**Accelerometer for RS-422-Interface**

# M-A552AR1x

# Data Sheet

**(P/N: X2F000031000300)**

# Notice of the Document

# Table of Contents

# 1. GENERAL DESCRIPTION

The M-A552 is a three axis digital output accelerometer featuring ultra-low noise, high stability, and low power consumption using fine processing technology of Quartz. Incorporating both high accuracy and durability, the versatile M-A552 is well suited to a wide-range of challenging applications such as SHM, seismic observation, condition monitoring for industrial equipment, and pose detection for industrial machinery (i.e. construction machinery/attachments, agricultural machinery/ implements, robots).
The M-A552 is packaged in a water-proof and dust-proof metallic case supporting RS422 interface. This ruggedized unit is suitable for industrial use that requires remote mounting, or long-distance wiring.

## 1.1. FEATURES

Table 1-1 Features

| Item | | Specification | Note |
|---|---|---|---|
| Sensor | | | |
| | Integrated sensor | Model: M-A552AR1x<br>Internal sensor M-A352<br><br>Output range ±15G<br>Initial Bias error ±4mG(@25℃)<br>Resolution 0.06uG<br>Low noise 0.5μg/√Hz typ.<br>Selectable output format: Acceleration / Tilt Angle<br>Programmable low-pass digital filters | |
| Interface | | | |
| | Protocol (DL layer) | RS-422 (TX/RX Pair, Full-Duplex transmission) | |
| | Bit rate | Maximum 460.8kbps (programmable) | 230.4kbps(default) |
| | Cable Length | 250m (max) | |
| Others | | | |
| | Trigger function | Internal timer event trigger function | External trigger not available |
| | Terminator | Included (120Ωtyp) | |
| General specification | | | |
| | Voltage supply | 9 ~ 32V | |
| | Power consumption | 40mA typ. (Vin=12V) | |
| | Operating temperature range | -30 ~ +70°C | |
| External dimension | | | |
| | Outer packaging | Overall metallic shield case | |
| | Size | 65×60×30 mm (Including projection) | |
| | Weight | 128g | |
| | Interface connecter | M12, 8pin-male, water-proof | |
| | Water-proof, Dust-proof | IP67 | |
| Regulation | | | |
| | EU | CE marking (EN61326、RoHS Directive) | Class A |
| | FCC | FCCpart15B | Class A |

## 1.2. APPLICATIONS

- Structural health monitoring
- Seismic measurements
- Vibration control and stabilization
- Motion analysis and control

## 1.3. BLOCK DIAGRAM

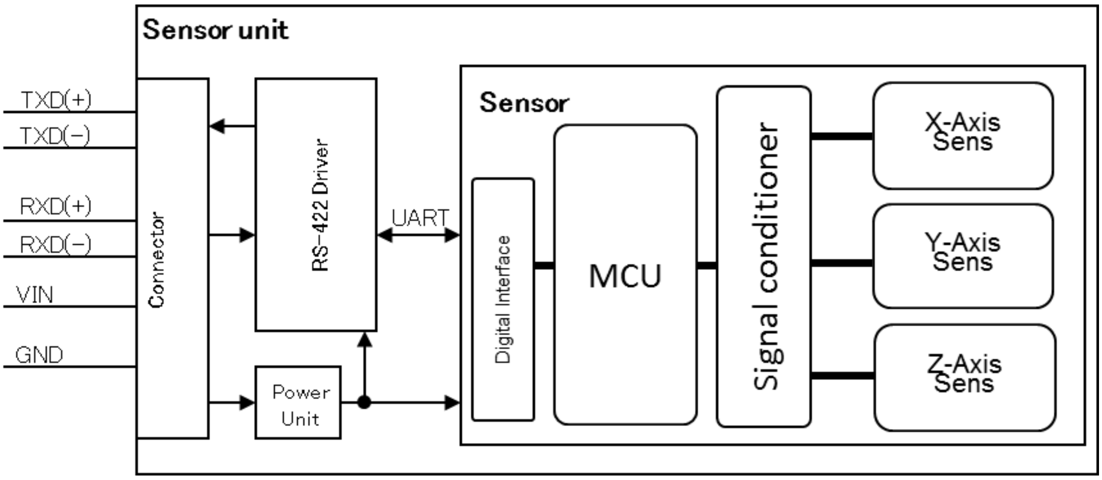



Figure 1-1 Block Diagram

# 2. SPECIFICATIONS

## 2.1. ABSOLUTE MAXIMUM RATINGS

Table 2-1 Absolute Maximum Ratings

| Parameter | Min | Typ. | Max | Unit |
|---|---|---|---|---|
| VIN to GND | −0.3 | | +32 | V |
| Voltage on any pin to GND | −32 | | +32 | V |
| Storage Temperature Range | −40 | | 85 | °C |
| Operating Temperature Range | -30 | | 70 | °C |
| Shock Resistance （Half-sine 1msec） *1 | | | 1,000 | G |

If the unit is operated beyond the absolute maximum rating, malfunction may occur, or the unit may fail completely. Although the unit may appear to operate normally, reliability may decrease.

**CAUTION:**

***1 Excessive vibration or shock independent of the above listed conditions may increase the possibility of malfunction or causing a failure!**

## 2.2. RECOMMENDED OPERATING CONDITION

Table 2-2 Recommended Operating Conditions

| Parameter | Term | Condition | Min. | Typ | Max. | Unit |
|---|---|---|---|---|---|---|
| Power supply voltage | $V_{IN}$ | $V_{IN}$ to GND *2 | 9 *1 | 12 | 24 | V |
| Port input voltage | $V_{PORT}$ | RD+/RD- to GND | | 5 | | V |
| Operating temperature | $T_{OPE}$ | | -20 | - | 70 | °C |

*1. *When power supply voltage is 9V or less, the master may not be able to communicate with this node normally even if the LED turns on.

*2. The power supply voltage must reach the recommended operating condition within 2 seconds after power is applied to this node.

## 2.3. PERFORMANCE & ELECTRICAL SPECIFICATIONS

Table 2-3 Sensor Specifications (Common Items)

VIN=12V, $T_A$=--30℃ to +70℃,±1G unless otherwise noted

| Parameter | Test Conditions / Comments | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| **MISALIGNMENT** | | | | | |
| Case to Axis | | | | ±0.5 | Deg |
| Axis to Axis | 1 σ, Axis-to-axis, Δ = 90° ideal,RT | | | ±0.1 | Deg |
| Cross Axis Sensitivity | | | ±0.2 | | % |
| **ACCELERATION** *1 | | | | | |
| Sensitivity | | | | | |
| Output Range | | | | ±15 | G |
| Scale Factor | | | 0.06 | | µG/LSB |
| Sensitivity Error | 25℃, ≤ 1G | | ±500 | | ppm |
| Nonlinearity | ≤ 1G, Best fit straight line, RT | | | ±0.03 | % of FS |
| Bias | | | | | |
| Initial Error | 1 σ, −30°C ≤ $T_A$ ≤ +70°C | | | ±4 | mG |
| Bias Repeatability | TA=25℃, VIN=12V<br>For 1 year after shipment | | 3 | | mG |
| Bias Temperature Error | 25℃ | | | ±2 | mG |
| Temperature sensitivity | | | ±0.1 | | mG/℃ |
| Bias Instability | AVR, Average | | 0.2 | | uG |
| Velocity Random Walk | Average | | 1.2E-4 | | (m/sec)/√hr |
| Noise | | | | | |
| Noise Density | $T_A$=25°C, average 0.5Hz to 6Hz, | | 0.5 | 2 | µG/√Hz |
| | $T_A$=25°C, peak 0.5Hz to 100Hz | | | 60 | µG/√Hz |
| Cantilever Resonance frequency | 25℃, VIN=12V | | 850 | | Hz |
| VRC | at 50Hz<br>25℃, VIN=12V | | | ±50 | µG/G$^2$ |
| Frequency Property | | | | | |
| -6 dB Bandwidth | User selectable | 9 | | 460 | Hz |
| **TILT ANGLE** *2 | | | | | |
| Sensitivity | | | | | |
| Output Range | | | | ±1.0472 (±60) | rad (deg) |
| Scale Factor | | | 0.002 | | µrad/LSB |
| Nonlinearity | 25℃, ±45deg | | | ±0.03 | % of FS |
| Misalignment | 1 σ, Axis-to-axis, Δ = 90° ideal | | | ±1.745 (±0.1) | mrad (deg) |
| Bias | | | | | |
| Bias Repeatability | TA=25℃, VIN=12V<br>For 1 year after shipment | | ±3 (±0.17) | | mrad (deg) |
| Bias Temperature Error | 25℃ | | | ±2 (±0.11) | mrad (deg) |
| Noise | | | | | |
| Noise Density | $T_A$=25°C, average 0.5Hz to 6Hz, | | 0.5 | 2 | µrad/√Hz, rms |
| **TEMPERATURE SENSOR** | | | | | |
| Output Range | | -30 | | 85 | ℃ |
| Scale Factor *3 | at 25℃<br>T[℃]=SF*a+34.987 | | -0.0037918 | | °C/LSB |

*1. The calibrated standard 1G gravitational acceleration value is 9.80665 m/s$^2$.

*2. The tilt angle is internally calculated from gravitational acceleration by the following expression.
Tilt Angle Calculation Formula

$$\theta = \mathrm{asin(G)}\,[rad]$$

*3. This is a reference value used for the internal temperature correction, and is not guaranteed to accurately output the interior temperature.

Table 2-4 Interface Specification

$T_A$=25°C, VIN=12V, unless otherwise noted

| Parameter | Test Conditions | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| Driver | | | | | |
| Differential Output Voltage | RL=120Ω,TD- to TD+ | 2 | 2.5 | | V |
| | RL=54Ω, TD- to TD+ | 1.5 | 2 | | V |
| Common Mode Output Voltage | RL=120Ω | 1 | 2.5 | 3 | V |
| Output Resistance | | | 120 | | Ω |
| Rise or Fall Time | RL=120Ω | | | 400 | ns |
| Receiver | | | | | |
| Differential Input voltage | | -25 | | 25 | V |
| Input Resistance | | | 120 | | Ω |
| FUNCTIONAL TIMES (Time until data is available) | | | | | |
| Power-On Start-Up Time *1 | | | | 900 | ms |
| Reset Recovery Time *1 | | | | 970 | ms |
| Flash Backup Time | | | | 310 | ms |
| Flash Reset Time | | | | 1900 | ms |
| Self Test Time | ACC Test / TEMP Test / VDD Test | | | 200 | ms |
| | Sensitivity Test | | 10 | 40 | s |
| | Flash Test | | | 5 | ms |
| Filter Setting Time | Built-In FIR Filter | | | 4 | ms |
| | User Defined FIR Filter | | | 100 | ms |
| User Filter Write Cycle, tUWC | | | | 7 | ms |
| User Filter Read Cycle, tURC | | | | 500 | us |
| OUTPUT DATA RATE *2 | | 50 | | 1,000 | Sps |
| Clock Accuracy | | | | ±0.001 | % |

*1. Do not access the device during startup or reinitialization.
*2. Data rate and optimum filter characteristics can be changed by command.

Table 2-5 Current Consumption

Ta=25°C,RL=120Ω,unless otherwise specified; all voltages are defined with respect to ground; positive currents flow into the sensor unit.;

| Parameter | Term | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Standby current | $I_{IN(ready)}$ | Vin=12V | - | 34.5 | - | mA |
| | | Vin=24V | - | 18.5 | - | mA |
| Operating current | | Vin=12V , 512Tap 460.8kbps, 1000sps | - | 38 | - | mA |
| Maximum input current | $I_{IN(max)}$ | | - | - | 60 | mA |

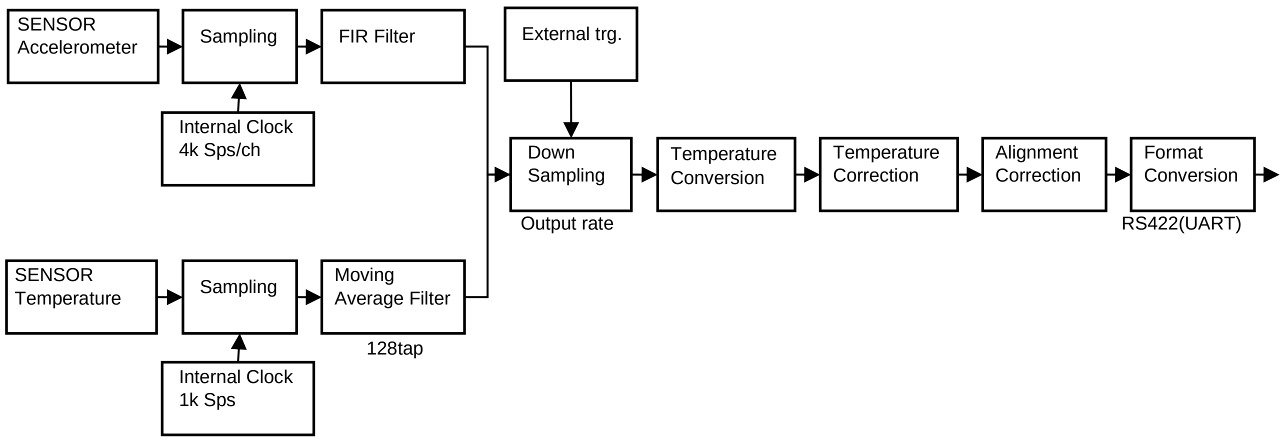



Figure 2-1 Functional Block Diagram

# 3. MECHANICAL DIMENSIONS

## 3.1. OUTLINE DIMENSIONS

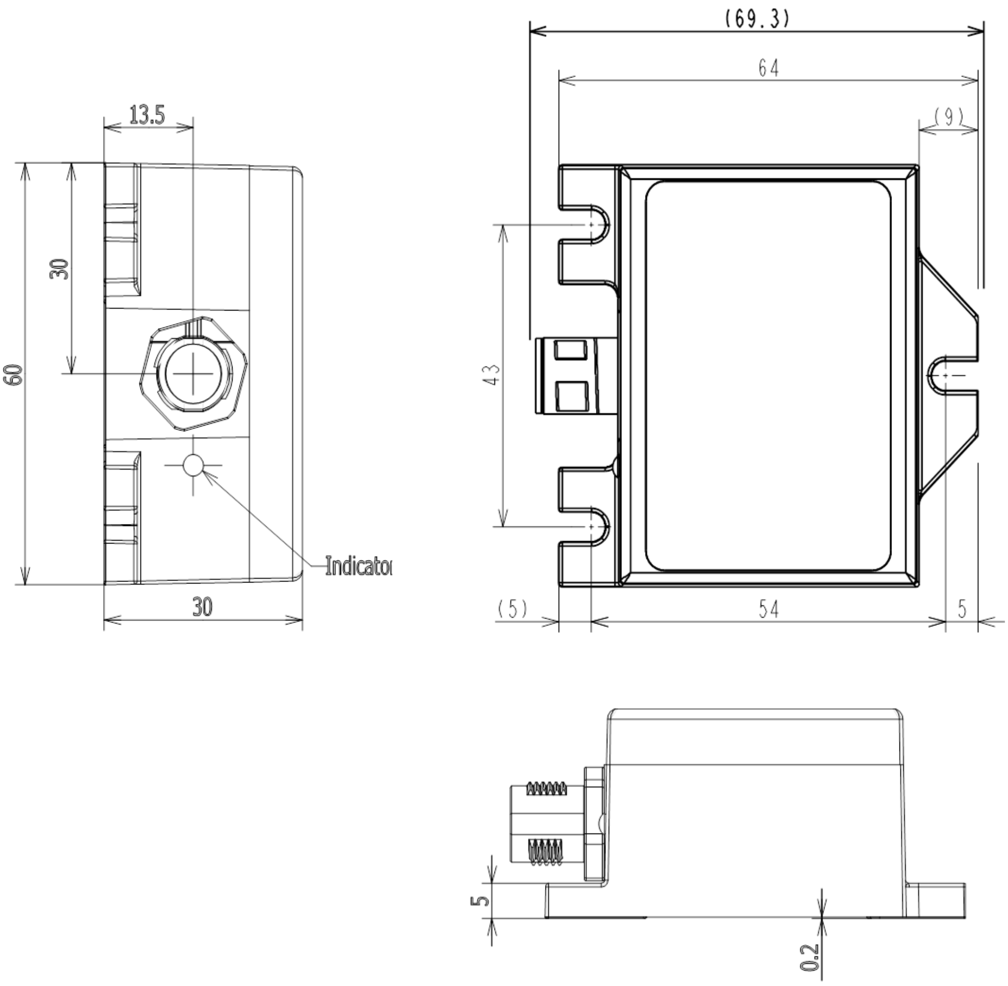



Figure 3-1 Outline Dimensions (millimeters)

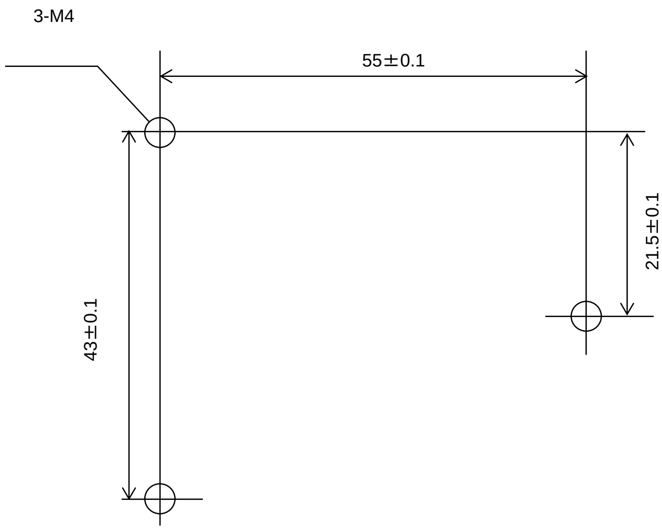


Figure 3-2 Recommended Mounting Dimension

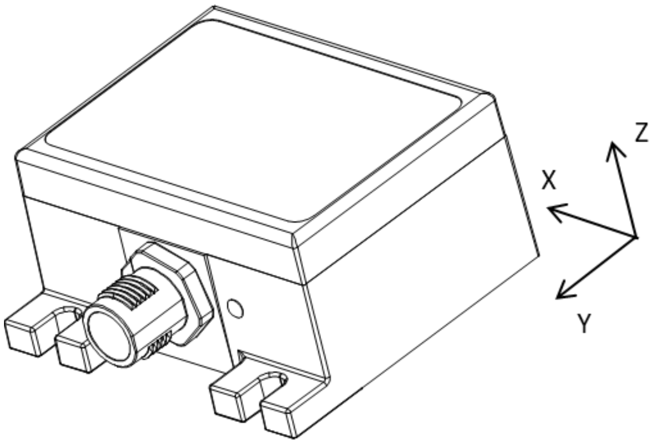



Figure 3-3 Axial direction

## 3.2. CONNECTOR SPECIFICATIONS

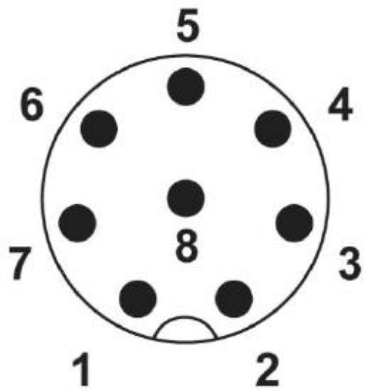



Figure 3-4 Connector Pin Layout

Table 3-1 Pin Function Description

| Pin No. | Mnemonic | Type*1 | Description |
|---|---|---|---|
| 1 | NC | N/A | Do Not Connect |
| 2 | VIN | S | Power Supply (9-32V) |
| 3 | GND | S | 0V |
| 4 | TD- | O | Transmit Data (-) |
| 5 | RD+ | I | Received Data (+) |
| 6 | TD+ | O | Transmit Data (+) |
| 7 | NC | N/A | Do Not Connect |
| 8 | RD- | I | Received Data (-) |

*1) Pin Type I :Input, O :Output, I/O :Input/Output, S :Supply, N/A :Not Applicable

Note: Please use an M12-8 pin mating female connector that corresponds to IP67 specifications. Table 3-2 describes the connector manufacturer and the model number which is used in this product.

Table 3-2 Connector Part Number

| Manufacturer | Part Number | RoHS Compliant |
|---|---|---|
| PHOENIX CONTACT | SACC-DSI-MS-8CON-M12-SCO SH | Yes |

# 4. TYPICAL PERFORMANCE CHARACTERISTICS

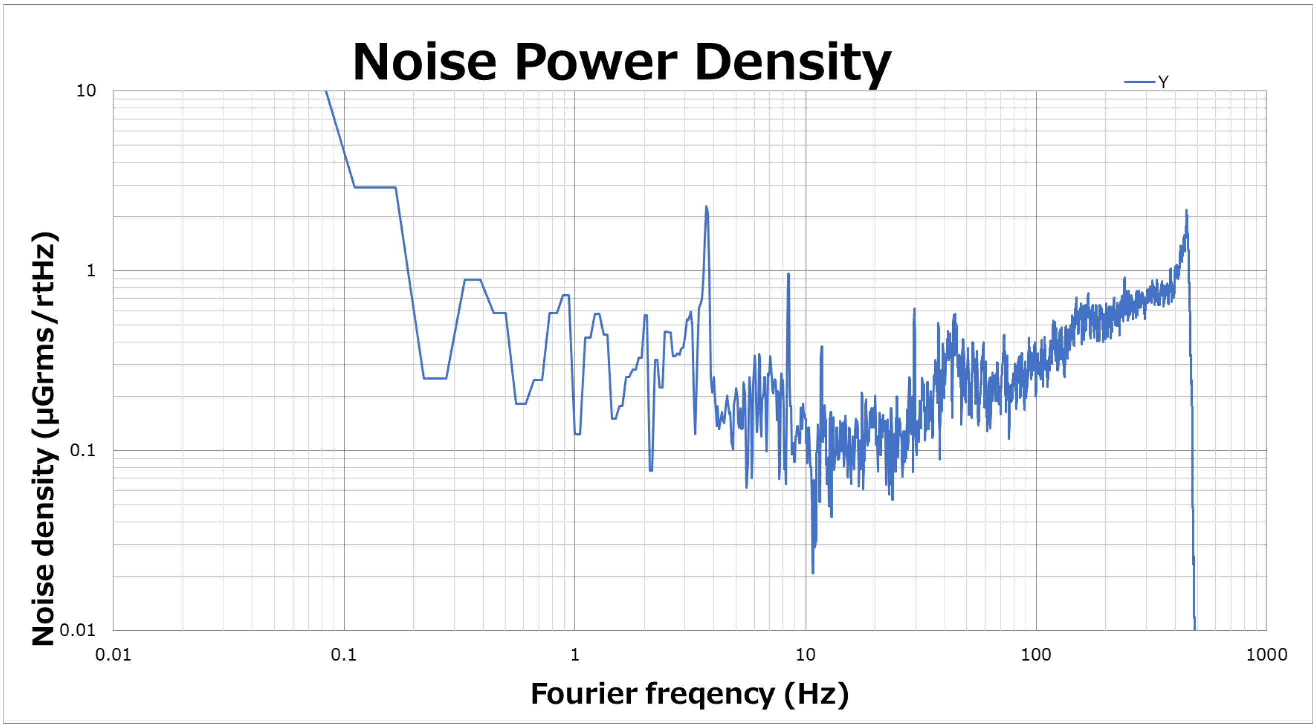



Figure 4-1 Noise Density Characteristics

The above graph is a typical example of the product characteristics, and is not guaranteed by the specification.

# 5. CONNECTION EXAMPLE

## 5.1. CONNECTION TO HOST

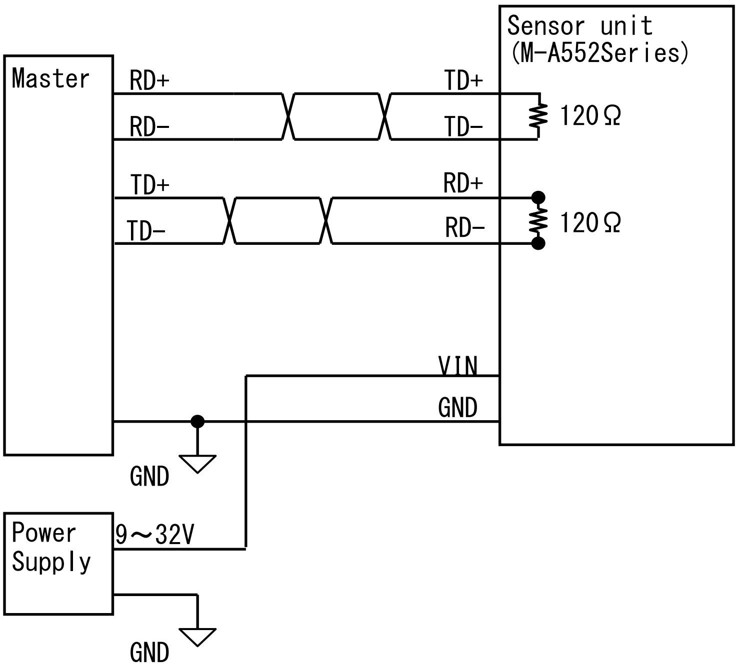



Figure 5-1 Connection Example

## 5.2. PRECAUTION FOR WIRING AND CABLING

- This product has internal terminator on the receiver port (RD).
- It is recommended that twisted pair cable with shielding should be used. Each signal pair should be connected to each cable pair. (ex: RD+ and RD-)
- It is recommended that shield connects to ground (at the host) when a cable with shield is used.
- Maximum recommended cable length is 250 meters as a guideline. However, even if the cable length is within the guidelines, the communication may be unstable or unusable depending on system environment. The cabling should be evaluated in the target system environment to confirm proper operation. (Ref: TIA-EIA-422-B ANNEX A)

## 5.3. PRECAUTION FOR SUPPLYING POWER

The user should be aware of serious risks on the power supply exposure to the following:

- High voltage noise by increased resistance and inductance on power supply line.
- Surge voltage from lightning and environmental equipment.

Figure 1-1Figure 5-2 describes the external reference protection circuit against the lightning surge with a surge level based on IEC61000-4-5, +/-1kV(power supply line to the power supply ground) and +/-2kV(power supply line to the earth).

| | |
|---|---|
| VP: | CAN_V+ (Power supply) |
| PGND: | CAN_GND (Power supply ground) |
| FGND: | EARTH (System ground earth) |
| U3039: | Surge absorber to power supply (Okaya Electric Industries) |
| ERZ-V14D390: | Surge absorber to earth ground (Panasonic) |

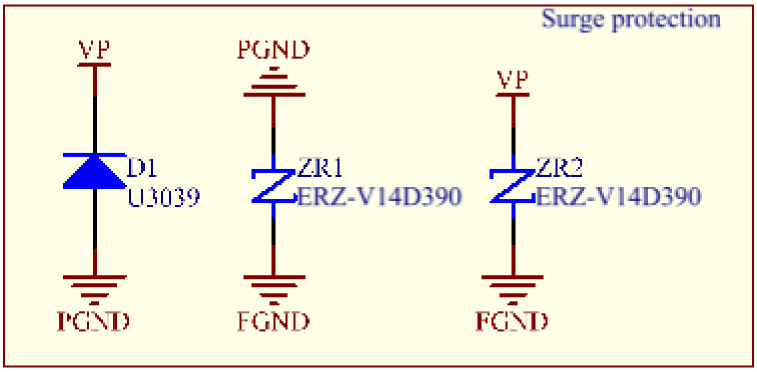



Figure 5-2 Surge Protection Circuit

## 5.4. OPERATION MODE

The following three operational modes are available in the device.

1) Configuration mode
2) Sampling mode

| Sampling condition | Measurement condition |
|---|---|
| ･Manual sampling | ･Standard noise floor |
| ･Auto sampling | ･Reduced noise floor |

In the Sampling mode, the device can operate with a standard noise floor condition or a reduced noise floor condition. When Auto sampling is active, all sensor outputs are sent automatically at the programmed output data rate without the request from the Host. These conditions can be switched between Manual sampling and Auto sampling by **UART_AUTO** (*3) (UART_CTRL[0x08(W1)] bit[0]), and between Standard noise floor and Reduced noise floor by **MESMOD_SEL** (see Figure 5.3).

Immediately after a hardware reset or power-on, internal initialization starts. During the internal initialization, all the register values and states of external pins are undefined. After the internal initialization is completed, the device goes into Configuration mode automatically, except when AUTO_START and UART_AUTO sampling are both enabled (the device then goes into Sampling mode automatically). To change the operation mode, write to **MODE_CMD** (MODE_CTRL[0x02(W0)] bit[9:8]) (*1) and make various changes to the sensor setting in Configuration mode (*2). After configuration is completed, go to Sampling mode to read out the temperature and acceleration data.
By executing software reset (Register: GLOB_CMD [0x0A (W1)], write 1 to SOFT_RST in bit [7]), internal initialization operation is executed regardless of the current operation mode and the system enters Configuration mode.

*1. The following explains register notation used in this document.
For example, MODE_CTRL[0x02(W0)] bit[9:8] refers to:
- MODE_CTRL: Register Name
- [0x02(W0)]: First number is the Register Address, (W0) refers to Window Number "0"
- bit[9:8]: Bits from 9 to 8

*2. Make sure that the device is in Configuration mode when you write to the registers to configure operational settings. In Sampling mode, writing to registers is ignored **except** the following cases.

- Writing to **MODE_CMD** (MODE_CTRL[0x02(W0)] bit[9:8])
- Writing to **SOFT_RST** (GLOB_CMD[0x0A(W1)] bit[7])
- Writing to **WINDOW_ID** (WIN_CTRL[0x7E(W0/W1)] bit[7:0])

*3. While the device is with UART Auto sampling and sensor sampling is active, register read access is not supported. Otherwise, the sampling data transmitted with the UART Auto sampling will be corrupted by the response data from the register read.

## 5.5. STATE TRANSITION

Immediately after power-on, internal initialization starts and then the device will be in configuration mode. However, if AUTO_START and UART_AUTO are both enabled, then the device automatically enters sampling mode after internal initialization. Various operation modes are executed by sending various commands from configuration mode. Table 5.3 shows the state transition diagram.

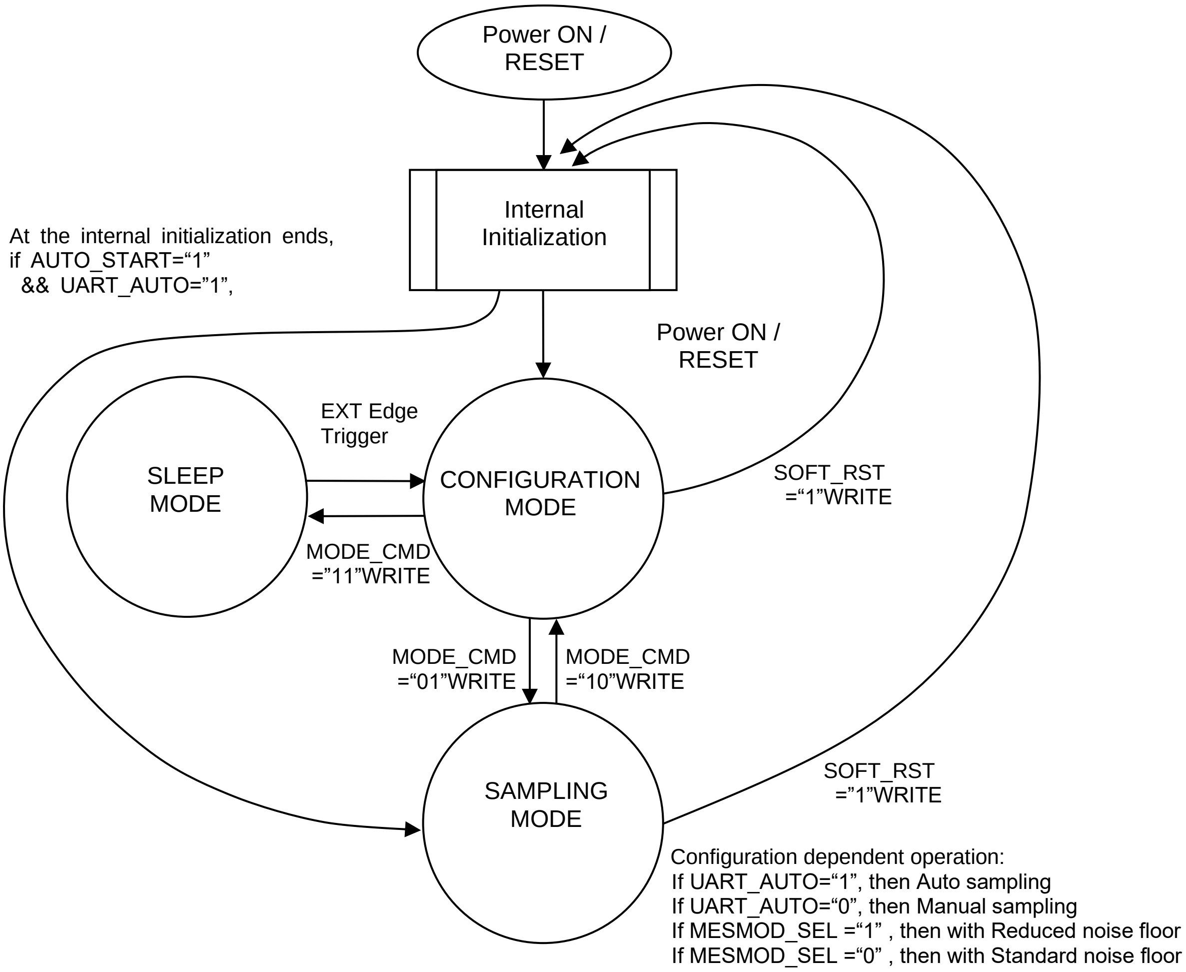



Figure 5-3 Operational State Diagram

## 5.6. Functional Block

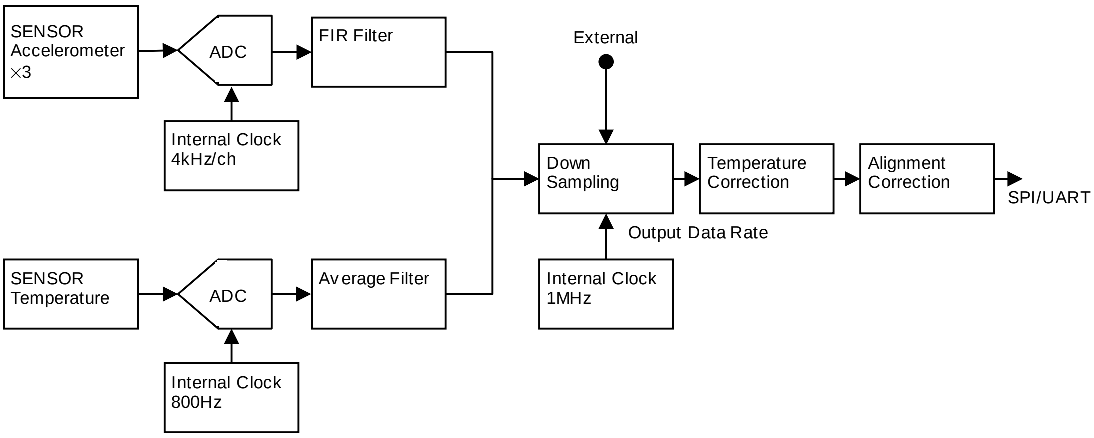



Figure 5-4 Functional Block Diagram

## 5.7. Sampling Counter

By reading COUNT[0x0A(W0)] register, the counter value, which is incremented based on the sampling completion timing of the internal A/D converter is returned. The count interval is 250usec/count and is based on the precision of the internal reference oscillator (crystal).

Additionally, during UART burst read or with UART Auto sampling, the counter value can be included in the response format by setting the **COUNT_OUT** (BURST_CTRL[0x0C(W1)] bit[1]). For information about the response format, see 6.3 DATA PACKET FORMAT.

## 5.8. Self Test

This product has the following self test functions. For information about the execution time of the self test, see “Self Test Time” in Table 2.4 Interface Specifications.

- Acceleration Value

This self test function can be used to check whether the outputs of the accelerometer are within the pre-determined range and operating properly.

The test result is OK if the absolute value of the output as a three dimensional vector is within the gravitational acceleration(0.8G to 1.2G).

When performing the self test, make sure the device does not move during the test and the test is conducted in a place without vibration.

To use this function, execute **ACC_TEST** of register: MSC_CTRL[0x02(W1)] bit[10], check the **ACC_ERR_ALL** of register DIAG_STAT[0x04(W0)] bit[1] for diagnostic result.

- Acceleration Sensitivity

  This self test function can be used to determine whether the acceleration sensitivity error is within ± 2.5 %. It takes up to 40 sec (typ.10 sec) per axis for the diagnostic.

  To use this function, execute **SENS_TEST** of register: MSC_CTRL[0x02(W1)],bit[14:12], check the **SENS_ERR** of register DIAG_STAT[0x04(W0)],bit[11:10] for diagnostic result.

  Note) It may lead to a diagnostic result of “unable to be determined” or may result in an inaccurate diagnostic when there are sudden changes in vibration level during the execution or the vibration level is lower than the noise floor of this product (refer to Section 4 Typical Performance Characteristics).

- Temperature Value

  Determine whether the temperature sensor is operating properly.

  To use this function, execute **TEMP_TEST** of register: MSC_CTRL [0x02 (W1)], bit [9], check the **TEMP_ERR** of register: DIAG_STAT [0x04 (W0)], bit [9] for diagnostic result.

- Power Supply Voltage Level

  Determine whether the power supply voltage of the internal sensor is within 3.0V to 3.6V.

  To use this function, execute **VDD_TEST** of register: MSC_CTRL[0x02(W1)],bit[8], check the **VDD_ERR** of register: DIAG_STAT[0x04(W0)],bit[8] for diagnostic result.

- Nonvolatile memory

  Determine whether the Nonvolatile memory is operating properly by consistency test of data in nonvolatile memory.

  To use this function, execute **FLASH_TEST** of register: MSC_CTRL[0x02(W1)],bit11], check the **FLASH_ERR** of register: DIAG_STAT[0x04(W0)],bit[2] for diagnostic result.

## 5.9. Threshold Detection of Accelerometer

When the acceleration value exceeds the preset threshold, an alarm is indicated.
The threshold can be set for each 1G step within the range of 0 to 15 G upper limit and -15 G to 0 G lower limit. At the time of shipment, the upper limit + 15 G and the lower limit -15 G are set

The alarm threshold is set in the registers: XA_ALARM [0x47 - 0x46 (W1)], YA_ALARM [0x49 - 0x48 (W1)], ZA_ALARM [0x4B – 0x4A (W1)] and the alarm indication is registered in FLAG [0x06 (W0)], displayed in ***ALARM_ERR** of bit [4: 2]. Reading ***ALARM_ERR** will reset the alarm display.

## 5.10. Checksum

A checksum can be appended to the response data during a UART Burst read or UART Auto sampling by enabling this function in **CHKSM_OUT** (BURST_CTRL [0x0C(W1)] bit 0).

The range of the data content for checksum is after the address byte (AD=0x80) of the response data (Figure 5.10). The checksum is calculated with a simple addition of the data content in units of 16-bit, and the resulting sum is truncated to 16-bits and appended as checksum just before delimiter byte (CR=0x0D).
For example:
Because the sum is "611B4" for the response data stream of "FE01 C455 4000 0052 33C0 0043 7BC8 004A 2608 FD73 3AA0 FF75 4C30 1F53 8FD0 0600 0014", the checksum is "11B4":

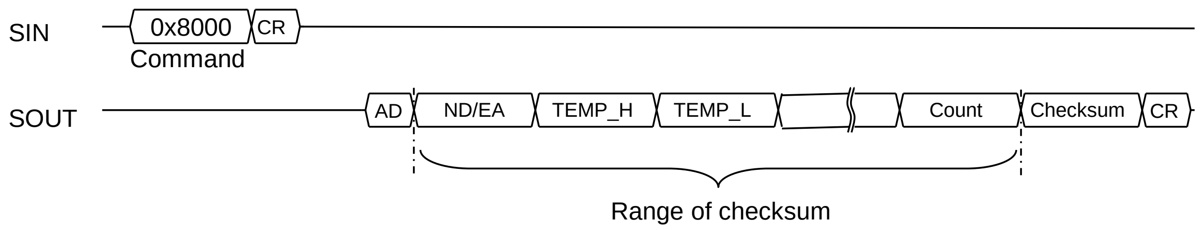



Figure 5-5 Checksum

## 5.11. Automatic Start (For UART Auto Sampling Only)

Automatic Start function is designed to be used in conjunction with the UART Auto sampling. When the power is supplied or the accelerometer is restart/reset, it allows the device to automatically enter Sampling mode after completing internal initialization. Please refer to Figure 5.3 for the state transition.

Follow the procedures below to enable the Automatic Start function:

- Write a “1” to both **UART_AUTO** (bit [0]) and **AUTO_START** (bit [1]) of UART_CTRL [0x08(W1)].
- Store the current register settings to non-volatile memory by writing a "1" to **FLASH_BACKUP** (GLOB_CMD [0x0A(W1)] bit [3]). After completion of the **FLASH_BACKUP** command, confirm the results by **FLASH_BU_ERR** (DIAG_STAT [0x04(W0)] bit [0]).
- The Accelerometer will automatically enter Sampling Mode after the power supply is cycled, or a hardware reset, or a software reset command is executed.

Follow the procedures below to disable this function.

- After entering sampling mode with automatic start, write "01" to **MODE_CMD** of register: MODE_CTRL [0x02 (W0)], bit [9: 8] and enter the configuration mode
- Write "0" to **AUTO_START** of register: UART_CTRL [0x08 (W1)], bit [1].
- The subsequent steps are the same as above. Please store the register setting to nonvolatile memory and restart or reset the accelerometer.

## 5.12. Bias Offset

This function adjusts acceleration bias of X, Y, Z axis. The user specified offset is applied to the measured acceleration value before being sent out the serial interface.

Set the bias offset value to the registers: XA_OFFSET [0x2F - 0x2C (W1)], YA_OFFSET [0x33 - 0x30 (W1)], ZA_OFFSET [0x37 - 0x34 (W1)]. The adjustment range is -15 G to +15 G. The data format is the same as the output format of register: ACCL [0x3A - 0x30 (W0)]. Both the X, Y, and Z axes are set to "0" when shipped.

## 5.13. Tilt Output / Combination Output

The device can be configured to output tilt angle by register setting. The tilt angle is calculated from the measured gravitational acceleration vector. The calculation formulas are as follows.

$$\theta = \mathrm{asin}\mathrm{G}[rad]$$

The device is configurable to select the measurement output type for each axis to be either acceleration or tilt angle. The measurement output type is selected with **OUTPUT_SEL_*** of register: SIG_CTRL [0x00 (W1)], bit [7: 5].

When both acceleration and tilt angle is outputting at the same time, set **OUTPUT_SEL** to "Tilt angle" and read register: ACCL [0x3A - 0x30 (W0)] and register: TILT [0x46 - 0x3C (W0)] in normal mode.

## 5.14. Measurement with Reduced Noise Floor Condition

The device can be configured to output data with a Reduced noise floor condition by register setting. Check the noise density characteristics shown in Figures 4.1-4.2 for standard noise floor level and reduced noise floor level.

Note) If the setting for the noise floor condition is switched from the standard condition to the reduced condition, current consumption during sampling increases.

Follow the procedures below to select a Reduced noise floor condition,

- Set **MESMOD_SEL (**SIG_CTRL[0x00(W1)], bit[4]) to "1: Enable".
- Store the current register settings to non-volatile memory by writing a "1" to **FLASH_BACKUP** (GLOB_CMD [0x0A(W1)], bit [3]). After completion of the **FLASH_BACKUP** command, confirm the results by **FLASH_BU_ERR** (DIAG_STAT [0x04(W0)], bit [0]) to be "0: No error".
- The Reduced noise floor condition will be applied after the power supply is cycled, or a hardware reset or a software reset command is executed. The status can be checked by **MESMOD _STAT** (GLOB_CMD[0x0A (W1)], bit[12]) to be "1: Reduced noise floor condition".
- Put the operation mode from configuration mode into sampling mode by writing "01" to **MODE_CMD** (MODE_CTRL[0x02(W0)], bit[9:8]) to start measurement with the Reduced noise floor condition.

Follow the procedures below to return to Standard noise floor .

- Set **MESMOD_SEL (**SIG_CTRL[0x00(W1)], bit[4]) to "0: Standard noise floor condition".
- The subsequent steps are the same as above. Please store the register setting to nonvolatile memory and restart or reset the device.

## 5.15. Bias Temperature Shock Compensation

The device is equipped with a bias stabilization function against thermal shock. The factory setting is set to "1: enable" for this function. When enabled, the time period for bias stabilization after power on is reduced, and the bias errors due to an environmental temperature change are reduced.

Note) This function when enabled may increase errors in estimation of inertial position when state estimation filters such as a Kalman filter are used for inertial navigation etc.

When data without bias temperature shock compensation is preferred, disable this function by setting **TEMP_STABIL** (SIG_CTRL[0x00(W1)], bit[2]) to "0: Disable".

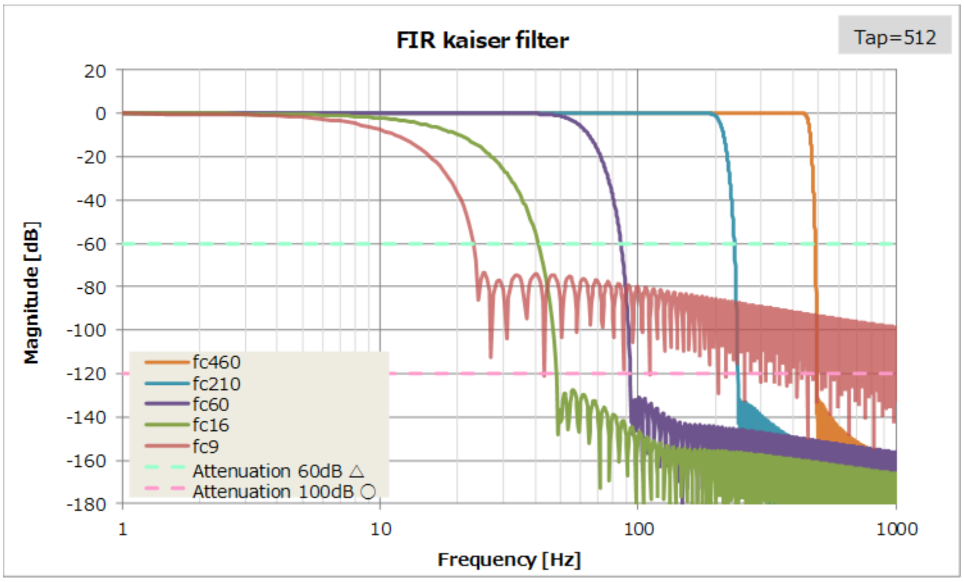
## 5.16. FILTER

The device has a programmable internal FIR filter. The intermediate sensor signal at 4k sps is processed by the FIR filter and decimated according to the output timing and sent out the serial interface. The number of TAPs and a cutoff frequency can be set with the FILTER_CTRL [0x06(W1)] register.

### 5.16.1. FIR Kaiser Filter

Filter parameters correspond to the Kaiser window parameters.
The number of TAPs can be set to 64, 128, or 512, and the cutoff frequency Fc can be selected

according to the output sample rate. Figure 5-6 to Figure 5-9 show the typical characteristic of the filters.



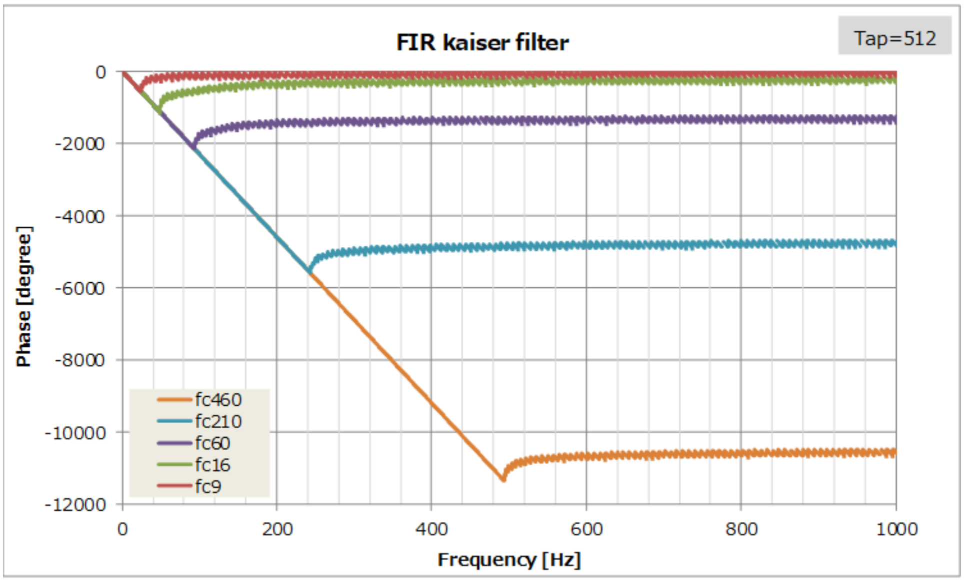




Figure 5-6 FIR Kaiser Characteristics－TAP512

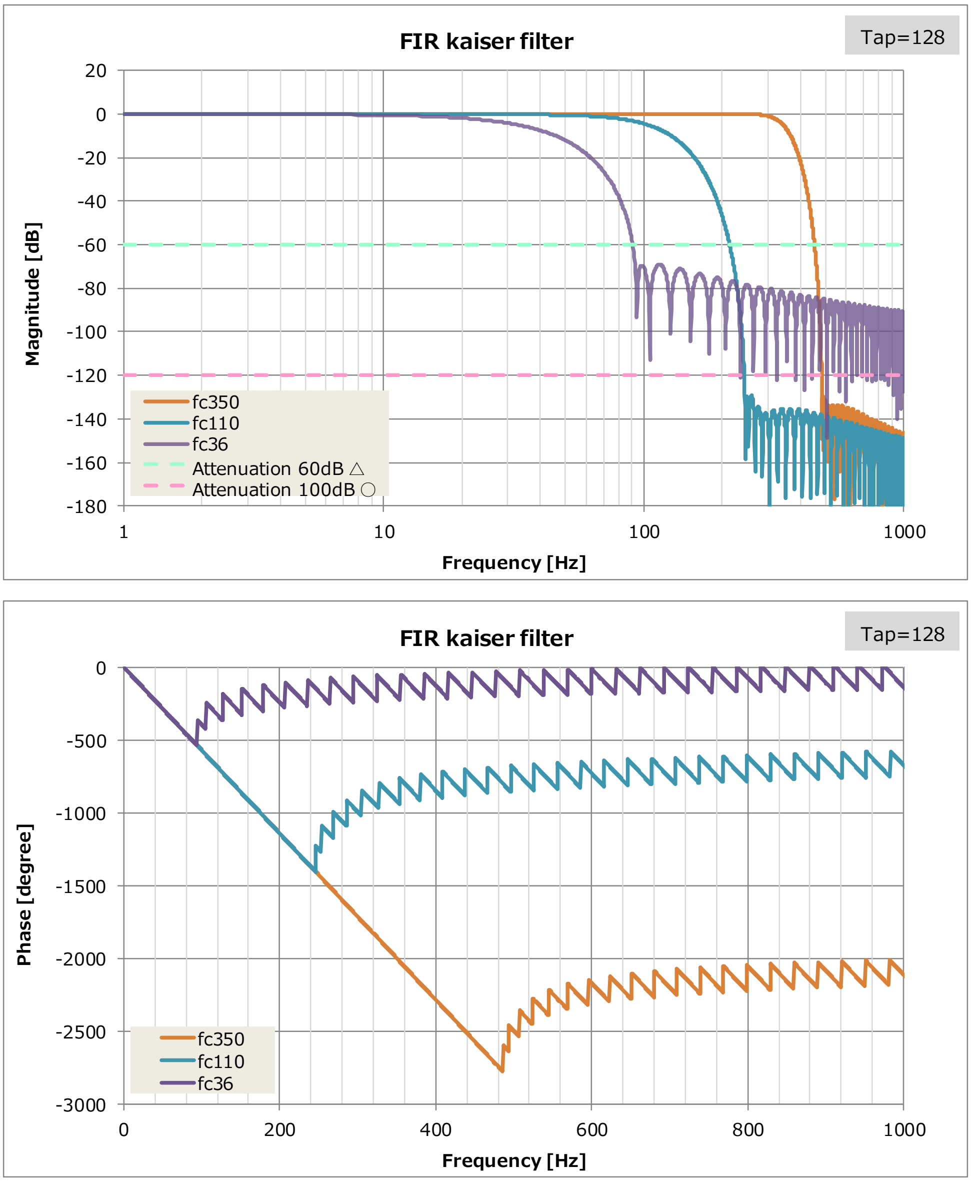
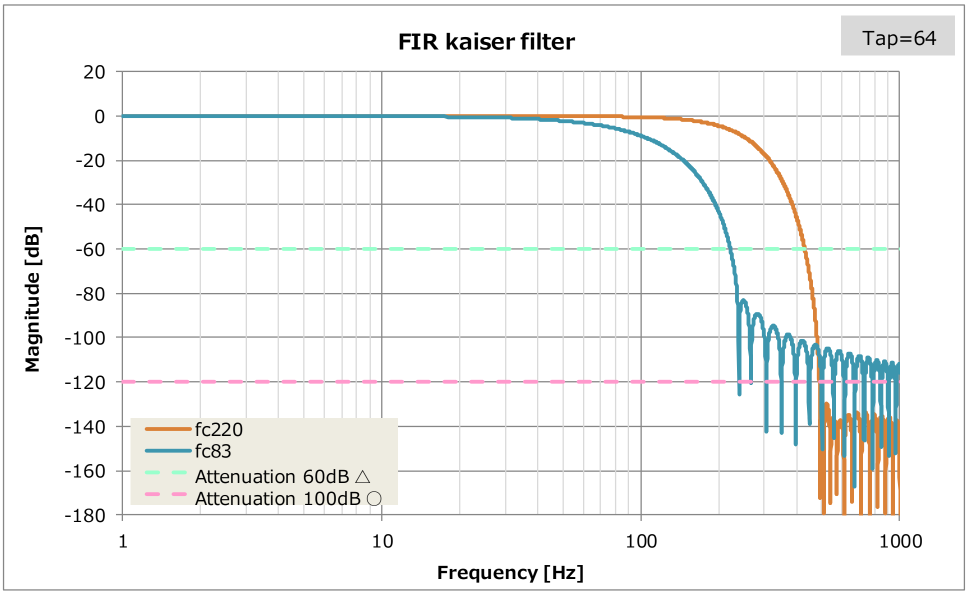



Figure 5-7 FIR Kaiser Characteristics－TAP128



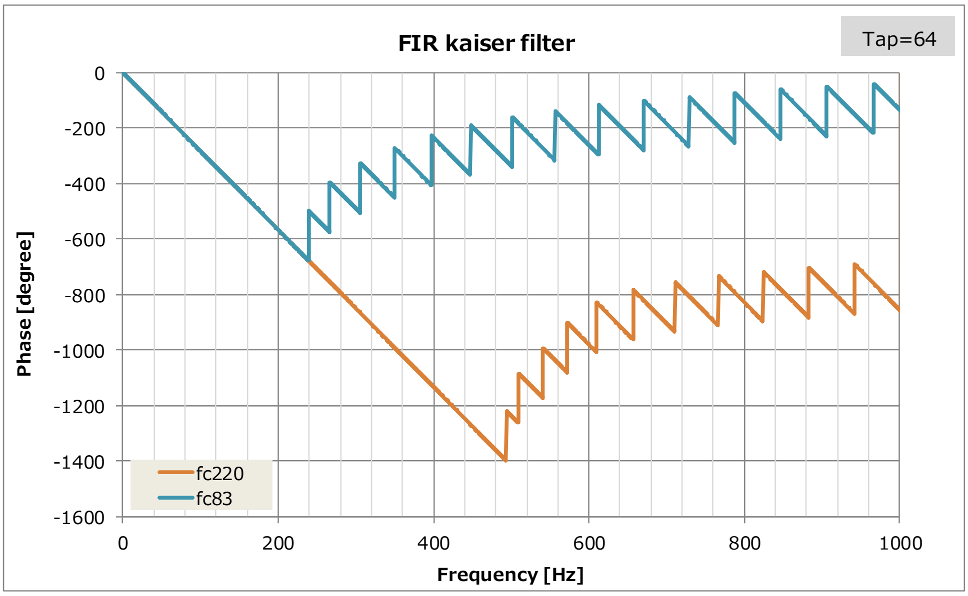




Figure 5-8 FIR Kaiser Characteristics－TAP64

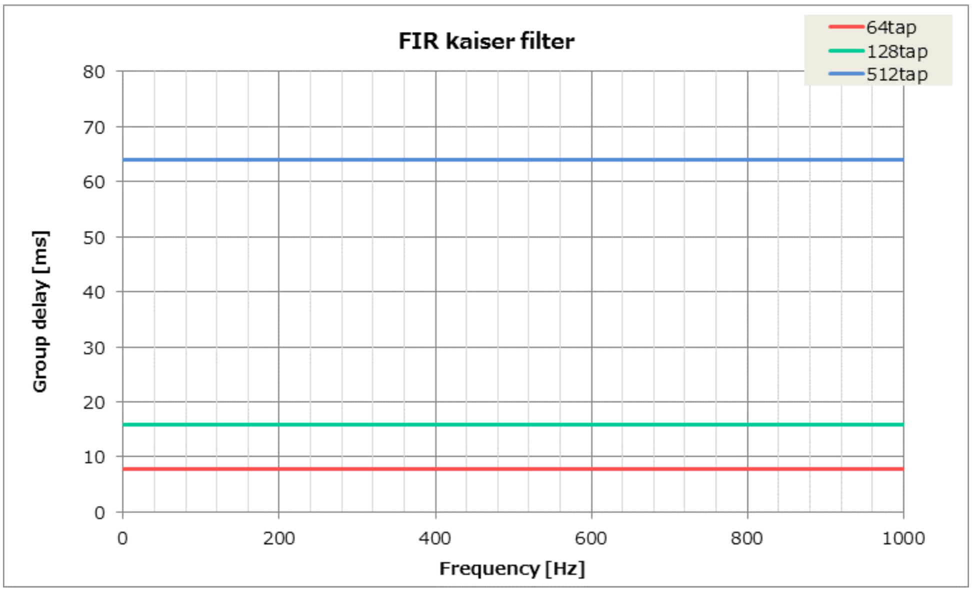



Figure 5-9 FIR Kaiser Characteristics－Group delay

### 5.16.2. User FIR Filter

The FIR filter can be arbitrarily defined by properly setting filter coefficients in the registers: FIR_UCMD [0x16 (W1)], FIR_UDATA [0x18 (W1)], FIR_UADDR [0x1A (W1)], and set **FILTER_SEL** of register: FILTER_CTRL [0x06(W1)] to "user defined FIR filter".

Follow the procedures below to program the user defined FIR filter.

- Register Programming Preparation

Set the filter coefficient value using signed 32 bit fixed point number with decimal point after bit [31]. For example, if the coefficient value in decimal form is 0.2195378928, the corresponding filter coefficient value in signed 32 bit fixed point form is $0.2195378928 * 2^{31} \fallingdotseq$ 0x1C19D153.

Table 5-1 shows the address ranges for the filter coefficients, and Figure 5-10 shows a N-tap FIR filter architecture and a coefficient memory map. The start address is common to each tap number and is at 0x0800. No specific values are set in memory at the time of factory shipment.

Table 5-1 User Defined FIR Filter Coefficient Address Ranges

| Tap | Coefficient Address Range |
|---|---|
| 4 | 0x0800-0x080F |
| 64 | 0x0800-0x08FF |
| 128 | 0x0800-0x09FF |
| 512 | 0x0800-0x0FFF |

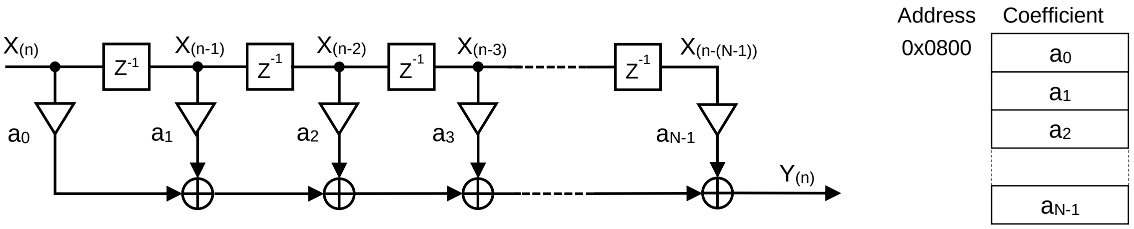



Figure 5-10 N-tap FIR Filter Architecture and Coefficient Memory Map

- Register Control (Write)

Set the filter coefficient address in register: FIR_UADDR [0x1B, 0x1A (W1)] and set the filter coefficient value in **FIR_UDATA** of register: FIR_UDATA [0x18 (W1)].

Set **FIR_UCMD** of register: FIR_UCMD [0x16 (W1)], bit [1: 0] to write the coefficient value. Next coefficient value can be set after waiting until the **FIR_UCMD** of register is to be “00: execution complete”

After the byte has completed writing, the address is automatically incremented by 1, so continuous programming of coefficients is possible without requiring additional address settings.

For the coefficient value, set the upper byte to the upper address and the lower byte to the lower address. Figure 5-11 shows the write sequence.

Please specify the type of filter,TAP setting and cutoff frequency using **FILTER_SEL** in register: FILTER_CTRL[0x06 (W1)], bit [3:0]. When selecting the user defined FIR filter, the **FILTER_SEL** register must reflect the filter coefficient data that are programmed in the device.

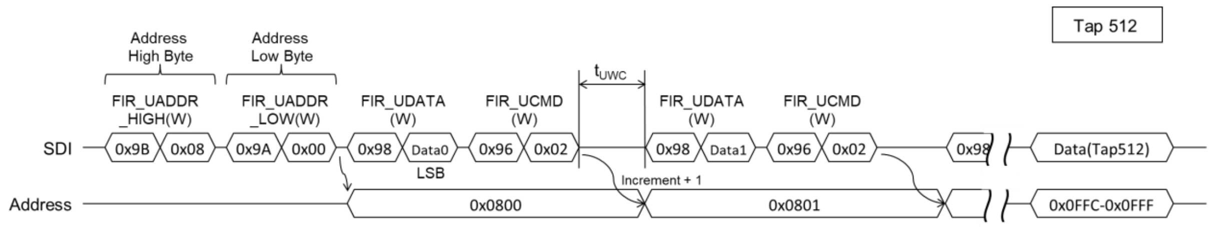



Figure 5-11 User Defined FIR Filter Coefficient Example Write Sequence (512 taps)

- Register Control (Read)

Set the filter coefficient address in register: FIR_UADDR [0x1B, 0x1A (W1)] and read the coefficient value using **FIR_UCMD** in register: FIR_UCMD [0x16 (W1)], bit [1: 0]. Next coefficient value can be read after waiting until the FIR_UCMD of register is to be “00: execution complete”

The address is automatically incremented by 1, so continuous execution of read commands is possible without requiring additional address settings. Figure 5-12 shows the read sequence.

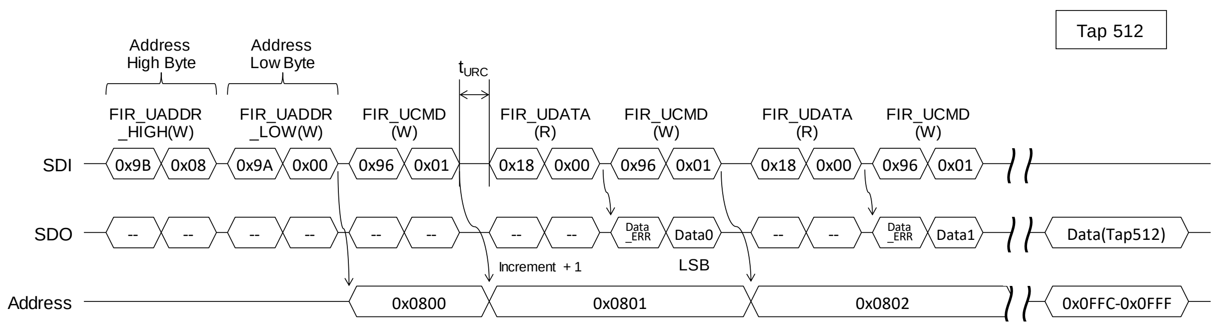



Figure 5-12 User Defined FIR Filter Coefficient Read Sequence (512 taps)

### 5.16.3. Notes For FIR Filter Usage

- Transient response

As shown inTable 5-2, transient response data is generated according to the combination of the tap number and the data output rate when sampling is started.

In the case of internal timer trigger measurement, the acceleration value of register ACCL [0x3A - 0x30(W0)] is not updated during this period.

In the case of automatic measurement, the device starts outputting data after the transient response.

Table 5-2 Transient Response Data Based on Output Data Rate and Filter Tap

| | 64 Taps | 128 Taps | 512 Taps |
|---|---|---|---|
| 1,000sps | 15 | 31 | 127 |
| 500sps | 7 | 15 | 63 |
| 200sps | | 7 | 31 |
| 100sps | | | 15 |
| 50sps | | | 7 |

- Supported Settings For Output Rate and Filter Cutoff Frequency

The host must set the cutoff frequency of the FIR filter and the output rate in proper combination to avoid aliasing.

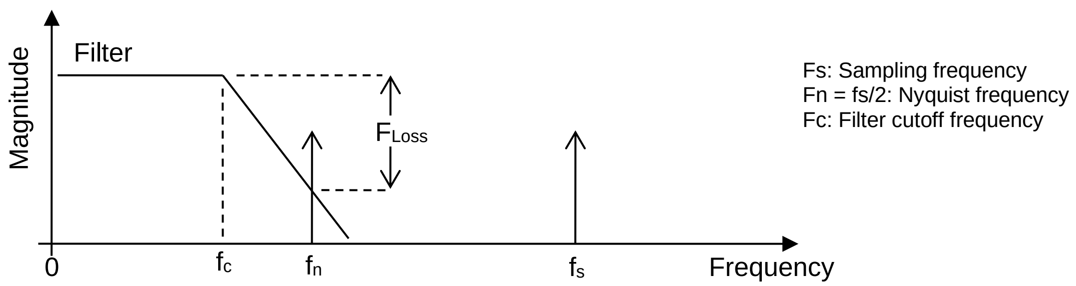



Figure 5-13 Anti-Aliasing Relationship Diagram

Table 5-3 Supported Settings for Output Rate and Filter Cutoff Frequency

| | | | | SMPL_CTRL Register（Internal Timer Trigger） | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Output Data Rate | | | | |
| | Tap | Fc | Group delay | 50 Sps | 100 Sps | 200 Sps | 500 Sps | 1,000 Sps |
| FILTER_CTRL Register | 512 Taps | 460 Hz | 63.875 ms | - | - | - | - | OK |
| | | 210 Hz | | - | - | - | OK | OK |
| | | 60 Hz | | - | - | OK[*1] | OK | OK |
| | | 16 Hz | | - | OK | OK | OK | OK |
| | | 9 Hz | | * | * | * | * | * |
| | 128 Taps | 350 Hz | 15.875 ms | - | - | - | - | OK |
| | | 110 Hz | | - | - | - | OK | OK |
| | | 36 Hz | | - | - | * | * | * |
| | 64 Taps | 220 Hz | 7.875 ms | - | - | - | - | OK |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 83 Hz | | - | - | - | * | * |

OK: $F_{Loss}$ < -120 dB Recommended setting
* : $F_{Loss}$ < -60 dB Although a possible setting, some decrease in measurement quality due to aliasing
ー : Fn < Fc Invalid setting. When using internal timer measurement, measurement data returns with error "0x64000000".
Note) These settings are valid when the user defined FIR filter function is used or the external trigger input function is active,

*1) The factory setting is Tap: 512, Fc: 60 Hz, ODR: 200 Hz

### 5.16.4. Long-Term Filter (HPF, LPF)

In addition to the FIR filter, this product has a simple filter for long-term measurement. This filter consists of a moving average operating at the output data rate. The number of taps can be set to a power of 2 in the range 2 to 4096.
LPF is a two-stage configuration with the set number of taps.
HPF consists of a single-stage configuration with the set number of taps and subtracting the moving average from the original data.

When using a long-term filter, set **FILT_EN** of register: LONGFILT_CTRL [0x1C (W1)], bit [0] to "1: valid" and select "LPF" or "HPF" in bit [1] **FILT_SEL**.
Set the number of taps to TAP_SIZE of register: LONGFILT_TAP [0x1E (W1)].

Please note that transient response data is generated according to the combination of the tap size and a kind of filters (HPF/LPF) when sampling is started. Numbers of the transient response data for long period filters are shown in Table 5-4.

Table 5-4 Numbers of Transient Response Data for Long Period Filters

| | Number of Transient Response Data |
|---|---|
| HPF | TAP Size |
| LPF | TAP Size * 2 |

Note) The maximum output rate is limited to 500 Sps when long-term filter is used.

# 6. HOST INTERFACE

## 6.1. SERIAL INTERFACE SPECIFICATION

### 6.1.1. COMUNICATION CONDITION

Table 6-1 shows the supported UART communication settings and Figure 6-1 shows the UART bit format. Please refer to **BAUD_RATE** (UART_CTRL [0x08(W1)] bit[9:8]) for changing the baud rate setting.

Table 6-1 UART Communication Settings

| Parameter | Settings |
|---|---|
| Transfer rate | 115.2kbps/ 230.4kbps/ 460.8kbps |
| Start | 1 bit |
| Data | 8 bits |
| Stop | 1 bit |
| Parity | None |
| Delimiter | CR(0x0D) |

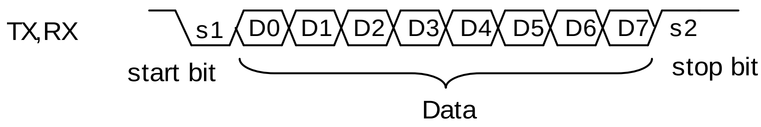



Figure 6-1 UART Bit Format

For the UART interface, a delimiter (1 byte) is placed at the end of each command (by the host) and response (by the IMU). In addition, for responses, the address (1 byte) specified by the command is added (by the IMU) to the beginning of the response.

Table 6-2 and Table 6-3 shows the timing of UART.

Table 6-2 UART Timing

| Parameter | Manual Sampling | | | | Auto Sampling | | Unit |
|---|---|---|---|---|---|---|---|
| | Normal Mode | | Burst Mode | | | | |
| | Minimum | Maximum | Minimum | Maximum | Minimum | Maximum | |
| $t_{STALL}$(115.2kbps) | - | 25 | - | 45 | - | - *2 | µs |
| $t_{STALL}$(230.4kbps) | - | 25 | - | 45 | - | - *2 | µs |
| $t_{STALL}$(460.8kbps) | - | 25 | - | 45 | - | - *2 | µs |
| $t_{WRITERATE}$(115.2kbps) | 660 | - | - | - | 660 | - | µs |
| $t_{WRITERATE}$(230.4kbps) | 350 | - | - | - | 350 | - | µs |
| $t_{WRITERATE}$(460.8kbps) | 200 | - | - | - | 200 | - | µs |
| $t_{READRATE}$(115.2kbps) | 660 | - | *1 | - | - *2 | - | µs |
| $t_{READRATE}$(230.4kbps) | 350 | - | *1 | - | - *2 | - | µs |
| $t_{READRATE}$(460.8kbps) | 200 | - | *1 | - | - *2 | - | µs |

*1) Please refer to Table 6.9.

*2) Register reading is not supported while in Sampling Mode with UART Auto Sampling enabled.

Table 6-3 UART Timing (tREADRATE requirements for Burst Mode)

| Parameter | Burst Mode (minimum) | Unit |
|---|---|---|
| $t_{READRATE}$(115.2kbps) | 660 + 86.8 * B | µs |
| $t_{READRATE}$(230.4kbps) | 350 + 43.4 * B | µs |
| $t_{READRATE}$(460.8kbps) | 200 + 21.7 * B | µs |

B= Number of receive data bytes (AD: address and CR: delimiter is not included).

Example tREADRATE Calculation:

BURST_CTRL[0x0C(W1)]: Set value 0x4702
B=18 byte for the above stated register setting
$t_{READRATE}$(460.8kbps) = 200 + (21.7 * 18) = 591(µs)

### 6.1.2. UART Read Timing (Normal Mode)

The response to the read command, i.e. the data from the internal register, is always returned 16-bit data at a time. The register address (AD) comes at the beginning of the response, for example, 0x02 for the MODE_CTRL [0x02(W0)] register.

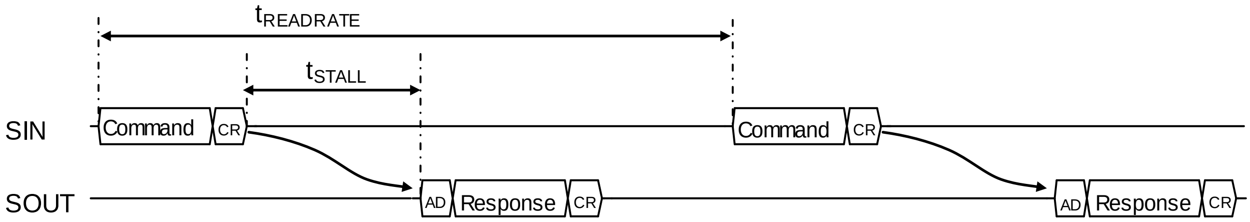



Figure 6-2 UART Read Timing (Normal Mode)

Table 6-4 Command Format (Read)

| First byte | | | | | | | | Second byte | | | | | | | | Third byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | A[6:0] | | | | | | | XX | | | | | | | | 0x0D | | | | | | | |

A[6:0] ･･･ Register address (even address)
XX ･･･ Don't Care
0x0D ･･･ Delimiter

Table 6-5 Response Format (Read)

| First byte | | | | | | | | Second byte | | | | | | | | Third byte | | | | | | | | Fourth byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | A[6:0] | | | | | | | D[15:8] | | | | | | | | D[7:0] | | | | | | | | 0x0D | | | | | | | |

A[6:0] ･･･ Register address (even address)
D[15:8] ･･･ Register read data (upper byte)
D[7:0] ･･･ Register read data (lower byte)
0x0D ･･･ Delimiter

### 6.1.3. UART Read Timing (Burst Mode)

Burst mode access of read data is supported using a “Burst Read Command” by writing 0x00 in **BURST_CMD** (BURST [0x00(W0)] bits[7:0]). In Burst Mode, ND/EA flag, temperature sensor value, 3-axis acceleration sensor value, etc. are consecutively sent as a response. The response format for the burst read output data is configured by register setting in BURST_CTRL [0x0C(W1)]. Please refer to 6.2 Data Packet Format for the response format.

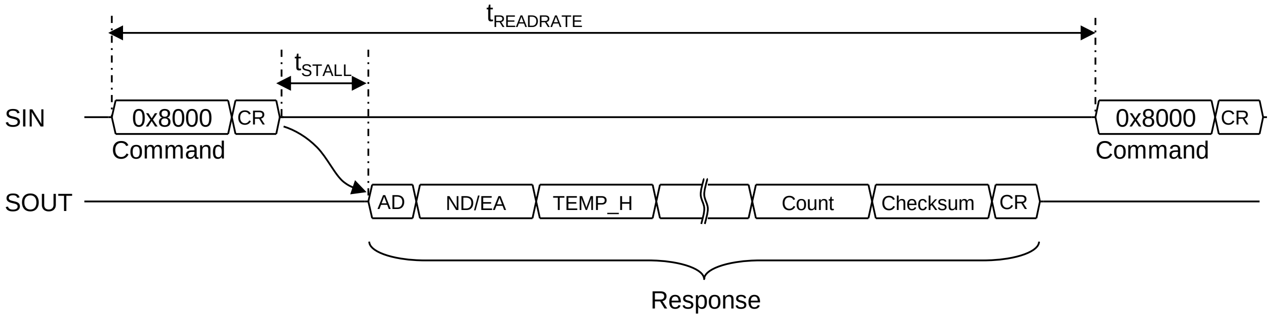



Figure 6-3 UART Read Timing (Burst Mode)

Table 6-6 Command Format (Burst Mode)

| First byte | | | | | | | | Second byte | | | | | | | | Third byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x80 | | | | | | | | 0x00 | | | | | | | | 0x0D | | | | | | | |

0x80 ··· Burst Command
0x00 ··· Burst Data 0x00
0x0D ··· Delimiter

### 6.1.4. UART Write Timing

A write command to a register will have no response. Unlike register reading, registers are written in 8-bit words.

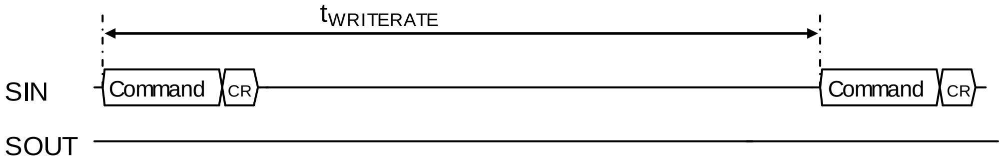



Figure 6-4 UART Write Timing

Table 6-7 Command Format (Write)

| First byte | | | | | | | | | Second byte | | | | | | | | Third byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | A[6:0] | | | | | | | | D[7:0] | | | | | | | | 0x0D | | | | | | | |

A[6:0] ··· Register address (even number or odd number)
D[7:0] ··· Register write data
0x0D ··· Delimiter

### 6.1.5. UART Auto Sampling Operation

When UART Auto sampling is active, all sensor outputs are sent as burst transfer automatically at the programmed output data rate without the request from the Host. For information about the response format, see 6.2 UART Data Packet Format. The response format for the burst read output data is configured by register setting in BURST_CTRL [0x0C(W1)].

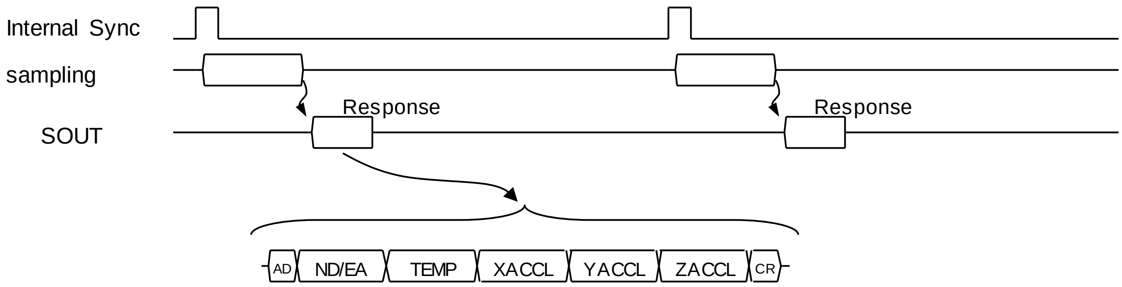



Figure 6-5 UART Auto Sampling Operation

## 6.2. Data Packet Format

The following table shows example of the data packet format sent to the host in the UART Burst Mode or UART Auto Sampling.

Table 6-8 UART Data Packet Format (UART Burst Mode / Auto Sampling) Example.
BURST_CTRL[0x0C(W1)]=0xC703 (Burst Output, Temp, Acceleration, Counter, Checksum)SIG_CTRL[0x00(W1)]=0x8E04 (Output Mode: Acceleration, Bias stabilization: Enable)

| Byte No. | Name | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ADDRESS | 0x80 | | | | | | | |
| 2 | ND | ND (Temp) | - | - | - | ND (XACCL) | ND (YACCL) | ND (ZACCL) | - |
| 3 | EA | - | - | - | XALARM _ERR | YALARM _ERR | ZALARM _ERR | ALIASI ERR | EA |
| 4 | TEMP_ HIGH_H | TEMP_HIGH [15:8] | | | | | | | |
| 5 | TEMP_ HIGH_L | TEMP_HIGH [7:0] | | | | | | | |
| 6 | TEMP_ LOW_H | TEMP_LOW [15:8] | | | | | | | |
| 7 | TEMP_ LOW_L | TEMP_LOW [7:0] | | | | | | | |
| 8 | XACCL _HIGH_H | XACCL_HIGH [15:8] | | | | | | | |
| 9 | XACCL _HIGH_L | XACCL _HIGH [7:0] | | | | | | | |
| 10 | XACCL _LOW_H | XACCL_LOW [15:8] | | | | | | | |
| 11 | XACCL _LOW_L | XACCL _LOW [7:0] | | | | | | | |
| 12 | YACCL _HIGH_H | YACCL _HIGH [15:8] | | | | | | | |
| 13 | YACCL _HIGH_L | YACCL _HIGH [7:0] | | | | | | | |
| 14 | YACCL _LOW_H | YACCL _LOW [15:8] | | | | | | | |
| 15 | YACCL _LOW_L | YACCL _LOW [7:0] | | | | | | | |
| 16 | ZACCL _HIGH_H | ZACCL _HIGH [15:8] | | | | | | | |
| 17 | ZACCL _HIGH_L | ZACCL _HIGH [7:0] | | | | | | | |
| 18 | ZACCL _LOW_H | ZACCL _LOW [15:8] | | | | | | | |
| 19 | ZACCL _LOW_L | ZACCL _LOW [7:0] | | | | | | | |
| 20 | COUNT_H | COUNT [15:8] | | | | | | | |
| 21 | COUNT_L | COUNT [7:0] | | | | | | | |
| 22 | CHECKSUM_H | CHECKSUM [15:8] | | | | | | | |
| 23 | CHECKSUM_L | CHECKSUM [7:0] | | | | | | | |
| 24 | CR | 0x0D | | | | | | | |

# 7. User Registers

A host device (for example, a microcontroller) can control the Accelerometer by accessing the control registers inside the device.

The registers are accessed in this device using a WINDOW method. The prescribed window number is first written to **WINDOW_ID** of WIN_CTRL[0x7E(W0/W1)] bit [7:0], then the desired register address can be accessed. The WIN_CTRL [0x7E(W0/W1)] register can always be accessed without needing to set the window number.

During the Power-On Start-Up Time or the Reset Recovery time specified in the Table 2-4 Interface Specifications, all the register values are undefined because internal initialization is in progress. Ensure the device registers are only accessed after the Power-On Start-Up Time or the Reset Recovery time is over.

For information about the initial values of the control registers after internal initialization is finished, see the “Default” column in the Table 7.1. The control registers with ○ mark in the “Flash Backup” column can be saved to the non-volatile memory by the user, and the initial values after the power on will be the values read from the non-volatile memory. If the read out from the non-volatile memory fails, the **FLASH_ERR** (DIAG_STAT [0x04(W0)] bit[2]) is set to 1 (error).

Please ensure that the device is in the Configuration Mode before writing to registers. In the Sampling Mode, writing to registers is ignored **except** for the following cases.

- MODE_CTRL [0x02(W0)] bit [9:8] in **MODE_CMD**
- GLOB_CMD [0x0A(W1)] bit [7] in **SOFT_RST**
- WIN_CTRL [0x7E(W0/W1)] bit [7:0] in **WINDOW_ID**

While with the UART Auto sampling and Sampling Mode is active, register read access is not supported. Otherwise, the sampling data transmitted in the UART Auto sampling will be corrupted by the response data from the register read.

Each register is 16-bit wide and one address is assigned to every 8 bits. Registers are read in 16-bit words and are written in 8-bit words. The byte order of each 16-bit register is little endian, but the byte order of the 16-bit data transferred over the digital interface is big endian.

Table 7.1 shows the register map, and Section 7.1 through Section 7.30 describes the registers in detail.

The “-” sign in the register assignment table in Section 7.1 through Section 7.30 means “reserved”.

Write a “0” to reserved bits during a write operation.

During a read operation, a reserved bit can return either 0 or 1 (“don’t care”).

Writing to a read-only register is prohibited.

NOTE) The explanation of the register notation MODE_CTRL [0x02(W0)] bit [9:8] is as follows:

- MODE_CTRL: Register name
- [0x02(W0)]: First number is the Register Address, (W0) means Window Number “0”
- bit[9:8]: Bits 9 to 8

Table 7-1 Register Map

| Name | Window ID | Address | (*3)Read Command 16bit Read | Write Command 8bitWrite | R/W | Flash Backup | Default | Function |
|---|---|---|---|---|---|---|---|---|
| BURST | 0 | 0x00 | - | 0x80 | W | - | - | Burst mode |
| | | 0x01 | | - | - | | - | |
| MODE_CTRL | 0 | 0x02 | 0x02XX | - | - | - | 0x00 | Operation mode control |
| | | 0x03 | | 0x83 | R/W | | 0x04 | |
| DIAG_STAT | 0 | 0x04 | 0x04XX | - | R | - | 0x00 | Diagnostic result |
| | | 0x05 | | - | R | | 0x00 | |
| FLAG | 0 | 0x06 | 0x06XX | - | R | - | 0x00 | ND/EA flag |
| | | 0x07 | | - | R | | 0x00 | |
| COUNT | 0 | 0x0A | 0x0AXX | - | R | - | 0x00 | Sampling count |
| | | 0x0B | | - | R | | 0x00 | |
| TEMP_HIGH | 0 | 0x0E | 0x0EXX | - | R | - | 0xFF | Temperature sensor value High |
| | | 0x0F | | - | R | | 0xFF | |
| TEMP_LOW | 0 | 0x10 | 0x10XX | - | R | - | 0xFF | Temperature sensor value Low |
| | | 0x11 | | - | R | | 0xFF | |
| XACCL_HIGH | 0 | 0x30 | 0x30XX | - | R | - | 0xFF | X acceleration sensor value High |
| | | 0x31 | | - | R | | 0xFF | |
| XACCL_LOW | 0 | 0x32 | 0x32XX | - | R | - | 0xFF | X acceleration sensor value Low |
| | | 0x33 | | - | R | | 0xFF | |
| YACCL_HIGH | 0 | 0x34 | 0x34XX | - | R | - | 0xFF | Y acceleration sensor value High |
| | | 0x35 | | - | R | | 0xFF | |
| YACCL_LOW | 0 | 0x36 | 0x36XX | - | R | - | 0xFF | Y acceleration sensor value Low |
| | | 0x37 | | - | R | | 0xFF | |
| ZACCL_HIGH | 0 | 0x38 | 0x38XX | - | R | - | 0xFF | Z acceleration sensor value High |
| | | 0x39 | | - | R | | 0xFF | |
| ZACCL_LOW | 0 | 0x3A | 0x3AXX | - | R | - | 0xFF | Z acceleration sensor value Low |
| | | 0x3B | | - | R | | 0xFF | |
| XTILT_HIGH | 0 | 0x3C | 0x3CXX | - | R | - | 0xFF | X Tilt sensor Value High |
| | | 0x3D | | - | R | | 0xFF | |
| XTILT_LOW | 0 | 0x3E | 0x3EXX | - | R | - | 0xFF | X Tilt sensor Value Low |
| | | 0x3F | | - | R | | 0xFF | |
| YTILT_HIGH | 0 | 0x40 | 0x40XX | - | R | - | 0xFF | Y Tilt sensor Value High |
| | | 0x41 | | - | R | | 0xFF | |
| YTILT_LOW | 0 | 0x42 | 0x42XX | - | R | - | 0xFF | Y Tilt sensor Value Low |
| | | 0x43 | | - | R | | 0xFF | |
| ZTILT_HIGH | 0 | 0x44 | 0x44XX | - | R | - | 0xFF | Z Tilt sensor Value High |
| | | 0x45 | | - | R | | 0xFF | |
| ZTILT_LOW | 0 | 0x46 | 0x46XX | - | R | - | 0xFF | Z Tilt sensor Value Low |
| | | 0x47 | | - | R | | 0xFF | |
| SIG_CTRL | 1 | 0x00 | 0x00XX | 0x80 | R/W | ○ | 0x04 | DataReady signal & polarity control |
| | | 0x01 | | 0x81 | R/W | | 0x8E | |
| MSC_CTRL | 1 | 0x02 | 0x02XX | 0x82 | R/W | ○ | 0x26 | Other control |
| | | 0x03 | | 0x83 | R/W | | 0x00 | |
| SMPL_CTRL | 1 | 0x04 | 0x04XX | - | - | ○ | 0x00 | Sampling control |
| | | 0x05 | | 0x85 | R/W | | 0x04 | |
| FILTER_CTRL | 1 | 0x06 | 0x06XX | 0x86 | R/W | ○ | 0x08 | Filter control |
| | | 0x07 | | - | - | | 0x00 | |
| UART_CTRL | 1 | 0x08 | 0x08XX | 0x88 | R/W | ○ | 0x00 | UART control |
| | | 0x09 | | 0x89 | R/W | | 0x02 | |
| GLOB_CMD | 1 | 0x0A | 0x0AXX | 0x8A | R/W | - | 0x00 | System control |
| | | 0x0B | | - | R | | 0x00 | |
| BURST_CTRL | 1 | 0x0C | 0x0CXX | 0x8C | R/W | ○ | 0x02 | Burst control |
| | | 0x0D | | 0x8D | R/W | | 0x47 | |
| FIR_UCMD | 1 | 0x16 | 0x16XX | 0x96 | R/W | - | 0x00 | User FIR Filter control |
| | | 0x17 | | - | - | | 0x00 | |
| FIR_UDATA | 1 | 0x18 | 0x18XX | 0x98 | R/W | - | 0x00 | User FIR Filter coefficient data |
| | | 0x19 | | - | R | | 0x00 | |
| FIR_UADDR | 1 | 0x1A | 0x1AXX | 0x9A | R/W | - | 0x00 | User FIR Filter coefficient Address |
| | | 0x1B | | 0x9B | R/W | | 0x08 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| LONGFILT_CTRL | 1 | 0x1C | 0x1CXX | 0x9C | R/W | ○ | 0x00 | Long period filter control |
| | | 0x1D | | - | - | | 0x00 | |
| LONGFILT_TAP | 1 | 0x1E | 0x1EXX | 0x9E | R/W | ○ | 0x0A | Long period filter tap number |
| | | 0x1F | | - | - | | 0x00 | |
| OFFSET_XA_HIGH | 1 | 0x2C | 0x2CXX | 0xAC | R/W | ○ | 0x00 | X acceleration offset value High |
| | | 0x2D | | 0xAD | R/W | | 0x00 | |
| OFFSET_XA_LOW | 1 | 0x2E | 0x2EXX | 0xAE | R/W | ○ | 0x00 | X acceleration offset value Low |
| | | 0x2F | | 0xAF | R/W | | 0x00 | |
| OFFSET_YA_HIGH | 1 | 0x30 | 0x30XX | 0xB0 | R/W | ○ | 0x00 | Y acceleration offset value High |
| | | 0x31 | | 0xB1 | R/W | | 0x00 | |
| OFFSET_YA_LOW | 1 | 0x32 | 0x32XX | 0xB2 | R/W | ○ | 0x00 | Y acceleration offset value Low |
| | | 0x33 | | 0xB3 | R/W | | 0x00 | |
| OFFSET_ZA_HIGH | 1 | 0x34 | 0x34XX | 0xB4 | R/W | ○ | 0x00 | Z acceleration offset value High |
| | | 0x35 | | 0xB5 | R/W | | 0x00 | |
| OFFSET_ZA_LOW | 1 | 0x36 | 0x36XX | 0xB6 | R/W | ○ | 0x00 | Z acceleration offset value Low |
| | | 0x37 | | 0xB7 | R/W | | 0x00 | |
| XALARM | 1 | 0x46 | 0x46XX | 0xC6 | R/W | ○ | 0xF1 | X acceleration alarm |
| | | 0x47 | | 0xC7 | R/W | | 0x0F | |
| YALARM | 1 | 0x48 | 0x48XX | 0xC8 | R/W | ○ | 0xF1 | Y acceleration alarm |
| | | 0x49 | | 0xC9 | R/W | | 0x0F | |
| ZALARM | 1 | 0x4A | 0x4AXX | 0xCA | R/W | ○ | 0xF1 | Z acceleration alarm |
| | | 0x4B | | 0xCB | R/W | | 0x0F | |
| PROD_ID1 | 1 | 0x6A | 0x6AXX | - | R | - | 0x41 | Product ID 1 |
| | | 0x6B | | - | R | | 0x33 | |
| PROD_ID2 | 1 | 0x6C | 0x6CXX | - | R | - | 0x35 | Product ID 2 |
| | | 0x6D | | - | R | | 0x32 | |
| PROD_ID3 | 1 | 0x6E | 0x6EXX | - | R | - | 0x41 | Product ID 3 |
| | | 0x6F | | - | R | | 0x44 | |
| PROD_ID4 | 1 | 0x70 | 0x70XX | - | R | - | 0x31 | Product ID 4 |
| | | 0x71 | | - | R | | 0x30 | |
| VERSION | 1 | 0x72 | 0x72XX | - | R | - | (*1) | Firmware version |
| | | 0x73 | | - | R | | | |
| SERIAL_NUM1 | 1 | 0x74 | 0x74XX | - | R | - | (*2) | Serial Number 1 |
| | | 0x75 | | - | R | | | |
| SERIAL_NUM2 | 1 | 0x76 | 0x76XX | - | R | - | | Serial Number 2 |
| | | 0x77 | | - | R | | | |
| SERIAL_NUM3 | 1 | 0x78 | 0x78XX | - | R | - | | Serial Number 3 |
| | | 0x79 | | - | R | | | |
| SERIAL_NUM4 | 1 | 0x7A | 0x7AXX | - | R | - | | Serial Number 4 |
| | | 0x7B | | - | R | | | |
| WIN_CTRL | 0,1 | 0x7E | 0x7EXX | 0xFE | R/W | - | 0x00 | Register Window Control |
| | | 0x7F | | - | - | | 0x00 | |

*1. It depends on the version of the installed firmware.

*2. It is determined by each individual serial number.

*3. Lower byte XX: Do not care

## 7.1. BURST Register (Window 0)

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|

| | | |
|---|---|---|
| 0x01 | - | - |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x00 | BURST_CMD | | | W |

**bit[7:0] BURST_CMD**

A burst mode read operation is initiated by writing 0x00 in **BURST_CMD** of this register.

NOTE) The data transmission format is described in 6.1.3 SPI Read Timing (Burst Mode) and 6.2.2 UART Read Timing (Burst Mode). Also refer to 6.3 Data Packet Format. The output data can be selected by setting BURST_CTRL [0x0C(W1)].

## 7.2. MODE_CTRL Register (Window 0)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x03 | - | - | - | - | MODE _STAT | | MODE_CMD | | R/W *1 |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x02 | - | - | - | - | - | - | - | - | - |

*1) MODE_STAT is read-only.

**bit[11:10] MODE_STAT**

This read-only status bit shows the current operation mode.

00: Sampling Mode
01: Configuration mode
10: Sleep Mode
11: (Not Used)

**bit[9:8] MODE_CMD**

Executes commands related to the operation mode.

00: Execute Complete.
01: Go to the Sampling Mode.
After the mode transition is completed, the bits automatically goes back to “00”.
10: Go to the Configuration Mode.
After the mode transition is completed, the bits automatically goes back to “00”.
11: Go to the Sleep Mode.
After the mode transition is completed, the bits automatically goes back to “00”.

## 7.3. DIAG_STAT Register (Window 0)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x05 | ACC_VS _ERR | ACC_X _ERR | ACC_Y _ERR | ACC_Z _ERR | SENS _ERR | | TEMP _ERR | VDD _ERR | R |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x04 | HARD _ERR | | | - | UART _OVF | FLASH _ERR | ACC_ERR _ALL | FLASH_ BU_ERR | R |

Note) When the host reads the diagnosis result, all the results (including the EA flag in the FLAG register) will be cleared to 0.

**bit[15] ACC_VES_ERR (ACC VEctorSynthesis ERRor)**
Shows the execution result of vector composite value of acceleration three axes in **ACC_TEST** of MSC_CTRL [0x02 (W1)], bit [10].
1: Error occurred
0: No error
If this error occurs, acceleration sensor is faulty.

**bit[14] ACC_X_ERR**
Shows the result of X axis acceleration sensor operation check in **ACC_TEST** of MSC_CTRL [0x02 (W1)], bit [10].
1: Error occurred
0: No error
If this error occurs, X axis acceleration sensor has failed (operation stop)

**bit[13] ACC_Y_ERR**
Shows the result of Y axis acceleration sensor operation check in **ACC_TEST** of MSC_CTRL [0x02 (W1)], bit [10].
1: Error occurred
0: No error
If this error occurs, Y axis acceleration sensor has failed (operation stop)

**bit[12] ACC_Z_ERR**
Shows the result of Z axis acceleration sensor operation check in **ACC_TEST** of MSC_CTRL [0x02 (W1)], bit [10].
1: Error occurred
0: No error
If this error occurs, Z axis acceleration sensor has failed (operation stop)

**bit[11:10] SENS_ERR (SENSitivity ERRor)**
Shows the execution result of **SENS_TEST** (Sensitivity Test) of MSC_CTRL [0x02 (W1)], bit [14:12].
11 : Not used
10 : Unable to be determined
01 : Error occurred
00 : No error
If this error occurs, acceleration sensor is faulty.

**bit[9] TEMP_ERR**
Shows the execution result of **TEMP_TEST** (Temp Sensor Check) of MSC_CTRL [0x02 (W1)], bit [9].
1: Error occurred
0: No error
If this error occurs, temperature sensor is faulty.

**bit[8] VDD_ERR**
Shows the execution result of **VDD_TEST** (Power Supply Voltage Check) of MSC_CTRL [0x02 (W1)], bit [8].
1: Error occurred
0: No error
If this error occurs, Check whether the power supply voltage level is within the specified range.

**bit[7:5] HARD_ERR**

Shows the result of the hardware check at startup.
Other than 000: Error occurred
000 : No error
When this error occurs, it indicates the device is faulty.

**bit[3] UART_OVF (UART OVer Flow)**

Shows an error occurred if the data transmission rate is faster than the UART baud rate.
1: Error occurred
0: No error
When this error occurs, review the settings for the baud rate（register：UART_CTRL[0x08(W1)], bit[9:8]）, data output rate（register：SMPL_CTRL[0x04(W1)],bit[11:8]）, UART Burst Mode / Auto sampling (register：BURST_CTRL[0x0C(W1)] ) in combination.

**bit[2] FLASH_ERR**

Shows the result of **FLASH_TEST** of MSC_CTRL [0x02(W1)] bit[11].
1: Error occurred
0: No error
This error indicates a failure occurred when reading data out from the non-volatile memory.

**bit[1] ACC_ERR_ALL (ACCTest ERRor All)**

Shows the logical sum of bit [15:12] of this register.
1: Error occurred
0: No error

**bit[0] FLASH_BU_ERR (FLASH BackUp ERRor)**

Shows the result of **FLASH_BACKUP** of GLOB_CMD [0x0A(W1)] bit [3].
1: Error occurred
0: No error

## 7.4. FLAG(ND/EA) Register (Window 0)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x07 | ND (Temp) | - | - | - | ND (XACCL) | ND (YACCL) | ND (ZACCL) | - | R |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x06 | - | - | - | XALARM _ERR | YALARM _ERR | ZALARM _ERR | ALIASI _ERR | EA | R |

Note) ALARM_ERR flags are cleared to "0" by reading this register.
Note) The EA flag is cleared to "0" by reading the DIAG_STAT register.
Note) The ALIASI_ERR flag is updated when writing to the SMPL_CTRL or FILTER_CTRL register.

**bit[15] ND(New Data) flag (Temperature)**

When new measurement data is set in temperature register: TEMP_HIGH [0x0E (W0)], this bit is set to "1". This bit is reset to “0” reading by the temperature register.

**bit[11:9] ND(New Data) flag (Acceleration)**

When new measurement data is set in acceleration register: XACCL_ HIGH[0x30(W0)], YACCL_ HIGH[0x34(W0)], ZACCL_ HIGH[0x38(W0)], this bit is set to "1". This bit is reset to “0” by reading the temperature register.

**bit[4] XALARM_ERR(XAcc_ALARM_ERRor)**

This bit indicates when the acceleration exceeds the value set in register: XA_ALARM[0x47-0x46(W1)] in the X axis during measurement.
1: detection
0: no detection

**bit[3] YALARM_ERR(YAcc_ALARM_ERRor)**

This bit indicates when the acceleration exceeds the value set in register: YA_ALARM[0x49-0x48(W1)] in the Y axis during measurement.

1: detection

0: no detection

**bit[2] ZALARM_ERR(ZAcc_ALARM_ERRor)**

This bit indicates when the acceleration exceeds the value set in register: ZA_ALARM[0x4B-0x4A(W1)] in the Z axis during measurement.

1: detection

0: no detection

**bit[1] ALIASI_ERR(ALIASIng_ERRor)**

This bit indicates the validation check of the combination setting of the output rate in register: SMPL_CTRL[0x04(W1)],bit[11:8] and filter cutoff frequency in register: FILTER_CTRL[0x06 (W1)],bit[3:0].

1: Abnormal Setting

0: Normal Setting

**bit[0] EA(All Error) flag**

When at least one failure is found in the diagnostic result (DIAG_STAT [0x04(W0)]), this bit is set to “1”(failure occurred). This bit is reset to “0” by reading the DIAG_STAT register.

1: Failure occurred

0: No Failure

## 7.5. COUNT Register (Window 0)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x0A | COUNT | | | R |

**bit[15:0] COUNT**

This register returns the sampling count value of the internal A/D converter.

Note) The time unit of the sampling counter value represents 250 μs/count.
Example: If the data output rate equals 1000Sps, the counter value sequence is 4,8,12, ... , 0xFFFC, 0, 4, ....

## 7.6. TEMP Register (Window 0)

| Addr (Hex) | Bit15 | ... | Bit0 | R/W |
|---|---|---|---|---|
| 0x0E | | TEMP_HIGH | | R |
| 0x10 | | TEMP_LOW | | R |

**bit[15:0] Temperature sensor output data**

The internal temperature sensor value can be read from this register.
The output data format is 32-bit two's complement format.

Please refer to the below formula for conversion to temperature in centigrade. Please refer to Table 2.3 Sensor Specification for the scale factor value.
The reference value in this register is for the temperature correction. There is no guarantee that the value provides the absolute value of the internal temperature.

T [℃]= SF * a + 34.987

SF: Scale Factor
A: Temperature sensor output data (decimal)

## 7.7. ACCL Register (Window 0)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x30 | | XACCL_HIGH | | R |
| 0x32 | | XACCL_LOW | | R |
| 0x34 | | YACCL_HIGH | | R |
| 0x36 | | YACCL_LOW | | R |
| 0x38 | | ZACCL_HIGH | | R |
| 0x3A | | ZACCL_LOW | | R |

**bit[15:0] Acceleration sensor output data**

These registers contain the 3-axis acceleration data for X, Y, and Z.
Register: SIG_CTRL [0x00 (W1)] provides the output mode selection OUTPUT_SEL of bit [7: 5] to specify the acceleration data as either "acceleration" or "Tilt angle".

The output data format
Unit [ G ]
32-bit two's complement format
bit31 : sign
bit30~24 : integer
bit23~0 : decimal

Note) When the combination of output rate and filter cutoff frequency is "abnormal setting", reading acceleration sensor value responds with error code "0x64000000"
Note) When the acceleration value exceeds the preset threshold value, reading acceleration value responds with the threshold value. For example, if the preset threshold values are set to +15 G and -15 G, the corresponding response is “0x0F000000” for +15 G or more, and “0xF1000000” for -15 G or less.

## 7.8. TILT Register (Window 0)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x3C | XTILT_HIGH | | | R |
| 0x3E | XTILT_LOW | | | R |
| 0x40 | YTILT_HIGH | | | R |
| 0x42 | YTILT_LOW | | | R |
| 0x44 | ZTILT_HIGH | | | R |
| 0x46 | ZTILT_LOW | | | R |

**bit[15:0] Tilt sensor output data**

These registers contain the 3-axis Tilt angle data for X, Y, and Z.
Register: SIG_CTRL [0x00 (W1)] The tilt angle is output only when the output mode selection OUTPUT_SEL of bit [7: 5] is set to "tilt angle".

The output data format
Unit [ radian ]
32-bit two's complement format
bit31 : sign
bit30~29 : integer
bit28~0 : decimal

Note) When the combination of output rate and filter cutoff frequency is "abnormal setting", reading tilt angle sensor value responds with error code "0x64000000"
Note) When the tilt angle value exceeds the dynamic range (±60 deg), reading tilt angle value responds with the value of +60 deg or -60 deg. For example, the corresponding response is “0x2182A470” for +60 deg or more, and “0xDE7D5B90” for - 60 deg or less.

## 7.9. SIG_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x01 | ND_EN (Temp) | - | - | - | ND_EN (XACCL) | ND_EN (YACCL) | ND_EN (ZACCL) | - | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x00 | OUTPUT_ SEL_X | OUTPUT_ SEL_Y | OUTPUT_ SEL_Z | MESMOD _SEL | - | TEMP_ STABIL | - | - | R/W |

**bit[15] ND_EN (Temp)**

Enables or disables the temperature sensor ND flags in FLAG [0x06(W0)] bit [15].
1: Enable
0: Disable

**bit[11] ND_EN (X Acceleration sensor)**

Enables or disables the X Acceleration sensor ND flags in FLAG [0x06(W0)] bit [11].
1: Enable
0: Disable

**bit[10] ND_EN (Y Acceleration sensor)**
Enables or disables the Y Acceleration sensor ND flags in FLAG [0x06(W0)] bit [10].
1: Enable
0: Disable

**bit[9] ND_EN (Z Acceleration sensor)**
Enables or disables the Z Acceleration sensor ND flags in FLAG [0x06(W0)] bit [9].
1: Enable
0: Disable

**bit[7] OUTPUT_SEL_X**
Sets the output mode on the X axis.
1: Tilt angle
0: Acceleration

**bit[6] OUTPUT_SEL_Y**
Sets the output mode on the Y axis.
1: Tilt angle
0: Acceleration

**bit[5] OUTPUT_SEL_Z**
Sets the output mode on the Z axis.
1: Tilt angle
0: Acceleration

**bit[4] MESMOD_SEL**
Sets the measurement condition.
1: Reduced noise floor condition
0: Standard noise floor condition
When **MESMOD_SEL** is set, the device can operate with the set measurement condition after completing internal initialization after powered on or a reset.
Write to this **MESMOD_SEL** bit. Then execute FLASH_BACKUP of GLOB_CMD [0x0A(W1)] bit [3] to preserve the current register settings. Read the **MESMOD_STAT** of register: GLOB_CMD[0x0A (W1)],bit[12] to check the current setting of measurement condition.

**bit[2] TEMP_STABIL**
Bias stabilization against thermal shock.
1: Enable
0: Disable

## 7.10. MSC_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x03 | - | Z_SENS _TEST | Y_SENS _TEST | X_SENS _TEST | FLASH _TEST | ACC _TEST | TEMP _TEST | VDD _TEST | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x02 | - | 0 | - | - | - | - | - | - | R/W |

Note) Although ACC_TEST, TEMP_TEST, and VDD_TEST can be executed at the same time, other tests cannot be executed at the same time.
When executing them in succession, confirm the execution of the previous command is finished by waiting until the bit changes from “1” to “0” and then execute the next command.

**bit[14] Z_SENS_TEST**
Write “1” to execute the self test to check if the Z axis accelerometer sensitivity is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **SENS_ERR** of DIAG_STAT [0x04(W0)] bit [11:10] to check the result.

**bit[13] Y_SENS_TEST**
Write “1” to execute the self test to check if the Y axis accelerometer sensitivity is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **SENS_ERR** of DIAG_STAT [0x04(W0)] bit [11:10] to check the result.

**bit[12] X_SENS_TEST**
Write “1” to execute the self test to check if the X axis accelerometer sensitivity is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **SENS_ERR** of DIAG_STAT [0x04(W0)] bit [11:10] to check the result.

**bit[11] FLASH_TEST**
Write “1” to execute the data consistency test for the non-volatile memory. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **FLASH_ERR** of DIAG_STAT [0x04(W0)] bit [2] to check the result.

**bit[10] ACC_TEST**
Write “1” to execute the self test to check if the accelerometer is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **ACC_ERR_ALL** of DIAG_STAT [0x04(W0)] bit [1] to check the results.

**bit[9] TEMP_TEST**
Write “1” to execute the self test to check if temperature sensor is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **TEMP_ERR** of DIAG_STAT [0x04(W0)] bit [9] to check the results.

**bit[8] VDD_TEST**
Write “1” to execute the self test to check if power supply voltage level is working properly. The read value of the bit is “1” during the test and “0” after the test is completed. After writing “1” to this bit, wait until this bit goes back to “0” and then read the **VDD_ERR** of DIAG_STAT [0x04(W0)] bit [8] to check the results.

**bit[6] Fixed to 0**
This function cannot be used. Fix to 0.

## 7.11. SMPL_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x05 | - | - | - | - | DOUT_RATE | | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x04 | - | | | - |

**bit[11:8] DOUT_RATE**

Specifies the data output rate. To avoid aliasing, refer to Table 5.4 Measurable output rate and cutoff frequency combination for output rate setting.

0000: Reserved
0001: Reserved
0010: 1,000Sps
0011: 500Sps
0100: 200Sps
0101: 100Sps
0110: 50Sps
0111-1111: not used

*1) The factory setting is ODR: 200 Hz
Note) The maximum output rate is limited to 500 Sps when long-term filter is used.

## 7.12. FILTER_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x07 | - | - | - | - | - | - | - | - | - |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x06 | - | - | FILTER_ STAT | - | FILTER_SEL | | | | R/W *1 |

*1) Only FILTER_STAT is read-only.

**bit[5] FILTER_STAT**

This read-only status bit shows the status of the filter setting

1: Filter setting is busy

0: Filter setting is completed

**bit[3:0] FILTER_SEL**

Specifies the type of filter,TAP setting and cutoff frequency.

For the FIR Kaiser filter, these bits also selects the cutoff frequency. When using user defined FIR filter, please program the filter coefficient data

0000: Reserved
0001: FIR Kaiser Filter TAP=64、fc=83
0010: FIR Kaiser Filter TAP=64、fc=220
0011: FIR Kaiser Filter TAP=128、fc=36
0100: FIR Kaiser Filter TAP=128、fc=110
0101: FIR Kaiser Filter TAP=128、fc=350
0110: FIR Kaiser Filter TAP=512、fc=9
0111: FIR Kaiser Filter TAP=512、fc=16
1000: FIR Kaiser Filter TAP=512、fc=60
1001: FIR Kaiser Filter TAP=512、fc=210
1010: FIR Kaiser Filter TAP=512、fc=460
1011: User Defined FIR Filter TAP=4
1100: User Defined FIR Filter TAP=64
1101: User Defined FIR Filter TAP=128
1110: User Defined FIR Filter TAP=512
1111: not used

After writing to this bit, FILTER_STAT changes to 1 (during execution).
Confirm the completion of the filter setting process by confirming that the FILTER_STAT bit returns to "0".

Note) For the combination of output rate and cutoff frequency considering avoidance of aliasing and transient response at sampling start, refer to 5.17.3 Notes on FIR filter.

Note) The factory settings are Tap: 512, Fc: 60 Hz

## 7.13. UART_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x09 | - | | | | | | BAUD_RATE | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|
| 0x08 | - | | | AUTO _START | UART _AUTO | R/W |

**bit[9:8] BAUD_RATE**

Note) The baud rate change using these **BAUD_RATE** bits become effective immediately after write access completes.

These bits specifies the Baud Rate of UART interface.
00: Reserved
01: 460.8kbps
10: 230.4kbps
11: 115.2kbps

**bit[1] AUTO_START (Only valid for UART Auto sampling)**

Enables or disables the Auto Start function.
1: Automatic Start is enabled
0: Automatic Start is disabled

When Auto Start is enabled, the device enters sampling mode and sends sampling data automatically after completing internal initialization after powered on.

Write a "1" to this **AUTO_START** bit and **UART_AUTO** bit of this register to enable this function.Then execute **FLASH_BACKUP** of GLOB_CMD [0x0A(W1)] bit [3] to preserve the current register settings.

**bit[0] UART_AUTO**

Enables or disables the UART Auto sampling function.
1: UART Auto sampling is selected
0: UART Manual sampling is selected

If UART Auto sampling is active, register values such as FLAG, temperature, and accelerations (XACCL, YACCL, ZACCL) are continuously transmitted automatically according to the data output rate set by SMPL_CTRL [0x04(W1)] register.

In UART Manual sampling, register data is transmitted as a response to a register read command.

Note) For more info on UART Auto sampling refer to 6.2.4 UART Auto Sampling Operation and 6.3 Data Packet Format. The burst output data is configured by register setting in BURST_CTRL [0x0C(W1)].

## 7.14. GLOB_CMD Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x0B | - | - | - | MESMOD _STAT | - | NOT _READY | - | - | R |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x0A | SOFT _RST | - | - | - | FLASH_ BACKUP | FLASH _RST | - | - | R/W |

**bit[12] MESMOD_STAT**

This read-only status bit shows the status of the measurement condition at a sampling mode.
1: Reduced noise floor condition
0: Standard noise floor condition

**bit[10] NOT_READY**

Indicates whether this product currently ready. Immediately after power on, this bit is “1” and becomes “0” when the product is ready. After the power on, wait until the Power-On Start-Up Time has elapsed and then wait until this bit becomes “0” before starting sensor measurement. This bit is read-only.
1: Not ready
0: Ready

**bit[7] SOFT_RST**

Write “1” to execute software reset. After the software reset is completed, the bit automatically goes back to “0”.

**bit[3] FLASH_BACKUP**

Write “1” to save the current values of the control registers with the ○ mark in the “Flash Backup” column of Table 7.1 to the non-volatile memory. After the execution is completed, the bit automatically goes back to “0”. After confirming this bit goes back to “0” and then check the result in **FLASH_BU_ERR** of DIAG_STAT [0x04(W0)] bit [0].

**bit[2] FLASH_RST**

Write “1” to resets the setting value saved in the nonvolatile memory to the factory default state. After completion of execution, it will automatically return to “0”. After confirming this bit goes back to “0” and then check the result in **FLASH_BU_ERR** of DIAG_STAT [0x04(W0)] bit [0]. The factory default state will be reflected to the registers after completing internal initialization after powered on or a reset.

## 7.15. BURST_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x0D | FLAG _OUT | TEMP _OUT | - | - | - | ACCX _OUT | ACCY _OUT | ACCZ _OUT | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x0C | - | - | - | - | - | - | COUNT _OUT | CHKSM _OUT | R/W |

These bits enable/disable the content in the output data for burst mode and UART Auto sampling.

**bit[15] FLAG_OUT**

Controls the output of FLAG status.
1: Enables output.
0: Disables output.

**bit[10] ACCX_OUT**

Controls the output of X axis acceleration / tilt angle. The output mode is selected by **OUTPUT_SEL_X** of register: SIG_CTRL [0x00 (W1)], bit [7].
1: Enables output.
0: Disables output.

**bit[9] ACCY_OUT**

Controls the output of Y axis acceleration / tilt angle. The output mode is selected by **OUTPUT_SEL_Y** of register: SIG_CTRL [0x00 (W1)], bit [6].
1: Enables output.
0: Disables output.

**bit[8] ACCZ_OUT**

Controls the output of Z axis acceleration / tilt angle. The output mode is selected by **OUTPUT_SEL_Z** of register: SIG_CTRL [0x00 (W1)], bit [5].
1: Enables output.
0: Disables output.

**bit[1] COUNT_OUT**

Controls the output of counter value.
1: Enables output.
0: Disables output.

**bit[0] CHKSM_OUT**

Controls the output of checksum.
1: Enables output.
0: Disables output.

Note) Please set "1: Enables output" to at least one bit of bit[8:10]. All outputs of acceleration / tilt angle values cannot be desabled at the same time.

## 7.16. FIR_UCMD Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x17 | - | - | - | - | - | - | - | - | - |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x16 | - | - | - | - | - | - | FIR_UCMD | | R/W |

**bit[1:0] FIR_UCMD (FIR Filter User CoMmanD)**

These bits set the control command for setting the coefficient data of the user defined FIR filter.

| | READ | WRITE |
|---|---|---|
| 00: | execution complete | do not execute |
| 01: | reading in progress | read |
| 10: | writing in progress | write |
| 11: | not used | not used |

## 7.17. FIR_UDATA Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x19 | DATA _ERR | - | - | - | - | - | - | - | R |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x18 | FIR_UDATA | | | | | | | | R/W |

**bit[15] DATA_ERR**

This bit shows the state of the read error on the coefficient data of the user defined FIR filter.
1: Read error
0: Normal operation

**bit[7:0] FIR_UDATA(FIR Filter User DATA)**

Set the coefficient data (binary) of the user defined FIR filter.

## 7.18. FIR_UADDR Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1B | FIR_UADDR _HIGH | | | | | | | | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1A | FIR_UADDR _LOW | | | | | | | | R/W |

**bit[15:8] FIR_UADDR_HIGH**

Upper address of the coefficient data of the user FIR filter.

**bit[7:0] FIR_UADDR_LOW**

Lower address of the coefficient data of the user FIR filter.

Note) This address is automatically incremented after the read / write command is executed.

Note) The setting range is from 0x0800 to 0x0FFF. It cannot be set outside the range.

## 7.19. LONGFILT_CTRL Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1D | - | - | - | - | - | - | - | - | - |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1C | - | - | - | - | - | - | FILT_SEL | FILT_EN | R/W*1 |

**bit[1] FILT_SEL**

This bit selects the type of long period filter.
1: HPF
0: LPF

**bit[0] FILT_EN**

This bit enable / disable long-period filter.
1: Enable
0: Disable

## 7.20. LONGFILT_TAP Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1F | - | - | - | - | - | - | - | - | - |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x1E | - | - | - | - | TAP_SIZE | | | | R/W |

**bit[3:0] TAP_SIZE**

These bits set the tap size of the long period filter.

0001: tap2
0010: tap4
0011: tap8
0100: tap16
0101: tap32
0110: tap64
0111: tap128
1000: tap256
1001: tap512
1010: tap1024
1011: tap2048
1100: tap4096
1101～1111: not used

## 7.21. XA_OFFSET Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x2D | XOFFSET_HIGH_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x2C | XOFFSET_HIGH_L | | | R/W |

**bit[15:0] XOFFSET_HIGH**
Sets the X-axis acceleration offset value upper word.

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x2F | XOFFSET_LOW_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x2E | XOFFSET_LOW_L | | | R/W |

**bit[15:0] XOFFSET_LOW**
Sets the X-axis acceleration offset value lower word.

## 7.22. YA_OFFSET Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x31 | YOFFSET_HIGH_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x30 | YOFFSET_HIGH_L | | | R/W |

**bit[15:0] YOFFSET_HIGH**
Sets the Y-axis acceleration offset value upper word.

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x33 | YOFFSET_LOW_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x32 | YOFFSET_LOW_L | | | R/W |

**bit[15:0] YOFFSET_LOW**
Sets the Y-axis acceleration offset value lower word.

## 7.23. ZA_OFFSET Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x35 | ZOFFSET_HIGH_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x34 | ZOFFSET_HIGH_L | | | R/W |

**bit[15:0] ZOFFSET_HIGH**

Sets the Z-axis acceleration offset value upper word.

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x37 | ZOFFSET_LOW_H | | | R/W |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x36 | ZOFFSET_LOW_L | | | R/W |

**bit[15:0] ZOFFSET_LOW**

Sets the Z-axis acceleration offset value upper word.

## 7.24. XA_ALARM Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x47 | XALARM _UP | | | | | | | | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x46 | XALARM _LO | | | | | | | | R/W |

**bit[15:8] XALARM_UP**

Sets the acceleration upper limit value to be determined by XALARM_ERR of register: FLAG [0x06 (W0)], bit [4].

Data format: 8bit, two's complement format
Setting unit: G
Setting range: -15 to +15 （can not be set to a value outside the range）

**bit[7:0] XALARM_LO**

Sets the acceleration lower limit value to be determined by XALARM_ERR of register: FLAG [0x06 (W0)], bit [4]
The setting specification is the same as XALARM_UP

## 7.25. YA_ALARM Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x49 | YALARM _UP | | | | | | | | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x48 | YALARM _LO | | | | | | | | R/W |

**bit[15:8] YALARM_UP**

Sets the acceleration upper limit value to be determined by YALARM_ERR of register: FLAG [0x06 (W0)], bit [3]

Data format: 8bit, two's complement format

Setting unit: G

Setting range: -15 to +15 （can not be set to a value outside the range）

**bit[7:0] YALARM_LO**

Sets the acceleration lower limit value to be determined by YALARM_ERR of register: FLAG [0x06 (W0)], bit [3]

The setting specification is the same as YALARM_UP

## 7.26. ZA_ALARM Register (Window 1)

| Addr (Hex) | Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x4B | ZALARM _UP | | | | | | | | R/W |

| Addr (Hex) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | R/W |
|---|---|---|---|---|---|---|---|---|---|
| 0x4A | ZALARM _LO | | | | | | | | R/W |

**bit[15:8] ZALARM_UP**

Sets the acceleration upper limit value to be determined by ZALARM_ERR of register: FLAG [0x06 (W0)], bit [2]

Data format: 8bit, two's complement format

Setting Unit: G

Setting rang: -15 to +15 （can not be set to a value outside the range）

**bit[7:0] ZALARM_LO**

Sets the acceleration lower limit value to be determined by ZALARM_ERR of register: FLAG [0x06 (W0)], bit [2]

The setting specification is the same as ZALARM_UP

## 7.27. PROD_ID Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x6A | PROD_ID1 | | | R |
| 0x6C | PROD_ID2 | | | R |
| 0x6E | PROD_ID3 | | | R |
| 0x70 | PROD_ID4 | | | R |

**bit[15:0] Product ID**

Note) These registers return the product model number of the internal sensor represented in ASCII code.

Product ID return value is A352AD10

PROD_ID1 : 0x3341
PROD_ID2 : 0x3235
PROD_ID3 : 0x4441
PROD_ID4 : 0x3031

## 7.28. VERSION Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x72 | VERSION | | | R |

**bit[15:0] Version**

This register returns the Firmware Version

## 7.29. SERIAL_NUM Register (Window 1)

| Addr (Hex) | Bit15 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x74 | SERIAL_NUM1 | | | R |
| 0x76 | SERIAL_NUM2 | | | R |
| 0x78 | SERIAL_NUM3 | | | R |
| 0x7A | SERIAL_NUM4 | | | R |

**bit[15:0] Serial Number**

Note) These registers return the serial number of the internal sensor represented in ASCII code.

For example, if the Serial Number is 01234567 then the return value is:

SERIAL_NUM1 : 0x3130
SERIAL_NUM2 : 0x3332
SERIAL_NUM3 : 0x3534
SERIAL_NUM4 : 0x3736

## 7.30. WIN_CTRL Register (Window 0,1)

| Addr (Hex) | Bit15 | . . . | Bit8 | R/W |
|---|---|---|---|---|
| 0x7F | - | | | - |

| Addr (Hex) | Bit7 | . . . | Bit0 | R/W |
|---|---|---|---|---|
| 0x7E | WINDOW_ID | | | R/W |

**bit[7:0] WINDOW_ID**

Selects the desired register window by writing the window number to this register.

0x00 :Window 0
0x01 :Window 1
0x02-0xFF: Unused

# 8. SAMPLE PROGRAM FLOW

The following describes the recommended procedures for operating this device

## 8.1. FLOW

### 8.1.1. Power-on sequence

Power-on sequence is as follows.

(a) power-on.

(b) Wait Power-On Start-Up Time.

(c) Wait until NOT_READY bit goes to 0. NOT_READY is GLOB_CMD[0x0A(W1)]'s bit[10].

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x0A,0x00,0x0d}. /* GLOB_CMD read command */

TXdata={0x0A,MSByte,LSByte,0x0d}. /* get response */

Confirm NOT_READY bit.

When NOT_READY becomes 0, it ends. Otherwise , please repeat (c).

(d) Confirm HARD_ERR bits. HARD_ERR is DIAG_STAT[0x04(W0)]'s bit[7:5].

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */

TXdata={0x04,0x00,0x0d}. /* DIAG_STAT read command */

TXdata={0x04,MSByte,LSByte,0x0d}. /* get response */

Confirm HARD_ERR is 000.

If HARD_ERR is 000, the Accelerometer is OK. Otherwise, the Accelerometer is faulty.

### 8.1.2. Register read and write

[Read Example]

To read a 16bit-data from a register(addr=0x02 / WINDOW=0).

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID write command.(WINDOW=0) */

TXdata={0x02,0x00,0x0d}. /* MODE_CTRL read command */

RXdata={0x02,0x04,0x00,0x0d} /* get response*/

0x04 in 2nd byte of RXdata is Configuration mode.

0x00 in 3rd byte of RXdata is Reserved.

Please note that read data unit is 16bit, and Most Significant Byte first.

------------------------------------------------------------

[Write Example]

To write a 8bit-data into a register(addr=0x03 / WINDOW=0).

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */

TXdata={0x83,0x01,0x0d}. /* MODE_CTRL(H) write command.(move to Sampling mode) */

There is no response at Write.

By sending this command, the Accelerometer moves to Sampling mode.

Please note that write data unit is 8bit.

### 8.1.3. Sampling data

[Sample Flow 1 (UART Auto sampling)]

Power-on sequence. Please refer to Chapter 8.1.1.

Filter setting sequence. Please refer to Chapter 8.1.9

```
TXdata={0xFE,0x01,0x0d}.    /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x85,0x04,0x0d}.    /* SMPL_CTRL(H) write command.(200Sps) */
TXdata={0x88,0x01,0x0d}.    /* UART_CTRL(L) write command.(UART Auto sampling) */
TXdata={0x8C,0x02,0x0d}.    /* BURST_CTRL(L) write command.(COUNT=on) */
TXdata={0x8D,0x47,0x0d}.    /* BURST_CTRL(H) write command.(TEMP=on, ACC_XYZ=on) */
TXdata={0xFE,0x00,0x0d}.    /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x83,0x01,0x0d}.    /* MODE_CTRL(H) write command.(move to Sampling mode) */
receive sampling data.
  (a)RXdata={0x80, TEMP_HIGH_Hi, TEMP_HIGH_Lo, TEMP_LOW_Hi, TEMP_LOW_Lo,
             XACCL_HIGH_Hi, XACCL_HIGH_Lo, XACCL_LOW_Hi, XACCL_LOW_Lo,
             YACCL_HIGH_Hi, YACCL_HIGH_Lo, YACCL_LOW_Hi, YACCL_LOW_Lo,
             ZACCL_HIGH_Hi, ZACCL_HIGH_Lo, ZACCL_LOW_Hi, ZACCL_LOW_Lo,
             COUNT_Hi, COUNT_Lo, 0x0d}
  repeat (a).
TXdata={0x83,0x02,0x0d}.    /* MODE_CTRL(H) write command.(return to Configulation mode) */
```

------------------------------------------------------------

[Sample Flow 2(UART burst mode)]

Power-on sequence. Please refer to Chapter 8.1.1

Filter setting sequence. Please refer to Chapter 8.1.9

```
TXdata={0xFE,0x01,0x0d}.    /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x85,0x04,0x0d}.    /* SMPL_CTRL(H) write command.(200Sps) */
TXdata={0x88,0x00,0x0d}.    /* UART_CTRL(L) write command.(UART Manual sampling) */
TXdata={0x8C,0x02,0x0d}.    /* BURST_CTRL(L) write command.(COUNT=on) */
TXdata={0x8D,0x47,0x0d}.    /* BURST_CTRL(H) write command.(TEMP=on, ACC_XYZ=on) */
TXdata={0xFE,0x00,0x0d}.    /* WINDOW_ID(L) write command.(WINDOW=0) */
```

TXdata={0x83,0x01,0x0d}. /* MODE_CTRL(H) write command.(move to Sampling mode) */

receive sampling data.

(a)Wait until Data Ready signal is asserted.

(b)TXdata={0x80,0x00,0x0d}. /* BURST(L) write command */

(c)RXdata={0x80, TEMP_HIGH_Hi, TEMP_HIGH_Lo, TEMP_LOW_Hi, TEMP_LOW_Lo,
XACCL_HIGH_Hi, XACCL_HIGH_Lo, XACCL_LOW_Hi, XACCL_LOW_Lo,
YACCL_HIGH_Hi, YACCL_HIGH_Lo, YACCL_LOW_Hi, YACCL_LOW_Lo,
ZACCL_HIGH_Hi, ZACCL_HIGH_Lo, ZACCL_LOW_Hi, ZACCL_LOW_Lo,
COUNT_Hi, COUNT_Lo, 0x0d}

repeat from (a) to (c).

TXdata={0x83,0x02,0x0d}. /* MODE_CTRL(H) write command.(return to Configulation mode) */

------------------------------------------------------------

[Notes]

Please note that read data unit is 16bit, and Most Significant Byte first.

Please note that write data unit is 8bit.

X ACCL_HIGH_Hi: means MSByte of ACCL_HIGH data

X ACCL_HIGH_Lo: means LSByte of ACCL_HIGH data

### 8.1.4. Self test

Selftest is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Send self test command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x83,0x04,0x0d}. /* MSC_CTRL(H) write command.(Acc Test) */

(b) Wait until selftest has finished.

Wait until ACC_TEST bit goes to 0. ACC_TEST is MSC_CTRL[0x02(W1)]'s bit[10].

TXdata={0x02,0x00,0x0d}. /* MSC_CTRL read command */

RXdata={0x02,MSByte,LSByte,0x0d}. /* get response */

Confirm ACC_TEST bit.

When ACC_TEST becomes 0, it ends. Otherwise , please repeat (b).

(c) Confirm the result.

Confirm ACC_ERR bits. ACC_ERR is DIAG_STAT[0x04(W0)]'s bit[15:12].

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x04,0x00,0x0d}. /* DIAG_STAT read command */
RXdata={0x04,MSByte,LSByte,0x0d}. /* get response */
Confirm each ACC_ERR is 0.
If each ACC_ERR is 0, the result is OK. Otherwise, the result is NG.

### 8.1.5. Flash Test

Flash test is as follows.

Power-on sequence. Please refer to Chapter 8.1.1
(a) Send flash test command.
TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x83,0x08,0x0d}. /* MSC_CTRL(H) write command.(Flash Test) */

(b) Wait until flash test has finished.
Wait until FLASH_TEST bit goes to 0. FLASH_TEST is MSC_CTRL[0x02(W1)]'s bit[11].
TXdata={0x02,0x00,0x0d}. /* MSC_CTRL read command */
RXdata={0x02,MSByte,LSByte,0x0d}. /* get response */
Confirm FLASH_TEST bit.
When FLASH_TEST becomes 0, it ends. Otherwise , please repeat (b).

(c) Confirm the result.
Confirm FLASH_ERR bits. FLASH_ERR is DIAG_STAT[0x04(W0)]'s bit[2].
TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x04,0x00,0x0d}. /* DIAG_STAT read command */
RXdata={0x04,MSByte,LSByte,0x0d}. /* get response */
Confirm FLASH_ERR is 0.
If FLASH_ERR is 0, the result is OK. Otherwise, the result is NG.

### 8.1.6. Software Reset

Software reset is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Send software reset command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x8A,0x80,0x0d}. /* GLOB_CMD(L) write command.(Software reset) */

(b) Wait Reset Recovery Time.

### 8.1.7. Flash Backup

Flash backup is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Send flash backup command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x8A,0x08,0x0d}. /* GLOB_CMD(L) write command.(Flash backup) */

(b) Wait until flash backup has finished.

Wait until FLASH_BACKUP bit goes to 0. FLASH_BACKUP is GLOB_CMD[0x0A(W1)]'s bit[3].

TXdata={0x0A,0x00,0x0d}. /* GLOB_CMD read command */

RXdata={0x0A,MSByte,LSByte,0x0d}. /* get response */

Confirm FLASH_BACKUP bit.

When FLASH_BACKUP becomes 0, it ends. Otherwise , please repeat (b).

(c) Confirm the result.

Confirm FLASH_BU_ERR bits. FLASH_BU_ERR is DIAG_STAT[0x04(W0)]'s bit[0].

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */

TXdata={0x04,0x00,0x0d}. /* DIAG_STAT read command */

RXdata={0x04,MSByte,LSByte,0x0d}. /* get response */

Confirm FLASH_BU_ERR is 0.

If FLASH_BU_ERR is 0, the result is OK. Otherwise, the result is NG.

### 8.1.8. Flash Reset

Flash Reset is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Send flash reset command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x8A,0x04,0x0d}. /* GLOB_CMD(L) write command.(Flash Reset) */

(b) Wait until flash reset has finished.

Wait until FLASH_RST bit goes to 0. FLASH_RST is GLOB_CMD[0x0A(W1)]'s bit[2].

TXdata={0x0A,0x00,0x0d}. /* GLOB_CMD read command */

RXdata={0x0A,MSByte,LSByte,0x0d}. /* get response */

Confirm FLASH_RST bit.

When FLASH_RST becomes 0, it ends. Otherwise , please repeat (b).

(c) Confirm the result.

Confirm FLASH_BU_ERR bits. FLASH_BU_ERR is DIAG_STAT[0x04(W0)]'s bit[0].

TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */

TXdata={0x04,0x00,0x0d}. /* DIAG_STAT read command */

RXdata={0x04,MSByte,LSByte,0x0d}. /* get response */

Confirm FLASH_BU_ERR is 0.

If FLASH_BU_ERR is 0, the result is OK. Otherwise, the result is NG.

(d) Power off and on , or reset.

#### 8.1.9. Filter setting

Filter setting is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Send filter setting command for FIR kaiser filter (TAP512, fc60).

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x86,0x08,0x0d}. /* FILTER_CTRL(L) write command.(Filter setting) */

(b) Wait until filter setting has finished.

Wait until FILTER_STAT bit goes to 0. FILTER_STAT is FILTER_CTRL[0x06(W1)]'s bit[5].

TXdata={0x06,0x00,0x0d}. /* FILTER_CTRL read command */

RXdata={0x06,MSByte,LSByte,0x0d}. /* get response */

Confirm FILTER_STAT bit.

When FILTER_STAT becomes 0, it ends. Otherwise , please repeat (b).

### 8.1.10. User Defined FIR Filter coefficients setting

User Defined FIR Filter coefficients setting is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

[Write Sequence]

(a) Send filter coefficient address command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

First, set the start address (0x0800),

TXdata={0x9B,0x08,0x0d}. /* FIR_UADDR(H) write command.(Address High Byte) */

TXdata={0x9A,0x00,0x0d}. /* FIR_UADDR(L) write command.(Address Low Byte) */

(b) Send filter coefficient data command.

For example, if the coefficient data is 0x1C19D153, send in order from the lower byte(0x53).

TXdata={0x98,0x53,0x0d}. /* FIR_UDATA(L) write command */

(c) Send filter coefficient control command.

TXdata={0x96,0x02,0x0d}. /* FIR_UCMD(L) write command (Wirte execution)*/

(d) Wait until Write execution has finished.

Wait until FIR_UCMD bit goes to 00. FIR_UCMD is FIR_UCMD[0x16(W1)]'s bit[1:0].

TXdata={0x16,0x00,0x0d}. /* FIR_UCMD read command */

RXdata={0x16,MSByte,LSByte,0x0d}. /* get response */

Confirm FIR_UCMD bit.

When FIR_UCMD becomes 00, it ends. Otherwise , please repeat (d).

(e) Repeat from (b) to (d) until sending all coefficients.

(f) Send filter setting command for User Defined FIR Filter. Please refer to Chapter 8.1.9

notes

The coefficient data unit is 32bit, and little-endian format.

After the byte has completed writing, the address is automatically incremented by 1.

[Read Sequence]

(a) Send filter coefficient address command.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

First, set the start address (0x0800),

TXdata={0x9B,0x08,0x0d}. /* FIR_UADDR(H) write command.(Address High Byte) */

TXdata={0x9A,0x00,0x0d}. /* FIR_UADDR(L) write command.(Address Low Byte) */

(b) Send filter coefficient control command.

TXdata={0x96,0x01,0x0d}. /* FIR_UCMD(L) write command (Read execution)*/

(c) Wait until Read execution has finished.

Wait until FIR_UCMD bit goes to 00. FIR_UCMD is FIR_UCMD[0x16(W1)]'s bit[1:0].

TXdata={0x16,0x00,0x0d}. /* FIR_UCMD read command */

RXdata={0x16,MSByte,LSByte,0x0d}. /* get response */

Confirm FIR_UCMD bit.

When FIR_UCMD becomes 00, it ends. Otherwise , please repeat (c).

(d) Send filter coefficient data command.

TXdata={0x18,0x00,0x0d}. /* FIR_UDATA read command */

RXdata={0x18,MSByte,LSByte,0x0d}. /* get response */

(e) Repeat from (b) to (d) until reading all coefficients.

notes

The coefficient data unit is 32bit, and little-endian format.

After the byte has completed reading, the address is automatically incremented by 1.

### 8.1.11. Auto Start

Auto Start is as follows.

Power-on sequence. Please refer to Chapter 8.1.1

(a) Set registers.

TXdata={0xFE,0x01,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=1) */

TXdata={0x85,0x04,0x0d}. /* SMPL_CTRL(H) write command.(200Sps) */

TXdata={0x86,0x08,0x0d}. /* FILTER_CTRL(L) write command.(Filter setting TAP=512 fc60) */

TXdata={0x88,0x03,0x0d}. /* UART_CTRL(L) write command.(UART Auto sampling, Auto start=on) */

TXdata={0x8C,0x02,0x0d}. /* BURST_CTRL(L) write command.(COUNT=on) */

TXdata={0x8D,0x47,0x0d}. /* BURST_CTRL(H) write command.(TEMP=on, ACC_XYZ=on) */

(b) Execute Flash backup. Please refer to Chapter 8.2.7.

(c) power-off.

(d) power-on.

(e) Wait Power-On Start-Up Time.

(f) receive sampling data.

(i) Wait until Data Ready signal is asserted.

(ii) RXdata={0x80, TEMP_HIGH_Hi, TEMP_HIGH_Lo, TEMP_LOW_Hi, TEMP_LOW_Lo,
XACCL_HIGH_Hi, XACCL_HIGH_Lo, XACCL_LOW_Hi, XACCL_LOW_Lo,
YACCL_HIGH_Hi, YACCL_HIGH_Lo, YACCL_LOW_Hi, YACCL_LOW_Lo,
ZACCL_HIGH_Hi, ZACCL_HIGH_Lo, ZACCL_LOW_Hi, ZACCL_LOW_Lo,
COUNT_Hi, COUNT_Lo, 0x0d}

repeat from (i) to (ii).

(g) If you want to stop sampling,

TXdata={0x83,0x02,0x0d}. /* MODE_CTRL(H) write command.(return to Configulation mode) */

# 9. HANDLING NOTES

## 9.1. CAUTIONS FOR ATTACHING

- The product contains quartz crystal oscillator created by microfabrication. Take precaution to prevent falling or excessive impact. Do not use the product after an accidental fall or it experiences excessive impact. The possibility of a failure and risk of malfunction from failure increases.
- Excessive vibration, shock, continuous stress, or sudden temperature change may increase the possibility of failure.
- Please consult us before the unit is used in an environment where there is acute vibration out of the measurement band.
- The product should be kept powered on for more than 15 minutes to measure with highest precision and accuracy.
- Do not connect the product to the network with the supply voltage turned on.
- When attaching the product, ensure that the product is properly mounted to avoid mechanical stress such as warping or twisting. In addition, ensure appropriate torque is applied when tightening the screws but not too excessive to cause the mount of the product to deform or break. Use screw locking techniques as necessary.
- When setting up the product, ensure that the equipment, jigs, tools, and workers maintain a good ground in order not to generate high voltage leakage. Applying over current or static electricity to the product may be damage the product permanently.
- When installing the product, ensure that metallic or other conductive material do not enter the product. Otherwise, malfunction or damage of the product may result.
- If excessive shock is applied to the product when, for example, the product falls, the quality of the product may be degraded. Ensure that the product does not fall when you handle it.
- Before you start using the product to obtain measurements, test it in the actual equipment under the actual operating environment to confirm proper operation.
- When connecting a cable to this product, tighten the screw enough after inserting it completely. This product may not satisfy IP67 if tightening is insufficient.
- Do not use the product in a situation where power is always applied to the joint of connector.
- Ensure that the signals are wired correctly with attention to the name and the polarity of each signal.
- Since the product has capacitors inside, Inrush current occurs immediately after power-on. Evaluate in the actual environment in order to check the effect of the supply voltage sag caused by inrush current in the system.

## 9.2. OTHER CAUTIONS

- This product is water-proof and dust-proof in conformity with IP67. We do not guarantee the operation of the product when the product is exposed to condensation, dust, oil, corrosive gas (salt, acid, alkaline, etc), or direct sunlight which surpass IP67. Do not use this product under water.
- This product is not designed to be radiation resistant.
- Never use this product if the operating condition is over the absolute maximum rating. Otherwise, permanent damage to the product may result.
- If the product is exposed to excessive external noise or other similar conditions, degradation of the precision, malfunction, or damage to the product may result. The system needs to be designed so that the noise itself is suppressed or the system is immune to the noise.
- This product is not designed to be used in equipment that demands extremely high reliability and where its failure may threaten human life or property (for example, aerospace equipment, submarine repeater, nuclear power control equipment, life support equipment, medical equipment, transportation control equipment, etc.). Seiko Epson Corporation will not be liable for any damages caused by the use of the product for those applications.
- Do not apply shock or vibration to the packing box. Do not spill water over the packing box. Do not store or use the product in an environment where dew condensation occurs due to rapid temperature change.
- Do not put mechanical stress on the product while it is stored.
- Do not alter or disassemble the product.

- Do not use in water except if it gets temporarily wet based on IP67. This product does not achieve the sufficient waterproof performance if the the connector is mated incorrectly or that the mating connector does not satisfy IP67.
- The power supply to this product must satisfy the voltage rating within 2 seconds after it is turned on.
- Do not use thinner or similar liquids on this product. When cleaning this product, alcohol may be used.
- It is recommended that this product be installed horizontally (±5deg.) for normal use.

## 9.3. LIMITED WARRANTY

- The product warranty period is one year from the date of shipment.
- If a defect due to a quality failure of the product is found during the warranty period, we will promptly provide a replacement.

# 10. PART NUMBER / ORDERING INFO.

The following is the ordering code for the product:

| Product Name | Model Name | Product Number | Comment |
|---|---|---|---|
| Accelerometer for RS-422 Interface | M-A552AR10 | X2F000031000300 | |

# 11. STANDARDS AND APPROVALS

The following standards are applied only to the unit that are so labeled. (EMC is tested using the EPSON power supplies)

Europe : CE marking

## 11.1. NOTICE

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
The connection of a non-shielded interface cable to this product will invalidate the EMC standards of the device.
Any changes or modifications not expressly approved by Seiko Epson Corporation could void your authority to operate the equipment.

## 11.2. CE MARKING

This product conforms to the following Directives and Norms,

EN61326-1 Class A
EN50581

## 11.3. RoHS & WEEE

The crossed out wheeled bin label that can be found on your product indicates that this product should not be disposed of via the normal household waste stream. To prevent possible harm to the environment or human health please separate this product from other waste streams to ensure that it can be recycled in an environmentally sound manner. For more details on available collection facilities please contact your local government office or the retailer where you purchased this product.

AEEE Yönetmeliğine Uygundur.

Обладнання відповідає вимогам Технічного регламенту обмеження використання деяких небезпечних речовин в електричному та електронному обладнанні

## 11.4. FCC COMPLIANCE STATEMENT FOR AMERICAN USERS

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

## 11.5. INDUSTRY ICES COMPLIANCE STATEMENT FOR CANADIAN USERS

CAN ICES-3(A)/NMB-3(A)

# 12. REVISION HISTORY

Attachment-1

| Rev. No. | Date | Page | Category | Contents |
|---|---|---|---|---|
| Rev. 20191015 | 2019/10/15 | All | New | Preliminary |
| Rev.20200930 | 2020/09/30 | - | Revised | Change contact of International sales operations (End of book) |
| Rev.20220401 | 2022/4/1 | 64 | Modify | Product Number Change |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## AMERICA

**EPSON AMERICA, INC.**
Headquarter:
3131 Katella Ave.
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

San Jose Office:
2860 Zanker Road, Suite 204,
San Jose, CA 95134, U.S.A
Phone: +1-800-463-7766

## EUROPE

**EPSON EUROPE ELECTRONICS GmbH**
Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

## ASIA

**EPSON (CHINA) CO., LTD.**
4F,Tower 1 of China Central Place,81 Jianguo Street, Chaoyang District, Beijing 100025 CHINA
Phone: +86-400-810-9972 X ext.2
Mail EPSON_MSM@ecc.epson.com.cm

**EPSON SINGAPORE PTE. LTD.**
438B Alexandra Road, Block B Alexandra TechnoPark,
#04-01/04, Singapore 119968
Phone: +65-6586-5500 FAX: +65-6271-3182

**EPSON TAIWAN TECHNOLOGY & TRADING LTD.**
15F, No.100, Song Ren Road, Taipei 110, TAIWAN
Phone: +886-2-8786-6688 Fax: +886-2-8786-6660

**EPSON KOREA Co., Ltd.**
10F Posco Tower Yeoksam, Teheranro 134 Gangnam-gu,
Seoul, 06235 KOREA
Phone: +82-2-558-4270 Fax: +82-2-3420-6699

## JAPAN

**SEIKO EPSON CORPORATION.**
**MD SALES & MARKETING DEPT.**
JR Shinjuku Miraina Tower, 4-1-6 Shinjuku, Shinjuku-ku, Tokyo,
160-8801, Japan
Phone: +81-3-6682-4322 FAX: +81-3-6682-5016

Document Code : 3Z60-0066-51E
Revised date, APR 2022