AN4662
Application note

LIS2HH12: MEMS digital output motion sensor ultra-low-power high-performance 3-axis "pico" accelerometer

Introduction

This document describes the low-voltage 3-axis digital output linear MEMS accelerometer provided in an LGA package.

The LIS2HH12 is an ultra-low-power high-performance 3-axis linear accelerometer belonging to the “pico” family, with a digital I^2C/SPI serial interface standard output.

The device features ultra-low-power operational modes that allow advanced power saving and smart sleep-to-wake-up (Activity) and return-to-sleep (Inactivity) functions.

The LIS2HH12 has dynamic user-selectable full scales of ±2g/±4g/±8g and it is capable of measuring accelerations with output data rates from 10 Hz to 800 Hz.

The device may be configured to generate two independent interrupt signals by detecting an inertial wake-up/free-fall event as well as by the position of the device itself. Thresholds and timing for both interrupt generators are programmable by the end user on the fly.

The LIS2HH12 has an integrated 32-level first-in, first-out (FIFO) buffer allowing the user to store data in order to limit intervention by the host processor.

The LIS2HH12 is available in a small thin plastic land grid array package (LGA) and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

The ultra-small size and weight of the SMD package make it an ideal choice for smartphones and wearable devices and in applications such as pedometers, tilt monitoring and activity recognition.

Contents

List of tables

List of figures

1 Registers

Table 1: Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TEMP_L	0Bh	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
TEMP_H	0Ch	TD15	TD14	TD13	TD12	TD11	TD10	TD9	TD8
RESERVED	00-0Eh								
WHO_AM_I	0Fh	0	1	0	0	0	0	0	1
ACT_THS	1Eh	ATHS7	ATHS6	ATHS5	ATHS4	ATHS3	ATHS2	ATHS1	ATHS0
ACT_DUR	1Fh	ADUR7	ADUR6	ADUR5	ADUR4	ADUR3	ADUR2	ADUR1	ADUR0
CTRL1	20h	HR	ODR2	ODR1	ODR0	BDU	Zen	Yen	Xen
CTRL2	21h	-	DFC1	DFC0	HPM1	HPM0	FDS	HPIS1	HPIS2
CTRL3	22h	FIFO_EN	STOP_FTH	INT1_INACT	INT1_IG2	INT1_IG1	INT1_OVR	INT1_FTH	INT1_DRDY
CTRL4	23h	BW2	BW1	FS1	FS0	BW_SCALE _ODR	IF_ADD _INC	I2C_DISABLE	SIM
CTRL5	24h	DEBUG	SOFT _RESET	DEC1	DEC0	ST2	ST1	H_LActive	PP_OD
CTRL6	25h	BOOT	-	INT2_BOOT	INT2_IG2	INT2_IG1	INT2_Empty	INT2_FTH	INT2_DRDY
CTRL7	26h	-	-	DCRM2	DCRM1	LIR2	LIR1	4D_IG2	4D_IG1
STATUS	27h	ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA
OUT_X_L	28h	XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0
OUT_X_H	29h	XD15	XD14	XD13	XD12	XD11	XD10	XD9	XD8
OUT_Y_L	2Ah	YD7	YD6	YD5	YD4	YD3	YD2	YD1	YD0
OUT_Y_H	2Bh	YD15	YD14	YD13	YD12	YD11	YD10	YD9	YD8
OUT_Z_L	2Ch	ZD7	ZD6	ZD5	ZD4	ZD3	ZD2	ZD1	ZD0
OUT_Z_H	2Dh	ZD15	ZD14	ZD13	ZD12	ZD11	ZD10	ZD9	ZD8
FIFO_CTRL	2E	FMODE2	FMODE1	FMODE0	FTH4	FTH3	FTH2	FTH1	FTH0

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_SRC	2F	FTH	OVR	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0
IG_CFG1	30h	AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
IG_SRC1	31h	-	IA	ZH	ZL	YH	YL	XH	XL
IG_THS_X1	32h	XTH1_7	XTH1_6	XTH1_5	XTH1_4	XTH1_3	XTH1_2	XTH1_1	XTH1_0
IG_THS_Y1	33h	YTH1_7	YTH1_6	YTH1_5	YTH1_4	YTH1_3	YTH1_2	YTH1_1	YTH1_0
IG_THS_Z1	34h	ZTH1_7	ZTH1_6	ZTH1_5	ZTH1_4	ZTH1_3	ZTH1_2	ZTH1_1	ZTH1_0
IG_DUR1	35h	WAIT1	DUR1_6	DUR1_5	DUR1_4	DUR1_3	DUR1_2	DUR1_1	DUR1_0
IG_CFG2	36h	AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE
IG_SRC2	37h	-	IA	ZH	ZL	YH	YL	XH	XL
IG_THS2	38h	TH2_7	TH2_6	TH2_5	TH2_4	TH2_3	TH2_2	TH2_1	TH2_0
IG_DUR2	39h	WAIT2	DUR2_6	DUR2_5	DUR2_4	DUR2_3	DUR2_2	DUR2_1	DUR2_0
XL_REFERENCE	3Ah	XREF7	XREF6	XREF5	XREF4	XREF3	XREF2	XREF1	XREF0
XH_REFERENCE	3Bh	XREF15	XREF14	XREF13	XREF12	XREF11	XREF10	XREF9	XREF8
YL_REFERENCE	3Ch	YREF7	YREF6	YREF5	YREF4	YREF3	YREF2	YREF1	YREF0
YH_REFERENCE	3Dh	YREF15	YREF14	YREF13	YREF12	YREF11	YREF10	YREF9	YREF8
ZL_REFERENCE	3Eh	ZREF7	ZREF6	ZREF5	ZREF4	ZREF3	ZREF2	ZREF1	ZREF0
ZH_REFERENCE	3Fh	ZREF15	ZREF14	ZREF13	ZREF12	ZREF11	ZREF10	ZREF9	ZREF8

2 Operating modes

The LIS2HH12 provides two different operating modes: power-down mode and active mode. In active mode six different output data rates (ODR) can be selected: 800 Hz, 400 Hz, 200 Hz, 100 Hz, 50 Hz and 10 Hz. From 800 Hz to 100 Hz the current consumption is constant for each ODR, while 50 Hz and 10 Hz further reduce the current consumption depending on the selected ODR.

After the power supply is applied, the LIS2HH12 performs a 20 ms boot procedure to load the calibration parameters from internal memory. At the end of the boot phase, the device is automatically configured in power-down mode.

Referring to the LIS2HH12 datasheet, the output data rate ODR[2:0] bits of CTRL1 register are used to select the operating modes (power-down mode, active mode) and related output data rate (*Table 2: "Data rate configuration"*).

Table 2: Data rate configuration

ODR2	ODR1	ODR0	Power mode selection
0	0	0	Power-down mode
0	0	1	Active mode 10 Hz
0	1	0	Active mode 50 Hz
0	1	1	Active mode 100 Hz
1	0	0	Active mode 200 Hz
1	0	1	Active mode 400 Hz
1	1	0	Active mode 800 Hz
1	1	1	N. A.

Table 3: "Power consumption (µA)" shows the typical values of power consumption for the different operating modes.

Table 3: Power consumption (µA)

ODR[Hz]	Current consumption [µA] @ Vdd = 2.5 V [typ.]
Power-down	5
10	50
50	110
100	180
200	180
400	180
800	180

2.1 Power-down mode

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

During power-down mode the Interrupt pins continue to be driven at the last logical state.

2.2 Active mode

In active mode, data are generated at the output data rate (ODR) selected through the ODR[2:0] bits of CTRL1 register. Six different ODRs can be selected in order to guarantee desired application bandwidth: 800 Hz, 400 Hz, 200 Hz, 100 Hz, 50 Hz and 10 Hz.

The three acceleration axes can be individually enabled through the Zen, Yen, and Xen bits of CTRL1 register. All combinations of single, two or three axes can be applied; output registers related to disabled axes are forced to value 0000h.

Data interrupt generation and the FIFO buffer are active and configurable respectively through the IG_CFGx (30h, 36h) and FIFO_CTRL (2Eh) registers.

The Activity/Inactivity recognition function can be enabled to reduce system power consumption.

2.3 Data stabilization time

The LIS2HH12 reading chain contains low-pass filtering to improve signal-to-noise performance and to reduce aliasing effects. For this reason the settling time of the filters must be considered when the device is switched from power-down to active mode or when the ODR is changed.

Table 4: "Number of samples to be discarded" clarifies the number of samples to be discarded depending on the internal filter bandwidth and output data rate selection. Bandwidth can be changed by setting the BW [2:1] bits in CTRL4 register and the output data rate can be changed by setting the ODR [2:0] bits in CTRL1.

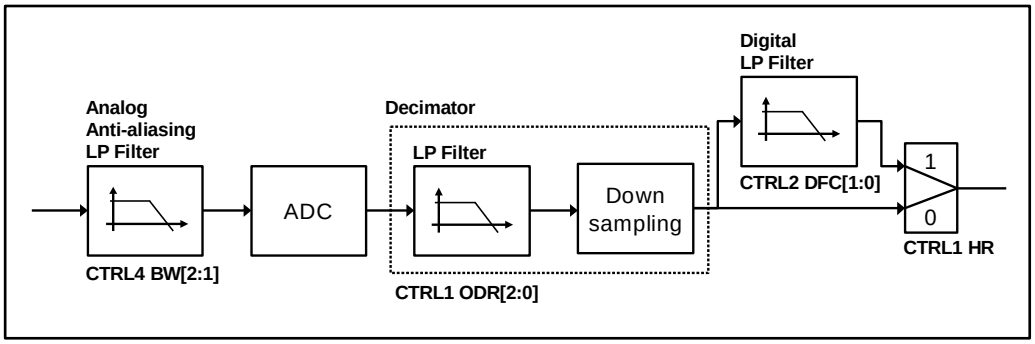
Table 4: Number of samples to be discarded

ODR [Hz]	BW = 400 Hz	BW = 200 Hz	BW = 100 Hz	BW = 50 Hz
10	1	-	-	-
50	1	-	-	-
100	1	1	1	1
200	1	1	1	4
400	1	1	4	7
800	1	4	7	14

2.5 System bandwidth

The sampling chain can be represented by a cascade of four blocks: analog low-pass filter, ADC converter, decimation filter and digital low-pass filter.

Figure 1: Sampling chain block diagram



The analog signal coming from the mechanical block is filtered by a low-pass anti-aliasing filter before being converted from the ADC. In active mode, for ODRs greater than 50 Hz, the filter cutoff frequency can be configured using the BW[2:1] bits in the CTRL4 register if the BW_SCALE_ODR bit in the CTRL4 register is set to 1, otherwise (default condition) the filter bandwidth is automatically set to ODR/2.

Table 5: Anti-aliasing filter bandwidth

CTRL4 BW[2:1]	Bandwidth [Hz]
00	400
01	200
10	100
11	50

Table 6: Anti-aliasing bandwidth options

ODR [Hz]	Analog filter cutoff frequency BW_SCALE_ODR = 0	Analog filter cutoff frequency BW_SCALE_ODR = 1
10/50	400 [Hz]	400 [Hz]
100	50 [Hz]	BW[2:1]
200	100 [Hz]	BW[2:1]
400	200 [Hz]	BW[2:1]
800	400 [Hz]	BW[2:1]

The ADC block is an analog-to-digital converter that converts the input analog signal with constant sampling rate at 3.2 kHz. During active mode the ADC is periodically turned ON/OFF with a duty cycle that depends on the selected data rate.

The decimation filter is a digital block that performs signal filtering and downsampling. The decimator executes an average of N samples in order to reduce the sampling rate from

3.2 kHz to the selected ODR. The cutoff frequency of the low-pass decimation filter depends on the selected ODR as summarized in *Table 7: "Decimation filter bandwidth"*.

Table 7: Decimation filter bandwidth

ODR[Hz]	Decimation filter cutoff frequency [Hz]
10/50	66
100	133
200	266
400	534
800	1090

For ODRs equal to or greater than 100 Hz the anti-aliasing bandwidth is guaranteed by the analog filter when BW_SCALE_ODR = 0.

The digital low-pass filter can be enabled by setting the HR bit to 1 in the CTRL1 register. When high-resolution mode is selected, the digital signal coming from the decimation filter is additionally low-pass filtered and bandwidth is selected by the DFC[1:0] bits in CTRL2 register.

Table 8: Digital low-pass filter bandwidth

HR bit	CTRL2 DFC[1:0]	LP cutoff frequency [Hz]
1	00	ODR/50
1	01	ODR/100
1	10	ODR/9
1	11	ODR/400

2.6 Activity/Inactivity function

The Activity/Inactivity recognition function allows reducing system power consumption and developing new smart applications.

When the Activity/Inactivity recognition function is activated, the LIS2HH12 is able to automatically decrease the sampling rate to 10 Hz, increasing the 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 activated by writing the desired threshold in ACT_THS (1Eh); the threshold is represented by 7 bits and it is applied to a positive and negative direction. The high-pass filter is automatically enabled.

Table 9: Activity/Inactivity control registers

Register	Resolution [LSB]
ACT_THS	Full Scale / 128 [mg]
ACT_DUR	8/ODR [s]

When the acceleration falls below the threshold for at least (8 ACT_DUR +1)/ODR time, the ODR [2:0] bits of CTRL1 are bypassed (Inactivity) and the ODR is internally set to 10 Hz although the content of CTRL1 is left untouched. When acceleration rises above the threshold, the CTRL1 register settings are immediately restored (Activity).

Once the Activity/Inactivity detection function is enabled, the status can be driven to the INT1 pin by setting the INT1_INACT bit to “1” in CTRL3 register.

The Activity/Inactivity function is automatically disabled by setting the ACT_THS register to 00h.

3 Startup sequence

Once the device is powered up, it automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed, i.e. after approximately 20 milliseconds, the device automatically enters power-down mode. To turn on the device and gather acceleration data, it is necessary to select one of the Active modes (*Table 2: "Data rate configuration"*) and to enable at least one of the axes through the CTRL1 register.

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

1. Write CTRL1 = 3Fh // X, Y, Z, enabled, ODR = 100 Hz, BDU enabled
2. Write CTRL3 = 01h // Data Ready Interrupt on INT1

3.1 Reading acceleration data

3.1.1 Using the status register

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

1. Read STATUS
2. If STATUS(ZYXDA) == 0 then go to 1
3. If STATUS(ZYXOR) == 1 then some data have been overwritten
4. Read OUTX_L
5. Read OUTX_H
6. Read OUTY_L
7. Read OUTY_H
8. Read OUTZ_L
9. Read OUTZ_H
10. Data processing
11. Go to 1

The check performed at step #3 allows understanding whether the read rate is adequate compared to the data production rate. If one or more acceleration samples have been overwritten by new data, due to a slow read rate, the ZYXOR bit of STATUS register is set to 1.

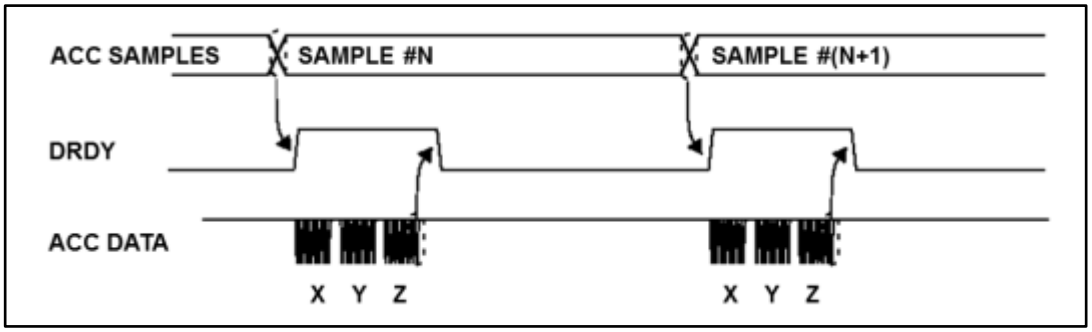
The overrun bits are automatically cleared when all the data present inside the output registers have been read and new data have not been produced in the meantime.

3.1.2 Using the data-ready (DRDY) signal

The device may be configured to have one HW signal to determine when a new set of measurement data is available to be read. This signal is related to the XYZDA bit of the STATUS register. The signal can be driven to the INT1 pin by setting the INT1_DRDY bit to 1 in the CTRL3 register or driven to the INT2 pin by setting the INT2_DRDY bit to 1 in the CTRL6 register. Interrupt polarity can be set to active-low or active-high through the H_LACTIVE bit of CTRL5 register.

The data-ready signal rises to 1 when a new set of acceleration data has been generated and is available to be read.The interrupt is reset when the higher part of the data of all the enabled channels has been read (29h, 2Bh, 2Dh). If the output registers have not been read for a long time, a dummy read must be performed in order to clear the data-ready signal.

Figure 2: Data-ready signal



3.1.3 Using the block data update (BDU) feature

If the reading of the acceleration data is particularly slow and cannot be synchronized (or it is not required) with either the XYZDA bit present inside the STATUS register or with the DRDY signal, it is strongly recommended to set the BDU (block data update) bit to 1 in CTRL1 register.

This feature avoids the reading of most significant and least significant parts of the acceleration data related to different samples. In particular, when the BDU is activated, the data registers related to each channel always contain the most recent acceleration data produced by the device, but, in case the reading of a given pair (i.e. OUT_X_H and OUT_X_L, OUT_Y_H and OUT-Y_L, OUT_Z_H and OUT_Z_L) has started, the refresh for that pair is blocked until both MSB and LSB parts of the data are read.

Note: BDU only guarantees that OUT_X(Y, Z)_L and OUT_X(X,Z)_H have been sampled at the same moment. However, if the reading speed is too slow, it may read X sampled at T1, Y sampled at T2 and Z sampled at T3.

3.2 Understanding acceleration data

The measured acceleration data are sent to the OUT_X_H, OUT_X_L, OUT_Y_H, OUT_Y_L, OUT_Z_H, and OUT_Z_L registers. These registers contain, respectively, the most significant part and the least significant part of the acceleration signals acting on the X, Y, and Z axes.

The complete acceleration data for the X (Y, Z) channel is given by the concatenation of OUT_X_H & OUT_X_L (OUT_Y_H & OUT_Y_L, OUT_Z_H & OUT_Z_L) and it is expressed as a 2’s complement number.

3.2.1 Data alignment

Acceleration data are represented as 16-bit 2's complement numbers and are left-justified.

3.2.2 Examples of acceleration data

Table 10: "Output data registers content vs. acceleration (FS = 2 g, Sensitivity 0.061 mg/digit)" provides a few basic examples of the data that is read from the data registers when the device is subject to a given acceleration. The values listed in the table are given under the hypothesis of perfect device calibration (i.e. no offset, no gain error,....).

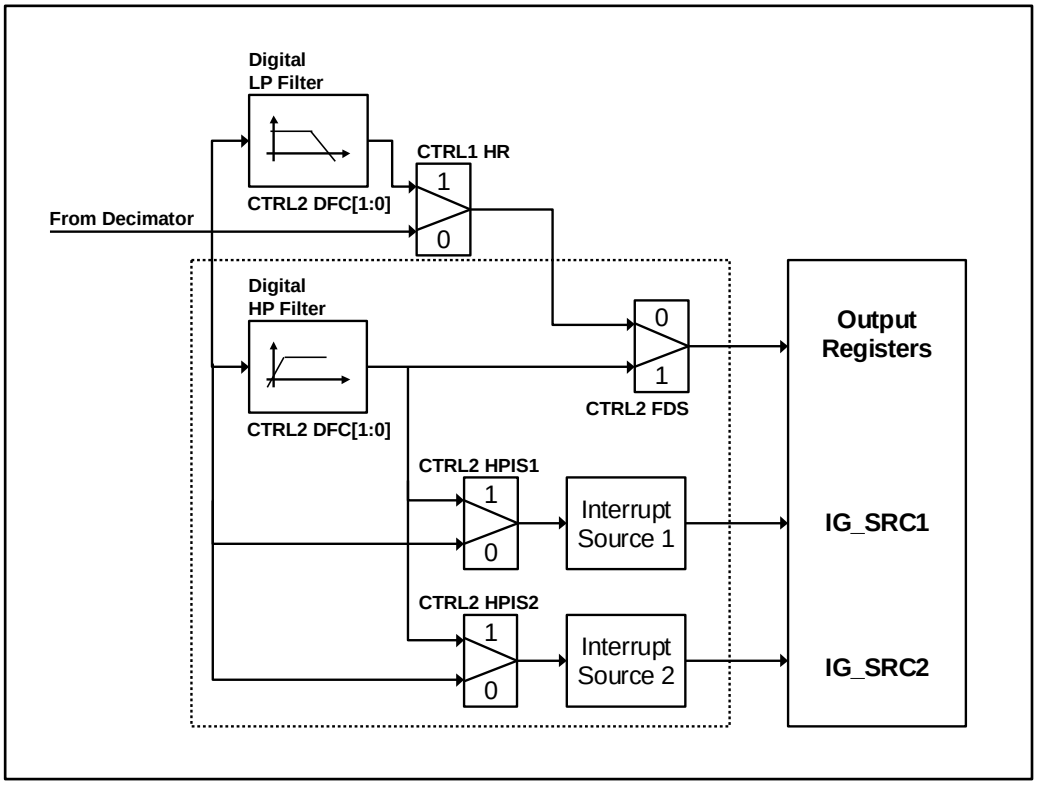
Table 10: Output data registers content vs. acceleration (FS = 2 g, Sensitivity 0.061 mg/digit)

Acceleration output	Register address	
	OUT_X_H (29h)	OUT_X_L (28h)
0 *g*	00h	00h
350 m*g*	16	69
1 *g*	40h	09h
-350 m*g*	E9	97
-1 *g*	BFh	F7h

4 High-pass filter

The LIS2HH12 provides an embedded high-pass filtering capability to easily delete the DC component of the measured acceleration. As shown in *Figure 3: "High-pass filter connections - block diagram"*, through FDS, the HPIS[2:1] bits of the CTRL2 register configuration, it is possible to independently apply the filter on the output data and/or on the interrupt data. This means that it is possible, for example, to get filtered data while the interrupt generation works on unfiltered data and vice versa.

Figure 3: High-pass filter connections - block diagram



4.1 Filter configuration

Referring to *Table 11: "High-pass filter mode configuration"*, two operating modes are possible for the high-pass filter.

Table 11: High-pass filter mode configuration

HPM1	HPM0	High-pass filter mode
0	0	Normal mode
0	1	Reference signal for filtering
1	0	Normal mode
1	1	N. A.

The bandwidth of the high-pass filter depends on the selected ODR and on the settings of the DCF[1:0] bits of the CTRL2 register. The high-pass filter cutoff frequencies (f_t) are shown in *Table 12: "High-pass filter cutoff frequency [Hz]"*.

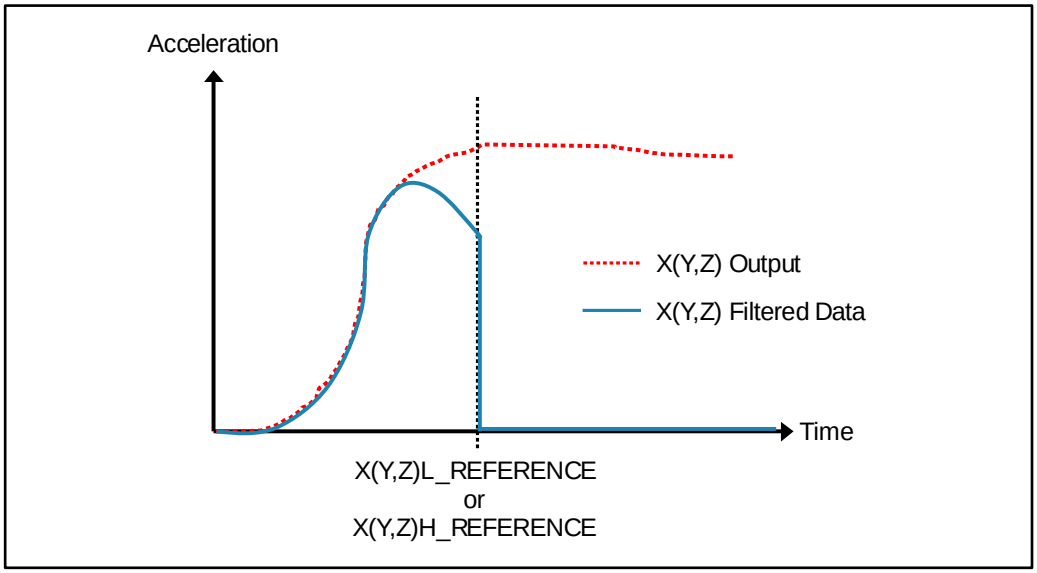
Table 12: High-pass filter cutoff frequency [Hz]

CTRL2 DCF [1:0]	HP cutoff frequency [Hz]
00	ODR/50
01	ODR/100
10	ODR/9
11	ODR/400

4.1.1 Normal mode

In this configuration the high-pass filter can be reset for each axis by reading one of the related reference registers (XL_REFERENCE or XH_REFERENCE, YL_REFERENCE or YH_REFERENCE, ZL_REFERENCE or ZH_REFERENCE). A reset operation instantly deletes the DC component of the acceleration and puts the corresponding axis to zero.

Figure 4: Reading X(Y,Z)L_REFERENCE or X(Y,Z)H_REFERENCE



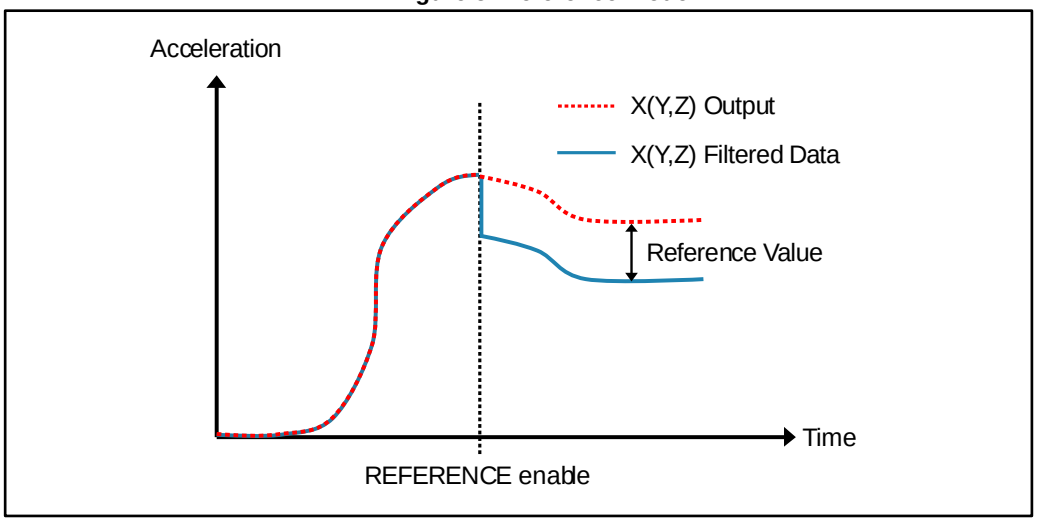
4.1.2 Reference signal for filtering

In this configuration the output data are calculated as the difference between the input acceleration (X, Y, Z) and the content of the related reference registers: XL(H)_REFERENCE, YL(H)_REFERENCE, ZL(H)_REFERENCE. The reference value for each axis is a combination of two registers of 8 bits that allow defining a 16-bit value. This value is expressed in 2's complement representation and the value of 1 LSB of this 16-bit value depends on the selected full scale: LSB = (FS)/(2^{15}) (*Table 13: "Reference mode LSB value"*).

Table 13: Reference mode LSB value

Full scale	Reference mode LSB value (mg)
2	~0.061
4	~0.122
8	~0.244

Figure 5: Reference mode



Reference mode could be used in a static condition when the sensor orientation is not known. This mode allows removing the DC component present on the sensor outputs without applying any filter architecture. This approach is suitable in applications that need to remove the DC component and are subjected to acceleration variations below the cutoff frequency of the HP filter (slow movement). The procedure to be used as a reference should be the following:

1. Place device in a defined position
2. Read OUTX_L(H), OUTY_L(H) and OUTZ_L(H)
3. Set X(Y, Z)L(H)_REFERENCE = OUTX(Y, Z)_L(H)
4. Enable Reference mode: CTRL2 HPM[1:0] = “01”
5. Drive Reference data to output registers: CTRL2 [FDS] = 1

5 Interrupt generation

The LIS2HH integrates two independent interrupt generators (IG1 and IG2) that can behave as free-fall, wake-up, 6D and 4D orientation detection. These signals can be independently driven to the two interrupt pins (INT1 and INT2) or checked by reading the IA bit of IG_SRC1(2).

5.1 Interrupt pin configuration

The device is provided with two pins which can be activated to generate either the data-ready or the interrupt signals. The functionality of these pins is selected through the CTRL3(22h) and CTRL6(25h) registers. Refer to *Table 14: "CTRL3 register"*, *Table 15: "CTRL3 description"*, *Table 16: "CTRL6 register"*, *Table 17: "CTRL6 description"* and to the block diagram given in *Figure 6: "Interrupt signals and interrupt pins"*.

Table 14: CTRL3 register

-	-	INT1 _INACT	INT1 _IG2	INT1 _IG1	INT1 _OVR	INT1 _FTH	INT1 _DRDY

Table 15: CTRL3 description

INT1_INACT	Inactivity interrupt on INT1. Default value 0. (0: disable; 1: enable)
INT1_IG2	Interrupt generator 2 on INT1. Default value 0. (0: disable; 1: enable)
INT1_IG1	Interrupt generator 1 on INT1. Default value 0. (0: disable; 1: enable)
INT1_OVR	FIFO overrun signal on INT1. Default value 0. (0: disable; 1: enable)
INT1_FTH	FIFO threshold signal on INT1. Default value 0. (0: disable; 1: enable)
INT1_DRDY	Data-Ready signal on INT1. Default value 0. (0: disable; 1: enable)

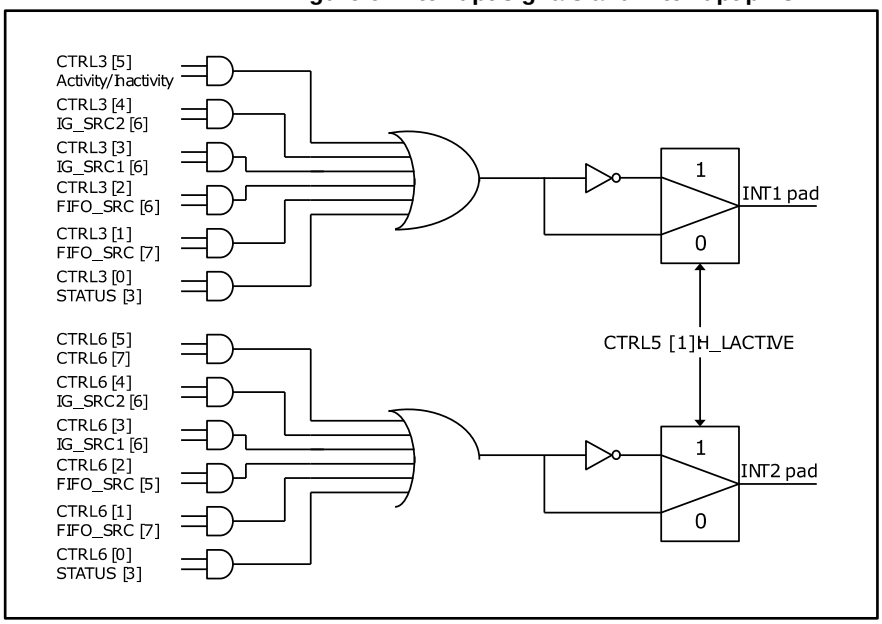
Table 16: CTRL6 register

BOOT	-	INT2 _BOOT	INT2 _IG2	INT2 _IG1	INT2 _EMPTY	INT2 _FTH	INT2 _DRDY

Table 17: CTRL6 description

INT2_BOOT	BOOT interrupt on INT2. Default value 0. (0: disable; 1: enable)
INT2_IG2	Interrupt generator 2 on INT2. Default value 0. (0: disable; 1: enable)
INT2_IG1	Interrupt generator 1 on INT2. Default value 0. (0: disable; 1: enable)
INT2_EMPTY	FIFO empty flag on INT2. Default value 0. (0: disable; 1: enable)
INT2_FTH	FIFO threshold signal on INT2. Default value 0. (0: disable; 1: enable)
INT2_DRDY	Data-Ready signal on INT2. Default value 0. (0: disable; 1: enable)

Figure 6: Interrupt signals and interrupt pins



If multiple interrupt signals are routed to the same pad (INTx), the logic level of this pad is the “OR” combination of the selected interrupt signals. In order to know what signal has generated the interrupt condition, the related source registers have to be read: IG_SRC1(2), FIFO_SRC, STATUS.

6 Inertial interrupt

The LIS2HH12 can provide two inertial interrupt signals and offers several possibilities to personalize these signals. The registers involved in the interrupt generation behavior are IG_CFG1(2), IG_SRC1(2), IG_THS_X(Y,Z)1, IG_THS2 and IG_DUR1(2).

Interrupt functionalities can be configured through the IG_CFG1(2) registers, the two MSB are used to configure the operating mode while the six LSB define the axis behavior with respect to the defined threshold.

Table 18: IG_CFG1(2) register

AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

Table 19: Interrupt mode configuration

AOI bit	6D bit	Interrupt mode
0	0	OR combination of axis requests
0	1	6-direction movement recognition
1	0	AND combination of axis requests
1	1	6-direction position recognition

Table 20: Interrupt behavior configuration

bit	Interrupt behavior
ZHIE	Enable interrupt request when Z output is higher than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
ZLIE	Enable interrupt request when Z output is lower than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
YHIE	Enable interrupt request when Y output is higher than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
YLIE	Enable interrupt request when Y output is lower than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
XHIE	Enable interrupt request when X output is higher than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
XLIE	Enable interrupt request when X output is lower than the threshold. Default value: 0 (0: disable interrupt request;1: enable interrupt request)

The X(Y,Z)HIE and X(Y,Z)LIE configuration has to be selected according to the target application: free-fall, wake-up, etc. Please note that enabling both X(Y,Z)HIE and X(Y,Z)LIE of same axis the interrupt request for that axis is always verified. Interrupt requests coming from all three axes are combined depending on the selected mode, in order to generate the interrupt condition. Whenever an interrupt condition is verified, the interrupt signal is generated and by reading the IG_SRC1(2) registers it is possible to understand which axis generated the request.

Table 21: IG_SRC1(2) register

-	IA	ZH	ZL	YH	YL	XH	XL

Table 22: IG_SRC1(2) description

IA	Interrupt Active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupt requests has been generated)
ZH	Z High. Default value: 0 (0: no request; 1: ZH interrupt request has occurred)
ZL	Z Low. Default value: 0 (0: no request; 1: ZL interrupt request has occurred)
YH	Y High. Default value: 0 (0: no request; 1: YH interrupt request has occurred)
YL	Y Low. Default value: 0 (0: no request; 1: YL interrupt request has occurred)
XH	X High. Default value: 0 (0: no request; 1: XH interrupt request has occurred)
XL	X Low. Default value: 0 (0: no request; 1: XL interrupt request has occurred)

6.1 Duration

The IG_DUR1(2) register content sets the minimum duration of the interrupt event to be recognized. Duration can also be represented as the number of consecutive samples that have to verify the interrupt condition before generating the interrupt signal. The duration counter starts to increase when the interrupt condition is verified and allows generating the interrupt only when the IG_DUR1(2) content value is reached; the steps and maximum time value depend on the selected ODR. In normal condition the duration counter is set to zero as soon as the interrupt condition is no longer verified unless the decrement function is enabled.

Duration time is measured in N/ODR [s], where N is the duration register content and ODR can assume the values: 10, 50, 100, 200, 400 and 800 Hz.

Table 23: Duration LSB value in normal mode

ODR (Hz)	Duration LSB value (ms)
10	100
50	20
100	10
200	5
400	2.5
800	1.25

The duration counter allows preventing an interrupt generation in the presence of a glitch bigger than the selected threshold.

6.1.1 Decrement function

The decrement functionality is enabled by setting the DCRM1(2) bit to “1” in CTRL7(26h) register. When this function is enabled, the interrupt pin is set to an inactive state as soon as the interrupt condition is no longer verified, but the duration counter is not set to zero immediately: it starts to decrease from the IG_DUR1(2) register value. If a new interrupt condition is verified before the duration counter reaches zero, it starts to increase again from the actual value and a new interrupt is generated when the duration counter reaches again the IG_DUR1(2) register value.

This function allows greater reactivity on consecutive interrupt conditions.

Figure 7: DCRM1(2) = 0

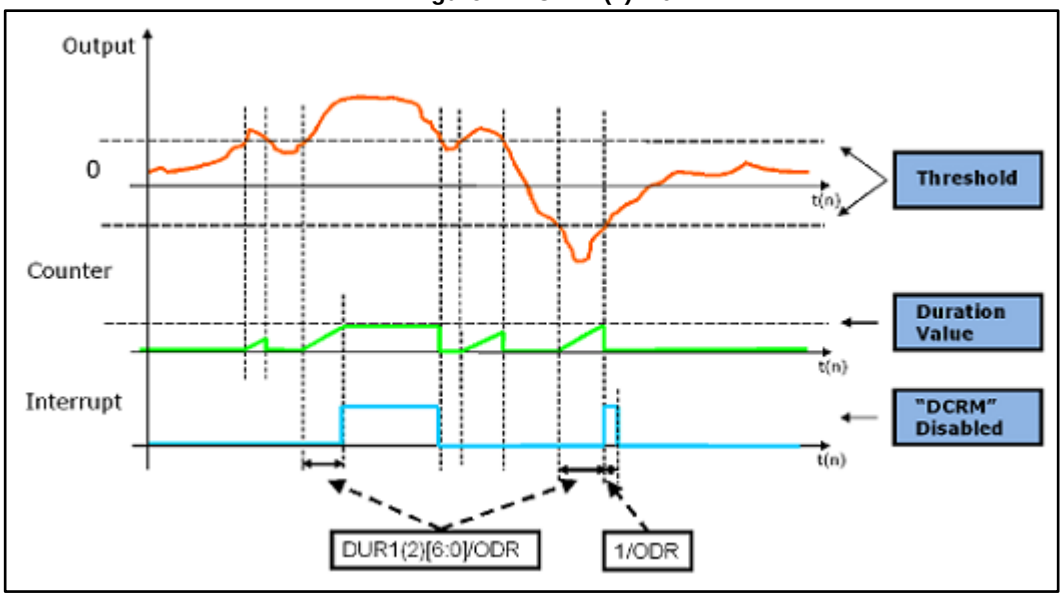
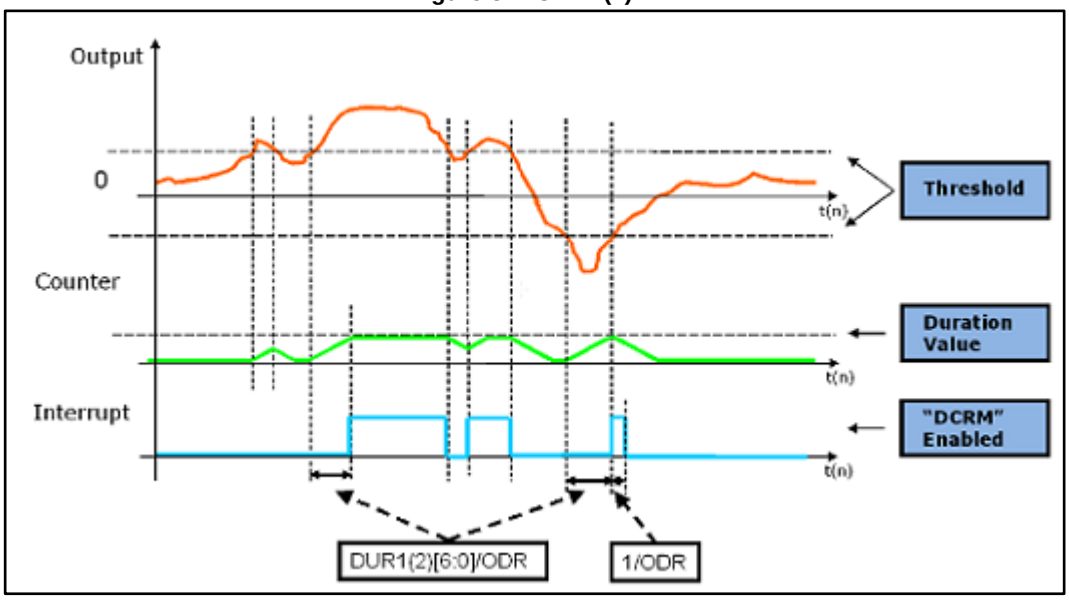


Figure 8: DCRM1(2) = 1

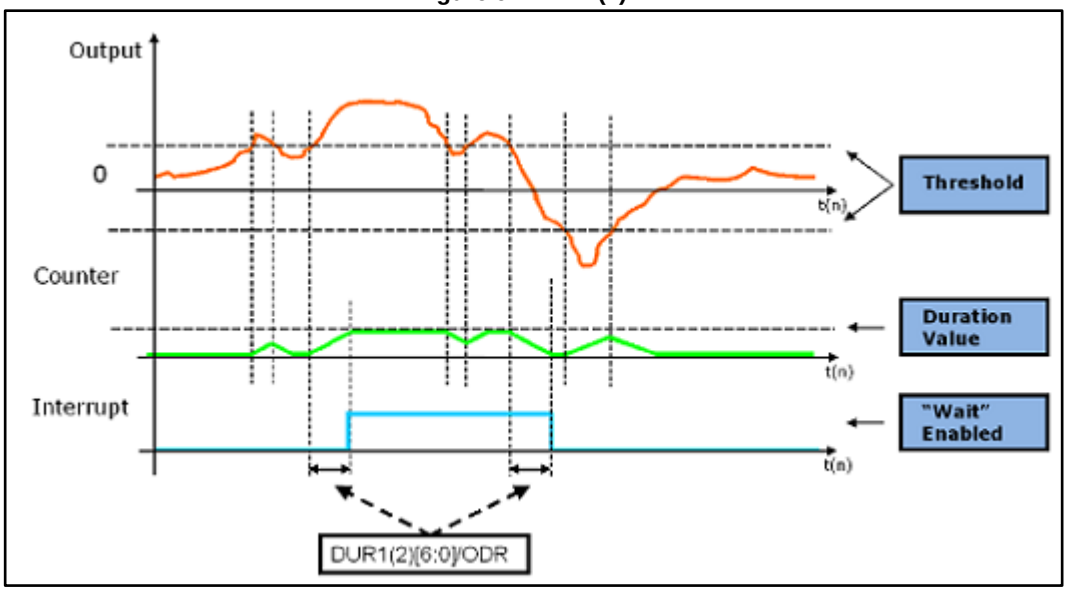


6.1.2 Wait function

The wait functionality is enabled by setting the WAIT1(2) bit to “1” in the IG_DUR1(2) register. When the interrupt condition is no longer verified, the duration counter is decremented one per sample. The interrupt pin signal remains active until the duration counter is different from zero (at least: DUR1(2)[6:0] / ODR [time]). If a new interrupt condition is verified and the duration counter is bigger than zero, the duration counter starts to increase again and the interrupt pin signal remains at the active state.

The wait function allows maintaining the interrupt in presence of a glitch lower than the selected threshold.

Figure 9: WAIT1(2) = 1



6.2 Threshold

The threshold registers define the reference accelerations used by the interrupt generator circuitry. The LIS2HH12 interrupt generator #1 allows selecting a different threshold for each axis while interrupt generator #2 supports the same threshold level for the X, Y and Z axis.

Registers used to configure this parameter are named IG_THS_X(Y,Z)1 for interrupt generator #1 and IG_THS2 for interrupt generator #2; the threshold LSB value is related to the selected full-scale and is equal to: FS/(2^8) (*Table 24: "Threshold LSB value"*).

Table 24: Threshold LSB value

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

6.3 Free-fall and wake-up interrupts

The LIS2HH12 interrupts signal can behave as free-fall and wake-up. All three device outputs are driven into a threshold comparator, but only the enabled conditions are evaluated. The ZHIE, ZLIE, YHIE, YLIE, XHIE, and XLIE bits of IG_CFG1(2) register allow deciding on which axis the interrupt decision must be performed and on which direction the threshold must be passed to generate the interrupt request. The interrupt signal is generated whenever the selected interrupt conditions are verified; the IG_SRC1(2) register has to be read to understand which condition happened.

The free-fall signal (FF) and wake-up signal (WU) interrupt generation block are represented in *Figure 10: "Free-fall, wake-up interrupt generator #1 diagram"* for interrupt generator #1 and in *Figure 11: "Free-fall, wake-up interrupt generator #2 diagram"* for interrupt generator #2.

The FF or WU interrupt generation can be selected through the AOI bit in the IG_CFG1(2) register. When the AOI bit is set to ‘0’, the selected interrupt conditions are combined in a

logical OR. In this case, the interrupt is generated when at least one of the enabled axes crosses the threshold written in IG_THS_X(Y,Z)1 for interrupt generator #1 and in IG_THS2 for interrupt generator #2. Otherwise, when the AOI bit is set to '1', the selected interrupt signals are placed in input in a logical AND. In this case, an interrupt signal is generated when all the enabled axes cross the threshold.

The LIR1(2) bit of CTRL7 register determines if the interrupt request must be latched or not. If the LIR1(2) bit is set to '0' (default value), the interrupt signal goes high when the interrupt condition is satisfied and returns to low immediately if the interrupt condition is no longer verified. Otherwise, if the LIR1(2) bit is set to '1', whenever an interrupt condition is applied, the interrupt signal remains high even if the condition returns to a non-interrupt status until a read of the IG_SRC1(2) register is performed.

The H_LACTIVE bit of CTRL5 register must be used to select the polarity of the interrupt pins. 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 CTRL5 allows changing the interrupt pins behavior 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.

Figure 10: Free-fall, wake-up interrupt generator #1 diagram

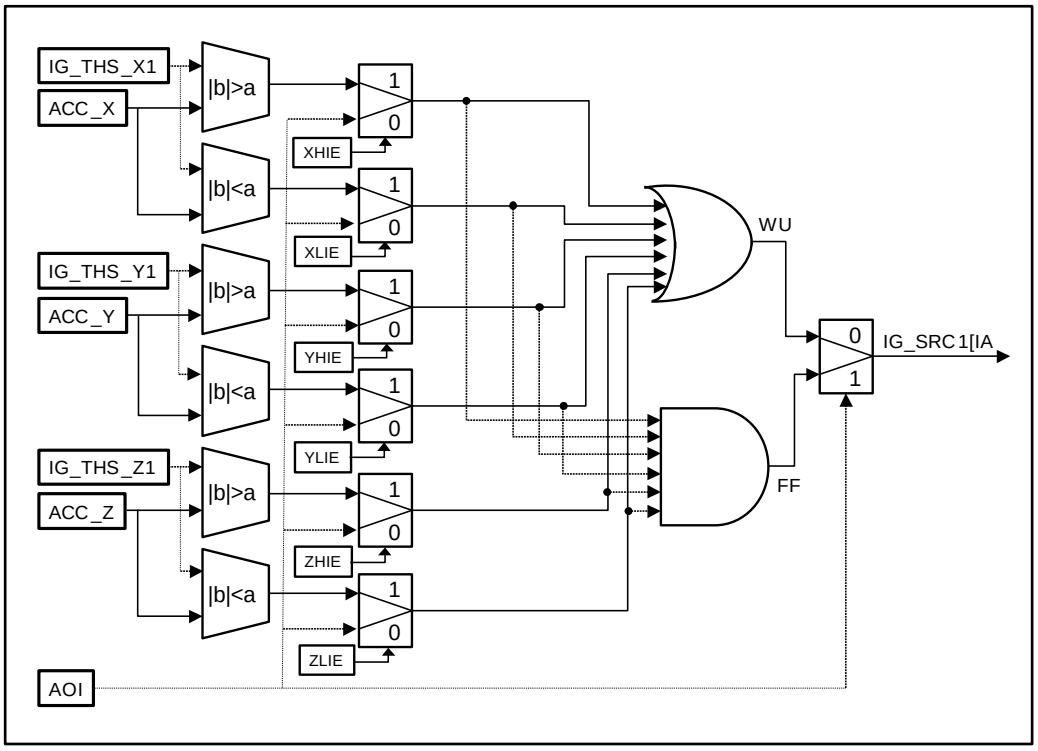
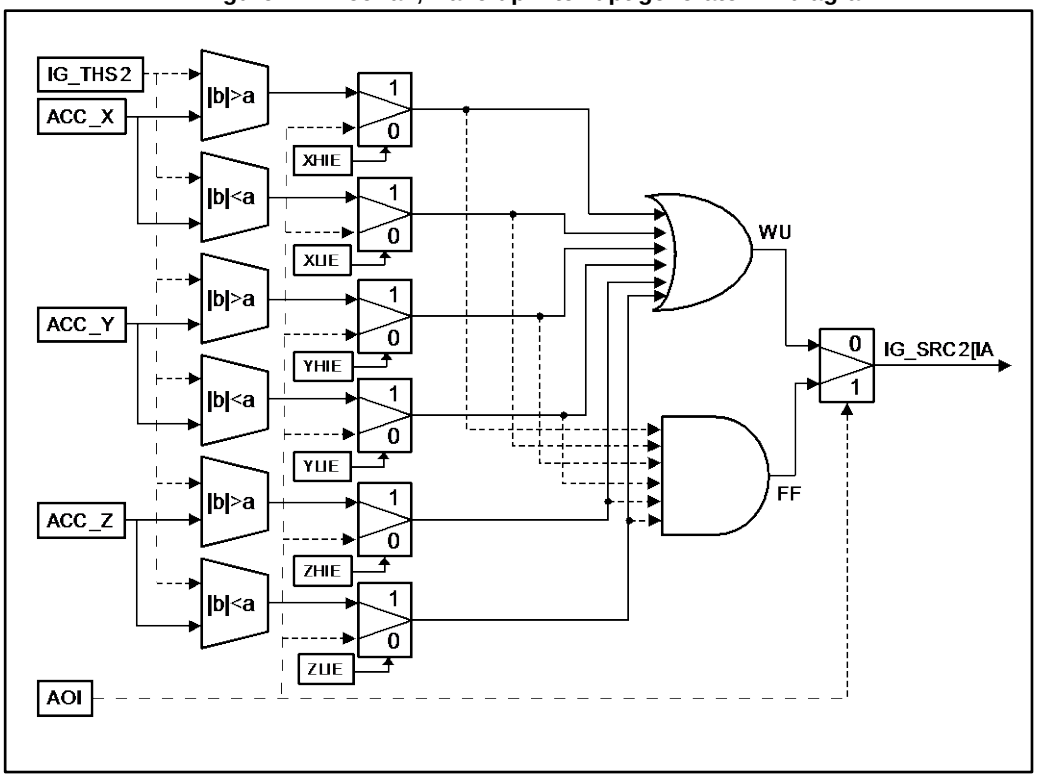


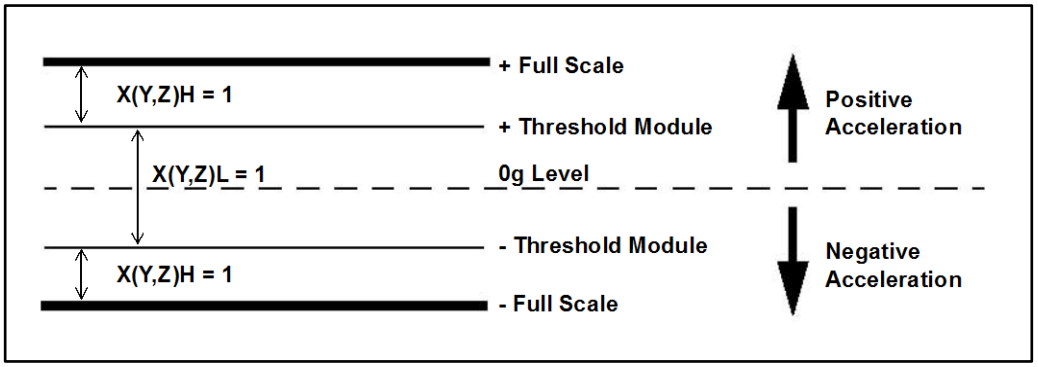
Figure 11: Free-fall, wake-up interrupt generator #2 diagram



The threshold modules which are used by the system to detect any free-fall or inertial wake-up event are defined by the IG_THS_X(Y,Z)1 and IG_THS2 registers. The threshold values are expressed over 8 bits as an unsigned number and are symmetrical around the zero-g level.

The XH(YH, ZH) bit of the IG_SRC1(2) register is true when the unsigned acceleration value of the X(Y, Z) channel is higher than IG_THS_X(Y,Z)1 for interrupt generator #1 and IG_THS2 for interrupt generator #2. Similarly, the XL(YL, ZL) bit is true when the unsigned acceleration value of the X(Y, Z) channel is lower than the selected threshold. Refer to *Figure 12: "FF_WU_CFG high and low"* for more details.

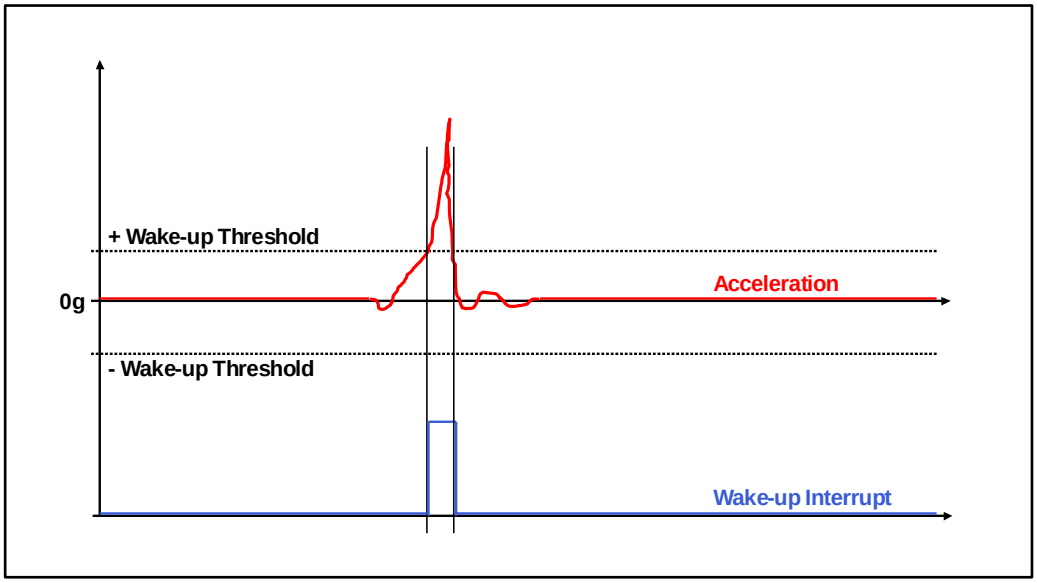
Figure 12: FF_WU_CFG high and low



6.3.1 Inertial wake-up

The wake-up interrupt refers to a specific configuration of the IG_SRC1(2) register that allows interrupt generation when the acceleration of one of the configured axis exceeds a defined threshold (*Figure 13: "Inertial wake-up interrupt"*).

Figure 13: Inertial wake-up interrupt



6.3.2 HP filter bypassed

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

0	-	// Place the device in flat position
1.	Write CTRL1 = 3Fh	// X, Y, Z, enabled, ODR = 100 Hz, BDU // enabled
2.	Write CTRL2 = 00h	// High-pass filter disabled
3.	Write CTRL3 = 08h	// Interrupt generator 1 on INT1 pin

4.	Write CTRL4 = 04h	// FS = 2g; Register address automatically
		// incremented during a multiple byte access with a
		// serial interface

5.	Write CTRL5 = 00h	// Interrupt active-high; Interrupt pins push-pull // configuration
6.	Write CTRL7 = 04h	// Interrupt 1 latched
7.	Write IG_THS_X(Y)1 = 20h	// Threshold = 250 mg [(2/256)*32 = 250 mg]
8.	Write IG_DUR1 = 00h	// No duration
9.	Write IG_CFG1 = 0Ah	// Enable XHIE and YHIE interrupt generation
10.	if (INT1 pin == 1) go to 11 else go to 10	// Waiting for wake-up event

11.	Read IG_SRC1	// Return the event that has triggered the interrupt // and clear the interrupt pin (latched)
12.	(Wake-up event has occurred; insert your code here)	// Event handling
13.	Go to 10	

6.3.3 Using the HP filter

The code provided below gives a basic routine which shows the practical use of the inertial wake-up feature performed on high-pass filtered data. In particular the device is configured to recognize when the high-frequency component of the acceleration applied along either the X, Y, or Z axis exceeds a preset threshold (250 mg used in this example).

The event which triggers the interrupt is latched inside the device and its occurrence is signaled through the use of the INT1 pin.

0	-	// Place the device in any orientation
1.	Write CTRL1 = 3Fh	// X, Y, Z, enabled, ODR = 100 Hz, BDU // enabled
2.	Write CTRL2 = 02h	// HP filter enabled for Interrupt Generator 1
3.	Write CTRL3 = 08h	// Interrupt generator 1 on INT1 pin

4.	Write CTRL4 = 04h	// FS = 2g; Register address automatically
		// incremented during a multiple byte access with a
		// serial interface

5.	Write CTRL5 = 00h	// Interrupt active high; Interrupt pins push-pull // configuration
6.	Write CTRL7 = 04h	// Interrupt 1 latched
7.	Write IG_THS_X(Y,Z)1 = 20h	// Threshold = 250 mg [(2/256)*32 = 250 mg]
8.	Write IG_DUR1 = 00h	// No duration

9.	Read registers:
	X(Y,Z)L_REFERENCE or
	X(Y,Z)H_REFERENCE
		// Dummy read to force immediately the HP filter output
		// to current acceleration value.
		// Allows to save the filter settling time

10.	Write IG_CFG1 = 2Ah	// Enable ZHIE, YHIE and XHIE interrupt generation
11.	if (INT1 pin == 1) go to 12 else go to 11	// Waiting for wake-up event
12.	Read IG_SRC1	// Return the event that has triggered the interrupt // and clear interrupt pin (latched)
13.	(Wake-up event has occurred; insert your code here)	// Event handling
14.	Go to 11	

At step 9, a dummy read of the X(Y,Z)L_REFERENCE or X(Y,Z)H_REFERENCE registers is performed to immediately force the HP filter output to the current acceleration/tilt state against which the device performed the threshold comparison.

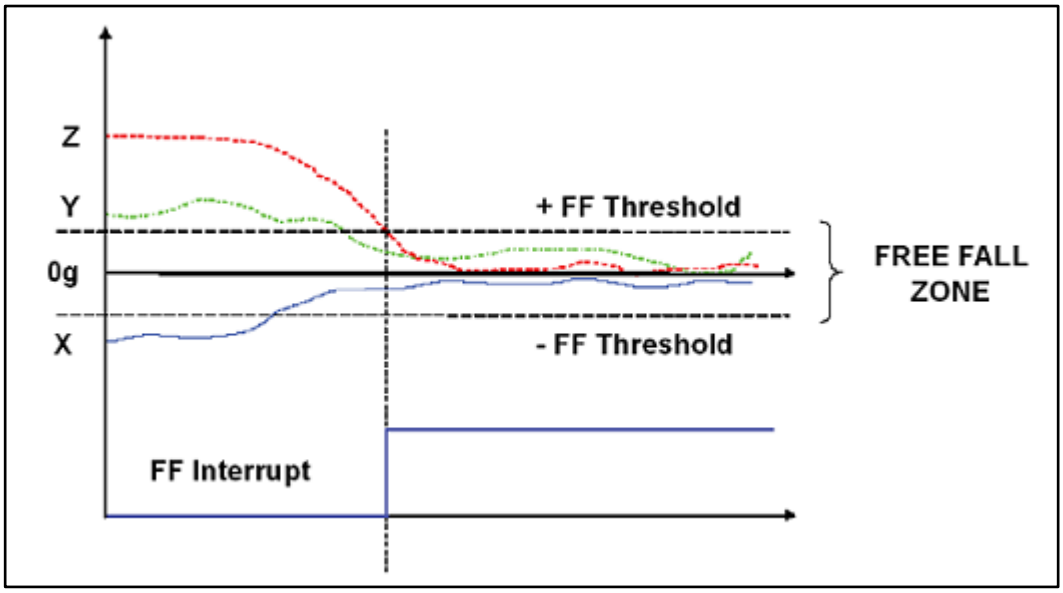
This reading may be performed any time it is required to remove the acceleration DC component without waiting for the settling time of the filter.

The HP filter is intended to be used when device orientation is unknown a priori and the acceleration gravity components on X, Y and Z have to be removed.

6.4 Free-fall detection

Free-fall detection refers to a specific configuration IG_SRC1(2) register 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 (*Figure 14: "Free-fall interrupt"*).

Figure 14: Free-fall interrupt



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

1.	Write CTRL1 = 3Fh	// X, Y, Z, enabled, ODR = 100 Hz, BDU // enabled
2.	Write CTRL2 = 00h	// High-pass filter disabled
3.	Write CTRL3 = 08h	// Interrupt generator 1 on INT1 pin
4.	Write CTRL4 = 04h	// FS = 2g; Register address automatically // incremented during a multiple byte access with a // serial interface
5.	Write CTRL5 = 00h	// Interrupt active high; Interrupt pins push-pull // configuration
6.	Write CTRL7 = 04h	// Interrupt 1 latched
7.	Write IG_THS_X(Y)1 = 2Dh	// Threshold = 352 mg [(2/256)*45 = 352 mg]
8.	Write IG_DUR1 = 03h	// Set three samples event Duration
9.	Write IG_CFG1 = 95h	// Enable logical AND comparison for XLIE, YLIE and // ZLIE: free-fall recognition
10.	if (INT1 pin == 1) go to 11 else go to 10	// Waiting for free-fall event
11.	Read IG_SRC1	// Return the event that has triggered the interrupt // and clear interrupt pin (latched)

12.	(Free-fall event has occurred; insert your code here)	// Event handling
13.	Go to 10	

The sample code exploits a threshold set to 352 mg for free-fall recognition and the event is notified by hardware through the INT1 pin. At step 8, the IG_DUR1 register is configured like this to ignore events that are shorter than 3/ODR = 3/100 Hz ~= 30 msec in order to avoid false detections.

Once the free-fall event has occurred, a read of the IG_SRC1 register clears the request and the device is ready to recognize other free-fall events.

7 6D/4D orientation detection

The LIS2HH12 provides advanced capability to detect the orientation of the device in space, enabling easy implementation of an energy-saving procedure and automatic image rotation for handheld devices.

7.1 6D orientation detection

Allows generating an Interrupt signal when one face of the device is stable in a known direction (almost flat): six orientations of the device in space can be detected. The 6D orientation direction function can be enabled through the AOI and 6D bits of the IG_CFG1(2) register, in detail ZHIE, ZLIE, YHIE, YLIE, XHIE and XLIE must be set to “1” in order to generate the interrupt in all the six possible directions. The 6D interrupt is asserted when only one axis exceeds the selected threshold and the accelerations measured from other two axes are lower than the threshold, while the interrupt is deasserted when more than one axis measures acceleration that exceeds the selected threshold.

When configured for 6D function, the ZH, ZL, YH, YL, XH, and XL bits of IG_SRC1(2) give information about the device orientation indicating which face is almost flat and in which direction (i.e. face up or face down). Only one of the six LSB of IG_SRC1(2) could be set to “1” when 6D detection is enabled: ZH or ZL or YH or YL or XH or XL.

In more detail:

- ZH (YH, XH) is equal to “1” 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 equal to “1” 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.

There are two possible configurations for the 6D function:

- 6D movement recognition: In this configuration the interrupt is generated when the device moves from a direction (known or unknown) to a different known direction. The interrupt signal is active only for 1/ODR [s] then it is automatically deasserted. This function is activated by setting the AOI-bit = 0 and 6D-bit = 1 in IG_CFG1(2) register. IG_CFG1(2) register must be set to 0x7F in order to generate the 6D movement interrupt in all six directions.
- 6D position recognition: In this configuration the interrupt is generated when the device is stable in one of the six known positions; the interrupt remains active as long as the position is maintained. This function is activated by setting the AOI-bit = 1 and
- 6D-bit = 1 in IG_CFG1(2) register. IG_CFG1(2) register must be set to 0xFF in order to generate the interrupt in all six positions.

Referring to *Figure 15: "6D movement vs. 6D position interrupt"*, the 6D movement line shows the behavior of the interrupt when the device is configured for 6D movement recognition on the X and Y positive axis (IG_CFG1(2) = 0x4A), while the 6D position line shows the behavior of the interrupt when the device is configured for 6D position recognition on the X and Y positive axis (IG_CFG1(2) = 0xCAh).

Referring to *Figure 16: "6D recognized positions"*, the device has been configured for 6D position function on the X, Y, and Z axis. *Table 25: "IG_SRC1(2) register in 6D position"* shows the content of the IG_SRC1(2) register for each position.

Figure 15: 6D movement vs. 6D position interrupt

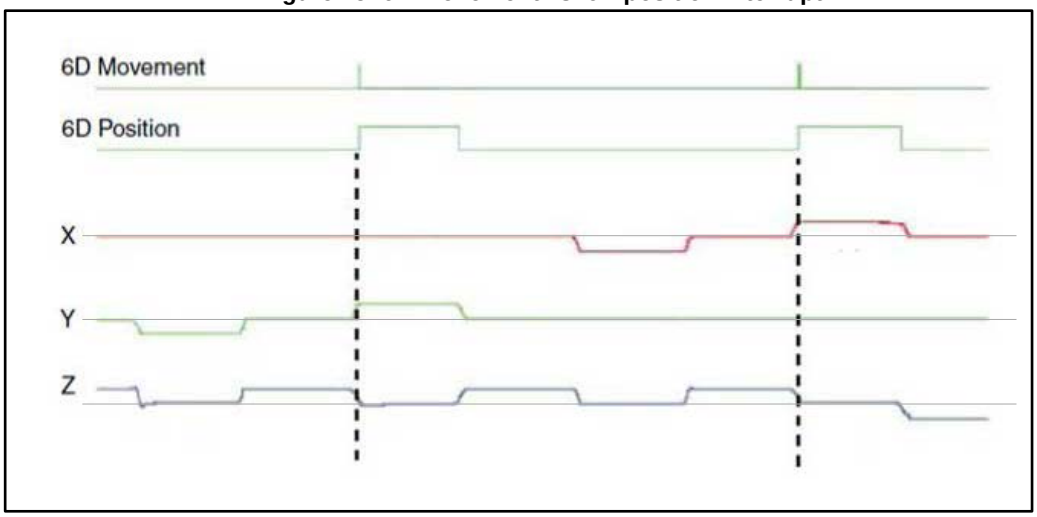


Figure 16: 6D recognized positions

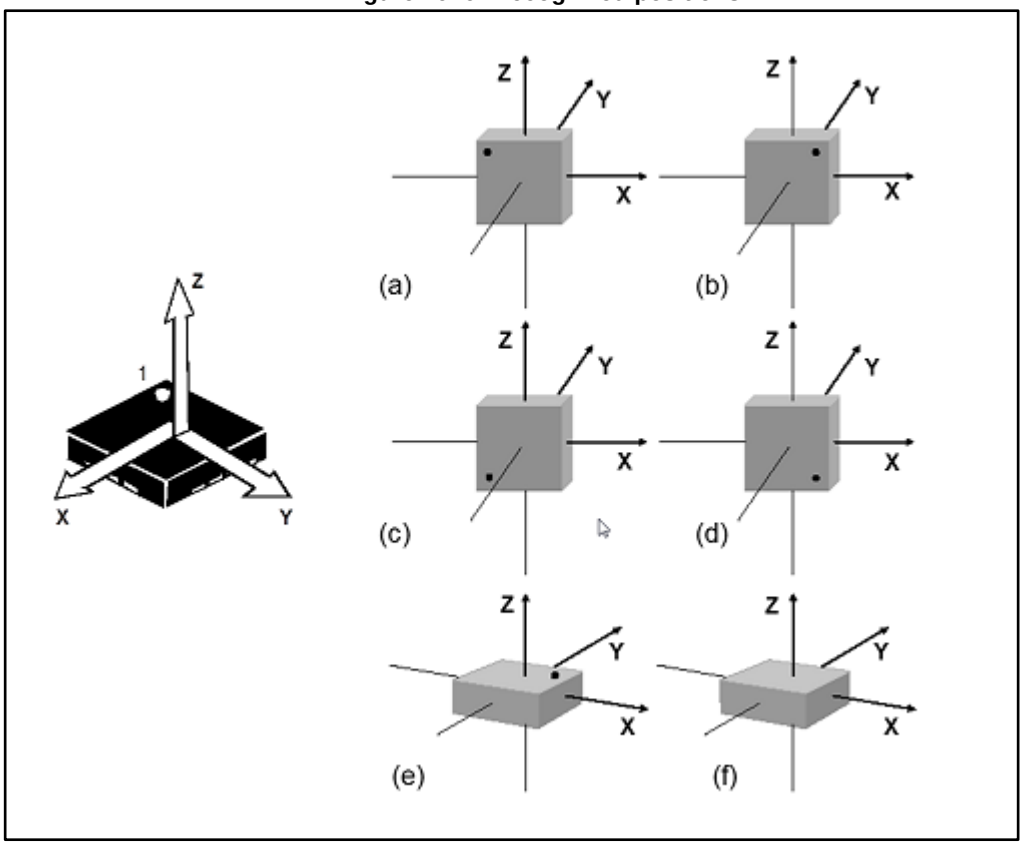


Table 25: IG_SRC1(2) register in 6D position

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

7.2 4D direction

Allows recognizing portrait and landscape conditions on X and Y also when the device is almost flat. When 4D direction is enabled, the Z output is not taken into account for portrait and landscape computation: YH, YL, XH, XL bits of IG_SRC1(2) are not related to the

Z-axis.

This mode can be enabled by setting the 4D_IG1(2) bit of CTRL7 to 1 when the 6D bit of IG_CFG1(2) is also set to 1. When 4D functionality is selected, ZHIE and ZLIE of IG_CFG1(2) register must be set to zero, otherwise an undesired interrupt could be generated. In this mode, the Z-axis position detection is disabled, therefore reducing position recognition to cases (a), (b), (c), and (d) of *Table 25: "IG_SRC1(2) register in 6D position"*.

1.	Write CTRL1 = 3Fh	// X, Y, Z, enabled, ODR = 100 Hz, BDU // enabled
2.	Write CTRL2 = 00h	// High-pass filter disabled
3.	Write CTRL3 = 08h	// Interrupt generator 1 on INT1 pin

4.	Write CTRL4 = 34h	// FS = 8g; Register address automatically
		// incremented during a multiple byte access with a
		// serial interface

5.	Write CTRL5 = 00h	// Interrupt active high; Interrupt pins push-pull // configuration
6.	Write CTRL7 = 01h	// Interrupt 1 not latched, 4D on INT1
7.	Write IG_THS_X(Y)1 = 0Ah	// X, Y Threshold = 10 * 31 mg = 310 mg
8.	Write IG_DUR1 = 00h	// 1° sample over threshold generates the Interrupt
9.	Write IG_CFG1 = CFh	// 6D Position, X and Y greater and lower than // threshold, ZHIE = ZLIE = 0

8 First-in first-out (FIFO) buffer

In order to limit intervention by the host processor and facilitate post-processing data for event recognition, the LIS2HH12 embeds a first-in first-out buffer (FIFO) for each of the three output channels, X, Y, and Z.

FIFO use allows consistent power saving for the system, since the host processor does not need to continuously poll data from the sensor, it can wake up only when needed and burst the significant data out from the FIFO.

The FIFO buffer can work according to six different modes that guarantee a high-level of flexibility during application development: Bypass mode, FIFO mode, Stream mode, Stream-to-FIFO mode, Bypass-to-Stream mode and Bypass-to-FIFO mode. Each mode is selected by the FMODE[2:0] bits in the FIFO_CTRL register.

Programmable FIFO threshold level, FIFO empty or FIFO overrun events are available in the FIFO_SRC register and can be set to generate a dedicated interrupt on the INT1 and INT2 pins.

8.1 FIFO description

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

The data sample set consists of 6 bytes (Xl, Xh, Yl, Yh, Zl, and Zh) which 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 values are overwritten.

Table 26: FIFO buffer full representation (32nd sample set stored)

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

Table 27: FIFO overrun representation (33rd sample set stored and 1st sample discarded)

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

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

When FIFO is enabled and the mode is different from Bypass, the LIS2HH12 output registers (28h to 2Dh) always contain the oldest FIFO sample set.

8.2 FIFO registers

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

8.2.1 CTRL3 register (0x22)

The FIFO_EN bit in CTRL3 register must be set to 1 in order to enable the internal first-in first-out buffer; when this bit is set, the accelerometer output registers (28h to 2Dh) don't contain the current acceleration value, but the oldest value stored in FIFO.

Table 28: CTRL3 register

FIFO_EN	STOP_FTH						

The STOP_FTH bit activates the FIFO threshold level feature, enabling this bit the physical FIFO dimension will be limited to the FTH[4:0] value of FIFO_CTRL register. If the STOP_FTH bit is set to '1', the FTH interrupt starts to perform as OVR interrupt.

8.2.3 FIFO_CTRL register (0x2E)

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

Table 29: FIFO_CTRL register

FMODE2	FMODE1	FMODE0	FTH4	FTH3	FTH2	FTH1	FTH0

FMODE[2:0] bits are dedicated to defining the FIFO buffer behavior selection:

FMODE[2:0] = (0,0,0): Bypass mode

FMODE[2:0] = (0,0,1): FIFO mode

FMODE[2:0] = (0,1,0): Stream mode

FMODE[2:0] = (0,1,1): Stream-to-FIFO mode

FMODE[2:0] = (1,0,0): Bypass-to-Stream mode

FMODE[2:0] = (1,0,1): Not Used

FMODE[2:0] = (1,1,0): Not Used

FMODE[2:0] = (1,1,1): Bypass-to-FIFO mode

FTH[4:0] bits are intended to define the threshold level; when the number of stored samples into FIFO exceeds this value the FTH bit is set to “1” in the FIFO_SRC register.

If STOP_FTH bit of CTRL3 register is set to ‘1’, the FTH[4:0] value also defines the FIFO phisical deep; in this case FTH interrupt start to perform as OVR interrupt.

8.2.4 FIFO_SRC register (0x2F)

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

Table 30: FIFO_SRC register

FTH	OVR	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0

The FTH bit is set to ‘1’ when FIFO content exceeds FTH[4:0] threshold level (FSS > FTH).

The OVR bit is set to ‘1’ when the FIFO buffer is full which means that the FIFO buffer contains 32 unread new samples and it is ready to be read. At the following ODR the new sample set replaces the oldest FIFO value. The OVR bit is reset to ‘0’ when the first sample set has been read.

The EMPTY flag is set high when all FIFO samples have been read and there aren’t unread samples in the buffer. When the EMPTY bit is equal to ‘1’, the latest sample set is already available in the FIFO and additional read operations continue to return the same value.

The FSS[4:0] field always contains the current number of unread samples stored in the FIFO buffer. When FIFO is enabled, this value increases at ODR frequency until the buffer is full, whereas, it decreases every time that one sample set is retrieved from FIFO.

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

Table 31: FIFO_SRC_REG startup behavior assuming FTH[4:0] = 15

FTH	OVRN	EMPTY	FSS[4:0]	Unread FIFO samples	Timing
0	0	1	00000	0	t0
0	0	0	00000	1	t0 + 1/ODR
0	0	0	00001	2	t0 + 2/ODR
0	0	0	00010	3	t0 + 3/ODR
...	...	...	...	...	...
0	0	0	01110	15	t0 + 15/ODR
1	0	0	01111	16	t0 + 16/ODR
...	...	...	...	...	...
1	0	0	11110	31	t0 + 31/ODR
1	1	0	11111	32	t0 + 32/ODR
1	1	0	11111	32 (oldest discarded)	t0 + 33/ODR

Table 32: FIFO_SRC_REG running behavior assuming FTH[4:0] = 15

FTH	OVRN	EMPTY	FSS[4:0]	Unread FIFO samples	Timing
0	0	1	00000	(#1 read sample)	t0
0	0	0	00001	2	t0 + 1/ODR
0	0	0	00010	3	t0 + 2/ODR
...	...	...	...	...	...
0	0	0	01110	15	t0 + 14/ODR
1	0	0	01111	16	t0 + 15ODR
...	...	...	...	...	...
1	0	0	11110	31	t0 + 30/ODR
1	0	0	11111	32	t0 + 31/ODR
1	1	0	11111	32 (read sample discarded)	t0 + 32/ODR

Condition OVR = 0 and FSS = 0 is reached only at startup because FIFO is really empty, in running mode the latest sample read still remains in the FIFO buffer.

WTM, OVR and EMPTY flags can be enabled to generate a dedicated interrupt on the INT1(2) pin by configuring bits INT1_OVR and INT1_FTH in CTRL3 register and bits INT2_EMPTY and INT2_FTH in CTRL6 register.

8.4 FIFO modes

The LIS2HH12 FIFO buffer can be configured to operate in six different modes selectable by the FMODE[2:0] field in the FIFO_CTRL register. Available configurations ensure a high level of flexibility and extend the number of functionalities usable in application development.

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

The FIFO_EN bit in CTRL3 register must be set to 1 in order to enable the internal first-in first-out buffer.

8.4.1 Bypass mode

In Bypass mode (FIFO_CTRL (FMODE [2:0])= 000), the FIFO is not operational and it remains empty.

Bypass mode is also used to reset the FIFO when in FIFO mode.

8.4.2 FIFO mode

In FIFO mode (FIFO_CTRL(FMODE [2:0])= 001) data from X, Y and Z channels are stored in the FIFO until it is full. The overrun interrupt can be enabled by CTRL3(INT1_OVR) = '1'. When the overrun interrupt occurs, FIFO is full and stops collecting data. Bypass mode has to be selected in FIFO_CTRL(FMODE [2:0]) in order to reset the FIFO content. After reset it is possible to restart FIFO mode by setting FIFO_CTRL(FMODE [2:0]) = 001.

The FIFO buffer can memorize 32 levels of X, Y and Z data, but the depth of the FIFO can be reduced using the CTRL3 (STOP_ FTH) bit. Setting the STOP_FTH bit = 1, FIFO depth is limited to FIFO_CTRL(FTH [4:0]).

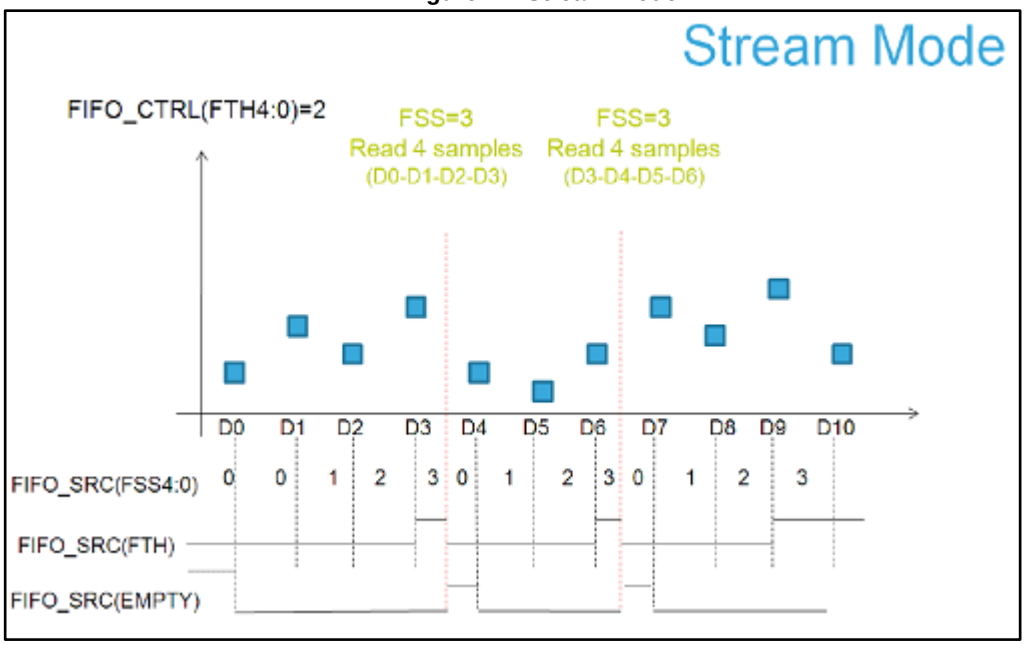
8.4.3 Stream mode

Stream mode (FIFO_CTRL(FMODE [2:0]) = 010) provides continuous FIFO update; when FIFO is full, as new data arrives the older is discarded.

The overrun interrupt can be enabled, CTRL3 (INT1_OVR)= '1', in order to read the whole FIFO content at once. In order to flush the FIFO, FIFO_SRC(FSS [4:0])+1 samples have to be read when the overrun interrupt is generated.

If data can't be lost and it is not possible to read at least one data sample set within one ODR period, a watermark interrupt can be enabled to partially read the FIFO and leave free memory slots for incoming data. In order to flush the FIFO, FIFO_SRC(FSS [4:0]) samples have to be read when the watermark interrupt is rising. In the latter case, if FIFO_SRC(FSS [4:0])+1 samples are retrieved on the watermark interrupt, the first sample read will be equal to the last sample of the previous burst (D3 in *Figure 17: "Stream mode"*).

Figure 17: Stream mode



Stream mode is intended to be used for reading all 32 samples of FIFO within an ODR after receiving an Overrun signal. However, also a watermark interrupt CTRL3 (INT1_FTH), CTRL6 (INT2_FTH) can be enabled in order to read data from the FIFO and leave free memory slots for incoming data.

8.4.4 Stream-to-FIFO mode

In Stream-to-FIFO mode (FIFO_CTRL(FMODE2:0) = 011), FIFO behavior changes according to the IG_SRC1 (IA) bit. When the IG_SRC1(IA) bit is equal to '1', FIFO operates in FIFO mode, when the IG_SRC1 (IA) bit is equal to '0', FIFO operates in Stream mode.

Interrupt generator 1 has to be properly set through the IG_CFG1, IGTHS_X1, IGTHS_Y1 and IGTHS_Z1 registers.

The CTRL7 (LIR1) bit should be set to '1' in order to latch the interrupt status and preserve the FIFO content. If the interrupt is not latched and the interrupt condition disappears, the FIFO behavior switches to Stream mode, overwriting the FIFO content.

8.4.5 Bypass-to-Stream mode

In Bypass-to-Stream mode (FIFO_CTRL (FMODE [2:0]) = 100) the FIFO buffer is reset

(By-pass mode) when G_SRC1 (IA) is equal to '0', it starts to collect data and to operate in Stream mode when G_SRC1 (IA) is equal to '1'.

Interrupt generator 1 has to be properly set through the IG_CFG1, IGTHS_X1, IGTHS_Y1 and IGTHS_Z1 registers.

The CTRL7 (LIR1) bit should be set to '1' in order to latch the interrupt status and preserve the FIFO content. If the interrupt is not latched and interrupt condition disappears, the FIFO behavior switches to Bypass and the content is reset.

8.4.7 Bypass-to-FIFO mode

In Bypass-to-FIFO mode (FIFO_CTRL (FMODE [2:0]) = 111), FIFO behavior changes according to IG_SRC1(IA) bit. When IG_SRC1(IA) bit is equal to '1' FIFO operates in FIFO-mode, when IG_SRC1(IA) bit is equal to '0' FIFO operates in Bypass mode (FIFO content reset). If an interrupt is generated FIFO starts collecting data until it is full.

Interrupt generator 1 has to be properly set through IG_CFG1, IGTHS_X1, IGTHS_Y1 and IGTHS_Z1 registers.

CTRL7 (LIR1) bit should be set to '1' in order to latch the interrupt status and preserve the FIFO content. If the interrupt is not latched and interrupt condition disappear, the FIFO behavior switch to bypass and content is reset.

8.5 Retrieving data from FIFO

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

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

The best way to perform data reading is to empty the FIFO buffer when the OVR bit is set to ‘1’, however the read can be performed at any time. The number of FIFO samples to read depends on the reading method:

- if OVR is equal to ‘1’, #FSS+1 samples have to be read. The empty bit has to become ‘1’.

- if OVR is equal to ‘0’, #FSS samples have to be read. The empty bit has to remain ‘0’.

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

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

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

In order to avoid loss of data, the right ODR must be selected according to the serial communication rate available.If standard I^2C mode is used (max rate

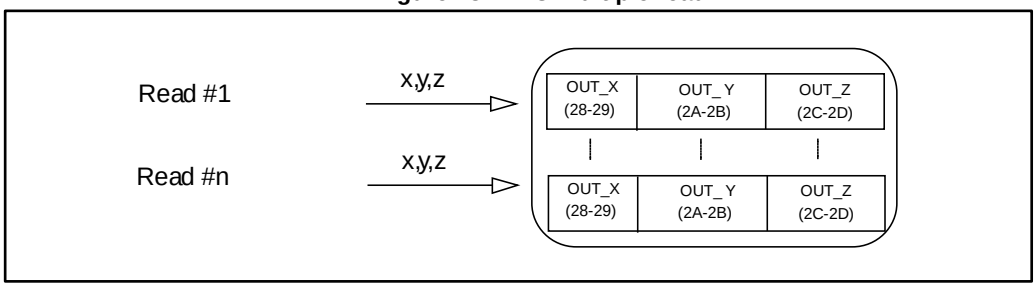
100 kHz) a single sample set reading takes 830 µs while total FIFO download is about

17.57 ms. The I^2C speed is lower than SPI and it needs about 29 clock pulses to start communication (Start, Slave Address, Device Address+Write, Restart, Device Address+Read) plus an additional 9 clock pulses for every byte to read. The complete FIFO read would be performed faster than 1*ODR which means that using standard I^2C, the selectable ODR must be lower than 57 Hz. If fast I^2C mode is used (max rate 400 kHz), the selectable ODR must be lower than 228 Hz.

8.5.2 FIFO multiple read (burst)

Starting from the address 0x28 a multiple read can be performed. Once the read reaches the address 0x2D the system automatically restarts from the address 0x28.

Figure 18: FIFO multiple read



9 Revision history

Table 33: Document revision history

Date	Revision	Changes
13-Apr-2015	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

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

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

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

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

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

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