## Abstract
## Single-Chip 3-Axis Accelerometer
## QMA7981

*Advanced Information*

The QMA7981 is a single chip three-axis accelerometer. This surface-mount, small sized chip has integrated acceleration transducer with signal conditioning ASIC, sensing tilt, motion, shock and vibration, targeted for applications such as screen rotation, step counting, sleep quality, gaming and personal navigation in mobile and wearable smart devices.

The QMA7981 is based on our state-of-the-art, high resolution single crystal silicon MEMS technology. Along with custom-designed 14-bit ADC ASIC, it offers the advantages of low noise, high accuracy, low power consumption, and offset trimming. The device supports digital interface IIC and SPI.
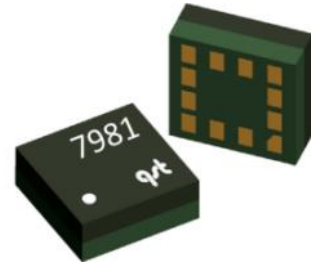
The QMA7981 is in a 2x2x0.95mm3 surface mount 12-pin land grid array (LGA) package.

## FEATURES

- 3-Axis Accelerometer in a 2x2x0.95 mm$^3$ Land Grid Array Package (LGA), guaranteed to operate over a temperature range of -40 °C to +85 °C.

- 14-Bit ADC with low noise accelerometer sensor

- I$^2$C Interface with Standard and Fast modes. Support SPI digital interface

- Built-In Self-Test

- Wide range operation voltage (1.71V To 3.6V) and low power consumption (2-50uA low power conversion current)

- RoHS compliant , halogen-free

- Built–in motion algorithm

## BENEFIT

- Small size for highly integrated products. Signals have been digitized and factory trimmed.

- High resolution allows for motion and tilt sensing

- High-Speed Interfaces for fast data communications.

- Enables low-cost functionality test after assembly in production

- Automatically maintains sensor's sensitivity under wide operation voltage range and compatible with battery powered applications

- Environmental protection and wide applications

- Low power and easy applications including step counting, sleep quality, gaming and personal navigation

## CONTENTS

# 1 INTERNAL SCHEMATIC DIAGRAM

## 1.1 Internal Schematic Diagram
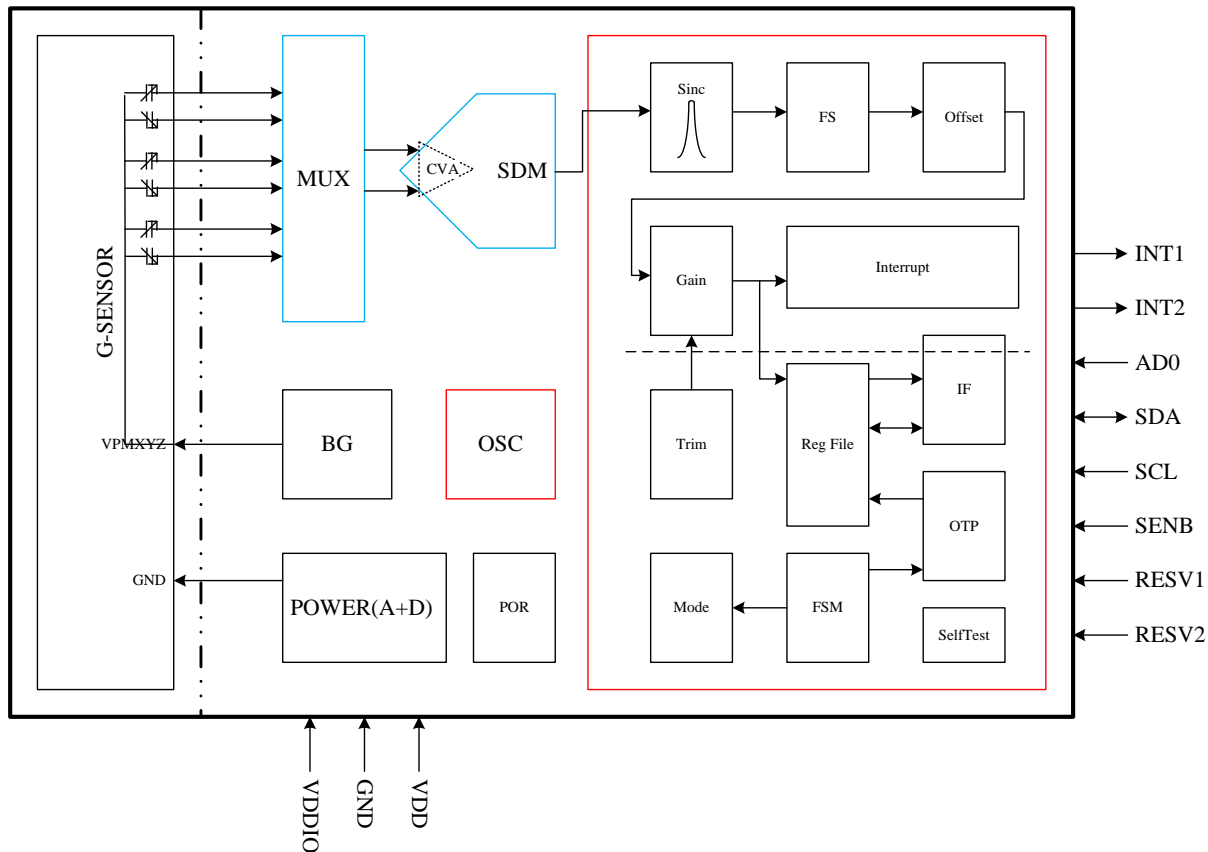


**Figure 1.    Block Diagram**

**Table 1.    Block Function**

| Block | Function |
|---|---|
| Transducer | 3-axis acceleration sensor |
| CVA | Charge-to-Voltage amplifier for sensor signals |
| Interrupt | Digital interrupt engine, to generate interrupt signal on data conversion, and motion function |
| FSM | Finite state machine, to control device in different mode |
| I$^2$C/SPI | Interface logic data I/O |
| OSC | Internal oscillator for internal operation |
| Power | Power block, including LDO |

## 2 SPECIFICATIONS AND I/O CHARACTERISTICS

### 2.1 Product Specifications

Table 2. Specifications (* Tested and specified at 25°C and 3.0V VDD except stated otherwise.)

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Supply voltage VDD | VDD, for internal blocks | 1.71 | 3.3 | 3.6 | V |
| I/O voltage VDDIO | VDDIO, for IO only | 1.71 | 3.3 | VDD | V |
| Standby current | VDD and VDDIO on | | 1 | | µA |
| Low power current | ODR=268 Hz | | 50 | | µA |
| | ODR=134 Hz | | 25.3 | | |
| | ODR=67 Hz | | 12.9 | | |
| | ODR=33.6 Hz | | 6.7 | | |
| | ODR=13.4 Hz | | 2.9 | | |
| | ODR=6.7 Hz | | 1.7 | | |
| Low noise current | ODR=32.5 Hz | | 100 | | µA |
| | ODR=21.6 Hz | | 83.3 | | |
| | ODR=13 Hz | | 50 | | |
| | ODR=6.5 Hz | | 25 | | |
| BW | Programmable bandwidth | | 0.16~168 | | Hz |
| Data output rate (ODR) | 2*BW | | 0.32~336 | | Samples /sec |
| Startup time | From the time when VDD reaches to 90% of final value to the time when device is ready for conversion | | 2 | | ms |
| Wakeup time | From the time device enters into active mode to the time device is ready for conversion | | 1 | | ms |
| Operating temperature | | -40 | | 85 | ℃ |
| Acceleration Full Range | | | ±2/±4/±8/ ±16/±32 | | g |
| Sensitivity | FS=±2g | | 4096 | | LSB/g |
| | FS=±4g | | 2048 | | |
| | FS=±8g | | 1024 | | |
| | FS=±16g | | 512 | | |
| | FS=±32g | | 256 | | |
| Sensitivity Temperature Drift | FS=±2g, Normal VDD Supplies | | ±0.02 | | %/℃ |
| Sensitivity tolerance | Gain accuracy | | ±4 | | % |
| Zero-g offset | FS=±2g, Normal VDD Supplies | | ±80 | | mg |
| Zero-g offset Temperature Drift | FS=±2g, Normal VDD Supplies | | ±2 | | mg/℃ |
| Noise density | FS=±2g, run state | | 200 | | µg/√Hz |
| Nonlinearity | FS=±2g, Best fit straight line, | | ±0.5 | | %FS |
| Cross Axis Sensitivity | | | 1 | | % |

## 2.2 Absolute Maximum Ratings

Table 3.    Absolute Maximum Ratings (Tested at 25°C except stated otherwise.)

| Parameters | Condition | Min | Max | Units |
|---|---|---|---|---|
| VDD | | -0.3 | 5.4 | V |
| VDDIO | | -0.3 | 5.4 | V |
| ESD | HBM | | 2 | kV |
| Shock Immunity | Duration < 200μS | | 10000 | g |
| Storage temperature | | -50 | 150 | ℃ |

## 2.3 I/O Characteristics

Table 4.    I/O Characteristics

| Parameter | Symbol | Pin | Condition | Min. | TYP. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Voltage Input High Level 1 | $V_{IH}1$ | SDA, SCL | | 0.7*VDDIO | | VDDIO+0.3 | V |
| Voltage Input Low Level 1 | $V_{IL}1$ | SDA, SCL | | -0.3 | | 0.3*VDDIO | V |
| Voltage Output High Level | $V_{OH}$ | INT1, INT2 | Output Current ≥-100μA | 0.8*VDDIO | | | V |
| Voltage Output Low Level | $V_{OL}$ | INT1, INT2, SDA | Output Current ≤100μA(INT) Output Current ≤1mA (SDA) | | | 0.2*VDDIO | V |

## 3  PACKAGE PIN CONFIGURATIONS

### 3.1  Package 3-D View

Arrow indicates direction of g field that generates a positive output reading in normal measurement configuration.
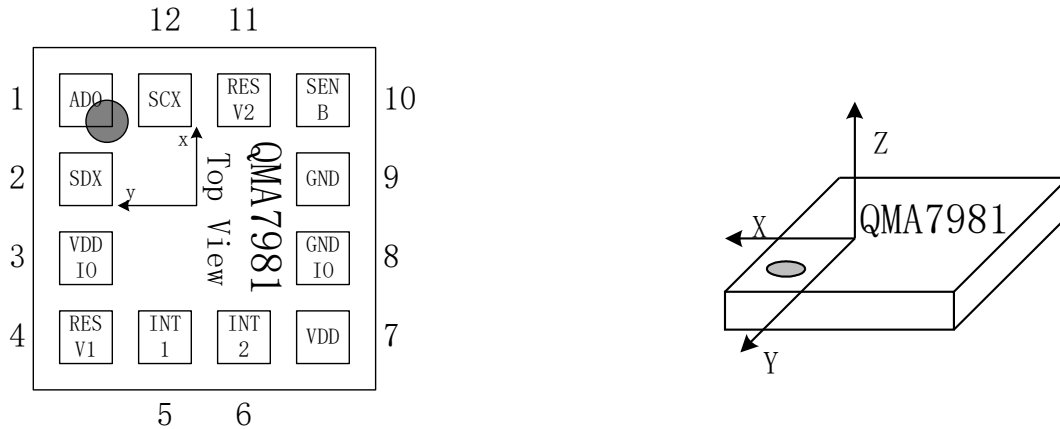


**Figure 2.    Package View**

**Table 5.    Pin Configurations**

| PIN No. | PIN NAME | I/O | Power Supply | TYPE | Function |
|---|---|---|---|---|---|
| 1 | AD0 | I | VDDIO | CMOS | LSB of I$^2$C address, or SDO of 4W SPI |
| 2 | SDX | IO | VDDIO | CMOS | Serial data for I$^2$C, or SDI of 4W SPI, or SDIO of 3W SPI |
| 3 | VDDIO | P | VDDIO | Power | Power supply to digital interface |
| 4 | RESV1 | I | VDDIO | CMOS | Reserved. Float or connect to GND |
| 5 | INT1 | O | VDDIO | CMOS | Interrupt 1 |
| 6 | INT2 | O | VDDIO | CMOS | Interrupt 2 |
| 7 | VDD | P | VDD | Power | Power supply to internal block |
| 8 | GNDIO | G | GND | Power | Ground to digital interface |
| 9 | GND | G | GND | Power | Ground to internal block |
| 10 | SENB | IO | VDDIO | CMOS | Protocol selection |
| 11 | RESV2 | IO | VDDIO | CMOS | Reserved. Float, or connect to GND |
| 12 | SCL | I | VDDIO | CMOS | Serial clock for I$^2$C, or Serial clock for SPI |

## 3.2 Package Outlines

### 3.2.1 Package Type
LGA (Land Grid Array)

### 3.2.2 Package Outline Drawing:
2.0mm (Length)*2.0mm (Width)*0.95mm (Height)



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN. | NOM. | MAX. |
|---|---|---|---|
| A | 0.90 | 0.95 | 1.00 |
| A1 | --- | 0.75 | --- |
| c | 0.16 | 0.20 | 0.24 |
| D | 1.95 | 2.00 | 2.05 |
| D1 | | 1.80 BSC | |
| E | 1.95 | 2.00 | 2.05 |
| E1 | | 1.80 BSC | |
| e | | 0.50 BSC | |
| b | 0.20 | 0.25 | 0.30 |
| L | 0.225 | 0.275 | 0.325 |
| x1 | | 0.10 REF | |
| y1 | | 0.10 REF | |
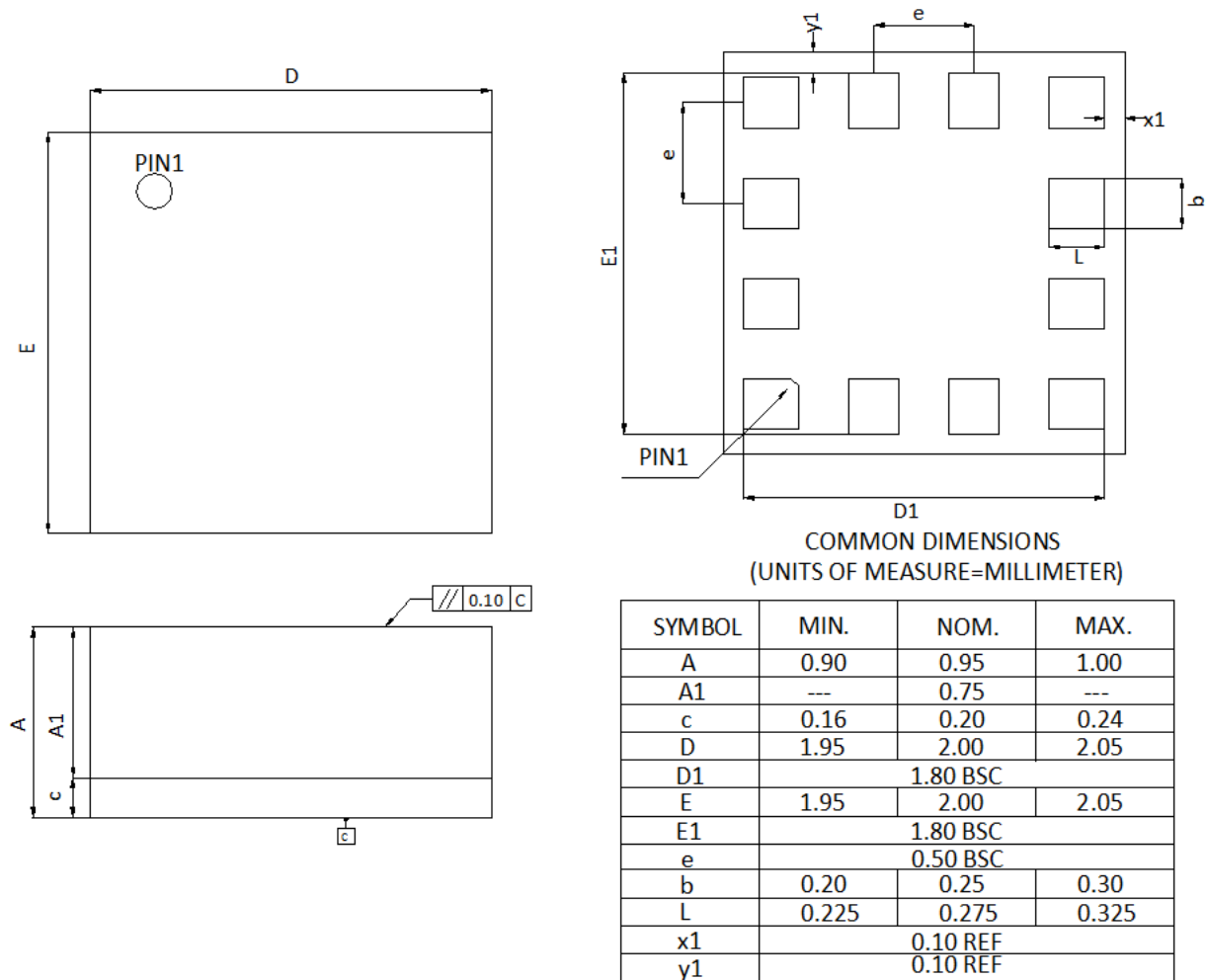
NOTE:
1. CONTROLLING DIMENSION: MILLIMETER.

**Figure 3. Package Outline Drawing**

**3.2.3    Marking:**
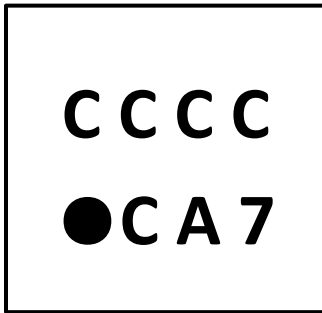
CCCC
●CA7

**Figure 4.    Marking Format**

| Marking Text | Description | Comments |
|---|---|---|
| Line 1 | CCCC: lot code | Lot code: 4 alphanumeric digits, variable to generate mass production trace-code |
| Line 2 | S: Sub-con ID<br>P: Part number<br>7: Fixed number | S: 1 alphanumeric digit, variable identify sub-con<br>P: 1 alphanumeric digit, variable to identify part number<br>7: "7" stand for product name QMA7981 |
| ● | Pin 1 identifier | -- |

# 4   EXTERNAL CONNECTION

## 4.1      I2C Dual Supply Connection
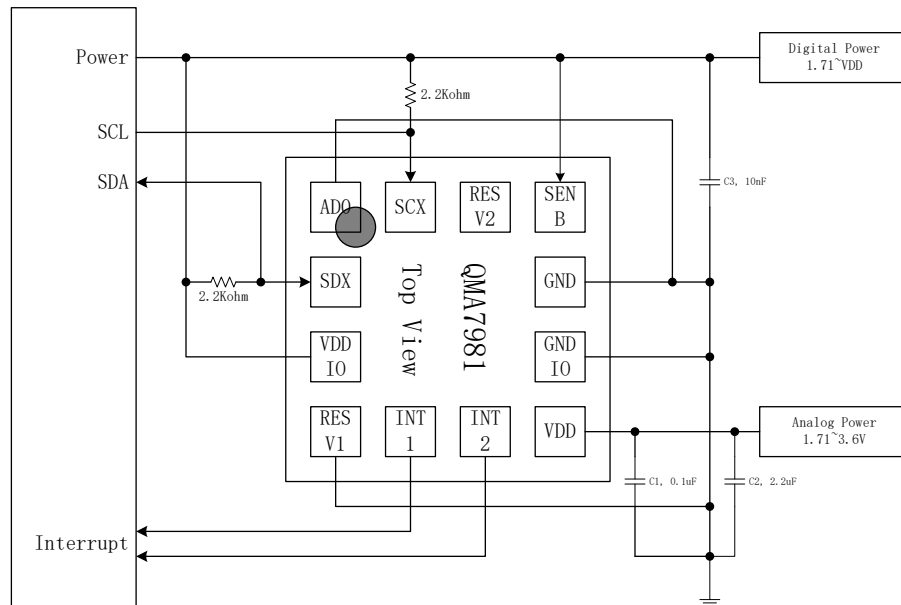


**Figure 5.      I2C Dual Supply Connection**

## 4.2      I2C Single Supply connection
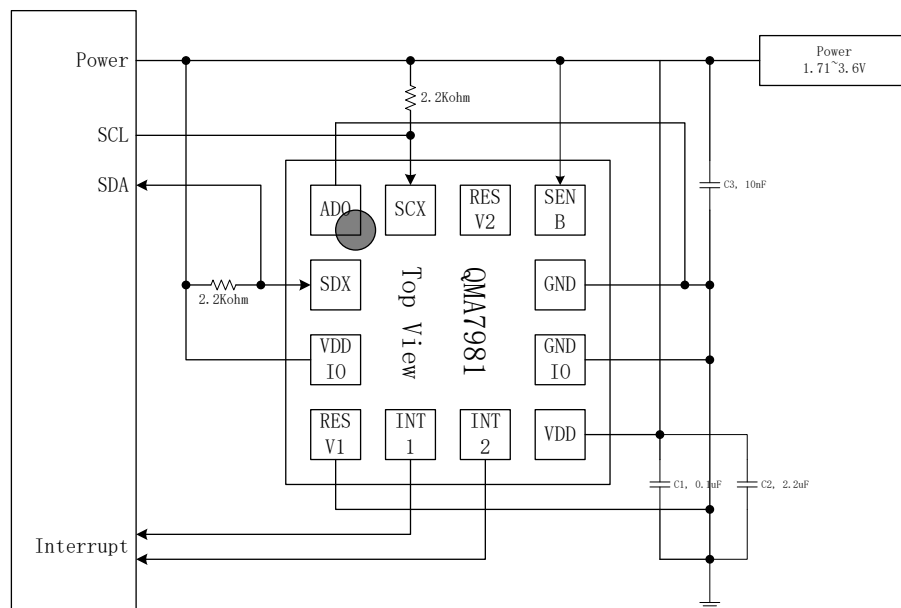


**Figure 6.    I2C Single Supply Connection**

## 4.3    SPI Dual Supply Connection
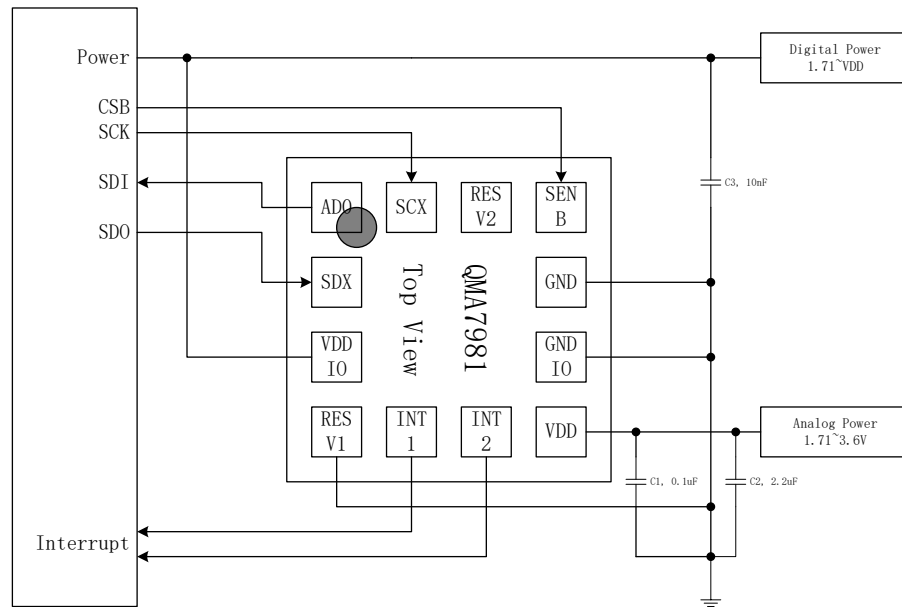


**Figure 7.    SPI Dual Supply Connection**

## 4.4    SPI Single Supply connection
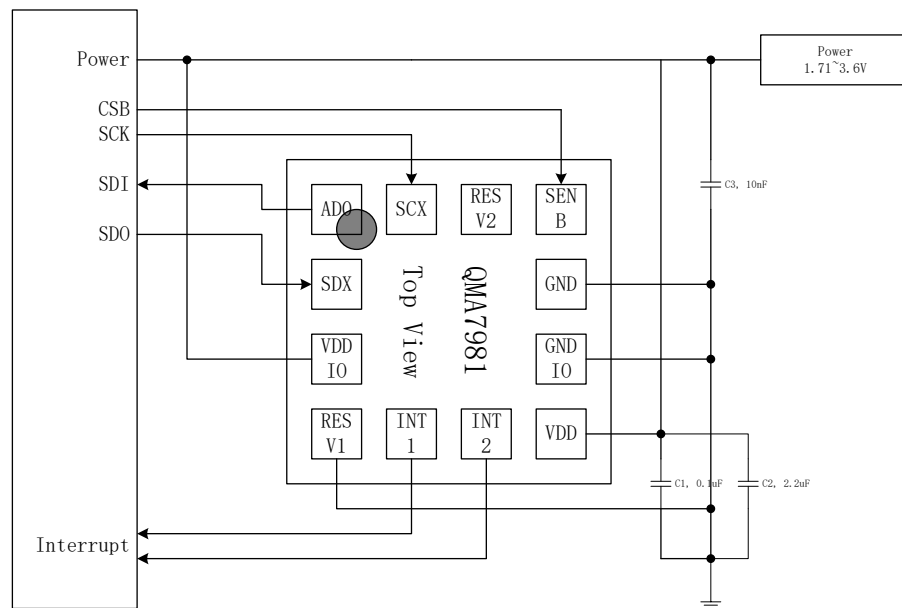


**Figure 8.    SPI Single Supply Connection**

# 5   BASIC DEVICE OPERATION

## 5.1      Acceleration sensor

The QMA7981 acceleration sensor circuit consists of tri-axial sensors and application specific support circuits to measure the acceleration of device. When a DC power supply is applied to the sensor, the sensor converts any accelerating incident in the sensitive axis directions to charge output.

## 5.2      Power Management

Device has two power supply pins. VDD is the main power supply for all of the internal blocks, including analog and digital. VDDIO is a separate power supply, for digital interface only.

The device contains a power-on-reset generator. It generates reset pulse as power on, which can load the register's default value, for the device to function properly.
To make sure the POR block functions well, we should have such constrains on the timing of VDD and VDDIO.

The device should turn-on both power pins in order to operate properly. When the device is powered on, all registers are reset by POR, then the device transits to the standby mode and waits for further commends.

Table 6 provides references for four power states.

**Table 6．   Power States**

| Power State | VDD | VDDIO | Power State description |
|---|---|---|---|
| 1 | 0V | 0V | Device Off, No Power Consumption |
| 2 | 0V | 1.71v~3.6v | Not allowed. User need to make sure that VDDIO is less than VDD. Otherwise, there will be leakage from VDDIO to VDD through internal ESD devices |
| 3 | 1.71v~3.6v | 0V | Device Off, Same Current as Standby Mode |
| 4 | 1.71v~3.6v | 1.71v~VDD | Device On, Normal Operation Mode, Enters Standby Mode after POR |

## 5.3      Power On/Off Time

Device has two power supply pins and two ground pins. VDD is the main power supply for all of the internal blocks, including analog and digital. VDDIO is a separate power supply, for digital interface only. GND is 0V supply for all of internal blocks, and GNDIO for digital interface.

There is no limitation on the voltage levels of VDD and VDDIO relative to each other, as long as they are within operating range.

The device contains a power-on-reset generator. It generates reset pulse as power on, which can load the register's default value, for the device to function properly.

To make sure the POR block functions well, we should have such constrains on the timing of VDD.
The power on/off time related to the device is in Table 7

**Table 7. Time Required for Power On/Off**

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| POR Completion Time | PORT | Time Period After VDD and VDDIO at Operating Voltage to Ready for I²C Commend and Analogy Measurement. | | | 250 | µs |
| Power off Voltage | SDV | Voltage that Device Considers to be Power Down. | | | 0.2 | V |
| Power on Interval | PINT | Time Period Required for Voltage Lower Than SDV to Enable Next POR | 100 | | | µs |
| Power on Time | PSUP | Time Period Required for Voltage from SDV to 90% of final value | | | 50 | ms |
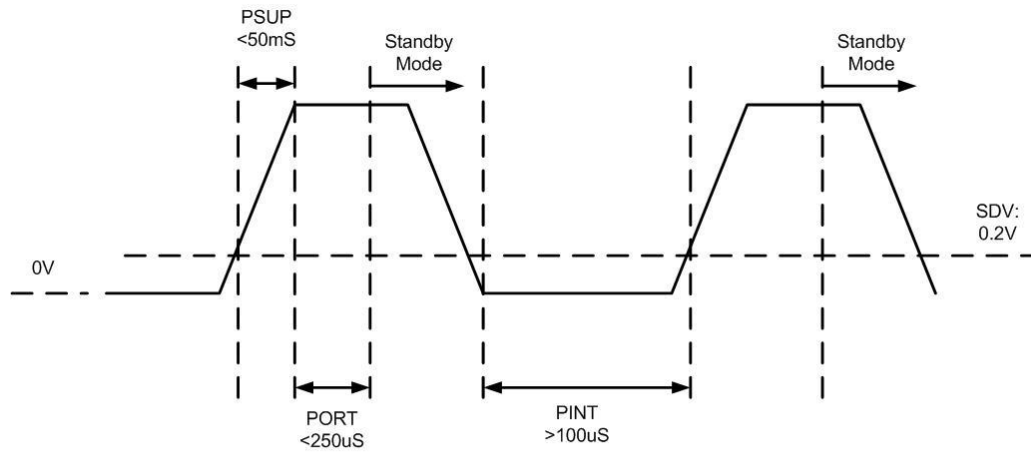


**Power On/Off Timing**

**Figure 9.    Power On/Off Timing**

## 5.4    Communication Bus Interface I²C and Its Addresses

This device will be connected to a serial interface bus as a slave device under the control of a master device, such as the processor. Control of this device is carried out via I²C.

This device is compliant with I²C -Bus Specification, document number:    9398 393 40011.    As an I²C compatible device, this device has a 7-bit serial address and supports I²C protocols.    This device supports standard and fast speed modes, 100 kHz and 400 kHz, respectively. External pull-up resistors are required to support all these modes.

There are two I²C addresses selected by connecting pin 1 (AD0) to GND or VDDIO.    The first six MSB are hardware configured to "001001" and the LSB can be configured by AD0.

**Table 8.  I²C Address Options**

| AD0 (pin 1) | I²C Slave Address(HEX) | I²C Slave Address(BIN) |
|---|---|---|
| Connect to GND | 12 | 0010010 |
| Connect to VDDIO | 13 | 0010011 |

# 6  MODES OF OPERATION

## 6.1  Modes Transition

QMA7981 has two different operational modes, controlled by register (0x11), MODE_BIT. The main purpose of these modes is for power management. The modes can be transited from one to another, as shown below, through I²C commands. The default mode after power-on is standby mode.
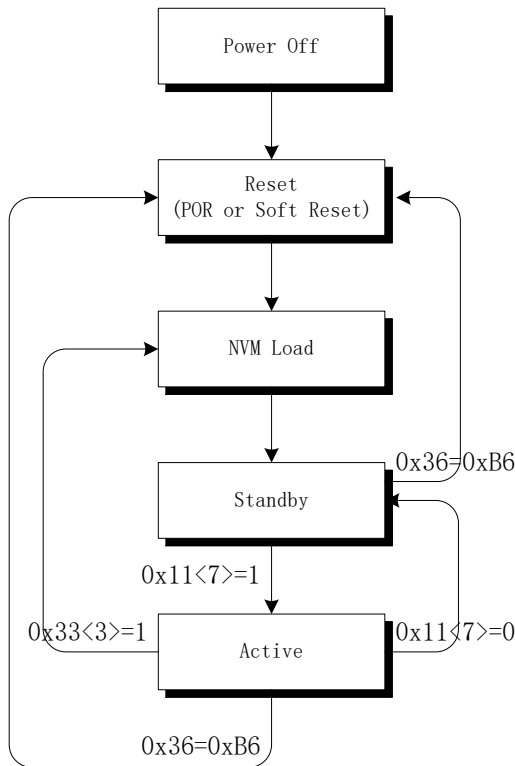


**Figure 10.  Basic operation flow after power-on**
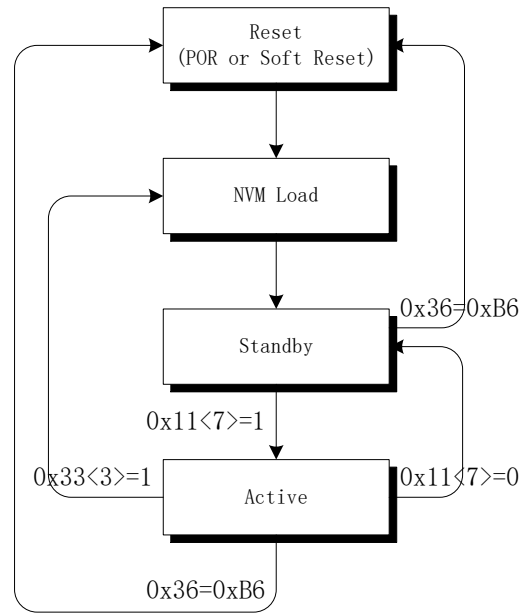
**Figure 11.  The work mode transferring**

The default mode after power on is standby mode. Through I²C instruction, device can switch between standby mode and active mode. With SOFTRESET by writing 0xB6 into register 0x36, all of the registers will get default values. SOFTRESET can be done both in active mode and in standby mode. Also, by writing 1 in NVM_LOAD (0x33<3>) when device is in active mode, the NVM related image registers will get default value from NVM, however, other registers will keep the values of their own.

## 6.2  Description of Modes

### 6.2.1 Active Mode

In active mode, the ADC digitizes the charge signals from transducer, and digital signal processor conditions these signals in digital domain, processes the interrupts, and send data to Data registers (0x01~0x06).

### 6.2.2 Standby Mode

In standby mode, most of the blocks are off, while device is ready for access through I²C. Standby mode is the default mode after power on or soft reset. Device can enter into this mode by set the soft reset register (0x36) to 0xB6 or set the MODE_BIT (0x11<7>) to logic 0.

Besides the above two modes, the device also contains NVM loading state. This state is used to reset the value of the NVM related image registers. There are two bits related to this state. When NVM_LOAD (0x33<3>) is set to 1, NVM loading starts. When the device is in NVM loading state, NVM_RDY (0x33<2>) is set to logic 0 by device. After NVM loading is finished, NVM_RDY (0x33<2>) is set back to logic 1 by device, and NVM_LOAD is reset to 0 by device automatically. NVM loading can only happen when NVM_LOAD is set to 1 in active mode. If the user sets this NVM_LOAD bit to 1 in standby mode, the device will not take the action until it enters into active state by setting MODE_BIT (0x11<7>) to logic 1.

After loading NVM, the device will enter into standby mode directly.

The loading time for NVM is about 100uS.

# 7 Functions and interrupts

ASIC support interrupts, such as STEP_INT, DRDY_INT, ANY_MOT_INT, SIG_MOT_INT, NO_MOT_INT, RAISE_INT, etc.

## 7.1 STEP_ INT

The STEP/STEP_QUIT detect that the user is entering/exiting step mode. When the user enter into step mode, at least one axis sensor data will vary periodically, by numbering the variation periods the step counter can be calculated.
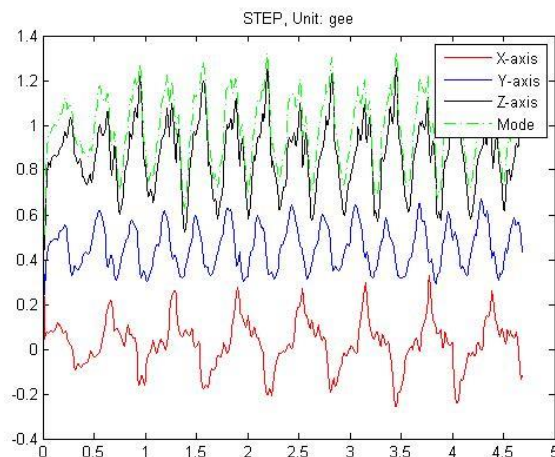


**Figure 10. STEP/STEP_QUIT**

Median data (max+min) /2 is called dynamic threshold, the max and min data can be updated by certainly samples, the sample number can be set by register STEP_SAMPLE_CNT (0x12<6:0>). When the sensor data decreasing (or increasing) through the dynamic threshold, a user run step is detected.

Register STEP_PRECISION (0x13<6:0>) is used as threshold when updating the new collected sensor data. Sensor data below the threshold will be discarded, this helps removing unstable variations causing failed detection.

The run step event happened at certain interval timing. All of the events outside the timing window will not be regarded as a run step and the step counter will not counted. The timing window can be set by register STEP_TIME_UP(0x15)    and

STEP_TIME_LOW(0x14), the conversion ODR numbers ranged from STEP_TIME_LOW *ODR to 8* STEP_TIME_UP*ODR .

STEP_COUNT_PEAK<2:0> is used to set a fixed peak value for step detection, 0.05G~0.4G can be set. STEP_COUNT_P2P<2:0> is used to set a peak to peak threshold for step detection, 0.3G~1G can be set.

To remove unstable variation which will cause false STEP event detection, the step counter considers steps as valid step events only after some continuous steps detected; the start threshold can be set by 0x1F<7:5>. Also, the step counter register STEP_CNT<23:0> ({0x0E,0x08,0x07}) will be updated immediately by the setting number, and interrupt STEP is also generated.

The related interrupt status bit is STEP_INT (0x0A<3>) and SIG_STEP (0x0A<6>). When the interrupt is generated, the value of STEP_INT will be set to logic 1, which will be cleared after the interrupt status register is read by user. STEP_IEN/SIG_STEP_IEN (0x16<3>/0x16<6>) is the enable bit for the STEP_INT/SIG_STEP_INT. Also, to get this interrupt on PIN_INT1 and/or PIN_INT2, we need to set INT1_STEP (0x19<3>)/INT1_SIG_STEP (0x19<6>) or INT2_STEP (0x1B<3>) /INT2_SIG_STEP (0x1B<6>) to logic 1, to map the internal interrupt to the interrupt PINs.

## 7.2 DRDY_INT

The width of the acceleration data is 14 bits, in two's complement representation. The data of each axis is split into 2 parts, the MSB part (one byte contains bit 13 to bit 6) and the LSB part (one byte contains bit 5 to bit 0). Reading data should start with LSB part. When user is reading the LSB byte of data, to ensure the integrity of the acceleration data, the content of MSB can be locked, by setting SHADOW_DIS (0x21<6>) to logic 0. This lock function can be disabled by setting SHADOW_DIS to logic 1. Without lock, the MSB and LSB content will be updated by new value immediately. The bit NEW_DATA in the LSB byte is the flag of the new data. If new data is updated, this NEW_DATA flag will be 1, and will be cleared when corresponding MSB or LSB is read by user.

Also, the user should note that even with SHADOW_DIS=0, the data of 3 axes are not guaranteed from the same time point.

The device supports four different acceleration measurement ranges. The range is setting through RANGE (0x0F<3:0>), and the details as following:

| RANGE | Acceleration range | Resolution |
|-------|--------------------|------------|
| 0001 | 2g | 244ug/LSB |
| 0010 | 4g | 488ug/LSB |
| 0100 | 8g | 977ug/LSB |
| 1000 | 16g | 1.95mg/LSB |
| 1111 | 32g | 3.91mg/LSB |
| Others | 2g | 244ug/LSB |

The interrupt for the new data serves for the synchronous data reading for the host. It is generated after storing a new value of z-axis acceleration data into data register. This interrupt will be cleared automatically when the next data conversion cycle starts, and the interrupt will be effective about 64*MCLK, and automatically cleared.
The interrupt mode for the new data is fixed to be non-latched.

## 7.3 ANY_MOT_INT

Any motion Any motion detection uses slope between two successive data to detect the changes in motion. It generates interrupt when a preset threshold ANY_MOT_TH (0x2E) is exceeded.

The time difference between two successive data depends on the output data rate (ODR).

$$\text{Slope}(t1) = \big(acc(t1) - acc(t0)\big) * ODR$$

The any motion detection criteria are fulfilled and interrupt is generated if any of enabled channels exceeds ANY_MOT_TH for ANY_MOT_DUR (0x2C<1:0>) consecutive times.

As long as all the enabled channels data fall or stay below ANY_MOT_TH for ANY_MOT_DUR consecutive times, the interrupt will be reset unless the interrupt signal is latched.

The any motion detection engine will send out the signals of axis which triggered the interrupt (ANY_MOT_FIRST_X (0x09<0>), ANY_MOT_FIRST_Y (0x09<1>), ANY_MOT_FIRST_Z (0x09<2>)) and the sign of the motion (ANY_MOT_SIGN (0x09<3>))

## 7.4 SIG_MOT_INT

A significant motion is a motion due to a change in user location.
The algorithm is as following:
1) Look for movement, same setting as any motion detection
2) If movement detected, sleep for T_Skip (0x2F<3:2>)
3) Look for movement
   a) If no movement detected within T_Proof (0x2F<5:4>), go back to 1
   b) If movement detected, report a significant movement, and generate the interrupt

The significant motion detection and any motion detection are exclusive, user can select either one through SIG_MOT_SEL (0x2F<0>).
If significant motion is detected, the engine will set SIG_MOT_INT (0x0A<0>).

## 7.5 NO_MOT_INT

No-motion interrupt is generated if the slope (absolute value of acceleration difference) on all selected axes is smaller than the programmable threshold for a programmable time. Figure shows the timing for the no-motion interrupt. Register (0x2C) NO_MOT_DUR defines the delay times before the no-motion interrupt is generated. Table lists the delay times adjustable with register (0x2C) NO_MOT_DUR.

The no-motion interrupt is enabled per axis by writing logic 1 to bits (0x18) NO_MOTION_EN_X, (0x18) NO_MOTION_EN_Y, and (0x18) NO_MOTION_EN_Z, respectively. The no-motion threshold is set through the (0x2D) NO_MOT_TH register. The meaning of an LSB of (0x2D) NO_MOT_TH depends on the selected g-range: it corresponds to 3.91mg in 2g-range (7.81mg in 4g-range, 15.6mg in 8g-range, 31.25mg in 16g-range, 62.5mg in 32g-range). Therefore the maximum value is 996mg in 2g-range (2g in 4g-range, 4g in 8g-range, 8g in 16g-range, and 16g in 32g-range). The time difference between the successive acceleration samples depends on the selected ODR and equates to 1/ODR.

## 7.6 RAISE_INT

Raise wake algorithm is used to detect the action of raise hand (or hand down). The interrupt is enabled by writing logic 1 to bits (0X16[1]) RAISE_EN, (0X16[2]) HD_EN. User can adjust the sensitivity through the registers. The register RAISE_WAKE_SUM_TH(0X2A[5:0]) defines the strength of hand action (raise and down). The register RAISE_DIFF_TH(0X2A[7:6],0X2B[1:0]) defines the differential values of twice actions, when the hand behavior almost done the differential value will be smaller and we can use this register to set the threshold. RAISE_WAKE_PERIOD and RAISE_WAKE_TIMEOUT_TH define the duration of the total hand action.

## 7.7 Interrupt configuration

The device has the above 3 interrupt engines. Each of the interrupts can be enabled and configured independently. If the trigger condition of the enabled interrupt fulfilled, the corresponding interrupt status bit will be set to logic 1, and the mapped interrupt pin will be activated. The device has two interrupt PINs, INT1 and INT2. Each of the interrupts can be mapped to either PIN or both PINs.

The interrupt status registers INT_ST(0x09~0x0d) will update when a new data word is written into the data registers. If an interrupt is disabled, the related active interrupt status bit is disabled immediately.

When interrupt condition is fulfilled, related bit of interrupt will be set, until the associated interrupt condition is no more valid. Read operation to related register will also clear the register.

Device supports 2 interrupt modes, non-latched, and latched mode. The interrupt modes are set through LATCH_INT (0x21<0>).

In non-latched mode, the mapped interrupt pin will be set and/or cleared same as associated interrupt register bit. Also, the mapped interrupt pin can be cleared with read operation to any of the INT_ST(0x09~0x0d).

Exception to this is the new data interrupt and step interrupt, which are automatically reset after a fixed time (T_Pulse = 64/MCLK), no matter LATCH_INT (0x21<0>) is set to 0 or 1.

In latched mode, the clearings of mapped pins are determined by INT_RD_CLR (0x21<7>).
If the condition for trigging the interrupt still holds, the interrupt status will be set again with the next change of the data registers.

Mapping the interrupt pins can be set by INT_MAP (0x19~0x1B).

The electrical interrupt pins can be set INT_PIN_CONF (0x20<3:0>). The active logic level can be set to 1 or 0, and the interrupt pin can be set to open-drain or push-pull.

# 8 I²C COMMUNICATION PROTOCOL

## 8.1 I²C Timings

Table 9 and Figure 11 describe the I²C communication protocol times

**Table 9. I²C Timings**

| Parameter | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| SCL Clock | $f_{scl}$ | | 0 | | 400 | kHz |
| SCL Low Period | $t_{low}$ | | 1 | | | µs |
| SCL High Period | $t_{high}$ | | 1 | | | µs |
| SDA Setup Time | $t_{sudat}$ | | 0.1 | | | µs |
| SDA Hold Time | $t_{hddat}$ | | 0 | | 0.9 | µs |
| Start Hold Time | $t_{hdsta}$ | | 0.6 | | | µs |
| Start Setup Time | $t_{susta}$ | | 0.6 | | | µs |
| Stop Setup Time | $t_{susto}$ | | 0.6 | | | µs |
| New Transmission Time | $t_{buf}$ | | 1.3 | | | µs |
| Rise Time | $t_r$ | | | | | µs |
| Fall Time | $t_f$ | | | | | µs |
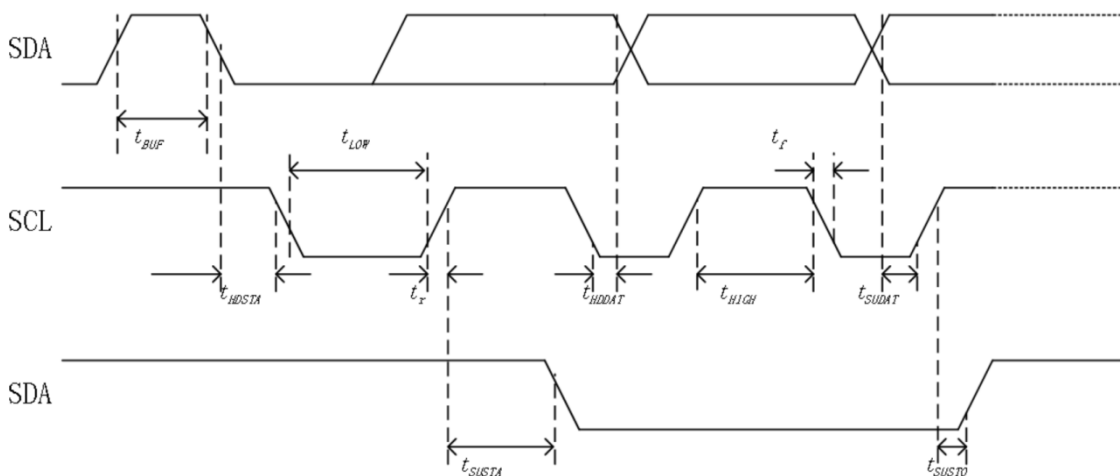


**Figure 11. I²C Timing Diagram**

## 8.2 I²C R/W Operation

### 8.2.1 Abbreviation

**Table 10. Abbreviation**

| SACK | Acknowledged by slave |
|------|------------------------|
| MACK | Acknowledged by master |
| NACK | Not acknowledged by master |
| RW | Read/Write |

### 8.2.2    Start/Stop/Ack

START: Data transmission begins with a high to transition on SDA while SCL is held high. Once I²C transmission starts, the bus is considered busy.

STOP:  STOP condition is a low to high transition on SDA line while SCL is held high.

ACK: Each byte of data transferred must be acknowledged. The transmitter must release the SDA line during the acknowledge pulse while the receiver mush then pull the SDA line low so that it remains stable low during the high period of the acknowledge clock cycle.

NACK: If the receiver doesn't pull down the SDA line during the high period of the acknowledge clock cycle, it's recognized as NACK by the transmitter.

### 8.2.3    I²C Write

I²C write sequence begins with start condition generated by master followed by 7 bits slave address and a write bit (R/W=0). The slave sends an acknowledge bit (ACK=0) and releases the bus. The master sends the one byte register address. The slave again acknowledges the transmission and waits for 8 bits data which shall be written to the specified register address. After the slave acknowledges the data byte, the master generates a stop signal and terminates the writing protocol.

**Table 11.    I²C Write**

| START | Slave Address | | | | | | | RW | SACK | Register Address (0x11) | | | | | | | | SACK | Data (0x80) | | | | | | | | SACK | STOP |
|-------|---|---|---|---|---|---|---|----|------|---|---|---|---|---|---|---|---|------|---|---|---|---|---|---|---|---|------|------|
| | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

### 8.2.4    I²C Read

I²C write sequence consists of a one-byte I²C write phase followed by the I²C read phase. A start condition must be generated between two phase. The I²C write phase addresses the slave and sends the register address to be read. After slave acknowledges the transmission, the master generates again a start condition and sends the slave address together with a read bit (R/W=1). Then master releases the bus and waits for the data bytes to be read out from slave. After each data byte the master has to generate an acknowledge bit (ACK = 0) to enable further data transfer. A NACK from the master stops the data being transferred from the slave. The slave releases the bus so that the master can generate a STOP condition and terminate the transmission.

The register address is automatically incremented and more than one byte can be sequentially read out. Once a new data read transmission starts, the start address will be set to the register address specified in the current I²C write command.

**Table 12.    I²C Read**

| START | Slave Address | | | | | | | RW | SACK | Register Address (0x00) | | | | | | | | SACK |
|-------|---|---|---|---|---|---|---|----|------|---|---|---|---|---|---|---|---|------|
| | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| START | Slave Address | | | | | | | RW | SACK | Data (0x00) | | | | | | | | MACK | Data (0x01) | | | | | | | |
|-------|---|---|---|---|---|---|---|----|------|---|---|---|---|---|---|---|---|------|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| MACK | Data (0x02) | | | | | | | | MACK | ............................ | MACK | Data (0x07) | | | | | | | | NACK | STOP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | ............................ | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

# 9    REGISTERS

## 9.1    Register Map

The table below provides a list of the 8-bit registers embedded in the device and their respective function and addresses

**Table 13. Register Map**

| Add. | Name | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 | R/W | DEF |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x3F | | RAISE_WAKE_PERIOD[10:8] | | | RAISE_WAKE_TIMEOUT_TH[11:8] | | | | | RW | 02 |
| 0x3E | RAISE_CFG | RAISE_WAKE_TIMEOUT_TH[7:0] | | | | | | | | RW | 00 |
| 0x36 | S_RESET | SOFTRESET: 0xB6 | | | | | | | | RW | 00 |
| 0x35 | | RAISE_WAKE_PERIOD[7:0] | | | | | | | | RW | 81 |
| 0x34 | | YZ_TH_SEL[2:0] | | | | Y_TH[4:0] | | | | RW | 9D |
| 0x32 | ST | SELFTEST_BIT | | | | | SELFTEST_SIGN | BP_AXIS_STEP<1:0> | | RW | 00 |
| 0x31 | | | | | | | | | | RW | 00 |
| 0x30 | RST_MOT | MO_BP_CO | STEP_BP_CO | | LOW_RST_N | HIGH_RST_N | NO_MOT_RST_N | SIG_MOT_RST_N | ANY_MOT_RST_N | RW | 1F |
| 0x2F | | | | SIG_MOT_TPROOF<1:0> | | SIG_MOT_TSKIP<1:0> | | | SIG_MOT_SEL | RW | 00 |
| 0x2E | | ANY_MOT_TH<7:0> | | | | | | | | RW | 00 |
| 0x2D | | NO_MOT_TH<7:0> | | | | | | | | RW | 00 |
| 0x2C | MOT_CFG | NO_MOT_DUR<5:0> | | | | | | ANY_MOT_DUR<1:0> | | RW | 00 |
| 0x2B | | HD_Z_TH[2:0] | | | HD_X_TH[2:0] | | | RAISE_WAKE_DIFF_TH[3:2] | | RW | 7C |
| 0x2A | RAISE_CFG | RAISE_WAKE_DIFF_TH[1:0] | | RAISE_WAKE_SUM_TH[5:0] | | | | | | RW | D8 |
| 0x29 | | OS_CUST_Z<7:0> | | | | | | | | RW | 00 |
| 0x28 | | OS_CUST_Y<7:0> | | | | | | | | RW | 00 |
| 0x27 | OS_CUST | OS_CUST_X<7:0> | | | | | | | | RW | 00 |
| 0x21 | INT_CFG | INT_RD_CLR | SHADOW_DIS | DIS_I2C | 1 | 1 | 1 | LATCH_INT_STEP | LATCH_INT | RW | 1C |
| 0x20 | INT_PIN_CFG | DIS_PU_SENB | DIS_IE_AD0 | EN_SPI3W | STEP_COUNT_PEAK<2> | INT2_OD | INT2_LVL | INT1_OD | INT1_LVL | RW | 05 |
| 0x1F | | STEP_START_CNT<2:0> | | | STEP_COUNT_PEAK<1:0> | | STEP_COUNT_P2P<2:0> | | | RW | A9 |
| 0x1E | | Z_TH[3:0] | | | | X_TH[3:0] | | | | RW | 66 |
| 0x1D | STEP_CFG | STEP_INTERVAL<6:0> | | | | | | | EN_RESET_DC | RW | 00 |
| 0x1C | | INT2_NO_MOT | 1 | 1 | INT2_DATA | INT2_LOW | INT2_HIGH | 1 | INT2_ANY_MOT | RW | 62 |
| 0x1B | | 1 | INT2_SIG_STEP | 1 | 1 | INT2_STEP | INT2_HD | INT2_RAISE | INT2_SIG_MOT | RW | B0 |
| 0x1A | | INT1_NO_MOT | 1 | 1 | INT1_DATA | INT1_LOW | INT1_HIGH | 1 | INT1_ANY_MOT | RW | 62 |
| 0x19 | INT_MAP | 1 | INT1_SIG_STEP | 1 | 1 | INT1_STEP | INT1_HD | INT1_RAISE | INT1_SIG_MOT | RW | B0 |
| 0x18 | | NO_MOT_EN_Z | NO_MOT_EN_Y | NO_MOT_EN_X | 1 | 1 | ANY_MOT_EN_Z | ANY_MOT_EN_Y | ANY_MOT_EN_X | RW | 18 |
| 0x17 | | 1 | 1 | 1 | INT_DATA_EN | LOW_EN | HIGH_EN_Z | HIGH_EN_Y | HIGH_EN_X | RW | E0 |
| 0x16 | INT_EN | 1 | SIG_STEP_IEN | 1 | 1 | STEP_IEN | HD_EN | RAISE_EN | 1 | RW | B1 |
| 0x15 | | STEP_TIME_UP<7:0> | | | | | | | | RW | 00 |
| 0x14 | | STEP_TIME_LOW<7:0> | | | | | | | | RW | 19 |
| 0x13 | | STEP_CLR | STEP_PRECISION<6:0> | | | | | | | RW | 7F |
| 0x12 | STEP_CFG | STEP_EN | STEP_SAMPLE_CNT<6:0> | | | | | | | RW | 14 |
| 0x11 | PM | MODE_BIT | 1 | T_RSTB_SINC_SEL<1:0> | | MCLK_SEL<3:0> | | | | RW | 40 |
| 0x10 | BW | 1 | 1 | 1 | BW<4:0> | | | | | RW | E0 |
| 0x0F | FSR | 1 | 1 | 1 | 1 | RANGE<3:0> | | | | RW | F0 |
| 0x0E | STEPCNT | STEP_CNT<23:16> | | | | | | | | R | 00 |
| 0x0D | | | | | | | | | | R | 00 |
| 0x0C | | | | | HIGH_INT | HIGH_SIGN | HIGH_FIRST_Z | HIGH_FIRST_Y | HIGH_FIRST_X | R | 00 |
| 0x0B | | | | | DATA_INT | LOW_INT | | | | R | 00 |
| 0x0A | | | SIG_STEP | | | STEP_INT | HD_INT | RAISE_INT | SIG_MOT_INT | R | 00 |
| 0x09 | INT_ST | NO_MOT | | | | ANY_MOT_SIGN | ANY_MOT_FIRST_Z | ANY_MOT_FIRST_Y | ANY_MOT_FIRST_X | R | 00 |
| 0x08 | | STEP_CNT<15:8> | | | | | | | | R | 00 |
| 0x07 | STEPCNT | STEP_CNT<7:0> | | | | | | | | R | 00 |
| 0x06 | | ACC_Z<13:6> | | | | | | | | R | 00 |
| 0x05 | | ACC_Z<5:0> | | | | | | | NEWDATA_Z | R | 00 |
| 0x04 | | ACC_Y<13:6> | | | | | | | | R | 00 |
| 0x03 | | ACC_Y<5:0> | | | | | | | NEWDATA_Y | R | 00 |
| 0x02 | | ACC_X<13:6> | | | | | | | | R | 00 |
| 0x01 | DATA | ACC_X<5:0> | | | | | | | NEWDATA_X | R | 00 |
| 0x00 | CHIP ID | CHIP ID to indicate the product version | | | | | | | | R | ANA |

## 9.2     Register Definition

Register 0x00 (CHIP ID)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| CHIP_ID<7:0> | | | | | | | | RW | 0xEX |

This register is used to identify the device

Register 0x01 ~ 0x02 (DXL, DXM)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| DX<5:0> | | | | | | | NEWDATA _X | R | 0x00 |
| DX<13:6> | | | | | | | | R | 0x00 |

DX:                                14bits acceleration data of x-channel. This data is in two's complement.
NEWDATA_X:                1, acceleration data of x-channel has been updated since last reading
                                     0, acceleration data of x-channel has not been updated since last reading

Register 0x03 ~ 0x04 (DYL, DYM)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| DY<5:0> | | | | | | | NEWDATA _Y | R | 0x00 |
| DY<13:6> | | | | | | | | R | 0x00 |

DY:                                14bits acceleration data of y-channel. This data is in two's complement.
NEWDATA_Y:                1, acceleration data of y-channel has been updated since last reading
                                     0, acceleration data of y-channel has not been updated since last reading

Register 0x05 ~ 0x06 (DZL, DZM)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| DZ<5:0> | | | | | | | NEWDATA _Z | R | 0x00 |
| DZ<13:6> | | | | | | | | R | 0x00 |

DZ:                                14bits acceleration data of z-channel. This data is in two's complement.
NEWDATA_Z:                1, acceleration data of z-channel has been updated since last reading
                                     0, acceleration data of z-channel has not been updated since last reading

Register 0x07 ~ 0x08 (STEP_CNT)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_CNT<7:0> | | | | | | | | R | 0x00 |
| STEP_CNT<15:8> | | | | | | | | R | 0x00 |

STEP_CNT<15:0>:              16 bits of step counter, out of total 24bits data. The MSB data are in 0x0e

Register 0x09 (INT_ST0)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| NO_MOT | SIG_STEP | | | ANY_MOT _SIGN | ANY_MOT _FIRST_Z | ANY_MOT _FIRST_Y | ANY_MOT _FIRST_X | R | 0x00 |

NO_MOT:

                                     1, no_motion interrupt active
                                     0, no_motion interrupt inactive
ANY_MOT_SIGN:

                                     1, sign of any_motion triggering signal is negative
                                     0, sign of any_motion triggering signal is positive
ANY_MOT_FIRST_Z:

                                     1, any_motion interrupt is triggered by Z axis
                                     0, any_motion interrupt is not triggered by Z axis
ANY_MOT_FIRST_Y:

                                     1, any_motion interrupt is triggered by Y axis
                                     0, any_motion interrupt is not triggered by Y axis
ANY_MOT_FIRST_X:

                                     1, any_motion interrupt is triggered by X axis
                                     0, any_motion interrupt is not triggered by X axis

Register 0x0a (INT_ST1)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|

| | | SIG_STEP | | | STEP_INT | HD_INT | RAISE_INT | SIG_MOT_INT | R | 0x00 |
|---|---|---|---|---|---|---|---|---|---|---|

SIG_STEP:                1, significant step is active

                                     0, significant step is inactive

STEP_INT:               1, step valid interrupt is active

                                     0, step quit interrupt is inactive

HD_INT:                 1, hand down interrupt is active

                                     0, hand down interrupt is inactive

RAISE_INT:            1, raise hand interrupt is active

                                   0, raise hand interrupt is inactive

SIG_MOT_INT:         1, significant interrupt is active

                                   0, significant interrupt is inactive

Register 0x0b (INT_ST2)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| | | | DATA_INT | | | | | R | 0x00 |

DATA_INT:               1, data ready interrupt active

                                   0, data ready interrupt inactive

Register 0x0e (STEP_CNT)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_CNT<23:16> | | | | | | | | R | 0x00 |

STEP_CNT<23:16>:        8bit MSB data of step counter, out of total 24bits data. The LSB data are in 0x07 and 0x08

Register 0x0f (FSR)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | RANGE<3:0> | | | | RW | 0xF0 |

RANGE<3:0>:            set the full scale of the accelerometer. Setting as following

| RANGE<3:0> | Acceleration range | Resolution |
|---|---|---|
| 0001 | 2g | 244ug/LSB |
| 0010 | 4g | 488g/LSB |
| 0100 | 8g | 977ug/LSB |
| 1000 | 16g | 1.95mg/LSB |
| 1111 | 32g | 3.91mg/LSB |
| Others | 2g | 244ug/LSB |

Register 0x10 (BW)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | BW<4:0> | | | | | RW | 0xE0 |

BW<4:0>:               bandwidth setting, as following

| BW<4:0> | ODR |
|---|---|
| xx000 | MCLK/7695 |
| xx001 | MCLK/3855 |
| xx010 | MCLK/1935 |
| xx011 | MCLK/975 |
| xx100 | |
| xx101 | MCLK/15375 |
| xx110 | MCLK/30735 |
| xx111 | MCLK/61455 |
| Others | MCLK/7695 |

Register 0x11 (PM)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| MODE_BIT | 1 | T_RSTB_SINC_SEL<1:0> | | MCLK_SEL<3:0> | | | | RW | 0x40 |

MODE_BIT:             1, set device into active mode

                                   0, set device into standby mode

T_RSTB_SINC_SEL<1:0>:  Reset clock setting. The preset time is reserved for CIC filter in digital

                                 11, T_RSTB_SINC=8*MCLK

                                 10, T_RSTB_SINC=6*MCLK

                                 01, T_RSTB_SINC=4*MCLK

                                 00, T_RSTB_SINC=3*MCLK

MCLK_SEL<3:0>:         set the master clock to digital

| MCLK_SEL<3:0> | Freq of MCLK |
|---|---|
| 0000 | 500KHz |
| 0001 | 333KHz |
| 0010 | 200KHz |
| 0011 | 100KHz |
| 0100 | 50KHz |
| 0101 | 25KHz |
| 0110 | 12.5KHz |
| 0111 | 5KHz |
| 1xxx | Reserved |

Register 0x12 (STEP_CONF0)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_EN | STEP_SAMPLE_CNT<6:0> | | | | | | | RW | 0x14 |

STEP_EN:                             enable step counter, this bit should be set when using step counter

STEP_SAMPLE_CNT:

sample count setting for dynamic threshold calculation. The actual value is STEP_SAMPLE_CNT<6:0>*8, default is 0xC, 96 sample count

Register 0x13 (STEP_CONF1)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_CLR | STEP_PRECISION<6:0> | | | | | | | RW | 0x7F |

STEP_CLR:                             clear step count in register 0x07 ,0x08 and 0x0E

STEP_PRECISION<6:0>:

threshold for acceleration change of two successive sample which is used to update sample_new register in step counter, the actual g value is STEP_PRECISION<6:0>*LSB*16 when STEP_PRECISION<6:0>!=0000000 & !=1111111

When STEP_PRECISION<6:0>=0000000, always use P2P/8

When STEP_PRECISION<6:0>=1111111, always use P2P/16

When STEP_PRECISION<6:0>=?, always use P2P/4

Register 0x14 (STEP_CONF2)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_TIME_LOW<7:0> | | | | | | | | RW | 0x19 |

STEP_TIME_LOW<7:0>:           the short time window for a valid step, the actual time is STEP_TIME_LOW<7:0>*(1/ODR)

Register 0x15 (STEP_CONF3)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| STEP_TIME_UP<7:0> | | | | | | | | RW | 0x00 |

STEP_TIME_UP<7:0>:             time window for quitting step counter, the actual time is     STEP_TIME_UP<7:0>*8*(1/ODR)

Register 0x16 (INT_EN0)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | SIG_STEP_IEN | 1 | 1 | STEP_IEN | HD_EN | RAISE_EN | 1 | RW | 0xB1 |

SIG_STEP_IEN:                1, enable significant step interrupt

                                        0, disable significant step interrupt

STEP_IEN:                       1, enable step valid interrupt

                                        0, disable step valid interrupt

HD_EN:                           1, enable hand-down interrupt

                                        0, disable hand-down interrupt

RAISE_EN:                      1, enable raise-hand interrupt

                                        0, disable raise-hand interrupt

Register 0x17 (INT_EN1)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | INT_DATA_EN | | | | | RW | 0xE0 |

INT_DATA_EN:                1, enable data ready interrupt

                                        0, disable data ready interrupt

Register 0x18 (INT_EN2)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| NO_MOT_EN_Z | NO_MOT_EN_Y | NO_MOT_EN_X | 1 | 1 | ANY_MOT_EN_Z | ANY_MOT_EN_Y | ANY_MOT_EN_X | RW | 0x18 |

| | |
|---|---|
| NO_MOT_EN_Z: | 1, enable no_motion interrupt on Z axis |
| | 0, disable no_motion interrupt on Z axis |
| NO_MOT_EN_Y: | 1, enable no_motion interrupt on Y axis |
| | 0, disable no_motion interrupt on Y axis |
| NO_MOT_EN_X: | 1, enable no_motion interrupt on X axis |
| | 0, disable no_motion interrupt on X axis |
| ANY_MOT_EN_Z: | 1, enable any_motion interrupt on Z axis |
| | 0, disable any_motion interrupt on Z axis |
| ANY_MOT_EN_Y: | 1, enable any_motion interrupt on Y axis |
| | 0, disable any_motion interrupt on Y axis |
| ANY_MOT_EN_X: | 1, enable any_motion interrupt on X axis |
| | 0, disable any_motion interrupt on X axis |

Register 0x19 (INT_MAP0)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | INT1_SIG_STEP | 1 | 1 | INT1_STEP | INT1_HD | INT1_RAISE | INT1_SIG_MOT | RW | 0xB0 |

INT1_SIG_STEP:

1, map significant step interrupt to INT1 pin
0, not map significant step interrupt to INT1 pin

INT1_STEP:

1, map step valid interrupt to INT1 pin
0, not map step valid    interrupt to INT1 pin

INT1_HD:

1, map hand down interrupt to INT1 pin
0, not map hand down interrupt to INT1 pin

INT1_RAISE:

1, map raise hand interrupt to INT1 pin
0, not map raise hand interrupt to INT1 pin

INT1_SIG_MOT:

1, map significant interrupt to INT1 pin
0, not map significant interrupt to INT1 pin

Register 0x1a (INT_MAP1)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| INT1_NO_MOT | 1 | 1 | INT1_DATA | | | 1 | INT1_ANY_MOT | RW | 0x62 |

| | |
|---|---|
| INT1_NO_MOT: | 1, map no_motion interrupt to INT1 pin |
| | 0, not map no_motion interrupt to INT1 pin |
| INT1_DATA: | 1, map data ready interrupt to INT1 pin |
| | 0, not map data ready interrupt to INT1 pin |
| INT1_ANY_MOT: | 1, map any motion interrupt to INT1 pin |
| | 0, not map any motion interrupt to INT1 pin |

Register 0x1b (INT_MAP2)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| 1 | INT2_SIG_STEP | 1 | 1 | INT2_STEP | INT2_HD | INT2_RAISE | INT2_SIG_MOT | RW | 0xB0 |

| | |
|---|---|
| INT2_SIG_STEP: | 1, map significant step interrupt to INT2 pin |
| | 0, not map significant step interrupt to INT2 pin |
| INT2_STEP: | 1, map step valid interrupt to INT2 pin |
| | 0, not map step valid interrupt to INT2 pin |
| INT2_HD: | 1, map hand down interrupt to INT2 pin |
| | 0, not map hand down interrupt to INT2 pin |
| INT2_RAISE: | 1, map raise hand interrupt to INT2 pin |
| | 0, not map raise hand interrupt to INT2 pin |
| INT2_SIG_MOT: | 1, map significant interrupt to INT2 pin |

0, not map significant interrupt to INT2 pin

Register 0x1c (INT_MAP3)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| INT2_NO_MOT | 1 | 1 | INT2_DATA | | | 1 | INT2_ANY_MOT | RW | 0x62 |

| | |
|---|---|
| INT2_NO_MOT: | 1, map no_motion interrupt to INT2 pin |
| | 0, not map no_motion interrupt to INT2 pin |
| INT2_DATA: | 1, map register data ready interrupt to INT2 pin |
| | 0, not map register data ready interrupt to INT2 pin |
| INT2_ANY_MOT: | 1, map any motion interrupt to INT2 pin |
| | 0, not map any motion interrupt to INT2 pin |

Register 0x1d (SIG_STEP_TH)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| STEP_INTERVAL<7:0> | | | | | | | | RW | 0x00 |

STEP_INTERVAL <7:0>: threshold of significant step. When MOD(STEP_CNT, STEP_INTERVAL)=0, SIG_STEP_INT will be generated.

Register 0x1e (raise hand: X_TH Z_TH)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| Z_TH<3:0> | | | | X_TH<3:0> | | | | RW | 0x66 |

| | |
|---|---|
| X_TH<3:0>: | 0~7.5, LSB 0.5 (unit : m/s2) |
| Z_TH<3:0>: | -8~7, LSB 1 (unit : m/s2) |

Register 0x1f

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| STEP_START_CNT<2:0> | | | STEP_COUNT_PEAK<1:0> | | STEP_COUNT_P2P<2:0> | | | RW | 0xA9 |

| | |
|---|---|
| STEP_START_CNT<2:0>: | th_step_pattern = 0/4/8/12/16/24/32/40 |
| STEP_COUNT_PEAK<2:0>: | FIXED_PEAK = 0.05g + 0.05g * STEP_COUNT_PEAK<2:0>. |
| | This FIXED_PEAK is used in algorithm of STEP COUNTER. |
| | STEP_COUNT_PEAK<2> is in register 0x20<4> and |
| | STEP_COUNT_PEAK[2:0]= {0x20[4], 0x1F[4:3]} |
| STEP_COUNT_P2P<2:0>: | FIXED_P2P = 0.3g + 0.1g * STEP_COUNT_P2P<2:0>. |
| | STEP_COUNT_P2P[3:0]= {0x1F[2:0]} |

Register 0x20 (INTPIN_CONF)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| DIS_PU_SENB | DIS_IE_AD0 | EN_SPI3W | STEP_COUNT_PEAK<2> | INT2_OD | INT2_LVL | INT1_OD | INT1_LVL | RW | 0x05 |

| | |
|---|---|
| DIS_PU_SENB: | 1, disable pull-up resistor of PIN_SENB |
| | 0, enable pull-up resistor of PIN_SENB |
| DIS_IE_AD0: | 1, disable input of AD0 |
| | 0, not disable input of AD0 |
| EN_SPI3W: | 1, enable 3W SPI |
| | 0, 4W SPI |
| STEP_COUNT_PEAK<2>: | Definition in 0x1F<4:3> |
| INT2_OD: | 1, open-drain for INT2 pin |
| | 0, push-pull for INT2 pin |
| INT2_LVL: | 1, logic high as active level for INT2 pin |
| | 0, logic low as active level for INT2 pin |
| INT1_OD: | 1, open-drain for INT1 pin |
| | 0, push-pull for INT1 pin |
| INT1_LVL: | 1, logic high as active level for INT1 pin |
| | 0, logic low as active level for INT1 pin |

Register 0x21 (INT_CFG)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| INT_RD_CLR | SHADOW_DIS | DIS_I2C | 1 | 1 | 1 | LATCH_INT_STEP | LATCH_INT | RW | 0x1C |

| | |
|---|---|
| INT_RD_CLR: | 1, clear all the interrupts in latched-mode, when any read operation to any of registers from 0x09 to 0x0D |
| | 0, clear the related interrupts, only when read the register INT_ST (0x09 to 0x0D), |
| | no matter the interrupts in latched-mode, or in non-latched-mode. |
| | Reading 0x09 will clear the register 0x09 only and the others keep the status |
| SHADOW_DIS: | 1, disable the shadowing function for the acceleration data |

0, enable the shadowing function for the acceleration data.
when shadowing is enabled, the MSB of the acceleration data is locked,
when corresponding LSB of the data is reading.
This can ensure the integrity of the acceleration data during the reading.
The MSB will be unlocked when the MSB is read.

DIS_I2C:                 1: disable I2C. Setting this bit to 1 in SPI mode is recommended
                         0: enable I2C
LATCH_INT_STEP:          1, step related interrupt is in latch mode
                         0, step related interrupt is in non-latch mode
LATCH_INT:               1, interrupt is in latch mode
                         0, interrupt is in non-latch mode

Register 0x27 (OS_CUST_X)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| OS_CUST_X<7:0> | | | | | | | | RW | 0x00 |

OS_CUST_X<7:0>:          offset calibration of X axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
                         7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x28 (OS_CUST_Y)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| OS_CUST_Y<7:0> | | | | | | | | RW | 0x00 |

OS_CUST_Y<7:0>:          offset calibration of Y axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
                         7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x29 (OS_CUST_Z)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| OS_CUST_Z<7:0> | | | | | | | | RW | 0x00 |

OS_CUST_Z<7:0>:          offset calibration of Z axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
                         7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x2a (RAISE_WAKE_SUM_TH RAISE_WAKE_DIFF_TH)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| RAISE_WAKE_DIFF_TH<1:0> | | RAISE_WAKE_SUM_TH<5:0> | | | | | | RW | 0xD8 |

RAISE_WAKE_SUM_TH <5:0>:   0 ~ 31.5 (LSB 0.5 m/s2)

| RAISE_WAKE_DIFF_TH<3:0> | UNIT (m/s$^2$) |
|---|---|
| 0 | 0.2 |
| 1 | 0.3 |
| 2 | 0.4 |
| 3 | 0.5 |
| 4 | 0.6 |
| 5 | 0.7 |
| 6 | 0.8 |
| 7 | 0.9 |
| 8 | 1.0 |
| 9 | 1.1 |
| 10 | 1.2 |
| default | 0.2 |

Register 0x2b (RAISE_WAKE_DIFF_TH HD_X_TH HD_Z_TH)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| HD_Z_TH<2:0> | | | HD_X_TH<2:0> | | | RAISE_WAKE_DIFF_TH<3:2> | | RW | 0x7C |

HD_X_TH<2:0> :           hand down x threshold, 0~7 (m/s2)
HD_Z_TH<2:0> :           hand down z threshold, 0~7 (m/s2)

Register 0x2c (MOT_CONF0)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|---|---|---|---|---|---|---|---|---|---|
| NO_MOT_DUR<5:0> | | | | | | ANY_MOT_DUR<1:0> | | RW | 0x00 |

NO_MOT_DUR<5:0>:         no motion interrupt will be triggered when slope < NO_MOT_TH for the times which defined by NO_MOT_DUR<5:0>
                         Duration = (NO_MOT_DUR<3:0> + 1) * 1s, if NO_MOT_DUR<5:4> =b00

Duration = (NO_MOT_DUR<3:0> + 4) * 5s, if NO_MOT_DUR<5:4> =b01
Duration = (NO_MOT_DUR<3:0> + 10) * 10s, if NO_MOT_DUR<5:4> =b1x

ANY_MOT_DUR<1:0>:      any motion interrupt will be triggered when slope > ANY_MOT_TH for (ANY_MOT_DUR<1:0> + 1) samples

Register 0x2d (MOT_CONF1)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| NO_MOT_TH<7:0> | | | | | | | | RW | 0x00 |

NO_MOT_TH<7:0>:      Threshold of no-motion interrupt. The threshold definition is as following
     TH= NO_MOT_TH<7:0> * 16 * LSB

Register 0x2e (MOT_CONF2)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| ANY_MOT_TH<7:0> | | | | | | | | RW | 0x00 |

ANY_MOT_TH<7:0>:      Threshold of any motion interrupt. The threshold definition is as following
     TH= ANY_MOT_TH<7:0> * 16 * LSB

Register 0x2f (MOT_CONF3)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| | | SIG_MOT_TPROOF<1:0> | | SIG_MOT_TSKIP<1:0> | | | SIG_MOT_SEL | RW | 0x00 |

SIG_MOT_TPROOF<1:0>:      00, T_PROOF=0.25s
     01, T_PROOF=0.5s
     10, T_PROOF=1s
     11, T_PROOF=2s
SIG_MOT_TSKIP<1:0>:      00, T_SKIP=1.5s
     01, T_SKIP=3s
     10, T_SKIP=6s
     11, T_SKIP=12s
SIG_MOT_SEL:      1, select significant motion interrupt
     0, select any motion interrupt

Register 0x30

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| MO_BP_CO | STEP_BP_CO | | | | NO_MOT_RST_N | SIG_MOT_RST_N | ANY_MOT_RST_N | RW | 0x1F |

MO_BP_CO:      1, motion detector will use data without OS_CUST
     0, motion detector will use data with OS_CUST
STEP_BP_CO:      1, pedometer will use data without OS_CUST
     0, pedometer will use data with OS_CUST
NO_MOT_RST_N:      0, Reset no motion detector. After reset, user should write 1 back.

SIG_MOT_RST_N:      0, Reset significant motion detector. After reset, user should write 1 back.

ANY_MOT_RST_N:      0, Reset any motion detector. After reset, user should write 1 back.

Register 0x32 (ST)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| SELFTEST_BIT | | | | | SELFTEST_SIGN | BP_AXIS_STEP<1:0> | | RW | 0x00 |

SELFTEST_BIT:      1, self-test enabled. When self-test enabled, a delay of 3ms is necessary for the value settling.
     0, normal
SELFTEST_SIGN:      1, set self-test excitation positive
     0, set self-test excitation negative
BP_AXIS_STEP<1:0>:      11, bypass Z axis, use only X and Y axes data for step counter algorithm
     10, bypass Y axis, use only X and Z axes data for step counter algorithm
     01, bypass X axis, use only Y and Z axes data for step counter algorithm
     00, use all of 3 axes data for step counter algorithm

Register 0x34 (Y_TH YZ_TH_SEL)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| YZ_TH_SEL<2:0> | | | Y_TH<4:0> | | | | | RW | 0x9D |

Y_TH: -16 ~ 15 (m/s2)

| YZ_TH_SEL<2:0> | | | UNIT (m/s2) | | |
|------|------|------|------|------|------|

| 0 | 7.0 |
|---|---|
| 1 | 7.5 |
| 2 | 8.0 |
| 3 | 8.5 |
| 4 | 9.0 |
| 5 | 9.5 |
| 6 | 10.0 |
| 7 | 10.5 |

Register 0x35 (RAISE_WAKE_PERIOD)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| RAISE_WAKE_PERIOD<7:0> | | | | | | | | RW | 0x81 |

RAISE_WAKE_PERIOD<10:0>:     * ODR period = wake count (EX. ODR = 1ms, 0X35 = 100 ➔ wake count = 0.1 sec)

Register 0x36 (SR)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| SOFT_RESET | | | | | | | | RW | 0x00 |

SOFT_RESET:                0xB6, soft reset all of the registers. After soft-reset, user should write 0x00 back

Register 0x3e (RAISE_WAKE_TIMEOUT_TH)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| RAISE_WAKE_TIMEOUT_TH<7:0> | | | | | | | | RW | 0x00 |

Register 0x3f (RAISE_WAKE_TIMEOUT_TH RAISE_WAKE_PERIOD RAISE_WAKE_EN)

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W | Default |
|------|------|------|------|------|------|------|------|-----|---------|
| | RAISE_WAKE_PERIOD<10:8> | | | RAISE_WAKE_TIMEOUT_TH<11:8> | | | | RW | 0x02 |

RAISE_WAKE_TIMEOUT_TH<11:0> * ODR period = timeout count (EX. ODR = 1ms, 0X3e = 100 ➔ timeout count = 0.1 sec)

## ORDERING INFORMATION

| Ordering Number | Temperature Range | Package | Packaging |
|-----------------|-------------------|---------|-----------|
| QMA7981-TR | -40℃~85℃ | LGA-12 | Tape and Reel: 5k pieces/reel |

**Caution**
This part is sensitive to damage
by electrostatic discharge. Use ESD
precautionary procedures when
touching, removing or inserting.

**CAUTION: ESDS CAT. 1B**

### FIND OUT MORE

For more information on QST's Accelerometer Sensors contact us at 86-21-50497300.

The application circuits herein constitute typical usage and interface of QST product. QST does not provide warranty or assume liability of customer-designed circuits derived from this description or depiction.

QST reserves the right to make changes to improve reliability, function or design. QST does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

ISO9001 : 2015

China Patents 201510000399.8, 201510000425.7, 201310426346.3, 201310426677.7, 201310426729.0, 201210585811.3 and 201210553014.7 apply to the technology described.