

---

## IIS2DLPC: high-performance ultra-low-power 3-axis accelerometer for industrial applications

### Introduction

This document is intended to provide usage information and application hints related to ST's IIS2DLPC motion sensor.

The IIS2DLPC is a 3D digital accelerometer system-in-package with a digital I<sup>2</sup>C/SPI serial interface standard output and a dynamic user-selectable full-scale acceleration range of  $\pm 2/\pm 4/\pm 8/\pm 16$  g. The device is capable of measuring accelerations with output data rates from 1.6 Hz to 1600 Hz.

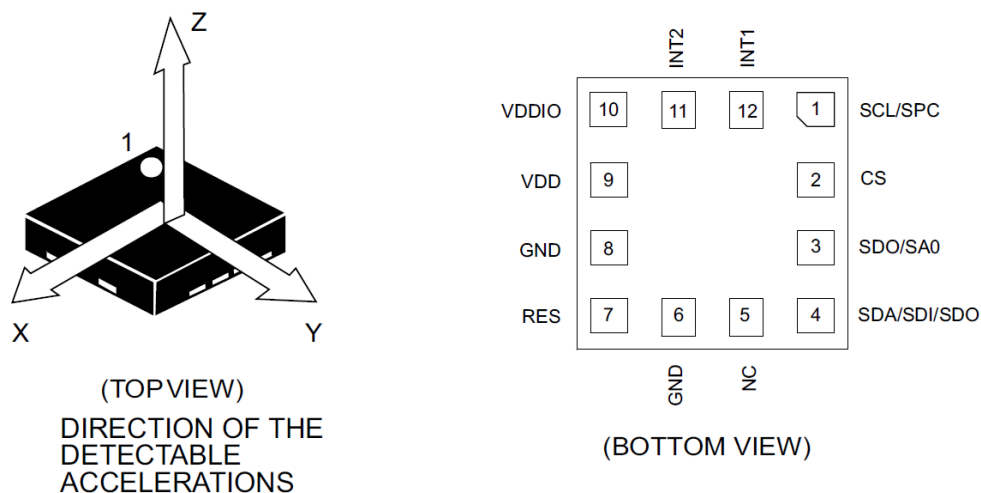
The IIS2DLPC has multiple operating modes which are reconfigurable on the fly and range from ultra-low-power mode (less than 0.4  $\mu$ A at 1.6 Hz ODR or single-shot) to high-performance, high-resolution mode (90  $\mu$ g/ $\sqrt{\text{Hz}}$ ) at 1.6 kHz ODR at 120  $\mu$ A). Ideal applications are industrial IoT and connected devices, anti-tampering devices for smart meters and portable healthcare devices.

The IIS2DLPC 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 and can be 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 IIS2DLPC 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.

# 1 Pin description

**Figure 1. Pin connections**



**Table 1. Internal pin status**

Pin #	Name	Function	Pin status
1	SCL SPC	I <sup>2</sup> C serial clock (SCL) SPI serial port clock (SPC)	Default: input without internal pull-up
2	CS	SPI enable I <sup>2</sup> C/SPI mode selection 1: SPI idle mode / I <sup>2</sup> C communication enabled 0: SPI communication mode / I <sup>2</sup> C disabled	Default: input with internal pull-up <sup>(1)</sup>
3	SDO SA0	Serial data output (SDO) I <sup>2</sup> C less significant bit of the device address (SA0)	Default: input with internal pull-up <sup>(2)</sup>
4	SDA SDI SDO	I <sup>2</sup> C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	Default: (SDA) input without internal pull-up
5	NC	Internally not connected. Can be tied to VDD, VDDIO, or GND.	
6	GND	0 V supply	
7	RES	Connect to GND	
8	GND	0 V supply	
9	VDD	Power supply	
10	VDD_IO	Power supply for I/O pins	
11	INT2	Interrupt pin 2 Clock input when selected in single data conversion on demand.	Default: push-pull output forced to ground
12	INT1	Interrupt pin 1	Default: push-pull output forced to ground

1. In order to disable the internal pull-up on the CS pin, write '1' to the CS\_PU\_DISC bit in CTRL2 (21h).
2. Internal pull-up on SDO/SA0 pin cannot be disabled: do not connect this pin to GND in low-power applications.

## 2 Registers



**Table 2. Registers**

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUT_T_L <sup>(1)</sup>	0Dh	TEMP3	TEMP2	TEMP1	TEMP0	0	0	0	0
OUT_T_H <sup>(1)</sup>	0Eh	TEMP11	TEMP10	TEMP9	TEMP8	TEMP7	TEMP6	TEMP5	TEMP4
WHO_AM_I <sup>(1)</sup>	0Fh	0	1	0	0	0	1	0	0
CTRL1	20h	ODR3	ODR2	ODR1	ODR0	MODE1	MODE0	LP_MODE1	LP_MODE0
CTRL2	21h	BOOT	SOFT_RESET	0	CS_PU_DISC	BDU	IF_ADD_INC	I2C_DISABLE	SIM
CTRL3	22h	ST2	ST1	PP_OD	LIR	H_LACTIVE	0	SLP_MODE_SEL	SLP_MODE_1
CTRL4_INT1_PAD_CTRL	23h	INT1_6D	INT1_SINGLE_TAP	INT1_WU	INT1_FF	INT1_TAP	INT1_DIFF5	INT1_FTH	INT1_DRDY
CTRL5_INT2_PAD_CTRL	24h	INT2_SLEEP_STATE	INT2_SLEEP_CHG	INT2_BOOT	INT2_DRDY_T	INT2_OVR	INT2_DIFF5	INT2_FTH	INT2_DRDY
CTRL6	25h	BW_FILT1	BW_FILT0	FS1	FS0	FDS	LOW_NOISE	0	0
OUT_T <sup>(1)</sup>	26h	TEMP7	TEMP6	TEMP5	TEMP4	TEMP3	TEMP2	TEMP1	TEMP0
STATUS <sup>(1)</sup>	27h	FIFO_THS	WU_IA	SLEEP_STATE	DOUBLE_TAP	SINGLE_TAP	6D_IA	FF_IA	DRDY
OUT_X_L <sup>(1)</sup>	28h	X_L7	X_L6	X_L5	X_L4	X_L3 <sup>(2)</sup>	X_L2 <sup>(2)</sup>	0	0
OUT_X_H <sup>(1)</sup>	29h	X_H7	X_H6	X_H5	X_H4	X_H3	X_H2	X_H1	X_H0
OUT_Y_L <sup>(1)</sup>	2Ah	Y_L7	Y_L6	Y_L5	Y_L4	Y_L3 <sup>(2)</sup>	Y_L2 <sup>(2)</sup>	0	0
OUT_Y_H <sup>(1)</sup>	2Bh	Y_H7	Y_H6	Y_H5	Y_H4	Y_H3	Y_H2	Y_H1	Y_H0
OUT_Z_L <sup>(1)</sup>	2Ch	Z_L7	Z_L6	Z_L5	Z_L4	Z_L3 <sup>(2)</sup>	Z_L2 <sup>(2)</sup>	0	0
OUT_Z_H <sup>(1)</sup>	2Dh	Z_H7	Z_H6	Z_H5	Z_H4	Z_H3	Z_H2	Z_H1	Z_H0
FIFO_CTRL	2Eh	FMode2	FMode1	FMode0	FTH4	FTH3	FTH2	FTH1	FTH0
FIFO_SAMPLES <sup>(1)</sup>	2Fh	FIFO_FTH	FIFO_OVR	Diff5	Diff4	Diff3	Diff2	Diff1	Diff0
TAP_THS_X	30h	4D_EN	6D_THS1	6D_THS0	TAP_THSX_4	TAP_THSX_3	TAP_THSX_2	TAP_THSX_1	TAP_THSX_0
TAP_THS_Y	31h	TAP_PRIOR_2	TAP_PRIOR_1	TAP_PRIOR_0	TAP_THSY_4	TAP_THSY_3	TAP_THSY_2	TAP_THSY_1	TAP_THSY_0
TAP_THS_Z	32h	TAP_X_EN	TAP_Y_EN	TAP_Z_EN	TAP_THSZ_4	TAP_THSZ_3	TAP_THSZ_2	TAP_THSZ_1	TAP_THSZ_0
INT_DUR	33h	LATENCY3	LATENCY2	LATENCY1	LATENCY0	QUIET1	QUIET0	SHOCK1	SHOCK0
WAKE_UP_THS	34h	SINGLE_DOUBLE_TAP	SLEEP_ON	WK_THS5	WK_THS4	WK_THS3	WK_THS 2	WK_THS 1	WK_THS 0
WAKE_UP_DUR	35h	FF_DUR5	WAKE_DUR1	WAKE_DUR0	STATIONARY	SLEEP_DUR3	SLEEP_DUR2	SLEEP_DUR1	SLEEP_DUR0

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FREE_FALL	36h	FF_DUR4	FF_DUR3	FF_DUR2	FF_DUR1	FF_DUR0	FF_THS2	FF_THS1	FF_THS0
STATUS_DUP <sup>(1)</sup>	37h	OVR	DRDY_T	SLEEP_STATE_IA	DOUBLE_TAP	SINGLE_TAP	6D_IA	FF_IA	DRDY
WAKE_UP_SRC <sup>(1)</sup>	38h	0	0	FF_IA	SLEEP_STATE_IA	WU_IA	X_WU	Y_WU	Z_WU
TAP_SRC <sup>(1)</sup>	39h	0	TAP_IA	SINGLE_TAP	DOUBLE_TAP	TAP_SIGN	X_TAP	Y_TAP	Z_TAP
SIXD_SRC <sup>(1)</sup>	3Ah	0	6D_IA	ZH	ZL	YH	YL	XH	XL
ALL_INT_SRC <sup>(1)</sup>	3Bh	0	0	SLEEP_CHANGE_IA	6D_IA	DOUBLE_TAP	SINGLE_TAP	WU_IA	FF_IA
X_OFS_USR	3Ch	X_OFS_USR_7	X_OFS_USR_6	X_OFS_USR_5	X_OFS_USR_4	X_OFS_USR_3	X_OFS_USR_2	X_OFS_USR_1	X_OFS_USR_0
Y_OFS_USR	3Dh	Y_OFS_USR_7	Y_OFS_USR_6	Y_OFS_USR_5	Y_OFS_USR_4	Y_OFS_USR_3	Y_OFS_USR_2	Y_OFS_USR_1	Y_OFS_USR_0
Z_OFS_USR	3Eh	Z_OFS_USR_7	Z_OFS_USR_6	Z_OFS_USR_5	Z_OFS_USR_4	Z_OFS_USR_3	Z_OFS_USR_2	Z_OFS_USR_1	Z_OFS_USR_0
CTRL7	3Fh	DRDY_PULSED	INT2_ON_INT1	INTERRUPTS_ENABLE	USR_OFF_ON_OUT	USR_OFF_ON_WU	USR_OFF_W	HP_REF_MODE	LPASS_ON6D

1. Read-only register
2. If Low-Power Mode 1 is enabled, this bit is set to 0.

## 3 Operating modes

### 3.1 Power mode

Five sets of operating modes have been designed to offer the customer a broad choice of noise/power-consumption combinations:

- 1 High-Performance Mode: focus on low noise
- 4 Low-Power Modes: trade-off between noise and power consumption

**Table 3. Accelerometer resolution**

High-Performance Mode	Low-Power Mode 4	Low-Power Mode 3	Low-Power Mode 2	Low-Power Mode 1
14-bit	14-bit	14-bit	14-bit	12-bit

These operating modes are selected by writing the MODE[1:0] and LP\_MODE[1:0] bits in CTRL1 (20h) shown in the table below.

**Table 4. CTRL1 register**

b7	b6	b5	b4	b3	b2	b1	b0
ODR3	ODR2	ODR1	ODR0	MODE1	MODE0	LP_MODE1	LP_MODE0

**Table 5. Mode selection**

MODE[1:0]	Mode and resolution
00	Low-Power Mode (12/14-bit resolution)
01	High-Performance Mode (14-bit resolution)
10	Single data conversion on-demand mode (12/14-bit resolution)
11	Not allowed

**Table 6. Low-power mode selection**

LP_MODE[1:0]	Mode and resolution
00	Low-Power Mode 1 (12-bit resolution)
01	Low-Power Mode 2 (14-bit resolution)
10	Low-Power Mode 3 (14-bit resolution)
11	Low-Power Mode 4 (14-bit resolution)

From each of these five sets, two configurations have been designed:

- Very low-power(low-noise off)
- Low-noise

Writing the LOW\_NOISE bit in CTRL6 (25h) selects the desired configuration. The LOW\_NOISE bit in CTRL6 (25h) impacts front-end noise and current consumption. Bandwidths and settling time are not impacted.

Table 7 shows the typical values of power consumption for the different operating modes.

**Table 7. Power consumption [ $\mu$ A]**

Output data rate	Power supply	High-Performance		Low-Power Mode 4		Low-Power Mode 3		Low-Power Mode 2		Low-Power Mode 1	
		LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE	LOW_NOISE
		0	1	0	1	0	1	0	1	0	1
1.6 Hz	1.8 V	-	-	0.65	0.7	0.55	0.6	0.45	0.5	0.38	0.4
12.5 Hz	1.8 V	90	120	4	5	2.5	3	1.6	2	1	1.1
25 Hz	1.8 V	90	120	8.5	10	4.5	6	3	3.5	1.5	2
50 Hz	1.8 V	90	120	16	20	9	11	5.5	7	3	3.5
100 Hz	1.8 V	90	120	32	39	17	21.5	10.5	13	5	6
200 Hz	1.8 V	90	120	63	77	34	42	20.5	25	10	12
400/800/1600 Hz	1.8 V	90	120	-	-	-	-	-	-	-	-
1.6 Hz	3.0 V	-	-	1.3	1.35	0.95	1	0.75	0.8	0.67	0.7
12.5 Hz	3.0 V	110	140	5.3	7	3	4	2	2.5	1.3	1.5
25 Hz	3.0 V	110	140	10.5	12.5	6	7	3.8	4.5	2.1	2.5
50 Hz	3.0 V	110	140	20.5	24.5	11.5	14	7	8.5	3.7	4.5
100 Hz	3.0 V	110	140	40	48.5	22	26.5	13.5	16	6.5	8
200 Hz	3.0 V	110	140	80	95.5	44	52.5	26	31	12.5	14.5
400/800/1600 Hz	3.0 V	110	140	-	-	-	-	-	-	-	-

Table 8 and Table 9 show the typical noise values for the different operating modes.

**Table 8. High-Performance Mode: noise density [ $\mu$ g/ $\sqrt{\text{Hz}}$ ]**

Full scale	LOW_NOISE	
	0	1
$\pm 2 g$	110	90
$\pm 4 g$	110	100
$\pm 8 g$	130	120
$\pm 16 g$	170	160

*Note: In High-Performance Mode the noise density is the same for all ODRs.*

**Table 9. Low-Power Mode: RMS noise [mg(RMS)]**

Full scale	Low-Power Mode 4		Low-Power Mode 3		Low-Power Mode 2		Low-Power Mode 1	
	0	1	0	1	0	1	0	1
	LOW_NOISE		LOW_NOISE		LOW_NOISE		LOW_NOISE	
$\pm 2\text{ g}$	1.6	1.3	2.1	1.8	3.0	2.4	5.5	4.5
$\pm 4\text{ g}$	1.7	1.4	2.3	1.9	3.2	2.7	6.5	5.4
$\pm 8\text{ g}$	1.7	1.5	2.4	2.1	3.3	2.8	6.8	5.8
$\pm 16\text{ g}$	2.0	1.8	2.7	2.4	3.7	3.3	7.7	7.0

*Note: In Low-Power Mode the RMS noise is the same for all ODRs.*

## 3.2 Continuous conversion

When bits MODE[1:0] in CTRL1 (20h) are set to Low-Power mode (00b) or High-Performance mode (01b), the device is in continuous conversion and the output data rate can be selected through the ODR[3:0] bits in CTRL1 (20h).

**Table 10. Output data rate selection**

ODR[3:0]	Mode and resolution
0000	Power-down
0001	High-Performance 12.5 Hz / Low-Power mode 1.6 Hz
0010	12.5 Hz (independent of power mode)
0011	25 Hz (independent of power mode)
0100	50 Hz (independent of power mode)
0101	100 Hz (independent of power mode)
0110	200 Hz (independent of power mode)
0111	High-Performance 400 Hz / Low-Power mode 200 Hz
1000	High-Performance 800 Hz / Low-Power mode 200 Hz
1001	High-Performance 1600 Hz / Low-Power mode 200 Hz



### 3.3 Single data conversion (on-demand mode)

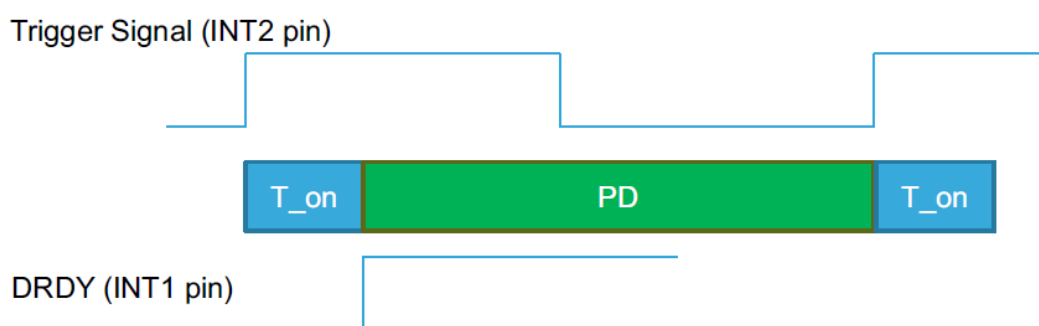
This mode is available only for low-power modes and it is enabled by writing the MODE[1:0] bits to '10' in CTRL1 (20h).

In this configuration the device waits for a trigger signal in order to generate new data according to the selected power mode LP\_MODE[1:0] bits in CTRL1 (20h), after that the device immediately enters power-down.

The trigger can be:

- **A rising edge on the INT2 pin (if SLP\_MODE\_SEL = '0' in register CTRL3 (22h)).** In this case the user can detect the end of the conversion using the DRDY bit of the STATUS register (27h) that can also be routed to the INT1 pin by setting the INT1\_DRDY bit to 1 in register CTRL4\_INT1\_PAD\_CTRL (23h). Minimum duration of trigger signal high level is 20 ns.
- **A write of SLP\_MODE\_1 to '1' in register CTRL3 (22h) (if SLP\_MODE\_SEL = '1' in register CTRL3 (22h)).** In this case, the user can detect the end of the conversion using the DRDY bit/signal as in the previous case, or by checking when the SLP\_MODE\_1 bit in register CTRL3 (22h) is automatically cleared.

**Figure 2. Single data conversion using INT2 as external trigger (SLP\_MODE\_SEL = 0)**



The maximum data rate using single data conversion mode is 200 Hz and the time of conversion depends on the low-power mode selected (refer to the following table).

**Table 11. Low-power mode selection**

Low-power	Typical time of conversion (T_on)
Mode 1	1.20 ms
Mode 2	1.70 ms
Mode 3	2.30 ms
Mode 4	3.55 ms

*Note: If the ODR[3:0] bits of the CTRL1 register are set to 0000b, the accelerometer is permanently configured in Power-down mode and no conversion can be triggered. When the single data conversion mode has to be used, the ODR[3:0] bits of the CTRL1 register must be different than 0000b.*

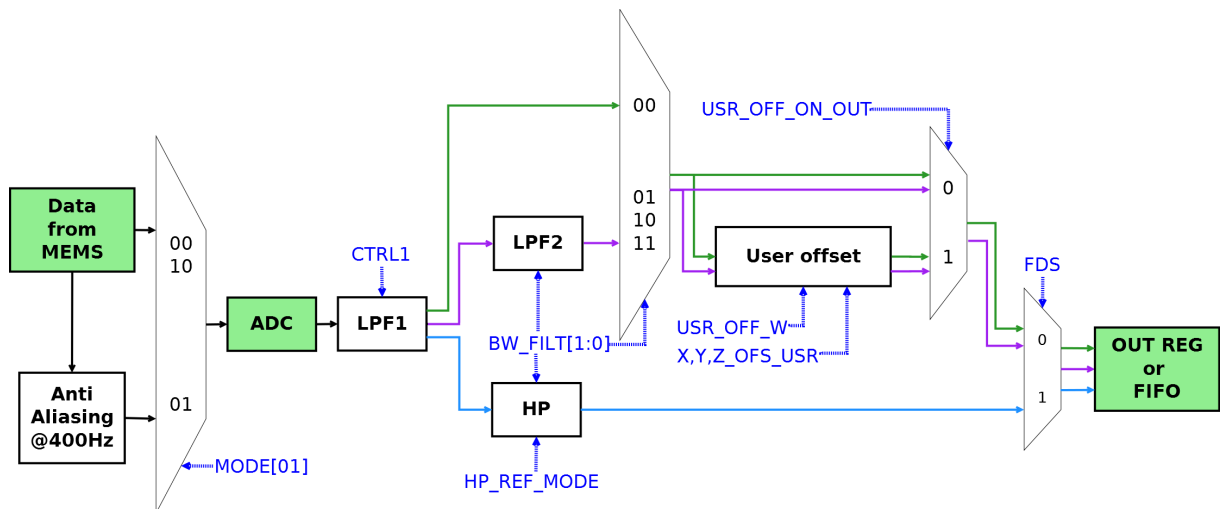
Interrupts, embedded features and FIFO are still supported when using single data conversion mode. Also the embedded filters LPF1, LPF2 and HP are available in single data conversion (on-demand mode) with the same bandwidth and settling time of the selected low-power mode (see [Section 3.4 Accelerometer bandwidth](#) for details).

### 3.4 Accelerometer bandwidth

The accelerometer sampling chain (Figure 3. Accelerometer filtering chain diagram) is represented by a cascade of a few blocks:

- ADC: Analog-to-digital converter
- Anti-Aliasing Filter: available only in High-Performance Mode (MODE[1:0] = 01) with a cutoff frequency of 400 Hz
- LPF1(2): low-pass filter 1(2)
- HP: high-pass filter
- User offset: configurable values that are subtracted from the sampled data (one for each axis)

Figure 3. Accelerometer filtering chain diagram



As shown in the figure above, data can be generated using three different filter paths:

- only LPF1 (green path) : in order to select this path set BW\_FILT[1:0] = 00 and FDS = 0. Additional details in [Table 12. Low-pass filter 1 bandwidth](#).
- LPF1 + LPF2 (purple path) : in order to select this path set BW\_FILT[1:0] to a value different from 00 and FDS = 0. Additional details in [Table 13. Bandwidth: low-pass path](#).
- LPF1 + HP (blue path): these outputs are available by setting FDS = 1. Additional details in [Table 14. Bandwidth: high-pass path](#).

**Table 12. Low-pass filter 1 bandwidth**

Mode	ODR selection	BW_FILTER[1:0] = 00	
		Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]
Low-Power Mode 4	@ each ODR	0	180
Low-Power Mode 3	@ each ODR	0	360
Low-Power Mode 2	@ each ODR	0	720
Low-Power Mode 1	@ each ODR	0	3200
High-Performance	@12.5 Hz	0	ODR/2
High-Performance	@25 Hz	0	ODR/2
High-Performance	@50 Hz	0	ODR/2
High-Performance	@100 Hz	1	ODR/2
High-Performance	@200 Hz	1	ODR/2
High-Performance	@400 Hz	1	ODR/2
High-Performance	@800 Hz	1	ODR/2
High-Performance	@1600 Hz	2	400

1. The starting condition of ODR[3:0], MODE[1:0], LP\_MODE[1:0] and BW\_FILTER[1:0] do not impact these values. The turn-on time (first sample available starting from power-down condition) is 1 / ODR.

**Table 13. Bandwidth: low-pass path**

Mode	ODR selection	BW_FILTER[1:0] = 01		BW_FILTER[1:0] = 10		BW_FILTER[1:0] = 11	
		Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]	Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]	Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]
LP Mode 4	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 3	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 2	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 1	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
HP	@12.5 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@25 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@50 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@100 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@200 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@400 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@800 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@1600 Hz	3	ODR/4	6	ODR/10	12	ODR/20

1. The starting condition of ODR[3:0], MODE[1:0], LP\_MODE[1:0] and BW\_FILTER[1:0] do not impact these values.

**Table 14. Bandwidth: high-pass path**

Mode	ODR selection	BW_FILT[1:0] = 01 / 00		BW_FILT[1:0] = 10		BW_FILT[1:0] = 11	
		Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]	Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]	Samples to discard <sup>(1)</sup> Settling @95%	Cutoff [Hz]
LP Mode 4	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 3	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 2	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
LP Mode 1	@ each ODR	1	ODR/4	5	ODR/10	11	ODR/20
HP	@12.5 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@25 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@50 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@100 Hz	1	ODR/4	5	ODR/10	11	ODR/20
HP	@200 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@400 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@800 Hz	2	ODR/4	5	ODR/10	11	ODR/20
HP	@1600 Hz	3	ODR/4	6	ODR/10	12	ODR/20

1. The starting condition of ODR[3:0], MODE[1:0], LP\_MODE[1:0] and BW\_FILT[1:0] do not impact these values.

Setting USR\_OFF\_ON\_OUT = 1 in CTRL7 does not change the bandwidth of the system. In this configuration, the values written in registers X\_OFS\_USR, Y\_OFS\_USR, Z\_OFS\_USR are subtracted from the respective axis. The offset values are signed values (two's complement).

The weight of the bits in registers X\_OFS\_USR, Y\_OFS\_USR, Z\_OFS\_USR is defined through the USR\_OFF\_W bit in CTRL7.

### 3.5 High-pass filter configuration

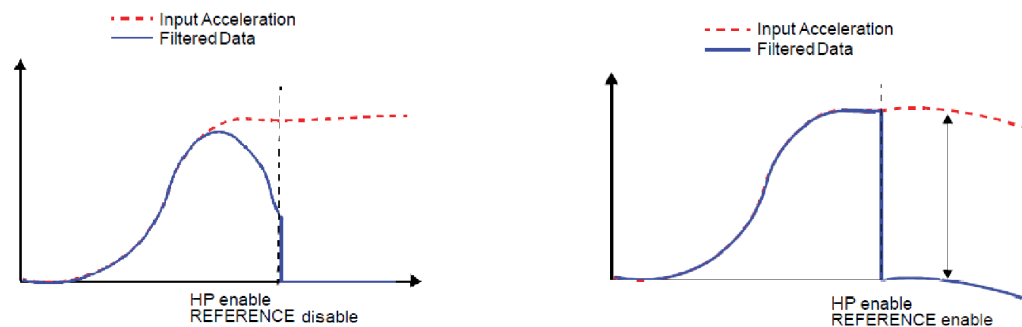
The IIS2DLPC provides an embedded high-pass filtering capability to easily delete the DC component of the measured acceleration. As shown in [Figure 3. Accelerometer filtering chain diagram](#), through the FDS bit in register CTRL6 the user can route the filter outputs to the output registers.

It is also possible to independently apply the filter to the embedded function data ([Figure 9. Embedded functions in Section 5 Interrupt generation and embedded functions](#)). This means that it is possible to get filtered data while the interrupt generation works on unfiltered data.

The high-pass filter can be configured in *reference mode* by setting the HP\_REF\_MODE bit in the CTRL7 register to 1. In this configuration the output data is calculated as the difference between the measured acceleration and the output values captured when *reference mode* was enabled. In this way only the difference is applied without any filtering.

As an example, this feature can be combined with the wake-up functionality described in [Section 5.4](#) in order to detect when the device is displaced with respect to a specific orientation, i.e. the orientation of the device when the HP\_REF\_MODE bit was set to 1. When the output acceleration exceeds the wake-up threshold defined by the WK\_THS[5:0] bits in the WAKE\_UP\_THS register for a duration longer than the one defined by the WAKE\_DUR[1:0] bits in the WAKE\_UP\_DUR register, an interrupt is generated. If the device is moved back to the original reference orientation, the interrupt is deactivated.

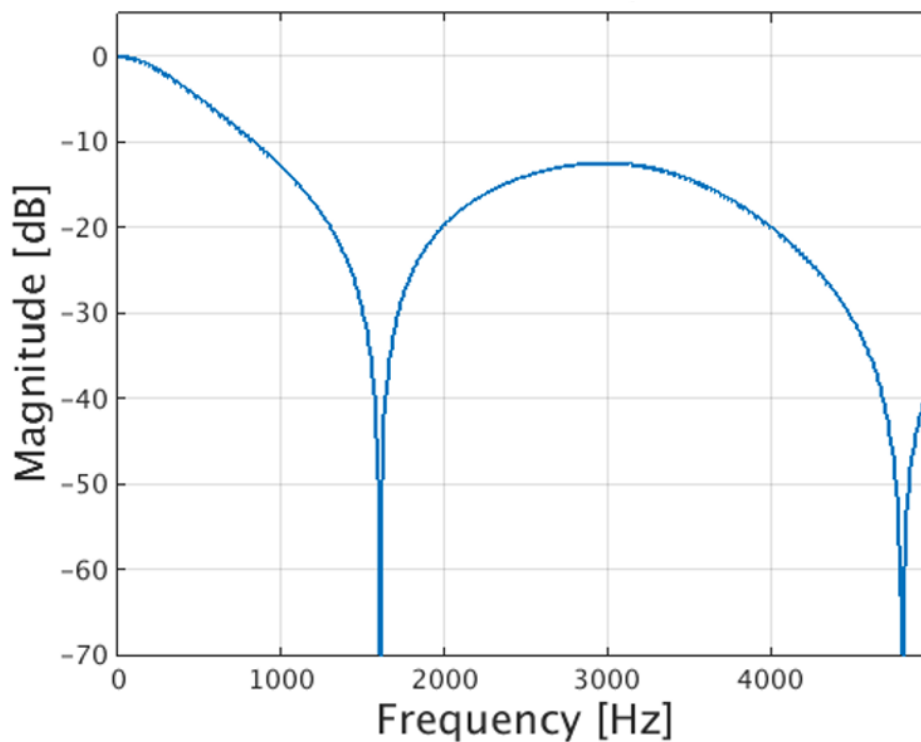
**Figure 4. High-pass filter in normal and reference mode**



### 3.6 Frequency response

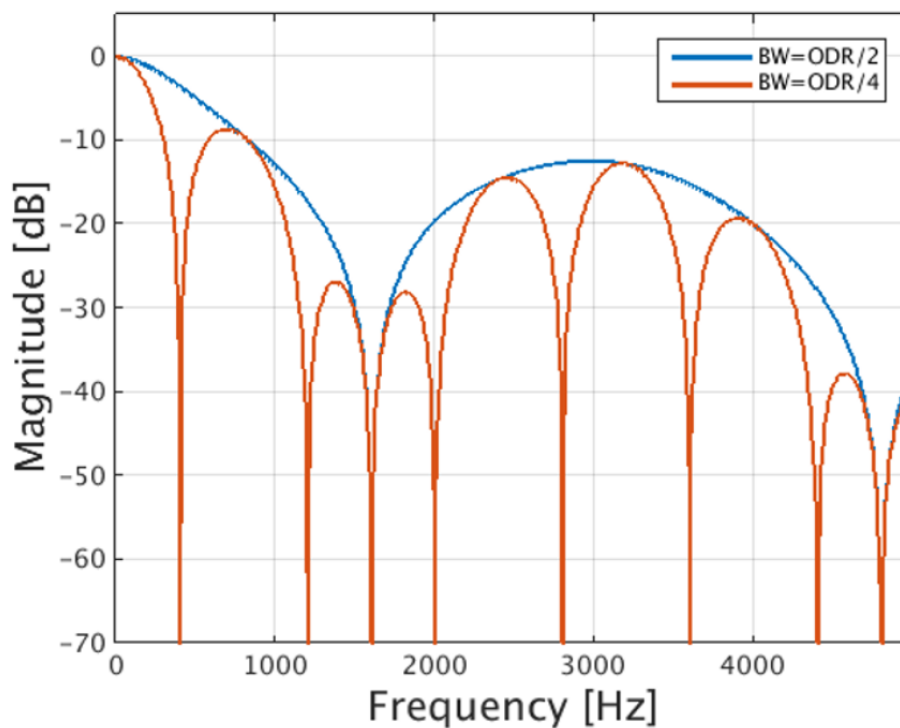
The following figures indicate the frequency response of the sensor in various configurations.

**Figure 5. LPF1 - High-Performance mode, ODR = 1600 Hz, BW = 400 Hz**



*Note: ODR 1600 Hz, BW = 400 Hz, Output of LPF1 (CTRL1.ODR = 1001; CTRL1.MODE = 01; CTRL6.BW\_FILT = 00)*

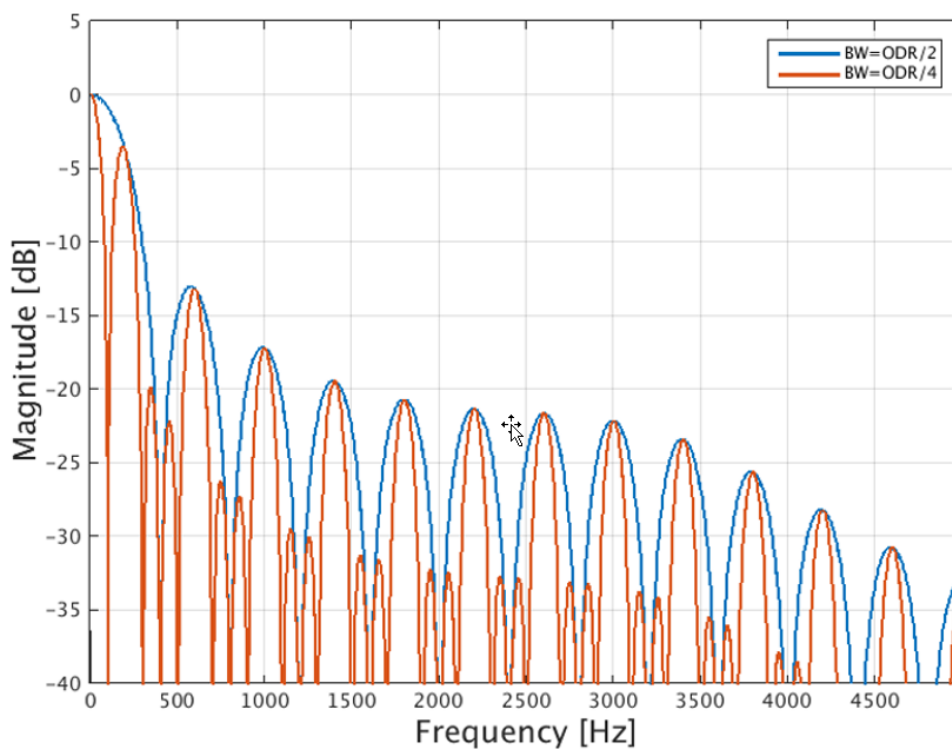
Figure 6. LPF1 / LPF2 - High-Performance mode, ODR 800 Hz



Note:

- blue: ODR 800 Hz, BW = 400 Hz, Output of LPF1 (CTRL1.ODR = 1000; CTRL1.MODE = 01; CTRL6.BW\_FILT = 00)
- orange: ODR 800 Hz, BW = 200 Hz, Output of LPF2 (CTRL1.ODR = 1000; CTRL1.MODE = 01; CTRL6.BW\_FILT = 01)

Figure 7. LPF1 / LPF2 - Low-Power mode 4 - ODR 200 Hz



Note:

- blue: ODR 200 Hz, BW = 180 Hz, Output of LPF1 (CTRL1.ODR = 0110; CTRL1.MODE = 00; CTRL1.MODE = 11; CTRL6.BW\_FILT = 00)
- orange: ODR 200 Hz, BW = 50 Hz, Output of LPF2 (CTRL1.ODR = 0110 ; CTRL1.MODE = 00; CTRL1.MODE = 11; CTRL6.BW\_FILT = 01)

Note: The frequency response is determined by CAD simulation.



## 4 Reading output data

### 4.1 Startup sequence

Once the device is powered up, it automatically downloads the calibration coefficients from the embedded non-volatile memory to the internal registers. When the boot procedure is completed, i.e. after approximately 20 milliseconds, the accelerometer automatically enters power-down. The default status of the pins with both VDD and VDDIO "on" is indicated in [Table 1. Internal pin status](#).

*Note: VDD cannot be lower than VDDIO. VDD = 0 V and VDDIO "on" is allowed: when this power supply configuration is applied, an internal pull-up is applied also to the SDA and SCL pins (the other pins maintain the default status indicated in [Table 1](#)).*

To turn on the accelerometer and gather acceleration data, it is necessary to select one of the operating modes through the CTRL1 register.

Refer to [Section 3 Operating modes](#) for a detailed description of data generation.

### 4.2 Using the status register

The device is provided with a STATUS register which can 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.

The read operations should be performed as follows:

1. Read STATUS
2. If DRDY = 0, then go to 1
3. Read OUT\_X\_L
4. Read OUT\_X\_H
5. Read OUT\_Y\_L
6. Read OUT\_Y\_H
7. Read OUT\_Z\_L
8. Read OUT\_Z\_H
9. Data processing
10. Go to 1

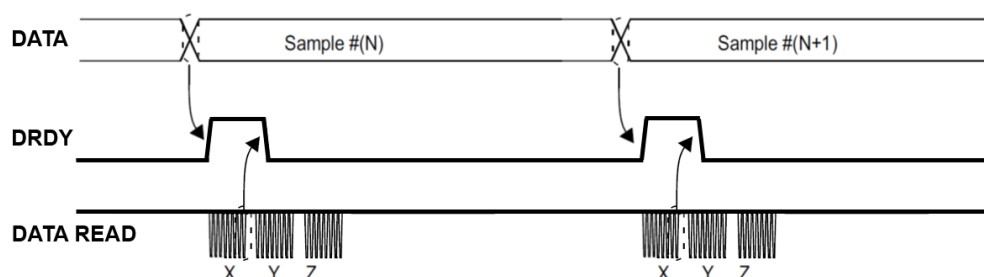
### 4.3 Using the data-ready signal

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

The data-ready signal is derived from the DRDY bit of the STATUS register. The signal can be driven to the INT1 pin by setting the INT1\_DRDY bit of the CTRL4\_INT1\_PAD\_CTRL register to 1 and to the INT2 pin by setting the INT2\_DRDY bit of the CTRL5\_INT2\_PAD\_CTRL register to 1.

The data-ready signal rises to 1 when a new set of data has been measured and is available to be read. In DRDY latched mode (DRDY\_PULSED bit = 0 in CTRL7 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 75  $\mu$ s (typical) if the accelerometer is configured in High-Performance mode, otherwise it can vary between 105  $\mu$ s and 175  $\mu$ s. Pulsed mode is not applied to the DRDY bit which is always latched.

Figure 8. Data-ready signal



#### 4.4 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 register or with the DRDY signal driven to the INT1/INT2 pins, it is strongly recommended to set the BDU (block data update) bit to 1 in the CTRL2 (21h) 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. OUT\_X\_H and OUT\_X\_L, OUT\_Y\_H and OUT\_Y\_L, OUT\_Z\_H and OUT\_Z\_L) 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 of one data channel 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.*

#### 4.5 Understanding output 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 output data for the X, Y, Z channels is given by the concatenation OUT\_X\_H & OUT\_X\_L, OUT\_Y\_H & OUT\_Y\_L, OUT\_Z\_H & OUT\_Z\_L.

Acceleration data is represented as 16-bit numbers, left-aligned and encoded in two's complement. These values (LSB) have different resolution according to the selected operating mode.

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

Table 15. Sensitivity

Full Scale	Sensitivity [mg/LSB]	
	12-bit format <sup>(1)</sup>	14-bit format
±2 g	0.976	0.244
±4 g	1.952	0.488
±8 g	3.904	0.976
±16 g	7.808	1.952

1. Only Low-Power Mode 1

### 4.5.1 Example of output data

Below is a simple example of how to use the LSB data and transform it into mg.  
 The values are given under the hypothesis of ideal device calibration (i.e., no offset, no gain error, etc.).  
 Get raw data from the sensor (High-Performance mode,  $\pm 2$  g):

```
OUT_X_L: 60h
```

```
OUT_X_H: FDh
```

```
OUT_Y_L: 78h
```

```
OUT_Y_H: 00h
```

```
OUT_Z_L: FCh
```

```
OUT_Z_H: 42h
```

Do register concatenation:

```
OUT_X_H & OUT_X_L: FD60h
```

```
OUT_Y_H & OUT_Y_L: 0078h
```

```
OUT_Z_H & OUT_Z_L: 42FCh
```

Apply sensitivity (e.g., 14-bit resolution, 0.244 at full scale  $\pm 2$  g):

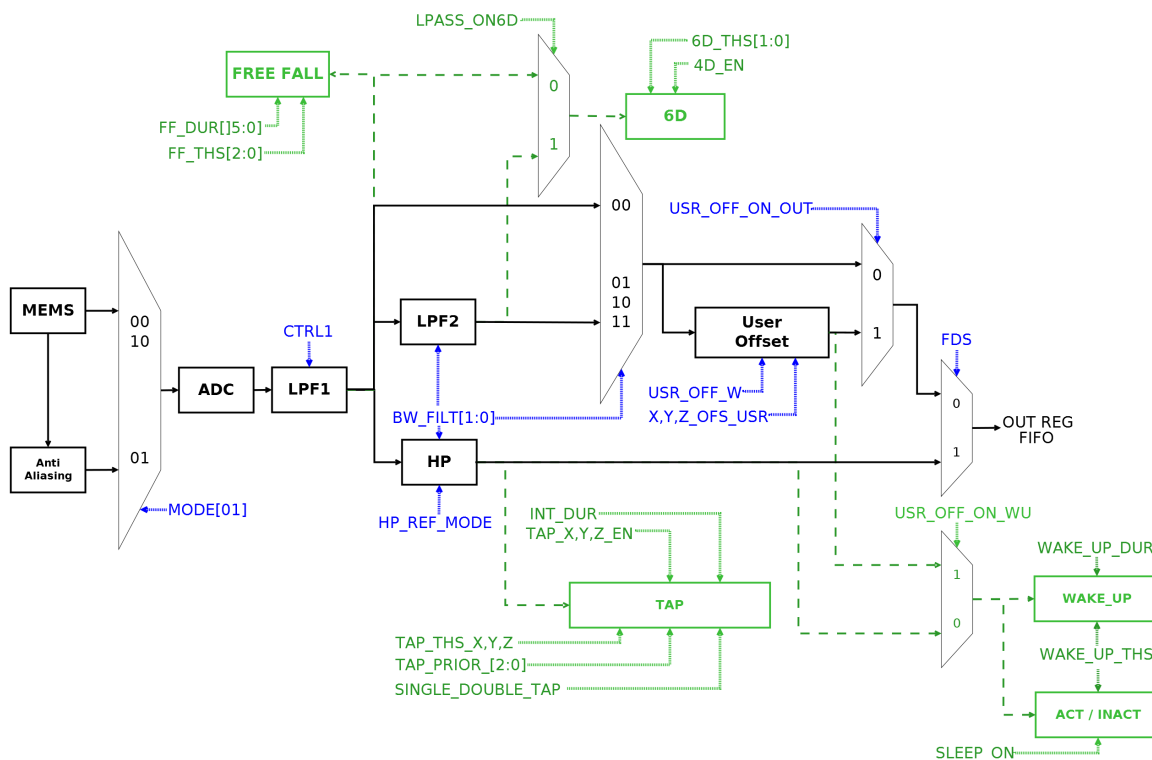
```
X: -672 / 4 * 0.244 = -41 mg
```

```
Y: +120 / 4 * 0.244 = +7 mg
```

```
Z: +17148 / 4 * 0.244 = +1046 mg
```

## 5 Interrupt generation and embedded functions

### Figure 9. Embedded functions



In order to generate an interrupt, the IIS2DLPC device has to be set in an active operating mode (not in power-down) because generation of the interrupt is based on accelerometer data.

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 the FIFO interrupt signals and sensor data-ready, can be driven to the INT1 and/or INT2 interrupt pins or checked by reading the dedicated source register bits.

The H\_LACTIVE bit of the CTRL3 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 the interrupt condition is reached.

The PP\_OD bit of CTRL3 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 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.

*Note: If latched mode is enabled (LIR = 1), it is not recommended to continuously poll ALL\_INT\_SRC or the dedicated source registers because by reading them the embedded functions are internally reset; a synchronous (with interrupt event) read of the source registers is recommended in this case.*

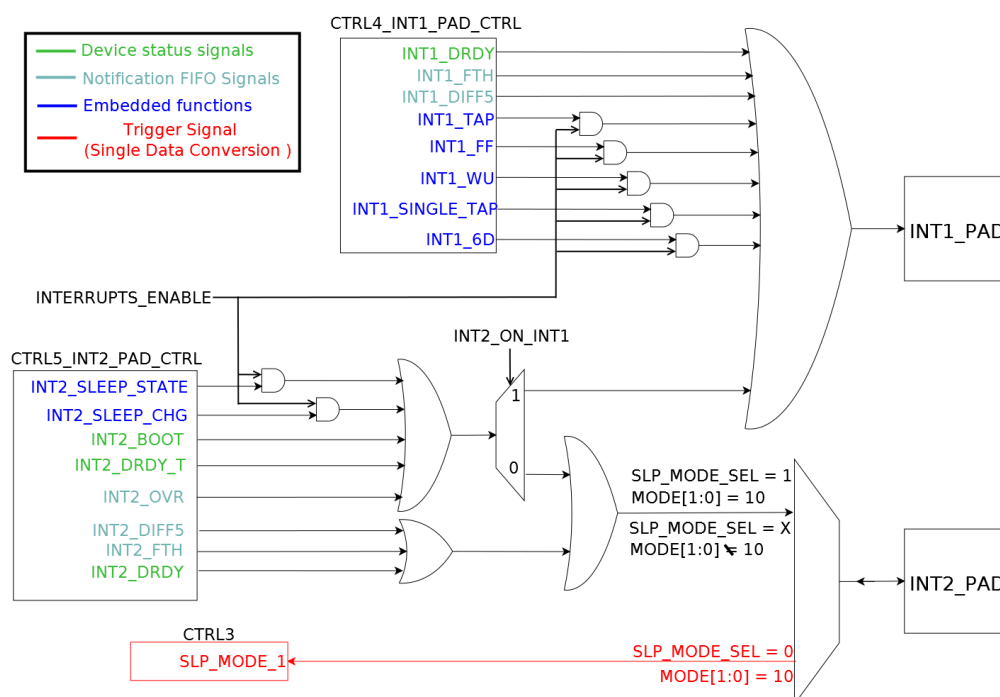
## 5.1 Interrupt pin configuration

The device is provided with two pins that can be activated to generate:

- *device status signals* (such as data-ready and boot);
- *embedded function interrupt signals*;
- *notification FIFO signals*.

All the *embedded function interrupt signals* are subordinate to the INTERRUPTS\_ENABLE bit in register CTRL7. If this bit is set, the embedded functions are enabled and the interrupts signals can be routed to the INT1 and/or INT2 pins; otherwise, if this bit is not set, the embedded functions are disabled. The INT2 pin can also become an input pin when it is used as an *external trigger in single data conversion (on-demand) mode*. In order to configure the device in this mode the user must set bits MODE[1:0] = 10 in register CTRL1 and bit SLP\_MODE\_SEL = 0 in register CTRL3. It is possible to route all the INT2 pin signals on the INT1 pin by setting bit INT2\_ON\_INT1 = 1 in register CTRL7.

Figure 10. Interrupt pin configuration



The description of the interrupt control registers appears below; 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 corresponding bit has to be set to 1.

Table 16. CTRL4\_INT1\_PAD\_CTRL

b7	b6	b5	b4	b3	b2	b1	b0
INT1_6D	INT1_SINGLE_TAP	INT1_WU	INT1_FF	INT1_TAP	INT1_DIFF5	INT1_FTH	INT1_DRDY

- INT1\_6D: 6D recognition is routed to the INT1 pin.
- INT1\_SINGLE\_TAP: Single-tap event recognition is routed to the INT1 pin.
- INT1\_WU: Wakeup event recognition is routed to the INT1 pin.
- INT1\_FF: Free-fall event recognition is routed to the INT1 pin.
- INT1\_TAP: Double-tap event recognition is routed to the INT1 pin.

- INT1\_DIFF5: FIFO full recognition is routed to the INT1 pin.
- INT1\_FTH: FIFO threshold event is routed to the INT1 pin.
- INT1\_DRDY: Accelerometer data-ready is routed to the INT1 pin.

**Table 17. CTRL5\_INT2\_PAD\_CTRL**

b7	b6	b5	b4	b3	b2	b1	b0
INT2_SLEEP_STATE	INT2_SLEEP_CHG	INT2_BOOT	INT2_DRDY_T	INT2_OVR	INT2_DIFF5	INT2_FTH	INT2_DRDY

- INT2\_SLEEP\_STATE: Enable routing of SLEEP\_STATE to the INT2 pin.
- INT2\_SLEEP\_CHG: Sleep change status routed to the INT2 pin.
- INT2\_BOOT: Boot state routed to the INT2 pin.
- INT2\_DRDY\_T: Temperature data-ready is routed to the INT2 pin.
- INT2\_OVR: FIFO overrun interrupt is routed to the INT2 pin.
- INT2\_DIFF5: FIFO full recognition is routed to the INT2 pin.
- INT2\_FTH: FIFO threshold event is routed to the INT2 pin.
- INT2\_DRDY: Accelerometer data-ready to the INT2 pin.

## 5.2 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.

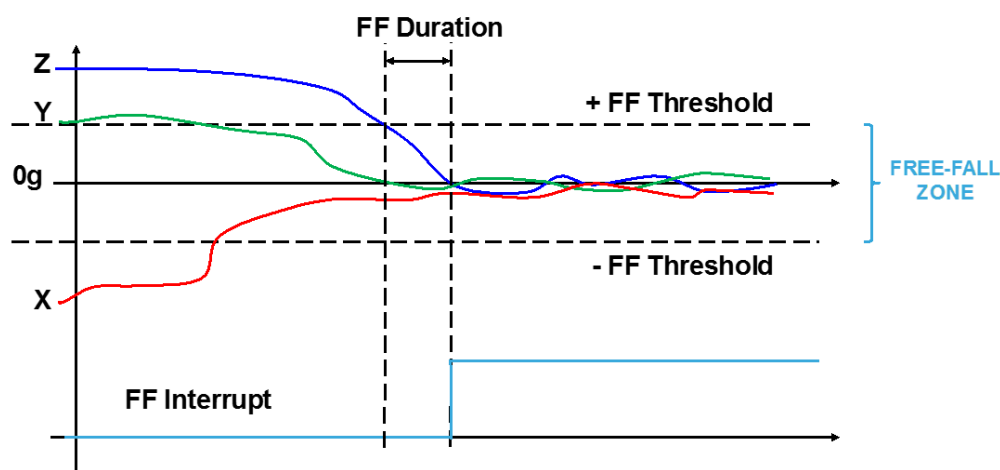
As indicated below, the STATUS register is partially duplicated at address 37h in order to allow a multiple read of consecutive registers.

- STATUS (27h) or STATUS\_DUP (37h);
- WAKE\_UP\_SRC (38h);
- TAP\_SRC (39h);
- SIXD\_SRC (3Ah);
- ALL\_INT\_SRC (3Bh).

### 5.3 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 11. Free-fall interrupt).

**Figure 11. Free-fall interrupt**



The free-fall event signal can be routed to the INT1 pin by setting the INT1\_FF bit of the CTRL4\_INT1\_PAD\_CTRL register to 1; it can also be checked by reading the FF\_IA bit of the STATUS register.

If latch mode is disabled (LIR bit of CTRL3 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 or ALL\_INT\_SRC 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 STATUS is reset when the free-fall condition is no longer verified).

The free-fall detection parameters can be modified by configuring the FREE\_FALL (contains bits FF\_THS[2:0] and FF\_DUR[4:0]) and WAKE\_UP\_DUR (contains MSB of duration parameter - FF\_DUR5) registers. The threshold value can be set through the FF\_THS[2:0] bits and is described in Table 18. Free-fall threshold value. The values given in this table are valid for any accelerometer full-scale configuration.

**Table 18. Free-fall threshold value**

FREE_FALL - FF_THS[2:0]	Threshold value
000	~156 mg
001	~219 mg
010	~250 mg
011	~312 mg
100	~344 mg
101	~406 mg
110	~469 mg
111	~500 mg

Duration time is measured in N/ODR, where N is the content of the FF\_DUR[5:0] field of the FREE\_FALL / WAKE\_UP\_DUR registers and ODR is the accelerometer data rate.

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

1. Write 64h in CTRL1 // Turn on the accelerometer  
// ODR = 200 Hz, High-Performance
2. Write 04h in CTRL6 // FS  $\pm 2$  g, LOW\_NOISE enabled
3. Write 00h in WAKE\_UP\_DUR // Set event duration (FF\_DUR5 = 0)
4. Write 33h in FREE\_FALL // Set FF threshold (FF\_THS[2:0] = 011b)  
// Set six sample event duration (FF\_DUR[5:0] = 000110b)
5. Write 10h in CTRL4\_INT1\_PAD\_CTRL // FF interrupt driven to INT1 pin
6. Write 10h in CTRL3 // Latch interrupt
7. Write 20h in CTRL7 // Enable interrupts

The sample code exploits a threshold set to  $\sim 312$  mg for free-fall recognition and the event is notified by hardware through the INT1 pin. The FF\_DUR[5:0] field of the FREE\_FALL / WAKE\_UP\_DUR registers is configured to ignore events that are shorter than  $6/\text{ODR} = 6/200 \text{ Hz} = 30 \text{ ms}$  in order to avoid false detections.



## 5.4 Wake-up interrupt

In the IIS2DLPC device the wake-up feature can use the high-pass filter or the offset outputs, this choice can be done through the `USR_OFF_ON_WU` bit in `CTRL7` as illustrated in [Figure 9. Embedded functions](#).

If “offset output” is selected, every axis can have offset with a different value, writing registers `X_OFS_USR`, `Y_OFS_USR`, `Z_OFS_USR`. Bit weight is defined through the `USR_OFF_W` bit in register `CTRL7`.

The wake-up interrupt signal is generated if a certain number of consecutive data exceed the configured threshold ([Figure 12. Wake-up event recognition \(using the HP filter\)](#)).

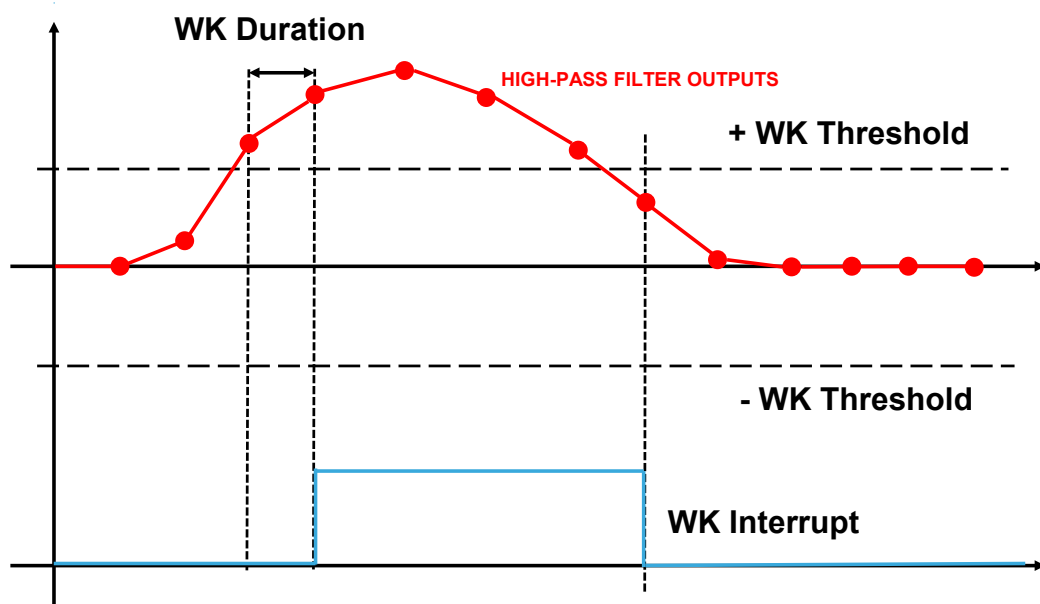
The unsigned threshold value is defined using the `WK_THS [5:0]` bits of the `WAKE_UP_THS` register; the value of 1 LSB of these 6 bits depends on the selected accelerometer full scale: 1 LSB =  $FS/64$ . The threshold is applied to both positive and negative data: for a wake-up interrupt generation 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 `WAKE_DUR [1:0]` bits of the `WAKE_UP_DUR` register: 1 LSB corresponds to  $1 \cdot ODR$  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` interrupt pin by setting the `INT1_WU` bit of the `CTRL4_INT1_PAD_CTRL` register to 1; it can also be checked by reading the `WU_IA` bit of the `STATUS` register. The `X_WU`, `Y_WU`, `Z_WU` bits of the `WAKE_UP_SRC` register indicate which axis has triggered the wake-up event.

If latch mode is disabled (`LIR` bit of `CTRL3` 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` or `ALL_INT_SRC` register. The `X_WU`, `Y_WU`, `Z_WU` bits are maintained at the state in which the interrupt was generated until the read is performed, and released at the next `ODR` cycle. In case the `WU_X`, `WU_Y`, `WU_Z` bits have to be evaluated (in addition to the `WU_IA` bit), it is recommended to directly read the `WAKE_UP_SRC` register (do not use `ALL_INT_SRC` register for this specific case). If 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 the `WAKE_UP_SRC` or `ALL_INT_SRC` register is reset when the wake-up condition is no longer verified).

**Figure 12. Wake-up event recognition (using the HP filter)**



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

1. Write 64h in CTRL1 // Turn on the accelerometer  
// ODR = 200 Hz, High-Performance
2. Write 04 in CTRL6 // FS  $\pm 2$  g, LOW\_NOISE enabled
3. Write 20h in CTRL7 // Use HP filter, enable interrupts
4. Write 00h in WAKE\_UP\_DUR // No duration
5. Write 02h in WAKE\_UP\_THS // Set wake-up threshold
6. Write 20h in CTRL4\_INT1\_PAD\_CTRL // Wake-up interrupt driven to INT1 pin

Since the duration time is set to zero, the wake-up interrupt signal is generated for each X,Y,Z data from HP filter exceeding the configured threshold. The WU\_THS field of the WAKE\_UP\_THS register is set to 000010b, therefore the wake-up threshold is 62.5 mg ( $= 2 * FS / 64$ ).

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

1. Write 04h in CTRL6 // FS  $\pm 2$  g LOW\_NOISE enabled
2. Write 34h in CTRL7 // Use X/Y/Z\_OFS\_USR registers  
// X/Y/Z\_OFS\_USR weight 15.6 mg/LSb  
// Enable interrupts
3. Write 00h in X\_OFS\_USR // Set X offset as 0
4. Write 00h in Y\_OFS\_USR // Set Y offset as 0
5. Write 40h in Z\_OFS\_USR // Set Z offset as 1 g
6. Write 00h in WAKE\_UP\_DUR // No duration
7. Write 02h in WAKE\_UP\_THS // Set wake-up threshold
8. Write 20h in CTRL4\_INT1\_PAD\_CTRL // Wake-up interrupt driven to INT1 pin
9. Write 64h in CTRL1 // Turn on the accelerometer  
// ODR = 200 Hz, High-Performance

Since the duration time is set to zero, the wake-up interrupt signal is generated for each X,Y,Z data from the difference between the data measured and the X\_OFS\_USR, Y\_OFS\_USR, Z\_OFS\_USR registers exceeding the configured threshold. The WU\_THS field of the WAKE\_UP\_THS register is set to 000010b, therefore the wake-up threshold is 62.5 mg ( $= 2 * FS / 64$ ).

## 5.5 6D/4D orientation detection

The IIS2DLPC 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.

### 5.5.1 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 SIXD\_SRC register indicate which axis has triggered the 6D event.

In more detail:

**Table 19. SIXD\_SRC 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 absolute value 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 absolute value bigger than the threshold.

The 6D\_THS[1:0] bits of the TAP\_THS\_X register are used to select the threshold value used to detect the change in device orientation. The threshold values given in [Table 20. Threshold for 4D/6D function](#) are valid for each accelerometer full-scale value.

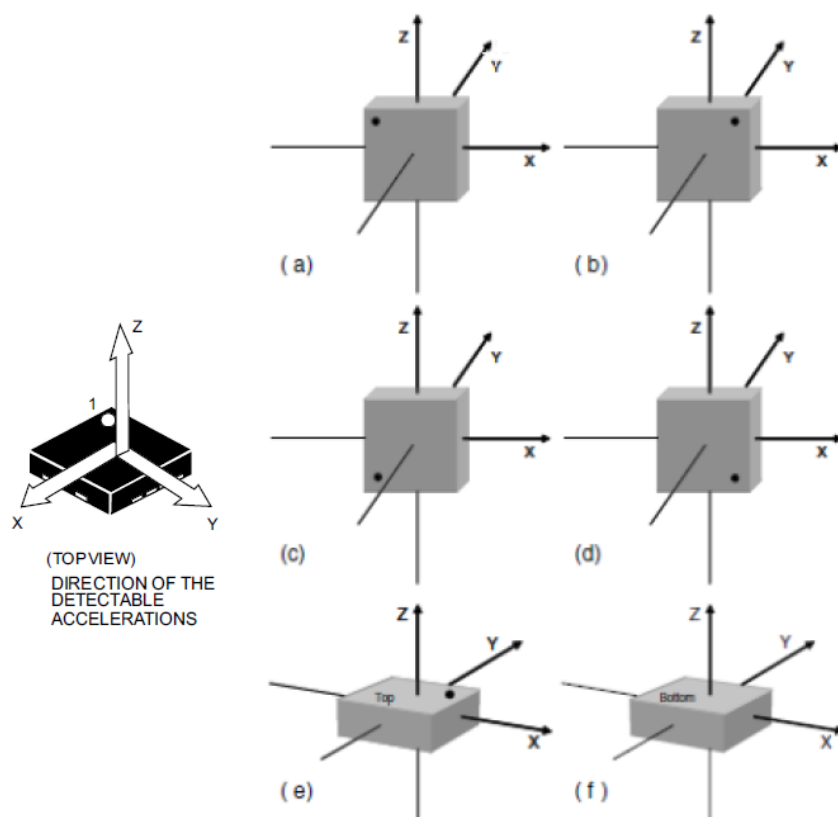
**Table 20. Threshold for 4D/6D function**

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

This interrupt signal can be driven to the INT1 interrupt pin by setting the INT1\_6D bit of the CTRL4\_INT1\_PAD\_CTRL register to 1; it can also be checked by reading the 6D\_IA bit of the SIXD\_SRC register.

If latch mode is disabled (LIR bit of CTRL3 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 SIXD\_SRC or ALL\_INT\_SRC register clears the request and the device is ready to recognize a different orientation. The XL, XH, YL, YH, ZL, ZH bits are not affected by the LIR configuration: they correspond to the current state of the device when the D6D\_SRC register is read. If latched 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 13. 6D recognized orientations](#), the content of the SIXD\_SRC register for each position is shown in [Table 21. SIXD\\_SRC register for 6D positions](#).

**Figure 13. 6D recognized orientations**

**Table 21. SIXD\_SRC register for 6D positions**

Case	6D_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

The following example implements a SW routine for 6D orientation detection:

- Write 64h in CTRL1 // Turn on the accelerometer  
// ODR = 200 Hz, High-Performance
- Write 04h in CTRL6 // FS  $\pm 2$  g, LOW\_NOISE enabled
- Write 20h in CTRL7 // Do not use low-pass filter for 6D, enable interrupts
- Write 40h in TAP\_THS\_X // Set 6D threshold (6D\_THS[1:0] = 10b = 60 degrees)
- Write 80h in CTRL4\_INT1\_PAD\_CTRL // 6D interrupt driven to INT1 pin

### 5.5.2 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\_THS\_X register to 1. In this configuration, Z-axis position detection is disabled, therefore reducing position recognition to cases (a), (b), (c), and (d) of [Table 21. SIXD\\_SRC register for 6D positions](#).

## 5.6 Single-tap and double-tap recognition

The single-tap and double-tap recognition functions featured in the IIS2DLPC 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. 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 IIS2DLPC device the single-tap and double-tap recognition functions use the high-pass filter to detect tap events.

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

The recommended accelerometer ODR for single and double-tap recognition is 400 Hz or higher.

### 5.6.1 Single-tap

If the device is configured for single-tap event detection, an interrupt is generated when the high-pass filtered data exceeds the programmed threshold and returns below it within the shock time window.

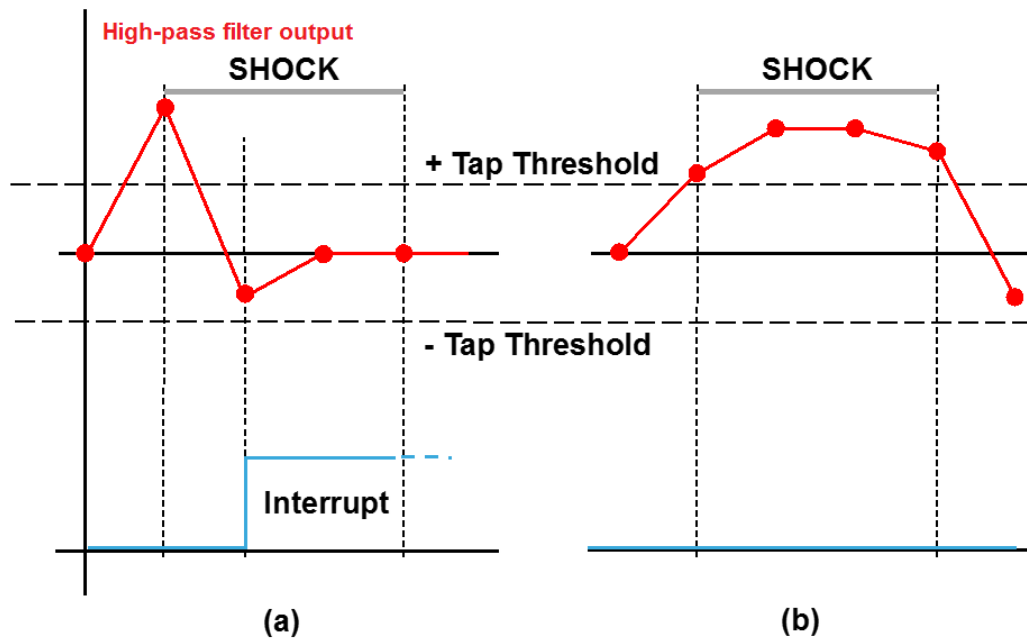
In the single-tap case, if the LIR bit of the CTRL3 register is set to 0, the interrupt is kept high for the duration of the quiet window.

In order to enable the latch feature on the single-tap interrupt signal, the LIR bit of CTRL3 has to be set to 1: the interrupt is kept high until the TAP\_SRC or ALL\_INT\_SRC register is read.

The SINGLE\_DOUBLE\_TAP bit of WAKE\_UP\_THS has to be set to 0 in order to enable single-tap recognition only.

In case (a) of [Figure 14. Single-tap event recognition](#) the single-tap event has been recognized, while in case (b) the tap has not been recognized because the signal falls below the threshold after the shock time window has expired.

Figure 14. Single-tap event recognition



### 5.6.2 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.

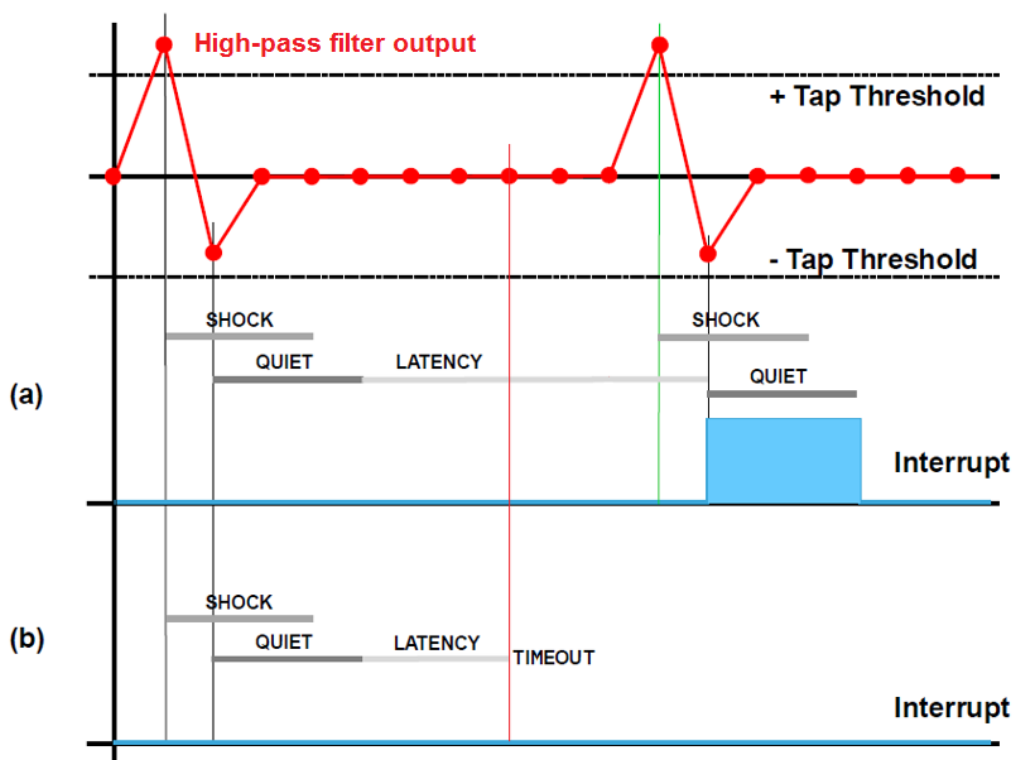
In particular, after the first tap has been recognized, the second tap detection procedure is delayed for an interval defined by the quiet time. This means that after the first tap has been recognized, the second tap detection procedure starts only if the high-pass filtered data exceeds the threshold after the quiet window but before the latency window has expired. In case (a) of Figure 15. 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 high-pass filtered data exceeds 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 high-pass filtered 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 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 or ALL\_INT\_SRC register is read.

Figure 15. Double-tap event recognition (LIR bit = 0)



### 5.6.3 Single-tap and double-tap recognition configuration

The IIS2DLPC 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 TAP\_THS\_Z register must be set to 1 to enable the tap recognition on X, Y, Z directions, respectively.

The TAP\_THSX[4:0], TAP\_THSY[4:0], TAP\_THSZ[4:0] bits of the TAP\_THS\_X, TAP\_THS\_Y, TAP\_THS\_Z registers 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 high-pass filtered data.

The user can also define the “priority report” of the single / double-tap interrupt event through TAP\_PRIOR\_[2:0] in register TAP\_THS\_Y. This feature is useful in case of a contemporary tap event on more than one axis because only the tap event related to the axis having higher priority is given in the source register.

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 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 latch mode is disabled (LIR bit of CTRL3 is set to 0), the quiet time also defines the length of the interrupt pulse (for both single and double-tap). The QUIET[1:0] bits of the INT\_DUR 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 LATENCY[3:0] bits of the INT\_DUR 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 LATENCY[3:0] bits are set to a different value, 1 LSB corresponds to 32/ODR time.

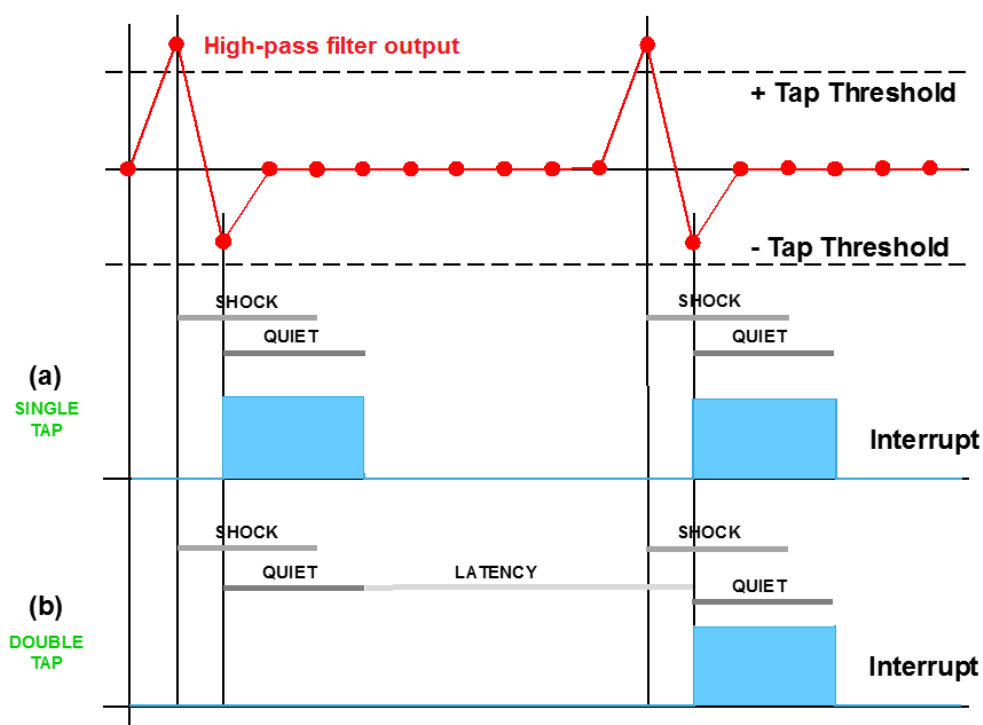
Figure 16. Single and double-tap recognition (LIR bit = 0) illustrates a single-tap event (a) and a double-tap event (b). These interrupt signals can be driven to the INT1 interrupt pin by setting the INT1\_SINGLE\_TAP bit of the CTRL4\_INT1\_PAD\_CTRL register to 1 for the single-tap case, and setting the INT1\_TAP bit of the CTRL4\_INT1\_PAD\_CTRL register to 1 for the double-tap case.

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.

No single/double-tap interrupt is generated if the accelerometer is in inactivity status (see Section 5.7 Activity/Inactivity recognition for more details).



Figure 16. Single and double-tap recognition (LIR bit = 0)



The tap interrupt signals can also be checked by reading the TAP\_SRC register, described in Table 22. TAP\_SRC register.

Table 22. TAP\_SRC 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 to 0, only single-tap recognition is enabled: double-tap recognition is disabled and cannot be detected. When the SINGLE\_DOUBLE\_TAP bit is set to 1, both single and double-tap recognition are enabled, and the single-tap event is always recognized first, followed by the double-tap event.

If 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, latch mode is applied to the single-tap interrupt signal; when it is set to 1, latch mode is applied to the double-tap interrupt signal only. The latched interrupt signal is kept high until the TAP\_SRC or ALL\_INT\_SRC register is read. The TAP\_SIGN, X\_TAP, Y\_TAP, Z\_TAP bits are maintained at the state in which the interrupt was generated until the read is performed, and released at the next ODR cycle. In case the TAP\_SIGN, X\_TAP, Y\_TAP, Z\_TAP bits have to be evaluated (in addition to the TAP\_IA bit), it is recommended to directly read the TAP\_SRC register (do not use ALL\_INT\_SRC register for this specific case). If latched mode is enabled but the interrupt signal is not driven to the interrupt pins, the latch feature does not take effect.

#### 5.6.4 Single-tap example

The following example code implements a SW routine for single-tap detection.

1. Write 74h in CTRL1 // Turn on the accelerometer  
// ODR = 400 Hz, High-Performance
2. Write 04h in CTRL6 // FS  $\pm 2$  g, LOW\_NOISE enabled
3. Write 09h in TAP\_THS\_X // Set tap threshold for X-axis
4. Write E9h in TAP\_THS\_Y // Set tap threshold for Y-axis  
// Set TAP priority Z-Y-X
5. Write E9h in TAP\_THS\_Z // Enable tap detection on X, Y, Z-axis  
// Set tap threshold for Z-axis
6. Write 06h in INT\_DUR // Set quiet and shock time windows
7. Write 00h in WAKE\_UP\_THS // Only single-tap enabled (SINGLE\_DOUBLE\_TAP = 0)
8. Write 40h in CTRL4\_INT1\_PAD\_CTRL // Single-tap interrupt driven to INT1 pin
9. Write 20h in CTRL7 // Enable interrupts

In this example the threshold for each axis is set to 01001b, therefore the tap threshold is 562.5 mg ( $= 9 * FS / 32$ ).

The SHOCK field of the INT\_DUR register is set to 10b: an interrupt is generated when the high-pass filtered 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 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 ( $= 1 * 4 / ODR$ ).

### 5.6.5 Double-tap example

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

```

1.  Write 74h in CTRL1                // Turn on the accelerometer
                                       // ODR = 400 Hz, High-Performance
2.  Write 04h in CTRL6                // FS  $\pm 2$  g, LOW_NOISE enabled
    Write 0Ch in TAP_THS_X            // Set tap threshold for X-axis
    Write ECh in TAP_THS_Y            // Set tap threshold for Y-axis
                                       // Set TAP priority Z-Y-X
3.  Write ECh in TAP_THS_Z            // Enable tap detection on X, Y, Z-axis
                                       // Set tap threshold for Z axis
4.  Write 7Fh in INT_DUR               // Set duration, quiet and shock time windows
5.  Write 80h in WAKE_UP_THS           // Single and double-tap enabled
                                       // (SINGLE_DOUBLE_TAP = 1)
6.  Write 08h in CTRL4_INT1_PAD_CTRL   // Single-tap interrupt driven to INT1 pin
7.  Write 20h in CTRL7                // Enable interrupts

```

In this example the threshold for each axis is set to 01100b, therefore the tap threshold is 750 mg ( $= 12 * FS / 32$ ). For interrupt generation, during the first and the second tap the high-pass filtered data must return below the threshold before the shock window has expired. The SHOCK field of the INT\_DUR 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 high-pass filtered data overthreshold during the quiet time window. Furthermore, since latch mode is disabled, the interrupt is kept high for the duration of the quiet window. The QUIET field of the INT\_DUR 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 register is set to 0111b, therefore the duration time is 560 ms ( $= 7 * 32 / ODR$ ).

## 5.7 Activity/Inactivity recognition

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

The activity/inactivity function is enabled by setting the SLEEP\_ON bit of WAKE\_UP\_THS (34h) register to 1. If the sleep state condition is detected, the IIS2DLPC automatically goes to 12.5 Hz ODR in the low-power mode previously selected by the LP\_MODE[1:0] bits in CTRL1 (20h). The IIS2DLPC wakes up from the sleep state as soon as a wake-up event has been detected, switching to the operating mode and ODR configured in CTRL1 (20h) register.

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 can use the high-pass filter or the offset outputs, this choice can be done through the USR\_OFF\_ON\_WU bit in CTRL7 as illustrated in [Figure 9. Embedded functions](#).

In case of "offset output" are selected, every axis can be offset with a different value, writing registers X\_OFS\_USR, Y\_OFS\_USR, Z\_OFS\_USR. Bit weight is defined through the USR\_OFF\_W bit in register CTRL7.

This function can be fully programmed by the user in terms of expected amplitude and timing of the high-pass filtered data by means of a dedicated set of registers ([Figure 17. Activity/Inactivity recognition \(using the HP filter\)](#)).

The unsigned threshold value is defined using the WK\_THS[5:0] bits in the WAKE\_UP\_THS 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 high-pass filtered data.

When a certain number of consecutive X,Y,Z high-pass filtered data is smaller than the configured threshold, the ODR [3:0] bits of the CTRL1 register are bypassed (inactivity) and the accelerometer is internally set to 12.5 Hz although the content of CTRL1 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 register: 1 LSB corresponds to 512/ODR time, where ODR is the accelerometer output data rate.

During the inactivity status of the device, the SLEEP\_STATE bit in STATUS is set high. This bit can be routed to the INT2 pin setting to 1 both the INT2\_SLEEP\_STATE and INT2\_SLEEP\_STATE\_CHG bits in CTRL5\_INT2\_PAD\_CTRL. Please note that this signal is not compatible with "latched notification mode", the LIR bit of CTRL3 should be set to 0.

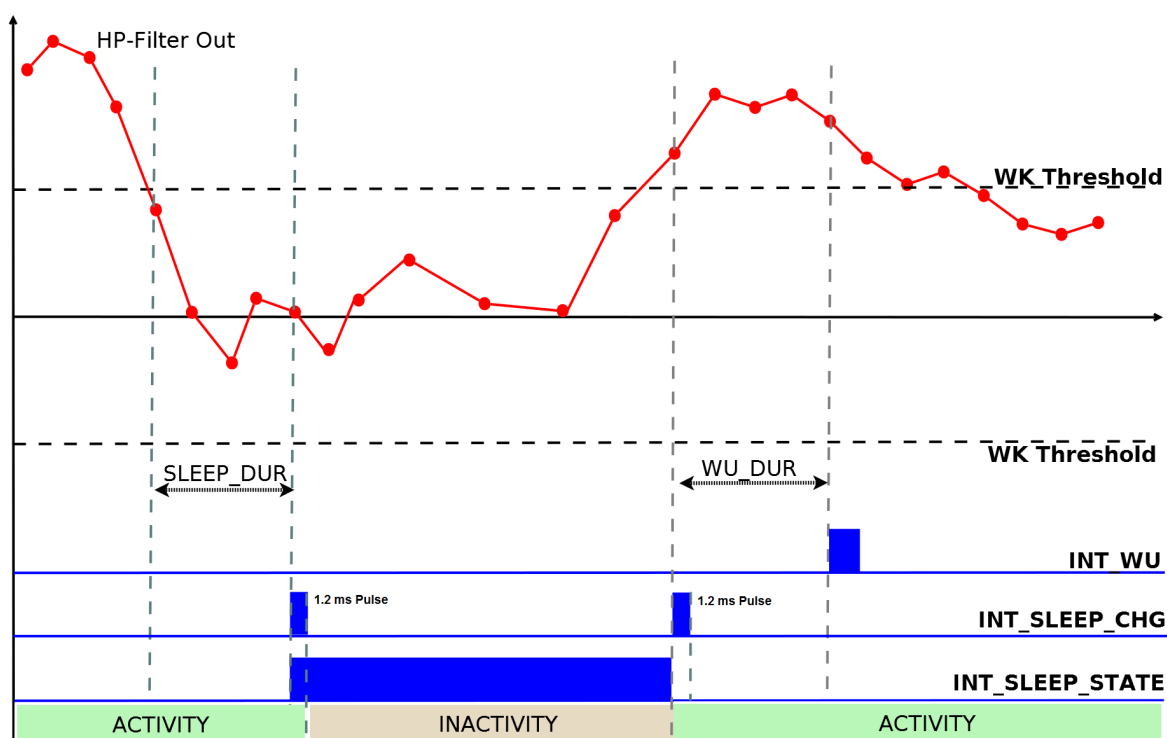
Every time the device status changes from activity to inactivity or vice-versa, the SLEEP\_CHANGE\_IA bit in ALL\_INT\_SRC is set for about 1.2 ms. This bit can be routed on the INT2 pin using the INT2\_SLEEP\_CHG bit in CTRL5\_INT2\_PAD\_CTRL. The typical duration of the sleep change event pulse on the interrupt pin depends on the selected power mode and Activity / Inactivity transition (when using High-Performance mode).

**Table 23. Pulse duration on interrupt pin**

Power mode	Activity to Inactivity pulse duration	Inactivity to Activity pulse duration
Low-Power Mode 1	1.20 ms	1.20 ms
Low-Power Mode 2	1.70 ms	1.70 ms
Low-Power Mode 3	2.30 ms	2.30 ms
Low-Power Mode 4	3.55 ms	3.55 ms
High-Performance mode	0.71 ms + pulse duration of the selected Low-Power Mode	1 / ODR

When a single sample of high-pass filtered data on one axis becomes bigger than the threshold, the CTRL1 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 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\_INT1\_PAD\_CTRL register to 1).

**Figure 17. Activity/Inactivity recognition (using the HP filter)**


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

- |                                     |   |
|-------------------------------------|---|
| 1. Write 50h in CTRL1               | // Turn on the accelerometer<br>// ODR = 200 Hz, FS = $\pm 2g$                    |
| 2. Write 42h in WAKE_UP_DUR         | // Set duration for inactivity detection<br>// Set duration for wake-up detection |
| 3. Write 42h in WAKE_UP_THS         | // Set activity/inactivity threshold<br>// Enable activity/inactivity detection   |
| 4. Write 20h in CTRL4_INT1_PAD_CTRL | // Activity (wakeup) interrupt driven to INT1 pin                                 |
| 5. Write 20h in CTRL7               | // Enable interrupts  |

In this example the WU\_THS field of the WAKE\_UP\_THS register is set to 000010b, therefore the activity/inactivity threshold is 62.5 mg ( $= 2 * FS / 64$ ).

Before inactivity detection, the X,Y,Z high-pass filtered data must be smaller than the configured threshold for a period of time defined by the SLEEP\_DUR field of the WAKE\_UP\_DUR 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 register settings immediately restored if the high-pass filtered data of (at least) one axis is 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 register: this field is set to 10b, corresponding to 10 ms ( $= 2 * 1 / ODR$ ).

The following routine describes how to route the sleep change event on the INT2 pin.

- |                                     |  |
|-------------------------------------|--|
| 1. Write 50h in CTRL1               | // Turn on the accelerometer                 |
|                                     | // ODR = 200 Hz, FS = $\pm 2 g$              |
| 2. Write 02h in WAKE_UP_DUR         | // Set duration for inactivity detection     |
| 3. Write 42h in WAKE_UP_THS         | // Set activity/inactivity threshold         |
|                                     | // Enable activity/inactivity detection      |
| 4. Write 40h in CTRL5_INT2_PAD_CTRL | // Sleep change interrupt driven to INT2 pin |
| 5. Write 20h in CTRL7               | // Enable interrupts                         |

This example is similar to the previous one but the event routed is "Sleep change" on the INT2 pin.

## 5.8 Stationary/Motion detection

Stationary / Motion detection is a particular case of the Activity / Inactivity functionality in which no ODR / power mode changes occur when a sleep condition (equivalent to Stationary condition) is detected. Stationary / Motion detection is activated by setting the STATIONARY bit to 1 in the WAKE\_UP\_DUR register. If both the STATIONARY bit and SLEEP\_ON bit in the WAKE\_UP\_THS register are set to 1, Stationary / Motion detection is selected.

## 5.9 Boot status

After the device is powered up, the IIS2DLPC performs a 20 ms boot procedure to load the trimming parameters (register addresses: 02h; from 07h to 0Bh; from 10h to 1Fh). After the boot is completed, the accelerometer is automatically configured in power-down mode.

During the boot time the registers are not accessible.

After power-up, the trimming parameters can be reloaded by setting the BOOT bit of the CTRL2 register to 1.

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 a reset to the default value of the control registers is required (registers addresses: from 20h to 25h; 2Eh; from 30h to 36h; from 3Ch to 3Fh), it can be performed by setting the SOFT\_RESET bit of the CTRL2 register to 1. The software reset procedure can take 5  $\mu$ s.

The boot status signal can be driven to the INT2 interrupt pin by setting the INT2\_BOOT bit of the CTRL5\_INT2\_PAD\_CTRL register to 1: the signal goes to '1' while a boot is taking place, and returns to '0' when it is done.

The flow must be performed serially (from ANY operating mode) as shown in the example below:

1. Set SOFT\_RESET bit to '1'
2. Wait 5  $\mu$ s (or wait until the SOFT\_RESET bit of the CTRL2 register returns to 0)
3. Set BOOT bit to '1'
4. Wait 20 ms

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 BOOT bit and SOFT\_RESET bit of the CTRL2 register).

## 6 First-in first-out (FIFO) buffer

In order to limit intervention by the host processor and facilitate post-processing data for recognition of events, the IIS2DLPC 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, 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 or INT2 pins.

### 6.1 FIFO description

The FIFO buffer is able to store up to 32 acceleration samples stored with the resolution according to bits MODE[1:0] and LP\_MODE[1:0] in register CTRL1.

The data sample set consists of 6 bytes (XI, Xh, Yl, Yh, Zl, and Zh) and they are released to the FIFO at the selected output data rate defined in ODR[3:0] register CTRL1.

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

**Table 24. FIFO buffer full representation (32<sup>nd</sup> sample set stored)**

Output registers	28h	29h	2Ah	2Bh	2Ch	2Dh
	XI	Xh	Yl	Yh	Zl	Zh
FIFO index	FIFO sample set					
FIFO(0)	XI(0)	Xh(0)	Yl(0)	Yh(0)	Zl(0)	Zh(0)
FIFO(1)	XI(1)	Xh(1)	Yl(1)	Yh(1)	Zl(1)	Zh(1)
FIFO(2)	XI(2)	Xh(2)	Yl(2)	Yh(2)	Zl(2)	Zh(2)
FIFO(3)	XI(3)	Xh(3)	Yl(3)	Yh(3)	Zl(3)	Zh(3)
...	...	...	...	...	...	...
FIFO(30)	XI(30)	Xh(30)	Yl(30)	Yh(30)	Zl(30)	Zh(30)
FIFO(31)	XI(31)	Xh(31)	Yl(31)	Yh(31)	Zl(31)	Zh(31)

**Table 25. FIFO buffer full representation (33<sup>rd</sup> sample set stored and 1<sup>st</sup> sample discarded)**

Output registers	28h	29h	2Ah	2Bh	2Ch	2Dh
	XI	Xh	Yl	Yh	Zl	Zh
FIFO index	Sample set					
FIFO(0)	XI(1)	Xh(1)	Yl(1)	Yh(1)	Zl(1)	Zh(1)
FIFO(1)	XI(2)	Xh(2)	Yl(2)	Yh(2)	Zl(2)	Zh(2)
FIFO(2)	XI(3)	Xh(3)	Yl(3)	Yh(3)	Zl(3)	Zh(3)
FIFO(3)	XI(4)	Xh(4)	Yl(4)	Yh(4)	Zl(4)	Zh(4)
...	...	...	...	...	...	...
FIFO(31)	XI(32)	Xh(32)	Yl(32)	Yh(32)	Zl(32)	Zh(32)

Table 24. FIFO buffer full representation (32<sup>nd</sup> sample set stored) represents the FIFO full status when 32 samples are stored in the buffer while Table 25. FIFO buffer full representation (33<sup>rd</sup> sample set stored and 1<sup>st</sup> sample discarded) represents the next step when the 33<sup>rd</sup> sample is inserted into FIFO and the 1<sup>st</sup> 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 IIS2DLPC output registers (28h to 2Dh) always contain the oldest FIFO sample set.

## 6.2 FIFO registers

The FIFO buffer is managed by two different registers, one allows enabling and configuring the FIFO behavior, the other one provides information about the buffer status.

A few other registers are used to route FIFO events on the INT1 and INT2 pins to interrupt the application processor. These are discussed in [Section 6.3 FIFO interrupts](#).

### 6.2.1 FIFO\_CTRL (2Eh)

The FIFO\_CTRL register contains the mode in which the FIFO is set. At reset, by default, the FIFO mode is Bypass which means that the FIFO is off; the FIFO is enabled and starts storing the samples as soon as the mode is set to a mode other than Bypass.

**Table 26. FIFO\_CTRL register**

b7	b6	b5	b4	b3	b2	b1	b0
FMode2	FMode1	FMode0	FTH4	FTH3	FTH2	FTH1	FTH0

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

FTH[4:0] bits are discussed in [Section 6.3.1 FIFO threshold](#).

### 6.2.2 FIFO\_SAMPLES (2Fh)

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

**Table 27. FIFO\_SAMPLES register**

b7	b6	b5	b4	b3	b2	b1	b0
FIFO_FTH	FIFO_OVR	Diff5	Diff4	Diff3	Diff2	Diff1	Diff0

- FIFO\_FTH bit is set high when FIFO content is greater than or equal to the watermark level. This flag can be routed to the INT1 or INT2 pin (see [Section 6.3 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 32 unread samples. The FIFO\_OVR bit is reset when the first sample set has been read.
- Diff5 bit is used together with bits Diff[4:0] to provide information of how many FIFO entries are used (000000b means FIFO empty, 100000b means FIFO full). This flag can be routed to the INT1 or INT2 pin (see [Section 6.3 FIFO interrupts](#)).

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



**Table 28. FIFO\_SAMPLES behavior assuming FTH[4:0] = 15**

FIFO_FTH	Diff5 (FIFO_FULL)	FIFO_OVR	Diff[4:0]	Unread FIFO samples	Timing
0	0	0	00000	0	t <sub>0</sub>
0	0	0	00001	1	t <sub>0</sub> + 1/ODR
0	0	0	00010	2	t <sub>0</sub> + 2/ODR
...	...	...	...	...	...
0	0	0	01110	14	t <sub>0</sub> + 14/ODR
1	0	0	01111	15	t <sub>0</sub> + 15/ODR
...	...	...	...	...	...
1	0	0	11111	31	t <sub>0</sub> + 31/ODR
1	1	0	00000	32	t <sub>0</sub> + 32/ODR
1	1	1	00000	32	t <sub>0</sub> + 33/ODR

## 6.3 FIFO interrupts

There are three specific FIFO events that can be routed to the pins in order to interrupt the main processor: FIFO threshold, FIFO full, and FIFO overrun.

All FIFO events can be routed to the INT1 and INT2 pins.

### 6.3.1 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 31 using the FTH[4:0] field in the FIFO\_CTRL register.

If the number of entries in FIFO (Diff[5:0]) is greater than or equal to the value programmed in FTH[4:0], the FIFO\_FTH bit is set high in the FIFO\_SAMPLES register.

Diff[5:0] increases by one step at the ODR frequency and decreases by one step every time that a sample set reading is performed by the user.

The threshold flag (FIFO\_FTH) can be routed to the INT1 and INT2 pins to provide a dedicated interrupt for the application processor that can, as a consequence, consume less power between interrupts. The INT1\_FTH bit of CTRL4\_INT1\_PAD\_CTRL register and the INT2\_FTH bit of CTRL5\_INT2\_PAD\_CTRL register are dedicated to this task.

### 6.3.2 FIFO full

It is possible to configure the device to generate an interrupt whenever the FIFO becomes full. To do so, just set the INT1\_DIFF5 bit of the CTRL4\_INT1\_PAD\_CTRL register to '1' (or the INT2\_DIFF5 bit of the CTRL5\_INT2\_PAD\_CTRL register to '1'). To avoid losing samples, the FIFO reading operation must start and complete inside 1 ODR window.

### 6.3.3 FIFO overrun

It is possible to configure the device to generate an interrupt if the overrun event occurs in FIFO. To do so just set the INT2\_OVR bit of the CTRL5\_INT2\_PAD\_CTRL register to '1'.

## 6.4 FIFO modes

The IIS2DLPC FIFO buffer can be configured to operate in five 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 usable functions in application development.

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

### 6.4.1 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 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.

### 6.4.2 FIFO mode

In FIFO mode, the buffer continues filling until full (32 sample sets stored). As soon as the FIFO\_OVR flag goes 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 register.

By selecting this mode, FIFO starts data collection and Diff[5:0] 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 32 sample set reading 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 reading procedure it is necessary to exit Bypass mode.

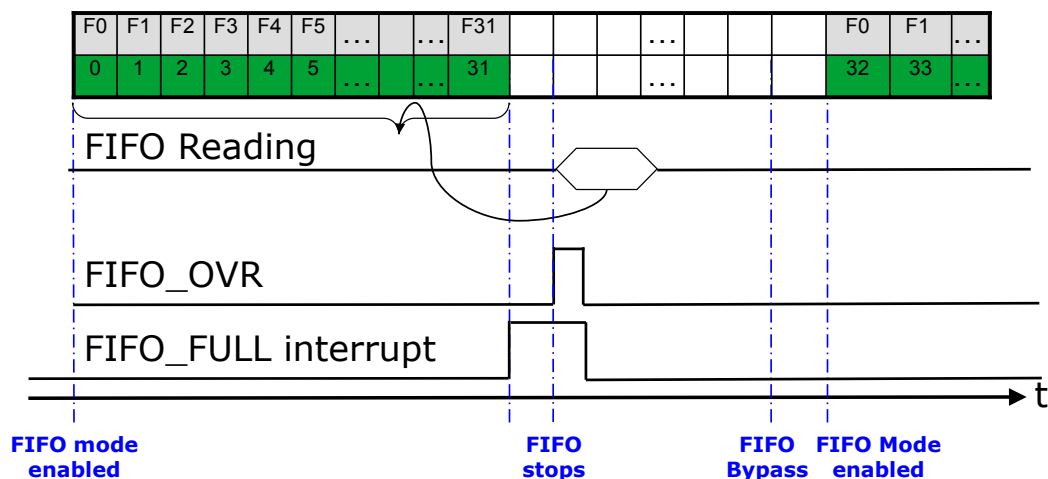
In order to serve the FIFO full (Diff5 bit) event as soon as possible, it is recommended to route it to the pin in order to generate an interrupt, which will then be managed by a specific handler:

1. Set INT1\_DIFF5 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 18. FIFO mode behavior**



As indicated in [Figure 18. FIFO mode behavior](#), when FIFO mode is enabled, the buffer starts to collect data and fills all 32 slots (from F0 to F31) 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 reading procedure may be performed inside an interrupt handler triggered by a FIFO FULL condition (Diff5) and it is composed of a 32 sample set of 6 bytes for a total of 192 bytes and retrieves data starting from the oldest sample stored in FIFO (F0). The 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.

### 6.4.3 Continuous mode

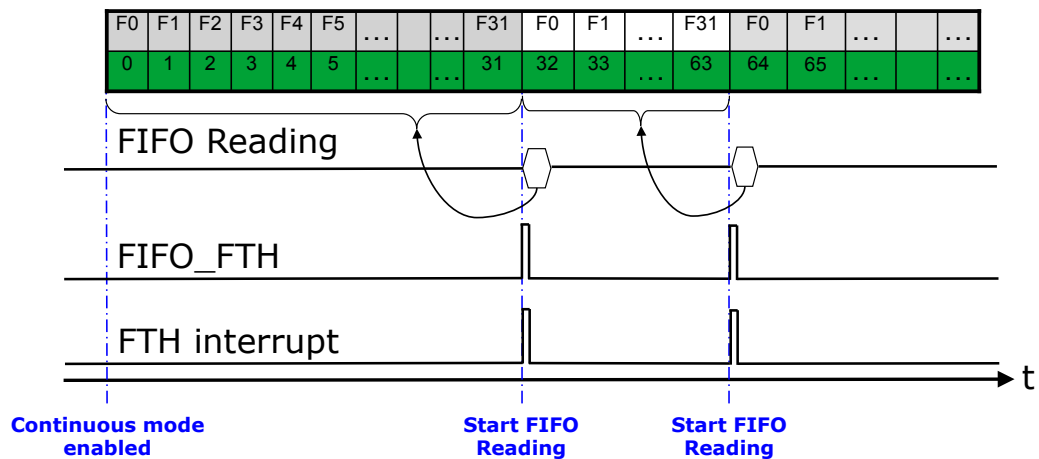
In Continuous mode FIFO continues filling, when the buffer is full, the FIFO index restarts from the beginning and older data is replaced by current data. The oldest values continue to be overwritten until a read operation frees FIFO slots. The host processor reading speed is most important in order 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[4:0] to 31.
2. Set INT1\_FTH to '1': Enable FIFO threshold interrupt
3. Activate Continuous mode by setting the FMode[2:0] field to 110b in the FIFO\_CTRL register (2Eh).

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

**Figure 19. Continuous mode with interrupt trigger**



As indicated in [Figure 19. Continuous mode with interrupt trigger](#), when Continuous mode is enabled, the FIFO buffer is continuously filling (from F0 to F31) at the selected output data rate. When the buffer is full, the FTH interrupt (as well as the FIFO\_FULL condition indicated by the Diff5 bit in FIFO\_SAMPLES (2Fh), which might also be used to trigger an interrupt) goes high, and the application processor may read all FIFO samples (32 \* 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 6.5 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/I<sup>2</sup>C register and the current oldest FIFO value is shifted into the output registers in order to allow the next read operation.

#### 6.4.4 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 (i.e. wake-up, free-fall, single or double tap, 6D/4D, or any combination of them) occurs.

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

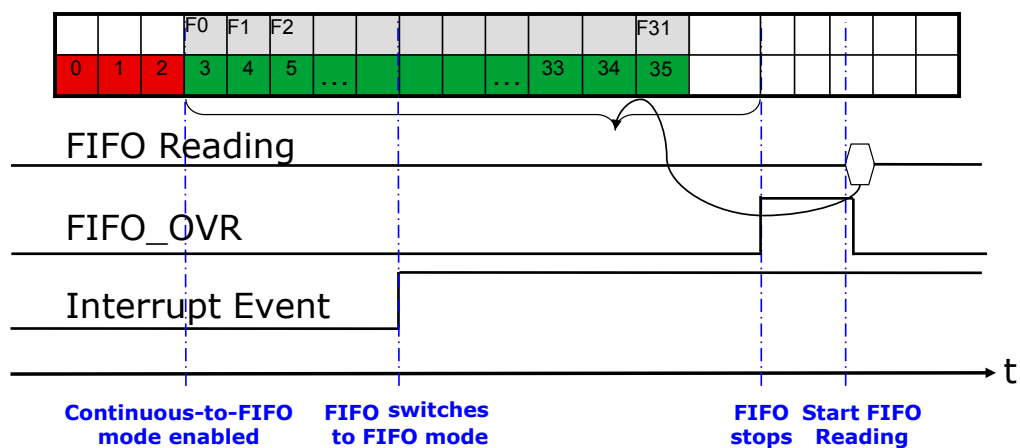
Follow these steps for Continuous-to-FIFO mode configuration:

1. Configure the desired interrupt generator by following the instructions in [Section 5 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 register (2Eh).

*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 or INT2 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 20. Continuous-to-FIFO mode: interrupt latched and non-latched](#)).

**Figure 20. Continuous-to-FIFO mode: interrupt latched and non-latched**



### 6.4.5 Bypass-to-Continuous mode

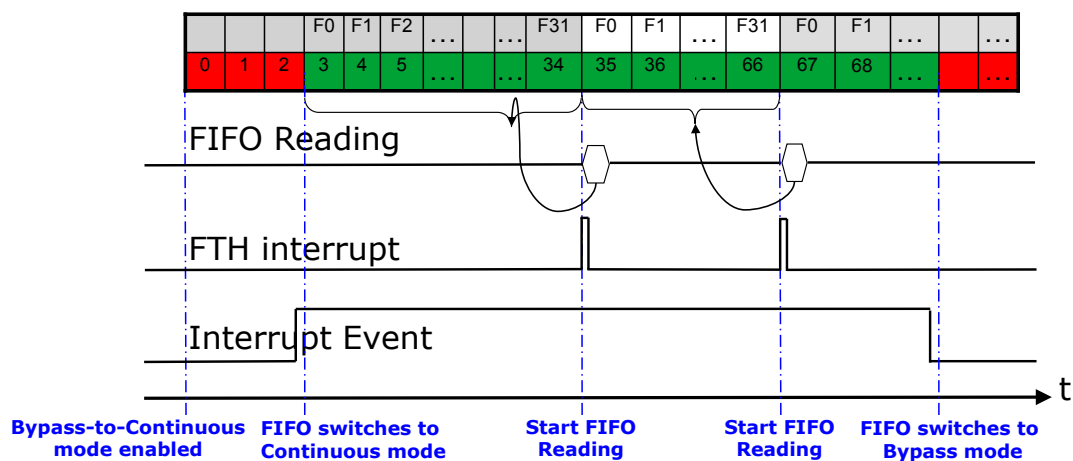
This mode is a combination of the Bypass and Continuous modes previously described. In Bypass-to-Continuous mode, the FIFO buffer starts in Bypass mode and switches to Continuous mode when the selected interrupt (i.e. wake-up, free-fall, single or double tap, 6D/4D, or any combination of them) occurs.

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

1. Configure desired interrupt generator by following the instructions in [Section 5 Interrupt generation and embedded functions](#) (be sure it is latched).
2. Set FTH[4:0] to 31.
3. Set INT1\_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 register (2Eh).

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

**Figure 21. Bypass-to-Continuous mode**



As indicated in [Figure 21. Bypass-to-Continuous mode](#) 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 (32 \* 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.

## 6.5 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/I<sup>2</sup>C 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 32 read operations from the accelerometer output registers. The size of the data stored in FIFO is dependent on the selected 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 reading byte combination in order to increase application flexibility (ex: 192 single byte read, 32 reads of 6 bytes, 1 multiple read of 192 bytes, etc.).

It is recommended to read all FIFO slots in a multiple byte read of 192 bytes (6 output registers by 32 slots). In order to minimize communication between the master and slave, the reading address may be automatically incremented by the device by setting the IF\_ADD\_INC bit of CTRL2 register to '1'; the device rolls back to 0x28 when register 0x2D is reached.

The I<sup>2</sup>C 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 I<sup>2</sup>C mode being used (max rate 100 kHz), a single sample set reading takes 830  $\mu$ s while total FIFO download takes about 17.57 ms (29 + 9 \* 192 clock pulses).

In the case of the SPI, instead, 8 clock pulses are required only once at the very beginning to get started (for the register address, including the r/w bit) plus an additional 8 clock pulses for every byte to read. With a 2 MHz clock a single sample set reading would take 28  $\mu$ s, while total FIFO download takes about 772  $\mu$ s.

If this recommendation were followed, using a standard I<sup>2</sup>C (100 kHz) the complete FIFO reading (17.57 ms) is taking 28/ODR with ODR at 1600 Hz. Using a SPI @ 2 MHz (10 MHz is the maximum supported by the device) the complete FIFO reading would take about two periods of data generation (2\*1/ODR) with ODR at 1600 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 6.3 FIFO interrupts](#)).

**Table 29. Example: threshold function of ODR**

ODR (Hz)	FTH_THS (I <sup>2</sup> C @ 100 kHz)	FTH_THS (I <sup>2</sup> C @ 400 kHz)	FTH_THS (SPI @ 2 MHz)
50	32	32	32
100	17	32	32
200	8	32	32
400	4	17	32
800	1	8	32
1600	-	4	25

## 7 Temperature sensor

The IIS2DLPC is provided with an internal temperature sensor that is suitable for ambient temperature measurement.

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

Bit DRDY\_T in STATUS\_DUP (37h) is set high when a new set of data is available and is reset when one of the temperature data outputs (OUT\_T\_H or OUT\_T) is read. The DRDY\_T bit can be routed on the INT2 pin through bit INT2\_DRDY\_T of the CTRL5\_INT2\_PAD\_CTRL register.

Temperature DRDY interrupt can be pulsed using the DRDY\_PULSED bit of the CTRL7 register: the pulse duration is 78  $\mu$ s (typical). Pulsed mode is not applied to the DRDY\_T bit which is always latched.

The temperature data is represented as a number of 12 bits in two's complement format, left-aligned in the OUT\_T\_L and OUT\_T\_H registers. A duplicated value of OUT\_T\_H in register OUT\_T is also available in order to provide 8 bits in two's complement format, temperature sequentially readable with the sensor outputs. See table below for temperature sensor details.

**Table 30. Temperature sensor characteristics**

Symbol	Parameter	Min.	Typ. <sup>(1)</sup>	Max.	Unit
TsDr	Temperature sensor output change vs. temperature		1 <sup>(2)</sup>		LSB/°C
			16 <sup>(3)</sup>		
TODR	Temperature refresh rate in High-Performance Mode for all ODRs or in Low-Power Modes for ODRs equal to 200/100/50 Hz		50		Hz
	Temperature refresh rate in Low-Power Modes for ODR equal to 25 Hz		25		
	Temperature refresh rate in Low-Power Modes for ODR equal to 12.5 Hz		12.5		
	Temperature refresh rate in Low-Power Modes for ODR equal to 1.6 Hz		1.6		

1. Typical specifications are not guaranteed.
2. 8-bit resolution (i.e. when using the OUT\_T register)
3. 12-bit resolution (i.e. when using the OUT\_T\_L and OUT\_T\_H registers)

### 7.1 Example of temperature data calculation

Table 31. 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 31. Output data registers content vs. temperature**

Temperature values	OUT_T (26h)	OUT_T_H (0Eh)	OUT_T_L (0Dh)
23.5 °C	FEh	FEh	80h
25.0 °C	00h	00h	00h
26.5 °C	01h	01h	80h

## 8 Self-test

The embedded self-test functions allow checking device functionality without moving it.

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. 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 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 data produced by the electrostatic test-force and gravity.

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 absolute value for each axis and verifying that it falls within a given range. The min and max values are provided in the datasheet.

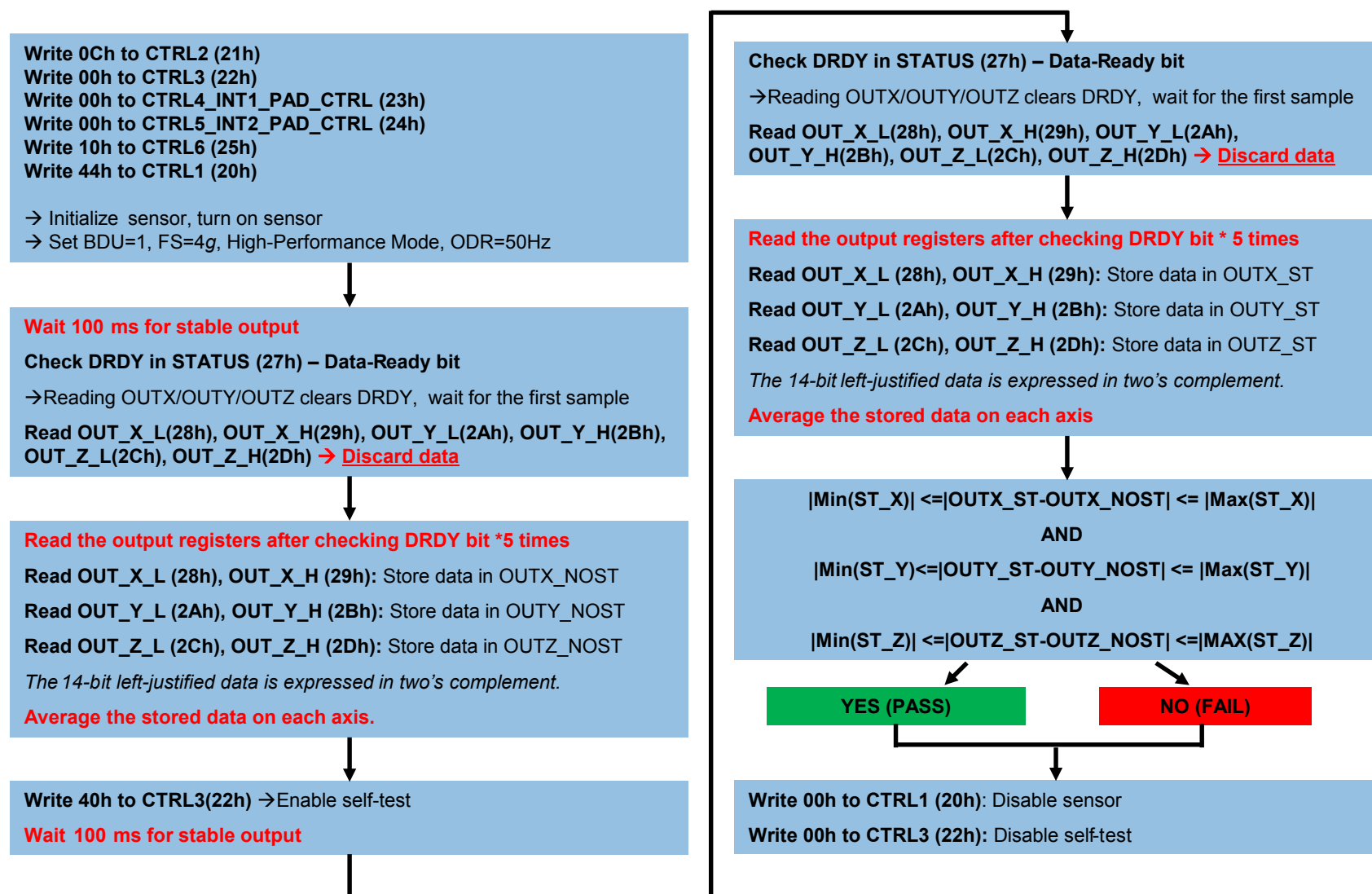
The complete accelerometer self-test procedure is indicated in [Figure 22. Accelerometer self-test procedure](#).

*Notes:*

1. *Keep the device still during the self-test procedure.*
2. *The full scale and data rate used in the self-test procedure are not mandatory but recommended.*
3. *Please refer to the datasheet for minimum and maximum values.*



Figure 22. Accelerometer self-test procedure



Note: The wait time of 100 ms is not mandatory but recommended. In any case, the settling time should be taken into account.

## Revision history

**Table 32. Document revision history**

Date	Version	Changes
27-Aug-2018	1	Initial release
21-Dec-2018	2	Updated Section 3.4 Accelerometer bandwidth Updated Section 5.7 Activity/Inactivity recognition Updated Figure 22. Accelerometer self-test procedure
14-Aug-2020	3	Updated pin status of pin 1 and pin 4 in Table 1. Internal pin status Updated Section 3.1 Power mode Updated Section 3.3 Single data conversion (on-demand mode) Updated Figure 3. Accelerometer filtering chain diagram Updated Section 4.1 Startup sequence Updated Section 4.3 Using the data-ready signal Updated Section 5 Interrupt generation and embedded functions Updated Section 5.1 Interrupt pin configuration Updated Section 5.3 Free-fall interrupt Updated Section 5.4 Wake-up interrupt Updated Section 5.5.1 6D orientation detection Updated Section 5.6.3 Single-tap and double-tap recognition configuration Updated Section 5.7 Activity/Inactivity recognition Updated Section 6.5 Retrieving data from FIFO Updated Section 7 Temperature sensor Updated Table 31. Output data registers content vs. temperature Minor textual changes
21-Jan-2021	4	Updated <a href="#">Section 3.5 High-pass filter configuration</a> Updated <a href="#">Section 5.3 Free-fall interrupt</a>

## Contents

<b>1</b>	<b>Pin description</b>	<b>2</b>
<b>2</b>	<b>Registers</b>	<b>3</b>
<b>3</b>	<b>Operating modes</b>	<b>5</b>
3.1	Power mode	5
3.2	Continuous conversion	8
3.3	Single data conversion (on-demand mode)	9
3.4	Accelerometer bandwidth	10
3.5	High-pass filter configuration	13
3.6	Frequency response	14
<b>4</b>	<b>Reading output data</b>	<b>17</b>
4.1	Startup sequence	17
4.2	Using the status register	17
4.3	Using the data-ready signal	17
4.4	Using the block data update (BDU) feature	18
4.5	Understanding output data	18
4.5.1	Example of output data	19
<b>5</b>	<b>Interrupt generation and embedded functions</b>	<b>20</b>
5.1	Interrupt pin configuration	21
5.2	Event status	22
5.3	Free-fall interrupt	23
5.4	Wake-up interrupt	25
5.5	6D/4D orientation detection	27
5.5.1	6D orientation detection	27
5.5.2	4D orientation detection	29
5.6	Single-tap and double-tap recognition	29
5.6.1	Single-tap	30
5.6.2	Double tap	31
5.6.3	Single-tap and double-tap recognition configuration	32
5.6.4	Single-tap example	34
5.6.5	Double-tap example	35

5.7	Activity/Inactivity recognition .....	35
5.8	Stationary/Motion detection .....	38
5.9	Boot status .....	38
<b>6</b>	<b>First-in first-out (FIFO) buffer .....</b>	<b>39</b>
6.1	FIFO description .....	39
6.2	FIFO registers .....	40
6.2.1	FIFO_CTRL (2Eh) .....	40
6.2.2	FIFO_SAMPLES (2Fh) .....	40
6.3	FIFO interrupts .....	41
6.3.1	FIFO threshold .....	41
6.3.2	FIFO full .....	41
6.3.3	FIFO overrun .....	41
6.4	FIFO modes .....	41
6.4.1	Bypass mode .....	42
6.4.2	FIFO mode .....	42
6.4.3	Continuous mode .....	43
6.4.4	Continuous-to-FIFO mode .....	44
6.4.5	Bypass-to-Continuous mode .....	45
6.5	Retrieving data from FIFO .....	46
<b>7</b>	<b>Temperature sensor .....</b>	<b>47</b>
7.1	Example of temperature data calculation .....	47
<b>8</b>	<b>Self-test .....</b>	<b>48</b>
	<b>Revision history .....</b>	<b>50</b>

## List of tables

<b>Table 1.</b>	Internal pin status . . . . .	2
<b>Table 2.</b>	Registers . . . . .	3
<b>Table 3.</b>	Accelerometer resolution . . . . .	5
<b>Table 4.</b>	CTRL1 register . . . . .	5
<b>Table 5.</b>	Mode selection . . . . .	5
<b>Table 6.</b>	Low-power mode selection . . . . .	5
<b>Table 7.</b>	Power consumption [ $\mu$ A] . . . . .	6
<b>Table 8.</b>	High-Performance Mode: noise density [ $\mu$ g/ $\sqrt{\text{Hz}}$ ] . . . . .	6
<b>Table 9.</b>	Low-Power Mode: RMS noise [mg(RMS)] . . . . .	7
<b>Table 10.</b>	Output data rate selection . . . . .	8
<b>Table 11.</b>	Low-power mode selection . . . . .	9
<b>Table 12.</b>	Low-pass filter 1 bandwidth . . . . .	11
<b>Table 13.</b>	Bandwidth: low-pass path . . . . .	11
<b>Table 14.</b>	Bandwidth: high-pass path . . . . .	12
<b>Table 15.</b>	Sensitivity . . . . .	18
<b>Table 16.</b>	CTRL4_INT1_PAD_CTRL . . . . .	21
<b>Table 17.</b>	CTRL5_INT2_PAD_CTRL . . . . .	22
<b>Table 18.</b>	Free-fall threshold value . . . . .	23
<b>Table 19.</b>	SIXD_SRC register . . . . .	27
<b>Table 20.</b>	Threshold for 4D/6D function . . . . .	27
<b>Table 21.</b>	SIXD_SRC register for 6D positions . . . . .	28
<b>Table 22.</b>	TAP_SRC register . . . . .	33
<b>Table 23.</b>	Pulse duration on interrupt pin . . . . .	36
<b>Table 24.</b>	FIFO buffer full representation (32 <sup>nd</sup> sample set stored) . . . . .	39
<b>Table 25.</b>	FIFO buffer full representation (33 <sup>rd</sup> sample set stored and 1 <sup>st</sup> sample discarded) . . . . .	39
<b>Table 26.</b>	FIFO_CTRL register . . . . .	40
<b>Table 27.</b>	FIFO_SAMPLES register . . . . .	40
<b>Table 28.</b>	FIFO_SAMPLES behavior assuming FTH[4:0] = 15 . . . . .	41
<b>Table 29.</b>	Example: threshold function of ODR . . . . .	46
<b>Table 30.</b>	Temperature sensor characteristics . . . . .	47
<b>Table 31.</b>	Output data registers content vs. temperature . . . . .	47
<b>Table 32.</b>	Document revision history . . . . .	50

## List of figures

<b>Figure 1.</b>	Pin connections . . . . .	2
<b>Figure 2.</b>	Single data conversion using INT2 as external trigger (SLP_MODE_SEL = 0) . . . . .	9
<b>Figure 3.</b>	Accelerometer filtering chain diagram . . . . .	10
<b>Figure 4.</b>	High-pass filter in normal and reference mode . . . . .	13
<b>Figure 5.</b>	LPF1 - High-Performance mode, ODR = 1600 Hz, BW = 400 Hz. . . . .	14
<b>Figure 6.</b>	LPF1 / LPF2 - High-Performance mode, ODR 800 Hz . . . . .	15
<b>Figure 7.</b>	LPF1 / LPF2 - Low-Power mode 4 - ODR 200 Hz . . . . .	16
<b>Figure 8.</b>	Data-ready signal . . . . .	18
<b>Figure 9.</b>	Embedded functions . . . . .	20
<b>Figure 10.</b>	Interrupt pin configuration . . . . .	21
<b>Figure 11.</b>	Free-fall interrupt . . . . .	23
<b>Figure 12.</b>	Wake-up event recognition (using the HP filter). . . . .	25
<b>Figure 13.</b>	6D recognized orientations . . . . .	28
<b>Figure 14.</b>	Single-tap event recognition . . . . .	30
<b>Figure 15.</b>	Double-tap event recognition (LIR bit = 0) . . . . .	31
<b>Figure 16.</b>	Single and double-tap recognition (LIR bit = 0) . . . . .	33
<b>Figure 17.</b>	Activity/Inactivity recognition (using the HP filter). . . . .	37
<b>Figure 18.</b>	FIFO mode behavior . . . . .	42
<b>Figure 19.</b>	Continuous mode with interrupt trigger. . . . .	43
<b>Figure 20.</b>	Continuous-to-FIFO mode: interrupt latched and non-latched . . . . .	44
<b>Figure 21.</b>	Bypass-to-Continuous mode . . . . .	45
<b>Figure 22.</b>	Accelerometer self-test procedure. . . . .	49

**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. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). 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.

© 2021 STMicroelectronics – All rights reserved