EXCEED YOUR VISION

Accelerometer

M-A352AD10
Data Sheet

SEIKO EPSON CORPORATION

Rev.20220401

Notice of the Document

NOTICE：PLEASE READ CAREFULLY BELOW BEFORE THE USE OF THIS DOCUMENT ©Seiko Epson Corporation 2022

The content of this document is subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purposes, in whole or in part, without the consent of Seiko Epson Corporation(“Epson”).
2. Before purchasing or using Epson products, please contact with our sales representative for the latest information and be always sure to check the latest information published on Epson’s official web sites and sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purpose only. Please use the application circuits, programs, usage, etc. in the design of your equipment or systems at your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties’ intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. In using Epson products, you shall be responsible for safe design in your products; your hardware, software and systems are designed enough to prevent any harm or damages to life, health or property even if any malfunction or failure might be caused by Epson products. In designing of your products with using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson’s web site, etc.). When using the information included in the above materials such as product data, chart, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both in stand-alone basis and within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incurred by due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) and applications individually listed in this document (“General Purpose”). Epson products are NOT intended for any use beyond the General Purpose that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to harm to life, health or serious property damage or severe impact on society, including, but not limited to listed below. Therefore, you are advised to use Epson products only for the General Purpose. Should you desire to buy and use Epson products for the particular purpose other than the General Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any particular purpose.

   【Particular purpose】

   Space equipment (artificial satellites, rockets, etc.)
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
   Medical equipment (other than applications individually listed in this document) / Relay equipment to be placed on sea floor
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment
   Other applications requiring similar levels of reliability as the above
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing military weapons of mass destruction, military purpose use, or any other military applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

2022.08

.

Table of Contents

1. General Description

The M-A352 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-A352 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).

Features

- Ultra-low noise：0.2μG/√Hz typ.
- Selectable output format: Acceleration / Tilt Angle
- Selectable interface: SPI / UART
- Programmable low-pass digital filters
- Low jitter external trigger function for synchronous sampling
- Solid metallic case (Aluminum, size : 48mm x 24mm x 16mm, weight: 25g)

Applications

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

Functional Block Diagram

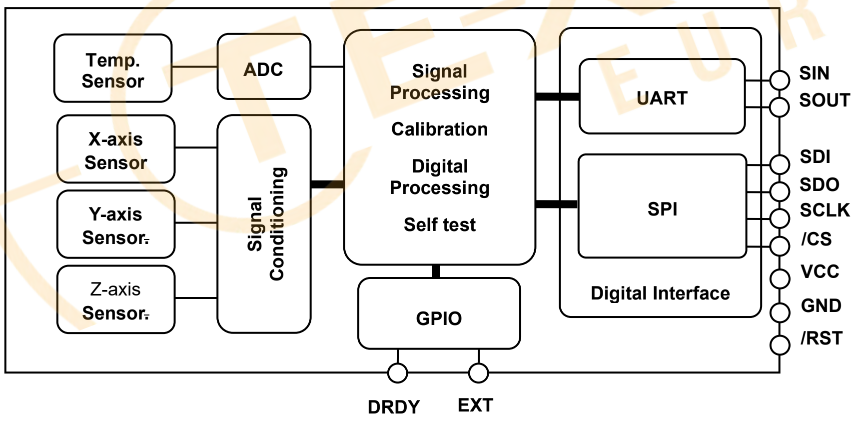


Figure 1.1 Block Diagram

2. Specifications

2.1 Absolute Maximum Ratings

Table 2.1 Absolute Maximum Rating

Parameter	Min	Typ	Max	Unit
VCC to GND	−0.3		3.6	V
Digital Input Voltage to GND	−0.3		VCC+0.3	V
Digital Output Voltage to GND	−0.3		VCC+0.3	V
Storage Temperature Range	−40		85	°C
Acceleration (Half-sine 0.2msec)			1,000	G

[*1] Precautions concerning ESD.
Electrostatic discharge (ESD) may damage this product.
Please take appropriate measures against electrostatic discharge (ESD) when storing and handling this product.
Damage by electrostatic discharge (ESD) can cause very small performance deterioration, partial malfunction, or complete breakdown.
This is a high precision product and may not conform to specification even with very small performance degradation due to improper usage or handling.

2.2 Recommended Operating Condition

Table 2.2 Recommended Operating Conditions

Parameter	Condition	Min	Typ	Max	Unit
VCC to GND		3.15	3.3	3.45	V
Digital Input Voltage to GND		GND		VCC	V
Digital Output Voltage to GND		−0.3		VCC +0.3	V
Operating Temperature Range		-30		85	°C
Start up Time	Power-on to start output			900	ms.
	Warm-up period for best performance. Bias stabilization against thermal shock = enable		5		min.
	Warm-up period for best performance. Bias stabilization against thermal shock = disable		15		min

2.3 Performance & Electrical Specifications

Table 2.3 Sensor Specification

Condition: T_A=-30°C to +85°C, VCC=3.15V~3.45V, ≤±1G, Normal Operation Mode, unless otherwise noted.

Parameter	Test Conditions / Comments	Min	Typ	Max	Unit
ACCELERATION					
Sensitivity					
Output Range	f = DC ~ 460Hz			±15	G
Scale Factor	2^{-24}G/LSB		0.06		µG/LSB
Sensitivity Error	25°C, ≤ 1G		±500		$\times 10^{-6}$ (ppm)
Nonlinearity	≤ 1G, Best fit straight line, RT			±0.03	% of FS
Cross Axis Sensitivity			±0.2		%
Misalignment	25°C			±0.1	Deg
Bias					
Initial Error	25°C			±2	mG
Bias Repeatability	TA=25°C and VCC=3.3V for one year after shipment		3		mG
Bias Temperature Error	25°C			±2	mG
Temperature Sensitivity			±0.1		mG/°C
Bias Instability	Allan variance, Average		0.2		µG
Velocity Random Walk	Average		1.2E-4		(m/sec)/√hr
Noise					
Noise Density	25°C, Avg, f = 0.5Hz ~ 6Hz		0.2	0.7	µG/√Hz, rms
Cantilever Resonance Frequency[*1]	25°C, VCC3.3V		850		Hz
VRC	at 50Hz, 25°C, VCC3.3V			±50	µG/G^2
Frequency Property					
-6 dB Bandwidth	User selectable	9		460	Hz
TILT ANGLE					
Sensitivity					
Dynamic Range	f = DC ~ 460Hz			±1.0472 (±60)	rad (deg)
Scale Factor	2^{-29}rad/LSB		0.002		µrad/LSB
Nonlinearity	25°C, ±45deg			±0.03	% of FS
Misalignment	25°C			±1.745 (±0.1)	mrad (deg)
Bias					
Bias Repeatability	TA=25°C and VCC=3.3V for one year after shipment		±3 (±0.17)		mrad (deg)
Bias Temperature Error	25°C			±2 (±0.11)	mrad (deg)
Noise					
Noise Density	25°C, Avg, f = 0.5Hz ~ 6Hz		0.2	0.7	µrad/√Hz, rms
TEMPERATURE SENSOR					
Output Range		-30		85	°C
Scale Factor [*2]	Output=2634(0x0A4A) at 25°C		-0.0037918		°C/LSB
RELIABILITY					
MTBF[*3]	JIS-C5003 TA=25℃	87,600			hour

[*1] Please make sure that a vibration on this product around the resonance frequency does not exceed 100 mG. Please take an appropriate action (e.g. installing a damper mechanism) if it exceeds 100 mG.

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

[*3] The MTBF is an estimated value derived from the result of high temperature operation with a system requirement of TA=25℃ and a 60% reliability level.

Note) The values in the specifications are based on the data calibrated at the factory. The values may change according to the way the product is used.

Note) The Max/Min value is the maximum/minimum value of the design or factory shipment examination, unless otherwise specified.
Note) The calibrated standard 1G gravitational acceleration value is 9.80665 m/s^2

Table 2.4 Interface Specification T_A=25°C, VCC=3.3V, unless otherwise noted

Parameter	Test Conditions	Min	Typ	Max	Unit
LOGIC INPUTS*1					
Positive Trigger Voltage	Schmitt	1.37		2.29	V
Negative Trigger Voltage	Schmitt	0.69		1.24	V
Hysteresis Voltage	Schmitt	0.53			V
Input Current, Ii	VI=Vcc or GND		0.5		µA
Input Capacitance, Ci			2.5		pF
RST Low Pulse Width		100			ms
Pull-up resistor			220		kΩ
Ext.Trigger Input Width, t_{ETW}		1			µs
Ext.Trigger Input Cycle, t_{ETC}		1		20	ms
Ext.Trigger Jitter, t_{ETJ}	Ext.Trigger input to resampling's completion	0		5	µs
Ext.Trigger Delay Time*3, t_{ETD}	Ext.Trigger input to DRDY asserted Long-period filter is disable			740	µs
Internal Timer Delay Time*3, t_{ITD}	Internal Timer input to DRDY asserted Long Period Filter = disable			430	µs
DIGITAL OUTPUTS*1					
Output High Voltage, VOH	ISOURCE=20µA	VCC-0.1			V
Output Low Voltage, VOL	ISINK=20µA			0.1	V
FUNCTIONAL TIMES*2	Time until data is available				
Power-On Start-Up Time				900	ms
Reset Recovery Time				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 / axis		10	40	s
	Flash Test			5	ms
Filter Setting Time	Built-In FIR Filter			4	ms
	User FIR Filter			100	ms
User Filter Write Cycle, t_{UWC}				7	ms
User Filter Read Cycle, t_{URC}				500	us
Sleep Wake-up Time, t_{WakeUp}				16	ms
OUTPUT DATA RATE		50		1,000	Sps
Clock Accuracy				±0.001	%
POWER SUPPLY	Operating voltage range, VCC	3.15	3.3	3.45	V
Power Supply Current	Standard noise floor condition, 200Sps, Average		13.2	18.0	mA
	Reduced noise floor condition, 200Sps, Average		16.2	20.0	mA
	Sleep mode		1.3	2.0	mA

Note) These parameters are not included in the factory test items but these characteristics are confirmed.

*1) Digital I/O signal pins operate at 3.3V inside the unit. All digital I/O signal pins (except RST) can tolerate 5V input.

*2) These specifications do not include the effect of temperature fluctuation and response time of the internal filter.

*3) It is not included the group delay of the built-in filter.

2.4 Timing Specifications

Table 2.5 Timing Specification T_A=25°C, VCC=3.3V, unless otherwise noted

Parameter	Description	Min	Typ	Max	Unit
NORMAL MODE					
fSCLK		0.01		2.0	MHz
tSTALL	Stall period between data	20			µs
tWRITERATE	Write rate	40			µs
tREADRATE	Read rate	40			µs
BURST MODE					
fSCLK		0.01		2.0	MHz
tSTALL1	Stall period between data	45			µs
tSTALL2	Stall period between data	0			µs
tREADRATE2	Read rate	8			µs
COMMON					
tCS	Chip select to clock edge	10			ns
tDAV	SO valid after SCLK edge			80	ns
tDSU	SI setup time before SCLK rising edge	10			ns
tDHD	SI hold time after SCLK rising edge	10			ns
tSCLKR, tSCLKF	SCLK rise/fall times			20	ns
tDF, tDR	SO rise/fall times			20	ns
tSFS	high after SCLK edge CS	80			ns

Note) These parameters are not included in the factory test items but these characteristics are confirmed.

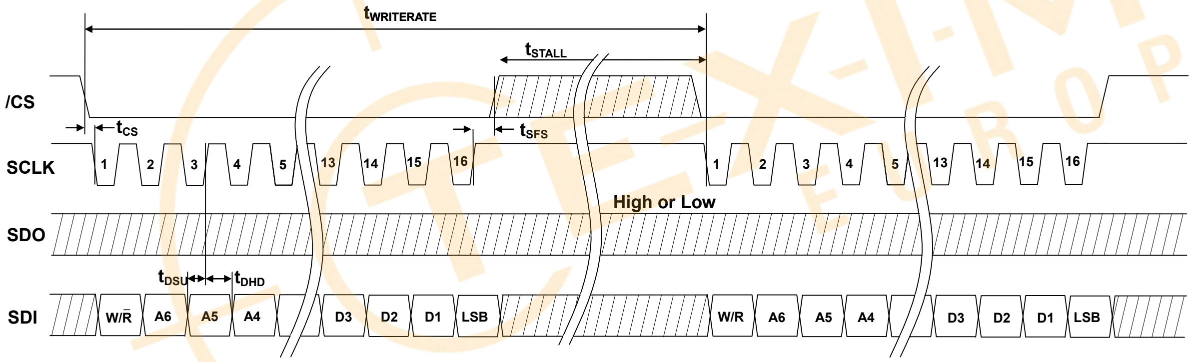


Figure 2.1 SPI Write Timing and Sequence

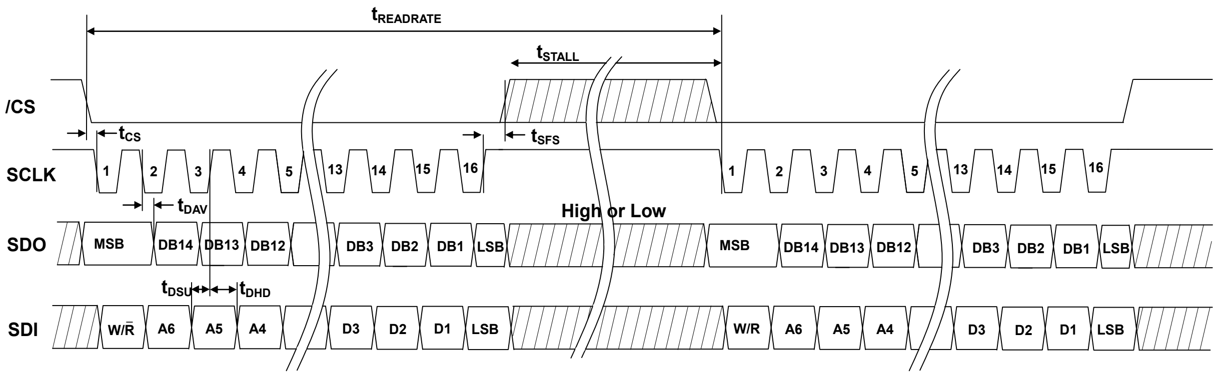


Figure 2.2 SPI Read Timing and Sequence

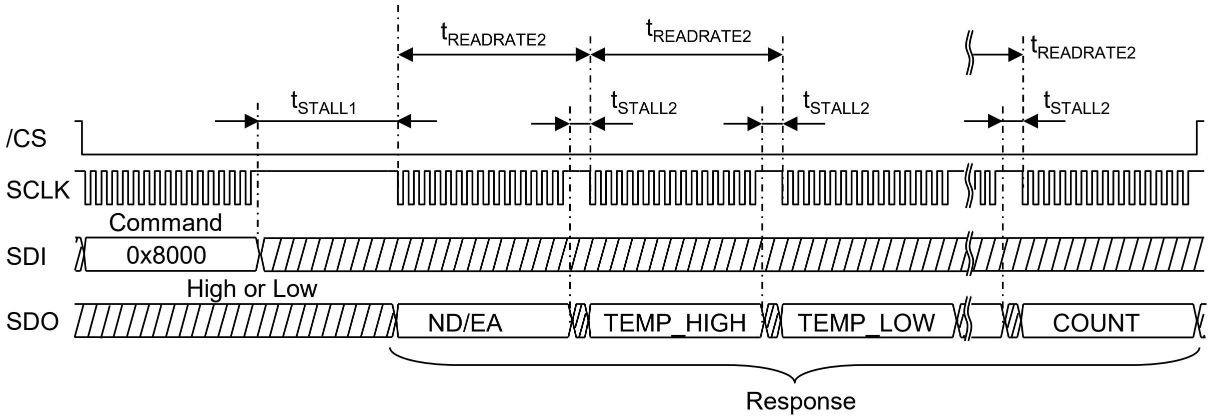


Figure 2.3 SPI Read Timing and Sequence (BURST MODE)

2.5 Socket Pin Layout and Functions

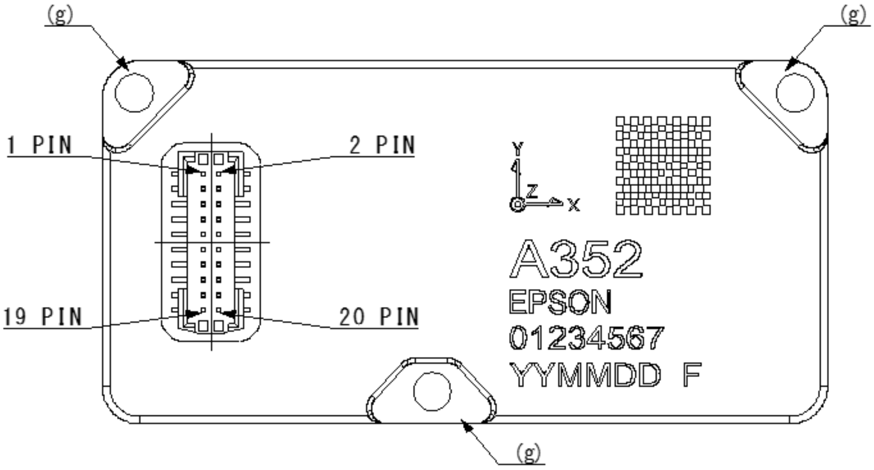


Figure 2.4 Socket Pin Assignment

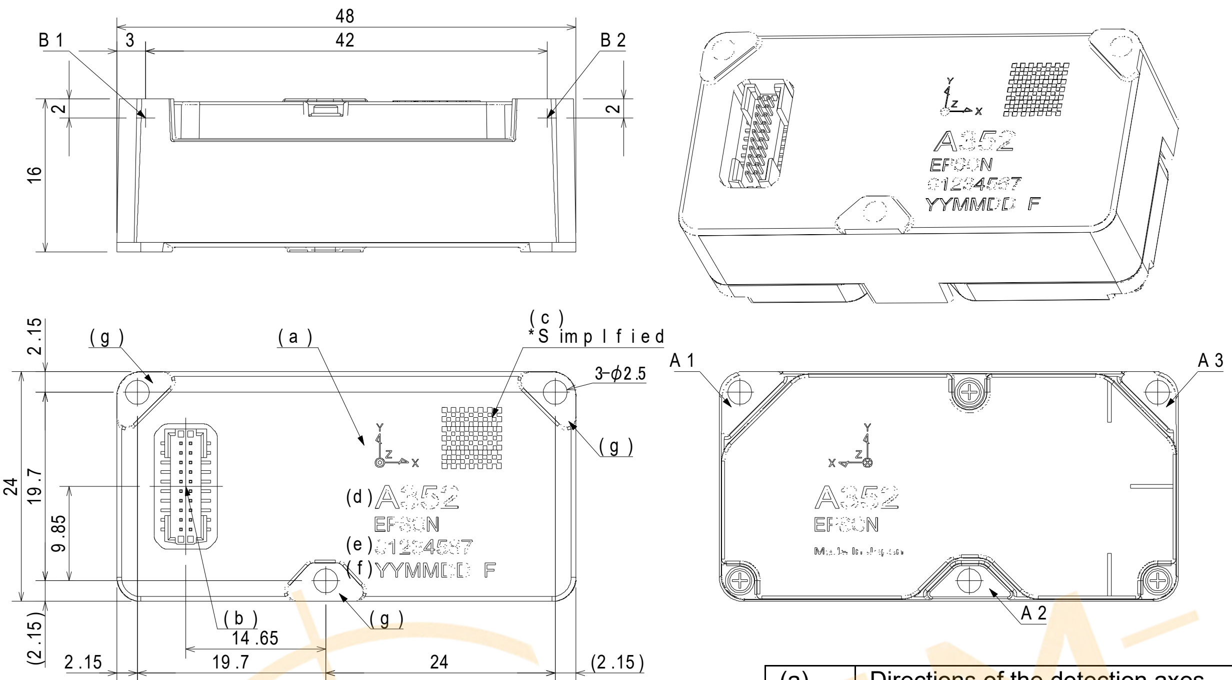
Table 2.6 Pin Function Descriptions

Pin No.	Mnemonic	Type*1	Description
1	SCLK	I	SPI Serial Clock *2
2	SDO	O	SPI Data Output *2
5	SDI	I	SPI Data Input *2
6	/CS	I	SPI Chip Select *2
7	SOUT	O	UART Data Output *2
9	SIN	I	UART Data Input *2
13	DRDY	O	Data Ready *3
14	EXT	I	External Trigger Input *4 (Sleep Wakeup Input)
16	/RST	I	Reset *5
10,11,12	VCC	S	Power Supply 3.3V
3,4,8,15	GND	S	Ground *6
17,18,19,20	NC	N/A	Do Not Connect

*1) Pin Type I :Input, O :Output, I/O :Input/Output, S :Supply, N/A :Not Applicable
*2) Please connect either SPI or UART. Connecting both SPI and UART at the same time may cause malfunction.
Please connect unused input pins to VCC via a resistor.
*3) Please refer to DRDY_ON of register: MSC_CTRL [0x02 (W1)], bit [2] for pin function selection.
*4) Please refer to EXT_SEL of register: MSC_CTRL [0x02 (W1)], bit [6] for pin function selection.
*5) When RST pin is not used, fix it to High (VCC) level via a resistor.
*6) Please connect (g) Frame Ground to any GND pin (No.3, 4, 8, 15).
Note) All input pins are weak pull-up inside this product.

3. Mechanical Dimensions

3.1 Outline Dimensions



(a)	Directions of the detection axes
(b)	Connector Position
(c)	Matrix Code (DataMatrix) Including Product Name & S/N & Date & Factory Code
(d)	Product Name
(e)	Serial Number
(f)	Date & Factory Code
(g)	Frame Ground
(x)	X Axis Sensor Position
(y)	Y Axis Sensor Position
(z)	Z Axis Sensor Position

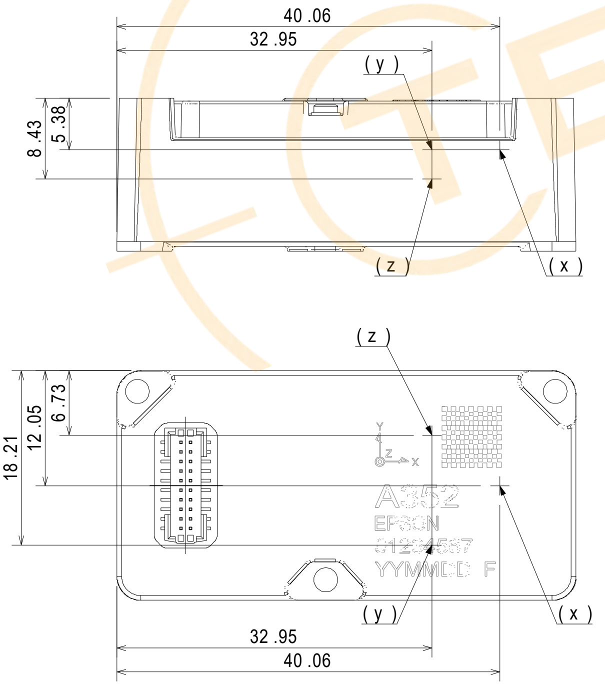


Table 3.1 Header Part Number

Maker	Parts Number	RoHS Compliant
Samtec	FTMH-110-02-L-DV-ES	Yes

Table 3.2 Recommended socket parts at the host side

Maker	Parts Number	RoHS Compliant
Samtec	CLM-110-02-H-D	Yes
Samtec	CLM-110-02-L-D	Yes

Figure 3.1 Outline Dimensions (millimeters)

*1) This product is calibrated based on the surfaces A1, A2, A3, and B1, B2.
*2) In order to demonstrate the performance of the product properly, please fix surfaces A1, A2, A3 to rugged parts with M2 screw.
*3) When high connection reliability is required, please tighten this product together with the board on which the connector is mounted.

4. Typical Performance Characteristics

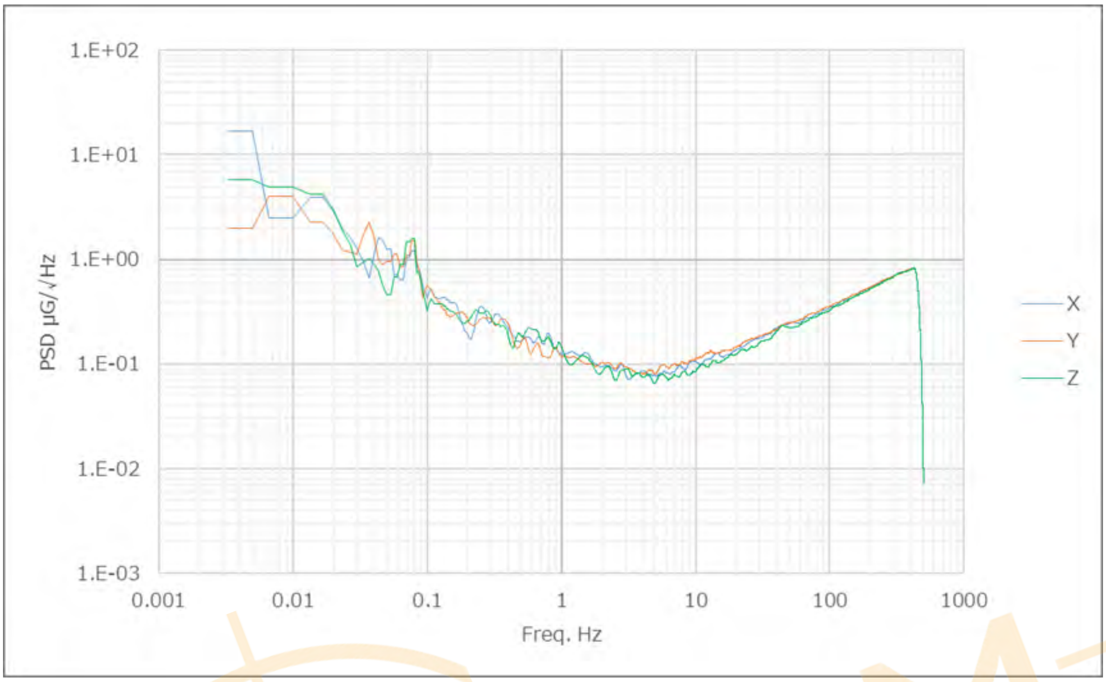


Figure 4.1 Noise Density Characteristic of Accelerometer (Standard Noise Floor Condition)

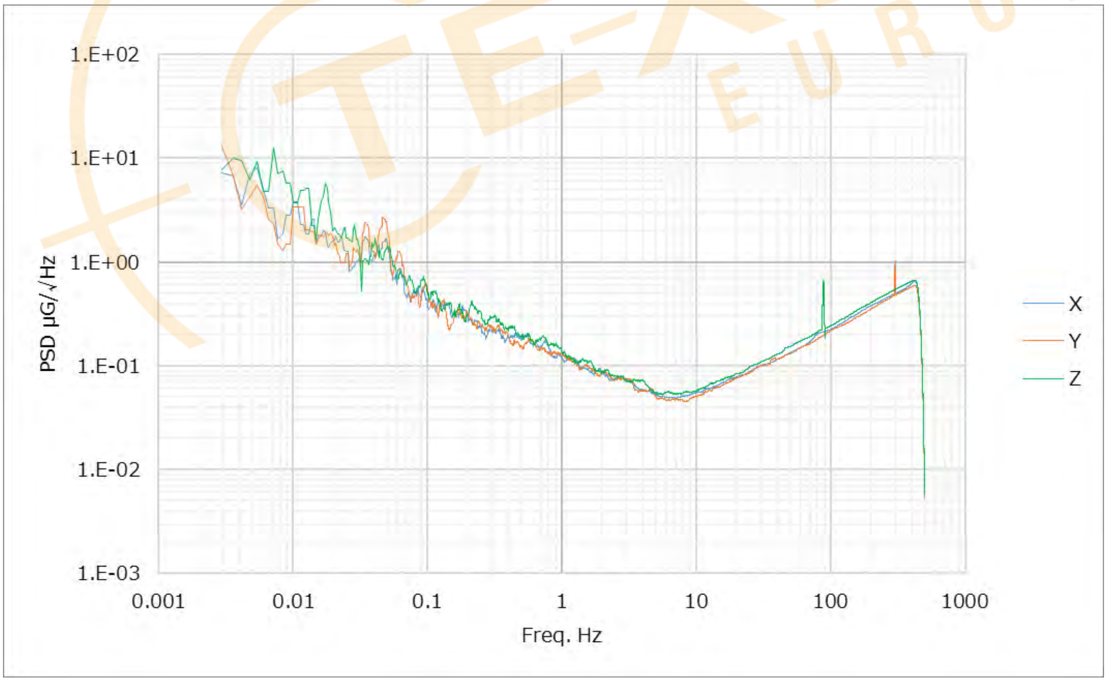


Figure 4.2 Noise Density Characteristic of Accelerometer (Reduced Noise Floor Condition)

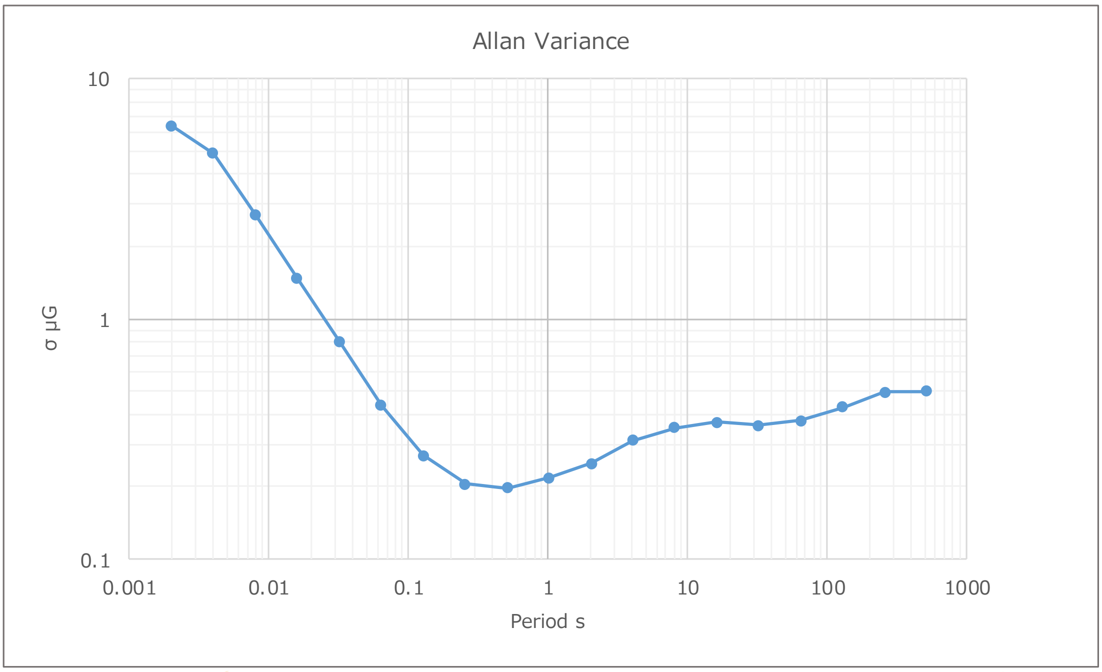


Figure 4.3 Allan Variance Characteristic of Accelerometer(Standard Noise Floor Condition)

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

5. Basic Operation

5.1 Connection To Host

The device supports two types of serial interface:UART and SPI. Only one interface type should be selected and used at any given time (not both). The example wiring connection is provided below as a reference.

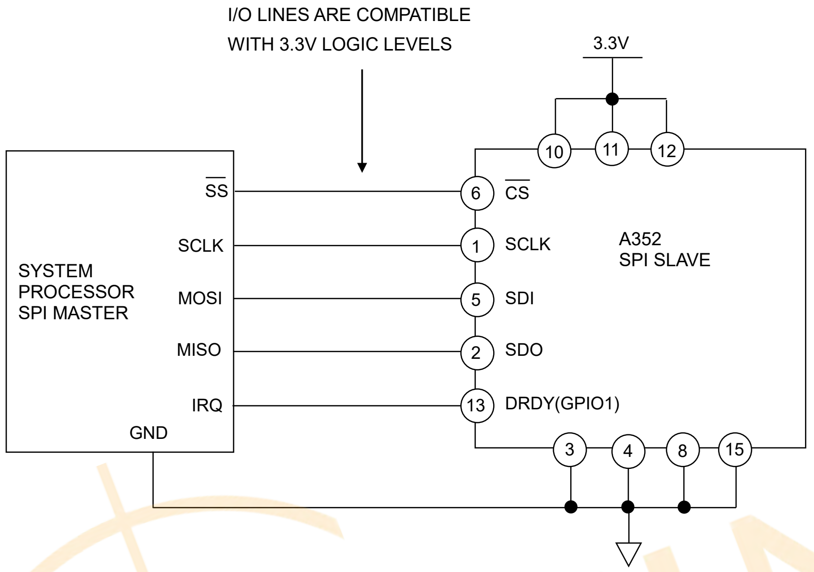


Figure 5.1 SPI Connection

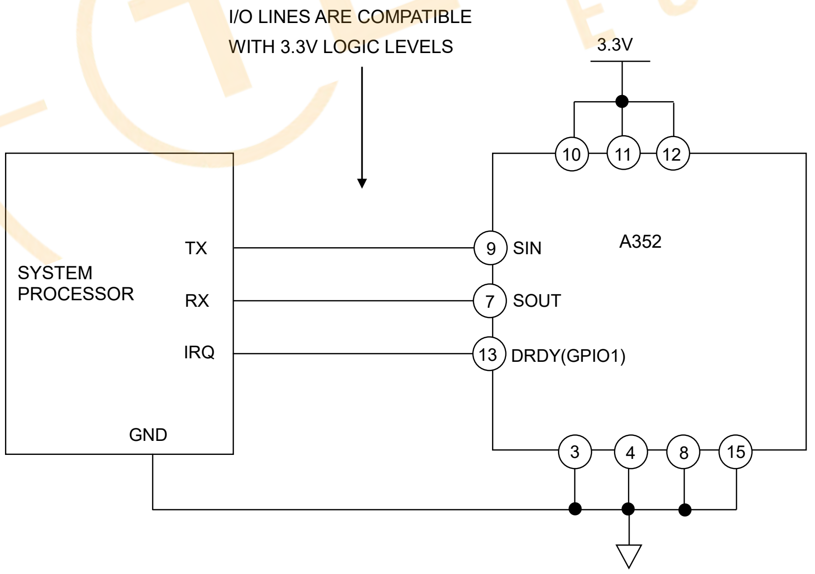


Figure 5.2 UART Connection

The device performance may be affected by signal overshoot or undershoot of the host interface. Care must be taken minimize signal integrity issues when designing the electrical interface so that the noise level is reduced as much as possible to within the tolerance of the communication timing specification.

5.2 Operation Mode

The following three operational modes are available in the device.

(1) Configuration mode
(2) Sampling mode

Sampling condition
・Manual sampling
・Auto sampling (UART Only)

Measurement condition
・Standard noise floor
・Reduced noise floor

(3) Sleep mode

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 (only available with a UART connection). These conditions can be switched between Manual sampling and Auto sampling by **UART_AUTO**, and between a Standard noise floor and a 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 for the UART version 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. When shifting to the sleep mode, the internal circuit operation stops and the current consumption during standby can be reduced. The return time from sleep mode can be shorter than the initialization time from startup. The device can wake up from sleep mode by detecting an edge trigger on the EXT pin.

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.

When the UART interface is used, writing to **UART_AUTO** (UART_CTRL[0x08(W1)] bit[0]) can switch between the Manual sampling and the Auto sampling(*3). When SPI interface is used, Manual sampling must be selected. Otherwise, the device does not work properly.

*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.

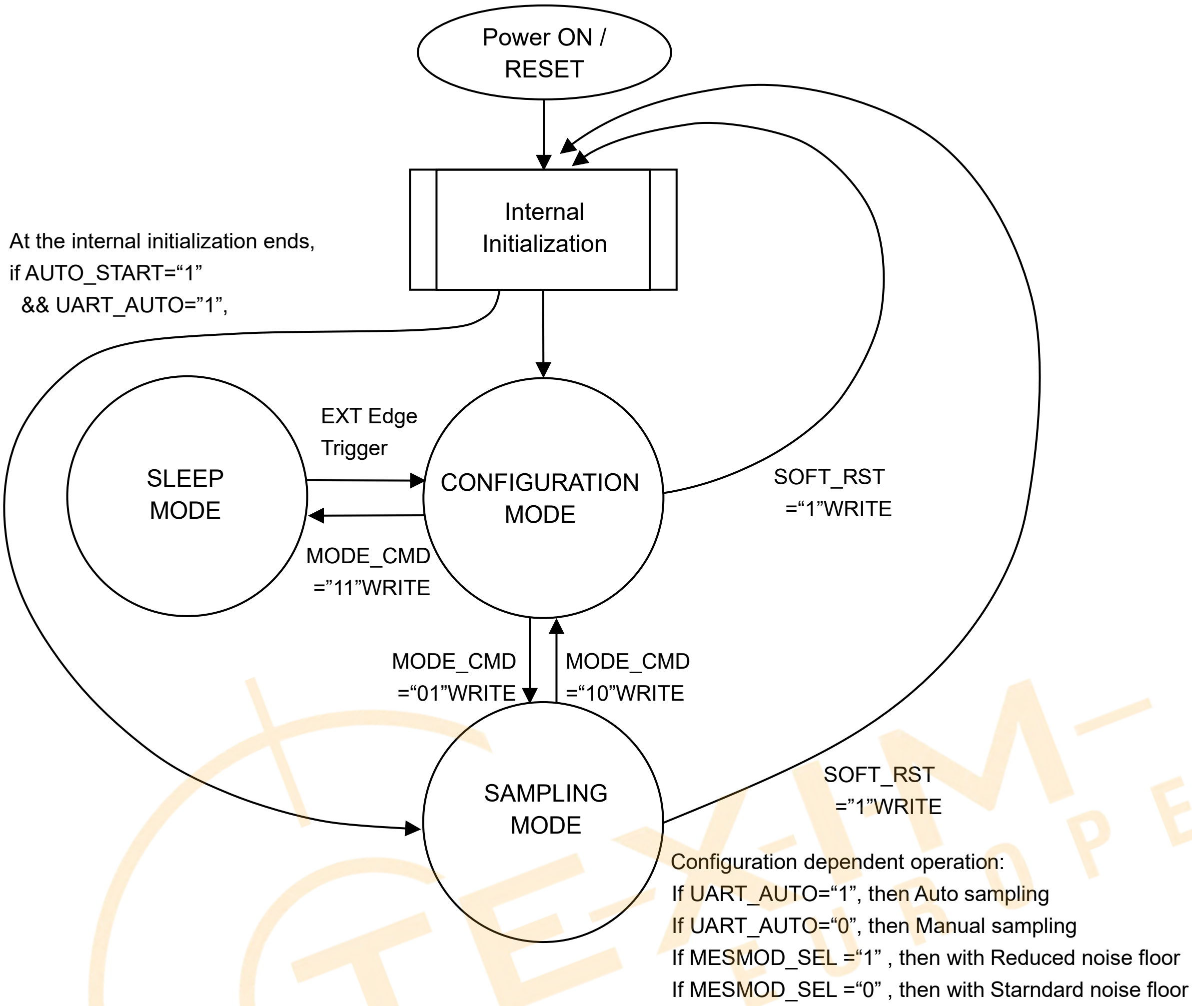


Figure 5.3 Operational State Diagram

5.3 Functional Block

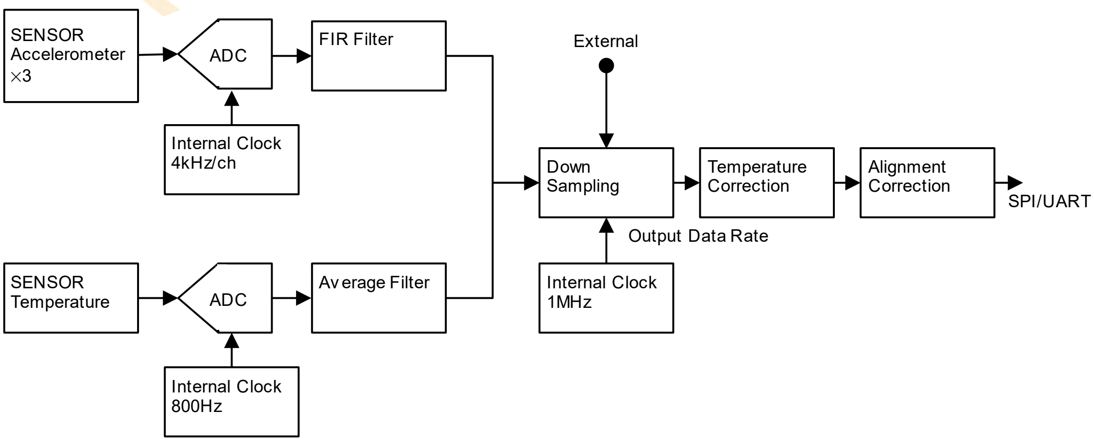


Figure 5.4 Functional Block Diagram

5.4 Data Output Timing

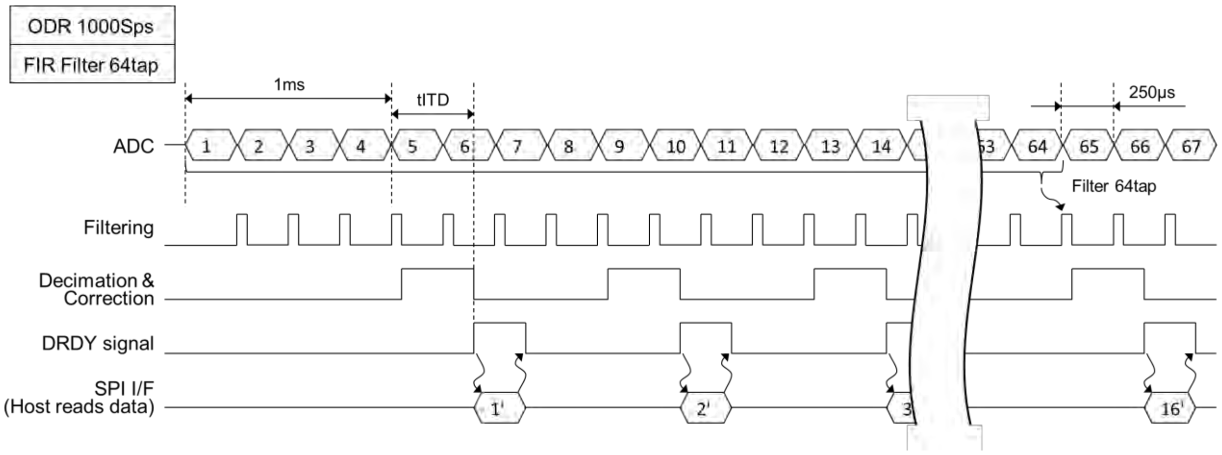


Figure 5.5 Data Output Timing – ODR 1,000 Sps

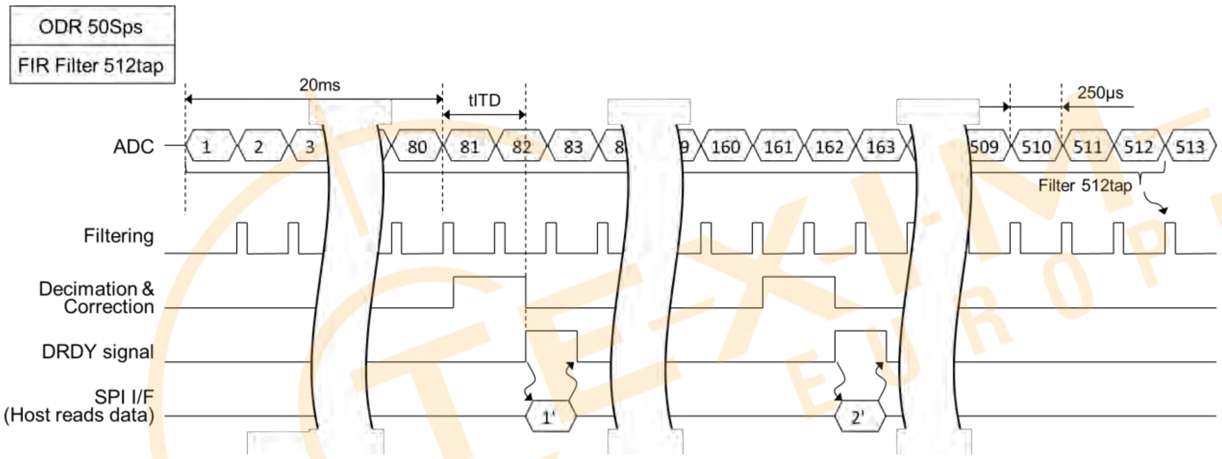


Figure 5.6 Data Output Timing – ODR 50 Sps

5.5 Data Ready Signal

The Data Ready Signal is asserted when one sampling cycle completes and registers are updated with new sensor values. When the sensor values are read out, the Data Ready signal becomes negated. With UART AUTO sampling enabled, the Data Ready signal becomes negated just before data is output.

The Data Ready Signal is output to the DRDY pin when the **DRDY_ON** (MSC_CTRL[0x02(W1)] bit[2]) is set to “1”. The polarity of the signal can be changed by the **DRDY_POL** of MSC_CTRL[0x02(W1)] bit[1] register.

The Data Ready Signal is the logical sum of all the ND flags corresponding to each sensor value. If all the ND flags are disabled in the **ND_EN** (SIG_CTRL[0x00(W1)] bit[15,11:9]), the Data Ready will not be asserted. On the other hand, if all the sensor values enabled in the **ND_EN** (SIG_CTRL[0x00(W1)] bit[15,11:9]) are not read out, the Data Ready signal is kept asserted and never becomes negated.

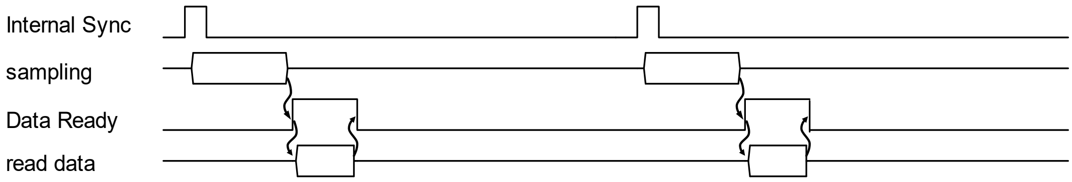


Figure 5.7 Data Ready Signal Timing

5.6 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, can be read. The count interval is 250usec/count and is based on the precision of the internal reference oscillator (crystal).

Additionally, during UART/SPI burst mode 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.

5.7 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 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.

5.8 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.

5.9 External Trigger Input

External Trigger Input function provides control of the sample data output timing by using an externally supplied input pulse signal to EXT pin. By enabling the **EXT_SEL** (MSC_CTRL[0x02(W1)] bit[6]), EXT pin can be used as External Trigger Input pin. The polarity of External Trigger Input (Positive Pulse / Negative Pulse) can be selected by **EXT_POL** (MSC_CTRL[0x02(W1)] bit[5]).

When this function is active, the operation is as follows:

- For UART Auto Sampling:

  When External Trigger Input pin is asserted, the latest sampling data is set to each register and sent to Host automatically.

- For all other modes:

  When External Trigger Input pin is asserted, the latest sampling data is set to each register and Data Ready signal is asserted. The Host should then read the sampling data synchronized with Data Ready signal.

Note) In case of External Trigger function usage please apply appropriate filter setting (**FILTER_SEL**) depending on the External Trigger period. Inappropriate filter setting may affect sensor noise performance.

The External Trigger Input Timing requirements and timing diagrams are shown in Figure 5.8, and Figure 5.9.

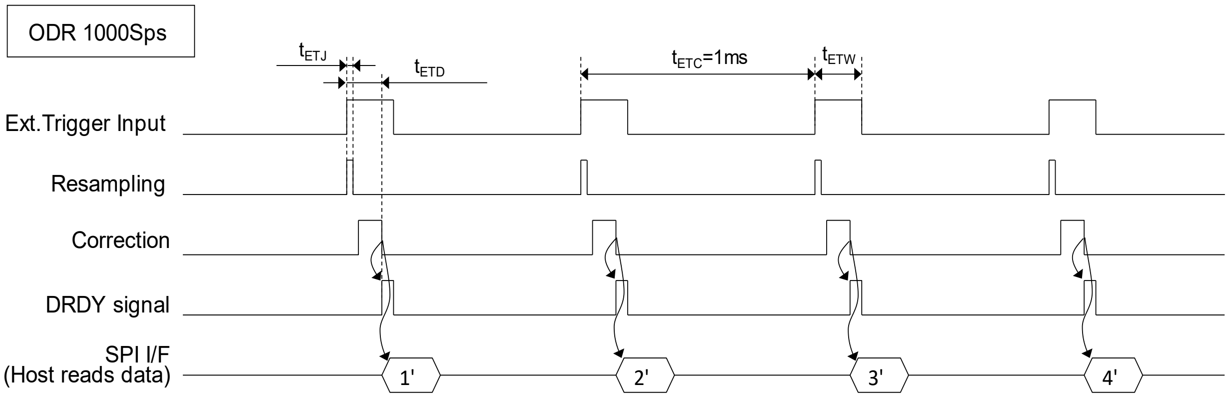


Figure 5.8 External Trigger Input (Auto Sampling)

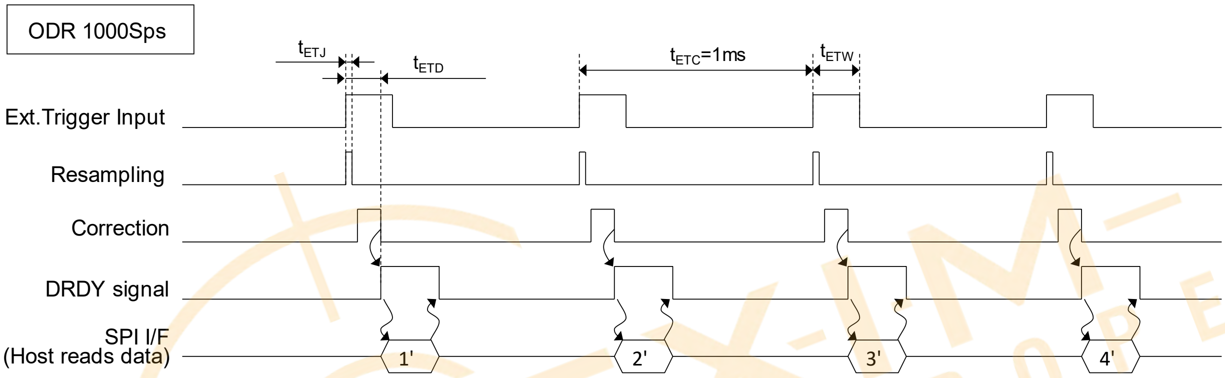


Figure 5.9 External Trigger Input (UART/SPI Manual Sampling)

5.10 Checksum

A checksum can be appended to the response data during UART/SPI Burst mode 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":

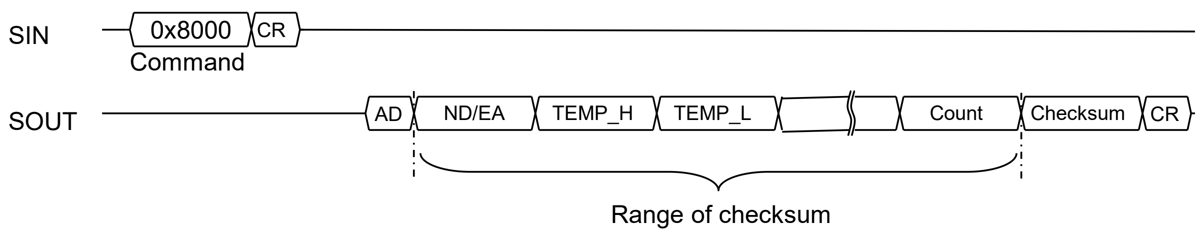


Figure 5.10 Checksum

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.

$$\theta = \mathrm{asinG}[rad]$$

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 Intermittent Measurement for Total Current Reduction

This explains how to realize intermittent measurement for reducing the device current consumption. The user can realize the intermittent measuremet by one of the methods below.

（1）Method of using a sleep mode

(2) Method of switching the device power directly on and off

Table 5.1 shows a summary of some essential items and characteristics.

Table 5.1 Summary of Intermittent Measurement Characteristics and Parameters

	(1) Using sleep mode	(2) Device power on and off
Switching method	Controlling a register and an EXT pin.	Switching the M-A352 power directly on and off
Current consumption at standby	1.3 mA (typ.)	0 mA
Wakeup time	16 msec (Max.)	900 msec (Max.)
Advantage	Short wakeup time from sleep mode to sampling mode	Minimum current consumption at standby (power off)
Disadvantage	-	Necessity for design considerations to correctly handle floating device interface pins, or unpowered pins during standby mode (power off) and transition current at wakeup (power on)
Example of intended use	Event-driven measurement	Occasional measurement and long standby (power off) time

Note) When returning to sampling mode, current consumption increases from a low level to the typical current at sampling mode. This causes an increase in internal heating of the device resulting in a transitional increase in temperature compensation errors.
Note) The extent of the errors depends on many variables such as standby time, environment conditions, etc, therefore, the user should evaluate carefully these effects when used for strict and high precision measurement scenarios.

Method of using sleep mode

The sleep mode function can be enabled by register setting. When shifting to sleep mode, internal circuit operation stops and current consumption during standby mode can be reduced to 1.3 mA (typ.). Wakeup time from sleep mode to sampling mode can be shorter than that from power on to start time (reduced from 900 msec to 16 msec).

Put the operation mode from configuration mode into sleep mode by writing "11" to **MODE_CMD** (MODE_CTRL[0x02(W0)], bit[9:8]). The device can wake up from sleep mode to configuration mode in Sleep Wake-up Time by detecting an edge trigger on the EXT pin. Timing sequence from configuration mode to sleep mode and vice versa are shown in Figure 5.11.

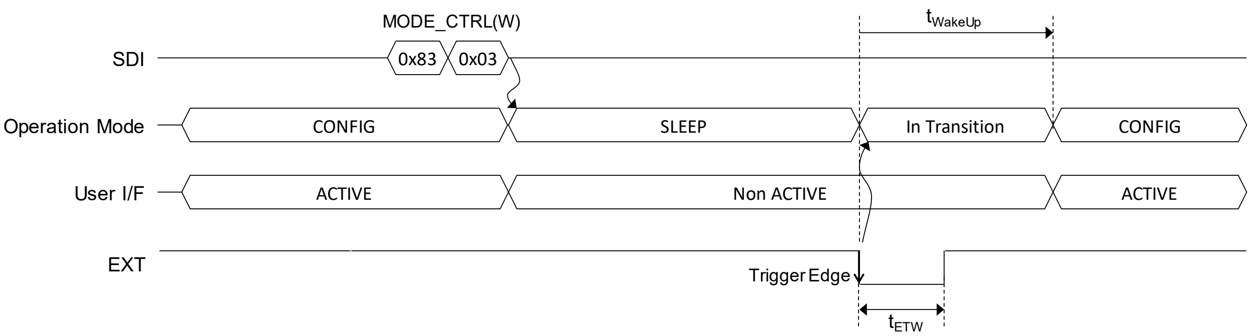


Figure 5.11 Timing Sequence from Configuration mode to Sleep mode and vice versa

Method of switching the device power directly on and off

When the device power is directly switched on and off, the current consumption during standby mode can be 0 mA. Wakeup time from standby to sampling mode is exactly the same as that from power on to start time (900 msec). Please refer to “5.2 Operation Mode” for timing sequence from the power on to sampling mode.

Note) The communication interface pins shown in Table 2.6 will be un-powered during standby (the device power being off), and transition current will appear at wakeup time (device power on). The system interface to the device may need additional design considerations to mitigate the effects, i.e. an unpowered device input pin appears as a short circuit to GND to the host system or when the device is powered on before the system interface to these pins are driven (floating input).

5.15 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 from 13.2 mA typ. to 16.2 mA typ.

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.

5.16 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”.

5.17 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.

5.17.1 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. Figures 5.12 to 5.15 show the typical characteristic of the filters.

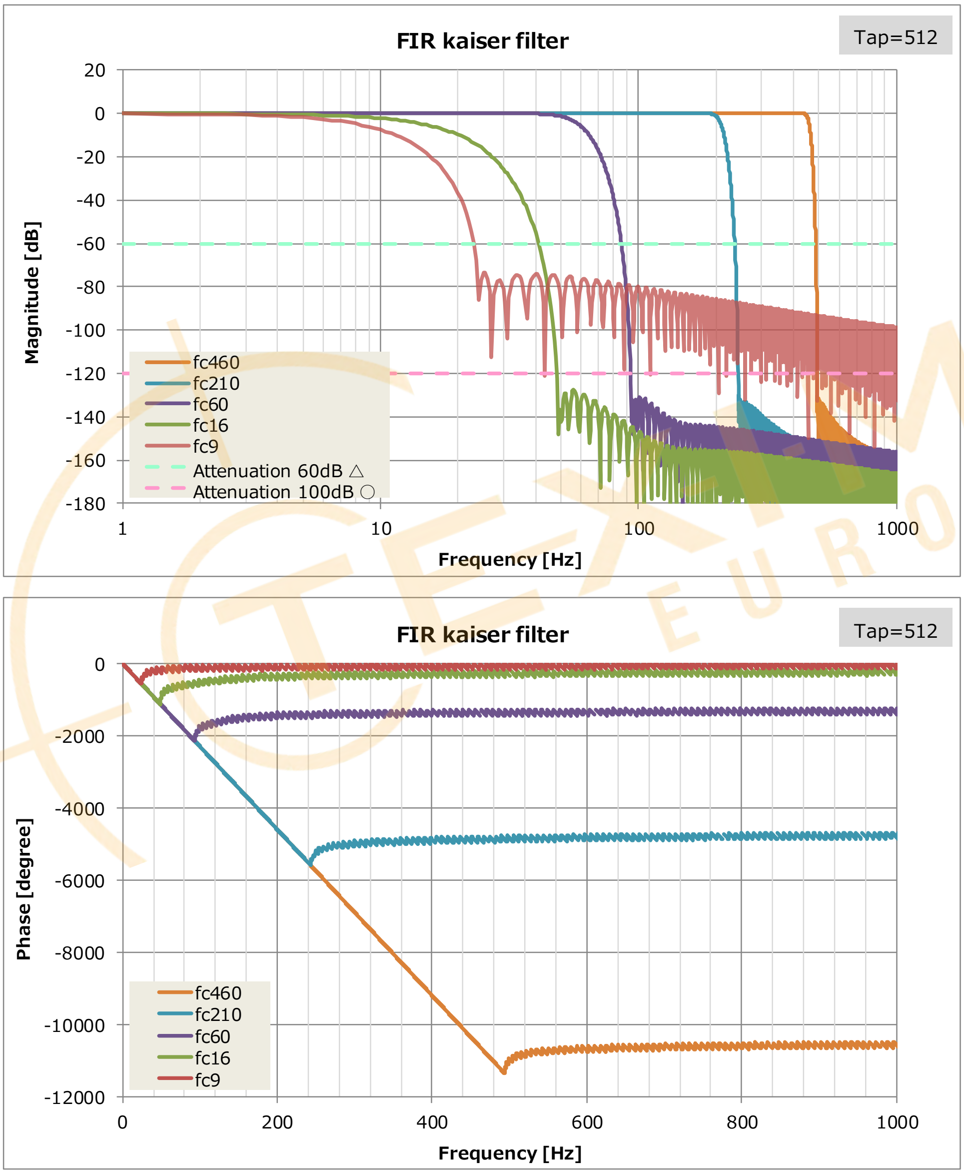
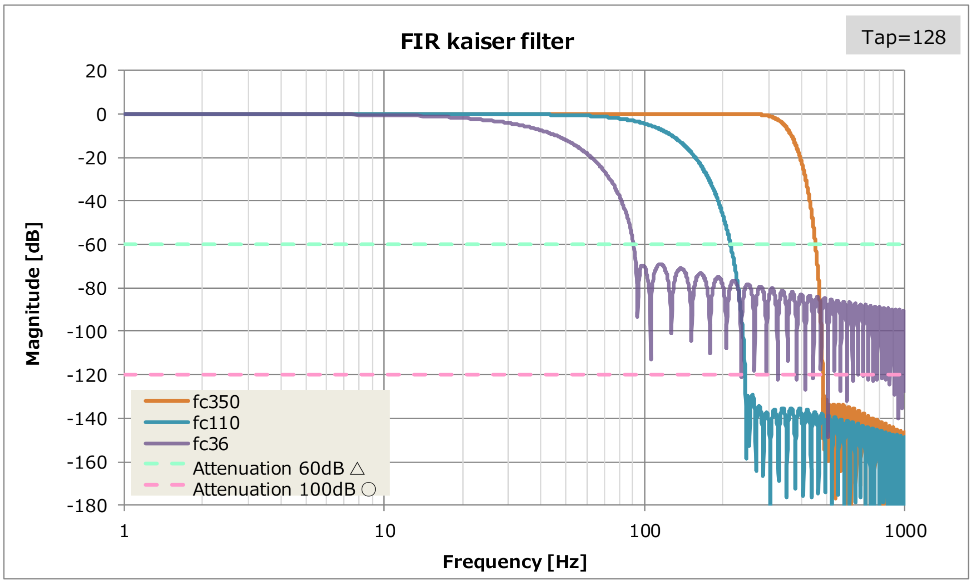


Figure 5.12 FIR Kaiser Filter Characteristic (512 taps)



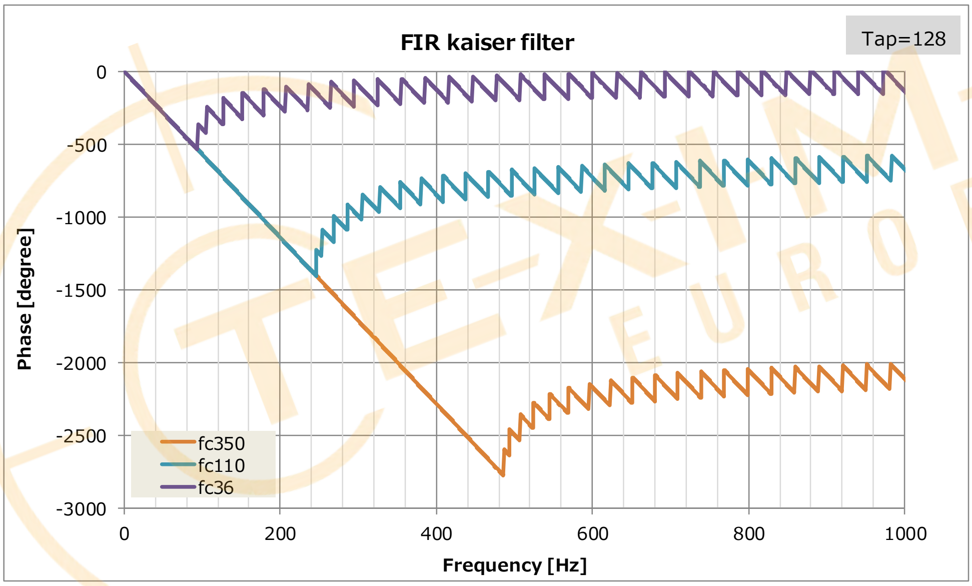


Figure 5.13 FIR Kaiser Filter Characteristic (128 taps)

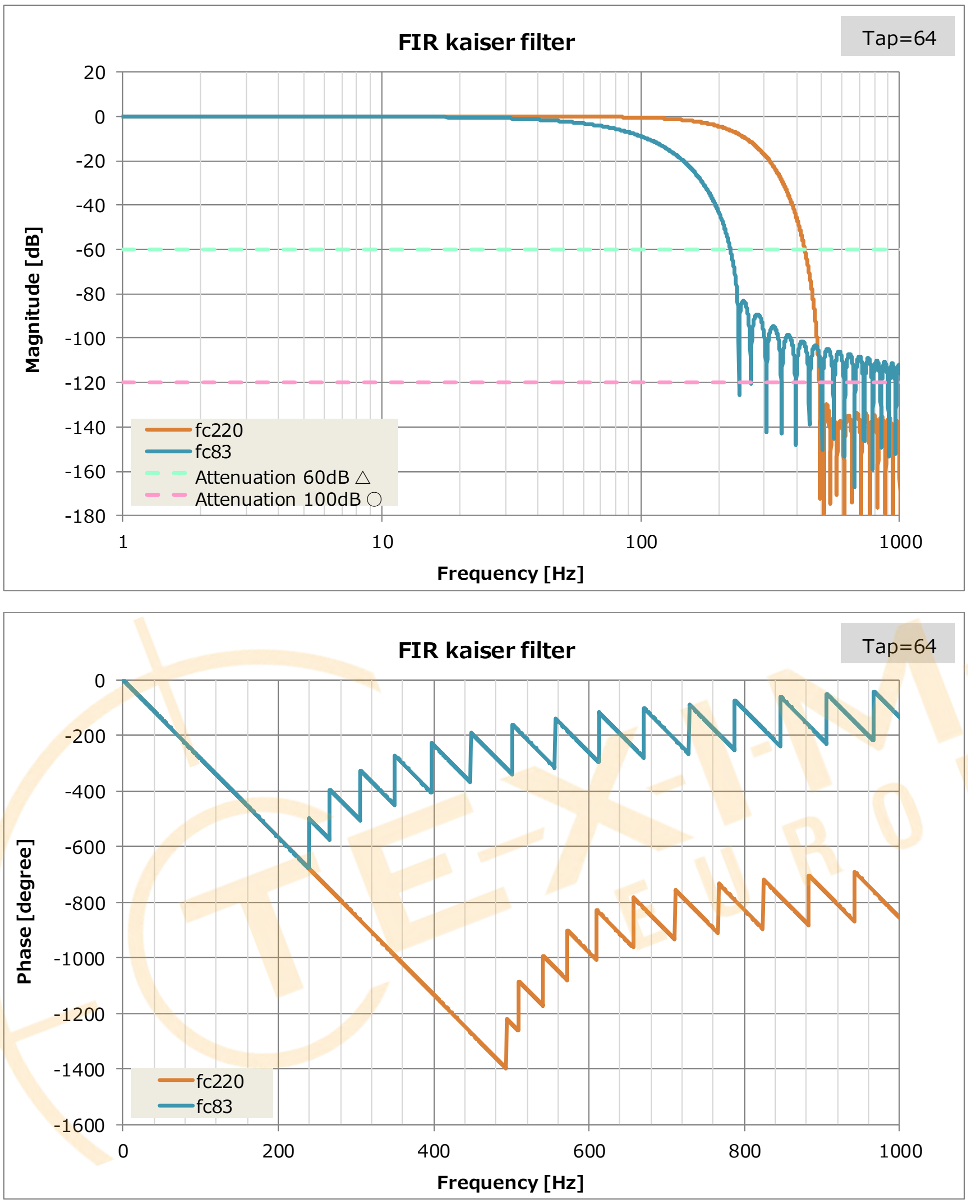


Figure 5.14 FIR Kaiser Filter Characteristic (64 taps)

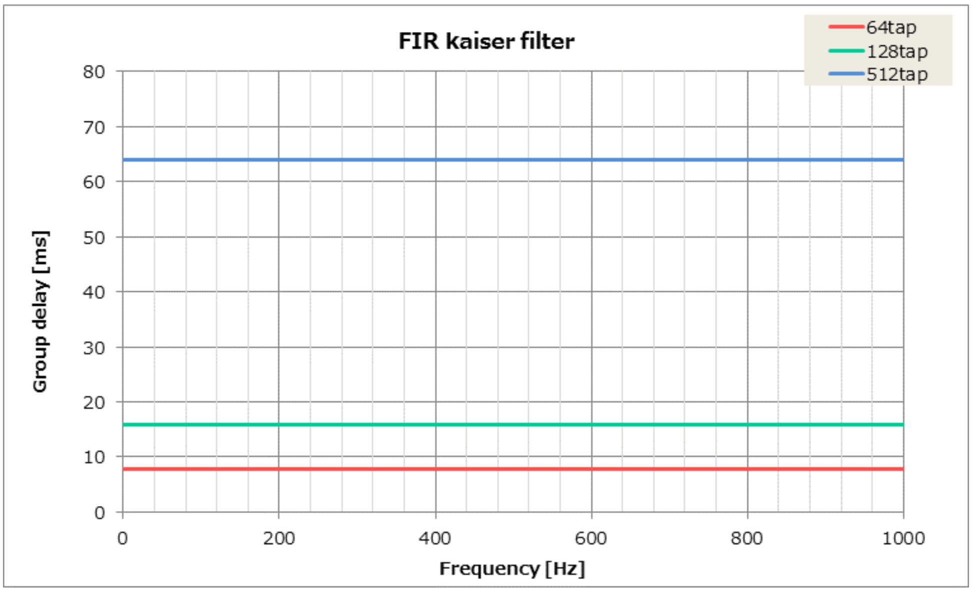


Figure 5.14 FIR Kaiser Filter Characteristic – Group Delay

5.17.2 User Defined 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 coefficent 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 5.2 shows the address ranges for the filter coefficients, and Figure 5.16 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 factory shipment.

Table 5.2 User Defined FIR Filter Coefficient Address Ranges

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

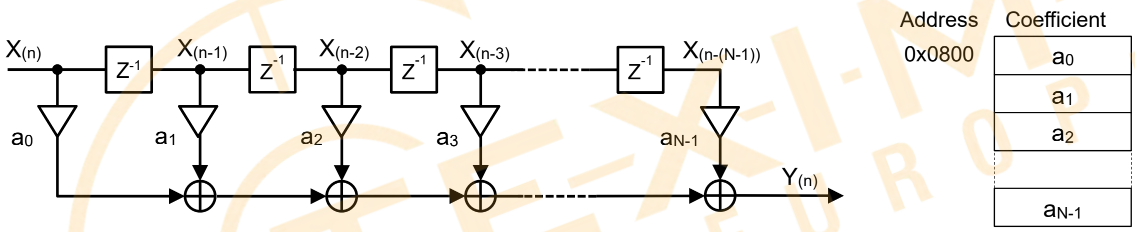


Figure 5.16 N-tap FIR Filter Architecture and Coefficient Memory Map

- 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 registeris to be “00: execution complete”

After the byte has completed writing, the address is automatically incremented by 1, so continuous programming of coefficients are 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 5.17 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.

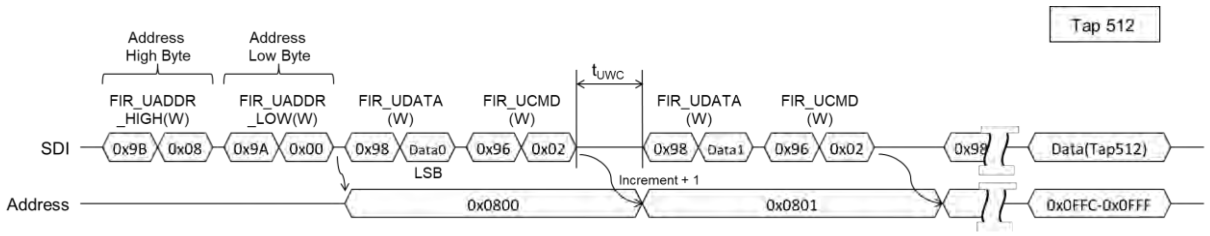


Figure 5.17 User Defined FIR Filter Coefficient Example Write Sequence (512 taps)

- 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 registeris 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 5.18 shows the read sequence.

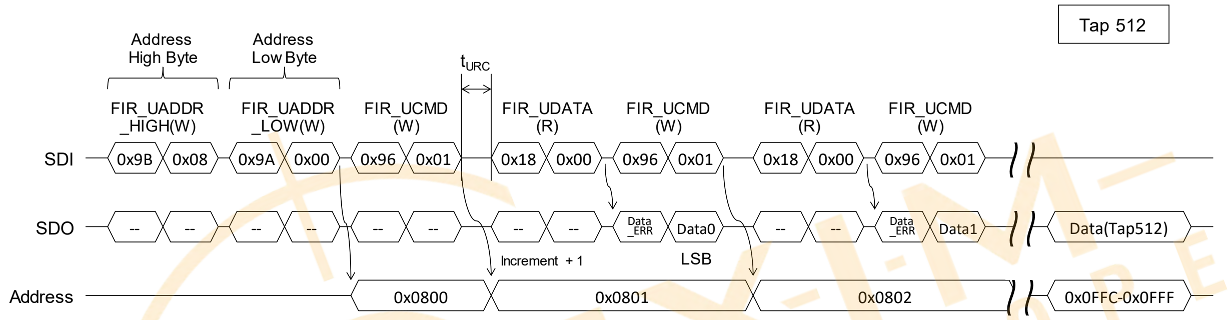


Figure 5.18 User Defined Filter Coefficient Read Sequence (512 taps)

5.17.3 Notes For FIR Filter Usage

- Transient response

As shown in Table 5.3, 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 5.3 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.

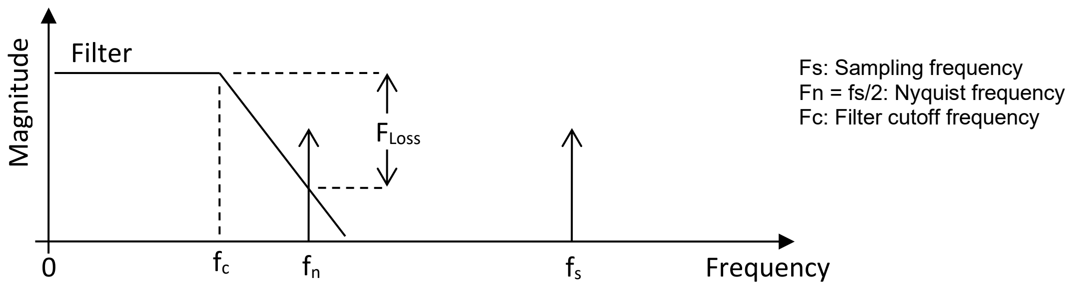


Figure 5.19 Anti-Aliasing Relationship Diagram

Table 5.4 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
— : 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

5.17.4 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 at an 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 5.5.

Table 5.5 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. Digital Interface

This device has the following two external interfaces.

(1) SPI interface

(2) UART interface

The SPI interface and the UART interface have almost the same functions, except for Auto sampling function for the UART interface. No hardware pin configuration is necessary for SPI/UART selection since both interfaces are always active. Connect desired interface pins to SPI or UART interface.

Note) Connecting both SPI and UART at the same time is not supported and may result in malfunction of the device.

The registers inside the device are accessed via the SPI or UART interfaces.

In this document, data sent to the device is called a “Command” and data sent back in response to the command is called a “Response”. There are two types of commands: write command and read command. The write command has no response. The write command always writes to the internal register in 8-bit words. The response to the read command, i.e. the data from the internal register, is always read in 16-bit words.

When reading from the registers, there is a burst mode in addition to the normal mode.

When the IMU output data rate is high (i.e. 1000sps), it may exceed the bandwidth of the host interface and cause the data transmission to be incorrect. In this case, the user must balance the transmission data rate and the bandwidth capability of the host interface.

Adjust the following settings accordingly to optimize the host interface bandwidth:

- For the UART, adjust the baud rate in **BAUD_RATE** (UART_CTRL [0x08(W1)] bit [9:8]).
- For the SPI, adjust the host side SPI clock frequency and SPI wait time.

Adjust the following settings accordingly to optimize the transmission data rate:

- The transmission data rate is affected by the data output rate setting in **DOUT_RATE** (SMPL_CTRL [0x04(W1)] bits [11:8]).
- The transmission data rate is also affected by the number of output bytes included in burst mode read transfer. The adjustment to the number of output bytes is in registers BURST_CTRL [0x0C(W1)].

Several concrete examples for setting the transmission data rate and host interface bandwidth are shown below:

(1) For UART Output:

- **BAUD_RATE** =“01" of UART_CTRL [0x08(W1)] bit [9,8]: 460800 baud
- **UART_AUTO** =“1" of UART_CTRL [0x08(W1)] bit [0]: UART Auto sampling
- **DOUT_RATE** = “0100” of SMPL_CTRL [0x04(W1)] bit [11:8]: 200Sps
- **BURST_CTRL** [0x0C(W1)] = “0x4702”: TEMP, Acceleration, and COUNT output

(2) For SPI Output:

- SPI Interface Transmission Setting: f_{SCLK}=1MHz and t_{STALL}=24us for normal mode
- **DOUT_RATE** = “0100” of SMPL_CTRL [0x04(W1)] bit [11:8]: 200Sps
- **BURST_CTRL** [0x0C(W1)]= “0x4702” : TEMP, Acceleration, and COUNT output

6.1 SPI Interface

Table 6.1 shows the communication settings of SPI interface and Table 6.2 shows the SPI timing for normal mode.

Table 6.1 SPI Communication Settings

Parameter	Setting
Mode	Slave
Word length	16 bits
Phase	Rising edge
Polarity	Negative logic

Table 6.2 SPI Timing (Normal Mode)

Parameter	Minimum	Maximum	Unit
t_{SCLK}	0.01	2.0	MHz
t_{STALL}	20	-	μs
$t_{WRITERATE}$	40	-	μs
$t_{READRATE}$	40	-	μs

6.1.1 SPI Read Timing (Normal Mode)

The response data to a read command, i.e. the data from the internal register, is always returned in 16-bit words. The SPI interface supports sending the next command during the same bus cycle as receiving a response to the read command (full-duplex).

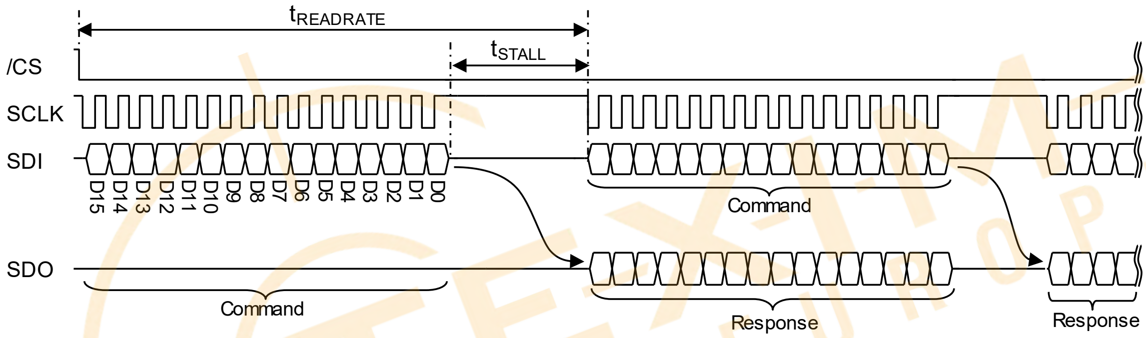


Figure 6.1 SPI Read Timing (Normal Mode)

Table 6.3 Command Format (Read)

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="7">A[6:0]</td><td colspan="8">XX</td></tr>
</table>

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

Table 6.4 Response Format (Read)

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">D[15:8]</td><td colspan="8">D[7:0]</td></tr>
</table>

D[15:8] ··· Register read data (upper byte)
D[7:0] ··· Register read data (lower byte)

6.1.2 SPI Write Timing (Normal Mode)

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

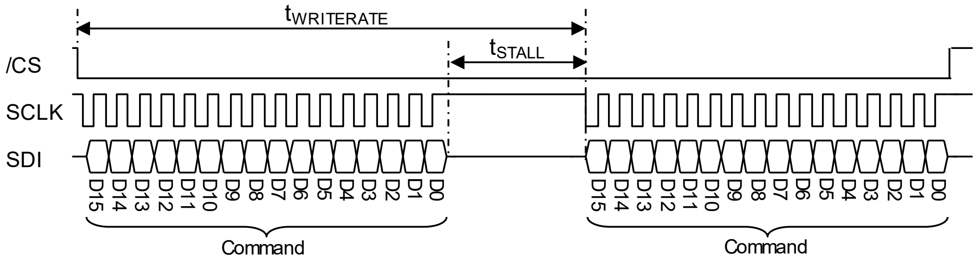


Figure 6.2 SPI Write Timing (Normal Mode)

Table 6.5 Command Format (Write)

15	14		13	12	11	10	9	8	7	6	5	4	3	2	1	0
1		A[6:0]							D[7:0]							

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

6.1.3 SPI 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 flag/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 6.3 Data Packet Format for the response format.

Table 6.6 SPI Timing (Burst Mode)

Parameter	Minimum	Maximum	Unit
f_{SCLK}	0.01	2.0	MHz
t_{STALL1}	45	-	μs
t_{STALL2}	0	-	μs
$t_{READRATE2}$	8	-	μs

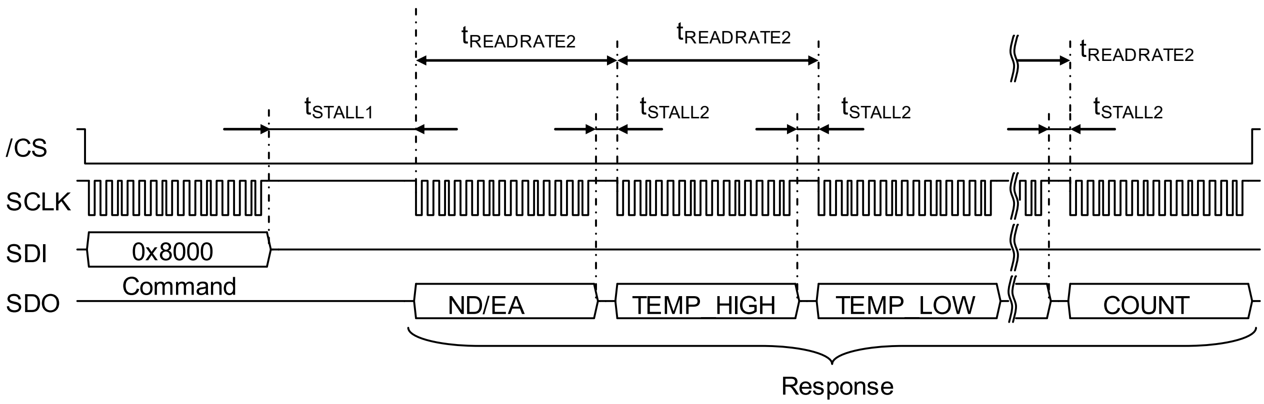


Figure 6.3 SPI Read Timing (Burst Mode)

6.2 UART Interface

Table 6.7 shows the supported UART communication settings and Figure 6.4 shows the UART bit format. Please refer to **BAUD_RATE** (UART_CTRL [0x08(W1)] bit[9:8]) for changing the baud rate setting.

Table 6.7 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)

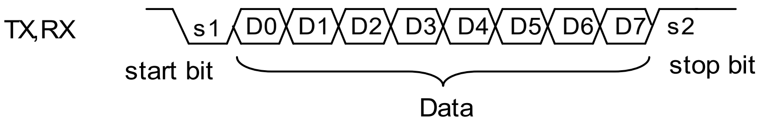


Figure 6.4 UART Bit Format

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 6.8 and Table 6.9 shows the timing of UART.

Table 6.8 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 6.9 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)

6.2.1 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.

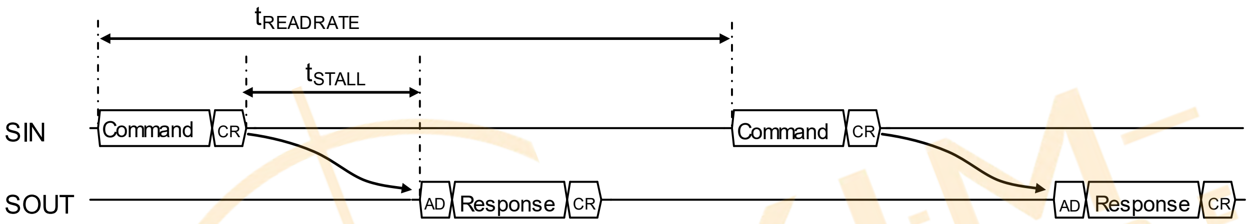


Figure 6.5 UART Read Timing (Normal Mode)

Table 6.10 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 6.11 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

6.2.2 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 6.3 Data Packet Format for the response format.

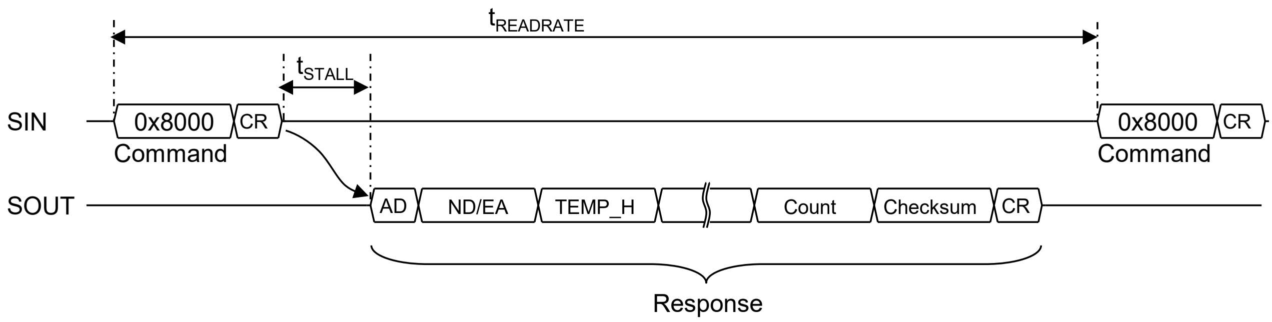


Figure 6.6 UART Read Timing (Burst Mode)

Table 6.12 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

6.2.3 UART Write Timing

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

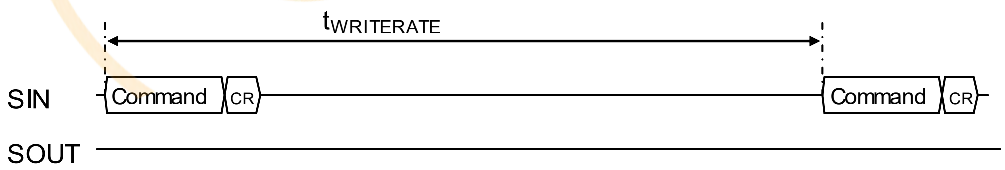


Figure 6.7 UART Write Timing

Table 6.13 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

6.2.4 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 6.3 UART Data Packet Format. The response format for the burst read output data is configured by register setting in BURST_CTRL [0x0C(W1)].

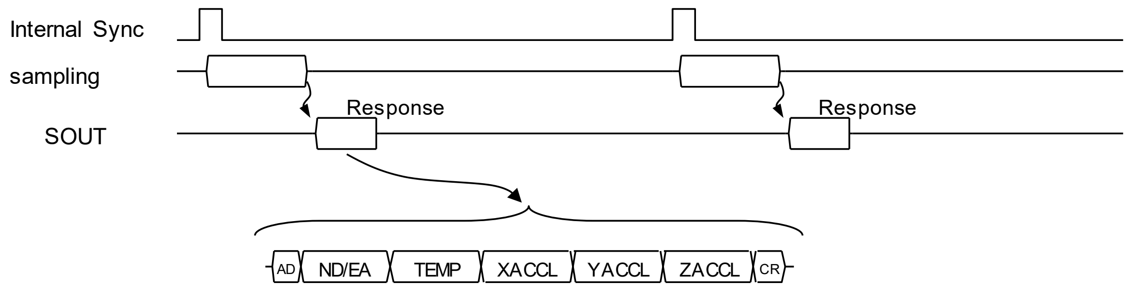


Figure 6.8 UART Auto Sampling Operation

6.3 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 6.14 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							

Table 6.15 Data Packet Format (SPI BURST MODE) Example

BURST_CTRL[0x0C(W1)]=0xC703 (Burst Output, Temp, Acceleration, Counter, Checksum)
SIG_CTRL[0x00(W1)]=0x8E04 (Output Mode : Acceleration, Bias stabilization: Enable)

Word No.	Bit15	. . .	Bit0
1	FLAG(ND/EA)		
2	TEMP_HIGH		
3	TEMP_LOW		
4	XACCL_HIGH		
5	XACCL_LOW		
6	YACCL _HIGH		
7	YACCL _LOW		
8	ZACCL _HIGH		
9	ZACCL _LOW		
10	COUNT		
11	CHECKSUM		

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		0x01	
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			SPI _OVF	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 00: Error occurred
00 : No error
When this error occurs, it indicates the device is faulty.
bit[4] SPI_OVF (SPI OVer Flow)
Shows an error occurred if the device received too many commands from the SPI interface in short period of time.
1: Error occurred
0: No error
When this error occurs, review the SPI command transmission interval and the SPI clock setting.
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 [°C]= 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	-	EXT _SEL	EXT _POL	-	-	DRDY _ON	DRDY _POL	-	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] EXT_SEL

Select the function of the EXT terminal. To use the internal timer trigger, select "0”.
1: External trigger input is enabled
0: External trigger input is disabled（internal timer trigger is enabled）

bit[5] EXT_POL

Selects the polarity of the External Counter Reset Input or External Trigger Input function.
1: Negative logic (falling edge)
0: Positive logic (rising edge)

bit[2] DRDY_ON

Selects the function of the DRDY terminal, when set to “1”, Data Ready signal is output.
1: Data Ready Signal is enabled
0: Data Ready Signal is disabled

bit[1] DRDY_POL

Selects the polarity of the Data Ready signal when selected in **DRDY_ON** above.
1: Active High
0: Active Low

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

Note) This register bit must be set to 0 when using the SPI interface.

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 range : -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 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 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 Sequence

The following describes the recommended procedures for operating this device.

8.1 SPI Sequence

8.1.1 Power-on Sequence (SPI)

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={0xFE01}/ RXdata={0x----}.        /* WINDOW_ID(L) write command.(WINDOW=1) */
 TXdata={0x0A00}/ RXdata={0x----}.        /* GLOB_CMD read command */
 TXdata={0x----}/ RXdata={GLOB_CMD}.      /* 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={0xFE00}/ RXdata={0x----}.        /* WINDOW_ID(L) write command.(WINDOW=0) */
 TXdata={0x0400}/ RXdata={0x----}.        /* DIAG_STAT read command */
 TXdata={0x----}/ RXdata={DIAG_STAT}.     /* get response */
```

Confirm HARD_ERR is 000.
If HARD_ERR is 000, the Accelerometer is OK. Otherwise, the Accelerometer is faulty.
-:don't care

8.1.2 Register Read and Write (SPI)

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

```
 TXdata={0xFE00}/ RXdata={0x----}.        /* WINDOW_ID write command.(WINDOW=0) */
 TXdata={0x0200}/ RXdata={0x----}.        /* MODE_CTRL read command */
 TXdata={0x----}/ RXdata={0x0400}.        /* get response*/
```

-:don't care

0x04 in high byte of RXdata is Configuration mode.
0x00 in low byte of RXdata is Reserved.
Please note that read data unit is 16bit, and Most Significant Bit first in 16bit SPI.

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

```
 TXdata={0xFE00}/ RXdata={0x----}.     /* WINDOW_ID(L) write command.(WINDOW=0) */
 TXdata={0x8301}/ RXdata={0x----}.    /* MODE_CTRL(H) write command.(move to Sampling mode) */
```

There is no response at Write.
-:don't care

By sending this command, the Accelerometer moves to Sampling mode.
Please note that write data unit is 8bit.

8.1.3 Sampling Data (SPI)

[Sample Flow 1 (SPI normal mode)]
Power-on sequence. Please refer to Chapter 8.1.1.
Filter setting sequence. Please refer to Chapter 8.1.9.

```
  TXdata={0xFE01}/ RXdata={0x----}.     /* WINDOW_ID(L) write command.(WINDOW=1) */
  TXdata={0x8504}/ RXdata={0x----}.     /* SMPL_CTRL(H) write command.(200Sps) */
  TXdata={0x8800}/ RXdata={0x----}.     /* UART_CTRL(L) write command.(disable UART Auto sampling) */
```

```
 TXdata={0xFE00}/ RXdata={0x----}.    /* WINDOW_ID(L) write command.(WINDOW=0) */
 TXdata={0x8301}/ RXdata={0x----}.    /* MODE_CTRL(H) write command.(move to Sampling mode) */
receive sampling data.
 (a)Wait until Data Ready signal is asserted.
 (b)
 TXdata={0x0E00}/ RXdata={----}.              /* TEMP_HIGH read command */
 TXdata={0x1000}/ RXdata={TEMP_HIGH}.         /* TEMP_LOW read command */
 TXdata={0x3000}/ RXdata={ TEMP_LOW }.        /* XACCL_HIGH read command */
 TXdata={0x3200}/ RXdata={XACCL_HIGH}.        /* XACCL_LOW read command */
 TXdata={0x3400}/ RXdata={XACCL_LOW}.         /* YACCL_HIGH read command */
 TXdata={0x3600}/ RXdata={YACCL_HIGH}.        /* YACCL_LOW read command */
 TXdata={0x3800}/ RXdata={YACCL_LOW}.         /* ZACCL_HIGH read command */
 TXdata={0x3A00}/ RXdata={ZACCL_HIGH}.        /* ZACCL_LOW read command */
 TXdata={0x0A00}/ RXdata={ZACCL_LOW}.         /* COUNT read command */
 TXdata={0x----}/ RXdata={COUNT}.
 repeat from (a) to (b).
 TXdata={0x8302}/ RXdata={0x----}.  /* MODE_CTRL(H) write command.(return to Configulation mode) */
 -:don't care
```

Note) Please remember to wait until Data Ready signal is asserted.

```
---------------------------------------------------------------
[Sample Flow 2 (SPI burst mode)]
 Power-on sequence. Please refer to Chapter 8.1.1.
 Filter setting sequence. Please refer to Chapter 8.1.9.
 TXdata={0xFE01}/ RXdata={0x----}.    /* WINDOW_ID(L) write command.(WINDOW=1) */
 TXdata={0x8504}/ RXdata={0x----}.    /* SMPL_CTRL(H) write command.(200Sps) */
 TXdata={0x8800}/ RXdata={0x----}.    /* UART_CTRL(L) write command.(disable UART Auto sampling) */
 TXdata={0x8C02}/ RXdata={0x----}.    /* BURST_CTRL(L) write command.(COUNT=on) */
 TXdata={0x8D47}/ RXdata={0x----}.    /* BURST_CTRL(H) write command.(TEMP=on, ACC_XYZ=on) */
 TXdata={0xFE00}/ RXdata={0x----}.    /* WINDOW_ID(L) write command.(WINDOW=0) */
 TXdata={0x8301}/ RXdata={0x----}.    /* MODE_CTRL(H) write command.(move to Sampling mode) */
receive sampling data.
 (a)Wait until Data Ready signal is asserted.
 (b)
 TXdata={0x8000}/ RXdata={0x----}.    /* BURST(L) write command */
 TXdata={0x----}/ RXdata={TEMP_HIGH}.
 TXdata={0x----}/ RXdata={TEMP_LOW}.
 TXdata={0x----}/ RXdata={XACCL_HIGH}.
 TXdata={0x----}/ RXdata={XACCL_LOW}.
 TXdata={0x----}/ RXdata={YACCL_HIGH}.
 TXdata={0x----}/ RXdata={YACCL_LOW}.
 TXdata={0x----}/ RXdata={ZACCL_HIGH}.
 TXdata={0x----}/ RXdata={ZACCL_LOW}.
 TXdata={0x----}/ RXdata={COUNT}.
 repeat from (a) to (b).
TXdata={0x8302}/ RXdata={0x----}. /* MODE_CTRL(H) write command.(return to Configulation mode) */
-:don't care
```

Note) Please remember to wait until Data Ready signal is asserted.

8.1.4 Selftest (SPI)

Selftest is as follows.

```
Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send self test command.
 TXdata={0xFE01}/ RXdata={0x----}.          /* WINDOW_ID(L) write command.(WINDOW=1) */
```

TXdata={0x8304}/ RXdata={0x----}. /* 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={0x0200}/ RXdata={0x----}. /* MSC_CTRL read command */
TXdata={0x----}/ RXdata={MSC_CTRL}. /* 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={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x0400}/ RXdata={0x----}. /* DIAG_STAT read command */
TXdata={0x----}/ RXdata={DIAG_STAT}. /* get response */
Confirm each ACC_ERR is 0.
If each ACC_ERR is 0, the result is OK. Otherwise, the result is NG.
-:don't care

8.1.5 Flash Test (SPI)

Flash test is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send flash test command.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8308}/ RXdata={0x----}. /* 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={0x0200}/ RXdata={0x----}. /* MSC_CTRL read command */
TXdata={0x----}/ RXdata={MSC_CTRL}. /* 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={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x0400}/ RXdata={0x----}. /* DIAG_STAT read command */
TXdata={0x----}/ RXdata={DIAG_STAT}. /* get response */
Confirm FLASH_ERR is 0.
If FLASH_ERR is 0, the result is OK. Otherwise, the result is NG.
-:don't care

8.1.6 Software Reset (SPI)

Software reset is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send software reset command.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8A80}/ RXdata={0x----}. /* GLOB_CMD(L) write command.(Software reset) */
(b) Wait Reset Recovery Time.
-:don't care

8.1.7 Flash Backup (SPI)

Flash backup is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send flash backup command.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8A08}/ RXdata={0x----}. /* 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={0x0A00}/ RXdata={0x----}. /* GLOB_CMD read command */
TXdata={0x----}/ RXdata={GLOB_CMD}. /* get response */
Confirm FLASH_BACKUP bit.
When FLASH_BACKUP becomes 0, it ends. Otherwise , please repeat (b).
(c) Confirm the result.
TXdata={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
Confirm FLASH_BU_ERR bits. FLASH_BU_ERR is DIAG_STAT[0x04(W0)]'s bit[0].
TXdata={0x0400}/ RXdata={0x----}. /* DIAG_STAT read command */
TXdata={0x----}/ RXdata={DIAG_STAT}. /* get response */
Confirm FLASH_BU_ERR is 0.
If FLASH_BU_ERR is 0, the result is OK. Otherwise, the result is NG.
-:don't care

8.1.8 Flash Reset (SPI)

Flash Reset is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send flash reset command.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8A04}/ RXdata={0x----}. /* 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={0x0A00}/ RXdata={0x----}. /* GLOB_CMD read command */
TXdata={0x----}/ RXdata={GLOB_CMD}. /* get response */
Confirm FLASH_RST bit.
When FLASH_RST becomes 0, it ends. Otherwise , please repeat (b).

(c) Confirm the result.
TXdata={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
Confirm FLASH_BU_ERR bits. FLASH_BU_ERR is DIAG_STAT[0x04(W0)]'s bit[0].
TXdata={0x0400}/ RXdata={0x----}. /* DIAG_STAT read command */
TXdata={0x----}/ RXdata={DIAG_STAT}. /* get response */
Confirm FLASH_BU_ERR is 0.
If FLASH_BU_ERR is 0, the result is OK. Otherwise, the result is NG.
-:don't care

(d) Power off and on , or reset.

8.1.9 Filter Setting (SPI)

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={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8608}/ RXdata={0x----}. /* 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={0x0600}/ RXdata={0x----}. /* FILTER_CTRL read command */

TXdata={0x----}/ RXdata={FILTER_CTRL}. /* 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 (SPI)

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={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
First, set the start address (0x0800),
TXdata={0x9B08}/ RXdata={0x----}. /* FIR_UADDR(H) write command.(Address High Byte) */
TXdata={0x9A00}/ RXdata={0x----}. /* 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={0x9853}/ RXdata={0x----}. /* FIR_UDATA(L) write command */

(c) Send filter coefficient control command.
TXdata={0x9602}/ RXdata={0x----}. /* 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={0x1600}/ RXdata={0x----}. /* FIR_UCMD read command */
TXdata={0x----}/ RXdata={FIR_UCMD}. /* 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={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
First, set the start address (0x0800),
TXdata={0x9B08}/ RXdata={0x----}. /* FIR_UADDR(H) write command.(Address High Byte) */
TXdata={0x9A00}/ RXdata={0x----}. /* FIR_UADDR(L) write command.(Address Low Byte) */

(b) Send filter coefficient control command.
TXdata={0x9601}/ RXdata={0x----}. /* 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={0x1600}/ RXdata={0x----}. /* FIR_UCMD read command */
TXdata={0x----}/ RXdata={FIR_UCMD}. /* get response */
Confirm FIR_UCMD bit.
When FIR_UCMD becomes 00, it ends. Otherwise , please repeat (c).

(d) Send filter coefficient data command.
TXdata={0x1800}/ RXdata={0x----}. /* FIR_UDATA read command */

TXdata={0x----}/ RXdata={FIR_UDATA}. /* 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 Sleep Sequence (SPI)

Sleep sequence is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Enter Sleep mode
TXdata={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x8303}/ RXdata={0x----}. /* MODE_CTRL(H) write command.(move to sleep mode) */

(b) Wake up from Sleep mode
Wake up from sleep mode and move to config mode by detecting an edge trigger on the EXT pin.
After waiting Sleep Wake-up Time, can access the registers in SPI interface.

notes
SPI communication is not possible during sleep mode.

8.1.12 Reduced Noise Floor Condition Setting (SPI)

Reduced noise floor condition setting is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send a Reduced noise floor condition selection command.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x8014}/ RXdata={0x----}. /* SIG_CTRL(L) write command.(select measurement condition) */

(b) Execute Flash backup. Please refer to Chapter 8.1.7.
(c) Power off and on.
(d) Wait Power-On Start-Up Time.

(e) Confirm measurement condition selection state.
TXdata={0xFE01}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=1) */
Confirm MESMOD_STAT bits. MESMOD_STAT is GLOB_CMD[0x0A(W1)]'s bit[12].
TXdata={0x0A00}/ RXdata={0x----}. /* GLOB_CMD read command */
TXdata={0x----}/ RXdata={GLOB_CMD}. /* get response */
Confirm MESMOD_STAT bit.
When MESMOD_STAT is 1, it is possible to select the Reduced noise floor condition.

(f) Select the Reduced noise floor condition
TXdata={0xFE00}/ RXdata={0x----}. /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x8301}/ RXdata={0x----}. /* MODE_CTRL(H) write command.(move to sampling mode) */

8.1.13 Bias Offset Setting (SPI)

Bias offset setting is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send bias offset setting command.
For example, if X axis bias offset value is +1.23G (0x013AE147),

```
TXdata={0xFE01}/ RXdata={0x----}.     /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0xAD01}/ RXdata={0x----}.     /* XA_OFFSET_HIGH(H) write command. */
TXdata={0xAC3A}/ RXdata={0x----}.     /* XA_OFFSET_HIGH(L) write command. */
TXdata={0xAFE1}/ RXdata={0x----}.     /* XA_OFFSET_LOW(H) write command. */
TXdata={0xAE47}/ RXdata={0x----}.     /* XA_OFFSET_LOW(L) write command. */
```

8.1.14 Alarm Threshold Setting (SPI)

Alarm threshold settig is as follows.

Power-on sequence. Please refer to Chapter 8.1.1.
(a) Send alarm threshold setting command.
For example, if X axis alarm threshold value is +5G/-5G,

```
TXdata={0xFE01}/ RXdata={0x----}.     /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0xC705}/ RXdata={0x----}.     /* XA_ALARM write command. (upper limit value)*/
TXdata={0xC6FB}/ RXdata={0x----}.     /* XA_ALARM write command. (lower limit value)*/
```

8.2 UART Sequence

8.2.1 Power-on Sequence (UART)

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.

8.2.2 Register Read and Write (UART)

[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.

8.2.3 Sampling Data (UART)

[Sample Flow 1 (UART Auto sampling)]
Power-on sequence. Please refer to Chapter 8.2.1.
Filter setting sequence. Please refer to Chapter 8.2.9.

```
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 8.2.1.
 Filter setting sequence. Please refer to Chapter 8.2.9.
 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)
*/

Note)  Please remember to wait until Data Ready signal is asserted.
------------------------------------------------------------
[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
```

8.2.4 Selftest (UART)

Selftest is as follows.

```
Power-on sequence. Please refer to Chapter 8.2.1.
(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.
```

8.2.5 Flash Test (UART)

Flash test is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(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.
```

8.2.6 Software Reset (UART)

Software reset is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(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.

8.2.7 Flash Backup (UART)

Flash backup is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(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.
```

8.2.8 Flash Reset (UART)

Flash Reset is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(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.

8.2.9 Filter Setting (UART)

Filter setting is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(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).

8.2.10 User Defined FIR Filter Coefficients Setting (UART)

User Defined FIR Filter coefficients setting is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.

[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 8.2.9.

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.

8.2.11 Sleep Sequence (UART)

Sleep sequence is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(a) Enter Sleep mode
 TXdata={0xFE,0x00,0x0d}. /* WINDOW_ID(L) write command.(WINDOW=0) */
 TXdata={0x83,0x03,0x0d}. /* MODE_CTRL(H) write command.(move to sleep mode) */

(b) Wake up from Sleep mode
 Wake up from sleep mode and move to config mode by detecting an edge trigger on the EXT pin.
 After waiting Sleep Wake-up Time, can access the registers in UART interface.

notes
UART communication is not possible during sleep mode.

8.2.12 Reduced Noise Floor Condition Setting (UART)

Reduced noise floor condition setting is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(a) Send a Reduced noise floor condition selection command.
```
TXdata={0xFE,0x01,0x0d}.          /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0x80,0x14,0x0d}.          /* SIG_CTRL(L) write command.(select measurement condition) */
```

(b) Execute Flash backup. Please refer to Chapter 8.2.7.
(c) Power off and on.
(d) Wait Power-On Start-Up Time.

(e) Confirm measurement condition selection state.
```
TXdata={0xFE,0x01,0x0d}.          /* WINDOW_ID(L) write command.(WINDOW=1) */
```
Confirm MESMOD_STAT bits. MESMOD_STAT is GLOB_CMD[0x0A(W1)]'s bit[12].
```
TXdata={0x0A,0x00,0x0d}.          /* GLOB_CMD read command */
RXdata={0x0A,MSByte,LSByte,0x0d}. /* get response */
```
Confirm MESMOD_STAT bit.
When MESMOD_STAT is 1, it is possible to select the Reduced noise floor condition.

(f) Select the Reduced noise floor condition
```
TXdata={0xFE,0x00,0x0d}.          /* WINDOW_ID(L) write command.(WINDOW=0) */
TXdata={0x83,0x01,0x0d}.          /* MODE_CTRL(H) write command.(move to Sampling mode) */
```

8.2.13 Bias Offset Setting (UART)

Bias offset setting is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(a) Send bias offset setting command.
For example, if X axis bias offset value is +1.23G (0x013AE147),
```
TXdata={0xFE,0x01,0x0d}.          /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0xAD,0x01,0x0d}.          /* XA_OFFSET_HIGH(H) write command. */
TXdata={0xAC,0x3A,0x0d}.          /* XA_OFFSET_HIGH(L) write command. */
TXdata={0xAF,0xE1,0x0d}.          /* XA_OFFSET_LOW(H) write command. */
TXdata={0xAE,0x47,0x0d}.          /* XA_OFFSET_LOW(L) write command. */
```

8.2.14 Alarm Threshold Setting (UART)

Alarm threshold settig is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.
(a) Send alarm threshold setting command.
For example, if X axis alarm threshold value is +5G/-5G,
```
TXdata={0xFE,0x01,0x0d}.          /* WINDOW_ID(L) write command.(WINDOW=1) */
TXdata={0xC7,0x05,0x0d}.          /* XA_ALARM write command. (upper limit value)*/
TXdata={0xC6,0xFB,0x0d}.          /* XA_ALARM write command. (lower limit value)*/
```

8.2.15 Auto Start (UART only)

Auto Start is as follows.

Power-on sequence. Please refer to Chapter 8.2.1.

```
(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 Use

- When you attach the product to a housing, equipment, jig, or tool, make sure you attach it properly so that no mechanical stress is added to create a distortion such as a warp or twist. In addition, tighten the screws firmly but not too firmly because the mount of the product may break. Use screw locking techniques as necessary.
- When you set up the product, make sure the equipment, jigs, tools, and workers maintain a good ground in order not to generate high voltage leakage. If you add overcurrent or static electricity to the product, the product may be damaged permanently.
- When you install the product, make sure metallic or other conductors do not enter the product. Otherwise, malfunction or damage of the product may result.
- If excessive shock is added to the product when, for example, the product falls, the quality of the product may be degraded. Make sure the product does not fall when you handle it.
- Before you start using the product, test it in the actual equipment under the actual operating environment.
- Since the product has capacitors inside, inrush current will occur during power-on. Evaluate in the actual environment in order to check the effect of the supply voltage drop by inrush current in the system.
- If water enters the product, malfunction or damage of the product may result. If the product can be exposed to water, the system must have a waterproof structure. We do not guarantee the operation of the product when the product is exposed to condensation, dust, oil, corrosive gas (salt, acid, alkaline, or the like) , or direct sunlight.
- This product is not designed to be radiation resistant.
- Never use this product if the operating condition is over the absolute maximum rating. If you do, the characteristics of the product may never recover.
- If the product is exposed to excessive exogenous noise or the like, degradation of the precision, malfunction, or damage of 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.
- Mechanical vibration or shock, continuous mechanical stress, rapid temperature change, or the like may cause cracks or disconnections at the various connecting parts.
- Take sufficient safety measure for the equipment this product is built into.
- This product is not intended for general use by the consumer but instead for engineering design. For the customer, please consider it safely with the proper use.
- This product is not designed to be used in the 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.). Therefore, Seiko Epson Corporation will not be liable for any damages caused by the use of the product for those applications.
- Do not alter or disassemble the product.

9.2 Cautions for Storage

- Do not add shock or vibration to the packing box. Do not spill water over the packing box. Do not store or use the product in the environment where dew condensation occurs due to rapid temperature change.
- To suppress the characteristic change by prolonged storage, it is recommended to maintain the environment at normal temperature and normal humidity. Normal temperature: +5 ~ +35 °C Normal humidity: 45%RH ~ 85%DH (JIS Z 8703).
- Do not store the product in a location subject to High Temperature, high humidity, under direct sunlight, corrosive gas or dust.
- Do not put mechanical stress on the product while it is stored.

9.3 Other Cautions

- When you connect the socket to the header of this product, make sure you do not insert the header in the reverse orientation. If you do, the product may be damaged permanently.
- The gloss marks derived from the adhesive material may have appeared on the casing surface of the product, but it does not affect the function and quality of the product.
- The Parting line as a result of die cast manufacturing process may have appeared on the casing surface of the product, but it is not an abnormality.
- Please take care not to tamper with or accidently disturb the assembly screw on the surface where the serial number is printed when attaching and detaching the product to the system. We do not guarantee the performance and the quality of the product in case the assembly screw is manipulated.
- 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.
- If a radio (transmission antenna) is set up near this product, degradation of the precision may result by radio frequency interference. Place the radio (transmission antenna) as far away as possible or add shielding to mitigate the effects of radio frequency interference.
- Never turn off power while the host communicates the product. Otherwise, malfunction of the product may result.
- Small performance deterioration due to long-term use and aging effects, etc. cannot be detected through the self-diagnosis test in this product. Discontinue use immediately even when the self-diagnosis test results in a “pass” when experiencing abnormality in the sensor performance.
- If noise is induced on the external trigger terminal, there is a possibility an invalid measurement process is unintentionally sent to the host. To prevent this, when using an external trigger, take precaution to minimize noise on the external trigger terminal.
- Exercise care and precaution with the packaging and during transport of the equipment that this product is installed on to avoid excessive vibration and or damage from impact.

9.4 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 product can be ordered with the following numbers. Please inquire separately about details.

Table 10.1 Product Model Number

Product Model Number	Product Name	Comments
X2F000011000100	M-A352AD10	-

11. Evaluation Tools

Evaluation tools can be provided for the M-A352. For details, contact our representatives.

Table 11.1 Evaluation Tool Model Number

Product Model Number	Product Name	Comments
X2H000021000200	M-G32EV041	USB Evaluation Board for M-A352AD10 *Works with Logger Software.
X2H000021000300	M-G32EV051	Relay board for M-A352AD10 *Combination with M-G32EV041 is possible.

Revision History

Rev. No.	Date	Page	Category	Contents
Rev 20191009	2019/10/09	All	New	Newly established
Rev 20191213	2019/12/13	P8	Update	Table 3.2 Recommended socket parts at the host side
Rev 20220401	2022/4/1	P76	Modify	Product Number Change

EPSON

International Sales Operations

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

Revised date APR. 2022 in JAPAN

Disclaimer

ALL PRODUCTS, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Texim Europe B.V. its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, “Texim”), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Texim makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product.

It is the customer’s responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application.

Parameters provided in datasheets and / or specifications may vary in different applications and performance may vary over time.

All operating parameters, including typical parameters, must be validated for each customer application by the customer’s technical experts.

Please contact us if you have any questions about the contents of the datasheet.

This may not be the latest version of the datasheet. Please check with us if a later version is available.

Texim Europe - contact details

Headquarters & Warehouse

Elektrostraat 17
NL-7483 PG Haaksbergen
The Netherlands

T: +31 (0)53 573 33 33
E: info@texim-europe.com
Homepage: www.texim-europe.com

The Netherlands

Elektrostraat 17
NL-7483 PG Haaksbergen

T: +31 (0)53 573 33 33
E: nl@texim-europe.com

Belgium

Zuiderlaan 14, box 10
B-1731 Zellik

T: +32 (0)2 462 01 00
E: belgium@texim-europe.com

UK & Ireland

St Mary's House, Church Lane
Carlton Le Moorland
Lincoln LN5 9HS

T: +44 (0)1522 789 555
E: uk@texim-europe.com

Germany - North

Bahnhofstrasse 92
D-25451 Quickborn

T: +49 (0)4106 627 07-0
E: germany@texim-europe.com

Germany - South

Martin-Kollar-Strasse 9
D-81829 München

T: +49 (0)89 436 086-0
E: muenchen@texim-europe.com

Austria

Warwitzstrasse 9
A-5020 Salzburg

T: +43 (0)662 216 026
E: austria@texim-europe.com

Nordic

Søndre Jagtvej 12
DK-2970 Hørsholm

T: +45 88 20 26 30
E: nordic@texim-europe.com

Italy

Via Matteotti 43
IT-20864 Agrate Brianza (MB)

T: +39 (0)39 9713293
E: italy@texim-europe.com