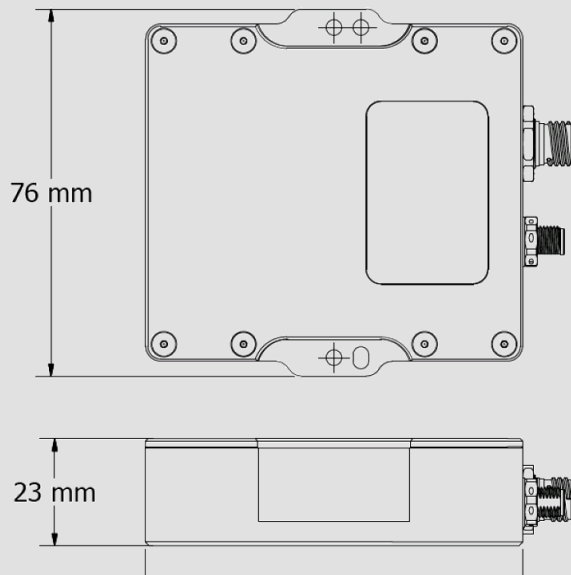


3DM-RQ1™-45

Data Communications Protocol





©2013 LORD Corporation
MicroStrain[®] Sensing Systems
459 Hurricane Lane
Suite 102
Williston, VT 05495
United States of America
Phone: 802-862-6629
Fax: 802-863-4093

www.microstrain.com
sensing_support@LORD.com

REVISED: June 25, 2014

Table of Contents

Table of Contents.....	3
3DM-RQ1 API.....	9
API Introduction.....	9
Command and Data Summary.....	10
Commands	10
Base Command Set (0x01).....	10
3DM Command Set (0x0C).....	10
Estimation Filter Command Set (0x0D)	11
System Command Set (0x7F).....	11
Data.....	12
IMU Data Set (set 0x80)	12
GPS Data Set (set 0x81).....	12
Estimation Filter Data Set (set 0x82).....	12
Basic Programming.....	14
MIP Packet Overview.....	14
Command Overview	16
Example “Ping” Command Packet:.....	16
Example “Ping” Reply Packet:.....	16
Data Overview	17
Example Data Packet:	17
Example Setup Sequence	19
Continuous Data Example Command Sequence	19
Polling Data Example Sequence	24
Parsing Incoming Packets	26
Multiple Rate Data	27
Data Synchronicity.....	28
Communications Bandwidth Management.....	29
UART Bandwidth Calculation.....	29
Command Reference.....	31

Base Commands.....	31
Ping (0x01, 0x01)	31
Set To Idle (0x01, 0x02).....	32
Resume (0x01, 0x06).....	33
Get Device Information (0x01, 0x03).....	34
Get Device Descriptor Sets (0x01, 0x04).....	35
Device Built-In Test (0x01, 0x05).....	36
Device Reset (0x01, 0x7E).....	37
3DM Commands.....	38
Poll IMU Data (0x0C, 0x01)	38
Poll GPS Data (0x0C, 0x02).....	39
Poll Estimation Filter Data (0x0C, 0x03).....	40
Get IMU Data Base Rate (0x0C, 0x06).....	42
Get GPS Data Base Rate (0x0C, 0x07)	43
Get Estimation Filter Data Base Rate (0x0C, 0x0B)	44
IMU Message Format (0x0C, 0x08).....	45
GPS Message Format (0x0C, 0x09)	47
Estimation Filter Message Format (0x0C, 0x0A)	49
Enable/Disable Continuous Data Stream (0x0C, 0x11).....	51
Device Startup Settings (0x0C, 0x30).....	53
Accel Bias (0x0C, 0x37).....	54
Gyro Bias (0x0C, 0x38).....	55
Capture Gyro Bias (0x0C, 0x39)	56
Magnetometer Hard Iron Offset (0x0C, 0x3A).....	58
Magnetometer Soft Iron Matrix (0x0C, 0x3B)	60
Coning and Sculling Enable (0x0C, 0x3E)	62
UART BAUD Rate (0x0C, 0x40).....	63
Advanced Low-Pass Filter Settings (0x0C, 0x50)	64
Device Status (0x0C, 0x64)	66
Raw RTCM 2.3 Message (0x0C, 0x20)	69
Estimation Filter Commands	70
Reset Filter (0x0D, 0x01)	70

Set Initial Attitude (0x0D, 0x02).....	71
Set Initial Heading (0x0D, 0x03).....	72
Vehicle Dynamics Mode (0x0D, 0x10)	73
Sensor to Vehicle Frame Transformation (0x0D, 0x11)	74
Sensor to Vehicle Frame Offset (0x0D, 0x12).....	76
Antenna Offset (0x0D, 0x13).....	77
Estimation Control Flags (0x0D, 0x14).....	78
GPS Source Control (0x0D, 0x15)	79
External GPS Update (0x0D, 0x16).....	80
External Heading Update (0x0D, 0x17)	82
Heading Update Control (0x0D, 0x18)	84
Auto-Initialization Control (0x0D, 0x19).....	85
Accelerometer White Noise Standard Deviation (0x0D, 0x1A).....	86
Gyroscope White Noise Standard Deviation (0x0D, 0x1B).....	88
Magnetometer White Noise Standard Deviation (0x0D, 0x42)	89
Accelerometer Bias Model Parameters (0x0D, 0x1C).....	91
Gyroscope Bias Model Parameters (0x0D, 0x1D).....	92
Zero Velocity Update (ZUPT) Control (0x0D, 0x1E)	93
External Heading Update with Timestamp (0x0D, 0x1F).....	95
Zero Angular Rate Update Control (0x0D, 0x20).....	96
Tare Orientation (0x0D, 0x21)	97
Commanded Zero-Velocity Update (0x0D, 0x22).....	98
Commanded Zero-Angular Rate Update (0x0D, 0x23)	99
Declination Source (0x0D, 0x43).....	100
Magnetometer Magnitude Error Adaptive Measurement (0x0D, 0x45).....	102
Magnetometer Dip Angle Error Adaptive Measurement (0x0D, 0x46).....	104
System Commands.....	106
Communication Mode (0x7F, 0x10).....	106
Hardware Control (0x7F, 0x11).....	107
Error Codes.....	109
Data Reference.....	110
IMU Data.....	110

Scaled Accelerometer Vector (0x80, 0x04)	110
Scaled Gyro Vector (0x80, 0x05)	110
Scaled Magnetometer Vector (0x80, 0x06)	111
Scaled Ambient Pressure (0x80, 0x17).....	111
Delta Theta Vector (0x80, 0x07).....	112
Delta Velocity Vector (0x80, 0x08)	112
GPS Correlation Timestamp (0x80, 0x12).....	113
GPS Data	114
LLH Position (0x81, 0x03)	114
ECEF Position (0x81, 0x04).....	115
NED Velocity (0x81, 0x05)	115
ECEF Velocity(0x81, 0x06).....	116
DOP Data (0x81, 0x07).....	117
UTC Time (0x81, 0x08).....	118
GPS Time (0x81, 0x09).....	118
Clock Information (0x81, 0x0A).....	119
GPS Fix Information (0x81, 0x0B).....	120
Space Vehicle Information (0x81, 0x0C).....	121
Hardware Status (0x81, 0x0D)	122
DGPS Information (0x81, 0x0E)	123
DGPS Channel Status (0x81, 0x0F).....	124
Estimation Filter Data.....	125
Filter Status (0x82, 0x10)	125
GPS Timestamp (0x82, 0x11).....	126
LLH Position (0x82, 0x01).....	127
NED Velocity (0x82, 0x02)	127
Orientation, Quaternion (0x82, 0x03).....	128
Orientation, Matrix (0x82, 0x04)	129
Orientation, Euler Angles (0x82, 0x05)	130
Gyro Bias (0x82, 0x06).....	131
Accel Bias (0x82, 0x07)	131
LLH Position Uncertainty (0x82, 0x08).....	132

NED Velocity Uncertainty (0x82, 0x09).....	132
Attitude Uncertainty, Euler Angles (0x82, 0x0A)	133
Gyro Bias Uncertainty (0x82, 0x0B)	134
Accel Bias Uncertainty (0x82, 0x0C).....	135
Linear Acceleration (0x82, 0x0D).....	135
Compensated Acceleration (0x82, 0x1C)	136
Compensated Angular Rate (0x82, 0x0E).....	137
WGS84 Local Gravity Magnitude (0x82, 0x0F)	137
Attitude Uncertainty, Quaternion Elements (0x82, 0x12).....	138
Gravity Vector (0x82, 0x13)	139
Heading Update Source State (0x82, 0x14).....	139
Magnetic Model Solution (0x82, 0x15)	140
Gyro Scale Factor (0x82, 0x16)	141
Accel Scale Factor (0x82, 0x17).....	141
Gyro Scale Factor Uncertainty (0x82, 0x18)	142
Accel Scale Factor Uncertainty (0x82, 0x19).....	143
Standard Atmosphere Model (0x82, 0x20).....	143
Pressure Altitude (0x82, 0x21)	144
GPS Antenna Offset Correction (0x82, 0x30).....	145
GPS Antenna Offset Correction Uncertainty (0x82, 0x31).....	145
MIP Packet Reference	147
Structure	147
Payload Length Range.....	147
Checksum Range.....	148
16-bit Fletcher Checksum Algorithm (C language)	148
Advanced Programming.....	149
Multiple Commands in a Single Packet	149
Direct Modes	150
Internal Diagnostic Functions.....	150
3DM-RQ1-45 Internal Diagnostic Commands	150
Handling High Rate Data.....	151
Runaway latency	151

Dropped packets..... 151
Creating Fixed Data Packet Format 151
Advanced Programming Models 152

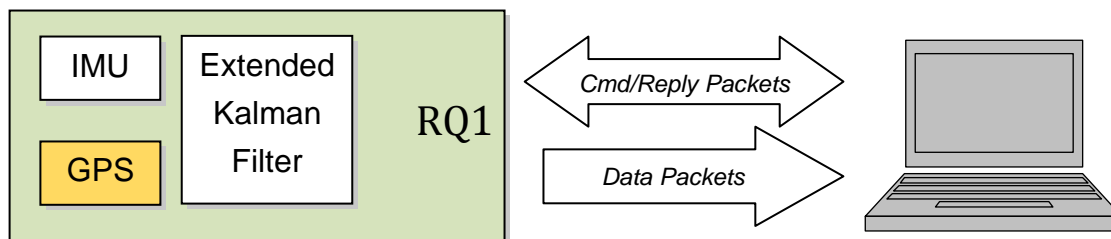
3DM-RQ1 API

API Introduction

The 3DM-RQ1 programming interface is comprised of a compact set of setup and control commands and a very flexible user-configurable data output format. The commands and data are divided into 4 command sets and 3 data sets corresponding to the internal architecture of the device. The four command sets consist of a set of “Base” commands (a set that is common across many types of devices), a set of unified “3DM” (3D Motion) commands that are specific to the MicroStrain inertial product line, a set of “Estimation Filter” commands that are specific to MicroStrain navigation and advanced AHRS devices, and a set of “System” commands that are specific to sensor systems comprised of more than one internal sensor block. The three data sets represent the three types of data that the RQ1 is capable of producing: “IMU” (Inertial Measurement Unit) data, “GPS” (Global Positioning Sensor) data, and “Estimation Filter” (Position, Velocity, and Attitude) data. The type of estimation filter used in the RQ1 is an Extended Kalman Filter (EKF).

Base commands	<i>Ping, Idle, Resume, Get ID Strings, etc.</i>
3DM commands	<i>Poll IMU Data, Poll GPS Data, etc.</i>
Estimation Filter commands	<i>Reset Filter, Sensor to Vehicle Frame Transformation, etc.</i>
System commands	<i>Switch Communications Mode, etc.</i>
IMU data	<i>Acceleration Vector, Gyro Vector, etc.</i>
GPS data	<i>Latitude, Longitude, UTC, Satellites in view, etc.</i>
Estimation Filter data	<i>Position, Velocity, Attitude, Acceleration Estimates, etc.</i>

The protocol is packet based. All commands, replies, and data are sent and received as fields in a message packet. Commands are all confirmed with an ack/nack (with a few exceptions). The packets have a descriptor type field based on their contents, so it is easy to identify if a packet contains commands, replies, IMU data, GPS data, or Estimation Filter data.



The 3DM-RQ1 has an advanced mode switch that allows the device to switch into direct “IMU” or “GPS” mode. In those modes, the device responds to the native protocols of the 3DM-RQ1 IMU or the u-blox6 GPS devices which are imbedded in the 3DM-RQ1. These modes can be used to access advanced or specialized features of these devices (see the [Advanced Programming](#) section).

Command and Data Summary

Below is a summary of the commands and data available in the programming interface. Commands and data are denoted by two values. The first value denotes the “descriptor set” that the command or data belongs to (Base command, 3DM command, Estimation Filter Command, IMU data, GPS data, or Estimation Filter data) and the second value denotes the unique command or data “descriptor” in that set. The pair of values constitutes a “full descriptor”.

Commands

Base Command Set (0x01)

- [Ping](#) (0x01, 0x01)
- [Set To Idle](#) (0x01, 0x02)
- [Get Device Information](#) (0x01, 0x03)
- [Get Device Descriptor Sets](#) (0x01, 0x04)
- [Device Built-In Test \(BIT\)](#) (0x01, 0x05)
- [Resume](#) (0x01, 0x06)
- [Device Reset](#) (0x01, 0x7E)

3DM Command Set (0x0C)

- [Poll IMU Data](#) (0x0C, 0x01)
- [Poll GPS Data](#) (0x0C, 0x02)
- [Poll Estimation Filter Data](#) (0x0C, 0x03)
- [Get IMU Data Rate Base](#) (0x0C, 0x06)
- [Get GPS Data Rate Base](#) (0x0C, 0x07)
- [Get Estimation Filter Data Rate Base](#) (0x0C, 0x0B)
- [IMU Message Format](#) (0x0C, 0x08)
- [GPS Message Format](#) (0x0C, 0x09)
- [Estimation Filter Message Format](#) (0x0C, 0x0A)
- [Enable/Disable Device Continuous Data Stream](#) (0x0C, 0x11)
- [Device Startup Settings](#) (0x0C, 0x30)
- [Accel Bias](#) (0x0C, 0x37)
- [Gyro Bias](#) (0x0C, 0x38)
- [Capture Gyro Bias](#) (0x0C, 0x39)
- [Magnetometer Hard Iron Offset](#) (0x0C, 0x3A)
- [Magnetometer Soft Iron Matrix](#) (0x0C, 0x3B)
- [Coning and Sculling Enable](#) (0x0C, 0x3E)
- [Change UART BAUD rate](#) (0x0C, 0x40)
- [Advanced Low-Pass Filter Settings](#) (0x0C, 0x50)
- [Device Status*](#) (0x0C, 0x64)
- [Raw RTCM 2.3 Message](#) (0x0C, 0x20)

Estimation Filter Command Set (0x0D)

• Reset Filter	(0x0D, 0x01)
• Set Initial Attitude	(0x0D, 0x02)
• Set Initial Heading	(0x0D, 0x03)
• Vehicle Dynamics Mode	(0x0D, 0x10)
• Sensor to Vehicle Frame Transformation	(0x0D, 0x11)
• Sensor to Vehicle Frame Offset	(0x0D, 0x12)
• Antenna Offset	(0x0D, 0x13)
• Estimation Control	(0x0D, 0x14)
• GPS Source Control	(0x0D, 0x15)
• External GPS Update	(0x0D, 0x16)
• External Heading Update	(0x0D, 0x17)
• Heading Update Control	(0x0D, 0x18)
• Auto-Initialization Control	(0x0D, 0x19)
• Accelerometer White Noise Standard Deviation	(0x0D, 0x1A)
• Gyroscope White Noise Standard Deviation	(0x0D, 0x1B)
• Magnetometer White Noise Standard Deviation	(0x0D, 0x42)
• Accelerometer Bias Model Parameters	(0x0D, 0x1C)
• Gyroscope Bias Model Parameters	(0x0D, 0x1D)
• Zero-Velocity Update Control	(0x0D, 0x1E)
• External Heading Update with Timestamp	(0x0D, 0x1F)
• Angular Rate Zero Update Control	(0x0D, 0x20)
• Tare Orientation	(0x0D, 0x21)
• Commanded ZUPT	(0x0D, 0x22)
• Commanded Zero-Angular Rate Update	(0x0D, 0x23)
• Declination Source	(0x0D, 0x43)
• Magnetometer Magnitude Error Adaptive Measurement	(0x0D, 0x45)
• Magnetometer Dip Angle Error Adaptive Measurement	(0x0D, 0x46)

System Command Set (0x7F)

• Communication Mode*	(0x7F, 0x10)
• Hardware Control	(0x7F, 0x11)

*Advanced Commands

Data

IMU Data Set (set 0x80)

- [Scaled Accelerometer Vector](#) (0x80, 0x04)
- [Scaled Gyro Vector](#) (0x80, 0x05)
- [Scaled Magnetometer Vector](#) (0x80, 0x06)
- [Scaled Ambient Pressure](#) (0x80, 0x17)
- [Delta Theta Vector](#) (0x80, 0x07)
- [Delta Velocity Vector](#) (0x80, 0x08)
- [GPS Correlated Timestamp](#) (0x80, 0x12)

GPS Data Set (set 0x81)

- [LLH Position](#) (0x81, 0x03)
- [ECEF Position](#) (0x81, 0x04)
- [NED Velocity](#) (0x81, 0x05)
- [ECEF Velocity](#) (0x81, 0x06)
- [Dilution of Precision \(DOP\)](#) (0x81, 0x07)
- [UTC Time](#) (0x81, 0x08)
- [GPS Time](#) (0x81, 0x09)
- [Clock Information](#) (0x81, 0x0A)
- [GPS Fix Information](#) (0x81, 0x0B)
- [Space-Vehicle Information \(SVI\)](#) (0x81, 0x0C)
- [Hardware Status](#) (0x81, 0x0D)
- [DGPS Information](#) (0x81, 0x0E)
- [DGPS Channel Status](#) (0x81, 0x0F)

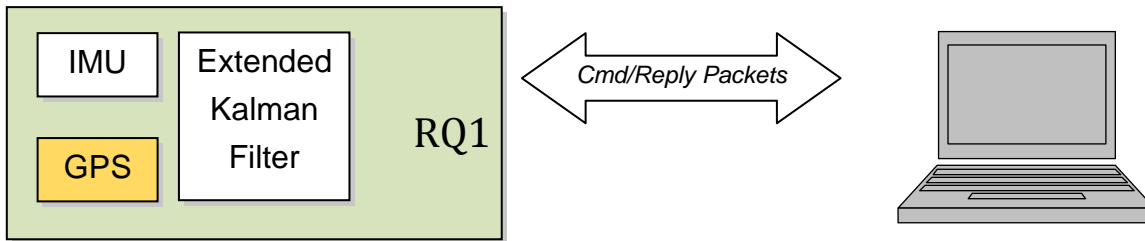
Estimation Filter Data Set (set 0x82)

- [Filter Status](#) (0x82, 0x10)
- [GPS Timestamp](#) (0x82, 0x11)
- [LLH Position](#) (0x82, 0x01)
- [NED Velocity](#) (0x82, 0x02)
- [Orientation, Quaternion](#) (0x82, 0x03)
- [Orientation, Matrix](#) (0x82, 0x04)
- [Orientation, Euler Angles](#) (0x82, 0x05)
- [Gyro Bias](#) (0x82, 0x06)
- [Accel Bias](#) (0x82, 0x07)
- [LLH Position Uncertainty](#) (0x82, 0x08)
- [NED Velocity Uncertainty](#) (0x82, 0x09)
- [Attitude Uncertainty, Euler Angles](#) (0x82, 0x0A)
- [Gyro Bias Uncertainty](#) (0x82, 0x0B)
- [Accel Bias Uncertainty](#) (0x82, 0x0C)
- [Linear Acceleration](#) (0x82, 0x0D)
- [Compensated Angular Rate](#) (0x82, 0x0E)
- [WGS84 Local Gravity Magnitude](#) (0x82, 0x0F)

- [Attitude Uncertainty, Quaternion Elements](#) (0x82, 0x12)
- [Gravity Vector](#) (0x82, 0x13)
- [Heading Update Source State](#) (0x82, 0x14)
- [Magnetic Model Solution](#) (0x82, 0x15)
- [Gyro Scale Factor](#) (0x82, 0x16)
- [Accel Scale Factor](#) (0x82, 0x17)
- [Gyro Scale Factor Uncertainty](#) (0x82, 0x18)
- [Accel Scale Factor Uncertainty](#) (0x82, 0x19)
- [Compensated Linear Acceleration](#) (0x82, 0x1C)
- [Standard Atmosphere Model](#) (0x82, 0x20)
- [Pressure Altitude](#) (0x82, 0x21)
- [GPS Antenna Offset Correction](#) (0x82, 0x30)
- [GPS Antenna Offset Correction Uncertainty](#) (0x82, 0x31)

Basic Programming

The 3DM-RQ1 is designed to stream IMU, and GPS, and Estimation Filter data packets over a common interface as efficiently as possible. To this end, programming the device consists of a configuration stage where the data messages and data rates are configured. The configuration stage is followed by a data streaming stage where the program starts the incoming data packet stream.



In this section there is an overview of the packet, an overview of command and reply packets, an overview of how an incoming data packet is constructed, and then an example setup command sequence that can be used directly with the 3DM-RQ1 either through a COM utility or as a template for software development.

MIP Packet Overview

This is an overview of the 3DM-RQ1 packet structure. The packet structure used is the MicroStrain “MIP” packet. A reference to the general packet structure is presented in the [MIP Packet Reference](#) section. An overview of the packet is presented here.

The MIP packet “wrapper” consists of a four byte header and two byte checksum footer:

Header				Packet Payload			Checksum	
SYNC1 “u”	SYNC2 “e”	Descriptor Set byte	Payload Length byte	Field Length byte	Field Descriptor byte	Field Data	MSB	LSB
0x75	0x65	0x80	0x0E	0x0E	0x03	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0x83	0xE1

Payload Length byte. This specifies the length of the packet payload. The packet payload may contain one or more fields and thus this byte also represents the sum of the lengths of all the fields in the payload.

Descriptor Set. Descriptors are grouped into different sets. The value 0x80 identifies this packet as an AHRS data packet. Fields in this packet will be from the AHRS data descriptor set.

Start of Packet (SOP) “sync” bytes. These are the same for every MIP packet and are used to identify the start of the packet.

2 byte Fletcher checksum of all the bytes in the packet.

The packet payload section contains one or more fields. Fields have a length byte, descriptor byte, and data. The diagram below shows a packet payload with a single field.

Header				Packet Payload			Checksum	
SYNC1 "u"	SYNC2 "e"	Descriptor Set byte	Payload Length byte	Field Length byte	Field Descriptor byte	Field Data	MSB	LSB
0x75	0x65	0x80	0x0E	0x0E	0x06	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0x86	0x08

Field Length byte. This represents a count of all the bytes in the field including the length byte, descriptor byte and field data.

Descriptor byte. This byte identifies the contents of the field data. This descriptor indicates that the data is a mag vector (set: 0x80, descriptor: 0x06)

Field data. The length of the data is Field Length – 2. This data is 12 bytes long (14 – 2) and represents the floating point magnetometer vector value from the AHRS data set.

Below is an example of a packet payload with two fields (gyro vector and mag vector). Note the payload length byte of 0x1C which is the sum of the two field length bytes 0x0E + 0x0E:

Header				Packet Payload (2 fields)						Checksum	
SYNC1 "u"	SYNC2 "e"	Descriptor Set	Payload Length	Field1 Len	Field1 Descriptor	Field1 Data	Field2 Len	Field2 Descriptor	Field2 Data	MSB	LSB
0x75	0x65	0x80	0x1C	0x0E	0x05	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0x0E	0x06	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0xB1	0x1E

Command Overview

The basic command sequence begins with the host sending a command to the device. A command packet contains a field with the command value and any command arguments.

The device responds by sending a reply packet. The reply contains at minimum an ACK/NACK field. If any additional data is included in a reply, it appears as a second field in the packet.

Example “Ping” Command Packet:

Below is an example of a “Ping” command packet from the Base command set. A “Ping” command has no arguments. Its function is to determine if a device is present and responsive:

Header				Packet Payload			Checksum	
SYNC1 “u”	SYNC2 “e”	Descriptor Set byte	Payload Length byte	Field Length byte	Field Descriptor byte	Field Data	MSB	LSB
0x75	0x65	0x01	0x02	0x02	0x01	N/A	0xE0	0xC6

Copy-Paste version: “7565 0102 0201 E0C6”

The packet header has the “ue” starting sync bytes characteristic of all [MIP packets](#). The descriptor set byte (0x01) identifies the data as being from the Base command set. The length of the payload portion is 2 bytes. The payload portion of the packet consists of one field. The field starts with the length of the field which is followed by the descriptor byte (0x01) of the field. The field descriptor value is the command value. Here the descriptor identifies the command as the “Ping” command from the Base command descriptor set. There are no parameters associated with the ping command, so the field data is empty. The checksum is a two byte [Fletcher checksum](#) (see the [MIP Packet Reference](#) for instructions on how to compute a Fletcher two byte checksum).

Example “Ping” Reply Packet:

The “Ping” command will generate a reply packet from the device. The reply packet will contain an ACK/NACK field. The ACK/NACK field contains an “echo” of the command byte plus an error code. An error code of 0 is an “ACK” and a non-zero error code is a “NACK”:

Header				Packet Payload			Checksum	
SYNC1 “u”	SYNC2 “e”	Descriptor Set byte	Payload Length byte	Field Length byte	Field Descriptor byte	Field Data: 2 bytes	MSB	LSB
0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x01 Error code: 0x00	0xD5	0x6A

Copy-Paste version: “7565 0104 04F1 0100 D56A”

The packet header has the “ue” starting sync bytes characteristic of all [MIP packets](#). The descriptor set byte (0x01) identifies the payload fields as being from the Base command set. The length of the payload portion is 4 bytes. The payload portion of the packet consists of one field. The field starts with the length of the field which is followed by the descriptor byte (0xF1) of the field. The field descriptor byte identifies the reply as the “ACK/NACK” from the Base command descriptor set. The field data consists of an “echo” of the original command (0x01) followed by the error

code for the command (0x00). In this case the error is zero, so the field represents an “ACK”. Some examples of non-zero error codes that might be sent are “timeout”, “not implemented”, and “invalid parameter in command”. The checksum is a two byte [Fletcher checksum](#) (see the [MIP Packet Reference](#) for instructions on how to compute a Fletcher two byte checksum).

The ACK/NACK descriptor value (0xF1) is the same in all descriptor sets. The value belongs to a set of reserved global descriptor values.

The reply packet may have additional fields that contain information in reply to the command. For example, requesting [Device Status](#) will result in a reply packet that contains two fields in the packet payload: an ACK/NACK field and a device status information field.

Data Overview

Data packets are generated by the device. When the device is powered up, it may be configured to immediately stream data packets out to the host or it may be “idle” and waiting for a command to either start continuous data or to get data by “polling” (one data packet per request). Either way, the data packet is generated by the device in the same way.

Example Data Packet:

Below is an example of a MIP data packet which has one field that contains the scaled accelerometer vector.

Header				Packet Payload			Checksum	
SYNC1 “u”	SYNC2 “e”	Descriptor Set byte	Payload Length byte	Field Length byte	Field Descriptor byte	Field Data: Accel vector (12 bytes, 3 float – X, Y, Z)	MSB	LSB
0x75	0x65	0x80	0x0E	0x0E	0x04	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0x92	0xC0

Copy-Paste version: “7565 800E 0E04 3E7A 63A0 BB8E 3B29 7FE5 BF7F 92C0”

The packet header has the “ue” starting sync bytes characteristic of all MIP packets. The descriptor set byte (0x80) identifies the payload field as being from the IMU data set. The length of the packet payload portion is 14 bytes (0x0E). The payload portion of the packet starts with the length of the field. The field descriptor byte (0x04) identifies the field data as the scaled accelerometer vector from the IMU data descriptor set. The field data itself is three single precision floating point values of 4 bytes each (total of 12 bytes) representing the X, Y, and Z axis values of the vector. The checksum is a two byte [Fletcher checksum](#) (see the [MIP Packet Reference](#) for instructions on how to compute a Fletcher two byte checksum).

The format of the field data is fully and unambiguously specified by the descriptor. In this example, the field descriptor (0x04) specifies that the field data holds an array of three single precision IEEE-754 floating point numbers in big-endian byte order and that the values represent units of “g’s” and the order of the values is X, Y, Z vector order. Any other specification would require a different descriptor (see the [Data Reference](#) section of this manual).

Each packet can contain any combination of data quantities from the same data descriptor set (any combination of GPS data OR any combination of IMU data OR and combination of Estimation Filter data– you cannot combine GPS data, IMU data, and Estimation Filter data in the same packet).

Data polling commands generate two individual reply packets: An ACK/NACK packet and a data packet. Enable/Disable continuous data commands generate an ACK/NACK packet followed by the continuous stream of data packets.

The IMU, GPS, and Estimation Filter data packets can be set up so that each data quantity is sent at a different rate. For example, you can setup continuous data to send the accelerometer vector at 100Hz and the magnetometer vector at 5Hz. This means that packets will be sent at 100Hz and each one will have the accelerometer vector but only every 20th packet will have the magnetometer vector. This helps reduce bandwidth and buffering requirements. An example of this is given in the [IMU Message Format](#) command.

Example Setup Sequence

Setup involves a series of command/reply pairs. The example below demonstrates actual setup sequences that you can send directly to the 3DM-RQ1 either programmatically or by using a COM utility. In most cases only minor alterations will be needed to adapt these examples for your application.

Continuous Data Example Command Sequence

Most applications will operate with the 3DM-RQ1 sending a continuous data stream. In the following example, the IMU data format is set, followed by the Estimation Filter data format. GPS data will not be included as we will not be cross-checking against the navigation solution. To reduce the amount of streaming data, if present during the configuration, the device is placed into the idle state while performing the device initialization; when configuration is complete, the required data streams are enabled to bring the device out of idle mode. Finally, the configuration is saved so that it will be loaded on subsequent power-ups, eliminating the need to perform the configuration again.

Step 1: Put the Device in Idle Mode (Disabling the IMU, GPS, and Estimation Filter data-streams)

Send the “[Set To Idle](#)” command to put the device in the idle state (reply is ACK/NACK). This is not required but reduces the parsing burden during initialization and makes visual confirmation of the commands easier:

Step 1	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command Set to Idle	0x75	0x65	0x01	0x02	0x02	0x02	N/A	0xE1	0xC7
Reply ACK/NACK	0x75	0x65	0x01	0x04	0x04	0xF1	Cmd echo: 0x02 Error code: 0x00	0xD6	0x6C

Copy-Paste version of the command: “7565 0102 0202 E1C7”

Step 2: Configure the IMU data-stream format

Send a “[Set IMU Message Format](#)” command (reply is ACK/NACK). This example requests scaled gyro, scaled accelerometer, and GPS Correlation Timestamp information at 100 Hz (500Hz base rate, with a rate decimation of 5 on the 3DM-RQ1 = 100 Hz.) This will result in a single IMU data packet sent at 100 Hz containing the scaled gyro field followed by the scaled accelerometer field followed by the IMU GPS Correlation Timestamp. This is a very typical configuration for a base level of inertial data. If different rates were requested, then each packet would only contain the data quantities that fall in the same decimation frame (see the [Multiple Rate Data](#) section). If the stream was not disabled in the previous step, the IMU data would begin stream immediately.

Please note, this command will not append the requested descriptors to the current IMU data-stream configuration, it will overwrite it completely.

Step 2	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command New IMU Message Format	0x75	0x65	0x0C	0x0D	0x0D	0x08	Function: 0x01 Desc count: 0x03 1 st Descriptor: 0x04 Rate Dec: 0x0005 2 nd Descriptor: 0x05 Rate Dec: 0x0005 3 rd Descriptor: 0x12 Rate Dec: 0x0005	0x2A	0x35
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Cmd echo: 0x08 Error code: 0x00	0xE7	0xBA

Copy-Paste version of the command: "7565 0C0D 0D08 0103 0400 0505 0005 1200 0536 65"

Step 3: Configure the Estimation Filter data-stream format

The following configuration command requests the Estimated LLH Position, Estimated NED Velocity, Estimated Orientation in Quaternion form, and Filter Status at 20 Hz (500Hz base rate, with a rate decimation of 25 = 20 Hz.) This will result in a single Estimation Filter packet sent at 20 Hz containing the requested fields in the requested order. If different rates were requested, the each packet would only contain the data quantities that fall in the same data rate frame (see the [Multiple Rate Data](#) section). If the stream was not disabled in the previous step, the Estimation Filter data would begin stream immediately.

Please note, this command will not append the requested descriptors to the current Estimation Filter data-stream configuration, it will overwrite it completely.

Step 3	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command New Estimation Filter Message Format	0x75	0x65	0x0C	0x10	0x10	0x0A	Function: 0x01 Desc Count: 0x04 Est. Posdesc: 0x01 Rate dec: 0x0019 Est.Veldesc: 0x02 Rate dec: 0x0019 Est. Quatdesc: 0x03	0x3F	0x31

							Rate dec: 0x0019 Filter Statusdesc: 0x10 Rate dec: 0x0019		
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Cmd echo: 0x0A Error code: 0x00	0xE9	0xBE

Copy-Paste version of the command: "7565 0C10 100A 0104 0100 1902 0019 0300 1910 0019 8FE9"

Step 4: Save the IMU and Estimation Filter MIP Message format

To save the IMU and Estimation Filter MIP Message format, use the "Save" function selector (0x03) in the IMU and Estimation Filter Message Format commands. Below we've combined the two commands as two fields in the same packet. Notice that the two reply ACKs comes in one packet also. Alternatively, they could be sent as separate packets.

Step 4	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command field 1 Save Current IMU Message Format	0x75	0x65	0x0C	0x08	0x04	0x08	Function: 0x03 Desc count: 0x00		
Command field 2 Save Current Estimation Filter Message Format					0x04	0x0A	Function: 0x03 Desc count: 0x00	0x0E	0x31
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x08	0x04	0xF1	Cmd echo: 0x08 Error code: 0x00		
Reply field 2 ACK/NACK					0x04	0xF1	Cmd echo: 0x0A Error code: 0x00	0xEA	0x71

Copy-Paste version of the command: "7565 0C08 0408 0300 040A 0300 0E31"

Step 5: Enable the IMU and Estimation Filter data-streams

Send an "Enable/Disable Continuous Stream" command to enable the IMU and Estimation Filter continuous streams (reply is ACK). These streams may have already been enabled by default, this step is to confirm they are enabled. These streams will begin streaming data immediately.

Step 5	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command field 1 Enable Continuous IMU Message	0x75	0x65	0x0C	0x0A	0x05	0x11	Fctn: 0x01 IMU: 0x01 ON: 0x01		
Command field 2 Enable Continuous Estimation Filter Message					0x05	0x11	Fctn: 0x01 Estimation Filter: 0x03 ON: 0x01	0x24	0xCC
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x08	0x04	0xF1	Cmd echo: 0x11 Error code: 0x00		
Reply field 2 ACK/NACK					0x04	0xF1	Cmd echo: 0x11 Error code: 0x00	0xFA	0xB5

Copy-Paste version of the command: "7565 0C0A 0511 0101 0105 1101 0301 24 CC"

Step 6 (Optional): Resume the Device

Sending the "[Resume](#)" command is another method of re-enabling transmission of enabled data streams (reply is ACK/NACK).

Step 6	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command Resume	0x75	0x65	0x01	0x02	0x02	0x06	N/A	0xE5	0xCB
Reply ACK/NACK	0x75	0x65	0x01	0x04	0x04	0xF1	Cmd echo: 0x06 Error code: 0x00	0xDA	0x74

Copy-Paste version of the command: "7565 0102 0206 E5CB"

Step 7: Initialize the Filter

At this point in the set-up, the GX3-RQ1 is streaming data, but the Kalman Filter is not yet initialized. For a successful initialization to occur, the GPS must have a fix and the initial orientation must be known. The orientation may be initialized in 4 different ways: Setting all of the attitude elements manually, setting only the heading and allowing the device to determine pitch and roll, using the internal IMU solution (which requires the magnetometers) to provide the initial orientation, or via auto-initialization, which uses the chosen heading update source to initialize. In this example,

we will assume the magnetometers are available and use the IMU solution to initialize the Kalman Filter. Once the attitude is initialized and the GPS fix becomes valid, the Kalman Filter estimation will propagate:

Step 7	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command Disable Continuous	0x75	0x65	0x0D	0x06	0x06	0x04	Declination: 0.0 deg 0x00000000	0xF7	0xE9
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Cmd echo: 0x04 Error code: 0x00	0xE4	0xB8

Copy-Paste version of the command: "7565 0D06 0604 0000 0000 F7E9"

Polling Data Example Sequence

Polling for data is less efficient than processing a continuous data stream, but may be more appropriate for certain applications. The main difference from the continuous data example is the inclusion of the Poll data commands in the data loop:

Step 1: Put the Device in Idle Mode (Disabling the IMU, GPS, and Estimation Filter data-streams)

Same as continuous streaming. See [above](#).

Step 2: Configure the IMU data-stream format

Same as continuous streaming. See [above](#).

Step 3: Configure the Estimation Filter data-stream format

Same as continuous streaming. See [above](#).

Step 4: Save the IMU and Estimation Filter MIP Message format

Same as continuous streaming. See [above](#).

Step 5: Resume the Device

Same as continuous streaming step 6. See [above](#).

Step 6: Initialize the Filter

Same as continuous streaming step 7. See [above](#).

Step 7: Send individual data polling commands

Send individual [Poll IMU Data](#) and [Poll Estimation Filter Data](#) commands in your data collection loop. After the ACK/NACK is sent by the device, a single data packet will be sent according to the settings in the previous steps. Note that the ACK/NACK has the same descriptor set value as the command, but the data packet has the descriptor set value for the type of data (IMU or Estimation Filter):

Step 7	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command Poll IMU Data	0x75	0x65	0x0C	0x04	0x04	0x01	Option: 0x00 Desc Count: 0x00	0xEF	0xDA
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Cmd echo: 0x01 Error code: 0x00	0xE0	0xAC
IMU Data Packet field 1 (Gyro Vector)	0x75	0x65	0x80	0x1C	0x0E	0x04	0x3E 7A 63 A0 0xBB 8E 3B 29		

							0x7F E5 BF 7F		
IMU Data Packet field 2(Accel Vector)					0x0E	0x03	0x3E 7A 63 A0 0xBB 8E 3B 29 0x7F E5 BF 7F	0xAD	0xDC

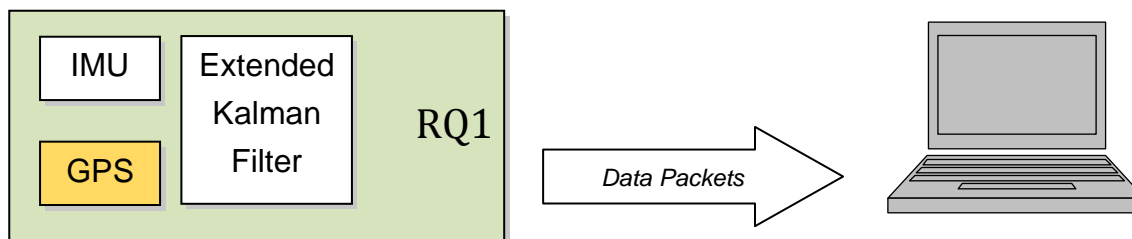
Copy-Paste version of the command: "7565 0C04 0401 0000EF DA"

You may specify the format of the data packet on a per-polling-command basis rather than using the pre-set data format (see the [Poll IMU Data](#) and [Poll Estimation Filter Data](#) sections)

The polling command has an option to suppress the ACK/NACK in order to keep the incoming stream clear of anything except data packets. Set the option byte to 0x01 for this feature.

Parsing Incoming Packets

Setup is usually the easy part of programming the 3DM-RQ1. Once you start continuous data streaming, parsing and processing the incoming data packet stream will become the primary focus. The stream of data from the IMU and Kalman Filter (Estimation Filter) are usually the dominant source of data since they come in the fastest. Polling for data may seem to be a logical solution to controlling the data flow, and this may be appropriate for some applications, but if your application requires the precise delivery of inertial data, it is often necessary to have the data stream drive the process rather than having the host try to control the data stream through polling.



The “descriptor set” qualifier in the MIP packet header is a feature that greatly aids the management of the incoming packet stream by making it easy to sort the packets into logical sub-streams and route those streams to appropriate handlers. The first step is to parse the incoming character stream into packets.

It is important to take an organized approach to parsing continuous data. The basic strategy is this: parse the incoming stream of characters for the packet starting sequence “ue” and then wait for the entire packet to come in based on the packet length byte which arrives after the “ue” and descriptor set byte. Make sure you have a timeout on your wait loop in case your stream is out of sync and the starting “ue” sequence winds up being a “ghost” sequence. If you timeout, restart the parsing with the first character after the ghost “ue”. Once the stream is in sync, it is rare that you will hit a timeout unless you have an unreliable communications link. After verifying the checksum, examine the “descriptor set” field in the header of the packet. This tells you immediately how to handle the packet.

Based on the value of the descriptor set field in the packet header, pass the packet to either a command handler (if it is a Base command or 3DM command descriptor set) or a data handler (if it is a GPS, IMU, or Estimation Filter data set). Since you know beforehand that the IMU and Estimation Filter data packets will be coming in fastest, you can tune your code to buffer or handle these packets at a high priority. Likewise, you know that the GPS packets will be coming in at a much lower rate but may have much more data to process. Again, you can tune your code to buffer or handle these slower packets appropriately. Replies to commands generally happen sequentially after a command so the incidence of these is under program control.

For multithreaded applications, it is often useful to use queues to buffer packets bound for different packet handler threads. The depth of the queue can be tuned so that no packets are dropped while waiting for their associated threads to process the packets in the queue. See [Advanced Programming Models](#) section for more information on this topic.

Once you have sorted the different packets and sent them to the proper packet handler, the packet handler may parse the packet payload fields and handle each of the fields as appropriate for the application. For simple applications, it is perfectly acceptable to have a single handler for all packet types. Likewise, it is perfectly acceptable for a single parser to handle both the packet type and the fields in the packet. The ability to sort the packets by type is just an option that simplifies the implementation of more sophisticated applications.

Multiple Rate Data

The message format commands ([IMU Message Format](#), [GPS Message Format](#), and [Estimation Filter Message Format](#)) allow you to set different data rates for different data quantities. This is a very useful feature especially for IMU data because some data, such as accelerometer and gyroscope data, usually requires higher data rates (>100Hz) than other IMU data such as Magnetometer (20Hz typical) data. The ability to send data at different rates reduces the parsing load on the user program and decreases the bandwidth requirements of the communications channel.

Multiple rate data is scheduled on a common sampling rate clock. This means that if there is more than one data rate scheduled, the schedules coincide periodically. For example, if you request Accelerometer data at 100Hz and Magnetometer data at 50Hz, the magnetometer schedule coincides with the Accelerometer schedule 50% of the time. When the schedules coincide, then the two data quantities are delivered in the same packet. In other words, in this example, you will receive data packets at 100Hz and every packet will have an accelerometer data field and EVERY OTHER packet will also include a magnetometer data field:

<i>Packet 1</i>	<i>Packet 2</i>	<i>Packet 3</i>	<i>Packet 4</i>	<i>Packet 5</i>	<i>Packet 6</i>	<i>Packet 7</i>	<i>Packet 8</i>	<i>....</i>
Accel	Accel Mag	Accel	Accel Mag	Accel	Accel Mag	Accel	Accel Mag	Accel

If a timestamp is included at 100Hz, then the timestamp will also be included in every packet in this example. It is important to note that *the data in a packet with a timestamp is always synchronous with the timestamp*. This assures that multiple rate data is always synchronous.

<i>Packet 1</i>	<i>Packet 2</i>	<i>Packet 3</i>	<i>Packet 4</i>	<i>Packet 5</i>	<i>Packet 6</i>	<i>....</i>
Accel Timestamp	Accel Mag Timestamp	Accel Timestamp	Accel Mag Timestamp	Accel Timestamp	Accel Mag Timestamp	Accel Timestamp

Data Synchronicity

Because the MIP packet allows multiple data fields to be in a single packet, it may be assumed that a single timestamp field in the packet applies to all the data in the packet. In other words, it may be assumed that all the data fields in the packet were sampled at the same time.

IMU, GPS, and Estimation Filter data are generated independently by three systems with different clocks. The importance of time is different in each system and the data they produce. The IMU data requires precise microsecond resolution and perfectly regular intervals in its timestamps. GPS data produces very precise UTC interval data but it is typically delivered in a 1 second time frame. The Kalman Filter resides on a separate processor and must derive its timing information from the two data sources.

The time base difference is one of the factors that necessitate separation of the GPS, IMU, and Estimation Filter data into separate packets. Conversely, the common time base of the different data quantities within one system is what allows grouping multiple data quantities into a single packet with a common timestamp. In other words, IMU data is always grouped with a timestamp generated from the IMU time base, and GPS data is always grouped with a timestamp from the GPS time base, etc.

All data streams (IMU, GPS, and Estimation Filter) on the 3DM-RQ1 output a “GPS Time”-formatted timestamp. This timestamp is synchronized between the 3 devices using the GPS 1PPS hardware beacon. This allows a precise common time base for all data. Due to the differences in clocks on each device, the period between two consecutive timestamp values may not be constant; this occurs because periodic corrections are applied to the IMU and Estimation Filter timestamps when the GPS 1PPS signal is asserted.

Communications Bandwidth Management

Because of the large amount and variety of data that is available from the 3DM-RQ1, it is quite easy to overdrive the bandwidth of the communications channel. This can result in dropped packets. The 3DM-RQ1 does not do analysis of the bandwidth requirements for any given output data configuration, it will simply drop a packet if its internal serial buffer is being filled faster than it is being emptied. It is up to the programmer to analyze the size of the data packets requested and the available bandwidth of the communications channel. Often the best way to determine this is empirically by trying different settings and watching for dropped packets. Below are some guidelines on how to determine maximum bandwidth for your application.

UART Bandwidth Calculation

Below is an equation for the maximum theoretical UART BAUD rate for a given message configuration. Although it is possible to calculate the approximate bandwidth required for a given setup, there is no guarantee that the system can support that setup due to internal processing delays. The best approach is to try a setting based on an initial estimate and watch for dropped packets. If there are dropped packets, increase the BAUD rate, reduce the data rate, or decrease the size or number of packets.

$$n(k \times f_{mr}) + n \sum (S_f \times f_{dr})$$

Where

$$\begin{aligned} S_f &= \text{Size of data field in bytes} \\ f_{dr} &= \text{field data rate in Hz} \\ f_{mr} &= \text{maximum data rate in Hz} \\ n &= \text{size of UART word} = 10\text{bits} \\ k &= \text{Size of MIP wrapper} = 6\text{ bytes} \end{aligned}$$

which becomes

$$60f_{mr} + 10 \sum (S_f \times f_{dr})$$

Example:

For an IMU message format of Accelerometer Vector (14 byte data field) + Internal Timestamp (6 byte data field), both at 100 Hz, the theoretical minimum BAUD rate would be:

$$\begin{aligned} &= 60 \times 100 + 10((14 \times 100) + (6 \times 100)) \\ &= 26000 \text{ BAUD} \end{aligned}$$

In practice, if you set the BAUD rate to 115200 the packets come through without any packet drops. If you set the BAUD rate to the next available lower rate of 19200, which is lower than the calculated minimum, you get

regular packet drops. The only way to determine a packet drop is by observing a timestamp in sequential packets. The interval should not change from packet to packet. If it does change then packets were dropped.

Command Reference

Base Commands

The Base command set is common to many MicroStrain devices. With the Base command set it is possible to identify many properties and do basic functions on a device even if you do not recognize its specialized functionality or data. The commands work the same way on all devices that implement this set.

Ping (0x01, 0x01)

Description	Send a “Ping” command								
Notes	Device responds with ACK/NACK packet if present.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x01			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, non-zero:NACK)				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Ping</i>	0x75	0x65	0x01	0x02	0x02	0x01		0xE0	0xC6
<i>Reply ACK/NACK</i>	0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x01 Error code: 0x00	0xD5	0x6A

Copy-Paste version of the command: “7565 0102 0201 E0C6”

Set To Idle (0x01, 0x02)

Description	Place device into idle mode.								
Notes	Command has no parameters. Device responds with ACK if successfully placed in idle mode. This command will suspend streaming (if enabled) or wake the device from sleep (if sleeping) to allow it to respond to status and setup commands. You may restore the device mode by issuing the Resume command.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x02			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, non-zero:NACK)				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Set To Idle</i>	0x75	0x65	0x01	0x02	0x02	0x02		0xE1	0xC7
<i>Reply ACK/NACK</i>	0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x02 Error code: 0x00	0xD6	0x6C

Copy-Paste version of the command: "7565 0102 0202 E1C7"

Resume (0x01, 0x06)

Description	Place device back into the mode it was in before issuing the Set To Idle command. If the Set To Idle command was not issued, then the device is placed in default mode.								
Notes	Command has no parameters. Device responds with ACK if stream successfully enabled.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x06			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0: ACK, non-zero: NACK)				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Set To Idle</i>	0x75	0x65	0x01	0x02	0x02	0x06		0xE5	0xCB
<i>Reply ACK/NACK</i>	0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x06 Error code: 0x00	0xDA	0x74

Copy-Paste version of the command: "7565 0102 0206 E5CB"

Get Device Information (0x01, 0x03)

Description	Get the device ID strings and firmware version								
Notes	Reply has two fields: "ACK/NACK" and "Device Info Field"								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>						
<i>Command</i>	0x02	0x03	N/A						
<i>Replyfield 1 ACK/NACK</i>	0x04	0xF1	U8 – echo the command byte U8 – error code (0: ACK, non-zero: NACK)						
<i>Reply field 2 Device Info Field</i>	0x52	0x81	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>			
			0	Firmware Version	U16	N/A			
			2	Model Name	String(16)	N/A			
			18	Model Number	String(16)	N/A			
			34	Serial Number	String(16)	N/A			
			50	Lot Number	String(16)	N/A			
			66	Device Options	String(16)	N/A			
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Get Device Info</i>	0x75	0x65	0x01	0x02	0x02	0x03		0xE2	0xC8
<i>ReplyField 1 ACK/NACK</i>	0x75	0x65	0x01	0x58	0x04	0xF1	Command echo: 0x03 Error code: 0x00		
<i>Reply Field 2 Device Info Field</i>					0x54	0x81	FW Version: 0x05FE " 3DM-RQ1" " 6232-4270" " 6232-00122" " " " 5g, 150d/s"	0x##	0x##

Copy-Paste version of the command: "7565 0102 0203 E2C8"

Get Device Descriptor Sets (0x01, 0x04)

Description	Get the set of descriptors that this device supports								
Notes	Reply has two fields: “ACK/NACK” and “Descriptors”. The “Descriptors” field is an array of 16 bit values. The MSB specifies the descriptor set and the LSB specifies the descriptor.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x04			N/A				
<i>Replyfield 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0: ACK, non-zero: NACK)				
<i>Reply field 2 Array of Descriptors</i>	2 x <Number of descriptors> + 2	0x82			<i>Binary Offset</i>	<i>Description</i>		<i>Data Type</i>	
					0	MSB: Descriptor Set LSB: Descriptor		U16	
					1	MSB: Descriptor Set LSB: Descriptor		U16	
					...	<etc>		...	
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Get Device Info</i>	0x75	0x65	0x01	0x02	0x02	0x04		0xE3	0xC9
<i>ReplyField 1 ACK/NACK</i>	0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x04 Error code: 0x00		
<i>Reply Field 2 Array of Descriptors</i>					<n*2>	0x82	0x0101 0x0102 0x0103 ... 0x0C01 0x0C02 ... <i>n</i> th descriptor: 0x0C72	0x##	0x##

Copy-Paste version of the command: “7565 0102 0204 E3C9”

Device Built-In Test (0x01, 0x05)

<p>Description</p>	<p>Run the device Built-In Test (BIT). The Built-In Test command always returns a 32 bit value. A value of 0 means that all tests passed. A non-zero value indicates that not all tests passed. The failure flags are device dependent. The flags for the 3DM-RQ1 are defined below. THE RQ1 MUST BE STATIONARY FOR THIS TEST.</p>																																																										
<p>Notes</p>	<p>3DM-RQ1 BIT Error Flags:</p> <table border="1" data-bbox="375 596 1403 1436"> <thead> <tr> <th>Byte</th> <th>Byte 1 (LSB)</th> <th>Byte 2</th> <th>Byte 3</th> <th>Byte 4 (MSB)</th> </tr> </thead> <tbody> <tr> <td>Device</td> <td>AP-1 Board</td> <td>Sensors</td> <td>GPS</td> <td>Navigation Algorithm</td> </tr> <tr> <td>Bit 1 (LSB)</td> <td>WDT Reset (Latching, Reset after first commanded BIT)</td> <td>IMU Communication Fault</td> <td>GPS Power Fault</td> <td>Solution Fault</td> </tr> <tr> <td>Bit 2</td> <td>Input Voltage Fault</td> <td>Reserved</td> <td>GPS Communication Fault</td> <td>Reserved</td> </tr> <tr> <td>Bit 3</td> <td>System Voltage Fault</td> <td>Reserved</td> <td>GPS Solution Fault</td> <td>Reserved</td> </tr> <tr> <td>Bit 4</td> <td>Temperature Sensor Fault</td> <td>Reserved</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>Bit 5</td> <td>Main Board Heater Feedback Fault</td> <td>Magnetometer Fault (if applicable)</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>Bit 6</td> <td>Main Board Cold Condition Detected (Latching)</td> <td>Pressure Sensor Fault (if applicable)</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>Bit 7</td> <td>Reserved</td> <td>Reserved</td> <td>Reserved</td> <td>Reserved</td> </tr> <tr> <td>Bit 8 (MSB)</td> <td>Reserved</td> <td>Reserved</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>									Byte	Byte 1 (LSB)	Byte 2	Byte 3	Byte 4 (MSB)	Device	AP-1 Board	Sensors	GPS	Navigation Algorithm	Bit 1 (LSB)	WDT Reset (Latching, Reset after first commanded BIT)	IMU Communication Fault	GPS Power Fault	Solution Fault	Bit 2	Input Voltage Fault	Reserved	GPS Communication Fault	Reserved	Bit 3	System Voltage Fault	Reserved	GPS Solution Fault	Reserved	Bit 4	Temperature Sensor Fault	Reserved	Reserved	Reserved	Bit 5	Main Board Heater Feedback Fault	Magnetometer Fault (if applicable)	Reserved	Reserved	Bit 6	Main Board Cold Condition Detected (Latching)	Pressure Sensor Fault (if applicable)	Reserved	Reserved	Bit 7	Reserved	Reserved	Reserved	Reserved	Bit 8 (MSB)	Reserved	Reserved	Reserved	Reserved
Byte	Byte 1 (LSB)	Byte 2	Byte 3	Byte 4 (MSB)																																																							
Device	AP-1 Board	Sensors	GPS	Navigation Algorithm																																																							
Bit 1 (LSB)	WDT Reset (Latching, Reset after first commanded BIT)	IMU Communication Fault	GPS Power Fault	Solution Fault																																																							
Bit 2	Input Voltage Fault	Reserved	GPS Communication Fault	Reserved																																																							
Bit 3	System Voltage Fault	Reserved	GPS Solution Fault	Reserved																																																							
Bit 4	Temperature Sensor Fault	Reserved	Reserved	Reserved																																																							
Bit 5	Main Board Heater Feedback Fault	Magnetometer Fault (if applicable)	Reserved	Reserved																																																							
Bit 6	Main Board Cold Condition Detected (Latching)	Pressure Sensor Fault (if applicable)	Reserved	Reserved																																																							
Bit 7	Reserved	Reserved	Reserved	Reserved																																																							
Bit 8 (MSB)	Reserved	Reserved	Reserved	Reserved																																																							
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>																																																								
<p><i>Command</i></p>	<p>0x02</p>	<p>0x05</p>	<p>N/A</p>																																																								
<p><i>Reply field 1 ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>	<p>U8 – echo the command byte U8 – error code (0:ACK, non-zero: NACK)</p>																																																								
<p><i>Reply field 2 BIT Error Flags</i></p>	<p>0x06</p>	<p>0x83</p>	<p>U32 – BIT Error Flags</p>																																																								
<p>Example</p>	<p>MIP Packet Header</p>				<p>Command/Reply Fields</p>			<p>Checksum</p>																																																			
	<p><i>Sync1</i></p>	<p><i>Sync2</i></p>	<p><i>Desc</i></p>	<p><i>Payload</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>MSB</i></p>	<p><i>LSB</i></p>																																																		

			Set	Length	Length	Desc.	Data		
Command Built-In Test	0x75	0x65	0x01	0x02	0x02	0x05	N/A	0xE4	0xCA
Reply field 1 ACK/NACK	0x75	0x65	0x01	0x0A	0x04	0xF1	Echo cmd:0x05 Error code: 0x00		
Reply field 2 BIT Error Flags					0x06	0x83	BIT Error Flags: 0x00000000	0x68	0x7D

Copy-Paste version of the command: "7565 0102 0205 E4CA"

Device Reset (0x01, 0x7E)

Description	Resets the Device.								
Notes	Device responds with ACK if it recognizes the command and then immediately resets.								
Field Format	<i>Field Length</i>		<i>Field Descriptor</i>		<i>Field Data</i>				
Command	0x02		0x7E		N/A				
Reply ACK/NACK	0x04		0xF1		U8 – echo the command descriptor U8 – error code (0: ACK, non-zero: NACK)				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
Command Set Reset	0x75	0x65	0x01	0x02	0x02	0x7E	N/A	0x5D	0x43
Reply ACK/NACK	0x75	0x65	0x01	0x04	0x04	0xF1	Command echo: 0x7E Error code: 0x00	0x52	0x64

Copy-Paste version of the command: "7565 0102 027E 5D43"

3DM Commands

The 3DM command set is common to the MicroStrain Inertial sensors that support the MIP packet protocol. Because of the unified set of commands, it is easy to migrate code from one inertial sensor to another.

Poll IMU Data (0x0C, 0x01)

Description	Poll the device for an IMU message with the specified format								
Notes	<p>This function polls for an IMU message using the provided format. The resulting message will maintain the order of descriptors sent in the command and any unrecognized descriptors are ignored. If the format is not provided, the device will attempt to use the stored format (set with the Set IMU Message Format command.) If no format is provided and there is no stored format, the device will respond with a NACK. The reply packet contains an ACK/NACK field. The polled data packet is sent separately as an IMU Data packet.</p> <p><i>Possible Option Selector Values:</i> 0x00 – Normal ACK/NACK Reply. 0x01 – Suppress the ACK/NACK reply.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	4 + 3*N	0x01			U8 – Option Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 Reserved)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Examples	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Poll IMU data (use stored format)</i>	0x75	0x65	0x0C	0x04	0x04	0x01	Option: 0x00 Desc count: 0x00	0xEF	0xDA
<i>Command Poll IMU data (use specified format)</i>	0x75	0x65	0x0C	0x0A	0x0A	0x01	Option: 0x00 Desc count: 0x02 1 st Descriptor: 0x04 Reserved: 0x0000 2 nd Descriptor: 0x05 Reserved: 0x0000	0x06	0x27
<i>Reply ACK/NACK</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x01 Error code: 0x00	0xE0	0xAC

Copy-Paste versions of the commands:

Stored format: "7565 0C04 0401 0000 EFDA"

Specified format: "7565 0C0A 0A01 0002 0400 0005 0000 0627"

Poll GPS Data (0x0C, 0x02)

Description	Poll the device for a GPS message with the specified format								
Notes	<p>This function polls for a GPS message using the provided format. The resulting message will maintain the order of descriptors sent in the command and any unrecognized descriptors are ignored. If the format is not provided, the device will attempt to use the stored format (set with the Set GPS Message Format command.) If no format is provided and there is no stored format, the device will respond with a NACK. The reply packet contains an ACK/NACK field. The polled data packet is sent separately as a GPS Data packet.</p> <p><i>Possible Option Selector Values:</i> 0x00 – Normal ACK/NACK Reply. 0x01 – Suppress the ACK/NACK reply.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	4 + 3*N	0x02			U8 – Option Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 Reserved)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Examples	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Poll GPS data (use stored format)</i>	0x75	0x65	0x0C	0x04	0x04	0x02	Option: 0x00 Desc count: 0x00	0xF0	0xDD
<i>Command Poll GPS data (use specified format)</i>	0x75	0x65	0x0C	0x0A	0x0A	0x02	Option: 0x00 Desc count: 0x02 1 st Descriptor: 0x03 Reserved: 0x0000 2 nd Descriptor: 0x05 Reserved: 0x0000	0x06	0x2A
<i>Reply ACK/NACK (Data)</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x02 Error code: 0x00	0xE1	0xAE

packet is sent separately if ACK)									
-----------------------------------	--	--	--	--	--	--	--	--	--

Copy-Paste versions of the commands:

Stored format: "7565 0C04 0402 0000 F0DD"

Specified format: "7565 0C0A 0A02 0002 0300 0005 0000 062A"

Poll Estimation Filter Data (0x0C, 0x03)

Description	Poll the device for a Estimation Filter message with the specified format								
Notes	<p>This function polls for a Estimation Filter message using the provided format. The resulting message will maintain the order of descriptors sent in the command and any unrecognized descriptors are ignored. If the format is not provided, the device will attempt to use the stored format (set with the Set Estimation Filter Message Format command.) If no format is provided and there is no stored format, the device will respond with a NACK. The reply packet contains an ACK/NACK field. The polled data packet is sent separately as a Estimation Filter Data packet.</p> <p><i>Possible Option Selector Values:</i> 0x00 – Normal ACK/NACK Reply. 0x01 – Suppress the ACK/NACK reply.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	4 + 3*N	0x03			U8 – Option Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 Reserved)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Examples	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Poll Estimation Filter data (use stored format)</i>	0x75	0x65	0x0C	0x04	0x04	0x03	Option: 0x00 Desc count: 0x00	0xF1	0xE0
<i>Command Poll Estimation</i>	0x75	0x65	0x0C	0x0A	0x0A	0x03	Option: 0x00 Desc count: 0x02	0x02	0x1E

<i>Filter data (use specified format)</i>							1 st Descriptor: 0x01 Reserved: 0x0000 2 nd Descriptor: 0x02 Reserved: 0x0000		
<i>Reply ACK/NACK (Data packet is sent separately if ACK)</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	<i>Echo cmd:</i> 0x03 <i>Error code:</i> 0x00	0xE2	0xB0

Copy-Paste versions of the commands:

Stored format: "7565 0C04 0403 0000 F1E0"

Specified format: "7565 0C0A 0A03 0002 0100 0002 0000 021E"

Get IMU Data Base Rate (0x0C, 0x06)

Description	Get the base rate for the IMU data in Hz.								
Notes	Returns the value used for data rate calculations. See the IMU Message Format command.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>						
<i>Command</i>	0x02	0x06	none						
<i>Reply field 1 ACK/NACK Field</i>	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)						
<i>Reply field 2 IMU Base Rate</i>	0x04	0x83	U16 – IMU data base rate (Hz)						
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	MSB	LSB
<i>Command Get IMU Base Rate</i>	0x75	0x65	0x0C	0x02	0x02	0x06		0xF0	0xF7
<i>Reply field 1 ACK/NACK</i>	0x75	0x65	0x0C	0x08	0x04	0xF1	<i>Echo cmd: 0x06 Error code: 0x00</i>		
<i>Reply field 2 IMU Base Rate</i>					0x04	0x83	<i>Base rate (Hz): 0x01F4</i>	0x65	0xFD

Copy-Paste version of the command: "7565 0C02 0206 F0F7"

Get GPS Data Base Rate (0x0C, 0x07)

Description	Get the base rate for the GPS data in Hz.								
Notes	Returns the value used for data rate calculations. See the GPS Message Format command.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>						
<i>Command</i>	0x02	0x06	none						
<i>Reply field 1 ACK/NACK Field</i>	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)						
<i>Reply field 2 GPS Base Rate</i>	0x04	0x84	U16 – GPS data base rate (Hz)						
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	MSB	LSB
<i>Command Get GPS Base Rate</i>	0x75	0x65	0x0C	0x02	0x02	0x07		0xF1	0xF8
<i>Reply field 1 ACK/NACK</i>	0x75	0x65	0x0C	0x08	0x04	0xF1	<i>Echo cmd: 0x07 Error code: 0x00</i>		
<i>Reply field 2 GPS Base Rate</i>					0x04	0x84	<i>Base rate (Hz): 0x0004</i>	0x76	0x14

Copy-Paste version of the command: "7565 0C02 0207 F1F8"

Get Estimation Filter Data Base Rate (0x0C, 0x0B)

Description	Get the base rate for the Estimation Filter data in Hz.								
Notes	Returns the value used for data rate calculations. See the Estimation Filter Message Format command.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>						
<i>Command</i>	0x02	0x0B	none						
<i>Reply field 1 ACK/NACK Field</i>	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)						
<i>Reply field 2 Estimation Filter Base Rate</i>	0x04	0x8A	U16 – GPS data base rate (Hz)						
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	MSB	LSB
<i>Command Get GPS Base Rate</i>	0x75	0x65	0x0C	0x02	0x02	0x0B		0xF5	0xFC
<i>Reply field 1 ACK/NACK</i>	0x75	0x65	0x0C	0x08	0x04	0xF1	<i>Echo cmd: 0x0B Error code: 0x00</i>		
<i>Reply field 2 Estimation Filter Base Rate</i>					0x04	0x8A	<i>Base rate (Hz): 0x01F4</i>	0x71	0x30

Copy-Paste version of the command: "7565 0C02 020B F5FC"

IMU Message Format (0x0C, 0x08)

<p>Description</p>	<p>Set, read, or save the format of the IMU message packet. This command sets the format for the IMU data packet when in standard mode. The resulting data messages will maintain the order of descriptors sent in the command. The command has a function selector and a descriptor array as parameters.</p>								
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p>The rate decimation field is calculated as follows for IMU messages:</p> <p style="text-align: center;">Data Rate = IMU Base Rate / Rate Decimation</p> <p>The device checks that all descriptors are valid prior to executing this command. If any of the descriptors are invalid for the IMU descriptor set, a NACK will be returned and the message format will be unchanged. The descriptor array only needs to be provided if the function selector is = 1 (Use new settings). For all other functions it may be empty (Number of Descriptors = 0).</p>								
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>			<p><i>Field Data</i></p>				
<p><i>Command</i></p>	<p>4 + 3*N</p>	<p>0x08</p>			<p>U8 - Function Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>				
<p><i>Reply ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>			<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>				
<p><i>Reply field 2 (function = 2)</i></p>	<p>3 + 3*N</p>	<p>0x80</p>			<p>U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>				
<p>Examples</p>	<p>MIP Packet Header</p>				<p>Command/Reply Fields</p>			<p>Checksum</p>	
	<p><i>Sync1</i></p>	<p><i>Sync2</i></p>	<p><i>Desc</i></p>	<p><i>Payload</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>MSB</i></p>	<p><i>LSB</i></p>

			Set	Length	Length	Desc.	Data		
Command IMU Message Format (use new settings)	0x75	0x65	0x0C	0x0A	0x0A	0x08	Function: 0x01 Desc count: 0x02 1 st Descriptor: 0x04 Rate Dec: 0x000A 2 nd Descriptor: 0x05 Rate Dec: 0x000A	0x22	0xA0
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x08 Error code: 0x00	0xE7	0xBA
Command IMU Message Format (read back current settings)	0x75	0x65	0x0C	0x04	0x04	0x08	Function: 0x02 Desc count: 0x00	0xF8	0xF3
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x0D	0x04	0xF1	Echo cmd: 0x08 Error code: 0x00		
Reply field 2 Current IMU Message Format					0x09	0x80	Desc count: 0x02 1 st Descriptor: 0x03 Rate Dec: 0x000A 2 nd Descriptor: 0x04 Rate Dec: 0x000A	0x98	0x0F

Copy-Paste version of the commands:

Use New Settings: "7565 0C0A 0A08 0102 0400 0A05 000A 22A0"

Read Current Settings: "7565 0C04 0408 0200 F8F3"

GPS Message Format (0x0C, 0x09)

<p>Description</p>	<p>Set, read, or save the format of the GPS message packet. This function sets the format for the GPS MIP data packet when in standard mode. The resulting message will maintain the order of descriptors sent in the command. The command has a function selector and a descriptor array as parameters.</p>								
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The rate decimation field is calculated as follows for GPS messages:</p> <p><i>Data Rate = GPS Base Rate / Rate Decimation</i></p> <p>The device checks that all descriptors are valid prior to executing this command. If any of the descriptors are invalid for the GPS data descriptor set, a NACK will be returned and the message format will be unchanged. The descriptor array only needs to be provided if the function selector is = 1 (Use new settings). For all other functions it may be empty (Number of Descriptors = 0).</p>								
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>			<p><i>Field Data</i></p>				
<p><i>Command</i></p>	<p>4 + 3*N</p>	<p>0x09</p>			<p>U8 - Function Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>				
<p><i>Reply field 1 ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>			<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>				
<p><i>Reply field 2 (function = 2)</i></p>	<p>3 + 3*N</p>	<p>0x81</p>			<p>U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>				
<p>Examples</p>	<p>MIP Packet Header</p>				<p>Command/Reply Fields</p>			<p>Checksum</p>	
	<p><i>Sync1</i></p>	<p><i>Sync2</i></p>	<p><i>Desc</i></p>	<p><i>Payload</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>Field</i></p>	<p><i>MSB</i></p>	<p><i>LSB</i></p>

			Set	Length	Length	Desc.	Data		
Command GPS Message Format (use new settings)	0x75	0x65	0x0C	0x0A	0x0A	0x09	Function: 0x01 Desc count: 0x02 1 st Descriptor: 0x03 Data rate: 0x0004 2 nd Descriptor: 0x05 Data rate: 0x0004	0x16	0x85
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x09 Error code: 0x00	0xE8	0xBC
Command GPS Message Format (read back current settings)	0x75	0x65	0x0C	0x04	0x04	0x09	Function: 0x02 Desc count: 0x00	0xF9	0xF6
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x0D	0x04	0xF1	Echo cmd: 0x09 Error code: 0x00		
Reply field 2 Current GPS Message Format					0x09	0x81	Desc count: 0x02 1 st Descriptor: 0x03 Data rate: 0x0004 2 nd Descriptor: 0x05 Datarate: 0x0004	0x8D	0xFE

Copy-Paste version of the commands:

Use New Settings: "7565 0C0A 0A09 0102 0300 0405 0004 1685"

Read Current Settings: "7565 0C04 0409 0200 F9F6"

Estimation Filter Message Format (0x0C, 0x0A)

<p>Description</p>	<p>Set, read, or save the format of the Estimation Filter message packet. This function sets the format for the Estimation Filter MIP data packet when in standard mode. The resulting message will maintain the order of descriptors sent in the command. The command has a function selector and a descriptor array as parameters.</p>		
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The rate decimation field is calculated as follows for Estimation Filter messages:</p> <p><i>Data Rate = <u>EF Base Rate</u> / Rate Decimation</i></p> <p>The device checks that all descriptors are valid prior to executing this command. If any of the descriptors are invalid for the Estimation Filter data descriptor set, a NACK will be returned and the message format will be unchanged. The descriptor array only needs to be provided if the function selector is = 1 (Use new settings). For all other functions it may be empty (Number of Descriptors = 0).</p>		
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>
<p><i>Command</i></p>	<p>4 + 3*N</p>	<p>0x0A</p>	<p>U8 - Function Selector U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>
<p><i>Reply field 1 ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>	<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>
<p><i>Reply field 2 (function = 2)</i></p>	<p>3 + 3*N</p>	<p>0x82</p>	<p>U8 – Number of Descriptors (N), N*(U8 – Descriptor, U16 – Rate Decimation)</p>
<p>Examples</p>	<p>MIP Packet Header</p>		<p>Command/Reply Fields</p>
			<p>Checksum</p>

	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command Estimation Filter Message Format (use new settings)	0x75	0x65	0x0C	0x0A	0x0A	0x0A	Function: 0x01 Desc count: 0x02 1 st Descriptor: 0x01 Data rate: 0x0001 2 nd Descriptor: 0x02 Data rate: 0x0001	0x0C	0x6A
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x0A Error code: 0x00	0xE9	0xBE
Command Estimation Filter Message Format (read back current settings)	0x75	0x65	0x0C	0x04	0x04	0x0A	Function: 0x02 Desc count: 0x00	0xFA	0xF9
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x0D	0x04	0xF1	Echo cmd: 0x0A Error code: 0x00		
Reply field 2 Current GPS Message Format					0x09	0x82	Desc count: 0x02 1 st Descriptor: 0x01 Data rate: 0x0001 2 nd Descriptor: 0x02 Datarate: 0x0001	0x84	0xED

Copy-Paste version of the commands:

Use New Settings: "7565 0C0A 0A0A 0102 0100 0102 0C6A"

Read Current Settings: "7565 0C04 040A 0200 FAF9"

Enable/Disable Continuous Data Stream (0x0C, 0x11)

<p>Description</p>	<p>Control the streaming of IMU, GPS, and Estimation Filter data. If disabled, the data from the selected device is not continuously transmitted. Upon enabling, the most current data will be transmitted (i.e. no stale data is transmitted.) The default for the device is all streams enabled. For all functions except 0x01 (use new setting), the new enable flag value is ignored.</p>								
<p>Notes</p>	<p>Possible function selector values:</p> <p style="padding-left: 40px;">0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings</p> <p><i>The device selector can be:</i></p> <p style="padding-left: 40px;">0x01 – IMU 0x02 – GPS 0x03 – Estimation Filter</p> <p><i>The enable flag can be either:</i></p> <p style="padding-left: 40px;">0x00 – disable the selected stream. 0x01 – enable the selected stream. (<i>default</i>)</p>								
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>			<p><i>Field Data</i></p>				
<p><i>Command</i></p>	<p>0x05</p>	<p>0x11</p>			<p>U8 – Function Selector U8 – Device Selector U8 – New Enable Flag</p>				
<p><i>Reply field 1 ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>			<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>				
<p><i>Reply field 2 (function = 2)</i></p>	<p>0x04</p>	<p>0x85</p>			<p>U8 – Device Selector U8 – Current Device Enable Flag</p>				
<p>Examples</p>	<p>MIP Packet Header</p>				<p>Command/Reply Fields</p>			<p>Checksum</p>	
	<p><i>Sync1</i></p>	<p><i>Sync2</i></p>	<p><i>Desc Set</i></p>	<p><i>Payload Length</i></p>	<p><i>Field Length</i></p>	<p><i>Field Desc.</i></p>	<p><i>Field Data</i></p>	<p><i>MSB</i></p>	<p><i>LSB</i></p>
<p><i>Command IMU Stream ON</i></p>	<p>0x75</p>	<p>0x65</p>	<p>0x0C</p>	<p>0x05</p>	<p>0x05</p>	<p>0x11</p>	<p>Function(Apply):0x01 Device (IMU): 0x01</p>	<p>0x04</p>	<p>0x1A</p>

							Stream (ON): 0x01		
<i>Command</i> <i>IMU Stream OFF</i>	0x75	0x65	0x0C	0x05	0x05	0x11	Function(Apply): 0x01 Device (IMU): 0x01 Stream (OFF): 0x00	0x03	0x19
<i>Command</i> <i>GPS Stream ON</i>	0x75	0x65	0x0C	0x05	0x05	0x11	Function(Apply): 0x01 Device (GPS): 0x02 Stream (ON): 0x01	0x05	0x1C
<i>Command</i> <i>GPS Stream OFF</i>	0x75	0x65	0x0C	0x05	0x05	0x11	Function(Apply): 0x01 Device (GPS): 0x02 Stream (OFF): 0x00	0x04	0x1B
<i>Command</i> <i>Estimation Filter</i> <i>Stream ON</i>	0x75	0x65	0x0C	0x05	0x05	0x11	Function(Apply): 0x01 Device (Estimation Filter): 0x03 Stream (ON): 0x01	0x06	0x1E
<i>Command</i> <i>Estimation Filter</i> <i>Stream OFF</i>	0x75	0x65	0x0C	0x05	0x05	0x11	Function(Apply): 0x01 Device (Estimation Filter): 0x03 Stream (OFF): 0x00	0x05	0x1D
<i>Reply</i> <i>ACK/NACK</i>	0x75	0x65	0x0C	0x05	0x05	0xF1	<i>Echo cmd:</i> 0x11 <i>Error code:</i> 0x00	0xEF	0xCA

Copy-Paste version of the 1st command: "7565 0C05 0511 0101 0104 1A"

Device Startup Settings (0x0C, 0x30)

Description	Save, Load, or Reset to Default the values for all device settings.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings</p> <p><i>**When a save current settings command is issued a brief data disturbance may occur.</i></p>								
Field Format	<i>Field Length</i>		<i>Field Descriptor</i>		<i>Field Data</i>				
Command	0x02		0x30		U8 –Function Selector				
Reply ACK/NACK	0x04		0xF1		U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Save All</i>	0x75	0x65	0x0C	0x03	0x03	0x30	<i>Fctn(Save):0x03</i>	0x1F	0x45
<i>Reply ACK/NACK</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	<i>Echo cmd:0x30 Error code:0x00</i>	0x0F	0x0A

Copy-Paste version of the command: "7565 0C03 0330 031F 45"

Accel Bias (0x0C, 0x37)

Advanced

Description	Set the value, or read the current value of the IMU7 Accelerometer Bias Vector. For all functions except 0x01 and 0x06 (apply new settings), the new vector value is ignored. The bias value is subtracted from the scaled accelerometer value prior to output.								
Notes	<p>Possible function selector values:</p> <ul style="list-style-type: none"> 0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings 0x06 – Apply new settings with no ACK/NACK Reply 								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x37			U8 – Function Selector float – X Accel Bias Value float – Y Accel Bias Value float – Z Accel Bias Value				
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0x9A			float – current X Accel Bias Value float – current Y Accel Bias Value float – current Z Accel Bias Value				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Accel Bias</i>	0x75	0x65	0x0C	0x0F	0x0F	0x37	<i>Fctn(Apply):0x01 Field (Bias): 0x00000000 0x00000000 0x00000000</i>	0x3C	0x75
<i>Reply</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	<i>Echo cmd:0x37</i>	0x16	0x18

ACK/NACK							Error code:0x00		
----------	--	--	--	--	--	--	-----------------	--	--

Copy-Paste version of the command: "7565 0C0F 0F37 0100 0000 0000 0000 0000 0000 003C 75"

Gyro Bias (0x0C, 0x38)

Advanced

Description	Set the value, or read the current value of the IMU7 Gyro Bias Vector. For all functions except 0x01 and 0x06 (apply new settings), the new vector value is ignored. The bias value is subtracted from the scaled Gyro value prior to output.								
Notes	<p>Possible function selector values:</p> <ul style="list-style-type: none"> 0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings 0x06 – Apply new settings with no ACK/NACK Reply 								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x38			U8 – Function Selector float – X Gyro Bias Value float – Y Gyro Bias Value float – Z Gyro Bias Value				
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0x9B			float – current X Gyro Bias Value float – current Y Gyro Bias Value float – current Z Gyro Bias Value				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Gyro Bias</i>	0x75	0x65	0x0C	0x0F	0x0F	0x38	<i>Fctn(Apply):0x01 Field (Bias):</i>	0x3D	0x83

							0x00000000 0x00000000 0x00000000		
Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd:0x38 Error code:0x00	0x17	0x1A

Copy-Paste version of the command: "7565 0C0F 0F38 0100 0000 0000 0000 0000 0000 003D 83"

Capture Gyro Bias (0x0C, 0x39)

Description	This command will cause the IMU7 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 Gyro Bias vector. The bias vector is not saved as a startup value. If you wish to save this vector, use the Gyro Bias command.								
Notes	Possible Sampling Time values: Total sampling time in units of milliseconds. Range of values: 1000 to 30000. Note: The IMU7 must be stationary during the execution of the Capture Gyro Bias Operation.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x04	0x39			U16 – Sampling Time (milliseconds)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0x9B			float – current X Gyro Bias Value float – current Y Gyro Bias Value float – current Z Gyro Bias Value				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>

<i>Command</i>	0x75	0x65	0x0C	0x04	0x04	0x39	Sampling Time: 0x2710	0x5E	0xE0
<i>Reply field 1</i> <i>ACK/NACK</i>	0x75	0x65	0x0C	0x12	0x04	0xF1	Echo cmd: 0x39 Error code: 0x00		
<i>Reply field 2</i> <i>Bias Vector</i>					0x0E	0x9B	Field (Bias): 0x00000000 0x00000000 0x00000000	0xCF	0x19

Copy-Paste version of the command: "7565 0C04 0439 2710 5EE0"

Magnetometer Hard Iron Offset (0x0C, 0x3A)

Description	This command will read or write values to the magnetometer Hard Iron Offset Vector. For all functions except 0x01 and 0x06 (apply new settings), the new vector value is ignored. The offset value is subtracted from the scaled Mag value prior to output.								
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings 0x06 – Apply new settings with no ACK/NACK Reply 								
Notes	<p><i>Default values:</i></p> <p>Hard Iron Offset: [0,0,0]</p>								
Field Format	<i>Field Length</i>		<i>Field Descriptor</i>			<i>Field Data</i>			
<i>Command</i>	0x0F		0x3A			U8 – Function Selector Float – X Hard Iron Offset Float – Y Hard Iron Offset Float – Z Hard Iron Offset			
<i>Reply field 1 ACK/NACK</i>	0x04		0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)			
<i>Reply field 2 (function = 2)</i>	0x0E		0x9C			Float – current X Hard Iron Offset Float – current Y Hard Iron Offset Float – current Z Hard Iron Offset			
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>

<p><i>Command</i></p> <p><i>Hard Iron Offset</i></p>	0x75	0x65	0x0C	0x0F	0x0F	0x3A	<p><i>Fctn(Apply):0x01</i></p> <p><i>Offset Vector:</i></p> <p>0x00000000</p> <p>0x00000000</p> <p>0x00000000</p>	0x3F	0x9F
<p><i>Reply field 1</i></p> <p><i>ACK/NACK</i></p>	0x75	0x65	0x0C	0x04	0x04	0xF1	<p><i>Echo cmd: 0x3A</i></p> <p><i>Error code: 0x00</i></p>	0x19	0x1E

Copy-Paste version of the command: "7565 0C0F 0F3A 0100 0000 0000 0000 0000 0000 003F 9F"

Magnetometer Soft Iron Matrix (0x0C, 0x3B)

Description	<p>This command will read or write values to the magnetometer Soft Iron Compensation Matrix. The values for this matrix are determined empirically by external software algorithms based on calibration data taken after the device is installed in its application. These values can be obtained and set by using the MicroStrain “GX Iron Calibration” application. The matrix is applied to the scaled magnetometer vector prior to output.</p>								
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings 0x06 – Apply new settings with no ACK/NACK Reply <p><i>Default values:</i></p> <p>Soft Iron Compensation Matrix (identity matrix; row order): [1,0,0][0,1,0][0,0,1]</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x27	0x3B			U8 – Function Selector float – m _{1,1} float – m _{1,2} float – m _{1,3} float – m _{2,1} float – m _{2,2} float – m _{2,3} float – m _{3,1} float – m _{3,2} float – m _{3,3}				
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x26	0x9D			float – m _{1,1} float – m _{1,2} float – m _{1,3} float – m _{2,1} float – m _{2,2} float – m _{2,3} float – m _{3,1} float – m _{3,2} float – m _{3,3}				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>

<p>Command</p> <p>Soft Iron Matrix</p>	0x75	0x65	0x0C	0x27	0x27	0x3B	<p>Fctn(Apply):0x01</p> <p>Comp Matrix:</p> <p>0x3F800000</p> <p>0x00000000</p> <p>0x00000000</p> <p>0x00000000</p> <p>0x3F800000</p> <p>0x00000000</p> <p>0x00000000</p> <p>0x00000000</p> <p>0x3F800000</p>	0xA D	0x59
<p>Reply field 1</p> <p>ACK/NACK</p>	0x75	0x65	0x0C	0x12	0x04	0xF1	<p>Echo cmd: 0x3B</p> <p>Error code: 0x00</p>	0x1 A	0x20

Copy-Paste version of the command: "7565 0C27 273B 013F 8000 0000 0000 0000 0000 0000 0000 0000 003F 8000 0000 0000 0000 0000 0000 003F 8000 00AD 59"

Coning and Sculling Enable (0x0C, 0x3E)

Description	Set, read, or save the Coning and Sculling compensation Enable. This function sets the Coning and Sculling compensation Enable. For all functions except 0x01 (use new setting), the new parameter values are ignored.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Apply new setting 0x02 – Read back current setting 0x03 – Save current settings as startup setting 0x04 – Load saved startup setting 0x05 – Load factory default setting</p> <p><i>The enable flag can be either:</i></p> <p>0x00 – disable the Coning and Sculling compensation. 0x01 – enable the Coning and Sculling compensation. (<i>default</i>)</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x10	0x3E			U8 – Function Selector U8 – New Coning and Sculling enable setting				
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x03	0x9E			U8 – Current Coning and Sculling enable setting				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Enable Settings</i>	0x75	0x65	0x0C	0x04	0x04	0x3E	<i>Fctn (Apply): 0x01 Enable: 0x01</i>	2E	94
<i>Reply ACK/NACK</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	<i>Echo cmd: 0x3E Error code: 0x00</i>	1D	26

Copy-Paste version of the command: "7565 0C04 043E 0101 2E94"

UART BAUD Rate (0x0C, 0x40)

Description	Change, read, or save the BAUD rate of the main communication channel (UART1). For all functions except 0x01 (use new settings), the new BAUD rate value is ignored.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p><i>Supported BAUD rates are:</i></p> <p>9600, 19200, 115200(<i>default</i>), 230400, 460800</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x07	0x40			U8 – Function Selector U32 –New BAUD rate				
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x06	0x87			U32 – Current BAUD rate				
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Set BAUD Rate Command</i>	0x75	0x65	0x0C	0x07	0x07	0x40	<i>Fctn(USE):0x01 BAUD (115200): 0x0001C200</i>	0xF8	0xDA
<i>Reply ACK/NACK</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	<i>Echo cmd:0x40 Error code:0x00</i>	0x1F	0x2A

Copy-Paste version of the command: "7565 0C07 0740 0100 01C2 00F8 DA"

Advanced Low-Pass Filter Settings (0x0C, 0x50)

<p>Description</p>	<p>Advanced configuration for low-pass filter settings. The scaled data quantities are by default filtered through a single-pole IIR low-pass filter which is configured with a -3dB cutoff frequency of half the reporting frequency (set by decimation factor in the IMU Message Format command) to prevent aliasing on a per data quantity basis. This advanced configuration command allows for the cutoff frequency to be configured independently of the data reporting frequency as well as allowing for a complete bypass of the digital low-pass filter for either or both scaled data quantities.</p>
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Possible data descriptors:</i></p> <ul style="list-style-type: none"> 0x04 – Scaled accel data 0x05 – Scaled gyro data 0x06 – Scaled mag data 0x17 – Scaled pressure data <p><i>Possible filter enable values:</i></p> <ul style="list-style-type: none"> 0x01 – Apply low-pass filter 0x00 – Do not apply low-pass filter <p><i>Manual filter bandwidth configuration:</i></p> <ul style="list-style-type: none"> 0x01 – Use user specified -3 dB cutoff frequency 0x00 – Automatically configure -3 dB cutoff frequency to half reporting rate <p><i>-3 dB Cutoff Frequency:</i></p> <p>Cutoff Frequency value specified must be no greater than 250 Hz.</p>

	<p>**This value in a write command is ignored if Automatic Bandwidth is selected.</p> <p><i>Reserved Byte:</i></p> <p>This byte is reserved for internal use and should be left in the 0x00 state</p>								
Field Format	Field Length	Field Descriptor				Field Data			
Command	0x09	0x50				U8 – Function Selector U8 – Data Descriptor U8 – Low-Pass Filter Enable/Disable U8 – Manual/Auto -3 dB Cutoff Frequency Configuration U16 – -3 dB Cutoff Frequency U8 – Reserved Byte			
Reply ACK/NACK	0x04	0xF1				U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)			
Reply field 2 (function = 2)	0x08	0x8B				U8 – Data Descriptor U8 – Filter (0x01: Enabled, 0x00: Disabled) U8 – Cutoff Frequency (0x00: Auto, 0x01: Manual) U16 – -3 dB Cutoff Frequency Hz U8 – Reserved			
Examples	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command IMU Message Format (use new settings)	0x75	0x65	0x0C	0x09	0x09	0x50	Function: 0x01 Scaled Accel: 0x04 Enable Filter: 0x01 Automatic Cutoff Configuration: 0x00 -3 dB Cutoff Frequency: 0x0000 (ignored for automatic cutoff configuration) Reserved: 0x00	0x4C	0x6D

Reply ACK/NACK	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd:0x50 Error code:0x00	0x2F	0x4A
-------------------	------	------	------	------	------	------	----------------------------------	------	------

Copy-Paste version of the commands: "7565 0C09 0950 0104 0100 0000 004E 80"

Device Status (0x0C, 0x64)

Description	Get the device-specific status for the 3DM-RQ1-45
Notes	<p>Reply has two fields: "ACK/NACK" and "Device Status Field". The device status field may be one of two selectable formats – basic and diagnostic.</p> <p>The reply data for this command is device specific. The reply is specified by two parameters in the command. The first parameter is the model number(which for the 3DM-RQ1-45 is always = 6232 (0x1858)). That is followed by a status selector byte which determines the type of data structure returned. In the case of the 3DM-RQ1-45, there are two selector values – one to return a basic status structure and a second to return an extensive diagnostics status structure. A list of available values for the selector values and specific fields in the data structure are as follows:</p> <p><i>Possible Status Selector Values:</i></p> <ul style="list-style-type: none"> 0x01 – Basic Status Structure 0x02 – Diagnostic Status Structure <p><i>Possible Communication Mode Values:</i></p> <ul style="list-style-type: none"> 0x01 – Standard MIP Mode 0x02 – Advanced IMU Direct Mode 0x03 – Advanced GPS Direct Mode <p><i>Possible Communication Device Values:</i></p> <ul style="list-style-type: none"> 0x01 - Com1 (Serial) <p><i>Possible System State Values:</i></p> <ul style="list-style-type: none"> 0x0001 – Initialization 0x0002 – Sensor Start-up 0x0003 – Running

<p><i>Possible Com1 State, GPS Driver State, GPS Port State, IMU Driver State, IMU Port State Values:</i></p> <p>0x00 – Not Initialized 0x01 – Initialized</p>						
Field Format	Field Length	Field Descriptor	Field Data			
Command	0x02	0x64	U16-Device Model Number: set = 6232 (0x1858) U8-Status Selector			
Reply field 1 ACK/NACK Field	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)			
Reply field 2 Basic Device Status Field	0x0F	0x90	Binary Offset	Description	Data Type	Units
			0	Echo of the Device Model Number	U16	N/A
			2	Echo of the selector byte	U8	N/A
			3	Status Flags (Reserved)	U32	N/A
			7	System state	U16	See Notes
			9	System Timer (since start-up)	U32	millisecond s
Reply field 2 Diagnostic Device Status Field	0x56	0x90	Binary Offset	Description	Data Type	Units
			0	Echo of the Device Model Number	U16	N/A
			2	Echo of the selector byte	U8	N/A
			3	Status Flags (Reserved)	U32	N/A
			7	System state	U16	See Notes
			9	System Timer (since start-up)	U32	millisecond s
			13	GPS Power On	U8	1 – on 0 – off
			14	Cold Start (T < -40.0 degC) on Power-up	U8	1 – yes 0 – no
			15	On-board Temperature	float	degC
			19	Last Temp Sensor Read (System Timer)	U32	millisecond s

23	Number of GPS 1PPS Pulses	U32	Count
27	Last GPS 1PPS (System Timer)	U32	milliseconds
31	IMU Stream Enabled	U8	1 – on 0 – off
32	GPS Stream Enabled	U8	1 – on 0 – off
33	Estimation Filter Stream Enabled	U8	1 – on 0 – off
34	Outgoing IMU Stream Dropped Packet Count	U32	Count
38	Outgoing GPS Stream Dropped Packet Count	U32	Count
42	Outgoing Estimation Filter Stream Dropped Packet Count	U32	Count
46	Number of bytes written to com port	U32	Count
50	Number of bytes read from com port	U32	Count
54	Number of overruns when writing to com port	U32	Count
58	Number of overruns when reading from com port	U32	Count
62	Number of IMU message parsing errors	U32	Count
66	Total IMU messages read	U32	Count
70	Last IMU message read (System Timer)	U32	Millisecond
74	Number of GPS message parsing errors	U32	Count
78	Total GPS messages read	U32	Count
82	Last GPS message read (System Timer)	U32	Millisecond

Example	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command Get Device Status (return Basic Status structure: selector =	0x75	0x65	0x0C	0x05	0x05	0x64	Model # (6232): 0x1858 Status Selector (basic status): 0x01	0xC1	0x51

1)									
Reply field 1 ACK/NACK	0x75	0x65	0x0C	0x15	0x04	0xF1	Echo cmd: 0x64 Error code: 0x00		
Reply field 2 Device Status (Basic Status structure)					0x0D	0x90	Echo Model#: 0x1858 Echo Selector: 0x01 Additional Data	0x##	0x##

Copy-Paste version of the command: "7565 0C05 0564 1852 01BF 4D"

Raw RTCM 2.3 Message (0x0C, 0x20)

Description	Send Raw RTCM 2.3 differential corrections to the device								
Notes	The device currently only accepts message types 1, 2, 3, and 9. RTK corrections are not supported. The RTCM data can be limited to 1 message per command or streamed into the device by dividing the stream into reasonably-sized chunks (no greater than 253 bytes.)								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x2 + Num_bytes	0x20			Num_bytes of the raw RTCM2.3 stream				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0C	2+Num	2+Num	0x20	Num_Bytes of raw RTCM2.3 data	0xXX	0xXX
<i>Reply ACK/NACK</i>	0x75	0x65	0x0C	0x04	0x04	0xF1	Echo cmd: 0x20 Error code: 0x00	0xFF	0xEA

Copy-Paste version of the command: N/A

Estimation Filter Commands

The Estimation Filter command set is specific to MicroStrain Inertial Navigation and advanced AHRS sensors.

Reset Filter (0x0D, 0x01)

Description	Reset the filter to the initialize state.								
Notes	If the auto-initialization feature is disabled, the initial attitude or heading must be set in order to enter the run state after a reset.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x01			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, non-zero:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x02	0x02	0x01		0xEC	0xF6
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x01</i> <i>Error code: 0x00</i>	0xE1	0xB2

Copy-Paste version of the command: "7565 0D02 0201 ECF6"

Set Initial Attitude (0x0D, 0x02)

Description	Set the initial attitude.								
Notes	<p>This command can only be issued in the “INIT” state and should be used with a good estimate of the vehicle attitude. The Euler Angles are the sensor body frame with respect to the local NED frame.</p> <p>The valid input ranges are as follows:</p> <p>Roll: $[-\pi, \pi]$ Pitch: $[-\frac{\pi}{2}, \frac{\pi}{2}]$ Yaw: $[-\pi, \pi]$</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0E	0x02			Float – Roll (radians) Float – Pitch (radians) Float – Heading (radians)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0E	0E	02	Roll:0x00000000 (0.0f) Pitch:0x00000000 (0.0f) Heading:0x00000000 (0.0f)	0x05	0x6F
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x02 Error code: 0x00	0xE2	0xB4

Copy-Paste version of the command: “7565 0D0E0E02 0000 0000 0000 0000 0000 0000 0000 0000 056F”

Set Initial Heading (0x0D, 0x03)

Description	Set the initial heading angle.								
Notes	This command can only be issued in the “INIT” state and should be used with a good estimation of Heading. The device will use this value in conjunction with the output of the accelerometers to determine the initial attitude estimate. The Euler Angles are the sensor body frame with respect to the local NED frame.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x06	0x03			Float – Heading (radians)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x06	0x06	0x03	<i>Heading:0x00000000 (0.0f)</i>	0xF6	0xE4
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x03 Error code: 0x00</i>	0xE3	0xB6

Copy-Paste version of the command: “7565 0D06 0603 0000 0000 F6E4”

Vehicle Dynamics Mode (0x0D, 0x10)

Description	Set, read, or save the vehicle dynamics mode. For all functions except 0x01 (use new settings), the new dynamics mode value is ignored.																												
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Apply new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Load factory default settings</p> <p><i>Possible Modes:</i></p> <table border="1" data-bbox="375 772 1398 1136"> <thead> <tr> <th>Mode</th> <th>Use</th> <th>Altitude Limits</th> <th>Velocity Limits</th> </tr> </thead> <tbody> <tr> <td>0x01 – Portable (default)</td> <td>Applications with low acceleration</td> <td>12,000 m</td> <td>Horizontal - 310 m/s Vertical - 50 m/s</td> </tr> <tr> <td>0x02 – Automotive</td> <td>Low vertical acceleration, wheeled-vehicle dynamics</td> <td>6000 m</td> <td>Horizontal - 84 m/s Vertical - 15 m/s</td> </tr> <tr> <td>0x03 – Airborne</td> <td>Airborne application up to 2 Gs</td> <td>50,000 m</td> <td>Horizontal - 250 m/s Vertical - 100 m/s</td> </tr> <tr> <td>0x04 – Airborne (High G)</td> <td>Airborne application up to 4 Gs</td> <td>50,000 m</td> <td>Horizontal - 250 m/s Vertical - 100 m/s</td> </tr> </tbody> </table>									Mode	Use	Altitude Limits	Velocity Limits	0x01 – Portable (default)	Applications with low acceleration	12,000 m	Horizontal - 310 m/s Vertical - 50 m/s	0x02 – Automotive	Low vertical acceleration, wheeled-vehicle dynamics	6000 m	Horizontal - 84 m/s Vertical - 15 m/s	0x03 – Airborne	Airborne application up to 2 Gs	50,000 m	Horizontal - 250 m/s Vertical - 100 m/s	0x04 – Airborne (High G)	Airborne application up to 4 Gs	50,000 m	Horizontal - 250 m/s Vertical - 100 m/s
Mode	Use	Altitude Limits	Velocity Limits																										
0x01 – Portable (default)	Applications with low acceleration	12,000 m	Horizontal - 310 m/s Vertical - 50 m/s																										
0x02 – Automotive	Low vertical acceleration, wheeled-vehicle dynamics	6000 m	Horizontal - 84 m/s Vertical - 15 m/s																										
0x03 – Airborne	Airborne application up to 2 Gs	50,000 m	Horizontal - 250 m/s Vertical - 100 m/s																										
0x04 – Airborne (High G)	Airborne application up to 4 Gs	50,000 m	Horizontal - 250 m/s Vertical - 100 m/s																										
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>																								
<i>Command</i>	0x04	0x10			U8 – Function Selector U8 – New Dynamics Mode																								
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)																								
<i>Reply field 2 (function = 2)</i>	3	0x80			U8 –Current Dynamics Mode																								
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>																					
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>																				
<i>Command DynamicsMode</i>	0x75	0x65	0x0D	0x04	0x04	0x10	<i>Fctn (Apply): 0x01 Mode</i>	0x01	0x10																				

							(Portable):0x01		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x10 Error code:0x00	0xF0	0xD0

Copy-Paste version of the command: "7565 0C04 0410 01010110"

Sensor to Vehicle Frame Transformation (0x0D, 0x11)

Description	Set the sensor to vehicle frame transformation matrix using Roll, Pitch, and Yaw Euler angles. These angles define the rotation <i>from</i> the sensor body frame <i>to</i> the fixed vehicle frame. Please reference the device Theory of Operation for more information.		
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>This transformation affects the following output quantities:</p> <p>IMU: Scaled Acceleration Scaled Gyro Delta Theta Delta Velocity</p> <p>Estimation Filter: Estimated Orientation, Quaternion Estimated Orientation, Matrix Estimated Orientation, Euler Angles Estimated Linear Acceleration Estimated Angular Rate Estimated Gravity Vector</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>

<i>Command</i>	0x0F	0x11	U8 – Function Selector Float – Roll Angle (radians) Float – Pitch Angle (radians) Float – Yaw Angle (radians)						
<i>Reply ACK/NACK</i>	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)						
<i>Reply field 2 (function = 2)</i>	0x0E	0x81	Float – Roll Angle (radians) Float – Pitch Angle (radians) Float – Yaw Angle (radians)						
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x11	<i>Fctn (Apply): 0x01</i> <i>Roll:0x00000000</i> (0.0f) <i>Pitch:0x00000000</i> (0.0f) <i>Yaw:0x00000000</i> (0.0f)	0x17	0x72
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x11</i> <i>Error code:0x00</i>	0xF1	0xD2

Copy-Paste version of the command: "7565 0D0F 0F11 0100 0000 0000 0000 0000 0000 0017 72"

Sensor to Vehicle Frame Offset (0x0D, 0x12)

Description	Set the sensor to vehicle frame offset, expressed in the sensor frame. Please reference the device Theory of Operation for more information.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p>This offset affects the following output quantities: Estimated LLH Position</p> <p>The offset vector magnitude is limited to 10 meters.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x12			U8 – Function Selector Float – X (meters, sensor body frame) Float – Y (meters, sensor body frame) Float – Z (meters, sensor body frame)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0x82			Float – X (meters, sensor body frame) Float – Y (meters, sensor body frame) Float – Z (meters, sensor body frame)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x12	<i>Fctn (Apply): 0x01</i> X:0x00000000 (0.0f) Y:0x00000000 (0.0f) Z:0x00000000	0x18	0x80

							(0.0f)		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x12 Error code:0x00	0xF2	0xD4

Copy-Paste version of the command: "7565 0D0F 0F12 0100 0000 0000 0000 0000 0000 0018 80"

Antenna Offset (0x0D, 0x13)

Description	Set the sensor to antenna offset, expressed in the sensor frame from the sensor inertial origin to the GPS antenna RF center.								
Notes	<p>Possible function selector values:</p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The offset vector magnitude is limited to 10 meters.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x13			U8 – Function Selector Float – X (meters, sensor body frame) Float – Y (meters, sensor body frame) Float – Z (meters, sensor body frame)				
Reply ACK/NACK	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
Reply field 2 (function = 2)	0x0E	0x83			Float – X (meters, sensor body frame) Float – Y (meters, sensor body frame) Float – Z (meters, sensor body frame)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x13	<i>Fctn (Apply):</i> 0x01	0x19	0x8E

							X:0x00000000 (0.0f) Y:0x00000000 (0.0f) Z:0x00000000 (0.0f)		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x13 Error code:0x00	0xF3	0xD6

Copy-Paste version of the command: "7565 0D0F 0F13 0100 0000 0000 0000 0000 0000 0019 8E"

Estimation Control Flags (0x0D, 0x14)

Description	Controls which parameters are estimated by the Kalman Filter.
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Available Flags :</i></p> <ul style="list-style-type: none"> 0x0001 – Enable Gyro Bias Estimation (Recommended) 0x0002 – Enable Accel Bias Estimation (Recommended) 0x0004 – Enable Gyro Scale Factor Estimation (Optional) 0x0008 – Enable Accel Scale Factor Estimation (Optional) 0x0010 – Enable GPS Antenna Offset Estimation (Optional) <p><i>Examples :</i></p> <ul style="list-style-type: none"> 0xFFFF – Enable all 0xFFF5 – Disable Accel Scale Factor and Bias Estimation 0xFFFA – Disable Gyro Scale Factor and Bias Estimation 0xFFEF – Disable GPS Antenna Offset Estimation <p>(note: Any bit without a designated function should be set to 1 for future compatibility.)</p>

Field Format	Field Length	Field Descriptor	Field Data						
Command	0x05	0x14	U8 – Function Selector U16 – Estimation Control Flags						
Reply ACK/NACK	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)						
Reply field 2 (function = 2)	0x04	0x84	U16 – Estimation Control Flags						
Example	MIP Packet Header				Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command	0x75	0x65	0x0D	0x05	0x05	0x14	Fctn (Apply): 0x01 Flags:0xFFFF (Enable all states)	0x04	0x27
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x14 Error code:0x00	0xF4	0xD8

Copy-Paste version of the command: "7565 0D05 0514 01FF FF04 27"

GPS Source Control (0x0D, 0x15)

Description	Control the source of GPS information used to update the Kalman Filter.
Notes	<p>Possible function selector values:</p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>Possible GPS Source values:</p>

0x01 – Internal GPS
 0x02 – External GPS (Requires user to provide GPS information via the “External GPS Update” command)

Changing the GPS source while the sensor is in the “running” state will temporarily place it back in the “init” state until the new source of GPS data is received.

Field Format	Field Length	Field Descriptor	Field Data
Command	0x04	0x15	U8 – Function Selector U8 –GPS Source
Reply ACK/NACK	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)
Reply field 2 (function = 2)	0x03	0x86	U8 – GPS Source

Example	MIP Packet Header				Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command	0x75	0x65	0x0D	0x04	0x04	0x15	Fctn (Apply): 0x01 Source:0x02 (External GPS)	0x07	0x20
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x15 Error code:0x00	0xF5	0xDA

Copy-Paste version of the command: “7565 0D04 0415 0102 0720”

External GPS Update (0x0D, 0x16)

Description	Trigger a filter update step using external GPS information.
Notes	GPS source control must be set to external for this command to update the filter; it will be ignored/NACK’d otherwise. The maximum rate for this message is 20 Hz.

Field Format	Field Length	Field Descriptor	Field Data						
Command	0x48	0x16	Double – GPS Time of Week (seconds) U16 – GPS Week Number Double – Latitude (deg) Double – Longitude (deg) Double – Altitude above WGS84 Ellipsoid (m) Float – North Velocity (m/s) Float – East Velocity (m/s) Float – Down Velocity (m/s) Float – North Position Uncertainty (m, 1-sigma) Float – East Position Uncertainty (m, 1-sigma) Float – Down Position Uncertainty (m, 1-sigma) Float – North Velocity Uncertainty (m/s, 1-sigma) Float – East Velocity Uncertainty (m/s, 1-sigma) Float – Down Velocity Uncertainty (m/s, 1-sigma)						
Reply ACK/NACK	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)						
Example	MIP Packet Header				Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Field Desc.	Field Data	MSB	LSB
Command	0x75	0x65	0x0D	48	48	16	GPS Tow: 0.0d GPS Week: 0x0000 Latitude:0.0d Longitude:0.0d Height:0.0d Vel North:0.0f Vel East:0.0f Vel Down:0.0f Pos Sigma (N)0.5f Pos Sigma (E)0.5f Pos Sigma (D)0.5f Vel Sigma (N)0.1f Vel Sigma (E)0.1f Vel Sigma (D)0.1f	0xFF	0xFF
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x16 Error code: 0x00	0xFF	0xDC

Copy-Paste version of the command: N/A

External Heading Update (0x0D, 0x17)

Description	Trigger a filter update step using external heading information <i>The heading must be the sensor frame with respect to the NED frame.</i>								
Notes	The heading update control must be set to external for this command to update the filter; it will be ignored/NACK'd otherwise. The maximum rate for this message is 20 Hz. Angle uncertainties of 0.0 will be NACK'd. Possible Heading Type Commands: 0x01 – True Heading 0x02 – Magnetic Heading* *Note: if the GPS (internal or external) is unavailable, magnetic heading updates will be NACK'd. This happens because the on-board magnetic model cannot be run without GPS updates.								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>						
<i>Command</i>	0x0B	0x17	Float – Heading Angle (radians, true north, +-PI) Float – Heading Angle Uncertainty (radians, 1-sigma) U8 – Heading type (1 – true, 2 – magnetic)						
<i>Reply ACK/NACK</i>	0x04	0xF1	U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)						
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0B	0x0B	0x17	Angle: 0.0f Angle Sigma: 0.01f Heading Type: 0x01(True)	0xXX	0xXX

Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x17 Error code: 0x00	0xF7	0xDE
-------------------	------	------	------	------	------	------	------------------------------------	------	------

Copy-Paste version of the command: N/A

Heading Update Control (0x0D, 0x18)

Description	Select the source for heading updates to the Kalman Filter.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p><i>Possible Enable Flag values:</i></p> <p>0x00 – Disable Heading Updates 0x02 – Use the Internal GPS Velocity Vector for Heading Updates** 0x03 – Use external heading updates</p> <p>**To use the Internal GPS velocity vector for heading updates, the target application must have no (or minimal) side-slip; this is true in most ground vehicle applications. <i>Additionally, when this option is selected, the X-axis of the sensor must be co-aligned with the direction of travel of the vehicle.</i></p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x04	0x18			U8 – Function Selector U8 –Enable Flag				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x03	0x87			U8 – Enable Flag				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x04	0x04	0x18	<i>Fctn (Apply): 0x01 Enable:0x01 (Enable Mag. Updates)</i>	0x09	0x28
<i>Reply</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x18</i>	0xF8	0xE0

ACK/NACK							Error code:0x00		
----------	--	--	--	--	--	--	-----------------	--	--

Copy-Paste version of the command: "7565 0D04 0418 0101 0928"

Auto-Initialization Control (0x0D, 0x19)

Description	Enable/Disable automatic initialization upon device startup.								
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Possible Enable Flag values:</i></p> <ul style="list-style-type: none"> 0x00 – Disable auto-initialization 0x01 – Enable auto-initialization* <p>*Note: Auto-initialization on the 3DM-RQ1 can only take place under one of the following conditions:</p> <ol style="list-style-type: none"> 1) The heading source is set to external and the user is providing external heading data 2) The heading source is set to internal GPS and the vehicle is moving with sufficient velocity to capture the heading. 								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x04	0x19			U8 – Function Selector U8 – Enable Flag				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x03	0x88			U8 – Enable Flag				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x04	0x04	0x19	<i>Fctn (Apply): 0x01</i>	0x0A	0x2B

							Enable:0x01 (Enable auto-initialization)		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x19 Error code:0x00	0xF9	0xE2

Copy-Paste version of the command: "7565 0D04 0419 0101 0A2B"

Accelerometer White Noise Standard Deviation (0x0D, 0x1A)

Description	Set the expected accelerometer white noise 1-sigma values. This function can be used to tune the filter performance in the target application.		
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>Each of the noise values must be greater than 0.0</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>
<i>Command</i>	0x0F	0x1A	U8 – Function Selector Float – X Accel Noise 1-sigma (meters/second ²) Float – Y Accel Noise 1-sigma (meters/second ²) Float – Z Accel Noise 1-sigma (meters/second ²)
Reply ACK/NACK	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)
Reply field 2 (function = 2)	0x0E	0x89	Float – X Accel Noise 1-sigma (meters/second ²) Float – Y Accel Noise 1-sigma

					(meters/second ²) Float – Z Accel Noise 1-sigma (meters/second ²)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x1A	<i>Fctn (Apply): 0x01</i> <i>X:(0.02f)</i> <i>Y:(0.02f)</i> <i>Z:(0.02f)</i>	0x60	0xA3
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x1A</i> <i>Error code:0x00</i>	0xFA	0xE4

Copy-Paste version of the command: "7565 0D0F 0F01 1A013CA3D70A3CA3D70A3CA3D760A3"

Gyroscope White Noise Standard Deviation (0x0D, 0x1B)

Description	Set the expected gyroscope white noise 1-sigma values. This function can be used to tune the filter performance in the target application.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p>Each of the noise values must be greater than 0.0</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x1B			U8 – Function Selector Float – X Gyro Noise 1-sigma (rad/second) Float – Y Gyro Noise 1-sigma (rad/second) Float – Z Gyro Noise 1-sigma (rad/second)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0x8A			Float – X Gyro Noise 1-sigma (rad/second) Float – Y Gyro Noise 1-sigma (rad/second) Float – Z Gyro Noise 1-sigma (rad/second)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x1B	<i>Fctn (Apply): 0x01</i> <i>X:(0.0000539f)</i> <i>Y:(0.0000539f)</i> <i>Z:(0.0000539f)</i>	0xDE	0xE8
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x1B</i> <i>Error code:0x00</i>	0xFB	0xE6

Copy-Paste version of the command: "7565 0D0F 0F1B 013A 0D4B AD3A 0D4B AD3A 0D4B ADDE E8"

Magnetometer White Noise Standard Deviation (0x0D, 0x42)

Description	Set the expected magnetometer white noise 1-sigma values. This function can be used to tune the filter performance in the target application.								
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>Each of the noise values must be greater than 0.0</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x0F	0x42			U8 – Function Selector Float – X Mag Noise 1-sigma (gauss) Float – Y Mag Noise 1-sigma (gauss) Float – Z Mag Noise 1-sigma (gauss)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x0E	0xB1			Float – X Mag Noise 1-sigma (gauss) Float – Y Mag Noise 1-sigma (gauss) Float – Z Mag Noise 1-sigma (gauss)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x0F	0x42	<i>Fctn (Apply): 0x01</i> X:(0.02f) Y:(0.02f) Z:(0.02f)	0x	0x

<i>Reply</i> <i>ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x42</i> <i>Error code:0x00</i>	0x22	0x34
---------------------------------	-------------	-------------	-------------	-------------	-------------	-------------	---	-------------	-------------

Copy-Paste version of the command: ""

Accelerometer Bias Model Parameters (0x0D, 0x1C)

Description	Set the accelerometer bias model parameters.								
Notes	<p><i>Possible function selector values:</i></p> <p>0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings</p> <p>Each of the noise values must be greater than 0.0</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x1B	0x1C			U8 – Function Selector Float – X Accel Bias Beta (1/second) Float – Y Accel Bias Beta (1/second) Float – Z Accel Bias Beta (1/second) Float – X Accel Bias Noise 1-sigma (m/s ²) Float – Y Accel Bias Noise 1-sigma (m/s ²) Float – Z Accel Bias Noise 1-sigma (m/s ²)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
<i>Reply field 2 (function = 2)</i>	0x1A	0x8B			Float – X Accel Bias Beta (1/second) Float – Y Accel Bias Beta (1/second) Float – Z Accel Bias Beta (1/second) Float – X Accel Bias Noise 1-sigma (m/s ²) Float – Y Accel Bias Noise 1-sigma (m/s ²) Float – Z Accel Bias Noise 1-sigma (m/s ²)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x1B	0x1C	Fctn (Apply): 0x01 X Beta: (0.01f) Y Beta: (0.01f) Z Beta: (0.01f) X Noise: (0.0006f) Y Noise: (0.0006f)	0xXX	0xXX

							Z Noise: (0.0006f)		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x1C Error code:0x00	0xFC	0xE8

Copy-Paste version of the command: N/A

Gyroscope Bias Model Parameters (0x0D, 0x1D)

Description	Set the gyroscope bias model parameters.		
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>Each of the noise values must be greater than 0.0</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>
<i>Command</i>	0x1B	0x1D	U8 – Function Selector Float – X Gyro Bias Beta (1/second) Float – Y Gyro Bias Beta (1/second) Float – Z Gyro Bias Beta (1/second) Float – X Gyro Bias Noise 1-sigma (rad /second) Float – Y Gyro Bias Noise 1-sigma (rad /second) Float – Z Gyro Bias Noise 1-sigma (rad /second)
Reply ACK/NACK	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)
Reply field 2 (function = 2)	0x1A	0x8C	Float – X Gyro Bias Beta (1/second) Float – Y Gyro Bias Beta (1/second) Float – Z Gyro Bias Beta (1/second) Float – X Gyro Bias Noise 1-sigma (rad /second)

							Float – Y Gyro Bias Noise 1-sigma (rad /second) Float – Z Gyro Bias Noise 1-sigma (rad /second)		
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x0F	0x1B	0x1D	<i>Fctn (Apply): 0x01</i> <i>X Beta: (0.01f)</i> <i>Y Beta: (0.01f)</i> <i>Z Beta: (0.01f)</i> <i>X Noise: (0.00016f)</i> <i>Y Noise: (0.00016f)</i> <i>Z Noise: (0.00016f)</i>	0xXX	0xXX
<i>Reply</i> <i>ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x1D</i> <i>Error code:0x00</i>	0xFD	0xEA

Copy-Paste version of the command: N/A

Zero Velocity Update (ZUPT) Control (0x0D, 0x1E)

Description	Control the use of zero velocity updates.		
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The ZUPT is triggered when the scalar magnitude of the GPS reported velocity vector is equal-to or less than the threshold value. The device will NACK threshold values that are less than zero (i.e. negative.)</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>
<i>Command</i>	0x08	0x1E	U8 – Function Selector

					U8 –Enable Value Float – ZUPT threshold (m/s)				
Reply ACK/NACK	0x04		0xF1		U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
Reply field 2 (function = 2)	0x07		0x8D		U8– Enable Value Float – ZUPT threshold (m/s)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x08	0x08	0x1E	<i>Fctn (Apply): 0x01</i> <i>Enable:0x01</i> (Enable ZUPTs) <i>Threshold:</i> 0x00000000 (0.0f)	0x17	0xBA
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x1E</i> <i>Error code:0x00</i>	0xFE	0xEC

Copy-Paste version of the command: "7565 0D08 081E 0101 00000000 17BA"

External Heading Update with Timestamp (0x0D, 0x1F)

Description	<p>Trigger a filter update step using external heading information that is time-tagged with a specific GPS time.</p> <p><i>The heading must be the sensor frame with respect to the NED frame.</i></p>								
Notes	<p>This is more accurate than the External Heading Update (0x0D, 0x17) and should be used in applications where the vehicle heading experiences high angular rate, which may cause significant error in the applied measurement due to the sampling, transmission, and processing time required for the command. Accurate time-stamping of the heading information is important. The maximum rate for this message is 20 Hz.</p> <p>Angle uncertainties of 0.0 will be NACK'd.</p> <p>Possible Heading Type Commands:</p> <p style="padding-left: 40px;">0x01 – True Heading 0x02 – Magnetic Heading*</p> <p>*Note: if the GPS is unavailable, magnetic heading updates will be NACK'd. This happens because the on-board magnetic model cannot be run without GPS updates.</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x15	0x1F			Double – GPS TOW (time-of-week, seconds) U16 – GPS week number Float – Heading Angle (radians, true north, +- PI) Float – Heading Angle Uncertainty (radians, 1-sigma) U8 – Heading type (1 – true, 2 – magnetic)				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, not 0:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x15	0x15	0x1F	GPS TOW: 30,000.0 GPS Week Number: 1700 Angle: 0.0f Angle Sigma: 0.01f	0xXX	0xXX

							Heading Type: 0x01 (True)		
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x1F Error code: 0x00	0xFF	0xEE

Copy-Paste version of the command: N/A

Zero Angular Rate Update Control (0x0D, 0x20)

Description	Control the use of zero angular rate updates.								
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The zero angular rate update is triggered when the scalar magnitude of the angular rate vector is equal-to or less than the threshold value. The device will NACK threshold values that are less than zero (i.e. negative.)</p>								
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x08	0x20			U8 – Function Selector U8 –Enable Value (0 – disable, 1 – enable) Float –Threshold (rad/s)				
Reply ACK/NACK	0x04	0xF1			U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)				
Reply field 2 (function = 2)	0x07	0x8E			U8– Enable Value Float – ZUPT threshold (rad/s)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>

Command	0x75	0x65	0x0D	0x08	0x08	0x20	Fctn (Apply): 0x01 Enable:0x01 (Enable) Threshold: 0x00000000 (0.0f)	0x19	0xC8
Reply ACK/NACK	0x75	0x65	0x0D	0x04	0x04	0xF1	Echo cmd: 0x20 Error code:0x00	0x00	0xF0

Copy-Paste version of the command: "7565 0D08 0820 0101 00000000 19C8"

Tare Orientation (0x0D, 0x21)

Description	This function uses the current device orientation relative to the NED frame as the current sensor to vehicle transformation. This command is provided as a convenient way to set the sensor to vehicle frame transformation.		
Notes	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Possible axis bitfield values:</i></p> <ul style="list-style-type: none"> 0x01 – Tare the roll axis 0x02 – Tare the pitch axis 0x04 – Tare the yaw axis <p>Example Combinations:</p> <ul style="list-style-type: none"> 0x03 – Tare the roll and pitch axis 0x07 – Tare all 3 axis <p>Note: The filter must be in the "running" state to use this command.</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>

<i>Command</i>	0x04	0x21	U8 – Function Selector U8 – Tare Axis Bitfield						
<i>Reply ACK/NACK</i>	0x04	0xF1	U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)						
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x04	0x04	0x21	<i>Fctn (Apply): 0x01 X:0x07 (All axis)</i>	0x18	0x49
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x21 Error code:0x00</i>	0x	0x

Copy-Paste version of the command: "7565 0D04 0421 0107 1849"

Commanded Zero-Velocity Update (0x0D, 0x22)

Description	Perform a commanded ZUPT. The maximum rate for this message is 20 Hz.								
Notes									
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x22			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, non-zero:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x02	0x02	0x22		0x0D	0x17
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x22 Error code: 0x00</i>	0x02	0xF4

Copy-Paste version of the command: "7565 0D02 0222 0D17"

Commanded Zero-Angular Rate Update (0x0D, 0x23)

Description	Perform a commanded zero-angular rate update. The maximum rate for this message is 10 Hz.								
Notes									
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>			<i>Field Data</i>				
<i>Command</i>	0x02	0x23			N/A				
<i>Reply ACK/NACK</i>	0x04	0xF1			U8 – echo the command byte U8 – error code (0:ACK, non-zero:NACK)				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x02	0x02	0x23		0x0E	0x18
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x23 Error code: 0x00</i>	0x03	0xF6

Copy-Paste version of the command: "7565 0D02 0222 0D17"

Declination Source (0x0D, 0x43)

<p>Description</p>	<p>Set/Get the local declination angle source. This can be used to correct for the difference in magnetic and true north. Normally, the device reports heading with-respect-to magnetic north, but when an accurate declination angle is provided, the device will report heading with respect to true north.</p>		
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Possible declination sources:</i></p> <ul style="list-style-type: none"> 0x01 – None 0x02 – World Magnetic Model 0x03 – Manual <p>Option description:</p> <p>None: orientation information will be reported with respect to magnetic north.</p> <p>World Magnetic Model: The declination will be sourced from the device’s internal world magnetic model.</p> <p>Manual: The user provides the declination angle. The device does not validate this angle and it is therefore up to the user to select the correct value.</p>		
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>
<p><i>Command</i></p>	<p>0x08</p>	<p>0x43</p>	<p>U8 – Function Selector U8 – Declination Source Float – Manual Declination angle (radians, only required if source = Manual)</p>
<p><i>Reply ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>	<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>

<i>Reply field 2 (function = 2)</i>	0x05	0xB2	U8 – Declination Source Float – Declination angle (radians)						
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x08	0x08	0x43	<i>Fctn (Apply): 0x01</i> <i>Source (Manual): 0x03</i> <i>Angle:0x00000000 (0.0f)</i>	0x	0x
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd: 0x43</i> <i>Error code:0x00</i>	0x23	0x36

Copy-Paste version of the command: ""

Magnetometer Magnitude Error Adaptive Measurement (0x0D, 0x45)

<p>Description</p>	<p>Set the magnetometer magnitude error adaptive measurement parameters. This function can be used to tune the filter performance in the target application.</p>		
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>Adaptive measurements can be enabled/disabled without the need for providing the additional parameters. In this case, only the function selector and enable value are required; all other parameters will remain at their previous values.</p>		
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>
<p><i>Command</i></p>	<p>0x1C (28)</p>	<p>0x45</p>	<p>U8 – Function Selector U8 – Enable (0 – Disable, 1 – Enable) Float – Low-pass filter cutoff frequency (Hz) Float – Low Limit (Gauss) Float – High Limit (Gauss) Float – Low Limit Uncertainty, 1-Sigma (Gauss) Float – High Limit Uncertainty, 1-Sigma (Gauss) Float – Minimum Uncertainty, 1-Sigma (Gauss)</p>
<p><i>Reply ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>	<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>
<p><i>Reply field 2 (function = 2)</i></p>	<p>0x1B (27)</p>	<p>0xB4</p>	<p>U8 – Enable (0 – Disable, 1 – Enable) Float – Low-pass filter cutoff frequency (Hz) Float – Low Limit (Gauss) Float – High Limit (Gauss) Float – Low Limit Uncertainty, 1-Sigma (Gauss)</p>

							Float – High Limit Uncertainty, 1-Sigma (Gauss) Float – Minimum Uncertainty, 1-Sigma (Gauss)		
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x1C	0x1C	0x45	<i>Fctn (Apply):</i> 0x01 <i>Enable:</i> 0x01 <i>Freq (Hz):</i> (1.0f) <i>Low Limit:</i> (-0.2) <i>High Limit:</i> (0.2f) <i>Low Limit 1 sigma:</i> (0.2f) <i>High Limit 1-sigma:</i> (0.2f) <i>Min 1-sigma:</i> (0.01f)	0x	0x
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd:</i> 0x45 <i>Error code:</i> 0x00	0x	0x

Copy-Paste version of the command: ""

Magnetometer Dip Angle Error Adaptive Measurement (0x0D, 0x46)

<p>Description</p>	<p>Set the magnetometer dip angle error adaptive measurement parameters. This function can be used to tune the filter performance in the target application.</p>		
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p>The adaptive function is linear, taking inputs from 0 to the stated high limit (in radians) and outputs from the minimum uncertainty to the high-limit uncertainty (in Gauss.)</p> <p>Adaptive measurements can be enabled/disabled without the need for providing the additional parameters. In this case, only the function selector and enable value are required; all other parameters will remain at their previous values.</p>		
<p>Field Format</p>	<p><i>Field Length</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>
<p><i>Command</i></p>	<p>0x14 (20)</p>	<p>0x46</p>	<p>U8 – Function Selector U8 – Enable (0 – Disable, 1 – Enable) Float – Low-pass filter cutoff frequency (Hz) Float – High Limit (Radians) Float – High Limit Uncertainty, 1-Sigma (Gauss) Float – Minimum Uncertainty, 1-Sigma (Gauss)</p>
<p><i>Reply ACK/NACK</i></p>	<p>0x04</p>	<p>0xF1</p>	<p>U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)</p>
<p><i>Reply field 2</i></p>	<p>0x13 (19)</p>	<p>0xB5</p>	<p>U8 – Enable (0 – Disable, 1 – Enable) Float – Low-pass filter cutoff frequency (Hz)</p>

(function = 2)							Float – High Limit (Radians) Float – High Limit Uncertainty, 1-Sigma (Gauss) Float – Minimum Uncertainty, 1-Sigma (Gauss)		
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command</i>	0x75	0x65	0x0D	0x14	0x14	0x46	<i>Fctn (Apply):</i> 0x01 <i>Enable:</i> 0x01 <i>Freq (Hz):</i> (10.0f) <i>High Limit (rad):</i> (0.3f) <i>High Limit 1-sigma:</i> (0.2f) <i>Min 1-sigma:</i> (0.01f)	0x	0x
<i>Reply ACK/NACK</i>	0x75	0x65	0x0D	0x04	0x04	0xF1	<i>Echo cmd:</i> 0x46 <i>Error code:</i> 0x00	0x	0x

Copy-Paste version of the command: ""

System Commands

The System Command set provides a set of advanced commands that are specific to devices such as the 3DM-GX3-35 that have multiple intelligent internal sensor blocks. These commands allow special mode such as talking directly to the native protocols of the embedded sensor blocks. For example, with the 3DM-GX3-35, you may switch into a mode that talks directly to the internal u-blox chip or directly to the embedded 3DM-GX3-25 IMU.

Communication Mode (0x7F, 0x10)

Advanced

<p>Description</p>	<p>Set, read, or save the device communication mode. This will change the communications protocol to and from “Estimation Filter” mode to “IMU Direct” (MIP IMU protocol) or “GPS Direct” (u-blox5 protocols). This command is always active, even when switched to the direct modes. This command responds with an ACK/NACK just prior to switching to the new protocol. For all functions except 0x01 (use new settings), the new communications mode value is ignored.</p>														
<p>Notes</p>	<p><i>Possible function selector values:</i></p> <ul style="list-style-type: none"> 0x01 – Use new settings 0x02 – Read back current settings. 0x03 – Save current settings as startup settings 0x04 – Load saved startup settings 0x05 – Reset to factory default settings <p><i>Possible Communications Modes:</i></p> <table border="1" data-bbox="451 1362 1265 1493"> <thead> <tr> <th>Value</th> <th>Mode</th> <th>Protocol(s)</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>Standard</td> <td>3DM-RQ1 MIP Packet (<i>default</i>)</td> </tr> <tr> <td>0x02</td> <td>IMU Direct</td> <td>MIP IMU</td> </tr> <tr> <td>0x03</td> <td>GPS Direct</td> <td>NMEA, UBX</td> </tr> </tbody> </table> <p><i>IMPORTANT: GPS message settings are automatically when switching from direct modes back into standard mode.</i></p> <p><i>Note: Switching to and from GPS Direct Mode takes longer than most commands to complete due to the amount of GPS setup data that needs to be stored/retrieved.</i></p>			Value	Mode	Protocol(s)	0x01	Standard	3DM-RQ1 MIP Packet (<i>default</i>)	0x02	IMU Direct	MIP IMU	0x03	GPS Direct	NMEA, UBX
Value	Mode	Protocol(s)													
0x01	Standard	3DM-RQ1 MIP Packet (<i>default</i>)													
0x02	IMU Direct	MIP IMU													
0x03	GPS Direct	NMEA, UBX													
<p>Field Format</p>	<p><i>Field</i></p>	<p><i>Field Descriptor</i></p>	<p><i>Field Data</i></p>												

	<i>Length</i>								
<i>Command</i>	0x04	0x10		U8 –Function Selector U8 –New Communications Mode					
<i>Reply field 1 ACK/NACK</i>	0x04	0xF1		U8 – echo the command descriptor U8 – error code (0:ACK, not 0:NACK)					
<i>Reply field 2 (function = 2)</i>	0x03	0x90		U8 –Current Communications Mode					
Example	MIP Packet Header				Command/Reply Fields			Checksum	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command COM Mode</i>	0x75	0x65	0x7F	0x04	0x04	0x10	<i>Fctn(USE):0x01 New mode (IMU Direct): 0x02</i>	0x74	0xBD
<i>Reply ACK/NACK</i>	0x75	0x65	0x7F	0x04	0x04	0xF1	<i>Echo cmd:0x10 Error code:0x00</i>	0x62	0x7C

Copy-Paste version of the command: "7565 7F04 0410 0102 74BD"

Hardware Control (0x7F, 0x11)

Description	Control device hardware (only available if the device has a heater)		
Notes	<p><i>Possible function selector values:</i></p> <p>0x06 – Apply new settings (no ACK)</p> <p><i>Possible Control Bitfield Values:</i></p> <p>0x00000001 (Bit 1) – 1 – Heater Enabled, 0 – Heater Disabled (Enabled by default)</p>		
Field Format	<i>Field Length</i>	<i>Field Descriptor</i>	<i>Field Data</i>
<i>Command</i>	0x07	0x11	U8 – Function Selector U32 – Control Bitfield

<i>Reply ACK/NACK</i>	N/A		N/A		N/A				
Example	<i>MIP Packet Header</i>				<i>Fields</i>			<i>Checksum</i>	
	<i>Sync1</i>	<i>Sync2</i>	<i>Desc Set</i>	<i>Payload Length</i>	<i>Field Length</i>	<i>Field Desc.</i>	<i>Field Data</i>	<i>MSB</i>	<i>LSB</i>
<i>Command Heater Off</i>	0x75	0x65	0x7F	0x07	0x07	0x11	<i>Fctn (Apply): 0x06</i> <i>Bitfield:</i> 0x00000000 (Heater off)	0x7E	0x5D

Copy-Paste version of the command: "7565 7F07 0711 0600 0000 007E 5D"

Error Codes

<i>Error Name</i>	<i>Error Value</i>	<i>Description</i>
MIP Unknown Command	0x01	The command descriptor is not supported by this device
MIP Invalid Checksum	0x02	An otherwise complete packet has a bad checksum
MIP Invalid Parameter	0x03	One or more parameters in the packet are invalid. This can refer to a value that is outside the allowed range for a command or a value that is not the expected size or type
MIP Command Failed	0x04	Device could not complete the command
MIP Command Timeout	0x05	Device did not complete the command within the expected time

Data Reference

IMU Data

Scaled Accelerometer Vector (0x80, 0x04)

Description	Scaled Accelerometer Vector					
Notes	This is a vector quantifying the direction and magnitude of the acceleration that the 3DM-RQ1 is exposed to. This quantity 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 3DM-RQ1's local coordinate system.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x04	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Accel	float	g
			4	Y Accel	float	g
			8	Z Accel	float	g

Scaled Gyro Vector (0x80, 0x05)

Description	Scaled Gyro Vector					
Notes	This is a vector quantifying the rate of rotation (angular rate) of the 3DM-RQ1. This quantity is fully temperature compensated and scaled into units of radians/second. It is expressed in terms of the 3DM-RQ1's local coordinate system.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x05	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Gyro	float	Radians/second
			4	Y Gyro	float	Radians/second
			8	Z Gyro	float	Radians/second

Scaled Magnetometer Vector (0x80, 0x06)

Description	Scaled Magnetometer Vector					
Notes	This is a vector which gives the instantaneous magnetometer direction and magnitude. This quantity is fully temperature compensated and scaled into units of Gauss. It is expressed in terms of the 3DM-RQ1's local coordinate system.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x06	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Mag	float	Gauss
			4	Y Mag	float	Gauss
			8	Z Mag	float	Gauss

Scaled Ambient Pressure (0x80, 0x17)

Description	Scaled Ambient Pressure					
Notes	This is a scalar which gives the instantaneous ambient pressure reading. This quantity is fully temperature compensated and scaled into units of milliBar.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	06 (0x06)	0x17	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Ambient Pressure	float	milliBar

Delta Theta Vector (0x80, 0x07)

Description	Time integral of angular rate.					
Notes	This is a vector which gives the time integral of angular rate over the interval set by the IMU message format command. It is expressed in terms of the 3DM-RQ1's local coordinate system in units of radians.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x07	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Delta Theta	float	radians
			4	Y Delta Theta	float	radians
			8	Z Delta Theta	float	radians

Delta Velocity Vector (0x80, 0x08)

Description	Time integral of velocity.					
Notes	This is a vector which gives the time integral of specific acceleration over the interval set by the IMU message format command. It is expressed in terms of the 3DM-RQ1's local coordinate system in units of g*second where g is the standard gravitational constant. To convert Delta Velocity into the more conventional units of m/sec, simply multiply by the standard gravitational constant, 9.80665 m/sec ²					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x08	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Delta Velocity	float	g*seconds
			4	Y Delta Velocity	float	g*seconds
			8	Z Delta Velocity	float	g*seconds

GPS Correlation Timestamp (0x80, 0x12)

Description	GPS correlation timestamp.					
Notes	<p>This timestamp has three fields:</p> <p style="padding-left: 40px;">Double GPS TOW U16 GPS Week number U16 Timestamp flags</p> <p><i>Timestamp Status Flags:</i></p> <p style="padding-left: 40px;">Bit0 – PPS Beacon Good If set, GPS PPS signal is present Bit1 – GPS Time Refresh (toggles with each refresh) Bit2 – GPS Time Initialized (set with the first GPS Time Refresh)</p> <p>This timestamp correlates the IMU packets with the GPS packets. It is identical to the GPS Time record except the flags are defined specifically for the IMU. When the GPS Time Initialized flag is asserted, the GPS Time and IMU GPS Timestamp are correlated. This flag is only set once upon the first valid GPS Time record. After that, each time the GPS Time becomes invalid (from a lack of signal) and then valid again (regains signal) the GPS Time Refresh flag will toggle. The GPS Time Initialized will remain set.</p> <p>The “PPS Beacon Good” flag in the Timestamp flags byte indicates if the PPS beacon coming from the GPS is present. If this flag is not asserted, it means that the IMU internal clock is being used for the PPS. The fractional portion of the GPS TOW represents the amount of time that has elapsed from the last PPS.</p> <p>If the GPS loses signal, the GPS and IMU timestamps become free running and will slowly drift away from each other. If the timestamp clocks have drifted apart, then there will be a jump in the timestamp when the PPS Beacon Good reasserts, reflecting the amount of drift of the clocks.</p> <p>See the Data Synchronicity section of this manual for more information on timestamps.</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x12	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	GPS Time of Week	Double	Seconds
			8	GPS Week Number	U16	
10	Timestamp Flags	U16	See Notes			

GPS Data

LLH Position (0x81, 0x03)

Description	Position Data in the Geodetic Frame					
Notes	Valid Flag Mapping: 0x0001 – Latitude & Longitude Valid 0x0002 – Ellipsoid Height Valid 0x0004 – MSL Height Valid 0x0008 – Horizontal Accuracy Valid 0x0010 – Vertical Accuracy Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	44 (0x2C)	0x03	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Latitude	Double	Decimal Degrees
			8	Longitude	Double	Decimal Degrees
			16	Height above Ellipsoid	Double	Meters
			24	Height above MSL	Double	Meters
			32	Horizontal Accuracy	Float	Meters
			36	Vertical Accuracy	Float	Meters
			40	Valid Flags	U16	See Notes

ECEF Position (0x81, 0x04)

Description	Position Data in the Earth-Centered, Earth-Fixed Frame					
Notes	Valid Flag Mapping: 0x0001 – ECEF Position Valid 0x0002 – Position Accuracy Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	32 (0x20)	0x04	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Position	Double	Meters
			8	Y Position	Double	Meters
			16	Z Position	Double	Meters
			24	Position Accuracy	Float	Meters
28	Valid Flags	U16	See Notes			

NED Velocity (0x81, 0x05)

Description	Velocity Data in the North-East-Down Frame					
Notes	Valid Flag Mapping: 0x0001 – NED Velocity Valid 0x0002 – Speed Valid 0x0004 – Ground Speed Valid 0x0008 – Heading Valid 0x0010 – Speed Accuracy Valid 0x0020 – Heading Accuracy Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	36(0x24)	0x05	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	North	Float	Meters / Sec
4	East	Float	Meters / Sec			

			8	Down	Float	Meters / Sec
			12	Speed	Float	Meters / Sec
			16	Ground Speed	Float	Meters / Sec
			20	Heading	Float	Decimal Degrees
			24	Speed Accuracy	Float	Meters / Sec
			28	Heading Accuracy	Float	Decimal Degrees
			32	Valid Flags	U16	See Notes

ECEF Velocity (0x81, 0x06)

Description	Velocity Data in the Earth-Centered, Earth-Fixed Frame					
Notes	Valid Flag Mapping: 0x0001 – ECEF Velocity Valid 0x0002 – Velocity Accuracy Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	20 (0x14)	0x06	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Velocity	Float	Meters / Sec
			4	Y Velocity	Float	Meters / Sec
			8	Z Velocity	Float	Meters / Sec
			12	Velocity Accuracy	Float	Meters / Sec
16	Valid Flags	U16	See Notes			

DOP Data (0x81, 0x07)

Description	Dilution of Precision Data					
Notes	Valid Flag Mapping: 0x0001 – GDOP Valid 0x0002 – PDOP Valid 0x0004 – HDOP Valid 0x0008 – VDOP Valid 0x0010 – TDOP Valid 0x0020 – NDOP Valid 0x0040 – EDOP Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	32 (0x20)	0x07	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Geometric DOP	Float	N/A
			4	Position DOP	Float	N/A
			8	Horizontal DOP	Float	N/A
			12	Vertical DOP	Float	N/A
			16	Time DOP	Float	N/A
			20	Northing DOP	Float	N/A
			24	Easting DOP	Float	N/A
28	Valid Flags	U16	See Notes			

UTC Time (0x81, 0x08)

Description	Coordinated Universal Time Data					
Notes	Valid Flag Mapping: 0x0001 – Date Valid 0x0002 – Time Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	15 (0x0F)	0x08	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Year	U16	Years (1999-2099)
			2	Month	U8	Months (1-12)
			3	Day	U8	Days (1-31)
			4	Hour	U8	Hours (0-23)
			5	Minute	U8	Minutes (0-59)
			6	Second	U8	Seconds (0-59)
			7	Millisecond	U32	Milliseconds
11	Valid Flags	U16	See Notes			

GPS Time (0x81, 0x09)

Description	Global Positioning System Time Data					
Notes	Valid Flag Mapping: 0x0001 – TOW Valid 0x0002 – Week Number Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x09	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Time of Week	Double	Seconds
			8	Week Number	U16	
		10	Valid Flags	U16	See Notes	

Clock Information (0x81, 0x0A)

Description	Detailed information about the GPS Clock					
Notes	Valid Flag Mapping: 0x0001 – Bias Valid 0x0002 – Drift Valid 0x0004 – Accuracy Estimate Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	28(0x1C)	0x0A	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Clock Bias	Double	Seconds
			8	Clock Drift	Double	Seconds/Second
			16	Accuracy Estimate	Double	Seconds
24	Valid Flags	U16	See Notes			

GPS Fix Information (0x81, 0x0B)

Description	Current GPS Fix Status Information					
Notes	<p>Valid Flag Mapping:</p> <p>0x0001 – Fix Type Valid 0x0002 – Number of SVs Valid 0x0004 – Fix Flags Valid</p> <p>Possible Fix Types values are:</p> <p>0x00 – 3D Fix 0x01 – 2D Fix 0x02 – Time Only 0x03 – None 0x04 – Invalid</p> <p>Possible Fix Flags are:</p> <p>0x0001 – SBAS Corrections Used 0x0002 – Differential (DGPS) Corrections Used</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	8(0x08)	0x0B	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Fix Type	U8	See Notes
			1	Number of SVs used for solution	U8	Count
			2	Fix Flags (Reserved)	U16	N/A
4	Valid Flags	U16	See Notes			

Space Vehicle Information (0x81, 0x0C)

Description	Individual Space Vehicle Information Entry					
Notes	<p>When enabled, these fields will arrive in a separate MIP packet.</p> <p>Valid Flag Mapping:</p> <p style="padding-left: 40px;">0x0001 – Channel Valid 0x0002 – SV ID Valid 0x0008 – Carrier to Noise Ratio Valid 0x0010 – Azimuth Valid 0x0020 – Elevation Valid 0x0040 – SV Flags Valid</p> <p>SV Flag Mapping:</p> <p style="padding-left: 40px;">0x0001 – SV Used for Navigation 0x0002 – SV Healthy</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14(0x0E)	0x0C	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Channel	U8	Channel Number
			1	Space Vehicle ID	U8	SV ID Number
			2	Carrier to Noise Ratio	U16	dBHz
			4	Azimuth	S16	Integer Degrees
			6	Elevation	S16	Integer Degrees
			8	Space Vehicle Flags	U16	See Notes
			10	Valid Flags	U16	See Notes

Hardware Status (0x81, 0x0D)

Description	GPS Hardware Status Information		
Notes	<p>Hardware status is only available at 1 Hz. Setting the rate higher than 1 Hz has no effect.</p> <p>Valid Flag Mapping:</p> <p style="padding-left: 40px;">0x0001 – Sensor State Valid 0x0002 – Antenna State Valid 0x0004 – Antenna Power Valid</p> <p>Possible Sensor State values:</p> <p style="padding-left: 40px;">0x00 – Sensor Off 0x01 – Sensor On 0x02 – Sensor State Unknown</p> <p>Possible Antenna State values:</p> <p style="padding-left: 40px;">0x01 – Antenna Init 0x02 – Antenna Short 0x03 – Antenna Open 0x04 – Antenna Good 0x05 – Antenna State Unknown.</p> <p>Possible Antenna Power values:</p> <p style="padding-left: 40px;">0x00 – Antenna Off 0x01 – Antenna On 0x02 – Antenna Power Unknown</p>		

Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	7(0x07)	0x0D	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Sensor State	U8	See Notes
			1	Antenna State	U8	See Notes
			2	Antenna Power	U8	See Notes
		3	Valid Flags	U16	See Notes	

DGPS Information (0x81, 0x0E)

Description	Individual DGPS Channel Status Entry					
Notes	<p>When enabled, a separate field for each active space vehicle will be sent in the packet.</p> <p>Valid Flag Mapping:</p> <p style="padding-left: 40px;">0x0001 – Latest Age Valid 0x0002 – Base Station ID Valid 0x0004 – Base Station Status Valid 0x0008 – Number of DGPS Channels Valid</p> <p>Possible Base Station Status Values:</p> <p style="padding-left: 40px;">0 – UDRE Scale Factor = 1.0 1 – UDRE Scale Factor = 0.75 2 – UDRE Scale Factor = 0.5 3 – UDRE Scale Factor = 0.3 4 – UDRE Scale Factor = 0.2 5 – UDRE Scale Factor = 0.1 6 – Reference Station Transmission Not Monitored 7 – Reference Station Not Working</p> <p>Note: UDRE = User Differential Range Error</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x0E	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Newest Age	float	Seconds
			4	Base Station ID	S16	N/A
			6	Base Station Status	S16	N/A
			8	Number of DGPS Channels	U16	Number
10	Valid Flags	U16	See Notes			

DGPS Channel Status (0x81, 0x0F)

Description	Individual DGPS Channel Status Entry					
Notes	<p>When enabled, a separate field for each active space vehicle will be sent in the packet.</p> <p>Valid Flag Mapping:</p> <p>0x0001 – SV ID Valid 0x0002 – Age Valid 0x0004 – Pseudorange Correction Valid 0x0008 – Pseudorange Rate Correction Valid</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	17(0x11)	0x0F	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Space Vehicle ID	U8	SV ID Number
			1	Age	float	Seconds
			5	Pseudorange Correction	float	Meters
			9	Pseudorange Rate Correction	float	Meters/Sec
13	Valid Flags	U16	See Notes			

Estimation Filter Data

Filter Status (0x82, 0x10)

Description	Estimation Filter Status
<p>Notes</p>	<p>Possible Filter States:</p> <ul style="list-style-type: none"> 0x00 – Startup 0x01 – Initialization (see status flags) 0x02 – Running, Solution Valid 0x03 – Running, Solution Error (see status flags) <p>Possible Dynamics Modes:</p> <ul style="list-style-type: none"> 0x01 – Portable 0x02 – Automotive 0x03 – Airborne <p>Possible Status Flags:</p> <p>Filter State = Initialization:</p> <ul style="list-style-type: none"> 0x1000 – Attitude not initialized 0x2000 – Position & Velocity not initialized <p>Filter State = Running:</p> <ul style="list-style-type: none"> 0x0001 – IMU Unavailable 0x0002 – GPS Unavailable 0x0008 – Matrix Singularity in calculation 0x0010 – Position Covariance High Warning* 0x0020 – Velocity Covariance High Warning* 0x0040 – Attitude Covariance High Warning* 0x0080 – NAN in Solution 0x0100 – Gyro bias estimate high warning 0x0200 – Accel bias estimate high warning 0x0400 – Gyro scale factor estimate high warning 0x0800 – Accel scale factor estimate high warning 0x2000 – GPS Antenna Offset Correction estimate high warning <p>*Note: The covariance high warnings are triggered when any axis of the</p>

	covariance vector exceeds normal operating limits. If more information is required, please inspect the relevant uncertainty packet to determine which axis is in error.					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	08 (0x08)	0x10	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Filter State	U16	See Notes
			2	Dynamics Mode	U16	See Notes
		4	Status Flags	U16	See Notes	

GPS Timestamp (0x82, 0x11)

Description	Estimation Filter Calculated Value Timestamp Data					
Notes	Valid Flag Mapping: 0x0000 – Time Invalid 0x0001 – Time Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	14 (0x0E)	0x11	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Time of Week	Double	Seconds
			8	Week Number	U16	
		10	Valid Flags	U16	See Notes	

LLH Position (0x82, 0x01)

Description	Estimated Position Data expressed in the Geodetic Frame					
Notes	Valid Flag Mapping: 0x0000 – Latitude, Longitude, & Height are Invalid 0x0001 – Latitude, Longitude, & Height Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	28 (0x1C)	0x01	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Latitude	Double	Decimal Degrees
			8	Longitude	Double	Decimal Degrees
			16	Height above Ellipsoid	Double	Meters
24	Valid Flags	U16	See Notes			

NED Velocity (0x82, 0x02)

Description	Estimated Velocity Data expressed in the Local-Level Frame					
Notes	Valid Flag Mapping: 0x0000 – NED Velocity is Invalid 0x0001 – NED Velocity Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x02	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	North	Float	Meters / Sec
			4	East	Float	Meters / Sec
			8	Down	Float	Meters / Sec
12	Valid Flags	U16	See Notes			

Orientation, Quaternion (0x82, 0x03)

Description	Estimated Orientation in quaternion form.					
Notes	<p>This is a 4 component quaternion which describes the orientation of the 3DM-GX3-45 with respect to the fixed earth coordinate quaternion.</p> $Q = \begin{bmatrix} q0 \\ q1 \\ q2 \\ q3 \end{bmatrix}$ <p>Q satisfies the following equation:</p> $V_E = Q \cdot V_IL \cdot Q^{-1}$ <p>Where: <i>V_IL</i> is a vector expressed in the 3DM-GX3[®]'s local coordinate system. <i>V_E</i> is the same vector expressed in the stationary, earth-fixed coordinate system</p> <p>Valid Flag Mapping:</p> <p>0x0000 – Quaternion is Invalid 0x0001 – Quaternion Valid 0x0002 – Quaternion is referenced to magnetic north</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	20 (0x14)	0x03	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	q ₀	float	n/a
			4	q ₁ * i	float	n/a
			8	q ₂ * j	float	n/a
			12	q ₃ * k	float	n/a
16	Valid Flags	U16	See Notes			

Orientation, Matrix (0x82, 0x04)

Description	Estimated Orientation in Matrix form.					
Notes	<p>This is a 9 component coordinate transformation matrix which describes the orientation of the 3DM-GX3[®] with respect to the fixed earth coordinate system.</p> $M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$ <p><i>M</i> satisfies the following equation:</p> $V_IL_i = M_{ij} \cdot V_E_j$ <p>Where: <i>V_IL</i> is a vector expressed in the 3DM-GX3[®]'s local coordinate system. <i>V_E</i> is the same vector expressed in the stationary, earth-fixed coordinate system</p> <p>Valid Flag Mapping:</p> <p>0x0000 – Orientation Matrix is Invalid 0x0001 – Orientation Matrix Valid 0x0002 – Orientation Matrix is referenced to magnetic north</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	40 (0x28)	0x04	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
0			M ₁₁	float	n/a	
4			M ₁₂	float	n/a	
8			M ₁₃	float	n/a	
12			M ₂₁	float	n/a	
16			M ₂₂	float	n/a	
20			M ₂₃	float	n/a	
24			M ₃₁	float	n/a	
28			M ₃₂	float	n/a	
32	M ₃₃	float	n/a			

			36	Valid Flags	U16	See Notes
--	--	--	----	-------------	-----	-----------

Orientation, Euler Angles (0x82, 0x05)

Description	Estimated Pitch, Roll, and Yaw (aircraft) angles					
Notes	<p>This is a 3 component vector containing the Roll, Pitch and Yaw angles in radians. It is computed by the INS from the orientation quaternion Q.</p> $Euler = \begin{bmatrix} Roll \\ Pitch \\ Yaw \end{bmatrix} \text{ (radians)}$ <p>Valid Flag Mapping:</p> <p>0x0000 – Euler Angles are Invalid 0x0001 – Euler Angles Valid 0x0002 – Euler Angles are referenced to magnetic north</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x05	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Roll	float	radians
			4	Pitch	float	radians
			8	Yaw	float	radians
		12	Valid Flags	U16	See Notes	

Gyro Bias (0x82, 0x06)

Description	Estimated Gyro Biases expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Gyro Bias are Invalid 0x0001 – Gyro Bias Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x06	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Gyro Bias	float	radians/sec
			4	Y Gyro Bias	float	radians/sec
			8	Z Gyro Bias	float	radians/sec
12	Valid Flags	U16	See Notes			

Accel Bias (0x82, 0x07)

Description	Estimated Accel Biases expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Accel Bias are Invalid 0x0001 – Accel Bias Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x07	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Accel Bias	float	Meters / Sec ²
			4	Y Accel Bias	float	Meters / Sec ²
			8	Z Accel Bias	float	Meters / Sec ²
12	Valid Flags	U16	See Notes			

LLH Position Uncertainty (0x82, 0x08)

Description	Estimated Position 1-sigma Uncertainty expressed in the Geodetic Frame					
Notes	Valid Flag Mapping: 0x0000 – Position Uncertainties are Invalid 0x0001 – Position Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x08	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Local-Level, 1-Sigma Position Uncertainty (North)	Float	Meters
			4	Local-Level, 1-Sigma Position Uncertainty (East)	Float	Meters
			8	Local-Level, 1-Sigma Position Uncertainty (Down)	Float	Meters
12	Valid Flags	U16	See Notes			

NED Velocity Uncertainty (0x82, 0x09)

Description	Estimated Velocity 1-sigma Uncertainty expressed in the Local-Level Frame					
Notes	Valid Flag Mapping: 0x0000 – NED Velocity Uncertainties are Invalid 0x0001 – NED Velocity Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x09	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Local-Level, 1-Sigma Velocity Uncertainty	Float	Meters / Sec

				(North)		
			4	Local-Level, 1-Sigma Velocity Uncertainty (East)	Float	Meters / Sec
			8	Local-Level, 1-Sigma Velocity Uncertainty (Down)	Float	Meters / Sec
			12	Valid Flags	U16	See Notes

Attitude Uncertainty, Euler Angles (0x82, 0x0A)

Description	Estimated attitude 1-sigma uncertainty expressed in Pitch, Roll, and Yaw (aircraft) elements.					
Notes	<p>This is a 3 component vector containing the Roll, Pitch and Yaw angle uncertainties in radians.</p> <p>IMPORTANT: These values are derived from the quaternion elements and become increasingly inaccurate as the pitch angle approaches +-90 degrees. To compensate for this limitation, these values will be marked as invalid when the pitch angle exceeds +-70 degrees.</p> <p>Valid Flag Mapping:</p> <p>0x0000 – Attitude Uncertainties are Invalid 0x0001 – Attitude Uncertainties Valid</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0A	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Attitude Uncertainty (Roll)	float	radians
			4	1-Sigma Attitude Uncertainty	float	radians

				(Pitch)		
			8	1-Sigma Attitude Uncertainty (Yaw)	float	radians
			12	Valid Flags	U16	See Notes

Gyro Bias Uncertainty (0x82, 0x0B)

Description	Estimated Gyro Bias 1-sigma Uncertainty expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Gyro Bias Uncertainties are Invalid 0x0001 – Gyro Bias Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0B	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Gyro Bias Uncertainty (X)	float	radians/sec
			4	1-Sigma Gyro Bias Uncertainty (Y)	float	radians/sec
			8	1-Sigma Gyro Bias Uncertainty (Z)	float	radians/sec
		12	Valid Flags	U16	See Notes	

Accel Bias Uncertainty (0x82, 0x0C)

Description	Estimated Accel Bias 1-sigma Uncertainty expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Accel Bias Uncertainties are Invalid 0x0001 – Accel Bias Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0C	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Accel Bias Uncertainty (X)	float	Meters / Sec ²
			4	1-Sigma Accel Bias Uncertainty (Y)	float	Meters / Sec ²
			8	1-Sigma Accel Bias Uncertainty (Z)	float	Meters / Sec ²
12	Valid Flags	U16	See Notes			

Linear Acceleration (0x82, 0x0D)

Description	Filter-Compensated Linear Acceleration Data (gravity vector removed) expressed in: 1) The Sensor Frame, if no sensor to body rotation has been defined. 2) The Vehicle Frame, if a sensor to body rotation has been defined.					
Notes	Valid Flag Mapping: 0x0000 – Linear Accelerations are Invalid 0x0001 – Linear Accelerations are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0D	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>

			0	X	Float	Meters / Sec ²
			4	Y	Float	Meters / Sec ²
			8	Z	Float	Meters / Sec ²
			12	Valid Flags	U16	See Notes

Compensated Acceleration (0x82, 0x1C)

Description	Filter-Compensated Acceleration Data expressed in: 1) The Sensor Frame, if no sensor to body rotation has been defined. 2) The Vehicle Frame, if a sensor to body rotation has been defined.					
Notes	Valid Flag Mapping: 0x0000 – Compensated Accelerations are Invalid 0x0001 – Compensated Accelerations are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0D	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X	Float	Meters / Sec ²
			4	Y	Float	Meters / Sec ²
			8	Z	Float	Meters / Sec ²
		12	Valid Flags	U16	See Notes	

Compensated Angular Rate (0x82, 0x0E)

Description	Filter-Compensated Angular Rate Data expressed in: 1) The Sensor Frame, if no sensor to body rotation has been defined. 2) The Vehicle Frame, if a sensor to body rotation has been defined.					
Notes	The estimated gyro bias has been removed from these angular rate values. Valid Flag Mapping: 0x0000 – Angular Rates are not Valid 0x0001 – Angular Rates are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x0E	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X	Float	Radians / Sec
			4	Y	Float	Radians / Sec
			8	Z	Float	Radians / Sec
12	Valid Flags	U16	See Notes			

WGS84 Local Gravity Magnitude (0x82, 0x0F)

Description	Local Magnitude of Earth’s gravity using the WGS84 gravity model.					
Notes	The GX3-45 implements the WGS84 gravity model, valid for altitudes of 20km or less. Valid Flag Mapping: 0x0000 – Gravity value is Invalid 0x0001 – Gravity value is Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	08(0x08)	0x0F	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Gravity Magnitude	Float	meters / sec ²
			4	Valid Flags	U16	See Notes

Attitude Uncertainty, Quaternion Elements (0x82, 0x12)

Description	Estimated attitude 1-sigma uncertainty expressed in quaternion components.					
Notes	<p>This is a 4 component vector containing the attitude uncertainty expressed in quaternion elements.</p> <p>Valid Flag Mapping:</p> <p style="padding-left: 40px;">0x0000 – Attitude uncertainties are Invalid</p> <p style="padding-left: 40px;">0x0001 – Attitude uncertainties are Valid</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	18 (0x12)	0x12	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Attitude Uncertainty (q0)	float	
			4	1-Sigma Attitude Uncertainty (q1)	float	
			8	1-Sigma Attitude Uncertainty (q2)	float	
			12	1-Sigma Attitude Uncertainty (q3)	float	
16	Valid Flags	U16	See Notes			

Gravity Vector (0x82, 0x13)

Description	Estimated Gravity Vector expressed in: 1) The Sensor Frame, if no sensor to body rotation has been defined. 2) The Vehicle Frame, if a sensor to body rotation has been defined.					
Notes	Valid Flag Mapping: 0x0000 – Gravity vector is Invalid 0x0001 – Gravity vector is Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x13	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X	Float	Meters / Sec ²
			4	Y	Float	Meters / Sec ²
			8	Z	Float	Meters / Sec ²
		12	Valid Flags	U16	See Notes	

Heading Update Source State (0x82, 0x14)

Description	Heading Update Source information expressed in the sensor frame.
Notes	<p>Heading updates can be applied from a number of sources (listed below.)</p> <p>The heading value is always relative to true north.</p> <p>Possible Sources:</p> <ul style="list-style-type: none"> 0x0000 – No source, heading updates disabled 0x0001 – Internal Magnetometer 0x0002 – Internal GPS Velocity Vector 0x0004 – External Heading Update Command <p>Valid Flag Mapping:</p> <ul style="list-style-type: none"> 0x0000 – No heading update received in 2 seconds. 0x0001 – The heading update source has provided data within 2 seconds.

Field Format	Field Length	Data Descriptor	Message Data			
	14(0x0E)	0x14	Binary Offset	Description	Data Type	Units
			0	Heading (True)	Float	Radians
			4	Heading 1-sigma Uncertainty	Float	Radians
			8	Source	U16	See Notes
10	Valid Flags	U16	See Notes			

Magnetic Model Solution (0x82, 0x15)

Description	Magnetic model solution expressed in the NED frame.					
Notes	<p>The World Magnetic Model 2010 is used. A valid GPS location is required for the model to be valid.</p> <p>Valid Flag Mapping:</p> <p>0x0000 – Magnetic model solution is invalid (note: this will be the state when the magnetic model is recalculating for the current time and location as well as when GPS is unavailable)</p> <p>0x0001 – Magnetic model solution is valid</p>					
Field Format	Field Length	Data Descriptor	Message Data			
	24 (0x18)	0x15	Binary Offset	Description	Data Type	Units
			0	Intensity (North)	Float	Gauss
			4	Intensity (East)	Float	Gauss
			8	Intensity (Down)	Float	Gauss
			12	Inclination	Float	Radians
			16	Declination	Float	Radians
20	Valid Flags	U16	See Notes			

Gyro Scale Factor (0x82, 0x16)

Description	Estimated Gyro Scale Factor expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Scale Factor values are Invalid 0x0001 – Scale Factor values are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x16	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Scale Factor	float	%/100
			4	Y Scale Factor	float	%/100
			8	Z Scale Factor	float	%/100
12	Valid Flags	U16	See Notes			

Accel Scale Factor (0x82, 0x17)

Description	Estimated Accel Scale Factor expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Scale Factor values are Invalid 0x0001 – Scale Factor values are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x17	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Scale Factor	float	%/100
			4	Y Scale Factor	float	%/100
			8	Z Scale Factor	float	%/100
12	Valid Flags	U16	See Notes			

Gyro Scale Factor Uncertainty (0x82, 0x18)

Description	Estimated Gyro Scale Factor 1-Sigma Uncertainty expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Gyro Scale Factor Uncertainties are Invalid 0x0001 – Gyro Scale Factor Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x18	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Gyro Scale Factor Uncertainty (X)	float	%/100
			4	1-Sigma Gyro Scale Factor Uncertainty (Y)	float	%/100
			8	1-Sigma Gyro Scale Factor Uncertainty (Z)	float	%/100
		12	Valid Flags	U16	See Notes	

Accel Scale Factor Uncertainty (0x82, 0x19)

Description	Estimated Accel Scale Factor 1-Sigma Uncertainty expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – Accel Scale Factor Uncertainties are Invalid 0x0001 – Accel Scale Factor Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x19	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Accel Scale Factor Uncertainty (X)	float	%/100
			4	1-Sigma Accel Scale Factor Uncertainty (Y)	float	%/100
			8	1-Sigma Accel Scale Factor Uncertainty (Z)	float	%/100
12	Valid Flags	U16	See Notes			

Standard Atmosphere Model (0x82, 0x20)

Description	Standard Atmosphere Model Solution.					
Notes	The US 1976 Standard Atmosphere Model is used. A valid GPS location is required for the model to be valid. Valid Flag Mapping: 0x0000 – Atmosphere model solution is invalid (note: this will be the state when GPS is unavailable) 0x0001 – Atmosphere model solution is valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			

			<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
	24 (0x18)	0x20	0	Geometric Altitude	Float	meters
			4	Geopotential Altitude	Float	meters
			8	Temperature	Float	degC
			12	Pressure	Float	mBar
			16	Density	Float	kg/m ³
			20	Valid Flags	U16	See Notes

Pressure Altitude (0x82, 0x21)

Description	Estimated Pressure Altitude.					
Notes	<p>The US 1976 Standard Atmosphere Model is used to calculate the pressure altitude in meters. A valid pressure sensor reading is required for the pressure altitude to be valid. The minimum pressure reading supported by the model is 0.0037 mBar, corresponding to an altitude of 84,852 meters.</p> <p>Valid Flag Mapping:</p> <p style="padding-left: 40px;">0x0000 – Pressure Altitude is Invalid</p> <p style="padding-left: 40px;">0x0001 – Pressure Altitude is Valid</p>					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	8 (0x08)	0x21	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	Pressure Altitude	float	meters
			4	Valid Flags	U16	See Notes

GPS Antenna Offset Correction (0x82, 0x30)

Description	Estimated GPS Antenna Offset Correction expressed in the Sensor Body Frame as the vector from the IMU to the GPS Antenna.					
Notes	Valid Flag Mapping: 0x0000 – GPS Antenna Offset Correction values are Invalid 0x0001 – GPS Antenna Offset Correction values are Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x30	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	X Offset Correction	float	meters
			4	Y Offset Correction	float	meters
			8	Z Offset Correction	float	meters
12	Valid Flags	U16	See Notes			

GPS Antenna Offset Correction Uncertainty (0x82, 0x31)

Description	Estimated GPS Antenna Offset Correction Uncertainty expressed in the Sensor Body Frame.					
Notes	Valid Flag Mapping: 0x0000 – GPS Antenna Offset Correction Uncertainties are Invalid 0x0001 – GPS Antenna Offset Correction Uncertainties Valid					
Field Format	<i>Field Length</i>	<i>Data Descriptor</i>	<i>Message Data</i>			
	16 (0x10)	0x31	<i>Binary Offset</i>	<i>Description</i>	<i>Data Type</i>	<i>Units</i>
			0	1-Sigma Offset Correction Uncertainty (X)	float	meters
			4	1-Sigma Gyro Scale Factor Uncertainty	float	meters

				(Y)		
			8	1-Sigma Offset Correction Uncertainty (Z)	float	meters
			12	Valid Flags	U16	See Notes

MIP Packet Reference

Structure

Commands and Data are sent and received as fields in the MicroStrain “MIP” packet format. Below is the general definition of the structure:

Header				Payload			Checksum	
SYNC1 “u”	SYNC2 “e”	Descriptor Set byte	Payload Length byte	Fields			MSB	LSB
0x75	0x65	<desc set selector>	$k_1+k_2+\dots+k_n$	MIP Field 1 length = k_1	...	MIP Field n length = k_n	0xMM	0xLL

Field Header		Field Data
Field Length byte	Field Descriptor byte	Field Data
k_n	<descriptor>	< k_n-2 bytes of data>

The packet always begins with the start-of-packet sequence “ue” (0x75, 0x65). The “Descriptor Set” byte in the header specifies which command or data set is contained in fields of the packet. The payload length byte specifies the sum of all the field length bytes in the payload section.

Payload Length Range

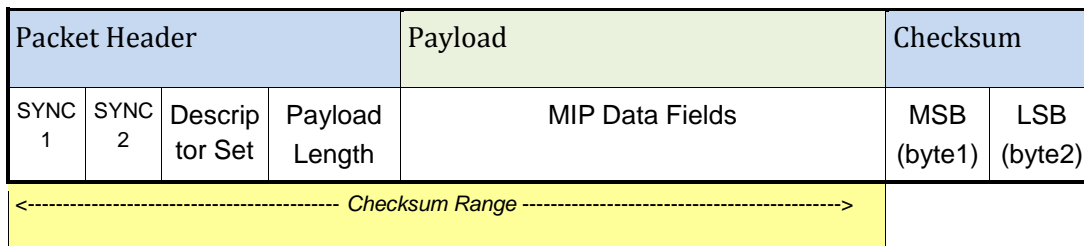
Packet Header				Payload		Checksum	
SYN C1	SYN C2	Descript or Set	Payload Length	MIP Data Fields		MSB	LSB
				<-----Payload Length Range ----->			

The payload section can be empty or can contain one or more fields. Each field has a length byte and a descriptor byte. The field length byte specifies the length of the entire field including the field length byte and field descriptor byte. The descriptor byte specifies the command or data that is contained in the field data. The descriptor can only be from the set of descriptors specified by the descriptor set byte in the header. The field data can be anything but is always rigidly defined. The definition of a descriptor is fundamentally described in a “.h” file that corresponds to the descriptor set that the descriptor belongs to.

MicroStrain provides a “Packet Builder” functionality in the “MIP Monitor” software utility to simplify the construction of a MIP packet. Most commands will have a single field in the packet, but multiple field packets are possible. Extensive examples complete with checksums are given in the command reference section.

Checksum Range

The checksum is a 2 byte Fletcher checksum and encompasses all the bytes in the packet:



16-bit Fletcher Checksum Algorithm (C language)

```
for(i=0; i<checksum_range; i++)
{
    checksum_byte1 += mip_packet[i];
    checksum_byte2 += checksum_byte1;
}
```

```
checksum = ((u16) checksum_byte1 << 8) + (u16) checksum_byte2;
```

Advanced Programming

Multiple Commands in a Single Packet

MIP packets may contain one or more individual commands. In the case that multiple commands are transmitted in a single MIP packet, the 3DM-RQ1-45 will respond with a single packet containing multiple replies. As with any packet, all commands must be from the same descriptor set (you cannot mix Base commands with 3DM commands in the same packet).

Below is an example that shows how you can combine the commands from step 2 and 3 of the [Example Setup Sequence](#) into a single packet. The commands are from the 3DM set. The command packet has two fields as does the reply packet (the fields are put on separate rows for clarity):

Step 2 and 3	MIP Packet Header				Command/Reply Fields			Checksum	
	Sync1	Sync2	Desc Set	Payload Length	Field Length	Cmd Desc.	Field Data	MSB	LSB
Command field 1 Set IMU Message Format	0x75	0x65	0x0C	0x14	0x0A	0x08	Function: 0x00 Desc count: 0x02 1 st Descriptor: 0x03 Rate Dec: 0x000A 2 nd Descriptor: 0x04 Rate Dec: 0x000A		
Command field 2 Set GPS Message Format					0x0A	0x09	Function: 0x00 Desc Count: 0x02 ECEP posdesc: 0x04 Rate dec: 0x0004 ECEP veldec: 0x06 Rate dec: 0x0004	0x50	0x98
Replyfield 1 ACK/NACK	0x75	0x65	0x0C	0x08	0x04	0xF1	Cmd echo: 0x08 Error code: 0x00		
Reply field 2 ACK/NACK					0x04	0xF1	Cmd echo: 0x09 Error code: 0x00	0xE9	0x6F

Copy-Paste version of the command: "7565 0C14 0A08 0002 0300 0A04 000A 0A09 0002 0400 0406 0004 5098"

Note that the only difference in the packet headers of the single command packets compared to the multiple command packets is the payload length. Parsing multiple fields in a single packet involves subtracting the field length of the next field from the payload length until the payload length is less than or equal to zero.

Direct Modes

The 3DM-RQ1-45 has special “direct” modes that switch the device into an IMU direct or GPS direct device. The [Device Communications Mode](#) command is used to switch between modes. When in these modes, the 3DM-RQ1-45 acts like a an “IMU only” or a u-blox GPS sensor respectively. Any code or tools developed for these devices may be used in these modes. For example, when in the “u-blox” direct mode, the u-blox “u-center” application works perfectly with the GPS chip embedded in the 3DM-RQ1-45.

These modes can be used to access advanced (native) data of the individual sensors, data that isn’t represented in the 3DM command sets of the GX3-45.

IMPORTANT: When you switch modes, you are switching to a new device protocol EXCEPT for two commands: the [Device Communications Mode](#) and [Device Status](#) commands. Those commands are always available regardless of which mode you are in. For example, if you switch to GPS direct mode, then the protocol recognized by the device is NMEA and UBX protocol, however the 3DM-RQ1-45 is still “listening” for mode switch or device status commands and will respond to them. It will not respond to any other 3DM-RQ1-45 Base or 3DM commands until switched back to the “Standard Mode”.

IMPORTANT: The GPS message settings required for Estimation Filter execution are automatically reloaded when switching from direct modes back in to standard mode.

Internal Diagnostic Functions

The 3DM-RQ1-45 supports two device specific internal functions used for diagnostics and system status. These are [Device Built In Test](#) and [Device Status](#). These commands are defined generically but the implementation is very specific to the hardware implemented on this device. Other MicroStrain devices will have their own implementations of these functions depending on the internal hardware of the devices.

3DM-RQ1-45 Internal Diagnostic Commands

- [Device Built In Test](#) (0x01, 0x05)
- [Device Status](#) (0x0C, 0x64)

Handling High Rate Data

The size of the data fields from an inertial device is substantially greater than on most other types of sensors. On top of that, in many applications it is desirable to receive that data with the lowest latency possible and thus the highest BAUD rate is selected. The result is that the port servicing requirements in terms of both speed and buffer size can be surprisingly large for inertial data. This can lead to a couple of common problems: runaway latency and dropped packets.

Runaway latency

Most operating systems provide drivers that have ample buffers and take care of port servicing at the hardware level. Dropping packets or losing data is not usually an issue on these systems. What can be an issue is latency, that is, when the buffer is not emptied by the application in a timely manner. In the worst case, the buffer is being filled faster than it is emptied and the application operates with increasingly “old” data – which causes runaway latency. It is important to monitor the incoming data buffer to make sure you do not reach this condition.

Dropped packets

Many applications do not use an operating system but are written from scratch or on top of proprietary application frameworks. These are most often embedded MCUs or small single board microcontrollers. On these systems, port handling is usually done in code at the hardware level. Collecting data from a port requires the use one of three techniques: register polling, hardware interrupts, or direct memory access (DMA). Register polling is very easy to do and is adequate for simple communications where data comes in very small chunks and at reasonable data rates. The problem with register polling is that you either waste time looping while waiting for a byte to come in at the port or you get too busy doing other tasks so that by the time you poll the port, the byte is lost because the next one overwrites it. This causes dropped packets. On these systems, it is imperative to utilize either a hardware interrupt or hardware DMA on the UART receiving data from the 3DM-RQ1. The DMA or UART interrupt service routine only takes processor time when a byte is ready and as long as the interrupts are preemptive, the processor will fetch every byte received. Using the interrupt routine to fill a ring buffer makes the most efficient use of an MCU and makes it easier to write your application main line code. This is essentially what drivers in operating systems do.

Creating Fixed Data Packet Format

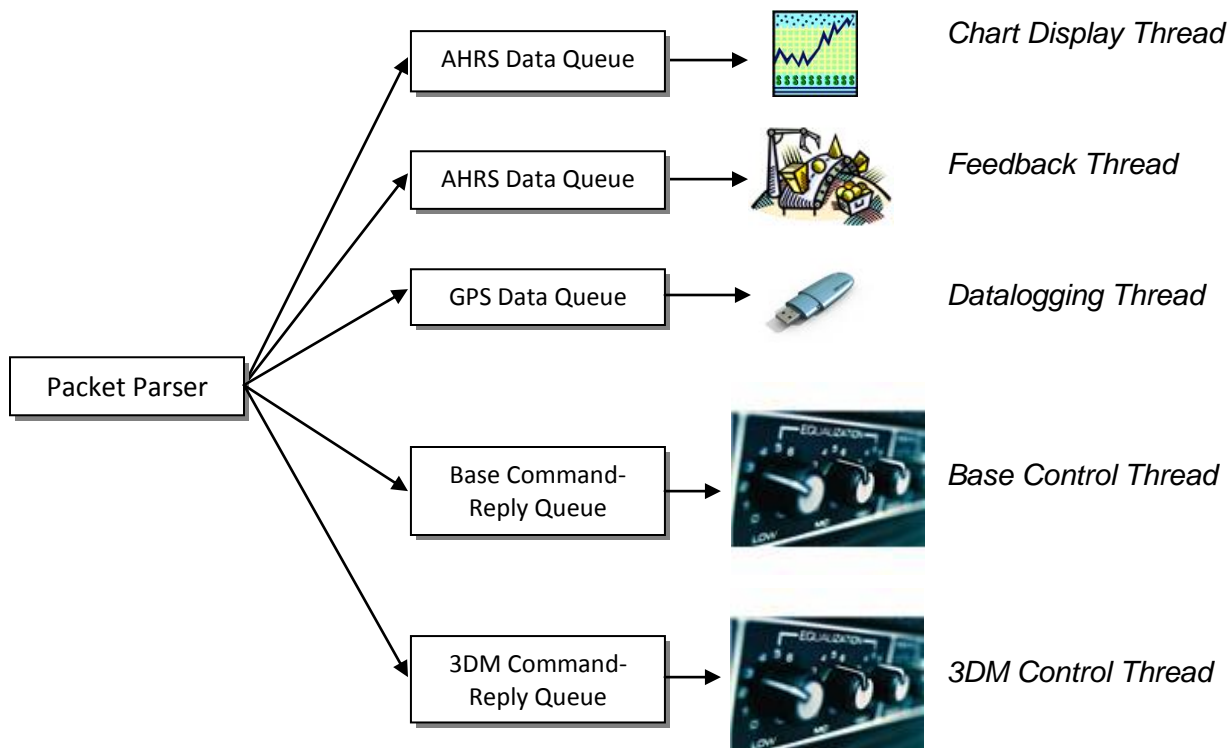
The MIP packet structure and protocol provides a great deal of flexibility to the user for creating a custom data stream. It does this by allowing selectable data fields and individual data rates for each field. The side effect of this feature is that packets vary in size depending on what data is being delivered in any particular time frame. For example, if acceleration data is configured for 100Hz and magnetometer data is configured for 25Hz, every fourth packet is larger than the previous three because of the additional magnetometer data. In some applications, this is undesirable and there may be a requirement for a fixed packet structure so that each data packet is exactly the same. A fixed packet structure allows you to find data fields by fixed offsets rather than parsing the packet for each field.

A fixed packet structure is easily achieved with MIP packet protocol by simply making sure the data rate for each data quantity is the same. The order of the data fields in the packet reflect the order of the fields in the [message format](#) command and thus are completely under the control of the user. Once an acceptable data packet structure is determined, and all the rates are set to the same decimation, use the “Save current settings as startup settings”

function selector in the message format command, and that format will be saved and used automatically on subsequent device startups. The message formats for each of the data classes (IMU, GPS, EF, etc) work the same way, however the available data rates for each class is different, so you will need to create a fixed message format for each one.

Advanced Programming Models

Many applications will only require a single threaded programming model which is simple to implement using a single program loop that services incoming packets. In other applications, advanced techniques such as multithreading or event based processes are required. The MIP packet design simplifies implementation of these models. It does this by limiting the packet size to a maximum of 261 bytes and it provides the “descriptor set” byte in the header. The limited packet size makes scalable packet buffers possible even with limited memory space. The descriptor set byte aids in sorting an incoming packet stream into one or more command-reply packet queues and/or data packet queues. A typical multithreaded environment will have a command/control thread and one or more data processing threads. Each of these threads can be fed with individual incoming packet queues, each containing packets that only pertain to that thread – sorted by descriptor set. Packet queues can easily be created dynamically as threads are created and destroyed. All packet queues can be fed by a single incoming packet parser that runs continuously independent of the queues. The packet queues are individually scaled as appropriate to the process; smaller queues for lower latency and larger queues for more efficient batch processing of packets.



Multithreaded application with multiple incoming packet queues

End of Document