QMI8A01
Low Noise, Wide Bandwidth 6D Inertial Measurement Unit with Motion Co-Processor

Features

- Low 13 mdps/√Hz gyroscope noise, low-latency, and wide bandwidth
- Low 150μg/√Hz accelerometer noise
- Host (slave) interface supports MIPI™ I3C, I²C, and 3-wire or 4-wire SPI
- Accelerometer and gyroscope sensors feature signal processing paths with digitally programmable data rates and filtering
- 3-axis gyroscope and 3-axis accelerometer in a small 3.0 x 3.5 x 1.1 mm 14-pin LGA package
- Large 1536-byte FIFO can be used to buffer sensor data to lower system power dissipation
- Integrated Tap, Any-Motion, No-Motion, Significant-Motion detection
- Large sensor dynamic ranges from ±16°/s to ±2048°/s for gyroscope and ±2 g to ±16 g for accelerometer
- Low power modes for effective power management
- Digitally programmable sampling rate and filters
- Embedded temperature sensor
- Wide extended operating temperature range (-40°C to 85°C)

Description

The QMI8A01 is a complete 6D MEMS inertial measurement unit (IMU). With tight board-level gyroscope sensitivity of ±1%, gyroscope noise density of 13 mdps/√Hz, and low latency, the QMI8A01 is ideal for consumer and industrial applications.

The QMI8A01 incorporates a 3-axis gyroscope and a 3-axis accelerometer. It provides a host-processor interface supporting I3C, I²C and 3-wire or 4-wire SPI.

With its built-in digital functionality, low power, and small size, the QMI8A01 is the ideal part for applications requiring motion-based functionality.

Applications

- Smartphones
- Game controllers, remote controls and pointing devices
- Robotic vacuums
- E-bikes and scooters
- Bluetooth headsets
- Automotive security systems
- Toys
- Portrait-landscape display control

Table of Contents

1 General Information

1.1 Ordering Information

Table 1. Ordering Information

Part Number	Package	Packing Method
QMI8A01	LGA14	Tape & Reel

1.2 Marking Information

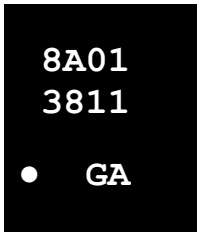
ROW	EXAMPLE	CODE/EXPLANATION
1		Line 1: DDDD – Device code
2		Line 2: YWLL – Y (Year code), W (biweekly code), LL (2-digits, Lot indication)
3		Line 3: GA – G (Assembly location), A (Product revision)

Figure 1. Top Mark

1.3 Internal Block Diagram

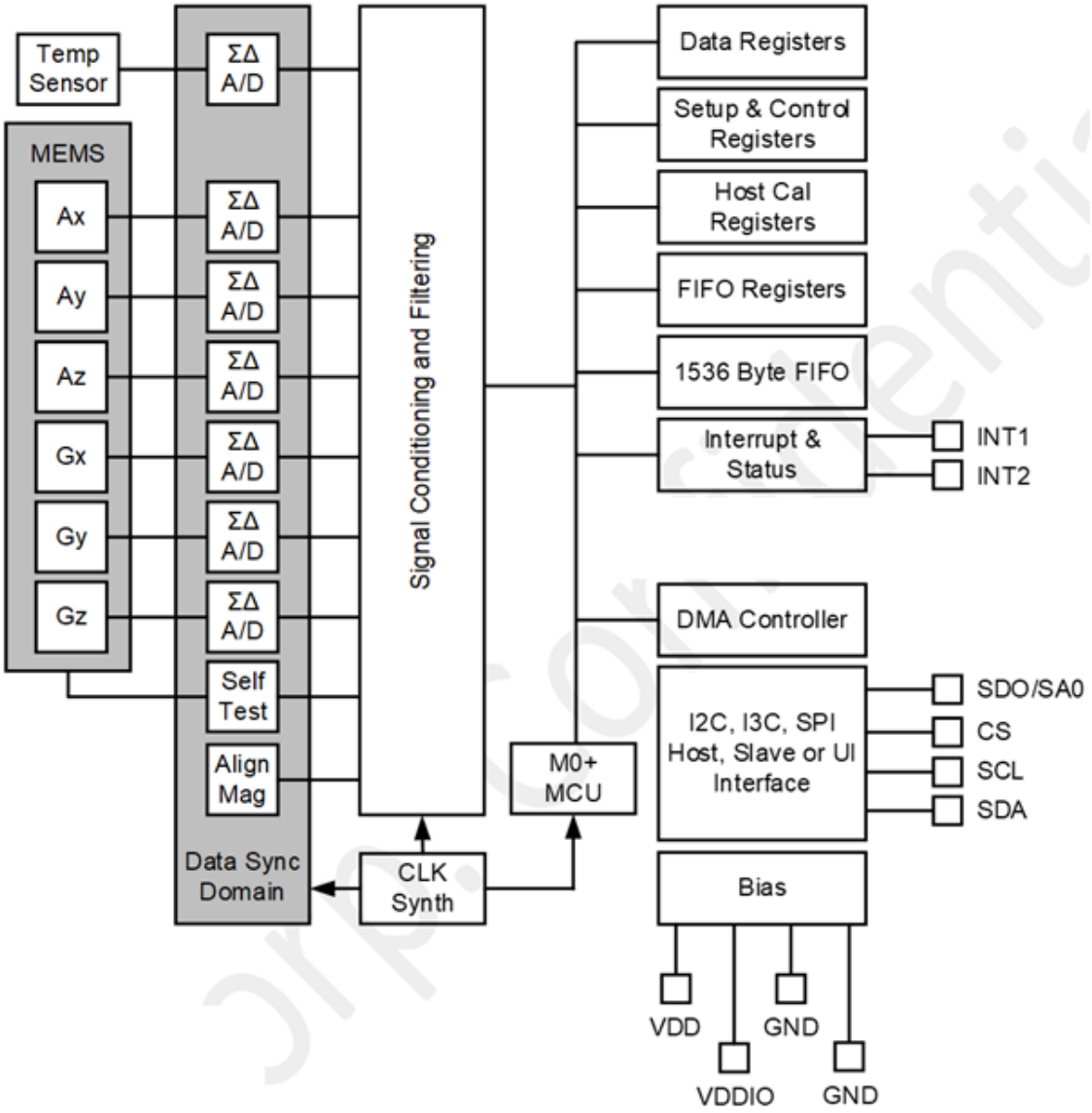


Figure 2. Internal Block Diagram

1.4 Interface Operating

The QMI8A01 can operate in below mode, as shown in the *Figure 3*.

The QMI8A01 is a slave device to a host processor that communicates to it using one of the following interfaces: I^2C, I^3C, and SPI (3-wire or 4-wire modes). This slave relationship to the host is the same for all operating modes.

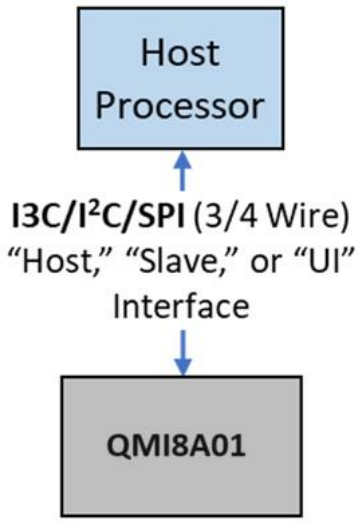


Figure 3. Operating Mode

1.5 Application Diagrams

The typical application diagrams are shown in this section.

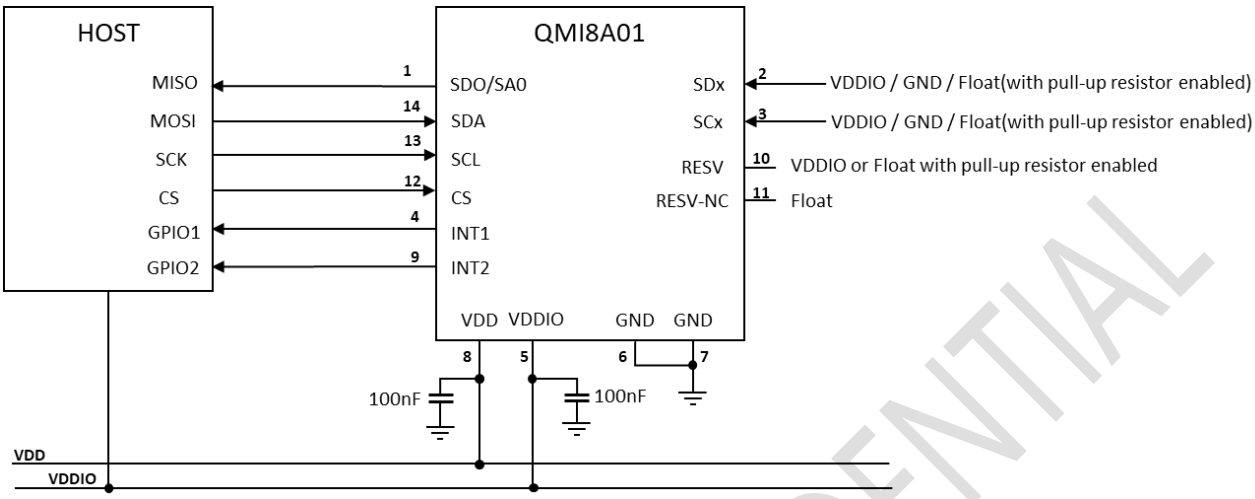


Figure 4. 4-Wire SPI-UI mode

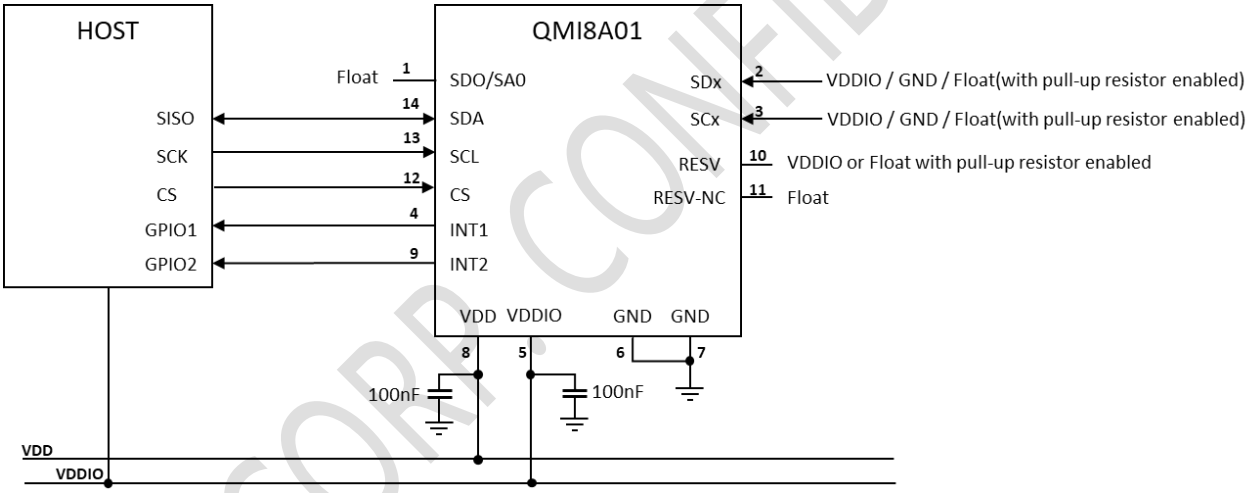


Figure 5. 3-Wire SPI-UI mode

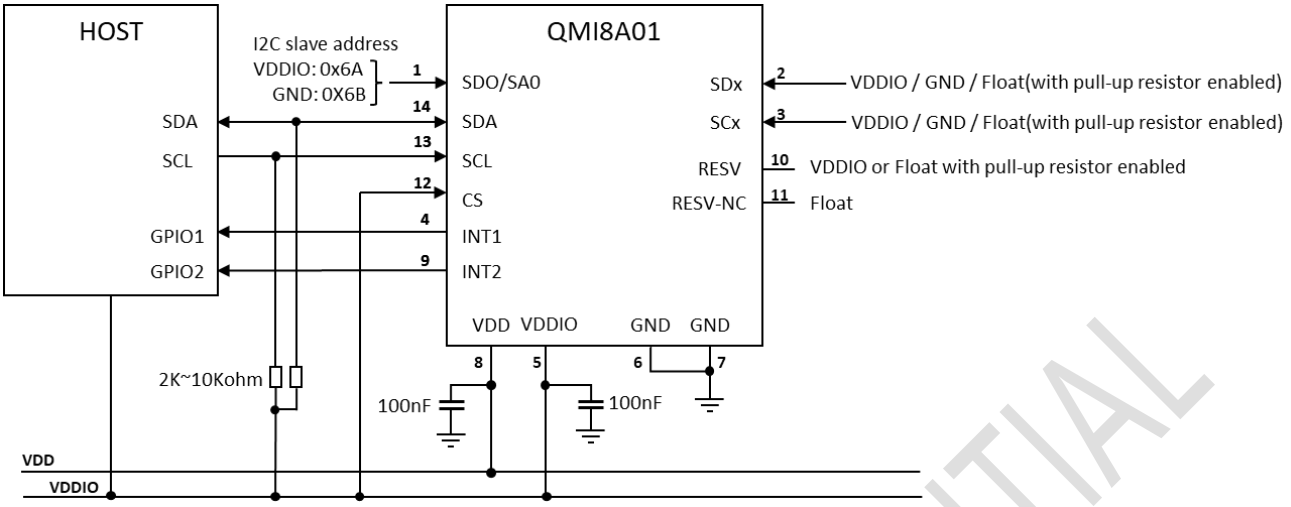


Figure 6. I3C/I2C-UI mode

There is an internal 200Kohm pull-up resistor on the SCL, SDA, CS, SDx, SCx, RESV (Pin10). By default, all those pull-up resistors are enabled. All the resistors can be disabled by CTRL9 command, refer to *1.6 Package & Pin Information* and *5.10.6 CTRL_CMD_SET_RPU* for details.

There is an internal 200Kohm pull-up resistor on SDO/SA0 pin. It is enabled during Power-On Reset or Soft-Reset, is automatically disabled after detecting the I2C slave address during the Reset Process. Therefore, in I2C/I3C mode leave the SDO/SA0 float or connect it to High level (recommended, to provide a stable level), will set the I2C slave address/I3C static address to 0x6A. And connect it to Low level, will set the I2C slave address/ I3C static address to 0x6B.

In 3-wire SPI mode, leave the SDO/SA0 pin float.

SCx and SDx can be connected to VDDIO or Logic High, GND or Logic Low, or be left float if internal pull-up resistors are enabled.

RESV (Pin 10) should NOT be connected to GND or Logic Low. It can be connected to VDDIO or Logic High, or leave it externally floating and enable the internal pull-up resistor (by default the pull-up resistor is enabled). Connecting it to VDDIO is preferred, which can provide a stable High level.

RESV-NC (Pin 11) is by default an output pin, should be float (no connection). In case of the necessity to connect it to High or Low level, the RESV(Pin 10) should be firmly connected to VDDIO, providing a stable High level, to disable the output of Pin 11.

Table 2 describes the names for the pins in different functions. The later descriptions will directly use the function name in different scenarios instead of the pin name.

Table 2. Pin Name Mapped to Function Name

Pin Number	Type	Pin Name	Function Name in 4-wire SPI	Function Name in 3-wire SPI	Function Name in Host I2C / I3C
1	O	SDO/SA0	SDO		SA0
12	I	CS	CS	CS	
13	IO	SCL	SPC	SPC	SCL
14	IO	SDA	SDI	SDIO	SDA

1.6 Package & Pin Information

The pinout of the QMI8A01 is shown in Figure 7 and Figure 8. The pin names and functionality are detailed in Table 3. The pin functionality is dictated by the QMI8A01's operating mode, as described in 1.5 Application Diagrams.

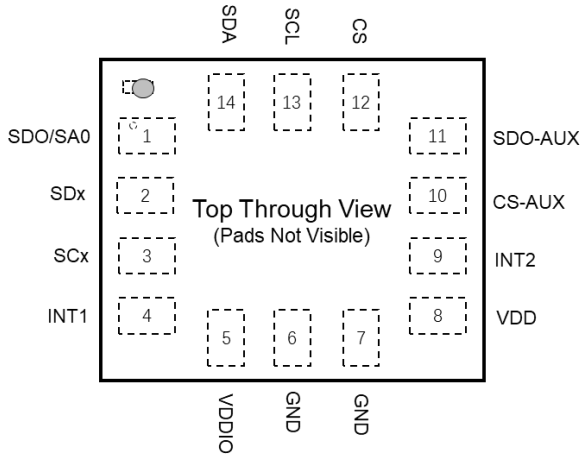


Figure 7. Pins Face Down (Top View)

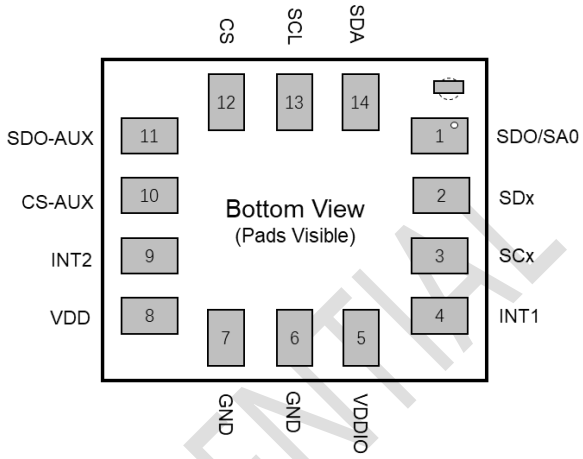


Figure 8. Pins Face Up (Bottom View)

Table 3. Pin Definitions

Pin Number	Type	Pin Name	Function
1	O	SDO/SA0(1)(3)	SPI-UI Data Out (SDO) in SPI-UI 4-Wire Mode SA0, I^2C Slave address and I^3C Static address select: If SA0 = 0, I^2C / I^3C address = 0x6B If SA0 = 1, I^2C / I^3C address = 0x6A
2	IO	SDx(1)(2)	Reserved. Connect to VDDIO, GND or No Connection
3	IO	SCx(1)(2)	Reserved. Connect to VDDIO, GND or No Connection
4	O	INT1	Programmable Interrupt 1
5	I	VDDIO	Power Supply for IO Pins
6	I	GND	Ground (0 V supply)
7	I	GND	Ground (0 V supply)
8	I	VDD	Power supply
9	O	INT2	Programmable Interrupt 2 (INT2) / Data Ready (DRDY)
10	IO	RESV(1)(2)	Reserved. Connect to VDDIO or Logic High, or No Connection and enable (by default) internal pull up resistor. *Refer to 1.5* Application Diagrams.
11	I	RESV-NC	Reserved. Float is preferred. *Refer to 1.5* Application Diagrams.
12	I	CS(1)(2)	I^2C/ I^3C /SPI-UI selection Pin. (If 1: I^2C-UI Mode: I^2C/I^3C communication enabled, SPI idle mode) (If 0: SPI-UI mode: I^2C/I^3C disabled)
13	IO	SCL(1)(2)	I^2C/I^3C-UI Data (SDA) in I^2C/ I^3C mode SPI-UI Serial Clock (SPC) (3) in SPI mode
14	IO	SDA(1)(2)	I^2C/I^3C-UI Data (SDA) SPI-UI Data In (SDI) (3) in 4 wire Mode SPI-UI Data IO (SDIO) (3) in 3 Wire Mode

Notes:

1. This pin has an internal 200KΩ pull up resistor.
2. The internal pull-up resistor can be disabled by CTRL9 command (CTRL_CMD_SET_RPU). Refer to ***5.10.6*** *CTRL_CMD_SET_RPU* for details.
3. Refer to Section 14 for detailed Host Serial Interface configuration information.

1.7 Recommended External Components

Table 4. Recommended External Components

Component	Description	Parameter	Typical
C_{p1}	Capacitor	Capacitance	100 nF
C_{p2}	Capacitor	Capacitance	100 nF
R_{pu}(4)	Resistor	Resistance	2KΩ ~ 10 kΩ

Note:

4. R_{pu} resistors are only needed when the Host Serial Interface is configured for I^2C (see I2C Interface section). They are not needed when the Host Serial Interface is configured for SPI or I3C. If pull-up resistors are used on SCL and SDA, then SPI, I3C and I^2C Modes are all possible. If a pull-down resistor is used on SA0, an alternate slave address is used for I^2C. SPI and I3C modes will be unaltered with the use of pull-up resistors for I^2C. Additionally, a suitable pull up resistance (R_{pu}) value should be selected, accounting for the tradeoff between current consumption and rise time.

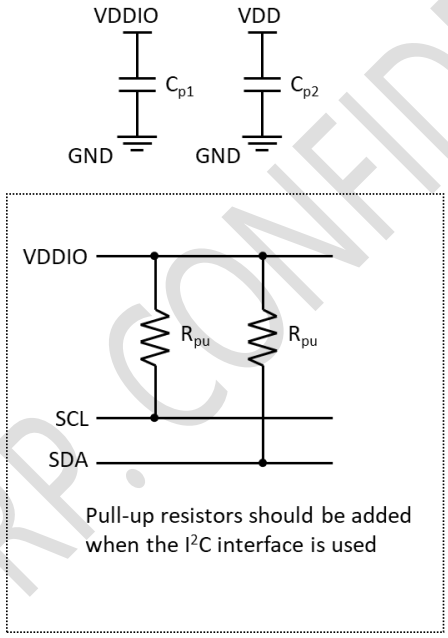


Figure 9. Typical Electrical Connections

2 QMI8A01 Chip Orientation Coordinate System

The QMI8A01 uses a right-handed coordinate system as the basis for the sensor frame of reference. Acceleration (a_x, a_y, a_z) are given with respect to the X-Y-Z coordinate system shown above. Increasing accelerations along the positive X-Y-Z axes are considered positive. Angular Rate (ω_x, ω_y, ω_z) in the counterclockwise direction around the respective axis are considered positive.

Figure 10 shows the various frames of reference and conventions for using the QMI8A01.

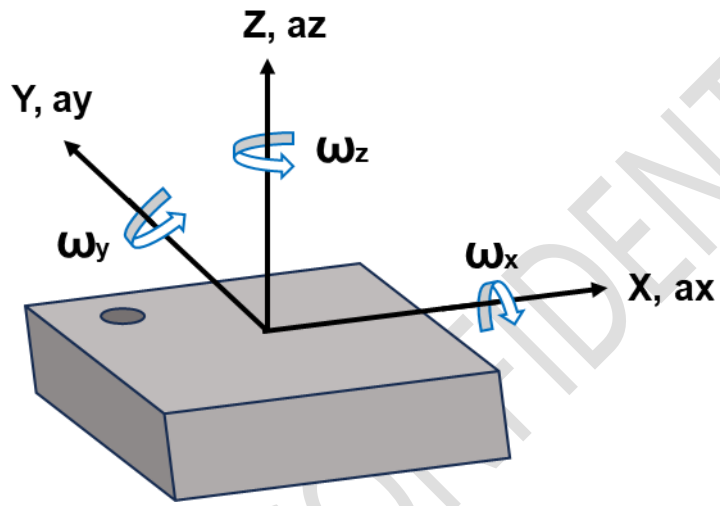


Figure 10. Chip Orientation Coordinate System

3 System, Electrical and Electro-Mechanical Characteristics

3.1 Absolute Maximum Ratings

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions. Stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only.

Table 5. Absolute Maximum Ratings

Symbol	Parameter		Min.	Max.	Unit
T_{STG}	Storage Temperature		-40	+125	°C
T_{Pmax}	Lead Soldering Temperature, 10 Seconds			+260	°C
VDD	Supply Voltage		-0.3	3.6	V
VDDIO	I/O Pins Supply Voltage		-0.3	3.6	V
S_g(5)	Acceleration g for 0.2 ms (Un-powered)			10,000	g
ESD(6)	Electrostatic Discharge Protection Level	Human Body Model per JES001-2014	±2000		V
		Charged Device Model per JESD22-C101	±500		

Notes:

5. This is a mechanical shock (g) sensitive device. Proper handling is required to prevent damage to the part.
6. This is an ESD-sensitive device. Proper handling is required to prevent damage to the part.

3.2 Recommended Operating Conditions

The Recommended Operating Conditions table defines the conditions for device operation. Recommended operating conditions are specified to ensure optimal performance.

Table 6. Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VDD	Supply Voltage	1.71	1.8	3.6	V
VDDIO	I/O Pins Supply Voltage	1.71	1.8	3.6	V
V_{IL}	Digital Low Level Input Voltage			0.3 *VDDIO	V
V_{IH}	Digital High Level Input Voltage	0.7 *VDDIO		VDDIO + 0.3	V
V_{OL}	Digital Low Level Output Voltage			0.1 *VDDIO	V
V_{OH}	Digital High Level Output Voltage	0.9 *VDDIO			V
V_{POR_RLS}	POR Threshold Voltage		1.1		V
T	Operating Temperature	-40		+85	°C

3.3 Power On Sequence of VDDIO and VDD

3.3.1 Power Off Condition

To ensure the POR block functions well, the constrains on the condition for power off (for both VDD and VDDIO) in Table 7. Refer to Figure 11 and Figure 12 for the illustration.

Table 7. Power Off Condition

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Power Off Voltage	SDV	Voltage that Device Considers to be Power Down, valid for VDD & VDDIO.			0.2	V
Power Off Interval	PINT	Time Period Required for Voltage Lower Than SDV to Enable Next POR, valid for VDD & VDDIO.	100			μs

3.3.2 Power-On Reset (POR)

Once the VDD & VDDIO are powered from 0V to a certain level, the internal power voltage detector will trigger the Power-On Reset (POR) automatically, and then exit/release the POR mode as the VDD voltage rises over the POR Threshold (approximately 1.1V). Refer to *Table 6*.

After POR release, there will be approximately a 200us Startup Delay, followed by the QMI8A01 Initialization.

The instability of VDDIO & VDD power lines, especially the power increase/drop with high slew rate, would interfere with the QMI8A01 Initialization and operation. Therefore, there should be no sudden transients and spikes on power lines after Startup Delay, to make sure the Initialization and later operations are properly implemented.

Normally it takes within approximately 15ms (refer to *System Turn On Time* in *Table 8* and *Table 9*) for QMI8A01 to finish the Initialization and during which, there should be no write/configuration to QMI8A01, to prevent possible interference and failure.

Note that the Software Reset is triggered by the reset command that host write to QMI8A01, which means POR is not involved, while the following sequence are similar (Startup Delay and Initialization) to POR. Refer to *7.4* for more details.

3.3.3 VDDIO and VDD Are Driven by Single Power

As shown in *Figure 11*, when QMI8A01 is driven by single/same power line, the power line should ramp up from POR Threshold (approximately 1.1V) to Min Operation Voltage(1.71V) with slew rate higher than 40V/s, otherwise the Initialization might fail, and the QMI8A01 I is not guaranteed to work.

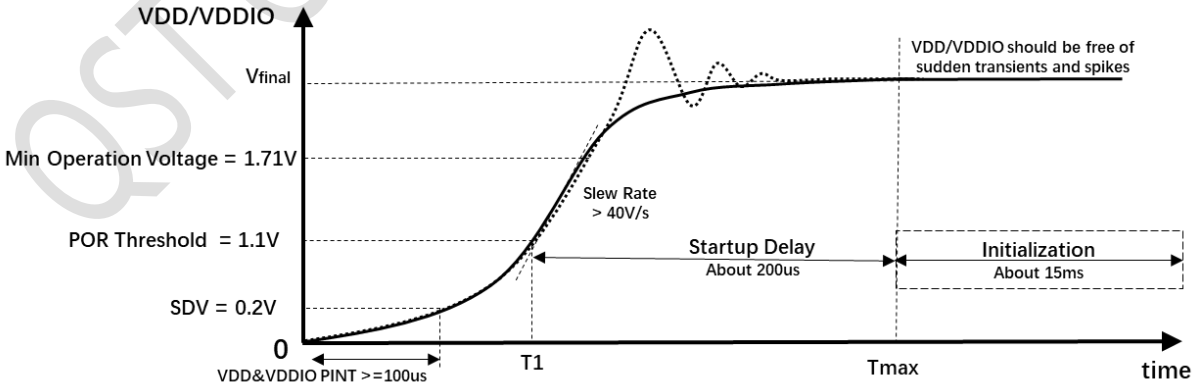


Figure 11. Power On Process When Driven by Single Power Line

3.3.4 VDDIO and VDD Are Driven by Separate Power Lines

As shown in *Figure 12*, when QMI8A01 is driven by separate power lines, the VDDIO should be always powered ahead of (no later than) VDD, which means:

VDDdelay >= 0, VDDIO should be driven no later than VDD

Note that the VDDdelay starts from the point when VDDIO rises over the POR Threshold and ends at the VDD rises over the POR Threshold. It should always be non-negative.

The power lines should ramp up from POR Threshold (approximately 1.1V) to Min Operation Voltage (1.71V) with slew rate higher than 40V/s, otherwise the Initialization might fail, and QMI8A01 might not work properly.

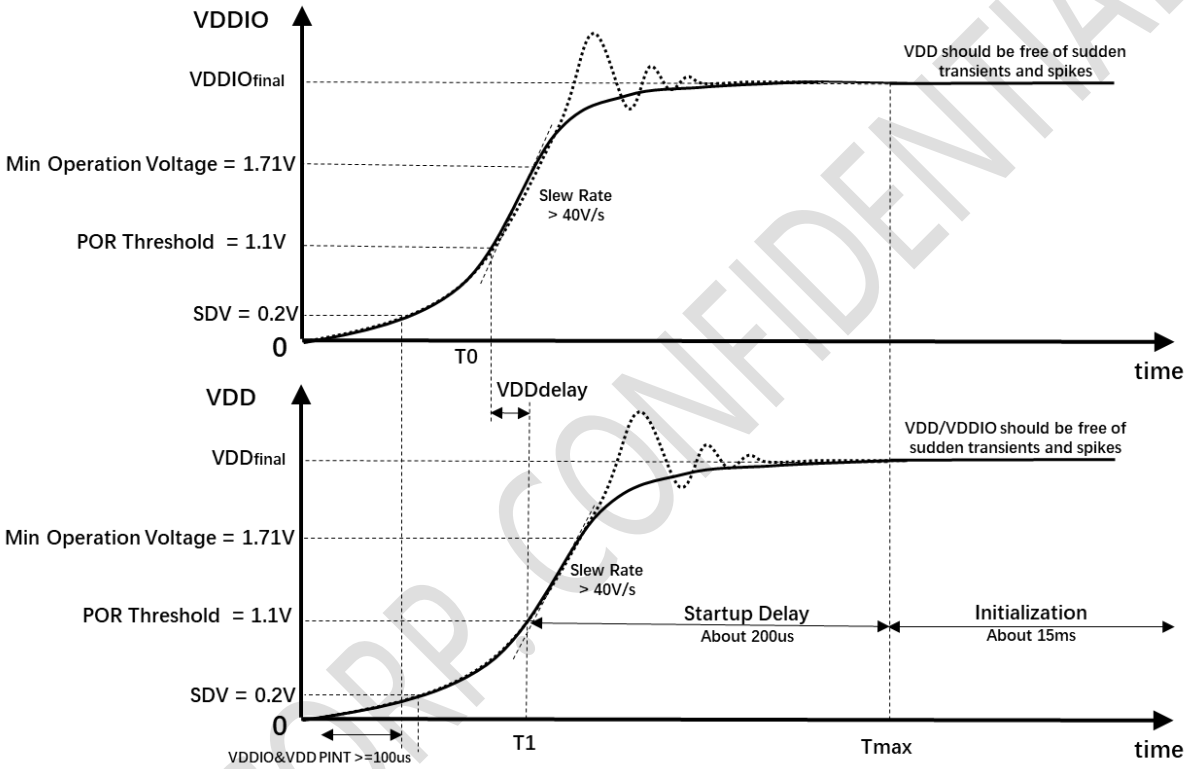


Figure 12. Power On Process When Driven by Separate Power Lines

3.4 Electro-Mechanical Specifications

VDD = VDDIO = 1.8 V, T = 25°C unless otherwise noted.

Table 8. Accelerometer Electro-Mechanical Specifications

Subsystem	Parameter	Typical		Unit	Comments
Accelerometer	Noise Density[7]	150		μg/√Hz	High-Resolution Mode
	Sensitivity Scale Factor	Scale Setting	Sensitivity	LSB/g	16-Bit Output
		±2 g	16,384		
		±4 g	8,192		
		±8 g	4,096		
		±16 g	2,048		
	Cross-Axis Sensitivity	±1		%	
	Temperature Coefficient of Offset (TCO) [7]	±0.5		mg/°C	Over-Temperature Range of -40°C to 85°C, at Board Level
	Temperature Coefficient of Sensitivity (TCS) [7]	±0.0025		%/°C	Over-Temperature Range of -40°C to 85°C, at Board Level
	Initial Offset Tolerance	±25		mg	Board Level
	Initial Sensitivity Tolerance	±0.4		%	Board Level
	Non-Linearity[7]	±0.75		%	Best Fit Line
	System Turn On Time[7][8]	15		ms	From Software Reset, No Power, or Power Down to Power-on Default state = t0 in Figure 16
	Accel Turn On Time[7][9]	3 ms + 3/ODR		ms	Accel Turn on from Power-On Default state or from Low Power state = t2 + t5 in Figure 16.

Note:

7. This parameter is evaluated based on samples in characterization, not tested on all parts in production.
8. System Turn-On Time defines the initialization duration of QMI8A01, it starts approximately 200us later than the release of POR (Power-On Reset) or the Software Reset. Refer to *3.3* and *7.4* for details.
9. Note that the settling time of the accelerometer data is minimum 3/ODR and will be longer according to different sensor configuration cases for example, when the accelerometer data filter is enabled. For more information, and for application-specific cases please consult the QST FAE team.

Table 9. Gyroscope Electro-Mechanical Specifications

Subsystem	Parameter	Typical		Unit	Comments
Gyroscope	Sensitivity	Scale Setting	Sensitivity	LSB/dps	16-Bit Output
		±16 dps	2048		
		±32 dps	1024		
		±64 dps	512		
		±128 dps	256		
		±256 dps	128		
		±512 dps	64		
		±1024 dps	32		
		±2048 dps	16		
	Natural Frequency	22.42		kHz	Precision +/- 2%(typical)
	Noise Density(7)	13		mdps/√Hz	High-Resolution Mode
	Non-Linearity(7)	±0.2		%	
	Cross-Axis Sensitivity	±2		%	
	g-Sensitivity	±0.1		dps/g	
	System Turn On Time(7)(10)	15		ms	From Software Reset, No Power, or Power Down to Power-on Default state = t0 in Figure 16
	Gyro Turn On Time(7)(11)	150 ms + 3/ODR		ms	Gyro Turn on from Power-On Default = t1 + t5 in Figure 16
	Temperature Coefficient of Offset (TCO) (7)	X/Y: ±0.025 Z: ±0.01		dps/°C	Over-Temperature Range of -40°C to 85°C, at Board Level
	Temperature Coefficient of Sensitivity (TCS) (7)	X/Y: ±0.035 Z: ±0.01		%/°C	Over-Temperature Range of -40°C to 85°C, at Board Level
	Initial Offset Tolerance	X/Y: ±5 Z: ±1		dps	Board Level
	Initial Sensitivity Tolerance	X/Y: ±0.75 Z: ±0.5		%	Board Level

Note:

10. System Turn-On Time defines the initialization duration of QMI8A01, it starts approximately 200us later than the release of POR (Power-On Reset) or the Software Reset. Refer to *3.3* and *7.4* for details.
11. Note that the settling time of the gyroscope data is minimum 3/ODR and will be longer according to different sensor configuration cases for example, when the gyroscope data filter is enabled. For more information, and for application-specific cases please consult the QST FAE team.

3.5 Accelerometer Programmable Characteristics

VDD = VDDIO = 1.8 V, T = 25°C unless otherwise noted. Typical numbers are provided below unless otherwise noted.

If only accelerometer is enabled, the ODR frequency is derived from the internal oscillator. If both accelerometer and gyroscope (6DOF mode) are enabled, the ODR frequency is derived from the natural frequency of gyroscope. *Table 10* shows the two ODR frequencies, which can be referenced for later descriptions in the datasheet. Refer to section *5.3* for detailed ODR configuration.

RMS noise can be calculated based on the noise density and the bandwidth.

Table 10. Accelerometer Noise Density

Mode	High-Resolution									Low-Power				Unit
ODR (Accel only)				1000	500	250	125	62.5	31.25	128	21	11	3	Hz
ODR (Accel + Gyro)	7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025					Hz
Typical Noise Density	150	150	150	150	150	150	150	150	150	125	180	285	700	$\mu g/\sqrt{Hz}$

Table 11. Accelerometer Filter Characteristics (Accelerometer only)(12)

Mode	High-Resolution									Low-Power				Unit
ODR	8000	4000	2000	1000	500	250	125	62.5	31.25	128	21	11	3	Hz
Bandwidth (Default, 27.5% of ODR)	NA	NA	NA	275	137.5	68.8	34.4	17.2	8.6	35.2	5.8	3.0	0.8	
Bandwidth with Low-Pass Filter Enabled Mode 00 (2.66% of ODR)	NA	NA	NA	26.6	13.3	6.7	3.3	1.7	0.8	3.4	0.6	0.3	0.1	
Bandwidth with Low-Pass Filter Enabled Mode 01 (3.63% of ODR)	NA	NA	NA	36.3	18.2	9.1	4.5	2.3	1.1	4.6	0.8	0.4	0.1	
Bandwidth with Low-Pass Filter Enabled Mode 10 (5.39% of ODR)	NA	NA	NA	53.9	27	13.5	6.7	3.4	1.7	6.9	1.1	0.6	0.2	
Bandwidth with Low-Pass Filter Enabled Mode 11 (13.37% of ODR)	NA	NA	NA	133.7	66.9	33.4	16.7	8.4	4.2	17.1	2.8	1.5	0.4	

Note:

12. When only accelerometer is enabled, the ODR is derived from the internal oscillator, rather than the nature frequency of Gyroscope.

Table 12. Accelerometer Filter Characteristics (6DOF)[13]

Mode	High-Resolution									Low-Power				Unit
ODR	7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025	128	21	11	3	Hz
Bandwidth (Default, 27.5% of ODR)	1973.0	986.5	493.2	246.6	123.3	61.7	30.8	15.4	7.7	NA	NA	NA	NA	
Bandwidth with Low-Pass Filter Enabled Mode 00 (2.66% of ODR)	190.8	95.4	47.7	23.9	11.9	6.0	3.0	1.5	0.7	NA	NA	NA	NA	
Bandwidth with Low-Pass Filter Enabled Mode 01 (3.63% of ODR)	260.4	130.2	65.1	32.6	16.3	8.1	4.1	2.0	1.0	NA	NA	NA	NA	
Bandwidth with Low-Pass Filter Enabled Mode 10 (5.39% of ODR)	386.7	193.4	96.7	48.3	24.2	12.1	6.0	3.0	1.5	NA	NA	NA	NA	
Bandwidth with Low-Pass Filter Enabled Mode 11 (13.37% of ODR)	959.2	479.6	239.8	119.9	60.0	30.0	15.0	7.5	3.7	NA	NA	NA	NA	

Note:

13. When both accelerometer and gyroscope are both enabled, all frequencies are synchronized to the nature frequency of gyroscope.

3.6 Gyroscope Programmable Characteristics

VDD = VDDIO = 1.8 V, T = 25°C, and represent typical numbers unless otherwise noted. All frequencies are synchronized to the gyroscope nature frequency.

Table 13 shows the noise density of gyroscope output over different ODR configurations.

The typical bandwidths of gyroscope over different ODR settings are listed in *Table 14*.

RMS noise can be calculated based on the noise density and the bandwidth.

Table 13. Gyroscope Noise Density

Mode	High-Resolution									Unit
ODR (Gyro and/or Accel)	7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025	Hz
Typical Noise Density	RSV(14)	RSV(14)	RSV(14)	13	13	13	13	13	13	mdps/√Hz

Note:

14. The gyroscope noise increases significantly when the ODR is set above 1KHz. It is not recommended to set the ODR beyond 1KHz for the applications that are impacted by gyroscope noise. For more information, please contact the QST FAE team.

Table 14. Gyroscope Filter Characteristics

Mode	High-Resolution									Unit
ODR	7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025	Hz
Bandwidth (Default, 27.5% of ODR)	1973.0	986.5	493.2	246.6	123.3	61.7	30.8	15.4	7.7	
Bandwidth with Low-Pass Filter Enabled Mode 00 (2.66% of ODR)	190.8	95.4	47.7	23.9	11.9	6.0	3.0	1.5	0.7	
Bandwidth with Low-Pass Filter Enabled Mode 01 (3.63% of ODR)	260.4	130.2	65.1	32.6	16.3	8.1	4.1	2.0	1.0	
Bandwidth with Low-Pass Filter Enabled Mode 10 (5.39% of ODR)	386.7	193.4	96.7	48.3	24.2	12.1	6.0	3.0	1.5	
Bandwidth with Low-Pass Filter Enabled Mode 11 (13.37% of ODR)	959.2	479.6	239.8	119.9	60.0	30.0	15.0	7.5	3.7	

3.7 Electrical Characteristics

VDD = VDDIO = 1.8 V, T = 25°C unless otherwise noted.

Table 15. Electrical Subsystem Characteristics

Symbol	Parameter		Min.	Typ.	Max.	Unit
f_{SPC}	Host SPI Interface Speed				15	MHz
f_{SCL}	Host I^2C Interface Speed (standard mode and Fast Mode are supported)				400	kHz
f_{SCL3}	Host I^3C Interface Speed	Standard Data Rate (SDR)			12.5	MHz

3.8 Current Consumption

VDD = VDDIO = 1.8 V, T = 25°C unless otherwise noted. IDD Current refers to the current flowing into the VDD pin. Typical numbers are provided below.

Table 16. Current Consumption for Accelerometer Only Sensor Mode (Gyroscope Disabled)

Mode		High-Resolution						Low-Power				Unit
ODR		1000	500	250	125	62.5	31.25	128	21	11	3	Hz
Typical Overall IDD Current	Filters Disabled (aLPF=0)	182	155	142	134	133	132	55	42	35	30	μA
	Filters Enabled (aLPF=1)	182	155	142	134	133	132	55	42	35	30	

Table 17. Current Consumption for Gyroscope Only Sensor Mode (Accelerometer Disabled)

Mode		High-Resolution								Unit
ODR		7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025	Hz
Typical Overall IDD Current	Filters Disabled (gLPF=0)	908	861	748	689	659	656	654	653	651	μA
	Filters Enabled (gLPF=1)	916	863	748	689	659	656	654	653	651	

Table 18. Current Consumption for 6DOF Sensor Mode (Accelerometer and Gyroscope Enabled)

Mode		High-Resolution								Unit
ODR		7174.4	3587.2	1793.6	896.8	448.4	224.2	112.1	56.05	28.025	Hz
Typical Overall IDD Current	Filters Disabled (aLPF=0; gLPF=0)	1004	956	843	786	757	754	752	751	750	μA
	Filters Enabled (aLPF=1; gLPF=1)	1031	970	850	789	758	756	753	751	750	

3.9 Temperature Sensor

The QMI8A01 is equipped with an internal 16-bit embedded temperature sensor that is automatically turned on by default whenever the accelerometer or gyroscope is enabled. The temperature sensor is used internally to correct the temperature dependency of calibration parameters of the accelerometer and gyroscope. The temperature compensation is optimal in the range of -40˚C to 85˚C with a resolution of 0.0625˚C (1/16 ˚C) or inversely, 16 LSB/ ˚C.

The QMI8A01 outputs the internal chip temperature that the HOST can read. The output is 16 bits, with a (1/256)˚C per LSB resolution. To read the temperature, the HOST needs to access the TEMP register *(see TEMP_L and TEMP_H in Data Output Registers in Table 29).*

The calculation formula is:

$$T = (TEMP\_H * 256 + TEMP\_L) / 256$$

Note that the TEMP_H is a signed value, that defines the sign of T. To read the temperature sensor data properly, the Host is expected to follow the guidelines in 11 Locking Mechanism.

Table 19. Temperature Sensor Specifications

Subsystem	Parameter	Typical	Unit
Digital Temperature Sensor	Range	-40 to +85	°C
	Internal Resolution	16	Bits
	Internal Sensitivity	256	LSB/°C
	Output Register Width	16	Bits
	Output Sensitivity	256	LSB/°C
	Refresh Rate	8	Hz

4 Register Map Overview

The QMI8A01 UI registers enable programming and control of the inertial measurement unit and associated on-chip signal processing. These registers are accessed through the UI interface – either SPI (4 wires or 3 wires) I3C, or I^2C.

4.1 UI Register Map Overview

UI register map may be classified into the following register categories:

- Chip Information Registers
- Setup and Control Registers: control various aspects of the IMU
- Host Controlled Calibration Registers: control and configure various aspects of the IMU via the host command interface called CTRL9
- Count Register for time stamping the sensor samples
- Sensor data registers
- FIFO Registers: to set up the FIFO and detect data availability and over-run
- Activity Detection status registers
- General Purpose Registers

Table 20. UI Register Overview

Name	Type	Register Address			Default	Comment
		Dec	Hex	Binary	Binary	
General Purpose Registers						
WHO_AM_I	r	0	00	00000000	00000101	Device Identifier
REVISION_ID	r	1	01	00000001	01111100	Device Revision ID
Setup and Control Registers						
CTRL1	rw	2	02	00000010	00100000	SPI Interface and Sensor Enable
CTRL2	rw	3	03	00000011	00000000	Accelerometer: Output Data Rate, Full Scale, Self-Test
CTRL3	rw	4	04	00000100	00000000	Gyroscope: Output Data Rate, Full Scale, Self-Test
Reserved	rw	5	05	00000101	00000000	Reserved
CTRL5	rw	6	06	00000110	00000000	Low pass filter setting
Reserved	rw	7	07	00000111	00000000	Reserved
CTRL7	rw	8	08	00001000	00000000	Enable Sensors
CTRL8	rw	9	09	00001001	00000000	Motion Detection Control
CTRL9	rw	10	0A	00001010	00000000	Host Commands
Host Controlled Calibration Registers (See CTRL9, Usage is Optional)						
CAL1_L	rw	11	0B	00001011	00000000	Calibration Register CAL1_L – lower 8 bits. CAL1_H – upper 8 bits.
CAL1_H	rw	12	0C	00001100	00000000	
CAL2_L	rw	13	0D	00001101	00000000	Calibration Register CAL2_L – lower 8 bits. CAL2_H – upper 8 bits.
CAL2_H	rw	14	0E	00001110	00000000	
CAL3_L	rw	15	0F	00001111	00000000	Calibration Register CAL3_L – lower 8 bits. CAL3_H – upper 8 bits.
CAL3_H	rw	16	10	00010000	00000000	
CAL4_L	rw	17	11	00010001	00000000	Calibration Register CAL4_L – lower 8 bits. CAL4_H – upper 8 bits.
CAL4_H	rw	18	12	00010010	00000000	
FIFO Registers						
FIFO_WTM_TH	rw	19	13	00010011	00000000	FIFO watermark level, in ODRs
FIFO_CTRL	rw	20	14	00010100	00000000	FIFO Setup

FIFO_SMPL_CNT	r	21	15	00010101	00000000	FIFO sample count LSBs
FIFO_STATUS	r	22	16	00010110	00000000	FIFO Status
FIFO_DATA	r	23	17	00010111	00000000	FIFO Data
Status Registers						
STATUSINT	r	45	2D	00101101	00000000	Sensor Data Availability with the Locking mechanism, CmdDone (CTRL9 protocol bit).
STATUS0	r	46	2E	00101110	00000000	Output Data Over Run and Data Availability.
STATUS1	r	47	2F	00101111	00000000	Miscellaneous Status: Any Motion, No Motion, Significant Motion, Tap.
Timestamp Register						
TIMESTAMP_ LOW	r	48	30	00110000	00000000	Sample Time Stamp TIMESTAMP_LOW – lower 8 bits. TIMESTAMP_MID – middle 8 bits. TIMESTAMP_HIGH – upper 8 bits
TIMESTAMP_MID	r	49	31	00110001	00000000	
TIMESTAMP_HIG H	r	50	32	00110010	00000000	
Data Output Registers (16 bits 2's Complement Except COD Sensor Data)						
TEMP_L	r	51	33	00110011	00000000	Temperature Output Data TEMP_L – lower 8 bits. TEMP_H – upper 8 bits
TEMP_H	r	52	34	00110100	00000000	
AX_L	r	53	35	00110101	00000000	X-axis Acceleration AX_L – lower 8 bits. AX_H – upper 8 bits
AX_H	r	54	36	00110110	00000000	
AY_L	r	55	37	00110111	00000000	Y-axis Acceleration AY_L – lower 8 bits. AY_H – upper 8 bits
AY_H	r	56	38	00111000	00000000	
AZ_L	r	57	39	00111001	00000000	Z-axis Acceleration AZ_L – lower 8 bits. AZ_H – upper 8 bits
AZ_H	r	58	3A	00111010	00000000	
GX_L	r	59	3B	00111011	00000000	X-axis Angular Rate GX_L – lower 8 bits. GX_H – upper 8 bits
GX_H	r	60	3C	00111100	00000000	
GY_L	r	61	3D	00111101	00000000	Y-axis Angular Rate GY_L – lower 8 bits. GY_H – upper 8 bits
GY_H	r	62	3E	00111110	00000000	
GZ_L	r	63	3F	00111111	00000000	Z-axis Angular Rate GZ_L – lower 8 bits. GZ_H – upper 8 bits
GZ_H	r	64	40	01000000	00000000	
COD Indication and General-Purpose Registers						
COD_STATUS	r	70	46	01000110	00000000	Calibration-On-Demand status register
dQW_L	r	73	49	01001001	00000000	General purpose register
dQW_H	r	74	4A	01001010	00000000	General purpose register
dQX_L	r	75	4B	01001011	00000000	General purpose register
dQX_H	r	76	4C	01001100	00000000	Reserved
dQY_L	r	77	4D	01001101	00000000	General purpose register
dQY_H	r	78	4E	01001110	00000000	Reserved
dQZ_L	r	79	4F	01001111	00000000	Reserved
dQZ_H	r	80	50	01010000	00000000	Reserved
dVX_L	r	81	51	01010001	00000000	General purpose register
dVX_H	r	82	52	01010010	00000000	General purpose register
dVY_L	r	83	53	01010011	00000000	General purpose register
dVY_H	r	84	54	01010100	00000000	General purpose register
dVZ_L	r	85	55	01010101	00000000	General purpose register
dVZ_H	r	86	56	01010110	00000000	General purpose register

Activity Detection Output Registers						
TAP_STATUS	r	89	59	01011001	00000000	Axis, direction, number of detected Tap
Reset Register						
RESET	w	96	60	01100000	00000000	Soft Reset Register

5 UI Sensor Configuration Settings and Output Data

5.1 Typical Sensor Mode Configuration and Output Data

In Typical sensor mode, QMI8A01 outputs raw sensor values. The sensors are configured and read using the registers described below. The accelerometer and gyroscope can be independently configured. *Table 21* summarizes these pertinent registers.

Table 21. Typical Sensor Mode Configuration and Output Data

Typical Sensor Configuration and Output Data			
Description	**Registers**	**Unit**	**Comments**
Sensor Enable, SPI 3 or 4 Wire	CTRL1		Control power states, configure SPI communications
Enable Sensor	CTRL7		Individually Enable/Disable the AttitudeEngine, Accelerometer, and Gyroscope Using sEN, aEN, and gEN bits, respectively.
Configure Accelerometer, Enable Self-Test	CTRL2		Configure Full Scale and Output Data Rate; Enable Self-Test
Configure Gyroscope, Enable Self-Test	CTRL3		Configure Full Scale and Output Data Rate; Enable Self-Test
Sensor Filters	CTRL5		Configure and Enable/Disable Low Pass Filters
Status	STATUSINT STATUS0, STATUS1		Data Availability, FIFO Ready to be Read, CTRL9 Protocol Bit
Time Stamp	TIMESTAMP[H,M,L]		Sample Time Stamp (Circular Register 0 – 0xFFFFFF)
Acceleration	A[X,Y,Z]_[H,L]	g	In Sensor Frame of Reference, Right-handed Coordinate System
Angular Rate	G[X,Y,Z]_[H,L]	dps	In Sensor Frame of Reference, Right-handed Coordinate System
Temperature	TEMP_[H,L]	° C	Temperature of the Sensor
FIFO Based Output	FIFO_DATA		1 Byte FIFO Data Outputs

5.2 Chip Information Register

Table 22. Chip Information Register Description

Register Name			
WHO_AM_I		**Register Address: 0 (0x00)**	
Bits	**Name**	**Default**	**Description**
7:0	WHO_AM_I	0x05	Device identifier 0x05 - to identify the device is a QST sensor. Read-only.
REVISION_ID		**Register Address: 1 (0x01)**	
Bits	**Name**	**Default**	**Description**
7:0	REVISION_ID	0x7C	Device Revision ID. Read-only.

5.3 Configuration Registers

This section describes the various operating modes and register configurations of the QMI8A01.

Table 23. Configuration Registers Description

<table>
<tr><th colspan="2">Register Name</th><th colspan="2"></th></tr>
<tr><td colspan="2">CTRL1</td><td colspan="2">Serial Interface and Sensor Enable. Register Address: 2 (0x02)</td></tr>
<tr><th>Bits</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>7</td><td>SIM</td><td>1’b0</td><td>0: Enables 4-wire SPI interface
1: Enables 3-wire SPI interface</td></tr>
<tr><td>6</td><td>ADDR_AI</td><td>1’b0</td><td>0: Serial interface (SPI, I^{2}C, I^{3}C) address non-increment.
1: Serial interface (SPI, I^{2}C, I^{3}C) address auto increment</td></tr>
<tr><td>5</td><td>BE</td><td>1’b1</td><td>0: Serial interface (SPI, I^{2}C, I^{3}C) read data Little-Endian
1: Serial interface (SPI, I^{2}C, I^{3}C) read data Big-Endian</td></tr>
<tr><td>4</td><td>INT2_EN</td><td>1’b0</td><td>0: INT2 pin is high-Z mode
1: INT2 pin output is enabled</td></tr>
<tr><td>3</td><td>INT1_EN</td><td>1’b0</td><td>0: INT1 pin is high-Z mode
1: INT1 pin output is enabled</td></tr>
<tr><td>2</td><td>FIFO_INT_SEL</td><td>1’b0</td><td>0: FIFO interrupt is mapped to INT2 pin
1: FIFO interrupt is mapped to INT1 pin</td></tr>
<tr><td>1</td><td>Reserved</td><td>1’b0</td><td></td></tr>
<tr><td>0</td><td>SensorDisable</td><td>1’b0</td><td>0: Enable internal high-speed oscillator
1: Disable internal high-speed oscillator. Refer to 7.1.</td></tr>
<tr><td colspan="2">CTRL2</td><td colspan="2">Accelerometer Settings: Address: 3 (0x03)</td></tr>
<tr><th>Bits</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>7</td><td>aST</td><td>1’b0</td><td>0: Disable Accelerometer Self-Test;
1: Enable Accelerometer Self-Test.</td></tr>
<tr><td>6:4</td><td>aFS<2:0></td><td>3’b0</td><td>Set Accelerometer Full-scale (1xx – N/A):
000 - Accelerometer Full-scale = ±2 g
001 - Accelerometer Full-scale = ±4 g
010 - Accelerometer Full-scale = ±8 g
011 – Accelerometer Full-scale = ±16 g</td></tr>
<tr><td>3:0</td><td>aODR<3:0>[15][16]</td><td>4’b0</td><td>Set Accelerometer Output Data Rate (ODR):
<table>
<tr><th>Setting</th><th>ODR Rate (Hz) (Accel only)</th><th>ODR Rate (Hz) (6DOF)[16]</th><th>Mode</th><th>Duty Cycle</th></tr>
<tr><td>0000</td><td>N/A</td><td>7174.4</td><td>Normal</td><td>100%</td></tr>
<tr><td>0001</td><td>N/A</td><td>3587.2</td><td>Normal</td><td>100%</td></tr>
<tr><td>0010</td><td>N/A</td><td>1793.6</td><td>Normal</td><td>100%</td></tr>
<tr><td>0011</td><td>1000</td><td>896.8</td><td>Normal</td><td>100%</td></tr>
<tr><td>0100</td><td>500</td><td>448.4</td><td>Normal</td><td>100%</td></tr>
<tr><td>0101</td><td>250</td><td>224.2</td><td>Normal</td><td>100%</td></tr>
<tr><td>0110</td><td>125</td><td>112.1</td><td>Normal</td><td>100%</td></tr>
<tr><td>0111</td><td>62.5</td><td>56.05</td><td>Normal</td><td>100%</td></tr>
<tr><td>1000</td><td>31.25</td><td>28.025</td><td>Normal</td><td>100%</td></tr>
<tr><td>1001</td><td>N/A</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1010</td><td>N/A</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1011</td><td>N/A</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1100</td><td>128</td><td>N/A</td><td>Low Power</td><td>100%</td></tr>
<tr><td>1101</td><td>21</td><td>N/A</td><td>Low Power</td><td>58%</td></tr>
<tr><td>1110</td><td>11</td><td>N/A</td><td>Low Power</td><td>31%</td></tr>
<tr><td>1111</td><td>3</td><td>N/A</td><td>Low Power</td><td>8.5%</td></tr>
</table></td></tr>
</table>

Table 24. Table 23 Configuration Register Description (Continued)

<table>
<tr><td colspan="2">Register Name</td><td colspan="2"></td></tr>
<tr><td colspan="2">CTRL3</td><td colspan="2">Gyroscope Settings: Address 4 (0x04)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7</td><td>gST</td><td>1’b0</td><td>0: Disable Gyro self-Test
1: Enable Gyro Self-Test.</td></tr>
<tr><td>6:4</td><td>gFS<2:0></td><td>3’b0</td><td>Set Gyroscope Full-scale:
000 - ±16 dps
001 - ±32 dps
010 - ±64 dps
011 - ±128 dps
100 - ±256 dps
101 - ±512 dps
110 - ±1024dps
111 - ±2048 dps</td></tr>
<tr><td>3:0</td><td>gODR<3:0> (16)</td><td>4’b0</td><td>Set Gyroscope Output Data Rate (ODR):
<table>
<tr><th>Setting</th><th>ODR Rate (Hz)</th><th>Mode</th><th>Duty Cycle</th></tr>
<tr><td>0000</td><td>7174.4</td><td>Normal</td><td>100%</td></tr>
<tr><td>0001</td><td>3587.2</td><td>Normal</td><td>100%</td></tr>
<tr><td>0010</td><td>1793.6</td><td>Normal</td><td>100%</td></tr>
<tr><td>0011</td><td>896.8</td><td>Normal</td><td>100%</td></tr>
<tr><td>0100</td><td>448.4</td><td>Normal</td><td>100%</td></tr>
<tr><td>0101</td><td>224.2</td><td>Normal</td><td>100%</td></tr>
<tr><td>0110</td><td>112.1</td><td>Normal</td><td>100%</td></tr>
<tr><td>0111</td><td>56.05</td><td>Normal</td><td>100%</td></tr>
<tr><td>1000</td><td>28.025</td><td>Normal</td><td>100%</td></tr>
<tr><td>1001</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1010</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1011</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1100</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1101</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1110</td><td>N/A</td><td></td><td></td></tr>
<tr><td>1111</td><td>N/A</td><td></td><td></td></tr>
</table></td></tr>
</table>

Note:

15. The accelerometer low power mode is only available when the gyroscope is disabled.
16. In 6DOF mode (accelerometer and gyroscope are both enabled), the ODR is derived from the nature frequency of gyroscope, refer to section 3.5 for more information.

Table 25. Table 23 Configuration Register Description (Continued)

<table>
<tr><th colspan="2">Register Name</th><th colspan="2"></th></tr>
<tr><td colspan="2">CTRL5</td><td colspan="2">Sensor Data Processing Settings. Register Address: 6 (0x06)</td></tr>
<tr><th>Bits</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>7</td><td>Reserved</td><td>1’b0</td><td></td></tr>
<tr><td>6:5</td><td>gLPF_MODE</td><td>2’b0</td><td>
<table>
<tr><td>gLPF_MODE</td><td>BW [Hz]</td></tr>
<tr><td>00</td><td>2.66% of ODR</td></tr>
<tr><td>01</td><td>3.63% of ODR</td></tr>
<tr><td>10</td><td>5.39% of ODR</td></tr>
<tr><td>11</td><td>13.37% of ODR</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>gLPF_EN</td><td>1’b0</td><td>0: Disable Gyroscope Low-Pass Filter.
1: Enable Gyroscope Low-Pass Filter with the mode given by gLPF_MODE.</td></tr>
<tr><td>3</td><td>Reserved</td><td>1’b0</td><td></td></tr>
<tr><td>2:1</td><td>aLPF_MODE</td><td>2’b0</td><td>
<table>
<tr><td>aLPF_MODE</td><td>BW [Hz]</td></tr>
<tr><td>00</td><td>2.66% of ODR</td></tr>
<tr><td>01</td><td>3.63% of ODR</td></tr>
<tr><td>10</td><td>5.39% of ODR</td></tr>
<tr><td>11</td><td>13.37% of ODR</td></tr>
</table>
</td></tr>
<tr><td>0</td><td>aLPF_EN</td><td>1’b0</td><td>0: Disable Accelerometer Low-Pass Filter.
1: Enable Accelerometer Low-Pass Filter with the mode given by aLPF_MODE.</td></tr>
</table>

Table 26. Table 23 Configuration Register Description (Continued)

Register Name			
CTRL7		**Enable Sensors and Configure Data Reads. Register Address: 8 (0x08)**	
Bits	**Name**	**Default**	**Description**
7	SyncSample	1'b0	0: Disable SyncSample mode 1: Enable SyncSample mode
6	Reserved	1'b0	
5	DRDY_DIS	1'b0	0: DRDY (Data Ready) is enabled, is driven to the INT2 pin 1: DRDY (Data Ready) is disabled, is blocked from the INT2 pin
4	gSN	1'b0	0: Gyroscope in Full Mode (Drive and Sense are enabled). 1: Gyroscope in Snooze Mode (only Drive enabled). This bit is effective only when gEN is set to 1. Refer to *7.1*.
3:2	Reserved	2'b0	
1	gEN	1'b0	0: Disable Gyroscope. 1: Enable Gyroscope.
0	aEN	1'b0	0: Disable Accelerometer. 1: Enable Accelerometer.
CTRL8		**Motion Detection Control. Register Address: 9 (0x09)**	
Bits	**Name**	**Default**	**Description**
7	CTRL9_HandShake_Type	1b'0	0: use INT1 as CTRL9 handshake 1: use STATUSINT.bit7 as CTRL9 handshake
6	ACTIVITY_INT_SEL	1b'0	0: INT2 is used for Activity Detection event interrupt 1: INT1 is used for Activity Detection event interrupt Note: this bit influences the Any/No/Sig-motion, Tap Detection interrupt
5	reserved	1b'0	
4	reserved	1b'0	
3	Sig-Motion_EN	1b'0	0: disable Significant Motion engine 1: enable Significant Motion engine
2	No-Motion_EN	1b'0	0: disable No Motion engine 1: enable No Motion engine
1	Any-Motion_EN	1b'0	0: disable Any Motion engine 1: enable Any Motion engine
0	Tap_EN	1b'0	0: disable Tap engine 1: enable Tap engine
Register Name			
CTRL9		Host Commands. Register Address: 10 (0x0A), Referred to: **CTRL 9 Functionality (Executing Pre-defined Commands)**	

5.4 FIFO Registers

Table 27. FIFO Control/Status/Data Registers

<table>
<tr><td colspan="2">Register Name</td><td colspan="2"></td></tr>
<tr><td colspan="2">FIFO_WTM_TH</td><td colspan="2">FIFO Watermark Register Address: 19 (0x13)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7:0</td><td>FIFO_WTM</td><td>8’h0</td><td>Number of ODRs (Samples) needed to trigger FIFO watermark</td></tr>
<tr><td colspan="2">FIFO_CTRL</td><td colspan="2">FIFO Control Register Address: 20 (0x14)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7</td><td>FIFO_RD_MODE</td><td>1’b0</td><td>0: FIFO is in Write mode, sensor data (if enabled) can be filled into FIFO
1: FIFO is in Read mode, FIFO data can be read via FIFO_DATA register
This bit is automatically set by using a CTRL9 command. It must be cleared again (by write 1’b0 to this bit) after the data read is complete so that filling data to the FIFO can resume. Refer to 5.10.6.3.</td></tr>
<tr><td>6:4</td><td>Reserved</td><td>3’b0</td><td></td></tr>
<tr><td>3:2</td><td>FIFO_SIZE</td><td>2’b0</td><td>
<table>
<tr><td>FIFO_SIZE[1:0]</td><td>FIFO Sample Size</td></tr>
<tr><td>00</td><td>16 samples</td></tr>
<tr><td>01</td><td>32 samples</td></tr>
<tr><td>10</td><td>64 samples</td></tr>
<tr><td>11</td><td>128 samples</td></tr>
</table>
</td></tr>
<tr><td>1:0</td><td>FIFO_MODE</td><td>2’b0</td><td>
<table>
<tr><td>FIFO_MODE[1:0]</td><td>FIFO Mode</td></tr>
<tr><td>00</td><td>Bypass (FIFO disable)</td></tr>
<tr><td>01</td><td>FIFO</td></tr>
<tr><td>10</td><td>Stream</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td colspan="2">FIFO_SMPL_CNT</td><td colspan="2">FIFO Sample Count Register Address: 21 (0x15)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7:0</td><td>FIFO_SMPL_CNT_LSB</td><td>8’b0</td><td>8 LS bits of FIFO Sample Count, in word (2bytes).</td></tr>
<tr><td colspan="2">FIFO_STATUS</td><td colspan="2">FIFO Status. Register Address 22 (0x16)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7</td><td>FIFO_FULL</td><td>1’b0</td><td>0: FIFO is not Full
1: FIFO is Full</td></tr>
<tr><td>6</td><td>FIFO_WTM</td><td>1’b0</td><td>0: FIFO Water Mark Level not hit.
1: FIFO Water Mark Level Hit</td></tr>
<tr><td>5</td><td>FIFO_OVFLOW</td><td>1’b0</td><td>0: FIFO Overflow has not happened
1: FIFO Overflow condition has happened (data dropping happened)</td></tr>
<tr><td>4</td><td>FIFO_NOT_EMPTY</td><td>1’b0</td><td>0: FIFO is Empty
1: FIFO is not Empty</td></tr>
<tr><td>3:2</td><td>Reserved</td><td>2’b0</td><td></td></tr>
<tr><td>1:0</td><td>FIFO_SMPL_CNT_MSB</td><td>2’b0</td><td>2 MS bits of FIFO Sample Count in word (2bytes).</td></tr>
<tr><td colspan="2">FIFO_DATA</td><td colspan="2">FIFO DATA Output Register Address: 23 (0x17)</td></tr>
<tr><td>Bits</td><td>Name</td><td>Default</td><td>Description</td></tr>
<tr><td>7:0</td><td>FIFO_DATA</td><td>8’b0</td><td>8 bit FIFO data output.</td></tr>
</table>

5.5 Status and Time Stamp Registers

Table 28. Status and Time Stamp Registers

Register Name			
STATUSINT		**Sensor Data Available and Lock Register Address: 45 (0x2D)**	
Bits	**Name**	**Default**	**Description**
7	Ctrl9 CmdDone	1'b0	Indicates CTRL9 Command was done, as part of CTRL9 protocol 0: Not Completed 1: Done
6:2	Reserved	5'b0	
1	Locked	1'b0	If syncSmpl (CTRL7.bit7) = 1: 0: Sensor Data is not locked. 1: Sensor Data is locked. If syncSmpl = 0, this bit shows the same value of INT1 level
0	Avail	1'b0	If syncSmpl (CTRL7.bit7) = 1: 0: Sensor Data is not available 1: Sensor Data is available for reading If syncSmpl = 0, this bit shows the same value of INT2 level
STATUS0		**Output Data Status Register Address: 46 (0x2E)**	
Bits	**Name**	**Default**	**Description**
7:2	Reserved	6'b0	
1	gDA	1'b0	Gyroscope new data available 0: No updates since last read. 1: New data available.
0	aDA	1'b0	Accelerometer new data available 0: No updates since last read. 1: New data available.
STATUS1		**Miscellaneous Status. Register Address 47 (0x2F)**	
Bits	**Name**	**Default**	**Description**
7	Significant Motion	1'b0	0: No Significant-Motion was detected 1: Significant-Motion was detected
6	No Motion	1'b0	0: No No-Motion was detected 1: No-Motion was detected
5	Any Motion	1'b0	0: No Any-Motion was detected 1: Any-Motion was detected
4	Reserved	1'b0	
3	Reserved	1'b0	
2	Reserved	1'b0	
1	TAP	1'b0	0: No Tap was detected 1: Tap was detected
0	Reserved	1'b0	
TIMESTAMP		**3 Bytes Sample Time Stamp – Output Count. Register Address: 48 - 50（0x30 - 0x32)**	
Bits	**Name**	**Default**	**Description**
7:0	TIMESTAMP_L<7:0>	0x00	

7:0	TIMESTAMP_M<15:8>	0x00	Sample time stamp. Count incremented by one for each sample (x, y, z data set) from sensor with highest ODR (circular register 0x0-0xFFFFFF).
7:0	TIMESTAMP_H<23:16>	0x00	

5.6 Sensor Data Output Registers

Table 29. Sensor Data Output Registers Description

Register Name			
TEMP_[H,L]		**Temp Sensor Output. Register Address: 0x33 – 0x34**	
Bits	**Name**	**Default**	**Description**
7:0	TEMP_L	0x00	Temperature output (°C) in two's complement. Refer to 3.9 for details.
7:0	TEMP_H	0x00	
Register Name			
A[X,Y,Z]_[H,L]		**Acceleration Output. Register Address: 0x35 – 0x3A**	
Bits	**Name**	**Default**	**Description**
7:0	AX_L<7:0>	0x00	X-axis acceleration in two's complement. AX_L – lower 8 bits. AX_H – upper 8 bits.
7:0	AX_H<15:8>	0x00	
7:0	AY_L<7:0>	0x00	Y-axis acceleration in two's complement. AY_L – lower 8 bits. AY_H – upper 8 bits.
7:0	AY_H<15:8>	0x00	
7:0	AZ_L<7:0>	0x00	Z-axis acceleration in two's complement. AZ_L – lower 8 bits. AZ_H – upper 8 bits.
7:0	AZ_H<15:8>	0x00	
Register Name			
G[X,Y.Z]_[H,L]		**Angular Rate Output. Register Address: 0x3B – 0x40**	
Bits	**Name**	**Default**	**Description**
7:0	GX_L<7:0>	0x00	X-axis angular rate in two's complement. GX_L – lower 8 bits. GX_H – upper 8 bits.
7:0	GX_H<15:8>	0x00	
7:0	GY_L<7:0>	0x00	Y-axis angular rate in two's complement. GY_L – lower 8 bits. GY_H – upper 8 bits.
7:0	GY_H<15:8>	0x00	
7:0	GZ_L<7:0>	0x00	Z-axis angular rate in two's complement. GZ_L – lower 8 bits. GZ_H – upper 8 bits.
7:0	GZ_H<15:8>	0x00	

5.7 Calibration-On-Demand (COD) Status Register

Register Name			
COD_STATUS		**Register Address: 70 (0x46)**	
Bits	**Name**	**Default**	**Description**
7	X_Limit_L_Fail	1'b0	0: COD passed for checking low sensitivity limit of X axis of gyroscope 1: COD failed for checking low sensitivity limit of X axis of gyroscope
6	X_Limit_H_Fail	1'b0	0: COD passed for checking high sensitivity limit of X axis of gyroscope 1: COD failed for checking high sensitivity limit of X axis of gyroscope
5	Y_Limit_L_Fail	1'b0	0: COD passed for checking low sensitivity limit of Y axis of gyroscope 1: COD failed for checking low sensitivity limit of Y axis of gyroscope
4	Y_Limit_H_Fail	1'b0	0: COD passed for checking high sensitivity limit of Y axis of gyroscope 1: COD failed for checking high sensitivity limit of Y axis of gyroscope
3	Accel_Check	1'b0	0: Accelerometer checked pass (no significant vibration happened during COD) 1: Accelerometer checked failed (significant vibration happened during COD)
2	Startup_Failed	1'b0	0: Gyroscope startup succeeded 1: Gyroscope startup failure happened when COD was called
1	Gyro_Enabled	1'b0	0: COD was called when gyroscope was not enabled 1: COD was called while gyroscope was enabled, COD return failure
0	COD_Failed	1'b0	0: COD succeeded, new gain parameters will be applied to GX & GY data 1: COD failed; no COD correction applied

Note the value of this register is only valid after the COD command. Refer to *12 Calibration-On-Demand (COD).*

5.8 Activity Detection Output Registers

Table 30. Activity Detection Output Registers

Register Name			
TAP_STATUS		**Register Address: 89 (0x59)**	
Bits	**Name**	**Default**	**Description**
7	TAP_POLARITY	1'b0	0: Tap was detected on the positive direction of the Tap axis 1: Tap was detected on the negative direction of the Tap axis
6	Reserved	1'b0	
5:4	TAP_AXIS	2'b0	0: No Tap was detected 1: Tap was detected on X axis 2: Tap was detected on Y axis 3: Tap was detected on Z axis
3:2	Reserved	2'b0	
1:0	TAP_NUM	2'b0	0: No Tap was detected 1: Single-Tap was detected 2: Double-Tap was detected 3: NA

5.9 Reset Register

Table 31. Reset Register Description

Register Name			
RESET		**Register Address: 96 (0x60)**	
Bits	**Name**	**Default**	**Description**
7:0	RESET	0x00	Soft Reset Register - Write 0xB0 to this register from any modes, will trigger the sensor reset process immediately. The register 0x4D will equals to 0x80 if there is a successful reset (Power-on Reset or Soft Reset) process. Refer to *7.4 Chip Reset Process*.

5.10 CTRL 9 Functionality (Executing Pre-defined Commands)

5.10.1 CTRL9 Protocol Description

The protocol for executing predefined commands from an external host processor on the QMI8A01 is facilitated by using the CTRL9 register. The register is available to the host via the UI SPI/I^2C/I^3C bus.

It operates by the host writing a pre-defined value (Command, refer to *5.10.2* CTRL9 Command List) to the CTRL9 register. The firmware of QMI8A01 evaluates this command and if a match is found it executes the corresponding pre-defined function.

Once the function has been executed, the QMI8A01 signals the completion by setting STATUSINT.bit7 to 1and raising INT1 interrupt if CTRL1.bit3 = 1 & CTRL8.bit7 == 0. The host must acknowledge this by writing CTRL_CMD_ACK (0x00) to CTRL9 register. After receiving the CTRL_CMD_ACK command, the QMI8A01 clears the STATUSINT.bit7 to 0 and pulls down the INT1 interrupt if CTRL1.bit3 = 1 & CTRL8.bit7 == 0.

This command presentation from the host to the QMI8A01 and the subsequent execution and handshake between the host and the QMI8A01 will be referred to as the "*CTRL9 Protocol*".

There are three types of interactions between the host and QMI8A01 that follow the CTRL9 Protocol.

WCtrl9: The host needs to supply data to QMI8A01 prior to the Ctrl9 protocol. (**Write – Ctrl9 Protocol**)

Ctrl9R: The host gets data from QMI8A01 following the Ctrl9 protocol. (**Ctrl9 protocol – Read**)

Ctrl9: No data transaction is required prior to or following the Ctrl9 protocol. (**Ctrl9**).

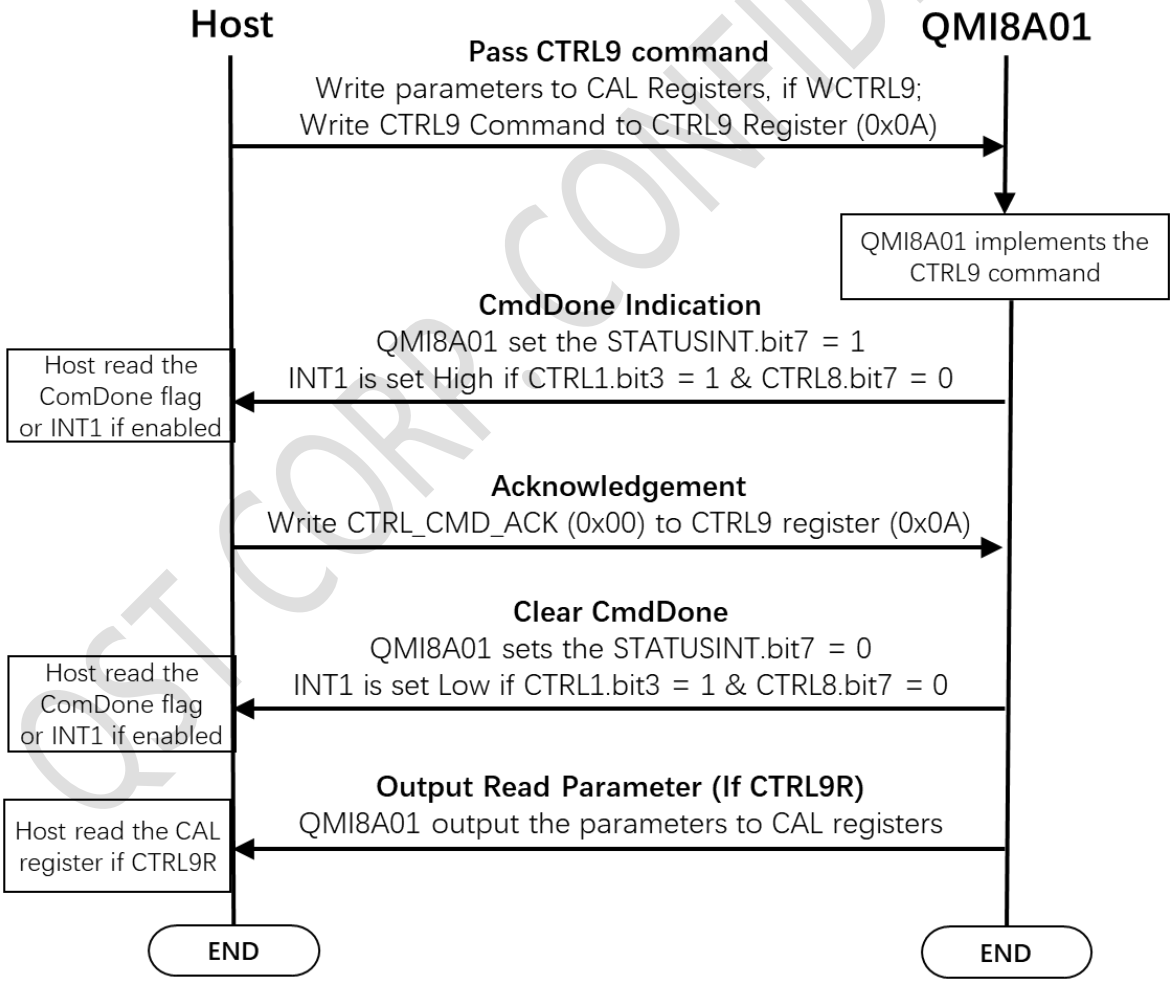


Figure 13. CTRL9 Protocol Flow Chart

5.10.2 **CTRL9 Command List**

The predefined CTRL9 commands are listed in Table 32 List of CTRL9 Command.

Table 32. List of CTRL9 Commands

Command Name	CTRL9 Command Value	Protocol Type	Description
CTRL_CMD_ACK	0x00	Ctrl9	Acknowledgement. Host acknowledges to QMI8A01, to end the protocol.
CTRL_CMD_RST_FIFO	0x04	Ctrl9	Reset FIFO from Host
CTRL_CMD_REQ_FIFO	0x05	Ctrl9R	Get FIFO data from Device
CTRL_CMD_ACCEL_HOST_DELTA_OFFSET	0x09	WCtrl9	Change accelerometer offset
CTRL_CMD_GYRO_HOST_DELTA_OFFSET	0x0A	WCtrl9	Change gyroscope offset
CTRL_CMD_CONFIGURE_TAP	0x0C	WCtrl9	Configure Tap detection
CTRL_CMD_CONFIGURE_MOTION	0x0E	WCtrl9	Configure Any Motion / No Motion / Significant Motion detection
CTRL_CMD_COPY_USID	0x10	Ctrl9R	Copy USID and FW Version to UI registers
CTRL_CMD_SET_RPU	0x11	WCtrl9	Configures IO pull-ups
CTRL_CMD_AHB_CLOCK_GATING	0x12	WCtrl9	Internal AHB clock gating switch
CTRL_CMD_ON_DEMAND_CALIBRATION	0xA2	WCtrl9	On-Demand Calibration on gyroscope
CTRL_CMD_APPLY_GYRO_GAINS	0xAA	WCtrl9	Restore the saved Gyroscope gains

5.10.3 **CAL Registers**

The set of CAL registers can be used for the parameter transferring, if WCTL9 or CTRL9R commands are implemented. Refer to *Table 32* and *5.10.6* for details.

Table 33. CAL Register Addresses

Register Name	Register Address	
	Dec	Hex
CAL1_L	11	0x0B
CAL1_H	12	0x0C
CAL2_L	13	0x0D
CAL2_H	14	0x0E
CAL3_L	15	0x0F
CAL3_H	16	0x10
CAL4_L	17	0x11
CAL4_H	18	0x12

5.10.4 WCtrl9 (Write – CTRL9 Protocol)

1. The host needs to provide the required data for this command to QMI8A01. The host typically does this by placing the data in a set of registers called the CAL registers. A maximum of eight CAL registers are used. Refer to *Table 33*.
2. Write Ctrl9 register 0x0A with the appropriate Command value, refer to *Table 32*.
3. The Device will set STATUSINT.bit7 to 1 and raise INT1(if CTRL1.bit3 = 1 & CTRL8.bit7 == 0), once it has executed the appropriate function based on the command value.
4. The host must acknowledge this by writing CTRL_CMD_ACK (0x00) to CTRL9 register, STATUSINT.bit7 (CmdDone) will be reset to 0 on receiving the CTRL_CMD_ACK command. And INT1 is pulled low upon the register read if CTRL1.bit3 = 1 & CTRL8.bit7 == 0.
5. If any data is expected from the device, it will be available at this time. The location of the data is specified separately for each of the Commands.

Refer to *5.10.6 for details.*

5.10.5 Ctrl9R (CTRL9 Protocol - Read)

1. Write Ctrl9 register 0x0A with the appropriate Command value.
2. The Device will set STATUSINT.bit7 to 1 and raise INT1 (if CTRL1.bit3 = 1 & CTRL8.bit7 == 0), once it has executed the appropriate function based on the command value.
3. The host must acknowledge this by writing CTRL_CMD_ACK (0x00) to CTRL9 register, STATUSINT.bit7 (CmdDone) will be reset to 0 on receiving the CTRL_CMD_ACK command. INT1 is pulled low upon the register read if CTRL1.bit3 = 1 & CTRL8.bit7 == 0.
4. Data is available from the device on the CAL registers. The location of the data is specified separately for each of the Commands.

Refer to *5.10.6 for details.*

5.10.6 CTRL9 Commands in Details

5.10.6.1 CTRL_CMD_ACK

Host acknowledges QMI8A01 when received the CmdDone information, to end the CTRL9 protocol.

5.10.6.2 CTRL_CMD_RST_FIFO

This CTRL9 command of writing 0x04 to the Ctrl9 register 0x0a allows the host to instruct the device to reset the FIFO. The FIFO data, sample count and flags will be cleared and reset to default status.

5.10.6.3 CTRL_CMD_REQ_FIFO

This CTRL9 Command is issued when the host wants to get data from the FIFO, by writing 0x05 through the CTRL9 process.

After successfully finishing the CTRL9 process, the FIFO read mode will be enabled, the device will direct the FIFO data to the FIFO_DATA register(0x17) until the FIFO is empty. After reading the FIFO data, the host must set FIFO_CTRL.FIFO_rd_mode to 0 by writing the FIFO_CTRL register, which will cause the FIFO_STATUS.FIFO_WTM/FIFO_FULL to be cleared and/or the INT pin (if enabled) be de-asserted. Refer to *5.10.4 5.10.5* for CTRL9 operation, and refer to *8.8* for details.

5.10.6.4 CTRL_CMD_ACCEL_HOST_DELTA_OFFSET

This CTRL9 Command is issued when the host wants to manually change the accelerometer offset. Each delta offset value should contain 16 bits and the format is signed 4.12 (12 fraction bits, unit is 1 / 2^12). The user must write the offset to the following registers:

Accel_Delta_X : {CAL1_H, CAL1_L}
Accel_Delta_Y : {CAL2_H, CAL2_L}
Accel_Delta_Z : {CAL3_H, CAL3_L}

Next, the Command is issued by writing 0x09 to CTRL9 register 0x0A. Note, this offset change is lost when the sensor is power cycled, or the system is reset.

5.10.6.5 CTRL_CMD_GYRO_HOST_DELTA_OFFSET

This CTRL9 Command is issued when the host wants to manually change the gyroscope offset. Each delta offset value should contain 16 bits and the format is signed 11.5 (5 fraction bits, unit is 1 / 2^5). The user must write the offset to the following registers:

Gyro_Delta_X : {CAL1_H, CAL1_L}
Gyro_Delta_Y : {CAL2_H, CAL2_L}
Gyro_Delta_Z : {CAL3_H, CAL3_L}

Next, the Command is issued by writing 0x0A to CTRL9 register 0x0A. Note, this offset change is lost when the sensor is power cycled, or the system is reset.

5.10.6.6 CTRL_CMD_CONFIGURE_TAP

This CTRL9 command is issued to configure the parameters of Tap detection. Refer to *10.3 Configure Tap for details.*

5.10.6.7 CTRL_CMD_CONFIGURE_MOTION

This CTRL9 command is issued to configure the parameters of Motion Detection. Refer to *9.4 Config Motion Detection.*

5.10.6.8 CTRL_CMD_COPY_USID

The USID is the unique ID of every single QMI8A01 part.

This CTRL9 Command copies the following data into UI registers. It is initiated by the host writing 0x10 to CTRL9. After issuing the command, the data will be available for the host to read from the registers shown below:

FW_Version byte 0 → dQW_L
FW_Version byte 1 → dQW_H
FW_Version byte 2 → dQX_L
USID_Byte_0 → dVX_L
USID_Byte_1 → dVX_H
USID_Byte_2 → dVY_L
USID_Byte_3 → dVY_H
USID_Byte_4 → dVZ_L
USID_Byte_5 → dVZ_H

Note that after the successful Power-On Reset or Soft-Reset, the FW_Version and USID will be copied automatically to the registers once for the host to read them. Those registers can be changed after enabling the sensors, so afterwards, the CTRL_CMD_COPY_USID command should be implemented to copy the FW_Version and USID to the registers before reading.

5.10.6.9 CTRL_CMD_SET_RPU

This CTRL9 Command is issued when the host configures the IO pull-up resistors. Each bit controls a combination of resistors as shown in *Table 34*:

Table 34. Pull-Up Resistor Table

Bit	Signal Name	Pins	Activity
0	aux_rpu_dis	SDx, SCx, RESV (Pin 10)	0: enable pull-up resistors(default) 1: disable pull-up resistors
1	icm_rpu_dis	SDx	0: enable pull-up resistor(default) 1: disable pull-up resistor
2	cs_rpu_dis	CS	0: enable pull-up resistor(default) 1: disable pull-up resistor
3	i2c_rpu_dis	SCL, SDA	0: enable pull-up resistors(default) 1: disable pull-up resistors
4:7	Reserved	NA	

The host writes the appropriate CAL1_L bit by issuing a WCtrl9 command with 0x11.

By default, all the pull-up resistors are enabled. Writing 1 to the bit will disable the pullup resistors accordingly, while writing 0 will enable the pull-up resistors.

5.10.6.10 CTRL_CMD_AHB_CLOCK_GATING

When locking Mechanism is set (CTRL7.bit7 == 1(syncSmpl)), the CTRL_CMD_AHB_CLOCK_GATING should be disabled to guarantee the locking mechanism of data reading, to prevent the possible misalignment. Refer to *12 Calibration-On-Demand (COD)* for details.

5.10.6.11 CTRL_CMD_ON_DEMAND_CALIBRATION

This CTRL9 Command enables the host to recalibrate the gyro sensitivity from time to time. Refer to *12 Calibration-On-Demand (COD).*

5.10.6.1 CTRL_CMD_ APPLY_GYRO_GAINS

This CTRL9 Command enables the host to restore the saved gyro gains to QMI8A01, to avoid run the COD again. This is not recommended when there are significant environmental changes, like significant PCB stress change. Refer to *12.4 Save and Restore the New Gain Parameters*.

6 Interrupts

The QMI8A01 has two Interrupt lines, INT1 and INT2.

Both INT1 and INT2 can be configured as High-Z mode or Push-Pull mode by configuring the CTRL1.bit3(INT1) or CTRL1.bit4(INT2). If CTRL1.bit3 (CTRL1.bit4) is set to 0, INT1(INT2) will be set in High-Z mode accordingly. While if CTRL1.bit3 (CTRL1.bit4) is set to 1, INT1(INT2) will be set in Push-Pull mode accordingly. By default, INT1 and INT2 are in High-Z mode.

There are two modes of the interrupt map, as described below. The host can configure multiple internal signal/interrupt sources to INT pins (INT1 and/or INT2). If driven to one INT pin, the multiple sources act in LOGIC-OR.

6.1 SyncSample mode

The SyncSample mode supports locking the values during the reading process. Refer to *11 Locking Mechanism.* For details to the sensor data registers.

.

Set CTRL7.bit7(SyncSample) == 1 will enable the SyncSample mode.

As illustrated in *Figure 14*. In SyncSample mode, the CTRL9 handshake signal will be routed to INT1. Check *5.10* for details.

The motion event interrupts (Any Motion, No Motion, Significant Motion, Tap) will be routed to INT1 if enabled.

FIFO function is not supported in this mode, and DRDY signal will be routed to INT2.

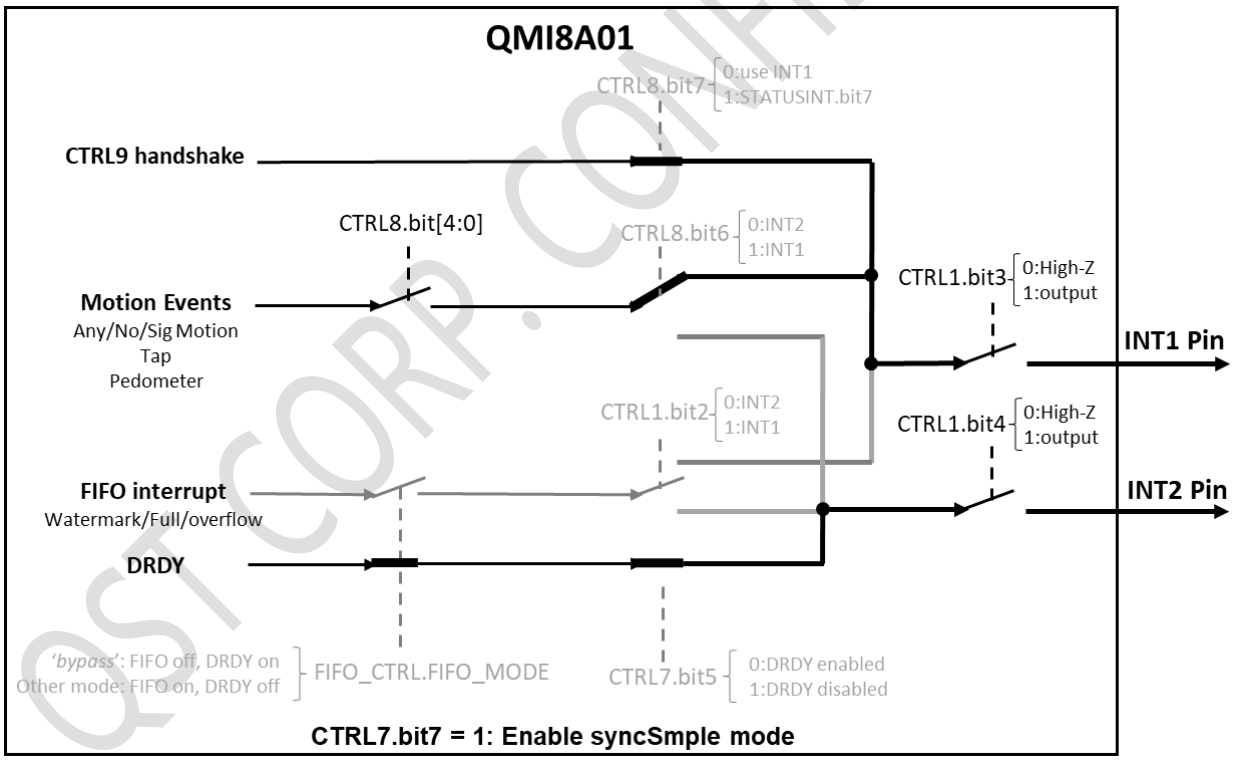


Figure 14. Interrupt Map of SyncSample Mode

6.2 Non-SyncSample mode

This mode supports FIFO function and free interrupts configuration, as illustrated in *Figure 15*.

If CTRL7.bit7(SyncSample) == 0, then bit 1 of STATUSINT register will have the same value as INT1 and bit 0 of STATUSINT register will have the same value as INT2.

In Non-SyncSample mode, there are two approaches of CTRL9 handshake. Host can check the INT1 pin high level for the handshark if set CTRL8.bit7 = 0; or poll the STATUSINT.bit7 for handshake if set CTRL8.bit7 = 1.

In Non-SyncSample mode, the motion event interrupt(s) can be configured to INT1 by setting CTRL8.bit6 = 1, or to INT2 by setting CTRL8.bit6 = 0. Note that the motion event engines can be enabled by CTRL8.bit[4:0], refer to *Table 23* for details.

In Non-SycnSample mode, the sensor data can be output through data register or FIFO. Configure the FIFO_CTRL.FIFO_MODE = 'bypass' mode, will enable the DRDY function and disable FIFO functionality; configure the FIFO_CTRL.FIFO_MODE = other mode, will enable the FIFO functionality and disable the DRDY function.

If FIFO mode is enabled, the FIFO interrupt can be configured to INT1 pin if CTRL1.bit2 is set to 1, or INT2 pin if CTRL1.bit2 is set to 0. Refer to *8 FIFO Description* for more details of FIFO interrupt behavior.

If DRDY mode is enabled, the DRDY signal will be routed to INT2 if the CTRL7.bit5(DRDY_DIS) is set to 0, or be blocked from INT2 pin if CTRL7.bit5(DRDY_DIS) is set to 1.

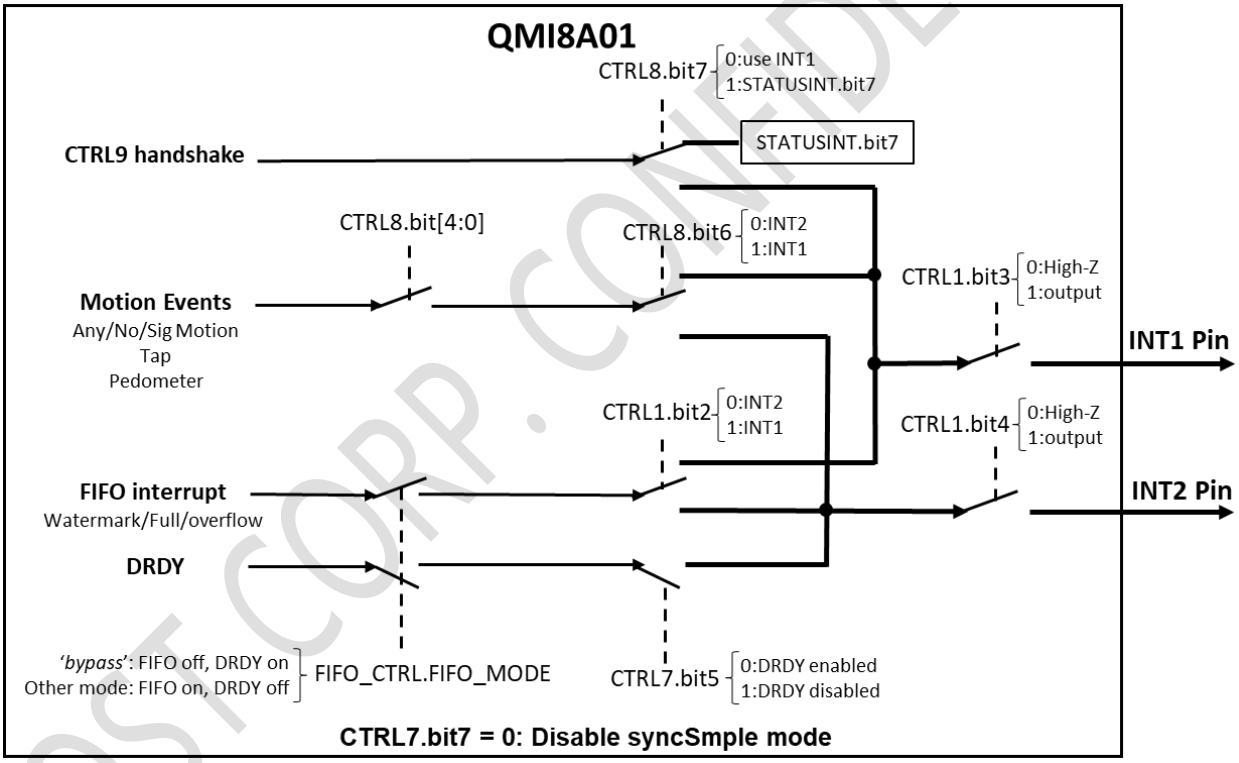


Figure 15. Interrupt Map of Non-SyncSample Mode

6.3 DRDY (Data Ready)

If DRDY mode is enabled (accelerometer and/or gyroscope are/is enabled, and FIFO mode is set to bypass) and DRDY_DIS == 0, DRDY (Data Ready) signal will be driven to the INT2 in edge-trigger mode, means the DRDY signal can be seen on INT2. The Sensor Data Output Registers (refer to *5.6*) are updated at the Output Data Rate (ODR), and DRDY signal is pulsed at the ODR frequency. The new data is updated to the data registers during the low level before the rising edge, and a rising edge on DRDY indicates that data is available for the host to read. DRDY is cleared automatically after a short duration. The DRDY pulse width is dependent on the sensor ODR set by CTRL2 and/or CTRL3 registers and enabled sensor(s).

In Non-SyncSample mode, it is the responsibility of the host to detect the rising edge and to read the data out during the high level of the INT2 pulse. Otherwise, there is the possibility that the updating of the new data happens during the host reading process and causes data mismatch.

In SyncSample mode, it is possible to lock the data in the data registers and read them in an unlimited delay, by following a process. Refer to *6.1 SyncSample mode.*

7 Operating Modes

The QMI8A01 offers a large number of operating modes that may be used to operate the device in a power efficient manner. These modes are described in *Table 35* and are shown in *Figure 16*; they may be configured using the control (CTRL) registers.

7.1 Operating Modes Descriptions

Table 35. Operating Modes

Mode	Description	Suggested Configuration
Power-On Default	All sensors are off, clock is turned on. Note this mode is the default state upon initial power up or after a reset.	CTRL1 sensorDisable = 0 CTRL7 aEN = 0, gEN = 0, CTRL2 aODR =000
Low Power	Same as Power-On Default mode, except in this mode the 250 kHz clock is turned on instead of the high-speed clock. To enter this mode requires host interaction to set CTRL2 aODR=11xx.	CTRL1 sensorDisable =0 CTRL7 aEN = 0, gEN = 0, CTRL2 aODR =11xx
Power-Down	All QMI8A01 functional blocks are switched off to minimize power consumption. Digital interfaces remain on allowing communication with the device. All configuration register values are preserved, and output data register values are maintained. The host must initiate this mode by setting sensorDisable=1.	CTRL1 sensorDisable =1 CTRL7 aEN = 0, gEN = 0
Normal Accel Only	Device configured as an accelerometer only.	CTRL7 aEN =1, gEN =0, CTRL2 aODR !=11xx
Low Power Accel Only	Device configured in low power accelerometer mode.	CTRL7 aEN =1, gEN =0, CTRL2 aODR =11xx
Snooze Gyro	Device configured as gyroscope drive only, the gyroscope MEMS will keep running at resonance frequency. Since the Sensing part is not enabled, there is no data from the gyroscope in this mode. This mode enables relative lower current consumption than Gyro-Only mode and can quickly generate data from clearing of gSN.	CTRL7 gSN=1, aEN =0, gEN =1
Gyro Only	Device configured as a gyroscope only.	CTRL7 gSN=0, aEN =0, gEN =1
Accel + Gyro (IMU)	Device configured as an Inertial Measurement Unit, i.e. an accelerometer and gyroscope combination sensors.	CTRL7 gSN=0, aEN =1, gEN =1 CTRL2 aODR != 11xx
Accel + Snooze Gyro	Accelerometer and gyroscope snooze are enabled. Only accelerometer data is available.	CTRL7 gSN=1, aEN =1, gEN =1 CTRL2 aODR != 11xx
Software Reset	Software Reset asserted	
No Power	VDDIO and VDD low	

7.2 General Mode Transitioning

Upon exiting the No Power state (i.e., on first applying power to the part) or exiting a Software Reset state, the part will enter the Power-On Default state. From there, the sensor can be configured in the various modes described in Table 35 and as shown in Figure 16. The figure illustrates the timing associated with various mode transitions, and values for these times are given in the section below and in Table 8 and Table 9.

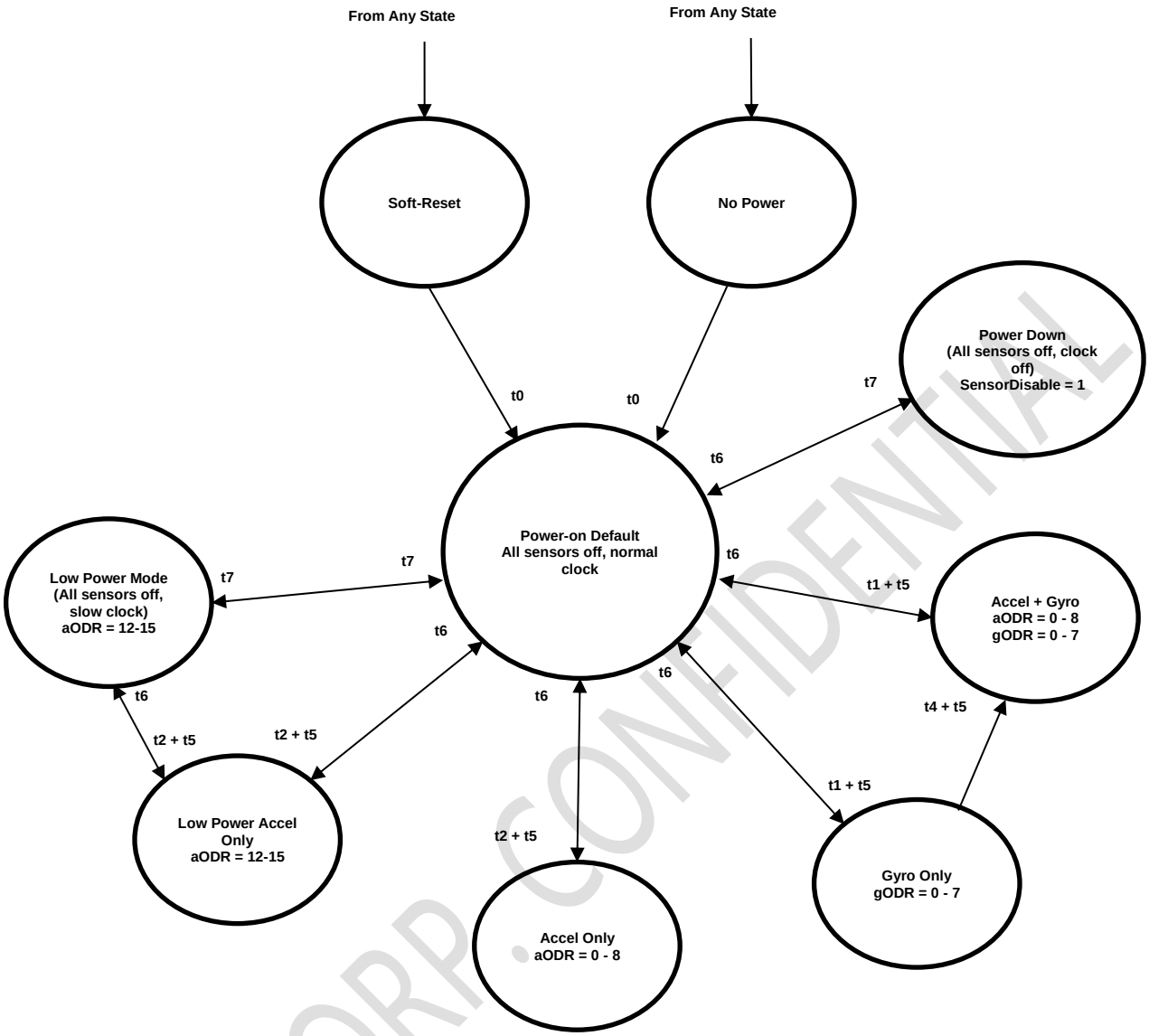


Figure 16. Operating Mode Transition Diagram

7.3 Transition Time

t0 is the *System Turn On Time* and is the time to enter the Power-On Default state from Software Reset, No Power, or Power down. Time t0 is maximum 15ms. This time only needs to be done once, upon transitioning from either a No Power or Power Down state, or whenever a reset is issued, which should not be done unless the intent is to have the device to go through its entire boot sequence (see the specification System Turn On Time in both Table 8 and Table 9).

The times t1, t2 and t4, are defined as the time it takes from issuing the enabling-sensor command (when the corresponding sensor is off) to DRDY(INT2) going high (data being present) or STATUSINT.bit0 is set to 1.

The time t5 is the time it takes to have the internal filtered data to output the correct representation of the inertial state. t5 is variable and is associated with the user selected Output Data Rate (ODR). We have defined minimum t5 = (3/ODR) to generally represent that time.

t6 is the time it takes to go from a sensor powered state to a state where the sensors are off. This time depends on the Output Data Rate (ODR) and ranges from 1/ODR to 2/ODR.

t7 is the transition time between various states where the sensors are off.

The Gyro Turn on Time (see Table 9) is comprised of t1 (the gyroscope wakeup time) and t5 (the part's filter settling time). t1 is typically 150 ms and t5 is defined as 3/ODR, where ODR is the output data rate in Hertz.

The Accel Turn on Time (see Table 8) is comprised of t2 (the accelerometer wakeup time) and t5 (the part's filter settling time). t2 is typically 3 ms, and t5 is defined as 3/ODR, where ODR is the output data rate in Hertz.

The t7 transition is dependent on data transfer rates and is for I2C at 400 kHz is <100 µs for SPI at 11 Mbps is around 40 µs.

7.4 Chip Reset Process

There are two approaches that can trigger the QMI8A01 to run the Reset process, Power-On Reset and Software Reset.

Power-On reset is initialized by driving the VDD & VDDIO lines to valid working range from the Power Off status (VDD = 0V, VDDIO = 0V). Refer to *3.2* for details. The Power-On Reset process starts from the release of POR, refer to *3.3* for details.

The Software Reset (Soft Reset) is generated by writing the 0xB0 to RESET register (0x60). Refer to *5.9*.

After the Reset is triggered (Power-On Reset and Software Reset), the QMI8A01 will run the reset process. The UI registers, internal RAM, FIFO will be set to default values, Analog and digital circuitries will be disabled, refer to *3.3* for details.

It takes maximum 15ms for the Reset process to be finished. Refer to *Table 8* and *Table 9* for details. Note that the VDDIO & VDD power lines are expected to be stable (no sharp pulse) once settled to the Final Value(s), before the System Turn On Time. Otherwise, the QMI8A01 initialization may be interfered by the unwanted power pulse, and result in failure of accelerometer or gyroscope startup, refer to *3.3* for details.

The register 0x4D will present 0x80 if there is a successful Reset (Power-on Reset or Soft Reset) process. Refer to *5.9*. Note that the content of register 0x4D could be overwritten after later operations, like enabling the sensor(s) (in CTRL7.bit [1:0]) or implementing the CTRL9 command, so host is expected to read the register immediately after POR or Software Reset to check the Reset result.

8 FIFO Description

8.1 FIFO Structure

The QMI8A01 contains a programmable 1536-byte FIFO. The FIFO's operating mode and configuration are set via the FIFO_CTRL register, refer to *Table 27*. FIFO data may consist of gyroscope and accelerometer data and is accessible via the serial interfaces (SPI/I2C/I3C), in burst reads. Depending on how many sensors are enabled, the host is expected to read increments of 6, 12 bytes, corresponding to one and two sensors active at the same time. This feature helps reduce overall system power consumption by enabling the host processor to read and process the sensor data in bursts and then enter a low-power mode. The interrupt function may be used to alert the host when the FIFO watermark level is reached.

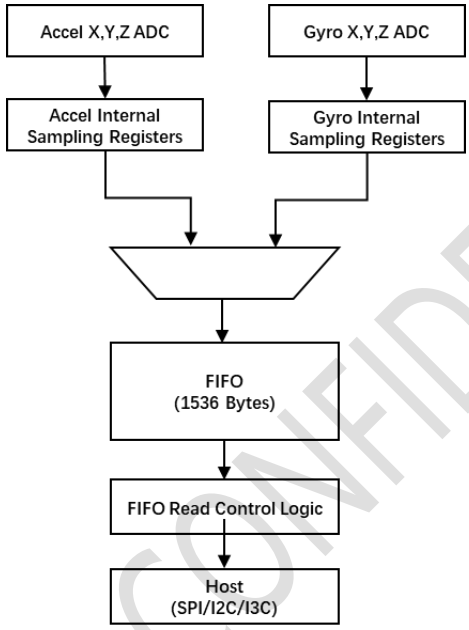


Figure 17. FIFO Data Flow

8.2 FIFO Size

There are four levels of FIFO size: 16 samples, 32 samples, 64 samples, 128 samples. The sample stands for 6 bytes of accelerometer data or 6 bytes of gyroscope data if one of them is enabled, or 6 bytes of accelerometer and 6 bytes of gyroscope data (total 12 bytes) if both are enabled.

The FIFO size is configured using the FIFO_CTRL.FIFO_SIZE[1:0]: 16 samples(0), 32 samples(1), 64 samples(2), 128 samples(3), refer to *Table 27.*

When the FIFO is enabled for two sensors (Accelerometer and Gyroscope), the sensors must be set at the same Output Data Rate (ODR), refer to *Table 23* for CTRL2 and CTRL3 registers.

8.3 Configure FIFO Mode

FIFO has multiple operating modes: Bypass, FIFO, Stream. The operating modes are set by the FIFO_CTRL.FIFO_MDOE [1:0] bits in the register, refer to *Table 27.*

Configure the FIFO_CTRL.FIFO_MODE to *'FIFO'*(1) or *'Stream'*(2) mode will enable the FIFO functionality. Configure the FIFO_MODE to *'Bypass'* (0) mode, will disable the FIFO functionality.

Once FIFO is enabled and the Accelerometer and/or Gyroscope is(are) enabled, the corresponding data will be filled into FIFO.

In '*FIFO*' mode, once FIFO is full, the data filling will stop, and new data will be discarded until the host reads out the FIFO data and releases the space for new data to be written to.

In '*Stream'* mode, once FIFO is full, the data filling will continue and the oldest data will be discarded, until host reads out the FIFO data and releases the space for new data to be written to.

8.4 FIFO Sample Count

The FIFO Sample Count indicates the filled content level of FIFO, which means the quantity of sensor data that was written into FIFO, and to be read out.

The FIFO Sample Count is a 10-bit value and stored in FIFO_SMPL_CNT and FIFO_STATUS[1:0], unit is word(two bytes). Host need to read the two registers, to calculate the number of bytes of FIFO content data by below formula:

FIFO_Sample_Count (in byte) = 2 * (fifo_smpl_cnt_msb[1:0] * 256 + fifo_smpl_cnt_lsb[7:0])

8.5 FIFO Watermark Interrupt

The FIFO_WTM register(0x13) indicates the expected level of FIFO data that the host wants to get the FIFO Watermark interrupt. The unit is sample, which means 6 bytes if one of accelerometer and gyroscope is enabled, and 12 bytes if both are enabled.

Note that the configured FIFO watermark should not be higher than the FIFO size configured by FIFO_CTRL.FIFO_SIZE.

If the FIFO_WTM is written with non-zero value, the FIFO watermark function is enabled. The internal FIFO watermark interrupt will be triggered if the FIFO content level reaches or is higher than the configured FIFO watermark level. The FIFO_STATUS.FIFO_WTM flag will be set to 1 and will be cleared to 0 if FIFO content drops lower than the FIFO watermark level.

The internal FIFO watermark interrupt signal can drive the INT1 pin if CTRL1.bit2 = 1, or INT2 pin if CTRL1.bit2 = 0. Note that the CTRL1.bit3(INT1) and CTRL1.bit4(INT2) controls the high-Z or push-pull mode of the INT pins, refer to *6 Interrupts* for details.

Once the corresponding INT pin is configured to the push-pull mode, the FIFO watermark interrupt can be seen on the corresponding INT pin. It will keep high level as long as the FIFO filled level is equal to or higher than the watermark, will drop to low level as long as the FIFO filled level is lower than the configured FIFO watermark after reading out by host and FIFO_RD_MODE is cleared.

8.6 FIFO Full

The FIFO_STATUS.FIFO_FULL flag is set if the FIFO filled level (samples) equals to the FIFO size configured by FIFO_CTRL.FIFO_SIZE and is cleared if FIFO filled level is lower than the FIFO size.

8.7 FIFO Read Mode

To read out the FIFO data, host need to set the FIFO into Read Mode, by issuing CTRL_CMD_REQ_FIFO command through CTRL9 process, refer to *5.10 CTRL 9 Functionality (Executing Pre-defined Commands)*.

Once FIFO Read Mode is enabled, FIFO data will be directed to the FIFO_DATA register, then the host can possibly read data from the FIFO_DATA register.

Note that once FIFO Read Mode is enabled, the new data won't be filled into FIFO, and will be discarded. The host is expected to read out the FIFO data and disable the FIFO Read Mode before the new data come. Otherwise, data dropping will happen. The time for the new data to come is defined by the sensor ODR, which is 1 / ODR.

8.8 Read FIFO Data

The FIFO data is read through the I^2C/I^3C/SPI interface by reading the FIFO_DATA register. Any time the FIFO_DATA register is read, data is erased from the FIFO memory, corresponds to First-In-First-Out concept.

Host is expected to read out the FIFO data, following below sequence:

1. Got FIFO watermark interrupt by INT pin or polling the FIFO_STATUS register (FIFO_WTM and/or FIFO_FULL).
2. Read the FIFO_SMPL_CNT and FIFO_STATUS registers, to calculate the level of FIFO content data, refer to 8.4 FIFO Sample Count.
3. Send CTRL_CMD_REQ_FIFO (0x05) by CTRL9 command, to enable FIFO read mode. Refer to *CTRL_CMD_REQ_FIFO* for details.
4. Read from the FIFO_DATA register per FIFO_Sample_Count.
5. Disable the FIFO Read Mode by setting FIFO_CTRL.FIFO_rd_mode to 0. New data will be filled into FIFO afterwards.

8.9 FIFO Data Pattern

Note that when only the accelerometer or gyroscope is enabled, the sensor data format at the host interface is:

AX_L[0] →AX_H[0] →AY_L[0] →AY_H[0] →AZ_L[0] →AZ_H[0] →AX_L[1] →…

or

GX_L[0] →GX_H[0] →GY_L[0] →GY_H[0] →GZ_L[0] →GZ_H[0] →GX_L[1] →…

When both accelerometer and gyroscope are enabled, the sensor data format is:

AX_L[0] →AX_H[0] →AY_L[0] →AY_H[0] →AZ_L[0] →AZ_H[0] →

GX_L[0] →GX_H[0] →GY_L[0] →GY_H[0] →GZ_L[0] →GZ_H[0] →

AX_L[1] →AX_H[1] →…

8.10 Reset FIFO

The FIFO content can be cleared/emptied by inserting CTRL_CMD_RST_FIFO through CTRL9 process, refer to 5.10 CTRL 9 Functionality (Executing Pre-defined Commands) for details.

9 Motion Detection

The calculation of the Motion Detection (No-, Any-, or Significant-Motion) is based on the accelerometer ODR defined by *CTRL2.aODR*, refer to *Table 23* for details.

The Motion Detection can only work in Non-SyncSample mode, refer to *6.2 Non-SyncSample mode* for details.

No-, Any-, or Significant-Motion interrupts can be issued to host. No-Motion interrupt indicates that the device is in idle/quiet status, host can run into sleep or low-power mode. Any-Motion interrupt indicates that the device is in movement, host can be awakened from sleep/low-power mode. Significant-Motion indicates host that the device is in significant and continuous movement.

9.1 Motion Detection Principle

Any-/No-/Significant-Motion detection is calculating on the slope of the acceleration of enabled axis:

Slope(n) = Acc(n) – Acc(n-1), for enabled axis/axes of accelerometer

9.1.1 Any-Motion Detection Principle

Any-Motion Detection is detecting the absolute slope that is higher than defined threshold (*AnyMotionXThr, AnyMotionYThr, AnyMotionZThr*), and last consecutively for *AnyMotionWindow* or more samples. As shown in *Figure 18*. If the conditions (threshold and duration) are fulfilled, Any-Motion event flag is set, and can be configured to drive the interrupt pin (INT1 or INT2), synced to the DRDY. When the slope falls within the range of *(-Threshold, +Threshold),* the Any-Motion conditions are not fulfilled, and the corresponding interrupt is reset.

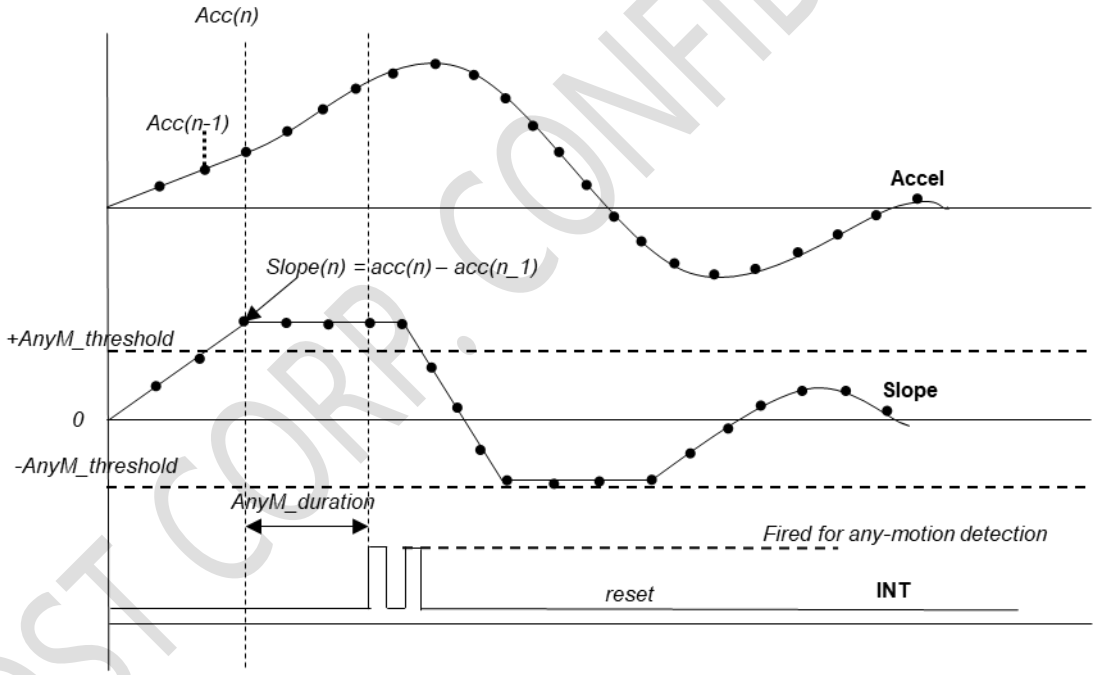


Figure 18. Any-Motion Detection Interrupt

9.1.2 No-Motion Detection Principle

No-Motion Detection is detecting the absolute of slope that is lower than defined threshold (*NoMotionXThr, NoMotionYThr, NoMotionZThr*), and last consecutively for *NoMotionWindow* or more samples. As shown in *Figure 19*. If the conditions (threshold and duration) are fulfilled, No-Motion event flag is set, and can be configured to drive the interrupt pin (INT1 or INT2), synced to the DRDY. When the slope falls out of the range of *(-Threshold, +Threshold),* the No-Motion conditions are not fulfilled, and the corresponding interrupt is reset.

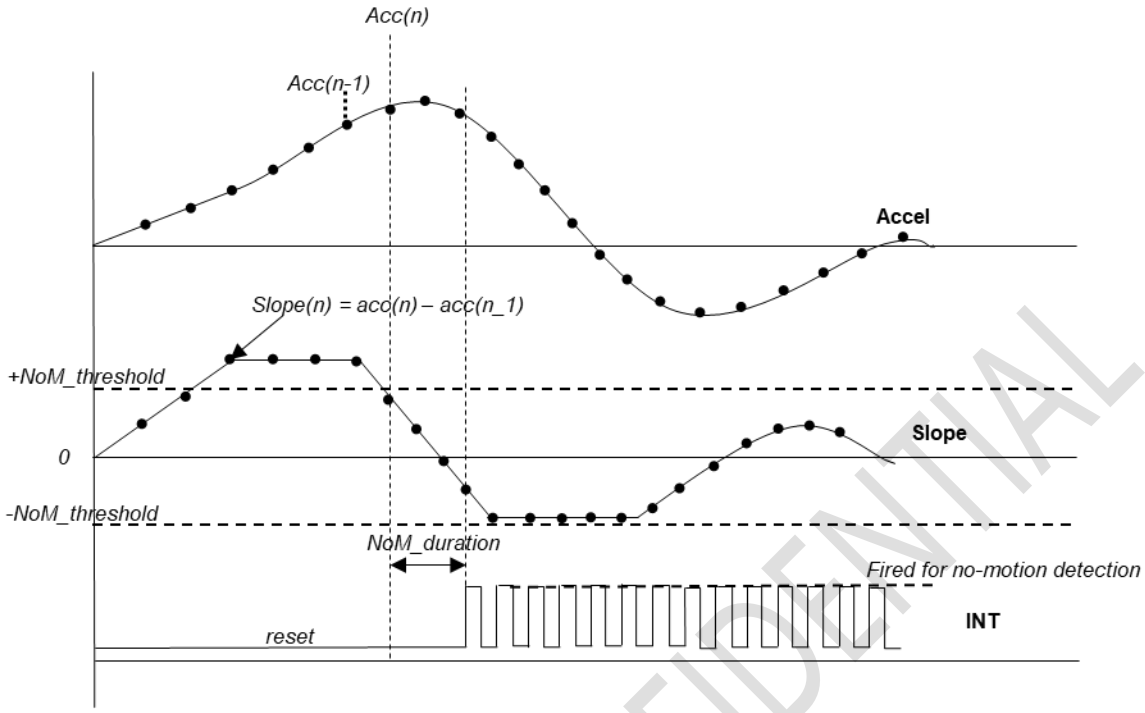


Figure 19. No-Motion interrupt detection

9.1.3 Significant-Motion Detection Principle

Significant-Motion Detection works based on Any-Motion and No-Motion events, so both Any-Motion and No-Motion should be configured and enabled to make sure Significant-Motion Detection runs properly. As shown in *Figure 20*.

After valid Any-Motion is detected, if a further Any-Motion is detected after *SigMotionWaitWindow* and before the *SigMotionConfirmWindow* time, the Significant-Motion event flag will be raised to the interrupt path. The Significant-Motion interrupt is cleared when No-Motion is detected.

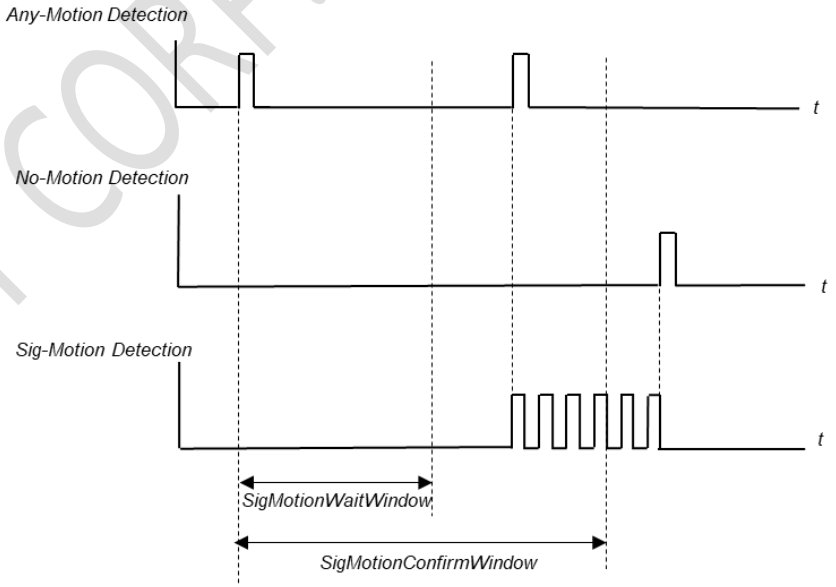


Figure 20. Significant-Motion interrupt detection

9.2 Motion Detection Flow

The flow chart of No-Motion Detection is show in Figure 21 No-Motion Detection Flow.

The flow chart of Any-Motion Detection is show in Figure 22 Any-Motion Detection Flow.

The flow chart of Significant-Motion Detection is show in Figure 23 Significant-Motion Detection Flow.

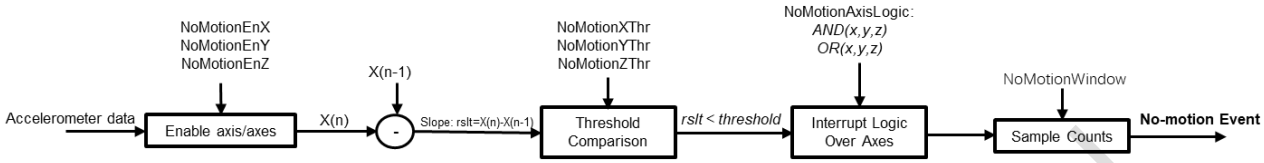


Figure 21. No-Motion Detection Flow

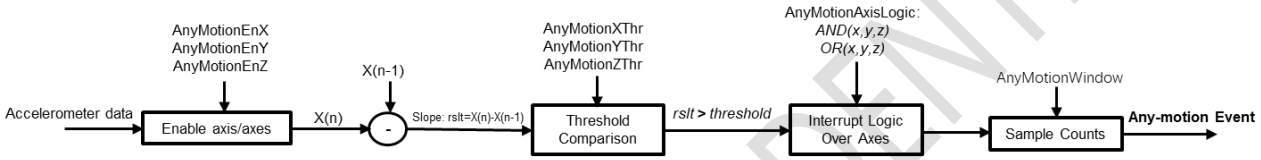


Figure 22. Any-Motion Detection Flow

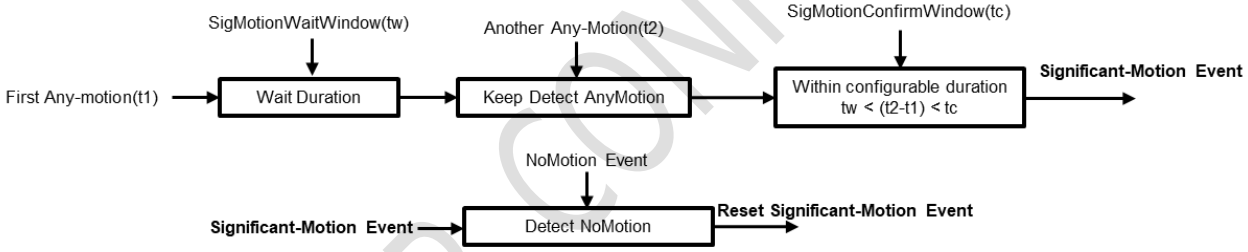


Figure 23. Significant-Motion Detection Flow

9.3 Parameters of Motion Detection

Table 36. Motion Detection Parameters

Parameter Name	Format	Resolution	Description
NoMotionAxisLogic	1-bit integer	1	0: Logic-OR between events of enabled axes for No-Motion detection 1: Logic-AND between events of enabled axes for No-Motion detection
NoMotionEnX NoMotionEnY NoMotionEnZ	1-bit integer	1	0: the corresponding axis is not involved for calculation 1: the corresponding axis data is calculated for No-Motion detection
NoMotionXThr NoMotionYThr NoMotionZThr	1-byte format unsigned, 5-bits fraction	0.03125g (1 / 32)	Defines the slope threshold of the corresponding axis for No-Motion detection
NoMotionWindow	1-byte integer	1 sample	Defines the minimum number of consecutive samples (duration) that the absolute of the slope of the enabled axis/axes data should keep lower than the threshold
AnyMotionAxisLogic	1-bit integer	1	0: Logic-OR between events of enabled axes for Any-Motion detection 1: Logic-AND between events of enabled axes for Any-Motion detection
AnyMotionEnX AnyMotionEnY AnyMotionEnZ	1-bit integer	1	0: the corresponding axis is not involved for calculation 1: the corresponding axis data is calculated for Any-Motion detection
AnyMotionXThr AnyMotionYThr AnyMotionZThr	1-byte format unsigned, 5-bits fraction	0.03125g (1 / 32)	Defines the slope threshold of the corresponding axis for Any-Motion detection
AnyMotionWindow	1-byte format unsigned, 5-bits fraction	1 sample	Defines the minimum number of consecutive samples (duration) that the absolute of the slope of the enabled axis/axes data should keep higher than the threshold
SigMotionWaitWindow	2-bytes integer	1 sample	Defines the wait window (idle time) starts from the first Any-Motion event until starting to detecting another Any-Motion event for confirmation
SigMotionConfirmWindow	2-bytes integer	1 sample	Defines the maximum duration for detecting the other Any-Motion event to confirm Significant-Motion, starts from the first Any-Motion event

Note: Once the Logic-OR is selected, the calculation on any enabled axis and reported one valid event, will trigger the corresponding Motion Detection (Any-Motion or No-Motion). Once the Logic-AND is selected, the calculation on all enabled axes should report valid event, to trigger the corresponding Motion Detection (Any-Motion or No-Motion).

The MOTION_MODE_CTRL byte consists of the parameters shown in *Table 37*.

Table 37. MOTION_MODE_CTRL Content

7	6	5	4	3	2	1	0
NoMotionAxisLogic	NoMotionEnZ	NoMotionEnY	NoMotionEnX	AnyMotionAxisLogic	AnyMotionEnZ	AnyMotionEnY	AnyMotionEnX

9.4 Config Motion Detection

The Motion Detection parameters are divided into two sets and can be passed to the QMI8A01 internal algorithm through two callings of CTRL9 command. As shown in *Table 38*.

Host should write the parameters to the corresponding registers, according to *Table 38*. Especially, write 0x01 to CAL4_H register for the first set of parameters, while write 0x02 to CAL4_H for the second set of parameters. Refer to *Table 33* for address of CAL registers.

Then trigger the CTRL9 command with 0x0E(CTRL_CMD_CONFIGURE_MOTION). Refer to *Table 32* for details.

Table 38. Write Motion Detection Parameters to QMI8A01

Register (bits)	First CTRL9 Command	Second CTRL9 Command
CAL1_L (7:0)	AnyMotionXThr.	AnyMotionWindow.
CAL1_H (7:0)	AnyMotionYThr.	NoMotionWindow
CAL2_L (7:0)	AnyMotionZThr.	SigMotionWaitWindow[7:0]
CAL2_H (7:0)	NoMotionXThr.	SigMotionWaitWindow [15:8]
CAL3_L (7:0)	NoMotionYThr.	SigMotionConfirmWindow[7:0]
CAL3_H (7:0)	NoMotionZThr.	SigMotionConfirmWindow[15:8]
CAL4_L (7:0)	MOTION_MODE_CTRL	NA
CAL4_H (7:4)	0x01(means 1st command)	0x02(means 2nd command)
CTRL9 code	0x0E(CTRL_CMD_CONFIGURE_MOTION)	0x0E(CTRL_CMD_CONFIGURE_MOTION)

Note: Configuration should be done when accelerometer and gyroscope are disabled (CTRL7.aEN = CTRL7.gEN =0).

9.5 Enabling Motion Detection

After successfully passing the parameters to QMI8A01 Motion Detection engine, host need to enable the Any-Motion Detection engine by setting CTRL8.bit1 to 1, enable the No-Motion Detection engine by setting CTRL8.bit2 to 1, enable the Significant-Motion Detection engine by setting CTRL8.bit3 to 1. If the accelerometer is configured and enabled too, the enabled engines will be started to detect the corresponding events.

On contrary, set the bit(s) in CTRL8.bit[3:1] to 0, will disable the corresponding engine. This can be done when accelerometer and/or gyroscope are(is) enabled or disabled. Refer to *5.3 Configuration* Registers for the details of CTRL8.

9.6 Motion Interrupt

Once the Any-Motion, No-Motion and/or Significant-Motion event(s) is/are reported, the corresponding Motion Detection Event will be generated.

The Motion Event (internal signal) can be selected to drive INT1(CTRL8.bit6 = 1) or INT2(CTRL8.bit6 = 0). And once the corresponding INT pin is enabled (by CTRL1.bit3 for INT1, or CTRL1.bit4 for INT2), the Motion interrupt can be seen on the INT pin (synced with DRDY). Refer to *5.3* Configuration Registers for details.

At the meantime, the STATUS1.bit[7:5] is updated and host can read these bits to confirm the Motion interrupt status. Note that the STATUS1.bit[5] = 1 represents the Any-Motion was detected, STATUS1.bit[6] = 1 represents the No-Motion was detected, STATUS1.bit[7] = 1 represents the Significant-Motion was detected. Refer to *5.3* Configuration Registers for details.

10 Tap

The Tap engine detects the Single-Tap or Double-Tap, if enabled.

The calculation of the Tap Detection is based on the accelerometer ODR defined by *CTRL2.aODR*, refer to *Table 23* for details.

The Tap detection can only work in Non-SyncSample mode, refer to for details 6.2 Non-SyncSample mode.

10.1 Tap Detection Principle

Figure 24 shows the principle of Tap detecting, includes Single Tap and Double Tap.

The acceleration data of the three axes (x, y, z) is dynamically averaged to get the Average of Acceleration. Alpha parameter defines the ratio/weight of the averaging calculation.

Linear Acceleration is calculated: Linear Acceleration = Acceleration – Average of Acceleration.

The Average Movement Magnitude indicates the movement energy level, is used to detect the Quiet status. It is calculated with *Gamma.*

If the square sum of the Linear Acceleration of three axes is higher than the *PeakMagThr*, the peak detecting is started. If later at the end of *PeakWindow*, the Average of the Movement Magnitude is lower than the *UDMThr (Undefined Motion Threshold)*, it means the vibration is low and return to Quiet status, it is considered a valid Peak is detected.

If a valid Peak is detected, and no further significant vibration within the *TapWindow* (be quiet after the Peak, no further Tap), a valid Tap is detected.

Once a valid Tap (first Tap) is detected, the second Tap of the Double-Tap should be detected after the TapWindow (Quiet time after first Tap) and before the *DTapWindow* (timeout of the Double Tap detection), and the Double-Tap event will be reported.

If the second Tap is detected within the *TapWindow*, it is considered as the Undefined Motion, and will reset the full Tap process, no Tap event will be reported.

If no further Tap is detected within the *DTapWindow*, Single-Tap event will be reported. The Tap detected after *DTapWindow*, is considered as the First Tap of the new round of detecting.

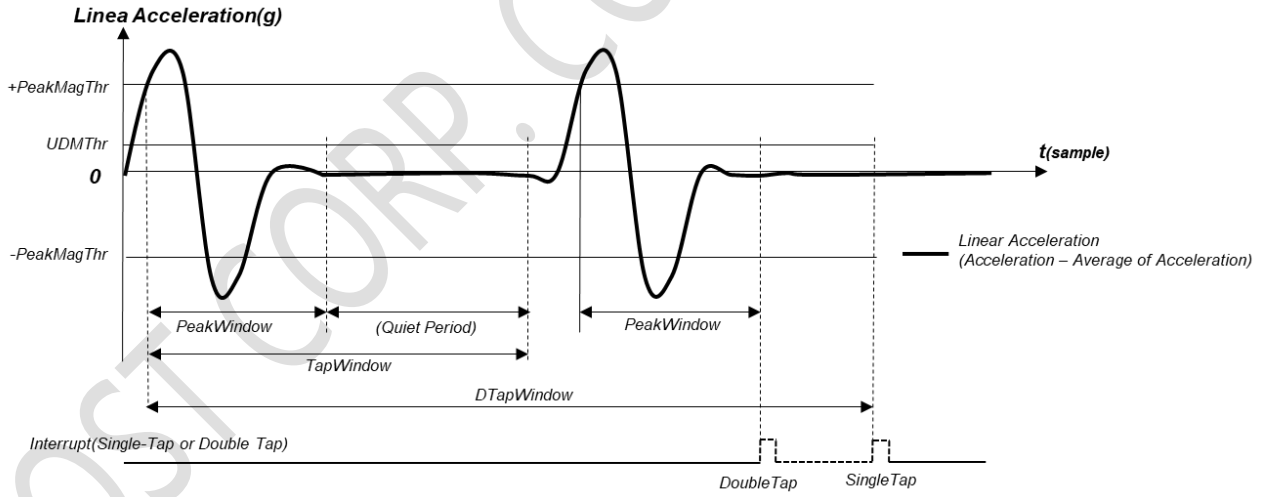


Figure 24. Tap Detect Principle

10.2 Tap Detection Parameters

The parameters listed in *Table 39* are used to configure the Tap Engine.

Table 39. Tap Parameters

Parameter Name	Format	Resolution	Description
Priority	1-byte integer	NA	Priority definition between the x, y, z axes of acceleration. Only Priority[2:0] bits are used. The axis that outputs the first peak of Linear Acceleration in a valid Tap detection, will be considered the Tap axis. However, there is possibility that two or three of the axes show same Linear Acceleration at exactly same time when reaching (or exceeding) the PeakMagThr. In this case, the defined priority is used to judge and select the defined priority axis as the Tap axis.
PeakWindow	1-byte integer	1 sample	Defines the maximum duration (in sample) for a valid peak. In a valid peak, the linear acceleration should reach or be higher than the *PeakMagThr* and should return to quiet (no significant movement) within *UDMThr*, at the end of *PeakWindow*. e.g., 20 @500Hz ODR
TapWindow	2-bytes integer	1 sample	Defines the minimum quiet time before the second Tap happens. After the first Tap is detected, there should be no significant movement (defined by *UDMThr*) during the *TapWindow*. The valid second tap should be detected after *TapWindow* and before *DTapWindow*. e.g., 50 @500Hz ODR
DTapWindow	2-bytes integer	1 sample	Defines the maximum time for a valid second Tap for Double Tap, count start from the first peak of the valid first Tap. e.g., 250 @500Hz ODR
Alpha	1-byte unsigned, 7-bits fraction	0.0078 (1/128)	Defines the ratio for calculation of the average of the acceleration. The bigger the *Alpha*, the bigger the weight of the latest data. e.g., 0.0625
Gamma	1-byte unsigned, 7-bits fraction	0.0078 (1/128)	Defines the ratio for calculating the average of the movement magnitude. The bigger the *Gamma*, the bigger the weight of the latest data. e.g., 0.25
PeakMagThr	2-bytes unsigned, 10-bits fraction	$0.001g^2$ (1/ 1024)	Threshold for peak detection. e.g., $0.8g^2$ (0x0320)
UDMThr	2-bytes unsigned, 10-bits fraction	0.001g (1/ 1024)	Undefined Motion threshold. This defines the threshold of the Linear Acceleration for quiet status. e.g., $0.4g^2$ (0x0190)

The *Priority[2:0]* is defined as below:

0: (X > Y> Z)

1: (X > Z > Y)

2: (Y > X > Z)

3: (Y > Z > X)

4: (Z > X > Y)

5: (Z > Y > X)

6, 7: (Same as 0)

10.3 Configure Tap

The Tap parameters are divided into two sets and can be passed to the QMI8A01 internal algorithm through two callings of CTRL9 command. As shown in *Table 40*.

Host should write the parameters to the corresponding registers, according to *Table 40*. Especially, write 0x01 to CAL4_H register for the first set of parameters, while write 0x02 to CAL4_H for the second set of parameters. Refer to *Table 33* for address of CAL registers.

Then trigger the CTRL9 comment with 0x0C(CTRL_CMD_CONFIGURE_TAP). Refer to *5.10* for details.

Table 40. Write Tap Parameters to QMI8A01

Register (bits)	First Command Set	Second Command Set
CAL1_L (7:0)	PeakWindow[7:0]	Alpha[7:0]
CAL1_H (7:0)	Priority[7:0] (actually only [2:0] is used)	Gamma[7:0]
CAL2_L (7:0)	TapWindow[7:0]	PeakMagThr[7:0]
CAL2_H (7:0)	TapWindow[15:8]	PeakMagThr[15:8]
CAL3_L (7:0)	DTapWindow[7:0]	UDMThr[7:0]
CAL3_H (7:0)	DtapWindow[15:8]	UDMThr[15:8]
CAL4_L (7:0)	NA	NA
CAL4_H (7:4)	0x01(means 1st command)	0x02(means 2nd command)
CTRL9 code	0x0C(CTRL_CMD_CONFIGURE_TAP)	0x0C(CTRL_CMD_CONFIGURE_TAP)

Note: Configuration should be done when accelerometer and gyroscope are disabled (CTRL7.aEN = CTRL7.gEN =0).

10.4 Enable Tap Detection

After successfully passing the parameters to QMI8A01 Tap engine, the host needs to enable the Tap engine by setting CTRL8.bit0 to 1. If the accelerometer is properly configured and enabled (CTRL7.aEN = 1), Tap engine will be started to detect the taps. To detect the Tap activity, it is recommended to set accelerometer ODR to higher than 200Hz (defined by CTRL2.aODR).

If CTRL8.bit0 is set to 0, or CTRL7.aEN = 0, the Tap engine will be stopped to detect the taps.

10.5 Tap Interrupt

Once the Tap event is reported (Single- or Double-Tap), it can generate the Tap interrupt signal.

The Tap event (internal signal) can be selected to drive INT1(CTRL8.bit6 = 1) or INT2(CTRL8.bit6 = 0). And once the corresponding INT pin is enabled (by CTRL1.bit3 for INT1, or CTRL1.bit4 for INT2), the Tap interrupt will be seen on the INT pin (synced with DRDY). Refer to *5.3 Configuration* Registers for details.

At the meantime, the STATUS1.bit1 is set and host can read this bit to confirm the Tap (Single- or Double-Tap) interrupt is generated.

10.6 Tap Detection Output

When the Tap (Single- or Double-Tap) is detected, the information of the Tap is presented in TAP_STATUS (register 0x59). The TAP_STATUS is updated and valid after the Tap event is detected (STATUSINT.bit1 = 1).

The TAP_NUM indicates the Single-Tap (TAP_STATUS.TAP_NUM = 1) or Double-Tap (TAP_STATUS.TAP_NUM = 2) was detected.

The TAP_STATUS.TAP_AXIS indicates the first valid peak of the Tap happens on X axis (TAP_STATUS.TAP_AXIS = 1), or Y axis (TAP_STATUS.TAP_AXIS = 2), or Z axis (TAP_STATUS.TAP_AXIS = 3). Note the TAP_AXIS judgement follows the definition of the Tap Priority. Refer to *Table 39* for details.

The TAP_STATUS.TAP_POLARITY indicates the direction from which the Tap moves towards QMI8A01. Note, this direction is derived from the value of the linear acceleration of first valid Peak.

11 Locking Mechanism

11.1 Locking Mechanism Principle

The Locking Mechanism function is enabled in SyncSample mode, can lock the sensor data and keep the values in data registers after a proper locking process.

As the concept of “shadow register”, which enables host to read the locked data in unlimited delay without the risk of mixing the two consecutive data if the new data comes and updates to the sensor data registers during the host read the sensor data registers. Refer to 6.1 SyncSample mode.

11.2 Locking Mechanism Data Reading Process

Figure 25 shows the process of reading data in Locking Mechanism.

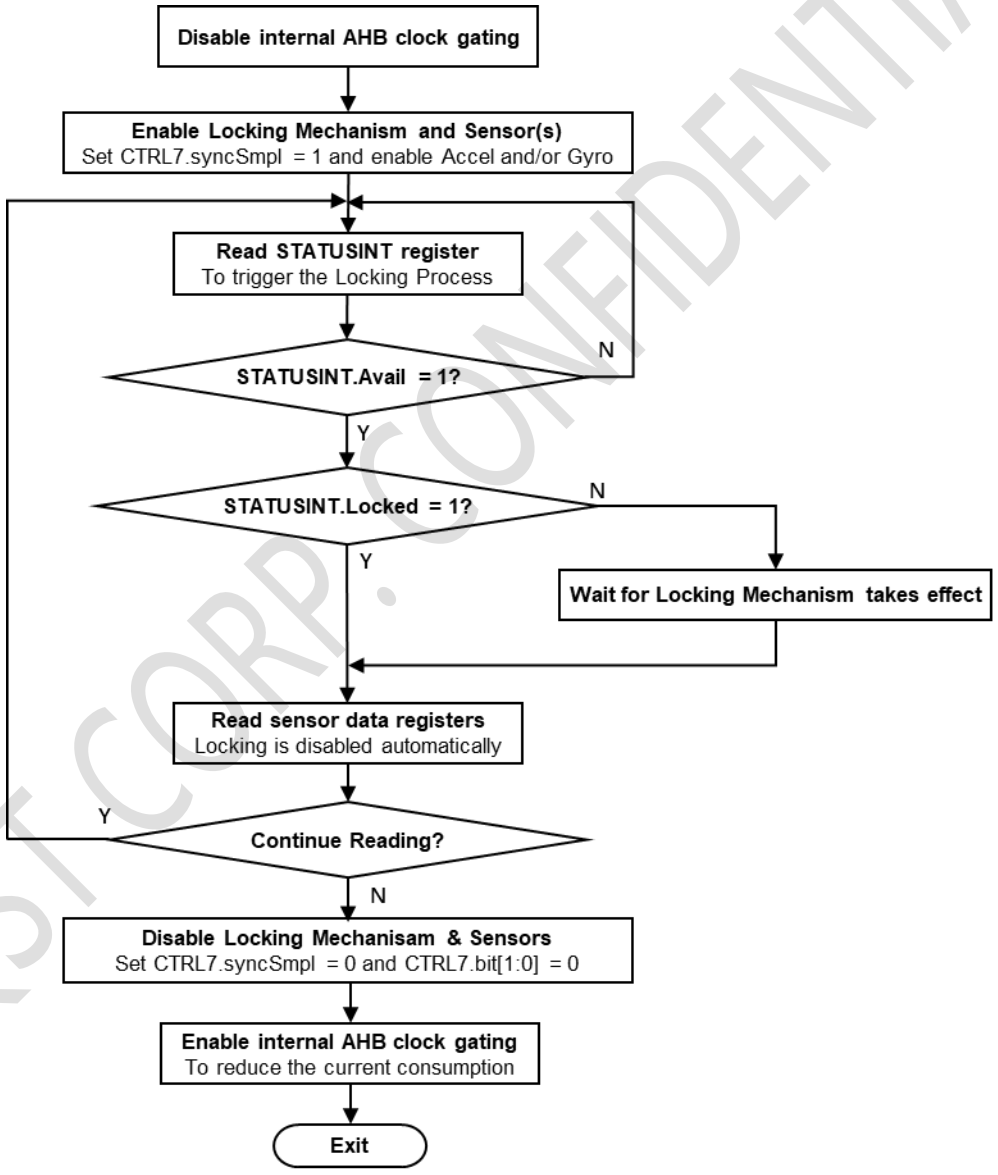


Figure 25. Data Reading Process in Locking Mechanism

11.2.1 **Disable/Enable AHB Clock Gating**

First, the host needs to disable the internal AHB clock gating, this will fully screen out the possibility of misalignment of the two consecutive data samples. Note this is ONLY needed when using SPI (accel only mode with ODR less than 500Hz), I2C or I3C interfaces (in all ODRs).

Host can disable the internal AHB clock gating by applying:

1- write 0x01 to CAL1_L register.

2- write 0x12 (CTRL_CMD_AHB_CLOCK_GATING) in CTRL9 protocol, refer to 5.9 and 5.10.6.10.

After disabling the sensor, enable back the clock gating by:

1- write 0x00 to CAL1_L register.

2- write 0x12 (CTRL_CMD_AHB_CLOCK_GATING) in CTRL9 protocol, refer to 5.9 and 5.10.6.10.

11.2.2 **Enable Locking Mechanism**

The locking Mechanism is enabled when setting CTRL7.bit7 to "1" (*syncSmpl*) and enable accelerometer and/or gyroscope.

1- Enable 6DOF in this mode, write 0x83 to CTRL7.

2- Enable only Accel in this mode, write 0x81 to CTRL7.

3- Enable only Gyro in this mode, write 0x82 to CTRL7.

11.2.3 **Reading Sensor Data**

When the Locking Mechanism is enabled, the reading to the STATUSINT register when new sensor data is available (STATUSINT.Avail = 1) will trigger the locking of the current sensor data sample. Once the data sample is locked, new data will be dropped, until the release of the locking. The Locking Mechanism is automatically released after the host reads GZ_H if gyroscope is enabled or AZ_H if accelerometer only is enabled.

Process of data reading in Locking Mechanism mode:

Read STATUSINT register, if STATUSINT.Avail = 1, the locking mechanism is started to take effect, go to step 2. If STATUSINT.Avail = 0, repeat step 1.

If STATUSINT.Avail = 1 and STATUSINT.Locked = 0, means data looking is in progress, and will be locked within Data_Lock_Delay. If STATUSINT.Locked = 1, go to step 4.

1- Wait for Data_Lock_Delay duration (refer to *11.3*), go to step 4.

Burst read the sensor data until the last byte of enabled sensor(s) (to release the locking), Refer to *Table 29*.

2- Repeat step 1 – 4, until Exit

11.3 Data_Lock_Delay

When the gyroscope is enabled the duration of Data_Lock_Delay is shown in Table 41.

Table 41. Data_Lock_Delay When Gyroscope Is Enabled

ODR setting	ODR(Hz)	Data_Lock_Delay (usec)
0	7174.4	2
1	3587.2	2
2	1793.6	4
3	896.8	6
4	448.4	12
5	224.2	12
6	112.1	12
7	56.05	12
8	28.025	12

When the gyro is not enabled (accel only mode) the value of Data_Lock_Delay is shown in Table 42.

Table 42. Data_Lock_Delay When Gyroscope Is Disabled

ODR setting	ODR(Hz)	Data_Lock_Delay (usec)
3	1000	6
4	500	12
5	250	24
6	125	48
7	62.5	48
8	31.25	48
12	128	40
13	21	100
14	11	200
15	3	270

11.4 Exit Locking Mechanism

Once pulling sensor data from QMI8A01 is finished, it can be configured to exit the Locking Mechanism mode, by disabling the sensors by setting CTRL7.bit[1:0] to 0, and enabling the AHB Clock Gating (refer to *11.2.1*).

11.5 On-The-Fly ODR Change in Locking Mechanism

The on-the-fly ODR changing is supported, so the host can change the ODR of sensor without disable the sensor.

An example sequence of changing ODR without disabling the sensor in locking mechanism is shown below:

1- Write CTRL2/CTRL3 to set the ODR's of accelerometer and gyroscope and full scales.

2- Write 0x81 / 0x82 / 0x83 to CTRL7.

3- Read Sensor Data according 11.2.3.

4- Changing ODR on-the-fly:

   a- Write 0x01/0x02/0x03 to CTRL7(clear the syncSmpl bit).

   b- Wait 1ms.

   c- Clear the Locking Mechanism in case the data is still locked from previous ODR by reading GZ_H if gyroscope is enabled or AZ_H if accelerometer only is enabled.

   d- Write CTRL2 / CTRL3 with the new ODR's.

   e- Write 0x81/0x82/0x83 to CTRL7.

5- Start poll and read Sensor Data using the new ODRs based on 11.2.3.

Note that the new data will be stable in at least 3 samples for filter to settle down, therefore, it is recommended to discard the first several samples at host side.

12 Calibration-On-Demand (COD)

12.1 COD Principle

The Calibration-On-Demand supports the on-demand calibration of Gyro X and Y axes. Based on the internal integrated functionality, the QMI8A01 can calibrate the internal gain of X & Y axes of gyroscope, result in a more precise sensitivity, and a tighter distribution of the X & Y axes sensitivity over QMI8A01 chips.

Note that the Z axis of gyroscope is not influenced by COD.

12.2 Run COD

To run the COD, host need to

1. Set CTRL7.aEN = 0 and CTRL7.gEN = 0, to disable the accelerometer and gyroscope.
2. Issue the CTRL_CMD_ON_DEMAND_CALIBRATION (0xA2) by CTRL9 command.
3. And wait approximately 1.5 seconds for QMI8A01 to finish the CTRL9 command.
4. Read the COD_STATUS register (0x46) to check the result/status of the COD implementation.

During the process, it is recommended to place the device in quiet, otherwise, the COD might fail and report error.

If succeeds, the recalibrated gain parameters will be applied to the sensor data afterwards. The updated gains are output to the UI registers and can be read by host, refer to *12.3*. The recalibrated gain parameters will be lost if a Power-On Reset or soft reset is implemented, QMI8A01 will then use the on-chip default gain parameters.

If failed, there is no influence on the operation of gyroscope, QMI8A01 will keep using the previous workable parameters (last successful COD parameters or the on-chip default parameters).

12.3 COD Status

If the COD command is successfully implemented, the COD_STATUS register will output 0x00 for the indication.

The non-zero value of COD_STATUS indicates different modes of failure. Refer to 5.7 Calibration-On-Demand (COD) Status Register for details.

12.4 Save and Restore the New Gain Parameters

After a successful COD, the new gains with COD correction will be applied to the future data of X and Y axes of Gyroscope. At the meantime, the new parameters are updated to the registers below, for host to read and save.

1. Gyro-X gain (16 bits) will be in dVX_L and dVX_H registers (0x51, 0x52)
2. Gyro-Y gain (16 bits) will be in dVY_L and dVY_H registers (0x53, 0x54)
3. Gyro-Z gain (16 bits) will be in dVZ_L and dVZ_H registers (0x55, 0x56)

If the host saved those gain parameters, it is possible to pass them back to the QMI8A01 (without invoking again the COD routine), using the CTRL9 command CTRL_CMD_APPLY_GYRO_GAINS (0xAA) as follow:

1. Disable Accelerometer and Gyroscope by setting CTRL7.aEN = 0 and CTRL7.gEN = 0
2. write Gyro-X gain (16 bits) to registers CAL1_L and CAL1_H registers (0x0B, 0x0C)
3. write Gyro-Y gain (16 bits) to registers CAL2_L and CAL2_H registers (0x0D, 0x0E)
4. write Gyro-Z gain (16 bits) to registers CAL3_L and CAL3_H registers (0x0F, 0x10)
5. Write 0xAA to CTRL9 and follow CTRL9 protocol.

Once the CTRL9 command is successfully finished, the restored gains will take effects for future data of Gyroscope.

Note that it is always recommended to run the COD from time to time to apply the precise and up-to-date correction of the Gyro-X and Gyro-Y sensitivity. Designer should be careful to restore the out-of-date gain parameters, especially when there is significant change of PCB stress.

13 Self-Test (Check-Alive)

13.1 Accelerometer Self-Test

The accelerometer Self-Test (Check-Alive) is used to determine if the accelerometer is functional and working within acceptable parameters.

It is implemented by applying an electrostatic force to actuate each of the three X, Y, and Z axis of the accelerometer. If the accelerometer mechanical structure responds to this input stimulus by sensing at least 200 mg, then the accelerometer can be considered functional.

The accelerometer Self-Test data is available to be read at registers dVX_L, dVX_H, dVY_L, dVY_H, dVZ_L and dVZ_H. The Host can initiate the Self-Test at any time with the following procedure.

Procedure for accelerometer Self-Test:

1- Disable the sensors (CTRL7 = 0x00).

2- Set proper accelerometer ODR (CTRL2.aODR) and bit CTRL2.aST (bit7) to 1 to trigger the Self-Test.

3- Wait for QMI8A01 to drive INT2 to High, if INT2 is enabled (CTRL1.bit4 = 1), or STATUSINT.bit0 is set to 1.

4- Set CTRL2.aST(bit7) to 0, to clear STATUSINT1.bit0 and/or INT2.

5- Check for QMI8A01 drives INT2 back to Low and sets STATUSINT1.bit0 to 0.

6- Read the Accel Self-Test result:

X channel: dVX_L and dVX_H (registers 0x51 and 0x52)

Y channel: dVY_L and dVY_H (registers 0x53 and 0x54)

Z channel: dVZ_L and dVZ_H (registers 0x55 and 0x56)

The results are 16-bits in format signed U5.11, resolution 0.5mg (1 / 2^11 g).

If the absolute results of all three axes are higher than 200mg, the accelerometer can be considered functional. Otherwise, the accelerometer cannot be considered functional.

Note that the Self-Test function will automatically set the full-scall to 16g and use the aODR set by user (CTRL2.aODR). At the end of Self-Test, QMI8A01 will update CTR2 with the original value user set before starting the Check-Alive) routine.

The typical time for Self-Test (from setting aST to 1, until the rising edge of INT2 if enabled, or STATUSINT.bit0 is set to 1) costs approximately 25 ODRs:

25ms @ 1KHz ODR

800ms @ 32Hz ODR

2.2s @ 11Hz ODR

13.2 Gyroscope Self-Test

The gyroscope Self-Test (Check-Alive) is used to determine if the gyroscope is functional.

It is implemented by applying an electrostatic force to actuate each of the three X, Y, and Z axis of the gyroscope and measures the mechanical response on the corresponding X, Y, and Z axis. If the equivalent magnitude of the gyroscope output is greater than 300dps for each axis, the gyroscope can be considered as functional.

The gyroscope Self-Test data is available to be read at output registers dVX_L, dVX_H, dVY_L, dVY_H, dVZ_L & dVZ_H. The Host can initiate the Self-Test anytime with the following procedure.

Procedure for gyroscope Self-Test:

1- Disable the sensors (CTRL7 = 0x00).

2- Set the bit gST to 1. (CTRL3.bit7 = 1’b1).

3- Wait for QMI8A01 to drive INT2 to High, if INT2 is enabled, or STATUSINT.bit0 is set to 1.

4- Set CTRL3.aST(bit7) to 0, to clear STATUSINT1.bit0 and/or INT2.

5- Check for QMI8A01 drives INT2 back to Low or sets STATUSINT1.bit0 to 0.

6- Read the Gyro Self-Test result:

X channel: dVX_L and dVX_H (registers 0x51 and 0x52)

Y channel: dVY_L and dVY_H (registers 0x53 and 0x54)

Z channel: dVZ_L and dVZ_H (registers 0x55 and 0x56)

Read the 16 bits result in format signed U12.4, resolution is 62.5mdps (1 / 2^4 dps).

If the absolute results of all three axes are higher than 300dps, the gyroscope can be considered functional. Otherwise, the gyroscope cannot be considered functional.

Note that the Self-Test function will automatically set the full-scale (gFS) and ODR (gODR) of CTRL3. At the end of Self-Test, QMI8A01 will update CTR3 with the original value user set before starting the Self-Test routine.

The typical time (from writing gST to 1, until the rising edge of INT2 if enabled, or STATUSINT.bit0 set to 1) cost for the Self-Test process is approximately 400ms.

14 Host Serial Interface

QMI8A01 Host Serial Interface supports slave interfaces of I^2C, MIPI I^3C and SPI. For SPI, it supports both 3-wire and 4-wire modes. The basic timing characteristics for the interface are described below. Through the QMI8A01 Host Serial Interface, the host can access, set up and control the QMI8A01 UI Registers.

14.1 Address Auto Increment

During burst reads and writes, the target address can be configured in Auto-Increment or Non-Increment, by CTRL1.ADDR_AI(bit 6), refer to *5.3*.

For example, during burst read (burst write is similar) started from register 0x0B(CAL1_L):

If ADDR_AI = 0, the register address will not increase, and QMI8A01 will output the content of CAL1_L, CAL1_L, CAL1_L …, as long as host continues the burst read clock.

If ADDR_AI = 1, the register address will automatically increase, and QMI8A01 will output the content of CAL1_L, CAL1_H, CAL2_L …, as long as host continues the burst read clock.

Note that the default value of ADDR_AI is 0, so it is recommended to set it to 1 from beginning, in case of burst read/write is required.

Note that, burst writes to Configuration Registers (refer to *Table 23*) are NOT supported. These registers should be written in single cycle mode only.

14.2 Serial Peripheral Interface (SPI)

14.2.1 SPI Features

QMI8A01 supports both 3-wire and 4-wire modes in the SPI slave interface. The SPI 4-wire mode uses two control lines (CS, SPC) and two data lines (SDI, SDO). The SPI 3-wire mode uses the same control lines and one bi-directional data line (SDIO). The SDI /SDIO pin is used for both 3- and 4-wire modes and is configured based on the mode selected.

SPI transactions can be done in either Mode 0 (CPOL=0, CPHA=0) or Mode 3 (CPOL=1, CPHA=1). The QMI8A01 interface automatically detects which mode is in use and configures clocking accordingly.

SPI 3- or 4-wire modes are configured by setting CTRL1.bit7. SPI 3-wire mode is selected when CTRL1.bit7 = 1. The default configuration is SPI 4-wire mode, that CTRL1.bit7 is 0.

Figure 26 shows the SPI address and data formats.

SPI Features

- Data is latched on the rising edge of the clock
- Data should change on falling edge of clock
- Maximum frequency is 15 MHz
- Data is delivered MSB first
- Support single read/writes and multi cycle (Burst) read/writes.
- **NOTE**: burst writes to Configuration Registers (refer to Table 23) are NOT supported. These registers should be written in single cycle mode only.
- Supports 7-bit Address format and 8-bit data format.

Address Format

MSB							LSB
Read	A6	A5	A4	A3	A2	A1	A0

Read – indicates read (1) or write (0) transaction relative to the SPI master

Data Format

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

Figure 26. SPI Address and Data Format

14.2.2 SPI Interface Connection

In a typical SPI Master and Multi-Slave configuration, the SPI master shares the SPI clock (SPC), the serial data input (SDI), and the Serial Data Output (SDO) with all the connected SPI slave devices. Unique Chip Select (CS) lines connect each SPI slave to the master.

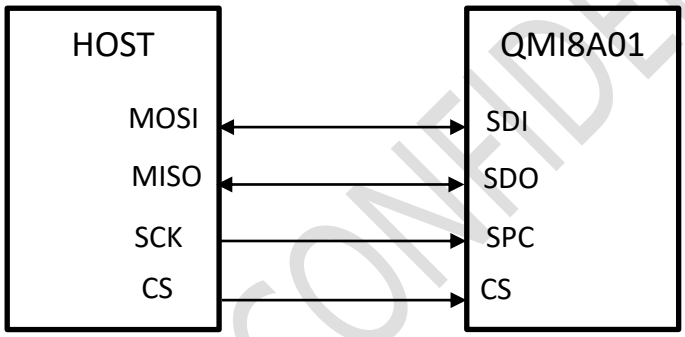


Figure 27. Typical SPI 4-Wire Connection

Figure 27 and Figure 28 show typical multi-slave 4- and 3-wire configurations. The primary difference between the two configurations is that the SDI and SDO lines are replaced by the bi-directional SDIO line. The SDIO line is driven by the master with both address and data when it is configured for write mode. During read mode, the SDIO line is driven by the master with the address, and subsequently driven by the “addressed” slave with data.

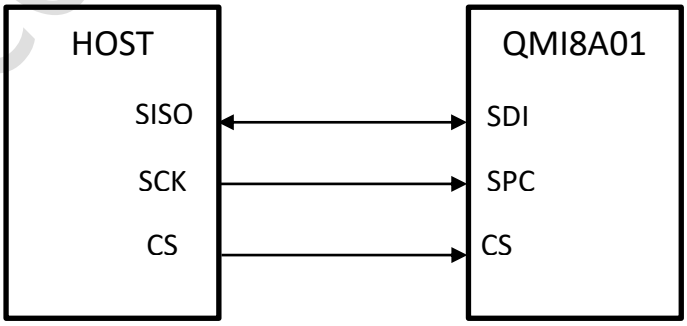


Figure 28. Typical SPI 3-Wire Connection

14.2.3 SPI Transaction Protocol

Figure 29 and Figure 30 illustrate the waveforms for both 4-wire and 3-wire SPI read and write transactions. Note that CS is active during the entire transaction.

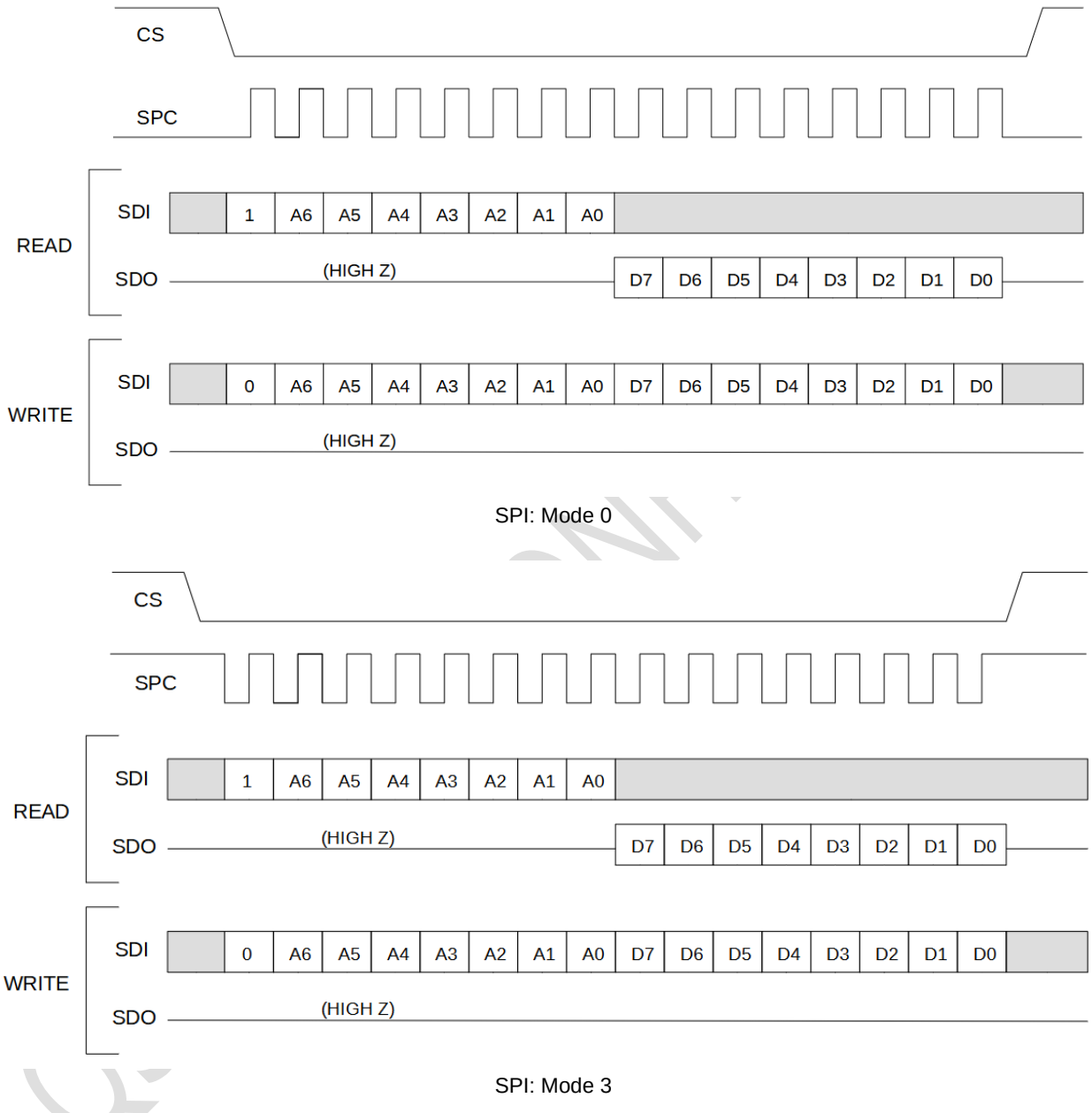


Figure 29. SPI 4-Wire Single Byte Read and Write (Mode 0 and Mode 3)

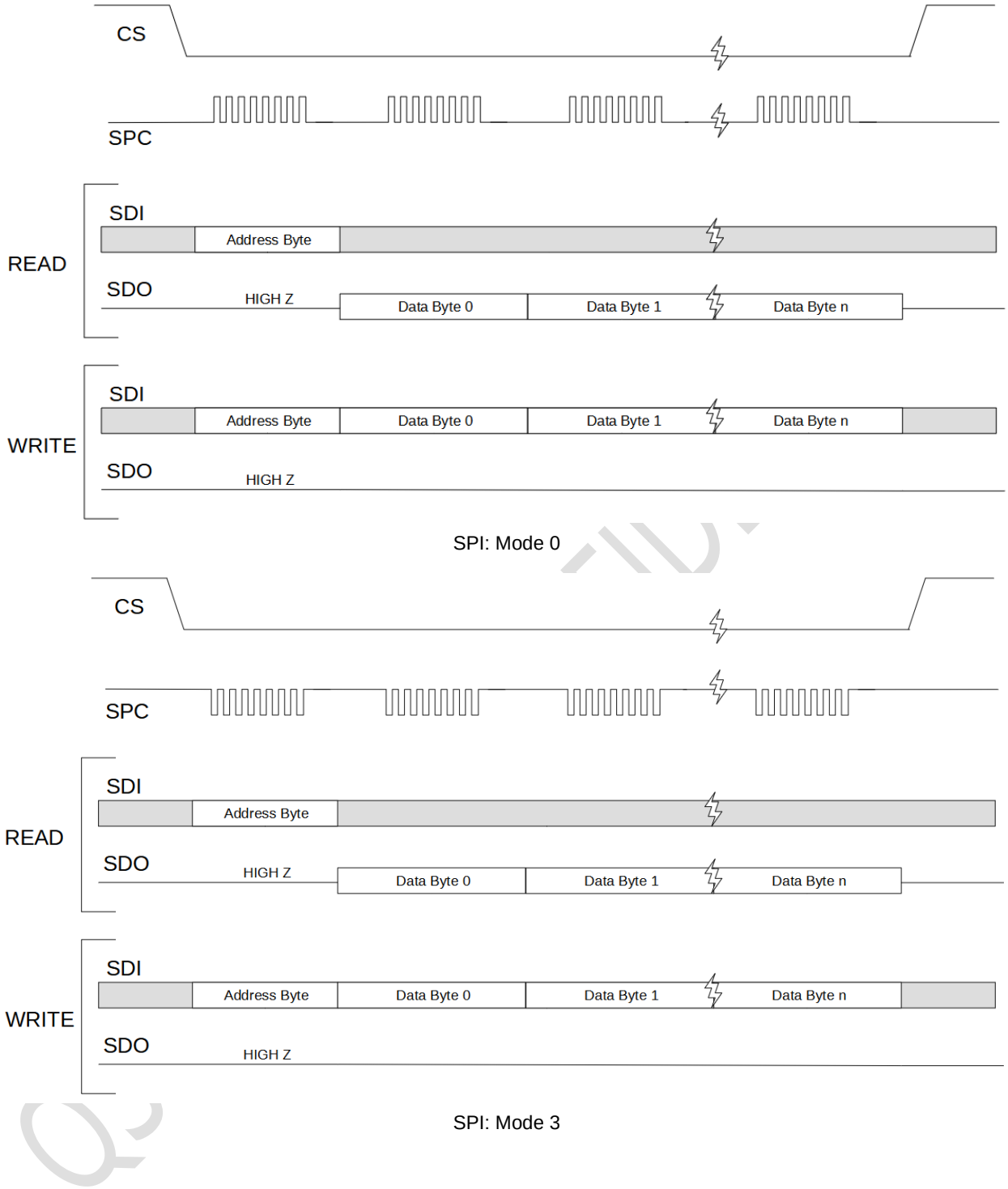


Figure 30. SPI 4-Wire Multi-Byte Read and Write Transactions

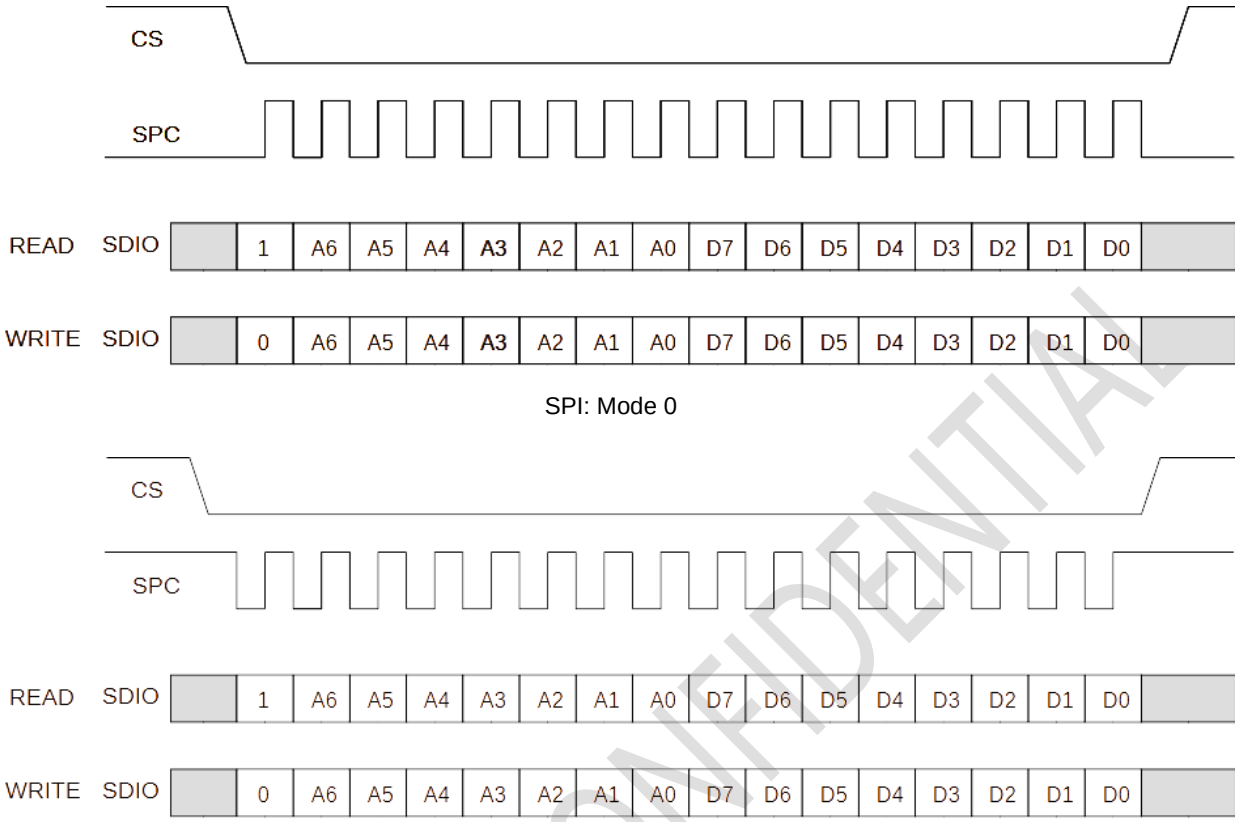


Figure 31. SPI 3-Wire Single Byte Read and Write Transactions

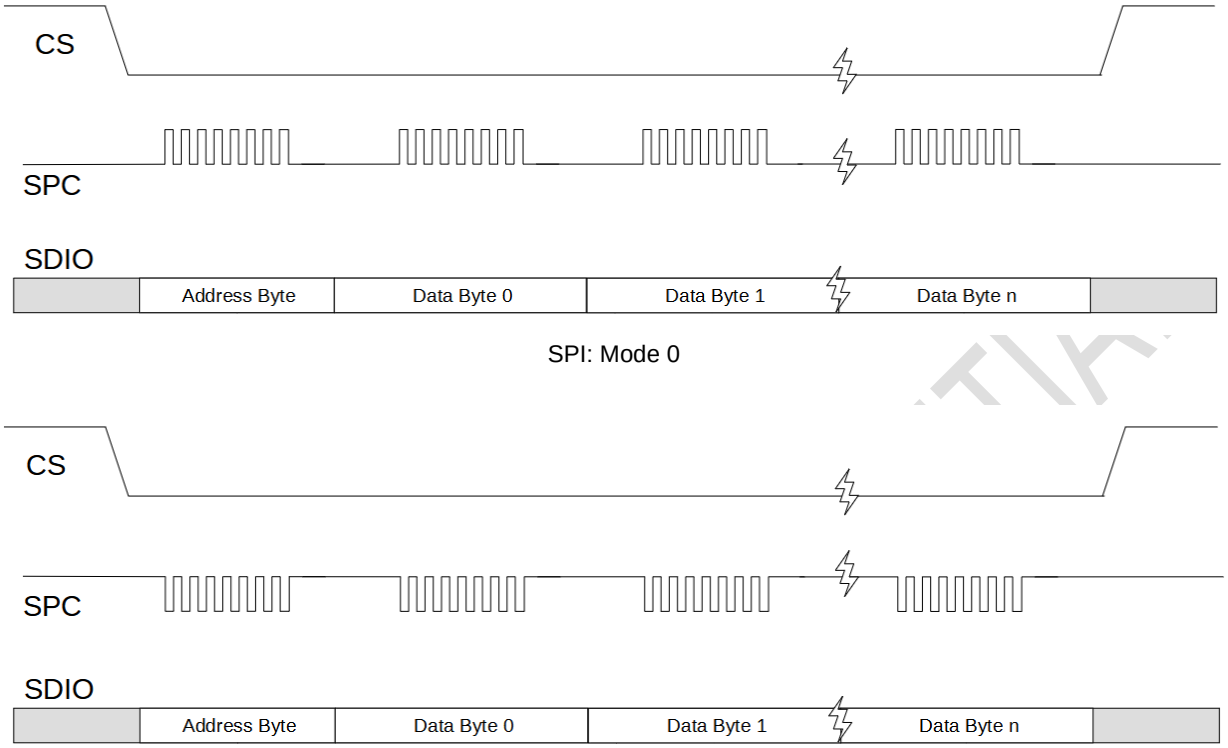


Figure 32. SPI 3-Wire Multi-Byte Read and Write Transactions

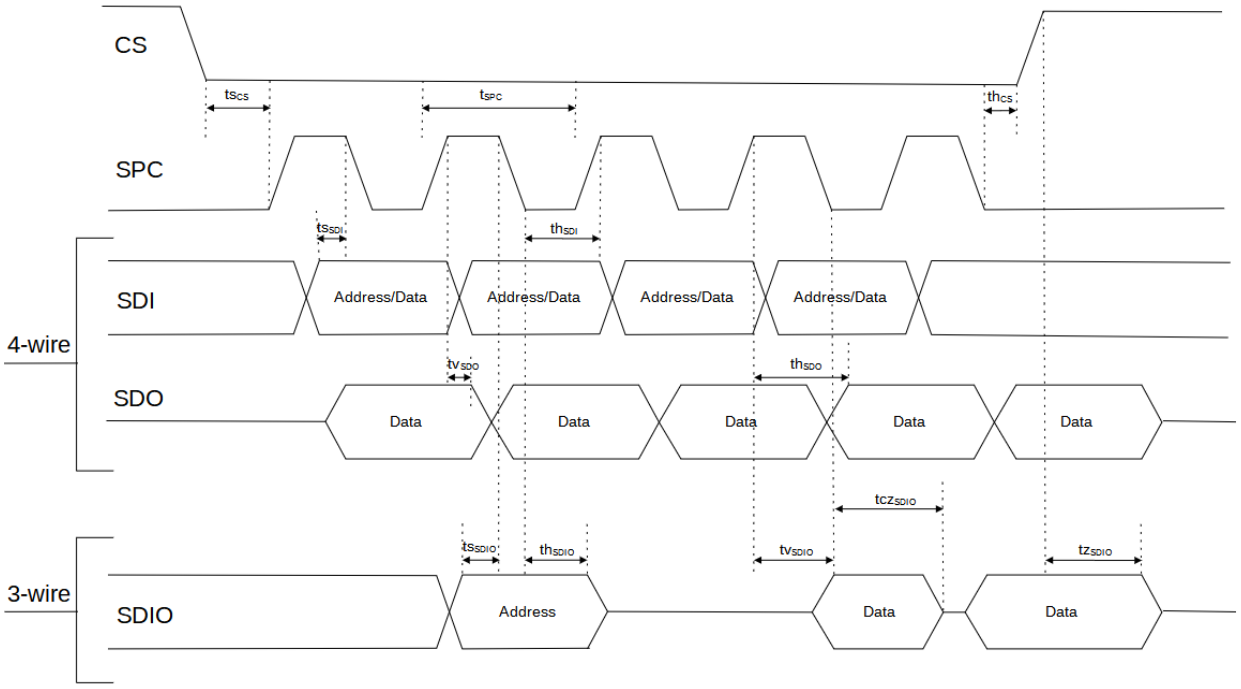
14.2.4 SPI Timing Characteristics

The typical operating conditions for the SPI interface are provided in Table 43. Please refer to Table 6 for the V_{IL}, V_{IH}, V_{OL}, V_{OH} definition to define the rising and falling edge condition of the timing symbols.

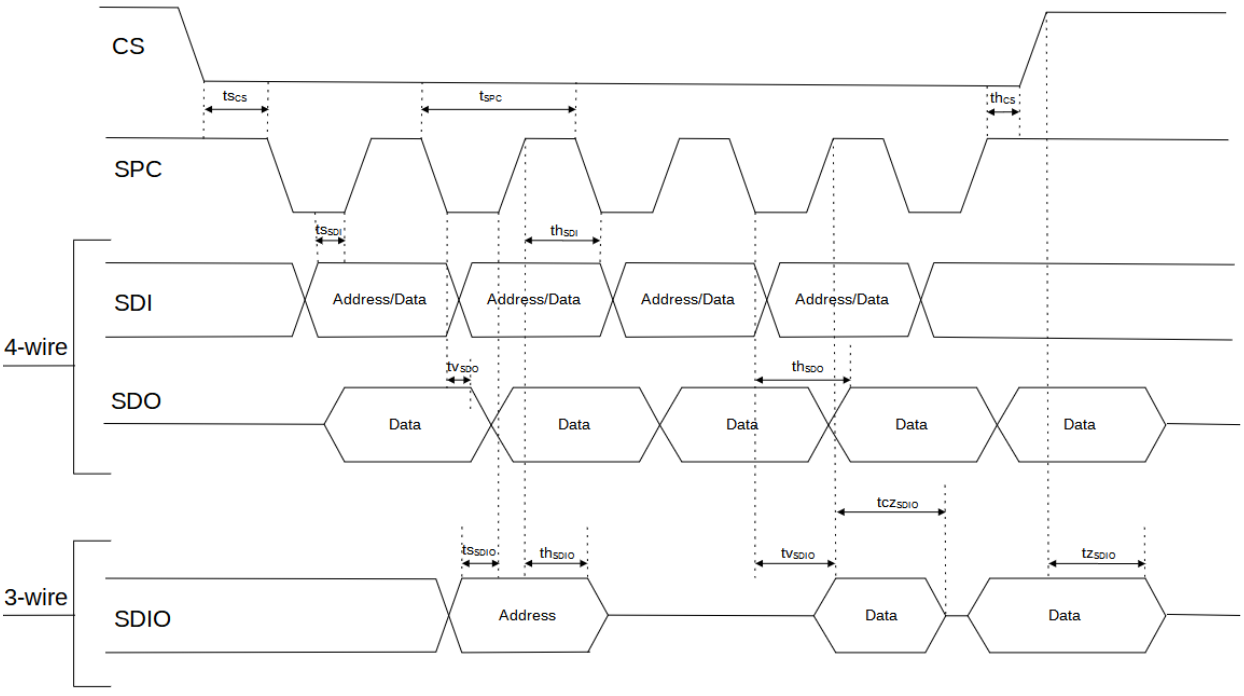
VDDIO = 1.8 V, T = 25°C unless otherwise noted.

Table 43. SPI Interface Timing Characteristics

Symbol	Parameter	Min.	Max.	Unit
t_{SPC}	SPI Clock Cycle	66.6		ns
f_{SPC}	SPI Clock Frequency		15	MHz
ts_{CS}	CS Setup Time	6		ns
th_{CS}	CS Hold Time	8		ns
ts_{SDI}	SDI Input Setup Time	5		ns
th_{SDI}	SDI Input Hold Time	15		ns
tv_{SDO}	SDO Time for Valid Output		50	ns
th_{SDO}	SDO Hold Time for Output	9		ns
td_{SDO}	SDO Disable Time for Output		50	ns
ts_{SDIO}	SDIO Address Setup Time	5		ns
th_{SDIO}	SDIO Address Hold Time	15		ns
tv_{SDIO}	SDIO Time for Valid Data		50	ns
tcz_{SDIO}	SDIO Time from SPC to High Z		50	ns
tz_{SDIO}	SDIO Time from CS to High Z		50	ns



SPI: Mode 0



SPI: Mode 3

Figure 33. Timing Characteristics for SPI 3- and 4-Wire Interfaces

14.3 I²C Interface

14.3.1 I²C Slave Address Selection

During the slave device selection phase, the I²C master supplies the 7-bit I²C slave device address to enable the QMI8A01. When SA0 is pulled down externally, the 7-bit device address becomes 0x6B (0b1101011). The 7-bit device address for the QMI8A01 is 0x6A (0b1101010) if SA0 is pulled up or left unconnected. Note that internally there is a weak pull-up of 200KΩ, and this pull-up resistor will be automatically disabled after the detection of I2C slave address during the Reset Process, refer to *7.4*.

14.3.2 I²C Interface Characteristics

Table 44 provides the I²C interface timing characteristics while Figure 34 and Figure 35 illustrate the I²C timing for both fast and standard modes, respectively. Please refer to Table 6 for the V_{IL}, V_{IH}, V_{OL}, V_{OH} definition to define the rising and falling edge condition of the timing symbols.

For additional technical details about the I²C standard, such as pull-up resistor sizing the user is referred to “UM10204 I²C-bus specification and user manual,” published by NXP B.V.

Table 44. I²C Timing Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
f_{SCL}	SCL Clock Frequency		0		400	KHz
t_{BUF}	Bus-Free Time between STOP and START Conditions		1300			ns
$t_{HD;STA}$	START or Repeated START Hold Time		600			ns
t_{LOW}	SCL LOW Period		1300			ns
t_{HIGH}	SCL HIGH Period		600			ns
$t_{SU;STA}$	Repeated START Setup Time		600			ns
$t_{SU;DAT}$	Data Setup Time		100			ns
$t_{HD;DAT}$	Data Hold Time	Standard Mode	0		3450	ns
		Fast Mode	0		900	
t_{RCL}, t_R	SCL Rise Time	Standard Mode			1000	ns
		Fast Mode	20 + 0.1 * C_B[17]		300	
t_{FCL}	SCL Fall Time	Standard Mode			300	ns
		Fast Mode	20 + 0.1 * C_B[17]		300	
t_{RDA}, t_{RCL1}	SDA Rise Time. Rise Time of SCL after a Repeated START Condition and after ACK Bit	Standard Mode			1000	ns
		Fast Mode	20 + 0.1 * C_B[17]		300	
t_{FDA}	SDA Fall Time	Standard Mode			300	ns
		Fast Mode	20 + 0.1 * C_B[17]		300	
$t_{SU;STO}$	Stop Condition Setup Time		600			ns

Note:

17. C_B is the bus capacitance.

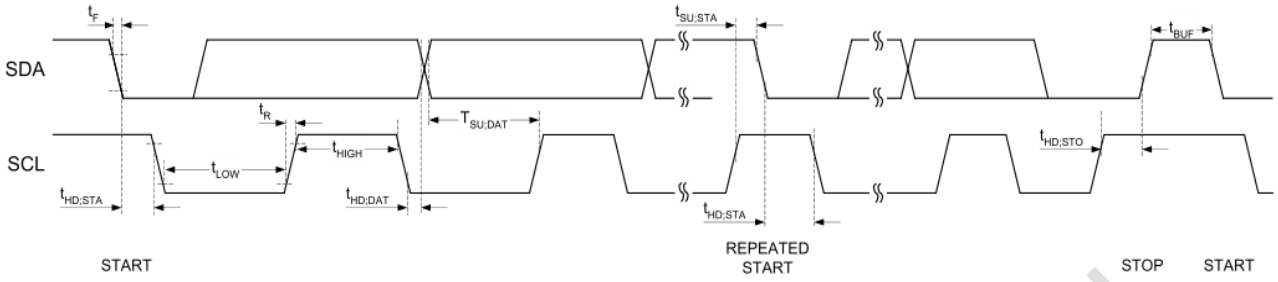
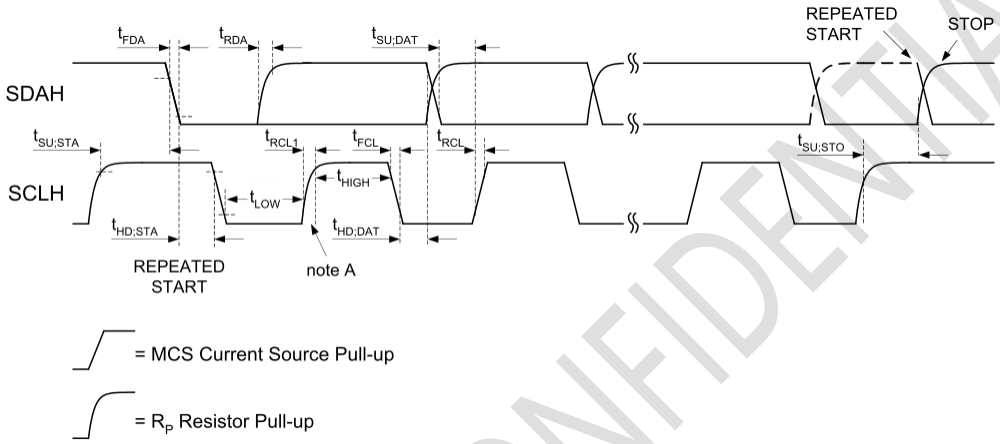


Figure 34. I2C Standard Mode Interface Timing



Note A: First rising edge of SCLH after Repeated Start and after each ACK bit.

Figure 35. I2C Fast Mode Interface Timing

14.4 I³C Interface

The QMI8A01 is compliant with the MIPI Alliance Basic Specification for I3C, version 1.0.

Note that HDR, IBI are not supported by QMI8A01.

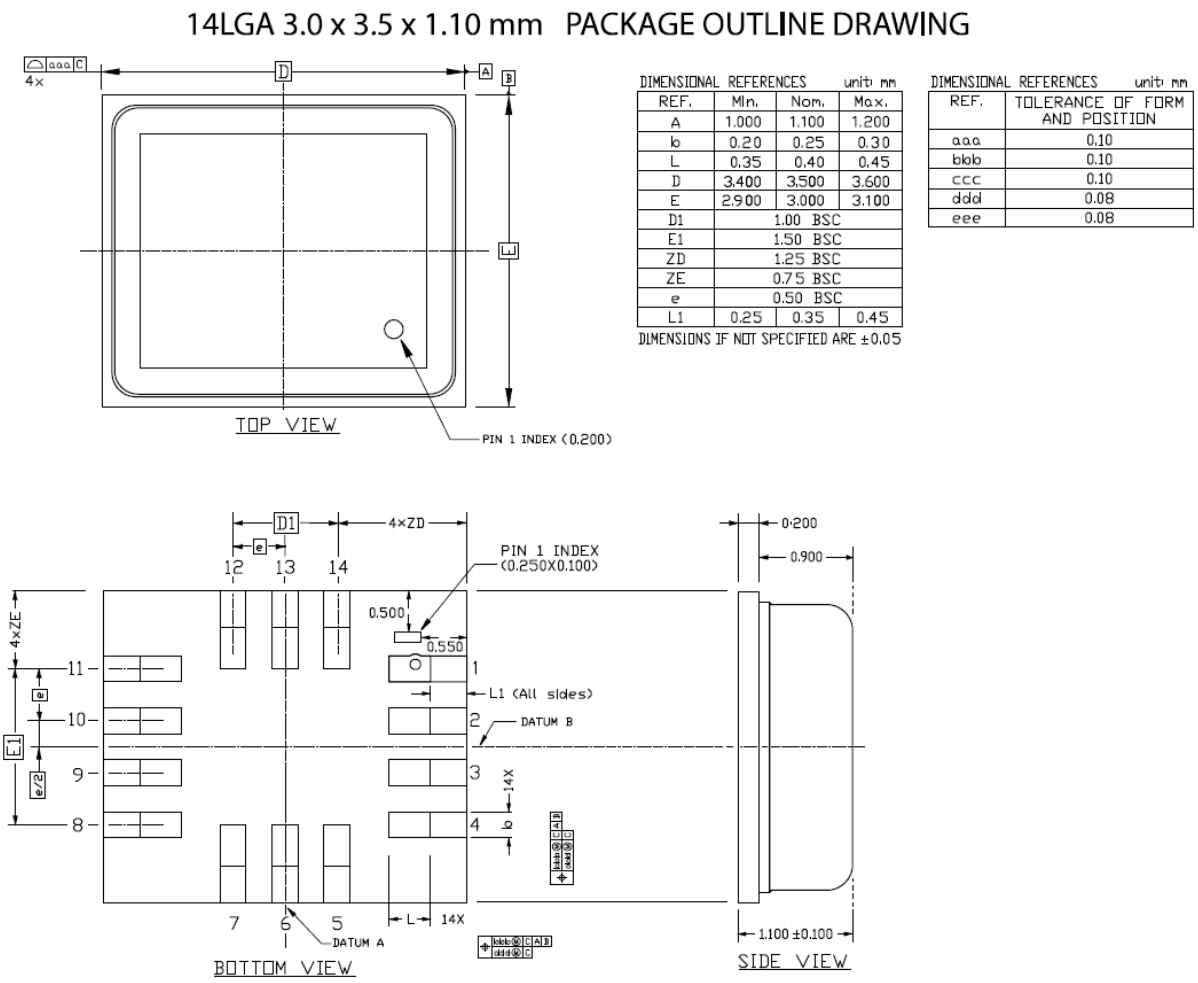
The PID of 8A01 is 0x086E00051000.

MIPI Manufacturer ID(VID) for QST Corp. is 0x0437.

The static address of I³C follows the descriptions in I²C, refer to *14.3.1*.

15 Package and Handling

15.1 Package Drawing



DIMENSIONAL REFERENCES unit: mm

REF.	Min.	Nom.	Max.
A	1.000	1.100	1.200
b	0.20	0.25	0.30
L	0.35	0.40	0.45
D	3.400	3.500	3.600
E	2.900	3.000	3.100
D1	1.00 BSC		
E1	1.50 BSC		
ZD	1.25 BSC		
ZE	0.75 BSC		
e	0.50 BSC		
L1	0.25	0.35	0.45

DIMENSIONS IF NOT SPECIFIED ARE ±0.05

DIMENSIONAL REFERENCES unit: mm

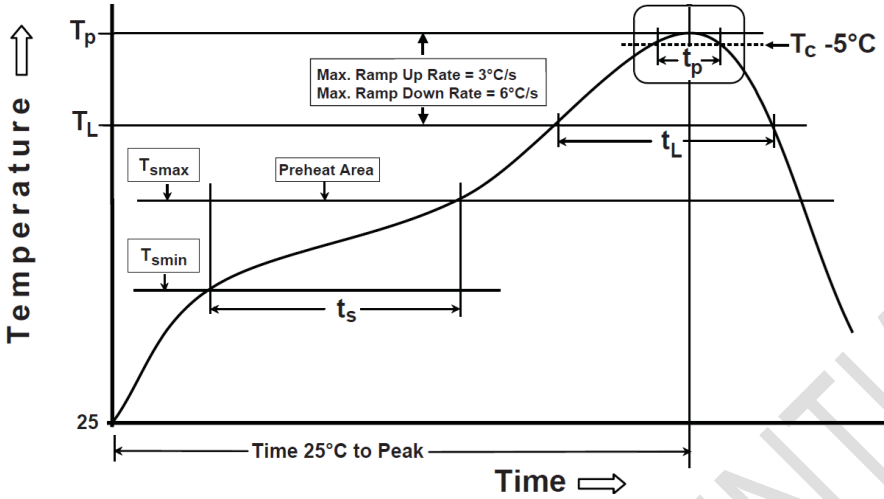
REF.	TOLERANCE OF FORM AND POSITION
aaa	0.10
bbb	0.10
ccc	0.10
ddd	0.08
eee	0.08

Figure 36. 14 Pin LGA 3.0 x 3.5 x 1.1 mm Package(错误!未找到引用源。)

Note:

18. The vent hole on pin 1 is designed to reduce the internal air pressure during SMT soldering to retain the metal cap. In production solder reflow, the solder paste cools and seals the vent hole. When reworking the PCB, don't heat the IMU from the top. This can result in the IMU sucking solder from pin1 into the cavity, damaging the IMU.

15.2 Reflow Specification



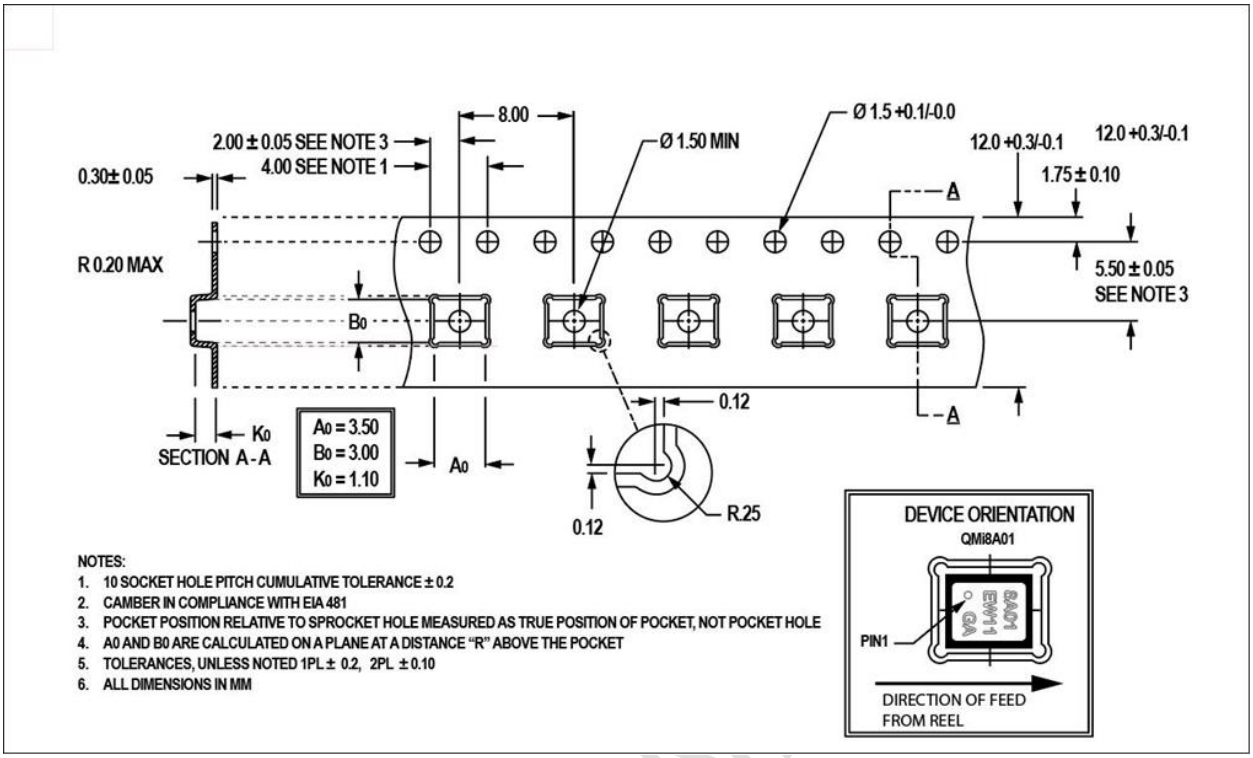
Note:

18. Figure from JEDEC J-STD-020

Profile Feature	Pb-Free Assembly Profile
Temperature Min. (T_{smin})	150°C
Temperature Max. (T_{smax})	200°C
Time (t_S) from (T_{smin} to T_{smax})	60-120 seconds
Ramp-up Rate (T_L to T_P)	3°C/second max.
Liquidous Temperature (T_L)	217°C
Time (t_L) Maintained above (T_L)	60-150 seconds
Peak Body Package Temperature (T_P)	260°C +0°C / -5°C
Time (t_P) within 5°C of 260°C	30 seconds
Ramp-down Rate (T_P to T_L)	6°C/second max.
Time 25°C to Peak Temperature	8 minutes max.

Figure 37. Reflow Profile

15.3 Carrier tape information



15.4 Storage Specifications

QMI8A01 storage specification conforms to IPC/JEDEC J-STD-020D.01 Moisture Sensitivity Level (MSL) 3.
Floor life after opening the moisture-sealed bag is 168 hours with storage conditions: Temperature: ambient to ≤30°C and Relative Humidity: 60%RH.