

Version

1.13

MICROSTRAIN, INC.

Microminiature Sensors

3DM-GX2™ Data Communications Protocol

For the Inertia-Link® and 3DM-GX2™ Orientation Sensors

©2007 by MicroStrain, Inc.
310 Hurricane Lane
Williston, VT 05495
Phone 802-862-6629
Fax 802-863-4093
www.microstrain.com
support@microstrain.com

REVISED: December 13, 2007

Contents

Overview	5
Introduction to the 3DM-GX2™ Data Communications Protocol	5
Serial Port Configuration	6
Communications Commands	6
Command Set Summary	7
Communications Modes	7
Polled Command Mode (default).....	7
Continuous Mode.....	8
Automatic Startup into Continuous Mode	8
Combined Continuous and Polled Mode	9
The Wireless Packet.....	9
Command Packet (from the host PC to the sensor):	10
The ACK/NACK byte.....	10
The Data Packet (from the sensor to the PC):.....	11
Example	11
Command Reference.....	13
Wireless Ping (wireless units only) (0x02).....	13
Raw Accelerometer and Angular Rate Sensor Outputs (0xC1).....	14
Acceleration & Angular Rate (0xC2)	14
DeltaAngle & DeltaVelocity (0xC3)	15
Set Continuous Mode (0xC4)	16
Orientation Matrix (0xC5)	17
Orientation Update Matrix (0xC6).....	17
Scaled Magnetometer Vector (0xC7)	18
Acceleration, Angular Rate & Orientation Matrix (0xC8)	18
Write Accelerometer Bias Correction (0xC9)	19
Write Gyro Bias Correction (0xCA).....	19
Acceleration, Angular Rate & Magnetometer Vector (0xCB)	20
Acceleration, Angular Rate & Magnetometer Vectors & Orientation Matrix (0xCC)	21
Capture Gyro Bias (0xCD)	22
Euler Angles (0xCE).....	23
Euler Angles and Angular Rates (0xCF)	23
Transfer Quantity to Non-Volatile Memory (0xD0)	23
Temperatures (0xD1).....	24
Gyro Stabilized Acceleration, Angular Rate & Magnetometer Vector (0xD2).....	24
DeltaAngle & DeltaVelocity & Magnetometer Vectors (0xD3)	25
Write Word to EEPROM (0xE4).....	26
Read Word from EEPROM (0xE5)	26
Stop Continuous Mode (no reply) (0xFA).....	26
Calculation Cycle, and Data Output Rate	27
Data Quantities Available	28
RawAccel – (3 components).....	28
RawAngRate – (3 components).....	28
Accel – (X, Y, Z components).....	29
StabAccel – (X, Y, Z components).....	29

AngRate – (X, Y and Z components)	30
DeltaVel– (X, Y, Z components).....	30
DeltaAng– (X, Y, Z components).....	30
Magnetometer– (X, Y, Z components)	31
StabMag – (X, Y, Z components).....	31
Timer	31
M	32
C	32
Euler Angles.....	33
AccelBias – (X, Y, Z components).....	33
GyroBias – (X, Y, Z components).....	33
Temperature – (Accel, GyroX, GyroY, GyroZ components).....	34
Orientation Conversion Formulas	35

Overview

This document describes how to communicate with the MicroStrain Inertia-Link® and 3DM-GX2™ series of orientation sensors, firmware version 2.1.03 and higher. These sensors are available with many communications options including USB, RS-232, RS485, RS-422, and wireless 802.15.4 (2.4 GHz). All versions of the sensor support the same MicroStrain **3DM-GX2™ Data Communications Protocol** described here.

Introduction to the 3DM-GX2™ Data Communications Protocol

The 3DM-GX2™ Data Communications Protocol is a set of serial commands and responses designed specifically for MicroStrain's orientation sensors. For the most part, the communications protocol consists of simple single byte binary commands with fixed length binary data records as replies. Most replies include an “echo” byte and a checksum to do simple data integrity checks. All communications with the sensor are accomplished using a standard serial “COM” port.

The Inertia-Link® and 3DM-GX2™ each have several communications options, including RS-232, RS485, RS-422, USB, and wireless (802.15.4). Regardless of which communications option is selected, all communications with the host system are made to look like a serial COM port. This is possible because of special device drivers that make the sensors “appear” to the computer as a serial port. These drivers are either installed along with the software that came with your sensor or automatically installed by the Windows “New Hardware Wizard”. Because all of the sensors appear as a COM port, it is possible to write software for one sensor that is automatically compatible with all the sensors.

Please consult the standard product catalog for a complete list of available communications options.

The only sensors that require special consideration are the wireless versions. Although communications with wireless sensors are through a virtual serial port, the wireless sensors require the use of a “packetized” version of the 3DM-GX2™ Data Communications protocol. This is necessary due to the additional information needed to deliver data to a sensor that may be sharing a radio channel with other sensors. This information is used by the wireless USB base station to route the data to the correct wireless sensor. The packetized commands differ from the non-packetized commands in that they are preceded by a packet header and followed by a packet footer.

It is important to note that you may use the “packetized” version of the protocol with all Inertia-Link® and 3DM-GX2™ sensors regardless of the communications option of the sensor. The sensor firmware detects and responds to both packetized commands and non-packetized commands. However, programming the packet protocol is more complex than programming for the non-packet protocol. Unless you are using or planning to use

the wireless version of the Inertia-Link® or 3DM-GX2™, you should use the standard non-packetized version of the commands.

The standard serial protocol is described below in “[Communications Commands](#)”. The packetized version of the commands is described in “[The Wireless Packet](#)” section.

Serial Port Configuration

All communications with Inertia-Link® and 3DM-GX2™ sensors are accomplished via a serial COM port. This is true for all communications options currently available for these sensors including RS-232, USB, and wireless (802.15.4). The serial port settings are the same for all modules and are shown in the table below:

COM Port Serial Settings	
Baud Rate	115.2K
Parity:	None
Data Bits:	8
Stop Bits:	1

Communications Commands

The host computer controls what data the Inertia-Link® or 3DM-GX2™ will output by issuing one or more single byte commands (in some cases, additional data bytes must follow the command byte). Each command will cause the Inertia-Link® or 3DM-GX2™ to transmit a data record of a fixed number of bytes.

Below is a summary of all commands available for the Inertia-Link® and 3DM-GX2™. Each of the commands is described in detail in the [Command Reference](#) section.

Command Set Summary

Command	Definition
0x02	Wireless Ping (wireless units only)
0xC1	Raw Accelerometer and Angular Rate Sensor Outputs
0xC2	Acceleration & Angular Rate
0xC3	DeltaAngle & DeltaVelocity
0xC4	Set Continuous Mode
0xC5	Orientation Matrix
0xC6	Attitude Update Matrix
0xC7	Magnetometer Vector
0xC8	Acceleration, Angular Rate & Orientation Matrix
0xC9	Write Accelerometer Bias Correction
0xCA	Write Gyro Bias Correction
0xCB	Acceleration, Angular Rate & Magnetometer Vector
0xCC	Accel, Ang Rate & Mag Vectors & Orientation Matrix
0xCD	Capture Gyro Bias
0xCE	Euler Angles
0xCF	Euler Angles and Angular Rates
0xD0	Transfer Quantity to Non-Volatile Memory
0xD1	Temperatures
0xD2	Gyro Stabilized Acceleration, Angular Rate & Magnetometer Vector
0xD3	DeltaAngle & DeltaVelocity & Magnetometer Vectors
0xE4	Write Word to EEPROM
0xE5	Read Word from EEPROM
0xFA	Stop Continuous Mode (no reply)

Communications Modes

There are two modes of communications for the Inertia-Link® and 3DM-GX2™: Polled mode and Continuous mode. In Polled mode, a single data record is transmitted for each command byte received. In Continuous mode, a command byte is stored in memory, and the corresponding data record is output continuously by the Inertia-Link® or 3DM-GX2™ with no further intervention by the host.

Polled Command Mode (default)

In polled mode, the Inertia-Link® or 3DM-GX2™ will transmit a data record each time a command byte is issued by the host computer. The sensor will not output unsolicited data records. The host may issue a command at any time. The Inertia-Link® or 3DM-GX2™ will respond by outputting the corresponding data record upon completion of the current calculation cycle. Multiple commands issued by the host will be buffered on-

board the Inertia-Link® or 3DM-GX2™, with one being processed at the completion of each successive calculation cycle.

Continuous Mode

In Continuous mode, a command byte is stored in memory, and the corresponding data record is output continuously by the Inertia-Link® or 3DM-GX2™ with no further intervention by the host. To enter continuous mode, the host computer must issue the “Set Continuous Mode” command byte (0xC4), followed by two confirmation bytes (0xC129), and followed by another command byte of the user’s choosing. The Inertia-Link® or 3DM-GX2™ responds by outputting the corresponding data packet at the completion of every calculation cycle. This provides a stream of data at the maximum possible rate, and at uniformly spaced time intervals (i.e., the calculation cycle time interval) with no gaps. The host computer must be capable of the buffering and interpreting the data stream at sufficient speed to prevent loss of data. For example, if the host issues the following bytes 0xC4, 0xC1, 0x29, 0xC2, the Inertia-Link® or 3DM-GX2™ will be set into continuous mode and will continuously output the “Acceleration & Angular Rate” data packet. For information on the data rate in the continuous mode see the [Calculation Cycle, and Data Output Rate](#) section.

Once continuous mode is set, it will remain in effect until it is terminated by the host issuing the “Set Continuous Mode” command byte with 0x00 as the Continuous Command Byte (i.e., 0xC4, 0xC1, 0x29, 0x00), or the power to the device is interrupted.

Wireless devices often require repeated transmissions of the termination packet. A single packet is not always successful. This is because the continuous data being broadcast by the Inertia-Link® or 3DM-GX2™ reduces the radio bandwidth available for commands from the host. The base station radio on the host will often time-out when trying to broadcast a command to an Inertia-Link® or 3DM-GX2™ that is in continuous mode.

A new alternate single byte termination command ([0xFA](#)) was added to version 2.1.00 of the firmware. This new single byte command does not generate a response packet. Because of the smaller size of the packet and the absence of the response packet, the wireless bandwidth requirement of this command is much lower than the standard termination packet and it terminates continuous mode much more successfully than the standard termination packet.

Automatic Startup into Continuous Mode

Normally, the Inertia-Link® or 3DM-GX2™ starts in Polled Mode on power-up. In some applications, it may be desirable to have the Inertia-Link® or 3DM-GX2™ enter continuous mode immediately on power-up. To accomplish this, the user can set the value of the LSB of EEPROM address 0xFCA6 to the desired command byte (the MSB of address 0xFCA6 should be set to 0x00 for non-wireless devices or 0x80 for wireless devices). On subsequent power-ups, the Inertia-Link® or 3DM-GX2™ will automatically enter continuous mode with the selected command active. Note that

following power-up the user may subsequently turn continuous mode off by issuing the “Set Continuous Mode” command with the Continuous Command Byte set to 0x00. To disable the automatic power-up entry into continuous mode, set the value of EEPROM address 0xFCA6 to 0x0000.

Combined Continuous and Polled Mode

While in continuous mode, the host computer may still issue individual commands as in polled mode. The responses to these commands will be interleaved with the continuous mode responses. At the completion of each calculation cycle, the Inertia-Link® or 3DM-GX2™ first outputs the response to the continuous mode command if continuous mode is active. The Inertia-Link® or 3DM-GX2™ then outputs the response to any individual command that has been issued in polled mode. In this case, two data packets will be output (one for the continuous command, and one for the polled command) during the same calculation cycle. The host computer’s data interpretation software must be capable of differentiating such data packets.

The Wireless Packet

The wireless versions of the Inertia-Link® and 3DM-GX2™ orientation sensors use the same command set and modes as the non-wireless versions, however these commands must be “wrapped” in a packet header and footer. This is necessary because a wireless sensor system includes a base station radio attached to the host system. The base station needs additional wireless routing information in order to connect to the correct wireless sensor.

*The packet structure used is the **MicroStrain Agile-Link Serial Packet Protocol (ASPP)**. The ASPP protocol is a comprehensive wireless protocol definition, however only the structures relevant to the 3DM-GX2™ Data Communications protocol are described in this document.*

Because the packet structure is only a “wrapper” for the communications commands and replies, the same modes (Polled, Continuous, and Combined Continuous and Polled) may be used. It is important to recognize that “over the air” packet collisions, communications timeouts, and radio interference will cause inevitable packet drops that don’t occur with wired communications. This is especially true when trying to send commands to wireless sensors while they are in continuous mode. The ASPP wireless protocol does not provide guaranteed packet delivery, so the user’s software should be tolerant of packet drops – both sent and received.

Command Packet (from the host PC to the sensor):

The basic structure of the ASPP Command Packet as used by the 3DM-GX2™ Data Communications Protocol looks like this:

Field Name	Bytes	Value
Start Of Packet (SOP):	1	0xAA
Delivery Flags:	1	0x0B
App Data Type:	1	0x00
Node Address:	2	<Node Address>
Command Length:	1	<length of command + 1>
Command MSB	1	0x00
Command LSB	1-100	<command>
Checksum:	2	<Sum of all bytes except the SOP>

The **node address** is a number that specifies the wireless address of the node. This is typically a number between 1 and 100 but can be any integer from 1 to 65534. The node address is a unique identifier for a sensor on a given wireless frequency. Each frequency, or channel, in the 802.15.4 wireless standard used by the wireless sensors can support up to 65534 nodes, each with a unique node address. Practically speaking, however, there can only be a few nodes active on each channel because the baud rate of the channel (approximately 230 Kbits per second) can only carry the data from a few active high data rate nodes simultaneously. The node number and channel on a sensor is preset at the factory but may be changed using the Inertia-Link® desktop application (version 1.1 or later). The channel on the base station is set by the desktop application to match the channel of the node that is selected for communications.

NOTE: The node number is preset at the factory to the last two digits of the unit serial number. For example, for an Inertia-Link® with a serial number of 4200-1076, the node number is preset at the factory to 76.

The ACK/NACK byte

The command packet is always followed by an ACK or NACK byte sent back to the PC from the base station or sensor. The ACK byte value is **0xAA** and the NACK byte value is **0x21**. The ACK signifies that the packet was accepted and broadcast by the base station to the sensor (wireless sensor) or accepted by the sensor (non-wireless sensor). A NACK indicates that the packet was not accepted. A NACK usually means that the packet was not well formed – for example, the SOP was not found or the checksum was incorrect.

The Data Packet (from the sensor to the PC):

The structure of a reply packet from the sensor is as follows:

Field Name	Bytes	Value
Start Of Packet (SOP):	1	0xAA
Delivery Flags:	1	0x07
App Data Type:	1	0x00
Node Address:	2	<Node Address>
Reply Length:	1	<len of response including command MSB and LSB>
Command MSB	1	0x00
Command LSB	1	<echo of the command>
Reply Data	1 - 100	<data of the command>
LQI	1	<Link quality>
RSSI	1	<Relative signal energy>
Checksum:	2	<Sum of all preceding bytes except SOP, LQI, RSSI>

Note that the items that are different from the command packet are the delivery flags, the command length, the reply data, and the LQI/RSSI bytes. In both the command and data packets, the combination of the command and reply data (the non-shaded items) are in the exact same format as with the non-packetized commands and replies. It is only the header fields (SOP, Delivery Flags, App Data Type, Node Address, and the length byte) and the footer fields (LQI, RSSI, and Checksum) that are different.

Example

An example of a standard 3DM-GX2™ Data Communications Protocol request for acceleration and angular rate (0xC2) looks like this:

PC sends command: **0xC2**

Sensor sends reply: **0xC2**, <28 bytes of data>, <2 byte checksum>

A packetized request for acceleration and angular rate data looks like this:

PC sends command: **0xAA, 0x0B, 0x00**, <2 byte node add.>, **2**, **0**, **0xC2**, <2 byte checksum>

Base station sends ACK: **0xAA**

Sensor sends reply: **0xAA, 0x07, 0x00**, <2 byte node add.>, **30**, **0**, **0xC1**, <28 bytes of data>, <LQI byte>, <RSSI byte>, <2 byte checksum>

A packetized request for acceleration and angular rate data that is rejected because of a packet error looks like this:

PC sends command: **0xAA, 0x0B, 0x00, <2 byte node add.>, 2, 0, 0xC2, <incorrect 2 byte checksum>**

Sensor sends NACK: **0x21**

Note that the ACK byte and the SOP byte (Start Of Packet) are the same value (0xAA). In other words, when sending a wireless packet, the host computer will see two 0xAA's as the first two bytes of the reply – the ACK and the SOP. Because the ACK byte is generated by the wireless basestation, you will not see it when communicating with non-wireless devices. With non-wireless devices (USB, RS-232 for example) you will only see a single 0xAA byte (the SOP byte) at the start of a reply.

If you are writing code that will communicate with both wireless and non-wireless devices using the packet protocol, it is advised that you structure your code. If the sensor is running in the polled command mode, the ACK/NACK byte should be read immediately after the command is sent. This way, it will not be accidentally interpreted as a packet SOP byte in the code that is reading data packets.

Command Reference

All commands are one byte in length. Several commands require that the host transmit additional data bytes following the command byte to fully define the action to be taken. All commands generate a response of a fixed number of bytes.

The response to most commands begins with a header byte (which has the same value as the corresponding command byte), and ends with a 16 bit checksum. The intervening bytes contain the data requested.

The 16 bit checksum is equal to the sum of all preceding bytes with rollover from 65535 to 0.

All multi-byte quantities are transmitted in big endian order (MSB first, LSB last). Floating point values are 32 bit, and conform to the IEEE-754 format.

Wireless Ping (wireless units only) (0x02)

Function:	The Inertia-Link® or 3DM-GX2™ will echo an 0x02 if the wireless communications is setup and working properly, otherwise the base station will return a 0x21
Command Byte:	0x02
Command Data:	2 bytes defined as follows
Byte 1	MSB of node address
Byte 2	LSB of node address
Response:	1 byte defined as follows
Byte 1	0x02 if wireless comm is OK. 0x21 if wireless comm is not OK

This command verifies that the communications channel is setup and working properly. It does not test the functionality of the Inertia-Link or 3DM-GX2. It is possible that the ping command could return an 0x02 but the device will not respond to any other commands. This would indicate that there is something wrong with the sensor even though the radio communications are OK. If the response is an 0x21, then there is a problem with the node address, the channel setting, or the radio signal (out of range or radio interference).

Raw Accelerometer and Angular Rate Sensor Outputs (0xC1)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the raw sensor voltages in the form of A/D converter codes (0 to 65535).
Command Byte:	0xC1
Command Data:	None
Response:	31 bytes defined as follows
Byte 1	<i>Header</i> = 0xC1
Bytes 2-5	<i>RawAccel</i> ₁
Bytes 6-9	<i>RawAccel</i> ₂
Bytes 10-13	<i>RawAccel</i> ₃
Bytes 14-17	<i>RawAngRate</i> ₁
Bytes 18-21	<i>RawAngRate</i> ₂
Bytes 22-25	<i>RawAngRate</i> ₃
Bytes 26-29	<i>Timer</i>
Bytes 30-31	<i>Checksum</i>

Acceleration & Angular Rate (0xC2)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the Acceleration and Angular Rate Vectors
Command Byte:	0xC2
Command Data:	None
Response:	31 bytes defined as follows
Byte 1	<i>Header</i> = 0xC2
Bytes 2-5	<i>Accel</i> _x
Bytes 6-9	<i>Accel</i> _y
Bytes 10-13	<i>Accel</i> _z
Bytes 14-17	<i>AngRate</i> _x
Bytes 18-21	<i>AngRate</i> _y
Bytes 22-25	<i>AngRate</i> _z
Bytes 26-29	<i>Timer</i>
Bytes 30-31	<i>Checksum</i>

DeltaAngle & DeltaVelocity (0xC3)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the DeltaAngle and DeltaVelocity Vectors
Command Byte:	0xC3
Command Data:	None
Response:	31 bytes defined as follows
Byte 1	<i>Header = 0xC3</i>
Bytes 2-5	<i>DeltaAng_x</i>
Bytes 6-9	<i>DeltaAng_y</i>
Bytes 10-13	<i>DeltaAng_z</i>
Bytes 14-17	<i>DeltaVel_x</i>
Bytes 18-21	<i>DeltaVel_y</i>
Bytes 22-25	<i>DeltaVel_z</i>
Bytes 26-29	<i>Timer</i>
Bytes 30-31	<i>Checksum</i>

Set Continuous Mode (0xC4)

Function:	The Inertia-Link® or 3DM-GX2™ will begin continuous output mode. The “Continuous Command Byte” determines which data record is generated. This should be set equal to the Command Byte corresponding to the desired data record. To stop continuous mode, set the Continuous Command Byte to 0; Note: The continuous mode set using this command will remain in effect until the device is de-powered. On subsequent power-up, the device will revert to its original state. To store a continuous mode setting that will persist through power-cycles, the Continuous Command Byte should be written to EEROM at address 0xFCA6. Note that for wireless devices a 0x80 should be written into the MSB of EEPROM address 0xFCA6. All other devices should write a 0x00 in the MSB.
Command Byte:	0xC4
Command Data:	3 bytes defined as follows
Byte 1	0xC1 (Confirms user intent)
Byte 2	0x29 (Confirms user intent)
Byte 3	Continuous Command Byte
Response:	8 bytes defined as follows
Byte 1	<i>Header</i> = 0xC4
Byte 2	Continuous Command Byte
Bytes 3-6	<i>Timer</i>
Bytes 7-8	<i>Checksum</i>

Orientation Matrix (0xC5)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing its orientation matrix, M .
Command Byte:	0xC5
Command Data:	None
Response:	43 bytes defined as follows
Byte 1	<i>Header</i> = 0xC5
Bytes 2-5	$M_{1,1}$
Bytes 6-9	$M_{1,2}$
Bytes 10-13	$M_{1,3}$
Bytes 14-17	$M_{2,1}$
Bytes 18-21	$M_{2,2}$
Bytes 22-25	$M_{2,3}$
Bytes 26-29	$M_{3,1}$
Bytes 30-33	$M_{3,2}$
Bytes 34-37	$M_{3,3}$
Bytes 38-41	<i>Timer</i>
Bytes 42-43	<i>Checksum</i>

Orientation Update Matrix (0xC6)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the orientation update matrix corresponding to the most recent calculation cycle.
Command Byte:	0xC6
Command Data:	None
Response:	43 bytes defined as follows
Byte 1	<i>Header</i> = 0xC6
Bytes 2-5	$C_{1,1}$
Bytes 6-9	$C_{1,2}$
Bytes 10-13	$C_{1,3}$
Bytes 14-17	$C_{2,1}$
Bytes 18-21	$C_{2,2}$
Bytes 22-25	$C_{2,3}$
Bytes 26-29	$C_{3,1}$
Bytes 30-33	$C_{3,2}$
Bytes 34-37	$C_{3,3}$
Bytes 38-41	<i>Timer</i>
Bytes 42-43	<i>Checksum</i>

Scaled Magnetometer Vector (0xC7)*version 2.1.00 or greater*

Function:	The 3DM-GX2™ will output a data record containing the magnetometer vector. The Inertia-Link® (which has no magnetometer) outputs NaNs for the vector components.
Command Byte:	0xC7
Command Data:	None
Response:	19 bytes defined as follows
Byte 1	<i>Header</i> = 0xC7
Bytes 2-5	<i>Mag_x</i>
Bytes 6-9	<i>Mag_y</i>
Bytes 10-13	<i>Mag_z</i>
Bytes 14-17	<i>Timer</i>
Bytes 18-19	<i>Checksum</i>

Acceleration, Angular Rate & Orientation Matrix (0xC8)

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the Acceleration and Angular Rate Vectors and the Orientation Matrix
Command Byte:	0xC8
Command Data:	None
Response:	67 bytes defined as follows
Byte 1	<i>Header</i> = 0xC8
Bytes 2-5	<i>Accel_x</i>
Bytes 6-9	<i>Accel_y</i>
Bytes 10-13	<i>Accel_z</i>
Bytes 14-17	<i>AngRate_x</i>
Bytes 18-21	<i>AngRate_y</i>
Bytes 22-25	<i>AngRate_z</i>
Bytes 26-29	<i>M_{1,1}</i>
Bytes 30-33	<i>M_{1,2}</i>
Bytes 34-37	<i>M_{1,3}</i>
Bytes 38-41	<i>M_{2,1}</i>
Bytes 42-45	<i>M_{2,2}</i>
Bytes 46-49	<i>M_{2,3}</i>
Bytes 50-53	<i>M_{3,1}</i>
Bytes 54-57	<i>M_{3,2}</i>
Bytes 58-61	<i>M_{3,3}</i>
Bytes 62-65	<i>Timer</i>
Bytes 66-67	<i>Checksum</i>

Write Accelerometer Bias Correction (0xC9)**version 2.1.00 or greater**

Function:	The specified <i>AccelBias</i> vector will be written to the Inertia-Link® or 3DM-GX2™. This vector will be subtracted from all subsequent measurements of <i>Accel</i> .
Command Byte:	0xC9
Command Data:	14 bytes defined as follows
Byte 1	0xB7
Byte 2	0x44
Bytes 3-6	<i>AccelBias_x</i>
Bytes 7-10	<i>AccelBias_y</i>
Bytes 11-14	<i>AccelBias_z</i>
Response:	19 bytes defined as follows
Byte 1	<i>Header</i> = 0xC9
Bytes 2-5	<i>AccelBias_x</i> (echo of input)
Bytes 6-9	<i>AccelBias_y</i> (echo of input)
Bytes 10-13	<i>AccelBias_z</i> (echo of input)
Bytes 14-17	<i>Timer</i>
Bytes 18-19	<i>Checksum</i>

Write Gyro Bias Correction (0xCA)**version 2.1.00 or greater**

Function:	The specified <i>GyroBias</i> vector will be written to the Inertia-Link® or 3DM-GX2™. This vector will be subtracted from all subsequent measurements of <i>AngRate</i> .
Command Byte:	0xCA
Command Data:	14 bytes defined as follows
Byte 1	0x12
Byte 2	0xA5
Bytes 3-6	<i>GyroBias_x</i>
Bytes 7-10	<i>GyroBias_y</i>
Bytes 11-14	<i>GyroBias_z</i>
Response:	19 bytes defined as follows
Byte 1	<i>Header</i> = 0xCA
Bytes 2-5	<i>GyroBias_x</i>
Bytes 6-9	<i>GyroBias_y</i>
Bytes 10-13	<i>GyroBias_z</i>
Bytes 14-17	<i>Timer</i>
Bytes 18-19	<i>Checksum</i>

Acceleration, Angular Rate & Magnetometer Vector (0xCB)*version 2.1.00 or greater*

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the acceleration, angular rate, and magnetometer vectors. On the Inertia-Link, magnetometer values will be NaN.
Command Byte:	0xCB
Command Data:	None
Response:	43 bytes defined as follows
Byte 1	<i>Header</i> = 0xCB
Bytes 2-5	<i>Accel_x</i>
Bytes 6-9	<i>Accel_y</i>
Bytes 10-13	<i>Accel_z</i>
Bytes 14-17	<i>AngRate_x</i>
Bytes 18-21	<i>AngRate_y</i>
Bytes 22-25	<i>AngRate_z</i>
Bytes 26-29	<i>Mag_x</i>
Bytes 30-33	<i>Mag_y</i>
Bytes 34-37	<i>Mag_z</i>
Bytes 38-41	<i>Timer</i>
Bytes 42-43	<i>Checksum</i>

Acceleration, Angular Rate & Magnetometer Vectors & Orientation Matrix (0xCC)

version 2.1.00 or greater

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the acceleration, angular rate and magnetometer vectors and the orientation matrix. On the Inertia-Link, magnetometer values will be NaN.
Command Byte:	0xCC
Command Data:	None
Response:	79 bytes defined as follows
Byte 1	<i>Header</i> = 0xCC
Bytes 2-5	<i>Accel_X</i>
Bytes 6-9	<i>Accel_Y</i>
Bytes 10-13	<i>Accel_Z</i>
Bytes 14-17	<i>AngRate_X</i>
Bytes 18-21	<i>AngRate_Y</i>
Bytes 22-25	<i>AngRate_Z</i>
Bytes 26-29	<i>Mag_X</i>
Bytes 30-33	<i>Mag_Y</i>
Bytes 34-37	<i>Mag_Z</i>
Bytes 38-41	<i>M_{1,1}</i>
Bytes 42-45	<i>M_{1,2}</i>
Bytes 46-49	<i>M_{1,3}</i>
Bytes 50-53	<i>M_{2,1}</i>
Bytes 54-57	<i>M_{2,2}</i>
Bytes 58-61	<i>M_{2,3}</i>
Bytes 62-65	<i>M_{3,1}</i>
Bytes 66-69	<i>M_{3,2}</i>
Bytes 70-73	<i>M_{3,3}</i>
Bytes 74-77	<i>Timer</i>
Bytes 78-79	<i>Checksum</i>

Capture Gyro Bias (0xCD)*version 2.1.00 or greater*

Function:	This command will cause the Inertia-Link® or 3DM-GX2™ to sample its sensors for the specified number of milliseconds. The resulting data will be used to initialize its orientation, and to estimate its gyro bias error. The estimated gyro bias error will be automatically written to the <i>GyroBias</i> vector. Notes: The Inertia-Link® or 3DM-GX2™ must be stationary during the execution of the Capture Gyro Bias Operation. The input parameter <i>SamplingTime</i> specifies the duration for which the sensors should be sampled and averaged. It is a 16 bit unsigned integer whose value gives the time in units of milliseconds. Recommended values are 10000 to 30000. The output, <i>GyroBias</i>, reflects the values automatically written into that vector as a result of the Capture Gyro Bias Operation.
Command Byte:	0xCD
Command Data:	4 bytes defined as follows
Byte 1	0xC1
Byte 2	0x29
Bytes 3-4	<i>SamplingTime</i>
Response:	19 bytes defined as follows
Byte 1	<i>Header</i> = 0xCD
Bytes 2-5	<i>GyroBias_x</i>
Bytes 6-9	<i>GyroBias_y</i>
Bytes 10-13	<i>GyroBias_z</i>
Bytes 14-17	<i>Timer</i>
Bytes 18-19	<i>Checksum</i>

Euler Angles (0xCE)**version 2.1.00 or greater**

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing its orientation in the form of Euler Angles
Command Byte:	0xCE
Command Data:	None
Response:	19 bytes defined as follows
Byte 1	<i>Header</i> = 0xCE
Bytes 2-5	<i>Roll</i>
Bytes 6-9	<i>Pitch</i>
Bytes 10-13	<i>Yaw</i>
Bytes 14-17	<i>Timer</i>
Bytes 18-19	<i>Checksum</i>

Euler Angles and Angular Rates (0xCF)**version 2.1.05 or greater**

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing its orientation in the form of Euler Angles, and the Angular Rates
Command Byte:	0xCF
Command Data:	None
Response:	31 bytes defined as follows
Byte 1	<i>Header</i> = 0xCF
Bytes 2-5	<i>Roll</i>
Bytes 6-9	<i>Pitch</i>
Bytes 10-13	<i>Yaw</i>
Bytes 14-17	<i>AngRate_x</i>
Bytes 18-21	<i>AngRate_y</i>
Bytes 22-25	<i>AngRate_z</i>
Bytes 26-29	<i>Timer</i>
Bytes 30-31	<i>Checksum</i>

Transfer Quantity to Non-Volatile Memory (0xD0)**version 2.1.12 or greater**

Function:	The current value of the Accelerometer Bias Correction, or the Gyro Bias Correction will be transferred to Non-volatile memory. When the system is subsequently powered on, it will read the non-volatile memory and load the corresponding values into RAM for immediate use. The input parameter “Transfer Quantity” is a 16 bit unsigned
-----------	---

	integer. Its value determines which data quantity is transferred to non-volatile memory <table> <tr> <th><i>Transfer Quantity</i></th><th>Quantity to be Transferred</th></tr> <tr> <td>1</td><td>Accelerometer Bias Correction</td></tr> <tr> <td>2</td><td>Gyro Bias Correction</td></tr> </table> All other values for <i>Transfer Quantity</i> will result in no action being taken. In such cases, the returned value of <i>Transfer Quantity</i> will be 0xFFFF.	<i>Transfer Quantity</i>	Quantity to be Transferred	1	Accelerometer Bias Correction	2	Gyro Bias Correction
<i>Transfer Quantity</i>	Quantity to be Transferred						
1	Accelerometer Bias Correction						
2	Gyro Bias Correction						
Command Byte:	0xD0						
Command Data:	4 Bytes defined as follows						
Byte 1	0xC1						
Byte 2	0x29						
Bytes 3-4	<i>Transfer Quantity</i> (16 bit unsigned integer)						
Response:	9 bytes defined as follows						
Byte 1	<i>Header</i> = 0xD0						
Bytes 2-3	<i>Transfer Quantity</i>						
Bytes 4-7	<i>Timer</i>						
Bytes 8-9	<i>Checksum</i>						

Temperatures (0xD1)

version 2.1.12 or greater

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing its 4 internal temperature measurements
Command Byte:	0xD1
Command Data:	None
Response:	15 bytes defined as follows
Byte 1	<i>Header</i> = 0xD1
Bytes 2-3	<i>Temperature</i> _{Accel}
Bytes 4-5	<i>Temperature</i> _{GyroX}
Bytes 6-7	<i>Temperature</i> _{GyroY}
Bytes 8-9	<i>Temperature</i> _{GyroZ}
Bytes 10-13	<i>Timer</i>
Bytes 14-15	<i>Checksum</i>

Gyro Stabilized Acceleration, Angular Rate & Magnetometer Vector (0xD2)

version 2.1.12 or greater

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the gyro-stabilized acceleration, angular rate, and magnetometer vectors. On the Inertia-Link, magnetometer
-----------	--

	values will be NaN.
Command Byte:	0xD2
Command Data:	None
Response:	43 bytes defined as follows
Byte 1	<i>Header</i> = 0xD2
Bytes 2-5	<i>StabAccel_x</i>
Bytes 6-9	<i>StabAccel_y</i>
Bytes 10-13	<i>StabAccel_z</i>
Bytes 14-17	<i>AngRate_x</i>
Bytes 18-21	<i>AngRate_y</i>
Bytes 22-25	<i>AngRate_z</i>
Bytes 26-29	<i>StabMag_x</i>
Bytes 30-33	<i>StabMag_y</i>
Bytes 34-37	<i>StabMag_z</i>
Bytes 38-41	<i>Timer</i>
Bytes 42-43	<i>Checksum</i>

DeltaAngle & DeltaVelocity & Magnetometer Vectors (0xD3)

version 2.1.13 or greater

Function:	The Inertia-Link® or 3DM-GX2™ will output a data record containing the DeltaAngle , DeltaVelocity and Magnetic Field Vectors . An Inertia-Link (which does not have magnetometers) will return NaN's for the magnetometer vector.
Command Byte:	0xD3
Command Data:	None
Response:	43 bytes defined as follows
Byte 1	<i>Header</i> = 0xD3
Bytes 2-5	<i>DeltaAng_x</i>
Bytes 6-9	<i>DeltaAng_y</i>
Bytes 10-13	<i>DeltaAng_z</i>
Bytes 14-17	<i>DeltaVel_x</i>
Bytes 18-21	<i>DeltaVel_y</i>
Bytes 22-25	<i>DeltaVel_z</i>
Bytes 26-29	<i>Mag_x</i>
Bytes 30-33	<i>Mag_y</i>
Bytes 34-37	<i>Mag_z</i>
Bytes 38-41	<i>Timer</i>
Bytes 42-43	<i>Checksum</i>

Write Word to EEPROM (0xE4)

Function:	The specified word will be written to EEPROM of the Inertia-Link® or 3DM-GX2™ at the specified address.
Command Byte:	0xE4
Command Data:	7 bytes defined as follows
Byte 1	0xC1
Byte 2	0x29
Byte 3	0x00
Byte 4-5	16 bit address of EEPROM word to be written to
Byte 6-7	Value of 16 bit word to be written to EEPROM
Response:	5 bytes defined as follows
Byte 1	<i>Header</i> = 0xE4
Bytes 2-3	EEPROM word (16 bit)
Bytes 4-5	<i>Checksum</i>

Read Word from EEPROM (0xE5)

Function:	The Inertia-Link® or 3DM-GX2™ will output the value of the 16bit word stored in EEPROM at the specified address
Command Byte:	0xE5
Command Data:	3 bytes defined as follows
Byte 1	0x00
Byte 2-3	16 bit address of EEPROM word to be read
Response:	5 bytes defined as follows
Byte 1	<i>Header</i> = 0xE5
Bytes 2-3	EEPROM word (16 bit)
Bytes 4-5	<i>Checksum</i>

Stop Continuous Mode (no reply) (0xFA)*version 2.1.00 or greater*

Function:	This single byte command will stop the continuous mode of the Inertia-Link® or 3DM-GX2™ without generating a reply. If not in continuous mode, it does nothing.
Command Byte:	0xFA
Command Data:	None

Calculation Cycle, and Data Output Rate

The on-board processor of the Inertia-Link® or 3DM-GX2™ continuously executes a calculation cycle. The steps in this cycle include the following:

1. Convert raw sensor outputs into digital form
2. Scale sensor outputs into physical units (including temperature, alignment, and G-sensitivity compensation).
3. Propagate and filter the orientation estimate.
4. If host has issued a command byte (or if operating in continuous mode), compute appropriate response data and transmit.

Step 4 in this cycle is only executed if the Inertia-Link® or 3DM-GX2™ has received a command byte from the host (or if the device is in continuous mode).

The calculation cycle continuously repeats itself (even if no data is requested by the host). The time required to complete a calculation cycle determines the fundamental limit on the maximum data output rate. Generally, only one new data record can be generated and output for each calculation cycle.

The duration of each calculation cycle is determined by the internal clock in combination with an adjustable divider

$$\text{CalcPeriod} = \frac{x}{51200}$$

Where

CalcPeriod = duration of the calculation cycle in seconds
 x = divider constant

The maximum Data Output Rate is the inverse of CalcPeriod

$$\text{DataRate} = \frac{51200}{x}$$

Where

DataRate = maximum data output rate in Hertz
 x = divider constant

DataRate quantifies the rate at which data records will be generated when running in continuous mode. In polled mode, this rate can also be achieved if the host program is capable of sending requests for data at a rate equal to or greater than the DataRate.

The factory default value of x is 512 giving a default data rate of 100Hz. x is stored as a 16 bit unsigned integer value in EEPROM address 0xFCA2. The value of x can be changed by utilizing the Write Word to EEPROM command. Valid values are within the range of 51200 (1Hz) to 170 (~300Hz).

Data Quantities Available

The Inertia-Link® and 3DM-GX2™ are capable of calculating and reporting data of various types. These can be accessed by selecting and sending the appropriate command byte (see [Command Set Summary](#) and [Command Reference](#) sections). The data that is available is the following:

RawAccel – (3 components)

These are the raw voltage outputs of the three axis accelerometer. They are expressed in terms of A/D converter codes where 0 represents 0.0 Volts, and 65535.0 represents 5 volts. They are not scaled into physical units, nor are the individual components necessarily orthogonal or aligned with the local coordinate system. These quantities are not recommended for most applications.

$$RawAccel = \begin{bmatrix} RawAccel_1 \\ RawAccel_2 \\ RawAccel_3 \end{bmatrix}$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Note: Although the A/D converter fundamentally produces integer quantities, floating point values are used to accommodate the Inertia-Link® and 3DM-GX2™'s ability to oversample and average successive readings.

RawAngRate – (3 components)

These are the raw voltage outputs of the three axis angular rate sensor. They are expressed in terms of A/D converter codes where 0 represents 0.0 Volts, and 65535.0 represents 5 volts. They are not scaled into physical units, nor are the individual components necessarily orthogonal, or forming a right-handed coordinate system. These quantities are not recommended for most applications.

$$RawAngRate = \begin{bmatrix} RawAngRate_1 \\ RawAngRate_2 \\ RawAngRate_3 \end{bmatrix}$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Note: Although the A/D converter fundamentally produces integer quantities, floating point values are used to accommodate the Inertia-Link® and 3DM-GX2's ability to oversample and average successive readings.

Accel – (X, Y, Z components)

This is a vector quantifying the direction and magnitude of the acceleration that the Inertia-Link® or 3DM-GX2™ is exposed to. This quantity is derived from RawAccel, but is fully temperature compensated and scaled into physical units of g ($1\ g = 9.80665\ \text{m/sec}^2$). It is expressed in terms of the Inertia-Link® or 3DM-GX2™'s local coordinate system.

$$\text{Accel} = \begin{bmatrix} \text{Accel}_x \\ \text{Accel}_y \\ \text{Accel}_z \end{bmatrix} (g)$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Note: The measured gravitational acceleration varies from the standard value by as much as +/-0.3% depending on geographic location. This is due to a combination of earth's non-spherical shape, centripetal acceleration from earth's rotation, and altitude. The standard value of $9.80665\ \text{m/sec}^2$ corresponds to sea-level locations at approximately 45.5 degrees latitude. A suitable model is the following:

$$g_l = 9.780318 \cdot \left[1 + 0.0053024 \cdot \sin^2(L) - 0.0000058 \cdot \sin^2(2L) \right] - 3.086 \cdot 10^{-6} h$$

where: g_l = locally measured acceleration of gravity (m/sec^2)
 L = latitude
 h = altitude above sea level in meters

For example, at a location of 10 degrees latitude and 1000 meters altitude, $g_l = 9.77879\ \text{m/sec}^2$. This is equal to $0.99716 \cdot g$. Therefore, under static conditions, the Accel vector output by the Inertia-Link® or 3DM-GX2™ would have a magnitude of $0.997g$.

StabAccel – (X, Y, Z components)

This is a vector which represents the complementary filter's best estimate of the vertical direction. Under stationary conditions, it should be equal to *Accel*. In dynamic conditions, *Accel* will be sensitive to both gravitational acceleration as well as linear acceleration. The Complementary filter computes *StabAccel* which is its estimate of the gravitation acceleration only, even though the system may be exposed to significant linear acceleration.

$$\text{StabAccel} = \begin{bmatrix} \text{StabAccel}_x \\ \text{StabAccel}_y \\ \text{StabAccel}_z \end{bmatrix} (g)$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

AngRate – (X, Y and Z components)

This is a vector quantifying the rate of rotation of the Inertia-Link® or 3DM-GX2™. This quantity is derived from the RawAngRate quantities, but is fully temperature compensated and scaled into units of radians/second. It is expressed in terms of the Inertia-Link® and 3DM-GX2™'s local coordinate system in units of radians/second.

$$AngRate = \begin{bmatrix} AngRate_x \\ AngRate_y \\ AngRate_z \end{bmatrix} \text{ (rad/sec)}$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

DeltaVel– (X, Y, Z components)

This is a vector which gives the time integral of *Accel* where the limits of integration are the beginning and end of the most recent sampling period (e.g., 0.01 seconds). It is expressed in terms of the Inertia-Link® and 3DM-GX2™'s local coordinate system in units of g*second where g is the standard gravitational constant. To convert *DeltaVel* into the more conventional units of m/sec, simply multiply by the standard gravitational constant, 9.80665 m/sec².

$$DeltaVel = \begin{bmatrix} DeltaVel_x \\ DeltaVel_y \\ DeltaVel_z \end{bmatrix} \text{ (g*sec)}$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

DeltaAng– (X, Y, Z components)

This is a vector which gives the time integral of *AngRate* where the limits of integration are the beginning and end of the most recent sampling period (eg., 0.01 seconds). It is expressed in terms of the Inertia-Link® and 3DM-GX2™'s local coordinate system in units of radians.

$$DeltaVel = \begin{bmatrix} DeltaVel_x \\ DeltaVel_y \\ DeltaVel_z \end{bmatrix} \text{ (rad)}$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Magnetometer– (X, Y, Z components)

This is a vector which gives the instantaneous magnetometer direction and magnitude. It is fully temperature compensated and is expressed in terms of the 3DM-GX2™'s local coordinate system in units of Gauss. The Inertia-Link® returns NaN for these values.

$$\text{Magnetometer} = \begin{bmatrix} \text{Mag}_x \\ \text{Mag}_y \\ \text{Mag}_z \end{bmatrix} (\text{G})$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

StabMag – (X, Y, Z components)

This is a vector which represents the complementary filter's best estimate of the geomagnetic field direction. In the absence of magnetic interference, it should be equal to *Magnetometer*. When transient magnetic interference is present, *Magnetometer* will be subject to transient (possibly large) errors. The Complementary filter computes *StabMag* which is its estimate of the geomagnetic field vector only, even though the system may be exposed to transient magnetic interference. Note that sustained magnetic interference cannot be adequately compensated for by the complementary filter.

$$\text{StabMag} = \begin{bmatrix} \text{StabMag}_x \\ \text{StabMag}_y \\ \text{StabMag}_z \end{bmatrix} (\text{G})$$

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Timer

This is a timer value which measures the time since system power-up. To convert the timer value to time in seconds, divide by 19,660,800. The system clock has an accuracy of +/- 0.01%.

Data Format: 32 bit unsigned integer value.

Note: The timer value rolls over from its maximum value to 0 approximately every 218 seconds. The host system must keep track of these rollovers if an absolute time is required over long periods.

M

This is a 9 component coordinate transformation matrix which describes the orientation of the Inertia-Link® or 3DM-GX2™ with respect to the fixed earth coordinate system.

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$$

M satisfies the following equation:

$$V_{ILi} = M_{ij} \cdot V_{Ej}$$

Where: V_{IL} is a vector expressed in the Inertia-Link® and 3DM-GX2™'s local coordinate system.

V_E is the same vector expressed in the stationary, earth-fixed coordinate system

Data Format: 9 element array of 32bit floating point values in IEEE-754 format.

*Note: The elements are arranged in Row major order:
($M_{1,1}$, $M_{1,2}$, $M_{1,3}$, $M_{2,1}$, ... , $M_{3,2}$, $M_{3,3}$).*

C

This is a 9 component coordinate transformation matrix which describes the change in orientation of the Inertia-Link® during the period of the most recent calculation cycle.

$$C = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,1} & C_{3,2} & C_{3,3} \end{bmatrix}$$

M satisfies the following equation:

$$M2_i = C_{ij} \cdot M1_{ij}$$

Where: $M1$ is the orientation matrix at the beginning of the calculation cycle.

$M2$ is the orientation matrix at the end of the calculation cycle.

Data Format: 9 element array of 32bit floating point values in IEEE-754 format.

Note: The elements are arranged in Row major order:

$$(C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, \dots, C_{3,2}, C_{3,3}).$$

Euler Angles

This is a 3 component vector containing the Roll, Pitch and Yaw angles in radians. It is computed on the Inertia-Link® from the orientation matrix M .

$$Euler = \begin{bmatrix} Roll \\ Pitch \\ Yaw \end{bmatrix} \text{ (radians)}$$

Data Format: 3 element vector of 32bit floating point values in IEEE-754 format.

AccelBias – (X, Y, Z components)

This is a 3 component vector quantity in units of g (same as the *Accel* vector). It is expressed in terms of the Inertia-Link® or 3DM-GX2™'s local coordinate system. On system power-up, *AccelBias* is initialized to a value of [0,0,0].

$$AccelBias = \begin{bmatrix} AccelBias_x \\ AccelBias_y \\ AccelBias_z \end{bmatrix} (g)$$

Due to imperfections in the accelerometer sensors, the measured *Accel* vector can be in error by an additive constant (bias). In some applications, it may be possible for the user's software algorithm to estimate this bias. If such an estimate is available, it can be written into the Inertia-Link®, in the form of the *AccelBias* vector. *AccelBias* will be subtracted from every measurement of the *Accel* vector made by the Inertia-Link® and 3DM-GX2™ to form a bias corrected version of the *Accel* quantity. This will then be used as the output quantity, and for all on-board filtering operations that use *Accel*. This provides a means by which bias errors in the accelerometer measurements can be compensated for.

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

GyroBias – (X, Y, Z components)

This is a 3 component vector quantity in units of rad/sec (same as the *AngRate* vector). It is expressed in terms of the Inertia-Link® or 3DM-GX2™'s local coordinate system. On system power-up, *GyroBias* is initialized to a value of [0,0,0].

$$\text{GyroBias} = \begin{bmatrix} \text{GyroBias}_x \\ \text{GyroBias}_y \\ \text{GyroBias}_z \end{bmatrix} (g)$$

Due to imperfections in the angular rate sensors, the measured *AngRate* vector can be in error by an additive constant (bias). In some applications, it may be possible for the user's software algorithm to estimate this bias. If such an estimate is available, it can be written into the Inertia-Link® and 3DM-GX2™, in the form of the *GyroBias* vector. *GyroBias* will be subtracted from every measurement of the *AngRate* vector made by the Inertia-Link® to form a bias corrected version of the *AngRate* quantity. This will then be used as the output quantity, and for all on-board filtering operations that use *AngRate*. This provides a means by which bias errors in the angular Rate measurements can be compensated for.

Data Format: 3 element array of 32bit floating point values in IEEE-754 format.

Temperature – (Accel, GyroX, GyroY, GyroZ components)

This is a 4 element array of 16 bit integers giving the raw A/D converter codes associated with each of the four on-board temperature sensors.

$$\text{Temperature} = \begin{bmatrix} \text{Temperature}_{\text{Accel}} \\ \text{Temperature}_{\text{GyroX}} \\ \text{Temperature}_{\text{GyroY}} \\ \text{Temperature}_{\text{GyroZ}} \end{bmatrix} (\text{A/D Converter Bits})$$

Temperature_{Accel} is derived from a PCboard mounted temperature sensor located near the accelerometer sensors. *Temperature_{GyroX}* is derived from the temperature sensor integral to the X gyro sensor. *Temperature_{GyroY}* and *Temperature_{GyroZ}* are derived from the temperature sensors integral to the Y and Z gyro sensors respectively. All temperature sensors are sampled using a 12 bit A/D converter with a 3.3V reference.

Temperature_{Accel} has the highest absolute accuracy. To convert to physical units, carry out the following conversion:

$$\text{Temperature}(\text{deg C}) = \left(\text{Temperature}_{\text{Accel}} \cdot \frac{3.3}{4096} - 0.5 \right) \cdot 100$$

The Gyro derived temperature outputs are optimized for repeatability rather than absolute accuracy. The conversion to degrees C depends on the model of gyro used.

For the ADXRS300, ADXRS150, ADXRS401 use the following:

$$Temperature(\text{deg C}) = \left(Temperature_{Accel} \cdot \frac{3.3}{4096} \cdot \frac{1}{0.0084} \right) + 273$$

For the ADXRS610, ADXRS612, ADXRS614 use the following:

$$Temperature(\text{deg C}) = \left(Temperature_{Accel} \cdot \frac{3.3}{4096} - 2.5 \right) \cdot \frac{1}{0.009} + 25$$

Orientation Conversion Formulas

The Inertia-Link® and 3DM-GX2™ can output orientation information in two different forms: Euler Angles or a 3x3 rotation matrix (also called a coordinate transformation matrix). These are essentially equivalent except that the Euler Angles have a mathematical singularity whenever Pitch is +/-90 degrees, and are therefore unsuitable for use under conditions where such orientations are likely to occur.

This document outlines how the two different means of expressing orientation are related to one-another.

The Inertia-Link® and 3DM-GX2™ fundamentally calculate orientation in the form of a rotation matrix, M .

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

M satisfies the vector equation,

$$VL = M \cdot VE \quad \text{where: } VE \text{ is a vector expressed in the Earth-Fixed coordinate system.}$$

VL is the same vector expressed in the Inertia-Link® or 3DM-GX2™ sensor's local coordinate system.

When the user requests orientation in the form of Euler Angles these are derived from the rotation matrix. Euler Angles consist of the *Pitch*, *Roll* and *Yaw* angles (or equivalently, the *Elevation*, *Bank*, and *Heading*). These are calculated using the “Aircraft” or “ZYX” formulation.

$$Pitch = \theta = \arcsin(-M_{13})$$

$$Roll = \phi = \arctan(M_{23} / M_{33})$$

$$Yaw = \psi = \arctan(M_{12} / M_{11})$$

Note: when computing the arc tan of a fraction, the possibility of quadrant ambiguity and division by zero problems occurs. Many programming languages include a function, typically called “atan2”, in which the numerator and denominator of the argument are input separately. This function then correctly returns the result under all conditions. The atan2 function should be used whenever possible.

The rotation matrix corresponding to a given set of Euler angles can be calculated using:

$$M = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \cos(\theta)\sin(\phi) \\ \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}$$

where $Pitch = \theta, Roll = \phi, Yaw = \psi$