

IMU (Inertial Measurement Unit): M-G330PDG0

Features

- Small Size, Lightweight : 24 x 24 x 10 mm, 10 grams
- Low-Noise, High-Stability
 - Gyro Bias Instability : 3 °/h
 - Angular Random Walk : 0.1 °/√h
- Initial Bias Error : 720 °/h
 (1σ, -10 °C to +60 °C)
 :1800 °/h
 (1σ, -40 °C to +85 °C)
 : 4 mG (1σ)
- 6 Degrees Of Freedom
 - Triple Gyroscopes : ±400 °/s
 - Tri-Axis Accelerometer : ±8 G/±16 G
- 16/32-bit Data Resolution
- Digital Serial Interface : SPI / UART
- Calibrated Stability
 (Bias, Scale Factor, Axial Alignment)
- Data Output Rate (Max.) : ~2k Sps
- External Trigger Input / External Counter Reset Input
- Delta Angle / Delta Velocity Output
- Attitude Output Accuracy : ±0.3 °
- Calibration Temperature Range : -40 °C to +85 °C
- Operating Temperature Range : -40 °C to +85 °C
- Single Voltage Supply : 3.3 V
- Low Power Consumption : 16 mA (Typ.)



Product Name and Number
M-G330PDG0 : X2G000201000100



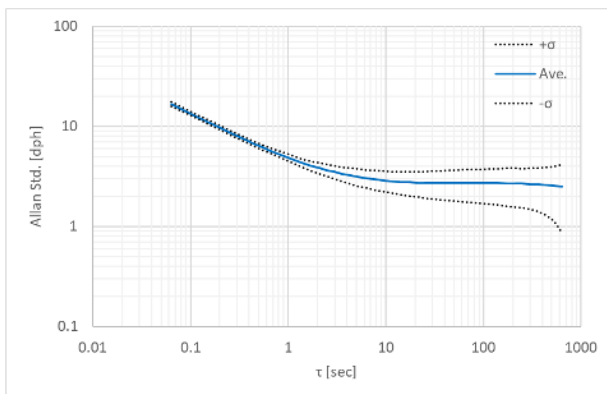
Description

The M-G330PDG0 is a small form factor inertial measurement unit (IMU) with 6 degrees of freedom: triaxial angular rates and linear accelerations, and provides high-stability and high-precision measurement capabilities with the use of high-precision compensation technology. A variety of calibration parameters are stored in memory of the IMU, and are automatically reflected in the measurement data being sent to the application after the power of the IMU is turned on. With general-purpose SPI / UART support for host communications, the M-G330PDG0 reduces technical barriers for users to introduce inertial measurement and minimizes design resources to implement inertial movement analysis and control applications. The features of the IMU such as high stability, high precision, and small size make it easy to create and differentiate applications in various fields of industrial systems.

Application

- Antenna Platform Stabilization
- Camera Gimbals
- Navigation Systems
- Vibration Control and Stabilization
- Pointing and Tracking Systems
- Autonomous Vehicle

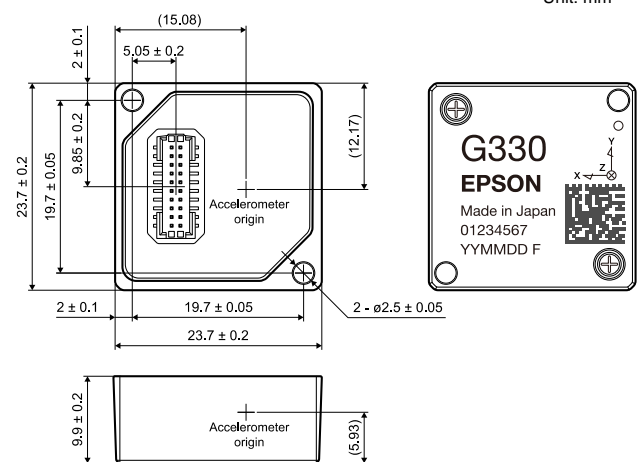
Typical Performance Characteristic



Gyro Allan Variance Characteristic

Outline Dimensions

Unit: mm



Block Diagram

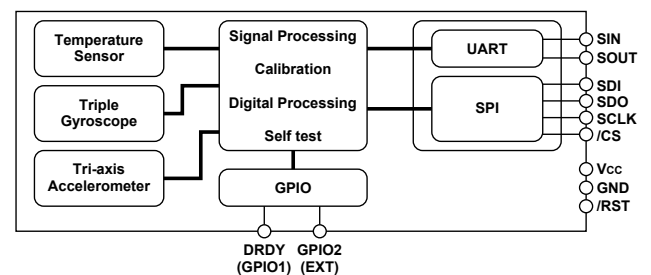


Table of Contents

Revision History	7
Ordering Information	8
Evaluation tools	8
Symbols	9
Notice of the Document	10
1. Product Specifications	11
1.1 Absolute Maximum Ratings	11
1.2 Recommended Operating Condition	11
1.3 Characteristics and Electrical Specifications	12
1.4 Timing Specifications	14
1.5 Connector Pin Layout and Functions	16
2. Mechanical Dimensions	17
2.1 Outline Dimensions	17
2.2 Connector Dimensions	18
3. Typical Performance Characteristics	19
4. Basic Operation	20
4.1 Connection to Host	20
4.2 Operation Mode	21
4.3 Functional Block Diagram	22
4.4 Data Output Timing	23
4.5 Data Ready Signal	24
4.6 Sampling Counter	24
4.7 GPIO	24
4.8 Self Test	24
4.9 External Trigger Input	25
4.10 External Counter Reset Input	27
4.11 Checksum	28
4.12 Automatic Start (For UART Auto Mode Only)	29
4.13 Filter	29
4.14 Delta Angle / Delta Velocity Output	32
4.15 Attitude Output	34
4.16 Quaternion Output	35
4.17 Range Over Function	36
4.18 Frame Alignment Correction	36
5. Digital Interface	37
5.1 SPI Interface	38
5.1.1 SPI Read Timing (Normal Mode)	38
5.1.2 SPI Write Timing (Normal Mode)	39
5.1.3 SPI Read Timing (Burst Mode)	39
5.2 UART Interface	40
5.2.1 UART Read Timing (Normal Mode)	41
5.2.2 UART Read Timing (Burst Mode)	41
5.2.3 UART Write Timing	42
5.2.4 UART Auto Mode Operation	42
5.3 Data Packet Format	43

6. User Registers	49
6.1 BURST Register (Window 0)	52
6.2 MODE_CTRL Register (Window 0).....	52
6.3 DIAG_STAT Register (Window 0)	52
6.4 FLAG (ND / EA) Register (Window 0).....	54
6.5 GPIO Register (Window 0).....	54
6.6 COUNT Register (Window 0).....	55
6.7 RANGE_OVER Register (Window 0).....	55
6.8 TEMP Register (Window 0).....	55
6.9 GYRO Register (Window 0).....	56
6.10 ACCL Register (Window 0)	56
6.11 ID Register (Window 0).....	56
6.12 QUATERNION Register (Window 0).....	57
6.13 DELTA_ANGLE Register (Window 0)	57
6.14 DELTA_VELOCITY Register (Window 0).....	57
6.15 ATTI Register (Window 0).....	58
6.16 SIG_CTRL Register (Window 1)	58
6.17 MSC_CTRL Register (Window 1)	59
6.18 SMPL_CTRL Register (Window 1).....	59
6.19 FILTER_CTRL Register (Window 1)	62
6.20 UART_CTRL Register (Window 1).....	63
6.21 GLOB_CMD Register (Window 1)	63
6.22 BURST_CTRL1 Register (Window 1).....	64
6.23 BURST_CTRL2 Register (Window 1).....	65
6.24 POL_CTRL Register (Window 1)	66
6.25 GLOB_CMD3 Register (Window 1)	66
6.26 ATTI_CTRL Register (Window 1).....	68
6.27 GLOB_CMD2 Register (Window 1)	70
6.28 R_MATRIX Register (Window 1).....	71
6.29 PROD_ID Register (Window 1).....	71
6.30 VERSION Register (Window 1)	71
6.31 SERIAL_NUM Register (Window 1).....	72
6.32 WIN_CTRL Register (Window 0,1)	72
7. Sample Program Sequence	73
7.1 SPI Sequence	73
7.1.1 Power-on Sequence (SPI)	73
7.1.2 Register Read and Write (SPI).....	73
7.1.3 Sampling Data (SPI)	73
7.1.4 Self Test (SPI)	76
7.1.5 Software Reset (SPI)	77
7.1.6 Flash Test (SPI).....	77
7.1.7 Flash Backup (SPI)	77
7.1.8 Filter Setting (SPI).....	78
7.1.9 Attitude Output (SPI).....	78
7.1.10 Frame Alignment Correction (SPI)	80
7.2 UART Sequence	81
7.2.1 Power-on Sequence (UART).....	81
7.2.2 Register Read and Write (UART).....	81
7.2.3 Sampling Data (UART).....	81
7.2.4 Self Test (UART)	84

7.2.5	Software Reset (UART).....	84
7.2.6	Flash Test (UART).....	84
7.2.7	Flash Backup (UART)	85
7.2.8	Initial Backup (UART).....	85
7.2.9	Filter Setting (UART).....	86
7.2.10	Auto Start (UART only).....	86
7.2.11	Attitude Output (UART).....	86
7.2.12	QUATERNION Output (UART).....	88
8.	Handling Notes.....	89
8.1	Cautions for Use	89
8.2	Cautions for Storage	89
8.3	Other Cautions.....	90
8.4	Limited Warranty	90
9.	Contact.....	91

List of Figures

Figure 1.1	SPI Write Timing and Sequence	14
Figure 1.2	SPI Read Timing and Sequence	14
Figure 1.3	SPI Read Timing and Sequence (BURST MODE)	15
Figure 1.4	Connector Pin Assignment	16
Figure 2.1	M-G330PDG0 Outline Dimensions	17
Figure 2.2	Header Pin Dimensions	18
Figure 3.1	Gyro Allan Variance Characteristic	19
Figure 3.2	Gyro Bias vs. Temperature Characteristic	19
Figure 3.3	Gyro Noise Frequency Characteristic	19
Figure 3.4	Accelerometer Allan Variance Characteristic	19
Figure 3.5	Accelerometer Bias vs. Temperature Characteristic	19
Figure 3.6	Accelerometer Noise Frequency Characteristic	19
Figure 4.1	SPI Connection	20
Figure 4.2	UART Connection	20
Figure 4.3	Operational State Diagram	22
Figure 4.4	Functional Block Diagram	22
Figure 4.5	Data Output Timing – Data Output Rate 1 kSps, FIR Filter TAP N = 2	23
Figure 4.6	Data Output Timing – Data Output Rate 500 Sps, FIR Filter TAP N = 4	23
Figure 4.7	Data Ready Signal Timing	24
Figure 4.8	External Trigger Input (UART / SPI Manual Mode)	26
Figure 4.9	External Trigger input Functional Block Diagram	26
Figure 4.10	External Counter Reset Input	28
Figure 4.11	Checksum	28
Figure 4.12	Moving Average Filter Characteristics	29
Figure 4.13	FIR Kaiser Filter Typical Characteristic 1	30
Figure 4.14	FIR Kaiser Filter Typical Characteristic 2	30
Figure 4.15	Delta Angle / Delta Velocity	32
Figure 4.16	Inclination Mode	34
Figure 4.17	Euler Angle Mode	34
Figure 5.1	SPI Read Timing (Normal Mode)	38
Figure 5.2	SPI Write Timing (Normal Mode)	39
Figure 5.3	SPI Timing (Burst Mode)	39
Figure 5.4	UART Bit Format	40
Figure 5.5	UART Read Timing (Normal Mode)	41
Figure 5.6	UART Read Timing (Burst Mode)	42
Figure 5.7	UART Write Timing	42
Figure 5.8	UART Auto Mode Operation	42
Figure 6.1	FLASH_BACKUP and INITIAL BACKUP	64

List of Tables

Table 1.1	Absolute Maximum Ratings	11
Table 1.2	Recommended Operating Conditions	11
Table 1.3	Sensor Specifications	12
Table 1.4	Interface Specifications	13
Table 1.5	Timing Specifications	14
Table 1.6	Pin Function Descriptions	16
Table 2.1	Header Part Number	18
Table 2.2	Socket Part Number	18
Table 4.1	External Trigger Input Timing Requirements	25
Table 4.2	External Counter Reset Input Timing	27
Table 4.3	Transient Response in Number of Samples Based on Output Data Rate vs Filter Tap	31
Table 5.1	SPI Communication Settings	38
Table 5.2	SPI Timing (Normal Mode)	38
Table 5.3	Command Format (Read)	38
Table 5.4	Response Format (Read)	39
Table 5.5	Command Format (Write)	39
Table 5.6	SPI Timing (Burst Mode)	39
Table 5.7	UART Communication Settings	40
Table 5.8	UART Timing	40
Table 5.9	UART Timing ($t_{READRATE}$ Requirements for Burst Mode)	41
Table 5.10	Command Format (Read)	41
Table 5.11	Response Format (Read)	41
Table 5.12	Command Format (Burst Mode)	42
Table 5.13	Command Format (Write)	42
Table 5.14	UART Data Packet Format (UART Burst / Auto Mode) Example: 16-bit Output	43
Table 5.15	UART Data Packet Format (UART Burst / Auto Mode) Example: 32-bit Output	44
Table 5.16	Data Packet Format (SPI Burst Mode) Example: 16-bit Output	45
Table 5.17	Data Packet Format (SPI Burst Mode) Example: 32-bit Output	45
Table 5.18	UART Data Packet Format (UART Burst / Auto Mode) Example: 32-bit Output	46
Table 5.19	Data Packet Format (SPI Burst Mode) Example: 32-bit Output	48
Table 6.1	Register Map	50
Table 6.2	SET_ERR Flag Output Rate and Filter Setting Table Evaluation Table	60
Table 6.3	Output Rate Limitations	61
Table 6.4	Delta Angle: Scale Factor & Range	67
Table 6.5	Delta Velocity: Scale Factor & Range	67
Table 6.6	ATTI_ON = "01", "10" Usage Restriction	68
Table 6.7	Attitude Output Axis Conversion	69
Table 6.8	Attitude Motion Profile Description	70

Revision History

Rev. No.	Date	Page	Description
Rev1.0	2022/12/28	ALL	Newly established
Rev1.1	2023/06/30	P54-55	Corrected Table 6.1 Register Map
		P67-68	Revised the explanation in Section 6.21, GLOB_CMD Register (Window 1)
		P88	Added a new section, "7.2.8 Initial Backup (UART)"
Rev1.2	2025/03/03	ALL	Eliminated the descriptions associated with the M-G366PDG0
		P13	Changed Flash Backup Time to 300ms
		P27-28	Changed "ADC sampling completion" to "sampling timing, etc." in the main text Table 4.2 and Figure 4.10: Changed "t _{ER2A} " to "t _{ER2S} "
		P32-33	Corrected Figure 4.15 for Delta Angle/Delta Velocity output Added annotation for Delta Angle/Delta Velocity output
		P34	Corrected the axis labels in Figure 4.17
		P36	Corrected the description of the range over function
		P55	Revised the explanation of the COUNT Register
Rev1.3	2026/01/30	P66	Corrected the description of POL_CTRL
		P12-13	Corrected the note *4) in Table 1.3 Sensor Specifications
		P32	Corrected Figure 4.15 Delta Angle / Delta Velocity to Filter TAP N=0
		P34	Added conditions for 4.15 Attitude Output
		P55-56	Corrected the notation of 6.8 TEMP Register, 6.9 GYRO Register, and 6.10 ACCL Register to SF:Scale Factor (16bit)
P90	Changed the wording of the Limited Warranty		

Ordering Information

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

Product Model Number	Product Name	Comments
X2G000201000100	M-G330PDG0	This product.

Evaluation tools

Evaluation tools can be provided for this product. Please inquire separately about details.

Product Model Number	Product Name	Comments
X2H000021000100	M-G32EV031	Breakout Board for IMU.
X2H000021000200	M-G32EV041	USB Evaluation Board for IMU. An option to connect M-G330PDG0 to the USB port of a PC.

Symbols**• Compliant with the EU RoHS directive**

* About products without the Pb-Free label

Product terminals are lead-free but the internal components of the product contain lead (high melting point solder lead as well as the lead contained in the glass of an electronic component are both not applicable under the EU RoHS directive).

Notice of the Document

Evaluation board/kit and Development tool important notice

1. This evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the requirements of design for finished products.
2. This evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by the use of it. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. The part used for this evaluation board/kit or development tool may be changed without any notice.

NOTICE: PLEASE READ CAREFULLY BELOW BEFORE USE THIS DOCUMENT

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

©Seiko Epson Corporation 2025. All rights reserved.

1. Product Specifications

1.1 Absolute Maximum Ratings

Table 1.1 Absolute Maximum Ratings

Parameter	Min.	Typ.	Max.	Unit
V _{CC} to GND	-0.3	—	4.8	V
Digital Input Voltage to GND	-0.3	—	V _{CC} + 0.3	V
Digital Output Voltage to GND	-0.3	—	V _{CC} + 0.3	V
Storage Temperature Range	-40	—	85	°C
Acceleration / Shock (Half-sine, 0.5 ms)	—	—	1000	G

Precautions about ESD

Electrostatic discharge (ESD) may damage the product.

When you store or handle the product, take appropriate preventive measures against electrostatic discharge (ESD).

Damages caused by electrostatic discharge (ESD) range from small performance degradation, partial malfunction, to complete breakdown.

This is a high-precision product. Even small performance degradation may cause the product not to conform to the specifications.

1.2 Recommended Operating Condition

Table 1.2 Recommended Operating Conditions

Parameter	Condition	Min.	Typ.	Max.	Unit
V _{CC} to GND		3.15	3.3	3.45	V
Digital Input Voltage to GND		GND	—	V _{CC}	V
Digital Output Voltage to GND		-0.3	—	V _{CC} + 0.3	V
Calibration Temperature Range	Performance parameters are applicable	-40	—	85	°C
Operating Temperature Range		-40	—	85	°C

1.3 Characteristics and Electrical Specifications

Table 1.3 Sensor Specifications

$T_A = 25\text{ }^\circ\text{C}$, $V_{CC} = 3.3\text{ V}$, angular rate = $0\text{ }^\circ/\text{s}$, $\leq \pm 1\text{ G}$, unless otherwise noted

Parameter	Test Conditions / Comments	Min.	Typ.	Max.	Unit
GYRO SENSOR					
Sensitivity					
Output Range		—	± 400	—	$^\circ/\text{s}$
Scale Factor	16 bits	-0.2%	66	+0.2%	LSB/($^\circ/\text{s}$)
	32 bits	-0.2%	$66 \times (2^{16})$	+0.2%	
Nonlinearity (Best fit straight line)	1σ	—	0.05	—	% of FS
Misalignment	1σ , Axis-to-axis, $\Delta = 90^\circ$ ideal	—	0.01	—	$^\circ$
Bias					
Initial Error	1σ , $-10\text{ }^\circ\text{C} \leq T_A \leq +60\text{ }^\circ\text{C}$	—	720	—	$^\circ/\text{h}$
	1σ , $-40\text{ }^\circ\text{C} \leq T_A \leq +85\text{ }^\circ\text{C}$	—	1800	—	
Repeatability	1σ , Turn-on to turn-on ^{*3}	—	36	—	$^\circ/\text{h}$
Bias Instability	Average	—	3	—	$^\circ/\text{h}$
Angular Random Walk	Average	—	0.1	—	$^\circ/\sqrt{\text{h}}$
Linear Acceleration Effect	Average	—	18	—	($^\circ/\text{h}$)/G
Noise Density	$f = 10\text{ Hz}$ to 20 Hz	—	7	—	($^\circ/\text{h}$)/ $\sqrt{\text{Hz}}$, rms
Frequency Property					
3 dB Bandwidth		—	500	—	Hz
ACCELEROMETERS					
Sensitivity					
Output Range		—	$\pm 8/\pm 16$ ^{*7}	—	G
Scale Factor	16 bits, Output Range ± 8	-0.2%	4	+0.2%	LSB/mG
	32 bits, Output Range ± 8	-0.2%	$4 \times (2^{16})$	+0.2%	
	16 bits, Output Range ± 16	-0.2%	2	+0.2%	
	32 bits, Output Range ± 16	-0.2%	$2 \times (2^{16})$	+0.2%	
Nonlinearity (Best fit straight line)	1σ , $< 1\text{ G}$	—	0.1	—	% of FS
Misalignment	1σ , Axis-to-axis, $\Delta = 90^\circ$ ideal	—	0.01	—	$^\circ$
Bias					
Initial Error	1σ , $-40\text{ }^\circ\text{C} \leq T_A \leq +85\text{ }^\circ\text{C}$	—	4	—	mG
Repeatability	1σ , Turn-on to turn-on ^{*3}	—	4	—	mG
Bias Instability	Average	—	34	—	μG
Velocity Random Walk	Average	—	0.03	—	(m/s)/ $\sqrt{\text{h}}$
Noise Density	$f = 10\text{ Hz}$ to 20 Hz	—	70	—	$\mu\text{G}/\sqrt{\text{Hz}}$, rms
Frequency Property					
3 dB Bandwidth		—	333	—	Hz
ATTITUDE OUTPUT					
Output Range	Inclination Mode	-80	—	+80	$^\circ$
	Euler Mode	ANG1: Roll	-45	+45	
		ANG2: Pitch	-180	+180	
		ANG3: Yaw ^{*4}	-180	+180	
Scale Factor	16 bits	—	0.00012207	—	rad/LSB
		—	0.00699411	—	$^\circ/\text{LSB}$
Accuracy ^{*4*6}	1σ , Static	—	0.3	—	$^\circ$
	1σ , Dynamic ^{*5} (100 $^\circ/\text{s}$, Max.)	—	0.3	—	
TEMPERATURE SENSOR					
Scale Factor ^{*1*2}	Output = 0 @ $+25\text{ }^\circ\text{C}$	—	0.00390625	—	$^\circ\text{C}/\text{LSB}$

*1) This is a reference value used for internal temperature compensation. There is no guarantee that the value gives an absolute value of the internal temperature.

*2) This is the temperature scale factor for the upper 16 bits (TEMP_HIGH).

*3) Turn-on to turn-on / Day by day, estimated variation during 5 consecutive days.

*4) Accuracy does not apply to ANG3 axis because it is not compensated for errors caused by bias drift.

*5) Dynamic accuracy is based on measurement data that has been measured from a stationary state.

The accuracy that can be achieved depends on the input movement.

*6) Attitude output accuracy is based on measurement data for GLOB_CMD2[0x16 (W1)] bit[5:4] = "00": modeA.

*7) Selectable by register setting.

- 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.
- The Typ. values in the specifications are average values or 1 σ values.
 - Unless otherwise noted, the Max./Min. values in the specifications are design values or Max./Min. values at the factory tests.
 - Acceleration characteristics do not depend on the output range.

Table 1.4 Interface Specifications

$T_A = 25\text{ }^\circ\text{C}$, $V_{CC} = 3.3\text{ V}$, unless otherwise noted

Parameter	Test Conditions	Min.	Typ.	Max.	Unit
LOGIC INPUTS^{*1}					
Positive Trigger Voltage	LVC MOS Schmitt	1.2	—	2.52	V
Negative Trigger Voltage	LVC MOS Schmitt	0.75	—	1.98	V
Hysteresis Voltage	LVC MOS Schmitt	0.3	—	—	V
Logic 1 Input Current, I_{INH}	$V_{IH} = 3.3\text{ V}$	—	0.1	—	μA
Logic 0 Input Current, I_{INL}	$V_{IL} = 0\text{ V}$	—	0.1	—	μA
Input Capacitance, C_{IN}		—	8	—	pF
RST Voltage Range		0	—	$V_{CC} + 0.3$	V
RST High-level Input Voltage, V_{IH}		$0.8 \times V_{CC}$	—	—	V
RST Low-level Input Voltage, V_{IL}		—	—	$0.2 \times V_{CC}$	V
RST Low Pulse Width		100	—	—	ms
Pull-up Resistor		32	80	224	k Ω
DIGITAL OUTPUTS^{*1}					
Output High Voltage, V_{OH}	$I_{SOURCE} = 1.4\text{ mA}$ LVC MOS	$V_{CC} - 0.4$	—	—	V
Output Low Voltage, V_{OL}	$I_{SINK} = 1.4\text{ mA}$ LVC MOS	—	—	0.4	V
FUNCTIONAL TIMES^{*2}					
Power-On Start-Up Time	Time until data is available	—	—	800	ms
Reset Recovery Time		—	—	800	ms
Flash Test Time		—	—	30	ms
Flash Backup Time		—	—	300	ms
Self Test Time		—	—	80	ms
Filter Setting Time		—	—	1	ms
Attitude_Motion_Profile Setting Time		—	—	1	ms
Data Output Rate	$DOUT_RATE = 0x00$	—	—	2000	Sps
Clock Accuracy		—	—	± 0.001	%
Power Supply	Operating voltage range, V_{CC}	3.15	3.3	3.45	V
Power Supply Current		—	16	—	mA

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

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

Note) The specifications above are not included in the factory test items but their characteristic is confirmed.

1.4 Timing Specifications

Table 1.5 Timing Specifications

T_A = 25 °C, V_{CC} = 3.3 V, unless otherwise noted

Parameter	Description	Min.	Typ.	Max.	Unit
NORMAL MODE					
f _{SCLK}		0.01	—	2.0	MHz
t _{STALL}	Stall period between data	20	—	—	μs
t _{WRITERATE}	Write rate	40	—	—	μs
t _{READRATE}	Read rate	40	—	—	μs
BURST MODE					
f _{SCLK}		0.01	—	1.0	MHz
t _{STALL1}	Stall period between data	45	—	—	μs
t _{STALL2}	Stall period between data	4	—	—	μs
t _{READRATE2}	Read rate	32	—	—	μs
COMMON					
t _{Cs}	Chip select to clock edge	10	—	—	ns
t _{DAV}	SO valid after SCLK edge	—	—	80	ns
t _{DSU}	SI setup time before SCLK rising edge	10	—	—	ns
t _{DHD}	SI hold time after SCLK rising edge	10	—	—	ns
t _{SCLKR} , t _{SCLKF}	SCLK rise/fall times	—	—	20	ns
t _{DR} , t _{DF}	SO rise/fall times	—	—	20	ns
t _{SFS}	High after SCLK edge CS	80	—	—	ns

Note) The specifications above are not included in the factory test items but their characteristic is confirmed.

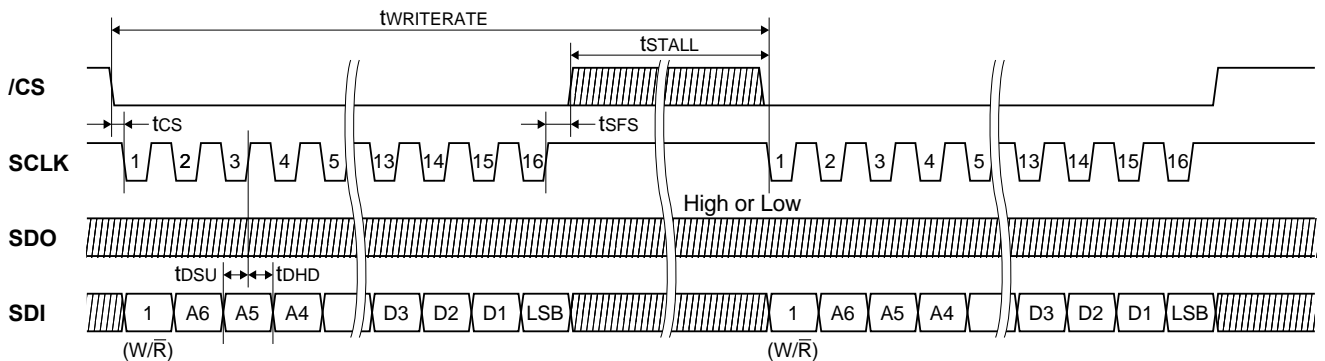


Figure 1.1 SPI Write Timing and Sequence

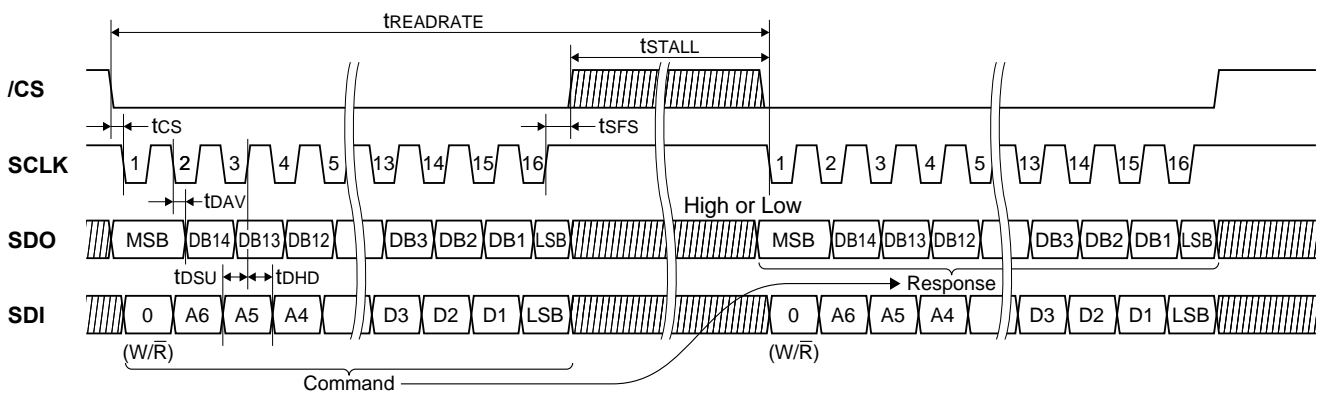


Figure 1.2 SPI Read Timing and Sequence

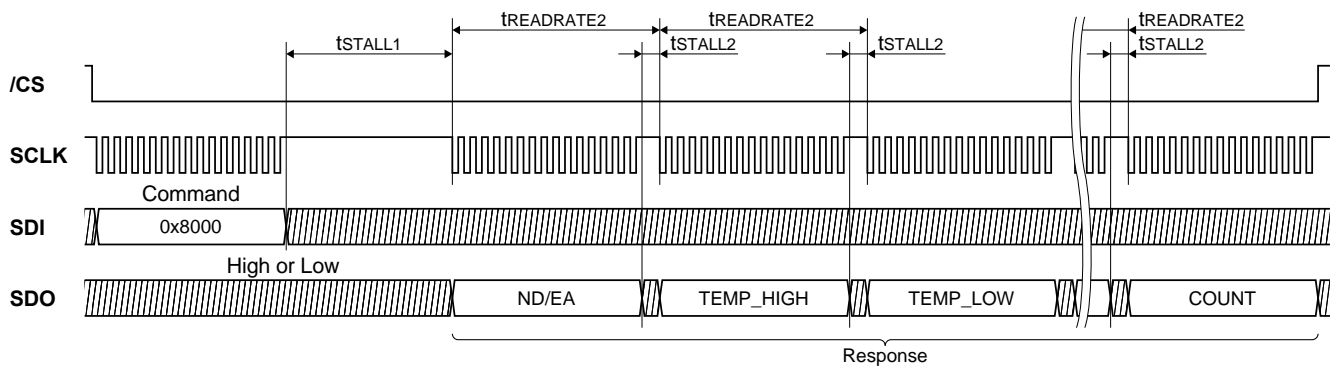


Figure 1.3 SPI Read Timing and Sequence (BURST MODE)

1.5 Connector Pin Layout and Functions

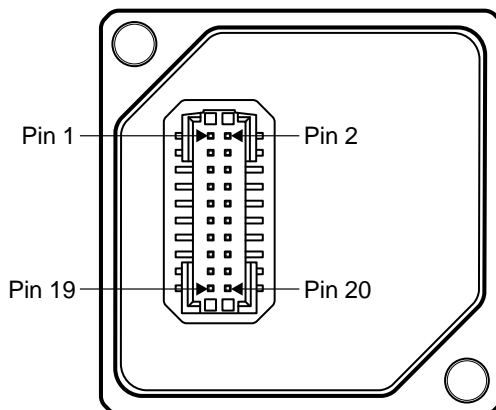


Figure 1.4 Connector Pin Assignment

Table 1.6 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 (GPIO1)	I/O	Data Ready ^{*3} (General Purpose I/O1)
14	GPIO2 (EXT)	I/O	General Purpose I/O2 ^{*4} (External Trigger Input or External Counter Reset Input)
16	/RST	I	Reset ^{*5}
10, 11, 12	V _{CC}	S	Power Supply 3.3 V
3, 4, 8, 15	GND	S	Ground
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) Connect either SPI or UART but not both. Connecting both SPI and UART at the same time may result in malfunction of the device. Regarding unused pins, please connect unused input pins to V_{CC} through resistor.

*3) Regarding Pin function selection, please refer to the **DRDY_ON** at register MSC_CTRL[0x02 (W1)] bit[2]

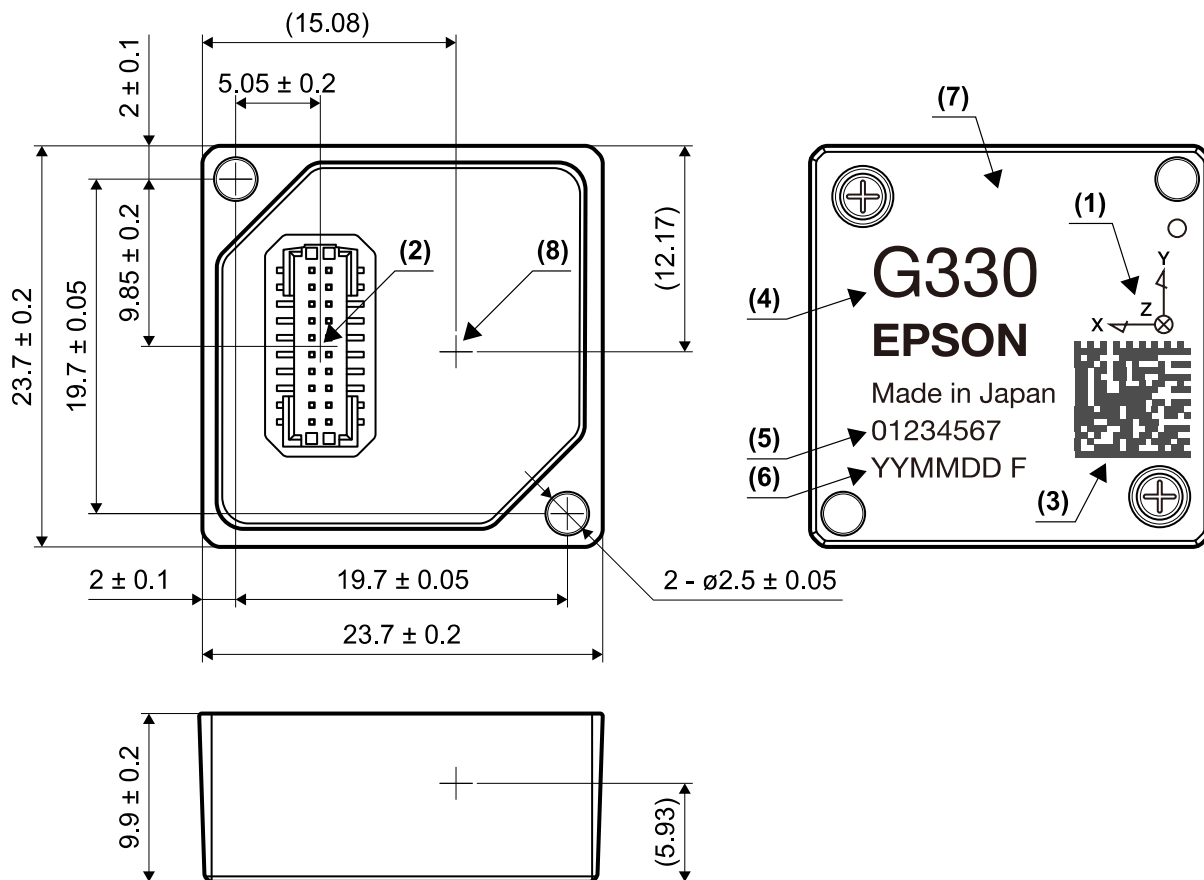
*4) Regarding Pin function selection, please refer to the **EXT_SEL** at register MSC_CTRL[0x02 (W1)] bit[7:6]

*5) If the /RST pin is not used, keep the pin at High (V_{CC}) voltage level.

Note) All input pins have weak pull up resistors inside the IMU.

2. Mechanical Dimensions

2.1 Outline Dimensions



* (data) reference dimensions

(Unit: mm)

No.	Description
(1)	Definition of sensor axes
(2)	Connector position
(3)	Matrix code (Data Matrix) Including product name, serial number, date, and factory code
(4)	Product name
(5)	Serial number
(6)	Date and factory code
(7)	Frame ground
(8)	Accelerometer origin

Figure 2.1 M-G330PDG0 Outline Dimensions

The right-hand screw rotation direction of the gyro sensor is positive (+).

2.2 Connector Dimensions

Figure 2.2 and Table 2.1 describes the connector manufacturer and the model number of the header built into the IMU.

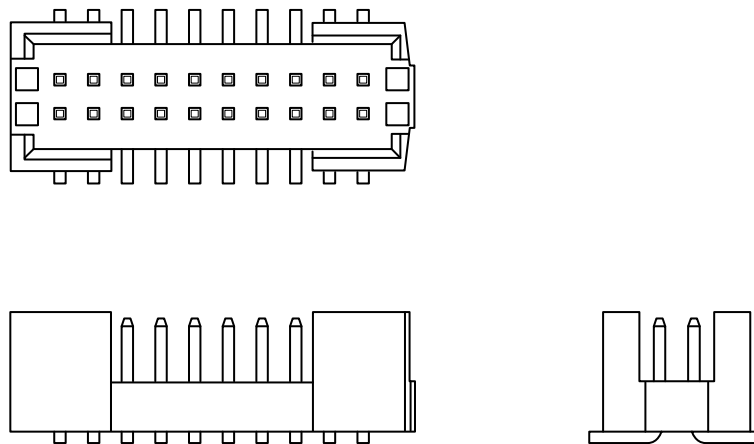


Figure 2.2 Header Pin Dimensions

Table 2.1 Header Part Number

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

* END SHROUDS is MOLDED TO POSITION END SHROUDS

Table 2.2 shows the connector manufacturer and the model number of the recommended socket used at the host side.

Table 2.2 Socket Part Number

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

3. Typical Performance Characteristics

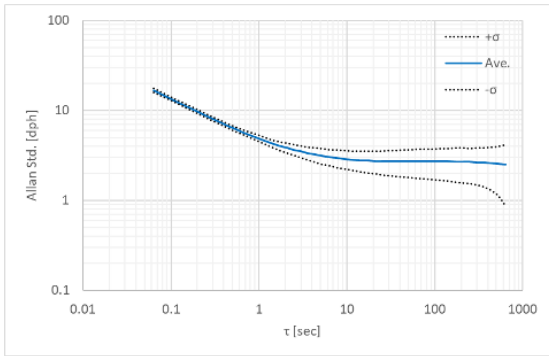


Figure 3.1 Gyro Allan Variance Characteristic

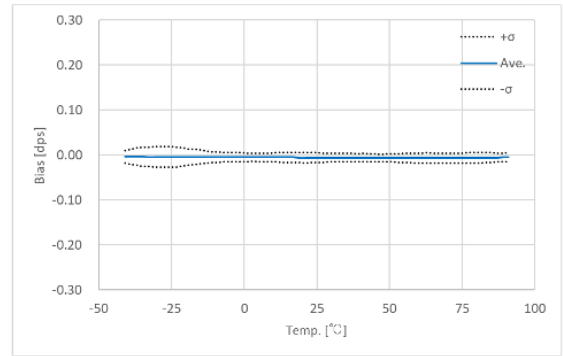


Figure 3.2 Gyro Bias vs. Temperature Characteristic

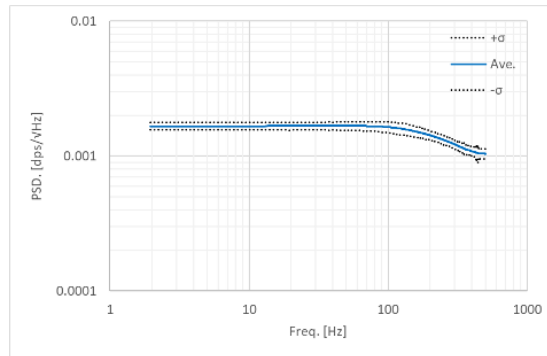


Figure 3.3 Gyro Noise Frequency Characteristic

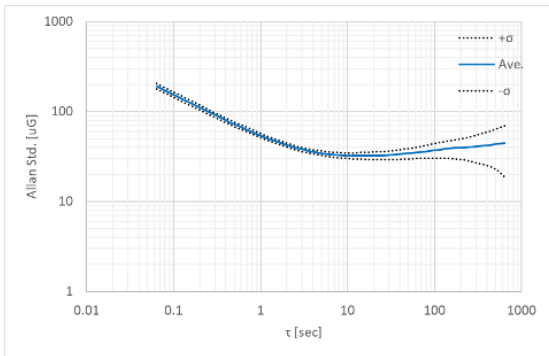


Figure 3.4 Accelerometer Allan Variance Characteristic

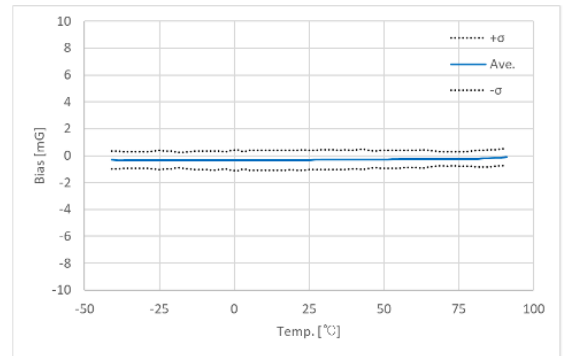


Figure 3.5 Accelerometer Bias vs. Temperature Characteristic

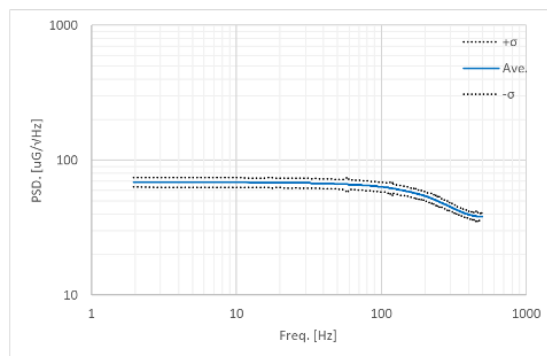


Figure 3.6 Accelerometer Noise Frequency Characteristic

The product characteristics shown above are typical examples and are not guaranteed as specifications.

4. Basic Operation

4.1 Connection to Host

The device is connected to the host via SPI or UART. The following is an example of the connection.

- NOTE:
- Connect either SPI or UART but not both. Connecting both SPI and UART at the same time may result in malfunction of the device.
 - Refer to Table 1.6 Pin Function Descriptions for the connection of unused pins.

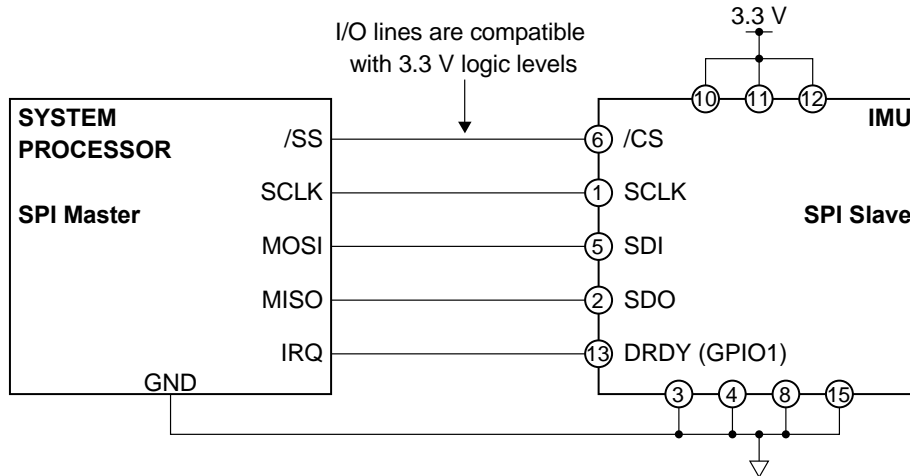


Figure 4.1 SPI Connection

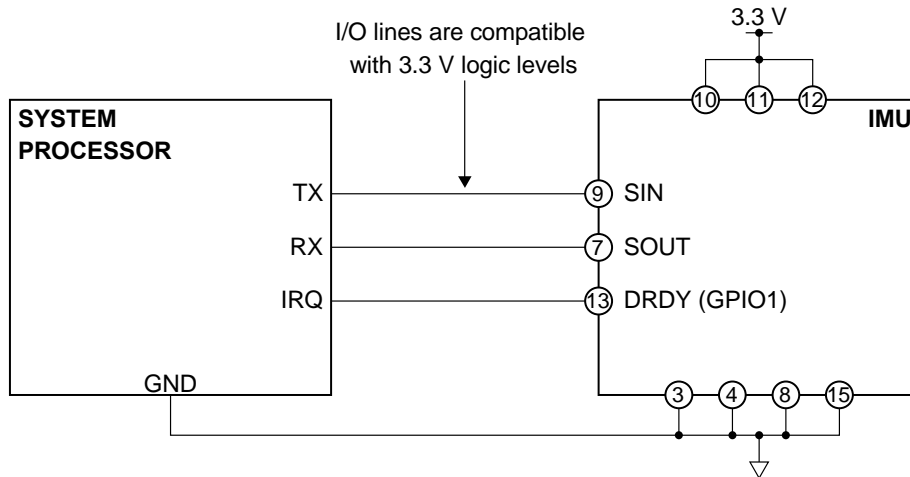


Figure 4.2 UART Connection

4.2 Operation Mode

The device has the following two operation modes. Only when the UART interface is used, Sampling mode has two submodes: Manual mode and Auto mode.

- (1) Configuration mode
- (2) Sampling mode
 - Manual mode
 - Auto mode (for UART only)

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. Configure various operational settings in Configuration mode^(*). After configuration is completed, go to the sampling mode to read out the temperature, angular rate, and acceleration data. To change the operation mode, write to **MODE_CMD** (MODE_CTRL[0x02 (W0)] bit[9:8]). When software reset is executed by writing "1" to **SOFT_RST** (GLOB_CMD[0x0A (W1)] bit[7]), internal initialization is executed and then the device goes into Configuration mode regardless of the current operation mode.

When the UART interface is used, writing to **UART_AUTO** (UART_CTRL[0x08 (W1)] bit[0]) can switch between the Manual mode and the Auto mode^(**).

NOTE: When the SPI interface is used, Manual mode must be selected. Otherwise, the device does not work properly.

*1) 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 **GPIO_DATA** (GPIO[0x08 (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])

*2) 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

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

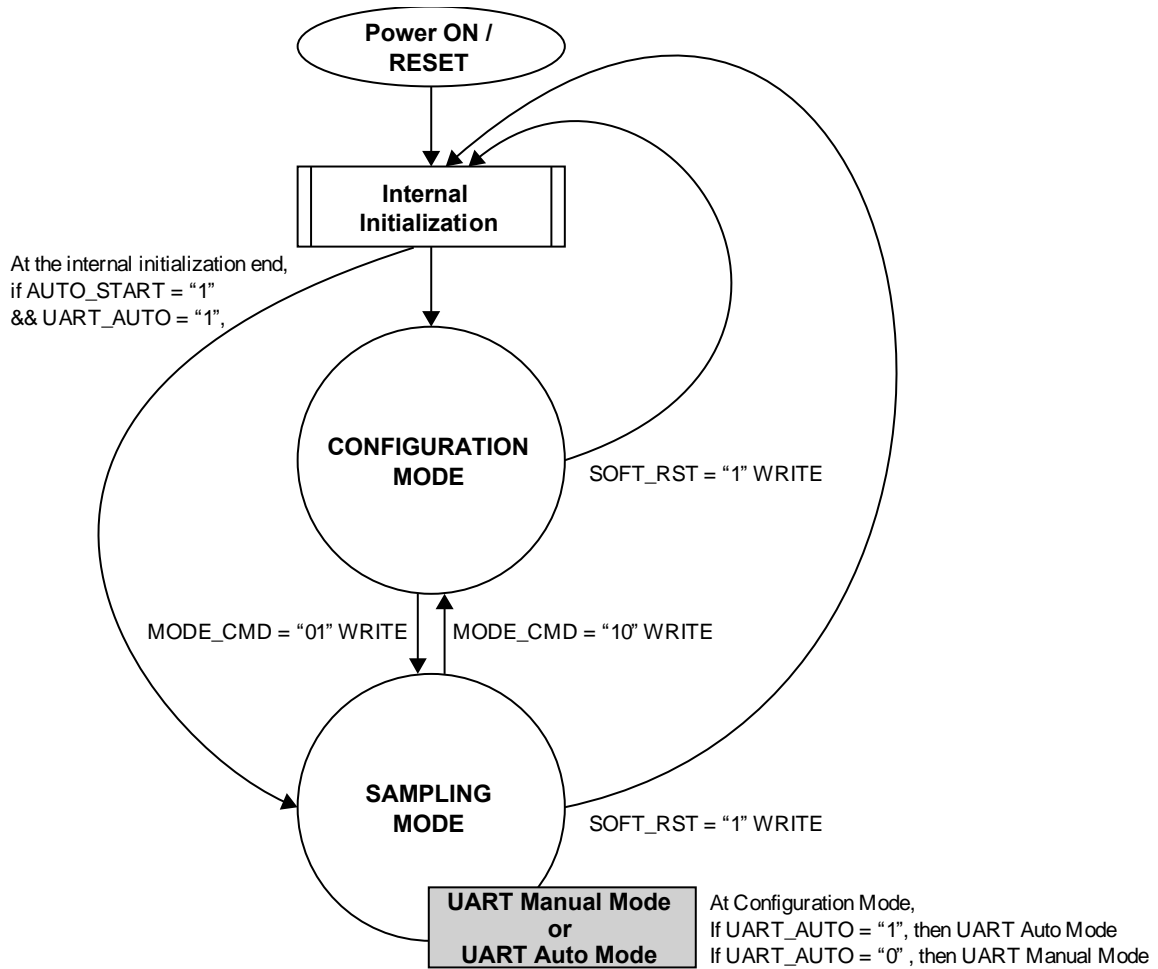


Figure 4.3 Operational State Diagram

4.3 Functional Block Diagram

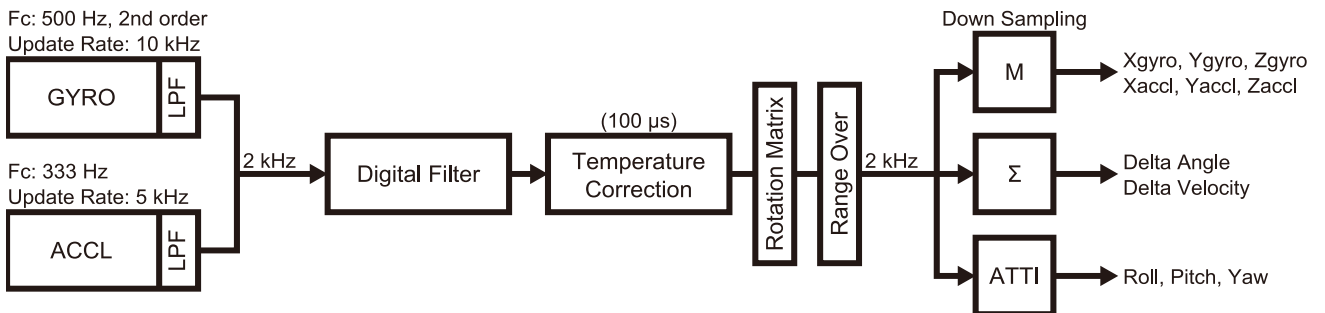


Figure 4.4 Functional Block Diagram

4.4 Data Output Timing

Data Output Rate: 1 kSps
 FIR Filter TAP N = 2

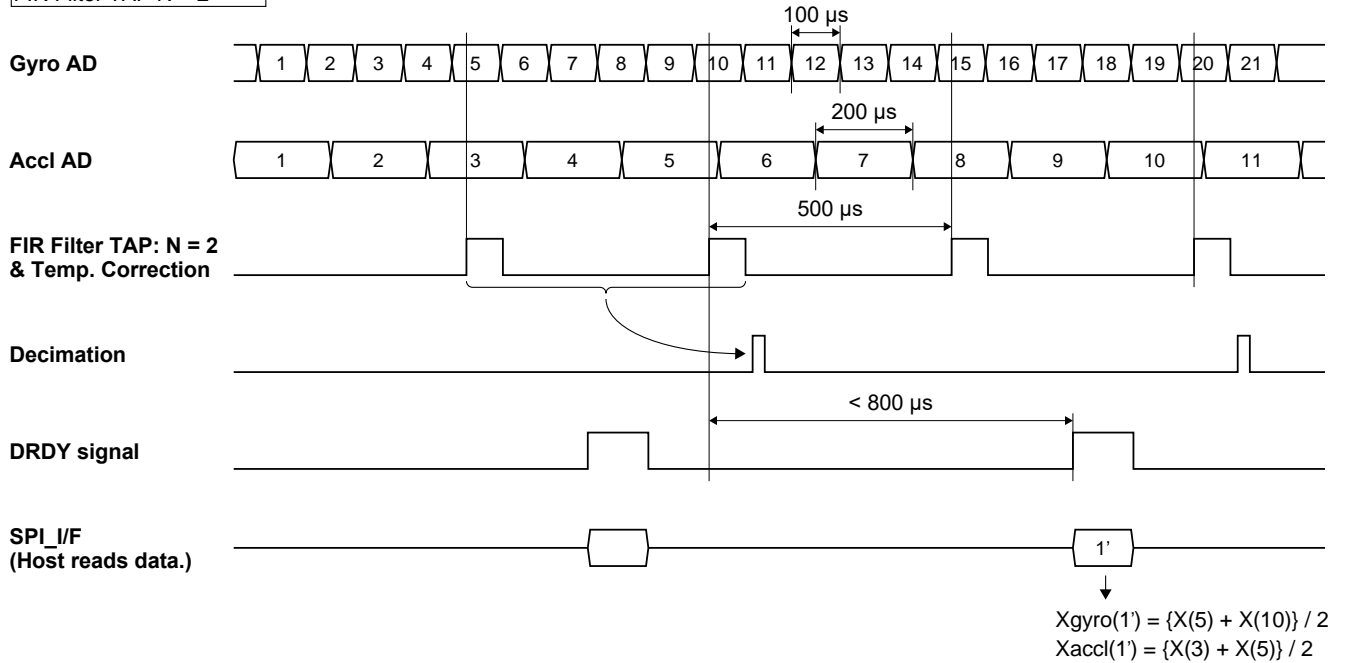


Figure 4.5 Data Output Timing – Data Output Rate 1 kSps, FIR Filter TAP N = 2

Data Output Rate: 500 Sps
 FIR Filter TAP N = 4

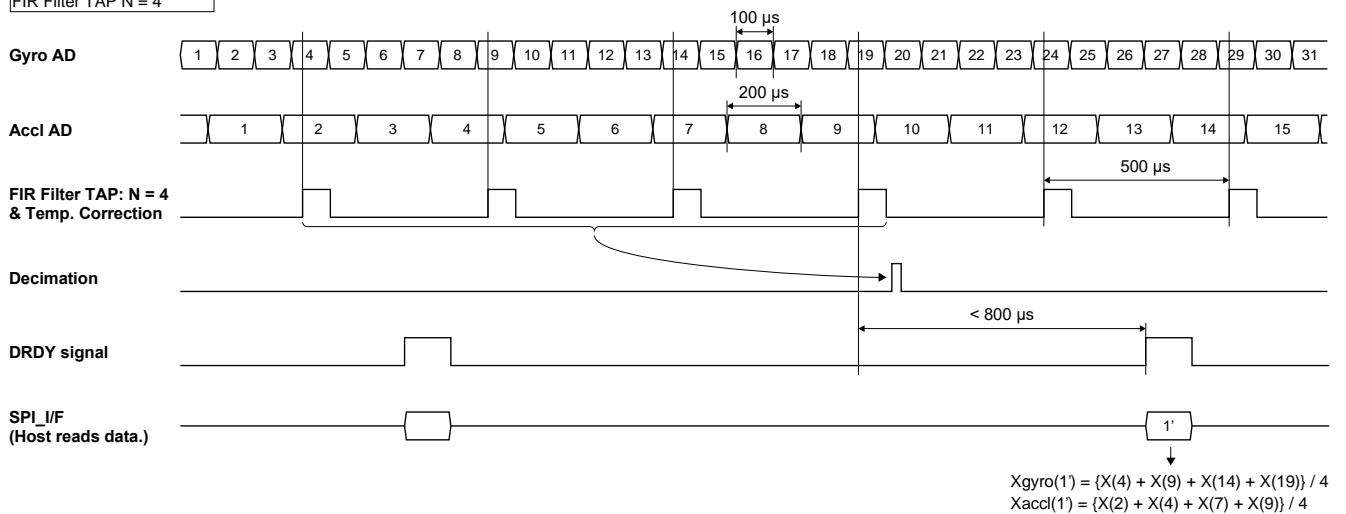


Figure 4.6 Data Output Timing – Data Output Rate 500 Sps, FIR Filter TAP N = 4

4.5 Data Ready Signal

The Data Ready signal is asserted when one sampling cycle completes and the registers are updated with new sensor values. When the sensor values are read out, the Data Ready signal becomes negated. In case of UART Auto mode, the Data Ready signal becomes negated just before data is output.

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

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 **ND_EN** (SIG_CTRL[0x00 (W1)] bit[15:9] [7:2]), the Data Ready signal will not be asserted. On the other hand, if all the sensor values enabled in **ND_EN** (SIG_CTRL[0x00 (W1)] bit[15:9]) are not read out, the Data Ready signal is kept asserted and never becomes negated.

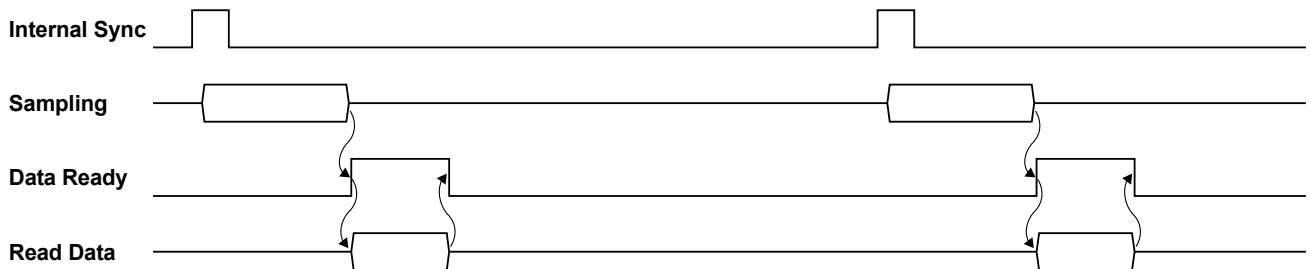


Figure 4.7 Data Ready Signal Timing

4.6 Sampling Counter

By reading the COUNT[0x0A (W0)] register, the counter value can be read which is incremented based on the sampling completion timing of the internal A/D converter. The count interval is 500 μ sec/count and is based on the precision of the internal reference oscillator (crystal).

Additionally, during UART / SPI Burst mode and in UART Auto mode, the counter value can be included in the burst response by setting **COUNT_OUT** (BURST_CTRL1[0x0C (W1)] bit[1]). For information about the response format, refer to 5.3 Data Packet Format.

4.7 GPIO

The device has two general purpose I/O ports (GPIO). By accessing the GPIO[0x08 (W0)] register, the direction (in/out) of each port can be configured and data can be read/written to. The GPIO port can be read in the normal mode, and also in the UART Burst mode or UART Auto mode.

GPIO1 is shared with the Data Ready signal. The switch between GPIO1 and the Data Ready signal can be controlled by **DRDY_ON** (MSC_CTRL[0x02 (W1)] bit[2]). When "0" is written to **DRDY_ON**, GPIO1 acts as a general purpose I/O port.

GPIO2 is shared with the EXT signal (External Trigger Input or External Counter Reset). The switch of GPIO2 and the EXT signal can be controlled by **EXT_SEL** (MSC_CTRL[0x02 (W1)] bit[7:6]). When "00" is written to **EXT_SEL**, GPIO2 acts as a general purpose I/O port.

4.8 Self Test

The self test function can be used to check whether the outputs of the gyroscope and the accelerometer are within the pre-determined range and operating properly. For the gyroscope, the test result is OK if the bias of the output for each X-, Y-, or Z-axis is close to zero when the device is not moving. For the accelerometer, the test result is OK if the absolute value of the output as a three-dimensional vector is equal to the gravitational acceleration. 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.

For information about the execution time of the self test, see "Self Test Time" in Table 1.4 Interface Specifications.

To use the self test function, see the descriptions of **SELF_TEST** (MSC_CTRL[0x02 (W1)] bit[10]) and **ST_ERR_ALL** (DIAG_STAT[0x04 (W0)] bit[1]).

NOTE: When executing the self-test, be sure to disable the external trigger function.

MSC_CTRL[0x02 (W1)] bit[7:6] = "00" or "01"

4.9 External Trigger Input

The External Trigger Input function provides control of the sample data output timing by using an externally supplied input pulse signal to the GPIO2 (EXT) pin. By enabling the **EXT_SEL** (MSC_CTRL[0x02 (W1)] bit[7:6]), GPIO2 pin can be used as the External Trigger Input pin. The polarity of the External Trigger Input is a positive pulse.

NOTE: The External trigger input function cannot be used in combination with the Attitude output function (ATTI_ON[0x15 (W1)] bit [10:9] = "10")

To enable the External Trigger Input function, program the settings as shown below.

- Set **EXT_SEL** (MSC_CTRL[0x02 (W1)] bit[7:6]) to "11" to enable the external trigger input function.
- For normal angular rate and linear acceleration output (**ATTI_ON** = "00"), set **FILTER_SEL** according to the External Trigger Input frequency to FILTER_CTRL[0x06 (W1)] bit[4:0]. The valid **FILTER_SEL** settings are described in Table 6.2 for a selected **DOUT_RATE**. But in this case, the table refers to the valid combination of External Trigger Input frequency and **FILTER_SEL** (instead of **DOUT_RATE** sps). If the External Trigger Input frequency is between two **DOUT_RATE** sps settings, choose the lower **DOUT_RATE** sps setting to determine supported **FILTER_SEL** settings.

NOTE: For normal angular rate and linear acceleration output, the minimum External Trigger Input frequency is 15.625 Hz.

- For Delta Angle / Delta Velocity output (**ATTI_ON** = "01"), **FILTER_SEL** can be set to TAP = 0 or other values depending on the application.

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

For UART Auto mode:

The FIR Filter & Temp Correction processing is performed at the internal 2 kHz rate timing for Gyro and Accl sampling data. When the external trigger input signal is asserted, the current sampling data after the FIR Filter & Temp Correction processing has been performed is automatically sent to the host.

For UART / SPI Manual mode and SPI Burst mode

The FIR Filter & Temp Correction processing is performed at the internal 2 kHz rate timing for Gyro and Accl sampling data. When the external trigger input signal is asserted, the current sampling data after FIR Filter & Temp Correction processing is updated in each register, and the Data Ready signal is asserted. The host should read the sampling data in synchronization with Data Ready.

The External Trigger Input timing requirements and diagrams are shown in Table 4.1, Figure 4.8, and Figure 4.9.

Table 4.1 External Trigger Input Timing Requirements

Parameter	Description	Min.	Max.	Unit
t _{ETW}	External Trigger Input width	100	–	ns
t _{ETC} ^{*1}	External Trigger Input cycle	1	1000	ms
t _{ETA2T_G}	Time from Gyro ADC's completion to External Trigger Input (This is a precision of External Sync.)	0	600	μs
t _{ETA2T_A}	Time from Accl ADC's completion to External Trigger Input (This is a precision of External Sync.)	0	700	μs
t _{ETD1} ^{*2}	Delay time from External Trigger Input to DRDY asserted	–	300	μs

*1) For normal angular rate and linear acceleration output (**ATTI_ON** = "00"), the minimum External Trigger Input frequency is 15.625 Hz, because the strongest low pass filtering is moving average TAP = 128.

*2) This does not include group delay of the internal filter.

Data Output Rate: 1 kSps
 FIR Filter TAP N = 2

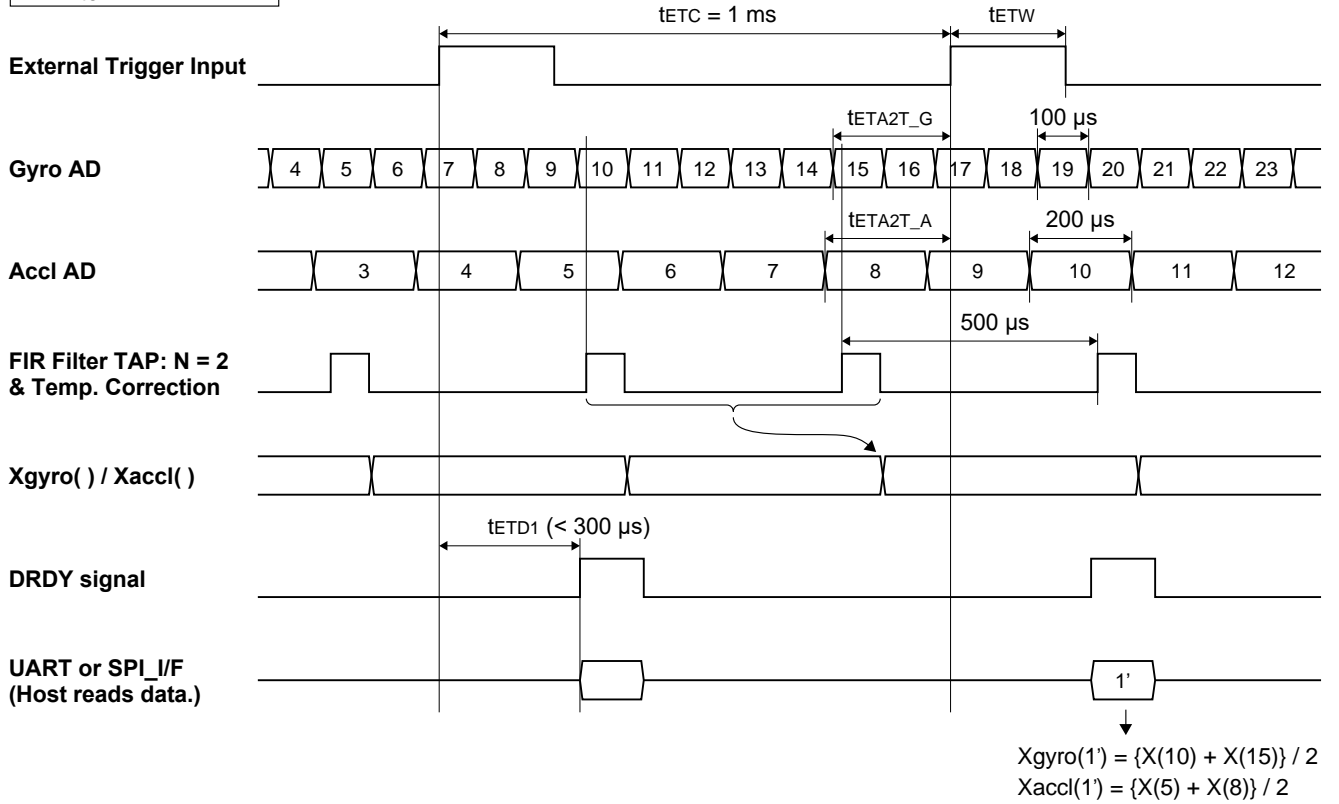


Figure 4.8 External Trigger Input (UART / SPI Manual Mode)

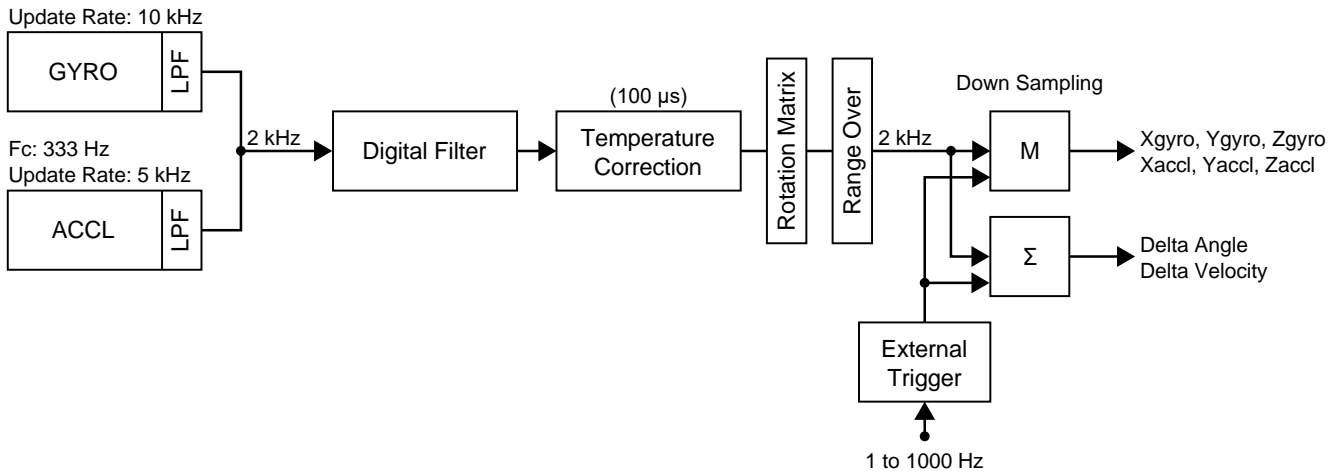


Figure 4.9 External Trigger input Functional Block Diagram

4.10 External Counter Reset Input

The External Counter Reset Input feature allows you to measure the time offset between the next sampling instant from an externally supplied input trigger on the GPIO2 (EXT) pin.

This function is enabled by writing to **EXT_SEL** (MSC_CTRL[0x02 (W1)] bit [7:6]) to select GPIO2 for use as an External Counter Reset Input pin. The active polarity of the input signal is positive pulse.

The following describes the operation when this function is active:

- The IMU has an internal 16-bit up counter incrementing at 62.5 kHz.
- The counter begins counting starting from 0 ⁽¹⁾ when Sampling mode begins. The counting resolution is 16 μ s.
- The counter can be reset by assertion of an external signal on the External Counter Reset Input pin. After the counter is reset, the count value is cleared and begins incrementing again from 0.
- The counter value is transferred at the sampling timing and stored in the COUNT[0x0A (W0)] register before the Data Ready signal is asserted.
- The Host can obtain the time offset from assertion of the most recent External Counter Reset Input signal to the sampling timing by reading the sampling data with the counter value when Data Ready signal is asserted.
- The counter stops counting ⁽²⁾ when Sampling mode is stopped.
- The counter will roll over and increment from 0 again if the count value increments past 65535.

*1) Enter Sampling mode from Configuration mode

*2) Leave Sampling mode and enter Configuration mode

NOTE: • When the External Counter Reset Input function is enabled, the COUNT[0x0A (W0)] register stores the counter value instead of the sampling count.

- This function cannot be used together with the Attitude output (ATTI_ON[0x15 (W1)] bit [10:9] = "10").

The timing specification and timing diagram for the External Counter Reset Input function are shown in Table 4.2 and Figure 4.10.

Table 4.2 External Counter Reset Input Timing

Parameter	Description	Min.	Max.	Unit
t _{ERW}	External Reset Input Width	100	–	ns
t _{ERC}	External Reset Input Cycle	1	1000	ms
t _{ER2S}	Time from External Reset Input to Sampling	(count ^{*1} x 16) + Δ _{ER2S}		μ s
Δ _{ER2S}	Precision of t _{ER2S}	-150	150	μ s

*1) The count value is read from register COUNT[0x0A (W0)] as indicated.

Data Output Rate: 1 kSps
 FIR Filter TAP N = 2

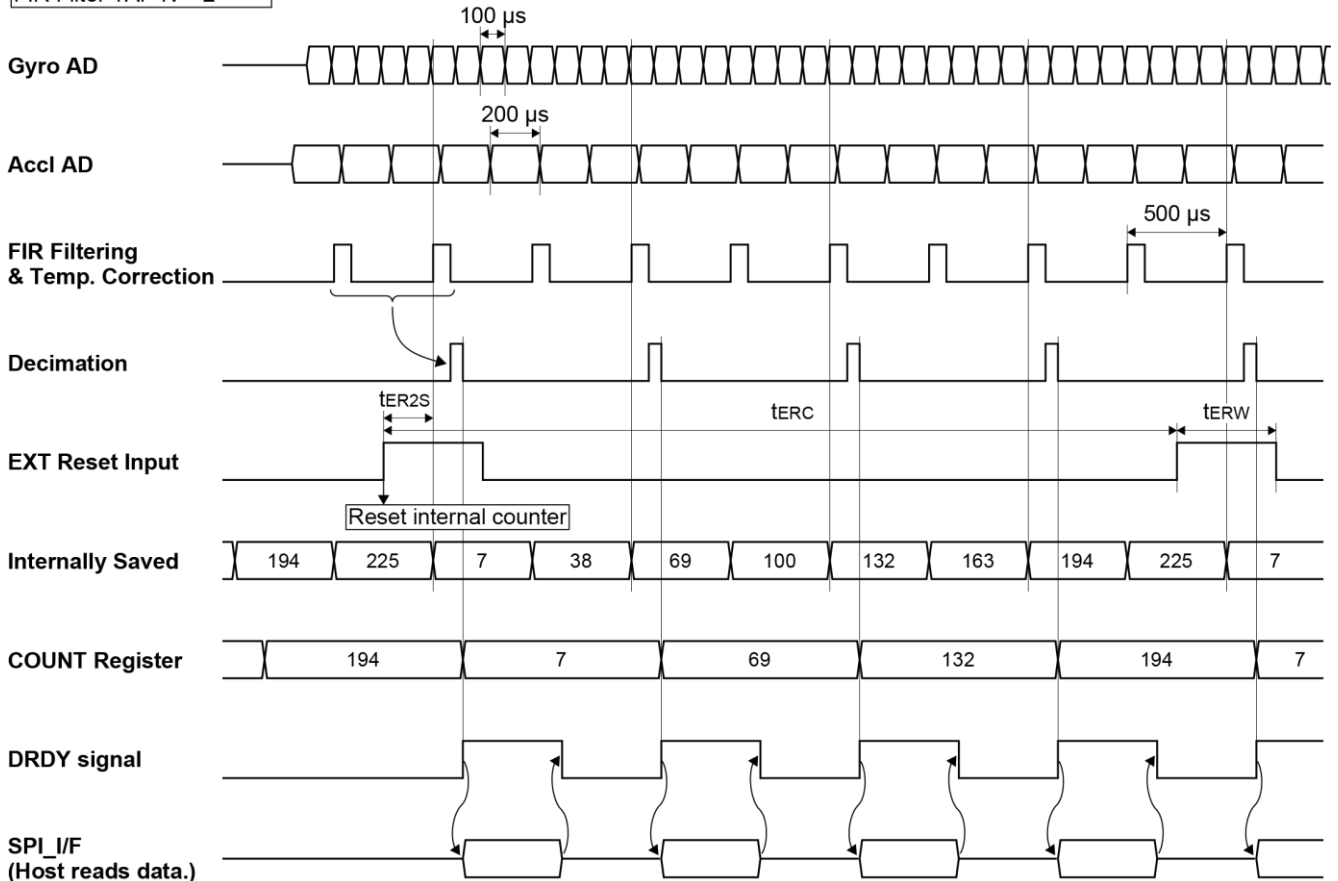


Figure 4.10 External Counter Reset Input

4.11 Checksum

A checksum can be appended to the response data during UART / SPI Burst mode or UART Auto mode by enabling this function in **CHKSM_OUT** (BURST_CTRL1[0x0C (W1)] bit[0]).

The checksum range of the data content is calculated immediately after the address byte (AD = 0x80) of the response data up to (not including) the delimiter byte (CR = 0x0D). The calculation method of checksum is a simple addition of the data content in units of 16 bits, and the resulting sum is truncated to 16 bits and appended as checksum just before the delimiter byte (CR = 0x0D).

Example:

Because the sum total is “611B4” for the following response data stream, the checksum is “11B4”:
 “FE01 C455 4000 0052 33C0 0043 7BC8 004A 2608 FD73 3AA0 FF75 4C30 1F53 8FD0 0600 0014”

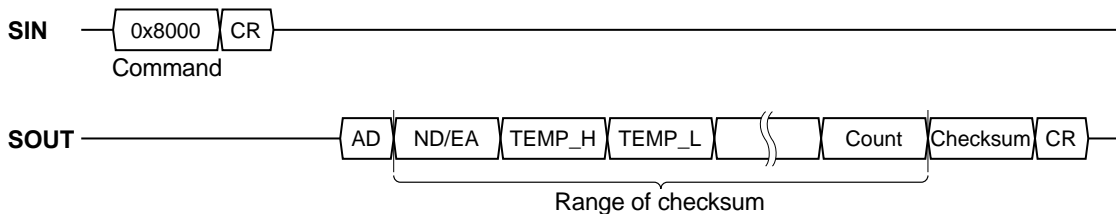


Figure 4.11 Checksum

4.12 Automatic Start (For UART Auto Mode Only)

The Automatic Start function, when enabled, allows the device to automatically enter Sampling mode after completing internal initialization when power is supplied or the IMU is reset. This function is designed to be used in conjunction with the UART Auto mode. Please refer to Figure 4.3 for the state transition.

Follow the procedure below to enable the Automatic Start function:

- Write “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 “1” to **FLASH_BACKUP** (**GLOB_CMD[0x0A (W1)]** bit[3]). After completion of the **FLASH_BACKUP** command, confirm the results by reading **FLASH_BU_ERR** (**DIAG_STAT[0x04 (W0)]** bit[0]).
- The IMU will automatically enter Sampling mode after the power supply is cycled, or a hardware reset, or a software reset command is executed.

The Automatic Start function can be enabled simultaneously with the External Trigger Input function.

Follow the procedure below to enable the Automatic Start with External Trigger Input function:

- Write “1” to both **UART_AUTO** (bit[0]) and **AUTO_START** (bit[1]) of **UART_CTRL[0x08 (W1)]**.
- Write “10” to **EXT_SEL** (**MSC_CTRL[0x02 (W1)]** bit[7:6]) to enable the External Trigger Input. Please connect the external trigger input signal to the GPIO2 pin.
- Store the current register settings to non-volatile memory by writing “1” to **FLASH_BACKUP** (**GLOB_CMD[0x0A (W1)]** bit[3]). After completion of the **FLASH_BACKUP** command, confirm the results by reading **FLASH_BU_ERR** (**DIAG_STAT[0x04 (W0)]** bit[0]).
- The IMU will automatically enter Sampling mode after the power supply is cycled, or a hardware reset, or a software reset command is executed.

4.13 Filter

This device contains built-in user configurable digital filters that are applied to the sensor data. The type of filter (moving average filter or FIR Kaiser filter) and the numbers of TAPs can be set with the **FILTER_CTRL[0x06 (W1)]** register.

(1) Moving Average Filter

TAP setting can be $N = 2, 4, 8, 16, 32, 64, \text{ or } 128$.

Figure 4.12 shows the characteristics of this filter.

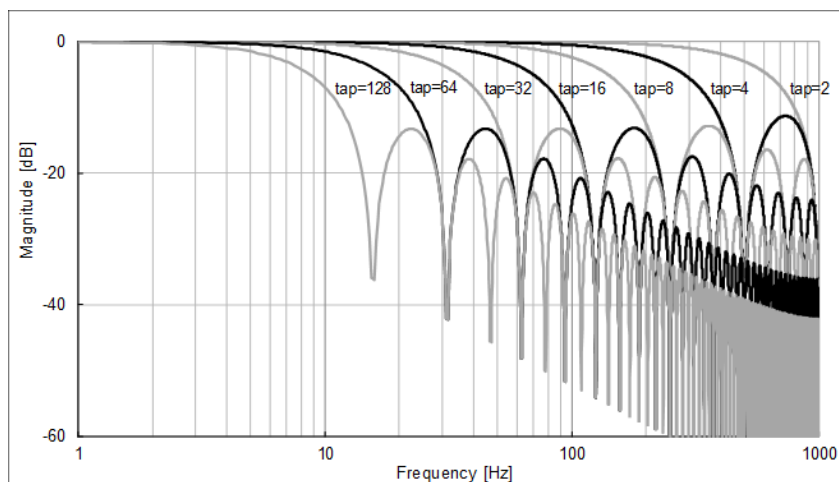


Figure 4.12 Moving Average Filter Characteristics

(2) FIR Kaiser Filter

Uses Kaiser Window (parameter = 8).

TAP setting can be N = 32, 64, or 128 with cutoff frequency $f_c = 50, 100, 200,$ or 400Hz .

Figure 4.13 and Figure 4.14 show the typical characteristic of this filter.

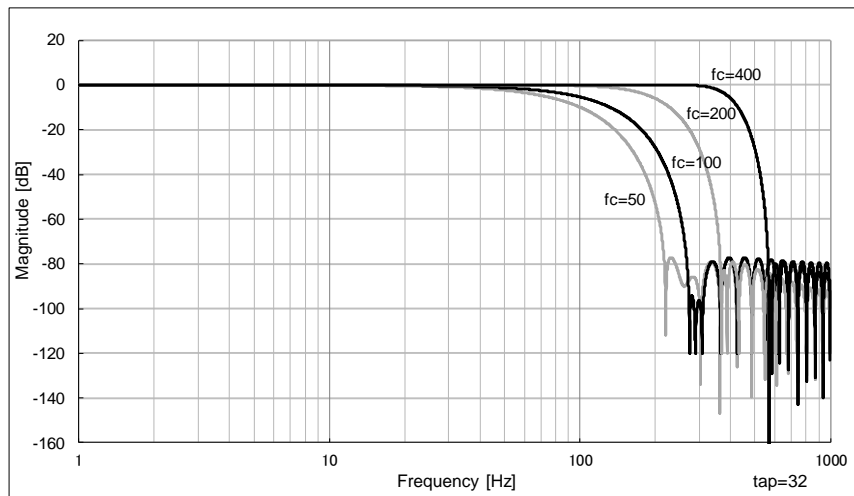


Figure 4.13 FIR Kaiser Filter Typical Characteristic 1

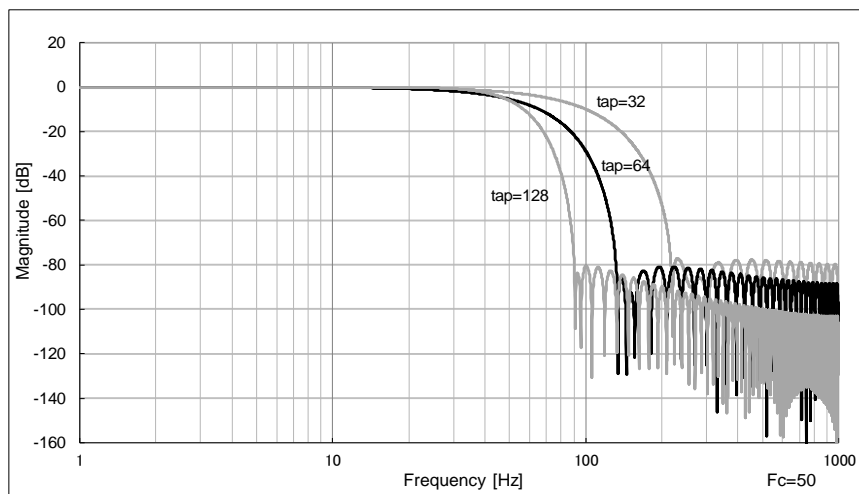


Figure 4.14 FIR Kaiser Filter Typical Characteristic 2

Please note that the transient response of the digital filter is a maximum of 127 samples from the sampling start time and varies depending on the output data rate and the filter tap setting. Refer to Table 4.3 which describes the transient response in terms of number of samples for valid combinations of output data rate and filter tap setting.

Table 4.3 Transient Response in Number of Samples Based on Output Data Rate vs Filter Tap

	TAP0	TAP2	TAP4	TAP8	TAP16	TAP32	TAP64	TAP128
2000 sps	0	1	3	7	15	31	63	127
1000 sps		0	1	3	7	15	31	63
500 sps			0	1	3	7	15	31
400 sps				1	3	6	12	25
250 sps				0	1	3	7	15
200 sps					1	3	6	12
125 sps					0	1	3	7
100 sps						1	3	6
80 sps						1	2	5
62.5 sps						0	1	3
50 sps							1	3
40 sps							1	2
31.25 sps							0	1
25 sps								1
20 sps								1
15.625 sps								0

	TAP32 Fc50	TAP32 Fc100	TAP32 Fc200	TAP32 Fc400	TAP64 Fc50	TAP64 Fc100	TAP64 Fc200	TAP64 Fc400	TAP128 Fc50	TAP128 Fc100	TAP128 Fc200	TAP128 Fc400
2000 sps	31	31	31	31	63	63	63	63	127	127	127	127
1000 sps	15	15	15	15	31	31	31	31	63	63	63	63
500 sps	7	7	7		15	15	15		31	31	31	
400 sps	6	6	6		12	12	12		25	25	25	
250 sps	3	3			7	7			15	15		
200 sps	3	3			6	6			12	12		
125 sps	1				3				7			
100 sps	1				3				6			
80 sps												
62.5 sps												
50 sps												
40 sps												
31.25 sps												
25 sps												
20 sps												
15.625 sps												

4.14 Delta Angle / Delta Velocity Output

Delta-Angle / Delta-Velocity Output is the function to output integrated angle increments and integrated velocity increments by mathematical accumulation of the angular rate and linear acceleration sensor values.

The Delta Angle register and Delta Velocity register[0x64–0x7B (W0)] represent the 32-bit numerical integration of angular rate and linear acceleration value before the down-sampling block. The integration of angular rate and linear acceleration value is reset when the Host reads the Delta Angle register and Delta Velocity register or automatically in UART Auto mode after sending the output data.

For example, in case of down-sampling ratio 4:1 (**DOUT_RATE** of **SMPL_CTRL**[0x05 (W1)] = 0x02), the integration data of four x-axis angular rate and x-axis linear acceleration values after filtering and temperature correction is stored in **XDLTA_HIGH** and **XDLTA_LOW** (Register[0x64–0x67 (W0)]), and **XDLTV_HIGH** and **XDLTV_LOW** (Register[0x70–0x73 (W0)]) respectively in 32-bit data format. Figure 4.15 shows the timing diagram in the case of down-sampling ratio 4:1, 500 sps and Filter TAP: N = 0.

- Point A: By reading **XDLTA_HIGH** and **XDLTA_LOW** from the Host at point A' (2 ms before point A), the integration of angular rate and acceleration value is reset to 0.
- Point B: "55" is output after integration of values 15, 20, 25, -5.

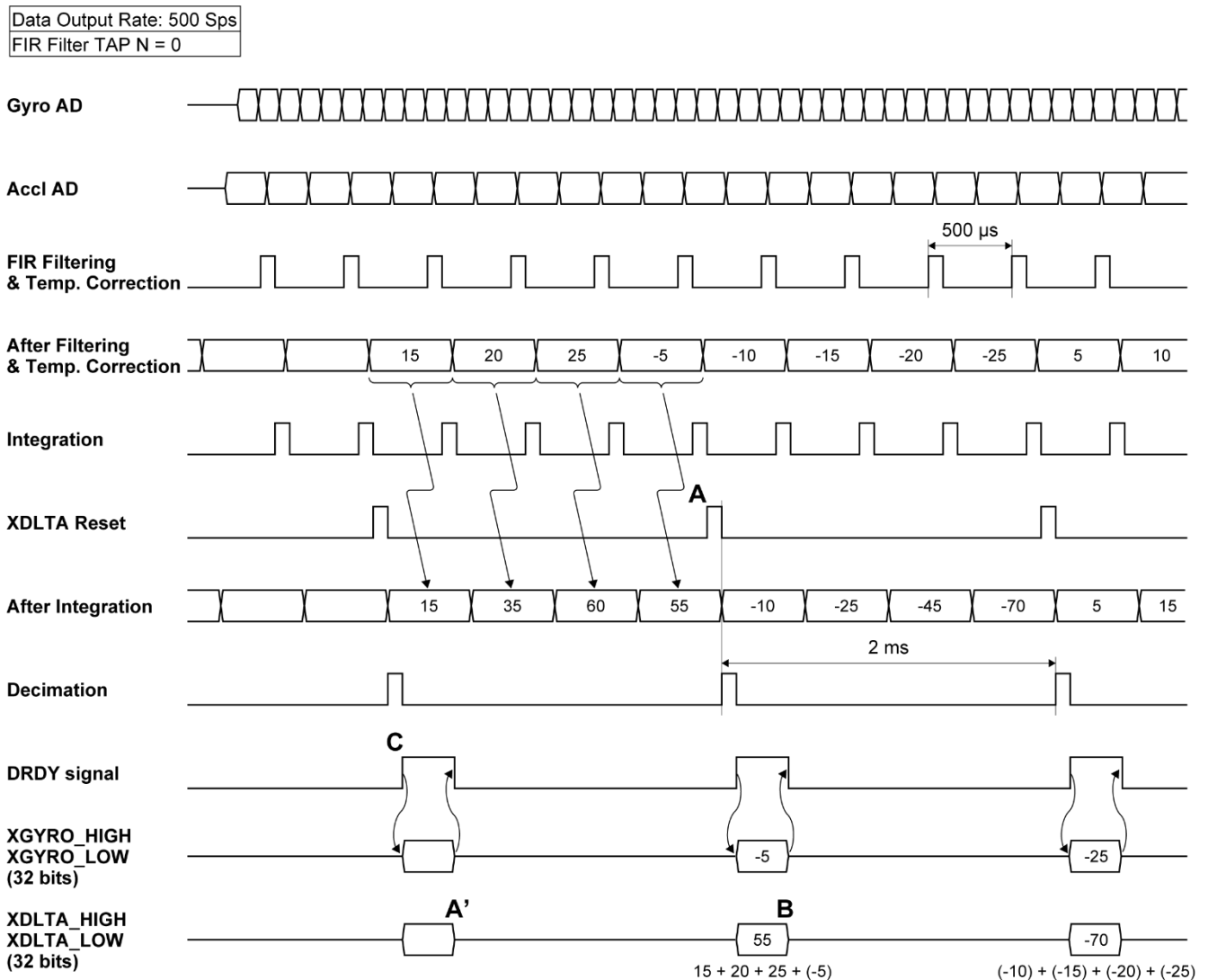


Figure 4.15 Delta Angle / Delta Velocity

As described above, the numerical integration of angular rate and linear acceleration values is reset by reading Delta Angle register and Delta Velocity register by the Host or automatically in UART Auto mode after sending the output data. However, when selecting 16-bit output mode in **DLTA_BIT** (BURST_CTRL2[0x0E (W1)] bit[11]) or **DLTV_BIT** (BURST_CTRL2[0x0E (W1)] bit[10]), the upper 16 bits of the value will be output and lower 16 bits will be added to the next integration cycle.

- NOTE:
- Delta Angle / Delta Velocity output is restricted and not supported for SPI Normal mode and UART Normal mode reads.
 - Delta Angle and Gyro sensor output cannot be used together, and Delta Velocity and Accelerometer sensor cannot be used together.
 - When Delta Angle / Delta Velocity is enabled, the **FILTER_SEL** can be set to TAP = 0 (or other value regardless of **DOUT_RATE** setting)
 - When using Delta Angle / Delta Velocity output with UART_AUTO disabled (UART_CTRL[0x08 (W1)] bit 0 = 0), send BURST_CMD within 500us of DRDY assertion (point C in Figure 4.15).

To enable Delta Angle / Delta Velocity output, program the settings as shown below.

1. Enable burst data output by writing "1" to **DLTA_OUT** or **DLTV_OUT** (BURST_CTRL1[0x0D (W1)] bit[11:10]).
2. Set the required bit length for **DLTA_BIT** or **DLTV_BIT** (BURST_CTRL2[0x0F (W1)] bit[11:10]).
3. Set the required scale factor for Delta Angle and/or Delta Velocity, respectively in **DLTA_RANGE_CTRL** or **DLTV_RANGE_CTRL** (GLOB_CMD3[0x12 (W1)] bit[7:4] & bit[3:0]).

NOTE: By setting an appropriate scale factor, overflow due to accumulation of angular velocity and acceleration values can be avoided. The appropriate scale factor depends on the operating conditions of the application.

4. Write "01" to **ATTI_ON** (ATTI_CTRL[0x15 (W1)] bit[10:9]) to enable the Delta Angle / Delta Velocity output function. *Section 5.3* provides an example of the Delta Angle / Delta Velocity output data fields for burst output in data packet format.

4.15 Attitude Output

This device supports Inclination mode or Euler Angle mode as an attitude output function. This function can be set in **ATTI_MODE** (ATTI_CTRL[0x14 (W1) bit[11]]). The definition of each mode is as follows.

For 200ms after sampling starts, the output of the tilt angle mode and Euler angle mode returns a fixed value of "0" to determine the initial attitude (ANG1_HIGH to ANG3_LOW[0x64 to 0x6E (W0)]). During this period, the static bias of the gyro is calculated and removed, and the attitude output data is returned after 200ms has elapsed.

(1) Inclination Mode

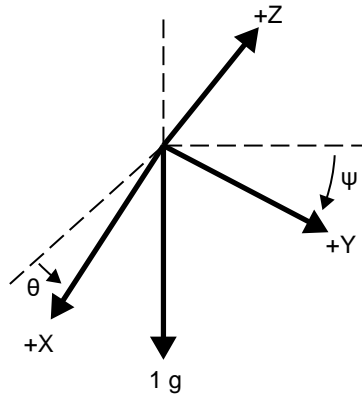


Figure 4.16 Inclination Mode

The inclination angle is the minimum angle that each axis rotates relative to the horizontal plane.

θ : ANG1 (ANG1_HIGH & LOW[0x64–0x67 (W0)]) Attitude angle data 1 (x-axis relative to horizontal*1)

ψ : ANG2 (ANG2_HIGH & LOW[0x68–0x6B (W0)]) Attitude angle data 2 (y-axis relative to horizontal*1)

*1) When **ATTI_CONV** (ATTI_CTRL[0x14 (W1)] bit[4:0]) = "00" (default setting).

The designation of the ANG1 and ANG2 axes are programmable in **ATTI_CONV**.

NOTE: For Inclination mode, ANG3 (ANG3_HIGH & LOW[0x6C–0x6F (W0)] Attitude angle data 3) returns a fixed value of "0"

(2) Euler Angle Mode

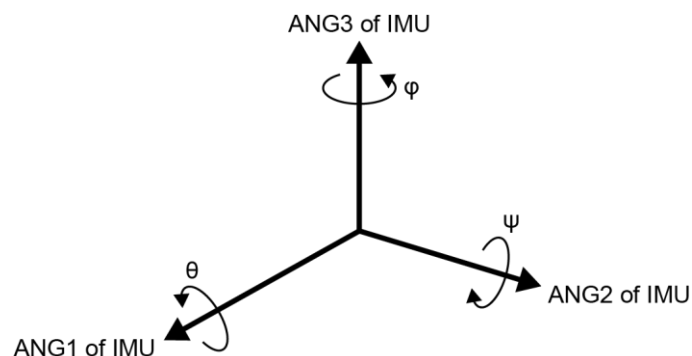


Figure 4.17 Euler Angle Mode

The order of the rotation for Euler Angle mode is ANG3 (Yaw), ANG1 (Roll), and ANG2 (Pitch) in a moving frame (each rotation is on the axes of a rotating coordinate system). The (+) rotation direction follows the "right hand" rule. The designation of the ANG1, ANG2, and ANG3 axes are programmable in **ATTI_CONV** (ATTI_CTRL[0x014 (W1)] bit[4:0]).

θ : ANG1 (ANG1_HIGH & LOW[0x64–0x67 (W0)]) Attitude angle data 1 (roll, x-axis rotation*1)

ψ : ANG2 (ANG2_HIGH & LOW[0x68–0x6B (W0)]) Attitude angle data 2 (pitch, y-axis rotation*1)

ϕ : ANG3 (ANG3_HIGH & LOW[0x6C–0x6F (W0)]) Attitude angle data 3 (yaw, z-axis rotation*1)

*1) When **ATTI_CONV** = "00" (default setting).

To enable Attitude output, program the settings as shown below.

1. Enable burst data output by writing "1" to **ATTI_OUT** (BURST_CTRL1[0x0D (W1)] bit[8]).
2. Set the required bit length in **ATTI_BIT** (BURST_CTRL2[0x0F (W1)] bit[8]).
3. Set the required attitude output mode in **ATTI_MODE** (ATTI_CTRL[0x15 (W1)] bit[11]).
4. Set the required attitude output axis conversion in **ATTI_CONV** (ATTI_CTRL[0x14 (W1)] bit[4:0]).
5. Set **ATTITUDE_MOTION_PROFILE** (GLOB_CMD2[0x16 (W1)] bit[5:4]).
6. Write "10" to **ATTI_ON** (ATTI_CTRL[0x15 (W1)] bit [10:9]) to enable the attitude output function. *Section 5.3* provides an example of the ANG1, ANG2, and ANG3 output data fields for burst output in data packet format.

NOTE: The Attitude output function cannot be used with the external counter reset input and external trigger input (MSC_CTRL[0x02 (W1)] bit[7:6] = "01", "11").

4.16 Quaternion Output

This device supports attitude output represented in quaternion format. The quaternion format is a 32-bit fixed point with the upper 2 bits (signed) of integer part and the lower 30 bits of fractional part.

q₀: QTN0_HIGH & LOW[0x50–0x53 (W0)]

q₁: QTN1_HIGH & LOW[0x54–0x57 (W0)]

q₂: QTN2_HIGH & LOW[0x58–0x5B (W0)]

q₃: QTN3_HIGH & LOW[0x5C–0x5F (W0)]

Each element of the quaternion is expressed as follows using the rotation axis unit vector "u" and the rotation angle "θ".

$$q_0 = \cos \frac{\theta}{2}$$

$$q_1 = u_x \sin \frac{\theta}{2}$$

$$q_2 = u_y \sin \frac{\theta}{2}$$

$$q_3 = u_z \sin \frac{\theta}{2}$$

To enable quaternion output, program the settings as shown below.

1. Enable burst data output by writing "1" to **QTN_OUT** (BURST_CTRL1[0x0D (W1)] bit[9]).
2. Set the required bit length in **QTN_BIT** (BURST_CTRL2[0x0F (W1)] bit[9]).
3. Set **ATTITUDE_MOTION_PROFILE** (GLOB_CMD2[0x16 (W1)] bit[5:4]).
4. Write "10" to **ATTI_ON** (ATTI_CTRL[0x15 (W1)] bit[10:9]) to enable the quaternion output function. *Section 5.3* provides an example of the quaternion output data fields for burst output in data packet format.

NOTE: The Quaternion output function cannot be used with the external counter reset input and external trigger input (MSC_CTRL[0x02 (W1)] bit[7:6] = "01", "11").

NOTE: When Quaternion output is enabled, set ATTI_CTRL[0x14 (W1)] bit[4:0] = "00000", ATTI_CONV: XYZ = FLU.

4.17 Range Over Function

This device supports the notification when a range over condition is detected in the sensor data. The range over thresholds are as follows.

Gyro Sensor: ± 400 [deg/s]

Accelerometer: ± 7.5 [G] for Output Range ± 8 G (**A_RANGE_CTRL** of GLOB_CMD3[0x13 (W1)] bit[8] = "0")
 ± 15 [G] for Output Range ± 16 G (**A_RANGE_CTRL** of GLOB_CMD3[0x13 (W1)] bit[8] = "1")

Detection is performed by "Range Over" block in the processing order as described in Figure 4.4 Functional Block Diagram. The host can confirm that a range over has occurred by reading the **RO** bit of FLAG (ND / EA) in the burst read data or the register FLAG[0x06 (W0)] bit[8]. The source of the range over occurrence can be confirmed by reading RANGE_OVER[0x0C (W0)] bit[13:8] and bit[0]. The **RO** bits (RANGE_OVER[0x0C (W0)] bit[13:8] bit[0]) are reset by reading the register, so that any subsequent range over detection can be resumed during sampling.

Refer to FLAG[0x06 (W0)], RANGE_OVER[0x0C (W0)] for register operation.

4.18 Frame Alignment Correction

The frame alignment of the three-axis gyro triad and accelerometer triad can be corrected by using the R_MATRIX function. The matrix coefficients are 16-bit fixed-point numbers with 2 bits representing the integer part including the sign bit, and 14 bits representing the fractional part. The default coefficient values for the R_MATRIX, if unchanged, will result in an identity matrix.

- R_MATRIX_M**[0x38–0x49 (W1)]: The coefficients for the 3 x 3 rotation matrix for gyro triad.

The coefficient setting register is common to angular velocity and acceleration.

Frame alignment correction of the gyroscope triad is represented by the 3 x 3 matrix multiplication operation:

$$\begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} = \begin{pmatrix} R_MATRIX_M11 & R_MATRIX_M12 & R_MATRIX_M13 \\ R_MATRIX_M21 & R_MATRIX_M22 & R_MATRIX_M23 \\ R_MATRIX_M31 & R_MATRIX_M32 & R_MATRIX_M33 \end{pmatrix} \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix}$$

G: Gyroscope output data after the rotation matrix operation.

Data is output to the XGYRO, YGYRO, and ZGYRO registers ([0x12–0x1D (W0)]).

M: Misalignment 3 x 3 matrix

g: Measurement data (3 axes)

Frame alignment correction of the accelerometer triad is represented by the 3 x 3 matrix multiplication operation:

$$\begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} = \begin{pmatrix} R_MATRIX_M11 & R_MATRIX_M12 & R_MATRIX_M13 \\ R_MATRIX_M21 & R_MATRIX_M22 & R_MATRIX_M23 \\ R_MATRIX_M31 & R_MATRIX_M32 & R_MATRIX_M33 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

A: Accelerometer output data after the rotation matrix operation.

Data is output to the XACCL, YACCL, and ZACCL registers ([0x1E–0x29 (W0)]).

M: Misalignment 3 x 3 matrix

a: Measurement data (3 axes)

NOTE: The initial value for M (misalignment matrix) is the identity matrix. When M is unmodified, G = g, A = a.

Refer to 7.1.10 Frame Alignment Correction (SPI) for an example of setting the rotation matrix coefficients.

NOTE: • When using the attitude output, set the rotation matrix coefficient that maintains the orthogonality between the axes.

- The frame alignment correction has effect on the Delta Angle / Delta Velocity output function. Also it has effect on the Attitude/Quaternion output function.

5. 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 additionally the UART interface supports Auto mode function. Because both interfaces are always active, the user needs only to connect the desired interface pins SPI or UART, without needing any hardware pin configuration or selection.

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 special mode called the Burst mode in addition to the Normal mode.

When the IMU output data rate is high (i.e., 1000 sps), it is possible to exceed the bandwidth of the host interface and causes 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)] bit[15:8]).
- 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_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)].

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

(1) For UART and 32-bit output:

Baudrate: 921600 baud, Data output rate: 2000 sps

- **BAUD_RATE** (UART_CTRL[0x08 (W1)] bit[9:8]) = "10": 921600 baud
- **UART_AUTO** (UART_CTRL[0x08 (W1)] bit[0]) = "1": UART Auto Mode
- **DOUT_RATE** (SMPL_CTRL[0x04 (W1)] bit[15:8]) = 0x00: 2000 sps
- BURST_CTRL1[0x0C (W1)] = 0xF006: FLAG, TEMP, angle rate, acceleration, GPIO, and COUNT outputs
- BURST_CTRL2[0x0E (W1)] = 0x7000: TEMP, angle rate, and acceleration are all output in 32 bits.

Baudrate: 460800 baud, Data output rate: 1000 sps

- **BAUD_RATE** (UART_CTRL[0x08 (W1)] bit[9:8]) = "00": 460800 baud
- **UART_AUTO** (UART_CTRL[0x08 (W1)] bit[0]) = "1": UART Auto Mode
- **DOUT_RATE** (SMPL_CTRL[0x04 (W1)] bit[15:8]) = 0x01: 1000 sps
- BURST_CTRL1[0x0C (W1)] = 0xF006: FLAG, TEMP, angle rate, acceleration, GPIO, and COUNT outputs
- BURST_CTRL2[0x0E (W1)] = 0x7000: TEMP, angle rate, and acceleration are all output in 32 bits.

(2) For SPI and 32-bit output:

- SPI Interface Transmission Setting: $f_{SCLK} = 1 \text{ MHz}$ and $t_{STALL} = 24 \mu\text{s}$ for normal mode
- **DOUT_RATE** (SMPL_CTRL[0x04 (W1)] bit[15:8]) = 0x01: 1000 sps
- BURST_CTRL1[0x0C (W1)] = 0xF006: FLAG, TEMP, angle rate, acceleration, GPIO, and COUNT outputs
- BURST_CTRL2[0x0E (W1)] = 0x7000: TEMP, angle rate, and acceleration are all output in 32 bits.

(3) For UART and 16-bit output:

- **BAUD_RATE** (UART_CTRL[0x08 (W1)] bit[9:8]) = "00": 460800 baud
- **UART_AUTO** (UART_CTRL[0x08 (W1)] bit[0]) = "1": UART Auto Mode
- **DOUT_RATE** (SMPL_CTRL[0x04 (W1)] bit[15:8]) = 0x01: 1000 sps
- BURST_CTRL1[0x0C (W1)] = 0xF006: FLAG, TEMP, angle rate, acceleration, GPIO, and COUNT outputs
- BURST_CTRL2[0x0E (W1)] = 0x0000: TEMP, angle rate, and acceleration are all output in 16 bits.

(4) For SPI and 16-bit output:

- SPI Interface Transmission Setting: $f_{SCLK} = 1 \text{ MHz}$ and $t_{STALL} = 24 \mu\text{s}$ for normal mode
- **DOUT_RATE** (SMPL_CTRL[0x04 (W1)] bit[15:8]) = 0x01: 1000 sps
- BURST_CTRL1[0x0C (W1)] = 0xF006: FLAG, TEMP, angle rate, acceleration, GPIO, and COUNT outputs
- BURST_CTRL2[0x0E (W1)] = 0x0000: All TEMP, angle rate, and acceleration are all output in 16 bits.

5.1 SPI Interface

Table 5.1 shows the communication settings of SPI interface and Table 5.2 shows the SPI timing for Normal mode.

Table 5.1 SPI Communication Settings

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

Table 5.2 SPI Timing (Normal Mode)

Parameter	Min.	Max.	Unit
f_{SCLK}	0.01	2.0	MHz
t_{STALL}	20	–	μs
$t_{WRITERATE}$	40	–	μs
$t_{READRATE}$	40	–	μs

5.1.1 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).

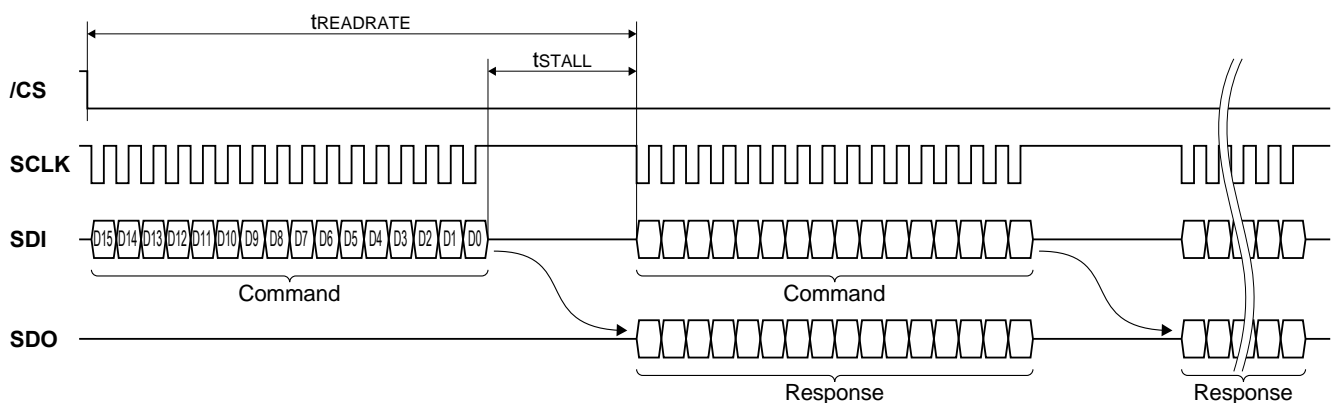


Figure 5.1 SPI Read Timing (Normal Mode)

Table 5.3 Command Format (Read)

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

A[6:0]: Register address (even address)

XX: Don't Care

Table 5.4 Response Format (Read)

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

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

5.1.2 SPI Write Timing (Normal Mode)

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

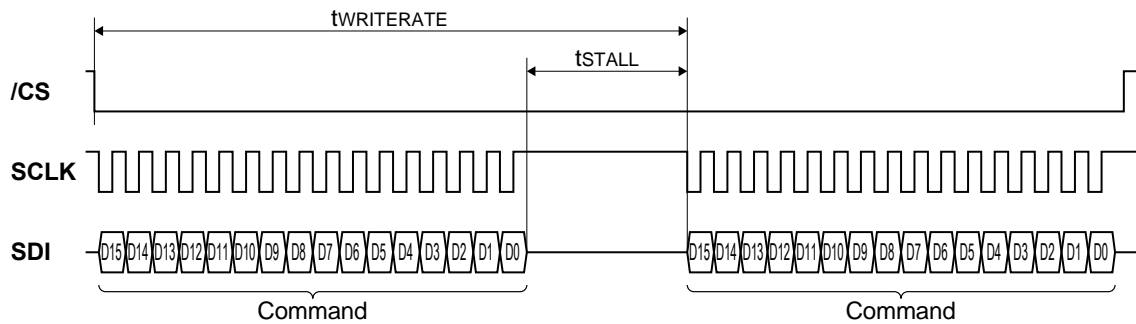


Figure 5.2 SPI Write Timing (Normal Mode)

Table 5.5 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 address)
 D[7:0]: Register write data

5.1.3 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)] bit[7:0]). In Burst mode, ND flag / EA flag, temperature sensor value, 3-axis gyroscope sensor value, 3-axis acceleration sensor value, GPIO, etc. can be consecutively sent as a response. The response format for the burst read output data is configured by register setting in BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)]. Please refer to 5.3 Data Packet Format for the response format.

Table 5.6 SPI Timing (Burst Mode)

Parameter	Min.	Max.	Unit
fSCLK	0.01	1.0	MHz
tSTALL1	45	–	µs
tSTALL2	4	–	µs
tREADRATE2	32	–	µs

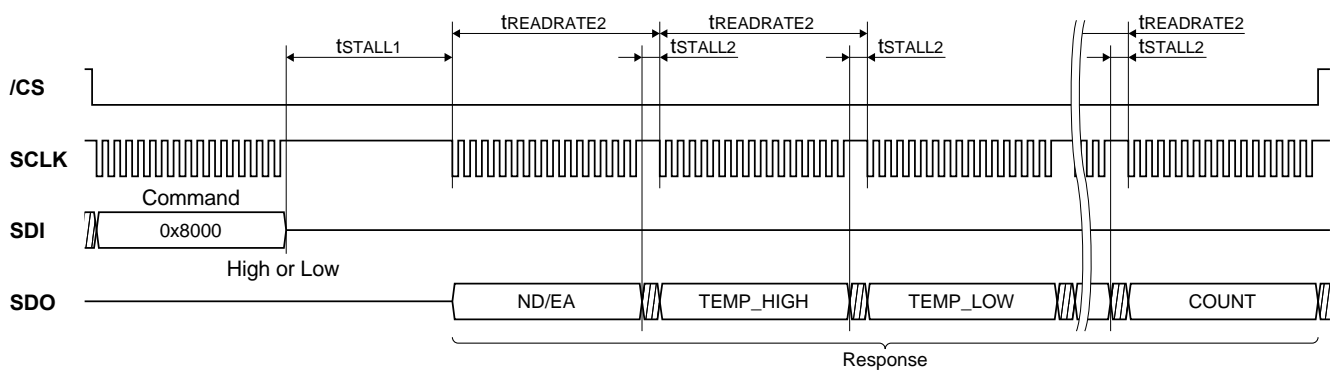


Figure 5.3 SPI Timing (Burst Mode)

NOTE: SPI Burst Read mode can support $t_{STALL2} = 0 \mu s$ under the following conditions.

- BURST_CTRL1[0x0C (W1)]: Set value 0xF003: FLAG, Temperature, Angular Rate, Acceleration, Count, Checksum
BURST_CTRL2[0x0E (W1)]: Set value 0x3000: Angular Rate & Acceleration are output in 32 bits.
- f_{SCLK} : Maximum 500 kHz
- t_{STALL1} : Minimum 45 μs .
- t_{STALL2} : 0 μs
- $t_{READRATE2}$: Minimum 32 μs .
- External Counter Reset or External Trigger Input is not supported
MSC_CTRL[0x02 (W1)] bit[7:6] = "00"
- Attitude Output is not supported
ATTI_ON[0x14 (W1)] bit[10] = "0"

5.2 UART Interface

Table 5.7 shows the supported UART communication settings and Figure 5.4 shows the UART bit format. Please refer to **BAUD_RATE** (UART_CTRL[0x08 (W1)] bit[9:8]) for changing the baud rate setting.

Table 5.7 UART Communication Settings

Parameter	Set value
Transfer rate	230.4k bps / 460.8 kbps / 921.6kbps
Start	1 bit
Data	8 bits
Stop	1 bit
Parity	None
Delimiter	CR (0x0D)

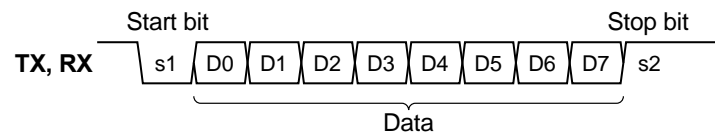


Figure 5.4 UART Bit Format

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 5.8 and Table 5.9 show the timings of UART.

Table 5.8 UART Timing

Parameter	Manual mode				Auto mode		Unit
	Normal mode		Burst mode		Min.	Max.	
	Min.	Max.	Min.	Max.			
t_{STALL} (230.4 kbps)	–	25	–	70	–	–*2	μs
t_{STALL} (460.8 kbps)	–	25	–	70	–	–*2	μs
t_{STALL} (921.6 kbps)	–	25	–	70	–	–*2	μs
$t_{WRITERATE}$ (230.4 kbps)	350	–	–	–	350	–	μs
$t_{WRITERATE}$ (460.8 kbps)	200	–	–	–	200	–	μs
$t_{WRITERATE}$ (921.6 kbps)	150	–	–	–	150	–	μs
$t_{READRATE}$ (230.4 kbps)	350	–	*1	–	–*2	–	μs
$t_{READRATE}$ (460.8 kbps)	200	–	*1	–	–*2	–	μs
$t_{READRATE}$ (921.6 kbps)	150	–	*1	–	–*2	–	μs

*1) Please refer to Table 5.9.

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

Table 5.9 UART Timing (t_{READRATE} Requirements for Burst Mode)

Parameter	Burst Mode (Min.)	Unit
t _{READRATE} (230.4kbps)	300 + (43.4 x B)	μs
t _{READRATE} (460.8kbps)	200 + (21.7 x B)	μs
t _{READRATE} (921.6kbps)	150 + (10.9 x B)	μs

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

Example t_{READRATE} Calculation:

BURST_CTRL1[0x0C (W1)]: Set value 0xF006
 BURST_CTRL2[0x0E (W1)]: Set value 0x7000
 B = 34 bytes for the above stated register setting
 t_{READRATE} (460.8 kbps) = 200 + (21.7 x 34) = 937.8 (μs)

5.2.1 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.

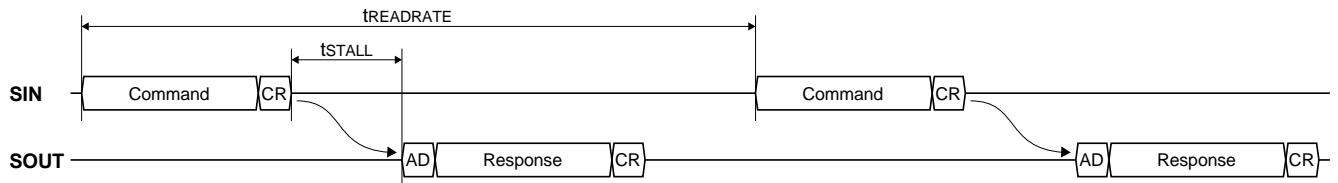


Figure 5.5 UART Read Timing (Normal Mode)

Table 5.10 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 5.11 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

5.2.2 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)] bit[7:0]). In Burst mode, ND flag / EA flag, temperature sensor value, 3-axis gyroscope sensor value, 3-axis acceleration sensor value, GPIO, etc. can be consecutively sent as a response. The response format for the burst read output data is configured by register setting in BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)]. Please refer to 5.3 Data Packet Format for the response format.

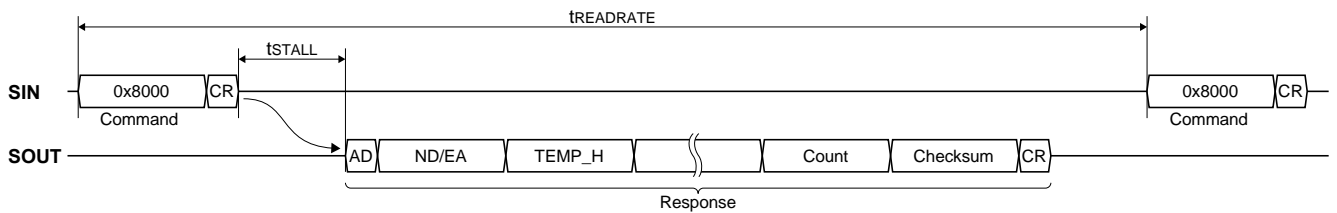


Figure 5.6 UART Read Timing (Burst Mode)

Table 5.12 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

5.2.3 UART Write Timing

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

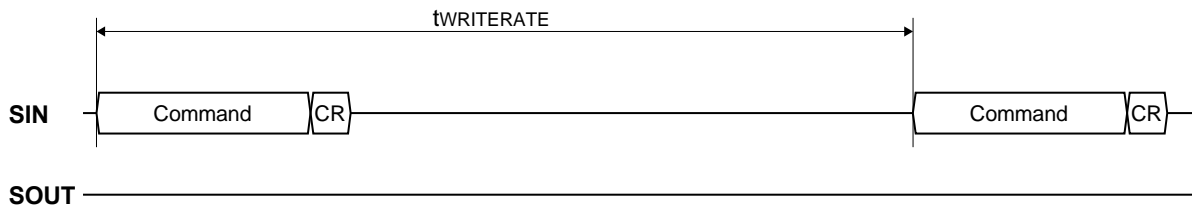


Figure 5.7 UART Write Timing

Table 5.13 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 address or odd address)
 D[7:0]: Register write data
 0x0D: Delimiter

5.2.4 UART Auto Mode Operation

When UART Auto mode 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 5.3 Data Packet Format. The response format for the burst read output data is configured by register setting in BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)].

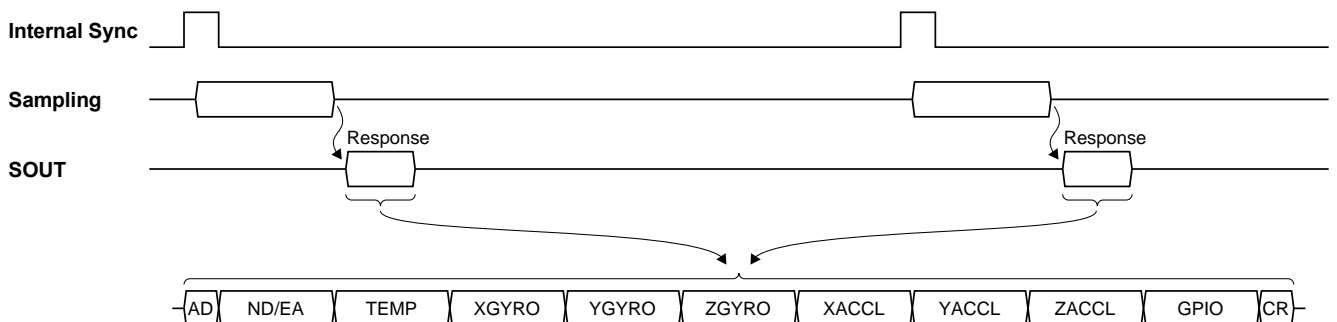


Figure 5.8 UART Auto Mode Operation

5.3 Data Packet Format

The following Table 5.14, Table 5.15, and Table 5.18 show examples of the data packet format sent to the host in UART Burst mode or UART Auto mode. Table 5.16, Table 5.17, and Table 5.19 show the data packet formats sent to the host in SPI Burst mode.

Table 5.14 UART Data Packet Format (UART Burst / Auto Mode) Example: 16-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF007 / BURST_CTRL2[0x0E (W1)] = 0x0000

Byte No.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	ADDRESS	0x80							
2	ND	ND (Temp)	ND (XGyro)	ND (YGyro)	ND (ZGyro)	ND (XACCL)	ND (YACCL)	ND (ZACCL)	RO
3	EA	ND (XDLTA)	ND (YDLTA)	ND (ZDLTA)	ND (XDLTV)	ND (YDLTV)	ND (ZDLTV)	–	EA
4	TEMP_HIGH_H	TEMP_HIGH[15:8]							
5	TEMP_HIGH_L	TEMP_HIGH[7:0]							
6	XGYRO_HIGH_H	XGYRO_HIGH[15:8]							
7	XGYRO_HIGH_L	XGYRO_HIGH[7:0]							
8	YGYRO_HIGH_H	YGYRO_HIGH[15:8]							
9	YGYRO_HIGH_L	YGYRO_HIGH[7:0]							
10	ZGYRO_HIGH_H	ZGYRO_HIGH[15:8]							
11	ZGYRO_HIGH_L	ZGYRO_HIGH[7:0]							
12	XACCL_HIGH_H	XACCL_HIGH[15:8]							
13	XACCL_HIGH_L	XACCL_HIGH[7:0]							
14	YACCL_HIGH_H	YACCL_HIGH[15:8]							
15	YACCL_HIGH_L	YACCL_HIGH[7:0]							
16	ZACCL_HIGH_H	ZACCL_HIGH[15:8]							
17	ZACCL_HIGH_L	ZACCL_HIGH[7:0]							
18	GPIO_H	–	–	–	–	–	–	GPIO _DATA2	GPIO _DATA1
19	GPIO_L	–	–	–	–	–	–	GPIO _DIR2	GPIO _DIR1
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 5.15 UART Data Packet Format (UART Burst / Auto Mode) Example: 32-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF007 / BURST_CTRL2[0x0E (W1)] = 0x7000

Byte No.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	ADDRESS	0x80							
2	ND	ND (Temp)	ND (XGyro)	ND (YGyro)	ND (ZGyro)	ND (XACCL)	ND (YACCL)	ND (ZACCL)	RO
3	EA	ND (XDLTA)	ND (YDLTA)	ND (ZDLTA)	ND (XDLTV)	ND (YDLTV)	ND (ZDLTV)	–	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	XGYRO_HIGH_H	XGYRO_HIGH[15:8]							
9	XGYRO_HIGH_L	XGYRO_HIGH[7:0]							
10	XGYRO_LOW_H	XGYRO_LOW[15:8]							
11	XGYRO_LOW_L	XGYRO_LOW[7:0]							
12	YGYRO_HIGH_H	YGYRO_HIGH[15:8]							
13	YGYRO_HIGH_L	YGYRO_HIGH[7:0]							
14	YGYRO_LOW_H	YGYRO_LOW[15:8]							
15	YGYRO_LOW_L	YGYRO_LOW[7:0]							
16	ZGYRO_HIGH_H	ZGYRO_HIGH[15:8]							
17	ZGYRO_HIGH_L	ZGYRO_HIGH[7:0]							
18	ZGYRO_LOW_H	ZGYRO_LOW[15:8]							
19	ZGYRO_LOW_L	ZGYRO_LOW[7:0]							
20	XACCL_HIGH_H	XACCL_HIGH[15:8]							
21	XACCL_HIGH_L	XACCL_HIGH[7:0]							
22	XACCL_LOW_H	XACCL_LOW[15:8]							
23	XACCL_LOW_L	XACCL_LOW[7:0]							
24	YACCL_HIGH_H	YACCL_HIGH[15:8]							
25	YACCL_HIGH_L	YACCL_HIGH[7:0]							
26	YACCL_LOW_H	YACCL_LOW[15:8]							
27	YACCL_LOW_L	YACCL_LOW[7:0]							
28	ZACCL_HIGH_H	ZACCL_HIGH[15:8]							
29	ZACCL_HIGH_L	ZACCL_HIGH[7:0]							
30	ZACCL_LOW_H	ZACCL_LOW[15:8]							
31	ZACCL_LOW_L	ZACCL_LOW[7:0]							
32	GPIO_H	–	–	–	–	–	–	GPIO _DATA2	GPIO _DATA1
33	GPIO_L	–	–	–	–	–	–	GPIO _DIR2	GPIO _DIR1
34	COUNT_H	COUNT[15:8]							
35	COUNT_L	COUNT[7:0]							
36	CHECKSUM_H	CHECKSUM[15:8]							
37	CHECKSUM_L	CHECKSUM[7:0]							
38	CR	0x0D							

Table 5.16 Data Packet Format (SPI Burst Mode) Example: 16-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF007 / BURST_CTRL2[0x0E (W1)] = 0x0000

Word No.	Bit 15	• • •	Bit 0
1			FLAG (ND / EA)
2			TEMP_HIGH
3			XGYRO_HIGH
4			YGYRO_HIGH
5			ZGYRO_HIGH
6			XACCL_HIGH
7			YACCL_HIGH
8			ZACCL_HIGH
9			GPIO
10			COUNT
11			CHECKSUM

Table 5.17 Data Packet Format (SPI Burst Mode) Example: 32-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF007 / BURST_CTRL2[0x0E (W1)] = 0x7000

Word No.	Bit 15	• • •	Bit 0
1			FLAG (ND / EA)
2			TEMP_HIGH
3			TEMP_LOW
4			XGYRO_HIGH
5			XGYRO_LOW
6			YGYRO_HIGH
7			YGYRO_LOW
8			ZGYRO_HIGH
9			ZGYRO_LOW
10			XACCL_HIGH
11			XACCL_LOW
12			YACCL_HIGH
13			YACCL_LOW
14			ZACCL_HIGH
15			ZACCL_LOW
16			GPIO
17			COUNT
18			CHECKSUM

Table 5.18 UART Data Packet Format (UART Burst / Auto Mode) Example: 32-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF107 / BURST_CTRL2[0x0E (W1)] = 0x7100

Byte No.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	ADDRESS	0x80							
2	ND	ND (Temp)	ND (XGyro)	ND (YGyro)	ND (ZGyro)	ND (XACCL)	ND (YACCL)	ND (ZACCL)	RO
3	EA	ND (XDLTA)	ND (YDLTA)	ND (ZDLTA)	ND (XDLTV)	ND (YDLTV)	ND (ZDLTV)	–	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	XGYRO_HIGH_H	XGYRO_HIGH[15:8]							
9	XGYRO_HIGH_L	XGYRO_HIGH[7:0]							
10	XGYRO_LOW_H	XGYRO_LOW[15:8]							
11	XGYRO_LOW_L	XGYRO_LOW[7:0]							
12	YGYRO_HIGH_H	YGYRO_HIGH[15:8]							
13	YGYRO_HIGH_L	YGYRO_HIGH[7:0]							
14	YGYRO_LOW_H	YGYRO_LOW[15:8]							
15	YGYRO_LOW_L	YGYRO_LOW[7:0]							
16	ZGYRO_HIGH_H	ZGYRO_HIGH[15:8]							
17	ZGYRO_HIGH_L	ZGYRO_HIGH[7:0]							
18	ZGYRO_LOW_H	ZGYRO_LOW[15:8]							
19	ZGYRO_LOW_L	ZGYRO_LOW[7:0]							
20	XACCL_HIGH_H	XACCL_HIGH[15:8]							
21	XACCL_HIGH_L	XACCL_HIGH[7:0]							
22	XACCL_LOW_H	XACCL_LOW[15:8]							
23	XACCL_LOW_L	XACCL_LOW[7:0]							
24	YACCL_HIGH_H	YACCL_HIGH[15:8]							
25	YACCL_HIGH_L	YACCL_HIGH[7:0]							
26	YACCL_LOW_H	YACCL_LOW[15:8]							
27	YACCL_LOW_L	YACCL_LOW[7:0]							
28	ZACCL_HIGH_H	ZACCL_HIGH[15:8]							
29	ZACCL_HIGH_L	ZACCL_HIGH[7:0]							
30	ZACCL_LOW_H	ZACCL_LOW[15:8]							
31	ZACCL_LOW_L	ZACCL_LOW[7:0]							
32	ANG1_HIGH_H	ANG1_HIGH[15:8]							
33	ANG1_HIGH_L	ANG1_HIGH[7:0]							
34	ANG1_LOW_H	ANG1_LOW[15:8]							
35	ANG1_LOW_L	ANG1_LOW[7:0]							

Byte No.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
36	ANG2_HIGH_H	ANG2_HIGH[15:8]							
37	ANG2_HIGH_L	ANG2_HIGH[7:0]							
38	ANG2_LOW_H	ANG2_LOW[15:8]							
39	ANG2_LOW_L	ANG2_LOW[7:0]							
40	ANG3_HIGH_H	ANG3_HIGH[15:8]							
41	ANG3_HIGH_L	ANG3_HIGH[7:0]							
42	ANG3_LOW_H	ANG3_LOW[15:8]							
43	ANG3_LOW_L	ANG3_LOW[7:0]							
44	GPIO_H	-	-	-	-	-	-	GPIO _DATA2	GPIO _DATA1
45	GPIO_L	-	-	-	-	-	-	GPIO _DIR2	GPIO _DIR1
46	COUNT_H	COUNT[15:8]							
47	COUNT_L	COUNT[7:0]							
48	CHECKSUM_H	CHECKSUM[15:8]							
49	CHECKSUM_L	CHECKSUM[7:0]							
50	CR	0x0D							

Table 5.19 Data Packet Format (SPI Burst Mode) Example: 32-bit Output

BURST_CTRL1[0x0C (W1)] = 0xF307 / BURST_CTRL2[0x0E (W1)] = 0x7300

Word No.	Bit 15	• • •	Bit 0
1			FLAG (ND / EA)
2			TEMP_HIGH
3			TEMP_LOW
4			XGYRO_HIGH
5			XGYRO_LOW
6			YGYRO_HIGH
7			YGYRO_LOW
8			ZGYRO_HIGH
9			ZGYRO_LOW
10			XACCL_HIGH
11			XACCL_LOW
12			YACCL_HIGH
13			YACCL_LOW
14			ZACCL_HIGH
15			ZACCL_LOW
16			QTN0_HIGH
17			QTN0_LOW
18			QTN1_HIGH
19			QTN1_LOW
20			QTN2_HIGH
21			QTN2_LOW
22			QTN3_HIGH
23			QTN3_LOW
24			ANG1_HIGH
25			ANG1_LOW
26			ANG2_HIGH
27			ANG2_LOW
28			ANG3_HIGH
29			ANG3_LOW
30			GPIO
31			COUNT
32			CHECKSUM

6. User Registers

A host device (for example, a microcontroller) can control the IMU 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 1.4 Interface Specifications, all the register values are undefined because internal initialization is in progress. Ensure the IMU registers are only accessed after the Power-On Start-Up 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 6.1. 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 IMU is in the Configuration mode before writing to registers. In the Sampling Mode, writing to registers is ignored except for the following cases.

- Writing to **MODE_CMD** (MODE_CTRL[0x02 (W0)] bit[9:8])
- Writing to **GPIO_DATA** (GPIO[0x08 (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])

While in the UART Auto mode and Sampling mode is active, register read access is not supported. Otherwise, the sampling data transmitted in the UART Auto mode 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 6.1 shows the register map, and Section 6.1 through Section 6.32 describes the registers in detail.

The “-” sign in the register assignment table in Section 6.1 through Section 6.32 means “reserved”.

Write “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, for example, 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 6.1 Register Map

Name	Window ID	Address	R/W	Flash Backup	Default	Function
BURST	0	0x00	W		-	Burst mode
MODE_CTRL	0	0x03, 0x02	R/W		0x0400	Operation mode control
DIAG_STAT	0	0x04	R		0x0000	Diagnostic result
FLAG	0	0x06	R		0x0000	ND flag / EA flag
GPIO	0	0x09, 0x08	R/W		0x0200	GPIO
COUNT	0	0x0A	R		0x0000	Sampling count value
RANGE_OVER	0	0x0C	R		0x0000	Range Over
TEMP_HIGH	0	0x0E	R		0xFFFF	Temperature sensor value High
TEMP_LOW	0	0x10	R		0xFFFF	Temperature sensor value Low
XGYRO_HIGH	0	0x12	R		0xFFFF	X gyroscope sensor value High
XGYRO_LOW	0	0x14	R		0xFFFF	X gyroscope sensor value Low
YGYRO_HIGH	0	0x16	R		0xFFFF	Y gyroscope sensor value High
YGYRO_LOW	0	0x18	R		0xFFFF	Y gyroscope sensor value Low
ZGYRO_HIGH	0	0x1A	R		0xFFFF	Z gyroscope sensor value High
ZGYRO_LOW	0	0x1C	R		0xFFFF	Z gyroscope sensor value Low
XACCL_HIGH	0	0x1E	R		0xFFFF	X acceleration sensor value High
XACCL_LOW	0	0x20	R		0xFFFF	X acceleration sensor value Low
YACCL_HIGH	0	0x22	R		0xFFFF	Y acceleration sensor value High
YACCL_LOW	0	0x24	R		0xFFFF	Y acceleration sensor value Low
ZACCL_HIGH	0	0x26	R		0xFFFF	Z acceleration sensor value High
ZACCL_LOW	0	0x28	R		0xFFFF	Z acceleration sensor value Low
ID	0	0x4C	R		0x5345	ID read function
QTN0_HIGH	0	0x50	R		0x0000	Quaternion q0 High
QTN0_LOW	0	0x52	R		0x0000	Quaternion q0 Low
QTN1_HIGH	0	0x54	R		0x0000	Quaternion q1 High
QTN1_LOW	0	0x56	R		0x0000	Quaternion q1 Low
QTN2_HIGH	0	0x58	R		0x0000	Quaternion q2 High
QTN2_LOW	0	0x5A	R		0x0000	Quaternion q2 Low
QTN3_HIGH	0	0x5C	R		0x0000	Quaternion q3 High
QTN3_LOW	0	0x5E	R		0x0000	Quaternion q3 Low
XDLTA_HIGH / ANG1_HIGH *1	0	0x64	R		0x0000	X delta angle value High / ANG1 attitude output High
XDLTA_LOW / ANG1_LOW *1	0	0x66	R		0x0000	X delta angle value Low / ANG1 attitude output Low
YDLTA_HIGH / ANG2_HIGH *1	0	0x68	R		0x0000	Y delta angle value High / ANG2 attitude output High
YDLTA_LOW / ANG2_LOW *1	0	0x6A	R		0x0000	Y delta angle value Low / ANG2 attitude output Low
ZDLTA_HIGH / ANG3_HIGH *1	0	0x6C	R		0x0000	Z delta angle value High / ANG3 attitude output High
ZDLTA_LOW / ANG3_LOW *1	0	0x6E	R		0x0000	Z delta angle value Low / ANG3 attitude output Low
XDLTV_HIGH	0	0x70	R		0x0000	X delta velocity value High
XDLTV_LOW	0	0x72	R		0x0000	X delta velocity value Low
YDLTV_HIGH	0	0x74	R		0x0000	Y delta velocity value High
YDLTV_LOW	0	0x76	R		0x0000	Y delta velocity value Low
ZDLTV_HIGH	0	0x78	R		0x0000	Z delta velocity value High
ZDLTV_LOW	0	0x7A	R		0x0000	Z delta velocity value Low
SIG_CTRL	1	0x01, 0x00	R/W	○	0xFE00	Data Ready signal & polarity control
MSC_CTRL	1	0x03, 0x02	R/W	○	0x0006	Other control
SMPL_CTRL	1	0x05, 0x04	R/W	○	0x0103	Sampling control
FILTER_CTRL	1	0x07, 0x06	R/W	○	0x0001	Filter control
UART_CTRL	1	0x09, 0x08	R/W	○	0x0000	UART control

Name	Window ID	Address	R/W	Flash Backup	Default	Function
GLOB_CMD	1	0x0B, 0x0A	R/W	○	0x0000	System control
BURST_CTRL1	1	0x0D, 0x0C	R/W	○	0xF006	Burst control 1
BURST_CTRL2	1	0x0F, 0x0E	R/W	○	0x0000	Burst control 2
POL_CTRL	1	0x11, 0x10	R/W	○	0x0000	Polarity control
GLOB_CMD3	1	0x13, 0x12	R/W	○	0x00CC	Acceleration range / Delta control
ATTI_CTRL	1	0x15, 0x14	R/W	○	0x0000	Attitude control
GLOB_CMD2	1	0x17, 0x16	R/W		0x0000	System control2
R_MATRIX_M11 *2	1	0x39, 0x38	R/W	◎	0x4000	Gyro R_Matrix coefficient
R_MATRIX_M12 *2	1	0x3B, 0x3A	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M13 *2	1	0x3D, 0x3C	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M21 *2	1	0x3F, 0x3E	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M22 *2	1	0x41, 0x40	R/W	◎	0x4000	Gyro R_Matrix coefficient
R_MATRIX_M23 *2	1	0x43, 0x42	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M31 *2	1	0x45, 0x44	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M32 *2	1	0x47, 0x46	R/W	◎	0x0000	Gyro R_Matrix coefficient
R_MATRIX_M33 *2	1	0x49, 0x48	R/W	◎	0x4000	Gyro R_Matrix coefficient
PROD_ID1	1	0x6A	R		*4	Product ID
PROD_ID2	1	0x6C	R		*4	Product ID
PROD_ID3	1	0x6E	R		*4	Product ID
PROD_ID4	1	0x70	R		*4	Product ID
VERSION *3	1	0x72	R		*4	Version
SERIAL_NUM1	1	0x74	R		*4	Serial number
SERIAL_NUM2	1	0x76	R		*4	Serial number
SERIAL_NUM3	1	0x78	R		*4	Serial number
SERIAL_NUM4	1	0x7A	R		*4	Serial number
WIN_CTRL	0,1	0x7F, 0x7E	R/W		0x0000	Register window control

*1 In ATTI_CTRL register, **ATTI_ON** setting determines if Delta Angle or Attitude is output.

*2 The R_MATRIX_*_ coefficient values are stored in non-volatile memory using GLOB_CMD2[0x17 (W1)] bit[8].

*3 Version is subject to change without notice.

*4 Refer to the description of each register for the default value.

6.1 BURST Register (Window 0)

Addr (Hex)	Bit 15	...	Bit 0	R/W
0x01		-		-
Addr (Hex)	Bit 7	...	Bit 0	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 5.1.3 SPI Read Timing (Burst Mode) and 5.2.2 UART Read Timing (Burst Mode). Also refer to 5.3 Data Packet Format. The output data can be selected by setting BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)].

6.2 MODE_CTRL Register (Window 0)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x03	-					MODE_STAT	MODE_CMD		R/W *1
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x02	-								-

*1) Only **MODE_STAT** is read-only.

bit[10] MODE_STAT

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

- 1: Configuration mode
- 0: Sampling mode

bit[9:8] MODE_CMD

Executes commands related to the operation mode.

- 01: Go to Sampling mode. After the mode transition is completed, the bits automatically go back to "00".
- 10: Go to Configuration mode. After the mode transition is completed, the bits automatically go back to "00".
- 11: (Not used)
- 00: (Not used)

6.3 DIAG_STAT Register (Window 0)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x05	-	ST_ERR (XGyro)	ST_ERR (YGyro)	ST_ERR (ZGyro)	ST_ERR (ACCL)	SET_ERR	DLTA_OVF	DLTV_OVF	R
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x04	-	HARD_ERR		SPI_OVF	UART_OVF	FLASH_ERR	ST_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[14:11] ST_ERR (SelfTest ERRor)

Shows the result of **SELF_TEST** (internal self test) of MSC_CTRL[0x02 (W1)] bit[10].

- 1: Error occurred
- 0: No error

bit[10] SET_ERR (SET ERRor)

Shows that a SET Error condition has occurred.

- 1: Error occurred
- 0: No error

SET Error condition occurs when Delta Angle / Delta Velocity is disabled and an invalid setting is detected with any of the following:

- Invalid combination of output rate setting (SMPL_CTRL[0x05 (W1)] bit[11:8]) and filter setting (FILTER_CTRL[0x06 (W1)] bit[4:0])
- Attitude output is enabled (ATTI_ON[0x15 (W1)] bit[10:9] = "10"), and external trigger input is enabled (MSC_CTRL[0x02 (W1)] bit[7:6] = "11").

bit[9] DLTA_OVF (DeLTa Angle OVer Flow)

Shows an overflow error condition of the Delta Angle.

- 1: Error occurred
0: No error

bit[8] DLTV_OVF (DeLTa V OVer Flow)

Shows an overflow error condition of the Delta Velocity.

- 1: Error occurred
0: No error

bit[6:5] HARD_ERR (HARD ERRor)

Shows the result of the hardware check at startup.

- Other than 00: Error occurred
00: No error

When this error occurs, it indicates the IMU 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 a 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 baud rate, data output rate, UART Burst / Auto mode in combination. Refer to **BAUD_RATE** of UART_CTRL[0x08 (W1)] bit[9:8], **DOUT_RATE** of SMPL_CTRL[0x04 (W1)] bit[15:8]. If using Burst mode with UART Auto mode, also review BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)] settings.

bit[2] FLASH_ERR (FLASH ERRor)

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] ST_ERR_ALL (SelfTest ERRor All)

Shows the logical sum of bit[14:11] 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] or **FLASH_ROTATION_BACKUP** of GLOB_CMD2[0x017 (W1)] bit[8].

- 1: Error occurred
0: No error

6.4 FLAG (ND / EA) Register (Window 0)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x07	ND (Temp)	ND (XGyro)	ND (YGyro)	ND (ZGyro)	ND (XACCL)	ND (YACCL)	ND (ZACCL)	RO	R
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x06	ND (XDLTA)	ND (YDLTA)	ND (ZDLTA)	ND (XDLTV)	ND (YDLTV)	ND (ZDLTV)	-	EA	R

bit[15:9] ND (New Data) flag (Temperature, Gyroscope, Acceleration)

When a new measuring data is set in each register of temperature (TEMP_HIGH), gyroscope (XGYRO_HIGH, YGYRO_HIGH, ZGYRO_HIGH), and acceleration (XACCL_HIGH, YACCL_HIGH, ZACCL_HIGH), the corresponding ND flag is set to "1". When the measurement output is read from the corresponding register, the flag is reset to "0".

bit[8] RO (Range Over) flag

When at least one over range condition is detected in RANGE_OVER[0x0C (W0)], this flag is set to "1".

bit[7:2] ND (New Data) flag (Delta Angle, Delta Velocity)

When a new measuring data is set in each register of Delta Angle (XDLTA_HIGH, YDLTA_HIGH, ZDLTA_HIGH), and Delta Velocity (XDLTV_HIGH, YDLTV_HIGH, ZDLTV_HIGH), the corresponding ND flag is set to "1". When the measurement output is read from the corresponding register, the flag is reset to "0".

bit[0] EA (All Error) flag

When at least one failure is found in the diagnostic result (DIAG_STAT[0x04 (W0)]), the flag is set to "1" (failure occurred).

6.5 GPIO Register (Window 0)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x09	-						GPIO_DATA2	GPIO_DATA1	R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x08	-						GPIO_DIR2	GPIO_DIR1	R/W

bit[9:8] GPIO_DATA

If the corresponding GPIO_DIR bit is set to "output", the value set in the GPIO_DATA is output to the GPIO port. If the corresponding GPIO_DIR bit is set to "input", the input level of the GPIO port is returned by reading the GPIO_DATA.

- 1: High Level
- 0: Low Level

bit[1:0] GPIO_DIR

Each bit controls the bitwise direction of the GPIO port.

- 1: Output
- 0: Input

- NOTE:
- GPIO1 is shared with the Data Ready signal function on the same pin. The selection between GPIO1 and the Data Ready signal is controlled with DRDY_ON of MSC_CTRL[0x02 (W1)] bit[2]. The pin functions as GPIO1 when DRDY_ON is "0" (disabled).
 - GPIO2 is shared with the EXT signal input function (External Trigger Input and External Counter Reset Input) on the same pin. The selection between GPIO2 and the EXT signal input is controlled with EXT_SEL of MSC_CTRL[0x02 (W1)] bit[7:6]. The pin functions as GPIO2 when EXT_SEL is "00" (GPIO2).

6.6 COUNT Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x0A	COUNT			R

bit[15:0] COUNT

The value returned by this register depends on whether the External Counter Reset Input function is enabled or not. The External Counter Reset Input is enabled when **EXT_SEL** of MSC_CTRL[0x02 (W1)] bit[7:6] = "01".

When the External Counter Reset Input function is disabled, this register returns the sampling count value at the sampling timing.

NOTE: The time unit of the sampling counter value represents 500 μs/count.

Example: If the data output rate equals 1000 Sps, the counter value sequence is 0, 2, 4, 6, ... , 0xFFFE, 0, 2, ...

When the External Counter Reset Input function is enabled, this register returns the timer counter value used by the External Counter Reset Input function.

6.7 RANGE_OVER Register (Window 0)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x0D	-		RO (XGyro)	RO (YGyro)	RO (ZGyro)	RO (XACCL)	RO (YACCL)	RO (ZACCL)	R
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0C	-							RO (Attitude)	R

bit[13:8] RO (Range Over) Flag (Gyroscope / Acceleration)

The specified gyroscope or acceleration sensor axis RO flag is set to "1" when the output value exceeds the sensing range. The flags are reset to "0" after reading this register.

bit[0] RO (Range Over) Flag (Attitude)

The attitude RO flag is set to "1" when the output value exceeds the sensing range. The flag is reset to "0" after reading this register.

Regardless of the acceleration Output Range setting value (GLOB_CMD3[0x13 (W1)] bit[8]), the range over threshold of the acceleration input for attitude output is ±8 G.

6.8 TEMP Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	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. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits (**TEMP_HIGH**).

Please refer to the below formula for conversion to temperature in centigrade. Please refer to Table 1.3 Sensor Specifications for the scale factor value.

For 32-bit usage: $T [^{\circ}\text{C}] = (\text{SF} / 65536) \times A + 25$

For 16-bit usage: $T [^{\circ}\text{C}] = \text{SF} \times A + 25$

SF: Scale Factor (16bit)

A: Temperature sensor output data (decimal)

NOTE: 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.

6.9 GYRO Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x12		XGYRO_HIGH		R
0x14		XGYRO_LOW		R
0x16		YGYRO_HIGH		R
0x18		YGYRO_LOW		R
0x1A		ZGYRO_HIGH		R
0x1C		ZGYRO_LOW		R

bit[15:0] Gyroscope output data

Returns the 3-axis gyroscope data for X, Y, and Z as referenced in Figure 2.1 M-G330PDG0 Outline Dimensions.

The output data format is 32-bit two's complement. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits.

Please refer to Table 1.3 Sensor Specifications for the scale factor value.

For 32-bit usage: $G \text{ [deg/s]} = ((1 / SF) / 65536) \times B$

For 16-bit usage: $G \text{ [deg/s]} = (1 / SF) \times B$

SF: Scale Factor (16bit)

B: Gyroscope output data (decimal)

6.10 ACCL Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x1E		XACCL_HIGH		R
0x20		XACCL_LOW		R
0x22		YACCL_HIGH		R
0x24		YACCL_LOW		R
0x26		ZACCL_HIGH		R
0x28		ZACCL_LOW		R

bit[15:0] Acceleration sensor output data

Returns the 3-axis acceleration data for X, Y, and Z as referenced in Figure 2.1 M-G330PDG0 Outline Dimensions.

The output data format is 32-bit two's complement. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits.

Please refer to Table 1.3 Sensor Specifications for the scale factor value.

For 32-bit usage: $A \text{ [mG]} = ((1 / SF) / 65536) \times C$

For 16-bit usage: $A \text{ [mG]} = (1 / SF) \times C$

SF: Scale Factor (16bit)

C: Acceleration sensor output data (decimal)

6.11 ID Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x4C		ID		R

bit[15:0] ID data

This register will return the value "0x5345" when read.

6.12 QUATERNION Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x50		QTN0_HIGH		R
0x52		QTN0_LOW		R
0x54		QTN1_HIGH		R
0x56		QTN1_LOW		R
0x58		QTN2_HIGH		R
0x5A		QTN2_LOW		R
0x5C		QTN3_HIGH		R
0x5E		QTN3_LOW		R

bit[15:0] Quaternion

Returns the quaternion output format representing the attitude in a 32-bit fixed point format.

The upper 2 bits (signed) are the integer part and the remaining 30 lower bits as the fractional part. For 16-bit usage, only the upper 16-bits are used. Please refer to 4.16 *Quaternion Output* for a detailed description.

6.13 DELTA_ANGLE Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x64		XDLTA_HIGH		R
0x66		XDLTA_LOW		R
0x68		YDLTA_HIGH		R
0x6A		YDLTA_LOW		R
0x6C		ZDLTA_HIGH		R
0x6E		ZDLTA_LOW		R

bit[15:0] Delta Angle (Gyroscope) output data

Returns the Delta Angle output format from the 3-axis gyroscope data for X, Y, and Z as referenced in Figure 2.1 M-G330PDG0 Outline Dimensions.

The output data format is 32-bit two's complement. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits.

Please refer to **DLTA_RANGE_CTRL** of GLOB_CMD3[0x12 (W1)] bit[7:4] for the appropriate delta angle scale factor value.

For 32-bit usage: $\Delta\text{Angle} [\text{deg}] = (\text{SF} / 65536) \times \text{D}$

For 16-bit usage: $\Delta\text{Angle} [\text{deg}] = \text{SF} \times \text{D}$

SF: Scale Factor (Refer to **DLTA_RANGE_CTRL**)

D: Gyroscope delta angle output data (decimal)

6.14 DELTA_VELOCITY Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x70		XDLTV_HIGH		R
0x72		XDLTV_LOW		R
0x74		YDLTV_HIGH		R
0x76		YDLTV_LOW		R
0x78		ZDLTV_HIGH		R
0x7A		ZDLTV_LOW		R

bit[15:0] Delta Velocity (Acceleration) output data

Returns the Delta Velocity output format from the 3-axis acceleration data for X, Y, and Z as referenced in Figure 2.1 M-G330PDG0 Outline Dimensions.

The output data format is 32-bit two's complement. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits.

Please refer to **DLTV_RANGE_CTRL** of GLOB_CMD3[0x12 (W1)] bit[3:0] for the appropriate delta velocity scale factor value.

For 32-bit usage: $\Delta\text{Velocity [m/s]} = (\text{SF} / 65536) \times E$

For 16-bit usage: $\Delta\text{Velocity [m/s]} = \text{SF} \times E$

SF: Scale Factor (Refer to **DLTV_RANGE_CTRL**)

E: Acceleration delta velocity output data (decimal)

6.15 ATTI Register (Window 0)

Addr (Hex)	bit 15	...	bit 0	R/W
0x64		ANG1_HIGH		R
0x66		ANG1_LOW		R
0x68		ANG2_HIGH		R
0x6A		ANG2_LOW		R
0x6C		ANG3_HIGH *1		R
0x6E		ANG3_LOW *1		R

bit[15:0] Attitude output data

Returns the attitude output data for ANG1, and ANG2 in Inclination mode, or returns the attitude output data for ANG1, ANG2, and ANG3 in Euler Angle mode.

The output data format is 32-bit two's complement. For 16-bit usage, treat the data as 16-bit two's complement using the upper 16 bits.

Please refer to Table 1.3 Sensor Specifications for the scale factor value.

For 32-bit usage: $\text{ATTI} = (\text{SF} / 65536) \times F$

For 16-bit usage: $\text{ATTI} = \text{SF} \times F$

SF: Scale Factor

F: Attitude output data (decimal)

*1) The ANG3 output, ANG3_HIGH & LOW[0x6C–0x6F (W0)], is only valid in Euler Angle mode and is fixed to 0 when in Inclination mode and should be ignored.

6.16 SIG_CTRL Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x01	ND_EN (Temp)	ND_EN (XGyro)	ND_EN (YGyro)	ND_EN (ZGyro)	ND_EN (XACCL)	ND_EN (YACCL)	ND_EN (ZACCL)	-	R/W

Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x00	ND_EN (XDLTA)	ND_EN (YDLTA)	ND_EN (ZDLTA)	ND_EN (XDLTV)	ND_EN (YDLTV)	ND_EN (ZDLTV)		-	R/W

bit[15:9] ND_EN (Temperature, Gyroscope, Acceleration)

Enables or disables the ND flags in FLAG[0x06 (W0)] bit[15:9].

1: Enable

0: Disable

bit[7:2] ND_EN (Delta Angle, Delta Velocity)

Enables or disables the ND flags in FLAG[0x06 (W0)] bit[7:2].

1: Enable

0: Disable

NOTE: **ND_EN** settings are ignored and have no effect when UART Auto mode is active.

6.17 MSC_CTRL Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W	
0x03	-				FLASH_TEST	SELF_TEST	-			R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W	
0x02	EXT_SEL		-			DRDY_ON	DRDY_POL	-	R/W	

NOTE: The FLASH_TEST, and SELF_TEST functions can not 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[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] SELF_TEST

Write “1” to execute the self test to check if the gyroscope and the accelerometer are 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 **ST_ERR_ALL** of DIAG_STAT[0x04 (W0)] bit[1] to check the results.

NOTE: When executing the self-test, be sure to disable the external trigger function.

MSC_CTRL[0x02 (W1)] bit[7:6] = “00” or “01”

bit[7:6] EXT_SEL

These bits select the function of GPIO2 pin to be GPIO2, External Counter Reset Input, or External Trigger Input.

00: GPIO2

01: External Counter Reset Input

10: Invalid

11: External Trigger Input

NOTE: External Trigger Input is not supported when Attitude Output is enabled (ATTI_ON[0x15 (W1)] bit[10:9] = “10”)

bit[2] DRDY_ON

Selects the function of the GPIO1 pin for either GPIO1 or Data Ready.

1: Data Ready Signal

0: GPIO1

bit[1] DRDY_POL

Selects the polarity of the Data Ready signal when selected in **DRDY_ON** above.

1: Active High

0: Active Low

6.18 SMPL_CTRL Register (Window 1)

Addr (Hex)	bit 15	...	bit 8	R/W
0x05	DOUT_RATE			R/W
Addr (Hex)	bit 7	...	bit 0	R/W
0x04	-			-

bit[15:8] DOUT_RATE

Specifies the data output rate.

The following lists the data output rate option with the recommended number of filter taps when using the moving average filter.

0x00: 2000 Sps	TAP ≥ 0
0x01: 1000 Sps	TAP ≥ 2
0x02: 500 Sps	TAP ≥ 4
0x03: 250 Sps	TAP ≥ 8
0x04: 125 Sps	TAP ≥ 16
0x05: 62.5 Sps	TAP ≥ 32
0x06: 31.25 Sps	TAP ≥ 64
0x07: 15.625 Sps	TAP = 128
0x08: 400 Sps	TAP ≥ 8
0x09: 200 Sps	TAP ≥ 16
0x0A: 100 Sps	TAP ≥ 32
0x0B: 80 Sps	TAP ≥ 32
0x0C: 50 Sps	TAP ≥ 64
0x0D: 40 Sps	TAP ≥ 64
0x0E: 25 Sps	TAP = 128
0x0F: 20 Sps	TAP = 128

- NOTE:
- When attitude output is enabled (ATTI_CTRL[0x14 (W1)] bit[10:9] = “10”), the data output rate must be set to 500 sps or less.
 - When Delta Angle / Delta Velocity is enabled, there is no restriction (SET_ERR event is disabled) on the combination in output rate setting and filter setting.
 - The SET_ERR bit in DIAG_STAT[0x05 (W0)] bit[10] will indicate an error if an invalid combination (denoted by “-”) in output rate setting (SMPL_CTRL[0x05 (W1)] bit[11:8]) and filter setting (FILTER_CTRL[0x06 (W1)] bit[4:0]) is selected as outlined in Table 6.2.
During a detected SET_ERR condition, the output registers TEMP_HIGH–TEMP_LOW / X_GYRO_HIGH–Z_GYRO_LOW / X_ACCL_HIGH–Z_ACCL_LOW will output a fixed value 0x7EF0.

Table 6.2 SET_ERR Flag Output Rate and Filter Setting Table Evaluation Table

- When the Attitude Output is disabled (ATTI_ON (ATTI_CTRL[0x14 (W1)] bit[10:9]) = “00” or “01”)

DOUT_RATE[3:0] (SMPL_CTRL[0x05 (W1)] bit[11:8])		FILTER_SEL (Filter_CTRL[0x06 (W1)] bit[4:0])																			
Setting	Sps	0	2	4	8	16	32	64	128	32 fc50	32 fc100	32 fc200	32 fc400	64 fc50	64 fc100	64 fc200	64 fc400	128 fc50	128 fc100	128 fc200	128 fc400
0x00	2000	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
0x01	1000	-	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
0x02	500	-	-	OK	OK	OK	OK	OK	OK	OK	OK	OK	-	OK	OK	OK	-	OK	OK	OK	-
0x08	400	-	-	-	OK	OK	OK	OK	OK	OK	OK	OK	-	OK	OK	OK	-	OK	OK	OK	-
0x03	250	-	-	-	OK	OK	OK	OK	OK	OK	OK	-	-	OK	OK	-	-	OK	OK	-	-
0x09	200	-	-	-	-	OK	OK	OK	OK	OK	OK	-	-	OK	OK	-	-	OK	OK	-	-
0x04	125	-	-	-	-	OK	OK	OK	OK	OK	-	-	-	OK	-	-	-	OK	-	-	-
0x0A	100	-	-	-	-	-	OK	OK	OK	OK	-	-	-	OK	-	-	-	OK	-	-	-
0x0B	80	-	-	-	-	-	OK	OK	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x05	62.5	-	-	-	-	-	OK	OK	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0C	50	-	-	-	-	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0D	40	-	-	-	-	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x06	31.25	-	-	-	-	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0E	25	-	-	-	-	-	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0F	20	-	-	-	-	-	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x07	15.625	-	-	-	-	-	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-

OK = Supported, - = Invalid

• When the Attitude Output is enabled (ATTI_ON (ATTI_CTRL[0x14 (W1)] bit[10:9]) = "10")

DOUT_RATE[3:0] (SMPL_CTRL[0x05 (W1)] bit[11:8])		FILTER_SEL (Filter_CTRL[0x06 (W1)] bit[4:0])																				
Setting	Sps	0	2	4	8	16	32	64	128	32 fc50	32 fc100	32 fc200	32 fc400	64 fc50	64 fc100	64 fc200	64 fc400	128 fc50	128 fc100	128 fc200	128 fc400	
0x00	2000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x01	1000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x02	500	-	-	-	-	OK	OK	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-
0x08	400	-	-	-	-	OK	OK	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-
0x03	250	-	-	-	-	OK	OK	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-
0x09	200	-	-	-	-	OK	OK	-	-	OK	OK	-	-	-	-	-	-	-	-	-	-	-
0x04	125	-	-	-	-	OK	OK	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0A	100	-	-	-	-	-	OK	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-
0x0B	80	-	-	-	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x05	62.5	-	-	-	-	-	OK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x0C	50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x0D	40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x06	31.25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x0E	25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x0F	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x07	15.625	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

OK = Supported, -- = Invalid

NOTE: There are limitations on the supported output features depending on the output rate as shown in the following table.

Table 6.3 Output Rate Limitations

Output Rate (Sps)	Gyro [0x12-0x1D (W0)] Accl [0x1C-0x29 (W0)]	Attitude [0x64-0x6F (W0)]	Quaternion [0x50-0x5F (W0)]	Rotation Matrix [0x38-0x5B (W0)]	Delta [0x64-0x7B (W0)]	Range Over [0x0C, 0x0D (W0)]
2000	OK	-	-	OK	-	OK
1000	OK	-	-	OK	OK	OK
500	OK	OK (Update 200 Sps)	OK (Update 200 Sps)	OK	OK	OK
400	OK	OK (Update 200 Sps)	OK (Update 200 Sps)	OK	OK	OK
250	OK	OK (Update 200 Sps)	OK (Update 200 Sps)	OK	OK	OK
200	OK	OK	OK	OK	OK	OK
125	OK	OK	OK	OK	OK	OK
100	OK	OK	OK	OK	OK	OK
80	OK	OK	OK	OK	OK	OK
62.5	OK	OK	OK	OK	OK	OK
50	OK	OK	OK	OK	OK	OK
40	OK	OK	OK	OK	OK	OK
31.25	OK	OK	OK	OK	OK	OK
25	OK	OK	OK	OK	OK	OK
20	OK	OK	OK	OK	OK	OK
15.625	OK	OK	OK	OK	OK	OK

OK = Supported, -- = Invalid

6.19 FILTER_CTRL Register (Window 1)

Addr (Hex)	bit 15	...						bit 8	R/W
0x07	-								-
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	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 completion status of the filter selection. After setting the **FILTER_SEL** in bit[4:0], this status bit will be set to "1". After completion of the filter setting operation, this bit will return to "0".

- 1: Filter setting is busy.
- 0: Filter setting is completed.

bit[4:0] FILTER_SEL

Specifies the type of filter (moving average filter and FIR Kaiser filter) and TAP setting. For the FIR Kaiser filter, these bits also select the cutoff frequency f_c in Hz.

00000: Moving average filter TAP = 0
 00001: Moving average filter TAP = 2
 00010: Moving average filter TAP = 4
 00011: Moving average filter TAP = 8
 00100: Moving average filter TAP = 16
 00101: Moving average filter TAP = 32
 00110: Moving average filter TAP = 64
 00111: Moving average filter TAP = 128
 01000: FIR Kaiser filter (parameter = 8) TAP = 32 and $f_c = 50$
 01001: FIR Kaiser filter (parameter = 8) TAP = 32 and $f_c = 100$
 01010: FIR Kaiser filter (parameter = 8) TAP = 32 and $f_c = 200$
 01011: FIR Kaiser filter (parameter = 8) TAP = 32 and $f_c = 400$
 01100: FIR Kaiser filter (parameter = 8) TAP = 64 and $f_c = 50$
 01101: FIR Kaiser filter (parameter = 8) TAP = 64 and $f_c = 100$
 01110: FIR Kaiser filter (parameter = 8) TAP = 64 and $f_c = 200$
 01111: FIR Kaiser filter (parameter = 8) TAP = 64 and $f_c = 400$
 10000: FIR Kaiser filter (parameter = 8) TAP = 128 and $f_c = 50$
 10001: FIR Kaiser filter (parameter = 8) TAP = 128 and $f_c = 100$
 10010: FIR Kaiser filter (parameter = 8) TAP = 128 and $f_c = 200$
 10011: FIR Kaiser filter (parameter = 8) TAP = 128 and $f_c = 400$
 10100-11111: Unused

After setting the filter with these bits, the completion of the operation requires the Filter Setting Time specified in Table 1.4 to elapse or confirming completion by checking **FILTER_STAT** (bit[5]).

NOTE: • Refer to 4.13 *Filter* for description of filter transient response from sampling start.

- When Attitude Output is enabled (ATTI_CTRL[0x14 (W1)] bit[10:9] = "10"), the filter must be set to the supported setting in Table 6.2.
- When acceleration range ± 8 G (GLOB_CMD3[0x13 (W1)] bit[8] **A_RANGE_CTRL** = "0"), TAP = 2 or more is recommended for the moving average filter

6.20 UART_CTRL Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x09	-						BAUD_RATE		R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x08	-						AUTO_START	UART_AUTO	R/W

bit[9:8] BAUD_RATE

These bits specify the Baud Rate of UART interface.

- 00: 460.8 kbps
- 01: 230.4 kbps
- 10: 921.6 kbps

NOTE: The baud rate change using these **BAUD_RATE** bits becomes effective immediately after write access completes.

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

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 when IMU is powered on or reset.

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

bit[0] UART_AUTO

Enables or disables the UART Auto mode function.

- 1: UART Auto mode is selected.
- 0: UART Manual mode is selected.

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

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

NOTE: • For more information on UART Auto mode, refer to *5.2.4 UART Auto Mode Operation* and *5.3 Data Packet Format*. The burst output data is configured by register setting in BURST_CTRL1[0x0C (W1)] and BURST_CTRL2[0x0E (W1)].

- This register bit must be set to "0" when using the SPI interface.

6.21 GLOB_CMD Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x0B	-						NOT_READY	-	R
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0A	SOFT_RST	-		INITIAL_BACKUP	FLASH_BACKUP	-		R/W	

bit[10] NOT_READY

Indicates whether the IMU is currently ready. Immediately after power on, this bit is "1" and becomes "0" when the IMU 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[4] INITIAL_BACKUP**
Write "1" to set the non-volatile memory for the registers to the factory default value. After the execution is completed, the bit automatically goes back to "0". After confirming this bit goes back to "0", check the result in **FLASH_BU_ERR** of **DIAG_STAT[0x04 (W0)] bit[0]**. The values saved in the non-volatile memory are reflected in the "Flash Backup" target registers after power on or a software reset.
- 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 6.1 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", check the result in **FLASH_BU_ERR** of **DIAG_STAT[0x04 (W0)] bit[0]**.

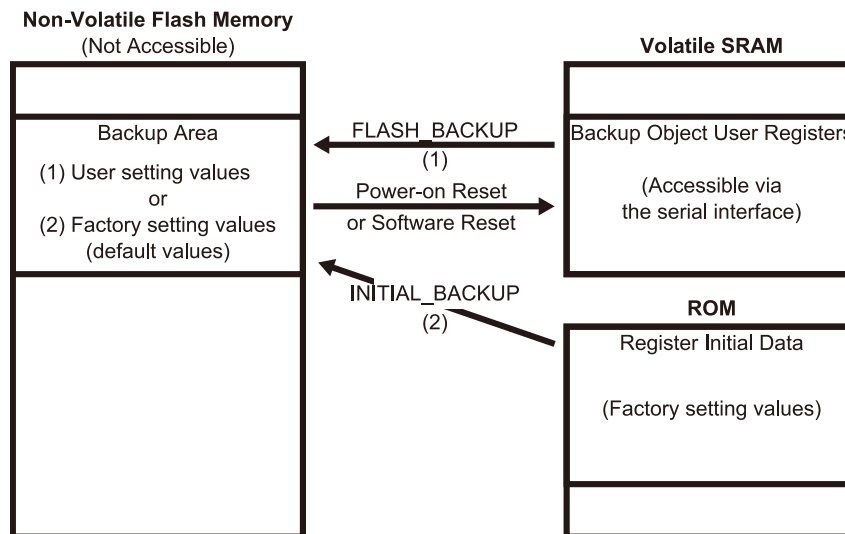


Figure 6.1 FLASH_BACKUP and INITIAL BACKUP

6.22 BURST_CTRL1 Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x0D	FLAG_OUT	TEMP_OUT	GYRO_OUT	ACCL_OUT	DLTA_OUT	DLTV_OUT	QTN_OUT	ATTI_OUT	R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0C			-			GPIO_OUT	COUNT_OUT	CHKSM_OUT	R/W

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

- bit[15] FLAG_OUT**
Controls the output of FLAG status.
1: Enables output.
0: Disables output.
- bit[14] TEMP_OUT**
Controls the output of temperature sensor.
1: Enables output.
0: Disables output.
- bit[13] GYRO_OUT**
Controls the output of gyroscope sensor.
1: Enables output.
0: Disables output.

- bit[12] ACCL_OUT**
Controls the output of acceleration sensor.
1: Enables output.
0: Disables output.
- bit[11] DLTA_OUT**
Controls the output of delta angle.
1: Enables output.
0: Disables output.
- bit[10] DLTV_OUT**
Controls the output of delta velocity.
1: Enables output.
0: Disables output.
- bit[9] QTN_OUT**
Controls the output of quaternion.
1: Enables output.
0: Disables output.
- bit[8] ATTI_OUT**
Controls the output of attitude.
1: Enables output.
0: Disables output.
- bit[2] GPIO_OUT**
Controls the output of GPIO status.
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.

6.23 BURST_CTRL2 Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x0F	-	TEMP_BIT	GYRO_BIT	ACCL_BIT	DLTA_BIT	DLTV_BIT	QTN_BIT	ATTI_BIT	R/W
Addr (Hex)	bit 7	...						bit 0	R/W
0x0E		-							-

These bits select the output bit length of output data for Burst mode and UART Auto mode.

- bit[14] TEMP_BIT**
Selects the bit length of the temperature output.
1: 32 bits
0: 16 bits

- bit[13] GYRO_BIT**
Selects the bit length of the gyroscope output.
1: 32 bits
0: 16 bits
- bit[12] ACCL_BIT**
Selects the bit length of the acceleration output.
1: 32 bits
0: 16 bits
- bit[11] DLTA_BIT**
Selects the bit length of the delta angle output.
1: 32 bits
0: 16 bits
- bit[10] DLTV_BIT**
Selects the bit length of the delta velocity output.
1: 32 bits
0: 16 bits
- bit[9] QTN_BIT**
Selects the bit length of quaternion output.
1: 32 bits
0: 16 bits
- bit[8] ATTI_BIT**
Selects the bit length of the attitude output.
1: 32 bits
0: 16 bits

6.24 POL_CTRL Register (Window 1)

Addr (Hex)	bit 15	...						bit 8	R/W
0x11	-								-
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x10	-	POL_CTRL (XGyro)	POL_CTRL (YGyro)	POL_CTRL (ZGyro)	POL_CTRL (XACCL)	POL_CTRL (YACCL)	POL_CTRL (ZACCL)	-	-

- bit[6:1] POL_CTRL**
Specifies whether to bitwise invert the output value of the following registers: angular rate (XGYRO, YGYRO, ZGYRO) and acceleration (XACCL, YACCL, ZACCL). This bitwise inversion will also have effect in the internal processing for the delta angle (XDLTA, YDLTA, ZDLTA), and delta velocity (XDLTV, YDLTV, ZDLTV).
1: Inverted
0: Not inverted

When using the 4.15 Attitude Output / 4.16 Quaternion function, please set all to "0".

6.25 GLOB_CMD3 Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x13	-							A_RANGE_CTRL	R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x12	DLTA_RANGE_CTRL				DLTV_RANGE_CTRL				R/W

bit[8] A_RANGE_CTRL
This bit sets the output range of the accelerometer.

1: ± 16 G
0: ± 8 G

bit[7:4] DLTA_RANGE_CTRL
These bits specify the scale factor and range of Delta Angle output.

NOTE: The user must carefully select the desired scale factor to optimize Delta Angle resolution but avoid an overflow condition in the internal Delta Angle processing.

bit[3:0] DLTV_RANGE_CTRL
These bits specify the scale factor and range of Delta Velocity output.

NOTE: The user must carefully select the desired scale factor to optimize Delta Velocity resolution but avoid an overflow condition in the internal Delta Velocity processing.

Table 6.4 Delta Angle: Scale Factor & Range

DLTA_RANGE_CTRL (bit[7:4])	Scale Factor [deg/LSB]	Range [\pm deg]
0000	7.576.E-06	2.482E-01
0001	1.515.E-05	4.965E-01
0010	3.030.E-05	9.930E-01
0011	6.061.E-05	1.986E+00
0100	1.212.E-04	3.972E+00
0101	2.424.E-04	7.944E+00
0110	4.848.E-04	1.589E+01
0111	9.697.E-04	3.178E+01
1000	1.939.E-03	6.355E+01
1001	3.879.E-03	1.271E+02
1010	7.758.E-03	2.542E+02
1011	1.552.E-02	5.084E+02
1100	3.103.E-02	1.017E+03
1101	6.206.E-02	2.034E+03
1110	1.241.E-01	4.067E+03
1111	2.482.E-01	8.134E+03

Table 6.5 Delta Velocity: Scale Factor & Range

OutputRange ± 8 G, A_RANGE_CTRL of GLOB_CMD3[0x13 (W1)] bit[8] = "0"			OutputRange ± 16 G, A_RANGE_CTRL of GLOB_CMD3[0x13 (W1)] bit[8] = "1"		
DLTV_RANGE_CTRL (bit[3:0])	Scale Factor [(m/s)/LSB]	Range [\pm (m/s)]	DLTV_RANGE_CTRL (bit[3:0])	Scale Factor [(m/s)/LSB]	Range [\pm (m/s)]
0000	1.226.E-06	4.017E-02	0000	2.452.E-06	8.034E-02
0001	2.452.E-06	8.034E-02	0001	4.903.E-06	1.607E-01
0010	4.903.E-06	1.607E-01	0010	9.807.E-06	3.213E-01
0011	9.807.E-06	3.213E-01	0011	1.961.E-05	6.427E-01
0100	1.961.E-05	6.427E-01	0100	3.923.E-05	1.285E+00
0101	3.923.E-05	1.285E+00	0101	7.845.E-05	2.571E+00
0110	7.845.E-05	2.571E+00	0110	1.569.E-04	5.142E+00
0111	1.569.E-04	5.142E+00	0111	3.138.E-04	1.028E+01
1000	3.138.E-04	1.028E+01	1000	6.276.E-04	2.057E+01
1001	6.276.E-04	2.057E+01	1001	1.255.E-03	4.113E+01
1010	1.255.E-03	4.113E+01	1010	2.511.E-03	8.226E+01
1011	2.511.E-03	8.226E+01	1011	5.021.E-03	1.645E+02
1100	5.021.E-03	1.645E+02	1100	1.004.E-02	3.291E+02
1101	1.004.E-02	3.291E+02	1101	2.008.E-02	6.581E+02
1110	2.008.E-02	6.581E+02	1110	4.017.E-02	1.316E+03
1111	4.017.E-02	1.316E+03	1111	8.034.E-02	2.632E+03

6.26 ATTI_CTRL Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x15	-				ATTI_MODE	ATTI_ON		-	R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x14	-			ATTI_CONV					R/W

bit[11] ATTI_MODE

This bit selects the output mode when attitude output is enabled.

1: Euler angle mode

0: Inclination mode

NOTE: **ATTI_MODE** does not affect the Quaternion output (refer to *Section 4.16*).

bit[10:9] ATTI_ON

These bits enable or disable the Attitude / Quaternion output or Delta Angle / Delta Velocity output.

00: Disable

01: Delta Angle / Delta Velocity Output

10: Attitude Output or Quaternion Output

11: Invalid

NOTE: **ATTI_ON** = "01", "10" has limitations when used with external counter reset input or external trigger input as described in Table 6.6.

Table 6.6 ATTI_ON = "01", "10" Usage Restriction

External Input Function	ATTI_ON = "00" Disable	ATTI_ON = "01" Delta Angle / Delta Velocity	ATTI_ON = "10" Attitude / Quaternion
External Counter Reset Input Reset Input EXT_SEL (MSC_CTRL[0x02 (W1)] bit[7:6]) = "01"	OK	OK	-
External Trigger Input EXT_SEL (MSC_CTRL[0x02 (W1)] bit[7:6]) = "11"	OK	OK	-

bit[4:0] ATTI_CONV

These bits select the attitude output axis transform reference conversion for Inclination or Euler Angle mode.

Table 6.7 Attitude Output Axis Conversion

Register	Attitude ^{*1}			Euler Output ^{*2}			Inclination Output ^{*3}		Note	
	Name	Forward	Left	Up	ANG1 (Roll)	ANG2 (Pitch)	ANG3 (Yaw)	ANG1		ANG2
0x00	a	X	Y	Z	X	Y	Z	X	Y	FLU ^{*4 *5}
0x01	b	X	Z	-Y	X	Z	-Y	X	Z	
0x02	c	X	-Y	-Z	X	-Y	-Z	X	-Y	
0x03	d	X	-Z	Y	X	-Z	Y	X	-Z	
0x04	e	Y	Z	X	Y	Z	X	Y	Z	
0x05	f	Y	X	-Z	Y	X	-Z	Y	X	
0x06	g	Y	-Z	-X	Y	-Z	-X	Y	-Z	
0x07	h	Y	-X	Z	Y	-X	Z	Y	-X	
0x08	i	Z	X	Y	Z	X	Y	Z	X	
0x09	j	Z	Y	-X	Z	Y	-X	Z	Y	
0x0A	k	Z	-X	-Y	Z	-X	-Y	Z	-X	
0x0B	l	Z	-Y	X	Z	-Y	X	Z	-Y	
0x0C	m	-X	Y	-Z	-X	Y	-Z	-X	Y	
0x0D	n	-X	-Z	-Y	-X	-Z	-Y	-X	-Z	
0x0E	o	-X	-Y	Z	-X	-Y	Z	-X	-Y	
0x0F	p	-X	Z	Y	-X	Z	Y	-X	Z	
0x10	q	-Y	Z	-X	-Y	Z	-X	-Y	Z	
0x11	r	-Y	-X	-Z	-Y	-X	-Z	-Y	-X	
0x12	s	-Y	-Z	X	-Y	-Z	X	-Y	-Z	
0x13	t	-Y	X	Z	-Y	X	Z	-Y	X	
0x14	u	-Z	X	-Y	-Z	X	-Y	-Z	X	
0x15	v	-Z	-Y	-X	-Z	-Y	-X	-Z	-Y	
0x16	w	-Z	-X	Y	-Z	-X	Y	-Z	-X	
0x17	X	-Z	Y	X	-Z	Y	X	-Z	Y	

*1 This is the direction that is indicated by the part marking on the device package.

*2 The Euler angle rotation order is ANG3 (Yaw) → ANG1 (Roll) → ANG2 (Pitch) in a moving frame (each rotation is on the axes of a rotating coordinate system). The (+) rotation direction follows the “right hand” rule.

*3 The inclination angle is referenced to the horizontal plane.

*4 The standard attitude reference and axis order for this device is Forward/Left/Up following the “right-hand rule”.

*5 When Quaternion output is enabled, set ATTI_CTRL[0x14 (W1)] bit[4:0] = “00000”, **ATTI_CONV**: XYZ = FLU.

6.27 GLOB_CMD2 Register (Window 1)

Addr (Hex)	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
0x17	-						INITIAL_ROTATION_BACKUP	FLASH_ROTATION_BACKUP	R/W
Addr (Hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x16	-	ATTITUDE_MOTION_PROFILE_STAT	ATTITUDE_MOTION_PROFILE	-				R/W *1	

*1) Only ATTITUDE_MOTION_PROFILE_STAT is read-only.

bit[9] INITIAL_ROTATION_BACKUP

Write "1" to set the non-volatile memory for the R_MATRIX registers with ☉ in the "Flash Backup" column in Table 6.1 to the factory default value. This bit automatically returns to "0" after execution is completed. The factory default value is reflected in the registers after power on or after a hardware or software reset.

bit[8] FLASH_ROTATION_BACKUP

Write "1" to save the current values of the R_MATRIX registers with the ☉ mark in the "Flash Backup" column of Table 6.1 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", check the result in FLASH_BU_ERR of DIAG_STAT[0x04 (W0)] bit[0].

bit[6] ATTITUDE_MOTION_PROFILE_STAT

This bit returns the completion status when setting the ATTITUDE_MOTION_PROFILE.
This bit is read only.

1: Busy
0: Completed

bit[5:4] ATTITUDE_MOTION_PROFILE

The setting of these bits can change the motion profile of the attitude / quaternion output function. The example settings are shown below. It is strongly recommended to evaluate all motion profiles to determine optimal performance.

00: modeA
01: modeB
10: modeC
11: Invalid

Table 6.8 Attitude Motion Profile Description

Attitude Motion Profile	Assumed Operating Speed	Application Example
00 = modeA	3 m/s	General (no specific application)
01 = modeB	20 m/s	Vehicle
10 = modeC	1 m/s	Construction machinery

When writing to these bits, the ATTITUDE_MOTION_PROFILE_STAT changes to "1" (busy). Confirm the completion of the setting process by checking that the ATTITUDE_MOTION_PROFILE_STAT bit returns to "0".

6.28 R_MATRIX Register (Window 1)

Addr (Hex)	bit 15	...	bit 0	R/W
0x38		R_MATRIX_M11		R/W
0x3A		R_MATRIX_M12		R/W
0x3C		R_MATRIX_M13		R/W
0x3E		R_MATRIX_M21		R/W
0x40		R_MATRIX_M22		R/W
0x42		R_MATRIX_M23		R/W
0x44		R_MATRIX_M31		R/W
0x46		R_MATRIX_M32		R/W
0x48		R_MATRIX_M33		R/W

bit[15:0] Rotation Matrix Coefficient

The frame alignment of gyroscope triad and acceleration sensor triad can be corrected by using the R_MATRIX function. Refer to 4.18 *Frame Alignment Correction* for more details.

6.29 PROD_ID Register (Window 1)

Addr (Hex)	bit 15	...	bit 0	R/W
0x6A		PROD_ID1		R
0x6C		PROD_ID2		R
0x6E		PROD_ID3		R
0x70		PROD_ID4		R

bit[15:0] Product ID

These registers return the product model number represented in ASCII code.

Product ID return value depends on the model.

PROD_ID	M-G330PDG0
PROD_ID1	0x3347
PROD_ID2	0x3033
PROD_ID3	0x4450
PROD_ID4	0x3047

6.30 VERSION Register (Window 1)

Addr (Hex)	bit 15	...	bit 0	R/W
0x72		VERSION		R

bit[15:0] Version

This register returns the Firmware Version.

6.31 SERIAL_NUM Register (Window 1)

Addr (Hex)	bit 15	...	bit 0	R/W
0x74	SERIAL_NUM1			R
0x76	SERIAL_NUM2			R
0x78	SERIAL_NUM3			R
0x7A	SERIAL_NUM4			R

bit[15:0] Serial Number

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

6.32 WIN_CTRL Register (Window 0,1)

Addr (Hex)	bit 15	...	bit 8	R/W
0x7F	-			-

Addr (Hex)	bit 7	...	bit 0	R/W
0x7E	WINDOW_ID			R/W

bit[7:0] WINDOW_ID

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

0x00: Window 0

0x01: Window 1

0x02–0xFF: Unused

7. Sample Program Sequence

The following describes the recommended procedures for operating this device.

7.1 SPI Sequence

7.1.1 Power-on Sequence (SPI)

The following shows a power-on sequence:

- (1) Power-on.
- (2) Wait 800 ms.
- (3) Wait until the NOT_READY bit goes to 0. NOT_READY is GLOB_CMD[0x0A (W1)]'s bit[10].


```
TXdata = {0xFE01} / RXdata = {0x----}          /* WINDOW = 1 */
TXdata = {0x0A00} / RXdata = {0x----}          /* GLOB_CMD read command */
TXdata = {0x0000} / RXdata = {GLOB_CMD}        /* get response */
```

 Confirm the NOT_READY bit.
 When NOT_READY becomes 0, this step ends. Otherwise, please repeat (3).
- (4) Confirm HARD_ERR bits. HARD_ERR is DIAG_STAT[0x04 (W0)]'s bit[6:5].


```
TXdata = {0xFE00} / RXdata = {0x----}          /* WINDOW_ID write command. (WINDOW = 0) */
TXdata = {0x0400} / RXdata = {0x----}          /* DIAG_STAT read command */
TXdata = {0x0000} / RXdata = {DIAG_STAT}       /* get response */
```

 Confirm if HARD_ERR is 00.
 If HARD_ERR is 00, the IMU is operating normally. Otherwise, the IMU is faulty.

-: don't care

7.1.2 Register Read and Write (SPI)

[Read Example]

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

```
TXdata = {0xFE00} / RXdata = {0x----}          /* WINDOW = 0 */
TXdata = {0x0200} / RXdata = {0x----}          /* read command */
TXdata = {0x----} / RXdata = {0x0400}         /* get response*/
```

-: don't care

0x04 in the upper byte of RXdata indicates the device is in Configuration mode.

0x00 in the lower byte of RXdata is Reserved.

Please note that read data unit is 16 bits, and Most Significant Bit first in 16-bit SPI.

[Write Example]

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

```
TXdata = {0xFE00} / RXdata = {0x----}          /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----}          /* write command */
```

There is no response at Write.

-: don't care

By sending this command, the IMU enters Sampling mode.

Please note that write data unit is 8 bits.

7.1.3 Sampling Data (SPI)

[Sample Flow 1 (SPI Normal mode)]

- (1) Power-on sequence. Please refer to *Section 7.1.1*.
- (2) Filter setting sequence. Please refer to *Section 7.1.8*.
- (3) Configure Sampling mode.


```
TXdata = {0xFE01} / RXdata = {0x----}          /* WINDOW = 1 */
```

```

TXdata = {0x8504} / RXdata = {0x----} /* 125 Sps */
TXdata = {0x8800} / RXdata = {0x----} /* disable UART Auto mode, just in case. */
TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----} /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```

TXdata = {0x0600} / RXdata = {0x----} /* FLAG read command */
TXdata = {0x0E00} / RXdata = {FLAG} /* TEMP_HIGH read command */
TXdata = {0x1000} / RXdata = {TEMP_HIGH} /* TEMP_LOW read command */
TXdata = {0x1200} / RXdata = {TEMP_LOW} /* XGYRO_HIGH read command */
TXdata = {0x1400} / RXdata = {XGYRO_HIGH} /* XGYRO_LOW read command */
TXdata = {0x1600} / RXdata = {XGYRO_LOW} /* YGYRO_HIGH read command */
TXdata = {0x1800} / RXdata = {YGYRO_HIGH} /* YGYRO_LOW read command */
TXdata = {0x1A00} / RXdata = {YGYRO_LOW} /* ZGYRO_HIGH read command */
TXdata = {0x1C00} / RXdata = {ZGYRO_HIGH} /* ZGYRO_LOW read command */
TXdata = {0x1E00} / RXdata = {ZGYRO_LOW} /* XACCL_HIGH read command */
TXdata = {0x2000} / RXdata = {XACCL_HIGH} /* XACCL_LOW read command */
TXdata = {0x2200} / RXdata = {XACCL_LOW} /* YACCL_HIGH read command */
TXdata = {0x2400} / RXdata = {YACCL_HIGH} /* YACCL_LOW read command */
TXdata = {0x2600} / RXdata = {YACCL_LOW} /* ZACCL_HIGH read command */
TXdata = {0x2800} / RXdata = {ZACCL_HIGH} /* ZACCL_LOW read command */
TXdata = {0x0800} / RXdata = {ZACCL_LOW} /* GPIO read command */
TXdata = {0x0A00} / RXdata = {GPIO} /* COUNT read command */
TXdata = {0x----} / RXdata = {COUNT}.

```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```

TXdata = {0x8302} / RXdata = {0x----} /* return to Configuration mode */

```

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

[Sample Flow 2 (SPI Normal mode)]

To read upper 16 bits of temperature, gyroscope, and accelerometer data:

(1) Power-on sequence. Please refer to *Section 7.1.1*.(2) Filter setting sequence. Please refer to *Section 7.1.8*.

(3) Configure Sampling mode.

```

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */
TXdata = {0x8504} / RXdata = {0x----} /* 125 Sps */
TXdata = {0x8800} / RXdata = {0x----} /* disable UART Auto mode, just in case. */
TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----} /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```

TXdata = {0x0600} / RXdata = {0x----} /* FLAG read command */
TXdata = {0x0E00} / RXdata = {FLAG} /* TEMP_HIGH read command */
TXdata = {0x1200} / RXdata = {TEMP_HIGH} /* XGYRO_HIGH read command */
TXdata = {0x1600} / RXdata = {XGYRO_HIGH} /* YGYRO_HIGH read command */
TXdata = {0x1A00} / RXdata = {YGYRO_HIGH} /* ZGYRO_HIGH read command */
TXdata = {0x1E00} / RXdata = {ZGYRO_HIGH} /* XACCL_HIGH read command */
TXdata = {0x2200} / RXdata = {XACCL_HIGH} /* YACCL_HIGH read command */

```

```

TXdata = {0x2600} / RXdata = {YACCL_HIGH} /* ZACCL_HIGH read command */
TXdata = {0x0800} / RXdata = {ZACCL_HIGH} /* GPIO read command */
TXdata = {0x0A00} / RXdata = {GPIO} /* COUNT read command */
TXdata = {0x----} / RXdata = {COUNT}

```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x8302} / RXdata = {0x----} /* return to Configuration mode */
```

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

[Sample Flow 3 (SPI Burst mode)]

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Filter setting sequence. Please refer to *Section 7.1.8*.

(3) Configure Sampling mode.

```

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */
TXdata = {0x8504} / RXdata = {0x----} /* 125 Sps */
TXdata = {0x8800} / RXdata = {0x----} /* disable UART Auto mode, just in case. */
TXdata = {0x8C06} / RXdata = {0x----} /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8DF0} / RXdata = {0x----} /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F70} / RXdata = {0x----} /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits */
TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----} /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```

TXdata = {0x8000} / RXdata = {0x----} /* BURST command */
TXdata = {0x----} / RXdata = {FLAG}.
TXdata = {0x----} / RXdata = {TEMP_HIGH}
TXdata = {0x----} / RXdata = {TEMP_LOW}
TXdata = {0x----} / RXdata = {XGYRO_HIGH}
TXdata = {0x----} / RXdata = {XGYRO_LOW}
TXdata = {0x----} / RXdata = {YGYRO_HIGH}
TXdata = {0x----} / RXdata = {YGYRO_LOW}
TXdata = {0x----} / RXdata = {ZGYRO_HIGH}
TXdata = {0x----} / RXdata = {ZGYRO_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 = {GPIO}
TXdata = {0x----} / RXdata = {COUNT}

```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x8302} / RXdata = {0x----} /* return to Configuration mode */
```

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

[Sample Flow 4 (SPI Burst mode)]

To read upper 16 bits of temperature, gyroscope, and accelerometer data:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Filter setting sequence. Please refer to *Section 7.1.8*.

(3) Configure Sampling mode.

```
TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */
TXdata = {0x8504} / RXdata = {0x----} /* 125 Sps */
TXdata = {0x8800} / RXdata = {0x----} /* disable UART Auto mode, just in case. */
TXdata = {0x8C06} / RXdata = {0x----} /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8DF0} / RXdata = {0x----} /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F00} / RXdata = {0x----} /* TEMP = 16 bits, Gyro = 16 bits, ACCL = 16 bits */
TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----} /* set to Sampling mode */
```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```
TXdata = {0x8000} / RXdata = {0x----} /* BURST command */
TXdata = {0x----} / RXdata = {FLAG}
TXdata = {0x----} / RXdata = {TEMP_HIGH}
TXdata = {0x----} / RXdata = {XGYRO_HIGH}
TXdata = {0x----} / RXdata = {YGYRO_HIGH}
TXdata = {0x----} / RXdata = {ZGYRO_HIGH}
TXdata = {0x----} / RXdata = {XACCL_HIGH}
TXdata = {0x----} / RXdata = {YACCL_HIGH}
TXdata = {0x----} / RXdata = {ZACCL_HIGH}
TXdata = {0x----} / RXdata = {GPIO}
TXdata = {0x----} / RXdata = {COUNT}
```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x8302} / RXdata = {0x----} /* return to Configuration mode */
```

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

7.1.4 Self Test (SPI)

The following shows a self test execution procedure:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Execute the self test.

(a) Send the self test command.

```
TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */
TXdata = {0x8304} / RXdata = {0x----} /* Self test command */
```

(b) Wait until the self test has finished.

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

```
TXdata = {0x0200} / RXdata = {0x----} /* MSC_CTRL read command */
TXdata = {0x0000} / RXdata = {MSC_CTRL} /* get response */
```

Confirm the SELF_TEST bit.

When SELF_TEST becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

Confirm the ST_ERR bits. ST_ERR is DIAG_STAT[0x04 (W0)]'s bit[14:11].

```
TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */
TXdata = {0x0400} / RXdata = {0x----} /* DIAG_STAT read command */
```

TXdata = {0x0000} / RXdata = {DIAG_STAT} /* get response */

Confirm if all the ST_ERR bits are 0.

If all the ST_ERR bits are 0, the test has finished successfully. Otherwise, an error has occurred.

-: don't care

7.1.5 Software Reset (SPI)

The following shows a software reset execution procedure:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Issue a software reset.

(a) Send software reset command.

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */

TXdata = {0x8A80} / RXdata = {0x----} /* Software reset command */

(b) Wait 800 ms.

-: don't care

7.1.6 Flash Test (SPI)

The following shows a flash test execution procedure:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Execute the flash test.

(a) Send the flash test command.

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */

TXdata = {0x8308} / RXdata = {0x----} /* Flash test command */

(b) Wait until the flash test has finished.

Wait until the 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 = {0x0000} / RXdata = {MSC_CTRL} /* get response */

Confirm the FLASH_TEST bit.

When FLASH_TEST becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

Confirm the FLASH_ERR bit. FLASH_ERR is DIAG_STAT[0x04 (W0)]'s bit[2].

TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */

TXdata = {0x0400} / RXdata = {0x----} /* DIAG_STAT read command */

TXdata = {0x0000} / RXdata = {DIAG_STAT} /* get response */

Confirm if FLASH_ERR is 0.

If FLASH_ERR is 0, the test has finished successfully. Otherwise, an error has occurred.

-: don't care

7.1.7 Flash Backup (SPI)

The following shows a flash backup execution procedure:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Write the desired settings to the registers that support flash backup shown in Table 6.1. Please refer to *Section 7.1.2*.

(3) Execute the flash backup.

(a) Send the flash backup command.

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */

TXdata = {0x8A08} / RXdata = {0x----} /* Flash backup command */

(b) Wait until the flash backup has finished.

Wait until the 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 = {0x0000} / RXdata = {GLOB_CMD} /* get response */

Confirm the FLASH_BACKUP bit.

When FLASH_BACKUP becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

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

TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */

TXdata = {0x0400} / RXdata = {0x----} /* DIAG_STAT read command */

TXdata = {0x0000} / RXdata = {DIAG_STAT} /* get response */

Confirm if FLASH_BU_ERR is 0.

If FLASH_BU_ERR is 0, the backup has finished successfully. Otherwise, an error has occurred.

-: don't care

7.1.8 Filter Setting (SPI)

The following shows a filter setting procedure:

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Configure the filter.

(a) Send the filter setting command for the moving average filter and TAP32.

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */

TXdata = {0x8605} / RXdata = {0x----} /* Filter setting command */

(b) Wait until the filter setting has finished.

Wait until the 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 = {0x0000} / RXdata = {FILTER_CTRL} /* get response */

Confirm the FILTER_STAT bit.

When FILTER_STAT becomes 0, this step ends. Otherwise, repeat (b).

7.1.9 Attitude Output (SPI)

[Sample Flow 1 (SPI Normal mode)]

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Filter setting sequence. Please refer to *Section 7.1.8*.

(3) Configure Sampling mode.

TXdata = {0xFE01} / RXdata = {0x----} /* WINDOW = 1 */

TXdata = {0x8509} / RXdata = {0x----} /* 200 Sps */

TXdata = {0x8800} / RXdata = {0x----} /* disable UART Auto mode, just in case. */

TXdata = {0x9504} / RXdata = {0x----} /* Inclination mode, enable ATTI_ON */

TXdata = {0xFE00} / RXdata = {0x----} /* WINDOW = 0 */

TXdata = {0x8301} / RXdata = {0x----} /* set to Sampling mode */

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

TXdata = {0x0600} / RXdata = {0x----} /* FLAG read command */

TXdata = {0x0E00} / RXdata = {FLAG} /* TEMP_HIGH read command */

TXdata = {0x1000} / RXdata = {TEMP_HIGH} /* TEMP_LOW read command */

TXdata = {0x1200} / RXdata = {TEMP_LOW} /* XGYRO_HIGH read command */

TXdata = {0x1400} / RXdata = {XGYRO_HIGH} /* XGYRO_LOW read command */

TXdata = {0x1600} / RXdata = {XGYRO_LOW} /* YGYRO_HIGH read command */

TXdata = {0x1800} / RXdata = {YGYRO_HIGH} /* YGYRO_LOW read command */

TXdata = {0x1A00} / RXdata = {YGYRO_LOW} /* ZGYRO_HIGH read command */

TXdata = {0x1C00} / RXdata = {ZGYRO_HIGH} /* ZGYRO_LOW read command */

TXdata = {0x1E00} / RXdata = {ZGYRO_LOW} /* XACCL_HIGH read command */

TXdata = {0x2000} / RXdata = {XACCL_HIGH} /* XACCL_LOW read command */

TXdata = {0x2200} / RXdata = {XACCL_LOW} /* YACCL_HIGH read command */

TXdata = {0x2400} / RXdata = {YACCL_HIGH} /* YACCL_LOW read command */

```

TXdata = {0x2600} / RXdata = {YACCL_LOW}      /* ZACCL_HIGH read command */
TXdata = {0x2800} / RXdata = {ZACCL_HIGH}      /* ZACCL_LOW read command */
TXdata = {0x6400} / RXdata = {ZACCL_LOW}        /* ANG1_HIGH read command */
TXdata = {0x6600} / RXdata = { ANG1_HIGH }      /* ANG1_LOW read command */
TXdata = {0x6800} / RXdata = { ANG1_LOW }       /* ANG2_HIGH read command */
TXdata = {0x6A00} / RXdata = { ANG2_HIGH }      /* ANG2_LOW read command */
TXdata = {0x0800} / RXdata = { ANG2_LOW }       /* GPIO read command */
TXdata = {0x0A00} / RXdata = {GPIO}             /* COUNT read command */
TXdata = {0x----} / RXdata = {COUNT}

```

/* ANG3 (YAW) is fixed at 0 in Inclination mode. */

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x8302} / RXdata = {0x----}          /* return to Configuration mode */
```

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

[Sample Flow 2 (SPI Burst mode)]

(1) Power-on sequence. Please refer to *Section 7.1.1*.

(2) Filter setting sequence. Please refer to *Section 7.1.8*.

(3) Configure Sampling mode.

```

TXdata = {0xFE01} / RXdata = {0x----}          /* WINDOW = 1 */
TXdata = {0x8509} / RXdata = {0x----}          /* 200 Sps */
TXdata = {0x8800} / RXdata = {0x----}          /* disable UART Auto mode, just in case. */
TXdata = {0x8C06} / RXdata = {0x----}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8DF1} / RXdata = {0x----}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on, ATTI = on */
TXdata = {0x8F71} / RXdata = {0x----}          /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits, ATTI = 32 bits */
TXdata = {0x9504} / RXdata = {0x----}          /* Inclination mode, enable ATTI_ON */
TXdata = {0xFE00} / RXdata = {0x----}          /* WINDOW = 0 */
TXdata = {0x8301} / RXdata = {0x----}          /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```

TXdata = {0x8000} / RXdata = {0x----}          /* BURST command */
TXdata = {0x----} / RXdata = {FLAG}
TXdata = {0x----} / RXdata = {TEMP_HIGH}
TXdata = {0x----} / RXdata = {TEMP_LOW}
TXdata = {0x----} / RXdata = {XGYRO_HIGH}
TXdata = {0x----} / RXdata = {XGYRO_LOW}
TXdata = {0x----} / RXdata = {YGYRO_HIGH}
TXdata = {0x----} / RXdata = {YGYRO_LOW}
TXdata = {0x----} / RXdata = {ZGYRO_HIGH}
TXdata = {0x----} / RXdata = {ZGYRO_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 = {ANG1_HIGH}
TXdata = {0x----} / RXdata = {ANG1_LOW}
TXdata = {0x----} / RXdata = {ANG2_HIGH}
TXdata = {0x----} / RXdata = {ANG2_LOW}
TXdata = {0x----} / RXdata = {ANG3_HIGH}

```

/* ANG3 is fixed at 0 in Inclination mode */

TXdata = {0x----} / RXdata = {ANG3_LOW} /* ANG3 is fixed at 0 in Inclination mode */
 TXdata = {0x----} / RXdata = {GPIO}
 TXdata = {0x----} / RXdata = {COUNT}

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

TXdata = {0x8302} / RXdata = {0x----} /* return to Configuration mode */

-: don't care

Note

Please remember to wait until the Data Ready signal is asserted.

7.1.10 Frame Alignment Correction (SPI)

[Sample Flow (SPI normal mode)]

Rotate the gyro triad and accelerometer triad -30 degrees around y-axis

Gyro sensor axis / Acceleration sensor axis

TXdata = {0xB86C} / RXdata = {0x----} /* R_MATRIX_G_M11 = 0.866 */
 TXdata = {0xB937} / RXdata = {0x----}
 TXdata = {0xBA00} / RXdata = {0x----} /* R_MATRIX_G_M12 = 0 */
 TXdata = {0xBB00} / RXdata = {0x----}
 TXdata = {0xBCFF} / RXdata = {0x----} /* R_MATRIX_G_M13 = 0.5 */
 TXdata = {0xBD1F} / RXdata = {0x----}
 TXdata = {0xBE00} / RXdata = {0x----} /* R_MATRIX_G_M21 = 0 */
 TXdata = {0xBF00} / RXdata = {0x----}
 TXdata = {0xC000} / RXdata = {0x----} /* R_MATRIX_G_M22 = 1 */
 TXdata = {0xC140} / RXdata = {0x----}
 TXdata = {0xC200} / RXdata = {0x----} /* R_MATRIX_G_M23 = 0 */
 TXdata = {0xC300} / RXdata = {0x----}
 TXdata = {0xC400} / RXdata = {0x----} /* R_MATRIX_G_M31 = - 0.5 */
 TXdata = {0xC5E0} / RXdata = {0x----}
 TXdata = {0xC600} / RXdata = {0x----} /* R_MATRIX_G_M32 = 0 */
 TXdata = {0xC700} / RXdata = {0x----}
 TXdata = {0xC86C} / RXdata = {0x----} /* R_MATRIX_G_M33 = 0.866 */
 TXdata = {0xC937} / RXdata = {0x----}

7.2 UART Sequence

7.2.1 Power-on Sequence (UART)

The following shows a power-on sequence:

- (1) Power-on.
- (2) Wait 800 ms.
- (3) Wait until the NOT_READY bit goes to 0. NOT_READY is GLOB_CMD[0x0A (W1)]'s bit[10].


```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x0A, 0x00, 0x0d}          /* GLOB_CMD read command */
RXdata = {0x0A, MSByte, LSByte, 0x0d} /* get response */
```

 Confirm the NOT_READY bit.
 When NOT_READY becomes 0, this step ends. Otherwise, please repeat (3).
- (4) Confirm the HARD_ERR bits. HARD_ERR is DIAG_STAT[0x04 (W0)]'s bit[6:5].


```
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x04, 0x00, 0x0d}          /* DIAG_STAT read command */
RXdata = {0x04, MSByte, LSByte, 0x0d} /* get response */
```

 Confirm if HARD_ERR is 00.
 If HARD_ERR is 00, the IMU is operating normally. Otherwise, the IMU is faulty.

7.2.2 Register Read and Write (UART)

[Read Example]

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

```
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x02, 0x00, 0x0d}          /* command */
RXdata = {0x02, 0x04, 0x00, 0x0d}    /* response */
```

0x04 in 2nd byte of RXdata indicates that the device is in Configuration mode.

0x00 in 3rd byte of RXdata is Reserved.

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

[Write Example]

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

```
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* command */
RXdata = w/o response
```

By sending this command, the IMU enters Sampling mode.

Please note that write data unit is 8 bits.

7.2.3 Sampling Data (UART)

[Sample Flow 1 (UART Auto mode)]

- (1) Power-on sequence. Please refer to *Section 7.2.1*.
- (2) Filter setting sequence. Please refer to *Section 7.2.9*.
- (3) Configure Sampling mode.


```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x04, 0x0d}          /* 125 Sps */
TXdata = {0x88, 0x01, 0x0d}          /* UART Auto mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF0, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F, 0x70, 0x0d}          /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits */
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */
```
- (4) Receive sampling data.

(a) Read data.

```
RXdata = {0x80,  FLAG_Hi,      FLAG_Lo,
              TEMP_HIGH_Hi,  TEMP_HIGH_Lo,  TEMP_LOW_Hi,  TEMP_LOW_Lo,
              XGYRO_HIGH_Hi, XGYRO_HIGH_Lo, XGYRO_LOW_Hi, XGYRO_LOW_Lo,
              YGYRO_HIGH_Hi, YGYRO_HIGH_Lo, YGYRO_LOW_Hi, YGYRO_LOW_Lo,
              ZGYRO_HIGH_Hi, ZGYRO_HIGH_Lo, ZGYRO_LOW_Hi, ZGYRO_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,
              GPIO_Hi,      GPIO_Lo,
              COUNT_Hi,     COUNT_Lo,     0x0d}
```

(b) Repeat (a).

(5) Exit Sampling mode.

```
TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */
```

[Sample Flow 2 (UART Auto mode)]

To read upper 16 bits of temperature, gyroscope, and accelerometer data.

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x04, 0x0d}          /* 125 Sps */
TXdata = {0x88, 0x01, 0x0d}          /* UART Auto mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF0, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F, 0x00, 0x0d}          /* TEMP = 16 bits, Gyro = 16 bits, ACCL = 16 bits */
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */
```

(4) Receive sampling data.

(a) Read data.

```
RXdata = {0x80,  FLAG_Hi,      FLAG_Lo,
              TEMP_HIGH_Hi,  TEMP_HIGH_Lo,
              XGYRO_HIGH_Hi, XGYRO_HIGH_Lo,
              YGYRO_HIGH_Hi, YGYRO_HIGH_Lo,
              ZGYRO_HIGH_Hi, ZGYRO_HIGH_Lo,
              XACCL_HIGH_Hi, XACCL_HIGH_Lo,
              YACCL_HIGH_Hi, YACCL_HIGH_Lo,
              ZACCL_HIGH_Hi, ZACCL_HIGH_Lo,
              GPIO_Hi,      GPIO_Lo,
              COUNT_Hi,     COUNT_Lo,     0x0d}
```

(b) Repeat (a).

(5) Exit Sampling mode.

```
TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */
```

[Sample Flow 3 (UART Burst mode)]

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x04, 0x0d}          /* 125 Sps */
TXdata = {0x88, 0x00, 0x0d}          /* UART Manual mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF0, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F, 0x70, 0x0d}          /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits */
```

```
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */
```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```
TXdata = {0x80, 0x00, 0x0d}          /* BURST command */
RXdata = {0x80,  FLAG_Hi,             FLAG_Lo,
          TEMP_HIGH_Hi,  TEMP_HIGH_Lo,  TEMP_LOW_Hi,  TEMP_LOW_Lo,
          XGYRO_HIGH_Hi, XGYRO_HIGH_Lo, XGYRO_LOW_Hi, XGYRO_LOW_Lo,
          YGYRO_HIGH_Hi, YGYRO_HIGH_Lo, YGYRO_LOW_Hi, YGYRO_LOW_Lo,
          ZGYRO_HIGH_Hi, ZGYRO_HIGH_Lo, ZGYRO_LOW_Hi, ZGYRO_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,
          GPIO_Hi,      GPIO_Lo,
          COUNT_Hi,    COUNT_Lo,      0x0d}
```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */
```

Note

Please remember to wait until the Data Ready signal is asserted.

[Sample Flow 4 (UART Burst mode)]

To read upper 16 bits of temperature, gyroscope, and accelerometer data.

(1) Power-on sequence. Please refer to *Section 7.2.1*.(2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x04, 0x0d}          /* 125 Sps */
TXdata = {0x88, 0x00, 0x0d}          /* UART Manual mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF0, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F, 0x00, 0x0d}          /* TEMP = 16 bits, Gyro = 16 bits, ACCL = 16 bits */
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */
```

(4) Receive sampling data.

(a) Wait until the Data Ready signal is asserted.

(b) Read data.

```
TXdata = {0x80, 0x00, 0x0d}          /* BURST command */
RXdata = {0x80,  FLAG_Hi,             FLAG_Lo,
          TEMP_HIGH_Hi,  TEMP_HIGH_Lo,
          XGYRO_HIGH_Hi, XGYRO_HIGH_Lo,
          YGYRO_HIGH_Hi, YGYRO_HIGH_Lo,
          ZGYRO_HIGH_Hi, ZGYRO_HIGH_Lo,
          XACCL_HIGH_Hi, XACCL_HIGH_Lo,
          YACCL_HIGH_Hi, YACCL_HIGH_Lo,
          ZACCL_HIGH_Hi, ZACCL_HIGH_Lo,
          GPIO_Hi,      GPIO_Lo,
          COUNT_Hi,    COUNT_Lo,      0x0d}
```

(c) Repeat from (a) to (b).

(5) Exit Sampling mode.

```
TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */
```

Note

Please remember to wait until the Data Ready signal is asserted.

- NOTE:
- Please note that read data unit is 16 bits, and Most Significant Byte first.
 - Please note that write data unit is 8 bits.
 - XGYRO_HIGH_Hi means MSByte of XGYRO_HIGH data
 - XGYRO_HIGH_Lo means LSByte of XGYRO_LOW data

7.2.4 Self Test (UART)

The following shows a self test execution procedure:

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Execute the self test.

(a) Send the self test command.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x83, 0x04, 0x0d}          /* Self test command */
```

(b) Wait until the self test has finished.

Wait until the SELF_TEST bit goes to 0. SELF_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 the SELF_TEST bit.

When SELF_TEST becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

Confirm the ST_ERR bits. ST_ERR is DIAG_STAT[0x04 (W0)]'s bit[14:11].

```
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x04, 0x00, 0x0d}          /* DIAG_STAT read command */
RXdata = {0x04, MSByte, LSByte, 0x0d} /* get response */
```

Confirm if all the ST_ERR bits are 0.

If all the ST_ERR bits are 0, the test has finished successfully. Otherwise, an error has occurred.

7.2.5 Software Reset (UART)

The following shows a software reset execution procedure:

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Issue a software reset.

(a) Send software reset command.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x8A, 0x80, 0x0d}          /* Software reset command */
```

(b) Wait 800 ms.

7.2.6 Flash Test (UART)

The following shows a flash test execution procedure:

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Execute the flash test.

(a) Send the flash test command.

```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x83, 0x08, 0x0d}          /* Flash test command */
```

(b) Wait until the flash test has finished.

Wait until the 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 the FLASH_TEST bit.

When FLASH_TEST becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

Confirm the FLASH_ERR bit. FLASH_ERR is DIAG_STAT[0x04 (W0)]'s bit[2].

TXdata = {0xFE, 0x00, 0x0d}

/* WINDOW = 0 */

TXdata = {0x04, 0x00, 0x0d}

/* DIAG_STAT read command */

RXdata = {0x04, MSByte, LSByte, 0x0d}

/* get response */

Confirm if FLASH_ERR is 0.

If FLASH_ERR is 0, the test has finished successfully. Otherwise, an error has occurred.

7.2.7 Flash Backup (UART)

The following shows a flash backup execution procedure:

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Write the desired settings to the registers that support flash backup shown in Table 6.1. Please refer to *Section 7.2.2*.

(3) Execute the flash backup.

(a) Send the flash backup command.

TXdata = {0xFE, 0x01, 0x0d}

/* WINDOW = 1 */

TXdata = {0x8A, 0x08, 0x0d}

/* Flash backup command */

(b) Wait until the flash backup has finished.

Wait until the 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 the FLASH_BACKUP bit.

When FLASH_BACKUP becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

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

TXdata = {0xFE, 0x00, 0x0d}

/* WINDOW = 0 */

TXdata = {0x04, 0x00, 0x0d}

/* DIAG_STAT read command */

RXdata = {0x04, MSByte, LSByte, 0x0d}

/* get response */

Confirm if FLASH_BU_ERR is 0.

If FLASH_BU_ERR is 0, the backup has finished successfully. Otherwise, an error has occurred.

7.2.8 Initial Backup (UART)

The following shows an initial backup execution procedure:

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Execute initial backup.

(a) Send the initial backup command.

TXdata = {0xFE, 0x01, 0x0d}

/* WINDOW = 1 */

TXdata = {0x8A, 0x10, 0x0d}

/* Initial backup command */

(b) Wait until the initial backup has finished.

Wait until the INITIAL_BACKUP bit goes to 0. INITIAL_BACKUP is GLOB_CMD[0x0A (W1)]'s bit[4].

TXdata = {0x0A, 0x00, 0x0d}

/* GLOB_CMD read command */

RXdata = {0x0A, MSByte, LSByte, 0x0d}

/* get response */

Confirm the INITIAL_BACKUP bit.

When INITIAL_BACKUP becomes 0, this step ends. Otherwise, repeat (b).

(c) Confirm the result.

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

TXdata = {0xFE, 0x00, 0x0d}

/* WINDOW = 0 */

TXdata = {0x04, 0x00, 0x0d}

/* DIAG_STAT read command */

RXdata = {0x04, MSByte, LSByte, 0x0d}

/* get response */

Confirm if FLASH_BU_ERR is 0.

If FLASH_BU_ERR is 0, the backup has finished successfully. Otherwise, an error has occurred.

(3) Issue a software reset. Please refer to *Section 7.2.5*.

7.2.9 Filter Setting (UART)

The following shows a filter setting procedure:

- (1) Power-on sequence. Please refer to *Section 7.2.1*.
- (2) Configure the filter.
 - (a) Send the filter setting command for the moving average filter and TAP32.


```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x86, 0x05, 0x0d}          /* Filter setting command */
```
 - (b) Wait until the filter setting has finished.

Wait until the 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 the FILTER_STAT bit.
When FILTER_STAT becomes 0, this step ends. Otherwise, repeat (b).

7.2.10 Auto Start (UART only)

The following shows a procedure to configure the Auto Start function and read data:

- (1) Power-on sequence. Please refer to *Section 7.2.1*.
- (2) Configure the following registers:


```
TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x04, 0x0d}          /* 125 Sps */
TXdata = {0x86, 0x04, 0x0d}          /* TAP = 16 */
TXdata = {0x88, 0x03, 0x0d}          /* UART Auto mode, Auto start = on */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF0, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on */
TXdata = {0x8F, 0x70, 0x0d}          /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits */
```
- (2) Execute Flash backup. Please refer to *Section 7.2.7*.
- (4) Power-off.
- (5) Power-on.
- (6) Wait 800 ms.
- (7) Receive sampling data.
 - (a) Read data.


```
RXdata = {0x80, FLAG_Hi, FLAG_Lo,
            TEMP_HIGH_Hi, TEMP_HIGH_Lo, TEMP_LOW_Hi, TEMP_LOW_Lo,
            XGYRO_HIGH_Hi, XGYRO_HIGH_Lo, XGYRO_LOW_Hi, XGYRO_LOW_Lo,
            YGYRO_HIGH_Hi, YGYRO_HIGH_Lo, YGYRO_LOW_Hi, YGYRO_LOW_Lo,
            ZGYRO_HIGH_Hi, ZGYRO_HIGH_Lo, ZGYRO_LOW_Hi, ZGYRO_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,
            GPIO_Hi, GPIO_Lo,
            COUNT_Hi, COUNT_Lo, 0x0d}
```
 - (b) Repeat (a).
- (8) If you want to stop sampling,


```
TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */
```

7.2.11 Attitude Output (UART)

[Sample Flow 1 (UART Auto mode)]

- (1) Power-on sequence. Please refer to *Section 7.2.1*.
- (2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```

TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x09, 0x0d}          /* 200 Sps */
TXdata = {0x88, 0x01, 0x0d}          /* UART Auto mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF1, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on, ATTI = on */
TXdata = {0x8F, 0x71, 0x0d}          /* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits, ATTI = 32 bits */
TXdata = {0x95, 0x04, 0x0d}          /* Inclination mode, enable ATTI_ON */
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Read data.

```

RXdata = {0x80,  FLAG_Hi,      FLAG_Lo,
            TEMP_HIGH_Hi,    TEMP_HIGH_Lo,  TEMP_LOW_Hi,  TEMP_LOW_Lo,
            XGYRO_HIGH_Hi,   XGYRO_HIGH_Lo, XGYRO_LOW_Hi, XGYRO_LOW_Lo,
            YGYRO_HIGH_Hi,   YGYRO_HIGH_Lo, YGYRO_LOW_Hi, YGYRO_LOW_Lo,
            ZGYRO_HIGH_Hi,   ZGYRO_HIGH_Lo, ZGYRO_LOW_Hi, ZGYRO_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,
            ANG1_HIGH_Hi,    ANG1_HIGH_Lo,  ANG1_LOW_Hi,  ANG1_LOW_Lo,
            ANG2_HIGH_Hi,    ANG2_HIGH_Lo,  ANG2_LOW_Hi,  ANG2_LOW_Lo,
            ANG3_HIGH_Hi,    ANG3_HIGH_Lo,  ANG3_LOW_Hi,  ANG3_LOW_Lo,
            GPIO_Hi,         GPIO_Lo,
            COUNT_Hi,        COUNT_Lo,      0x0d}

```

/* ANG3 (YAW) is fixed at 0 in Inclination mode. */

(b) Repeat (a).

(5) Exit Sampling mode.

```

TXdata = {0x83, 0x02, 0x0d}          /* return to Configuration mode */

```

[Sample Flow 2 (UART Auto mode)]

To read upper 16 bits of temperature, gyroscope, and accelerometer data.

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```

TXdata = {0xFE, 0x01, 0x0d}          /* WINDOW = 1 */
TXdata = {0x85, 0x09, 0x0d}          /* 200 Sps */
TXdata = {0x88, 0x01, 0x0d}          /* UART Auto mode */
TXdata = {0x8C, 0x06, 0x0d}          /* GPIO = on, COUNT = on, CheckSum = off */
TXdata = {0x8D, 0xF1, 0x0d}          /* FLAG = on, TEMP = on, Gyro = on, ACCL = on, ATTI = on */
TXdata = {0x8F, 0x00, 0x0d}          /* TEMP = 16 bits, Gyro = 16 bits, ACCL = 16 bits, ATTI = 16 bits */
TXdata = {0x95, 0x04, 0x0d}          /* Inclination mode, enable ATTI_ON */
TXdata = {0xFE, 0x00, 0x0d}          /* WINDOW = 0 */
TXdata = {0x83, 0x01, 0x0d}          /* set to Sampling mode */

```

(4) Receive sampling data.

(a) Read data.

```

RXdata = {0x80,  FLAG_Hi,      FLAG_Lo,
            TEMP_HIGH_Hi,    TEMP_HIGH_Lo,
            XGYRO_HIGH_Hi,   XGYRO_HIGH_Lo,
            YGYRO_HIGH_Hi,   YGYRO_HIGH_Lo,
            ZGYRO_HIGH_Hi,   ZGYRO_HIGH_Lo,
            XACCL_HIGH_Hi,   XACCL_HIGH_Lo,
            YACCL_HIGH_Hi,   YACCL_HIGH_Lo,
            ZACCL_HIGH_Hi,   ZACCL_HIGH_Lo,

```

```

ANG1_HIGH_Hi,   ANG1_HIGH_Lo,
ANG2_HIGH_Hi,   ANG2_HIGH_Lo,
ANG3_HIGH_Hi,   ANG3_HIGH_Lo,
GPIO_Hi,        GPIO_Lo,
COUNT_Hi,      COUNT_Lo,      0x0d}

```

/* ANG3 (YAW) is fixed at 0 in Inclination mode. */

(b) Repeat (a).

(5) Exit Sampling mode.

```
TXdata = {0x83, 0x02, 0x0d}
```

/* return to Configuration mode */

7.2.12 QUATERNION Output (UART)

[Sample Flow (UART Auto mode)]

(1) Power-on sequence. Please refer to *Section 7.2.1*.

(2) Filter setting sequence. Please refer to *Section 7.2.9*.

(3) Configure Sampling mode.

```
TXdata = {0xFE, 0x01, 0x0d}
```

/* WINDOW = 1 */

```
TXdata = {0x85, 0x09, 0x0d}
```

/* 200 Sps */

```
TXdata = {0x88, 0x01, 0x0d}
```

/* UART Auto mode */

```
TXdata = {0x8C, 0x06, 0x0d}
```

/* GPIO = on, COUNT = on, CheckSum = off */

```
TXdata = {0x8D, 0xF3, 0x0d}
```

/* FLAG = on, TEMP = on, Gyro = on, ACCL = on, QTN = on, ATTI = on */

```
TXdata = {0x8F, 0x73, 0x0d}
```

/* TEMP = 32 bits, Gyro = 32 bits, ACCL = 32 bits, QTN = 32 bits, ATTI = 32 bits */

```
TXdata = {0x95, 0x04, 0x0d}
```

/* Inclination mode, enable QTN_ON, enable ATTI_ON */

```
TXdata = {0xFE, 0x00, 0x0d}
```

/* WINDOW = 0 */

```
TXdata = {0x83, 0x01, 0x0d}
```

/* set to Sampling mode */

(4) Receive sampling data.

```

RXdata = {0x80,   FLAG_Hi,           FLAG_Lo,
           TEMP_HIGH_Hi,   TEMP_HIGH_Lo,   TEMP_LOW_Hi,   TEMP_LOW_Lo,
           XGYRO_HIGH_Hi,  XGYRO_HIGH_Lo, XGYRO_LOW_Hi, XGYRO_LOW_Lo,
           YGYRO_HIGH_Hi,  YGYRO_HIGH_Lo, YGYRO_LOW_Hi, YGYRO_LOW_Lo,
           ZGYRO_HIGH_Hi,  ZGYRO_HIGH_Lo, ZGYRO_LOW_Hi, ZGYRO_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,
           QTN0_HIGH_Hi,   QTN0_HIGH_Lo, QTN0_LOW_Hi,   QTN0_LOW_Lo,
           QTN1_HIGH_Hi,   QTN1_HIGH_Lo, QTN1_LOW_Hi,   QTN1_LOW_Lo,
           QTN2_HIGH_Hi,   QTN2_HIGH_Lo, QTN2_LOW_Hi,   QTN2_LOW_Lo,
           QTN3_HIGH_Hi,   QTN3_HIGH_Lo, QTN3_LOW_Hi,   QTN3_LOW_Lo,
           ROLL_HIGH_Hi,   ROLL_HIGH_Lo, ROLL_LOW_Hi,   ROLL_LOW_Lo,
           PITCH_HIGH_Hi,  PITCH_HIGH_Lo, PITCH_LOW_Hi,  PITCH_LOW_Lo,
           YAW_HIGH_Hi,    YAW_HIGH_Lo,   YAW_LOW_Hi,    YAW_LOW_Lo,
           GPIO_Hi,        GPIO_Lo,
           COUNT_Hi,       COUNT_Lo,       0x0d}

```

/* YAW is fixed at 0 in Inclination mode */

/* Quaternion is fixed point with upper 2 (signed) integer bits and lower 30 decimal bits */

8. Handling Notes

8.1 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.
- The casing of this product is electrically conductive. When the product is connected or mounted to the circuit board, ensure the board substrate or board wiring pattern does not short-circuit or contact to the case.

8.2 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 °C to +35 °C Normal humidity: 45% RH to 85% RH (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.

8.3 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 IMU may be damaged permanently. In addition, if you attach the product to the equipment, etc. using connection harness, connect the connection harness to the product first, and then attach it to the equipment, etc.
- 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 accidentally 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.
- 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-test in this product. Discontinue use immediately even when the self-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.

8.4 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 and is clearly attributable to us, we will provide a replacement.

9. Contact

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.cn

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.**

https://global.epson.com/products_and_drivers/sensing_system/contact/

Product Information on www server

https://global.epson.com/products_and_drivers/sensing_system/