode than the number used in High-Resolution mode, thus ensuring a lower power consumption.

The table below summarizes the current consumption of the two power modes with offset cancellation disabled/enabled (the sensor offset cancellation feature can be configured using the OFF_CANC bit of CFG_REG_B_M register - please refer to *Section 8: Magnetometer offset cancellation*).

Table 7. Current consumption

ODR [Hz]	Current consumption (LP = 0 and OFF_CANC = 0) [uA]	Current consumption (LP = 1 and OFF_CANC = 0) [uA]	Current consumption (LP = 0 and OFF_CANC = 1) [uA]	Current consumption (LP = 1 and OFF_CANC = 1) [uA]
10	100	25	120	50
20	200	50	235	100
50	475	125	575	235
100	950	250	1130	460

The following table summarizes the turn-on time of the device in the two different power modes with the offset cancellation function enabled or disabled (see *Section 8: Magnetometer offset cancellation*).

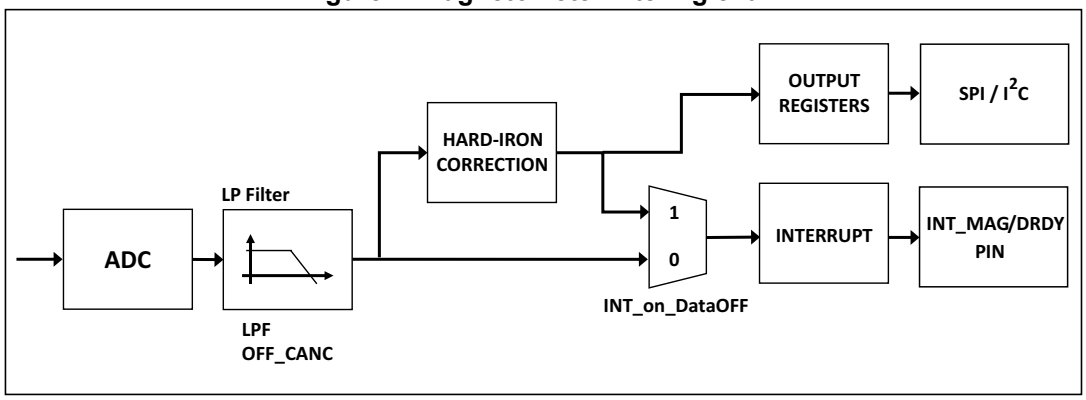
Table 8. Operating mode and turn-on time

Operating mode	Turn-on time	
CFG_REG_A_M[LP]	CFG_REG_A_M[OFF_CANC = 0]	CFG_REG_A_M[OFF_CANC = 1]
0 (High-Resolution)	9.4 ms	9.4 ms + 1/ODR
1 (Low-Power)	6.4 ms	6.4 ms + 1/ODR

5 Magnetometer low-pass filter

The ISM303DAC magnetometer embeds a digital low-pass filter in order to reduce noise. The filter can be enabled by setting the LPF bit in CFG_REG_B_M.

Figure 2. Magnetometer filtering chain



The table below summarizes the bandwidth and RMS noise values in different device configurations.

When the low-pass filter is enabled, the bandwidth is reduced while noise performance is improved without any increase in power consumption.

Note: The offset cancellation feature (please refer to Section 8: Magnetometer offset cancellation) works as a two-level moving average filter, thus giving the same bandwidth and noise performance as with the LPF filter enabled.

Table 9. RMS noise of operating modes

LPF or OFF_CANC	LP = 0		LP = 1	
	BW [Hz]	Noise RMS [mg]	BW [Hz]	Noise RMS [mg]
0 (disable)	ODR/2	4.5	ODR/2	9
1 (enable)	ODR/4	3	ODR/4	6

6 Reading magnetometer output data

6.1 Startup sequence

Once the device is powered up, the magnetometer automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed (it takes 20 ms), the magnetometer automatically enters Idle 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 = 80h	// Temperature compensation enabled
	// ODR = 10 Hz
	// Continuous and High-Resolution modes

2. Write CFG_REG_C_M = 01h	// Configure INT_MAG/DRDY pin as digital output and route data-ready signal

Writing 81h in CFG_REG_A_M instead of 80h will set the magnetometer to operate in Single mode instead of Continuous mode.

6.2 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 (Zyxda = 1).

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

6.3 Using the data-ready signal

The magnetometer 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 (DRDY) is represented by the Zyxda bit of the STATUS_REG_M register. This signal can be driven to the INT_MAG/DRDY pin by setting the INT_MAG bit of the CFG_REG_C_M register to 1.

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 (OUTX_H_REG_M, OUTY_H_REG_M and OUTZ_H_REG_M registers).

6.4 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 data-ready 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.

6.5 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 mG / LSB, in order to obtain the corresponding value in mG.

6.5.1 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_REG_M & OUTZ_L_REG_M: FECBh
```

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
```

6.5.2 Big-little endian selection

The ISM303DAC magnetometer allows swapping the content of the lower and the upper part of the output data registers (i.e. OUTX_H_REG_M with OUTX_L_REG_M) in order 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. This mode corresponds to the BLE bit of the CFG_REG_C_M register set 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. This mode corresponds to the BLE bit of the CFG_REG_C_M register set to 1.

7 Magnetometer reboot and software reset

After the device is powered up, the ISM303DAC magnetometer performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed, the magnetometer is automatically configured in Idle mode.

During the boot time the registers are not accessible.

After power-up, the trimming parameters can be re-loaded by setting the REBOOT bit of the CFG_REG_A_M register to 1.

No toggle of the device power lines is required; after the reboot is completed, the device enters in Idle mode (regardless of the selected operating mode) after performing one measurement.

If the reset to the default value of the control registers is required, it can be performed by setting the SOFT_RST bit of the CFG_REG_A_M register to 1. The software reset procedure can take 5 µs; the status of the reset is signaled by the status of the SOFT_RST bit of the CFG_REG_A_M register: once the reset is completed, this bit is automatically set low.

In order to avoid conflicts, the reboot and the software reset must not be executed at the same time (do not set to 1 at the same time both the REBOOT bit and SOFT_RST bit of the CFG_REG_A_M register).

The flow must be performed serially as shown in the example below:

1. Set the SOFT_RST bit of the CFG_REG_A_M register to 1;
2. Wait 5 µs (or wait until the SOFT_RST bit of the CFG_REG_A_M register returns to 0);
3. Set the REBOOT bit of the CFG_REG_A_M register to 1;
4. Wait 20 ms.

8 Magnetometer offset cancellation

The ISM303DAC magnetometer is based on AMR technology: a set pulse is needed to set an initial operating condition.

Offset cancellation is the result of performing a set and reset pulse in the magnetic sensor and it can be enabled to remove the intrinsic sensor offset.

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:

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

If the magnetometer is operating in Continuous mode, the offset cancellation is enabled by setting the OFF_CANC bit to 1 in CFG_REG_B_M. In this case, set/reset pulses are continuously performed; a set pulse is applied to one measurement, a reset pulse is applied to the next measurement. If the offset cancellation is disabled (OFF_CANC = 0) and Continuous mode is selected, 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 from Idle mode (a set of the magnetic sensor is performed anyway, even if the offset cancellation is disabled).

If the magnetometer is operating in Single mode, in order to enable the offset cancellation, both OFF_CANC and OFF_CANC_ONE_SHOT bits must be set to 1 in CFG_REG_B_M. Enabling these bits, the impulse polarity is inverted between a single read and the next one. While offset cancellation is automatically managed by the device in Continuous mode, if this feature is enabled in Single mode, 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.

9 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 address 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_X_REG_H_M, OFFSET_Y_REG_H_M and OFFSET_Z_REG_H_M registers should contain the MSBs of the hard-iron distortion field estimated along the X, Y and Z axes respectively. The OFFSET_X_REG_L_M, OFFSET_Y_REG_L_M and OFFSET_Z_REG_L_M registers should contain the LSBs of the hard-iron distortion field estimated along the X, Y and Z axes respectively. Hard-iron data have the same format and sensitivity of the magnetic output data. The hard-iron values stored in dedicated registers are automatically subtracted from the output data.

10 Magnetometer interrupt generation

In the ISM303DAC, the magnetometer interrupt signal generation is based on the comparison between the magnetometer output data and the programmable threshold.

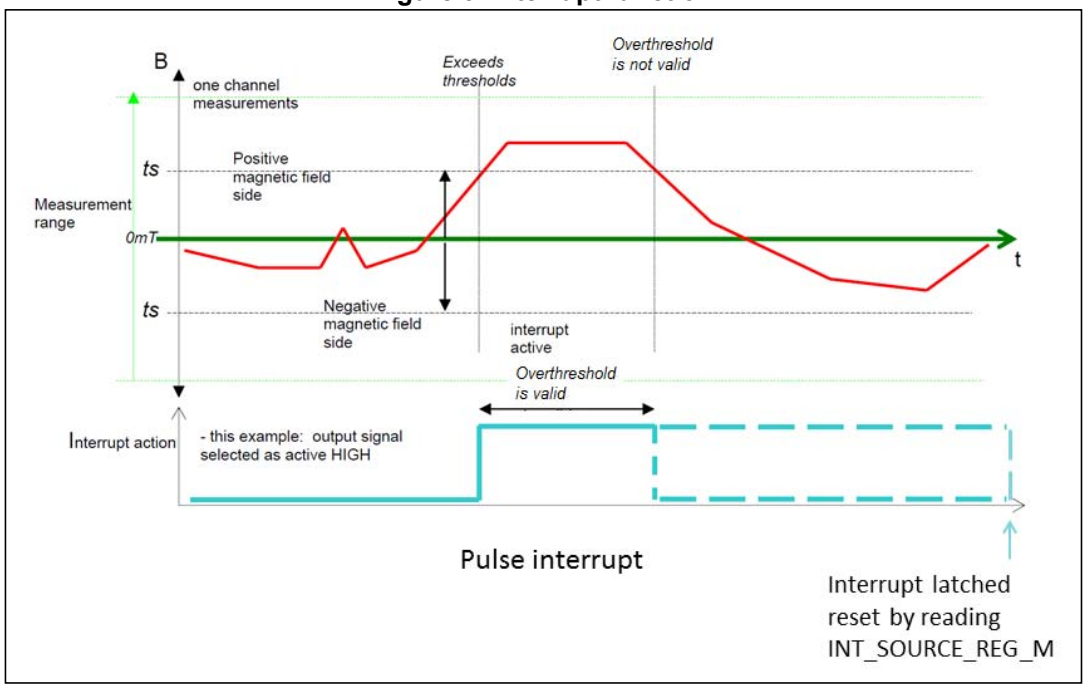
To enable the interrupt function, the IEN bit in INT_CTRL_REG_M must be set to 1. The interrupt function can be selectively enabled on each axis. In order to do this, the XIEN, YIEN, and ZIEN bits in INT_CTRL_REG_M need be set properly.

The threshold value can be programmed by setting the INT_THS_L_REG_M and INT_THS_H_REG_M registers.

The threshold is expressed in absolute value as a 15-bit unsigned number. The threshold has the same sensitivity as the magnetic data.

When magnetic data exceeds the positive or the negative threshold, an interrupt signal is generated and the INT bit in the INT_SOURCE_REG_M register goes high. Information about which axis has triggered the wake-up event is also available in the INT_SOURCE_REG_M register; in particular, when magnetic data exceeds the positive threshold the P_TH_S_[X, Y, Z] bit is set to 1, while if data exceeds the negative threshold the N_TH_S_[X, Y, Z] bit is set to 1. If magnetic data lay between the positive and the negative thresholds, no interrupt signal is generated.

Figure 3. Interrupt function



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.

The hardware interrupt signal can be either pulsed or latched:

- Pulsed interrupt signal: it goes to active level when the magnetic data exceeds one of the two thresholds and goes low when the magnetic data are between the two thresholds (positive or negative). This kind of interrupt is selected by setting the IEL bit in INT_CTRL_REG_M register to 0.
- Latched interrupt signal: it goes to active level when the data exceed one of the two thresholds but is reset only once the source register is read and not when the magnetic data returns between the two thresholds. This kind of interrupt is selected by setting the IEL bit in INT_CTRL_REG_M register to 1.

The interrupt signal polarity can be set using the IEA bit in INT_CTRL_REG_M.

If IEA is set to 1, then the interrupt signal is active high, while if it is set to 0, the interrupt signal is active low.

10.1 Interrupt configuration example

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

1.	Write 80h in CFG_REG_A_M	// Temperature compensation enabled
		// ODR = 10 Hz
		// Continuous mode and High-Resolution

2.	Write 40h in CFG_REG_C_M	// Configure INT_MAG/DRDY pin as digital output and route interrupt signal
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.

10.2 Overflow interrupt

The MROI bit in INT_SOURCE_REG_M alerts the user if a measurement range overflow has occurred at internal ADC level. This function is enabled only if the interrupt generator is active (IEN bit = 1). MROI behavior is always latched: once the internal measurement range overflow has occurred, the MROI bit is reset by reading INT_SOURCE_REG_M.

11 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 to 1 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.

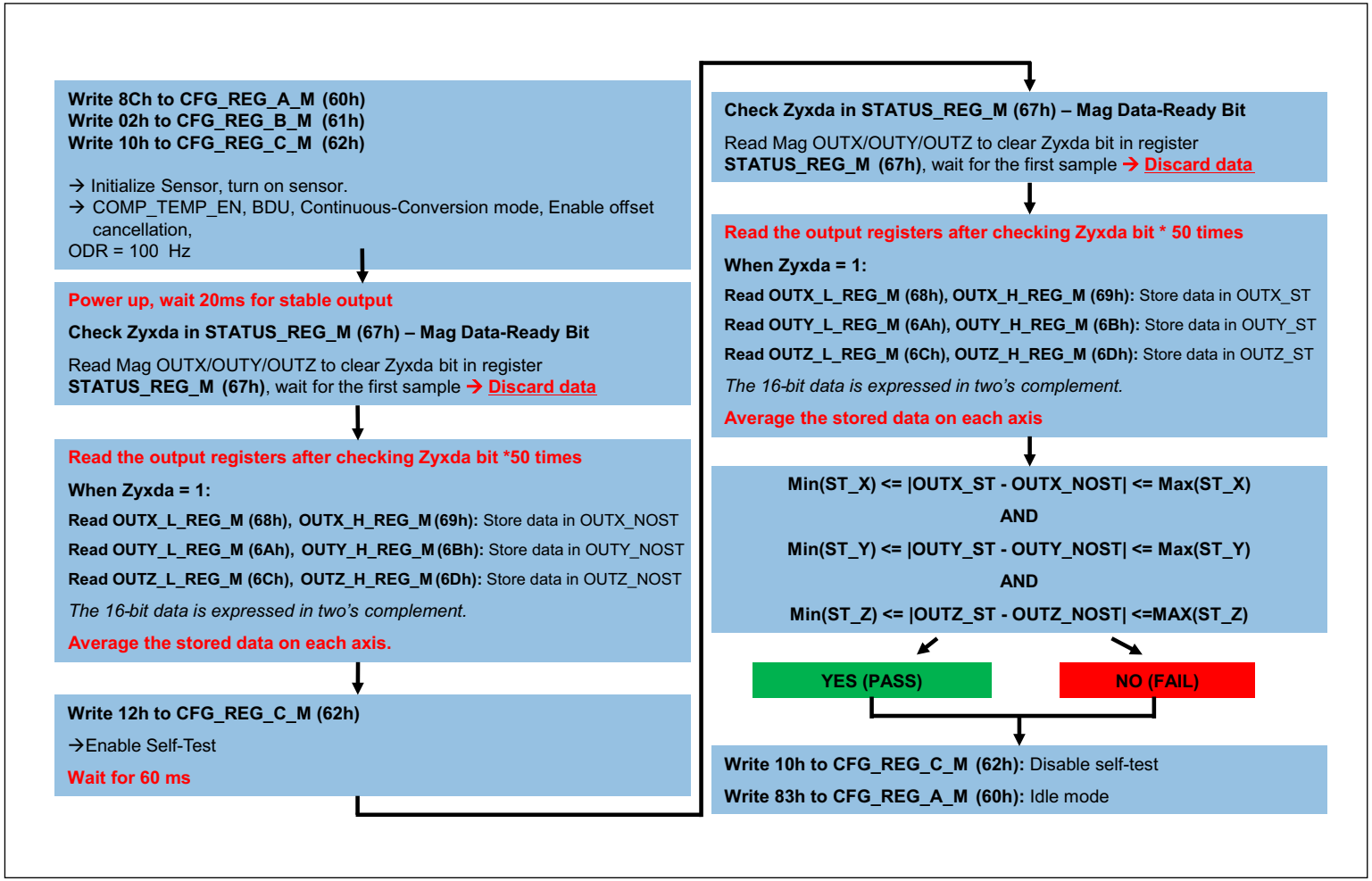
The procedure consists of:

1. enabling the magnetometer;
2. averaging fifty samples before enabling the self-test;
3. averaging fifty 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 4*.

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

Figure 4. Magnetometer self-test procedure



12 Accelerometer operating modes

The ISM303DAC accelerometer provides two power modes: high-resolution (HR)/high-frequency mode (HF) and low-power (LP) mode.

After the power supply is applied, the accelerometer performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed, the accelerometer is automatically configured in power-down mode.

Referring to the ISM303DAC datasheet, the output data rate (ODR) and high-frequency (HF_ODR) bits of CTRL1_A register are used to select the power mode and the output data rate of the accelerometer sensor (*Table 10: Accelerometer ODR and power mode selection*).

Table 10. Accelerometer ODR and power mode selection

ODR [3:0]	HF_ODR	Mode	ODR selection [Hz]	Resolution (bit number)
0000	0	Power-down	Power-down	-
1000	0	LP	1	10
1001	0	LP	12.5	10
1010	0	LP	25	10
1011	0	LP	50	10
1100	0	LP	100	10
1101	0	LP	200	10
1110	0	LP	400	10
1111	0	LP	800	10
0001	0	HR	12.5	14
0010	0	HR	25	14
0011	0	HR	50	14
0100	0	HR	100	14
0101	0	HR	200	14
0110	0	HR	400	14
0111	0	HR	800	14
0101	1	HF	1600	12
0110	1	HF	3200	12
0111	1	HF	6400	12

The output data have different resolution and are left-aligned. For example in case of the 10-bit resolution the output data are the 10 most significant bits of OUT_H_A & OUT_L_A concatenation, and the raw value has to be right-shifted by 6.

Table 11 shows the typical values of the ISM303DAC accelerometer power consumption for the different operating modes.

Table 11. Power consumption

ODR [Hz]	HR/HF (µA)	LP (µA)
1	162	4.5
12.5	162	6
25	162	8
50	162	10
100	162	16
200	162	28
400	162	52
800	162	100
1600	162	-
3200	162	-
6400	162	-

12.1 Power-down

When the accelerometer is in power-down, almost all internal blocks of the device are switched off to minimize power consumption. Digital interfaces (I²C 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 power-down.

12.2 High-resolution/high-frequency mode

In HR/HF mode, all accelerometer circuitry is always on and data are generated at the data rate selected through the ODR bits. Data interrupt generation is active.

HR mode works with a 14-bit resolution, while HF with a 12-bit resolution (see *Table 10: Accelerometer ODR and power mode selection*).

12.3 Low-power mode

In low-power mode the accelerometer circuitry is periodically turned on/off with a duty cycle that is a function of the selected ODR. This mode differs from HR/HF mode in the available output data rates. In low-power mode we have same data rates as HR mode (from 12.5 Hz to 800 Hz, but with a lower consumption) plus the 1 Hz case.

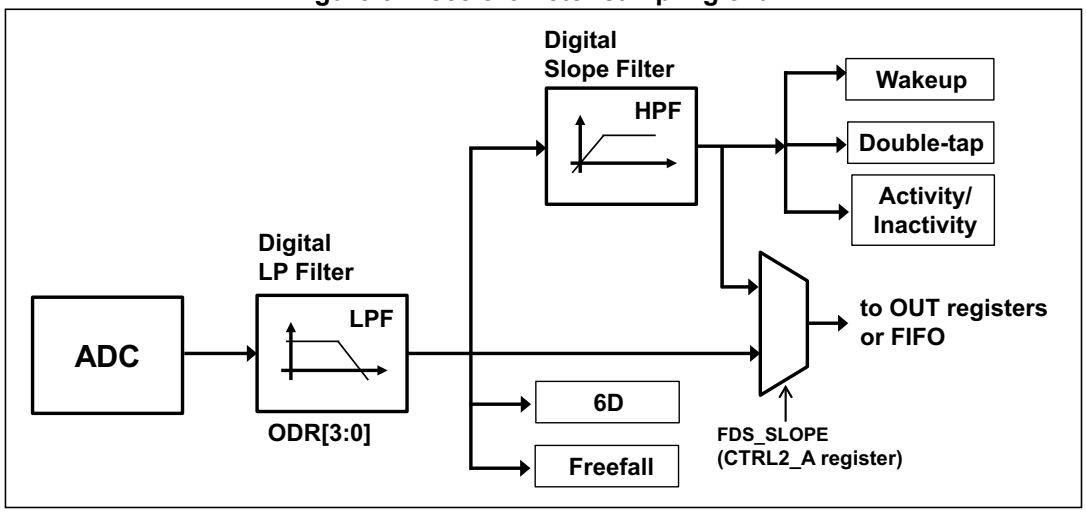
Data interrupt generation is active.

LP mode works with a 10-bit resolution (see *Table 10: Accelerometer ODR and power mode selection*).

12.4 Accelerometer bandwidth

The accelerometer sampling chain (*Figure 5*) is represented by a cascade of a few blocks: an ADC converter, a digital low-pass filter and a digital slope filter.

Figure 5. Accelerometer sampling chain



The digital signal is filtered by a low-pass digital filter (LPF) whose cutoff frequency depends on the selected accelerometer ODR, as shown in *Table 12*.

Table 12. Accelerometer LPF1 cutoff frequency

Mode	ODR selection [Hz]	LPF cutoff [Hz]
LP	1	3200
LP	12.5	3200
LP	25	3200
LP	50	3200
LP	100	3200
LP	200	3200
LP	400	3200
LP	800	3200
HR	12.5	5.5
HR	25	11
HR	50	22
HR	100	44
HR	200	88
HR	400	177
HR	800	355
HF	1600	710
HF	3200	1420
HF	6400	2840

The selection of the signal (LPF or HPF) which is sent to the OUT registers is determined by the FDS_SLOPE bit of CTRL2_A register. When it is logic '1', the HPF signal is selected, LPF otherwise.

The signal that is sent to the digital functions (wakeup, double-tap, activity/inactivity and 6D orientation) is always the HPF signal. Anti-aliasing filtering is guaranteed by the ADC sampling frequency and the digital LPF cutoff frequency. Anti-aliasing filtering is available in HR/HF mode only. When the accelerometer is in LP mode, the circuitry is periodically turned on/off (reducing power consumption) with a fixed on-time and a period that is a function of the selected ODR. For this reason the LPF cutoff is fixed to 3.2 kHz, so the user must take care to select the proper ODR value Vs application sampling frequency in order to avoid aliasing (based on the noise characteristics of the system in use).

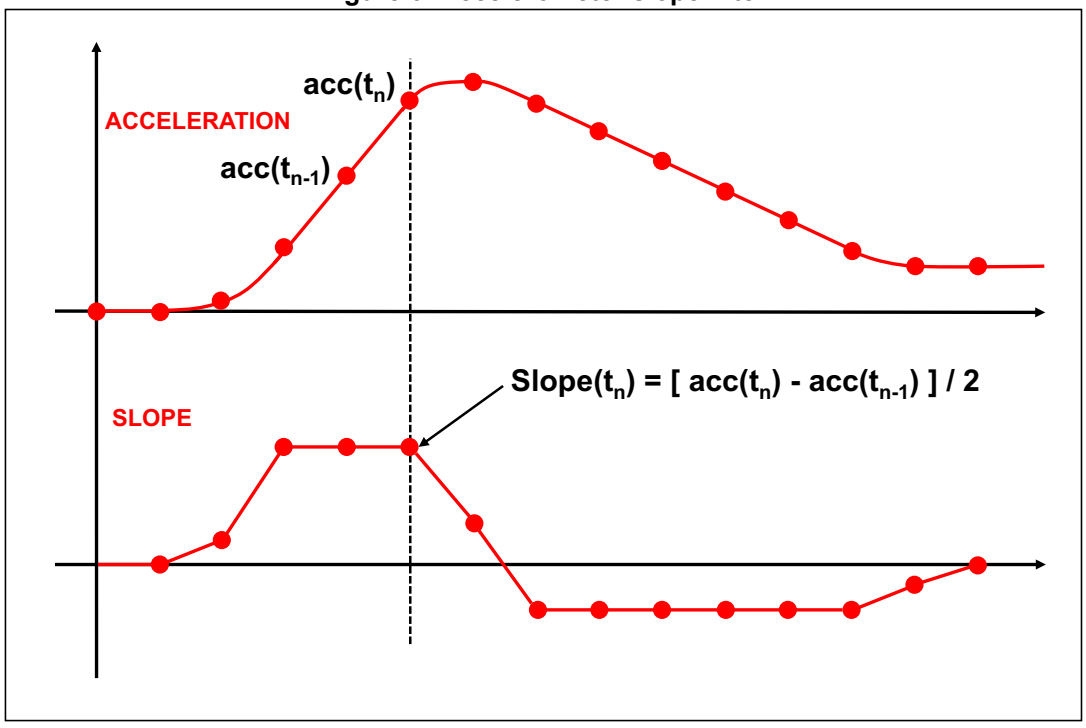
12.4.1 Accelerometer slope filter

As shown in *Figure 6*, the ISM303DAC accelerometer embeds a digital slope filter which is used for wakeup and single/double-tap features. The slope filter output data is computed using the following formula:

$$slope(t_n) = [\ acc(t_n) - acc(t_n\text{-}1)\ ] / 2$$

An example of a slope data signal is illustrated in *Figure 6*.

Figure 6. Accelerometer slope filter



Slope filter bandwidth is ~ ODR/4 and its data is available in the output registers and FIFO by setting the FDS_SLOPE bit of CTRL2_A (21h) to '1'.

13 Reading accelerometer output data

13.1 Startup sequence

Once the device is powered up, the accelerometer automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed, i.e. after approximately 20 milliseconds, the accelerometer automatically enters power-down.

To turn on the accelerometer and gather acceleration data, it is necessary to select one of the operating modes through the CTRL1_A register.

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

1. Write CTRL1_A = 60h // Acc = 400 Hz (high-resolution mode)
2. Write CTRL4_A = 01h // Acc data-ready interrupt on INT1_XL

13.2 Using the status register

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

For the accelerometer, the reads should be performed as follows:

1. Read STATUS_A
2. If DRDY = 0, then go to 1
3. Read OUTX_L_A
4. Read OUTX_H_A
5. Read OUTY_L_A
6. Read OUTY_H_A
7. Read OUTZ_L_A
8. Read OUTZ_H_A
9. Data processing
10. Go to 1

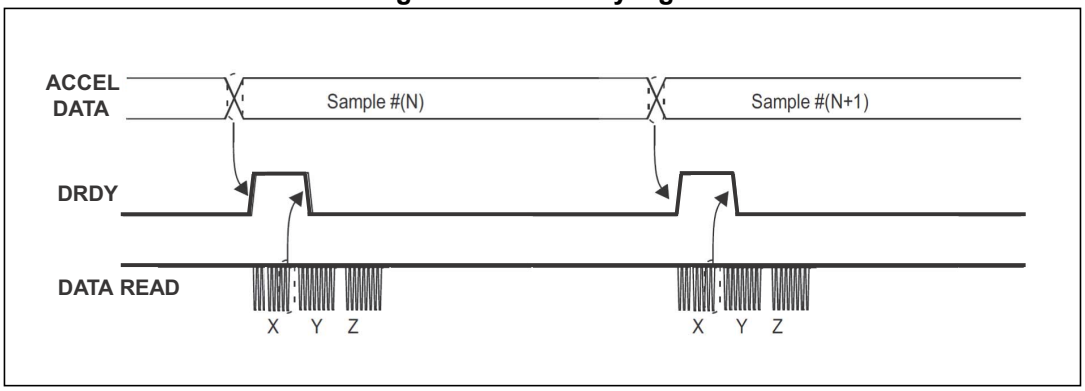
13.3 Using the data-ready signal

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

For the accelerometer sensor, the data-ready signal is represented by the DRDY bit of the STATUS_A register. The signal can be driven to the INT1_XL pin by setting to 1 the INT1_DRDY bit of the CTRL4_A register and to the INT2_XL pin by setting to 1 the INT2_DRDY bit of the CTRL5_A register.

The data-ready signal rises to 1 when a new set of data has been generated and it is available for reading. In DRDY latched mode (DRDY_PULSED bit = 0 in CTRL5_A register), which is the default condition, the signal gets reset when the higher part of one of the channels has been read (29h, 2Bh, 2Dh). In DRDY pulsed mode (DRDY_PULSED = 1), the pulse duration is about 75 µs.

Figure 7. Data-ready signal



13.4 Using the block data update (BDU) feature

If reading the accelerometer data is particularly slow and cannot be synchronized (or it is not required) with either the DRDY event bit in the STATUS_A register or with the DRDY signal driven to the INT1_XL/INT2_XL pins, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL1_A (20h) 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_A and OUTX_L_A, OUTY_H_A and OUTY_L_A, OUTZ_H_A and OUTZ_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 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.

13.5 Understanding output data

The measured acceleration data are sent to the OUTX_H_A, OUTX_L_A, OUTY_H_A, OUTY_L_A, OUTZ_H_A, and OUTZ_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 output data for the X, Y, Z channels is given by the concatenation OUTX_H_A & OUTX_L_A, OUTY_H_A & OUTY_L_A, OUTZ_H_A & OUTZ_L_A and it is expressed as a two's complement number.

Acceleration data is represented as 16-bit numbers, called LSB, but has different resolution according to the selected operating mode (LP/HR/HF). See *Table 10: Accelerometer ODR and power mode selection*.

After calculating the LSB, it must be multiplied by the proper sensitivity parameter to obtain the corresponding value in m*g*.

13.5.1 Example of output data

Hereafter there is a simple example of how to use the LSB data and transform it into m*g*.

The values are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....).

Get raw data from the sensor (HR mode, ODR 200 Hz):

```
OUTX_L_A: 5Ch
OUTX_H_A: FDh
OUTY_L_A: 74h
OUTY_H_A: 00h
OUTZ_L_A: F8h
OUTZ_H_A: 42h
```

Do registers concatenation:

```
OUTX_H_A & OUTX_L_A: FD5Ch
OUTY_H_A & OUTY_L_A: 0074h
OUTZ_H_A & OUTZ_L_A: 42F8h
```

Apply sensitivity (e.g. 0.061 at full scale ±2 *g*):

```
X: -676 * 0.061 = -41 mg
Y: +116 * 0.061 = +7 mg
Z: +17144 * 0.061 = +1046 mg
```

14 Accelerometer interrupt generation and embedded functions

In the ISM303DAC accelerometer the interrupt generation is based on accelerometer data, so, for interrupt-generation purposes, the accelerometer sensor has to be set in an active operating mode (not in power-down).

The interrupt generator can be configured to detect:

- Free-fall;
- Wake-up;
- 6D/4D orientation detection;
- Single-tap and double-tap sensing;
- Activity/Inactivity detection.

All these interrupt signals, together with FIFO interrupt signals and sensor data-ready, can be independently driven to the INT1_XL and INT2_XL interrupt pins or checked by reading the dedicated source register bits.

The H_LACTIVE bit of the CTRL3_A register must be used to select the polarity of the interrupt pins also when the DRDY signal is routed to them. If this bit is set to 0 (default value), the interrupt pins are active high and they change from low to high level when the related interrupt condition is verified. Otherwise, if the H_LACTIVE bit is set to 1 (active low), the interrupt pins are normally at high level and they change from high to low when interrupt condition is reached.

The PP_OD bit of CTRL3_A allows changing the behavior of the interrupt pins also when the DRDY signal is routed to them from push-pull to open drain. If the PP_OD bit is set to 0, the interrupt pins are in push-pull configuration (low-impedance output for both high and low level). When the PP_OD bit is set to 1, only the interrupt active state is a low-impedance output.

The LIR bit of CTRL3_A allows applying the latched mode to the interrupt signals (not affecting the DRDY signal). When the LIR bit is set to 1, once the interrupt pin is asserted, it must be reset by reading the related interrupt source register. If the LIR bit is set to 0, the interrupt signal is automatically reset when the interrupt condition is no longer verified or after a certain amount of time in function of the type of interrupt.

14.1 Interrupt pin configuration

The device is provided with two pins that can be activated to generate either data ready or interrupt signals. The functionality of these pins is selected through the CTRL4_A register for the INT1_XL pin, and through the CTRL5_A register for the INT2_XL pin.

Hereafter the description of these interrupt control registers; the default value of their bits is equal to 0, which corresponds to 'disable'. In order to enable the routing of a specific interrupt signal on the pin, the related bit has to be set to 1.

Table 13. CTRL4_A register

b7	b6	b5	b4	b3	b2	b1	b0
0	INT1_ S_TAP	INT1_ WU	INT1_ FF	INT1_ TAP	INT1_ 6D	INT1_ FTH	INT1_ DRDY

- INT1_S_TAP: Single-tap event recognition is routed on the INT1_XL pin.
- INT1_WU: Wakeup event recognition is routed on the INT1_XL pin.
- INT1_FF: Free-fall event recognition is routed on the INT1_XL pin.
- INT1_TAP: Tap event recognition is routed on the INT1_XL pin.
- INT1_6D: 6D event recognition is routed on the INT1_XL pin.
- INT1_FTH: FIFO threshold event is routed on the INT1_XL pin.
- INT1_DRDY: Accelerometer data-ready is routed on the INT1_XL pin.

Table 14. CTRL5_A register

b7	b6	b5	b4	b3	b2	b1	b0
DRDY_ PULSED	INT2_ BOOT	INT2_ON_ INT1	0	0	0	INT2_ FTH	INT2_ DRDY

- DRDY_PULSED: Data-ready interrupt mode selection: latched mode / pulsed mode.
- INT2_BOOT: Boot state routed on the INT2_XL pin.
- INT2_ON_INT1: All INT2 signals are routed also to the INT1_XL pin.
- INT2_FTH: FIFO threshold event is routed on the INT2_XL pin.
- INT2_DRDY: Accelerometer data-ready on the INT2_XL pin.

14.2 Event status

If multiple interrupt signals are routed on the same pin (INTx), 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 proper status register, which also will clear the event.

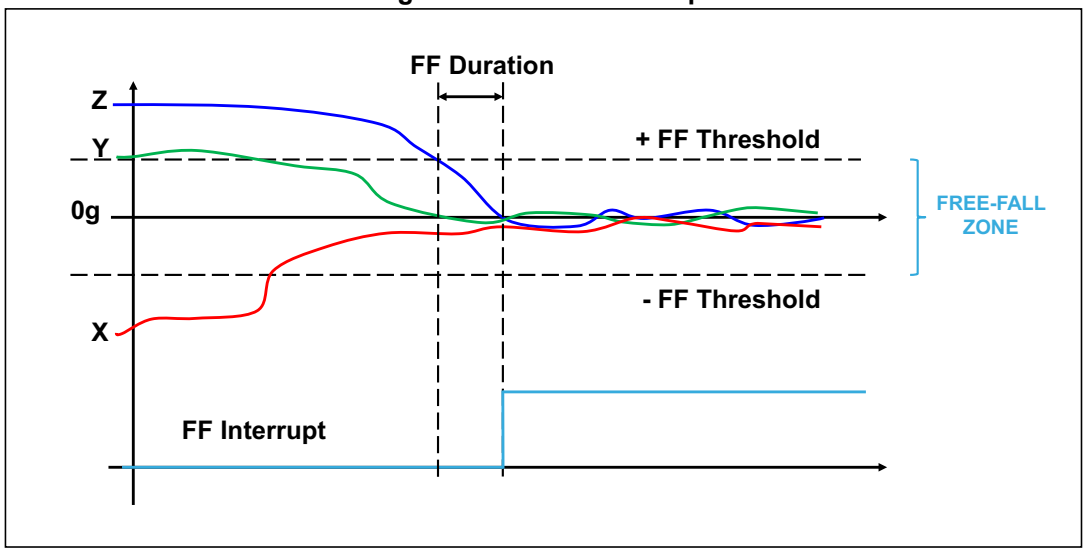
As indicated below, the STATUS_A register is duplicated at address 36h in order to allow a multiple read of consecutive registers (36h/37h/38h/39h).

- STATUS_A (27h) or STATUS_DUP_A (36h)
- WAKE_UP_SRC_A (37h)
- TAP_SRC_A (38h)
- 6D_SRC_A (39h)

14.3 Free-fall interrupt

Free-fall detection refers to a specific register configuration that allows recognizing when the device is in free-fall: the acceleration measured 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. Configurable threshold and duration parameters are associated to free-fall event detection: the threshold parameter defines the free-fall zone amplitude; the duration parameter defines the minimum duration of the free-fall interrupt event to be recognized (*Figure 8: Free-fall interrupt*).

Figure 8. Free-fall interrupt



The free-fall event signal can be routed to the INT1_XL pin by setting to 1 the INT1_FF bit of the CTRL4_A register; it can also be checked by reading the FF_IA bit of the WAKE_UP_SRC_A register.

If latch mode is disabled (LIR bit of CTRL3_A is set to 0), the interrupt signal is automatically reset when the free-fall condition is no longer verified. If latch mode is enabled and the free-fall interrupt signal is driven to the interrupt pins, once a free-fall event has occurred and the interrupt pin is asserted, it must be reset by reading the WAKE_UP_SRC_A register. If the latch mode is enabled, but the interrupt signal is not driven to the interrupt pins, the latch

feature does not take effect (the FF_IA bit in WAKE_UP_SRC_A is reset when the free-fall condition is no longer verified).

The register used to configure the threshold parameter is named FREE_FALL_A; the unsigned threshold value is related to the value of the FF_THS[2:0] field value as indicated in *Table 15* and is expressed in units of 31.25 m*g*. The LSB values given in this table are valid for any accelerometer full-scale value.

Table 15. Free-fall threshold LSB value

FREE_FALL_A - FF_THS[2:0]	Threshold LSB value
000	5
001	7
010	8
011	10
100	11
101	13
110	15
111	16

Duration time is measured in N/ODR, where N is the content of the FF_DUR[5:0] field of the FREE_FALL_A / WAKE_UP_DUR_A registers and ODR is the accelerometer data rate.

A basic SW routine for free-fall event recognition is given below.

1. Write 60h in CTRL1_A // Turn on the accelerometer
// ODR = 400 Hz, FS = ±2 *g*
2. Write 00h in WAKE_UP_DUR_A // Set event duration (FF_DUR5 = 0)
3. Write 33h in FREE_FALL_A // Set FF threshold (FF_THS[2:0] = 011b)
// Set six sample event duration (FF_DUR[5:0] = 000110b)
4. Write 10h in CTRL4_A // FF interrupt driven to INT1_XL pin
5. Write 04h in CTRL3_A // Latch interrupt

The sample code exploits a threshold set to ~310 m*g* (31.25 m*g* * 10) for free-fall recognition and the event is notified by hardware through the INT1_XL pin. The FF_DUR[5:0] field of FREE_FALL_A / WAKE_UP_DUR_A registers is configured like this to ignore events that are shorter than 6/ODR = 6/400 Hz = 15 msec in order to avoid false detections.

14.4 Wake-up interrupt

In the ISM303DAC device the wake-up feature is implemented using the slope filter (see *Section 12.4.1: Accelerometer slope filter* for more details), as illustrated in *Figure 6: Accelerometer slope filter*. The wake-up interrupt signal is generated if a certain number of consecutive slope filtered data exceed the configured threshold (*Figure 9: Wake-up interrupt*).

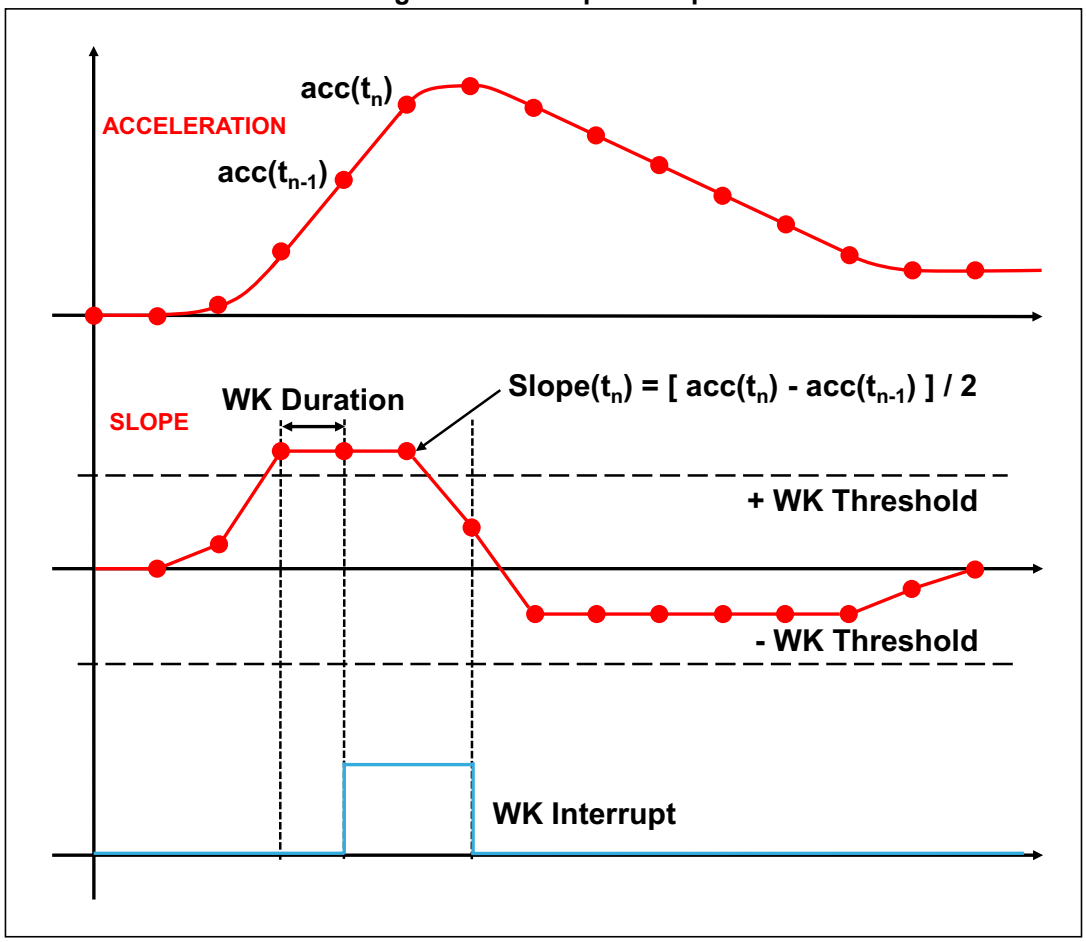
The unsigned threshold value is defined using the WU_THS[5:0] bits of the WAKE_UP_THS_A register; the value of 1 LSB of these 6 bits depends on the selected accelerometer full scale: 1 LSB = FS/64. The threshold is applied to both positive and negative data: for a wake-up interrupt generation at least one of the three axes must be bigger than the threshold.

The duration parameter defines the minimum duration of the wake-up event to be recognized; its value is set using the WU_DUR[1:0] bits of the WAKE_UP_DUR_A register: 1 LSB corresponds to 1*ODR time, where ODR is the accelerometer output data rate. It is important to appropriately define the duration parameter to avoid unwanted wake-up interrupts due to spurious spikes of the input signal.

This interrupt signal can be driven to the INT1_XL interrupt pin by setting to 1 the INT1_WU bit of the CTRL4_A register; it can also be checked by reading the WU_IA bit of the WAKE_UP_SRC_A register. The X_WU, Y_WU, Z_WU bits of the WAKE_UP_SRC_A register indicate which axis has triggered the wake-up event.

If latch mode is disabled (LIR bit of CTRL3_A is set to 0), the interrupt signal is automatically reset when the filtered data falls below the threshold. If latch mode is enabled and the wake-up interrupt signal is driven to the interrupt pins, once a wake-up event has occurred and the interrupt pin is asserted, it must be reset by reading the WAKE_UP_SRC_A register. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect (the WU_IA bit in WAKE_UP_SRC_A is reset when the free-fall condition is no longer verified).

Figure 9. Wake-up interrupt



The example code which implements the SW routine for the wake-up event recognition is given below.

1.	Write 60h in CTRL1_A	// Turn on the accelerometer // ODR = 400 Hz, FS = ±2 *g*
2.	Write 00h in WAKE_UP_DUR_A	// No duration
3.	Write 02h in WAKE_UP_THS_A	// Set wake-up threshold
4.	Write 20h in CTRL4_A	// Wake-up interrupt driven to INT1_XL pin

Since the duration time is set to zero, the wake-up interrupt signal is generated for each X,Y,Z slope data exceeding the configured threshold. The WU_THS field of the WAKE_UP_THS_A register is set to 000010b, therefore the wake-up threshold is 62.5 m*g* (= 2 * FS / 64).

14.5 6D/4D orientation detection

The ISM303DAC device provides the capability to detect the orientation of the device in space, enabling easy implementation of energy-saving procedures and automatic image rotation for mobile devices.

14.5.1 6D orientation detection

Six orientations of the device in space can be detected; the interrupt signal is asserted when the device switches from one orientation to another. The interrupt is not re-asserted as long as the position is maintained.

6D interrupt is generated when only one axis exceeds a selected threshold and the acceleration values measured from the other two axes are lower than the threshold: the ZH, ZL, YH, YL, XH, XL bits of the 6D_SRC_A register indicate which axis has triggered the 6D event.

In more detail:

Table 16. 6D_SRC_A register

b7	b6	b5	b4	b3	b2	b1	b0
0	6D_IA	ZH	ZL	YH	YL	XH	XL

- 6D_IA is set high when the device switches from one orientation to another.
- ZH (YH, XH) is set high when the face perpendicular to the Z (Y,X) axis is almost flat and the acceleration measured on the Z (Y,X) axis is positive and in the module bigger than the threshold.
- ZL (YL, XL) is set high when the face perpendicular to the Z (Y,X) axis is almost flat and the acceleration measured on the Z (Y,X) axis is negative and in the module bigger than the threshold.

The 6D_THS[1:0] bits of the TAP_6D_THS_A register are used to select the threshold value used to detect the change in device orientation. The threshold values given in *Table 17: Threshold for 4D/6D function* are valid for each accelerometer full-scale value.

Table 17. Threshold for 4D/6D function

6D_THS[1:0]	Threshold value [degrees]
00	80
01	70
10	60
11	50

This interrupt signal can be driven to the INT1_XL interrupt pin by setting to 1 the INT1_6D bit of the CTRL4_A register; it can also be checked by reading the 6D_IA bit of the 6D_SRC_A register.

If latch mode is disabled (LIR bit of CTRL3_A is set to 0), the interrupt signal is active only for 1/ODR[s] then it is automatically deasserted (ODR is the accelerometer output data rate). If latch mode is enabled and the 6D interrupt signal is driven to the interrupt pins, once

an orientation change has occurred and the interrupt pin is asserted, a read of the 6D_SRC_A register clears the request and the device is ready to recognize a different orientation. If the latch mode is enabled, but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

Referring to the six possible cases illustrated in *Figure 10: 6D recognized orientations*, the content of the 6D_SRC_A register for each position is shown in *Table 18: 6D_SRC_A register for 6D positions*.

Figure 10. 6D recognized orientations

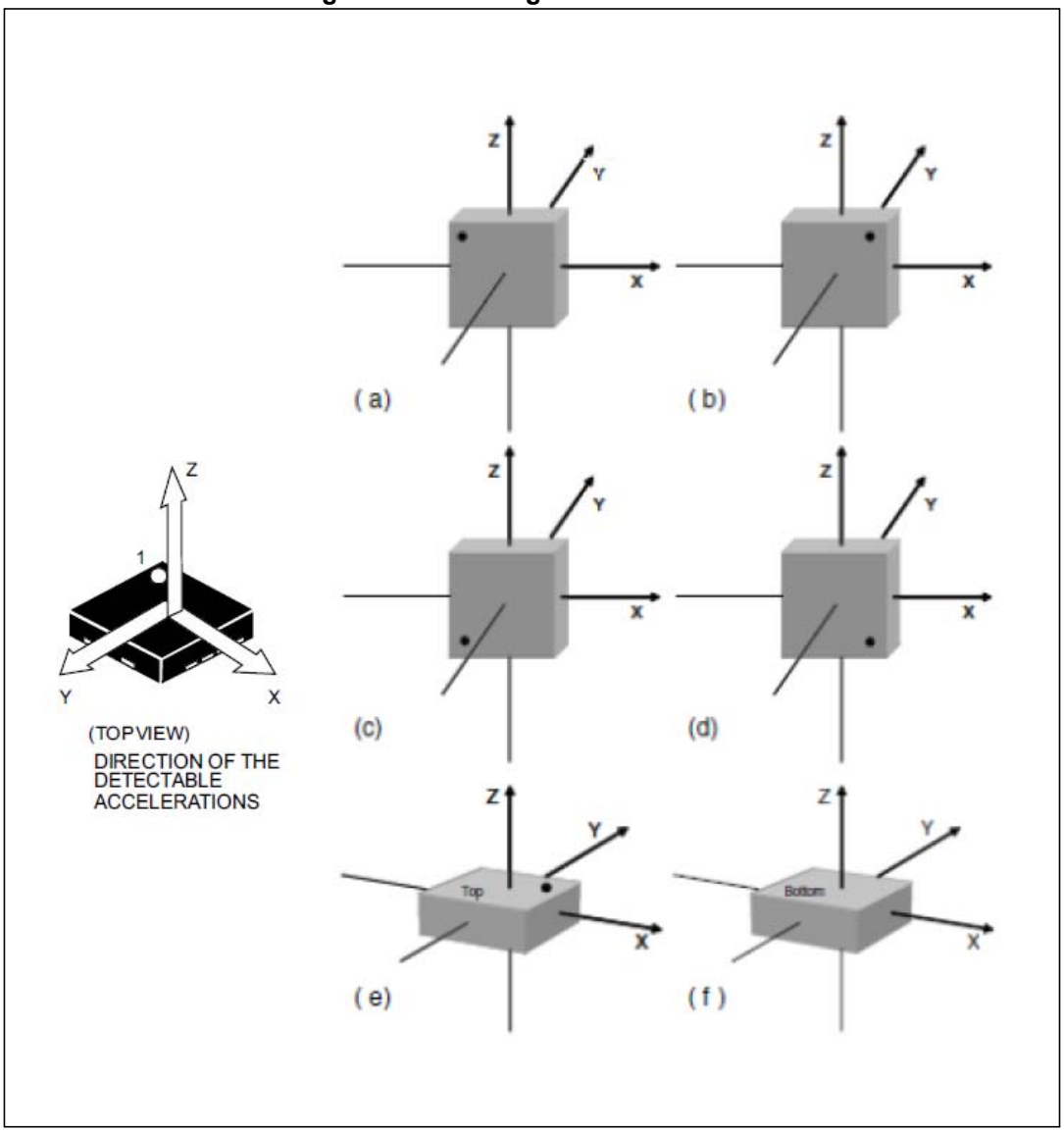


Table 18. 6D_SRC_A register for 6D positions

Case	6D_IA	ZH	ZL	YH	YL	XH	XL
(a)	1	0	0	0	1	0	0
(b)	1	0	0	0	0	0	1
(c)	1	0	0	0	0	1	0
(d)	1	0	0	1	0	0	0
(e)	1	1	0	0	0	0	0
(f)	1	0	1	0	0	0	0

Hereafter an example which implements the SW routine for 6D orientation detection:

1. Write 60h in CTRL1_A // Turn on the accelerometer
// ODR = 400 Hz, FS = ±2 *g*
2. Write 40h in TAP_6D_THS_A // Set 6D threshold (6D_THS[1:0] = 10b = 60 degrees)
3. Write 04h in CTRL4_A // 6D interrupt driven to INT1_XL pin

14.5.2 4D orientation detection

The 4D direction function is a subset of the 6D function especially defined to be implemented in mobile devices for portrait and landscape computation. It can be enabled by setting the 4D_EN bit of the TAP_6D_THS_A register 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 18: 6D_SRC_A register for 6D positions*.

14.6 Single-tap and double-tap recognition

The single-tap and double-tap recognition functions featured in the ISM303DAC 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 the interrupt pin INT1_XL. 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.

In the ISM303DAC device the single-tap and double-tap recognition functions use the slope between two consecutive acceleration samples to detect the tap events; the slope data is computed using the following formula:

$$slope(t_n) = [\ acc(t_n) - acc(t_n\text{-}1)\ ] / 2$$

This function can be fully programmed by the user in terms of expected amplitude and timing of the slope data by means of a dedicated set of registers.

Single and double-tap recognition is meaningful only for ODR ≥ 400 Hz.

14.6.1 Single tap

If the device is configured for single-tap event detection, an interrupt is generated when the slope data on the selected channel exceed the programmed threshold, and return below it within the shock time window.

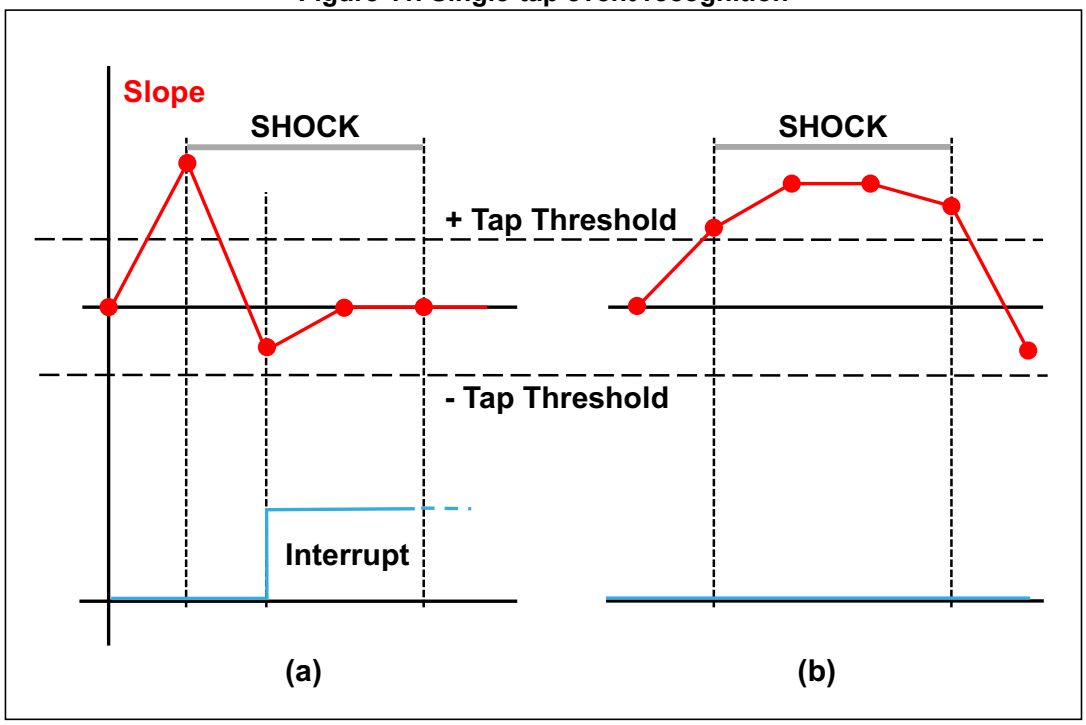
In the single-tap case, if the LIR bit of the CTRL3_A register is set to 0, the interrupt is kept high for the duration of the quiet window.

In order to enable the latch feature on the single-tap interrupt signal, the LIR bit of CTRL3_A has to be set to 1: the interrupt is kept high until the TAP_SRC_A register is read.

The SINGLE_DOUBLE_TAP bit of WAKE_UP_THS_A has to be set to 0 in order to enable single-tap recognition only.

In case (a) of *Figure 11: Single-tap event recognition* the single-tap event has been recognized, while in case (b) the tap has not been recognized because the slope data fall under the threshold after the shock time window has expired.

Figure 11. Single-tap event recognition



14.6.2 Double tap

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

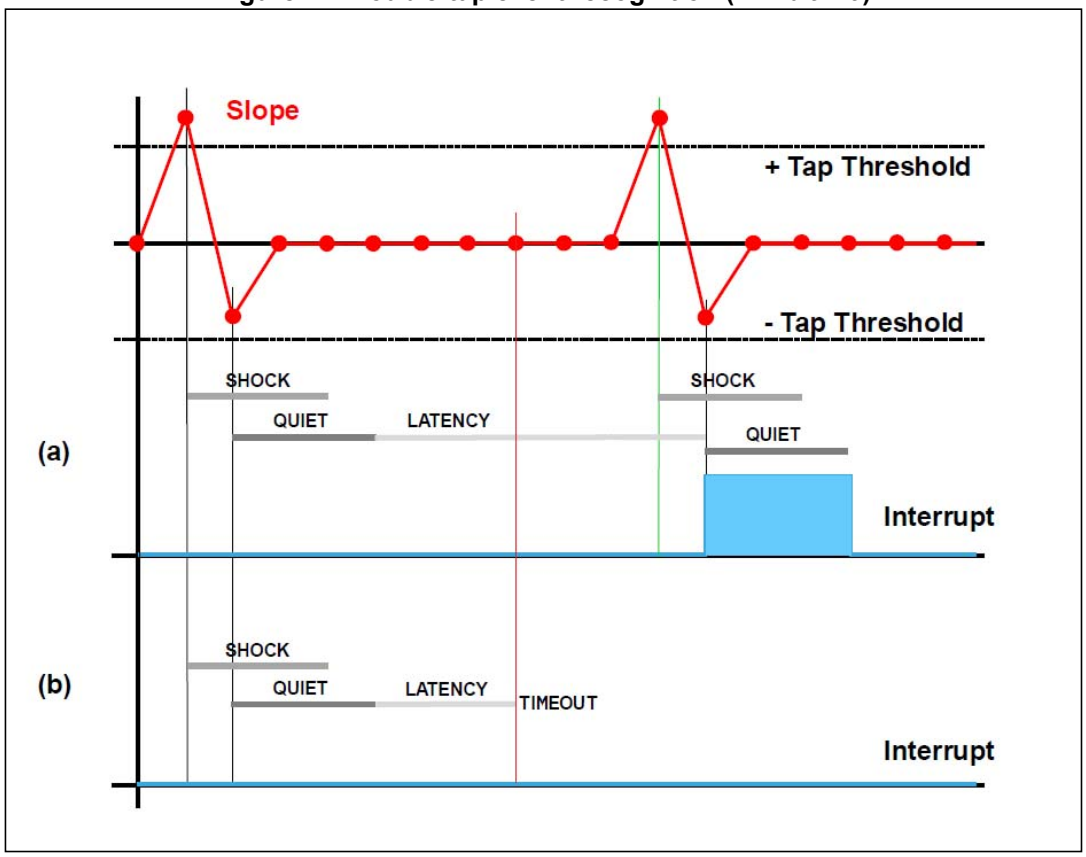
In particular, after the first tap has been recognized, the second tap detection procedure is delayed for an interval defined by the quiet time. This means that after the first tap has been recognized, the second tap detection procedure starts only if the slope data exceed the threshold after the quiet window but before the latency window has expired. In case (a) of *Figure 12: Double-tap event recognition (LIR bit = 0)*, a double-tap event has been correctly recognized, while in case (b) the interrupt has not been generated because the slope data exceed the threshold after the latency window interval has expired.

Once the second tap detection procedure is initiated, the second tap is recognized with the same rule as the first: the slope data must return below the threshold before the shock window has expired.

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

In the double-tap case, if the LIR bit of the CTRL3_A register is set to 0, the interrupt is kept high for the duration of the quiet window. If the LIR bit is set to 1, the interrupt is kept high until the TAP_SRC _A register is read.

Figure 12. Double-tap event recognition (LIR bit = 0)



14.6.3 Single-tap and double-tap recognition configuration

The ISM303DAC device can be configured to output an interrupt signal when tapped (once or twice) in any direction: the TAP_X_EN, TAP_Y_EN and TAP_Z_EN bits of the CTRL3_A register must be set to 1 to enable the tap recognition on X, Y, Z directions, respectively.

Configurable parameters for tap recognition functionality are the tap threshold and the shock, quiet and latency time windows. Valid ODRs are 400 Hz, 800 Hz and 1600 Hz.

The TAP_THS[4:0] bits of the TAP_6D_THS_A register are used to select the unsigned threshold value used to detect the tap event. The value of 1 LSB of these 5 bits depends on the selected accelerometer full scale: 1 LSB = FS/32. The unsigned threshold is applied to both positive and negative slope data.

The shock time window defines the maximum duration of the overthreshold event: the acceleration must return below the threshold before the shock window has expired, otherwise the tap event is not detected. The SHOCK[1:0] bits of the INT_DUR_A register are used to set the shock time window value: the default value of these bits is 00b and corresponds to 4/ODR time, where ODR is the accelerometer output data rate. If the SHOCK[1:0] bits are set to a different value, 1 LSB corresponds to 8/ODR time.

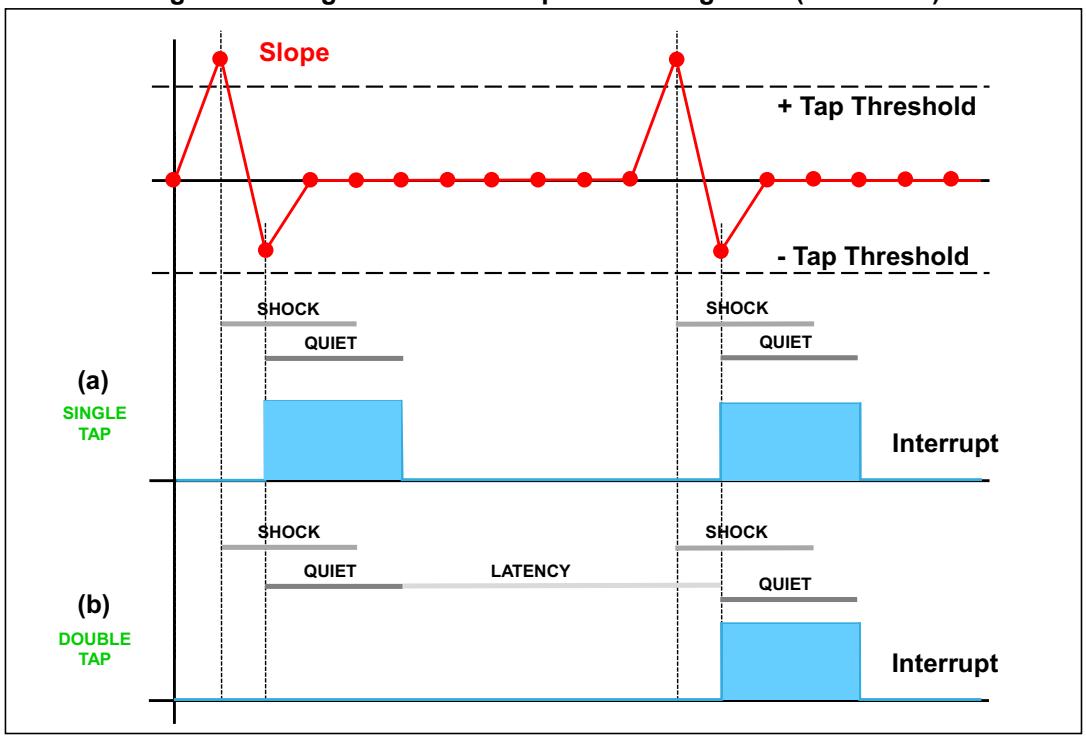
In the double-tap case, the quiet time window defines the time after the first tap recognition in which there must not be any overthreshold. When the latch mode is disabled (LIR bit of CTRL3_A is set to 0), the quiet time also defines the length of the interrupt pulse (in both single and double-tap case). The QUIET[1:0] bits of the INT_DUR_A register are used to set the quiet time window value: the default value of these bits is 00b and corresponds to 2/ODR time, where ODR is the accelerometer output data rate. If the QUIET[1:0] bits are set to a different value, 1 LSB corresponds to 4/ODR time.

In the double-tap case, the latency time window defines the maximum time between two consecutive detected taps. The latency time period starts just after the completion of the quiet time of the first tap. The LAT[3:0] bits of the INT_DUR_A register are used to set the latency time window value: the default value of these bits is 0000b and corresponds to 16/ODR time, where ODR is the accelerometer output data rate. If the LAT[3:0] bits are set to a different value, 1 LSB corresponds to 32/ODR time.

Figure 13: Single and double-tap event recognition (LIR bit = 0) illustrates a single-tap event (a) and a double-tap event (b). These interrupt signals can be driven to the INT1_XL interrupt pin by setting to 1 the INT1_S_TAP bit of the CTRL4_A register for the single-tap case, and setting to 1 the INT1_TAP bit of the CTRL4_A register for the double-tap case.

No single/double-tap interrupt is generated if the accelerometer is in inactivity status (see *Section 14.7: Activity/Inactivity recognition* for more details.

Figure 13. Single and double-tap event recognition (LIR bit = 0)



The tap interrupt signals can also be checked by reading the TAP_SRC_A (38h) register, described in *Table 19: TAP_SRC_A register*.

Table 19. TAP_SRC_A register

b7	b6	b5	b4	b3	b2	b1	b0
0	TAP_IA	SINGLE _TAP	DOUBLE _TAP	TAP _SIGN	X_TAP	Y_TAP	Z_TAP

- TAP_IA is set high when a single-tap or double-tap event has been detected.
- SINGLE_TAP is set high when a single tap has been detected.
- DOUBLE_TAP is set high when a double tap has been detected.
- TAP_SIGN indicates the acceleration sign when the tap event is detected. It is set low in case of positive sign and it is set high in case of negative sign.
- X_TAP (Y_TAP, Z_TAP) is set high when the tap event has been detected on the X (Y, Z) axis

Single and double-tap recognition works independently. Setting the SINGLE_DOUBLE_TAP bit of WAKE_UP_THS_A to 0, only the single-tap recognition is enabled: double-tap recognition is disabled and cannot be detected. When the SINGLE_DOUBLE_TAP is set to 1, both single and double-tap recognition are enabled.

If the latch mode is enabled and the interrupt signal is driven to the interrupt pins, the value assigned to SINGLE_DOUBLE_TAP also affects the behavior of the interrupt signal: when it is set to 0, the latch mode is applied to the single-tap interrupt signal; when it is set to 1, the latch mode is applied to the double-tap interrupt signal only. The latched interrupt signal is kept high until the TAP_SRC_A register is read. If the latch mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

14.6.4 Single-tap example

Hereafter an example code which implements the SW routine for single-tap detection.

1. Write 60h in CTRL1_A // Turn on the accelerometer
// ODR = 400 Hz, FS = ±2 *g*
2. Write 38h in CTRL3_A // Enable tap detection on X, Y, Z-axis
3. Write 09h in TAP_6D_THS_A // Set tap threshold
4. Write 06h in INT_DUR_A // Set quiet and shock time windows
5. Write 00h in WAKE_UP_THS_A // Only single tap enabled (SINGLE_DOUBLE_TAP = 0)
6. Write 40h in CTRL4_A // Single-tap interrupt driven to INT1_XL pin

In this example the TAP_THS field of the TAP_6D_THS_A register is set to 01001b, therefore the tap threshold is 562.5 m*g* (= 9 * FS / 32).

The SHOCK field of the INT_DUR_A register is set to 10b: an interrupt is generated when the slope data exceeds the programmed threshold, and returns below it within 40 ms (= 2 * 8 / ODR) corresponding to the shock time window.

The QUIET field of the INT_DUR_A register is set to 01b: since the latch mode is disabled, the interrupt is kept high for the duration of the quiet window, therefore 10 ms (= 1 * 4 / ODR).

14.6.5 Double-tap example

The example code which implements the SW routine for single-tap detection is given below.

1. Write 60h in CTRL1_A // Turn on the accelerometer
// ODR = 400 Hz, FS = ±2 *g*
2. Write 38h in TAP_CFG_A // Enable tap detection on X, Y, Z-axis
3. Write 0Ch in TAP_6D_THS_A // Set tap threshold
4. Write 7Fh into INT_DUR_A // Set duration, quiet and shock time windows
5. Write 80h in WAKE_UP_THS_A // Single & double-tap enabled (SINGLE_DOUBLE_TAP = 1)
6. Write 08h in CTRL4_A // Double-tap interrupt driven to INT1_XL pin

In this example the TAP_THS field of the TAP_6D_THS_A register is set to 01100b, therefore the tap threshold is 750 m*g* (= 12 * FS / 32).

For interrupt generation, during the first and the second tap the slope data must return below the threshold before the shock window has expired. The SHOCK field of the INT_DUR_A register is set to 11b, therefore the shock time is 60 ms (= 3 * 8 / ODR).

For interrupt generation, after the first tap recognition there must not be any slope data overthreshold during the quiet time window. Furthermore, since the latch mode is disabled, the interrupt is kept high for the duration of the quiet window. The QUIET field of the INT_DUR_A register is set to 11b, therefore the quiet time is 30 ms (= 3 * 4 / ODR).

For the maximum time between two consecutive detected taps, the LAT field of the INT_DUR_A register is set to 0111b, therefore the duration time is 560 ms (= 7 * 32 / ODR).

14.7 Activity/Inactivity recognition

The activity/inactivity recognition function allows reducing system power consumption and developing new smart applications.

When the activity/inactivity recognition function is activated, the ISM303DAC device is able to automatically enter low-power mode and decrease the accelerometer sampling rate to 12.5 Hz, increasing back the accelerometer ODR and bandwidth as soon as the wake-up interrupt event has been detected.

With this feature the system may be efficiently switched from low-power consumption to full performance and vice-versa depending on user-selectable acceleration events, thus ensuring power saving and flexibility.

The activity/inactivity recognition function is enabled by setting to 1 the SLEEP_ON bit of the WAKE_UP_THS_A register.

The activity/inactivity recognition function uses the slope between two consecutive acceleration samples to detect the activity/inactivity event; the slope data is computed using the following formula:

$$slope(t_n) = [\ acc(t_n) - acc(t_n\text{-}1)\ ] / 2$$

This function can be fully programmed by the user in terms of expected amplitude and timing of the slope data by means of a dedicated set of registers (*Figure 14: Activity/Inactivity recognition*).

The unsigned threshold value is defined using the WK_THS[5:0] bits in the WAKE_UP_THS_A register; the value of 1 LSB of these 6 bits depends on the selected accelerometer full scale: 1 LSB = 1 / 64 of FS. The threshold is applied to both positive and negative slope data.

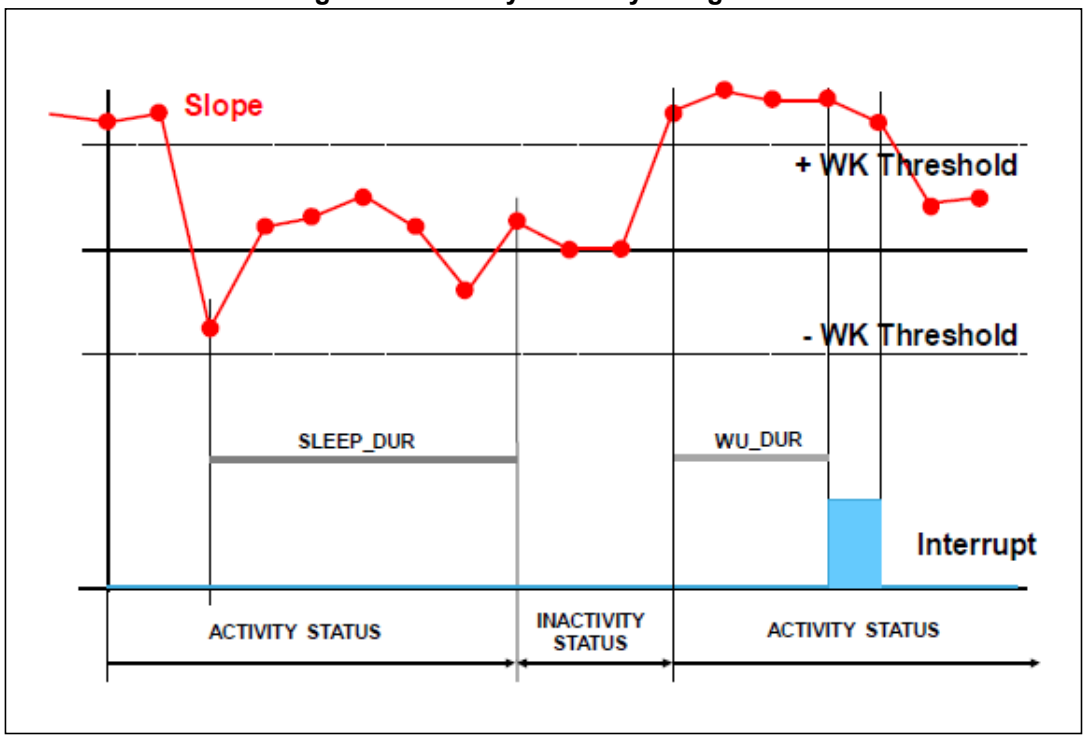
When a certain number of consecutive X,Y,Z slope data is smaller than the configured threshold, the ODR [3:0] bits of the CTRL1_A register are bypassed (inactivity) and the accelerometer is internally set to 12.5 Hz although the content of CTRL1_A is left untouched. The duration of the inactivity status to be recognized is defined by the SLEEP_DUR[3:0] bits of the WAKE_UP_DUR_A register: 1 LSB corresponds to 512/ODR time, where ODR is the accelerometer output data rate.

When the inactivity status is detected, no interrupt is generated to the application processor (SLEEP_STATE_IA bit of the WAKE_UP_SRC_A register cannot be routed on the pin).

When a single sample of slope data on one axis becomes bigger than the threshold, the CTRL1_A register settings are immediately restored (activity). The wake-up interrupt event can be delayed in function of the value of the WU_DUR[1:0] bits of the WAKE_UP_DUR_A register: 1 LSB corresponds to 1/ODR time, where ODR is the accelerometer output data rate. In order to generate the interrupt at the same time as the Inactivity/Activity event, WU_DUR[1:0] have to be set to 0.

When the wake-up event is detected, the interrupt is set high for 1/ODR period, then it is automatically deasserted (the WU_IA event on the pin must be routed by setting the INT1_WU bit of CTRL4_A register to 1).

Figure 14. Activity/Inactivity recognition



The code provided below is a basic routine for activity/inactivity detection implementation.

1.	Write 50h in CTRL1_A	// Turn on the accelerometer // ODR = 200 Hz, FS = ±2 *g*
2.	Write 42h in WAKE_UP_DUR_A	// Set duration for inactivity detection // Set duration for wake-up detection
3.	Write 42h in WAKE_UP_THS_A	// Set activity/inactivity threshold // Enable activity/inactivity detection
4.	Write 20h in CTRL4_A	// Activity (wakeup) interrupt driven to INT1_XL pin

In this example the WU_THS field of the WAKE_UP_THS_A register is set to 000010b, therefore the activity/inactivity threshold is 62.5 m*g* (= 2 * FS / 64).

Before inactivity detection, the X,Y,Z slope data must be smaller than the configured threshold for a period of time defined by the SLEEP_DUR field of the WAKE_UP_DUR_A register: this field is set to 0010b, corresponding to 5.12 s (= 2 * 512 / ODR). After this period of time has elapsed, the accelerometer ODR is internally set to 12.5 Hz.

The activity status is detected and the CTRL1_A register settings immediately restored if the slope data of (at least) one axis are bigger than the threshold and the wake-up interrupt was notified after an interval defined by the WU_DUR field of the WAKE_UP_DUR_A register: this field is set to 10b, corresponding to 10 ms (= 2 * 1 / ODR).

14.8 Boot status

After the device is powered up, the ISM303DAC accelerometer performs a 20 ms boot procedure to load the trimming parameters. After the boot is completed the accelerometer is automatically configured in power-down mode.

After power-up, the trimming parameters can be re-loaded by setting to 1 the BOOT bit of the CTRL2_A register.

No toggle of the device power lines is required and the content of the device control registers is not modified, so the device operating mode doesn't change after boot. If the reset to the default value of the control registers is required, it can be performed by setting to 1 the SW_RESET bit of the CTRL2_A register.

The boot status signal can be driven to the INT2_XL interrupt pin by setting to 1 the INT2_BOOT bit of the CTRL5_A register: the signal goes to '1' while a boot is taking place, and returns to '0' when it is done.

To return the device to the power-down default settings, follow these steps from ANY operating mode:

1. Set SW_RESET bit to '1'
2. Wait until SW_RESET bit returns to '0'
3. Set BOOT bit to '1'
4. Wait 20 ms

15 Accelerometer First-In First-Out (FIFO) buffer

In order to limit intervention by the host processor and facilitate post-processing data for events recognition, the ISM303DAC embeds a first-in, first-out buffer (FIFO) for each of the three accelerometer output channels, X, Y, and Z.

FIFO use allows 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 five different modes that guarantee a high level of flexibility during application development: Bypass mode, FIFO mode, Continuous mode, Bypass-to-Continuous and Continuous-to-FIFO mode.

A programmable watermark level and the FIFO_FULL event can be enabled to generate dedicated interrupts on the INT1_XL pin.

15.1 FIFO description

The FIFO buffer is able to store up to 256 acceleration samples of 14 bits for each channel or store the output of the acceleration module computation up to 768 entries (see *Section 15.4: Module-to-FIFO*); data are stored in the 14-bit 2's complement left-justified representation, which means that they always have to be right-shifted by two.

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 20. FIFO buffer full representation (256th sample set stored)

Output registers	28h	29h	2Ah	2Bh	2Ch	2Dh
	Xl	Xh	Yl	Yh	Zl	Zh
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(254)	Xl(254)	Xh(254)	Yl(254)	Yh(254)	Zl(254)	Zh(254)
FIFO(255)	Xl(255)	Xh(255)	Yl(255)	Yh(255)	Zl(255)	Zh(255)

Table 21. FIFO buffer full representation (257th sample set stored and 1st sample discarded)

Output registers	28h	29h	2Ah	2Bh	2Ch	2Dh
	Xl	Xh	Yl	Yh	Zl	Zh
FIFO index	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(255)	Xl(256)	Xh(256)	Yl(256)	Yh(256)	Zl(256)	Zh(256)

Table 20: FIFO buffer full representation (256th sample set stored) represents the FIFO full status when 256 samples are stored in the buffer while *Table 21: FIFO buffer full representation (257th sample set stored and 1st sample discarded)* represents the next step when the 257th 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 different from Bypass, the ISM303DAC accelerometer output registers (28h to 2Dh) always contain the oldest FIFO sample set.

15.2 FIFO registers

The FIFO buffer is managed by four different accelerometer registers, two of these allow enabling and configuring the FIFO behavior, the other two provide information about the buffer status.

A few other registers are used to route FIFO events on the pin to interrupt the application processor. These are discussed in *Section 15.3: FIFO interrupts*.

15.2.1 FIFO_CTRL_A register (25h)

The FIFO_CTRL_A register contains the mode at which the FIFO is set. At reset by default the FIFO mode is Bypass which means off; the FIFO gets enabled and starts storing the samples (or the module) as soon as the mode is set to a mode other than Bypass.

Table 22. FIFO_CTRL_A register

b7	b6	b5	b4	b3	b2	b1	b0
FMODE2	FMODE1	FMODE0	0	MODULE_ TO_FIFO	0	0	IF_CS_ PU_DIS

The FMODE[2:0] bits select the FIFO buffer behavior:

1. FMODE[2:0] = 000b: Bypass mode (FIFO turned off)
2. FMODE[2:0] = 001b: FIFO mode
3. FMODE[2:0] = 011b: Continuous-to-FIFO mode
4. FMODE[2:0] = 100b: Bypass-to-continuous mode
5. FMODE[2:0] = 110b: Continuous mode

MODULE_TO_FIFO enables the module of the X/Y/Z samples (sqrt(x2+y2+z2)) to go in FIFO instead of the samples themselves. Please note that the MODULE_ON bit of the FUNC_CTRL_A register must be on.

15.2.2 FIFO_THS_A register (2Eh)

This register may be used to set the FIFO threshold level.

Table 23. FIFO_THS_A register

b7	b6	b5	b4	b3	b2	b1	b0
FTH7	FTH6	FTH5	FTH4	FTH3	FTH2	FTH1	FTH0

The FTH[7:0] bits define the watermark level; when FIFO content is greater than or equal to this value, the FTH bit is set to 1 in the FIFO_SRC_A register.

15.2.3 FIFO_SRC_A (2Fh)

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

Table 24. FIFO_SRC_A register

b7	b6	b5	b4	b3	b2	b1	b0
FTH	FIFO_ OVR	DIFF8	0	0	0	0	0

- FTH bit is set high when FIFO content exceeds watermark level. This flag can be routed to the pin (see *Section 15.3: FIFO interrupts*).
- FIFO_OVR bit is set high when the first sample is overwritten after the FIFO buffer is full. This means that the FIFO buffer contains 256 unread samples. The FIFO_OVR bit is reset when the first sample set has been read.
- DIFF8 bit (or FIFO_FULL bit) is used together with bits of FIFO_SAMPLES (DIFF[7:0]) to provide information of how many FIFO entries are used (000000000b means FIFO empty, 100000000b means FIFO full). This flag can be routed to the pin (see *Section 15.3: FIFO interrupts*).

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

Table 25. FIFO_SRC_A behavior assuming FTH[7:0] = 15

FTH	DIFF8 (FIFO_FULL)	FIFO_OVR	DIFF[8:0]	Unread FIFO samples	Timing
0	0	0	000000000	0	t0
0	0	0	000000001	1	t0 + 1/ODR
0	0	0	000000010	2	t0 + 2/ODR
...	...	...	...	...	...
0	0	0	000001110	14	t0 + 14/ODR
1	0	0	000001111	15	t0 + 15/ODR
...	...	...	...	...	...
1	0	0	011111111	255	t0 + 255/ODR
1	1	0	100000000	256	t0 + 256/ODR
1	1	1	100000000	256	t0 + 257/ODR

15.2.4 FIFO_SAMPLES_A (30h)

The content of this register is used together with the DIFF8 bit of the FIFO_SRC_A register (DIFF[7:0]) to provide information of how many FIFO entries are used (000000000b means FIFO empty, 100000000b means FIFO full).

Table 26. FIFO_SAMPLES_A register

b7	b6	b5	b4	b3	b2	b1	b0
DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0

15.3 FIFO interrupts

There are two specific FIFO events that can be routed to the pin in order to interrupt the main processor: FIFO threshold and FIFO full.

The third FIFO event, FIFO_OVR, cannot be routed on the pin, but can instead be polled by reading the corresponding bit in the FIFO_SRC_A register.

15.3.1 FIFO threshold

The FIFO threshold is a configurable feature 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 threshold level. The user can select the desired level in a range from 0 to 255 using the FTH[7:0] field in the FIFO_THS_A register.

If the number of entries in FIFO (DIFF[8:0]) is greater than or equal to the value programmed in FTH[7:0], the FTH bit is set high in the FIFO_SRC_A register.

DIFF[8:0] increases by one step at the ODR frequency and decreases by one step every time that a sample set read is performed by the user.

The threshold flag (FTH) can be routed to the INT1_XL and INT2_XL pin to provide a dedicated interrupt for the application processor that can consume less power between each interrupt. The INT1_FTH bit of CTRL4_A register and the INT2_FTH bit of CTRL5_A register are dedicated to this task.

15.3.2 FIFO full

It is possible to configure the device to generate an interrupt whenever the FIFO gets full. To do so just set the INT1_FSS7 bit of the WAKE_UP_DUR_A register to '1'. To avoid losing samples, the FIFO read operation must start and complete inside 1 ODR window.

15.4 Module-to-FIFO

If the module computation is on (MODULE_ON bit of FUNC_CTRL_A register is '1') and the MODULE_TO_FIFO bit of FIFO_CTRL_A is set to '1', then the FIFO buffer will not contain the acceleration samples but instead the computation of their module (sqrt(x2+y2+z2)).

Since the module is a number of 14-bit in size, the FIFO buffer may contain up to 768 module entries (256 * 3). When the module is stored in FIFO, the threshold set in FIFO_THS_A and the number of stored data available in the FIFO_SAMPLES_A register are represented by 1LSb = 3 samples.

The following is a simple procedure to enable the FIFO to contain the acceleration sample module:

1. Set MODULE_ON to 1 in FUNC_CTRL_A (3Fh) to enable module computation
2. Set MODULE_TO_FIFO bit of FIFO_CTRL_A (25h) to '1' to save module in FIFO buffer
3. Enable the FIFO in one of the operative modes (see *Section 15.5: FIFO modes*)

15.5 FIFO modes

The ISM303DAC FIFO buffer can be configured to operate in five different modes selectable by the FMODE[2:0] field in FIFO_CTRL_A register. Available configurations ensure a high-level of flexibility and extend the number of functions usable in application development.

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

15.5.1 Bypass mode

When Bypass mode is enabled, the FIFO is not operational: 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.

Bypass mode is activated by setting the FMODE[2:0] field to 000b in the FIFO_CTRL_A register.

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.

15.5.2 FIFO mode

In FIFO mode, the buffer continues filling until full (256 sample sets stored). As soon as the FIFO_OVR flag gets to '1', the FIFO stops collecting data and its content remains unchanged until a different mode is selected.

FIFO mode is activated by setting the FMODE[2:0] field to 001b in the FIFO_CTRL_A register.

By selecting this mode, FIFO starts data collection and DIFF[8:0] changes according to the number of samples stored. At the end of the procedure, the FIFO_OVR flag rises to 1, and data can then be retrieved, performing a 256 sample set read from the output registers. 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 read procedure it is necessary to exit Bypass mode.

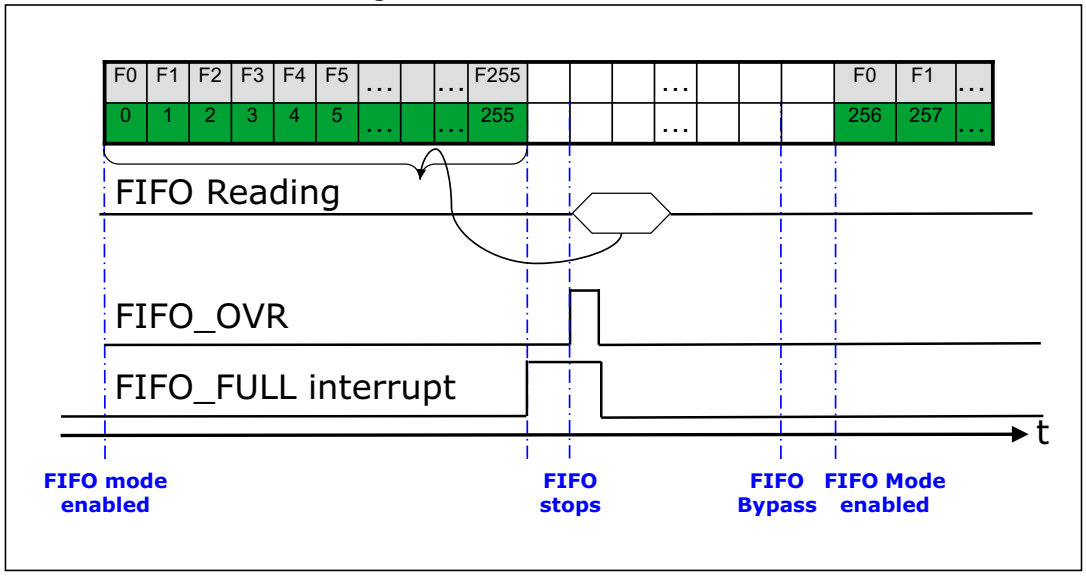
In order to serve the FIFO_FULL event as soon as possible, it is recommended to route it to the pin in order to generate an interrupt, which will then be managed by a specific handler:

1. Set INT1_FSS7 to '1': Enables FIFO_FULL interrupt
2. Set FMODE[2:0] = 001b: Enables FIFO mode

When the FIFO_FULL interrupt is generated or the FIFO_OVR bit is high (polling mode):

1. Read data from the accelerometer output registers

Figure 15. FIFO mode behavior



As indicated in *Figure 15: FIFO mode behavior*, when FIFO mode is enabled, the buffer starts to collect data and fills all 256 slots (from F0 to F255) at the selected output data rate. When the buffer is full, as the next sample comes in and overrides the buffer, the FIFO_OVR 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 read procedure may be performed inside an interrupt handler triggered by a FIFO_FULL condition (DIFF8) and it is composed of a 256 sample set of 6 bytes for a total of 1236 bytes and retrieves data starting from the oldest sample stored in FIFO (F0). The FIFO_OVR 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.

15.5.3 Continuous mode

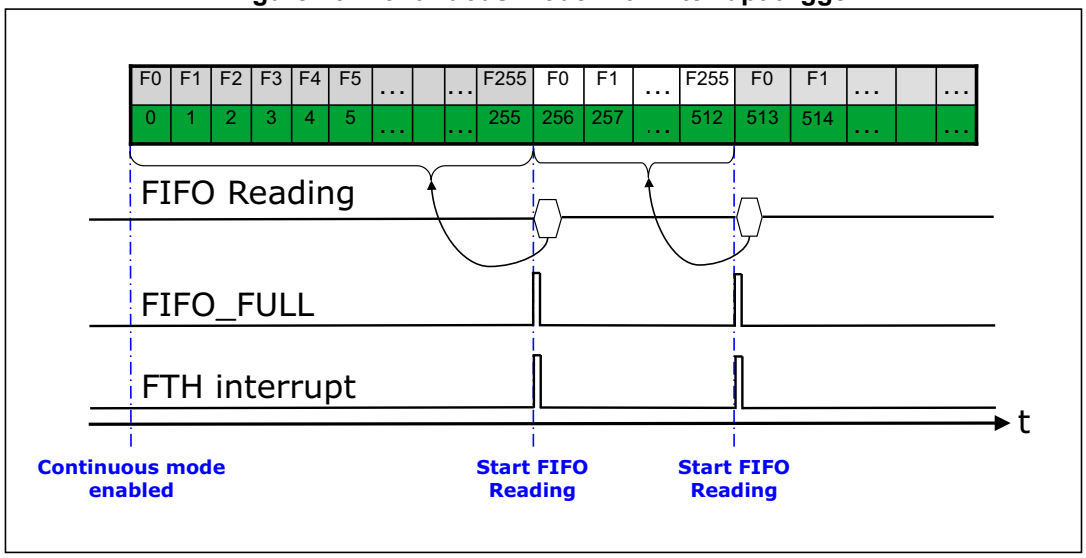
In Continuous mode FIFO continues filling, when the buffer is full, the FIFO index restarts from the beginning and older data is replaced by current data. The oldest values continue to be overwritten until a read operation frees FIFO slots. The host processor reading speed is most important in order to free slots faster than new data is made available. FMODE[2:0] in Bypass configuration is used to stop this mode.

Follow these steps for FIFO Continuous configuration which sets a threshold to generate an interrupt to trigger a read by the application processor:

1. Set FTH[7:0] to 255.
2. Set INT2_FTH to '1': Enable FIFO threshold interrupt
3. Activate Continuous mode by setting the FMODE[2:0] field to 110b in the FIFO_CTRL_A register (25h).

When the FTH interrupt is generated, data is read from the accelerometer output registers.

Figure 16. Continuous mode with interrupt trigger



As indicated in *Figure 16*, when Continuous mode is enabled, the FIFO buffer is continuously filling (from F0 to F255) at the selected output data rate. When the buffer is full, the FTH interrupt (as well as the FIFO_FULL condition indicated by the DIFF8 bit in FIFO_SRC_A (2Fh), which might also be used to trigger an interrupt) goes high, and the application processor may read all FIFO samples (256 * 6 bytes) as soon as possible to avoid loss of data and to limit intervention by the host processor which increases system efficiency. See *Section 15.6: Retrieving data from FIFO* for more details on FIFO reading speed.

When a read command is sent to the device, the content of the output registers is moved to the SPI/I²C register and the current oldest FIFO value is shifted into the output registers in order to allow the next read operation.

15.5.4 Continuous-to-FIFO mode

This mode is a combination of the Continuous and FIFO modes previously described. In Continuous-to-FIFO mode, the FIFO buffer starts operating in Continuous mode and switches to FIFO mode when the selected interrupt (e.g. wakeup, free-fall, tap, ...) occurs.

This mode 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 FIFO buffer is full and stopped.

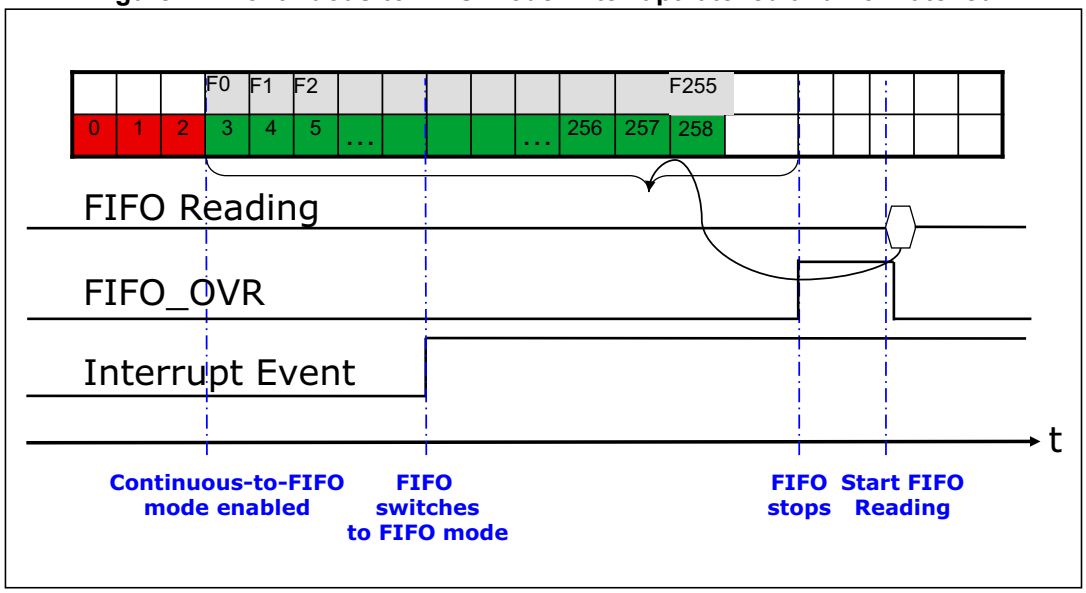
Follow these steps for Continuous-to-FIFO mode configuration:

1. Configure the desired interrupt generator by following the instructions in *Section 14: Accelerometer interrupt generation and embedded functions* (be sure it is latched).
2. Activate Continuous-to-FIFO mode by setting the FMODE[2:0] field to 011b in the FIFO_CTRL_A register (25h).

Note: When the requested event takes place, the FIFO mode change is triggered if and only if the event flag is routed to the INT1_XL or INT2_XL pin.

While in Continuous mode the FIFO buffer continues filling; when the requested event takes place the FIFO mode changes; then, as soon as the buffer becomes full, the FIFO_OVR bit is set high and the next samples overwrite the oldest and the FIFO stops collecting data (see *Figure 17*).

Figure 17. Continuous-to-FIFO mode: interrupt latched and non-latched



15.5.5 Bypass-to-Continuous mode

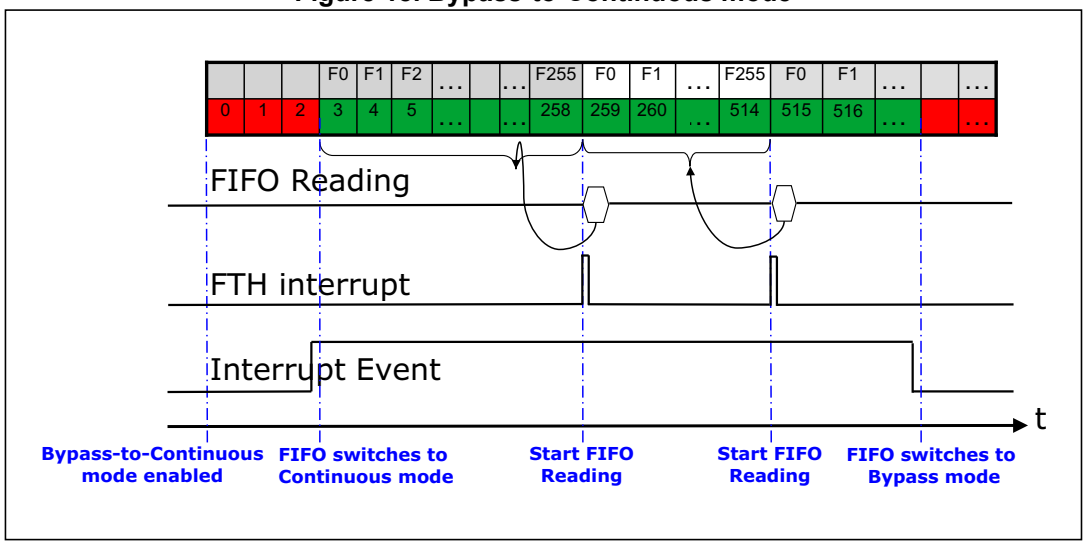
This mode is a combination of the Bypass and Continuous modes previously described. In Bypass-to-Continuous mode, the FIFO buffer starts in Bypass mode and switches to Continuous mode when the selected interrupt (e.g. wakeup, free-fall, tap, ...) occurs.

Follow these steps for Bypass-to-Continuous mode configuration:

1. Configure desired interrupt generator by following the instructions in *Section 14: Accelerometer interrupt generation and embedded functions* (be sure it is latched).
2. Set FTH[7:0] to 255.
3. Set INT2_FTH to '1': Enables FIFO threshold interrupt
4. Activate Bypass-to-Continuous mode by setting the FMODE[2:0] field to 100b in the FIFO_CTRL_A register (25h).

When the FTH interrupt is generated, data is read from the accelerometer output registers.

Figure 18. Bypass-to-Continuous mode



As indicated in *Figure 18*, the FIFO is initially in Bypass mode, so no samples enter in the FIFO buffer. As soon as an event occurs (e.g. a wakeup or a free-fall event) the FIFO switches to Continuous mode and starts to store the samples at the configured data rate. When the programmed threshold is reached, the FTH interrupt goes high, and the application processor may start reading all FIFO samples (256 * 6 bytes) as soon as possible to avoid loss of data.

If the FIFO_OVR flag was set, it will go to 0 as soon as the first FIFO set is read, creating space for new data. Since the FIFO is still in Continuous mode, the FIFO eventually reaches the threshold again and the situation repeats.

Finally, either the interrupt event is cleared or the FIFO enters directly Bypass mode and then it stops collecting data.

15.6 Retrieving data from FIFO

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

Whenever the output registers are read, their content is moved to the SPI/I²C output buffer. FIFO slots are ideally shifted up one level in order to release room for receiving a new sample and the output registers load the current oldest value stored in the FIFO buffer.

The whole FIFO content is retrieved by performing 256 read operations from the accelerometer output registers. The size of the data stored in FIFO is always 14-bit regardless of the power mode. Every other read 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 read byte combination in order to increase application flexibility (ex: 1536 single byte read, 256 reads of 6 bytes, 1 multiple read of 1536 bytes, etc.).

It is recommended to read all FIFO slots in a multiple byte read of 1536 bytes (6 output registers by 256 slots). In order to minimize communication between the master and slave the read address may be automatically incremented by the device by setting the IF_ADD_INC bit of CTRL2_A register to '1'; the device rolls back to 0x28 when register 0x2D is reached.

The I²C speed is lower than SPI and it needs about 29 clock pulses to start communication (Start, Slave Address, Register Address+Write, Restart, Register Address+Read) plus an additional 9 clock pulses for every byte to read (total of 83 clock pulses). So, in the case of standard I²C mode being used (max rate 100 kHz), a single sample set read takes 830 µs while total FIFO download takes about 138.53 ms (29 + 9 * 1536 clock pulses).

In the case of the SPI, instead, 9 clock pulses are required only once at the very beginning to get started (r/w + Register Address) plus additional 8 clock pulses for every byte to read. With a 2 MHz clock a single sample set read would take 28.5 µs, while total FIFO download takes about 6.15 ms.

If this recommendation were followed, using a standard I²C (100 kHz) the complete FIFO read (138.53 ms) would take 14/ODR with ODR at 100 Hz. Using a SPI @2 MHz (10 MHz is the maximum supported by the device) the complete FIFO read would take 1/ODR with ODR at 100 Hz.

So, in order to not lose samples, the application will read samples before the FIFO becomes full, setting a threshold and using the FTH interrupt (see *Section 15.3: FIFO interrupts*).

Table 27. Example: threshold function of ODR

ODR (Hz)	FTH_THS (I²C @ 100 kHz)	FTH_THS (I²C @ 400 kHz)	FTH_THS (SPI @ 2 MHz)
50	36	147	256
100	17	73	256
200	8	36	208
400	4	17	103
800	1	8	51
1600	-	4	25

16 Accelerometer self-test

When the accelerometer self-test is enabled, an actuation force is applied to the sensor, leading to a deflection of the moveable part of the sensor and generates an acceleration. In this case the sensor outputs exhibit a change in their DC levels which are related to the selected full scale through the sensitivity value.

The accelerometer self-test function is off when the ST[2:1] bits of the CTRL3_A register are programmed to 00b; it is enabled when the ST[2:1] bits are set to 01b (positive sign self-test) or 10b (negative sign self-test).

When the accelerometer self-test is activated, the sensor 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.

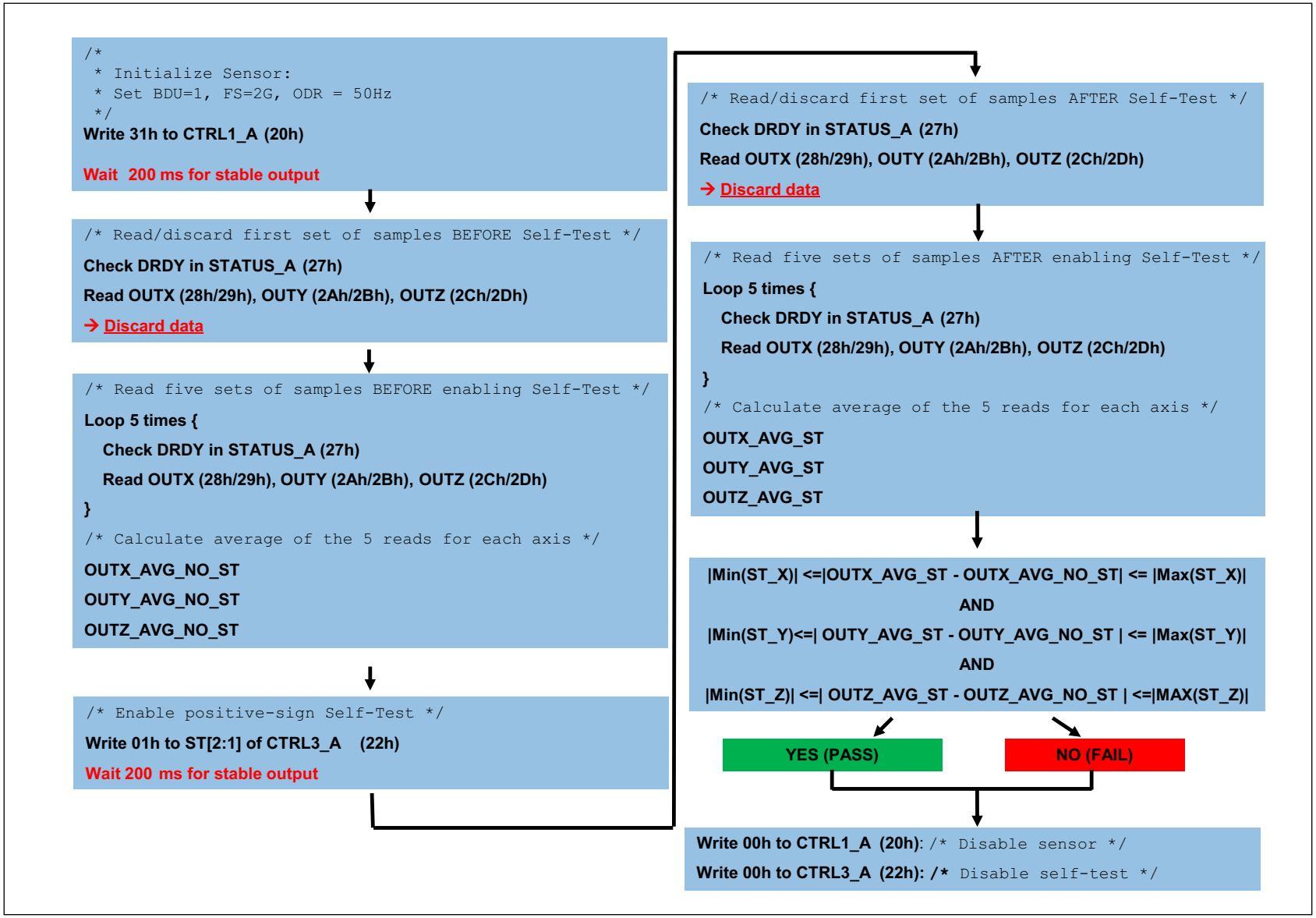
The procedure consists of:

1. enabling the accelerometer
2. averaging five samples before enabling the self-test
3. averaging five samples after enabling the self-test
4. computing the difference in module for each axis and verify it falls in a given range. The min and max value are provided in the datasheet.

The complete accelerometer self-test procedure is indicated in *Figure 19*.

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

Figure 19. Accelerometer self-test procedure



17 Temperature sensor

The ISM303DAC is provided with an internal temperature sensor that is suitable for ambient temperature measurement.

If the accelerometer sensor is in power-down mode, the temperature sensor is off and is showing the last value measured.

The output data rate of temperature sensor is fixed at 12.5 Hz.

The temperature data is given by the OUT_T_A register and it is represented as a number of 8 bits in two's complement format, with a sensitivity of +1 LSB/°C. The output zero level corresponds to 25 °C ±15 °C.

17.1 Example of temperature data calculation

Table 28: Output data registers content vs. temperature provides a few basic examples of the data that is read from the temperature data registers at different ambient temperature values. The values listed in this table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....).

Table 28. Output data registers content vs. temperature

Temperature values	OUT_T_A (26h)
23 °C	FEh
24 °C	FFh
25 °C	00h
27 °C	02h

18 Revision history

Table 29. Document revision history

Date	Revision	Changes
19-Jan-2018	1	Initial release