

Ultralow-power accelerometer with AI & anti-aliasing



LGA-12L
2.0 x 2.0 x 0.74 (max) mm

Product status link

[LIS2DUX12](#)

Product summary

| | |
|------------------------|---------------|
| Order code | LIS2DUX12TR |
| Temperature range [°C] | -40 to +85 |
| Package | LGA-12L |
| Packing | Tape and reel |

Product resources

[TN0018](#) (design and soldering)

Product label



Features

- Supply voltage range from 1.62 V to 3.6 V
 - Independent IO supply (1.62 V to 3.6 V) for I²C and SPI interfaces
 - Independent IO supply (extended range: 1.08 V to 3.6 V) for MIPI I3C[®] interface
- Four operating power modes
- Ultralow power consumption
 - High-performance mode with anti-aliasing filter: 10.8 μA
 - Low-power mode with anti-aliasing filter: 6.2 μA
 - Ultralow-power mode: 2.7 μA
 - Power-down: 0.01 μA
- Low noise down to 220 μg/√Hz
- ±2g/±4g/±8g/±16g programmable full-scale
- ODR from 1.6 Hz to 800 Hz
- Embedded machine learning core
- Programmable finite state machine
- Adaptive self-configuration (ASC) based on the sensor processing output (FSM / MLC)
- Embedded temperature sensor
- Embedded FIFO: up to 512 samples of accelerometer and temperature data in high resolution or up to 768 samples of acceleration data at low resolution
- High-speed I²C/SPI/MIPI I3C[®] digital output interface
- Embedded digital functions (free-fall, wake-up, single/double/triple-tap recognition, activity/inactivity, 6D/4D orientation)
- Advanced pedometer, step detector and step counter
- Significant motion detection, tilt detection
- Self-test
- Small package: 2.0 x 2.0 x 0.74 (max) mm LGA 12-lead
- 10000 g high shock survivability
- [ECOPACK](#) and RoHS compliant

Applications

- [Wearable devices](#) (wristband and smart watches)
- [Game controllers](#)
- Hearing aids and portable healthcare devices
- True wireless stereo
- Wireless sensor nodes
- Motion-activated user interfaces (screen rotation, tap, double tap)
- [Asset trackers](#)

Description

The LIS2DUX12 is a smart, digital, 3-axis linear accelerometer whose MEMS and ASIC have been expressly designed to combine the lowest current consumption possible with features such as always-on anti-aliasing filtering, a finite state machine (FSM) and machine learning core (MLC) with adaptive self-configuration (ASC).

The FSM and MLC with ASC deliver outstanding always-on, edge processing capabilities to the LIS2DUX12. The LIS2DUX12 MIPI I3C[®] slave interface and embedded 128-level FIFO buffer complete a set of features that make this accelerometer a reference in terms of system integration from a standpoint of the bill of materials, processing, or power consumption.

The LIS2DUX12 has user-selectable full scales of $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ and is capable of measuring accelerations with output data rates from 1.6 Hz to 800 Hz.

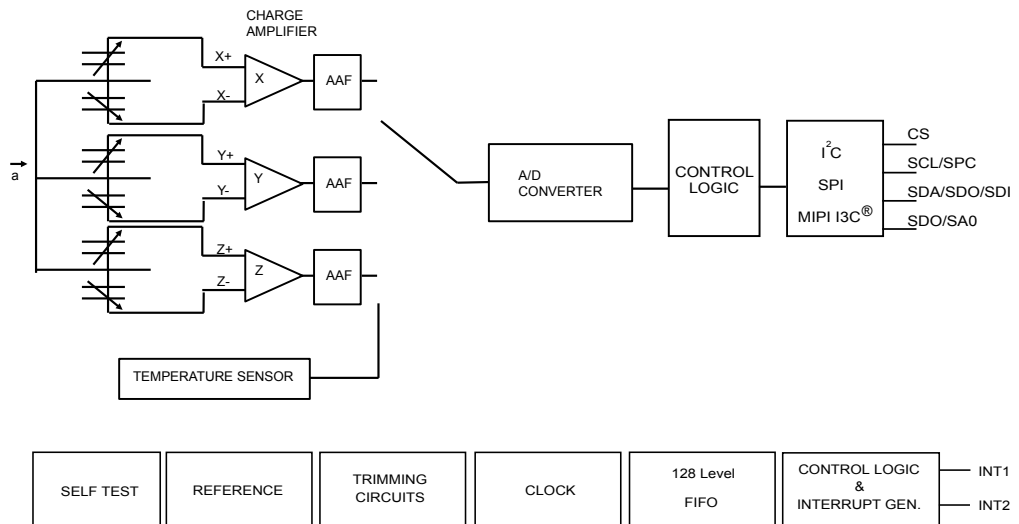
The LIS2DUX12 has a dedicated internal engine to process motion and acceleration detection including free-fall, wake-up, single/double/triple-tap recognition, activity/inactivity, and 6D/4D orientation.

The LIS2DUX12 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 $+85^{\circ}\text{C}$.

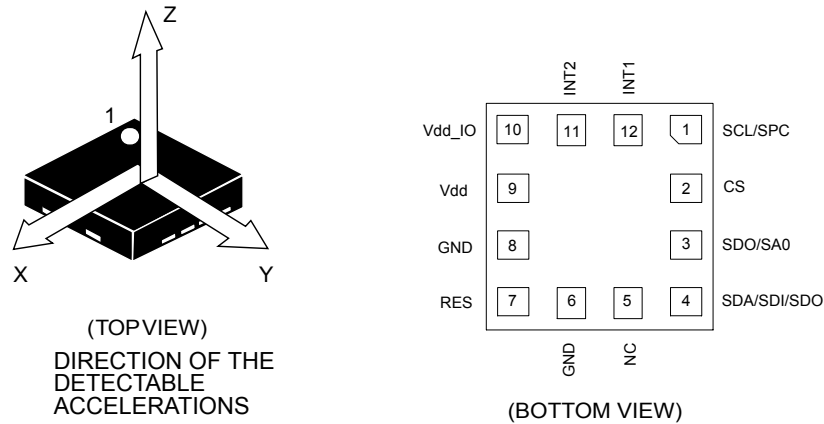
1 Block diagram and pin description

1.1 Block diagram

Figure 1. Block diagram



1.2 Pin description

Figure 2. Pin connections

Table 1. Pin description

| Pin# | Name | Function |
|-------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | SCL SPC | I ² C/MIPI I3C [®] serial clock (SCL) SPI serial port clock (SPC) |
| 2 ⁽¹⁾ | CS | SPI/I ² C/MIPI I3C [®] mode selection (1: SPI idle mode / I ² C/MIPI I3C [®] enabled; 0: SPI enabled / I ² C/MIPI I3C [®] disabled) |
| 3 ⁽²⁾ | SDO SA0 | SPI serial data output (SDO) I ² C less significant bit of the device address (SA0) |
| 4 ⁽²⁾ | SDA SDI SDO | I ² C/MIPI I3C [®] serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO) |
| 5 | NC | Internally not connected. Can be tied to Vdd, Vdd_IO, or GND. |
| 6 | GND | 0 V supply |
| 7 | RES | Connect to GND if not used as interrupt pin 1 ⁽³⁾ |
| 8 | GND | 0 V supply |
| 9 | Vdd | Power supply |
| 10 | Vdd_IO | Power supply for I/O pins |
| 11 ⁽⁴⁾ | INT2 | Interrupt pin 2. Clock input when selected in one-shot mode |
| 12 ⁽⁴⁾ | INT1 | Interrupt pin 1 |

1. The CS pin is internally pulled up by default. The pull-up of the CS pin can be disconnected by setting the bit CS_PU_DIS of register PIN_CTRL (0Ch) to 1.
2. The internal pull-up of the SDO/SA0 and SDA/SDI/SDO pins is disconnected by default. The pull-up of the SDO/SA0 pin can be enabled by setting bit SDO_PU_EN of register PIN_CTRL (0Ch) to 1. The pull-up of the SDA/SDI/SDO pin can be enabled by setting bit SDA_PU_EN of register PIN_CTRL (0Ch) to 1.
3. When the INT1_ON_RES bit of register CTRL1 (10h) is set to 1, the interrupt signals configured on the INT1 pin are routed to the RES pin.
4. The INT1 and INT2 pins are internally pulled down by default. The internal pull-down of the INT1 pin can be disconnected by setting the PD_DIS_INT1 bit in PIN_CTRL (0Ch) to 1. The internal pull-down of the INT2 pin can be disconnected by setting the PD_DIS_INT2 bit in PIN_CTRL (0Ch) to 1.

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

@ Vdd = 1.8 V, T = 25 °C unless otherwise noted.

The product is factory calibrated at 1.8 V. The operational power supply range is from 1.62 V to 3.6 V.

Table 2. Mechanical characteristics

| Symbol | Parameter | Test conditions | Min. | Typ. ⁽¹⁾ | Max. | Unit |
|--------|---------------------------------------------|---------------------------------------|------|---------------------|------|----------|
| FS | Measurement range | | | ±2 | | g |
| | | | | ±4 | | |
| | | | | ±8 | | |
| | | | | ±16 | | |
| So | Sensitivity ⁽²⁾ | @ FS ±2 g | | 0.061 | | mg/digit |
| | | @ FS ±4 g | | 0.122 | | |
| | | @ FS ±8 g | | 0.244 | | |
| | | @ FS ±16 g | | 0.488 | | |
| An | Noise density - high-performance mode | @ FS ±8 g ODR = 800 Hz, BW = ODR/2 | | 220 | | µg/√Hz |
| TyOff | Zero-g level offset accuracy ⁽³⁾ | | | ±30 | | mg |
| TCO | Zero-g offset change vs. temperature | | | ±1 | | mg/°C |
| TCS | Sensitivity change vs. temperature | | | ±0.035 | | %/°C |
| ST | Self-test positive difference | X-axis | 50 | - | 700 | mg |
| | | Y-axis | 50 | - | 700 | |
| | | Z-axis | 200 | - | 1200 | |

1. Typical specifications are not guaranteed.
2. 16-bit format
3. Values after factory calibration test and trimming

2.2 Electrical characteristics

@ Vdd = 1.8 V, T = 25 °C unless otherwise noted.

The product is factory calibrated at 1.8 V. The operational power supply range is from 1.62 V to 3.6 V.

Table 3. Electrical characteristics

| Symbol | Parameter | Test conditions | Min. | Typ. ⁽¹⁾ | Max. | Unit |
|-----------------|----------------------------------------------|-----------------------------------------------------------------------------------------------|----------------|---------------------|------------|------|
| Vdd | Supply voltage | | 1.62 | 1.8 | 3.6 | V |
| Vdd_IO | I/O pins supply voltage ⁽²⁾ | I ² C and SPI interfaces | 1.62 | | 3.6 | V |
| | | MIPI I3C [®] interface | 1.08 | | 3.6 | |
| IddHP | Current consumption in high-performance mode | FS = ±8 g ODR = all ODRs BW = ODR/2 with anti-aliasing filter Temperature sensor off | | 10.8 | | μA |
| IddLP | Current consumption in low-power mode | FS = ±8 g ODR = 50 Hz, BW = ODR/2 with anti-aliasing filter Temperature sensor off | | 6.2 | | μA |
| IddULP | Current consumption in ultralow-power mode | FS = ±8 g ODR = 1.6 Hz, BW = ODR/2 Temperature sensor off | | 2.7 | | μA |
| IddPD | Current consumption in deep power-down | | | 10 | | nA |
| V _{IH} | Digital high-level input voltage | | 0.7*Vdd_IO | | | V |
| V _{IL} | Digital low-level input voltage | | | | 0.3*Vdd_IO | V |
| V _{OH} | Digital high-level output voltage | I _{OH} = 4 mA ⁽³⁾ | Vdd_IO - 0.2 V | | | |
| V _{OL} | Digital low-level output voltage | I _{OL} = 4 mA ⁽³⁾ | | | 0.2 V | |

1. Typical specifications are not guaranteed.
2. It is possible to remove Vdd, maintaining Vdd_IO without blocking the communication busses. In this condition the measurement chain is powered off.
3. 4 mA is the maximum driving capability, that is, the maximum DC current that can be sourced/sunk by the digital pin in order to guarantee the correct digital output voltage levels V_{OH} and V_{OL}.

2.3 Temperature sensor characteristics

@ Vdd = 1.8 V, T = 25 °C unless otherwise noted.

Table 4. Temperature sensor characteristics

| Symbol | Parameter | Min. | Typ. ⁽¹⁾ | Max. | Unit |
|--------|--------------------------------------------------|------|----------------------|------|--------|
| Top | Operating temperature range | -40 | | +85 | °C |
| Toff | Temperature offset ⁽²⁾ | -15 | | +15 | °C |
| TSDr | Temperature sensor output change vs. temperature | | 0.045 ⁽³⁾ | | °C/LSB |
| TODR | Temperature refresh rate | | ODR | | Hz |

1. Typical specifications are not guaranteed.
2. The output of the temperature sensor is 0 LSB (typ.) at 25 °C.
3. 12-bit resolution.

2.4 Communication interface characteristics

2.4.1 SPI - serial peripheral interface

Subject to general operating conditions for V_{DD} and Top.

Table 5. SPI slave timing values

| Symbol | Parameter | Value ⁽¹⁾ | | | Unit | |
|-----------------|-------------------------|----------------------|-----|-----|------|----|
| | | Min | Typ | Max | | |
| $f_{c(SPC)}$ | SPI clock frequency | | | 10 | MHz | |
| $t_{c(SPC)}$ | SPI clock period | | | 100 | ns | |
| $t_{high(SPC)}$ | SPI clock high | 45 | | | | |
| $t_{low(SPC)}$ | SPI clock low | 45 | | | | |
| $t_{su(CS)}$ | CS setup time (mode 3) | 5 | | | | |
| | CS setup time (mode 0) | 20 | | | | |
| $t_{h(CS)}$ | CS hold time (mode 3) | 20 | | | | |
| | CS hold time (mode 0) | 20 | | | | |
| $t_{su(SI)}$ | SDI input setup time | 5 | | | | |
| $t_{h(SI)}$ | SDI input hold time | 15 | | | | |
| $t_{v(SO)}$ | SDO valid output time | | 15 | 25 | | |
| $t_{dis(SO)}$ | SDO output disable time | | | 50 | | |
| C_{load} | Bus capacitance | | | 100 | | pF |

1. Values are evaluated at 10 MHz clock frequency for SPI with both 4 and 3 wires, based on characterization results, not tested in production

Figure 3. SPI slave timing in mode 0

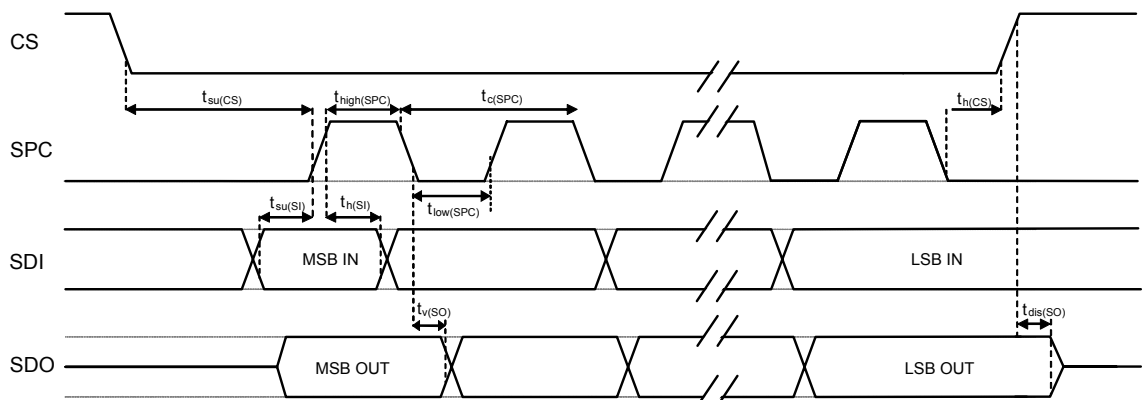
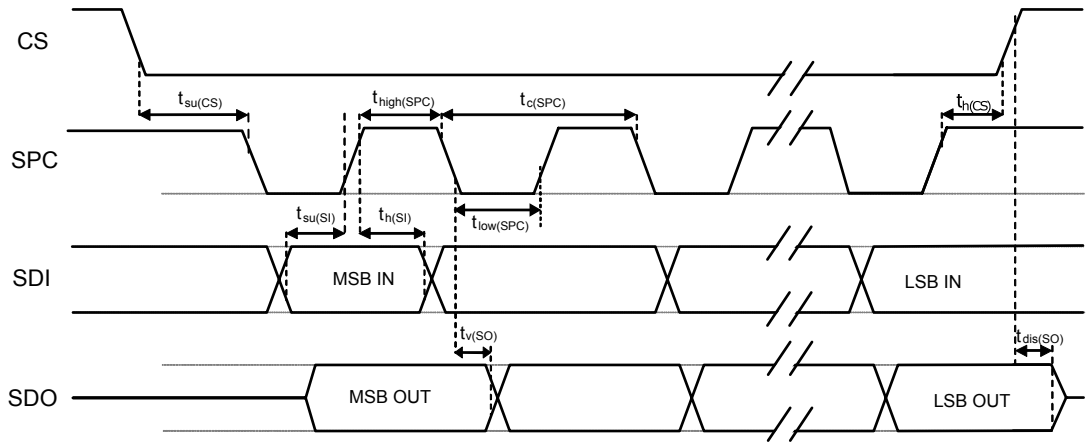


Figure 4. SPI slave timing in mode 3



Note: Measurement points are done at $0.3 \cdot V_{dd_IO}$ and $0.7 \cdot V_{dd_IO}$ for both input and output ports.

2.4.2 I²C - inter-IC control interface

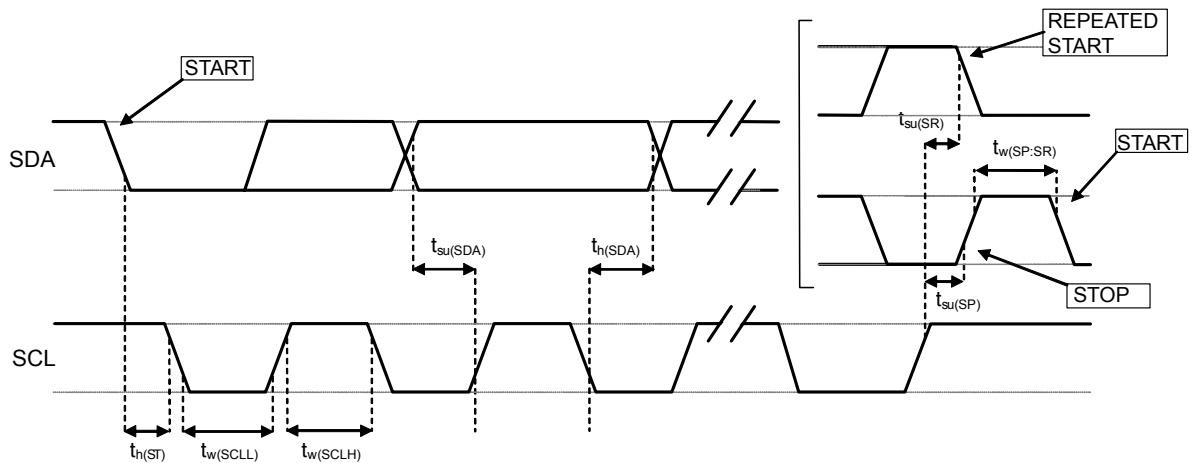
 Subject to general operating conditions for V_{dd} and Top.

Table 6. I²C slave timing values

| Symbol | Parameter | I ² C fast mode ⁽¹⁾⁽²⁾ | | I ² C fast mode plus ⁽¹⁾⁽²⁾ | | Unit |
|-----------------------|------------------------------------------------|----------------------------------------------|-----|---------------------------------------------------|------|------|
| | | Min | Max | Min | Max | |
| f _(SCL) | SCL clock frequency | 0 | 400 | 0 | 1000 | kHz |
| t _{w(SCLL)} | SCL clock low time | 1.3 | | 0.5 | | μs |
| t _{w(SCLH)} | SCL clock high time | 0.6 | | 0.26 | | |
| t _{su(SDA)} | SDA setup time | 100 | | 50 | | ns |
| t _{h(SDA)} | SDA data hold time | 0 | 0.9 | 0 | | μs |
| t _{h(ST)} | START/REPEATED START condition hold time | 0.6 | | 0.26 | | |
| t _{su(SR)} | REPEATED START condition setup time | 0.6 | | 0.26 | | |
| t _{su(SP)} | STOP condition setup time | 0.6 | | 0.26 | | |
| t _{w(SP:SR)} | Bus free time between STOP and START condition | 1.3 | | 0.5 | | |
| | Data valid time | | 0.9 | | 0.45 | |
| | Data valid acknowledge time | | 0.9 | | 0.45 | |
| C _B | Capacitive load for each bus line | | 400 | | 550 | |

 1. Data based on standard I²C protocol requirement, not tested in production.

 2. Data for I²C fast mode and I²C fast mode plus have been validated by characterization, not tested in production.

Figure 5. I²C slave timing diagram

 Note: Measurement points are done at 0.3·V_{dd_IO} and 0.7·V_{dd_IO} for both ports.

2.5 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 7. Absolute maximum ratings

| Symbol | Ratings | Maximum value | Unit |
|--------------------|-------------------------------------------------------------------------|---------------------------------|------|
| V _{dd} | Supply voltage | -0.3 to 4.3 | V |
| V _{dd_IO} | I/O pins supply voltage | -0.3 to 4.3 | V |
| V _{in} | Input voltage on any control pin (CS, SCL/SPC, SDA/SDI/SDO, SDO/SA0) | -0.3 to V _{dd_IO} +0.3 | V |
| A _{POW} | Acceleration (any axis, powered, V _{dd} = 1.8 V) | 3000 g for 0.5 ms | g |
| | | 10000 g for 0.2 ms | g |
| A _{UNP} | Acceleration (any axis, unpowered) | 3000 g for 0.5 ms | g |
| | | 10000 g for 0.2 ms | g |
| T _{OP} | Operating temperature range | -40 to +85 | °C |
| T _{STG} | Storage temperature range | -40 to +125 | °C |
| ESD | Electrostatic discharge protection | 2 (HBM) | kV |

Note: Supply voltage on any pin should never exceed 4.3 V.



This device is sensitive to mechanical shock, improper handling can cause permanent damage to the part.



This device is sensitive to electrostatic discharge (ESD), improper handling can cause permanent damage to the part.

3 Terminology and functionality

3.1 Terminology

3.1.1 Sensitivity

Sensitivity describes the gain of the sensor and can be determined by applying 1 g acceleration to it. As the sensor can measure DC accelerations this can be done easily by pointing the axis of interest towards the center of the Earth, noting the output value, rotating the sensor by 180 degrees (pointing to the sky) and noting the output value again. By doing so, ± 1 g acceleration is applied to the sensor. Subtracting the larger output value from the smaller one, and dividing the result by 2, leads to the actual sensitivity of the sensor. This value changes very little over temperature and time. The sensitivity tolerance describes the range of sensitivities of a large population of sensors.

3.1.2 Zero-g level offset

Zero-g level offset describes the deviation of an actual output signal from the ideal output signal if no acceleration is present. A sensor in a steady state on a horizontal surface measures 0 g on the X-axis and 0 g on the Y-axis whereas the Z-axis measures 1 g. The output is ideally in the middle of the dynamic range of the sensor (content of OUT registers 00h, data expressed as two's complement number). A deviation from ideal value in this case is called zero-g level offset. Offset is to some extent a result of stress to the MEMS sensor and therefore the offset can slightly change after mounting the sensor onto a printed circuit board or exposing it to extensive mechanical stress. Offset changes little over temperature, see "Zero-g level offset change vs. temperature".

3.2 Functionality

3.2.1 Operating modes

The LIS2DUX12 has 4 operating modes: high-performance, low-power, ultralow-power, and one-shot mode. In low-power mode, an aggressive anti-aliasing filter is active, but the overall current consumption remains extraordinarily low (refer to [Table 3](#)).

3.2.2 One-shot modes

The device features two one-shot modes: one-shot triggered by the INT2 pin, and one-shot triggered by the I²C/SPI/I³C digital interface.

3.2.3 Self-test

Self-test mode allows checking the sensor functionality without moving it, applying an actuation force to the sensor and simulating a definite input acceleration.

3.2.4 Activity/inactivity, Android stationary/motion detection functions

The activity/inactivity function recognizes the device's sleep state and allows reducing system power consumption.

When the activity/inactivity function is activated by setting the SLEEP_ON bit in [WAKE_UP_THS \(1Ch\)](#), the device automatically goes to the inactivity output data rate selected by the INACT_ODR[1:0] bits in register [CTRL4 \(13h\)](#).

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

The Android stationary/motion detection function only recognizes the device's sleep state.

When the Android stationary/motion detection function is activated by setting to a stationary condition the INACT_ODR[1:0] bits in register [CTRL4 \(13h\)](#), the device detects acceleration below a fixed threshold but does not change the ODR after sleep state detection.

The activity/inactivity recognition and Android stationary/motion detection functions are activated by writing the desired threshold in the [WAKE_UP_THS \(1Ch\)](#) register. The high-pass filter is automatically enabled.

If the device is in sleep (inactivity/stationary) mode, when at least one of the axes exceeds the threshold in [WAKE_UP_THS \(1Ch\)](#), the device goes into a sleep-to-wake state (as wake-up).

For the activity/inactivity function, the device, in a wake-up state, returns to the ODR before sleep state detection.

Activity/inactivity, Android stationary/motion detection threshold and duration can be configured in the following control registers:

[WAKE_UP_THS \(1Ch\)](#)

[WAKE_UP_DUR \(1Dh\)](#)

3.2.5 Interrupt event recognition

The device may be configured to generate interrupt signals coming from an independent inertial wake-up/free-fall event or from the position of the device itself. Thresholds and timing of this interrupt generator are programmable by the end user in runtime.

Automatic programmable sleep-to-wake-up and return-to-sleep functions are also available for enhanced power saving.

The device interrupts signal can behave as:

- Free-fall: 3-axis underthreshold recognition;
- Wake-up: axis recognition;
- Wake-to-sleep: change of state recognition active-sleep (also known as activity-inactivity);
- 6D and 4D orientation detection: change of position recognition;
- Tap-tap: single, double, triple axis and sign recognition.

All these functions are parallel but during sleep it is not possible to recognize a tap-tap event. All these signals can be driven to the two interrupt pins (INT1 and INT2) through registers [MD1_CFG \(1Fh\)](#) and [MD2_CFG \(20h\)](#).

All these functions are enabled by setting the INTERRUPTS_ENABLE bit in register [INTERRUPT_CFG \(17h\)](#) to 1.

It is possible to configure the duration of the interrupt using the LIR bit in [INTERRUPT_CFG \(17h\)](#) as shown in the following table.

Table 8. Configuration of duration of interrupt

| LIR | Interrupt type |
|-----|----------------|
| 0 | Level mode |
| 1 | Latched mode |

- Interrupt level mode: the interrupt signal goes high when an interrupt event occurs and is reset when the acceleration data fall below the threshold.
- Interrupt latched mode: the interrupt signal on the INT1/INT2 pins is the OR of the interrupt flags enabled through the [MD1_CFG \(1Fh\)](#) and [MD2_CFG \(20h\)](#) registers. Each interrupt flag goes to 1 when an interrupt event occurs and is reset when the dedicated source register is read. The interrupt generator block is inhibited 1 ODR after the reset event. It is possible to reset all the interrupt flags simultaneously by reading the [ALL_INT_SRC \(24h\)](#) register.

3.3 Sensing element

A proprietary process is used to create a surface micromachined accelerometer. The technology allows processing suspended silicon structures which are attached to the substrate in a few points called anchors and are free to move in the direction of the sensed acceleration. In order to be compatible with the traditional packaging techniques, a cap is placed on top of the sensing element to avoid blocking the moving parts during the molding phase of the plastic encapsulation. When an acceleration is applied to the sensor the proof mass displaces from its nominal position, causing an imbalance in the capacitive half-bridge. This imbalance is measured using charge integration in response to a voltage pulse applied to the capacitor.

At steady-state the nominal value of the capacitors are a few pF and when an acceleration is applied, the maximum variation of the capacitive load is in the fF range.

3.4 IC interface

The complete measurement chain is composed of a low-noise capacitive amplifier which converts the capacitive unbalancing of the MEMS sensor into an analog voltage using an analog-to-digital converter.

The acceleration data may be accessed through an I²C/SPI/MIPI I3C[®] interface thus making the device particularly suitable for direct interfacing with a microcontroller.

The LIS2DUX12 features a data-ready signal which indicates when a new set of measured acceleration data is available, thus simplifying data synchronization in the digital system that uses the device.

3.5 Factory calibration

The IC interface is factory-calibrated for sensitivity (S_0) and zero- g level offset.

The trim values are stored inside the device in nonvolatile memory. Any time the device is turned on, the trimming parameters are downloaded into the registers to be used during active operation. This allows using the device without further calibration. If an accidental write occurs in the registers where the trimming parameters are stored, the BOOT bit in CTRL4 (13h) can help to retrieve the correct trimming parameters from nonvolatile memory without the need to switch on/off the device. This bit is automatically reset at the end of the download operation. Setting this bit has no impact on the control registers.

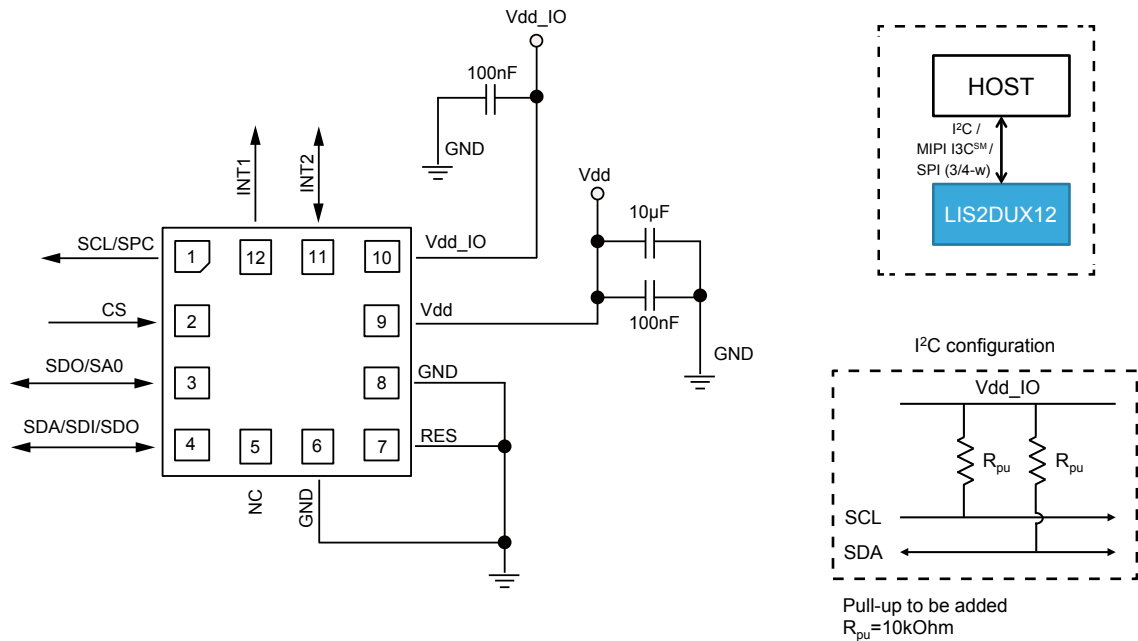
3.6 Temperature sensor

The temperature is available in OUT_T_L (2Eh), OUT_T_H (2Fh) stored as two's complement data, left-justified in 12-bit mode.

Refer to [Table 4. Temperature sensor characteristics](#) for the conversion factor.

4 Application hints

Figure 6. LIS2DUX12 electrical connections (top view)



The device core is supplied through the Vdd line while the I/O pins are supplied through the Vdd_IO line. Power supply decoupling capacitors (100 nF ceramic, 10 µF aluminum) should be placed as near as possible to pin 9 of the device (common design practice).

All the voltage and ground supplies must be present at the same time to have proper behavior of the IC (refer to Figure 6). It is possible to remove Vdd while maintaining Vdd_IO without blocking the communication bus, in this condition the measurement chain is powered off.

The functionality of the device and the measured acceleration data are selectable and accessible through the I²C/MIPI I3C[®] or SPI interfaces. When using the I²C, CS must be tied high (that is, connected to Vdd_IO).

The functions, the threshold and the timing of the two interrupt pins (INT1 and INT2) can be completely programmed by the user through the I²C/MIPI I3C[®]/SPI interface.

Table 9. Internal pin status

| Pin # | Name | Function | Pin status |
|-------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1 | SCL SPC | I ² C/MIPI I3C [®] serial clock (SCL) SPI serial port clock (SPC) | Default: input without pull-up |
| 2 | CS | SPI/I ² C/MIPI I3C [®] mode selection 1: SPI idle mode / I ² C/MIPI I3C [®] enabled 0: SPI enabled / I ² C/MIPI I3C [®] disabled | Default: input with internal pull-up |
| 3 | SDO SA0 | Serial data output (SDO) I ² C less significant bit of the device address (SA0) | Default: input without internal pull-up |
| 4 | SDA SDI SDO | I ² C/MIPI I3C [®] 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, Vdd_IO, or GND. | |
| 6 | GND | 0 V supply | |
| 7 | RES | Connect to GND if not used as interrupt pin 1 ⁽¹⁾ | |
| 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 one-shot mode. | Default: input with internal pull-down ⁽²⁾⁽³⁾ |
| 12 | INT1 | Interrupt pin 1 | Default: input with internal pull-down ⁽⁴⁾⁽⁵⁾ |

1. When the `INT1_ON_RES` bit of register `CTRL1` (10h) is set to 1, the interrupt signals configured on the `INT1` pin are routed to the `RES` pin.
2. The internal pull-down can be disconnected by setting the `PD_DIS_INT2` bit of register `PIN_CTRL` (0Ch) to 1. When this is done, the `INT2` pin needs to be externally biased.
3. The `INT2` pin is configured as "push-pull output forced to GND" (and the internal pull-down disabled) as soon as it is configured with the interrupt configuration bits in the `CTRL3` (12h), `MD2_CFG` (20h), `EMB_FUNC_INT2` (0Eh), `FSM_INT2` (0Fh), and `MLC_INT2` (11h) registers.
4. The internal pull-down can be disconnected by setting the `PD_DIS_INT1` bit of register `PIN_CTRL` (0Ch) to 1. When this is done, the `INT1` pin needs to be externally biased.
5. The `INT1` pin is configured as "push-pull output forced to GND" (and the internal pull-down disabled) as soon as it is configured with the interrupt configuration bits in the `CTRL2` (11h), `MD1_CFG` (1Fh), `EMB_FUNC_INT1` (0Ah), `FSM_INT1` (0Bh), and `MLC_INT1` (0Dh) registers.

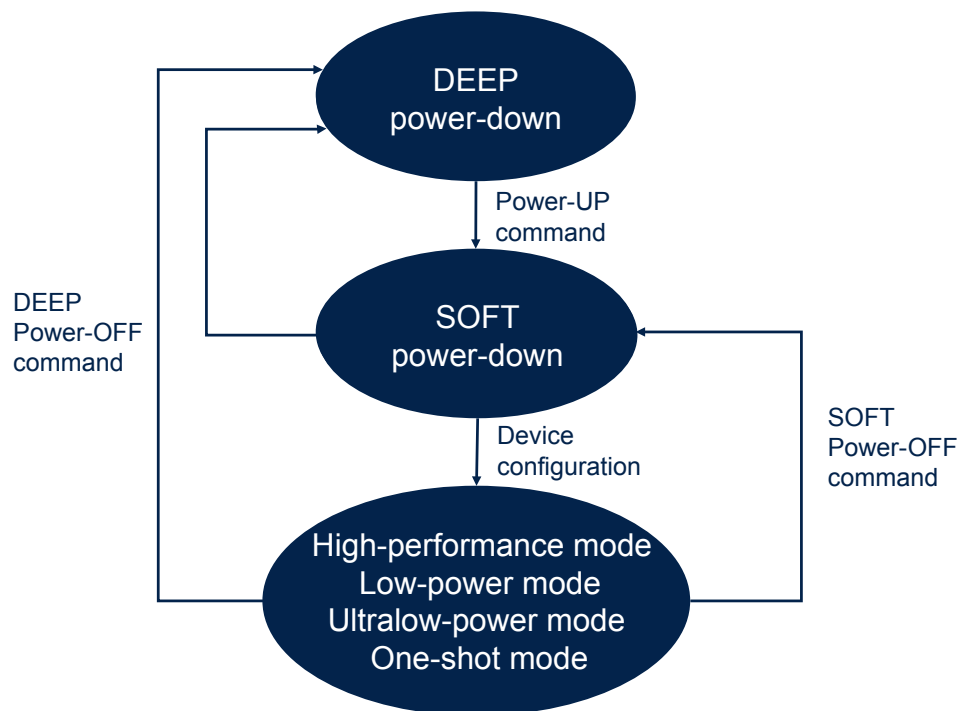
4.1 Power-up sequence

When Vdd and Vdd_IO are set, the device enters a deep power-down state: this is an ultralow current condition in which the device is powered but with a very aggressive trade-off in current consumption.

In order to wake up the device, the first step is to perform a power-up command bringing the device into a soft power-down state: the device is now ready to be configured and generate data.

The LIS2DUX12 offers 4 different operating modes able to generate accelerometer and temperature data: high-performance mode, low-power mode, ultralow-power mode, and one-shot mode.

Figure 7. Power-up sequence



4.1.1 Power-up command

The power-up command allows the LIS2DUX12 to transition from deep power-down to soft power-down. It differs if either the I²C/I³C interface or the SPI interface is used. See [Section 6](#) for more details on the digital interfaces.

4.1.1.1 I²C/I³C interfaces

If the I²C or I³C interfaces are used, the following sequence should be provided to the device:

- ST/SR+ STATIC ADDRESS+R/W (both R and W sequences are supported)

The device generates a NACK and starts power-up. The operation takes 25 ms (maximum) and once completed LIS2DUX12 is in the SOFT_PD state. It is possible to verify the correct transition in the soft power-down state providing again the power-up command (ST/SR+ STATIC ADDRESS+R/W) and checking the ACK generation from the device.

To guarantee the current execution of the power-up command, the I²C/I³C master should operate at open-drain speed using I²C fast mode plus reference timing.

In the LIS2DUX12 if the bus is at 1.2 V and the device is in deep power-down, the master should use I²C fast mode timing instead of I²C fast mode plus to perform the power-up command. Once the device is awake, the soft power-down rules apply in the same way as the other voltages.

If the I³C interface is used, a dynamic address should be assigned before starting device configuration. Once the DAA procedure is performed, the dynamic address is stored inside the device and it is maintained if the device returns to the deep power-down state. In this event, no other DAA procedures are needed and the power-up commands can be executed directly using the dynamic address.

4.1.1.2 SPI interface

If the SPI interface is used, the LIS2DUX12 can move from deep power-down to soft power-down by writing the bit SOFT_PD of IF_WAKE_UP (3Eh) to 1. The device starts the power-up and this operation takes 25 ms (maximum). In order to verify that device has correctly completed the transition to soft power-down, the who_am_I value (expected to be equal to 47h) can be checked by reading register WHO_AM_I (0Fh).

Note: When the power-up command is performed using the SPI interface, the I²C and the I³C interfaces are automatically disabled as soon as the device exits deep power-down condition. They are automatically re-enabled after a deep power-off command is executed.

4.1.2 Soft power-off command

When the LIS2DUX12 is configured in one of the four operating modes, it is possible to switch to the soft power-down condition, writing the ODR[3:0] bits in register CTRL5 (14h) to 0000.

4.1.3 Deep power-off command

When device is in soft power-down or it is configured in one of the four operating modes, it is possible to switch to the deep power-down condition, writing the bit DEEP_PD in register SLEEP (3Dh) to 1. When a transition to deep power-down occurs, all the registers are reset to their default value and a new power-up command and device configuration need to be done.

5 Digital main blocks and embedded low-power features

The LIS2DUX12 has been designed to be fully compliant with Android, featuring the following on-chip functions:

- FIFO data buffering
 - 100% efficiency with flexible configurations and partitioning
 - Possibility to store timestamp
- Event-detection interrupts (fully configurable)
 - Free-fall
 - Wake-up
 - 6D/4D orientation
 - Single/double/triple-tap detection
 - Activity/inactivity recognition
 - Stationary/motion detection
- Specific IP blocks (called "embedded functions") with negligible power consumption and high-performance
 - Pedometer functions: step detector and step counters
 - Tilt
 - Significant motion detection
 - Finite state machine (FSM)
 - Machine learning core (MLC) with exportable features and filters for AI applications

5.1 FIFO

The LIS2DUX12 embeds 128 slots of 7 bytes each (1 byte TAG + 6 bytes DATA). This allows consistent power saving for the system, since the host processor does not need to continuously poll data from the sensor, but it can wake up only when needed and burst the significant data out from the FIFO.

FIFO is designed in order to allow the batching of different kinds of sensors. It is possible to store in FIFO the data of the accelerometer and temperature physical sensors along with the data of virtual sensors like the step counter, the MLC features / filters / results and FSM results.

The reconstruction of a FIFO stream is a simple task thanks to the FIFO_DATA_OUT_TAG byte that allows recognizing the meaning of a word in FIFO.

The applications have maximum flexibility in choosing the rate of batching for physical sensors with FIFO dedicated configurations.

FIFO allows correctly reconstructing the timestamp information for each sensor stored in FIFO. Also, if a change in the ODR or BDR (batch data rate) configuration is performed, the application can correctly reconstruct the timestamp and know exactly when the change was applied in a FIFO stream without disabling FIFO batching. FIFO stores information of the new configuration and timestamp in which the change was applied in the device.

In order to maximize the amount of data collected in FIFO, it is possible to double the slots of FIFO data (from 128 to 256) by writing the FIFO_DEPTH bit to 1 in FIFO_CTRL (15h) with 2x depth mode. When this mode is enabled, the most significant 8 bits for each acceleration data are stored in FIFO. Each FIFO data word contains data of two consecutive ODRs, the actual and the previous one.

In high-resolution batch mode, accelerometer and temperature data are stored in FIFO in 12-bit format at the ODR rate.

In 2x depth batch mode, each FIFO word contains two accelerometer data in 8-bit format at the ODR/2 rate.

It is possible to avoid storing the temperature data in FIFO by setting the XL_ONLY_FIFO bit in FIFO_WTM (16h) register to 1. In this case, the accelerometer data are stored in FIFO in 16-bit format at the ODR rate.

The FIFO buffer can work according to six different modes:

- Bypass mode
- FIFO mode
- Continuous mode
- Continuous-to-FIFO mode
- Bypass-to-continuous
- Bypass-to-FIFO

Each mode is selected by the FIFO_MODE[2:0] bits in the FIFO_CTRL (15h) register. A programmable FIFO watermark is selected in the FIFO_WTM (16h) register.

FIFO status is available in the FIFO_STATUS1 (26h) and FIFO_STATUS2 (27h) registers and can be used to generate dedicated interrupts on the INT1 and INT2 pins using the CTRL2 (11h) and CTRL3 (12h) registers.

The FIFO_WTM_IA bit in the FIFO_STATUS1 (26h) register goes to 1 when the number of unread samples is greater than or equal to FTH[6:0] in FIFO_WTM (16h). If FTH[6:0] is equal to 0, the FIFO_WTM_IA bit in FIFO_STATUS1 (26h) goes to 0.

The FIFO_OVR_IA bit in FIFO_STATUS1 (26h) is equal to 1 if a FIFO sample is overwritten.

FSS[7:0] in FIFO_STATUS2 (27h) contains stored data levels of unread samples.

When FSS[7:0] is equal to 00000000, FIFO is empty. When FSS[7:0] is equal to 10000000, FIFO is full and the unread samples are 128.

5.1.1 Bypass mode

In bypass mode FIFO_CTRL (15h) (FIFO_MODE[2:0] = 000), the FIFO is not operational, no data is collected in FIFO memory, and it remains empty with only the actual sample available in the output registers.

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

5.1.2 FIFO mode

In FIFO mode FIFO_CTRL (15h) (FIFO_MODE[2:0] = 001) data from the output channels are stored in the FIFO memory until it is full. When 128 unread samples are stored in memory, data collecting is stopped.

To reset FIFO content, bypass mode should be selected by writing FIFO_CTRL (15h) (FIFO_MODE[2:0]) to 000. After this reset command, it is possible to restart FIFO mode, writing FIFO_CTRL (15h) (FIFO_MODE[2:0]) to 001.

5.1.3 Continuous mode

Continuous mode FIFO_CTRL (15h) (FIFO_MODE[2:0] = 110) provides a continuous FIFO update: when 128 unread samples are stored in memory, as new data arrives, the oldest data is discarded and overwritten by the newer.

A FIFO threshold flag FIFO_WTM_IA bit in FIFO_STATUS1 (26h) is asserted when the number of unread samples in FIFO is greater than or equal to FTH[6:0] in FIFO_WTM (16h).

It is possible to route the FIFO_WTM_IA bit to the INT1 pin by writing the INT1_FIFO_TH bit to 1 in register CTRL2 (11h) or to the INT2 pin by writing the INT2_FIFO_TH bit to 1 in register CTRL3 (12h).

If an overrun occurs, the oldest sample in FIFO is overwritten and the FIFO_OVR_IA flag in FIFO_STATUS1 (26h) is asserted.

5.1.4 Continuous-to-FIFO mode

In continuous-to-FIFO mode FIFO_MODE[2:0] = 011 in the FIFO_CTRL (15h) register, FIFO operates in continuous mode and FIFO mode starts upon an edge trigger event. When the FIFO is full, data collecting is stopped. The trigger event could be single/double/triple-tap, wake-up, free-fall, 6D interrupt or any combination of these events, but every interrupt has to be routed to the corresponding pin to be used as a trigger.

5.1.5 Bypass-to-continuous mode

In bypass-to-continuous mode [FIFO_CTRL \(15h\)](#) ([FIFO_MODE\[2:0\]](#) = 100), data measurement storage inside FIFO starts in continuous mode upon an edge trigger event.

The trigger event could be single/double/triple-tap, wake-up, free-fall, 6D interrupt or any combination of these events, but every interrupt has to be routed to the corresponding pin to be used as a trigger. The sample that generated the trigger is available in FIFO.

5.1.6 Bypass-to-FIFO

In bypass-to-FIFO mode [FIFO_CTRL \(15h\)](#)([FIFO_MODE_\[2:0\]](#) = 111), data measurement storage inside FIFO operates in FIFO mode when selected triggers are equal to 1, otherwise FIFO content is reset (bypass mode).

The trigger event could be single/double/triple-tap, wake-up, free-fall, 6D interrupt or any combination of these events, but every interrupt has to be routed to the corresponding pin to be used as a trigger.

The sample that generated the trigger is available in FIFO.

5.1.7 FIFO reading procedure

When FIFO is enabled and the mode is different from bypass, reading the FIFO output registers return the oldest FIFO sample set. Whenever these registers are read, their content is moved to the SPI/I²C/MIPI I3C[®] output buffer.

The data stored in FIFO are accessible from dedicated registers and each FIFO word is composed of 7 bytes: one tag byte ([FIFO_DATA_OUT_TAG \(40h\)](#)), in order to identify the sensor, and 6 bytes of fixed data ([FIFO_DATA_OUT](#) registers from (41h) to (46h)).

The [FSS\[7:0\]](#) field in the [FIFO_STATUS2 \(27h\)](#) registers contains the number of words (1 byte TAG + 6 bytes DATA) collected in FIFO.

Meta information about accelerometer configuration changes can be managed by enabling the [CFG_CHG_EN](#) bit in [FIFO_CTRL \(15h\)](#).

The entire FIFO content is retrieved by performing a certain number of read operations from the FIFO output registers until the buffer becomes empty ([FSS\[7:0\]](#) bits of the [FIFO_STATUS2 \(27h\)](#) register are equal to 0).

FIFO output data must be read with multiples of 7-byte reads starting from the [FIFO_DATA_OUT_TAG \(40h\)](#) register.

The rounding function (automatic wraparound) from address [FIFO_DATA_OUT_Z_L \(45h\)](#) and [FIFO_DATA_OUT_Z_H \(46h\)](#) to [FIFO_DATA_OUT_TAG \(40h\)](#) is done automatically in the device, in order to allow reading many words with a unique multiple read operation.

The recommended way to retrieve data from the FIFO is the following:

1. Read the [FIFO_STATUS2 \(27h\)](#) register to check how many words are stored in the FIFO. This information is contained in the [FSS\[7:0\]](#) bits.
2. For each word in FIFO, read the FIFO word (tag and output data) and interpret it on the basis of the FIFO tag.
3. Go to step 1.

5.1.8 FIFO empty condition

When FIFO is emptied, a dedicated FIFO tag value (equal to 00000) is used in order to recognize an empty condition and no duplicated samples are read.

5.2 Pedometer functions: step detector and step counters

The LIS2DUX12 embeds an advanced pedometer with an algorithm running in an ultralow-power domain in order to ensure extensive battery life in battery-constrained applications.

Leveraging on enhanced configurability, the advanced embedded pedometer is suitable for a large range of applications from mobile to wearable devices.

The algorithm processes and analyzes the accelerometer waveform in order to count the user's steps during walking and running activities.

The pedometer works at 25 Hz and it is not affected by the selected device power mode (ultralow-power, low-power, high-performance), thus guaranteeing an ultralow-power experience and extreme flexibility in conjunction with other device functionalities.

The accelerometer operating mode can be changed at runtime and is based on user requirements without impacting the performance of the pedometer.

The pedometer output can be batched in the device's FIFO buffer, in order to decrease overall system current consumption.

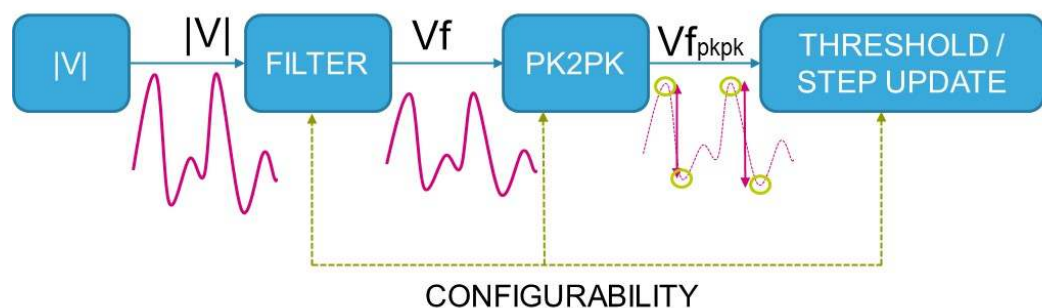
ST freely provides the support and the tools for easily configuring the device and tuning the algorithm configuration for a best-in-class user experience.

5.3 Pedometer algorithm

The pedometer algorithm is composed of a cascade of four stages:

1. Computation of the acceleration magnitude signal in order to detect the signal independently from device orientation;
2. FIR filter to extract relevant frequency components and to smooth the signal by cutting off high frequencies;
3. Peak detector to find the maximum and minimum of the waveform and compute the peak-to-peak value;
4. Step count: if the peak-to-peak value is greater than the settled threshold, a step is counted.

Figure 8. Four-stage pedometer algorithm



The LIS2DUX12 embeds a dynamic internal threshold for step detection that is updated after each peak-to-peak evaluation: the internal threshold is increased with a configurable speed if a step is detected or decreased with a configurable speed if a step is not detected.

This approach ensures high accuracy when the user starts to walk and a false peak rejection when the user is walking or running.

An internal configurable debounce algorithm can be also set to filter false walks: indeed, an accelerometer pattern is recognized as a walk or run only if a minimum number of steps are counted.

The LIS2DUX12 has been designed to reject a false-positive signal inside the algorithm core.

On top of the mechanisms detailed above, the LIS2DUX12 allows enabling and configuring a dedicated false-positive rejection block to further boost pedometer accuracy.

5.4 Tilt detection

The tilt function helps to detect activity change and has been implemented in hardware using only the accelerometer to achieve targets of both ultralow power consumption and robustness during the short duration of dynamic accelerations.

The tilt function is based on a trigger of an event each time the device's tilt changes and can be used with different scenarios, for example:

- Triggers when the phone is in a front pants pocket and the user goes from sitting to standing or standing to sitting;
- Does not trigger when the phone is in a front pants pocket and the user is walking, running, or going upstairs.

5.5 Significant motion detection

The significant motion detection (SMD) function generates an interrupt when a 'significant motion', that could be due to a change in user location, is detected. In the LIS2DUX12 device this function has been implemented in hardware using only the accelerometer.

SMD functionality can be used in location-based applications in order to receive a notification indicating when the user is changing location.

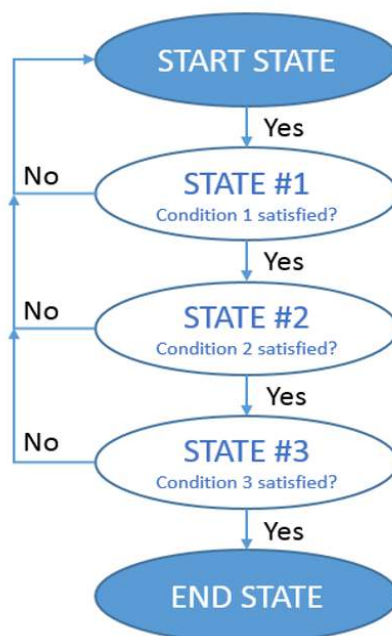
5.6 Finite state machine

The LIS2DUX12 can be configured to generate interrupt signals activated by user-defined motion patterns. To do this, up to 8 embedded finite state machines can be programmed independently for motion detection such as glance gestures, absolute wrist tilt, shake and double-shake detection.

Definition of finite state machine

A state machine is a mathematical abstraction used to design logic connections. It is a behavioral model composed of a finite number of states and transitions between states, similar to a flow chart in which one can inspect the way logic runs when certain conditions are met. The state machine begins with a start state, goes to different states through transitions dependent on the inputs, and can finally end in a specific state (called stop state). The current state is determined by the past states of the system. The following figure shows a generic state machine.

Figure 9. Generic state machine



Finite state machine in the LIS2DUX12

LIS2DUX12 accelerometer data can be used as input of up to 8 programs in the embedded finite state machine (FSM). The embedded temperature sensor data can also be processed by FSM logic (Figure 10. State machine in the LIS2DUX12).

All 8 finite state machines are independent: each one has its dedicated memory area and it is independently executed. An interrupt is generated when the end state is reached or when some specific command is performed.

Figure 10. State machine in the LIS2DUX12



5.7 Machine learning core

The LIS2DUX12 embeds a dedicated core for machine learning processing that provides system flexibility, allowing some algorithms run in the application processor to be moved to the MEMS sensor with the advantage of consistent reduction in power consumption.

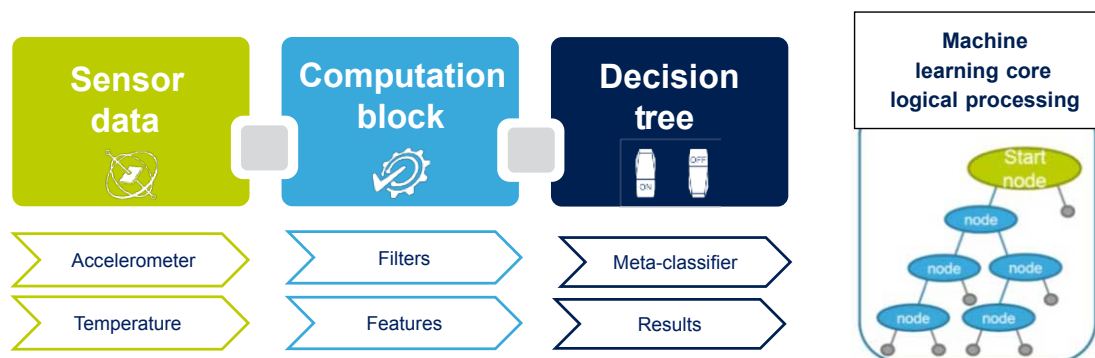
Machine learning core logic allows identifying if a data pattern matches a user-defined set of classes. Typical examples of applications could be activity detection like running, walking, driving, and so on.

The LIS2DUX12 machine learning core works on data patterns coming from the accelerometer sensor, but it is also possible to process the embedded temperature sensor data.

The input data can be filtered using a dedicated configurable computation block containing filters and features computed in a fixed time window defined by the user. Computed feature values and filtered data values can also be read through the FIFO buffer.

Machine learning processing is based on logical processing composed of a series of configurable nodes characterized by "if-then-else" conditions where the "feature" values are evaluated against defined thresholds.

Figure 11. Machine learning core in the LIS2DUX12



The LIS2DUX12 can be configured to run up to 4 decision trees simultaneously and independently and every decision tree can generate up to 16 results. The total number of nodes can be up to 128.

The results of the machine learning processing are available in dedicated output registers readable from the application processor at any time.

The LIS2DUX12 machine learning core can be configured to generate an interrupt when a change in the result occurs.

5.8 Adaptive self-configuration (ASC)

The LIS2DUX12 supports the adaptive self-configuration (ASC) feature, which allows the FSM to automatically reconfigure the device in real time based on the detection of a specific motion pattern or based on the output of a specific decision tree configured in the MLC, without any intervention from the host processor. The FSM can write a subset of the device registers using the SETR command, which allows indicating the register address and the new value to be written in such a register. The access to these device registers is mutually exclusive with respect to the host.

6 Digital interfaces

The registers embedded inside the LIS2DUX12 may be accessed through both the I²C, MIPI I3C[®] and SPI serial interfaces. The latter may be software configured to operate either in 3-wire or 4-wire interface mode.

The serial interfaces are mapped to the same pins. To select/exploit the I²C/MIPI I3C[®] interface, the CS line must be tied high (that is, connected to Vdd_IO).

Table 10. Serial interface pin description

| Pin name | Pin description |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS | SPI/I ² C/MIPI I3C [®] mode selection 1: SPI idle mode / I ² C/MIPI I3C [®] enabled 0: SPI enabled / I ² C/MIPI I3C [®] disabled |
| SCL | I ² C/MIPI I3C [®] serial clock (SCL) |
| SPC | SPI serial port clock (SPC) |
| SDA | I ² C/MIPI I3C [®] serial data (SDA) |
| SDI | SPI serial data input (SDI) |
| SDO | 3-wire interface serial data output (SDO) |
| SA0 | I ² C address selection (SA0) |
| SDO | SPI serial data output (SDO) |

6.1 I²C serial interface

The LIS2DUX12 I²C is a bus slave. The I²C is employed to write data into registers whose content can also be read back.

The relevant I²C terminology is given in the table below.

Table 11. I²C terminology

| Term | Description |
|-------------|------------------------------------------------------------------------------------------|
| Transmitter | The device that sends data to the bus |
| Receiver | The device that receives data from the bus |
| Master | The device that initiates a transfer, generates clock signals, and terminates a transfer |
| Slave | The device addressed by the master |

There are two signals associated with the I²C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. Both lines must be connected to Vdd_IO through an external pull-up resistor. When the bus is free, both the lines are high. The I²C interface is compliant with fast mode (400 kHz) and fast mode + (1000 kHz) I²C standards as well as with normal mode.

6.1.1 I²C operation

The transaction on the bus is started through a start (ST) signal. A start condition is defined as a high to low transition on the data line while the SCL line is held high. After this has been transmitted by the master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and the eighth bit tells whether the master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If they match, the device considers itself addressed by the master.

The slave address (SAD) associated to the LIS2DUX12 is 001100xb where the x bit is modified by the SA0/SDO pin in order to modify the device address. If the SA0/SDO pin is connected to the supply voltage, the address is 0011001b, otherwise if the SA0/SDO pin is connected to ground, the address is 0011000b. This solution permits to connect and address two different accelerometers to the same I²C lines.

Data transfer with acknowledge is mandatory. The transmitter must release the SDA line during the acknowledge pulse. The receiver must then pull the data line low so that it remains stable low during the high period of the acknowledge clock pulse. A receiver which has been addressed is obliged to generate an acknowledge after each byte of data received.

The I²C embedded inside the LIS2DUX12 behaves like a slave device and the following protocol must be adhered to. After the start condition (ST) a slave address is sent. Once a slave acknowledge (SAK) has been returned, an 8-bit subaddress (SUB) is transmitted: the 7 LSb represents the actual register address while the CTRL1 (10h) (IF_ADD_INC) bit defines the address increment.

The slave address is completed with a read/write bit. If the bit is 1 (read), a repeated start (SR) condition must be issued after the two subaddress bytes. If the bit is 0 (write) the master transmits to the slave with direction unchanged. Table 12 explains how the SAD+read/write bit pattern is composed, listing all the possible configurations.

Table 12. SAD+read/write patterns

| Command | SAD[6:1] | SAD[0] = SA0 | R/W | SAD+R/W |
|---------|----------|--------------|-----|----------------|
| Read | 001100 | 0 | 1 | 00110001 (31h) |
| Write | 001100 | 0 | 0 | 00110000 (30h) |
| Read | 001100 | 1 | 1 | 00110011 (33h) |
| Write | 001100 | 1 | 0 | 00110010 (32h) |

Table 13. Transfer when master is writing one byte to slave

| | | | | | | | | |
|--------|----|---------|-----|-----|-----|------|-----|----|
| Master | ST | SAD + W | | SUB | | DATA | | SP |
| Slave | | | SAK | | SAK | | SAK | |

Table 14. Transfer when master is writing multiple bytes to slave

| | | | | | | | | | | |
|--------|----|---------|-----|-----|-----|------|-----|------|-----|----|
| Master | ST | SAD + W | | SUB | | DATA | | DATA | | SP |
| Slave | | | SAK | | SAK | | SAK | | SAK | |

Table 15. Transfer when master is receiving (reading) one byte of data from slave

| | | | | | | | | | | | |
|--------|----|---------|-----|-----|-----|----|---------|-----|------|------|----|
| Master | ST | SAD + W | | SUB | | SR | SAD + R | | | NMAK | SP |
| Slave | | | SAK | | SAK | | | SAK | DATA | | |

Table 16. Transfer when master is receiving (reading) multiple bytes of data from slave

| | | | | | | | | | | | | | | | |
|--------|----|-------|-----|-----|-----|----|-------|-----|------|-----|------|-----|------|------|----|
| Master | ST | SAD+W | | SUB | | SR | SAD+R | | | MAK | | MAK | | NMAK | SP |
| Slave | | | SAK | | SAK | | | SAK | DATA | | DATA | | DATA | | |

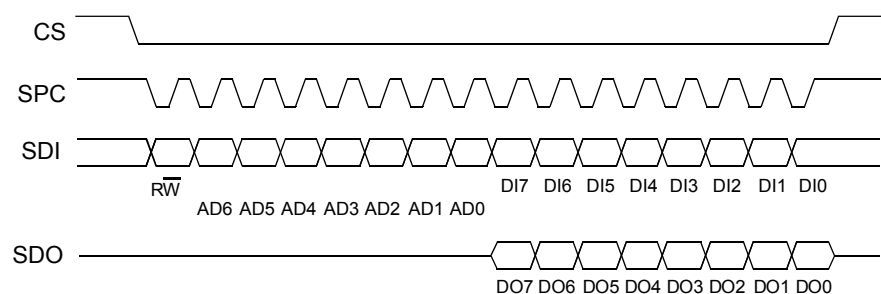
Data are transmitted in byte format (DATA). Each data transfer contains 8 bits. The number of bytes transferred per transfer is unlimited. Data is transferred with the most significant bit (MSb) first. If a slave receiver doesn't acknowledge the slave address (that is, it is not able to receive because it is performing some real-time function) the data line must be left high by the slave. The master can then abort the transfer. A low to high transition on the SDA line while the SCL line is high is defined as a stop condition. Each data transfer must be terminated by the generation of a stop (SP) condition.

In the presented communication format MAK is master acknowledge and NMAK is no master acknowledge.

6.2 SPI bus interface

The LIS2DUX12 SPI is a bus slave. The SPI allows writing to and reading from the registers of the device. The serial interface interacts with the application using 4 wires: **CS**, **SPC**, **SDI** and **SDO**.

Figure 12. Read and write protocol



CS enables the serial port and it is controlled by the SPI master. It goes low at the start of the transmission and goes back high at the end. **SPC** is the serial port clock and it is controlled by the SPI master. It is stopped high when **CS** is high (no transmission). **SDI** and **SDO** are respectively the serial port data input and output. Those lines are driven at the falling edge of **SPC** and should be captured at the rising edge of **SPC**.

Both the read register and write register commands are completed in 16 clock pulses or in multiples of 8 in case of multiple read/write bytes. Bit duration is the time between two falling edges of **SPC**. The first bit (bit 0) starts at the first falling edge of **SPC** after the falling edge of **CS** while the last bit (bit 15, bit 23, ...) starts at the last falling edge of **SPC** just before the rising edge of **CS**.

bit 0: \overline{RW} bit. When 0, the data $DI(7:0)$ is written into the device. When 1, the data $DO(7:0)$ from the device is read. In latter case, the chip drives **SDO** at the start of bit 8.

bit 1-7: address $AD(6:0)$. This is the address field of the indexed register.

bit 8-15: data $DI(7:0)$ (write mode). This is the data that is written into the device (MSb first).

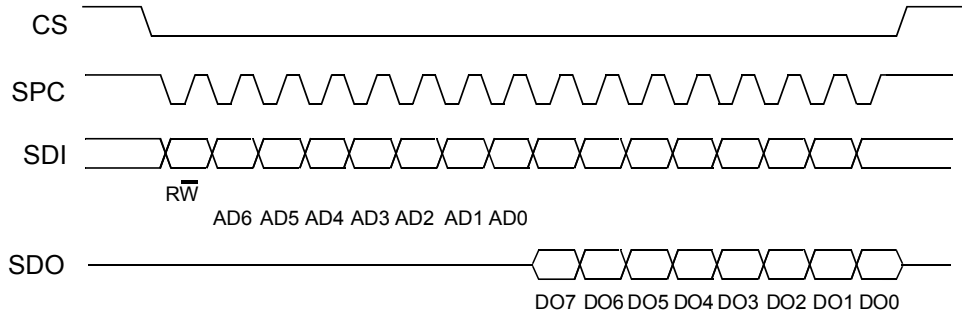
bit 8-15: data $DO(7:0)$ (read mode). This is the data that is read from the device (MSb first).

In multiple read/write commands additional blocks of 8 clock periods are added. When the **CTRL1 (10h)** (**IF_ADD_INC**) bit is 0, the address used to read/write data remains the same for every block. When the **CTRL1 (10h)** (**IF_ADD_INC**) bit is 1, the address used to read/write data is increased at every block.

The function and the behavior of **SDI** and **SDO** remain unchanged.

6.2.1 SPI read

Figure 13. SPI read protocol



The SPI read command is performed with 16 clock pulses. A multiple byte read command is performed by adding blocks of 8 clock pulses to the previous one.

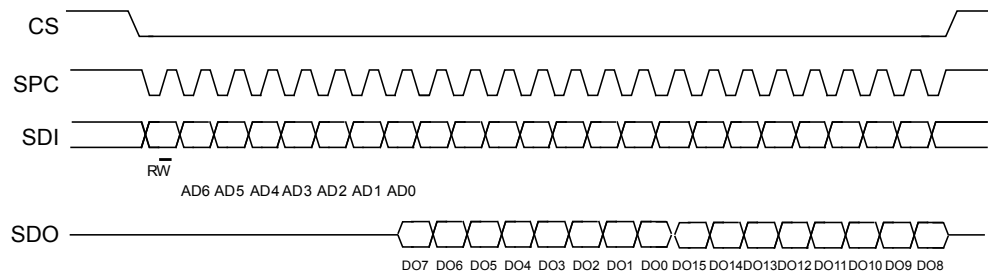
bit 0: READ bit. The value is 1.

bit 1-7: address AD(6:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

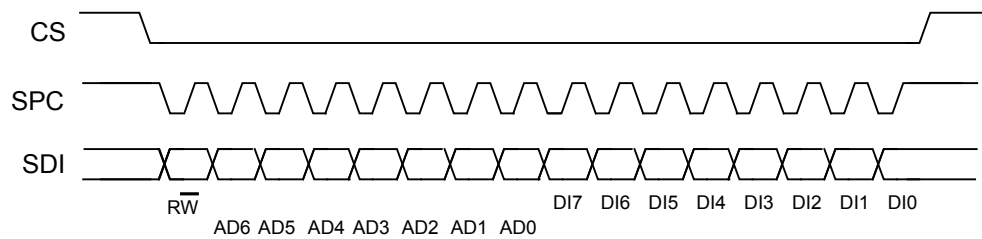
bit 16-... : data DO(...-8). Additional data in multiple byte reads.

Figure 14. Multiple byte SPI read protocol (2-byte example)



6.2.2 SPI write

Figure 15. SPI write protocol



The SPI write command is performed with 16 clock pulses. A multiple byte write command is performed by adding blocks of 8 clock pulses to the previous one.

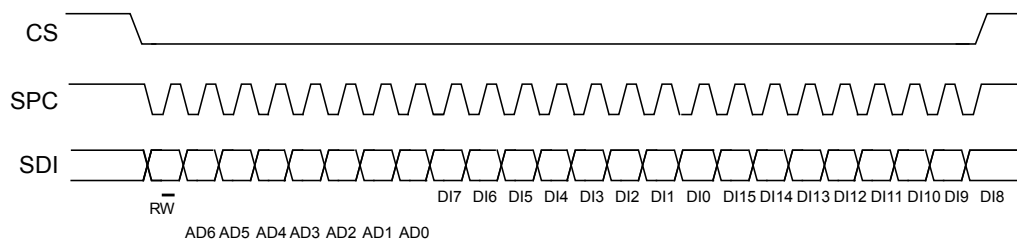
bit 0: WRITE bit. The value is 0.

bit 1-7: address AD(6:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written inside the device (MSb first).

bit 16-... : data DI(...-8). Additional data in multiple byte writes.

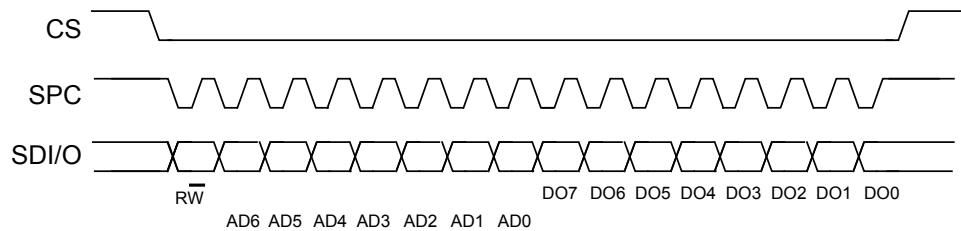
Figure 16. Multiple byte SPI write protocol (2-byte example)



6.2.3 SPI read in 3-wire mode

3-wire mode is entered by setting the PIN_CTRL (0Ch) (SIM) bit equal to 1 (SPI serial interface mode selection).

Figure 17. SPI read protocol in 3-wire mode



The SPI read command is performed with 16 clock pulses:

bit 0: READ bit. The value is 1.

bit 1-7: address AD(6:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

A multiple read command is also available in 3-wire mode.

6.3 MIPI I3C® interface

6.3.1 MIPI I3C® slave interface

The LIS2DUX12 interface includes a MIPI I3C® SDR-only slave interface able to work up to 12.5 MHz of the SCL frequency (compliant with release v1.1 of the specification) with MIPI I3C® SDR embedded features:

- CCC command
- Direct CCC communication (SET and GET)
- Broadcast CCC communication
- Private communications
- Private read and write for single byte
- Multiple read and write
- In-band interrupt request
- Slave reset pattern for reset
- Error detection and recovery (S0 - S6)
- Group address

6.3.2 MIPI I3C® CCC supported commands

The list of MIPI I3C® CCC commands supported by the device is detailed in the following table.

Table 17. MIPI I3C® CCC commands

| Command | Command code | Default | Description |
|-----------|----------------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| ENTDAA | 0x07 | | DAA procedure |
| SETDASA | 0x87 | | Assign dynamic address using static address. Static address values are 0x18 / 0x19 depending on SDO pin. |
| ENEC | 0x80 / 0x00 | | Slave activity control (direct and broadcast) |
| DISEC | 0x81 / 0x01 | | Slave activity control (direct and broadcast) |
| ENTAS0 | 0x82 / 0x02 | | Enter activity state (direct and broadcast) |
| RSTDAA | 0x86 ⁽¹⁾ / 0x06 | | Reset the assigned dynamic address (direct ⁽¹⁾ and broadcast) |
| SETMWL | 0x89 / 0x08 | | Define maximum write length during private write (direct and broadcast) |
| SETMRL | 0x8A / 0x09 | | Define maximum read length during private read (direct and broadcast) |
| SETNEWDA | 0x88 | | Change dynamic address |
| GETMWL | 0x8B | 0x00 0x08 (2 byte) | Get maximum write length during private write |
| GETMRL | 0x8C | 0x00 0x10 0x08 (3 byte) | Get maximum read length during private read |
| GETPID | 0x8D | 0x02 0x08 0x00 0x47 0x92 0x0B | SDO = 1 |
| | | 0x02 0x08 0x00 0x47 0x12 0x0B | SDO = 0 |
| GETBCR | 0x8E | 0x0F (1 byte) | Bus characteristics register |
| GETDCR | 0x8F | 0x41 default | MIPI I3C® device characteristics register |
| GETSTATUS | 0x90 | 0x00 0x00 (2 byte) | Status register |
| GETMXDS | 0x94 | 0x08 0x60 (2 byte) | Return max data speed |
| SETGRPA | 0x9B | | Group address assignment |

| Command | Command code | Default | Description |
|---------|--------------|------------------------------|-------------------------------------------------------------------------------|
| RSTGRPA | 0x2C / 0x9C | | Reset the group address |
| RSTACT | 0x9A / 0x2A | | Configure slave reset action |
| GETCAPS | 0x95 | 0x00 0x11 0x18 0x00 | Provide information about device capabilities and supported extended features |

1. Direct RSTDAAC can be disabled by writing bit DIS_DRSTDAAC in I3C_IF_CTRL (33h) to 1.

6.3.3 Anti-spike filter management on mixed I²C/MIPI I3C® bus

In the LIS2DUX12, the SDA and SCL lines are common to both I²C and I3C. The I²C bus requires anti-spike filters on the SDA and SCL pins that are not compatible with I3C timing.

The device acts as a standard I²C target as long as it is in deep power-down or in soft power-down with an I²C static address.

When in deep power-down, the master must emit START, STATIC ADDRESS R/W (or dynamic address if previously assigned) at open-drain speed using I²C fast mode plus reference timing to perform a power-up command.

When the device is in soft power-down for the first time after a transition from the deep power-down state, the device is capable of detecting and disabling the I²C anti-spike filter after detecting the broadcast address (7'h7E/W). In order to guarantee proper behavior of the device, the I3C master must emit the first START, 7'h7E/W at open-drain speed using I²C fast mode plus reference timing.

After detecting the broadcast address, the device can receive the I3C dynamic address following the I3C push-pull timing. If the device is not assigned a dynamic address, then it continues to operate as an I²C device with no anti-spike filter. For the case in which the host decides to keep the device as I²C with an anti-spike filter, there is a configuration required to keep the anti-spike filter active. This configuration is done by writing the ASF_ON bit to 1 in the I3C_IF_CTRL (33h) register. This configuration forces the anti-spike filter to always be turned on instead of being managed by the communication on the bus.

7 Register mapping

The table given below provides a list of the 8-bit registers embedded in the device and the corresponding addresses.

Table 18. Register map

| Name | Type ⁽¹⁾ | Register address | | Default | Comment |
|--------------------------|---------------------|------------------|----------|----------|---------|
| | | Hex | Binary | | |
| PIN_CTRL | R/W | 0C | 00001100 | 00000000 | |
| WAKE_UP_DUR_EXT | R/W | 0E | 00001110 | 00000000 | |
| WHO_AM_I | R | 0F | 00001111 | 01000111 | |
| CTRL1 | R/W | 10 | 00010000 | 00010000 | |
| CTRL2 | R/W | 11 | 00010001 | 00000000 | |
| CTRL3 | R/W | 12 | 00010010 | 00000000 | |
| CTRL4 | R/W | 13 | 00010011 | 00000000 | |
| CTRL5 | R/W | 14 | 00010100 | 00000000 | |
| FIFO_CTRL | R/W | 15 | 00010101 | 00000000 | |
| FIFO_WTM | R/W | 16 | 00010110 | 00000000 | |
| INTERRUPT_CFG | R/W | 17 | 00010111 | 00000000 | |
| SIXD | R/W | 18 | 00011000 | 00000000 | |
| WAKE_UP_THS | R/W | 1C | 00011100 | 00000000 | |
| WAKE_UP_DUR | R/W | 1D | 00011101 | 00000000 | |
| FREE_FALL | R/W | 1E | 00011110 | 00000000 | |
| MD1_CFG | R/W | 1F | 00011111 | 00000000 | |
| MD2_CFG | R/W | 20 | 00100000 | 00000000 | |
| WAKE_UP_SRC | R | 21 | 00100001 | 00000000 | |
| TAP_SRC | R | 22 | 00100010 | 00000000 | |
| SIXD_SRC | R | 23 | 00100011 | 00000000 | |
| ALL_INT_SRC | R | 24 | 00100100 | 00000000 | |
| STATUS | R | 25 | 00100101 | 00000000 | |
| FIFO_STATUS1 | R | 26 | 00100110 | 00000000 | |
| FIFO_STATUS2 | R | 27 | 00100111 | 00000000 | |
| OUT_X_L | R | 28 | 00101000 | 00000000 | |
| OUT_X_H | R | 29 | 00101001 | 00000000 | |
| OUT_Y_L | R | 2A | 00101010 | 00000000 | |
| OUT_Y_H | R | 2B | 00101011 | 00000000 | |
| OUT_Z_L | R | 2C | 00101100 | 00000000 | |
| OUT_Z_H | R | 2D | 00101101 | 00000000 | |
| OUT_T_L | R | 2E | 00101110 | 00000000 | |
| OUT_T_H | R | 2F | 00101111 | 00000000 | |
| SELF_TEST | R/W | 32 | 00110010 | 00000000 | |
| I3C_IF_CTRL | R/W | 33 | 00110011 | 00000000 | |
| EMB_FUNC_STATUS_MAINPAGE | R | 34 | 00110100 | 00000000 | |

| Name | Type ⁽¹⁾ | Register address | | Default | Comment |
|---------------------|---------------------|------------------|----------|----------|---------|
| | | Hex | Binary | | |
| FSM_STATUS_MAINPAGE | R | 35 | 00110101 | 00000000 | |
| MLC_STATUS_MAINPAGE | R | 36 | 00110110 | 00000000 | |
| SLEEP | R/W | 3D | 00111101 | 00000000 | |
| IF_WAKE_UP | W | 3E | 00111110 | 00000000 | |
| FUNC_CFG_ACCESS | R/W | 3F | 00111111 | 00000000 | |
| FIFO_DATA_OUT_TAG | R | 40 | 01000000 | 00000000 | |
| FIFO_DATA_OUT_X_L | R | 41 | 01000001 | 00000000 | |
| FIFO_DATA_OUT_X_H | R | 42 | 01000010 | 00000000 | |
| FIFO_DATA_OUT_Y_L | R | 43 | 01000011 | 00000000 | |
| FIFO_DATA_OUT_Y_H | R | 44 | 01000100 | 00000000 | |
| FIFO_DATA_OUT_Z_L | R | 45 | 01000101 | 00000000 | |
| FIFO_DATA_OUT_Z_H | R | 46 | 01000110 | 00000000 | |
| FIFO_BATCH_DEC | R/W | 47 | 01000111 | 00000000 | |
| TAP_CFG0 | R/W | 6F | 01101111 | 00000000 | |
| TAP_CFG1 | R/W | 70 | 01110000 | 00000000 | |
| TAP_CFG2 | R/W | 71 | 01110001 | 00000000 | |
| TAP_CFG3 | R/W | 72 | 01110010 | 00000000 | |
| TAP_CFG4 | R/W | 73 | 01110011 | 00000000 | |
| TAP_CFG5 | R/W | 74 | 01110100 | 00000000 | |
| TAP_CFG6 | R/W | 75 | 01110101 | 00000000 | |
| TIMESTAMP3 | R | 7A | 01111010 | 00000000 | |
| TIMESTAMP2 | R | 7B | 01111011 | 00000000 | |
| TIMESTAMP1 | R | 7C | 01111100 | 00000000 | |
| TIMESTAMP0 | R | 7D | 01111101 | 00000000 | |

1. R = read-only register, R/W = readable/writable register

Reserved registers must not be changed. Writing to those registers may cause permanent damage to the device. The content of the registers that are loaded at boot should not be changed. They contain the factory calibration values. Their content is automatically restored when the device is powered up.

8 Register description

8.1 PIN_CTRL (0Ch)

R/W

Table 19. Table 20. PIN_CTRL register

| | | | | | | | |
|-----------|-----------|-------------|-------------|-----------|-----------|-------|-----|
| SDO_PU_EN | SDA_PU_EN | PD_DIS_INT2 | PD_DIS_INT1 | H_LACTIVE | CS_PU_DIS | PP_OD | SIM |
|-----------|-----------|-------------|-------------|-----------|-----------|-------|-----|

Table 20. PIN_CTRL register description

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------|
| SDO_PU_EN | If 1, enables the internal pull-up of the SDO/SA0 pin. |
| SDA_PU_EN | If 1, enables the internal pull-up of the SDA/SDI/SDO pin. |
| PD_DIS_INT2 | If 1, disables the internal pull-down of the INT2 pin. |
| PD_DIS_INT1 | If 1, disables the internal pull-down of the INT1 pin. |
| H_LACTIVE | Interrupt active level (0: interrupts active-high (default); 1: Interrupts active-low) |
| CS_PU_DIS | If 1, disables the internal pull-up of the CS pin. |
| PP_OD | Push-pull/open-drain mode for INT pins (0: INT pins in push-pull mode (default); 1: INT pins in open-drain mode) |
| SIM | SPI 3 or 4-wire mode (0: 4-wire SPI (default); 1: 3-wire SPI) |

8.2 WAKE_UP_DUR_EXT (0Eh)

R/W

Table 21. WAKE_UP_DUR_EXT register

| | | | | | | | |
|------------------|------------------|------------------|-----------------|------------------|------------------|------------------|------------------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | WU_DUR_EXTENDED | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|------------------|------------------|------------------|-----------------|------------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 22. WAKE_UP_DUR_EXT register description

| | |
|-----------------|-----------------------------------------------------------------------------------------------------------------------|
| WU_DUR_EXTENDED | This bit is used to select the resolution of WAKE_UP_DUR[1:0] bits in register WAKE_UP_DUR (1Dh). Default value: 0 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------|

8.3 WHO_AM_I (0Fh)

This register is a read-only register. Its value is fixed at 47h.

Table 23. WHO_AM_I register default values

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

8.4 CTRL1 (10h)

R/W

Table 24. CTRL1 register

| | | | | | | | |
|------------------|-------------|----------|------------|-------------|---------|---------|---------|
| 0 ⁽¹⁾ | INT1_ON_RES | SW_RESET | IF_ADD_INC | DRDY_PULSED | WU_X_EN | WU_Y_EN | WU_Z_EN |
|------------------|-------------|----------|------------|-------------|---------|---------|---------|

1. This bit must be set to 0 for the correct operation of the device.

Table 25. CTRL1 register description

| | |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INT1_ON_RES | Enables routing the interrupt signals configured on the INT1 pin to the RES pin. Default value: 0 (0: disabled; 1: enabled) |
| SW_RESET | Software reset, resets all CTRL registers to their default values. Default value: 0 (0: disabled; 1: enabled) This bit is automatically reset to 0 at the end of the procedure. |
| IF_ADD_INC | Register address is automatically incremented during a multiple-byte access with a serial interface. (0: disabled; 1: enabled (default)) |
| DRDY_PULSED | Enables pulsed data-ready mode (0: data-ready latched mode (returns to 0 only after reading over an interface) (default); 1: data-ready pulsed mode (the data-ready pulses are typ. 90 µs long)) |
| WU_X_EN | Enables wake-up event detection status on X-axis. Default value: 0 (0: disabled; 1: enabled) |
| WU_Y_EN | Enables wake-up event detection status on Y-axis. Default value: 0 (0: disabled; 1: enabled) |
| WU_Z_EN | Enables wake-up event detection status on Z-axis. Default value: 0 (0: disabled; 1: enabled) |

8.5 CTRL2 (11h)

R/W

Table 26. CTRL2 register

| | | | | | | | |
|-----------|----------------|--------------|---------------|-----------|------------------|------------------|------------------|
| INT1_BOOT | INT1_FIFO_FULL | INT1_FIFO_TH | INT1_FIFO_OVR | INT1_DRDY | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|-----------|----------------|--------------|---------------|-----------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 27. CTRL2 register description

| | |
|----------------|------------------------------------------------------------------------------------------|
| INT1_BOOT | Enables boot status on INT1 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT1_FIFO_FULL | Enables FIFO full on INT1 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT1_FIFO_TH | Enables FIFO threshold interrupt on INT1 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT1_FIFO_OVR | Enables overrun interrupt on INT1 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT1_DRDY | Data-ready interrupt on INT1 pin. Default value: 0 (0: disabled; 1: enabled) |

8.6 CTRL3 (12h)

R/W

Table 28. CTRL3 register

| | | | | | | | |
|-----------|----------------|--------------|---------------|-----------|-------|-----------|-----------|
| INT2_BOOT | INT2_FIFO_FULL | INT2_FIFO_TH | INT2_FIFO_OVR | INT2_DRDY | HP_EN | ST_SIGN_Y | ST_SIGN_X |
|-----------|----------------|--------------|---------------|-----------|-------|-----------|-----------|

Table 29. CTRL3 register description

| | |
|----------------|--------------------------------------------------------------------------------------------------|
| INT2_BOOT | Enables boot status on INT2 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT2_FIFO_FULL | Enables FIFO full on INT2 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT2_FIFO_TH | Enables FIFO threshold interrupt on INT2 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT2_FIFO_OVR | Enables overrun interrupt on INT2 pin. Default value: 0 (0: disabled; 1: enabled) |
| INT2_DRDY | Data-ready interrupt on INT2 pin. Default value: 0 (0: disabled; 1: enabled) |
| HP_EN | Enables high-performance mode. Default value: 0 (0: low-power mode; 1: high-performance mode) |
| ST_SIGN_Y | Configures the sign of the self-test for the Y-axis. Default value: 0 |
| ST_SIGN_X | Configures the sign of the self-test for the X-axis. Default value: 0 |

8.7 CTRL4 (13h)

R/W

Table 30. CTRL4 register

| | | | | | | | |
|------------|------------|-----|-------------|---------|------------------|-----|------|
| INACT_ODR1 | INACT_ODR0 | BDU | EMB_FUNC_EN | FIFO_EN | 0 ⁽¹⁾ | SOC | BOOT |
|------------|------------|-----|-------------|---------|------------------|-----|------|

1. This bit must be set to 0 for the correct operation of the device.

Table 31. CTRL4 register description

| | |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INACT_ODR[1:0] | If the activity/inactivity function is enabled, then these bits select the accelerometer ODR during inactivity status, see Table 32 . |
| BDU | Sensing chain block data update (0: output registers MSByte and LSByte independent continuous update (default); 1: output registers are not updated until MSByte and LSByte have both been read) |
| EMB_FUNC_EN | Enables embedded functions. To be set to 1 before configuring the embedded functions. Default value: 0 (0: disabled; 1: enabled) |
| FIFO_EN | Enables batching in FIFO. To be set to 1 before configuring the FIFO buffer. Default value: 0 (0: disabled; 1: enabled) |
| SOC | Start of conversion bit. When one-shot mode using the interface (ODR[3:0] = 1111 in the CTRL5 (14h) register) is enabled, this bit provides the start for the measurement. This bit is automatically cleared. |
| BOOT | Reboots memory content. Default value: 0 (0: normal operating mode; 1: reboot memory content) This bit is automatically cleared. For proper execution of the boot procedure, set the device in high-performance mode. |

Table 32. ODR frequency in inactivity state

| INACT_ODR1 | INACT_ODR0 | Frequency [Hz] |
|------------|------------|-----------------------------------------------------------------------------------------------------------------|
| 0 | 0 | Stationary/motion detection: when selected, no ODR change is done if inactivity condition is detected (default) |
| 0 | 1 | 1.6 |
| 1 | 0 | 3 |
| 1 | 1 | 25 |

8.8 CTRL5 (14h)

R/W

Table 33. CTRL5 register

| | | | | | | | |
|------|------|------|------|-----|-----|-----|-----|
| ODR3 | ODR2 | ODR1 | ODR0 | BW1 | BW0 | FS1 | FS0 |
|------|------|------|------|-----|-----|-----|-----|

Table 34. CTRL5 register description

| | |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ODR[3:0] | ODR selection, see Table 35 . |
| BW[1:0] | <p>Selects the device bandwidth, which is dependent on the ODR selected. In high-performance mode (all ODR values) and in low-power mode for ODR ≥ 50 Hz, the available bandwidths are:</p> <p>ODR/2 (BW[1:0] = 00);</p> <p>ODR/4 (BW[1:0] = 01);</p> <p>ODR/8 (BW[1:0] = 10);</p> <p>ODR/16 (BW[1:0] = 11).</p> <p>In low-power mode for ODR < 50 Hz, refer to Table 36.</p> |
| FS[1:0] | Sets the full scale, see Table 37 . |

Table 35. Operating modes

| ODR[3:0] | Operating mode |
|----------|------------------------------|
| 0000 | Power-down |
| 0001 | 1.6 Hz in ultralow-power |
| 0010 | 3 Hz in ultralow-power |
| 0011 | 25 Hz in ultralow-power |
| 0100 | 6 Hz |
| 0101 | 12.5 Hz |
| 0110 | 25 Hz |
| 0111 | 50 Hz |
| 1000 | 100 Hz |
| 1001 | 200 Hz |
| 1010 | 400 Hz |
| 1011 | 800 Hz |
| 1110 | One-shot using the INT2 pin |
| 1111 | One-shot using the interface |

Table 36. Bandwidth selection (low-power mode with ODR < 50 Hz)

| ODR [Hz] | BW[1:0] | BW [Hz] |
|----------|---------|---------|
| 6 | 00 | - |
| | 01 | - |
| | 10 | - |
| | 11 | 3 |
| 12.5 | 00 | - |
| | 01 | - |
| | 10 | 6 |
| | 11 | 3 |
| 25 | 00 | - |
| | 01 | 12.5 |
| | 10 | 6 |
| | 11 | 3 |

Table 37. Full-scale selection

| FS1 | FS0 | Full scale |
|-----|-----|------------|
| 0 | 0 | $\pm 2 g$ |
| 0 | 1 | $\pm 4 g$ |
| 1 | 0 | $\pm 8 g$ |
| 1 | 1 | $\pm 16 g$ |

8.9 FIFO_CTRL (15h)

R/W

| | | | | | | | |
|------------|------------|------------------|------------------|-------------|------------|------------|------------|
| CFG_CHG_EN | FIFO_DEPTH | 0 ⁽¹⁾ | 0 ⁽¹⁾ | STOP_ON_FTH | FIFO_MODE2 | FIFO_MODE1 | FIFO_MODE0 |
|------------|------------|------------------|------------------|-------------|------------|------------|------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 38. FIFO_CTRL register description

| | |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CFG_CHG_EN | Enables batching in FIFO of the device configuration and timestamp value when the ODR (output data rate) or the BDR (batch data rate) changes. Default value: 0 (0: disabled; 1: enabled) |
| FIFO_DEPTH | If 1, enables 2x depth mode for FIFO buffer. |
| STOP_ON_FTH | Sensing chain FIFO stop values memorization at threshold level. (0: FIFO depth is not limited (default); 1: FIFO depth is limited to threshold level) |
| FIFO_MODE[2:0] ⁽¹⁾ | Different FIFO modes are enabled as shown in Table 39. |

1. User must set the FIFO_EN bit to 1 in the CTRL4 (13h) register before setting the FIFO_MODE[2:0] bits.

Table 39. Selection of FIFO mode

| FIFO_MODE2 | FIFO_MODE1 | FIFO_MODE0 | Mode |
|------------|------------|------------|-------------------------------------------------------------------------------------|
| 0 | 0 | 0 | Bypass mode |
| 0 | 0 | 1 | FIFO mode: stops collecting data when FIFO is full |
| 0 | 1 | 0 | Reserved |
| 0 | 1 | 1 | Continuous-to-FIFO: stream mode until trigger is deasserted, then FIFO mode |
| 1 | 0 | 0 | Bypass-to-continuous: bypass mode until trigger is deasserted, then continuous mode |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Continuous mode: if the FIFO is full, the new sample overwrites the older sample. |
| 1 | 1 | 1 | Bypass-to-FIFO: bypass mode until trigger is deasserted, then FIFO mode |

8.10 FIFO_WTM (16h)

R/W

Table 40. FIFO_WTM register

| | | | | | | | |
|--------------|------|------|------|------|------|------|------|
| XL_ONLY_FIFO | FTH6 | FTH5 | FTH4 | FTH3 | FTH2 | FTH1 | FTH0 |
|--------------|------|------|------|------|------|------|------|

Table 41. FIFO_WTM register description

| | |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XL_ONLY_FIFO | FIFO data configuration. If this bit is set to 0 (default), both the accelerometer data and the temperature data are stored in FIFO. If this bit is set to 1, only accelerometer data are stored in FIFO. |
| FTH[6:0] | FIFO watermark threshold, maximum value is 127. |

8.11 INTERRUPT_CFG (17h)

R/W

Table 42. INTERRUPT_CFG register

| | | | | | | | |
|--------------|------------------|------------|------------------|----------------------|---------------------|-----|-------------------|
| TIMESTAMP_EN | 0 ⁽¹⁾ | WAKE_THS_W | 0 ⁽¹⁾ | SLEEP_STAT_US_ON_INT | DIS_RST_LIR_ALL_INT | LIR | INTERRUPTS_ENABLE |
|--------------|------------------|------------|------------------|----------------------|---------------------|-----|-------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 43. INTERRUPT_CFG register description

| | |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIMESTAMP_EN | Enables timestamp counter. The counter is readable in TIMESTAMP3 (7Ah) , TIMESTAMP2 (7Bh) , TIMESTAMP1 (7Ch) , and TIMESTAMP0 (7Dh) . Default value: 0 (0: disabled; 1: enabled) |
| WAKE_THS_W | Weight of 1 LSB of wake-up threshold. Default value: 0 (0: 1 LSB = $FS_{XL} / (2^6)$; 1: 1 LSB = $FS_{XL} / (2^8)$) |
| SLEEP_STATUS_ON_INT | Sends the sleep status instead of sleep change to INT pins (only if INT1_SLEEP_CHANGE or INT2_SLEEP_CHANGE bits are enabled, respectively in registers MD1_CFG (1Fh) and MD2_CFG (20h)). Default value: 0 (0: sleep change on INT pins; 1: sleep status on INT pins) |
| DIS_RST_LIR_ALL_INT | If 1, disables the reset of the interrupt flags when ALL_INT_SRC (24h) is read. |
| LIR | Interrupt mode configuration (see Table 8). Default value: 0 (0: interrupt level mode; 1: interrupt latched mode) |
| INTERRUPTS_ENABLE | Enables basic interrupts (6D/4D, free-fall, wake-up, single/double/triple-tap, activity/inactivity). Default value: 0 (0: interrupt disabled; 1: interrupt enabled) |

8.12 SIXD (18h)

R/W

Table 44. SIXD register

| | | | | | | | |
|--------|----------|----------|------------------|------------------|------------------|------------------|------------------|
| D4D_EN | D6D_THS1 | D6D_THS0 | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|--------|----------|----------|------------------|------------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 45. SIXD register description

| | |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| D4D_EN | Enables 4D orientation detection. Z-axis position detection is disabled. Default value: 0 (0: disabled; 1: enabled) |
| D6D_THS[1:0] | Thresholds for 4D/6D function (00: 80 degrees (default); 01: 70 degrees; 10: 60 degrees; 11: 50 degrees) |

8.13 WAKE_UP_THS (1Ch)

R/W

Table 46. WAKE_UP_THS register

| | | | | | | | |
|------------------|----------|---------|---------|---------|---------|---------|---------|
| 0 ⁽¹⁾ | SLEEP_ON | WK_THS5 | WK_THS4 | WK_THS3 | WK_THS2 | WK_THS1 | WK_THS0 |
|------------------|----------|---------|---------|---------|---------|---------|---------|

1. This bit must be set to 0 for the correct operation of the device.

Table 47. WAKE_UP_THS register description

| | |
|-------------|---------------------------------------------------------------------------------------------------------|
| SLEEP_ON | If 1, activity/inactivity function is enabled. |
| WK_THS[5:0] | Threshold for wake-up: 1 LSB weight depends on WAKE_THS_W in INTERRUPT_CFG (17h). Default value: 000000 |

8.14 WAKE_UP_DUR (1Dh)

R/W

Table 48. WAKE_UP_DUR register

| | | | | | | | |
|---------|-----------|-----------|-----------|------------|------------|------------|------------|
| FF_DUR5 | WAKE_DUR1 | WAKE_DUR0 | ST_SIGN_Z | SLEEP_DUR3 | SLEEP_DUR2 | SLEEP_DUR1 | SLEEP_DUR0 |
|---------|-----------|-----------|-----------|------------|------------|------------|------------|

Table 49. WAKE_UP_DUR register description

| | |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FF_DUR5 | Free-fall duration. Default value: 0 In conjunction with FF_DUR[4:0] in FREE_FALL (1Eh) . 1 LSB: 1 ODR_time |
| WAKE_DUR[1:0] | Wake-up duration. Default value: 00 When the WU_DUR_EXTENDED bit in register WAKE_UP_DUR_EXT (0Eh) is set to 0, 1LSB of WAKE_DUR = 1 ODR_time, otherwise the following durations are selectable: (00: 3 ODR_time; 01: 7 ODR_time; 10: 11 ODR_time; 11: 15 ODR_time) |
| ST_SIGN_Z | Configures the sign of the self-test for the Z-axis. Default value: 0 |
| SLEEP_DUR[3:0] | Duration to go in sleep mode. Default value: 0000 which corresponds to 16 ODR_time 1 LSB: 512 ODR_time |

8.15 FREE_FALL (1Eh)

R/W

Table 50. FREE_FALL register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| FF_DUR4 | FF_DUR3 | FF_DUR2 | FF_DUR1 | FF_DUR0 | FF_THS2 | FF_THS1 | FF_THS0 |
|---------|---------|---------|---------|---------|---------|---------|---------|

Table 51. FREE_FALL register description

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| FF_DUR[4:0] | Free-fall duration. Default value: 0 In conjunction with FF_DUR5 in WAKE_UP_DUR (1Dh) . 1 LSB: 1 ODR_time |
| FF_THS[2:0] | Free-fall threshold (000: 156 mg; 001: 219 mg; 010: 250 mg; 011: 312 mg; 100: 344 mg; 101: 406 mg; 110: 469 mg; 111: 500 mg) |

8.16 MD1_CFG (1Fh)

R/W

Each bit in this register enables a signal to be carried over INT1; the pin's output is the OR combination of the signals selected here and in register CTRL2 (11h).

Table 52. MD1_CFG register

| | | | | | | | |
|-------------------|------------------|---------|---------|----------|---------|----------------|---------------|
| INT1_SLEEP_CHANGE | 0 ⁽¹⁾ | INT1_WU | INT1_FF | INT1_TAP | INT1_6D | INT1_TIMESTAMP | INT1_EMB_FUNC |
|-------------------|------------------|---------|---------|----------|---------|----------------|---------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 53. MD1_CFG register description

| | |
|-------------------|-------------------------------------------------------------------------------------------|
| INT1_SLEEP_CHANGE | Enables sleep change (or sleep status, depending on SLEEP_STATUS_ON_INT bit) on INT1 pin. |
| INT1_WU | Enables routing wake-up event to the INT1 pin. |
| INT1_FF | Enables routing free-fall event to the INT1 pin. |
| INT1_TAP | Enables routing tap event to the INT1 pin. |
| INT1_6D | Enables routing 6D recognition event to the INT1 pin. |
| INT1_TIMESTAMP | Enables routing the alert of timestamp overflow within 2.5 ms to the INT1 pin. |
| INT1_EMB_FUNC | Enables routing embedded functions event to the INT1 pin. |

8.17 MD2_CFG (20h)

R/W

Each bit in this register enables a signal to be carried over INT2; the pin's output is the OR combination of the signals selected here and in register CTRL3 (12h).

Table 54. MD2_CFG register

| | | | | | | | |
|-------------------|------------------|---------|---------|----------|---------|----------------|---------------|
| INT2_SLEEP_CHANGE | 0 ⁽¹⁾ | INT2_WU | INT2_FF | INT2_TAP | INT2_6D | INT2_TIMESTAMP | INT2_EMB_FUNC |
|-------------------|------------------|---------|---------|----------|---------|----------------|---------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 55. MD2_CFG register description

| | |
|-------------------|-------------------------------------------------------------------------------------------|
| INT2_SLEEP_CHANGE | Enables sleep change (or sleep status, depending on SLEEP_STATUS_ON_INT bit) on INT2 pin. |
| INT2_WU | Enables routing wake-up event to the INT2 pin. |
| INT2_FF | Enables routing free-fall event to the INT2 pin. |
| INT2_TAP | Enables routing tap event to the INT2 pin. |
| INT2_6D | Enables routing 6D recognition event to the INT2 pin. |
| INT2_TIMESTAMP | Enables routing the alert of timestamp overflow within 2.5 ms to the INT2 pin. |
| INT2_EMB_FUNC | Enables routing embedded functions event to INT2. |

8.18 WAKE_UP_SRC (21h)

R

Table 56. WAKE_UP_SRC register

| | | | | | | | |
|---|-----------------|-------|-------------|-------|------|------|------|
| - | SLEEP_CHANGE_IA | FF_IA | SLEEP_STATE | WU_IA | X_WU | Y_WU | Z_WU |
|---|-----------------|-------|-------------|-------|------|------|------|

Table 57. WAKE_UP_SRC register description

| | |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| SLEEP_CHANGE_IA | Detection of change in activity/inactivity status. Default value: 0 (0: change status not detected; 1: change status detected) |
| FF_IA | Free-fall event detection status. Default value: 0 (0: free-fall event not detected; 1: free-fall event detected) |
| SLEEP_STATE | Sleep status bit. Default value: 0 (0: activity status; 1: inactivity status) |
| WU_IA | Wake-up event detection status. Default value: 0 (0: wake-up event not detected; 1: wake-up event detected) |
| X_WU | Wake-up event detection status on X-axis. Default value: 0 (0: wake-up event on X-axis not detected; 1: wake-up event on X-axis detected) |
| Y_WU | Wake-up event detection status on Y-axis. Default value: 0 (0: wake-up event on Y-axis not detected; 1: wake-up event on Y-axis detected) |
| Z_WU | Wake-up event detection status on Z-axis. Default value: 0 (0: wake-up event on Z-axis not detected; 1: wake-up event on Z-axis detected) |

8.19 TAP_SRC (22h)

R

Table 58. TAP_SRC register

| | | | | | | | |
|--------|---------------|---------------|---------------|---|---|---|---|
| TAP_IA | SINGLE_TAP_IA | DOUBLE_TAP_IA | TRIPLE_TAP_IA | - | - | - | - |
|--------|---------------|---------------|---------------|---|---|---|---|

Table 59. TAP_SRC register description

| | |
|---------------|-------------------------------------------------------------------------------------------------------------------------|
| TAP_IA | Tap event detection status. Default: 0 (0: tap event not detected; 1: tap event detected) |
| SINGLE_TAP_IA | Single-tap event status. Default value: 0 (0: single-tap event not detected; 1: single-tap event detected) |
| DOUBLE_TAP_IA | Double-tap event detection status. Default value: 0 (0: double-tap event not detected; 1: double-tap event detected) |
| TRIPLE_TAP_IA | Triple-tap event detection status. Default value: 0 (0: triple-tap event not detected; 1: triple-tap event detected) |

8.20 SIXD_SRC (23h)

R

Table 60. SIXD_SRC register

| | | | | | | | |
|---|--------|----|----|----|----|----|----|
| - | D6D_IA | ZH | ZL | YH | YL | XH | XL |
|---|--------|----|----|----|----|----|----|

Table 61. SIXD_SRC register description

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| D6D_IA | Source of change in 6D/4D orientation. Default value: 0 (0: change orientation not detected; 1: change orientation detected) |
| ZH | Z-axis high event (over threshold). Default value: 0 (0: event not detected; 1: event (over threshold) detected) |
| ZL | Z-axis low event (under threshold). Default value: 0 (0: event not detected; 1: event (under threshold) detected) |
| YH | Y-axis high event (over threshold). Default value: 0 (0: event not detected; 1: event (over-threshold) detected) |
| YL | Y-axis low event (under threshold). Default value: 0 (0: event not detected; 1: event (under threshold) detected) |
| XH | X-axis high event (over threshold). Default value: 0 (0: event not detected; 1: event (over threshold) detected) |
| XL | X-axis low event (under threshold). Default value: 0 (0: event not detected; 1: event (under threshold) detected) |

8.21 ALL_INT_SRC (24h)

R

Table 62. ALL_INT_SRC register

| | | | | | | | |
|---|---------------------|------------|----------------|----------------|----------------|-----------|-----------|
| - | SLEEP_CHANGE_IA_ALL | D6D_IA_ALL | TRIPLE_TAP_ALL | DOUBLE_TAP_ALL | SINGLE_TAP_ALL | WU_IA_ALL | FF_IA_ALL |
|---|---------------------|------------|----------------|----------------|----------------|-----------|-----------|

Table 63. ALL_INT_SRC register description

| | |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| SLEEP_CHANGE_IA_ALL | Detection of change in activity/inactivity status. Default value: 0 (0: change in status not detected; 1: change in status detected) |
| D6D_IA_ALL | Source of change in 6D/4D orientation. Default value: 0 (0: change in orientation not detected; 1: change in orientation detected) |
| TRIPLE_TAP_ALL | Triple-tap event status. Default value: 0 (0: event not detected, 1: event detected) |
| DOUBLE_TAP_ALL | Double-tap event status. Default value: 0 (0: event not detected, 1: event detected) |
| SINGLE_TAP_ALL | Single-tap event status. Default value: 0 (0: event not detected, 1: event detected) |
| WU_IA_ALL | Wake-up event status. Default value: 0 (0: event not detected, 1: event detected) |
| FF_IA_ALL | Free-fall event status. Default value: 0 (0: event not detected, 1: event detected) |

8.22 STATUS (25h)

R

Table 64. STATUS register

| | | | | | | | |
|---|---|------------|---|---|---|---|------|
| - | - | INT_GLOBAL | - | - | - | - | DRDY |
|---|---|------------|---|---|---|---|------|

Table 65. STATUS register description

| | |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INT_GLOBAL | This bit is 1 if one of the following events occur: <ul style="list-style-type: none"> detection of change in activity/inactivity status source of change in 6D/4D orientation single/double/triple-tap event status wake-up event detection status free-fall event detection status sleep event status |
| DRDY | This bit is 1 when new accelerometer data is available and until the MSB of one of the output registers has been read. |

8.23 FIFO_STATUS1 (26h)

R

Table 66. FIFO_STATUS1 register

| | | | | | | | |
|-------------|-------------|---|---|---|---|---|---|
| FIFO_WTM_IA | FIFO_OVR_IA | - | - | - | - | - | - |
|-------------|-------------|---|---|---|---|---|---|

Table 67. FIFO_STATUS1 register description

| | |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FIFO_WTM_IA | FIFO watermark status. The watermark is set through bits FTH[6:0] in FIFO_WTM (16h). (0: FIFO filling is lower than WTM; 1: FIFO filling is equal to or higher than WTM) |
| FIFO_OVR_IA | FIFO overrun status: 1 if FIFO has overwritten data. |

8.24 FIFO_STATUS2 (27h)

R

Table 68. FIFO_STATUS2 register

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| FSS7 | FSS6 | FSS5 | FSS4 | FSS3 | FSS2 | FSS1 | FSS0 |
|------|------|------|------|------|------|------|------|

Table 69. FIFO_STATUS2 register description

| | |
|----------|--------------------------------------|
| FSS[7:0] | Number of unread data stored in FIFO |
|----------|--------------------------------------|

8.25 OUT_X_L (28h)

R

Table 70. OUT_X_L register

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUTX7 | OUTX6 | OUTX5 | OUTX4 | OUTX3 | OUTX2 | OUTX1 | OUTX0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

Table 71. OUT_X_L register description

| | |
|-----------|-----------------------|
| OUTX[7:0] | LSBs of X data output |
|-----------|-----------------------|

8.26 OUT_X_H (29h)

R

Table 72. OUT_X_H register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|-------|-------|
| OUTX15 | OUTX14 | OUTX13 | OUTX12 | OUTX11 | OUTX10 | OUTX9 | OUTX8 |
|--------|--------|--------|--------|--------|--------|-------|-------|

Table 73. OUT_X_H register description

| | |
|------------|-----------------------|
| OUTX[15:8] | MSBs of X data output |
|------------|-----------------------|

8.27 OUT_Y_L (2Ah)

R

Table 74. OUT_Y_L register

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUTY7 | OUTY6 | OUTY5 | OUTY4 | OUTY3 | OUTY2 | OUTY1 | OUTY0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

Table 75. OUT_Y_L register description

| | |
|-----------|-----------------------|
| OUTY[7:0] | LSBs of Y data output |
|-----------|-----------------------|

8.28 OUT_Y_H (2Bh)

R

Table 76. OUT_Y_H register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|-------|-------|
| OUTY15 | OUTY14 | OUTY13 | OUTY12 | OUTY11 | OUTY10 | OUTY9 | OUTY8 |
|--------|--------|--------|--------|--------|--------|-------|-------|

Table 77. OUT_Y_H register description

| | |
|------------|------------------------|
| OUTY[15:8] | MSBs of Y data output. |
|------------|------------------------|

8.29 OUT_Z_L (2Ch)

R

Table 78. OUT_Z_L register

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUTZ7 | OUTZ6 | OUTZ5 | OUTZ4 | OUTZ3 | OUTZ2 | OUTZ1 | OUTZ0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

Table 79. OUT_Z_L register description

| | |
|-----------|-----------------------|
| OUTZ[7:0] | LSBs of Z data output |
|-----------|-----------------------|

8.30 OUT_Z_H (2Dh)

R

Table 80. OUT_Z_H register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|-------|-------|
| OUTZ15 | OUTZ14 | OUTZ13 | OUTZ12 | OUTZ11 | OUTZ10 | OUTZ9 | OUTZ8 |
|--------|--------|--------|--------|--------|--------|-------|-------|

Table 81. OUT_Z_H register description

| | |
|------------|-----------------------|
| OUTZ[15:8] | MSBs of Z data output |
|------------|-----------------------|

8.31 OUT_T_L (2Eh)
R

Table 82. OUT_T_L register

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| OUTT7 | OUTT6 | OUTT5 | OUTT4 | OUTT3 | OUTT2 | OUTT1 | OUTT0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

Table 83. OUT_T_L register description

| | |
|-----------|----------------------------------------------------------------------------------------------------------------------|
| OUTT[7:0] | LSBs of temperature data output. The four least significant bits are always zero since the data is in 12-bit format. |
|-----------|----------------------------------------------------------------------------------------------------------------------|

8.32 OUT_T_H (2Fh)
R

Table 84. OUT_T_H register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|-------|-------|
| OUTT15 | OUTT14 | OUTT13 | OUTT12 | OUTT11 | OUTT10 | OUTT9 | OUTT8 |
|--------|--------|--------|--------|--------|--------|-------|-------|

Table 85. OUT_T_H register description

| | |
|------------|---------------------------------|
| OUTT[15:8] | MSBs of temperature data output |
|------------|---------------------------------|

8.33 SELF_TEST (32h)

R/W

Table 86. SELF_TEST register

| | | | | | | | |
|---|---|-----|-----|---|---|---|---|
| - | - | ST1 | ST0 | - | - | - | - |
|---|---|-----|-----|---|---|---|---|

Table 87. SELF_TEST register description

| | |
|---------|--------------------------------------------------------------------|
| ST[1:0] | These bits enable data acquisition during the self-test procedure. |
|---------|--------------------------------------------------------------------|

8.34 I3C_IF_CTRL (33h)

R/W

Table 88. I3C_IF_CTRL register

| | | | | | | | |
|-------------|---|--------|---|---|---|---------------|---------------|
| DIS_DRSTDAA | - | ASF_ON | - | - | - | BUS_ACT_SEL_1 | BUS_ACT_SEL_0 |
|-------------|---|--------|---|---|---|---------------|---------------|

Table 89. I3C_IF_CTRL register description

| | |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| DIS_DRSTDAA | If 0, direct RSTDAA is supported. If 1, direct RSTDAA is disabled. |
| ASF_ON | If 1, enables the anti-spike filter even if the dynamic address is assigned. |
| BUS_ACT_SEL_[1:0] | Bus available time selection for IBI (in-band interrupt): (00: 20 μ s; 01: 50 μ s (default); 10: 1 ms; 11: 25 ms) |

8.35 EMB_FUNC_STATUS_MAINPAGE (34h)

R

Table 90. EMB_FUNC_STATUS_MAINPAGE register

| | | | | | | | |
|-----------|---|-----------|---------|-------------|---|---|---|
| IS_FSM_LC | 0 | IS_SIGMOT | IS_TILT | IS_STEP_DET | 0 | 0 | 0 |
|-----------|---|-----------|---------|-------------|---|---|---|

Table 91. EMB_FUNC_STATUS_MAINPAGE register description

| | |
|-------------|-------------------------------------------------------------------------------------------------------------|
| IS_FSM_LC | Interrupt status bit for FSM long counter timeout interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_SIGMOT | Interrupt status bit for significant motion detection. (1: interrupt detected; 0: no interrupt) |
| IS_TILT | Interrupt status bit for tilt detection. (1: interrupt detected; 0: no interrupt) |
| IS_STEP_DET | Interrupt status bit for step detection. (1: interrupt detected; 0: no interrupt) |

8.36 FSM_STATUS_MAINPAGE (35h)

R

Table 92. FSM_STATUS_MAINPAGE register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| IS_FSM8 | IS_FSM7 | IS_FSM6 | IS_FSM5 | IS_FSM4 | IS_FSM3 | IS_FSM2 | IS_FSM1 |
|---------|---------|---------|---------|---------|---------|---------|---------|

Table 93. FSM_STATUS_MAINPAGE register description

| | |
|---------|-----------------------------------------------------------------------------------------|
| IS_FSM8 | Interrupt status bit for FSM8 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM7 | Interrupt status bit for FSM7 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM6 | Interrupt status bit for FSM6 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM5 | Interrupt status bit for FSM5 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM4 | Interrupt status bit for FSM4 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM3 | Interrupt status bit for FSM3 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM2 | Interrupt status bit for FSM2 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM1 | Interrupt status bit for FSM1 interrupt event. (1: interrupt detected; 0: no interrupt) |

8.37 MLC_STATUS_MAINPAGE (36h)

R

Table 94. MLC_STATUS_MAINPAGE register

| | | | | | | | |
|---|---|---|---|---------|---------|---------|---------|
| 0 | 0 | 0 | 0 | IS_MLC4 | IS_MLC3 | IS_MLC2 | IS_MLC1 |
|---|---|---|---|---------|---------|---------|---------|

Table 95. MLC_STATUS_MAINPAGE register description

| | |
|---------|-----------------------------------------------------------------------------------------|
| IS_MLC4 | Interrupt status bit for MLC4 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC3 | Interrupt status bit for MLC3 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC2 | Interrupt status bit for MLC2 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC1 | Interrupt status bit for MLC1 interrupt event. (1: interrupt detected; 0: no interrupt) |

8.38 SLEEP (3Dh)

R/W

Table 96. SLEEP register

| | | | | | | | |
|---|---|---|---|---|---|---|---------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | DEEP_PD |
|---|---|---|---|---|---|---|---------|

Table 97. SLEEP register description

| | |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEEP_PD | If this bit is set to 1, the device enters deep a power-down state. Once the device is in a deep power-down state, the register is no longer accessible. |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|

8.39 IF_WAKE_UP (3Eh)

W

Table 98. IF_WAKE_UP register

| | | | | | | | |
|---|---|---|---|---|---|---|---------|
| - | - | - | - | - | - | - | SOFT_PD |
|---|---|---|---|---|---|---|---------|

Table 99. IF_WAKE_UP register description

| | |
|---------|--------------------------------------------------------------------------------------------------------------------------------|
| SOFT_PD | This bit allows the transition from deep power-down to soft power-down when the SPI interface is used. This bit is write only. |
|---------|--------------------------------------------------------------------------------------------------------------------------------|

8.40 FUNC_CFG_ACCESS (3Fh)

Enable embedded functions register (R/W)

Table 100. FUNC_CFG_ACCESS register

| | | | | | | | |
|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|----------------|
| EMB_FUNC_REG_ACCESS | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | FSM_WR_CTRL_EN |
|---------------------|------------------|------------------|------------------|------------------|------------------|------------------|----------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 101. FUNC_CFG_ACCESS register description

| | |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EMB_FUNC_REG_ACCESS | Enables access to the embedded functions configuration registers ⁽¹⁾ . Default value: 0 |
| FSM_WR_CTRL_EN | Enables the control of the CTRL registers to FSM (FSM can change some configurations of the device autonomously). Default value: 0 (0: disabled; 1: enabled) |

1. Details concerning the embedded functions configuration registers are available in [Section 9 Embedded functions register mapping](#) and [Section 10 Embedded functions register description](#)

8.41 FIFO_DATA_OUT_TAG (40h)

This register contains the TAG values that distinguish the different kinds of data that can be batched in FIFO (R).

Table 102. FIFO_DATA_OUT_TAG register

| | | | | | | | |
|--------------|--------------|--------------|--------------|--------------|---|---|---|
| TAG_SENSOR_4 | TAG_SENSOR_3 | TAG_SENSOR_2 | TAG_SENSOR_1 | TAG_SENSOR_0 | 0 | 0 | - |
|--------------|--------------|--------------|--------------|--------------|---|---|---|

Table 103. FIFO_DATA_OUT_TAG register description

| | |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAG_SENSOR_[4:0] | FIFO tag. Identifies the sensor in FIFO_DATA_OUT_X_L (41h) and FIFO_DATA_OUT_X_H (42h), FIFO_DATA_OUT_Y_L (43h) and FIFO_DATA_OUT_Y_H (44h), and FIFO_DATA_OUT_Z_L (45h) and FIFO_DATA_OUT_Z_H (46h). For details, refer to Table 104. |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Table 104. Identification of sensor in FIFO

| TAG_SENSOR_[4:0] | Data in FIFO |
|------------------|-----------------------------------------|
| 00000 | FIFO empty |
| 00010 | Accelerometer and temperature |
| 00011 | Accelerometer only data (2x depth mode) |
| 00100 | Timestamp or CFG_CHG |
| 10010 | Step counter |
| 11010 | MLC result |
| 11011 | MLC filter |
| 11100 | MLC feature |
| 11101 | FSM result |

8.42 FIFO_DATA_OUT_X_L (41h) and FIFO_DATA_OUT_X_H (42h)

FIFO data output X (R)

Table 105. FIFO_DATA_OUT_X_H and FIFO_DATA_OUT_X_L registers

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Table 106. FIFO_DATA_OUT_X_H and FIFO_DATA_OUT_X_L register description

| | |
|---------|--------------------|
| D[15:0] | FIFO X-axis output |
|---------|--------------------|

8.43 FIFO_DATA_OUT_Y_L (43h) and FIFO_DATA_OUT_Y_H (44h)

FIFO data output Y (R)

Table 107. FIFO_DATA_OUT_Y_H and FIFO_DATA_OUT_Y_L registers

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Table 108. FIFO_DATA_OUT_Y_H and FIFO_DATA_OUT_Y_L register description

| | |
|---------|--------------------|
| D[15:0] | FIFO Y-axis output |
|---------|--------------------|

8.44 FIFO_DATA_OUT_Z_L (45h) and FIFO_DATA_OUT_Z_H (46h)

FIFO data output Z (R)

Table 109. FIFO_DATA_OUT_Z_H and FIFO_DATA_OUT_Z_L registers

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Table 110. FIFO_DATA_OUT_Z_H and FIFO_DATA_OUT_Z_L register description

| | |
|---------|--------------------|
| D[15:0] | FIFO Z-axis output |
|---------|--------------------|

8.45 FIFO_BATCH_DEC (47h)

R/W

Table 111. FIFO_BATCH_DEC register

| | | | | | | | |
|---|---|---|----------------|----------------|----------|----------|----------|
| 0 | 0 | 0 | DEC_TS_BATCH_1 | DEC_TS_BATCH_0 | BDR_XL_2 | BDR_XL_1 | BDR_XL_0 |
|---|---|---|----------------|----------------|----------|----------|----------|

Table 112. FIFO_BATCH_DEC register description

| | |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEC_TS_BATCH_[1:0] | Selects decimation for timestamp batching in FIFO. The write rate is the accelerometer BDR divided by the decimation decoder. (00: Timestamp not batched in FIFO (default); 01: Decimation 1: BDR_XL[Hz]; 10: Decimation 8: BDR_XL[Hz]/8 [Hz]; 11: Decimation 32: BDR_XL[Hz]/32 [Hz]) |
| BDR_XL_[2:0] | Selects the batch data rate (write frequency in FIFO) for accelerometer data, see Table 113 . |

Table 113. Accelerometer batch data rate

| BDR_XL_[2:0] | Accelerometer batch frequency |
|--------------|-----------------------------------|
| 000 | ODR (default) |
| 001 | ODR/2 |
| 010 | ODR/4 |
| 011 | ODR/8 |
| 100 | ODR/16 |
| 101 | ODR/32 |
| 110 | ODR/64 |
| 111 | Accelerometer not batched in FIFO |

8.46 TAP_CFG0 (6Fh)

R/W

Table 114. TAP_CFG0 register

| | | | | | | | |
|-------|-------|-----------|-----------|-----------|-----------|-----------|---|
| AXIS1 | AXIS0 | INVERT_T4 | INVERT_T3 | INVERT_T2 | INVERT_T1 | INVERT_T0 | - |
|-------|-------|-----------|-----------|-----------|-----------|-----------|---|

Table 115. TAP_CFG0 register description

| | |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AXIS[1:0] | Selection of axis for tap event research: (00: no axis (default); 01: X-axis; 10: Y-axis; 11: Z-axis) |
| INVERT_T[4:0] | These bits enable the search of the inverted peak by selecting the maximum number of samples between the first and second (inverted) peak in tap detection. (0: disabled) 1 LSB = 1 sample (maximum 31 samples) |

8.47 TAP_CFG1 (70h)

R/W

Table 116. TAP_CFG1 register

| | | | | | | | |
|----------------|----------------|----------------|----------------|---------------|---------------|---------------|---------------|
| PRE_STILL_THS3 | PRE_STILL_THS2 | PRE_STILL_THS1 | PRE_STILL_THS0 | POST_STILL_T3 | POST_STILL_T2 | POST_STILL_T1 | POST_STILL_T0 |
|----------------|----------------|----------------|----------------|---------------|---------------|---------------|---------------|

Table 117. TAP_CFG1 register description

| | |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PRE_STILL_THS[3:0] ⁽¹⁾ | Threshold for stationary condition before shock. 1 LSB = 62.5 mg (maximum 937.5 mg) |
| POST_STILL_T[3:0] | Number of samples during stationary condition after shock and wait phases. These bits are used together with POST_STILL_T[5:4] in register TAP_CFG2 (71h) . 1 LSB = 4 samples (maximum 252 samples). |

1. The PRE_STILL_THS[3:0] field must be set to a value greater than 0.

8.48 TAP_CFG2 (71h)

R/W

Table 118. TAP_CFG2 register

| | | | | | | | |
|---------------|---------------|---------|---------|---------|---------|---------|---------|
| POST_STILL_T5 | POST_STILL_T4 | WAIT_T5 | WAIT_T4 | WAIT_T3 | WAIT_T2 | WAIT_T1 | WAIT_T0 |
|---------------|---------------|---------|---------|---------|---------|---------|---------|

Table 119. TAP_CFG2 register description

| | |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST_STILL_T[5:4] | Number of samples during stationary condition after shock and wait phases. These bits are used together with POST_STILL_T[3:0] in register TAP_CFG1 (70h). 1 LSB = 4 samples (maximum 252 samples). |
| WAIT_T[5:0] | These bits program the number of samples to wait for the shock to finish. 1 LSB = 2 samples (maximum 126 samples). |

8.49 TAP_CFG3 (72h)

R/W

Table 120. TAP_CFG3 register

| | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|------------|------------|------------|------------|
| POST_STILL_THS3 | POST_STILL_THS2 | POST_STILL_THS1 | POST_STILL_THS0 | LATENCY_T3 | LATENCY_T2 | LATENCY_T1 | LATENCY_T0 |
|-----------------|-----------------|-----------------|-----------------|------------|------------|------------|------------|

Table 121. TAP_CFG3 register description

| | |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST_STILL_THS[3:0] ⁽¹⁾ | Threshold for stationary condition after shock and wait phases. 1 LSB = 62.5 mg (maximum 937.5 mg) |
| LATENCY_T[3:0] | Maximum number of samples between consecutive taps event to detect double or triple tap. The default value of these bits is 0000b which corresponds to 16 samples. If the LATENCY_T[3:0] bits are set to a different value, 1LSB corresponds to 32 samples (maximum 480 samples). |

1. The POST_STILL_THS[3:0] field must be set to a value greater than 0.

8.50 TAP_CFG4 (73h)

R/W

Table 122. TAP_CFG4 register

| | | | | | | | |
|------------------|---|-----------|-----------|-----------|-----------|-----------|-----------|
| WAIT_END_LATENCY | 0 | PEAK_THS5 | PEAK_THS4 | PEAK_THS3 | PEAK_THS2 | PEAK_THS1 | PEAK_THS0 |
|------------------|---|-----------|-----------|-----------|-----------|-----------|-----------|

Table 123. TAP_CFG4 register description

| | |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WAIT_END_LATENCY | This bit enables the feature to wait for the end of the latency window to exclusively determine if the event is a single, double or triple tap. (0: tap event flag is raised immediately for every detected tap; 1: in case of consecutive taps, only the flag for the highest level of tap is raised. The tap event flag is raised immediately if the highest level of tap enabled in TAP_CFG5 (74h) (single, double or triple) is reached, otherwise it is raised at the end of the latency window if no additional taps are detected within the window.) |
| PEAK_THS[5:0] | Threshold for peak detection. 1 LSB = 62.5 mg (maximum 3937.5 mg) |

8.51 TAP_CFG5 (74h)

R/W

Table 124. TAP_CFG5 register

| | | | | | | | |
|-------------------|-------------------|-------------------|----------------|----------------|----------------|----------------|----------------|
| TRIPLE_ TAP_EN | DOUBLE_ TAP_EN | SINGLE_ TAP_EN | REBOUND_ T4 | REBOUND_ T3 | REBOUND_ T2 | REBOUND_ T1 | REBOUND_ T0 |
|-------------------|-------------------|-------------------|----------------|----------------|----------------|----------------|----------------|

Table 125. TAP_CFG5 register description

| | |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TRIPLE_TAP_EN | This bit enables the triple-tap event. Default value: 0 (0: disabled; 1: enabled) |
| DOUBLE_TAP_EN | This bit enables the double-tap event. Default value: 0 (0: disabled; 1: enabled) |
| SINGLE_TAP_EN | This bit enables the single-tap event. Default value: 0 (0: disabled; 1: enabled) |
| REBOUND_T[4:0] | These bits program the number of samples to wait for the rebound to finish. The default value of these bits is 00000b which means that the rebound logic is disabled. If the REBOUND_T[4:0] bits are set to a different value, 1LSB corresponds to 2 samples (maximum 62 samples). |

8.52 TAP_CFG6 (75h)

R/W

Table 126. TAP_CFG6 register

| | | | | | | | |
|-------------------|-------------------|-------------------|-------------------|------------------|------------------|------------------|------------------|
| PRE_STILL_ ST3 | PRE_STILL_ ST2 | PRE_STILL_ ST1 | PRE_STILL_ ST0 | PRE_STILL_ N3 | PRE_STILL_ N2 | PRE_STILL_ N1 | PRE_STILL_ N0 |
|-------------------|-------------------|-------------------|-------------------|------------------|------------------|------------------|------------------|

Table 127. TAP_CFG6 register description

| | |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PRE_STILL_ST[3:0] | Selection of starting sample for stationary condition before shock (from the oldest sample in a buffer of 14 samples). 1 LSB = 1 sample (0: 1 st sample, 13: 14 th sample, maximum value is 13) |
| PRE_STILL_N[3:0] | Selection of number of samples for stationary condition before shock. 1 LSB = 1 sample (maximum 14 samples). If this field is set to 0, the stationary condition before shock is disabled. |

8.53 **TIMESTAMP3 (7Ah), TIMESTAMP2 (7Bh), TIMESTAMP1 (7Ch), and TIMESTAMP0 (7Dh)**

R

Table 128. TIMESTAMP output registers

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| D31 | D30 | D29 | D28 | D27 | D26 | D25 | D24 |
| D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 |
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Table 129. TIMESTAMP output register description

| | |
|---------|---------------------------------------------------------|
| D[31:0] | Timestamp output registers: 1LSB = 10 μ s (typical) |
|---------|---------------------------------------------------------|

9 Embedded functions register mapping

The table given below provides a list of the registers for the embedded functions available in the device and the corresponding addresses. Embedded functions registers are accessible when the EMB_FUNC_REG_ACCESS bit is set to 1 in the FUNC_CFG_ACCESS (3Fh) register and the EMB_FUNC_EN bit is set to 1 in the CTRL4 (13h) register.

Table 130. Register address map - embedded functions

| Name | Type | Register address | | Default | Comment |
|----------------------|------|------------------|----------|----------|---------|
| | | Hex | Binary | | |
| PAGE_SEL | R/W | 02 | 0000010 | 00000001 | |
| RESERVED | - | 03 | | | |
| EMB_FUNC_EN_A | R/W | 04 | 00000100 | 00000000 | |
| EMB_FUNC_EN_B | R/W | 05 | 00000101 | 00000000 | |
| EMB_FUNC_EXEC_STATUS | R | 07 | 00000111 | output | |
| PAGE_ADDRESS | R/W | 08 | 00001000 | 00000000 | |
| PAGE_VALUE | R/W | 09 | 00001001 | 00000000 | |
| EMB_FUNC_INT1 | R/W | 0A | 00001010 | 00000000 | |
| FSM_INT1 | R/W | 0B | 00001011 | 00000000 | |
| RESERVED | - | 0C | | | |
| MLC_INT1 | R/W | 0D | 00001101 | 00000000 | |
| EMB_FUNC_INT2 | R/W | 0E | 00001110 | 00000000 | |
| FSM_INT2 | R/W | 0F | 00001111 | 00000000 | |
| RESERVED | - | 10 | | | |
| MLC_INT2 | R/W | 11 | 00010001 | 00000000 | |
| EMB_FUNC_STATUS | R | 12 | 00010010 | output | |
| FSM_STATUS | R | 13 | 00010011 | output | |
| RESERVED | - | 14 | | | |
| MLC_STATUS | R | 15 | 00010101 | output | |
| PAGE_RW | R/W | 17 | 00010111 | 00000000 | |
| EMB_FUNC_FIFO_EN | R/W | 18 | 00011000 | | |
| RESERVED | - | 19 | | | |
| FSM_ENABLE | R/W | 1A | 00011010 | 00000000 | |
| RESERVED | - | 1B | | | |
| FSM_LONG_COUNTER_L | R/W | 1C | 00011100 | 00000000 | |
| FSM_LONG_COUNTER_H | R/W | 1D | 00011101 | 00000000 | |
| RESERVED | - | 1E | | | |
| INT_ACK_MASK | R/W | 1F | 00011111 | 00000000 | |
| FSM_OUTS1 | R | 20 | 00100000 | output | |
| FSM_OUTS2 | R | 21 | 00100001 | output | |
| FSM_OUTS3 | R | 22 | 00100010 | output | |
| FSM_OUTS4 | R | 23 | 00100011 | output | |
| FSM_OUTS5 | R | 24 | 00100100 | output | |

| Name | Type | Register address | | Default | Comment |
|-----------------|------|------------------|----------|----------|---------|
| | | Hex | Binary | | |
| FSM_OUTS6 | R | 25 | 00100101 | output | |
| FSM_OUTS7 | R | 26 | 00100110 | output | |
| FSM_OUTS8 | R | 27 | 00100111 | output | |
| STEP_COUNTER_L | R | 28 | 00101000 | output | |
| STEP_COUNTER_H | R | 29 | 00101001 | output | |
| EMB_FUNC_SRC | R | 2A | 00101010 | output | |
| RESERVED | - | 2B | | | |
| EMB_FUNC_INIT_A | R/W | 2C | 00101100 | 00000000 | |
| EMB_FUNC_INIT_B | R/W | 2D | 00101101 | 00000000 | |
| RESERVED | - | 2E-33 | | | |
| MLC1_SRC | R | 34 | 00110100 | output | |
| MLC2_SRC | R | 35 | 00110101 | output | |
| MLC3_SRC | R | 36 | 00110110 | output | |
| MLC4_SRC | R | 37 | 00110111 | output | |
| FSM_ODR | R/W | 39 | 00111001 | 01001000 | |
| MLC_ODR | R/W | 3A | 00111010 | 00010001 | |

Reserved registers must not be changed. Writing to those registers may cause permanent damage to the device. The content of the registers that are loaded at boot should not be changed. They contain the factory calibration values. Their content is automatically restored when the device is powered up.

10 Embedded functions register description

10.1 PAGE_SEL (02h)

Enable advanced features dedicated page (R/W)

Table 131. PAGE_SEL register

| | | | | | | | |
|-----------|-----------|-----------|-----------|------------------|------------------|------------------|------------------|
| PAGE_SEL3 | PAGE_SEL2 | PAGE_SEL1 | PAGE_SEL0 | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 1 ⁽²⁾ |
|-----------|-----------|-----------|-----------|------------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.
2. This bit must be set to 1 for the correct operation of the device.

Table 132. PAGE_SEL register description

| | |
|---------------|-------------------------------------------------------------------|
| PAGE_SEL[3:0] | Selects the advanced features dedicated page. Default value: 0000 |
|---------------|-------------------------------------------------------------------|

10.2 EMB_FUNC_EN_A (04h)

Enable embedded functions register (R/W)

Table 133. EMB_FUNC_EN_A register

| | | | | | | | |
|-------------------|------------------|----------------|---------|---------|------------------|------------------|------------------|
| MLC_BEFORE_FSM_EN | 0 ⁽¹⁾ | SIGN_MOTION_EN | TILT_EN | PEDO_EN | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|-------------------|------------------|----------------|---------|---------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 134. EMB_FUNC_EN_A register description

| | |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MLC_BEFORE_FSM_EN ⁽¹⁾ | Enables machine learning core function. When the machine learning core is enabled by setting this bit to 1, the MLC algorithms are executed before the FSM programs. Default value: 0 (0: machine learning core function disabled; 1: machine learning core function enabled and executed before FSM programs) |
| SIGN_MOTION_EN | Enables significant motion detection function. Default value: 0 (0: significant motion detection function disabled; 1: significant motion detection function enabled) |
| TILT_EN | Enables tilt calculation. Default value: 0 (0: tilt algorithm disabled; 1: tilt algorithm enabled) |
| PEDO_EN | Enables pedometer algorithm. Default value: 0 (0: pedometer algorithm disabled; 1: pedometer algorithm enabled) |

1. The MLC_EN bit in the *EMB_FUNC_EN_B (05h)* register must be set to 0 when using this bit.

DS14071 - Rev 2

page 64/102

10.3 EMB_FUNC_EN_B (05h)

Enable embedded functions register (R/W)

Table 135. EMB_FUNC_EN_B register

| | | | | | | | |
|------------------|------------------|------------------|--------|------------------|------------------|------------------|--------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | MLC_EN | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | FSM_EN |
|------------------|------------------|------------------|--------|------------------|------------------|------------------|--------|

1. This bit must be set to 0 for the correct operation of the device.

Table 136. EMB_FUNC_EN_B register description

| | |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MLC_EN ⁽¹⁾ | Enables machine learning core function. When the machine learning core is enabled by setting this bit to 1, the MLC algorithms are executed after executing the FSM programs. Default value: 0 (0: machine learning core function disabled; 1: machine learning core function enabled and executed after FSM programs) |
| FSM_EN | Enables finite state machine (FSM) function. Default value: 0 (0: FSM function disabled; 1: FSM function enabled) |

1. The MLC_BEFORE_FSM_EN bit in the EMB_FUNC_EN_A (04h) register must be set to 0 when using this bit.

10.4 EMB_FUNC_EXEC_STATUS (07h)

Embedded functions execution status register (R)

Table 137. EMB_FUNC_EXEC_STATUS register

| | | | | | | | |
|---|---|---|---|---|---|-------------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 0 | EMB_FUNC_EXEC_OVR | EMB_FUNC_ENDOP |
|---|---|---|---|---|---|-------------------|----------------|

Table 138. EMB_FUNC_EXEC_STATUS register description

| | |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EMB_FUNC_EXEC_OVR | This bit is set to 1 when the execution of the embedded functions program exceeds maximum time (new data are generated before the end of the algorithms). Default value: 0 |
| EMB_FUNC_ENDOP | When this bit is set to 1, no embedded function is running. Default value: 0 |

10.5 PAGE_ADDRESS (08h)

Page address register (R/W)

Table 139. PAGE_ADDRESS register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| PAGE_ADDR7 | PAGE_ADDR6 | PAGE_ADDR5 | PAGE_ADDR4 | PAGE_ADDR3 | PAGE_ADDR2 | PAGE_ADDR1 | PAGE_ADDR0 |
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 140. PAGE_ADDRESS register description

| | |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAGE_ADDR[7:0] | After setting the bit PAGE_WRITE / PAGE_READ in register PAGE_RW (17h), this register is used to set the address of the register to be written/read in the advanced features page selected through the bits PAGE_SEL[3:0] in register PAGE_SEL (02h). |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10.6 PAGE_VALUE (09h)

Page value register (R/W)

Table 141. PAGE_VALUE register

| | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| PAGE_VALUE7 | PAGE_VALUE6 | PAGE_VALUE5 | PAGE_VALUE4 | PAGE_VALUE3 | PAGE_VALUE2 | PAGE_VALUE1 | PAGE_VALUE0 |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|

Table 142. PAGE_VALUE register description

| | |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAGE_VALUE[7:0] | These bits are used to write (if the bit PAGE_WRITE = 1 in register PAGE_RW (17h)) or read (if the bit PAGE_READ = 1 in register PAGE_RW (17h)) the data at the address PAGE_ADDR[7:0] of the selected advanced features page. |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10.7 EMB_FUNC_INT1 (0Ah)

INT1 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT1. The pin's output supplies the OR combination of the selected signals.

Table 143. EMB_FUNC_INT1 register

| | | | | | | | |
|-------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|
| INT1_FSM_LC | 0 ⁽¹⁾ | INT1_SIG_MOT | INT1_TILT | INT1_STEP_DET | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|-------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 144. EMB_FUNC_INT1 register description

| | |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| INT1_FSM_LC ⁽¹⁾ | Enables routing FSM long counter timeout interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_SIG_MOT ⁽¹⁾ | Enables routing significant motion event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_TILT ⁽¹⁾ | Enables routing tilt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_STEP_DETECTOR ⁽¹⁾ | Enables routing pedometer step recognition event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |

1. This bit is active if the INT1_EMB_FUNC bit of MD1_CFG (1Fh) is set to 1.

10.8 FSM_INT1 (0Bh)

INT1 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT1. The pin's output supplies the OR combination of the selected signals.

Table 145. FSM_INT1 register

| | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| INT1_FSM8 | INT1_FSM7 | INT1_FSM6 | INT1_FSM5 | INT1_FSM4 | INT1_FSM3 | INT1_FSM2 | INT1_FSM1 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|

Table 146. FSM_INT1 register description

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| INT1_FSM8 ⁽¹⁾ | Enables routing FSM8 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM7 ⁽¹⁾ | Enables routing FSM7 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM6 ⁽¹⁾ | Enables routing FSM6 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM5 ⁽¹⁾ | Enables routing FSM5 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM4 ⁽¹⁾ | Enables routing FSM4 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM3 ⁽¹⁾ | Enables routing FSM3 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM2 ⁽¹⁾ | Enables routing FSM2 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_FSM1 ⁽¹⁾ | Enables routing FSM1 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |

1. This bit is active if the INT1_EMB_FUNC bit of MD1_CFG (1Fh) is set to 1.

10.9 MLC_INT1 (0Dh)

INT1 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT1. The pin's output supplies the OR combination of the selected signals.

Table 147. MLC_INT1 register

| | | | | | | | |
|------------------|------------------|------------------|------------------|-----------|-----------|-----------|-----------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | INT1_MLC4 | INT1_MLC3 | INT1_MLC2 | INT1_MLC1 |
|------------------|------------------|------------------|------------------|-----------|-----------|-----------|-----------|

1. This bit must be set to 0 for the correct operation of the device.

Table 148. MLC_INT1 register description

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| INT1_MLC4 ⁽¹⁾ | Enables routing MLC4 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_MLC3 ⁽¹⁾ | Enables routing MLC3 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_MLC2 ⁽¹⁾ | Enables routing MLC2 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |
| INT1_MLC1 ⁽¹⁾ | Enables routing MLC1 interrupt event to INT1. Default value: 0 (0: routing to INT1 disabled; 1: routing to INT1 enabled) |

1. This bit is active if the INT1_EMB_FUNC bit of MD1_CFG (1Fh) is set to 1.

10.10 EMB_FUNC_INT2 (0Eh)

INT2 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT2. The pin's output supplies the OR combination of the selected signals.

Table 149. EMB_FUNC_INT2 register

| | | | | | | | |
|-------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|
| INT2_FSM_LC | 0 ⁽¹⁾ | INT2_SIG_MOT | INT2_TILT | INT2_STEP_DET | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|-------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 150. EMB_FUNC_INT2 register description

| | |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| INT2_FSM_LC ⁽¹⁾ | Enables routing FSM long counter timeout interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_SIG_MOT ⁽¹⁾ | Enables routing significant motion event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_TILT ⁽¹⁾ | Enables routing tilt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_STEP_DETECT ⁽¹⁾ | Enables routing pedometer step recognition event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |

1. This bit is active if the INT2_EMB_FUNC bit of MD2_CFG (20h) is set to 1.

10.11 FSM_INT2 (0Fh)

INT2 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT2. The pin's output supplies the OR combination of the selected signals.

Table 151. FSM_INT2 register

| | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| INT2_FSM8 | INT2_FSM7 | INT2_FSM6 | INT2_FSM5 | INT2_FSM4 | INT2_FSM3 | INT2_FSM2 | INT2_FSM1 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|

Table 152. FSM_INT2 register description

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| INT2_FSM8 ⁽¹⁾ | Enables routing FSM8 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM7 ⁽¹⁾ | Enables routing FSM7 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM6 ⁽¹⁾ | Enables routing FSM6 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM5 ⁽¹⁾ | Enables routing FSM5 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM4 ⁽¹⁾ | Enables routing FSM4 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM3 ⁽¹⁾ | Enables routing FSM3 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM2 ⁽¹⁾ | Enables routing FSM2 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_FSM1 ⁽¹⁾ | Enables routing FSM1 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |

1. This bit is active if the INT2_EMB_FUNC bit of MD2_CFG (20h) is set to 1.

10.12 MLC_INT2 (11h)

INT2 pin control register (R/W)

Each bit in this register enables a signal to be carried over INT2. The pin's output supplies the OR combination of the selected signals.

Table 153. MLC_INT2 register

| | | | | | | | |
|------------------|------------------|------------------|------------------|-----------|-----------|-----------|-----------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | INT2_MLC4 | INT2_MLC3 | INT2_MLC2 | INT2_MLC1 |
|------------------|------------------|------------------|------------------|-----------|-----------|-----------|-----------|

1. This bit must be set to 0 for the correct operation of the device.

Table 154. MLC_INT2 register description

| | |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| INT2_MLC4 ⁽¹⁾ | Enables routing MLC4 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_MLC3 ⁽¹⁾ | Enables routing MLC3 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_MLC2 ⁽¹⁾ | Enables routing MLC2 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |
| INT2_MLC1 ⁽¹⁾ | Enables routing MLC1 interrupt event to INT2. Default value: 0 (0: routing to INT2 disabled; 1: routing to INT2 enabled) |

1. This bit is active if the INT2_EMB_FUNC bit of MD2_CFG (20h) is set to 1.

10.13 EMB_FUNC_STATUS (12h)

Embedded function status register (R)

Table 155. EMB_FUNC_STATUS register

| | | | | | | | |
|-----------|---|-----------|---------|-------------|---|---|---|
| IS_FSM_LC | 0 | IS_SIGMOT | IS_TILT | IS_STEP_DET | 0 | 0 | 0 |
|-----------|---|-----------|---------|-------------|---|---|---|

Table 156. EMB_FUNC_STATUS register description

| | |
|-------------|----------------------------------------------------------------------------------------------------------------|
| IS_FSM_LC | Interrupt status bit for FSM long counter timeout interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_SIGMOT | Interrupt status bit for significant motion detection (1: interrupt detected; 0: no interrupt) |
| IS_TILT | Interrupt status bit for tilt detection (1: interrupt detected; 0: no interrupt) |
| IS_STEP_DET | Interrupt status bit for step detection (1: interrupt detected; 0: no interrupt) |

10.14 FSM_STATUS (13h)

Finite state machine status register (R)

Table 157. FSM_STATUS register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| IS_FSM8 | IS_FSM7 | IS_FSM6 | IS_FSM5 | IS_FSM4 | IS_FSM3 | IS_FSM2 | IS_FSM1 |
|---------|---------|---------|---------|---------|---------|---------|---------|

Table 158. FSM_STATUS register description

| | |
|---------|-----------------------------------------------------------------------------------------|
| IS_FSM8 | Interrupt status bit for FSM8 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM7 | Interrupt status bit for FSM7 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM6 | Interrupt status bit for FSM6 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM5 | Interrupt status bit for FSM5 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM4 | Interrupt status bit for FSM4 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM3 | Interrupt status bit for FSM3 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM2 | Interrupt status bit for FSM2 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_FSM1 | Interrupt status bit for FSM1 interrupt event. (1: interrupt detected; 0: no interrupt) |

10.15 MLC_STATUS (15h)

Machine learning core status register (R)

Table 159. MLC_STATUS register

| | | | | | | | |
|---|---|---|---|---------|---------|--------|---------|
| 0 | 0 | 0 | 0 | IS_MLC4 | IS_MLC3 | IS_MLC | IS_MLC1 |
|---|---|---|---|---------|---------|--------|---------|

Table 160. MLC_STATUS register description

| | |
|---------|-----------------------------------------------------------------------------------------|
| IS_MLC4 | Interrupt status bit for MLC4 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC3 | Interrupt status bit for MLC3 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC2 | Interrupt status bit for MLC2 interrupt event. (1: interrupt detected; 0: no interrupt) |
| IS_MLC1 | Interrupt status bit for MLC1 interrupt event. (1: interrupt detected; 0: no interrupt) |

10.16 PAGE_RW (17h)

Enable read and write mode of advanced features dedicated page (R/W)

Table 161. PAGE_RW register

| | | | | | | | |
|--------------|------------|-----------|------------------|------------------|------------------|------------------|------------------|
| EMB_FUNC_LIR | PAGE_WRITE | PAGE_READ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|--------------|------------|-----------|------------------|------------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 162. PAGE_RW register description

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EMB_FUNC_LIR | Latched interrupt mode for embedded functions. Default value: 0 (0: embedded functions interrupt request not latched; 1: embedded functions interrupt request latched) |
| PAGE_WRITE | Enable writes to the selected advanced features dedicated page. ⁽¹⁾ Default value: 0 (1: enable; 0: disable) |
| PAGE_READ | Enable reads from the selected advanced features dedicated page. ⁽¹⁾ Default value: 0 (1: enable; 0: disable) |

1. Page selected by PAGE_SEL[3:0] in PAGE_SEL (02h) register.

10.17 EMB_FUNC_FIFO_EN (18h)

Embedded functions FIFO configuration register (R/W)

Table 163. EMB_FUNC_FIFO_EN register

| | | | | | | | |
|------------------|------------------|------------------|------------------|-------------|----------------------------|-------------|----------------------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | FSM_FIFO_EN | MLC_FILTER_FEATURE_FIFO_EN | MLC_FIFO_EN | STEP_COUNTER_FIFO_EN |
|------------------|------------------|------------------|------------------|-------------|----------------------------|-------------|----------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 164. EMB_FUNC_FIFO_EN register description

| | |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| FSM_FIFO_EN | Enables batching in FIFO buffer of finite state machine results. Default value: 0 (0: disabled; 1: enabled) |
| MLC_FILTER_FEATURE_FIFO_EN | Enables batching in FIFO buffer of machine learning core filters and features. Default value: 0 (0: disabled; 1: enabled) |
| MLC_FIFO_EN | Enables batching in FIFO buffer of machine learning core results. Default value: 0 (0: disabled; 1: enabled) |
| STEP_COUNTER_FIFO_EN | Enables batching in FIFO buffer of step counter value. Default value: 0 (0: disabled; 1: enabled) |

10.18 FSM_ENABLE (1Ah)

Enable FSM register (R/W)

Table 165. FSM_ENABLE register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| FSM8_EN | FSM7_EN | FSM6_EN | FSM5_EN | FSM4_EN | FSM3_EN | FSM2_EN | FSM1_EN |
|---------|---------|---------|---------|---------|---------|---------|---------|

Table 166. FSM_ENABLE register description

| | |
|---------|--------------------------------------------------------------------|
| FSM8_EN | Enables FSM8. Default value: 0 (0: FSM8 disabled; 1: FSM8 enabled) |
| FSM7_EN | Enables FSM7. Default value: 0 (0: FSM7 disabled; 1: FSM7 enabled) |
| FSM6_EN | Enables FSM6. Default value: 0 (0: FSM6 disabled; 1: FSM6 enabled) |
| FSM5_EN | Enables FSM5. Default value: 0 (0: FSM5 disabled; 1: FSM5 enabled) |
| FSM4_EN | Enables FSM4. Default value: 0 (0: FSM4 disabled; 1: FSM4 enabled) |
| FSM3_EN | Enables FSM3. Default value: 0 (0: FSM3 disabled; 1: FSM3 enabled) |
| FSM2_EN | Enables FSM2. Default value: 0 (0: FSM2 disabled; 1: FSM2 enabled) |
| FSM1_EN | Enables FSM1. Default value: 0 (0: FSM1 disabled; 1: FSM1 enabled) |

10.19 FSM_LONG_COUNTER_L (1Ch) and FSM_LONG_COUNTER_H (1Dh)

FSM long counter status register (R/W)

Long counter value is an unsigned integer value (16-bit format).

Table 167. FSM_LONG_COUNTER_L register

| | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|
| FSM_LC_7 | FSM_LC_6 | FSM_LC_5 | FSM_LC_4 | FSM_LC_3 | FSM_LC_2 | FSM_LC_1 | FSM_LC_0 |
|----------|----------|----------|----------|----------|----------|----------|----------|

Table 168. FSM_LONG_COUNTER_L register description

| | |
|--------------|--------------------------------------------------------------|
| FSM_LC_[7:0] | Long counter current value (LSbyte). Default value: 00000000 |
|--------------|--------------------------------------------------------------|

Table 169. FSM_LONG_COUNTER_H register

| | | | | | | | |
|---|-----------|-----------|-----------|-----------|-----------|----------|----------|
| - | FSM_LC_14 | FSM_LC_13 | FSM_LC_12 | FSM_LC_11 | FSM_LC_10 | FSM_LC_9 | FSM_LC_8 |
|---|-----------|-----------|-----------|-----------|-----------|----------|----------|

Table 170. FSM_LONG_COUNTER_H register description

| | |
|---------------|--------------------------------------------------------------|
| FSM_LC_[14:8] | Long counter current value (MSbyte). Default value: 00000000 |
|---------------|--------------------------------------------------------------|

10.20 INT_ACK_MASK (1Fh)

Reset status register (R/W)

Table 171. INT_ACK_MASK register

| IACK_MASK7 | IACK_MASK6 | IACK_MASK5 | IACK_MASK4 | IACK_MASK3 | IACK_MASK2 | IACK_MASK1 | IACK_MASK0 |
|------------|------------|------------|------------|------------|------------|------------|------------|
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 172. INT_ACK_MASK register description

| | |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IACK_MASK7 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 7 of the status register is not reset. When this bit is set to 0, bit 7 of the status register is reset. Default value: 0 |
| IACK_MASK6 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 6 of the status register is not reset. When this bit is set to 0, bit 6 of the status register is reset. Default value: 0 |
| IACK_MASK5 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 5 of the status register is not reset. When this bit is set to 0, bit 5 of the status register is reset. Default value: 0 |
| IACK_MASK4 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 4 of the status register is not reset. When this bit is set to 0, bit 4 of the status register is reset. Default value: 0 |
| IACK_MASK3 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 3 of the status register is not reset. When this bit is set to 0, bit 3 of the status register is reset. Default value: 0 |
| IACK_MASK2 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 2 of the status register is not reset. When this bit is set to 0, bit 2 of the status register is reset. Default value: 0 |
| IACK_MASK1 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 1 of the status register is not reset. When this bit is set to 0, bit 1 of the status register is reset. Default value: 0 |
| IACK_MASK0 | If set to 1, when reading the EMB_FUNC_STATUS (12h) / EMB_FUNC_STATUS_MAINPAGE (34h) , FSM_STATUS (13h) / FSM_STATUS_MAINPAGE (35h) and MLC_STATUS (15h) / MLC_STATUS_MAINPAGE (36h) registers in latched mode (when the EMB_FUNC_LIR bit is set to 1 in the PAGE_RW (17h) register), bit 0 of the status register is not reset. When this bit is set to 0, bit 0 of the status register is reset. Default value: 0 |

10.21 FSM_OUTS1 (20h)

FSM1 output register (R)

Table 173. FSM_OUTS1 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 174. FSM_OUTS1 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM1 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM1 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM1 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM1 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM1 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM1 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM1 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM1 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.22 FSM_OUTS2 (21h)

FSM2 output register (R)

Table 175. FSM_OUTS2 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 176. FSM_OUTS2 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM2 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM2 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM2 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM2 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM2 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM2 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM2 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM2 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.23 FSM_OUTS3 (22h)

FSM3 output register (R)

Table 177. FSM_OUTS3 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 178. FSM_OUTS3 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM3 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM3 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM3 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM3 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM3 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM3 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM3 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM3 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.24 FSM_OUTS4 (23h)

FSM4 output register (R)

Table 179. FSM_OUTS4 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 180. FSM_OUTS4 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM4 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM4 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM4 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM4 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM4 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM4 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM4 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM4 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.25 FSM_OUTS5 (24h)

FSM5 output register (R)

Table 181. FSM_OUTS5 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 182. FSM_OUTS5 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM5 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM5 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM5 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM5 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM5 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM5 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM5 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM5 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.26 FSM_OUTS6 (25h)

FSM6 output register (R)

Table 183. FSM_OUTS6 register

| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 184. FSM_OUTS6 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM6 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM6 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM6 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM6 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM6 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM6 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM6 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM6 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.27 FSM_OUTS7 (26h)

FSM7 output register (R)

Table 185. FSM_OUTS7 register

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 186. FSM_OUTS7 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM7 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM7 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM7 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM7 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM7 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM7 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM7 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM7 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.28 FSM_OUTS8 (27h)

FSM8 output register (R)

Table 187. FSM_OUTS8 register

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| P_X | N_X | P_Y | N_Y | P_Z | N_Z | P_V | N_V |
|-----|-----|-----|-----|-----|-----|-----|-----|

Table 188. FSM_OUTS8 register description

| | |
|-----|------------------------------------------------------------------------------------------------|
| P_X | FSM8 output: positive event detected on the X-axis. (0: event not detected; 1: event detected) |
| N_X | FSM8 output: negative event detected on the X-axis. (0: event not detected; 1: event detected) |
| P_Y | FSM8 output: positive event detected on the Y-axis. (0: event not detected; 1: event detected) |
| N_Y | FSM8 output: negative event detected on the Y-axis. (0: event not detected; 1: event detected) |
| P_Z | FSM8 output: positive event detected on the Z-axis. (0: event not detected; 1: event detected) |
| N_Z | FSM8 output: negative event detected on the Z-axis. (0: event not detected; 1: event detected) |
| P_V | FSM8 output: positive event detected on the vector. (0: event not detected; 1: event detected) |
| N_V | FSM8 output: negative event detected on the vector. (0: event not detected; 1: event detected) |

10.29 STEP_COUNTER_L (28h) and STEP_COUNTER_H (29h)

Step counter output register (R)

Table 189. STEP_COUNTER_L register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| STEP_7 | STEP_6 | STEP_5 | STEP_4 | STEP_3 | STEP_2 | STEP_1 | STEP_0 |
|--------|--------|--------|--------|--------|--------|--------|--------|

Table 190. STEP_COUNTER_L register description

| | |
|------------|------------------------------|
| STEP_[7:0] | Step counter output (LSbyte) |
|------------|------------------------------|

Table 191. STEP_COUNTER_H register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|--------|--------|
| STEP_15 | STEP_14 | STEP_13 | STEP_12 | STEP_11 | STEP_10 | STEP_9 | STEP_8 |
|---------|---------|---------|---------|---------|---------|--------|--------|

Table 192. STEP_COUNTER_H register description

| | |
|-------------|------------------------------|
| STEP_[15:8] | Step counter output (MSbyte) |
|-------------|------------------------------|

10.30 EMB_FUNC_SRC (2Ah)

Embedded function source register (R/W)

Table 193. EMB_FUNC_SRC register

| | | | | | | | |
|---------------|------------------|---------------|---------------------|---------------|----------------------|------------------|------------------|
| PEDO_RST_STEP | 0 ⁽¹⁾ | STEP_DETECTED | STEP_COUNT_DELTA_IA | STEP_OVERFLOW | STEP_COUNTER_BIT_SET | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|---------------|------------------|---------------|---------------------|---------------|----------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 194. EMB_FUNC_SRC register description

| | |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PEDO_RST_STEP | Reset pedometer step counter. Read/write bit. (0: disabled; 1: enabled) |
| STEP_DETECTED | Step detector event detection status. Read-only bit. (0: step detection event not detected; 1: step detection event detected) |
| STEP_COUNT_DELTA_IA | Pedometer step recognition on delta time status. Read-only bit. (0: no step recognized during delta time; 1: at least one step recognized during delta time) |
| STEP_OVERFLOW | Step counter overflow status. Read-only bit. (0: step counter value < 2 ¹⁶ ; 1: step counter value reached 2 ¹⁶) |
| STEP_COUNTER_BIT_SET | This bit is equal to 1 when the step count is increased. If a timer period is programmed in PEDO_SC_DELTAT_L (AAh) and PEDO_SC_DELTAT_H (ABh) embedded advanced features (page 1) registers, this bit is kept to 0. Read-only bit. |

10.31 EMB_FUNC_INIT_A (2Ch)

Embedded functions initialization register (R/W)

Table 195. EMB_FUNC_INIT_A register

| | | | | | | | |
|---------------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|
| MLC_BEFORE_FSM_INIT | 0 ⁽¹⁾ | SIG_MOT_INIT | TILT_INIT | STEP_DET_INIT | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|---------------------|------------------|--------------|-----------|---------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 196. EMB_FUNC_INIT_A register description

| | |
|---------------------|------------------------------------------------------------------------------------------|
| MLC_BEFORE_FSM_INIT | Machine learning core initialization request (MLC executed before FSM). Default value: 0 |
| SIG_MOT_INIT | Significant motion detection algorithm initialization request. Default value: 0 |
| TILT_INIT | Tilt algorithm initialization request. Default value: 0 |
| STEP_DET_INIT | Pedometer step counter/detector algorithm initialization request. Default value: 0 |

10.32 EMB_FUNC_INIT_B (2Dh)

Embedded functions initialization register (R/W)

Table 197. EMB_FUNC_INIT_B register

| | | | | | | | |
|------------------|------------------|------------------|----------|------------------|------------------|------------------|----------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | MLC_INIT | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | FSM_INIT |
|------------------|------------------|------------------|----------|------------------|------------------|------------------|----------|

1. This bit must be set to 0 for the correct operation of the device.

Table 198. EMB_FUNC_INIT_B register description

| | |
|----------|-----------------------------------------------------------------------------------------|
| MLC_INIT | Machine learning core initialization request (MLC executed after FSM). Default value: 0 |
| FSM_INIT | FSM initialization request. Default value: 0 |

10.33 MLC1_SRC (34h)

Machine learning core source register (R)

Table 199. MLC1_SRC register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| MLC1_SRC_7 | MLC1_SRC_6 | MLC1_SRC_5 | MLC1_SRC_4 | MLC1_SRC_3 | MLC1_SRC_2 | MLC1_SRC_1 | MLC1_SRC_0 |
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 200. MLC1_SRC register description

| | |
|----------------|------------------------------------|
| MLC1_SRC_[7:0] | Output value of MLC1 decision tree |
|----------------|------------------------------------|

10.34 MLC2_SRC (35h)

Machine learning core source register (R)

Table 201. MLC2_SRC register

| | | | | | | | |
|------------|------------|------------|------------|------------|-------------|------------|------------|
| MLC2_SRC_7 | MLC2_SRC_6 | MLC2_SRC_5 | MLC2_SRC_4 | MLC2_SRC_3 | MLCS2_SRC_2 | MLC2_SRC_1 | MLC2_SRC_0 |
|------------|------------|------------|------------|------------|-------------|------------|------------|

Table 202. MLC2_SRC register description

| | |
|----------------|------------------------------------|
| MLC2_SRC_[7:0] | Output value of MLC2 decision tree |
|----------------|------------------------------------|

10.35 MLC3_SRC (36h)

Machine learning core source register (R)

Table 203. MLC3_SRC register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| MLC3_SRC_7 | MLC3_SRC_6 | MLC3_SRC_5 | MLC3_SRC_4 | MLC3_SRC_3 | MLC3_SRC_2 | MLC3_SRC_1 | MLC3_SRC_0 |
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 204. MLC3_SRC register description

| | |
|----------------|------------------------------------|
| MLC3_SRC_[7:0] | Output value of MLC3 decision tree |
|----------------|------------------------------------|

10.36 MLC4_SRC (37h)

Machine learning core source register (R)

Table 205. MLC4_SRC register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| MLC4_SRC_7 | MLC4_SRC_6 | MLC4_SRC_5 | MLC4_SRC_4 | MLC4_SRC_3 | MLC4_SRC_2 | MLC4_SRC_1 | MLC4_SRC_0 |
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 206. MLC4_SRC register description

| | |
|----------------|------------------------------------|
| MLC4_SRC_[7:0] | Output value of MLC4 decision tree |
|----------------|------------------------------------|

10.37 FSM_ODR (39h)

Finite state machine output data rate configuration register (R/W)

Table 207. FSM_ODR register

| | | | | | | | |
|------------------|------------------|-----------|-----------|-----------|------------------|------------------|------------------|
| 0 ⁽¹⁾ | 1 ⁽²⁾ | FSM_ODR_2 | FSM_ODR_1 | FSM_ODR_0 | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|------------------|------------------|-----------|-----------|-----------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.
2. This bit must be set to 1 for the correct operation of the device.

Table 208. FSM_ODR register description

| | |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FSM_ODR_[2:0] | Finite state machine ODR configuration: (000: 12.5 Hz; 001: 25 Hz (default); 010: 50 Hz; 011: 100 Hz; 100: 200 Hz; 101: 400 Hz; 110: 800 Hz) |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10.38 MLC_ODR (3Ah)

Machine learning core output data rate configuration register (R/W)

Table 209. MLC_ODR register

| | | | | | | | |
|------------------|-----------|-----------|-----------|------------------|------------------|------------------|------------------|
| 0 ⁽¹⁾ | MLC_ODR_2 | MLC_ODR_1 | MLC_ODR_0 | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 1 ⁽²⁾ |
|------------------|-----------|-----------|-----------|------------------|------------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.
2. This bit must be set to 1 for the correct operation of the device.

Table 210. MLC_ODR register description

| | |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|
| MLC_ODR_[2:0] | Machine learning core ODR configuration: (000: 12.5 Hz; 001: 25 Hz (default); 010: 50 Hz; 011: 100 Hz; 100: 200 Hz) |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|

11 Embedded advanced features pages

The table given below provides a list of the registers for the embedded advanced features page 0. These registers are accessible when PAGE_SEL[3:0] are set to 0000 in PAGE_SEL (02h).

Note: The content of these registers is loaded when the embedded functions are enabled by setting the EMB_FUNC_EN bit to 1 in the CTRL4 (13h) register. The embedded functions must be enabled in order for these registers to become accessible.

Table 211. Register address map - embedded advanced features page 0

| Name | Type | Register address | | Default | Comment |
|---------------------|------|------------------|----------|----------|---------|
| | | Hex | Binary | | |
| FSM_LC_TIMEOUT_L | R/W | 54 | 01010100 | 00000000 | |
| FSM_LC_TIMEOUT_H | R/W | 55 | 01010101 | 00000000 | |
| FSM_PROGRAMS | R/W | 56 | 01010110 | 00000000 | |
| FSM_START_ADD_L | R/W | 58 | 01011000 | 00000000 | |
| FSM_START_ADD_H | R/W | 59 | 01011001 | 00000000 | |
| PEDO_CMD_REG | R/W | 5D | 01011101 | 00000000 | |
| PEDO_DEB_STEPS_CONF | R/W | 5E | 01011110 | 00001010 | |
| PEDO_SC_DELTAT_L | R/W | AA | 10101010 | 00000000 | |
| PEDO_SC_DELTAT_H | R/W | AB | 10101011 | 00000000 | |
| T_SENSITIVITY_L | R/W | B6 | 10110110 | 11010001 | |
| T_SENSITIVITY_H | R/W | B7 | 10110111 | 00011001 | |

Reserved registers must not be changed. Writing to those registers may cause permanent damage to the device. The content of the registers that are loaded at boot should not be changed. They contain the factory calibration values. Their content is automatically restored when the device is powered up.

Write procedure example: write value 06h in register at address 5Eh (PEDO_DEB_STEPS_CONF) in Page 0

1. Write bit EMB_FUNC_REG_ACCESS = 1 in FUNC_CFG_ACCESS (3Fh) // Enable access to embedded functions registers
2. Write bit PAGE_WRITE = 1 in PAGE_RW (17h) register // Select write operation mode
3. Write 0000 in PAGE_SEL[3:0] field of register PAGE_SEL (02h) // Select page 0
4. Write 5Eh in PAGE_ADDRESS (08h) register // Set address
5. Write 06h in PAGE_VALUE (09h) register // Set value to be written
6. Write bit PAGE_WRITE = 0 in PAGE_RW (17h) register // Write operation disabled
7. Write bit EMB_FUNC_REG_ACCESS = 0 in FUNC_CFG_ACCESS (3Fh) // Disable access to embedded functions registers

Read procedure example: read value of register at address 5Eh (PEDO_DEB_STEPS_CONF) in Page 0

1. Write bit EMB_FUNC_REG_ACCESS = 1 in FUNC_CFG_ACCESS (3Fh) // Enable access to embedded functions registers
2. Write bit PAGE_READ = 1 in PAGE_RW (17h) register // Select read operation mode
3. Write 0000 in PAGE_SEL[3:0] field of register PAGE_SEL (02h) // Select page 0
4. Write 5Eh in PAGE_ADDRESS (08h) register // Set address
5. Read value of PAGE_VALUE (09h) register // Get register value
6. Write bit PAGE_READ = 0 in PAGE_RW (17h) register // Read operation disabled
7. Write bit EMB_FUNC_REG_ACCESS = 0 in FUNC_CFG_ACCESS (3Fh) // Disable access to embedded functions registers

Note: Steps 1 and 2 of both procedures are intended to be performed at the beginning of the procedure. Steps 6 and 7 of both procedures are intended to be performed at the end of the procedure. If the procedure involves multiple operations, only steps 3, 4 and 5 must be repeated for each operation. If, in particular, the multiple operations involve consecutive registers, only step 5 can be performed.

12 Embedded advanced features register description

12.1 Page 0 - embedded advanced features registers

12.1.1 FSM_LC_TIMEOUT_L (54h) and FSM_LC_TIMEOUT_H (55h)

FSM long counter timeout register (R/W)

The long counter timeout value is an unsigned integer value (16-bit format). When the long counter value reached this value, the FSM generates an interrupt.

Table 212. FSM_LC_TIMEOUT_L register

| | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| FSM_LC_TIMEOUT7 | FSM_LC_TIMEOUT6 | FSM_LC_TIMEOUT5 | FSM_LC_TIMEOUT4 | FSM_LC_TIMEOUT3 | FSM_LC_TIMEOUT2 | FSM_LC_TIMEOUT1 | FSM_LC_TIMEOUT0 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|

Table 213. FSM_LC_TIMEOUT_L register description

| | |
|---------------------|------------------------------------------------------------------|
| FSM_LC_TIMEOUT[7:0] | FSM long counter timeout value (LSbyte). Default value: 00000000 |
|---------------------|------------------------------------------------------------------|

Table 214. FSM_LC_TIMEOUT_H register

| | | | | | | | |
|------------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|
| FSM_LC_TIMEOUT15 | FSM_LC_TIMEOUT14 | FSM_LC_TIMEOUT13 | FSM_LC_TIMEOUT12 | FSM_LC_TIMEOUT11 | FSM_LC_TIMEOUT10 | FSM_LC_TIMEOUT9 | FSM_LC_TIMEOUT8 |
|------------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|

Table 215. FSM_LC_TIMEOUT_H register description

| | |
|----------------------|------------------------------------------------------------------|
| FSM_LC_TIMEOUT[15:8] | FSM long counter timeout value (MSbyte). Default value: 00000000 |
|----------------------|------------------------------------------------------------------|

12.1.2 FSM_PROGRAMS (56h)

FSM number of programs register (R/W)

Table 216. FSM_PROGRAMS register

| | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| FSM_N_PROG7 | FSM_N_PROG6 | FSM_N_PROG5 | FSM_N_PROG4 | FSM_N_PROG3 | FSM_N_PROG2 | FSM_N_PROG1 | FSM_N_PROG0 |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|

Table 217. FSM_PROGRAMS register description

| | |
|-----------------|-------------------------------------------------------------------------------------|
| FSM_N_PROG[7:0] | Number of FSM programs; must be less than or equal to 8. Default value: 00000000 |
|-----------------|-------------------------------------------------------------------------------------|

12.1.3 FSM_START_ADD_L (58h) and FSM_START_ADD_H (59h)

FSM start address register (R/W). First available address is 0x35C.

Table 218. FSM_START_ADD_L register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| FSM_START7 | FSM_START6 | FSM_START5 | FSM_START4 | FSM_START3 | FSM_START2 | FSM_START1 | FSM_START0 |
|------------|------------|------------|------------|------------|------------|------------|------------|

Table 219. FSM_START_ADD_L register description

| | |
|----------------|-----------------------------------------------------------|
| FSM_START[7:0] | FSM start address value (LSbyte). Default value: 00000000 |
|----------------|-----------------------------------------------------------|

Table 220. FSM_START_ADD_H register

| | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|
| FSM_START15 | FSM_START14 | FSM_START13 | FSM_START12 | FSM_START11 | FSM_START10 | FSM_START9 | FSM_START8 |
|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|

Table 221. FSM_START_ADD_H register description

| | |
|-----------------|-----------------------------------------------------------|
| FSM_START[15:8] | FSM start address value (MSbyte). Default value: 00000000 |
|-----------------|-----------------------------------------------------------|

12.1.4 PEDO_CMD_REG (5Dh)

Pedometer configuration register (R/W)

Table 222. PEDO_CMD_REG register

| | | | | | | | |
|------------------|------------------|------------------|------------------|----------------|-----------------|------------------|------------------|
| 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | 0 ⁽¹⁾ | CARRY_COUNT_EN | FP_REJECTION_EN | 0 ⁽¹⁾ | 0 ⁽¹⁾ |
|------------------|------------------|------------------|------------------|----------------|-----------------|------------------|------------------|

1. This bit must be set to 0 for the correct operation of the device.

Table 223. PEDO_CMD_REG register description

| | |
|--------------------------------|-------------------------------------------------------------------------|
| CARRY_COUNT_EN | Set when user wants to generate interrupt only on count overflow event. |
| FP_REJECTION_EN ⁽¹⁾ | Enables the false-positive rejection feature |

1. This bit is active if the MLC_EN bit of EMB_FUNC_EN_B (05h) or the MLC_BEFORE_FSM_EN bit in the EMB_FUNC_EN_A (04h) register is set to 1.

12.1.5 PEDO_DEB_STEPS_CONF (5Eh)

Pedometer debounce configuration register (R/W)

Table 224. PEDO_DEB_STEPS_CONF register

| | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| DEB_STEP7 | DEB_STEP6 | DEB_STEP5 | DEB_STEP4 | DEB_STEP3 | DEB_STEP2 | DEB_STEP1 | DEB_STEP0 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|

Table 225. PEDO_DEB_STEPS_CONF register description

| | |
|---------------|------------------------------------------------------------------------------------------------------------------|
| DEB_STEP[7:0] | Debounce threshold. Minimum number of steps to increment the step counter (debounce). Default value: 00001010 |
|---------------|------------------------------------------------------------------------------------------------------------------|

12.1.6 PEDO_SC_DELTAT_L (AAh) and PEDO_SC_DELTAT_H (ABh)

Time period register for step detection on delta time (R/W)

Table 226. PEDO_SC_DELTAT_L register

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| PD_SC_7 | PD_SC_6 | PD_SC_5 | PD_SC_4 | PD_SC_3 | PD_SC_2 | PD_SC_1 | PD_SC_0 |
|---------|---------|---------|---------|---------|---------|---------|---------|

Table 227. PEDO_SC_DELTAT_H register

| | | | | | | | |
|----------|----------|----------|----------|----------|----------|---------|---------|
| PD_SC_15 | PD_SC_14 | PD_SC_13 | PD_SC_12 | PD_SC_11 | PD_SC_10 | PD_SC_9 | PD_SC_8 |
|----------|----------|----------|----------|----------|----------|---------|---------|

Table 228. PEDO_SC_DELTAT_H/L register description

| | |
|--------------|-----------------------------------|
| PD_SC_[15:0] | Time period value (1LSB = 6.4 ms) |
|--------------|-----------------------------------|

12.1.7 T_SENSITIVITY_L (B6h) and T_SENSITIVITY_H (B7h)

Temperature sensor sensitivity value register (R/W)

This sensitivity value is also applied to the data processed in the finite state machine (FSM) and machine learning core (MLC) blocks.

This register corresponds to the conversion value of the embedded temperature sensor. The register value is expressed as half-precision floating-point format: S EEEEEEEEEEEEEEE (S: 1 sign bit; E: 5 exponent bits; F: 10 fraction bits).

The default value of T_S_[15:0] is 0x19D1, this value corresponds to 0.045 °C/LSB with 12-bit resolution.

Table 229. T_SENSITIVITY_L register

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| T_S_7 | T_S_6 | T_S_5 | T_S_4 | T_S_3 | T_S_2 | T_S_1 | T_S_0 |
|-------|-------|-------|-------|-------|-------|-------|-------|

Table 230. T_SENSITIVITY_L register description

| | |
|-----------|------------------------------------------------------------------|
| T_S_[7:0] | Temperature sensor sensitivity (LSbyte). Default value: 11010001 |
|-----------|------------------------------------------------------------------|

Table 231. T_SENSITIVITY_H register

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|-------|-------|
| T_S_15 | T_S_14 | T_S_13 | T_S_12 | T_S_11 | T_S_10 | T_S_9 | T_S_8 |
|--------|--------|--------|--------|--------|--------|-------|-------|

Table 232. T_SENSITIVITY_H register description

| | |
|------------|------------------------------------------------------------------|
| T_S_[15:8] | Temperature sensor sensitivity (MSbyte). Default value: 00011001 |
|------------|------------------------------------------------------------------|

13 Package information

In order to meet environmental requirements, ST offers these devices in different grades of **ECOPACK** packages, depending on their level of environmental compliance. ECOPACK specifications, grade definitions and product status are available at: www.st.com. ECOPACK is an ST trademark.

13.1 Soldering information

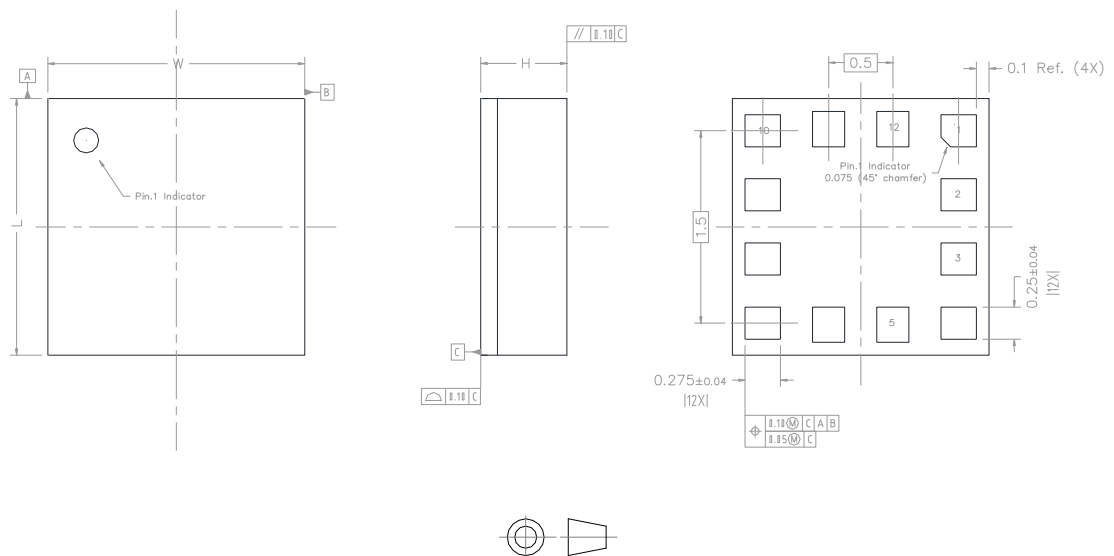
The LGA package is compliant with the **ECOPACK** and RoHS standard.

It is qualified for soldering heat resistance according to JEDEC J-STD-020.

For the land pattern and soldering recommendations, consult technical note **TN0018** available on www.st.com.

13.2 LGA-12 package information

Figure 18. LGA-12 2.0 x 2.0 x 0.74 mm package outline and mechanical data



Dimensions are in millimeter unless otherwise specified
General Tolerance is +/-0.15mm unless otherwise specified

OUTER DIMENSIONS

| ITEM | DIMENSION [mm] | TOLERANCE [mm] |
|------------|----------------|----------------|
| Length [L] | 2 | ±0.1 |
| Width [W] | 2 | ±0.1 |
| Height [H] | 0.74 MAX | / |

DM00170568

Revision history

Table 233. Document revision history

| Date | Version | Changes |
|-------------|---------|-----------------------|
| 09-Nov-2022 | 2 | Minor textual updates |

Contents

| | | |
|----------|--------------------------------------------------------------------|-----------|
| 1 | Block diagram and pin description | 3 |
| 1.1 | Block diagram | 3 |
| 1.2 | Pin description | 4 |
| 2 | Mechanical and electrical specifications | 5 |
| 2.1 | Mechanical characteristics | 5 |
| 2.2 | Electrical characteristics | 6 |
| 2.3 | Temperature sensor characteristics | 6 |
| 2.4 | Communication interface characteristics | 7 |
| 2.4.1 | SPI - serial peripheral interface | 7 |
| 2.4.2 | I ² C - inter-IC control interface | 9 |
| 2.5 | Absolute maximum ratings | 10 |
| 3 | Terminology and functionality | 11 |
| 3.1 | Terminology | 11 |
| 3.1.1 | Sensitivity | 11 |
| 3.1.2 | Zero-g level offset | 11 |
| 3.2 | Functionality | 11 |
| 3.2.1 | Operating modes | 11 |
| 3.2.2 | One-shot modes | 11 |
| 3.2.3 | Self-test | 11 |
| 3.2.4 | Activity/inactivity, Android stationary/motion detection functions | 12 |
| 3.2.5 | Interrupt event recognition | 12 |
| 3.3 | Sensing element | 13 |
| 3.4 | IC interface | 13 |
| 3.5 | Factory calibration | 13 |
| 3.6 | Temperature sensor | 13 |
| 4 | Application hints | 14 |
| 4.1 | Power-up sequence | 16 |
| 4.1.1 | Power-up command | 17 |
| 4.1.2 | Soft power-off command | 17 |
| 4.1.3 | Deep power-off command | 17 |
| 5 | Digital main blocks and embedded low-power features | 18 |
| 5.1 | FIFO | 18 |
| 5.1.1 | Bypass mode | 19 |
| 5.1.2 | FIFO mode | 19 |
| 5.1.3 | Continuous mode | 19 |

| | | |
|----------|----------------------------------------------------------------------------------|-----------|
| 5.1.4 | Continuous-to-FIFO mode | 19 |
| 5.1.5 | Bypass-to-continuous mode | 20 |
| 5.1.6 | Bypass-to-FIFO | 20 |
| 5.1.7 | FIFO reading procedure | 20 |
| 5.1.8 | FIFO empty condition | 20 |
| 5.2 | Pedometer functions: step detector and step counters | 21 |
| 5.3 | Pedometer algorithm | 21 |
| 5.4 | Tilt detection | 22 |
| 5.5 | Significant motion detection | 22 |
| 5.6 | Finite state machine | 22 |
| 5.7 | Machine learning core | 23 |
| 5.8 | Adaptive self-configuration (ASC) | 24 |
| 6 | Digital interfaces | 25 |
| 6.1 | I ² C serial interface | 25 |
| 6.1.1 | I ² C operation | 26 |
| 6.2 | SPI bus interface | 27 |
| 6.2.1 | SPI read | 28 |
| 6.2.2 | SPI write | 29 |
| 6.2.3 | SPI read in 3-wire mode | 30 |
| 6.3 | MIPI I3C [®] interface | 31 |
| 6.3.1 | MIPI I3C [®] slave interface | 31 |
| 6.3.2 | MIPI I3C [®] CCC supported commands | 32 |
| 6.3.3 | Anti-spike filter management on mixed I ² C/MIPI I3C [®] bus | 33 |
| 7 | Register mapping | 34 |
| 8 | Register description | 36 |
| 8.1 | PIN_CTRL (0Ch) | 36 |
| 8.2 | WAKE_UP_DUR_EXT (0Eh) | 36 |
| 8.3 | WHO_AM_I (0Fh) | 37 |
| 8.4 | CTRL1 (10h) | 37 |
| 8.5 | CTRL2 (11h) | 38 |
| 8.6 | CTRL3 (12h) | 38 |
| 8.7 | CTRL4 (13h) | 39 |
| 8.8 | CTRL5 (14h) | 40 |
| 8.9 | FIFO_CTRL (15h) | 42 |
| 8.10 | FIFO_WTM (16h) | 43 |
| 8.11 | INTERRUPT_CFG (17h) | 43 |

| | | |
|-------------|----------------------------------------------------------|----|
| 8.12 | SIXD (18h)..... | 44 |
| 8.13 | WAKE_UP_THS (1Ch)..... | 44 |
| 8.14 | WAKE_UP_DUR (1Dh)..... | 45 |
| 8.15 | FREE_FALL (1Eh)..... | 45 |
| 8.16 | MD1_CFG (1Fh)..... | 46 |
| 8.17 | MD2_CFG (20h)..... | 46 |
| 8.18 | WAKE_UP_SRC (21h)..... | 47 |
| 8.19 | TAP_SRC (22h)..... | 48 |
| 8.20 | SIXD_SRC (23h)..... | 48 |
| 8.21 | ALL_INT_SRC (24h)..... | 49 |
| 8.22 | STATUS (25h)..... | 49 |
| 8.23 | FIFO_STATUS1 (26h)..... | 50 |
| 8.24 | FIFO_STATUS2 (27h)..... | 50 |
| 8.25 | OUT_X_L (28h)..... | 50 |
| 8.26 | OUT_X_H (29h)..... | 50 |
| 8.27 | OUT_Y_L (2Ah)..... | 51 |
| 8.28 | OUT_Y_H (2Bh)..... | 51 |
| 8.29 | OUT_Z_L (2Ch)..... | 51 |
| 8.30 | OUT_Z_H (2Dh)..... | 51 |
| 8.31 | OUT_T_L (2Eh)..... | 52 |
| 8.32 | OUT_T_H (2Fh)..... | 52 |
| 8.33 | SELF_TEST (32h)..... | 53 |
| 8.34 | I3C_IF_CTRL (33h)..... | 53 |
| 8.35 | EMB_FUNC_STATUS_MAINPAGE (34h)..... | 53 |
| 8.36 | FSM_STATUS_MAINPAGE (35h)..... | 54 |
| 8.37 | MLC_STATUS_MAINPAGE (36h)..... | 54 |
| 8.38 | SLEEP (3Dh)..... | 54 |
| 8.39 | IF_WAKE_UP (3Eh)..... | 55 |
| 8.40 | FUNC_CFG_ACCESS (3Fh)..... | 55 |
| 8.41 | FIFO_DATA_OUT_TAG (40h)..... | 55 |
| 8.42 | FIFO_DATA_OUT_X_L (41h) and FIFO_DATA_OUT_X_H (42h)..... | 56 |
| 8.43 | FIFO_DATA_OUT_Y_L (43h) and FIFO_DATA_OUT_Y_H (44h)..... | 56 |
| 8.44 | FIFO_DATA_OUT_Z_L (45h) and FIFO_DATA_OUT_Z_H (46h)..... | 57 |
| 8.45 | FIFO_BATCH_DEC (47h)..... | 57 |
| 8.46 | TAP_CFG0 (6Fh)..... | 58 |
| 8.47 | TAP_CFG1 (70h)..... | 58 |

| | | |
|-----------|----------------------------------------------------------------------------|-----------|
| 8.48 | TAP_CFG2 (71h) | 59 |
| 8.49 | TAP_CFG3 (72h) | 59 |
| 8.50 | TAP_CFG4 (73h) | 59 |
| 8.51 | TAP_CFG5 (74h) | 60 |
| 8.52 | TAP_CFG6 (75h) | 60 |
| 8.53 | TIMESTAMP3 (7Ah), TIMESTAMP2 (7Bh), TIMESTAMP1 (7Ch), and TIMESTAMP0 (7Dh) | 61 |
| 9 | Embedded functions register mapping | 62 |
| 10 | Embedded functions register description | 64 |
| 10.1 | PAGE_SEL (02h) | 64 |
| 10.2 | EMB_FUNC_EN_A (04h) | 64 |
| 10.3 | EMB_FUNC_EN_B (05h) | 65 |
| 10.4 | EMB_FUNC_EXEC_STATUS (07h) | 65 |
| 10.5 | PAGE_ADDRESS (08h) | 65 |
| 10.6 | PAGE_VALUE (09h) | 66 |
| 10.7 | EMB_FUNC_INT1 (0Ah) | 66 |
| 10.8 | FSM_INT1 (0Bh) | 67 |
| 10.9 | MLC_INT1 (0Dh) | 68 |
| 10.10 | EMB_FUNC_INT2 (0Eh) | 68 |
| 10.11 | FSM_INT2 (0Fh) | 69 |
| 10.12 | MLC_INT2 (11h) | 70 |
| 10.13 | EMB_FUNC_STATUS (12h) | 70 |
| 10.14 | FSM_STATUS (13h) | 71 |
| 10.15 | MLC_STATUS (15h) | 71 |
| 10.16 | PAGE_RW (17h) | 72 |
| 10.17 | EMB_FUNC_FIFO_EN (18h) | 72 |
| 10.18 | FSM_ENABLE (1Ah) | 73 |
| 10.19 | FSM_LONG_COUNTER_L (1Ch) and FSM_LONG_COUNTER_H (1Dh) | 73 |
| 10.20 | INT_ACK_MASK (1Fh) | 74 |
| 10.21 | FSM_OUTS1 (20h) | 75 |
| 10.22 | FSM_OUTS2 (21h) | 75 |
| 10.23 | FSM_OUTS3 (22h) | 76 |
| 10.24 | FSM_OUTS4 (23h) | 76 |
| 10.25 | FSM_OUTS5 (24h) | 77 |
| 10.26 | FSM_OUTS6 (25h) | 77 |
| 10.27 | FSM_OUTS7 (26h) | 78 |

| | | |
|-----------|--------------------------------------------------------------|------------|
| 10.28 | FSM_OUTS8 (27h) | 78 |
| 10.29 | STEP_COUNTER_L (28h) and STEP_COUNTER_H (29h) | 79 |
| 10.30 | EMB_FUNC_SRC (2Ah) | 79 |
| 10.31 | EMB_FUNC_INIT_A (2Ch) | 80 |
| 10.32 | EMB_FUNC_INIT_B (2Dh) | 80 |
| 10.33 | MLC1_SRC (34h) | 80 |
| 10.34 | MLC2_SRC (35h) | 81 |
| 10.35 | MLC3_SRC (36h) | 81 |
| 10.36 | MLC4_SRC (37h) | 81 |
| 10.37 | FSM_ODR (39h) | 82 |
| 10.38 | MLC_ODR (3Ah) | 82 |
| 11 | Embedded advanced features pages | 83 |
| 12 | Embedded advanced features register description | 85 |
| 12.1 | Page 0 - embedded advanced features registers | 85 |
| 12.1.1 | FSM_LC_TIMEOUT_L (54h) and FSM_LC_TIMEOUT_H (55h) | 85 |
| 12.1.2 | FSM_PROGRAMS (56h) | 85 |
| 12.1.3 | FSM_START_ADD_L (58h) and FSM_START_ADD_H (59h) | 86 |
| 12.1.4 | PEDO_CMD_REG (5Dh) | 86 |
| 12.1.5 | PEDO_DEB_STEPS_CONF (5Eh) | 87 |
| 12.1.6 | PEDO_SC_DELTAT_L (AAh) and PEDO_SC_DELTAT_H (ABh) | 87 |
| 12.1.7 | T_SENSITIVITY_L (B6h) and T_SENSITIVITY_H (B7h) | 88 |
| 13 | Package information | 89 |
| 13.1 | Soldering information | 89 |
| 13.2 | LGA-12 package information | 89 |
| | Revision history | 90 |
| | List of tables | 96 |
| | List of figures | 101 |

List of tables

| | | |
|------------------|-----------------------------------------------------------------------------------------|----|
| Table 1. | Pin description | 4 |
| Table 2. | Mechanical characteristics | 5 |
| Table 3. | Electrical characteristics | 6 |
| Table 4. | Temperature sensor characteristics | 6 |
| Table 5. | SPI slave timing values | 7 |
| Table 6. | I ² C slave timing values | 9 |
| Table 7. | Absolute maximum ratings | 10 |
| Table 8. | Configuration of duration of interrupt | 12 |
| Table 9. | Internal pin status | 15 |
| Table 10. | Serial interface pin description | 25 |
| Table 11. | I ² C terminology | 25 |
| Table 12. | SAD+read/write patterns | 26 |
| Table 13. | Transfer when master is writing one byte to slave | 26 |
| Table 14. | Transfer when master is writing multiple bytes to slave | 26 |
| Table 15. | Transfer when master is receiving (reading) one byte of data from slave | 26 |
| Table 16. | Transfer when master is receiving (reading) multiple bytes of data from slave | 26 |
| Table 17. | MIPI I3C [®] CCC commands | 32 |
| Table 18. | Register map | 34 |
| Table 19. | Table 20. PIN_CTRL register | 36 |
| Table 20. | PIN_CTRL register description | 36 |
| Table 21. | WAKE_UP_DUR_EXT register | 36 |
| Table 22. | WAKE_UP_DUR_EXT register description | 36 |
| Table 23. | WHO_AM_I register default values | 37 |
| Table 24. | CTRL1 register | 37 |
| Table 25. | CTRL1 register description | 37 |
| Table 26. | CTRL2 register | 38 |
| Table 27. | CTRL2 register description | 38 |
| Table 28. | CTRL3 register | 38 |
| Table 29. | CTRL3 register description | 38 |
| Table 30. | CTRL4 register | 39 |
| Table 31. | CTRL4 register description | 39 |
| Table 32. | ODR frequency in inactivity state | 39 |
| Table 33. | CTRL5 register | 40 |
| Table 34. | CTRL5 register description | 40 |
| Table 35. | Operating modes | 40 |
| Table 36. | Bandwidth selection (low-power mode with ODR < 50 Hz) | 41 |
| Table 37. | Full-scale selection | 41 |
| Table 38. | FIFO_CTRL register description | 42 |
| Table 39. | Selection of FIFO mode | 42 |
| Table 40. | FIFO_WTM register | 43 |
| Table 41. | FIFO_WTM register description | 43 |
| Table 42. | INTERRUPT_CFG register | 43 |
| Table 43. | INTERRUPT_CFG register description | 43 |
| Table 44. | SIXD register | 44 |
| Table 45. | SIXD register description | 44 |
| Table 46. | WAKE_UP_THS register | 44 |
| Table 47. | WAKE_UP_THS register description | 44 |
| Table 48. | WAKE_UP_DUR register | 45 |
| Table 49. | WAKE_UP_DUR register description | 45 |
| Table 50. | FREE_FALL register | 45 |
| Table 51. | FREE_FALL register description | 45 |
| Table 52. | MD1_CFG register | 46 |
| Table 53. | MD1_CFG register description | 46 |

| | | |
|-------------------|------------------------------------------------------------------------|----|
| Table 54. | MD2_CFG register | 46 |
| Table 55. | MD2_CFG register description | 46 |
| Table 56. | WAKE_UP_SRC register | 47 |
| Table 57. | WAKE_UP_SRC register description | 47 |
| Table 58. | TAP_SRC register | 48 |
| Table 59. | TAP_SRC register description | 48 |
| Table 60. | SIXD_SRC register | 48 |
| Table 61. | SIXD_SRC register description | 48 |
| Table 62. | ALL_INT_SRC register | 49 |
| Table 63. | ALL_INT_SRC register description | 49 |
| Table 64. | STATUS register | 49 |
| Table 65. | STATUS register description | 49 |
| Table 66. | FIFO_STATUS1 register | 50 |
| Table 67. | FIFO_STATUS1 register description | 50 |
| Table 68. | FIFO_STATUS2 register | 50 |
| Table 69. | FIFO_STATUS2 register description | 50 |
| Table 70. | OUT_X_L register | 50 |
| Table 71. | OUT_X_L register description | 50 |
| Table 72. | OUT_X_H register | 50 |
| Table 73. | OUT_X_H register description | 50 |
| Table 74. | OUT_Y_L register | 51 |
| Table 75. | OUT_Y_L register description | 51 |
| Table 76. | OUT_Y_H register | 51 |
| Table 77. | OUT_Y_H register description | 51 |
| Table 78. | OUT_Z_L register | 51 |
| Table 79. | OUT_Z_L register description | 51 |
| Table 80. | OUT_Z_H register | 51 |
| Table 81. | OUT_Z_H register description | 51 |
| Table 82. | OUT_T_L register | 52 |
| Table 83. | OUT_T_L register description | 52 |
| Table 84. | OUT_T_H register | 52 |
| Table 85. | OUT_T_H register description | 52 |
| Table 86. | SELF_TEST register | 53 |
| Table 87. | SELF_TEST register description | 53 |
| Table 88. | I3C_IF_CTRL register | 53 |
| Table 89. | I3C_IF_CTRL register description | 53 |
| Table 90. | EMB_FUNC_STATUS_MAINPAGE register | 53 |
| Table 91. | EMB_FUNC_STATUS_MAINPAGE register description | 53 |
| Table 92. | FSM_STATUS_MAINPAGE register | 54 |
| Table 93. | FSM_STATUS_MAINPAGE register description | 54 |
| Table 94. | MLC_STATUS_MAINPAGE register | 54 |
| Table 95. | MLC_STATUS_MAINPAGE register description | 54 |
| Table 96. | SLEEP register | 54 |
| Table 97. | SLEEP register description | 54 |
| Table 98. | IF_WAKE_UP register | 55 |
| Table 99. | IF_WAKE_UP register description | 55 |
| Table 100. | FUNC_CFG_ACCESS register | 55 |
| Table 101. | FUNC_CFG_ACCESS register description | 55 |
| Table 102. | FIFO_DATA_OUT_TAG register | 55 |
| Table 103. | FIFO_DATA_OUT_TAG register description | 55 |
| Table 104. | Identification of sensor in FIFO | 56 |
| Table 105. | FIFO_DATA_OUT_X_H and FIFO_DATA_OUT_X_L registers | 56 |
| Table 106. | FIFO_DATA_OUT_X_H and FIFO_DATA_OUT_X_L register description | 56 |
| Table 107. | FIFO_DATA_OUT_Y_H and FIFO_DATA_OUT_Y_L registers | 56 |
| Table 108. | FIFO_DATA_OUT_Y_H and FIFO_DATA_OUT_Y_L register description | 56 |

| | | |
|-------------------|------------------------------------------------------------------------|----|
| Table 109. | FIFO_DATA_OUT_Z_H and FIFO_DATA_OUT_Z_L registers | 57 |
| Table 110. | FIFO_DATA_OUT_Z_H and FIFO_DATA_OUT_Z_L register description | 57 |
| Table 111. | FIFO_BATCH_DEC register | 57 |
| Table 112. | FIFO_BATCH_DEC register description | 57 |
| Table 113. | Accelerometer batch data rate | 57 |
| Table 114. | TAP_CFG0 register | 58 |
| Table 115. | TAP_CFG0 register description | 58 |
| Table 116. | TAP_CFG1 register | 58 |
| Table 117. | TAP_CFG1 register description | 58 |
| Table 118. | TAP_CFG2 register | 59 |
| Table 119. | TAP_CFG2 register description | 59 |
| Table 120. | TAP_CFG3 register | 59 |
| Table 121. | TAP_CFG3 register description | 59 |
| Table 122. | TAP_CFG4 register | 59 |
| Table 123. | TAP_CFG4 register description | 59 |
| Table 124. | TAP_CFG5 register | 60 |
| Table 125. | TAP_CFG5 register description | 60 |
| Table 126. | TAP_CFG6 register | 60 |
| Table 127. | TAP_CFG6 register description | 60 |
| Table 128. | TIMESTAMP output registers | 61 |
| Table 129. | TIMESTAMP output register description | 61 |
| Table 130. | Register address map - embedded functions | 62 |
| Table 131. | PAGE_SEL register | 64 |
| Table 132. | PAGE_SEL register description | 64 |
| Table 133. | EMB_FUNC_EN_A register. | 64 |
| Table 134. | EMB_FUNC_EN_A register description | 64 |
| Table 135. | EMB_FUNC_EN_B register. | 65 |
| Table 136. | EMB_FUNC_EN_B register description | 65 |
| Table 137. | EMB_FUNC_EXEC_STATUS register | 65 |
| Table 138. | EMB_FUNC_EXEC_STATUS register description | 65 |
| Table 139. | PAGE_ADDRESS register | 65 |
| Table 140. | PAGE_ADDRESS register description | 65 |
| Table 141. | PAGE_VALUE register | 66 |
| Table 142. | PAGE_VALUE register description | 66 |
| Table 143. | EMB_FUNC_INT1 register | 66 |
| Table 144. | EMB_FUNC_INT1 register description | 66 |
| Table 145. | FSM_INT1 register. | 67 |
| Table 146. | FSM_INT1 register description | 67 |
| Table 147. | MLC_INT1 register. | 68 |
| Table 148. | MLC_INT1 register description | 68 |
| Table 149. | EMB_FUNC_INT2 register | 68 |
| Table 150. | EMB_FUNC_INT2 register description | 68 |
| Table 151. | FSM_INT2 register. | 69 |
| Table 152. | FSM_INT2 register description | 69 |
| Table 153. | MLC_INT2 register. | 70 |
| Table 154. | MLC_INT2 register description | 70 |
| Table 155. | EMB_FUNC_STATUS register. | 70 |
| Table 156. | EMB_FUNC_STATUS register description. | 70 |
| Table 157. | FSM_STATUS register | 71 |
| Table 158. | FSM_STATUS register description | 71 |
| Table 159. | MLC_STATUS register | 71 |
| Table 160. | MLC_STATUS register description | 71 |
| Table 161. | PAGE_RW register | 72 |
| Table 162. | PAGE_RW register description | 72 |
| Table 163. | EMB_FUNC_FIFO_EN register | 72 |

| | | |
|-------------------|--------------------------------------------------------------------|----|
| Table 164. | EMB_FUNC_FIFO_EN register description | 72 |
| Table 165. | FSM_ENABLE register | 73 |
| Table 166. | FSM_ENABLE register description | 73 |
| Table 167. | FSM_LONG_COUNTER_L register | 73 |
| Table 168. | FSM_LONG_COUNTER_L register description | 73 |
| Table 169. | FSM_LONG_COUNTER_H register | 73 |
| Table 170. | FSM_LONG_COUNTER_H register description | 73 |
| Table 171. | INT_ACK_MASK register | 74 |
| Table 172. | INT_ACK_MASK register description | 74 |
| Table 173. | FSM_OUTS1 register | 75 |
| Table 174. | FSM_OUTS1 register description | 75 |
| Table 175. | FSM_OUTS2 register | 75 |
| Table 176. | FSM_OUTS2 register description | 75 |
| Table 177. | FSM_OUTS3 register | 76 |
| Table 178. | FSM_OUTS3 register description | 76 |
| Table 179. | FSM_OUTS4 register | 76 |
| Table 180. | FSM_OUTS4 register description | 76 |
| Table 181. | FSM_OUTS5 register | 77 |
| Table 182. | FSM_OUTS5 register description | 77 |
| Table 183. | FSM_OUTS6 register | 77 |
| Table 184. | FSM_OUTS6 register description | 77 |
| Table 185. | FSM_OUTS7 register | 78 |
| Table 186. | FSM_OUTS7 register description | 78 |
| Table 187. | FSM_OUTS8 register | 78 |
| Table 188. | FSM_OUTS8 register description | 78 |
| Table 189. | STEP_COUNTER_L register | 79 |
| Table 190. | STEP_COUNTER_L register description | 79 |
| Table 191. | STEP_COUNTER_H register | 79 |
| Table 192. | STEP_COUNTER_H register description | 79 |
| Table 193. | EMB_FUNC_SRC register | 79 |
| Table 194. | EMB_FUNC_SRC register description | 79 |
| Table 195. | EMB_FUNC_INIT_A register | 80 |
| Table 196. | EMB_FUNC_INIT_A register description | 80 |
| Table 197. | EMB_FUNC_INIT_B register | 80 |
| Table 198. | EMB_FUNC_INIT_B register description | 80 |
| Table 199. | MLC1_SRC register | 80 |
| Table 200. | MLC1_SRC register description | 80 |
| Table 201. | MLC2_SRC register | 81 |
| Table 202. | MLC2_SRC register description | 81 |
| Table 203. | MLC3_SRC register | 81 |
| Table 204. | MLC3_SRC register description | 81 |
| Table 205. | MLC4_SRC register | 81 |
| Table 206. | MLC4_SRC register description | 81 |
| Table 207. | FSM_ODR register | 82 |
| Table 208. | FSM_ODR register description | 82 |
| Table 209. | MLC_ODR register | 82 |
| Table 210. | MLC_ODR register description | 82 |
| Table 211. | Register address map - embedded advanced features page 0 | 83 |
| Table 212. | FSM_LC_TIMEOUT_L register | 85 |
| Table 213. | FSM_LC_TIMEOUT_L register description | 85 |
| Table 214. | FSM_LC_TIMEOUT_H register | 85 |
| Table 215. | FSM_LC_TIMEOUT_H register description | 85 |
| Table 216. | FSM_PROGRAMS register | 85 |
| Table 217. | FSM_PROGRAMS register description | 85 |
| Table 218. | FSM_START_ADD_L register | 86 |

| | |
|----------------------------------------------------------------------|----|
| Table 219. FSM_START_ADD_L register description | 86 |
| Table 220. FSM_START_ADD_H register | 86 |
| Table 221. FSM_START_ADD_H register description | 86 |
| Table 222. PEDO_CMD_REG register | 86 |
| Table 223. PEDO_CMD_REG register description | 86 |
| Table 224. PEDO_DEB_STEPS_CONF register | 87 |
| Table 225. PEDO_DEB_STEPS_CONF register description | 87 |
| Table 226. PEDO_SC_DELTAT_L register | 87 |
| Table 227. PEDO_SC_DELTAT_H register | 87 |
| Table 228. PEDO_SC_DELTAT_H/L register description | 87 |
| Table 229. T_SENSITIVITY_L register | 88 |
| Table 230. T_SENSITIVITY_L register description | 88 |
| Table 231. T_SENSITIVITY_H register | 88 |
| Table 232. T_SENSITIVITY_H register description | 88 |
| Table 233. Document revision history | 90 |

List of figures

| | | |
|-------------------|--------------------------------------------------------------------------|----|
| Figure 1. | Block diagram | 3 |
| Figure 2. | Pin connections | 4 |
| Figure 3. | SPI slave timing in mode 0 | 7 |
| Figure 4. | SPI slave timing in mode 3 | 8 |
| Figure 5. | I ² C slave timing diagram | 9 |
| Figure 6. | LIS2DUX12 electrical connections (top view) | 14 |
| Figure 7. | Power-up sequence | 16 |
| Figure 8. | Four-stage pedometer algorithm | 21 |
| Figure 9. | Generic state machine | 22 |
| Figure 10. | State machine in the LIS2DUX12 | 23 |
| Figure 11. | Machine learning core in the LIS2DUX12 | 23 |
| Figure 12. | Read and write protocol | 27 |
| Figure 13. | SPI read protocol | 28 |
| Figure 14. | Multiple byte SPI read protocol (2-byte example) | 28 |
| Figure 15. | SPI write protocol | 29 |
| Figure 16. | Multiple byte SPI write protocol (2-byte example) | 29 |
| Figure 17. | SPI read protocol in 3-wire mode | 30 |
| Figure 18. | LGA-12 2.0 x 2.0 x 0.74 mm package outline and mechanical data | 89 |

IMPORTANT NOTICE – 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 acknowledgment.

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, refer to 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.

© 2022 STMicroelectronics – All rights reserved