Application note

LIS3DHH: three-axis digital output accelerometer

Introduction

This document is intended to provide usage information and application hints related to ST's LIS3DHH motion sensor.

The LIS3DHH is a 3D digital accelerometer system-in-package with a digital SPI serial interface standard output, performing with a zero-*g* noise density of 45 µg/√Hz. Thanks to its ultra-low noise performance and excellent stability over temperature and time, the device is suitable for applications targeting precision inclinometers, platform and antenna stabilization and leveling instruments.

The device has a full-scale acceleration range of ±2.5 *g* and is capable of measuring accelerations with an output data rate of 1100 Hz. The LIS3DHH embeds a DSP engine with a configurable low-pass filter.

The LIS3DHH has an integrated 32-level first-in, first-out (FIFO) buffer allowing the user to store data in order to limit intervention by the host processor.

The LIS3DHH is available in a high-performance (low-stress) ceramic cavity land grid array (CC LGA) package and can operate within a temperature range of -40 °C to +85 °C.

1 Pin description

Figure 1. Pin connections

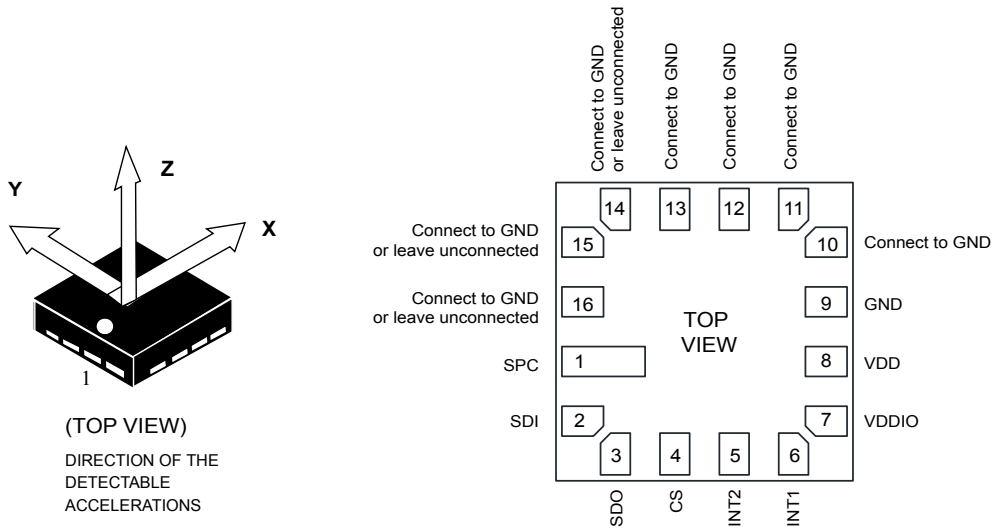


Table 1. Pin status

Pin #	Name	Function	Pin status
1	SPC	SPI serial port clock	Default: input open-drain
2	SDI	SPI serial data input	Default: input open-drain
3	SDO	SPI serial data output	Default: input open-drain
4	CS	SPI mode selection (1:SPI idle mode; 0: SPI communication mode)	Default: input open-drain
5	INT2	Programmable interrupt 2	Default: push-pull output forced to ground
6	INT1	Programmable interrupt 1	Default: push-pull output forced to ground
7	Vdd_IO	Power supply for I/O pins	Power supply for I/O pins
8	Vdd	Power supply	Power supply
9	GND	0 V power supply	0 V power supply
10	Reserved	Connect to GND	Connect to GND
11	Reserved	Connect to GND	Connect to GND
12	Reserved	Connect to GND	Connect to GND
13	Reserved	Connect to GND	Connect to GND
14	Reserved	Connect to GND or leave unconnected	Connect to GND or leave unconnected
15	Reserved	Connect to GND or leave unconnected	Connect to GND or leave unconnected
16	Reserved	Connect to GND or leave unconnected	Connect to GND or leave unconnected

Note: The INT1 and INT2 pins can be configured to operate in open-drain mode by asserting the bits PP_OD_INT1 and PP_OD_INT2 of CTRL_REG4 respectively.

2 Registers

Table 2. Registers

Register name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WHO_AM_I	0Fh	0	0	0	1	0	0	0	1
CTRL_REG1	20h	NORM_ MOD_EN	IF_ADD_INC	0	0	BOOT	SW_RESET	DRDY_PULSE	BDU
INT1_CTRL	21h	INT1_DRDY	INT1_BOOT	INT1_OVR	INT1_FSS5	INT1_FTH	INT1_EXT	0	0
INT2_CTRL	22h	INT2_DRDY	INT2_BOOT	INT2_OVR	INT2_FSS5	INT2_FTH	0	0	0
CTRL_REG4	23h	DSP_LP_ TYPE	DSP_BW_ SEL	ST2	ST1	PP_OD_ INT2	PP_OD_ INT1	FIFO_EN	1(1)
CTRL_REG5	24h	0	0	0	0	0	0	0	FIFO_SPI _HS_ON
OUT_TEMP_L	25h	Temp3	Temp2	Temp1	Temp0	0	0	0	0
OUT_TEMP_H	26h	Temp11	Temp10	Temp9	Temp8	Temp7	Temp6	Temp5	Temp4
STATUS	27h	ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA
OUT_X_L_XL	28h	D7	D6	D5	D4	D3	D2	D1	D0
OUT_X_H_XL	29h	D15	D14	D13	D12	D11	D10	D9	D8
OUT_Y_L_XL	2Ah	D7	D6	D5	D4	D3	D2	D1	D0
OUT_Y_H_XL	2Bh	D15	D14	D13	D12	D11	D10	D9	D8
OUT_Z_L_XL	2Ch	D7	D6	D5	D4	D3	D2	D1	D0
OUT_Z_H_XL	2Dh	D15	D14	D13	D12	D11	D10	D9	D8
FIFO_CTRL	2Eh	FMODE2	FMODE1	FMODE0	FTH4	FTH3	FTH2	FTH1	FTH0
FIFO_SRC	2Fh	FTH	OVRN	FSS5	FSS4	FSS3	FSS2	FSS1	FSS0

1. *This bit must be set to 1 by the user for correct operation of the device.*

3 Operating modes

The LIS3DHH provides two possible operating configurations:

- Power-Down mode;
- Normal mode.

After the power supply is applied (the required power-up sequence is given in the datasheet), the LIS3DHH performs a 10 ms boot procedure to load the trimming parameters. After the boot is completed, the accelerometer is automatically configured in Power-Down mode.

When the sensor is in Power-Down mode, almost all internal blocks of the device are switched off. The SPI digital interface remains active to allow communication with the device. The content of the configuration registers is preserved and the output data registers are not updated, keeping the last data sampled in memory before going into Power-Down mode.

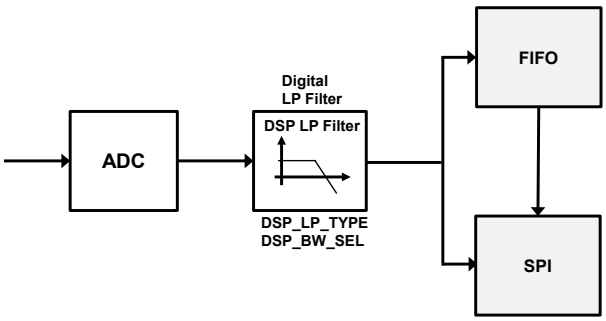
The LIS3DHH can be configured in Normal mode by setting the NORM_MOD_EN bit of the CTRL_REG1 register: the sensor provides acceleration data at an output data rate of 1100 Hz. The device has a full-scale acceleration range of ±2.5 *g*.

3.1 Accelerometer filtering chain

The accelerometer sampling chain is represented by a cascade of two main blocks: an ADC converter and a digital low-pass filter.

Figure 2. Accelerometer filtering chain shows the accelerometer sampling chain.

Figure 2. Accelerometer filtering chain



The desired filter type (FIR or IIR) and bandwidth (235 Hz or 440 Hz) of the digital low-pass filter can be configured, respectively, by the DSP_LP_TYPE and DSP_BW_SEL bits of the CTRL_REG4 register. Table 3 summarizes all the possible configurations and the relative settling time in terms of samples to be discarded. It is recommended to discard the number of samples specified when exiting Power-Down mode or if reconfiguring the filtering chain at runtime.

Table 3. Digital LP filter type and bandwidth

DSP_LP_TYPE	DSP_BW_SEL	Filter	Bandwidth [Hz]	Settling time (samples to be discarded)(1)
0	0	Linear phase FIR	440	1
	1	Linear phase FIR	235	2
1	0	Nonlinear phase IIR	440	2
	1	Nonlinear phase IIR	235	4

1. *Settling time @ 99% of the final value.*

3.2 Boot status

After the device is powered up, the LIS3DHH performs a 10 ms boot procedure to load the trimming parameters. After the boot is completed, the sensor is automatically configured in Power-Down mode. During the boot time the registers are not accessible.

After power up, the trimming parameters can be re-loaded by setting the BOOT bit of the CTRL_REG1 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 the reset to the default value of the control registers is required, it can be performed by setting the SW_RESET bit of the CTRL_REG1 register to 1. The SW_RESET procedure can take 5 µs; the status of reset is signaled by the status of the SW_RESET bit of the CTRL_REG1 register: once the reset is completed, this bit is automatically set low.

The reset procedure reconfigures the following registers to their default values: CTRL_REG1, INT1_CTRL, INT2_CTRL, CTRL_REG4, CTRL_REG5 and FIFO_CTRL.

The boot status signal can be driven to the interrupt pins by setting the INT1_BOOT bit of the INT1_CTRL register to 1 or the INT2_BOOT bit of the INT2_CTRL register: this signal is set high while the boot is running and it is set low again at the end of the boot procedure.

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

1. Set the SW_RESET bit of the CTRL_REG1 register to 1;
2. Wait 5 µs (or wait until the SW_RESET bit of the CTRL_REG1 register returns to 0);
3. Set the BOOT bit of the CTRL_REG1 register to 1;
4. Wait 10 ms.

In order to avoid conflicts, the reboot and the sw reset must not be executed at the same time (do not set to 1 at the same time both the BOOT bit and SW_RESET bit of the CTRL_REG1 register).

4 Reading output data

4.1 Startup sequence

Once the device is powered up, it automatically downloads the calibration coefficients from the embedded flash to the internal registers. When the boot procedure is completed, the accelerometer automatically enters Power-Down mode.

To turn on the accelerometer and gather acceleration data through the SPI interface, Normal mode must be selected through the NORM_MOD_EN bit of the CTRL_REG1 register.

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

1.	Write CTRL_REG1 = C0h	// Enables Normal mode and auto-increment
2.	Write CTRL_REG4 = 01h	// Bit 0 must be set to 1 for correct operation of the device
3.	Write INT1_CTRL = 80h	// Data-ready interrupt on INT1 pin

4.2 Using the status register

The device is provided with a STATUS_REG register which should be polled to check when a new set of data is available. The ZYXDA bit is set to 1 when a new set of data is available.

The read(s) should be performed as follows:

1. Read STATUS
2. If ZYXDA = 0, then go to 1
3. Read OUTX_L_XL
4. Read OUTX_H_XL
5. Read OUTY_L_XL
6. Read OUTY_H_XL
7. Read OUTZ_L_XL
8. Read OUTZ_H_XL
9. Data processing
10. Go to 1

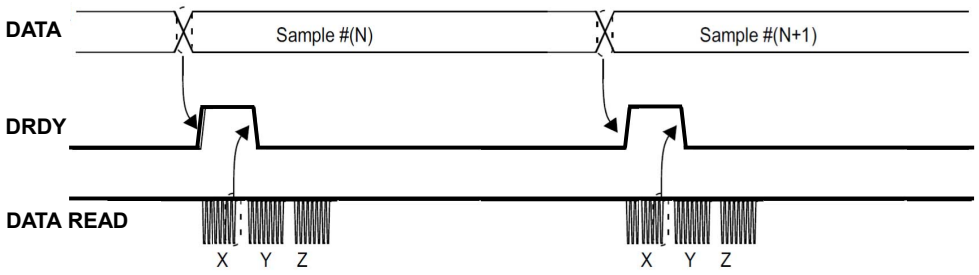
4.3 Using the data-ready signal

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

The data-ready signal is represented by the ZYXDA bit of the STATUS register. The signal can be driven to the INT1 pin by setting the INT1_DRDY bit of the INT1_CTRL register to 1 and to the INT2 pin by setting the INT2_DRDY bit of the INT2_CTRL register to 1.

The data-ready signal rises to 1 when a new set of data has been generated and is available to be read. The data-ready signal can be either latched or pulsed: if the DRDY_PULSE bit of the CTRL_REG1 register is set to 0 (default value), then the data-ready signal is latched and the interrupt is reset when the higher part of one of the channels is read (29h, 2Bh, 2Dh registers). If the DRDY_PULSE bit of the CTRL_REG1 register is set to 1, then the data-ready is pulsed and the duration of the pulse observed on the interrupt pins is typically 227 µs. Pulsed mode is not applied to the ZYXDA bit which is always latched.

Figure 3. Data-ready signal



4.4 Using the block data update (BDU) feature

If reading the output data is particularly slow and cannot be synchronized (or it is not required) with either the ZYXDA 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 CTRL_REG1 register.

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

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

4.5 Understanding output data

The measured acceleration data are sent to the OUTX_H_XL, OUTX_L_XL, OUTY_H_XL, OUTY_L_XL, OUTZ_H_XL, and OUTZ_L_XL registers. These registers contain, respectively, the most significant part and the least significant part of the acceleration signals acting on the X, Y, and Z axes.

The complete output data for the X, Y, Z channels is given by the concatenation OUTX_H_XL & OUTX_L_XL, OUTY_H_XL & OUTY_L_XL, OUTZ_H_XL & OUTZ_L_XL and it is expressed as a two's complement number.

Acceleration output data are represented as 16-bit numbers.

The rounding function is automatically active to auto-address the LIS3DHH registers for a circular burst-mode read. Basically, with a multiple read operation on the accelerometer output registers, the address of the register that is being read goes automatically from the first register to the last register (2Dh) of the pattern and then goes back to the first one (28h).

Note: The IF_ADD_INC bit in CTRL_REG1 must be set to 1 in order to enable SPI interface multiple byte access.

4.5.1 Examples of output data

Table 4. Content of output data registers vs. acceleration provides a few basic examples of the accelerometer data that is read in the data registers when the device is subjected to a given acceleration.

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

Table 4. Content of output data registers vs. acceleration

Acceleration values	Register address	
	OUTX_H_XL (29h)	OUTX_L_XL (28h)
0 *g*	00h	00h
350 m*g*	11h	FDh
1 *g*	33h	66h
-350 m*g*	EEh	03h
-1 *g*	CCh	9Ah

5 First-in first-out (FIFO) buffer

In order to limit intervention by the host processor and to avoid loss of data, the LIS3DHH embeds a first-in, first-out buffer (FIFO) for each of the three output channels, X, Y, and Z..

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 threshold level, the FIFO full and the FIFO overrun events can be enabled to generate dedicated interrupts on the INT1 or INT2 pins. The INT1 pin can be reconfigured as an input in order to guarantee the function of an external asynchronous input trigger for Continuous-to-FIFO and Bypass-to-Continuous modes.

5.1 FIFO description

The FIFO buffer is able to store up to 32 acceleration samples.

The data sample set consists of 6 bytes (Xl, Xh, Yl, Yh, Zl, and Zh) and they are released to the FIFO at 1100 Hz output data rate.

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

Table 5. FIFO buffer full representation (32nd sample set stored)

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

Table 6. FIFO buffer full representation (33rd sample set stored and 1st sample discarded)

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

Table 5. FIFO buffer full representation (32nd sample set stored) represents the FIFO full status when 32 samples are stored in the buffer while Table 6. FIFO buffer full representation (33rd sample set stored and 1st sample discarded) represents the next step when the 33rd sample is inserted into FIFO and the 1st sample is overwritten. The new oldest sample set is made available in the output registers.

5.2 FIFO registers

The FIFO buffer is managed by two different accelerometer registers: the FIFO_CTRL register allows enabling and configuring FIFO behavior, the FIFO_SRC register provides information about the buffer status.

A few other registers are used to route FIFO events on the pin to interrupt the application processor. These are discussed in Section 5.3 FIFO interrupts.

5.2.1 FIFO_CTRL (2Eh)

The FIFO_CTRL register contains the mode in which the FIFO is set. At reset, by default, the FIFO mode is set in Bypass mode, 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 7. 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 5.3.1 FIFO threshold.

5.2.2 FIFO_SRC (2Fh)

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

Table 8. FIFO_CTRL register

b7	b6	b5	b4	b3	b2	b1	b0
FTH	OVRN	FSS5	FSS4	FSS3	FSS2	FSS1	FSS0

- FTH bit is set high when FIFO content exceeds the threshold level selected through the FTH[4:0] bits of the FIFO_CTRL register. This flag can be routed to the INT1 and INT2 pins (see Section 5.3 FIFO interrupts).
- OVRN 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 OVRN bit is reset when the first sample set has been read.
- FSS[5:0] bits of FIFO_SRC register are used to provide information about how many FIFO entries are used (000000b means FIFO empty, 100000b means FIFO full). The FSS5 bit can be routed to the interrupt pins (see Section 5.3 FIFO interrupts).

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

Table 9. FIFO_SRC behavior assuming FTH[4:0] = 15

FTH	OVR	FSS	Unread FIFO samples	Timing
0	0	000000	0	t0
0	0	000001	1	t0 + 1/ODR
0	0	000010	2	t0 + 2/ODR

FTH	OVR	FSS	Unread FIFO samples	Timing
...	...	...	...	...
0	0	001110	14	t0 + 14/ODR
1	0	001111	15	t0 + 15/ODR
...	…	...	...	...
1	0	011111	31	t0 + 31/ODR
1	0	100000	32	t0 + 32/ODR
1	1	100000	33	t0 + 33/ODR

5.3 FIFO interrupts

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

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

5.3.1 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 (FSS[5:0]) is greater than or equal to the value programmed in FTH[4:0], the FTH bit is set high in the FIFO_SRC register.

FSS[5:0] increases by one step at the output data rate frequency and decreases by one step every time that a read of a sample set is performed by the user.

The threshold flag (FTH) can be routed to the INT1 and INT2 pins to provide a dedicated interrupt signal for the application processor. The INT1_FTH bit of INT1_CTRL register and the INT2_FTH bit of INT2_CTRL register are dedicated to this task.

5.3.2 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_FSS5 bit of the INT1_ CTRL register to 1 (or the INT2_FSS5 bit of the INT2_ CTRL register to 1). To avoid losing samples, the FIFO read operations must start and complete within one output data rate time window.

5.3.3 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 INT1_OVRN or INT2_OVRN bit of the INT1_CTRL or INT2_CTRL register to 1.

5.4 FIFO modes

After the LIS3DHH FIFO feature is enabled by setting the FIFO_EN bit in the CTRL_REG4 register to 1, the 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, Continuous-to-FIFO and Bypass-to-Continuous modes are described in the following paragraphs.

5.4.1 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 entire buffer content.

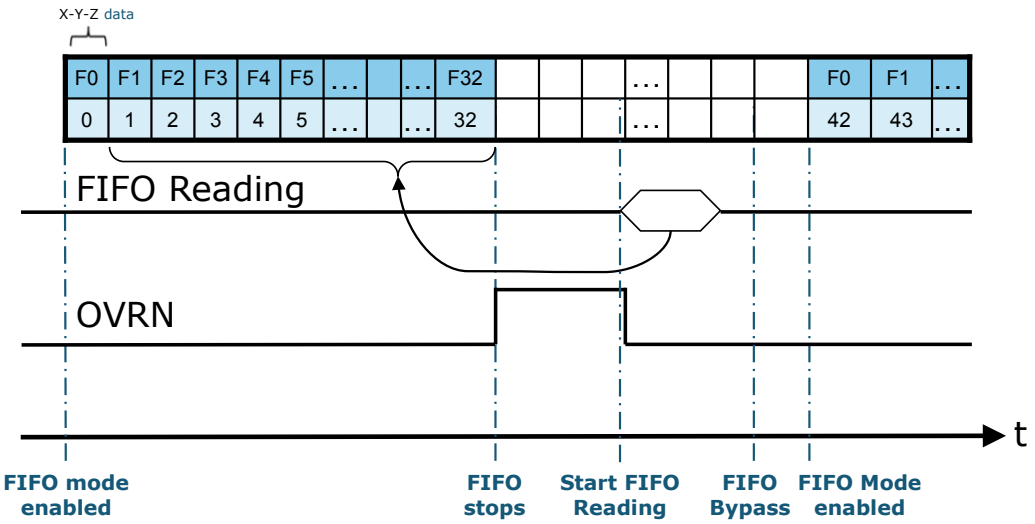
5.4.2 FIFO mode

In FIFO mode, the buffer continues filling until it becomes full. Then it stops collecting data and the FIFO 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 FSS[5:0] changes according to the number of samples stored. At the end of the procedure, the OVRN flag rises to 1 and data can then be retrieved, performing a 32 sample set read from the output registers. Communication speed is not so important in FIFO mode because data collection is stopped and there is no risk of overwriting acquired data. Before restarting FIFO mode, at the end of the read operation, it is necessary to exit Bypass mode.

Figure 4. FIFO mode



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 OVRN 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 OVRN 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.

If data stored in FIFO are read before OVRN event, FIFO does not stop collecting data. For example, if reading FIFO data is synchronized with an FSS5 event, FIFO continuously streams data in the buffer.

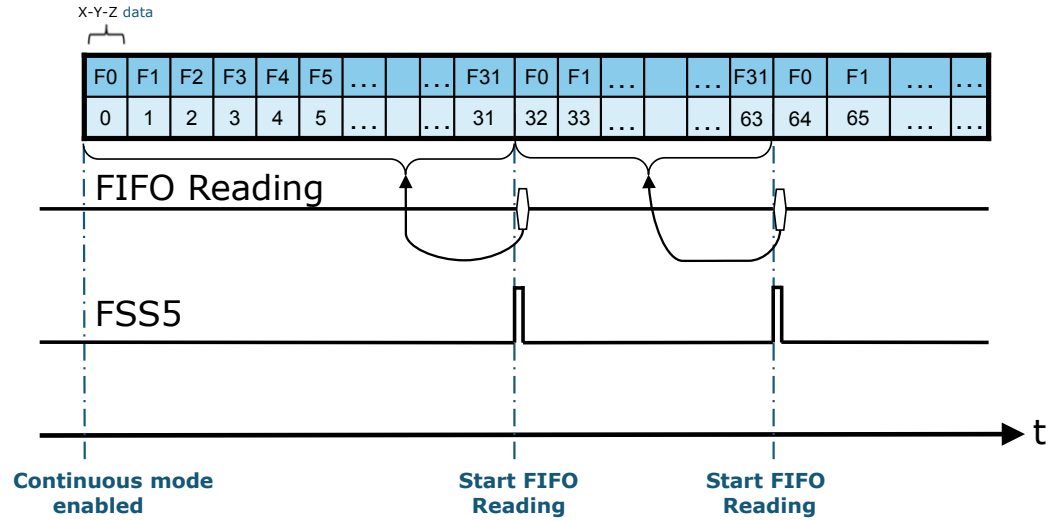
5.4.3 Continuous mode

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 very 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.

As indicated in the following figure, when Continuous mode is enabled, the FIFO buffer is continuously filling (from F0 to F31). When the buffer is full, the FSS5 bit 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 5.5 Retrieving data from the 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 register and the current oldest FIFO value is shifted into the output registers in order to allow the next read operation.

Figure 5. Continuous mode



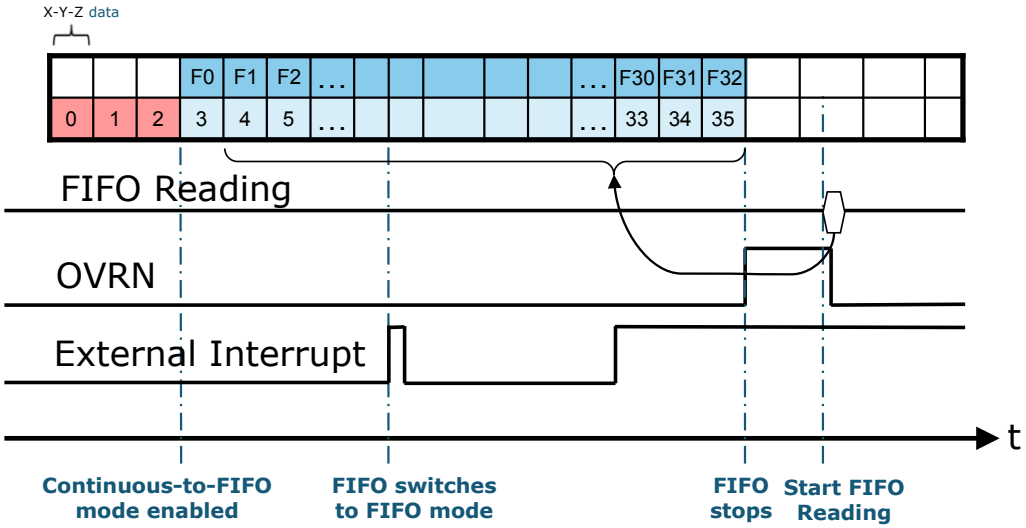
The user can synchronize reading FIFO data to the FSS5 signal, corresponding to the FIFO full event, or to the FTH signal, to check if the number of samples stored in FIFO is equal to or greater than the desired FIFO threshold level (configured through the FTH[4:0] bits of the FIFO_CTRL register).

5.4.4 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 an external event occurs. The INT1 pin needs to be configured as an input by setting the INT1_EXT bit of the INT1_CTRL register and the FIFO switches mode when the external interrupt signal is captured.

While in Continuous mode the FIFO buffer continues filling, when an external event takes place the FIFO mode changes. Then, as soon as the buffer becomes full, the OVRN bit is set high and the next samples overwrite the oldest and the FIFO stops collecting data.

Figure 6. Continuous-to-FIFO mode

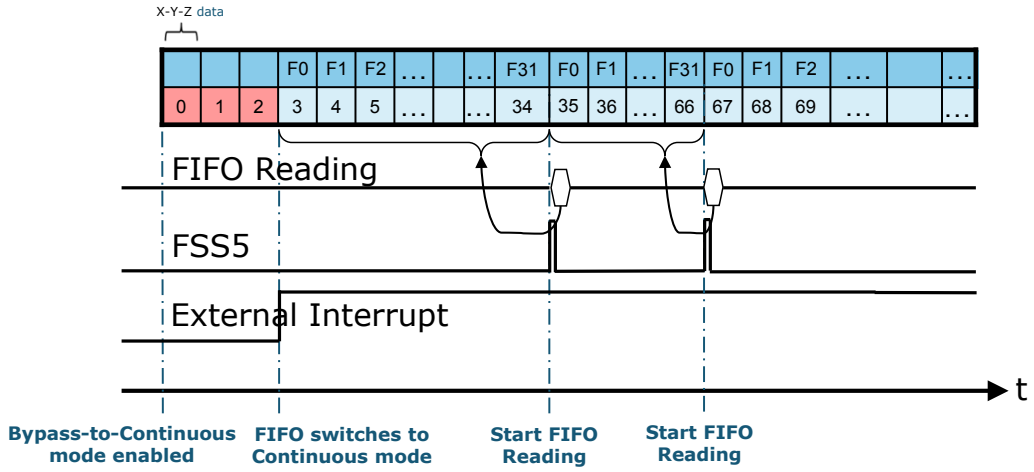


5.4.5 Bypass-to-Continuous mode

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 an external event occurs. The INT1 pin need to be configured as an input by setting the INT1_EXT bit of the INT1_CTRL register and the FIFO switches mode when the external interrupt signal is captured.

While in Bypass mode the FIFO buffer in empty, when an external event takes place the FIFO mode changes and starts to stream data in Continuous mode.

Figure 7. Bypass-to-Continuous mode



5.5 Retrieving data from the 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 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 entire FIFO content is retrieved by performing 32 read operations from the accelerometer output registers.

Data can be retrieved from FIFO using every read byte combination in order to increase application flexibility (ex: 192 single byte reads, 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 the CTRL2 register to 1. The device rolls back to 0x28 when register 0x2D is reached.

Note: it is recommended to set the FIFO_SPI_HS_ON bit in the CTRL_REG5 for SPI clock frequencies higher than 6 MHz.

6 Temperature sensor

The LIS3DHH is provided with an internal temperature sensor that is suitable for ambient temperature measurement.

If the accelerometer is in Power-Down mode, the temperature sensor is off. If the accelerometer is in Normal mode, the output data rate of temperature sensor is typically 68.75 Hz (accelerometer ODR / 16).

The temperature data is given by the concatenation of the OUT_TEMP_H and OUT_TEMP_L registers and it is represented as a number of 12 bits in two's complement format left-justified, with a sensitivity of 16 LSB/°C. The output zero level corresponds to 25 °C.

6.1 Example of temperature data calculation

The following table 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 10. Content of output data registers vs. temperature

Temperature values	Register address	
	OUT_TEMP_H (26h)	OUT_TEMP_L (25h)
0°C	E7h	00h
25°C	00h	00h
50°C	19h	00h

7 Self-test

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

When the self-test is enabled, an actuation force is applied to the sensor, simulating a definite input acceleration. In this case, the sensor outputs exhibit a change in their DC levels which are related to the selected full scale through the sensitivity value.

The accelerometer self-test function can be configured through the ST[1:0] bits of the CTRL_REG4 register. The self-test is off when the ST[1:0] bits are programmed to 00b; it is enabled when the ST[1:0] bits are set to 01b (positive sign self-test) or 10b (negative sign self-test).

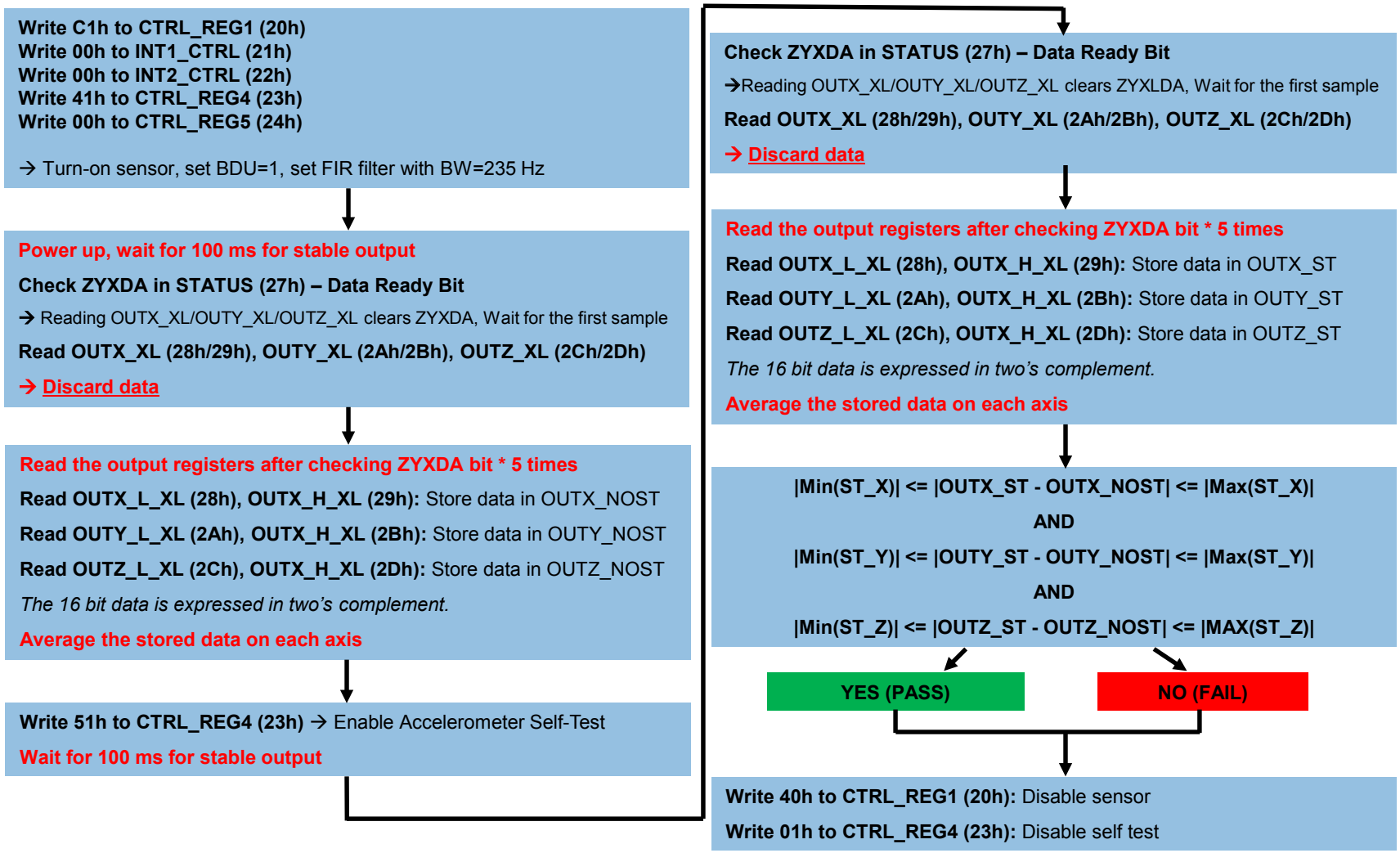
When the accelerometer self-test is activated, the sensor output level is given by the algebraic sum of the signals produced by the acceleration acting on the sensor and by the electrostatic test-force.

The procedure consists of:

1. enabling the accelerometer
2. averaging 5 samples before enabling the self-test
3. averaging 5 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 minimum and maximum values are provided in the datasheet.

The complete self-test procedure is indicated in the following figure.

Figure 8. **Accelerometer self-test procedure**



Revision history

Table 11. **Document revision history**

Date	Revision	Changes
27-Feb-2018	1	Initial release
11-June-2018	2	Updated Table 10. Content of output data registers vs. temperature

Contents

List of tables

List of figures