DCP Manual Version 1.08

3DM-GX3® Data Communications Protocol

Firmware Versions 0.4.14, 1.1.27 and higher

MicroStrain® Little Sensors, Big Ideas®

www.microstrain.com

459 Hurricane Lane
Williston, VT 05495
United States of America
Phone: 802-862-6629
Fax: 802-863-4093
www.microstrain.com
support@microstrain.com

REVISED: May 26, 2010

Contents

Overview

The 3DM-GX3® utilizes MicroStrain's fourth generation orientation sensor core and offers more performance and features than previous models including:

- 32-bit low power high performance processor
- USB and serial UART host interface
- 17 bit resolution on Gyros, Accelerometers, and Magnetometers
- Full 1kHz Coning and Sculling integration
- Cascaded adjustable FIR filters
- Data rates of up to 1000Hz
- Smallest and lightest full function 6-DOF orientation sensor
- Fully calibrated and temperature compensated outputs

This document describes how to communicate with the MicroStrain 3DM-GX3® Attitude Heading Reference System (AHRS). The communications interface is compatible with our Inertia-Link® and 3DM-GX2® devices but has expanded functionality.

This version of the Data Communications Protocol includes documentation of the new features supported starting with version 1.1.27 of the firmware. If you are upgrading from 0.4.14 of the firmware review the important additions and changes flagged with asterisks in the Command Set Summary and highlighted in green in the document.

Introduction to the MicroStrain 3DM-GX3® Data Communications Protocol

The 3DM-GX3® 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 (echo of the command byte) and a checksum to do simple data integrity checks.

The standard serial protocol is described below in Communications Commands.

Serial Port Configuration

All communications with 3DM-GX3® sensors are accomplished via a real (RS-232 or UART) or virtual (USB) serial port.

The default serial port settings for all devices are shown in Table 1 below:

COM Port Default Serial Settings	
Baud Rate	**115.2K**
Parity:	**None**
Data Bits:	**8**
Stop Bits:	**1**
RTS/CTS:	**Disabled**

Table 1

The baud rate may be changed as described in the Communications Settings section. The BAUD rate setting has no effect on USB communications.

Communications Commands

The host computer controls what data the 3DM-GX3® will output by issuing one or more single byte commands (in some cases, additional data bytes must follow the command byte). Most commands will cause the 3DM-GX3® to transmit a response of a fixed number of bytes.

The device can also easily be put into Continuous Mode which will continuously output a pre-selected data quantity without being prompted by the host. The device can also be programmed to power-on directly into Continuous Mode without requiring any host control.

Table 2 lists a summary of all commands available for the 3DM-GX3®. Each of the commands is described in detail in the Command Reference section.

Command Set Summary

Command	Definition
0xC1	Raw Accelerometer and Angular Rate Sensor Outputs
0xC2	Acceleration & Angular Rate
0xC3	DeltaAngle & DeltaVelocity
0xC4	Set Continuous Mode
0xC5	Orientation Matrix
0xC6	Orientation Update Matrix
0xC7	Magnetometer Vector
0xC8	Acceleration, Angular Rate & Orientation Matrix
0xC9	Write Accel Bias Correction
0xCA	Write Gyro Bias Correction
0xCB	Acceleration, Angular Rate & Magnetometer Vector
0xCC	Acceleration, Angular Rate & Magnetometer 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
0xD3	DeltaAngle & DeltaVelocity & Magnetometer Vectors
0xD4	Mode
0xD5	Mode Preset
0xD6	Continuous Preset
0xD7	Timer
0xD9	Communications Settings
0xDA	Stationary Test
0xDB	Sampling Settings*
0xDD	Realign Up and North*
0xDF	Quaternion*
0xE4	Write Word to EEPROM
0xE5	Read Word from EEPROM
0xE9	Read Firmware Version Number
0xEA	Read Device ID String
0xFA	Stop Continuous Mode (no reply)
0xFD	Firmware Update (no reply)
0xFE	Device Reset (no reply)

Table 2

These features are new or have new elements to them starting with firmware version 1.1.27

Device States and Modes

The 3DM-GX3® operates in one of two states: **Awake** or **Sleep**. In the awake state, there are three possible modes of operation **Continuous**, **Active**, or **Idle**. In the sleep state there are two possible modes: **Sleep**, **Deep Sleep**. Note that the Sleep and Deep Sleep modes are only available when the unit is connected to the host via the Serial port (LVTTL UART on the OEM version). The Sleep state is not available when connected via USB.

The 3DM-GX3® can transition from the sleep state to the awake state in one of two ways:

1. Power On
2. Sending any character to the serial port.

When the 3DM-GX3® goes into the awake state, it will go into the mode pre-selected by a Mode Preset command.

The 3DM-GX3® can transition from the awake state to the sleep state via a UART Mode command.

Active Mode

In active mode, the 3DM-GX3® has all sensors powered on and is performing continuous sampling and data conditioning. The communications channel is open and the 3DM-GX3® can respond to any configuration, status, or data command. The host may issue any command at any time. The sensor will not output unsolicited data records. The 3DM-GX3® will respond to data commands by outputting the corresponding data record. Multiple commands issued by the host will be buffered on-board the device, with one being processed at the completion of each successive sampling cycle.

Continuous Mode

In continuous mode, the 3DM-GX3® will output a data record continuously with no further action by the host. The data record output corresponds to the preset data command set by the last call to Continuous Preset. The host computer must be capable of buffering and interpreting the data stream at sufficient speed to prevent loss of data. 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 issuing a different Mode command or the power to the device is interrupted.

An alternate single byte command, Set Continuous Mode, can be used to start continuous mode. This command does not change the continuous preset value.

An alternate single byte termination command, Stop Continuous Mode, may be used to stop the continuous mode and put the sensor into active mode. The benefit of this command is that it does not generate a response packet. This can be advantageous where the introduction of a response packet in the middle of a data stream can cause a parser to get out of sync with the stream.

Although the 3DM-GX3® will still act on and respond to all other commands while in continuous mode, it is better to change to active or idle mode before doing configuration commands. This makes it easier for the host to parse the incoming stream of response packets.

***Note**: From the factory, the continuous preset value is 0 which results in no action when changing to continuous mode with the Mode command. The continuous preset value must be set to a valid data command for continuous mode to be activated.*

***Note**: When using the mode command to set the mode to continuous, the 3DM-GX3® will send the mode command reply packet before entering into continuous mode. In other words, you will receive the reply packet followed by the first data packet specified by the Continuous Preset command.*

Idle Mode

The idle mode is the same as the active mode in that the 3DM-GX3® will respond to any configuration, status, or data command. However, in the idle mode, the sensors are turned off and data commands return invalid sensor data. For all calculated floating point data, the values returned will be NaN. For raw data, the values will just be undetermined. This mode is useful when doing configuration on a battery powered system and you need to minimize the power consumption. Idle mode draws approximately one half of the current compared to active mode.

Sleep Mode (Serial Connection only)

The sleep mode is a lower power mode than idle mode. The difference is that the processor and clocks are turned off as well as the sensors.

Deep Sleep Mode (Soft Power Off – Serial connection only)

The deep sleep mode is the lowest power mode other than fully turning the device off. This mode may be used as an alternative to having a physical power off circuit at very low power cost.

Wake up and Settling Time

The 3DM-GX3® is put into a sleep state by calling the Mode command with one of the sleep modes as a parameter. When the 3DM-GX3® is awakened, it goes into one of the awake state modes depending on what was preset using the Mode Preset command.

The latency between a wakeup event (Power on or UART command) and the first valid data available from the 3DM-GX3® is largely determined by the sensor power settling time and filter initialization time. The approximate wake time to valid data (in seconds) can be computed by the formula:

$$T_{wake} = 0.053 + 2M/1000$$

Equation 1

where M is the filter width (see Filtering).

Summary of 3DM-GX3® States and Modes:

"Awake" state modes	*Method of entering mode*	*Device Power (@ 3.2VDC)*	*Functionality*
Continuous	Host sends "Continuous" command OR Wakeup mode is preset to "Continuous"	~270mW	Processor and Sensors on. Data is sent continuously at a pre-configured data rate (user programmable). Processor responsive to all commands.
Active (default)	Host sends "Active" command OR Wakeup mode is preset to "Active"	~270mW	Processor and Sensors on. Processor responsive to all commands. All data and configuration commands are valid in this mode.
Idle	Host sends "Idle" command OR Wakeup mode is preset to "Idle"	~120mW	Processor on, Sensors OFF. Processor responsive to all commands. All configuration commands are valid, data commands return invalid data

Table 3

"Sleep" state modes	*Method of entering mode*	*Device Power*	*Functionality*
Sleep	Host sends "Sleep" command.	~58mW	Processor in low power state. All sensors are powered OFF. Any char sent to the serial port will wakeup the device (does not have to be a valid command). On wakeup, the device goes directly into preset "Awake" mode (Continuous, Active, or Idle).
Deep Sleep (Soft Power Off)	Host sends "Deep Sleep" command.	~45mW	Processor is off. All sensors are powered OFF. Any char sent to the UART will wakeup the device (does not have to be a valid command). On wakeup, the device goes through restart and warmup process and then into preset "Awake" mode (Continuous, Active, or Idle).
Power Off	nEnable signal (OEM only)	~10uW	LDO regulators are disabled. 3DM-GX3® is physically turned off. On power-on, the device goes through restart and warmup process and then into preset "Awake" mode (Continuous, Active, or Idle).

Table 4

Command Reference

All commands, at minimum, consist of a command byte. Several commands require additional data bytes following the command byte to fully define the action to be taken. The number of bytes in a response varies from command to command, but the response for a given command is always a fixed length. There are no variable length responses.

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 (see Calculating The Checksum). It is important that the host software evaluate the checksum from all responses to prevent errors due to out-of-sync data streams.

Some commands that change settings or memory values contain a confirmation byte sequence as part of the command data. This is to prevent accidental changes in settings from spurious serial input. If the confirmation sequence is invalid, a 3 byte error reply will be sent instead of the expected data reply. This reply consists of the following bytes: 0x21 0x00 0x21. The first byte is the error command byte. The second and third bytes are the checksum, which is always 0x00 0x21.

On commands that change settings, range checking is employed and any value that is out of range will be brought into range. The new in-range value will be used for the setting and be returned in the reply. In cases where only a certain set of values are allowed, if the value is not part of the allowed set, it is ignored and the current value of the setting remains in force and is returned in the reply. Note that the reply always reflects the actual settings of the device.

All multi-byte quantities are transmitted in Big Endian order (MSB first, LSB last) by default. All data quantities are expressed as floating point values except where noted. Floating point values are 32 bit (4 bytes), and conform to the IEEE-754 format.

The data quantities returned for Temperature and certain values used in Write Word To EEPROM and Read Word From EEPROM use 16 bit unsigned integers constructed from 2 bytes. The first byte is referred to as the MSB (most significant byte) and the second byte is referred to as the LSB (least significant byte). Values are constructed using the following formula: (256 * MSB) + LSB = value.

Raw Accelerometer and Angular Rate Sensor Outputs (0xC1)

Function:	The 3DM-GX3® will output a data record containing the raw sensor voltage values in the range of 17 bit integer converter codes (0 to 131071). The value is conveyed in 32 bit IEEE-754 floating point format.	
Command:		
Byte 1	0xC1	
Response:		
Byte 1	0xC1	
Bytes 2-5	*$RawAccel_1$*	*(IEEE-754 Floating Point)*
Bytes 6-9	*$RawAccel_2$*	*(IEEE-754 Floating Point)*
Bytes 10-13	*$RawAccel_3$*	*(IEEE-754 Floating Point)*
Bytes 14-17	*$RawAngRate_1$*	*(IEEE-754 Floating Point)*
Bytes 18-21	*$RawAngRate_2$*	*(IEEE-754 Floating Point)*
Bytes 22-25	*$RawAngRate_3$*	*(IEEE-754 Floating Point)*
Bytes 26-29	*Timer*	
Bytes 30-31	*Checksum*	

Acceleration & Angular Rate (0xC2)

Function:	The 3DM-GX3® will output a data record containing the Acceleration and Angular Rate vectors.
Command:	
Byte 1	0xC2
Response:	
Byte 1	0xC2
Bytes 2-5	$Accel_X$ *(IEEE-754 Floating Point)*
Bytes 6-9	$Accel_Y$ *(IEEE-754 Floating Point)*
Bytes 10-13	$Accel_Z$ *(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$ *(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$ *(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$ *(IEEE-754 Floating Point)*
Bytes 26-29	*Timer*
Bytes 30-31	*Checksum*

DeltaAngle & DeltaVelocity (0xC3)

Function:	The 3DM-GX3® will output a data record containing the DeltaAngle and DeltaVelocity Vectors
Command:	
Byte 1	0xC3
Response:	
Byte 1	0xC3
Bytes 2-5	$DeltaAng_X$ *(IEEE-754 Floating Point)*
Bytes 6-9	$DeltaAng_Y$ *(IEEE-754 Floating Point)*
Bytes 10-13	$DeltaAng_Z$ *(IEEE-754 Floating Point)*
Bytes 14-17	$DeltaVel_X$ *(IEEE-754 Floating Point)*
Bytes 18-21	$DeltaVel_Y$ *(IEEE-754 Floating Point)*
Bytes 22-25	$DeltaVel_Z$ *(IEEE-754 Floating Point)*
Bytes 26-29	*Timer*
Bytes 30-31	*Checksum*

Set Continuous Mode (0xC4)

Function:	The 3DM-GX3® 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 or use the Stop Continuous Mode command. 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, see the Mode Preset and Continuous Preset commands.
Command:	
Byte 1	0xC4
Byte 2	0xC1 *(Confirms user intent)*
Byte 3	0x29 *(Confirms user intent)*
Byte 4	Continuous Command Byte *(8 bit unsigned integer)*
Response:	
Byte 1	0xC4
Byte 2	Continuous Command Byte
Bytes 3-6	*Timer*
Bytes 7-8	*Checksum*

Orientation Matrix (0xC5)

Function:	The 3DM-GX3® will output a data record containing the orientation matrix, M.	
Command:		
Byte 1	0xC5	
Response:		
Byte 1	0xC5	
Bytes 2-5	$M_{1,1}$	*(IEEE-754 Floating Point)*
Bytes 6-9	$M_{1,2}$	*(IEEE-754 Floating Point)*
Bytes 10-13	$M_{1,3}$	*(IEEE-754 Floating Point)*
Bytes 14-17	$M_{2,1}$	*(IEEE-754 Floating Point)*
Bytes 18-21	$M_{2,2}$	*(IEEE-754 Floating Point)*
Bytes 22-25	$M_{2,3}$	*(IEEE-754 Floating Point)*
Bytes 26-29	$M_{3,1}$	*(IEEE-754 Floating Point)*
Bytes 30-33	$M_{3,2}$	*(IEEE-754 Floating Point)*
Bytes 34-37	$M_{3,3}$	*(IEEE-754 Floating Point)*
Bytes 38-41	*Timer*	
Bytes 42-43	*Checksum*	

Orientation Update Matrix (0xC6)

Function:	The 3DM-GX3® will output a data record containing the orientation update matrix, C, corresponding to the most recent calculation cycle.	
Command:		
Byte 1	0xC6	
Response:		
Byte 1	0xC6	
Bytes 2-5	$C_{1,1}$	*(IEEE-754 Floating Point)*
Bytes 6-9	$C_{1,2}$	*(IEEE-754 Floating Point)*
Bytes 10-13	$C_{1,3}$	*(IEEE-754 Floating Point)*
Bytes 14-17	$C_{2,1}$	*(IEEE-754 Floating Point)*
Bytes 18-21	$C_{2,2}$	*(IEEE-754 Floating Point)*
Bytes 22-25	$C_{2,3}$	*(IEEE-754 Floating Point)*
Bytes 26-29	$C_{3,1}$	*(IEEE-754 Floating Point)*
Bytes 30-33	$C_{3,2}$	*(IEEE-754 Floating Point)*
Bytes 34-37	$C_{3,3}$	*(IEEE-754 Floating Point)*
Bytes 38-41	*Timer*	
Bytes 42-43	*Checksum*	

Scaled Magnetometer Vector (0xC7)

Function:	The 3DM-GX3® will output a data record containing the magnetometer vectors.
Command:	
Byte 1	0xC7
Response:	
Byte 1	0xC7
Bytes 2-5	*Mag*$_X$ *(IEEE-754 Floating Point)*
Bytes 6-9	*Mag*$_Y$ *(IEEE-754 Floating Point)*
Bytes 10-13	*Mag*$_Z$ *(IEEE-754 Floating Point)*
Bytes 14-17	*Timer*
Bytes 18-19	*Checksum*

Acceleration, Angular Rate & Orientation Matrix (0xC8)

Function:	The 3DM-GX3® will output a data record containing the Acceleration and Angular Rate Vectors and the Orientation Matrix.	
Command:		
Byte 1	0xC8	
Response:		
Byte 1	0xC8	
Bytes 2-5	$Accel_X$	*(IEEE-754 Floating Point)*
Bytes 6-9	$Accel_Y$	*(IEEE-754 Floating Point)*
Bytes 10-13	$Accel_Z$	*(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$	*(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$	*(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$	*(IEEE-754 Floating Point)*
Bytes 26-29	$M_{1,1}$	*(IEEE-754 Floating Point)*
Bytes 30-33	$M_{1,2}$	*(IEEE-754 Floating Point)*
Bytes 34-37	$M_{1,3}$	*(IEEE-754 Floating Point)*
Bytes 38-41	$M_{2,1}$	*(IEEE-754 Floating Point)*
Bytes 42-45	$M_{2,2}$	*(IEEE-754 Floating Point)*
Bytes 46-49	$M_{2,3}$	*(IEEE-754 Floating Point)*
Bytes 50-53	$M_{3,1}$	*(IEEE-754 Floating Point)*
Bytes 54-57	$M_{3,2}$	*(IEEE-754 Floating Point)*
Bytes 58-61	$M_{3,3}$	*(IEEE-754 Floating Point)*
Bytes 62-65	*Timer*	
Bytes 66-67	*Checksum*	

Write Accel Bias Correction (0xC9)

Function:	The specified AccelBias vector will be written to the 3DM-GX3®. This vector will be subtracted from all subsequent measurements of *Accel*.	
Command:		
Byte 1	0xC9	
Byte 2	0xB7 (Confirms user intent)	
Byte 3	0x44 (Confirms user intent)	
Bytes 4-7	$AccelBias_X$	*(IEEE-754 Floating Point)*
Bytes 8-11	$AccelBias_Y$	*(IEEE-754 Floating Point)*
Bytes 12-15	$AccelBias_Z$	*(IEEE-754 Floating Point)*
Response:		
Byte 1	0xC9	
Bytes 2-5	$AccelBias_X$	*(IEEE-754 Floating Point)*
Bytes 6-9	$AccelBias_Y$	*(IEEE-754 Floating Point)*
Bytes 10-13	$AccelBias_Z$	*(IEEE-754 Floating Point)*
Bytes 14-17	*Timer*	
Bytes 18-19	*Checksum*	

Write Gyro Bias Correction (0xCA)

Function:	The specified GyroBias vector will be written to the 3DM-GX3®. This vector will be subtracted from all subsequent measurements of *AngRate*.	
Command:		
Byte 1	0xCA	
Byte 2	0x12 (Confirms user intent)	
Byte 3	0xA5 (Confirms user intent)	
Bytes 4-7	$GyroBias_X$	*(IEEE-754 Floating Point)*
Bytes 8-11	$GyroBias_Y$	*(IEEE-754 Floating Point)*
Bytes 12-15	$GyroBias_Z$	*(IEEE-754 Floating Point)*
Response:		
Byte 1	0xCA	
Bytes 2-5	$GyroBias_X$	*(IEEE-754 Floating Point)*
Bytes 6-9	$GyroBias_Y$	*(IEEE-754 Floating Point)*
Bytes 10-13	$GyroBias_Z$	*(IEEE-754 Floating Point)*
Bytes 14-17	*Timer*	
Bytes 18-19	*Checksum*	

Acceleration, Angular Rate & Magnetometer Vector (0xCB)

Function:	The 3DM-GX3® will output a data record containing the acceleration, angular rate, and magnetometer vectors. If the magnetometer is not turned on or not present, the Mag vector values will be *NaN*.	
Command:		
Byte 1	0xCB	
Response:		
Byte 1	0xCB	
Bytes 2-5	$Accel_X$	*(IEEE-754 Floating Point)*
Bytes 6-9	$Accel_Y$	*(IEEE-754 Floating Point)*
Bytes 10-13	$Accel_Z$	*(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$	*(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$	*(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$	*(IEEE-754 Floating Point)*
Bytes 26-29	Mag_X	*(IEEE-754 Floating Point)*
Bytes 30-33	Mag_Y	*(IEEE-754 Floating Point)*
Bytes 34-37	Mag_Z	*(IEEE-754 Floating Point)*
Bytes 38-41	*Timer*	
Bytes 42-43	*Checksum*	

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

Function:	The 3DM-GX3® will output a data record containing the acceleration, angular rate and magnetometer vectors and the orientation matrix.	
Command:		
Byte 1	0xCC	
Response:		
Byte 1	0xCC	
Bytes 2-5	$Accel_X$	*(IEEE-754 Floating Point)*
Bytes 6-9	$Accel_Y$	*(IEEE-754 Floating Point)*
Bytes 10-13	$Accel_Z$	*(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$	*(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$	*(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$	*(IEEE-754 Floating Point)*
Bytes 26-29	Mag_X	*(IEEE-754 Floating Point)*
Bytes 30-33	Mag_Y	*(IEEE-754 Floating Point)*
Bytes 34-37	Mag_Z	*(IEEE-754 Floating Point)*
Bytes 38-41	$M_{1,1}$	*(IEEE-754 Floating Point)*
Bytes 42-45	$M_{1,2}$	*(IEEE-754 Floating Point)*
Bytes 46-49	$M_{1,3}$	*(IEEE-754 Floating Point)*
Bytes 50-53	$M_{2,1}$	*(IEEE-754 Floating Point)*
Bytes 54-57	$M_{2,2}$	*(IEEE-754 Floating Point)*
Bytes 58-61	$M_{2,3}$	*(IEEE-754 Floating Point)*
Bytes 62-65	$M_{3,1}$	*(IEEE-754 Floating Point)*
Bytes 66-69	$M_{3,2}$	*(IEEE-754 Floating Point)*
Bytes 70-73	$M_{3,3}$	*(IEEE-754 Floating Point)*
Bytes 74-77	*Timer*	
Bytes 78-79	*Checksum*	

Capture Gyro Bias (0xCD)

Function:	This command will cause the 3DM-GX3® 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 GyroBias vector. Notes: The 3DM-GX3® must be stationary during the execution of the Capture Gyro Bias Operation. The input parameter *SamplingTime* 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, *GyroBias*, reflects the values automatically written into that vector as a result of the Capture Gyro Bias Operation.
Command:	
Byte 1	0xCD
Byte 2	0xC1 (Confirms user intent)
Byte 3	0x29 (Confirms user intent)
Bytes 4-5	*SamplingTime* *(16 bit unsigned integer)*
Response:	
Byte 1	0xCD
Bytes 2-5	*$GyroBias_X$* *(IEEE-754 Floating Point)*
Bytes 6-9	*$GyroBias_Y$* *(IEEE-754 Floating Point)*
Bytes 10-13	*$GyroBias_Z$* *(IEEE-754 Floating Point)*
Bytes 14-17	*Timer*
Bytes 18-19	*Checksum*

Euler Angles (0xCE)

Function:	The 3DM-GX3® will output a data record containing Euler Angles.
Command:	
Byte 1	0xCE
Response:	
Byte 1	0xCE
Bytes 2-5	*Roll* *(IEEE-754 Floating Point)*
Bytes 6-9	*Pitch* *(IEEE-754 Floating Point)*
Bytes 10-13	*Yaw* *(IEEE-754 Floating Point)*
Bytes 14-17	*Timer*
Bytes 18-19	*Checksum*

Euler Angles and Angular Rates (0xCF)

Function:	The 3DM-GX3® will output a data record containing Euler Angles and Angular Rates.
Command:	
Byte 1	0xCF
Response:	
Byte 1	0xCF
Bytes 2-5	*Roll* *(IEEE-754 Floating Point)*
Bytes 6-9	*Pitch* *(IEEE-754 Floating Point)*
Bytes 10-13	*Yaw* *(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$ *(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$ *(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$ *(IEEE-754 Floating Point)*
Bytes 26-29	*Timer*
Bytes 30-31	*Checksum*

Transfer Quantity to Non-Volatile Memory (0xD0)

<table>
<tr><td>Function:</td><td>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 “<i>Transfer Quantity</i>” 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.</td></tr>
<tr><td>Command:</td><td></td></tr>
<tr><td>Byte 1</td><td>0xD0</td></tr>
<tr><td>Byte 2</td><td>0xC1 (Confirms user intent)</td></tr>
<tr><td>Byte 3</td><td>0x29 (Confirms user intent)</td></tr>
<tr><td>Bytes 4-5</td><td><i>Transfer Quantity</i> <i>(16 bit unsigned integer)</i></td></tr>
<tr><td>Response:</td><td></td></tr>
<tr><td>Byte 1</td><td>0xD0</td></tr>
<tr><td>Bytes 2-3</td><td><i>Transfer Quantity</i></td></tr>
<tr><td>Bytes 4-7</td><td><i>Timer</i></td></tr>
<tr><td>Bytes 8-9</td><td><i>Checksum</i></td></tr>
</table>

Temperatures (0xD1)

Function:	The 3DM-GX3® will output a data record containing its 4 internal temperature measurements. These temperatures are grouped with one or more sensors by proximity. See Temperatures section for equations to convert to real units.
Command:	
Byte 1	0xD1
Response:	
Byte 1	0xD1
Bytes 2-3	$Temperature_{\mathrm{Mag}}$ *(16 bit unsigned integer)*
Bytes 4-5	$Temperature_{\mathrm{Gyro3(Y)\ Accel3(X)}}$ *(16 bit unsigned integer)*
Bytes 6-7	$Temperature_{\mathrm{Gyro2(X)\ Accel1(Z)\_Accel2(Y)}}$ *(16 bit unsigned integer)*
Bytes 8-9	$Temperature_{\mathrm{Gyro1(Z)}}$ *(16 bit unsigned integer)*
Bytes 10-13	*Timer*
Bytes 14-15	*Checksum*

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

Function:	The 3DM-GX3® will output a data record containing the gyro-stabilized acceleration, angular rate, and gyro-stabilized magnetometer vectors.
Command:	
Byte 1	0xD2
Response:	
Byte 1	0xD2
Bytes 2-5	$StabAccel_X$ *(IEEE-754 Floating Point)*
Bytes 6-9	$StabAccel_Y$ *(IEEE-754 Floating Point)*
Bytes 10-13	$StabAccel_Z$ *(IEEE-754 Floating Point)*
Bytes 14-17	$AngRate_X$ *(IEEE-754 Floating Point)*
Bytes 18-21	$AngRate_Y$ *(IEEE-754 Floating Point)*
Bytes 22-25	$AngRate_Z$ *(IEEE-754 Floating Point)*
Bytes 26-29	$StabMag_X$ *(IEEE-754 Floating Point)*
Bytes 30-33	$StabMag_Y$ *(IEEE-754 Floating Point)*
Bytes 34-37	$StabMag_Z$ *(IEEE-754 Floating Point)*
Bytes 38-41	*Timer*
Bytes 42-43	*Checksum*

DeltaAngle & DeltaVelocity & Magnetometer Vectors (0xD3)

Function:	The 3DM-GX3® will output a data record containing the DeltaAngle, DeltaVelocity and Magnetic Field vectors. If the magnetometer is turned off or not present, the magnetometer vector will contain *NaNs*.
Command:	
Byte 1	0xD3
Response:	
Byte 1	0xD3
Bytes 2-5	$\mathit{DeltaAng}_X$ *(IEEE-754 Floating Point)*
Bytes 6-9	$\mathit{DeltaAng}_Y$ *(IEEE-754 Floating Point)*
Bytes 10-13	$\mathit{DeltaAng}_Z$ *(IEEE-754 Floating Point)*
Bytes 14-17	$\mathit{DeltaVel}_X$ *(IEEE-754 Floating Point)*
Bytes 18-21	$\mathit{DeltaVel}_Y$ *(IEEE-754 Floating Point)*
Bytes 22-25	$\mathit{DeltaVel}_Z$ *(IEEE-754 Floating Point)*
Bytes 26-29	Mag_X *(IEEE-754 Floating Point)*
Bytes 30-33	Mag_Y *(IEEE-754 Floating Point)*
Bytes 34-37	Mag_Z *(IEEE-754 Floating Point)*
Bytes 38-41	*Timer*
Bytes 42-43	*Checksum*

Mode (0xD4)

Function:	Put the 3DM-GX3® into new mode or read current mode.
Command:	
Byte 1	0xD4
Byte 2	0xA3 (Confirms user intent)
Byte 3	0x47 (Confirms user intent)
Byte 4	Mode selector: *(8 bit unsigned integer)* 0: No mode change: just read current mode 1: Put in Active mode (default) 2: Put in Continuous mode (use Continuous Preset first) 3: Put in Idle mode 4: Put in Sleep mode 5: Put in Deep Sleep mode
Response:	
Byte 1	0xD4
Byte 2	Current or new mode *(8 bit unsigned integer)*
Bytes 3-4	*Checksum*

Mode Preset (0xD5)

Function:	Store or read the value for the mode preset. The mode preset is the mode that the 3DM-GX3® will enter when powered up or woken from sleep. Stored in non-volatile memory.
Command:	
Byte 1	0xD5
Byte 2	0xBA (Confirms user intent)
Byte 3	0x89 (Confirms user intent)
Byte 4	Mode selector: *(8 bit unsigned integer)* 0: No change: just read current mode preset 1: Set to wake up in Active mode (default) 2: Set to wake up in Continuous mode 3: Set to wake up in Idle mode
Response:	
Byte 1	0xD5
Byte 2	Current or new preset mode
Bytes 3-4	Checksum

Continuous Preset (0xD6)

Function:	Store or read the data command that will be used for Continuous mode. Stored in non-volatile memory.
Command:	
Byte 1	0xD6
Byte 2	0xC6 (Confirms user intent)
Byte 3	0x6B (Confirms user intent)
Byte 4	Data command to be stored and used for Continuous mode. If this value is 0 or an invalid data command value, no command will be stored and the currently stored value will be returned in the response. *(8 bit unsigned integer)*
Response:	
Byte 1	0xD6
Byte 2	Data command used for Continuous mode
Bytes 3-4	Checksum

Timer (0xD7)

Function:	Set or read the 32 bit time stamp. Timestamp is in 1/62,500 second increments (16uS). If a PPS counter is implemented, you can use this function to set or read that counter also.
Command:	
Byte 1	0xD7
Byte 2	0xC1 (Confirms user intent)
Byte 3	0x29 (Confirms user intent)
Byte 4	Function selector: *(8 bit unsigned integer)* 0: Do not change the time stamp, just return current value 1: Restart the time stamp at the new value 2: Restart the PPS Seconds counter at the new value
Byte 5-8	New Timer Value
Response:	
Byte 1	0xD7
Byte 2 - 5	New or current time stamp value
Bytes 6-7	*Checksum*

Communications Settings (0xD9)

Function:	Set or Read the parameters of the specified communications port. The response is always given at the current settings. The settings are only changed after the response has been sent. (Note that after a firmware update, the communications settings on UART1 are always reset to 115200, 8-1-none). UART1 is the only UART available on the 3DM-GX3®-25
Command:	
Byte 1	0xD9
Byte 2	0xC3 (Confirms user intent)
Byte 3	0x55 (Confirms user intent)
Byte 4	Port Selector: *(8 bit unsigned integer)* 1: UART1, Primary UART for host communication 2: UART2, *Not Used on 3DM-GX3®-25* 3: UART3, *Not Used on 3DM-GX3®-25*
Byte 5	Function selector: *(8 bit unsigned integer)* 0: Do not change the parameters, just return current values (all other parameter values are ignored) 1: Change the parameters to the new values temporarily 2: Change the parameters and make them permanent (remember them in EEPROM)
Byte 6-9	BAUD Rate: *(32 bit unsigned integer)* 115200 (default), 230400, 460800, or 921600 *Warning! Check to make sure your host is able to handle the higher BAUD rates before changing this value*
Byte 10	Port Configuration: *(8 bit unsigned integer)* Bit 0: 0: Reserved. Set to 0 Bit 1: 0: Selected UART Disabled * 1: Selected UART Enabled * UART 1 cannot be disabled
Byte 11	Reserved: Set to 0
Response:	
Byte 1	0xD9
Byte 2	Port Selector
Byte 3-6	BAUD Rate
Byte 7	Port Configuration bits
Byte 8	Reserved
Bytes 9-10	Checksum

Stationary Test (0xDA)

Function:	The 3DM-GX3® will output a 32 bit value with single bit flags representing the status of various internal functions. If the bit is set, it means the corresponding function is operating within specification. The bit flags are defined as follows:

```
u32 st3VGood: 1;      // 3V supply is above 2.9V
u32 st5VGood: 1;      // 5V supply is within 5%
u32 stHSEReady:1;      // High speed external clk is stable
u32 stHSIReady:1;      // High speed internal clk is stable
u32 stPLLReady:1;      // PLL is locked and stable
u32 reserved:1;
u32 stGy1Good:1;       // GY1 passed test
u32 stGy2Good:1;       // GY2 passed test
u32 stGy3Good: 1;      // GY3 passed test
u32 stAcc1Good:1;      // Acc1 passed test
u32 stAcc2Good:1;      // Acc2 passed test
u32 stAcc3Good:1;      // Acc3 passed test
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 reserved:1;
u32 stSysTickGood:1;   // System tick and tick clock are in
agreement
```

	Note that the 5 volt flag will be zero in the “Idle” mode as the 5 volt supply is turned off to conserve power. During “Active” and “Continuous” modes, all flags should be “1”. Also note that this is a “static” test – that is, the device should be stationary for this test. If the device is not stationary, the Gyro, Accel, and Mag tests are likely to fail. The other tests are valid for a non-stationary device.
Command:	
Byte 1	0xDA
Response:	
Byte 1	0xDA
Bytes 2-5	Test Result *(32 bit unsigned integer)*
Bytes 6-7	Checksum

*Sampling Settings (0xDB)**

Function:	Set or Read the parameters that establish data rate, oversampling rate, decimation rate, digital filter bandwidth, and post processing
Command:	
Byte 1	0xDB
Byte 2	0xA8 (Confirms user intent)
Byte 3	0xB9 (Confirms user intent)
Byte 4	Function selector: *(8 bit unsigned integer)* 0: Do not change the parameters, just return current values (parameter values are ignored) 1*: Change the parameters to the new values. 2*: Change the parameters and store in non-volatile memory. 3**: Change the parameters to the new values but do not send a reply.
Byte 5-6	Data Rate decimation value. This value is divided into a fixed 1000Hz reference rate to establish the data output rate. Setting this value to 10 gives an output data rate of 1000/10 = 100 samples/sec. *When using the UART for communications, at data rates higher than 250, the UART baud rate must be increased (see command 0xD9).* Minimum Value is 1, Maximum value is 1000.
Byte 7-8	Data conditioning function selector: Bit 0: if set - Calculate orientation. *Default is "1"* Bit 1: if set - Enable Coning&Sculling. *Default is "1"* Bit 2 – 3: *reserved. Default is "0"* Bit 4: if set – Floating Point data is sent in Little Endian format (only floating point data from IMU to HOST is affected). *Default is "0"* Bit 5: if set – NaN data is suppressed. *Default is "0"* Bit 6: if set, enable finite size correction *Default is "0"* Bit 7: *reserved. Default is "0"* Bit 8**: if set, disables magnetometer *Default is "0"* Bit 9**: if set, enables magnetometer low power/low resolution mode. *Default is "0"* Bit 10**: if set, disables magnetic north compensation *Default is "0"* Bit 11**: if set, disables gravity compensation *Default is "0"* Bit 12**: if set, enables Quaternion calculation *Default is "0"* Bit 13 – 15**: *reserved. Default is "0"*

Byte 9	Gyro and Accel digital filter window size. First null is 1000 divided by this value. Minimum value is 1, maximum value is 32. *Default is 15*
Byte 10	Mag digital filter window size. First null is 1000 divided by this value. Minimum value is 1, maximum value is 32 *Default is 17.*
Byte 11-12**	Up Compensation in seconds. Determines how quickly the gravitational vector corrects the gyro stabilized pitch and roll. Minimum value is 1, maximum value is 1000. *Default is 10*
Byte 13-14**	North Compensation in seconds. Determines how quickly the magnetometer vector corrects the gyro stabilized yaw. Minimum value is 1, maximum value is 1000. *Default is 10*
Byte 15-20	*reserved. Set to "0"*
Response:	*(not sent for function selector 3)*
Byte 1	0xDB
Byte 2-3	Data Rate decimation value
Byte 4-5	Data conditioning function selector value
Byte 6	Gyro and Accel Digital filter window size value
Byte 7	Mag Digital filter window size value
Byte 8-9	Up Compensation value
Byte 10-11	North Compensation value
Byte 12-17	Reserved values
Bytes 18-19	Checksum

** Firmware versions previous to 1.1.27 stored new settings in non-volatile memory when using the "1" command.. Version 1.1.27 and later only store new settings in non-volatile memory only if the "2" command is used.*

*** New feature of firmware version 1.1.27 and later*

Realign Up and North (0xDD)

Firmware version 1.1.27 and above

Function:	This command will realign the "Up" and "North" vectors (Gyro Stabilized) using the specified time constants. This temporarily changes the Up and North comp gains to accelerate the realignment to the gravitational and magnetic north vectors (see Bytes 11-14 of the Sampling Settings command).
Command:	
Byte 1	0xDD
Byte 2	0x54 (Confirms user intent)
Byte 3	0x4C (Confirms user intent)
Byte 4	Function selector: *(8 bit unsigned integer)* 0: Realign Up and North. 3: Realign Up and North but do not send a reply.
Byte 5	Up Realign: 1-100: Realign time in 1/10 seconds (to 90%) 255: Do not realign this vector
Byte 6	North Realign:1-100: Realign time in 1/10 seconds (to 90%) 255: Do not realign this vector
Bytes 7-10	Reserved: Set to 0
Response:	*(not sent for function selector 3)*
Byte 1	0xDD
Byte 2-5	Reserved: returns undetermined value
Bytes 6-7	*Checksum*

Quaternion (0xDF)

Firmware version 1.1.27 and above

Function:	The 3DM-GX3® will output a data record containing the orientation in Quaternion form Q. The values are conveyed in 32 bit IEEE-754 floating point format. Note that the default sampling settings leave the quaternion calculation "off". To enable this output, the data conditioning function selector in Sampling Settings must have bit 0, bit 1 and bit 11 set. Turning this calculation on reduces the maximum data rate to 500Hz and slightly increases power consumption.
Command:	
Byte 1	0xDF
Response:	
Byte 1	0xDF
Bytes 2-5	*q0* *(IEEE-754 Floating Point)*
Bytes 6-9	*q1* *(IEEE-754 Floating Point)*
Bytes 10-13	*q2* *(IEEE-754 Floating Point)*
Bytes 14-17	*q3* *(IEEE-754 Floating Point)*
Bytes 18-21	*Timer*
Bytes 22-23	*Checksum*

Write Word to EEPROM (0xE4)

Function:	The specified word will be written to EEPROM of the 3DM-GX3® at the specified address. The only valid addresses for user read or write are from 0x0000 – 0x00EF and 0x0400->0x042F. All other addresses will be ignored. ***Important! These EEPROM locations are not for general purpose use. They are all reserved by MicroStrain. Do not write to any EEPROM location unless specifically instructed by a MicroStrain engineer. Doing otherwise could result in device malfunction and is not covered under warranty.***
Command:	
Byte 1	0xE4
Byte 2	0xC1 (Confirms user intent)
Byte 3	0x29 (Confirms user intent)
Byte 4	0x00
Byte 5-6	16 bit address of EEPROM word to be written *(16 bit unsigned integer)*
Byte 7-8	Value of 16 bit word to be written to EEPROM *(16 bit unsigned integer)*
Response:	
Byte 1	*Header* = 0xE4
Bytes 2-3	EEPROM word (16 bit)
Bytes 4-5	*Checksum*

Read Word from EEPROM (0xE5)

Function:	The 3DM-GX3® will output the value of the 16bit word stored in EEPROM at the specified address. The only valid addresses for user read or write are from 0x0000 – 0x00EF and 0x0400->0x042F. All other addresses will be ignored.
Command:	
Byte 1	0xE5
Byte 2	0x00
Byte 3-4	16 bit address of EEPROM word to be read *(16 bit unsigned integer)*
Response:	
Byte 1	0xE5
Bytes 2-3	EEPROM word (16 bit)
Bytes 4-5	*Checksum*

Read Firmware Version Number (0xE9)

Function:	The 3DM-GX3® will report its firmware version number to the host. This is a 32 bit integer value which is hard-coded in firmware.
Command:	
Byte 1	0xE9
Response:	
Byte 1	0xE9
Byte 2	Firmware Number Byte 1 (Most significant)
Byte 3	Firmware Number Byte 2
Byte 4	Firmware Number Byte 3
Byte 5	Firmware Number Byte 4 (Least significant)
Bytes 6-7	*Checksum*

Read Device ID String (0xEA)

Function:	Reads the specified device ID string from the 3DM-GX3®.
Command:	
Byte 1	0xEA
Byte 2	ID String Selector: *(8 bit unsigned integer)* 0: Model number string 1: Serial number string 2: Model name string 3: Device options string 4: Lot number string
Response:	
Byte 1	0xEA
Byte 2	Echo of the ID String Selector
Byte 3-18	16 character ASCII ID string
Bytes 19-20	*Checksum*

Stop Continuous Mode (no reply) (0xFA)

Function:	This command will stop the continuous mode of the 3DM-GX3® and place it in Active mode without generating a response packet.
Command:	
Byte 1	0xFA
Byte 2	0x75 (Confirms user intent)
Byte 3	0xB4 (Confirms user intent)
Response:	
	None

Device Reset (no reply) (0xFE)

Function:	This command will do a soft reset of the 3DM-GX3®.
Command:	
Byte 1	0xFE
Byte 2	0x9E (Confirms user intent)
Byte 3	0x3A (Confirms user intent)
Response:	
	None

Calculating The Checksum

The 16 bit checksum is equal to the sum of all preceding bytes with rollover from 65535 to 0. The C code snippet below demonstrates how to calculate the checksum and compare it to the big-endian checksum value extracted from a reply packet.

```
unsigned char     tGoodResponse;
unsigned short    tix;
unsigned short    tChksum;
unsigned short    tResponseChksum;
unsigned char     tResponse[128];
unsigned short    tResponseLen = 31;

//-------- Get a data reply packet from the device
GetAccelAngRateResponse(tResponse, tResponseLen);

//-------- Calculate the checksum
tChksum = 0;
for (tix = 0; tix < tResponseLen - 2; tix++)
      tChksum += tResponse[tix];

//-------- Extract the big-endian checksum from reply
tResponseChksum = 0;
tResponseChksum = tResponse[tResponseLen - 2] << 8;
tResponseChksum += tResponse[tResponseLen - 1];

//-------- Compare the checksums
tGoodResponse = (tChksum == tResponseChksum);
```

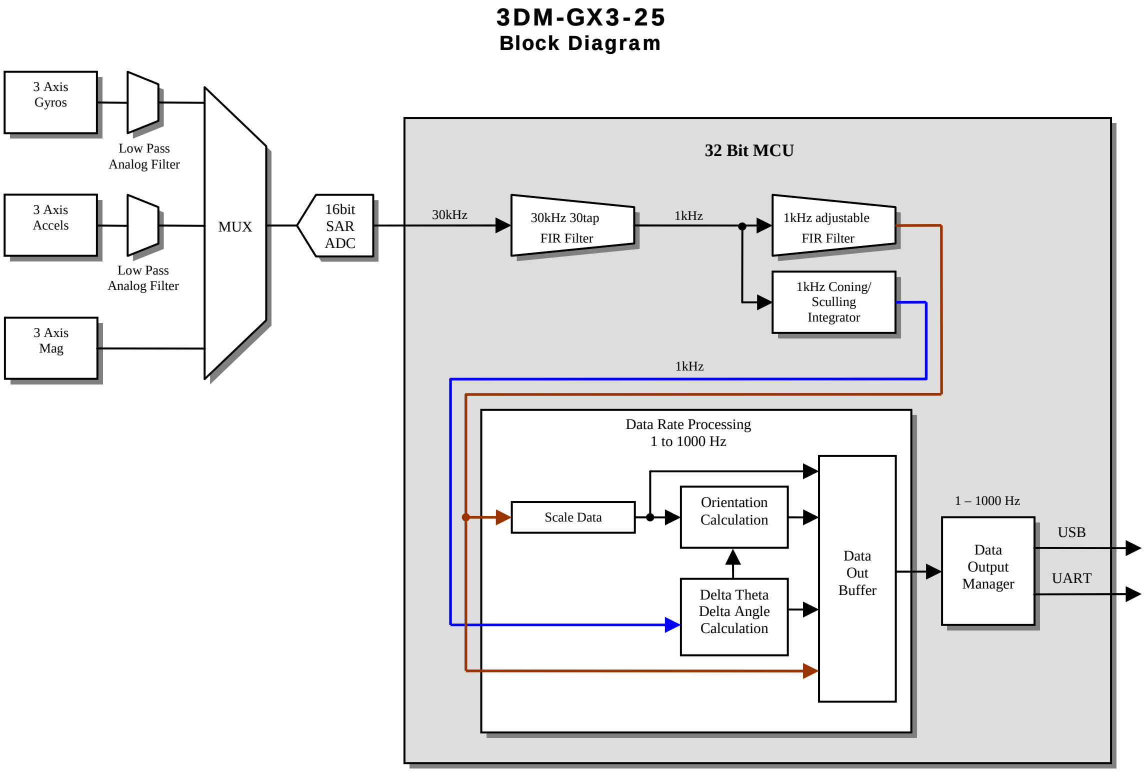
Internal Architecture

Calculation Cycle

The on-board processor of the 3DM-GX3® continuously executes a calculation cycle in Active or Continuous mode. The steps in this cycle include the following:

1. Convert raw sensor outputs into digital form
2. Apply first stage digital filter to the raw data to decrease noise and limit bandwidth.
3. Scale sensor outputs into physical units (including temperature, alignment, and G-sensitivity compensation).
4. Apply Coning and Sculling correction if enabled.
5. Propagate and filter the orientation estimate if enabled.
6. If host has issued a command byte (or if operating in continuous mode), compute appropriate response data and transmit.

Step 6 in this cycle is only executed if the 3DM-GX3® has received a command byte from the host or if the device is in continuous mode.



3DM-GX3-25
Block Diagram

Figure 1

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.

Data Output Rate

The data output rate may be set by manipulating the Decimation value using the Sampling Settings command. The BAUD rate may be set using the Communications Settings command.

Warning! Check to make sure your host is able to handle the higher BAUD rates before setting the BAUD rate to a value other than 115200

Below is a table of maximum output rates (data point per second) at various settings for most commands in continuous mode (Commands with 31 byte result). Output rates greater than this may cause skipping of packets. Longer data quantities may require lower data output rates. Coning & Sculling only needs to be turned on for DeltaAngle, DeltaVelocity and Orientation data commands. The Orientation calculation may only be turned on for output rates of 500 and below (true for serial and USB connections):

BAUD Rate	***Coning&Sculling***	***Orientation***	***Decimation Value***	***Max Output Rate***
115200	ON	ON	4	250
115200	OFF	OFF	4	250
230400	ON	ON	3	333
230400	OFF	OFF	3	333
460800	ON	ON	2	500
460800	OFF	OFF	2	500
921600	ON	OFF	1	1000
921600	OFF	OFF	1	1000

Table 5

Skipping can be detected by examining the timestamps of adjacent packets. The difference between the timestamps should match the period of the data output rate.

USB vs. Serial Output

Most of the 3DM-GX3 series include both a USB interface and a Serial interface (either RS-232 or TTL). There are some important characteristics to consider when choosing to use one of these interfaces. Each has its strengths and weaknesses.

USB

USB has several compelling features:

1. Power is supplied by the cable from the host computer (no external supply required)
2. The bandwidth is very high allowing even the largest records to be transferred at high data rates (up to 1000Hz).
3. USB is a universal standard and is thus supported by most host devices such as laptop computers.

The disadvantages of USB are:

1. It requires special drivers for each device type. The 3DM-GX3 uses a generic “Virtual COM Port” type of driver interface which is supported by the usbser.sys driver built into Windows.
2. The transfer of data from a connected device is completely controlled by the host – that is if the host doesn’t want to listen to a USB device, it can ignore it momentarily.
3. The latency of the data is not deterministic. The host will get to you when they can get to you which may be right now or maybe in a millisecond or two.

It is the second disadvantage that can sometimes cause “packet drops” or gaps in data. The 3DM-GX3 can buffer a small amount of data during these host “snubs” but at high data rates, gaps can occur. You can avoid these gaps by minimizing the user interface activity and/or the number of programs being run simultaneously on the host machine. It is good practice to always monitor the timestamp of the packets and check that the interval is what you expect it to be. You should not experience packet drops frequently, but they cannot be completely avoided.

The third disadvantage could be a problem in systems where latency must be short and/or guaranteed as in autopilots or platform stabilization systems where sensor output is part of a feedback network.

Fortunately, the serial interface provides almost a mirror opposite of the USB advantages and disadvantages...

Serial

The advantages of serial are:

1. It does not require special drivers – serial data is transferred “as is” with no additional wrapping, handshaking, or protocol.
2. The transfer of data is controlled by the sensor. Data is always transmitted as soon as it is ready.

3. The latency of the data is deterministic. Because there is no flow control, the data is always transmitted immediately with very little latency.

The disadvantages of serial are:

1. It requires an external power supply
2. There is no flow control - the host has to be fast enough to process the data as fast or faster than it is coming in from the sensor or it may drop packets.
3. The bandwidth of the serial UART is limited and so some larger data packets cannot be accommodated at the highest sample rates.

Most microcontrollers and almost all PC's can handle even the highest data rates of the 3DM-GX3, however managing the volume of the incoming data either through storage or processing algorithms can be problematic.

In general it is very convenient to develop interface code on a host computer using the USB interface and then switch to a serial interface for the embedded target. For the most part the interfaces will perform equivalently, but you must pay attention to the differences especially in applications that cannot tolerate packet drops or inconsistent latency. In addition, it is always good practice to monitor the timestamps and checksums of the incoming packets so you can be aware of packet drops and re-sync with the data stream if you do miss a few bytes.

Filtering

Noise filtering on the 3DM-GX3® is accomplished using analog anti-aliasing filters followed by a two stage digital moving average filter. The analog filters are fixed and have bandwidths characterized in Table 6.

Analog Anti Alias Filter Bandwidths	
Accelerometer (5g)	226Hz (1 pole RC)
Gyroscope (300d/s)	358Hz (2 pole RC)
Accelerometer (50g)	360Hz (2 pole Bessel)
Gyroscope (1200d/s)	660Hz (2 pole RC)
Magnetometer	*no filter*

Table 6

Two Stage Digital Filter

The digital filter has two stages. The first stage is a fixed 30kHz 30 tap moving average filter. The second stage is a 1kHz variable width moving average filter. The second stage is adjustable by means of the filter window size (aka filter width; filter taps, filter points). The transfer function of the digital filter is as follows:

$$H[f] = \left| \frac{\sin(M\pi f/1000)}{M\sin(\pi f/1000)} \times \frac{\sin(30\pi f/30000)}{30\sin(\pi f/30000)} \right|$$

Equation 2

M is the width of the second stage filter and f is the input frequency in *Hz*. For example, for an input frequency of 75*Hz*, and a filter width of 10, the attenuation is:

$$H[f] = \left| \frac{\sin(10\pi 75/1000)}{10\sin(\pi 75/1000)} \times \frac{\sin(30\pi 75/30000)}{30\sin(\pi 75/30000)} \right|$$

$$H[f] = 0.300$$

Example 1

The first stage 30kHz filter removes high frequency spectral noise produced by the MEMs sensors and is a smaller factor in attenuating signals in the hundred hertz range.

Magnetometer Digital Filter (High Resolution)

The magnetometer has special sampling criteria that result in an oversample rate of ¼ of the fixed oversample rate of 30000 which is used for the other sensors. This means the oversample rate of the magnetometer is 7500Hz. A fixed 2 point averaging filter is applied to the signal followed by a 7 point averaging filter. The result of this filter is fed at 1kHz to an adjustable filter with a window size from 1 to 32.

Note: The Magnetometer does not have anti-aliasing filters. Magnetic noise above 3750Hz will be aliased.

The transfer function for the magnetometer becomes:

$$H[f] = \left| \frac{\sin(2\pi f/7500)}{2\sin(\pi f/7500)} \times \frac{\sin(7\pi f/3750)}{7\sin(\pi f/3750)} \times \frac{\sin(M_m \pi f/1000)}{M_m \sin(\pi f/1000)} \right|$$

Equation 3

Where f is the input frequency in Hz and M_m is the magnetometer filter width. Note that the first stage of the filter changes the sampling frequency of the second stage to 3750Hz. This also results in a first null at 3750Hz.

As an example, an input frequency of 60Hz and filter window width, M_m, of 16 is attenuated as follows:

$$M_m = 16$$

$$H[f] = \left| \frac{\sin(2\pi 60/7500)}{2\sin(\pi 60/7500)} \times \frac{\sin(7\pi 60/3750)}{7\sin(\pi 60/3750)} \times \frac{\sin(16\pi 60/1000)}{16\sin(\pi 60/1000)} \right|$$

$$= 0.9997 \times 0.9799 \times 0.0418$$

$$H[f] = 0.041$$

Example 2

As can be seen, the first two filter terms are close to 1 so a simplified transfer function can be used for the purpose of calculating the attenuation of the adjustable filter:

$$H[f] \sim \left| \frac{\sin(M_m \pi f / 1000)}{M_m \sin(\pi f / 1000)} \right|$$

Equation 4

Magnetometer Digital Filter (Low Power)

Firmware version 1.1.27 and above

The magnetometer may be put into a lower power mode which results in lower resolution and slightly increased noise. The filtering is affected in two ways: (1) it removes the second stage filtering and (2) it changes the adjustable filter sample rate factor from 1000 to [datarate] where the [datarate] is in Hz (see Sampling Settings command). The result is that the simplified filtering transfer function becomes:

$$H[f] \sim \left| \frac{\sin(M_m \pi f / [datarate])}{M_m \sin(\pi f / [datarate])} \right|$$

Equation 5

Digital Filter Characteristics

One of the most important aspects of the moving average FIR filter is that it is the best filter to use with respect to step response in the time domain. This is important for systems that want to avoid overshoot and ringing from stepped input signals. The attenuation is moderate in the frequency domain but reasonable for applications that require a low cutoff frequency and have moderate noise in the near-band spectrum. The analog anti-alias filters (on the accelerometers and gyros) cascade with the digital filter to improve attenuation of above-band signals and noise. The first stages of the magnetometer filter attenuate high frequency noise up to 3750Hz. The adjustable stage of the magnetometer can be adjusted to filter out the highly prevalent 50/60Hz power line noise.

Figure 1 shows frequency response curves for a 3 point, 11 point, and 31 point single stage moving average filter.

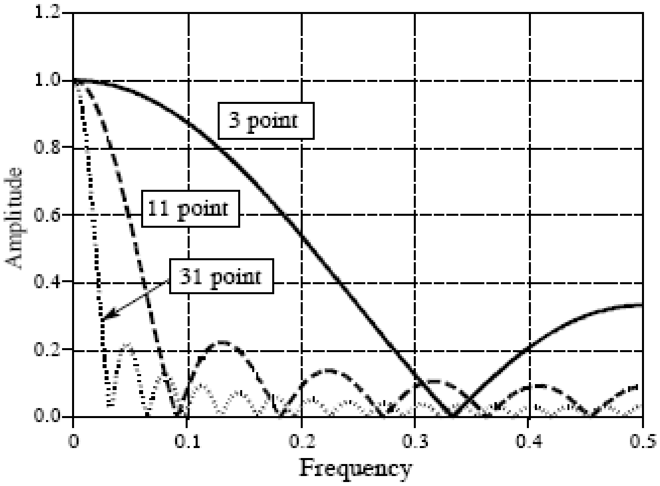


Figure 2

Best Performance

The best performance of the 3DM-GX3® occurs after all the sensors have warmed up and the operating temperature gradients of the unit have stabilized. These are the conditions that the 3DM-GX3® is calibrated under and where the best performance will be realized.

Data Quantities Available

The 3DM-GX3® is 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 131071 represents ~5 volts (17 bits of resolution). The accelerometer signals do not actually register the full range of possible codes – the range varies slightly from device to device. The raw accel values 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. They are useful for validating individual accel raw signal quality.

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

Equation 6

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 3DM-GX3®*'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 131071 represents ~5 volts (17 bits of resolution). The gyro signals do not actually register the full range of possible codes – the range varies slightly from device to device. The raw values are not scaled into physical units, nor are the individual components necessarily orthogonal, or forming a right-handed coordinate system. These quantities are not recommends for most applications.

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

Equation 7

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 3DM-GX3$^{®}$*'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 3DM-GX3$^{®}$ 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 m/sec^2). It is expressed in terms of the 3DM-GX3$^{®}$'s local coordinate system.

$$Accel = \begin{bmatrix} Accel_X \\ Accel_Y \\ Accel_Z \end{bmatrix} (g)$$

Equation 8

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 m/sec^2 corresponds to sea-level locations at approximately 45.5 degrees latitude. A suitable model is the following:

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

Equation 9

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 = 9.77879m/sec^2. This is equal to 0.99716**g*. Therefore, under static conditions, the Accel vector output by the 3DM-GX3® would have a magnitude of 0.997*g*.

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 thought the system may be exposed to significant linear acceleration.

$$StabAccel = \begin{bmatrix} StabAccel_X \\ StabAccel_Y \\ StabAccel_Z \end{bmatrix} (g)$$

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 3DM-GX3®. 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 3DM-GX3®'s local coordinate system in units of radians/second.

$$AngRate = \begin{bmatrix} AngRate_X \\ AngRate_Y \\ AngRate_Z \end{bmatrix} \text{(rad/sec)}$$

Equation 10

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 3DM-GX3®'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/sec^2 .

$$DeltaVel = \begin{bmatrix} DeltaVel_X \\ DeltaVel_Y \\ DeltaVel_Z \end{bmatrix} \text{(g*sec)}$$

Equation 11

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 3DM-GX3®'s local coordinate system in units of radians.

$$DeltaVel = \begin{bmatrix} DeltaVel_X \\ DeltaVel_Y \\ DeltaVel_Z \end{bmatrix} \text{(rad)}$$

Equation 12

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-GX3®'s local coordinate system in units of Gauss. The 3DM-GX3® returns NaN for these values.

$$Magnetometer = \begin{bmatrix} Mag_X \\ Mag_Y \\ Mag_Z \end{bmatrix} \text{(G)}$$

Equation 13

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 thought the system may be exposed to transient magnetic interference. Note that sustained magnetic interference cannot be adequately compensated for by the complementary filter.

$$StabMag = \begin{bmatrix} StabMag_X \\ StabMag_Y \\ StabMag \end{bmatrix} (G)$$

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 or timer set command. To convert the timer value to time in seconds, divide by 62,500. 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 68719 seconds (~1145 minutes or ~19 hours). 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 3DM-GX3® 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:

$$V\_IL_i = M_{ij} \cdot V\_E_j$$

Where: *V_IL* is a vector expressed in the 3DM-GX3®'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:
($M_{1,1}, M_{1,2}, M_{1,3}, M_{2,1}, \ldots , M_{3,2}, M_{3,3}$*).*

C

This is a 9 component coordinate transformation matrix which describes the change in orientation of the 3DM-GX3® 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:
($C_{1,1}, C_{1,2}, C_{1,3}, C_{2,1}, \ldots , C_{3,2}, C_{3,3}$*).*

Q

This is a 4 component quaternion which describes the orientation of the 3DM-GX3 with respect to the fixed earth coordinate quaternion.

$$Q = \begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix}$$

Q satisfies the following equation:

$V\_IL_i = Q \bullet V\_E \bullet Q^{-1}$

Where: *V_IL* is a vector expressed in the 3DM-GX3®'s local coordinate system.
V_E is the same vector expressed in the stationary, earth-fixed coordinate system

Data Format: 4 element array of 32bit floating point values in IEEE-754 format.

Note: • indicates a quaternion product

Euler Angles

This is a 3 component vector containing the Roll, Pitch and Yaw angles in radians. It is computed on the 3DM-GX3® 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 3DM-GX3®'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 3DM-GX3®, in the form of the *AccelBias* vector. *AccelBias* will be subtracted from every measurement of the *Accel* vector made by the 3DM-GX3® to form a bias corrected version of the *Accel* quantity. This will then 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 3DM-GX3®'s local coordinate system. On system power-up, *GyroBias* is initialized to a value of [0,0,0].

$$GyroBias = \begin{bmatrix} GyroBias_X \\ GyroBias_Y \\ GyroBias_Z \end{bmatrix} (g)$$

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 3DM-GX3®, in the form of the *GyroBias* vector. *GyroBias* will be subtracted from every measurement of the *AngRate* vector made by the 3DM-GX3® to form a bias corrected version of the *AngRate* quantity. This will then 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 – (Mag,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.

$$Temperature = \begin{bmatrix} Temperature_{Mag} \\ Temperature_{GyroY,AccX} \\ Temperature_{GyroX,AccZ,AccY} \\ Temperature_{GyroZ} \end{bmatrix} \text{(A/D Converter Bits)}$$

Equation 14

$Temperature_{Mag}$ is derived from a PC board mounted temperature sensor located near the 3-axis magnetometer sensor. $Temperature_{GyroY,AccX}$ is derived from the temperature sensor integral to the Y gyro sensor. $Temperature_{GyroX,AccZ,AccY}$ and $Temperature_{GyroZ}$ are derived from the temperature sensors integral to the X and Z gyro sensors respectively. Because of the high density of the circuit, the Accel temperatures are presumed to be the same as that of the Gyro that has the closest proximity. All temperature sensors are sampled using a 12 bit A/D converter with a 3.0V reference.

$Temperature_{Mag}$ has the highest absolute accuracy. To convert $Temperature_{Mag}$ to Centigrade units, carry out the following conversion:

$$Temperature = -1481.96 + \sqrt{2.1962 \times 10^{6} + \frac{(1.8639 - (\frac{3.0 \times T_{Mag}}{4096})}{3.88 \times 10^{-6}}}$$

Equation 15

Alternatively, a less accurate, but easier $Temperature_{Mag}$ conversion for the temperature range of -40 to +85 can be made using the following equation:

$$Temperature = -\frac{(\frac{3 \times T_{Mag}}{4096}) - 1.8583}{11.67 \times 10^{-3}}$$

Equation 16

Note: The temperature conversion formula does not apply to the Gyro derived temperature values. The Gyro derived temperature outputs are optimized for repeatability rather than absolute accuracy. The values returned for the Gyro derived temperatures should be regarded as dimensionless relative measurement rather than an absolute temperature measurement.

Orientation Conversion Formulas

The 3DM-GX3® can output orientation information in three different forms: Euler Angles, a 3x3 rotation matrix (also called a coordinate transformation matrix) or a quaternion. 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.

The 3DM-GX3® fundamentally calculates orientation in the form of a rotation matrix, *M.*

$$M = \begin{bmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{bmatrix}$$

M satisfies the vector equation,

$VL = M \cdot VE$ where: *VE* is a vector expressed in the Earth-Fixed coordinate system.
VL is the same vector expressed in the 3DM-GX3® sensor's local coordinate system.

M to Euler

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.

$$Pitch = \theta = \arcsin(-M13)$$
$$Roll = \phi = \arctan(M23 / M33)$$
$$Yaw = \psi = \arctan(M12 / M11)$$

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.

Euler to M

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

M to Q

When the user requests orientation from the 3DM-G in the form of Quaternions, *Q*, these are derived from the rotation matrix.

$$Q = \begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix}$$ where *q0* is the scalar component, and *q*1, *q*2, *q*3 are the vector components.

The quaternion satisfies the quaternion product equation

$VL = Q \cdot VE \cdot Q^{*}$ where: *VE* is a vector expressed in the Earth-Fixed coordinate system.
VL is the same vector expressed in the 3DM-G's local coordinate system.

When converting from a rotation matrix to quaternions, there are several different formulations that can be used. In practice, the numerical resolution of these may be quite different depending on the orientation. Therefore, it is recommended that a test be made of which formulation will yield the most favorable results. This can be done in the following manner:

$$test1 = M11 + M22 + M33$$
$$test2 = M11 - M22 - M33$$
$$test3 = -M11 + M22 - M33$$
$$test4 = -M11 - M22 + M33$$

max = largest of (*test*1, *test*2, *test*3, *test*4) if *max* = *test*1 then carry out:

$$S = 2\sqrt{1 + M11 + M22 + M33}$$

$$\begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} S/4 \\ (M23 - M32)/S \\ (M31 - M13)/S \\ (M12 - M21)/S \end{bmatrix}$$

if *max* = *test*2 then carry out:

$$S = 2\sqrt{1 + M11 - M22 - M33}$$

$$\begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} (M32 - M23)/S \\ -S/4 \\ -(M21 + M12)/S \\ -(M13 + M31)/S \end{bmatrix}$$

if *max* = *test*3 then carry out:

$$S = 2\sqrt{1 - M11 + M22 - M33}$$

$$\begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} (M13 - M31)/S \\ -(M21 + M12)/S \\ -S/4 \\ -(M32 + M23)/S \end{bmatrix}$$

if *max* = *test*4 then carry out:

$$S = 2\sqrt{1 - M11 - M22 + M33}$$

$$\begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} (M21 - M12)/S \\ -(M13 + M31)/S \\ -(M32 + M23)/S \\ -S/4 \end{bmatrix}$$

Note that $\begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix} = \begin{bmatrix} -q0 \\ -q1 \\ -q2 \\ -q3 \end{bmatrix}$

It is conventional to select the signs such that $q0$ is positive.

Q to M

To convert from a know quaternion to a rotation matrix, the following can be used:

$$M = 2\begin{bmatrix} q0^2 - 1/2 + q1^2 & q1q2 + q0q3 & q1q3 - q0q2 \\ q1q2 - q0q3 & q0^2 - 1/2 + q2^2 & q2q3 + q0q1 \\ q1q3 + q0q2 & q2q3 - q0q1 & q0^2 - 1/2 + q3^2 \end{bmatrix}$$

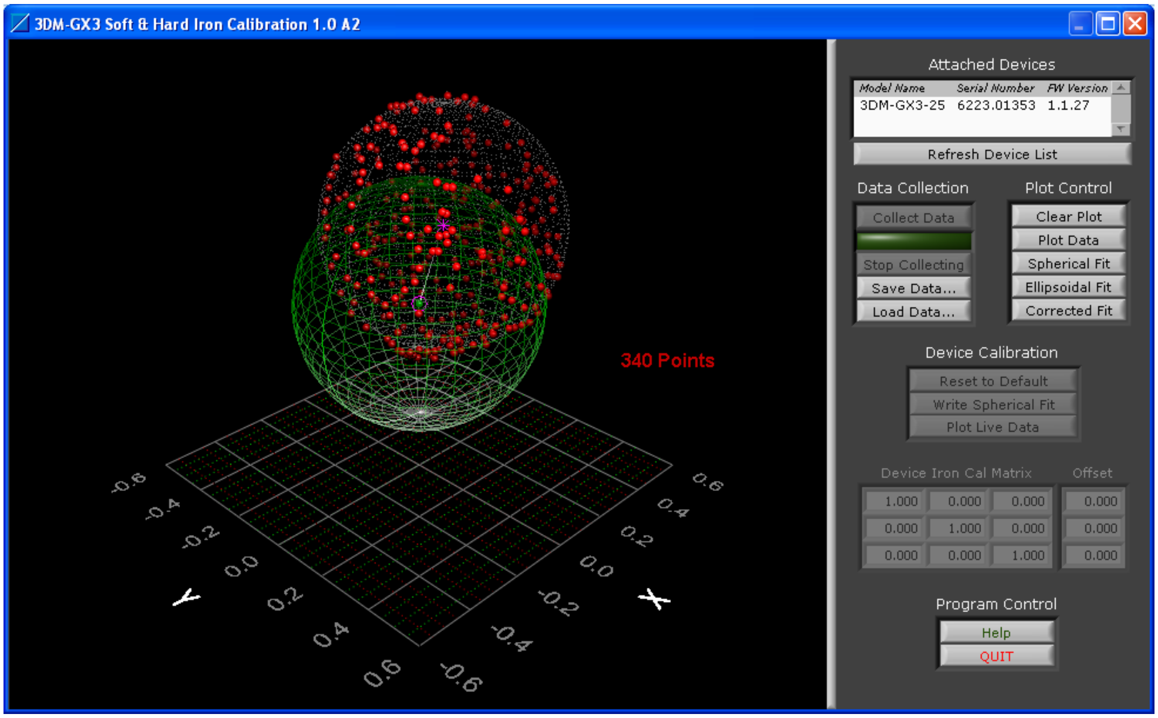
Q to Euler

To convert between Euler Angles and Quaternions, the appropriate conversion to a rotation matrix can be made as an intermediate step.

Magnetometer Iron Calibration

Iron Calibration is a necessary procedure for best magnetometer performance once the 3DM-GX3® is installed in its target application. MicroStrain provides an application "3DM-GX3 Soft and Hard Iron Calibration" that graphically and dynamically allows you to collect data and calibrate the sensor while actively installed in your application.

Iron calibration compensates for magnetic field *distortion* and *bias* that cause an ellipsoidal distortion and offset of the magnetometer readings. These are commonly referred to as *Soft Iron* distortion and *Hard Iron* offset and hence calibration is often referred to as *Soft and Hard Iron Calibration*. The 3DM-GX3 can compensate for these effects as long as they are not extreme; however it has to be calibrated after the device is installed in its final location. This means that iron calibration must be done by the user. Figure 3 shows a screen shot from the "3DM-GX3 Soft and Hard Iron Calibration" application demonstrating the effect of a steel bolt.



Model Name	Serial Number	FW Version
3DM-GX3-25	6223.01353	1.1.27

Device Iron Cal Matrix			Offset
1.000	0.000	0.000	0.000
0.000	1.000	0.000	0.000
0.000	0.000	1.000	0.000

Figure 3

Red dots are the magnetometer readings (with no iron calibration) when a steel bolt is mounted adjacent to the 3DM-GX3. The green sphere represents corrected magnetometer readings after iron calibration.

For best results, the 3DM-GX3 should be mounted with non-ferrous and non-magnetic materials and be kept as far away from features that may be ferrous or magnetic such as steel frames, bolts, motors, etc.

End of Document