苏州敏芯微电子技术股份有限公司

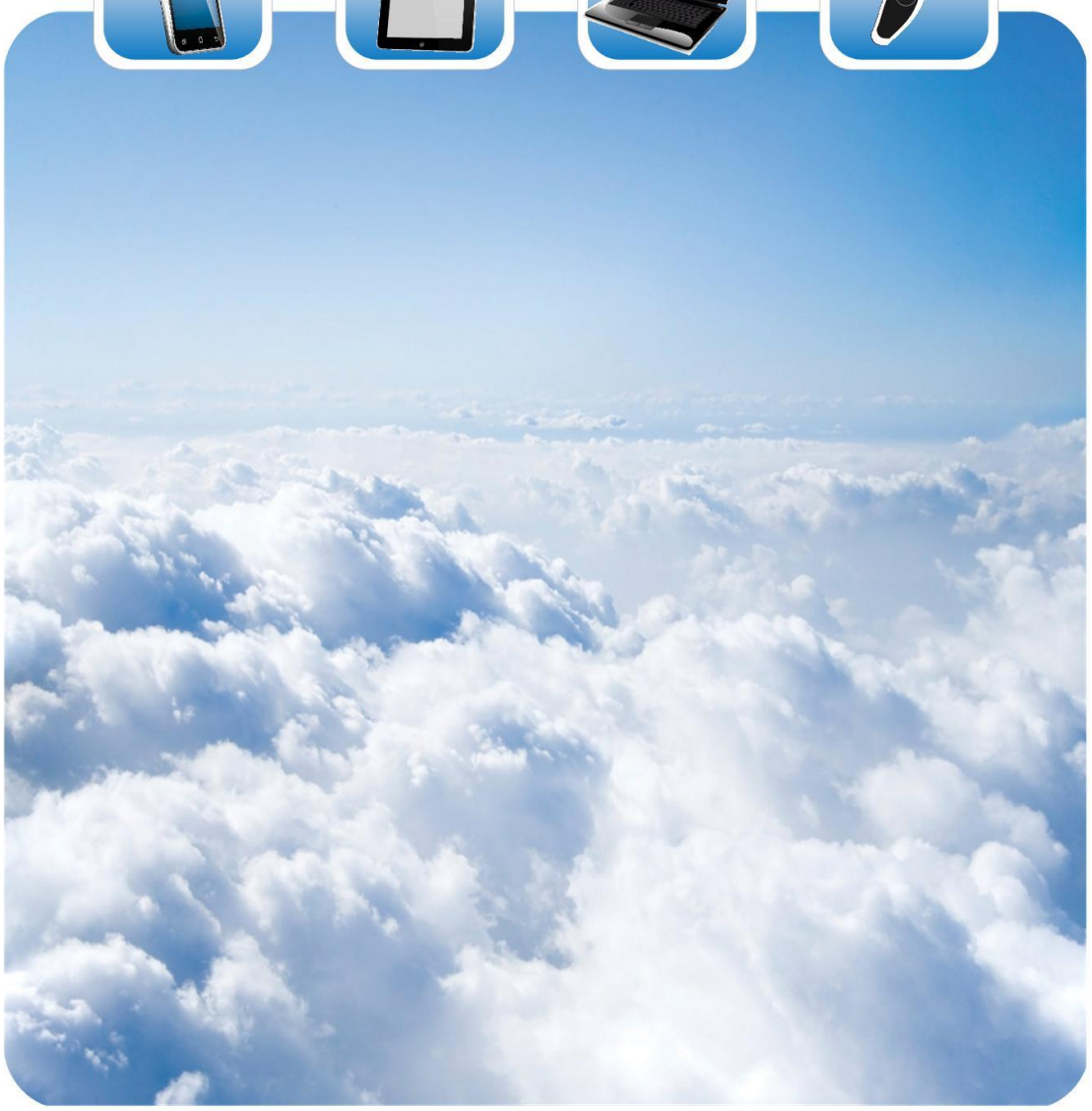
MEMSensing Microsystems (Suzhou, China) Co., Ltd.

# Data Sheet

V 1.1 / Mar. 2020

## MSA310

## Digital Tri-axial Accelerometer



## GENERAL DESCRIPTION

MSA310 is a tri-axial, low-g accelerometer with I²C digital output for consumer applications.

It has dynamical user selectable full scales range of ±2g/±4g/±8g/±16g and allows acceleration measurements with output data rates from 1Hz to 1000Hz.

MSA310 is available in an ultra small (2mm x 2mm, height 0.95mm) LGA package and is guaranteed to operate over -40°C to +85°C.

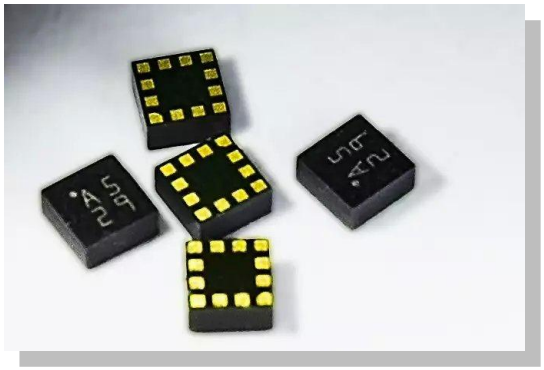
## FEATURES

- ✧ Ultra small package 2x2x0.95 mm LGA-12 pins
- ✧ User selectable range, ±2g, ±4g, ±8g, ±16g
- ✧ 1.62V to 3.6V supply voltage,
- ✧ User selectable data output rate
- ✧ I²C Interface
- ✧ One interrupt pin
- ✧ 12 bits resolution
- ✧ Integrated FIFO with a depth of 64 samples
- ✧ Low power consumption
- ✧ Factory programmed offset and sensitivity
- ✧ RoHS compliant

## APPLICATIONS

- ✧ User interface for mobile phone and tablet
- ✧ Display orientation
- ✧ Gesture recognition
- ✧ Vibration monitoring
- ✧ Inclination and tilt sensing
- ✧ Pedometer
- ✧ Gaming
- ✧ Free fall detection

## PRODUCT VIEW



# CONTENTS

GENERAL DESCRIPTION ..... 2
FEATURES ..... 2
APPLICATIONS ..... 2
PRODUCT VIEW ..... 2
1. Pin Description ..... 5
2. Specification ..... 6
*2.1 Absolute Maximum Ratings* ..... 6
*2.2 Sensor Characteristics* ..... 6
*2.3 Electrical Characteristics* ..... 7
3. Function Blocks ..... 8
4. Functional Description ..... 9
*4.1 Supply Voltage And Power Management* ..... 9
*4.2 Power Modes* ..... 9
*4.3 Sensor Data* ..... 10
*4.4 Bandwidth* ..... 10
*4.5 Interrupt Controller* ..... 11
4.5.1 General features ..... 11
4.5.2 Mapping ..... 12
4.5.3 Electrical behavior ..... 12
4.5.4 New data interrupt ..... 13
4.5.5 Active detection ..... 13
4.5.6 Tap detection ..... 14
4.5.7 Orientation recognition ..... 16
4.5.8 Freefall interrupt ..... 18
5. FIFO Operation ..... 19
*5.1 FIFO Operating Modes* ..... 19
*5.2 FIFO Data Readout* ..... 20
*5.3 FIFO Sample Counter and Overrun Flag* ..... 21
*5.4 FIFO Interrupts* ..... 21
6. Register Map ..... 23
7. Register Description ..... 24
8. Digital Interface and Application Collection ..... 33

*8.1 I2C Interface Specification* ..... 33
*8.2 Application collection* ..... 35
9. Package Description ..... 36
*9.1 Outline Dimensions* ..... 36
*9.2 Sensor orientation* ..... 37
*9.3 Tape and reel* ..... 38
*9.4 Reflow Profile* ..... 39
Revision History ..... 40

# 1. Pin Description

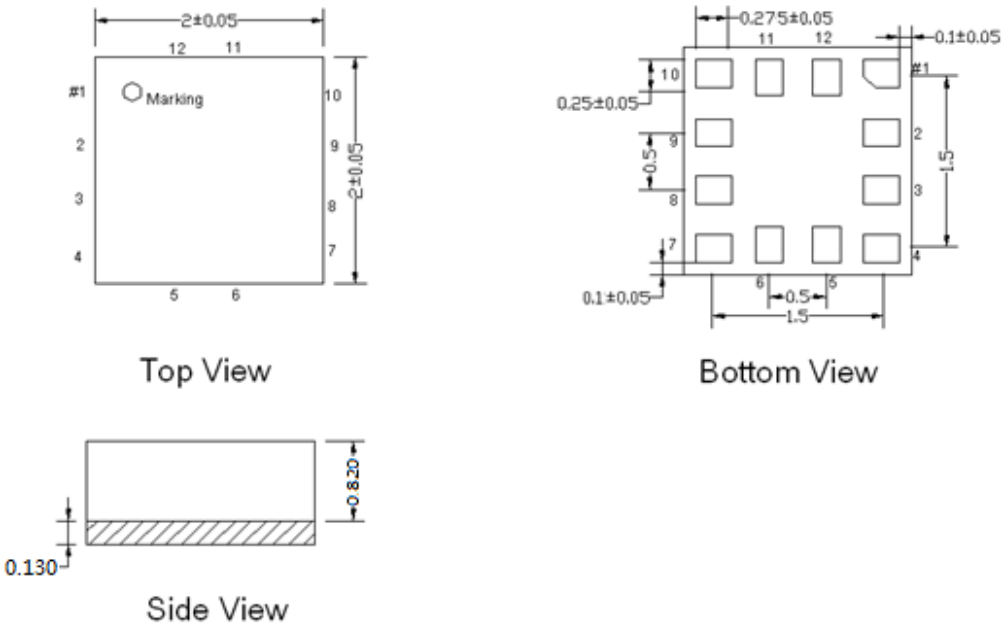



Figure 1: Pin Description

Table 1: Pin Definitions

| No. | Name | Function |
|---|---|---|
| 1 | NC | NC |
| 2 | SDA | Data input/output in I²C |
| 3 | NC | NC |
| 4 | NC | NC , Connected to GND recommended |
| 5 | INT1 | Interrupt pin |
| 6 | NC | NC |
| 7 | VDD | Power supply |
| 8 | NC | NC , Connected to GND recommended |
| 9 | GND | Ground |
| 10 | NC | NC |
| 11 | NC | NC |
| 12 | SCL | Clock for I²C interface |

# 2. Specification

## *2.1 Absolute Maximum Ratings*

Table 2: Absolute Maximum Ratings

| Parameter | Maximum value | Unit |
|---|---|---|
| Supply Voltage | −0.3 to 3.6 | V |
| Mechanical Shock | 10,000 | g |
| Operating Temperature | -40 to 85 | °C |
| Storage Temperature | -40 to 125 | °C |

## *2.2 Sensor Characteristics*

Table 3: Sensor Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| FS | Full scale range | | | ±2 | | g |
| | | | | ±4 | | g |
| | | | | ±8 | | g |
| | | | | ±16 | | g |
| S2g | Sensitivity | each axis at 25°C | | 1024 | | LSB/g |
| S4g | | each axis at 25°C | | 512 | | LSB/g |
| S8g | | each axis at 25°C | | 256 | | LSB/g |
| S16g | | each axis at 25°C | | 128 | | LSB/g |
| TCS | Sensitivity Temperature Coefficient | FS=±2g | | 0.02 | | %/°C |
| Off | Zero g Offset @ 25°C, X/Y/Z axis | FS=±2g | | ±80 | | mg |
| TCO | Zero g Offset Temperature Coefficient | FS=±2g , -40 – +85 °C | | ±1 | | mg/°C |
| Nrms | Noise Density | FS=±2g | | 300 | | μg/√Hz |
| NL | Non-Linearity | FS=±2g , best straight line | | 2 | | %FS |
| Cs | Cross Axis Sensitivity | Between any two axes | | 2 | | % |

## *2.3 Electrical Characteristics*

Table 4: Electrical Characteristics

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| VDD | Supply Voltage | | 1.62 | 1.8 | 3.6 | V |
| IDD | Supply Current | Normal operation | | 120 | | μA |
| | | Power down mode | | | 1 | μA |
| | | Low Power mode | | 2 | | μA |
| VIL | Voltage input low level | I2C | | | 0.3VDD | V |
| VIH | Voltage input high level | I2C | 0.7VDD | | | V |
| VOL | Voltage output low level | I2C | | | 0.1VDD | V |
| VOH | Voltage output high level | I2C | 0.9VDD | | | V |
| Twup | Wake up time | From power down mode | | 1 | | ms |
| Tsup | Start up time | From power off | | 3 | | ms |
| BW | Bandwidth | | 1.95 | | 500 | Hz |
| ODR | Output data rate | | 1 | | 1000 | Hz |

# 3. Function Blocks

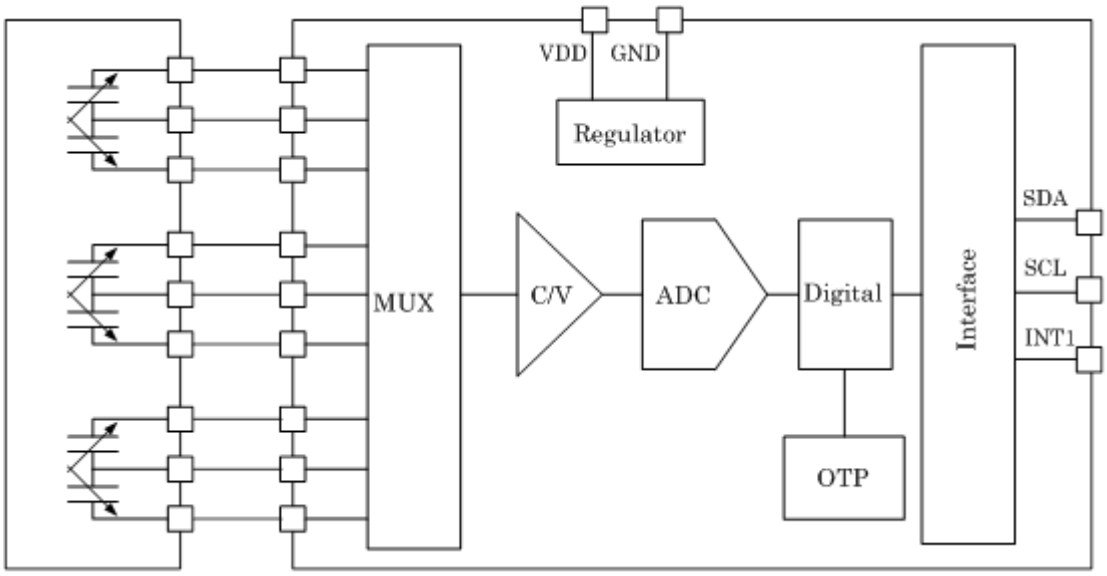



Figure 2: Function Block Diagram

# 4. Functional Description

## *4.1 Supply Voltage And Power Management*

The MSA310 has a power supply pin:

VDD is the power supply for all internal analog and digital function blocks through several regulators.

## *4.2 Power Modes*

The MSA310 has three different power modes. Besides normal mode, which represents the fully operational state of the device, there are two special energy saving modes: low-power mode and suspend mode.

**In normal mode:**
All parts of the electronic circuit are held powered-up and data acquisition is performed continuously.

**In suspend mode:**
The whole analog part, oscillators included, is powered down. No data acquisition is performed, the only supported operation is reading registers (latest acceleration data are kept). Suspend mode is entered by writing '11' to the (Register 0x11) 'POWER_MODE' bits.

**In low-power mode:**
The device is periodically switching between a sleep phase and a wake-up phase. The wake-up phase essentially corresponds to operation in normal mode with complete power-up of the circuitry. During the sleep phase the analog part except the oscillators is powered down. Low power mode is entered by writing '01' to the 'POWER_MODE' bits.

During the wake-up phase the number of samples required by any enabled interrupt is processed. If an interrupt is detected, the device stays in the wake-up phase as long as the interrupt condition endures (non-latched interrupt), or until the latch time expires (temporary interrupt), or until the interrupt is reset

(latched interrupt). If no interrupt is detected, the device enters the sleep phase.

### *4.3 Sensor Data*

The acceleration output is 12-bits two's complement data. The 12-bits data for each axis is split into MSB part (one byte containing bits 11 to 4) and LSB lower part (one byte containing bits 3 to 0).

The 'NEW_DATA_INT' bit is set when z_axis data is ready. It is reset when register (0x02) or (0x0A) is read. To enable 'NEW_DATA_INT' function, (0x17) 'NEW_DATA_INT_EN' should be set.

### *4.4 Bandwidth*

The bandwidth of the acceleration data is always half of the update rate in normal mode and listed in below table.

Table 5: bandwidth under different ODR and BW settings in normal mode

| **ODR** | **BW** |
|---|---|
| 1000Hz (1010-1111) | 500Hz |
| 500Hz (1001) | 250Hz |
| 250Hz (1000) | 125Hz |
| 125Hz (0111) | 62.5Hz |
| 62.5Hz (0110) | 31.25Hz |
| 31.25Hz (0101) | 15.63Hz |
| 15.63Hz (0100) | 7.81Hz |
| 7.81Hz (0011) | 3.9Hz |
| 3.9Hz (0010) | 1.95Hz |

MSA310 supports four different acceleration measurement range, it is selected by setting (0x0F) as follows:

Table 6: acceleration measurement range

| Range | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| Acceleration range | +/-2g | +/-4g | +/-8g | +/-16g |

## *4.5 Interrupt Controller*

Interrupt engines are integrated in the MSA310. Each interrupt can be independently enabled and configured. If the condition of an enabled interrupt is fulfilled, the corresponding status bit is set to 1 and the selected interrupt pin is activated. There is only one interrupt pin, and interrupts can be freely mapped. The pin state is a logic 'OR' combination of all mapped interrupts.

### 4.5.1 General features

An interrupt is cleared depending on the selected interrupt mode, which is common to all interrupts. There are three different interrupt modes: non-latched, latched and temporary. The mode is selected by the 'LATCH_INT' bits according to the following table.

Table 7: Interrupt mode selection

| **Latch_int** | **Interrupt mode** |
|---|---|
| 0000 | non-latched |
| 0001 | temporary latched 250ms |
| 0010 | temporary latched 500ms |
| 0011 | temporary latched 1s |
| 0100 | temporary latched 2s |
| 0101 | temporary latched 4s |
| 0110 | temporary latched 8s |
| 0111 | Latched |
| 1000 | non-latched |
| 1001 | temporary latched 1ms |
| 1010 | temporary latched 1ms |
| 1011 | temporary latched 2ms |
| 1100 | temporary latched 25ms |
| 1101 | temporary latched 50ms |
| 1110 | temporary latched 100ms |
| 1111 | Latched |

An interrupt is generated if its activation condition is met. It cannot be cleared

as long as the activation condition is fulfilled. In the non-latched mode the interrupt status bit and interrupt pin are cleared as soon as the activation condition is no more valid. Exceptions to this behavior are the new data and orientation, which are automatically reset after a fixed time.

In the latched mode an asserted interrupt status and the interrupt pin are cleared by writing 1 to bit 'RESET_INT'. If the activation condition still holds when it is cleared, the interrupt status is asserted again with the next change of the acceleration registers.

In the temporary mode an asserted interrupt and selected pin are cleared after a defined period of time. The behavior of the different interrupt modes is shown in the following figure.

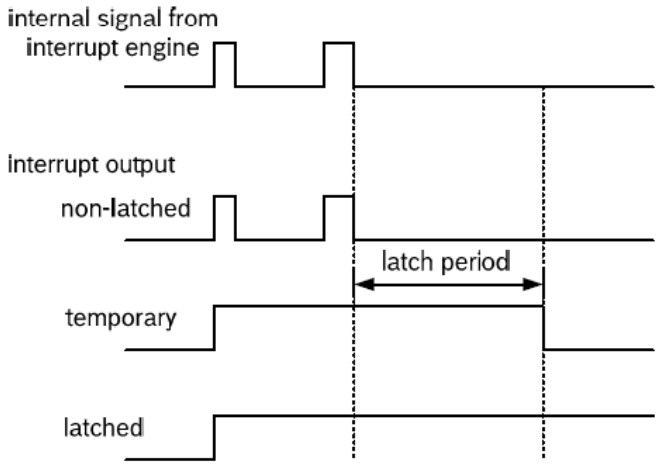



Figure 3: Interrupt mode

### 4.5.2 Mapping

The mapping of interrupts to the interrupt pins is done by register (0x19, 0x1A), setting int1_int type (int1_orient/int1_s_tap/int1_freefall, etc) to 1 can map this type interrupt to INT1 pin.

### 4.5.3 Electrical behavior

Interrupt pin can be configured to show desired electrical behavior. The 'active'

level of interrupt pin is determined by (0x20) 'INT1_LVL'. If (0x20) 'INT1_LVL' = '1' ('0') = '1' ('0'), then INT1 is active '1' ('0'). In addition to that, also the electric type of the interrupt pins can be selected. By setting bits (0x20) 'INT1_OD' to '0', the interrupt pin output type is push-pull, by setting the configuration bits to '1', the output type is open-drain.

#### 4.5.4 New data interrupt

This interrupt serves for synchronous reading of acceleration data. It is generated after an acceleration data was calculated. The interrupt is cleared automatically before the next acceleration data is ready.

#### 4.5.5 Active detection

Active detection uses the slope between successive acceleration signals to detect changes in motion. An interrupt is generated when the slope (absolute value of acceleration difference) exceeds a preset threshold.

The threshold is set with the value of register (0x28) active_th, with the LSB corresponding to 4 LSB of acceleration data. That is 3.9mg in 2g-range (7.8mg in 4g-range, 15.6mg in 8g-range and 31.3mg in 16g-range). And the maximum value is 1g in 2g-range, 2g in 4g-range, 4g in 8g-range and 8g in 16g-range.

The time difference between the successive acceleration signals depends is fixed to 1ms.

Active detection can be enabled (disabled) for each axis separately by writing '1' to bits (0x16) 'ACTIVE_EN_X/Y/Z'. The active interrupt is generated if the slope of any of the enabled axes exceeds the threshold for ['ACTIVE_DUR'+1] consecutive times. As soon as the slopes of all enabled axes fall below this threshold for ['ACTIVE_DUR'+1] consecutive times, the interrupt is cleared unless the interrupt signal is latched.

The interrupt status is stored in bit (0x09) 'ACTIVE_INT'. The bit (0x0B) 'ACTIVE_FIRST_X/Y/Z' record which axis triggered the active interrupt first and the sign of this acceleration data that triggered the interrupt is recorded in the

bit (0x0B) 'ACTIVE_SIGN'.

### 4.5.6 Tap detection

Tap detection has a functional similarity with a common laptop touch-pad or clicking keys of a computer mouse. A tap event is detected if a pre-defined pattern of the acceleration slope is fulfilled at least for one axis. Two different tap events are distinguished: A single tap is a single event within a certain time, followed by a certain quiet time. A double tap consists of a first such event followed by a second event within a defined time.

Single tap interrupt is enabled by writing 1 to bit (0x16) 'S_TAP_INT_EN' and double tap interrupt is enabled by writing 1 to bit (0x16) 'D_TAP_INT_EN'. The status of the single tap interrupt is stored in bit (0x09) 'S_TAP_INT' and the status of the double tap interrupt is stored in bit (0x09) 'D_TAP_INT'.

The slope threshold for detecting a tap event is set by bits (0x2B) "TAP_TH" with the LSB corresponding to 64LSB of acceleration data that is 62.5mg in 2g-range, 125mg in 4g-range, 250mg in 8g-range, 500mg in 16g-range. And the maximum value equals to the full scale in each range.

In the following figure the meaning of different timing parameter is visualized.

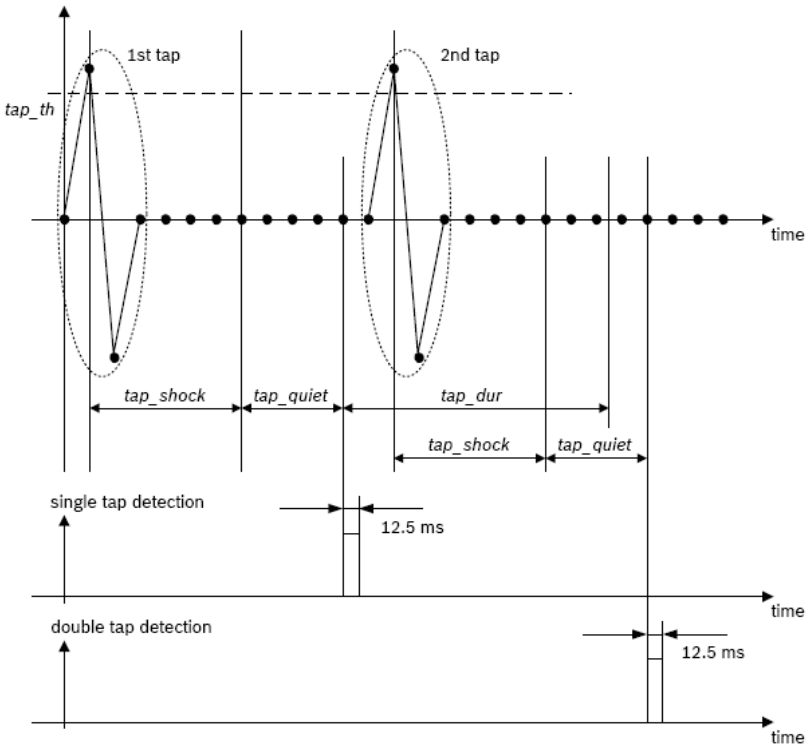



Figure 4: Timing of Tap detection

The parameter (0x2A) 'TAP_SHOCK' and (0x2A) 'TAP_QUIET' apply to both single and double tap detection, while (0x2A) 'TAP_DUR' applies to double detection only. Within the duration of (0x2A) 'TAP_SHOCK' any slope exceeding (0x2B) 'TAP_TH' after the first event is ignored, within the duration of (0x2A) 'TAP_QUIET' there must be no slope exceeding 'TAP_TH', otherwise the first event will be cancelled.

A single tap is detected and the single tap interrupt is generated after the combination durations of (0x2A) 'TAP_SHOCK' and (0x2A) 'TAP_QUIET', if the corresponding slope conditions are fulfilled. The interrupt is cleared after a delay of 12.5ms in non-latched mode.

A double tap is detected and the double tap interrupt is generated if an event fulfilling the conditions for a single tap occurs within the set duration in (0x2A) 'TAP_DUR' after the completion of the first tap event. The interrupt is cleared after a delay of 12.5ms in non-latched mode.

The sign of the slope of the first tap which triggered the interrupt is stored in

bit (0x0B) 'TAP_SIGN' (0 means positive, 1 means negative). The axis which triggered the interrupt is indicated by bits (0x0B) 'TAP_FIRST_X/Y/Z'.

### 4.5.7 Orientation recognition

The orientation recognition feature informs on an orientation change of sensor with respect to the gravitation field vector 'g'. The measured acceleration vector components with respect to the gravitation field are defined as shown in the following figure.

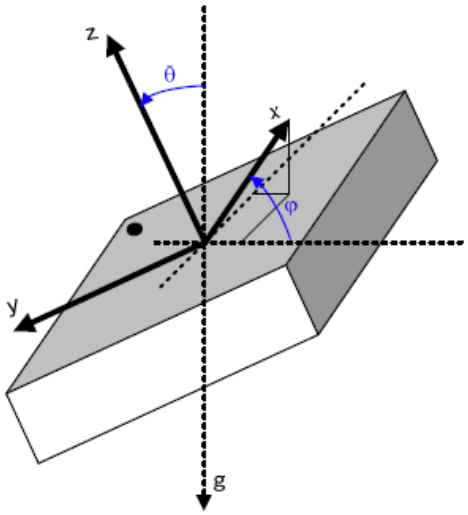



Figure 5: Definition of vector components

Therefore, the magnitudes of the acceleration vectors are calculated as follows:

$$acc\_x = 1g.\sin\theta\cdot\cos\varphi$$
$$acc\_y = -1g.\sin\theta\cdot\sin\varphi$$
$$acc\_z = 1g\cdot\cos\theta$$

Depending on the magnitudes of the acceleration vectors the orientation of the device in the space is determined and stored in the bits (0x0C) 'ORIENT'. There are three orientation calculation modes with different t thresholds for switching between different orientations: symmetrical, high-asymmetrical and low-asymmetrical. The mode is selected by setting the (0x2C) 'ORIENT_MODE' bits. For each orientation mode, the 'ORIENT' bits have a different meaning as show in below table.

Table 8: meaning of 'orient' bits in symmetric mode

| Orient | Name | Angle | Condition |
|---|---|---|---|
| X00 | Portrait upright | $315^o < \varphi < 45^o$ | \|acc_y\| < \|acc_x\| - 'hyst' & acc_x >= 0 |
| X01 | Portrait upside down | $135^o < \varphi < 225^o$ | \|acc_y\| < \|acc_x\| - 'hyst' & acc_x < 0 |
| X10 | Landscape left | $45^o < \varphi < 135^o$ | \|acc_y\| >= \|acc_x\| + 'hyst' & acc_y < 0 |
| X11 | Landscape right | $225^o < \varphi < 315^o$ | \|acc_y\| >= \|acc_x\| + 'hyst' & acc_y >= 0 |

Table 9: meaning of 'orient' bits in high-asymmetric mode

| Orient | Name | Angle | Condition |
|---|---|---|---|
| X00 | Portrait upright | $297^o < \varphi < 63^o$ | \|acc_y\|< 2*\|acc_x\| - 'hyst' & acc_x >= 0 |
| X01 | Portrait upside down | $117^o < \varphi < 243^o$ | \|acc_y\| < 2*\|acc_x\| - 'hyst' & acc_x < 0 |
| X10 | Landscape left | $63^o < \varphi < 117^o$ | \|acc_y\| >= 2*\|acc_x\| + 'hyst' & acc_y < 0 |
| X11 | Landscape right | $243^o < \varphi < 297^o$ | \|acc_y\| >= 2*\|acc_x\| + 'hyst' & acc_y >= 0 |

Table 10: meaning of 'orient' bits in low-asymmetric mode

| Orient | Name | Angle | Condition |
|---|---|---|---|
| X00 | Portrait upright | $333^o < \varphi < 27^o$ | \|acc_y\|< 0.5*\|acc_x\| - 'hyst' & acc_x >= 0 |
| X01 | Portrait upside down | $153^o < \varphi < 207^o$ | \|acc_y\| < 0.5*\|acc_x\| - 'hyst' & acc_x < 0 |
| X10 | Landscape left | $27^o < \varphi < 153^o$ | \|acc_y\| >= 0.5*\|acc_x\| + 'hyst' & acc_y < 0 |
| X11 | Landscape right | $207^o < \varphi < 333^o$ | \|acc_y\| >= 0.5*\|acc_x\| + 'hyst' & acc_y >= 0 |

In the preceding tables, the parameter 'HYST' stands for a hysteresis which can be selected by bits (0x2C) 'ORIENT_HYST'. 1LSB of (0x2C) 'ORIENT_HYST' always corresponds to 62.5mg in any g-range. The MSB of 'orient' bits contains information about the direction of the z-axis. It is set to 0(1) if acc_z >=0 (acc_z<0). The hysteresis for z axis is fixed to 0.2g.

The orient interrupt is enabled by writing (0x16) 'ORIENT_EN' bit. The interrupt is generated if the value of 'orient' has changed. It is automatically cleared after one stable period of the orient value in non-latched mode. In temporary latched or latched mode, the orient value is kept fixed as long as the interrupt persists. After cleaning the interrupt, the 'ORIENT' will updated with the next following value change.

The change of the 'ORIENT' value and the generation of the interrupt can be blocked according to conditions selected by setting the value of bits (0x2C) 'ORIENT_BLOCKING' as described by below table.

Table 11: blocking conditions for orientation recognition

| Orient_blocking | Conditions |
|---|---|
| 00b | No blocking |
| 01b | Z blocking |
| 10b | Z blocking or acceleration slope in any axis > 0.2g |
| 11b | No blocking |

The Z blocking is defined by the following inequality：

$$|acc\_z| > z\_blocking$$

The parameter z_blocking of the above given equation stands for the contents of the 'z_blocking' bits. Hereby it is possible to define a blocking value between 0g and 0.9375g with an LSB = 0.0625g.

### 4.5.8 Freefall interrupt

This interrupt is based on the comparison of acceleration data against a low-g threshold. The interrupt is enabled by writing 1 to the bit (0x17) 'FREEFALL_INT_EN'. There are two modes available: single mode and sum mode. In single mode the acceleration of each axis is compared with the threshold. In sum mode, the sum of absolute values of all accelerations |acc_x| + |acc_y| + |acc_z| is compared with the threshold. The mode is selected by the bit (0x24) 'FREEFALL_MODE'. The freefall threshold is set through the (0x23) 'FREEFALL_TH' bits with 1 LSB corresponding to an acceleration of 7.81mg. A hysteresis can be selected by setting the 'FREEFALL_HY' bits with 1 LSB corresponding to 125mg.

The freefall interrupt is generated if the absolute values of the acceleration of all axes or their sum are lower than the threshold for at least the time defined by (0x22) 'FREEFALL_DUR' bits. The interrupt is reset if the absolute value of at least one axis or the sum is higher than the threshold plus the hysteresis for at least one data acquisition. The interrupt status is stored in bit (0x09) 'FREEFALL_INT'.

# 5. FIFO Operation

## *5.1 FIFO Operating Modes*

The MSA310 features an integrated FIFO memory capable of storing up to 64 samples. Conceptually each sample consists of three 16 bit words corresponding to the x, y and z- axis, which are sampled at the same point in time. At the core of the FIFO is a buffer memory, which can be configured to operate in the following modes:

**BYPASS Mode**: In bypass mode, only the current sensor data can be read out from the FIFO address. Essentially, the FIFO behaves like the STREAM mode with a depth of 1. Compared to reading the data from the normal data registers, the advantage to the user is that the packages X, Y, Z are from the same timestamp, while the data registers are updated sequentially and hence mixing of data from different axes can occur.

**FIFO Mode**: In FIFO mode the acceleration data of the selected axes are stored in the buffer memory. If enabled, a watermark interrupt is triggered when the buffer has filled up to a configurable level. The buffer will be continuously filled until the fill level reaches 64 samples. When it is full the data collection is stopped, and all additional samples are ignored. Once the buffer is full, a FIFO-full interrupt is generated if it has been enabled.

**STREAM Mode**: In STREAM mode the acceleration data of the selected axes are stored in the buffer until it is full. The buffer has a depth of 64 samples. When the buffer is full the data collection continues and oldest entry is discarded. If enabled, a watermark interrupt is triggered when the buffer is filled to a configurable level. Once the buffer is full, a FIFO-full interrupt is generated if it has been enabled.

**Trigger Mode**: In trigger mode, FIFO accumulates samples, holding the latest 64 samples from measurements of the x, y, and z-axes. After a trigger event occurs and an interrupt is sent to the INT1 pin, FIFO keeps the last n samples (where n is the value specified by the 'watermark_samples' bits) and

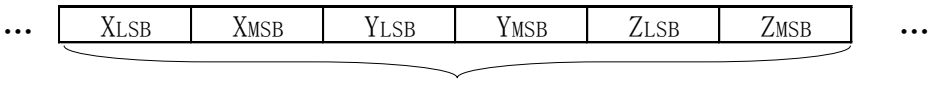
then operates in FIFO mode, collecting new samples only when FIFO is not full. Additional trigger events cannot be recognized until the trigger mode is reset. To reset the trigger mode, set the device to bypass mode and then set the device back to trigger mode. Note that the FIFO data should be read first because placing the device into bypass mode clears FIFO.

The primary FIFO operating mode is selected with register (0x14) fifo_mode according to '00b' for BYPASS mode, '01b' for FIFO mode, '10b' for STREAM mode, and '11b' for Trigger mode. Writing to register (0x14) clears the buffer content and resets the FIFO-full and watermark interupts. When reading register (0x14) fifo_mode always contains the current operating mode.

### *5.2 FIFO Data Readout*

The FIFO stores the data that are also available at the acceleration read-out registers (0x02) to (0x07). Thus, all configuration settings apply to the FIFO data as well as the acceleration data readout registers. The FIFO read out is possible through register (0xFF). The readout can be performed using burst mode since the read address counter is no longer incremented, when it has reached address (0xFF). This implies that the trapping also occurs when the burst read access starts below address (0xFF). A single burst can read out one or more samples at a time. Register (0x15) FIFO_data_select controls the acceleration data of which axes are stored in the FIFO. Possible settings for register (0x15) FIFO_data_select are '00b' for x, y and z-axis, '01b' for x-axis only, '10b' for y-axis only, '11b' for z-axis only. The depth of the FIFO is independent of whether all or a single axis have been selected. Writing to register (0x14) clears the buffer content and resets the FIFO-full and watermark interrupts.

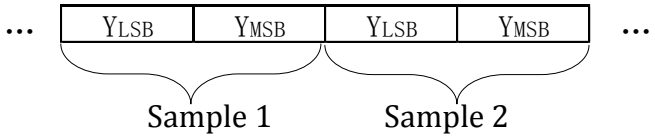
If all axes are enabled, the fomat of the data read-out from register (0xFF) is as follows:




Sample 1

If only one axis is enabled, the format of the data read-out from register (0xFF)

is as follows (example showm: y-axis only, other axes are equivalent).




If a sample is not completely read due to an incomplete read operation, the remaining part of the sample is discarded. In this case the FIFO aligns to the next sample during the next read operation. In order for the discarding mechanism to operate correctly, there must be a delay of at least 1.5 us between the last data bit of the partially read sample and the first address bit of the next FIFO read access. Othenwise samples must not be read out partially.

If the FIFO is read beyond the FIFO fill level zeroes (0) will be read. If the FIFO is read beyond the FIFO fill level the read or burst read access time must not exceed the sampling time $t_{SAMPLE}$. Otherwise samples may be lost.

### *5.3 FIFO Sample Counter and Overrun Flag*

Register (0x0D) FIFO_entries reflects the current fill level of the buffer. If additional samples are witten to the buffer although the FIFO is full, the (0x0A) Overrun_int bit is set to '1'. The FIFO buffer is cleared, the FIFO fill level indicated in register (0x0D) FIFO_entries and the (0x0A) Overrun_int bit are both set to '0' each time one a write access to one of the FIFO configuration registers (0x14) or (0x15) occurs. The (0x0A) Overrun_int bit is not reset when the FIFO fill level (0x0 D) FIFO_entries has decremented to '0' due to reading from register (0xFF).

### *5.4 FIFO Interrupts*

The FIFO controller can generate two different interrupt events, a FIFO-full and a watermark event. The FIFO-full and watermark interrupts are functional in all FIFO operating modes. The watermark interrupt is asserted when the fill level in the buffer has reached the sample count defined by register (0x14) watermark_samples. In order to enable (disable) the watermark interrupt, the (0x17)

watermark_int_en bit must be set to '1' ('0'). To map the watermark interrupt signal to INT1 pin, (0x1A) int_Watermark (bit must be set to '1'. The status of the watermark interupt may be read back through the (0x0A) Watermark_int bit. Writing to register (0x14) watermark_samples clears the FIFO buffer.

The FiFO-full interrupt is triggered when the buffer has been completely filled. In FIFO mode this occurs 64, in STREAM mode 63 samples, and in BYPASS mode 1 sample after the buffer has been cleared. In order to enable the FIFO-full interrupt, bit (0x17) overrun_int_en as well as bit (0x1A) int_overrun must also be set to '1'. The status of the FIFO-full interrupt may be read back through bit (0x0A) Overrun_int.

# 6. Register Map

| Address | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|
| 0x00 | | | SOFT RESET | | | SOFT RESET | | | 0x00 |
| 0x01 | PARTID[7:0] | | | | | | | | 0x13 |
| 0x02 | ACC_X[3:0] | | | | | | | | 0x00 |
| 0x03 | ACC_X[11:4] | | | | | | | | 0x00 |
| 0x04 | ACC_Y[3:0] | | | | | | | | 0x00 |
| 0x05 | ACC_Y[11:4] | | | | | | | | 0x00 |
| 0x06 | ACC_Z[3:0] | | | | | | | | 0x00 |
| 0x07 | ACC_Z[11:4] | | | | | | | | 0x00 |
| 0x09 | | ORIENT_INT | S_TAP_INT | D_TAP_INT | | ACTIVE_INT | | FREEFALL_INT | 0x00 |
| 0x0A | | | | | | OVERRUN_INT | WATERMARK_INT | NEW_DATA_INT | 0x00 |
| 0x0B | TAP_SIGN | TAP_FIST_X | TAP_FIRST_Y | TAP_FIRST_Z | ACTIVE_SIGN | ACTIVE_FIST_X | ACTIVE_FIST_Y | ACTIVE_FIST_Z | 0x00 |
| 0x0C | | ORIENT[2:0] | | | | | | | 0x00 |
| 0x0D | | FIFO_ENTRIES[6:0] | | | | | | | 0x00 |
| 0x0F | | | | | | | FS[1:0] | | 0x00 |
| 0x10 | X-AXIS_DIS | Y_AXIS_DIS | Z_AXIS_DIS | | ODR[3:0] | | | | 0x0F |
| 0x11 | PWR_MODE | | SLEEPTIMER_MODE | LOW_POWER_BW[3:0] | | | | | 0xDE |
| 0x12 | | | | | X_POLARITY | Y_POLARITY | Z_POLARITY | X_Y_SWAP | 0x06 |
| 0x14 | FIFO_MODE[1:0] | | WATERMARK_SAMPLES[5:0] | | | | | | 0x20 |
| 0x15 | | | | | | | FIFO_data_select[1:0] | | 0x00 |
| 0x16 | | ORIENT_INT_EN | S_TAP_INT_EN | D_TAP_INT_EN | | ACTIVE_INT_EN_Z | ACTIVE_INT_EN_Y | ACTIVE_INT_EN_X | 0x00 |
| 0x17 | | OVERRUN_INT_EN | WATERMARK_INT_EN | NEW_DATA_INT_EN | FREEFALL_INT_EN | | | | 0x00 |
| 0x19 | | INT1_ORIENT | INT1_S_TAP | INT1_D_TAP | | INT1_ACTIVE | | INT1_FREEFALL | 0x00 |
| 0x1A | | | | | | INT1_OVERRUN | INT_WATERMARK | INT1_NEW_DATA | 0x00 |
| 0x20 | | | | | | | INT1_OD | INT1_LVL | 0x00 |
| 0x21 | RESET_INT | | | | LATCH_INT[3:0] | | | | 0x00 |
| 0x22 | FREEFALL_DUR[7:0] | | | | | | | | 0x09 |
| 0x23 | FREEFALL_TH[7:0] | | | | | | | | 0x30 |
| 0x24 | | | | | | FREEFALL_MODE | FREEFALL_HY[1:0] | | 0x01 |
| 0x27 | | | | | | | ACTIVE_DUR[1:0] | | 0x00 |
| 0x28 | ACTIVE_TH | | | | | | | | 0x0A |
| 0x2A | TAP_QUIET | TAP_SHOCK | | | | TAP_DUR[2:0] | | | 0x04 |
| 0x2B | | | | TAP_TH[4:0] | | | | | 0x0A |
| 0x2C | | ORIENT_HYST[2:0] | | | ORIENT_BLOCKING[1:0] | | ORIENT_MODE[1:0] | | 0x18 |
| 0x2D | | | | | Z_BLOCKING[3:0] | | | | 0x08 |
| 0x38 | OFFSET_X[7:0] | | | | | | | | 0x00 |
| 0x39 | OFFSET_Y[7:0] | | | | | | | | 0x00 |
| 0x3A | OFFSET_Z[7:0] | | | | | | | | 0x00 |
| 0xFF | FIFO_DATA_OUT[7:0] | | | | | | | | 0x00 |

# 7. Register Description

### *Register 0x00(Soft Reset) :Write only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | Soft Reset | | | Soft Reset | | | 0x00 |

Soft Reset: 0:soft reset disable, 1:soft reset enable

### *Reg 0x01(PartID):Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| PARTID[7:0] | | | | | | | | 0x13 |

### *Reg 0x02/0x03(X_axis Data LSB/MSB) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| ACC_X[3:0] | | | | | | | | 0x00 |
| ACC_X[11:4] | | | | | | | | 0x00 |

### *Reg 0x04/0x05(Y_axis Data LSB/MSB) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| ACC_Y[3:0] | | | | | | | | 0x00 |
| ACC_Y[11:4] | | | | | | | | 0x00 |

### *Reg 0x06/0x07(Z_axis Data LSB/MSB) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| ACC_Z[3:0] | | | | | | | | 0x00 |
| ACC_Z[11:4] | | | | | | | | 0x00 |

### *Reg 0x09(Motion_Interrupt) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | ORIENT_INT | S_TAP_INT | D_TAP_INT | | ACTIVE_INT | | FREEFALL_INT | 0x00 |

ORIENT_INT: orientation interrupt status, 0: inactive, 1: active

S_TAP_INT: single tap interrupt status, 0: inactive, 1: active

D_TAP_INT: double tap interrupt status, 0: inactive, 1: active

ACTIVE_INT: active interrupt status, 0: inactive, 1: active

FREEFALL_INT: freefall interrupt status, 0: inactive, 1: active

### *Reg 0x0A(Data_Interrupt) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | OVERRUN_INT | WATERMARK_INT | NEW_DATA_INT | 0x00 |

OVERRUN_INT: FIFO Overrun interrupt status, 0: inactive,1: active

WATERMARK_INT: FIFO Watermark interrupt status, 0: inactive,1: active

NEW_DATA_INT: new data interrupt status, 0: inactive,1: active

### *Reg 0x0B(Tap_Active_Status) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| TAP_SIGN | TAP_FIRST_X | TAP_FIRST_Y | TAP_FIRST_Z | ACTIVE_SIGN | ACTIVE_FIRST_X | ACTIVE_FIRST_Y | ACTIVE_FIRST_Z | 0x00 |

TAP_SIGN: sign of tap triggering signal, 0 : positive, 1: negative

TAP_FIRST_X: tap interrupt triggered by x axis, 1: positive, 0: negative

TAP_FIRST_Y: tap interrupt triggered by y axis, 1: positive, 0: negative

TAP_FIRST_Z: tap interrupt triggered by z axis, 1: positive, 0: negative

ACTIVE_SIGN: sign of active interrupt, 0: positive, 1: negative

ACTIVE_FIRST_X: active interrupt triggered by x axis, 1: positive, 0: negative

ACTIVE_FIRST_Y: active interrupt triggered by y axis, 1: positive, 0: negative

ACTIVE_FIRST_Z: active interrupt triggered by z axis, 1: positive, 0: negative

### *Reg 0x0C(Orientation _Status) :Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | ORIENT[2:0] | | | | | | | 0x00 |

ORIENT[2]: orientation value of z axis, 0:upward looking, 1:downward looking

ORIENT[1:0]: orientation value of x/y axes

00: portrait upright

01: portrait upside down

10: landscape left,

11: landscape right

### *Reg 0x0D(FIFO status)： Read only*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| | FIFO_ENTRIES[6:0] | 0x00 |

FIFO_ENTRIES[6:0]: reports how many data stored in the FIFO

### *Reg 0x0F(Range)：Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | | FS[1:0] | | 0x00 |

FS[1:0]: acceleration range of x/y/z axes, 00:+/-2g, 01:+/-4g, 10:+/-8g, 11:+/-16g

### *Reg0x10(ODR) ：Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| X_AXIS_DIS | Y_AXIS_DIS | Z_AXIS_DIS | | ODR[3:0] | | | | 0x0F |

X_AXIS_DIS: 0: enable, 1: disable

Y_AXIS_DIS: 0: enable, 1: disable

Z_AXIS_DIS: 0: enable, 1: disable

ODR[3:0]: 0000: 1Hz (not available in normal mode)

0001: 1.95Hz (not available in normal mode)

0010: 3.9Hz

0011: 7.81Hz

0100: 15.63Hz

0101: 31.25Hz

0110: 62.5Hz

0111: 125Hz

1000: 250Hz

1001: 500Hz (not available in low power mode)

1010-1111:1000Hz (not available in low power mode)

### *Reg0x11(Power Mode/Bandwidth) ：Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| PWR_MODE | | SLEEPTIMER_MODE | LOW_POWER_BW[3:0] | | | | | 0xDE |

PWR_MODE: 00: normal mode, 01: low power mode, 11: suspend mode

SLEEPTIMER_MODE: 0: event-driven mode, 1: equidistant-sampling mode

LOW_POWER_BW[3:0]:

0000-0010:1.95Hz

0011:3.9Hz

0100:7.81Hz

0101:15.63Hz

0110: 31.25Hz

0111: 62.5Hz

1000: 125Hz

1001: 250Hz

1010-1111:500Hz

### *Reg 0x12(Swap_Polarity) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | X_POLARITY | Y_POLARITY | Z_POLARITY | X_Y_SWAP | 0x06 |

X_POLARITY: the polarity of X axis, 0: not reverse, 1: reverse

Y_POLARITY: the polarity of Y axis, 0: not reverse, 1: reverse

Z_POLARITY: the polarity of Z axis, 0: not reverse, 1: reverse

X_Y_SWAP: output of X/Y axis, 0: not swap, 1: swap

### *Reg 0x14(FIFO_Config_0) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| FIFO_MODE[1:0] | | WATERMARK_SAMPLES[5:0] | | | | | | 0x20 |

FIFO_MODE[1:0]: 00: bypass mode, 01: FIFO mode, 10: stream mode, 11: trigger mode

WATERMARK_SAMPLES[5:0]: indicate how many data entries are needed to trig a watermark interrupt.

### *Reg 0x15(FIFO_Config_1) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | | FIFO_DATA_SELECT[1:0] | | 0x00 |

FIFO_DATA_SELECT[1:0]: select acceleration data stored in FIFO 00: X+Y+Z 01: X only 10: Y only 11: Z only

### *Reg 0x16(Int_Set_0) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | ORIENT_INT_EN | S_TAP_INT_EN | D_TAP_INT_EN | | ACTIVE_INT_EN_Z | ACTIVE_INT_EN_Y | ACTIVE_INT_EN_X | 0x00 |

ORIENT_INT_EN: orient interrupt, 0: disable, 1: enable

S_TAP_INT_EN: single tap interrupt, 0: disable, 1: enable

D_TAP_INT_EN: double tap interrupt, 0: disable, 1: enable

ACTIVE_INT_EN_Z: active interrupt for the z axis, 0: disable, 1: enable

ACTIVE_INT_EN_Y: active interrupt for the y axis, 0: disable, 1: enable

ACTIVE_INT_EN_X: active interrupt for the x axis, 0: disable, 1: enable

### *Reg 0x17(Int_Set_1) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | OVERRUN_INT_EN | WATERMARK_INT_EN | NEW_DATA_INT_EN | FREEFALL_INT_EN | | | | 0x00 |

OVERRUN_INT_EN: overrun interrupt, 0: disable, 1: enable

WATERMARK_INT_EN: watermark interrupt, 0: disable, 1: enable

NEW_DATA_INT_EN: new data interrupt, 0: disable, 1: enable

FREEFALL_INT_EN: freefall interrupt, 0: disable, 1: enable

### *Reg 0x19(Int_Map_0) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | INT1_ORIENT | INT1_S_TAP | INT1_D_TAP | | INT1_ACTIVE | | INT1_FREEFALL | 0x00 |

INT1_ORIENT: map orientation interrupt to INT1, 0: disable, 1: enable

INT1_S_TAP: map single tap interrupt to INT1, 0: disable, 1: enable

INT1_D_TAP: map double tap interrupt to INT1, 0: disable, 1: enable

INT1_ACTIVE: map active interrupt to INT1, 0: disable, 1: enable

INT1_FREEFALL: map freefall interrupt to INT1, 0: disable, 1: enable

### *Reg 0x1A(Int_Map_1) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | INT1_OVERRUN | INT1_WATERMARK | INT1_NEW_DATA | 0x00 |

INT1_OVERRUN: map overrun interrupt to INT1, 0: disable, 1: enable

INT1_WATERMARK: map watermark interrupt to INT1, 0: disable, 1: enable

INT1_NEW_DATA: map new data interrupt to INT1, 0: disable, 1: enable

### *Reg 0x20 (Int_Config) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|

| | | | |
|---|---|---|---|
| | INT1_OD | INT1_LVL | 0x00 |

INT1_OD: select output for INT1, 0: push-pull, 1: open-drain

INT1_LVL: select active level for INT1, 0: low, 1: high

### *Reg 0x21 (Int_Latch) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| RESET_INT | | | | LATCH_INT[3:0] | | | | 0x00 |

RESET_INT: reset or not, 1: reset all latched interrupt, 0: not reset all latched interrupt

LATCH_INT[3:0]: 0000: non-latched

0001: temporary latched 250ms

0010: temporary latched 500ms

0011: temporary latched 1s

0100: temporary latched 2s

0101: temporary latched 4s

0110: temporary latched 8s

0111: latched

1000: non-latched

1001: temporary latched 1ms

1010: temporary latched 1ms

1011: temporary latched 2ms

1100: temporary latched 25ms

1101: temporary latched 50ms

1110: temporary latched 100ms

1111: latched

### *Reg 0x22 (Freefall_Dur) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| FREEFALL_DUR[7:0] | | | | | | | | 0x09 |

FREEFALL_DUR[7:0]: delay_time is (FREEFALL_DUR[7:0] +1) ×2ms, range from 2ms to 512ms, the default delay time is 20ms

### *Reg 0x23 (Freefall_Th) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| FREEFALL_TH[7:0] | | | | | | | | 0x30 |

FREEFALL_TH[7:0]: threshold value is freefall_th[7:0] ×7.81mg, default value is 375mg

### *Reg 0x24 (Freefall_Hy) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | FREEFALL_MODE | FREEFALL_HY[1:0] | | 0x01 |

FREEFALL_MODE: 0: single mode, 1: sum_mode

FREEFALL_HY[1:0]: freefall hysteresis time is FREEFALL_HY[1:0] ×125mg

### *Reg 0x27 (Active_Dur) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ACTIVE_DUR[1:0] | | 0x00 |

ACTIVE_DUR[1:0]: active duration time is (ACTIVE_DUR[1:0]+1)ms

### *Reg 0x28(Active_Th) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| ACTIVE_TH[7:0] | | | | | | | | 0x0A |

ACTIVE_TH[7:0]: threshold of active interrupt

3.91mg/LSB (2g range)

7.81mg/LSB (4g range)

15.625mg/LSB (8g range)

31.25mg/LSB (16g range)

### *Reg 0x2A(Tap_Dur) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| TAP_QUIET | TAP_SHOCK | | | | TAP_DUR[2:0] | | | 0x04 |

TAP_QUIET: 0: tap quiet duration 30ms, 1: tap quiet duration 20ms

TAP_SHOCK: 0: tap shock duration 50ms, 1: tap shock duration 70ms

TAP_DUR[2:0]: selects the length of the time window for the second shock

000: 50ms

001:100ms

010:150ms

011:200ms

100:250ms

101:375ms

110:500ms

111:700ms

### *Reg 0x2B(Tap_Th) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | TAP_Th[4:0] | | | | | 0x0A |

TAP_TH[4:0]: threshold of tap interrupt.

62.5mg/LSB(2g range)

125mg/LSB(4g range)

250mg/LSB(8g range)

500mg/LSB(16g range)

### *Reg 0x2C(Orient_Hy) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | ORIENT_HYST[2:0] | | | ORIENT_BLOCKING[1:0] | | ORIENT_MODE[1:0] | | 0x18 |

ORIENT_HYST[2:0]: set the hysteresis of the orientation interrupt, 1LSB is 62.5mg.

ORIENT_BLOCKING[1:0]: select the block mode

00: no blocking

01: z_axis blocking

10: z_axis blocking or slope in any axis > 0.2g

11: no blocking

ORIENT_MODE[1:0]: set the thresholds

00: symmetrical

01: high-asymmetrical

10: low-asymmetrical

11: symmetrical

### *Reg 0x2D(Z_Block) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| | | | | Z_BLOCKING[3:0] | | | | 0x08 |

Z_BLOCKING[3:0]: defines the block acc_z between 0g to 0.9375g

### *Reg 0x38/0x39/0x3A (Offset_compensation) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| OFFSET_X[7:0] | | | | | | | | 0x00 |

| | |
|---|---|
| OFFSET_Y[7:0] | 0x00 |
| OFFSET_Z[7:0] | 0x00 |

OFFSET_X[7:0]: the offset compensation value for X axis, 1LSB is 3.9mg

OFFSET_Y[7:0]: the offset compensation value for Y axis, 1LSB is 3.9mg

OFFSET_Z[7:0]: the offset compensation value for Z axis, 1LSB is 3.9mg

### *Reg 0xFF (FIFO_DATA) ： Read/Write*

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Default |
|---|---|---|---|---|---|---|---|---|
| FIFO_DATA_OUT[7:0] | | | | | | | | 0x00 |

FIFO_DATA_OUT[7:0]: FIFO data output register, return 0 if FIFO is empty

# 8. Digital Interface and Application Collection

The MSA310 supports $I^2C$ digital interface protocols for communications as slave with a host device.

## *8.1 I2C Interface Specification*

I2C bus uses SCL and SDA as signal lines. Both lines are connected to VDD externally via pull-up resistors so that they are pulled high when the bus is free.

The 7-bits I2C device address of MSA310 is shown as following table.

Table 12: $I^2C$ Address.

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | W/R |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0/1 |

Table 13: Electrical specification of the I2C interface pins

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $f_{scl}$ | Clock frequency | | | 400 | kHz |
| $t_{scl\_l}$ | SCL low pulse | | 1.3 | | us |
| $t_{scl\_h}$ | SCL high pulse | | 0.6 | | us |
| $T_{sda\_setup}$ | SDA setup time | | 0.1 | | us |
| $T_{sda\_hold}$ | SDA hold time | | 0.0 | | us |
| $t_{susta}$ | Setup Time for a repeated start condition | | 0.6 | | us |
| $t_{hdsta}$ | Hold time for a start condition | | 0.6 | | us |
| $t_{susto}$ | Setup Time for a stop condition | | 0.6 | | us |
| $t_{buf}$ | Time before a new | | 1.3 | | us |

| | | | | | |
|---|---|---|---|---|---|
| | transmission can start | | | | |

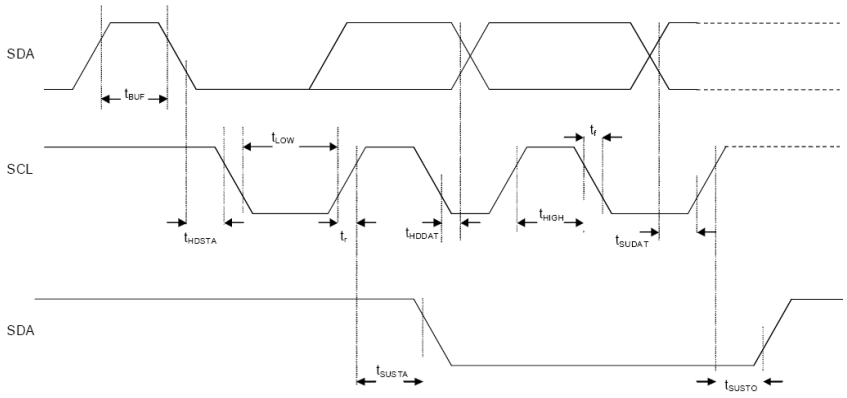



Figure 6: I2C Timing Diagram

The I2C interface protocol has special bus signal conditions. Start (S), stop (P) and binary data conditions are shown below. At start condition, SCL is high and SDA has a falling edge. Then the slave address is sent. After the 7 address bits, the direction control bit R/W selects the read or write operation. When a slave device recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle.

At stop condition, SCL is also high, but SDA has a rising edge. Data must be held stable at SDA when SCL is high. Data can change value at SDA only when SCL is low.

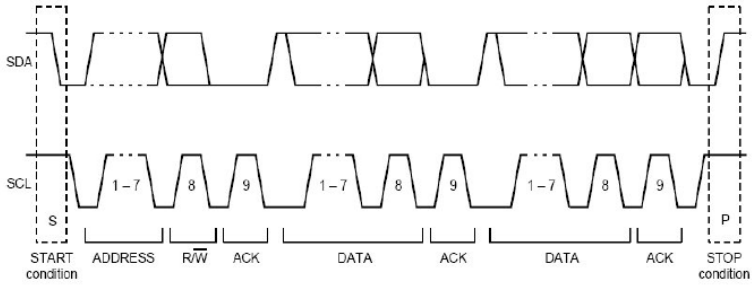



Figure7: I2C Protocol

## *8.2 Application collection*

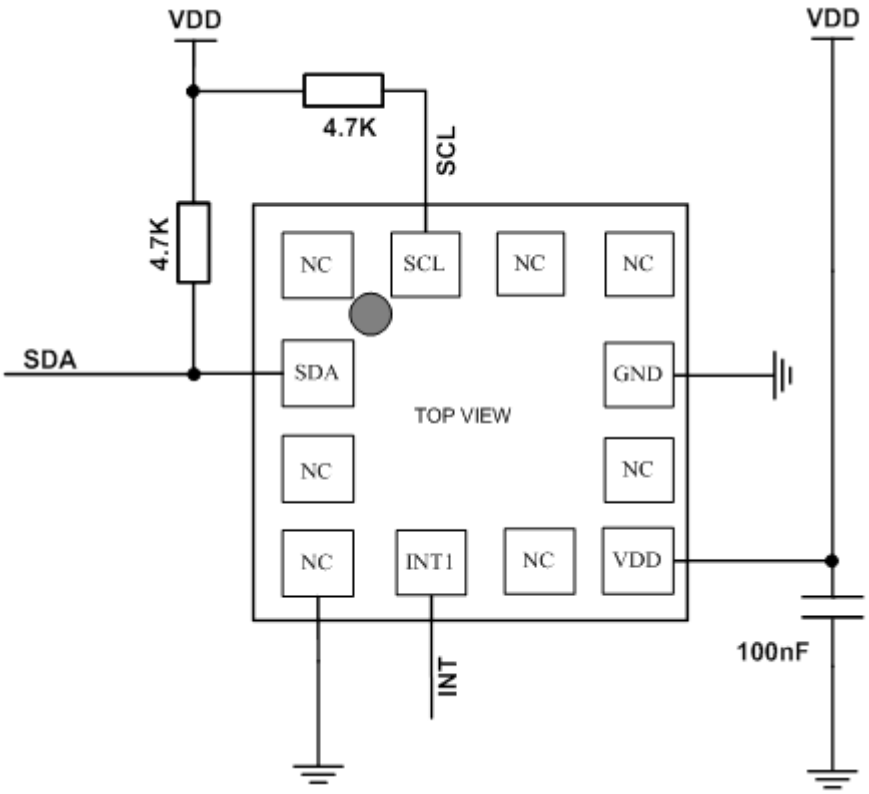
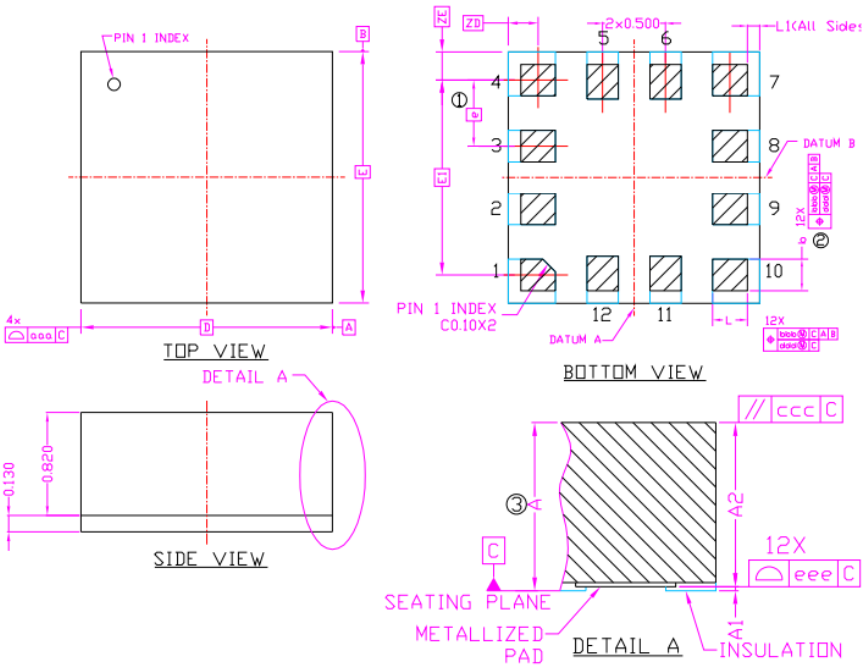



Figure 8: I²C application electrical collection

# 9. Package Description

## *9.1 Outline Dimensions*




DIMENSIONAL REFERENCES unit: mm

| REF. | Min. | Nom. | Max. |
|---|---|---|---|
| A | 0.90 | 0.95 | 1.00 |
| A1 | - | - | 0.05 |
| A2 | - | - | 0.95 |
| b | 0.20 | 0.25 | 0.30 |
| L | 0.225 | 0.275 | 0.325 |
| D | 1.95 | 2.00 | 2.05 |
| E | 1.95 | 2.00 | 2.05 |
| E1 | 1.55 BSC | | |
| ZD | 0.2375 BSC | | |
| ZE | 0.225 BSC | | |
| e | 0.525 BSC | | |
| L1 | 0.05 | 0.10 | 0.15 |

Figure 9: Package Pin Dimensions

## *9.2 Sensor orientation*


Figure 10: Orientation of sensing axis

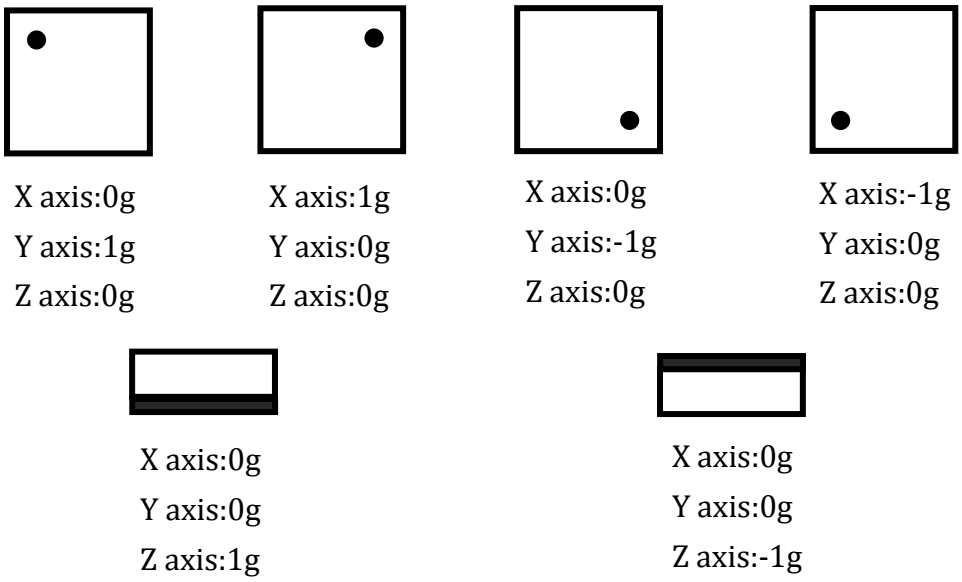


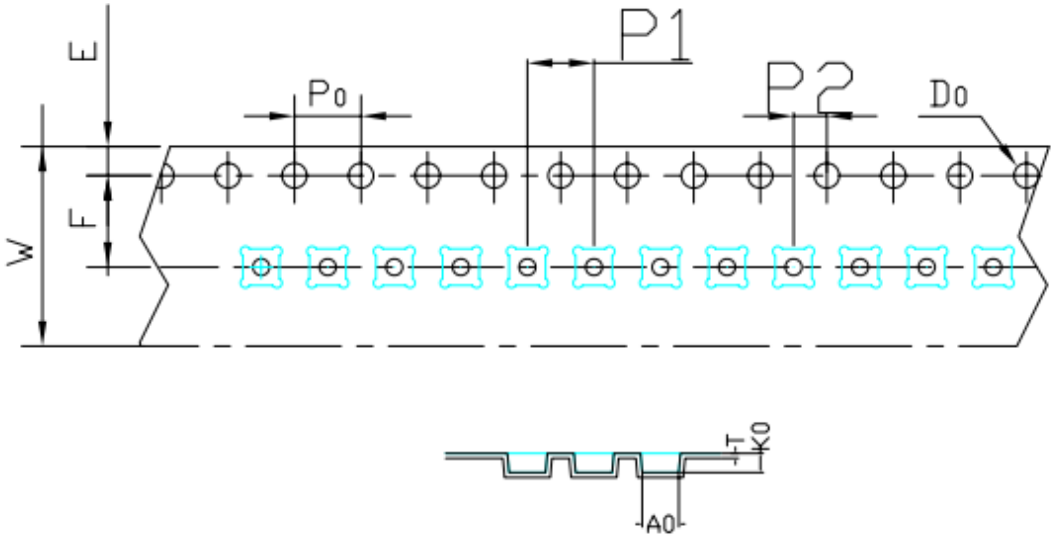

Figure 11: Output signal of sensing axis orientation

### *9.3 Tape and reel*

The MSA310 is shipped in a standard pizza box.

The box dimension for 1 reel is: L*W*H=35cm*35cm*5cm.

MSA310 quantity: 5000pcs per reel, please handle with care.




| | | | |
|---|---|---|---|
| | | W | 12.0±0.30 |
| A | 2.20±0.10 | E | 1.75±0.10 |
| B | 2.20±0.10 | F | 5.50±0.10 |
| K | 1.30±0.10 | P0 | 4.00±0.10 |
| D | 1.50 +0.10 −0.00 | P1 | 4.00±0.10 |
| T | 0.30±0.05 | P2 | 2.00±0.10 |

Figure12: Tape and reel dimension in mm

### *9.4 Reflow Profile*

| Profile Feature | Pb-Free Assembly |
|---|---|
| Average Ramp-Up Rate($Ts_{max}$ to Tp) | 3°C/second max |
| **Preheat** | |
| Temperature Min($Ts_{min}$) | 150°C |
| Temperature Max($Ts_{max}$) | 200°C |
| Time($ts_{min}$ to $ts_{max}$) | 60-180 seconds |
| **Time maintained above:** | |
| Temperature($T_L$) | 217°C |
| Time($t_L$) | 60-150 seconds |
| Peak/Classification Temperature($T_P$) | 260°C |
| Time within 5°C of actual Peak Temperature($t_P$) | 20-40 seconds |
| Ramp-Down Rate | 6°C/second max |
| Time 25°C to Peak Temperature | 8 minutes max |

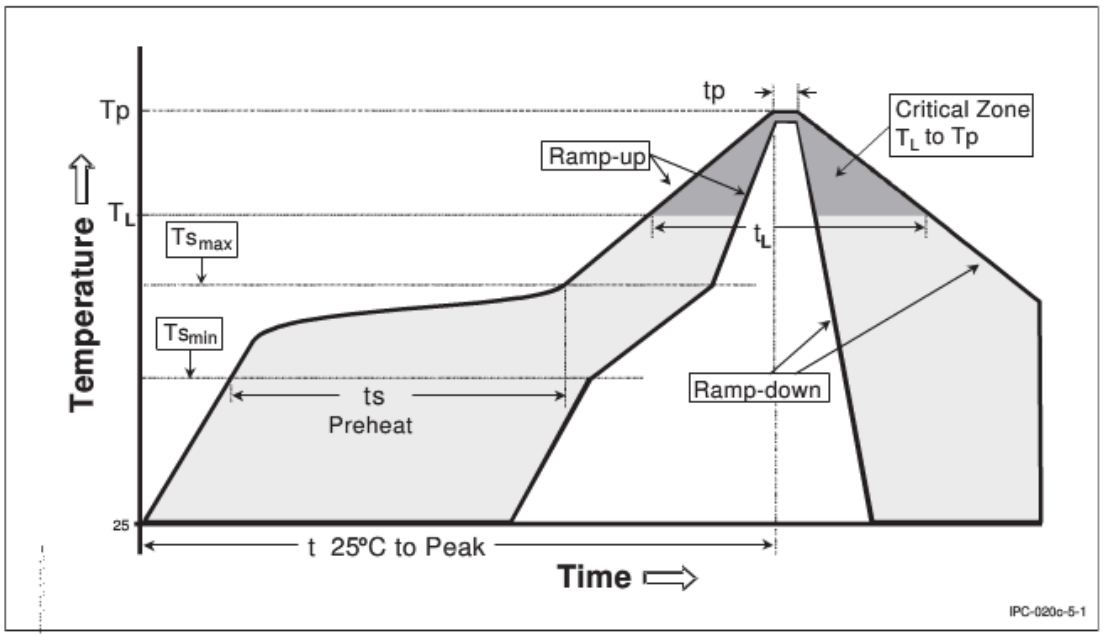



Figure 13: Reflow Profile

# Revision History

| Revision | Chapter | Subjects (major changes since last revision) | Date |
|---|---|---|---|
| 1.0 | - | Initial Release | 2020-09-30 |
| 1.1 | 6,7 | Update 0x12 register default value | 2021-03-04 |
| | | | |
| | | | |
| | | | |



| 公司销售、技术支持联系方式 | (http://www.memsensing.com) |
|---|---|
| *MEMSensing Microsystems(Suzhou, China)* Co., Ltd.<br>No. 99 Jinji Lake Avenue, Bldg. NW-09, Suite 501, Suzhou Industrial Park, China 215123<br>Phone: +86 512 62956055<br>Fax: +86 512 62956056 | 苏州敏芯微电子技术股份有限公司<br>苏州工业园区金鸡湖大道 99 号，NW-09 楼，501 室<br>中国 215123<br>电话: +86 512 62956055<br>传真: +86 512 62956056 |