Crossbow®

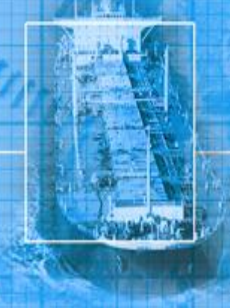
440 Series Inertial Systems

www.xbow.com

440 Series User's Manual

Document 7430-0131-01

Revision E, September 2010



Crossbow Technology, Inc., 1421 McCarthy Blvd., Milpitas, CA 95035
Tel: 408-965-3300, Fax: 408-324-4840
email: info@xbow.com, website: www.xbow.com

⚠ WARNING

This product has been developed by Crossbow exclusively for commercial applications. It has not been tested for, and Crossbow makes no representation or warranty as to conformance with, any military specifications or that the product is appropriate for any military application or end-use. Additionally, any use of this product for nuclear, chemical, biological weapons, or weapons research, or for any use in missiles, rockets, and/or UAV's of 300km or greater range, or any other activity prohibited by the Export Administration Regulations, is expressly prohibited without the written consent of Crossbow and without obtaining appropriate US export license(s) when required by US law. Diversion contrary to U.S. law is prohibited.

Table of Contents

About this Manual

The following annotations have been used to provide additional information.

◀ NOTE

Note provides additional information about the topic.

☑ EXAMPLE

Examples are given throughout the manual to help the reader understand the terminology.

⚐ IMPORTANT

This symbol defines items that have significant meaning to the user

⚠ WARNING

The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1

1.1 Heading 2

1.1.1 Heading 3

Normal

1 Introduction

1.1 Manual Overview

This manual provides a comprehensive introduction to Crossbow's 440 Series Inertial System products. For users wishing to get started quickly, please refer to the three page quick start guide included with each shipment. The following table highlights the content in each section and suggests how to use this manual.

Table 1: Manual Content

Manual Section	**Who Should Read ?**
Section 1: Manual Overview	All customers should read sections 1.1 and 1.2. Customers who have previously used Crossbow's 300/400 Series or 420 Series products should also read Section 1.3.
Section 2: Connections	Customers who are connecting the 440 Series products into a system with their own power supply and cable.
Section 3: Installation and Operation of NAV-VIEW 2.X	Customers who are installing the 440 Series products into a system and need details on using NAV-VIEW 2.X.
Section 4: Theory of Operation	All customers should read Section 4. As the 440 Series products are inter-related, use the chart at the beginning of Section 4 to ensure that you get an overview of all of the functions and features of your 440 Series system. For example, if you have purchased a NAV440, you should read not only the section on the NAV440, but also familiarize yourself with the theory of operation for the IMU440, VG440, and AHRS440. The NAV440 builds on the capabilities of the IMU440, VG440 and AHRS440.
Section 5: Application Guide	Customers who want product configuration tips for operating the 440 Series Inertial Systems in a wide range of applications – fixed wing, rotary wing, unmanned vehicles, land vehicles, marine vessels, and more. Review the part of Section 5 that is relevant to your application. Note: NAV and AHRS 440 Series units are preconfigured for airborne applications with "normal" dynamics. VG440 Series units are preconfigured for land applications with "automotive testing" dynamics. All 440 series products allow for complete flexibility in configuration by the user.
Section 6-9: Programming, Communicating, Advanced Commands and BIT	Customers who wish to communicate with the 440 Series system for sensor and navigation data, should review Section 6 and 7. Section 8 is for users who wish to configure the 440 Series operating parameters (e.g., baud rate or power-up output rate) without NAV-VIEW 2.X.

1.2 Overview of the 440 Series Inertial Systems

This manual provides a comprehensive introduction to the use of Crossbow's 440 Series Inertial System products listed in Table 2. This manual is intended to be used as a detailed technical reference and operating guide for the 440 Series. Crossbow's 440 Series products combine the latest in high-performance commercial MEMS (Micro-electromechanical Systems) sensors and digital signal processing techniques to provide a small, cost-effective alternative to existing IMU systems and mechanical gyros.

Table 2: 440 Series Feature Description

Product	Features
IMU440	6- DOF Digital IMU
VG440	6-DOF IMU, plus Dynamic Roll/Pitch
AHRS440	6-DOF IMU with 3-Axis Internal Magnetometer Dynamic Roll, Pitch, and Heading
NAV440	6-DOF IMU with 3-Axis Internal Magnetometer, and Internal WAAS Capable GPS Receiver Position, Dynamic Velocity, and Dynamic Roll, Pitch, Heading

The 440 Series is Crossbow's third generation of MEMS-based Inertial Systems, building on over a decade of field experience, and encompassing thousands of deployed units and millions of operational hours in a wide range of land, marine, airborne, and instrumentation applications.

At the core of the 440 Series is a rugged 6-DOF (Degrees of Freedom) MEMS inertial sensor cluster that is common across all members of the 440 Series. The 6-DOF MEMS inertial sensor cluster includes three axes of MEMS angular rate sensing and three axes of MEMS linear acceleration sensing. These sensors are based on rugged, field proven silicon bulk micromachining technology. Each sensor within the cluster is individually factory calibrated for temperature and non-linearity effects during Crossbow's manufacturing and test process using automated thermal chambers and rate tables.

Coupled to the 6-DOF MEMS inertial sensor cluster is a high performance DSP processor that utilizes the inertial sensor measurements to accurately compute navigation information including attitude, heading, and linear velocity thru dynamic maneuvers (actual measurements are a function of the 440 Series product as shown in Table 2). In addition, the DSP processor makes use of internal and external magnetic sensor and/or GPS data to aid the performance of the inertial algorithms and help correct long term drift and estimate errors from the inertial sensors and computations. The navigation algorithm utilizes a multi-state configurable Extended Kalman Filter (EKF) to correct for drift errors and estimate sensor bias values. This algorithm runs on a 150MHz 32-bit DSP (Digital Signal Processor) that has approximately four times the computational power of Crossbow's earlier generation Inertial Systems.

Another unique feature of the 440 Series is the extensive field configurability of the units. This field configurability allows the 440 Series of Inertial Systems to satisfy a wide range of applications and performance requirements with a single mass produced hardware platform. The basic configurability includes parameters such as baud rate, packet type, and update rate, and the advanced configurability includes the defining of custom axes and how the sensor feedback is utilized in the Kalman filter during the navigation process.

The 440 Series is packaged in a fully sealed lightweight housing which provides EMI, vibration, and moisture resistance to levels consistent with most land, marine, and airborne environments. The 440 Series utilizes an RS-232 serial link for data communication, and each data transmission includes a BIT (Built-In-Test) message providing system health status. The 440 Series is supported by Crossbow's NAV-VIEW 2.X, a powerful PC-based operating tool that provides complete field configuration, diagnostics, charting of sensor performance, and data logging with playback.

1.3 Summary of Major Changes from the 300/400 Series and the 420 Series

1.3.1 Mechanical Size and Footprint

The mechanical footprint of Crossbow's new 440 Series Inertial Systems is compatible with prior generation Inertial Systems including Crossbow's 400 Series (IMU400, VG400, AHRS400) and the NAV420 Series products.   The mounting plate foot print is the same and the connector location is identical.   The 440 Series products are shorter than their equivalent 400 Series product (i.e. the AHRS440 is shorter than the AHRS400, etc).   The NAV440 is dimensionally equivalent to the NAV420. For detailed mechanical and installation drawings, see appendix A of the manual.

1.3.2 Connector Pin Out & Operating Voltage, Current

The DB-15 male connector on Crossbow's 440 Series Inertial Systems has a backward compatible pin out with the 400 Series and 420 Series.   However, the 440 Series has a secondary optional-use communications port for internal or external GPS.

1.3.3 Software Compatibility

Crossbow's 440 Series Inertial Systems are NOT software compatible with any previous Crossbow products. The 440 Series products utilize a new extensible communication protocol which is documented in section 7 of this manual. Additionally, the 440 Series includes numerous enhancements and features that allow for better performance in many applications than the comparable 400 or 420 Series products.

1.3.4 Operating Performance and Accuracy

The 440 Series has been characterized in a wide range of land and airborne applications.   In the qualification testing, the dynamic accuracy of the 440 Series has shown superior performance when compared to the equivalent model of 400 and 420 Series, reducing attitude estimation errors in half during certain critical dynamic maneuvers without the use of GPS aiding. With GPS aiding in the NAV440, attitude estimation is improved by an order of magnitude compared with 400 series products. Recommended product configuration settings are discussed in Section 5, and theory of operation is discussed in Section 4.

2 Connections

2.1 Connections

The 440 Series has a male DB-15 connector. The signals are as shown in Table 3.

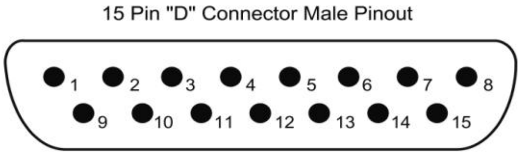


Table 3 Connector Pin Assignments

Pin	Signal
1	RS-232 Transmit Data (A Port)
2	RS-232 Receive Data (A Port)
3	Positive Power Input (+Vcc)
4	Power Ground
5	Chassis Ground
6	NC – Factory use only
7	RS-232 GPS Tx (B Port)
8	RS-232 GPS Rx (B Port)
9	Signal Ground
10	1 PPS OUT
11	1 PPS IN (IMU/VG/AHRS only)
12	NC – factory use only
13	Hardware BIT Error
14	NC – factory use only
15	NC – factory use only

2.2 I/O Cable

The user must provide a shielded cable with the shield connected to the I/O connector shell in order to provide the required EMI protection. The cable sent with the unit is intended to provide the user with the ability to test the unit right out of the box, and will not provide adequate shielding for all environments. Case ground (see below) must be used to provide full EMI protection. In addition, users should take care to ground the cable shield on only one end of the cable.

⚠ WARNING

The 440 Series is shipped with an EMI filter attached to the DB-15 connector. This connector must remain in place to ensure proper shielding from EMI interference.

⚠ WARNING

The cable sent with the unit is intended to provide the user with the ability to test the unit right out of the box, and will not provide adequate shielding for all environments.

⚠ WARNING

Case ground (see below) must be used to provide full EMI protection. In addition, users should take care to ground the cable shield on only one end of the cable.

2.3 Power Input and Power Input Ground

Power is applied to the 440 Series on pins 3 and 4. Pin 4 is ground; Pin 3 should have 9 to 42 VDC unregulated at 350 mA. If you are using the cable supplied with the 440 Series, the power supply wires are broken out of the cable at the DB-9 connector. The red wire is connected to the positive power input; the black wire is connected to the power supply ground.

⚠ WARNING

Do not reverse the power leads or damage may occur.

2.4 Case Ground

The case is electrically connected to Pin 5 of the DB-15 connector. The Pin 5 should be electrically connected to the user's cable shield, especially if the chassis does not make good ground contact. The case should be bolted to a good conducting surface that is grounded.

2.5 Serial Data Interface (A Port)

The serial interface is standard RS-232, 9600, 19200, 38400, or 57600 baud, 8 data bits, 1 start bit, 1 stop bit, no parity, and no flow control and will output at a user configurable output rate. These settings allow interaction via a standard PC serial port. The serial data settings can be configured on a 440 Series unit with NAV-VIEW 2.X. In order to set the serial data interface, select Unit Configuration, under the Menu Tab.

2.6 Serial GPS Interface (NAV440) (B Port Output)

The internal GPS receiver in NAV440 products outputs data in NMEA-0183 format as defined by the National Marine Electronics Association (NMEA), Standard For Interfacing Marine Electronic Devices, Version 2.20, January 1, 1997.

The packets are sent at 9600 Baud, 8 data bits, 1 start bit, 1 stop bit, no parity bit, and no flow control.

The GPS receiver outputs the following messages as shown in Table 4. (Refer to Appendix B for the detailed message format)

Table 4 GPS Output Packet Format

NMEA Record	**Description**
GGA	Global positioning system fixed data
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed data

2.7 External GPS Aiding (VG440 and AHRS440) (B Port Input)

The VG440/AHRS440 allows the use of an external GPS receiver to be connected with its GPS port. The user is required to configure the GPS receiver to output the GPS messages that the 440 Series expects. The table below shows the supported GPS protocols and guidelines for configuration. Note that the details of the GPS messages can be found in the respective GPS protocol documents. The user must configure the VG/AHRS440 to accept external GPS information using NAV-VIEW 2.X as described in Chapter 3. If the VG/AHRS440 is parsing valid external GPS data and the GPS receiver has 3D lock, then the comStatus→noExternalGPS flag will be zero, otherwise it will be one. See section 9 for a complete description of system status indications.

Since NMEA protocol does not provide vertical velocity, the vertical velocity that the 440 Series estimates (based upon GPS altitude changes) may not be sufficient for airborne applications. Therefore, the NMEA protocol is not recommended for airborne applications.

Configuration of An External GPS Receiver for VG440/AHRS440

Protocols	Required Messages	Required Message Rate	Baud rate
Ublox binary	NAV-LLH, NAV-VELNED, NAV-STATUS	4 Hz	9600, 19200, 38400, 57600$
NovAtel OEM4 and OEMV Binary	BestPosB, BestVelB	4Hz	9600, 19200, 38400, 57600$
NovAtel OEM4 ASCII	PosVelNavDopA	4Hz	19200, 38400, 57600$
NMEA*	GPGGA, GPVTG	4Hz	9600, 19200, 38400, 57600$

*Not recommended for airborne applications.

$57600 is the preferred baud rate for optimum performance

The GPS serial communication port should be configured to 8 data bits, 1 start bit, 1 stop bit, no parity bit, and no flow control.

2.8 Hardware BIT Error Output

The hardware BIT error output pin is the ultimate indication of system failure. This indication is available in most software output packets as the masterFail flag. It is the logical AND of the hardwareError, comError, and softwareError flags monitored by the system. In the event of a communication failure, the hardware BIT error pin may be used to detect a masterFail assertion. This pin is open-collector and requires a 1k to 10k ohm pull-up resister. The system will drive this pin low to assert a system failure.

2.9 1 PPS Input Interface (IMU, VG, and AHRS440)

The 1PPS input signal allows the user of IMU, VG, and AHRS 440 products to force synchronization of sensor data collection to a 1Hz rising-edge signal. The signal must maintain 0.0-0.2 V zero logic and 3.0-5.0 volts high logic and stay within 100ms of the internal system 1 second timing. Sending this signal to the system will align the sensor data collection and algorithm processing to its rising edge and 10ms boundaries thereafter. When the system is synchronized to 1PPS, the hardwareStatus→unlocked1PPS flag will be zero, otherwise, it will be one.

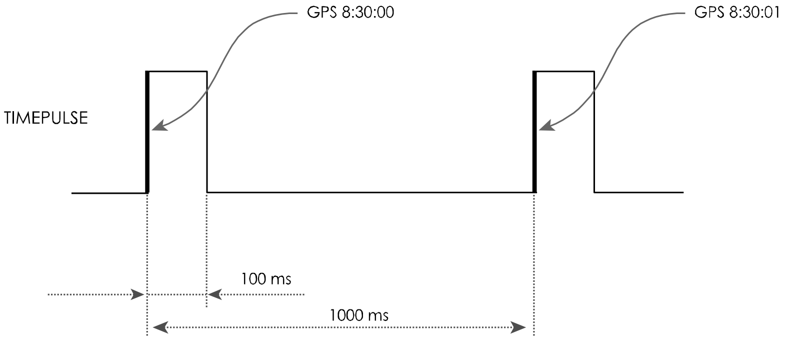
2.10 1 PPS Output Interface (NAV440)

The 1PPS output signal is provided by the internal GPS receiver (when GPS timing is known) on NAV440 products. On IMU, VG, and AHRS 440 products, this signal is simply a buffered version of the 1PPS input signal provided by the user. The 1PPS output signal is open-collector and should be interfaced to a rising-edge trigger with pull up resistor between 1k and 10k ohms. The 440 Series products synchronize sensor data collection to this 1PPS signal internally when available. Therefore, the 100Hz navigation algorithm will run exactly 100 times each second with no slip when locked to 1PPS. Packet data is valid on the rising edge of 1PPS and 10ms boundaries thereafter. There is, however, up to 500µs of additional latency in sensor data collection. If 1PPS is provided by the internal GPS receiver in NAV products, then the rising edge of 1PPS will correspond to the UTC second boundary. When the system is synchronized to 1PPS, the hardwareStatus→unlocked1PPS flag will be zero, otherwise, it will be one.

Figure 1 shows the sequential order of the signal present at 1 PPS OUT pin.

The 1 PPS signal is aligned to the sampling clock of 23.104 MHz. This results in a timing resolution of 43 ns.

Figure 1 1PPS Output Signal



2.11 GPS Antenna Connection (NAV440)

The GPS receiver needs to receive signals from as many satellites as possible. A GPS receiver doesn't work properly in narrow streets and underground parking lots or if objects or human beings cover the antenna. Poor visibility may result in position drift or a prolonged Time-To-First-Fix (TTFF). A good sky visibility is therefore a prerequisite. Even the best receiver can't make up for signal loss due to a poor antenna, in-band jamming or a poor RF cable.

The 440 Series unit ships with an external active antenna that must be connected properly to SMA jack located next to the DB-15 connector. Placing the antenna on a 4 inch or larger ground plane is highly recommended.

IMPORTANT

Place the antenna with optimal sky visibility and use a ground plane. Route the GPS Antenna RF cable away from sources of radiated energy (i.e. switching power supplies).

2.12 No Connection

During normal operation of the 440 Series, no connection is made to the factory test pins. These pins have internal pull-up mechanisms and must have no connections for the 440 Series to operate properly.

2.13 Quick Digital interface connection

On a standard DB-9 COM port connector, make the connections as described in Table 3.

Table 5 DB-9 COM Port Connections

COM Port Connector			**440 Series Connector**	
Pin #	Signal		Pin #	Signal
2	RxD		1	TxD
3	TxD		2	RxD
5	GND		9	GND

3 Installation and Operation of NAV-VIEW 2.X

NAV-VIEW 2.X has been completely redesigned to allow users to control all aspects of the 440 Series operation including data recording, configuration and data transfer. For the first time, you will be able to control the orientation of the unit, sampling rate, packet type, hard iron calibration and filter settings through NAV-VIEW.

3.1 NAV-VIEW 2.X Computer Requirements

The following are minimum requirements for the installation of the NAV-VIEW 2.X Software:

- CPU: Pentium-class (1.5GHz minimum)
- RAM Memory: 500MB minimum, 1GB+ recommended
- Hard Drive Free Memory: 20MB
- Operating System: Windows 2000™, or XP™,
- Properly installed Microsoft .NET 2.0 or higher

3.1.1 Install NAV-VIEW 2.X

To install NAV-VIEW 2.X onto your computer:

1. Insert the CD "440 Series Inertial System" in the CD-ROM drive.

2. Locate the "NAV-VIEW 2.X" folder. Double click on the "setup.exe" file.

3. Follow the setup wizard instructions. You will install NAV-VIEW 2.X and .NET 2.0 framework.

3.2 Connections

The 440 Series Inertial Systems products are shipped with a cable to connect the 440 Series to a PC Serial port.

1. Connect the 15-pin end of the digital signal cable labeled "DMU440" to the port on the 440 Series product.

2. Connect the 9-pin end of the cable marked "DMU440 to User" to the serial port of your computer.

3. The additional black and red wires on the cable connect power to the 440 Series product. Match red to (+) power and black to (-) ground. The input voltage can range from 9-42 VDC with a maximum current draw of 350 mA.

4. Allow at least 60 seconds after power up for the 440 Series product to initialize. The 440 Series needs to be held motionless during this period.

⚠ **WARNING**

Do not reverse the power leads! Reversing the power leads to the 440 Series can damage the unit; although there is reverse power protection, Crossbow Technology is not responsible for resulting damage to the unit should the reverse voltage protection electronics fail.

3.3 Setting up NAV-VIEW 2.X

With the 440 Series product powered up and connected to your PC serial port, open the NAV-VIEW 2.X software application.

1. NAV-VIEW 2.X should automatically detect the 440 Series product and display the serial number and firmware version if it is connected.

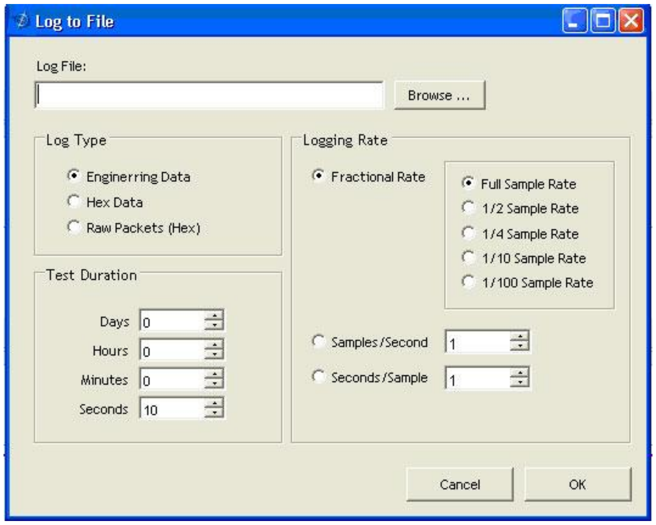
2. If NAV-VIEW 2.X does not connect, check that you have the correct COM port selected. You will find this under the “Setup” menu. Select the appropriate COM port and allow the unit to automatically match the baud rate by leaving the “Auto: match baud rate” selection marked.
3. If the status indicator at the bottom is green and states, Unit Connected, you’re ready to go. If the status indicator doesn’t say connected and is red, check the connections between the 440 Series product and the computer, check the power supply, and verify that the COM port is not occupied by another device.
4. Under the “View” menu you have several choices of data presentation. Graph display is the default setting and will provide a real time graph of all the 440 Series data. The remaining choices will be discussed in the following pages.

3.4 Data Recording

NAV-VIEW 2.X allows the user to log data to a text file (.txt) using the simple interface at the top of the screen. Customers can now tailor the type of data, rate of logging and can even establish predetermined recording lengths.

To begin logging data follow the steps below:

1. Locate the 📂 icon at the top of the page or select “Log to File” from the “File” drop down menu.
2. The following menu will appear.



Log File:

Log Type
● Enginerring Data
○ Hex Data
○ Raw Packets (Hex)

Test Duration
Days 0
Hours 0
Minutes 0
Seconds 10

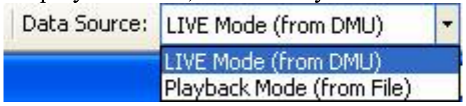
Logging Rate
● Fractional Rate
● Full Sample Rate
○ 1/2 Sample Rate
○ 1/4 Sample Rate
○ 1/10 Sample Rate
○ 1/100 Sample Rate
○ Samples/Second 1
○ Seconds/Sample 1

3. Select the “Browse” box to enter the file name and location that you wish to save your data to.
4. Select the type of data you wish to record. “Engineering Data” records the converted values provided from the system in engineering units, “Hex Data” provides the raw hex values separated into columns displaying the value, and the “Raw Packets” will simply record the raw hex strings as they are sent from the unit.
5. Users can also select a predetermined “Test Duration” from the menu. Using the arrows, simply select the duration of your data recording.
6. Logging Rate can also be adjusted using the features on the right side of the menu.
7. Once you have completed the customization of your data recording, you will be returned to the main screen where you can start the recording process using the ● button at the top of the page or select “Start Logging” from the “File” menu. Stopping the data recording can be accomplished using the ■ button and the recording can also be paused using the ⏸ button.

3.5 Data Playback

In addition to data recording, NAV-VIEW 2.X allows the user to replay saved data that has been stored in a log file.

1. To playback data, select “Playback Mode” from the “Data Source” drop down menu at the top.


Data Source: LIVE Mode (from DMU)

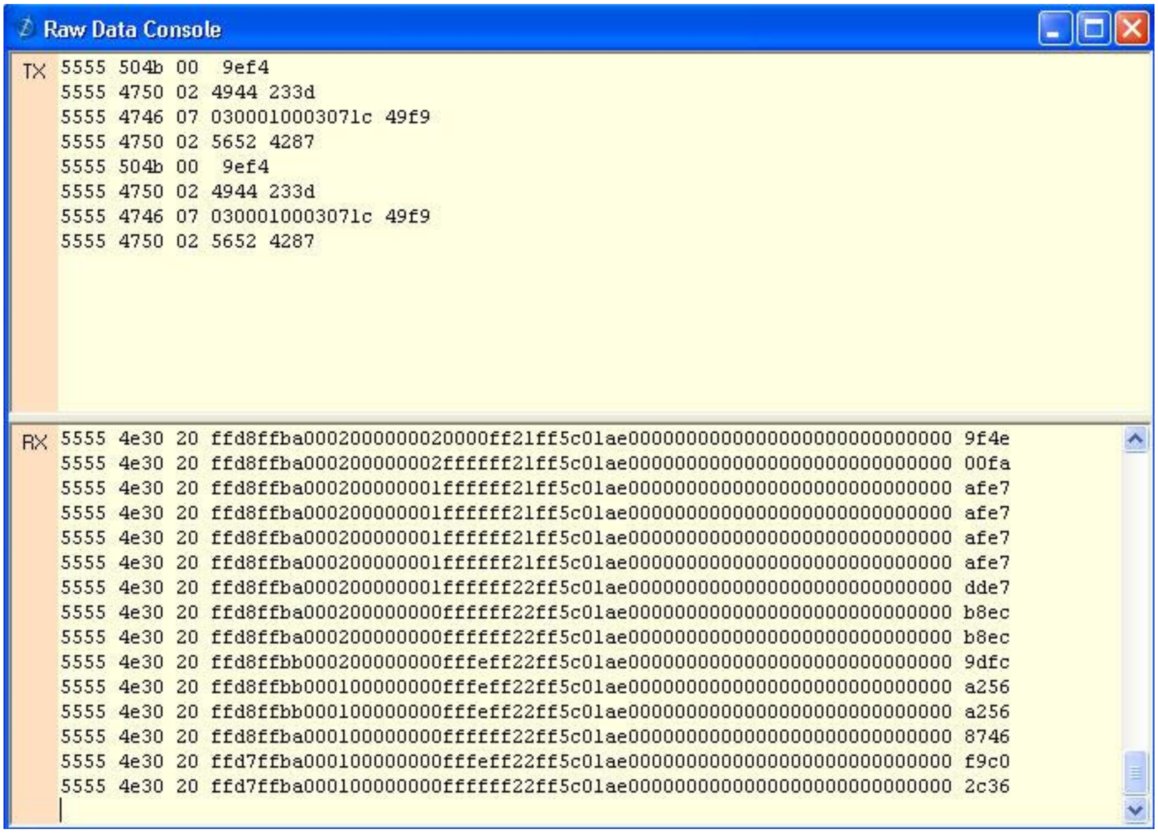
2. Selecting Playback mode will open a text prompt which will allow users to specify the location of the file they wish to play back. All three file formats are supported (Engineering, Hex, and Raw) for playback. In addition, each time recording is stopped/started a new section is created. These sections can be individually played back by using the drop down menu and associated VCR controls.

3. Once the file is selected, users can utilize the VCR style controls at the top of the page to start, stop, and pause the playback of the data.
4. NAV-VIEW 2.X also provides users with the ability to alter the start time for data playback. Using the slidebar at the top of the page users can adjust the starting time.

3.6 Raw Data Console

NAV-VIEW 2.X offers some unique debugging tools that may assist programmers in the development process. One such tool is the Raw Data Console. From the "View" drop down menu, simply select the "Raw Data Console". This console provides users with a simple display of the packets that have been transmitted to the unit (Tx) and the messages received (Rx). An example is provided below.

Figure 2 Raw Data Console

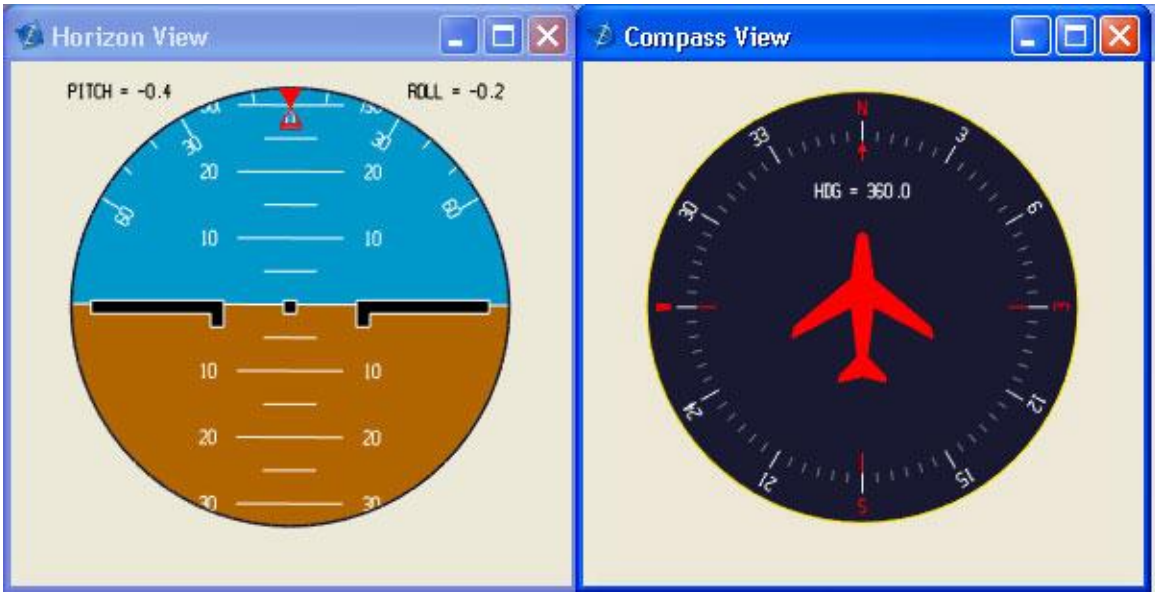


```
TX 5555 504b 00  9ef4
   5555 4750 02 4944 233d
   5555 4746 07 030001000307lc 49f9
   5555 4750 02 5652 4287
   5555 504b 00  9ef4
   5555 4750 02 4944 233d
   5555 4746 07 030001000307lc 49f9
   5555 4750 02 5652 4287

RX 5555 4e30 20 ffd8ffba0002000000020000ff21ff5c01ae0000000000000000000000000000 9f4e
   5555 4e30 20 ffd8ffba000200000002ffffff21ff5c01ae0000000000000000000000000000 00fa
   5555 4e30 20 ffd8ffba000200000001ffffff21ff5c01ae0000000000000000000000000000 afe7
   5555 4e30 20 ffd8ffba000200000001ffffff21ff5c01ae0000000000000000000000000000 afe7
   5555 4e30 20 ffd8ffba000200000001ffffff21ff5c01ae0000000000000000000000000000 afe7
   5555 4e30 20 ffd8ffba000200000001ffffff21ff5c01ae0000000000000000000000000000 afe7
   5555 4e30 20 ffd8ffba000200000001ffffff22ff5c01ae0000000000000000000000000000 dde7
   5555 4e30 20 ffd8ffba000200000000ffffff22ff5c01ae0000000000000000000000000000 b8ec
   5555 4e30 20 ffd8ffba000200000000ffffff22ff5c01ae0000000000000000000000000000 b8ec
   5555 4e30 20 ffd8ffbb000200000000fffeff22ff5c01ae0000000000000000000000000000 9dfc
   5555 4e30 20 ffd8ffbb000100000000fffeff22ff5c01ae0000000000000000000000000000 a256
   5555 4e30 20 ffd8ffbb000100000000fffeff22ff5c01ae0000000000000000000000000000 a256
   5555 4e30 20 ffd8ffba000100000000ffffff22ff5c01ae0000000000000000000000000000 8746
   5555 4e30 20 ffd7ffba000100000000fffeff22ff5c01ae0000000000000000000000000000 f9c0
   5555 4e30 20 ffd7ffba000100000000ffffff22ff5c01ae0000000000000000000000000000 2c36
```

3.7 Horizon and Compass View

If the 440 Series product you have connected is capable of providing heading and angle information (see Table 2), NAV-VIEW 2.X can provide a compass and a simulated artificial horizon view. To activate these views, simply select "Horizon View" and/or "Compass View" from the "View" drop down menu at the top of the page.

Figure 3 Horizon and Compass View



PITCH = -0.4

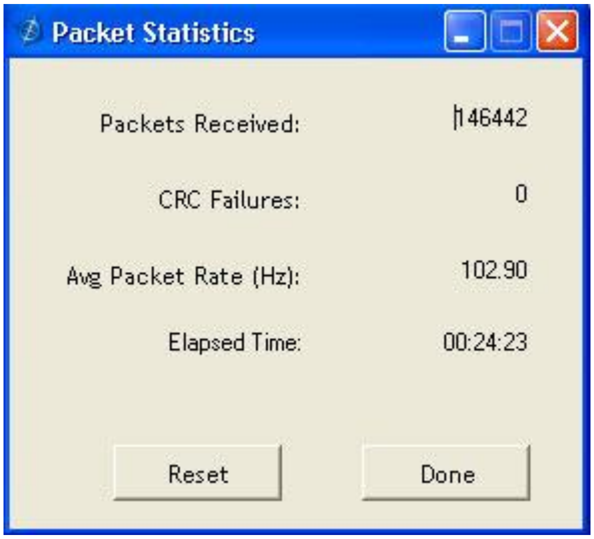
ROLL = -0.2

HDG = 360.0

3.8 Packet Statistics View

Packet statistics can be obtained from the "View" menu by selecting the "Packet Statistics" option. This view simply provides the user with a short list of vital statistics (including Packet Rate, CRC Failures, and overall Elapsed Time) that are calculated over a one second window. This tool should be used to gather information regarding the overall health of the user configuration. Incorrectly configured communication settings can result in a large number of CRC Failures and poor data transfer.

Figure 4 Packet Statistics



Packets Received:	146442
CRC Failures:	0
Avg Packet Rate (Hz):	102.90
Elapsed Time:	00:24:23

3.9 Unit Configuration

The Unit Configuration window gives the user the ability to view and alter the system settings. This window is accessed through the “Unit Configuration” menu item under the configuration menu. Under the “General” tab, users have the ability to verify the current configuration by selecting the “Get All Values” button. This button simply provides users with the currently set configuration of the unit and displays the values in the left column of boxes.

There are three tabs within the “Unit Configuration” menu; General, Advanced and BIT Configuration. The General tab displays some of the most commonly used settings. The Advanced and BIT Configuration menus provide users with more detailed setting information that they can tailor to meet their specific needs.

To alter a setting, simply select the check box on the left of the value that you wish to modify and then select the value using the drop down menu on the right side. Once you have selected the appropriate value, these settings can be set temporarily or permanently (a software reset or power cycle is required for the changes to take affect) by selecting from the choices at the bottom of the dialog box. Once the settings have been altered a “Success” box will appear at the bottom of the page.

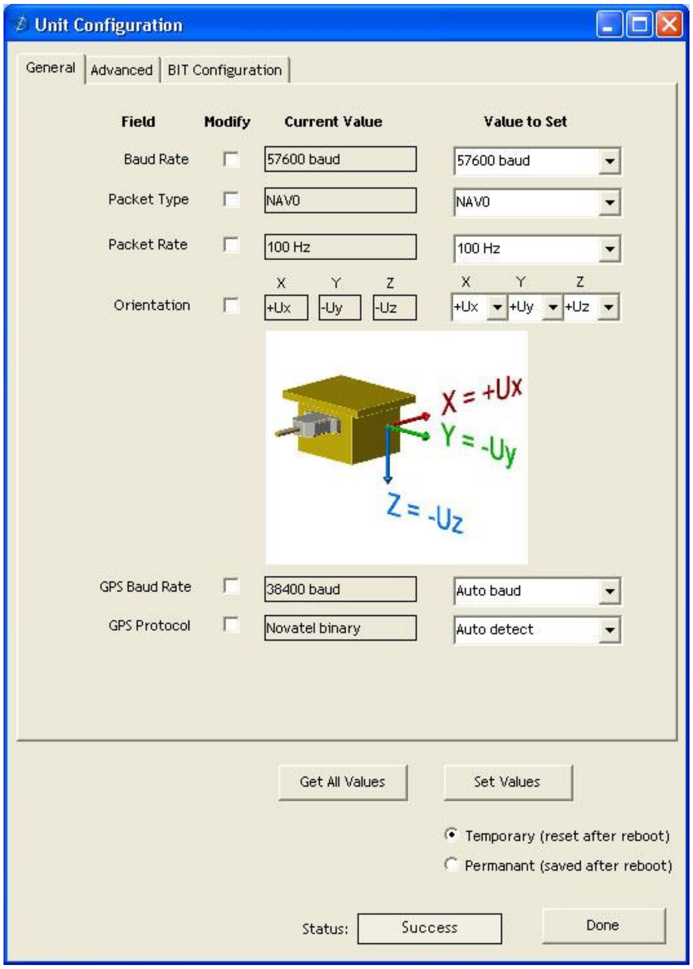
IMPORTANT

Caution must be taken to ensure that the settings selected are compatible with the system that is being configured. In most cases a “FAIL” message will appear if incompatible selections are made by the user, however it is the users responsibility to ensure proper configuration of the unit.

IMPORTANT

Unit orientation selections must conform to the right hand coordinate system as noted in Section 4.1 of this user manual. Selecting orientations that do not conform to this criteria are not allowed.

Figure 5 Unit Configuration



Unit Configuration

General | Advanced | BIT Configuration

Field	Modify	Current Value	Value to Set
Baud Rate	□	57600 baud	57600 baud
Packet Type	□	NAV0	NAV0
Packet Rate	□	100 Hz	100 Hz
Orientation	□	X +Ux, Y -Uy, Z -Uz	X +Ux, Y +Uy, Z +Uz
GPS Baud Rate	□	38400 baud	Auto baud
GPS Protocol	□	Novatel binary	Auto detect

● Temporary (reset after reboot)

○ Permanant (saved after reboot)

Status: Success

3.10 Advanced Configuration

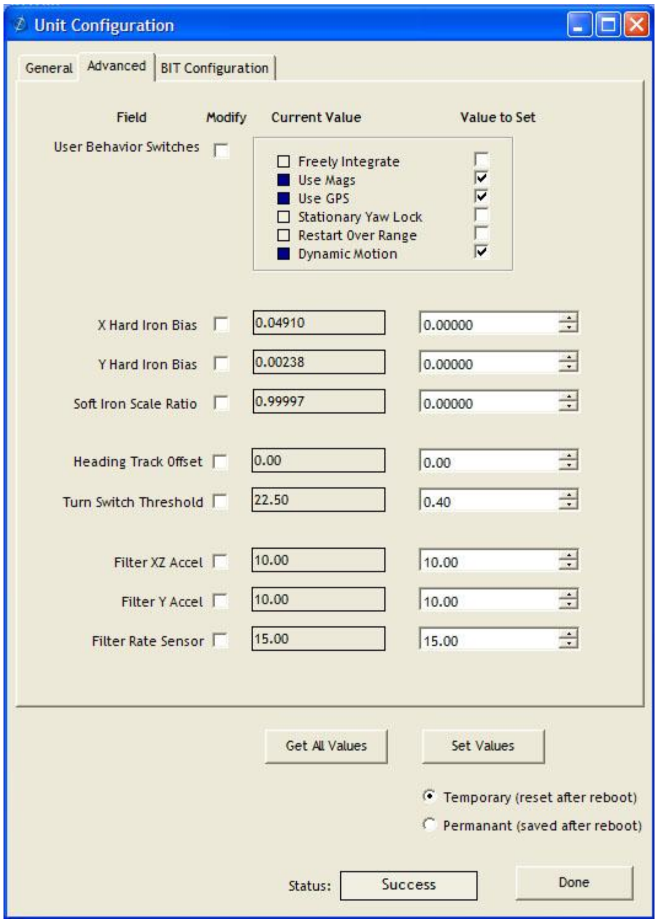
Users who wish to access some of the more advanced features of NAV-VIEW 2.X and the 440 Series products can do so by selecting the “Advanced” tab at the top of the “Unit Configuration” window.

⚠ WARNING

Users are strongly encouraged to read and thoroughly understand the consequences of altering the settings in the “Advanced” tab before making changes to the unit configuration. These settings are discussed in detail in Chapter 4 below.

Behavior switches are identified at the top of the page with marked boxes. A blue box will appear if a switch has been enabled similar to the figure below. The values can be set in the same manner as noted in the previous section. To set a value, users select the appropriate “Modify” checkbox on the left side of the menu and select or enable the appropriate value they wish to set. At the bottom of the page, users have the option of temporarily or permanently setting values. When all selections have been finalized, simply press the “Set Values” button to change the selected settings.

Figure 6 Advanced Settings



Field	Modify	Current Value	Value to Set
User Behavior Switches	□	□ Freely Integrate	□
		■ Use Mags	☑
		■ Use GPS	☑
		□ Stationary Yaw Lock	□
		□ Restart Over Range	□
		■ Dynamic Motion	☑
X Hard Iron Bias	□	0.04910	0.00000
Y Hard Iron Bias	□	0.00238	0.00000
Soft Iron Scale Ratio	□	0.99997	0.00000
Heading Track Offset	□	0.00	0.00
Turn Switch Threshold	□	22.50	0.40
Filter XZ Accel	□	10.00	10.00
Filter Y Accel	□	10.00	10.00
Filter Rate Sensor	□	15.00	15.00

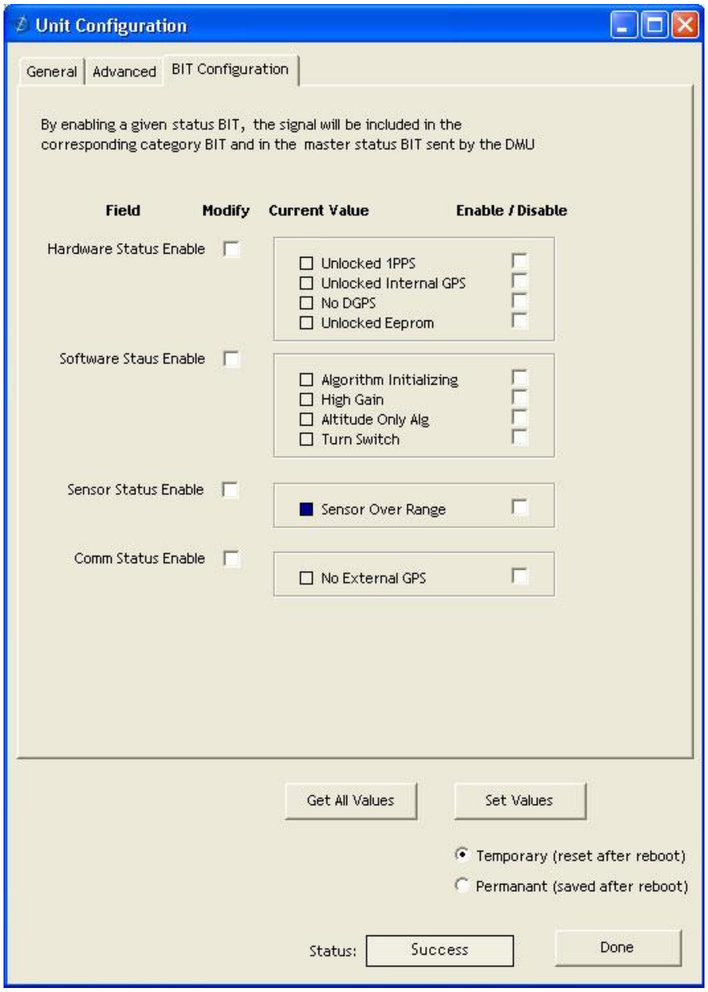
● Temporary (reset after reboot)
○ Permanant (saved after reboot)

Status: Success

3.11 Bit Configuration

The third and final tab of the unit configuration window is “Bit Configuration”. This tab allows the users to alter the logic of individual status flags that affect the masterStatus flag in the master BITstatus field (available in most output packets). By enabling individual status flags users can determine which flags are logically OR’ed to generate the masterStatus flag. This gives the user the flexibility to listen to certain indications that affect their specific application. The masterFail and all error flags are not configurable. These flags represent serious errors and should never be ignored.

Figure 7 BIT Configuration



By enabling a given status BIT, the signal will be included in the corresponding category BIT and in the master status BIT sent by the DMU

Field	Modify	Current Value	Enable / Disable
Hardware Status Enable	□	□ Unlocked 1PPS	□
		□ Unlocked Internal GPS	□
		□ No DGPS	□
		□ Unlocked Eeprom	□
Software Staus Enable	□	□ Algorithm Initializing	□
		□ High Gain	□
		□ Altitude Only Alg	□
		□ Turn Switch	□
Sensor Status Enable	□	■ Sensor Over Range	□
Comm Status Enable	□	□ No External GPS	□

● Temporary (reset after reboot)

○ Permanant (saved after reboot)

Status: Success

3.12 Mag Alignment Procedure

IMPORTANT

The following section only applies to 440 Series products with magnetometers (AHRS and NAV440). If your particular model does not utilize magnetometers for heading or performance you can disregard Section 3.12.

3.12.1 Hard Iron/Soft Iron Overview

The AHRS and NAV440 products use magnetic sensors to compute heading. Ideally, the magnetic sensors would be measuring only earth's magnetic field to compute the heading angle. In the real world, however, residual magnetism in your system adds to the total magnetic field measured. This residual magnetism (called hard iron and soft iron) will create errors in the heading measurement if it is not accounted for. In addition, magnetic material can change the direction of the magnetic field as a function of the input magnetic field. This dependence of the local magnetic field on input direction is called the soft iron effect.

The AHRS and NAV440 products can actually measure the constant magnetic field that is associated with your system and correct for it. The AHRS and NAV440 products can also make a correction for some soft iron effects. The process of measuring these non-ideal effects and correcting for them is called the "Mag Alignment Procedure". Performing a "Mag Alignment Procedure" will help correct for magnetic fields that are fixed with respect to the 440 Series product. It cannot correct for time varying fields, or fields created by ferrous material that moves with respect to the 440 Series product.

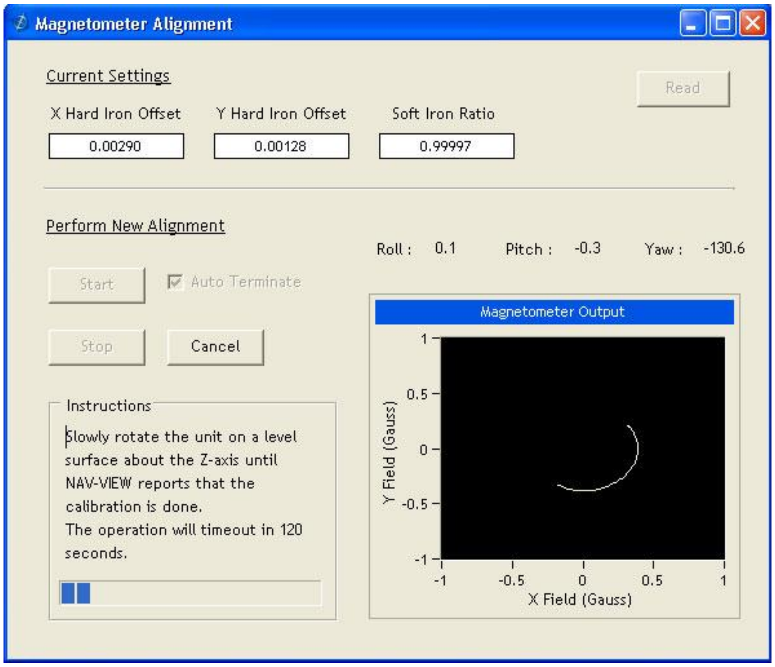
The AHRS and NAV440 products account for the extra magnetic field by making a series of measurements, and using these measurements to model the hard iron and soft iron environment in your system using a two-dimensional algorithm. The AHRS and NAV440 products will calculate the hard iron magnetic fields and soft iron corrections and store these as calibration constants in the EEPROM.

The "Mag Alignment Procedure" should always be performed with the AHRS or NAV440 product installed in the user system. If you perform the calibration process with the 440 Series product by itself, you will not be correcting for the magnetism in the user system. If you then install the 440 Series product in the system (i.e. a vehicle), and the vehicle is magnetic, you will still see errors arising from the magnetism of the vehicle.

3.12.2 Mag Alignment Procedure Using NAV-VIEW 2.X

The Mag Alignment Procedure using NAV-VIEW 2.X can be performed using the following steps below:

1. Select "Mag Alignment" from the "Configuration" drop down menu at the top.
2. If you can complete your 360 degree turn within 120 seconds, select the "Auto-Terminate" box.
3. Select the "Start" button to begin the "MagAlign" Procedure and follow the instructions at the bottom of the screen as shown in the figure below.



X Hard Iron Offset: 0.00290

Y Hard Iron Offset: 0.00128

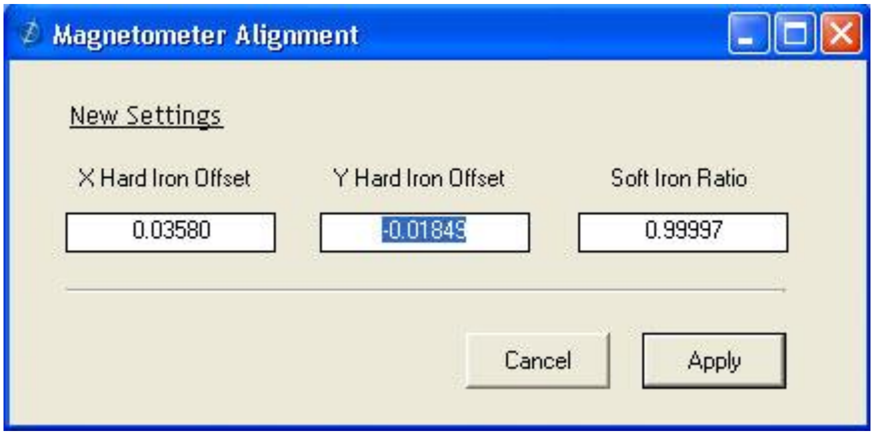
Soft Iron Ratio: 0.99997

Roll : 0.1 Pitch : -0.3 Yaw : -130.6

Instructions

Slowly rotate the unit on a level surface about the Z-axis until NAV-VIEW reports that the calibration is done.
The operation will timeout in 120 seconds.

4. Rotate the AHRS or NAV440 product through 380 degrees of rotation or until you receive a message to stop.
5. Once you have completed your rotation, you will be given data concerning the calibration accuracy. The X and Y offset values indicate how far the magnetic field has been offset due to hard iron affects from components surrounding the unit. In addition, you will see a soft iron ratio indicating the effect of soft iron on the AHRS of NAV440 product.
6. Save this data to the AHRS or NAV440 product by selecting the "Apply" button.



X Hard Iron Offset: 0.03580

Y Hard Iron Offset: -0.01849

Soft Iron Ratio: 0.99997

7. Upon completion of the "Mag Alignment Procedure", the heading accuracy should be verified with all third party systems active using a known reference such as a compass rose, GPS track or a calibrated compass. Heading inaccuracies greater than the values specified on the data sheet or fluctuating heading performance may be an indication of magnetic field disturbances near the unit.

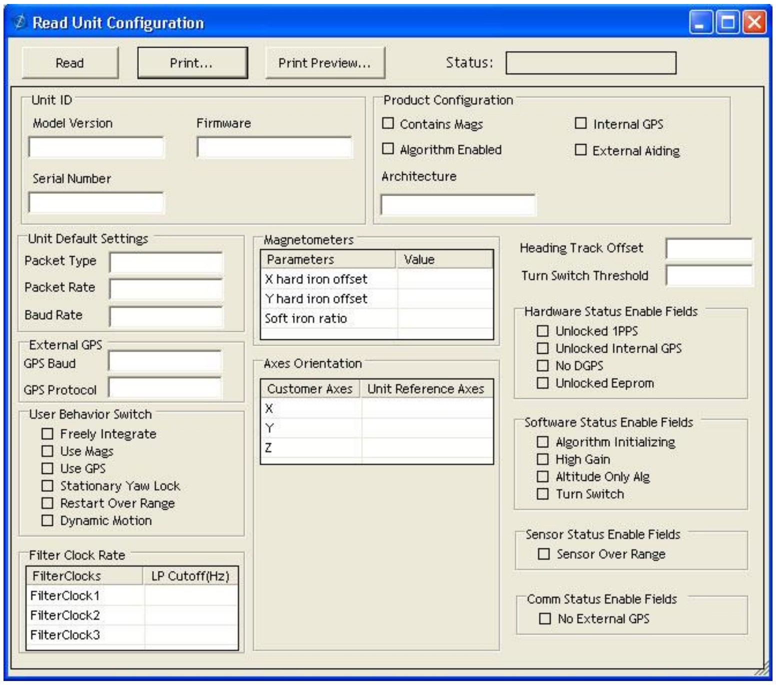
IMPORTANT

An acceptable calibration will provide X and Y Hard Iron Offset Values of <0.1 and a Soft Iron Ratio >0.95. If this procedure generates any values larger than stated above, the system will assert the softwareError→dataError→magAlignOutOfBounds error flag. See section 9 for details on error flag handling.

3.13 Read Unit Configuration

NAV-VIEW 2.X allows users to view the current settings and calibration data for a given 440 series unit by accessing the "Read Configuration" selection from the "Configuration" drop down menu. From this dialog, users can print a copy of the unit's current configuration and calibration values with the click of a button. Simply select the "Read" button at the top of the dialog box and upon completion select the "Print" or "Print Preview" buttons to print a copy to your local network printer. This information can be helpful when storing hard copies of unit configuration, replicating the original data sheet and for troubleshooting if you need to contact Crossbow's Support Staff.

Figure 8 Read Configuration



4 Theory of Operation

This section of the manual covers detailed theory of operation for each member of the 440 Series starting with the basic IMU440 and then reviewing each major variant (VG, AHRS and NAV) with their associated additional features, outputs, and performance. Table 6 shows the basic features of each member of the 440 Series with cross references to important sections for review.

Table 6 440 Series Overview

Product	Features	Learning More
IMU440	6- DOF Digital IMU	Read 4.1 and 4.2
VG440	6-DOF IMU, plus Dynamic Roll/Pitch (optional external GPS)	Read 4.1, 4.2 and 4.3
AHRS440	6-DOF IMU with 3-Axis Internal Magnetometer Dynamic Roll, Pitch, and Heading (Optional External GPS)	Read 4.1, 4.2, 4.3 and 4.4
NAV440	6-DOF IMU with 3-Axis Internal Magnetometer, and Internal WAAS Capable GPS Receiver Position, Dynamic Velocity, and Dynamic Roll, Pitch, Heading	Read 4.1, 4.2, 4.3, 4.4 and 4.5

Figure 9 shows the 440 Series hardware block diagram. At the core of the 440 Series is a rugged 6-DOF (Degrees of Freedom) MEMS inertial sensor cluster that is common across all members of the 440 Series. The 6-DOF MEMS inertial sensor cluster includes three axes of MEMS angular rate sensing and three axes of MEMS linear acceleration sensing. These sensors are based on rugged, field proven silicon bulk micromachining technology. Each sensor within the cluster is individually factory calibrated using Crossbow's automated manufacturing process. Sensor errors are compensated for temperature bias, scale factor, non-linearity and misalignment effects using a proprietary algorithm from data collected during manufacturing. Accelerometer, rate gyro, and magnetometer sensor bias shifts over temperature (-40 ^{0}C to +71 ^{0}C) are compensated and verified using calibrated thermal chambers and rate tables. The 6-DOF sensor cluster data is fed into a high-speed A/D and DSP processor after passing thru a series of programmable low-pass filters. The DSP processor outputs attitude and navigation data via the RS-232 port. As shown in the block diagram (Figure 9), the NAV440 and AHRS440 include an internal 3-axis magnetometer, and the NAV440 includes an internal WAAS capable GPS receiver. In addition, the AHRS440 and VG440 can accept input from external GPS sources as noted in Chapter 2 of this manual.

Figure 9 440 Series Hardware Block Diagram

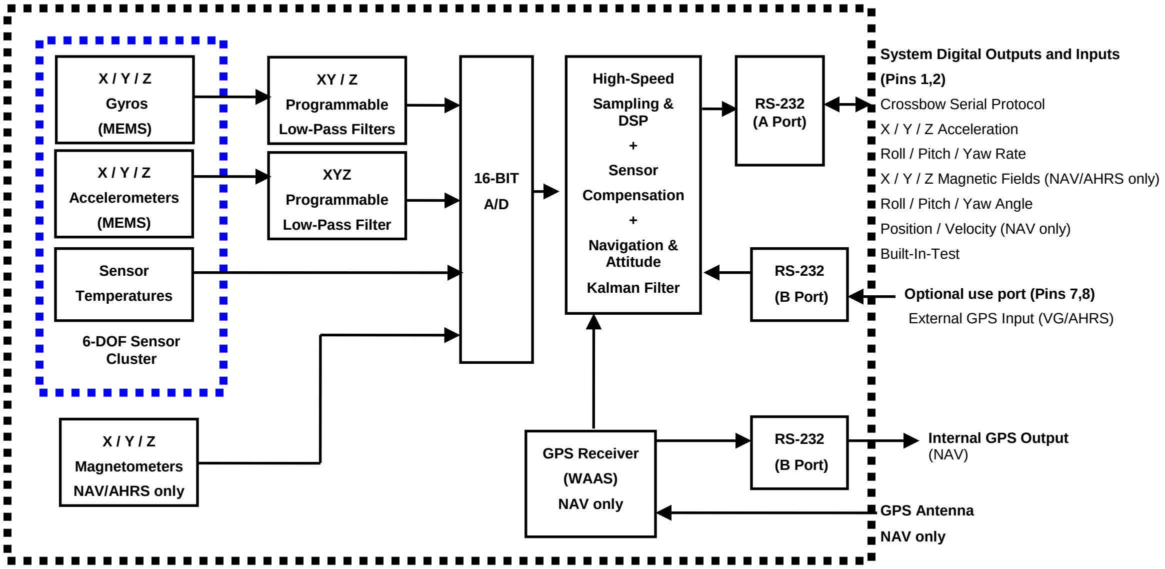
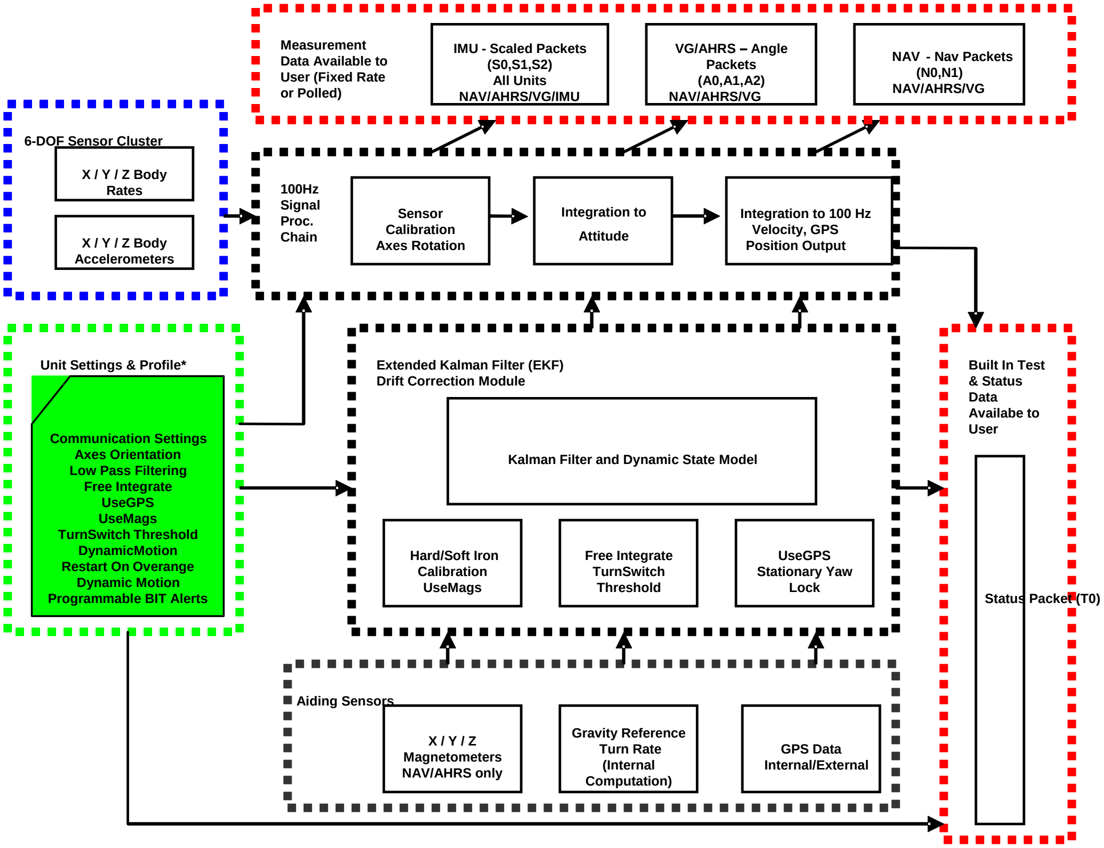


Figure 10 shows the software block diagram. The 6-DOF inertial sensor cluster data is fed into a high speed 100Hz signal processing chain. These 6-DOF signals pass through one or more of the processing blocks and these signals are converted into output measurement data as shown. Measurement data packets are available at fixed continuous output rates or on a polled basis. The type of measurement data packets available depends on the unit type according to the software block diagram and Table 6. Aiding sensor data is used by an Extended Kalman Filter (EKF) for drift correction in the NAV, AHRS and VG Series products. Built-In-Test and Status data is available in the measurement packet or via the special Status Packet T0.

As shown in the software block diagram, the 440 Series has a unit setting and profile block which configures the algorithm to user and application specific needs. This feature is one of the more powerful features in the 440 Series architecture as it allows the 440 Series to work in a wide range of commercial applications by settings different modes of operation for the 440 Series.

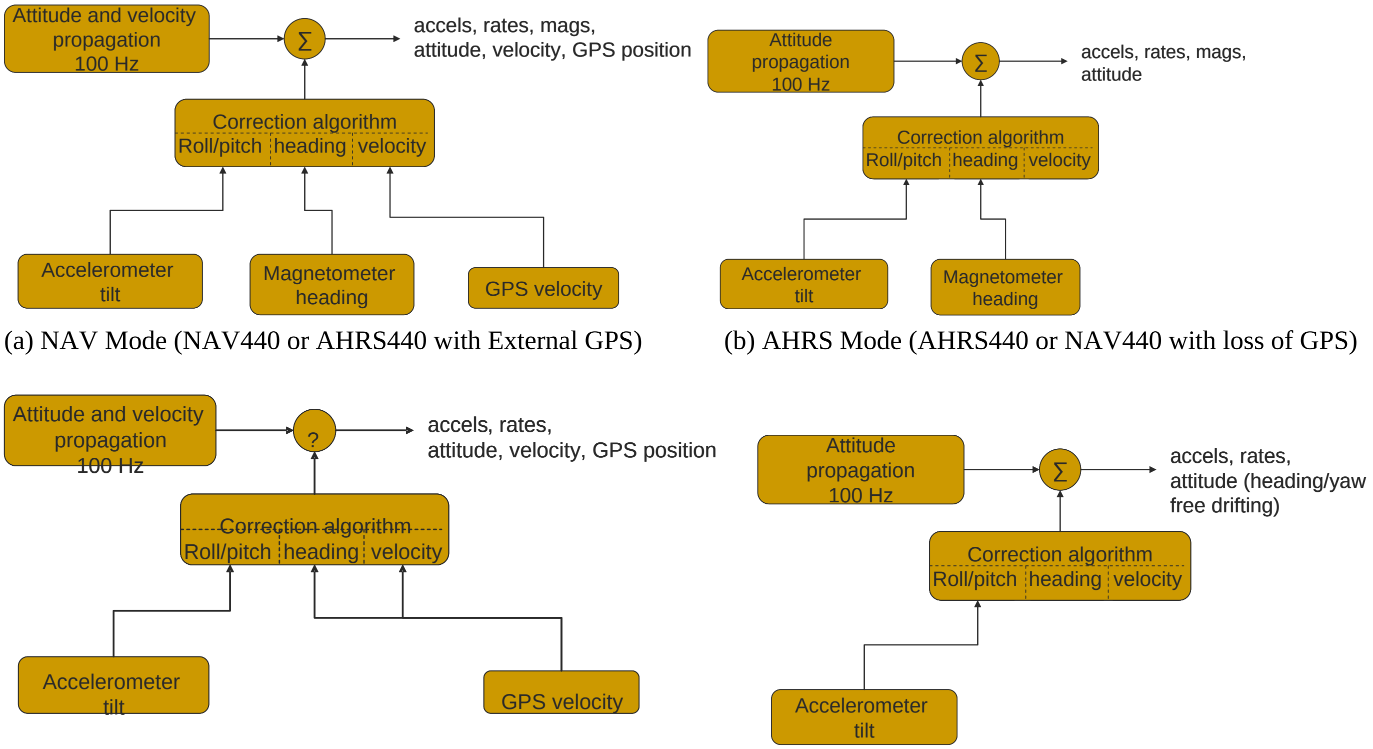
Figure 10 440 Series Software Block Diagram


Measurement Data Available to User (Fixed Rate or Polled)
IMU - Scaled Packets (S0,S1,S2) All Units NAV/AHRS/VG/IMU
VG/AHRS – Angle Packets (A0,A1,A2) NAV/AHRS/VG
NAV - Nav Packets (N0,N1) NAV/AHRS/VG
6-DOF Sensor Cluster
X / Y / Z Body Rates
X / Y / Z Body Accelerometers
100Hz Signal Proc. Chain
Sensor Calibration Axes Rotation
Integration to Attitude
Integration to 100 Hz Velocity, GPS Position Output
Unit Settings & Profile*
Communication Settings
Axes Orientation
Low Pass Filtering
Free Integrate
UseGPS
UseMags
TurnSwitch Threshold
DynamicMotion
Restart On Overange
Dynamic Motion
Programmable BIT Alerts
Extended Kalman Filter (EKF) Drift Correction Module
Kalman Filter and Dynamic State Model
Hard/Soft Iron Calibration UseMags
Free Integrate TurnSwitch Threshold
UseGPS Stationary Yaw Lock
Built In Test & Status Data Availabe to User
Status Packet (T0)
Aiding Sensors
X / Y / Z Magnetometers NAV/AHRS only
Gravity Reference Turn Rate (Internal Computation)
GPS Data Internal/External

Simplified functional block diagrams for NAV, AHRS and VG series products derived from Figure 10 are shown in Figure 11 to highlight key features of each product. The 440 series products are mainly differentiated by types of aiding sensors used in the EKF for the drift correction of the 6-DOF inertial sensor cluster.

For the AHRS product, a 3-axis magnetometer is used for correcting the drift on yaw/heading angle. For the NAV product, a 3-axis magnetometer and a GPS receiver are used for correcting the drift on yaw/heading angle, increasing the accuracy of the attitude estimation by incorporating these sensor signals into the EKF, and providing a navigation solution. The common aiding sensor for the drift correction for the attitude (i.e., roll and pitch only) is a 3-axis accelerometer. This is the default configuration for the VG product.

Figure 11 Functional Block Diagram of NAV, AHRS and VG Default Operating Mode.



(a) NAV Mode (NAV440 or AHRS440 with External GPS)

(b) AHRS Mode (AHRS440 or NAV440 with loss of GPS)

(c) VGS Mode (VG440 with External GPS)

(d) VG Mode (VG440)

4.1 440 Series Default Coordinate System

The 440 Series Inertial System default coordinate system is shown in Figure 12. As with many elements of the 440 Series, the coordinate system is configurable with either NAV-VIEW 2.X or by sending the appropriate serial commands. These configurable elements are known as ***Advanced Settings***. This section of the manual describes the default coordinate system settings of the 440 Series when it leaves the factory.

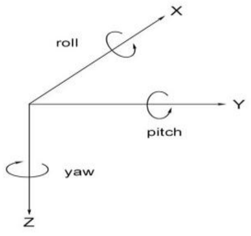
With the 440 Series product connector facing you, and the mounting plate down, the axes are defined as follows:

Figure 12 440 Series Default Coordinate System

X-axis – from face with connector through the 440 unit

Y-axis – along the face with connector from left to right

Z-axis – along the face with the connector from top to bottom



The axes form an orthogonal SAE right-handed coordinate system. Acceleration is positive when it is oriented towards the positive side of the coordinate axis. For example, with a 440 Series product sitting on a level table, it will measure zero g along the x- and y-axes and -1 g along the z-axis. Normal Force acceleration is directed upward, and thus will be defined as negative for the 440 Series z-axis.

The angular rate sensors are aligned with these same axes. The rate sensors measure angular rotation rate around a given axis. The rate measurements are labeled by the appropriate axis. The direction of a positive rotation is defined by the right-hand rule. With the thumb of your right hand pointing along the axis in a positive direction, your fingers curl around in the positive rotation direction. For example, if the 440 Series product is sitting on a level surface and you rotate it clockwise on that surface, this will be a positive rotation around the z-axis. The x- and y-axis rate sensors would measure zero angular rates, and the z-axis sensor would measure a positive angular rate.

The magnetic sensors are aligned with the same axes definitions and sign as the linear accelerometers. For example, when oriented towards magnetic North, you will read approximately +0.25 Gauss along X, 0.0 Gauss along Y, and +0.35 Gauss along Z direction (North America). Magnetic values at other geographic locations can be found at http://www.ngdc.noaa.gov/seg/WMM/DoDWMM.shtml.

Pitch is defined positive for a positive rotation around the y-axis (pitch up). Roll is defined as positive for a positive rotation around the x-axis (roll right). Yaw is defined as positive for a positive rotation around the z-axis (turn right).

The angles are defined as standard Euler angles using a 3-2-1 system. To rotate from the body frame to an earth-level frame, roll first, then pitch, and then yaw.

The position output from GPS is represented in Latitude, Longitude, and Altitude (LLA) convention on the WGS84 Ellipsoid. This is the most commonly used spherical coordinate system. The GPS velocity is defined in North, East and Down reference frame. The users can convert this into Cartesian coordinate system, called Earth-Centered, Earth-Fixed (ECEF). ECEF uses three-dimensional XYZ coordinates (in meters) to describe the location of a GPS user or satellite. Several online resources are available to help users with this transformation. For example, refer to the application note on Crossbow website**,** **http://www.xbow.com/Support/appnotes.htm**

4.1.1 Advanced Settings

The 440 Series Inertial Systems have a number of advanced settings that can be changed. The specific settings available vary from unit to unit, and a detailed description of each unit (IMU, VG, AHRS, and NAV) is found in the subsequent sections of this manual. All units support baud rate, power-up output packet type, output rate, sensor low pass filtering, and custom axes configuration. The units can be configured using NAV-VIEW 2.X, as described in Section 3, and also directly with serial commands as described in Sections 6-9.

4.2 IMU440 Theory of Operation

The product name, IMU440, stands for Inertial Measurement Unit 440, and the name is indicative of the inertial measurement unit functionality that the IMU440 provides by providing inertial rate and acceleration data in 6-DOF (six degrees of freedom). The IMU440 signal processing chain consists of the 6-DOF sensor cluster, programmable low-pass filters, analog to digital conversion, and the DSP signal processor for sensor error compensation. The IMU440, as with other 440 Series variants, has an RS-232 serial communications link.

After passing thru a digitally controlled programmable analog low-pass filter, the rate and acceleration analog sensor signals are sampled and converted to digital data at 1 kHz. The sensor data is filtered and down-sampled to 100Hz by the DSP using FIR filters. The factory calibration data, stored in EEPROM, is used by the DSP to remove temperature bias, misalignment, scale factor errors, and non-linearities from the sensor data. Additionally any advanced user settings such as axes rotation are applied to the IMU data. The 100Hz IMU data is continuously being maintained inside the IMU440. Digital IMU data is output over the RS-232 serial link at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. The digital IMU data is available in one of several measurement packet formats including Scaled Sensor Data ('S1' Packet) and Delta-Theta, Delta-V ('S2' Packet). In the Scaled Sensor Data ('S1' Packet) data is output in scaled engineering units. In the Delta-Theta, Delta-V format ('S2' Packet) scaled sensor data is integrated with respect to the time of the last output packet and the data is reported in units of accumulated (i.e., delta) degrees and meters/second. See Section 7 of the manual for full packet descriptions.

⚐ IMPORTANT

The Delta-Theta, Delta-V packet is only recommended for use in continuous output mode at 5Hz or greater. Polled requests for this packet will produce values accumulated since the last poll request, and thus, are subject to overflow (data type wrap around).

4.2.1 IMU440 Advanced Settings

The IMU440 advanced settings are described in Table 7 below. All of the advanced settings are accessible thru NAV-VIEW 2.X under the Configuration Menu, Unit Configuration settings.

Table 7 IMU440 Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	9600, 19200, 57600 also available
Packet Type	S1	S2 also available
Packet Rate	100Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the IMU440 will only send measurement packets in response to GP commands.
Orientation	See Fig 12.	To configure the axis orientation, select the desired measurement for each axis: NAV-VIEW 2.X will show the corresponding image of the IMU440, so it easy to visualize the mode of operation. Refer to Section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Analog Filter Clocks 1,2, & 3	25 Hz	The low pass filters are set to a default of 25 Hz for the accelerometers, and 25 Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There are two settings for the accelerometers, one for the X and Y axes, and a separate setting for the Z axis. The reason for a separate setting in the Z-axis is that in many installations, the Z-axis vibration level is much higher than in the X and Y axes, and it can prove helpful to filter the Z-axis at a lower cutoff than the X and Y axes. See note on Filter settings below.
BIT		See 4.2.2

NOTE on Analog Filter Clocks

Why change the filter settings? Generally there is no reason to change the low-pass filter settings on the IMU440 or other 440 Series Inertial Systems. However, when a 440 Series product is installed in an environment with a lot of vibration, it can be helpful to reduce the vibration-based signal energy and noise prior to further processing on the signal. Installing the IMU440 in the target environment and reviewing the data with NAV-VIEW 2.X can be helpful to determine if changing the filter settings would be helpful. Although the filter settings can be helpful in reducing vibration based noise in the signal, low filter settings (e.g., 1Hz) also reduce the bandwidth of the signal, i.e. can wash out the signals containing the dynamics of a target. Treat the filter settings with caution.

The analog filter clock settings are shown in default IMU440 coordinate system. If the user has configured the coordinate system to a non-standard or custom configuration, the user must take care to apply the appropriate rotation and configure the filter settings accordingly.

4.2.2 IMU440 Built-In Test

The IMU440 Built-In Test capability allows users of the IMU440 to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the IMU440. See Sections 6-8 Programming Guide, for details on the 'T0' packet.

The BIT word, which is contained within each measurement packet, is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced BIT (Section 9) for details regarding the configuration of the masterStatus flags. Table 8 shows the BIT definition and default settings for BIT programmable alerts in the IMU440.

Table 8 IMU440 Default BIT Status Definition

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT

HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = Alert, Sensor Over Range	Status
hardwareStatus	9	Disabled	Status
comStatus	10	Disabled	Status
softwareStatus	11	Disabled	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The IMU440 also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). In the case of the IMU440 which has fewer features and options than other 440 Series products, the only meaningful parameter is sensor over-range. It is recommended that users leave the default configuration, which is sensorStatus enabled and flag on sensor over-range. The over-range only applies to the rotational rate sensors. Because instantaneous acceleration levels due to vibration can exceed the accelerometer sensor range in many applications, none of the 440 Series products trigger over-range on accelerometer readings.

4.3 VG440 Theory of Operation

The VG440 supports all of the features and operating modes of the IMU440, and it includes additional internal software, running on the DSP processor, for the computation of dynamic roll and pitch. The product name, VG440, stands for Vertical Gyro 440, and it is indicative of the vertical gyro functionality that the VG440 replicates by providing dynamic roll and pitch measurements, in addition to the IMU data. The dynamic roll and pitch measurements are stabilized by the using the accelerometers as a long-term gravity reference. Unlike the VG400 and earlier Crossbow VG Series products, the VG440 can also output a free integrating yaw angle measurement that is not stabilized by a magnetometer or compass heading (see AHRS440 or NAV440 for stabilized heading). At a fixed 100Hz rate, the VG440 continuously maintains both the digital IMU data as well as the dynamic roll and pitch data. As shown in the software block diagram Figure 10, after the Sensor Calibration block, the IMU data is passed into an Integration to Orientation block (Please refer to the Figure 11 if external GPS aiding will be used). The Integration to Orientation block integrates body frame sensed angular rate to orientation at a fixed 100 times per second within all of the 440 Series products. For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, a quaternion formulation is used in the algorithm to provide attitude propagation.

As also shown in the software block diagram, the Integration to Orientation block receives drift corrections from the Extended Kalman Filter or Drift Correction Module. In general, rate sensors and accelerometers suffer from bias drift, misalignment errors, acceleration errors (g-sensitivity), nonlinearity (square terms), and scale factor errors. The largest error in the orientation propagation is associated with the rate sensor bias terms. The Extended Kalman Filter (EKF) module provides an on-the-fly calibration for drift errors, including the rate sensor bias, by providing corrections to the Integration to Orientation block and a characterization of the gyro bias state. In the VG440, the internally computed gravity reference vector provides a reference measurement for the EKF when the VG440 is in quasi-static motion to correct roll and pitch angle drift and to estimate the X and Y gyro rate bias. Because the gravity vector has no horizontal component, the EKF has no ability to estimate either the yaw angle error or the Z gyro rate bias. The VG440 adaptively tunes the EKF feedback in order to best balance the bias estimation and attitude correction with distortion free performance during dynamics when the object is accelerating either linearly (speed changes) or centripetally (false gravity forces from turns). Because centripetal and other dynamic accelerations are often associated with yaw rate, the VG440 maintains a low-passed filtered yaw rate signal and compares it to the turnSwitch threshold field (user adjustable). When the user platform to which the VG440 is attached exceeds the turnSwitch threshold yaw rate, the VG440 lowers the feedback gains from the accelerometers to allow the attitude estimate to coast through the dynamic situation with primary reliance on angular rate sensors. This situation is indicated by the softwareStatus→turnSwitch status flag. Using the turn switch maintains better attitude accuracy during short-term dynamic situations, but care must be taken to ensure that the duty cycle of the turn switch generally stays below 10% during the vehicle mission. A high turn switch duty cycle does not allow the system to apply enough rate sensor bias correction and could allow the attitude estimate to become unstable.

The VG440 algorithm has two major phases of operation. The first phase of operation is the initialization phase. During the initialization phase, the VG440 is expected to be stationary or quasi-static so the EKF weights the accelerometer gravity reference heavily in order to rapidly estimate the roll and pitch angles, and X, Y rate sensor bias. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the VG440 operates with lower levels of feedback (also referred to as EKF gain) from the accelerometers to continuously estimate and correct for roll and pitch errors, as well as to estimate X and Y rate sensor bias.

If a user wants to reset the algorithm or re-enter the initialization phase, sending the algorithm reset command, 'AR', will force the algorithm into the reset phase.

The VG440 outputs digital measurement data over the RS-232 serial link at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. In addition to the scaled sensor packets described in the IMU440 section, the VG440 has additional measurement output packets including the default 'A2' Angle Packet which outputs the roll angle, pitch angle, and digital IMU data. 'N0' and 'N1' packets are also available for use with an external GPS receiver. See Section 6 and 7 of the manual for full packet descriptions.

4.3.1 VG440 Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the VG440 provides additional advanced settings which are selectable for tailoring the VG440 to a specific application requirements. These VG440 advanced settings are shown in Tabel 9 below:

Table 9 VG440 Series Advanced Settings

Setting	**Default**	**Comments**
Baud Rate	38,400 baud	9600, 19200, 57600 also available
Packet Type	A2	S1, S2, N0, N1 also available
Packet Rate	25Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the VG440 will only send measurement packets in response to GP commands.
Orientation	See Fig. 12	To configure the axis orientation, select the desired measurement for each axes: NAV-VIEW 2.X will show the corresponding image of the VG440, so it easy to visualize the mode of operation. See Section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Analog Filter Clocks 1, 2 & 3	5Hz accels 20Hz rates	The low pass filters are set to a default of 5Hz for the accelerometers, and 20 Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There are two settings for the accelerometers, one for the X and Y axes, and a separate setting for the Z axis. The reason for a separate setting in the Z-axis is that in many installations, the Z-axis vibration level is much higher than in the X and Y axes, and it can prove helpful to filter the Z-axis at a lower cutoff than the X and Y axes.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the VG440 into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular rate with no kalman filter based corrections of roll, pitch, or yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the VG440 has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the VG440 has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.

Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the VG440. Algorithm reset returns the VG440 to a high gain state, where the VG440 rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.
Dynamic Motion	ON	The default setting is ON for the VG440. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	10.0 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the VG440 monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are reduced because they are likely corrupted.
BIT		See 4.3.2

4.3.2 VG440 Built-In Test

As with the IMU440, the VG440 Built-In Test capability allows users of the VG440 to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the VG440. See Sections 6 and 7 for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced BIT (Section 9) for details on configuring the masterStatus flags. Table 10 shows the BIT definition and default settings for BIT programmable alerts in the VG440.

Table 10 VG440 Default BIT Status Definition

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alerts	Status
hardwareStatus	9	Disabled	Status

comStatus	10	0 = nominal, 1 = No External GPS Comm	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization or High Gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The VG440 also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). Like the IMU440, Crossbow recommends for the vast majority of users, that the default Status byte for the VG440 is sufficient. For users, who wish to have additional visibility to when the VG440 EFK algorithm estimates that the VG440 is turning about its Z or Yaw axis, the softwareStatus bit can be configured to go high during a turn. In other words, the turnSwitch will turn on the softwareStatus bit. In the VG440, the turnSwitch is by default set at 10.0 deg/sec about the z-axis.

4.4 AHRS440 Theory of Operation

The AHRS440 supports all of the features and operating modes of the IMU440 and VG440, and it includes an additional internal 3-axis magnetometer and associated software running on the DSP processor, for the computation of dynamic heading, as well as dynamic roll and pitch. The product name, AHRS440, stands for Attitude Heading Reference System 440, and it is indicative of the attitude and heading reference functionality that the AHRS440 replicates by providing dynamic heading, roll, and pitch measurements, in addition to the VG and IMU data. The dynamic heading measurement is stabilized using the 3-axis magnetometer as a magnetic north reference. As in the VG440, the dynamic roll and pitch measurements are stabilized using the accelerometers as a long-term gravity reference. Unlike the AHRS400 and earlier Crossbow AHRS Series products, the AHRS440 can be configured to turn on and off the magnetic reference for user defined periods of time (see Section 8 Advanced Commands). In addition, the AHRS440 can accept external GPS data (refer to the NAV440 section for details) for improved performance.

At a fixed 100Hz rate, the AHRS440 continuously maintains the digital IMU data as well as the dynamic roll, pitch, and heading. As showing in Figure 10, after the Sensor Calibration Block, the IMU data is passed to the Integration to Orientation block. The Integration to Orientation block integrates body frame sensed angular rate to orientation at a fixed 100 times per second within all of the 440 Series products. For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, a quaternion formulation is used in the algorithm to provide attitude propagation.

As also shown in the software block diagram, the Integration to Orientation block receives drift corrections from the Extended Kalman Filter or Drift Correction Module. In general, rate sensors and accelerometers suffer from bias drift, misalignment errors, acceleration errors (g-sensitivity), nonlinearity (square terms), and scale factor errors. The largest error in the orientation propagation is associated with the rate sensor bias terms. The Extended Kalman Filter (EKF) module provides an on-the-fly calibration for drift errors, including the rate sensor bias, by providing corrections to the Integration to Orientation block and a characterization of the gyro bias state. In the AHRS440, the internally computed gravity reference vector and the distortion corrected magnetic field vector provide an attitude and a heading reference measurement for the EKF when the AHRS440 is in quasi-static motion to correct roll, pitch, and heading angle drift and to estimate the X, Y and Z gyro rate bias. The AHRS440 adaptively tunes the EKF feedback gains in order to best balance the bias estimation and attitude correction with distortion free performance during dynamics when the object is accelerating either linearly (speed changes) or centripetally (false gravity forces from turns). Because centripetal and other dynamic accelerations are often associated with yaw rate, the AHRS440 maintains a low-passed filtered yaw rate signal and compares it to the turnSwitch threshold field (user adjustable). When the user platform (with the AHRS440 attached) exceeds the turnSwitch threshold yaw rate, the AHRS440 lowers the feedback gains from the accelerometers to allow the attitude estimate to coast through the dynamic situation with primary reliance on angular rate sensors. This situation is indicated by the softwareStatus→turnSwitch status flag. Using the turn switch maintains better attitude accuracy during short-term dynamic situations, but care must be taken to ensure that the duty cycle of the turn switch generally stays below 10% during the vehicle mission. A high turn switch duty cycle does not allow the system to apply enough rate sensor bias correction and could allow the attitude estimate to become unstable.

As described in 4.3 VG440 theory of operation, the AHRS440 algorithm also has two major phases of operation. The first phase of operation is the high-gain initialization phase. During the initialization phase, the AHRS440 is expected to be stationary or quasi-static so the EKF weights the accelerometer gravity reference and Earth's magnetic field reference heavily in order to rapidly estimate the X, Y, and Z rate sensor bias, and the initial attitude and heading of the AHRS440. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the AHRS440 operates with lower levels of feedback (also referred to as EKF gain) from the accelerometers and magnetometers to continuously estimate and correct for roll, pitch, and heading (yaw) errors, as well as to estimate X, Y, and Z rate sensor bias.

The AHRS440 digital data is output over the RS-232 serial link at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. The AHRS400 supports the same scaled sensor and angle mode packet format of the VG440. The AHRS440 defaults to the 'A1' Angle Packet which outputs the roll angle, pitch angle, yaw angle, and digital IMU data. In the AHRS440, the 'A0' and 'A1' packets contain accurate magnetometer readings. See Sections 6 and 7 of the manual for full packet descriptions.

IMPORTANT

For proper operation, the AHRS440 relies on magnetic field readings from its internal 3-axis magnetometer. The AHRS440 must be installed correctly and calibrated for hard-iron and soft iron effects to avoid any system performance degradation. See section 4.4.1 for information and tips regarding installation and calibration.

4.4.1 AHRS440 Magnetometer Calibration and Alignment, Theory of Operation

The AHRS440 uses magnetic sensors to compute heading. Ideally, the magnetic sensors would measure only the earth's magnetic field to compute the heading angle. In the real world, however, residual magnetism in your system will add to the magnetic field measured by the AHRS440. This extra magnetic field will create errors in the heading measurement if they are not accounted for. These extra magnetic fields are called hard iron magnetic fields. In addition, magnetic material can change the direction of the magnetic field as a function of the input magnetic field. This dependence of the local magnetic field on input direction is called the soft iron effect. The AHRS440 can actually measure any constant magnetic field that is associated with your system and correct for it. The AHRS440 can also make a correction for some soft iron effects. The process of measuring these non-ideal effects and correcting for them is called hard iron and soft iron calibration. This calibration will help correct for magnetic fields that are fixed with respect to the AHRS440. It cannot help for time varying fields, or fields created by parts that move with respect to the AHRS440. Because time varying fields cannot be compensated, selection of a proper installation location is important.

During the calibration procedure, the AHRS440 makes a series of measurements while the user system is being turned through a complete 360 degree circle. A 360 degree rotation gives the AHRS440 visibility to hard and soft iron distortion in the horizontal plane. Using NAV-VIEW 2.X, a user can see the hard and soft iron effects by selecting the Misalignment option on the Configuration Menu, and viewing the magnetic circle during the calibration.

The AHRS440 uses these measurements to model the hard iron and soft iron environment in your system, and store these as calibration constants in the EEPROM. The status of the AHRS440 magnetometer calibration is indicated by the softwareError→dataError→magAlignOutOfBounds error flag available in the 'T0' packet. The user can access the hardIron and softIronScaleRatio calibration data as configuration fields in NAV-VIEW 2.X, or by using the communication protocol. Also, the softwareError bit of the masterFail byte within the BIT word is transmitted in every measurement packet. When the AHRS440 has not been properly calibrated, this softwareError bit will be set to fail (high).

In order for the AHRS440 calibration to work properly, the AHRS440 must be installed in your system prior to calibration. If you perform the calibration process with the AHRS440 by itself, you will only correct for the magnetism in the AHRS440 itself. If you then install the AHRS440 in a vehicle (for instance), and the vehicle is magnetic, you will still see errors arising from the magnetism of the vehicle. The AHRS440 must be calibrated after installation and prior to use of the system

The AHRS440 also provides a command interface for initiating the hard iron / soft iron calibration without the using NAV-VIEW 2.X. The user can send a 'WC' command to initiate the calibration, and then rotate the user system through 360 degrees. The 'WC' command has two options – auto-termination and manual termination. With, auto-termination, the AHRS440 tracks the yaw movement and after 380 degrees of rotation returns the calibration complete response, 'CC'. The auto-termination sequence can falsely terminate if the 360 degree rotation is not completed within 2 minutes of the 'WC' command initiation. Manual termination requires the user to send a second 'WC' command with the termination code in the payload. Manual termination is a good option when the user system moves very slowly (e.g., large marine vessel) and completing the 360 degree rotation may require more than two minutes.

The calibration complete, 'CC', command response message contains the X and Y hard iron bias, as well as the soft iron ratio. This information can be interpreted to give an indication of the quality of the calibration. See Section 3.12 for more information on the hard iron bias and soft iron ratio. Section 7 has programming details for the 'WC' and 'CC' commands.

⚠ WARNING

The AHRS440 and NAV440 units must be mounted at least 24" away from large ferrous objects and fluctuating magnetic fields. Failure to locate the unit in a clean magnetic environment will affect the attitude solution.

4.4.2 AHRS440 Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the AHRS440 provides additional advanced settings which are selectable for tailoring the AHRS440 to a specific application requirements. The AHRS440 advanced settings are shown in Table 11 below:

Table 11 AHRS440 Series Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	9600, 19200, 57600 also available
Packet Type	A1	S0, S1, S2, A0, A2, N0, N1 also available
Packet Rate	25 Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the VG440 will only send measurement packets in response to GP commands.
Orientation	See Fig. 12	To configure the axis orientation, select the desired measurement for each axes: NAV-VIEW 2.X will show the corresponding image of the AHRS440, so it easy to visualize the mode of operation. See section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Analog Filter Clocks 1, 2 & 3	5 Hz accels 20 Hz rates	The low pass filters are set to a default of 5Hz for the accelerometers, and 20Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There are two settings for the accelerometers, one for the X and Y axes, and a separate setting for the Z axis. The reason for a separate setting in the Z-axis is that in many installations, the Z-axis vibration level is much higher than in the X and Y axes, and it can prove helpful to filter the Z-axis at a lower cutoff than the X and Y axes.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the AHRS440 into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular rate with no kalman filter based corrections of roll, pitch, or yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch or magnetometer based signals to the yaw. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the AHRS440 has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the AHRS440 has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.
Use Mags	ON	The Use Mags setting allows users to turn on and off the magnetometer feedback for yaw/heading stabilization. The default setting is ON for the AHRS440. When Use Mags is turned ON, the AHRS440 uses the magnetic field sensor readings to stabilize the drift in yaw, and it slaves the yaw to the compass reading provided from the magnetic field sensor readings. When Use Mags is turned OFF, the heading (yaw) angle measurement of the AHRS440 will drift and freely integrate. In effect, this setting converts an AHRS440 into the functionality of the VG440. However, unlike a VG440 this can be done on a selectable basis and changed in real time during a mission. The reason for this setting is to give the user an ability to turn off the magnetometer stabilization when severe magnetic distortion may be occurring. This setting is desirable when the user system temporarily moves

		in close proximity to a large ferrous object. When the Use Mags switch is turned from OFF to ON, the AHRS440 will reinitialize the yaw/heading angle with the compass reading provided from the magnetic field sensor readings.
Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the AHRS440. Algorithm reset returns the AHRS440 to a high gain state, where the AHRS440 rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.
Dynamic Motion	ON	The default setting is ON for the AHRS440. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	0.5 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the AHRS440 monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are reduced because they are likely corrupted.
BIT		See 4.4.3

4.4.3 AHRS440 Built-In Test

As with the IMU440 and VG440, the Built-In Test capability allows users of the AHRS440 to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the AHRS440. See Sections 6 and 7 of the Programming Guide, for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit)) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag. The softwareError bit also provides useful information regarding the status and quality of the AHRS440 magnetic alignment. If the AHRS440 has not been properly magnetically calibrated, the AHRS440 shall indicate a softwareError.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which has been enabled. See Section 9 Advanced BIT for details on configuring the masterStatus flags. Table 12 shows the BIT definition and default settings for BIT programmable alerts in the AHRS440.

Table 12 AHRS440 Default BIT Status Definitions

BITstatus Field	*Bits*	*Meaning*	*Category*
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT

softwareError	3	0 = normal, 1 = internal software error or magAlignOutofBounds	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alerts	Status
hardwareStatus	9	Disabled	Status
comStatus	10	0 = nominal, 1 = No External GPS Comm	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization, or High Gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The AHRS440 also allows a user to configure the Status byte within the BIT message.   To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm).   Like the VG440 and IMU440, Crossbow recommends for the vast majority of users, that the default Status byte for the AHRS440 is sufficient.   For users, who wish to have additional visibility to when the AHRS440 EFK algorithm estimates that the AHRS440 is turning about its Z or Yaw axis, the softwareStatus bit can be configured to go high during a turn. In other words, the turnSwitch will turn on the softwareStatus bit.   In the AHRS440, the turnSwitch is by default set at 0.5 deg/sec about the Z-axis.

4.5 NAV440 Theory of Operation

The NAV440 supports all of the features and operating modes of the IMU/VG/AHRS440, and it includes an additional internal WAAS capable GPS receiver and associated software running on the DSP processor, for the computation of navigation information as well as orientation information.   The product name, NAV440, stands for Navigation System 440, and it is indicative of the navigation reference functionality that the NAV440 provides by outputting GPS information (Latitude, Longitude, and Altitude), inertially-aided 3-axis velocity information, as well as heading, roll, and pitch measurements, in addition to digital IMU data. An AHRS440 configured and properly connected to an external GPS also behaves as a NAV440.

At a fixed 100Hz rate, the NAV440 continuously maintains the digital IMU data; the dynamic roll, pitch, and heading data; as well as the navigation data. As shown in the software block diagram in Figure 10, after the Sensor Calibration block, the IMU data is passed into an Integration to Orientation block. The Integration to Orientation block integrates body frame sensed angular rate to orientation at a fixed 100 times per second within all of the 440 Series products (except IMU440). For improved accuracy and to avoid singularities when dealing with the cosine rotation matrix, a quaternion formulation is used in the algorithm to provide attitude propagation. Following the integration to orientation block, the body frame accelerometer signals are rotated into the NED level frame and are integrated to velocity. At this point, the data is blended with GPS position data, and output as a complete navigation solution.

As shown in Figure 10, the Integration to Orientation and the Integration to Velocity signal processing blocks receive drift corrections from the Extended Kalman Filter (EKF) drift correction module. The drift correction module uses data from the aiding sensors, when they are available, to correct the errors in the velocity, attitude, and heading outputs. Additionally, when aiding sensors are available corrections to the rate gyro and accelerometers are performed.

The NAV440 blends GPS derived heading and accelerometer measurements into the EKF update depending on the health and status of the associated sensors. If the GPS link is lost or poor, the Kalman Filter solution stops tracking accelerometer bias, but the algorithm continues to apply gyro bias correction and provides stabilized angle outputs. The EKF tracking states are reduced to angles and gyro bias only. The accelerometers will continue to integrate velocity, however, accelerometer noise, bias, and attitude error will cause the velocity estimates to start drifting within a few seconds. The attitude tracking performance will degrade, the heading will freely drift, and the filter will revert to the VG only EKF formulation. The UTC packet synchronization will drift due to internal clock drift.

The status of GPS signal acquisition can be monitored from the hardwareStatus BIT as discussed in Section 4.5.3 NAV440 Built in Test. From a cold start, it typically takes 40 seconds for GPS to lock. The actual lock time depends on the antenna's view of the sky and the number of satellites in view.

The DSP performs time-triggered trajectory propagation at 100Hz and will synchronize the sensor sampling with the GPS UTC (Universal Coordinated Time) second boundary when available.

As with the AHRS440 and VG440, the algorithm has two major phases of operation. Immediately after power-up, the NAV440 uses the accelerometers and magnetometers to compute the initial roll, pitch and yaw angles. The roll and pitch attitude will be initialized using the accelerometer's reference of gravity, and yaw will be initialized using the leveled magnetometers X and Y axis reference of the earth's magnetic field. During the first 60 seconds of startup, the NAV440 should remain approximately motionless in order to properly initialize the rate sensor bias. The initialization phase lasts approximately 60 seconds, and the initialization phase can be monitored in the softwareStatus BIT transmitted by default in each measurement packet. After the initialization phase, the NAV440 operates with lower levels of feedback (also referred to as EKF gain) from the GPS, accelerometers, and magnetometers.

Digital data is output over the RS-232 serial link at a selectable fixed rate (100, 50, 25, 20, 10, 5 or 2 Hz) or on as requested basis using the GP, 'Get Packet' command. In addition to the angle mode packets of the AHRS440 and scaled sensor packets of the IMU440, the NAV440 has additional output measurement packets including the default 'N1' Navigation Packet which outputs the Latitude, Longitude, Altitude, X,Y,Z velocities, accelerations, and roll angle, pitch angle, yaw angle, and digital IMU data. See Section 6 and 7 of the manual for full packet descriptions.

⚐ IMPORTANT

For proper operation, the NAV440 relies on magnetic field readings from its internal 3-axis magnetometer. The NAV440 must be installed correctly and calibrated for hard-iron and soft iron effects to avoid any system performance degradation.. See section 4.4.1 for information and tips regarding installation and calibration and why magnetic calibration is necessary. Please review this section of the manual before proceeding to use the NAV440.

⚐ IMPORTANT

For optimal performance the NAV440 utilizes GPS readings from its internal GPS receiver. The GPS receiver requires proper antennae installation for operation. See section 4.4.1 for information and tips regarding installation and calibration.

4.5.1 NAV440 Magnetometer Calibration and Alignment

The NAV440 requires the three axis magnetic field sensor to be calibrated while installed in its operating environment. See section 4.4.1 for information and tips regarding installation and calibration and why magnetic calibration is necessary. Please review this section of the manual before proceeding to use the NAV440.

4.5.2 NAV440 Advanced Settings

In addition to the configurable baud rate, packet rate, axis orientation, and sensor low-pass filter settings, the NAV440 provides additional advanced settings which are selectable for tailoring the NAV440 to a specific application requirements. The NAV440 advanced settings are shown in Table 13 below:

Table 13 NAV440 Series Advanced Settings

Setting	Default	Comments
Baud Rate	38,400 baud	9600, 19200, 57600 also available
Packet Type	N1	S0, S1, S2, A0, A1, A2, N0 also available
Packet Rate	25 Hz	This setting sets the rate at which selected Packet Type, packets are output. If polled mode is desired, then select Quiet. If Quiet is selected, the NAV440 will only send measurement packets in response to GP commands.
Orientation	See Fig. 12	To configure the axis orientation, select the desired measurement for each axes; NAV-VIEW 2.X will show the corresponding image of the NAV440, so it easy to visualize the mode of operation. See section 8.4 Orientation Field settings for the twenty four possible orientation settings. The default setting points the connector AFT.
Analog Filter Clocks 1, 2 & 3	5 Hz 20 Hz	The low pass filters are set to a default of 5Hz for the accelerometers, and 20Hz for the angular rate sensors. There is one filter setting for all three angular rate sensors. There are two settings for the accelerometers, one for the X and Y axes, and a separate setting for the Z axis. The reason for a separate setting in the Z-axis is that in many installations, the Z-axis vibration level is much higher than in the X and Y axes, and it can prove helpful to filter the Z-axis at a lower cutoff than the X and Y axes.
Freely Integrate	OFF	The Freely Integrate setting allows a user to turn the NAV440 into a 'free gyro'. In free gyro mode, the roll, pitch and yaw are computed exclusively from angular

		rate with no kalman filter based corrections of roll, pitch, and yaw. When turned on, there is no coupling of acceleration based signals into the roll and pitch or magnetometer based signal to the yaw. As a result, the roll, pitch, and yaw outputs will drift roughly linearly with time due to sensor bias. For best performance, the Freely Integrate mode should be used after the algorithm has initialized. This allows the Kalman Filter to estimate the roll and pitch rate sensor bias prior to entering the free gyro mode. Upon exiting the 'free gyro' mode (OFF), one of two behaviors will occur (1) If the NAV440 has been in freely integrate mode for less than sixty seconds, the algorithm will resume operation at normal gain settings (2) If the NAV440 has been in freely integrate mode for greater than sixty seconds, the algorithm will force a reset and reinitialize with high gains automatically.
Use GPS	ON	The Use GPS setting allows users to turn on and off the GPS feedback. The default setting is ON for the NAV440. When Use GPS is turned OFF, the NAV440's behavior will revert to that of an AHRS440. See the AHRS440 Theory of Operation for detailed description.
Stationary Yaw Lock	OFF	This setting defaults to OFF on the NAV440, and it is recommended to be OFF for the NAV440. The stationary yaw lock setting is only recommended for consideration when the NAV440 is operating with GPS (Use GPS = ON) and WITHOUT magnetometer feedback (Use Mags = OFF). Stationary yaw lock may be appropriate if the user platform is a wheeled land vehicle.
Use Mags	ON	The Use Mags setting allows users to turn on and off the magnetometer feedback for yaw/heading stabilization. The default setting is ON for the NAV440. When Use Mags is turned ON, the NAV440 uses the magnetic field sensor readings to stabilize the drift in yaw, and it slaves the yaw to the compass reading provided from the magnetic field sensor readings. When UseMags is turned OFF, the heading (yaw) angle measurement of the NAV440 will be slaved to the GPS heading if GPS is available, otherwise the heading will drift feely. The reason for this setting is to give the user an ability to turn off the magnetometer stabilization when severe magnetic distortion may be occurring. This setting is desirable when the user vehicle temporarily moves in close proximity to a large ferrous object. When the Use Mags switch is turned from OFF to ON, the NAV440 will reinitialize the yaw/heading angle with the compass reading provided from the magnetic field sensor readings.
Restart On Over Range	OFF	This setting forces an algorithm reset when a sensor over range occurs i.e., a rotational rate on any of the three axes exceeds the maximum range. The default setting is OFF for the NAV440. Algorithm reset returns the NAV440 to a high gain state, where the NAV440 rapidly estimates the gyro bias and uses the accelerometer feedback heavily. This setting is recommended when the source of over-range is likely to be sustained and potentially much greater than the rate sensor operating limit. Large and sustained angular rate over-ranges result in unrecoverable errors in roll and pitch outputs. An unrecoverable error is one where the EKF can not stabilize the resulting roll and pitch reading. If the over-ranges are expected to be of short duration (<1 sec) and a modest percentage over the maximum operating range, it is recommended that the restart on over range setting be turned off. Handling of an inertial rate sensor over-range is controlled using the restartOnOverRange switch. If this switch is off, the system will flag the overRange status flag and continue to operate through it. If this switch is on, the system will flag a masterFail error during an over-range condition and continue to operate with this flag until a quasi-static condition is met to allow for an algorithm restart. The quasi-static condition required is that the absolute value of each low-passed rate sensor fall below 3 deg/sec to begin initialization. The system will then attempt a normal algorithm start.

Dynamic Motion	ON	The default setting is ON for the NAV440. Turning off the dynamic motion setting results in a higher gain state that uses the accelerometer feedback heavily. During periods of time when there is known low dynamic acceleration, this switch can be turned off to allow the attitude estimate to quickly stabilize.
Turn Switch threshold	0.5 deg/sec	With respect to centripetal or false gravity forces from turning dynamics (or coordinated turn), the NAV440 monitors the yaw-rate. If the yaw rate exceeds a given Turnswitch threshold, the feedback gains from the accelerometer signals for attitude correction are reduced because they are likely corrupted.
BIT		See 4.5.3

4.5.3 NAV440 Built-In Test

As with the IMU, VG and AHRS440, the Built-In Test capability allows users of the NAV440 to monitor health, diagnostic, and system status information of the unit in real-time. The Built-In Test information consists of a BIT word (2 bytes) transmitted in every measurement packet. In addition, there is a diagnostic packet 'T0' that can be requested via the Get Packet 'GP' command which contains a complete set of status for each hardware and software subsystem in the NAV440. See Section 6 and 7 Programming Guide, for details on the 'T0' packet.

The BIT word contained within each measurement packet is detailed below. The LSB (Least Significant Bit) is the Error byte, and the MSB (Most Significant Bit) is a Status byte with programmable alerts. Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is the masterFail flag. The softwareError bit also provides useful information regarding the status and quality of the NAV440 magnetic alignment. If the NAV440 has not been properly magnetically calibrated, the NAV440 shall indicate a softwareError.

The masterStatus flag is a configurable indication that can be modified by the user. This flag is asserted as a result of any asserted alert signals which have been enabled. See Advanced Settings for details for configuring the masterStatus flags. Table 14 shows the BIT definition and default settings for BIT programmable alerts in the NAV440.

Table 14 NAV440 Default BIT Status Definitions

BITstatus Field	***Bits***	***Meaning***	***Category***
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error or magAlignOutofBounds	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = one or more status alert	Status
hardwareStatus	9	0 = nominal, 1 = Internal GPS unlocked or 1PPS invalid	Status
comStatus	10	Disabled	Status
softwareStatus	11	0 = nominal, 1 = Algorithm Initialization or high gain	Status
sensorStatus	12	0 = nominal, 1 = Sensor Over Range	Status
Reserved	13:15	N/A	

The NAV440 also allows a user to configure the Status byte within the BIT message. To configure the word, select the BIT Configuration tab from the Unit Configuration menu. The dialog box allows selection of which status types to enable (hardware, software, sensor, and comm). Like the IMU, VG and AHRS440, Crossbow recommends for the vast majority of users, that the default Status byte for the NAV440 is sufficient. For users, who wish to have additional visibility or alerts relative to the GPS sensor status or algorithm status, they can configure additional triggers for both the softwareStatus and hardwareStatus (See Sections 9 of the user's manual for all the BIT fields).

5 Application Guide

5.1 Introduction

This section provides recommended advanced settings for tailoring the 440 series of inertial systems to different types of application and platform requirements.

5.2 Fixed Wing Aircraft

A fixed-wing aircraft is a heavier-than-air craft where movement of the wings in relation to the aircraft is not used to generate lift. The term is used to distinguish from rotary-wing aircraft, where the movement of the wing surfaces relative to the aircraft generates lift. The fixed wing aircraft can range in size from the smallest experimental plane to the largest commercial jet. The dynamic characteristics of the fixed wing aircraft depends upon types of aircraft (i.e., glider, propeller aircraft, and jet aircraft) and mission phases (i.e., launch, landing, and maneuver). In order to meet application requirements, users must dial in proper advanced settings so that the 440 series can provide the best possible solution under given dynamic conditions. For example, Table 17 provides the recommended advanced settings for four different dynamic conditions.

Table 17: Recommended Advanced Settings for Fixed Wing Aircraft

Recommended Product	***AHRS440 or NAV440***			
Recommended Settings	***Dynamic Condition***			
	Pre-launch or known straight and level un-accelerated flight	***Launch***	***Normal Dynamics (Default)***	***High Dynamics***
UseMags	ON	ON	ON	ON
UseGPS	ON	ON (< 4g)	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF§	OFF	OFF (< 2g)
Stationary Yaw Lock	OFF	OFF	OFF	OFF
Restart Over Range	ON	OFF	OFF	OFF
Dynamic Motion	OFF	ON	ON	ON
Turn Switch Threshold	0.5 deg/s	0.5 deg/s	0.5 deg/s	0.5 deg/s
XY Filter Accel	5 Hz	5 Hz	5 Hz*	15 Hz
Z Filter Accel	5 Hz	5 Hz	5 Hz*	15 Hz
Filter Rate Sensor	20 Hz	20 Hz	20 Hz*	20 Hz

*A cutoff frequency of filters may be varied depending on the fastest dynamic mode of the aircraft. For example, the conventional aircraft has five dynamic modes, short-period, phugoid, spiral, dutch-roll, and roll, and the fastest one is the roll mode. The natural frequency of this mode is around 6~8 radian/sec or (about 2 Hz) in most cases. Therefore, the recommended filter setting would not reject desired frequency components (or dynamic modes) that one wants to capture. However, the larger the bandwidth (or cutoff frequency) is, the noisier the corresponding signal is, which may result in the performance degradation. If the aircraft is operated under severe vibrations, also, the recommended filter setting may need to be further reduced in order to reject the frequency components caused by the vibration.

§FreelyIntegrate should only be set to "ON" for severe launch conditions. Normal takeoff dynamics that a standard aircraft would experience will see the best performance with this setting in the "OFF" position.

5.3 Rotorcraft

Rotorcraft is a category of heavier-than-air flying machines that use lift generated by rotors. They may also include the use of static lifting surfaces, but the primary distinguishing feature being lift provided by rotating lift structures. Rotorcraft includes helicopters, autogyros, gyrodynes and tiltrotors.

The rotor blade dynamics itself is much faster than that of the fixed wing aircraft and contains high frequency components. At the same time, however, it may cause severe vibrations on the airframe. Also, the overall dynamics (translational and rotational motion) of the rotor craft is much slower than the fixed wing aircraft due to a mechanical mechanism of rotors generating the aerodynamic forces and moments. Table 18 provides the recommended advanced settings for two different dynamic conditions.

Table 18: Recommended Advanced Settings for Rotorcraft

Recommended Product	***AHRS440 or NAV440***	
Recommended Settings	***Dynamic Condition***	
	Normal Dynamics	***High Dynamics (with uncoordinated tail motion)***
UseMags	ON	ON
UseGPS	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF (< 2g)
Stationary Yaw Lock	OFF	OFF
Restart Over Range	OFF	ON
Dynamic Motion	ON	ON
Turn Switch Threshold	1.0 deg/s §	30.0 deg/s §
XY Filter Accel	5 Hz*	5 Hz
Z Filter Accel	5 Hz*	5 Hz
Filter Rate Sensor	20 Hz*	20 Hz

§The helicopter can change its heading angle rapidly unlike the aircraft which requires banking. A turn switch threshold that is too low may cause turn switch activation with high duty cycle causing random walk in roll and pitch angles due to low feedback gains.

*A cutoff frequency must be far away from major frequency components caused by the rotor vibration.

5.4 Land Vehicle

Some examples of land vehicles are: Automobiles, trucks, heavy equipment, trains, snowmobiles, and other tracked vehicles. Table 19 provides the recommended advanced settings for two different types of application.

Table 19: Recommended Advanced Settings for Land Vehicle

Recommended Product	VG440 or NAV440	
Recommended Settings	Dynamic Condition	
	Heavy Equipment Application	Automotive Testing (IMU and VG default)
UseMags	ON[§]	ON[§]
UseGPS	ON	ON (< 4g)
FreelyIntegrate	OFF	OFF
Stationary Yaw Lock	OFF	OFF
Restart Over Range	ON	OFF
Dynamic Motion	ON	ON
Turn Switch Threshold	5.0 deg/s	10.0 deg/s
XY Filter Accel	5 Hz	5 Hz
Z Filter Accel	5 Hz	5 Hz
Filter Rate Sensor	20 Hz	20 Hz

[§] When not in distorted magnetic environment.

5.5 Water Vehicle

Water vehicle is a craft or vessel designed to float on or submerge and provide transport over and under water. Table 20 provides the recommended advanced settings for two different types of application.

Table 20: Recommended Advanced Settings for Water Vehicle

Recommended Product	***NAV440***	
Recommended Settings	***Application***	
	Surfaced	***Submerged***
UseMags	ON§	ON§
UseGPS	ON	OFF
FreeIntegrate	OFF	OFF
Stationary Yaw Lock	OFF	OFF
Restart Over Range	OFF	OFF
Dynamic Motion	ON	ON
Turn Switch Threshold	10 deg/s	5 deg/s
XY Filter Accel	5 Hz	2 Hz
Z Filter Accel	5 Hz	2 Hz
Filter Rate Sensor	15 Hz	10 Hz

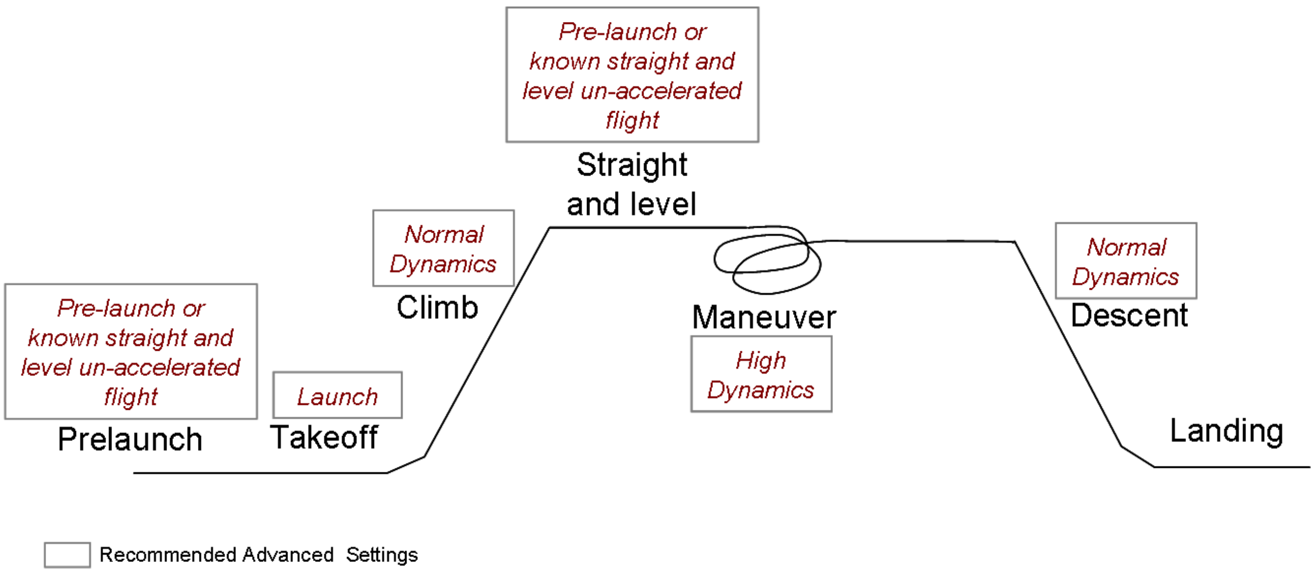
§When not in distorted magnetic environment.

☑ EXAMPLE

Figure 13 shows a typical flight profile of the fixed wing aircraft and the corresponding advanced settings that one can configure adaptively depending on a flight phase:

- **Prelaunch** is the phase of flight in which an aircraft goes through a series of checkups (hardware and software) on the ground before takeoff. The aircraft is a static condition,
- **Takeoff** is the phase of flight in which an aircraft goes through a transition from moving along the ground (taxiing) to flying in the air, usually along a runway. The aircraft is under horizontal acceleration and may suffer from vibrations coming from an engine and ground contact forces transmitted from its landing gear..
- **Climb** is the phase of a flight, after take off, consisting of getting the aircraft to the desired flight level altitude. More generally, the term 'climb' means increasing the altitude. The aircraft is under vertical acceleration until it reaches the steady-state climb rate.
- **Straight and level flight** is the phase of flight in which an aircraft reaches its nominal flight altitude and maintains its speed and altitude. The aircraft is under equilibrium.
- **Maneuver** is the phase of flight in which an aircraft accelerates, decelerates, and turns. The aircraft is under non-gravitational acceleration and/or deceleration.
- **Descent** is the phase of flight in which an aircraft decreases altitude for an approach to landing. The aircraft is under vertical deceleration until it captures a glide slope.
- **Landing** is the last part of a flight, where the aircraft returns to the ground.

Figure 13 Typical flight profiles of fixed wing aircraft and the corresponding advanced settings.


Pre-launch or known straight and level un-accelerated flight
Straight and level
Normal Dynamics
Climb
Maneuver
High Dynamics
Normal Dynamics
Descent
Pre-launch or known straight and level un-accelerated flight
Prelaunch
Launch
Takeoff
Landing
□ Recommended Advanced Settings

6 Programming Guide

The 440 Series contains a number of different products which have different measurement capabilities. Depending on the model you purchased, various commands and output modes are supported. However, all models support a common packet structure that includes both command or input data packets (data sent to the 440 Series) and measurement output or response packet formats (data sent from the 440 Series). This section of the manual explains these packet formats as well as the supported commands. NAV-VIEW 2.X also features a number of tools that can help a user understand the packet types available and the information contained within the packets. This section of the manual assumes that the user is familiar with ANSI C programming language and data type conventions.

For an example of the code required to parse input data packets, please see refer to Appendix C.

For qualified commercial OEM users, a source code license of NAV-VIEW 2.X can be made available under certain conditions. Please contact your Crossbow Technology representative for more information.

6.1 General Settings

The serial port settings are RS232 with 1 start bit, 8 data bits, no parity bit, 1 stop bit, and no flow control. Standard baud rates supported are: 9600, 19200, 38400, and 57600.

Common definitions include:

- A word is defined to be 2 bytes or 16 bits.
- All communications to and from the unit are packets that start with a single word alternating bit preamble 0x5555. This is the ASCII string "UU".
- All multiple byte values are transmitted Big Endian (Most Significant Byte First).
- All communication packets end with a single word CRC (2 bytes). CRC's are calculated on all packet bytes excluding the preamble and CRC itself. Input packets with incorrect CRC's will be ignored.
- Each complete communication packet must be transmitted to the 440 Series inertial system within a 4 second period.

6.2 Number Formats

Number Format Conventions include:

- 0x as a prefix to hexadecimal values
- single quotes (‘’) to delimit ASCII characters
- no prefix or delimiters to specify decimal values.

The following table defines number formats:

Table 21: Number Formats

Descriptor	Description	Size (bytes)	Comment	Range
U1	Unsigned Char	1		0 to 255
U2	Unsigned Short	2		0 to 65535
U4	Unsigned Int	4		0 to 2^32-1
I2	Signed Short	2	2's Complement	-2^15 to 2^15-1
I2*	Signed Short	2	Shifted 2's Complement	Shifted to specified range
I4	Signed Int	4	2's Complement	-2^31 to 2^31-1
F4	Floating Point	4	IEEE754 Single Precision	-1*2^127 to 2^127
SN	String	N	ASCII	

6.3 Packet Format

All of the Input and Output packets, except the Ping command, conform to the following structure:

0x5555	*<2-byte packet type (U2)>*	*<payload byte-length (U1)>*	*<variable length payload>*	*<2-byte CRC (U2)>*

The Ping Command does not require a CRC, so a 440 Series unit can be pinged from a terminal emulator. To Ping a 440 Series unit, type the ASCII string 'UUPK'. If properly connected, the 440 Series unit will respond with 'PK'. All other communications with the 440 Series unit require the 2-byte CRC. {Note: A 440 Series unit will also respond to a ping command using the full packet formation with payload 0 and correctly calculated CRC. Example: 0x5555504B009ef4 }.

6.3.1 Packet Header

The packet header is always the bit pattern 0x5555.

6.3.2 Packet Type

The packet type is always two bytes long in unsigned short integer format. Most input and output packet types can be interpreted as a pair of ASCII characters. As a semantic aid consider the following single character acronyms:

P = packet

F = fields

Refers to Fields which are settings or data contained in the unit

E = EEPROM

Refers to factory data stored in EEPROM

R = read

Reads default non-volatile fields

G = get

Gets current volatile fields or settings

W = write

Writes default non-volatile fields. These fields are stored in non-volatile memory and determine the unit's behavior on power up. Modifying default fields take effect on the next power up and thereafter.

S = set

Sets current volatile fields or settings. Modifying current fields will take effect immediately by modifying internal RAM and are lost on a power cycle.

6.3.3 Payload Length

The payload length is always a one byte unsigned character with a range of 0-255. The payload length byte is the length(in bytes) of the *<variable length payload>* portion of the packet ONLY, and does not include the CRC.

6.3.4 Payload

The payload is of variable length based on the packet type.

6.3.5 16-bit CRC-CCITT

Packets end with a 16-bit CRC-CCITT calculated on the entire packet excluding the 0x5555 header and the CRC field itself. A discussion of the 16-bit CRC-CCITT and sample code for implementing the computation of the CRC is included at the end of this document. This 16-bit CRC standard is maintained by the International Telecommunication Union (ITU). The highlights are:

Width = 16 bits

Polynomial 0x1021

Initial value = 0xFFFF

No XOR performed on the final value.

See Appendix C for sample code that implements the 16-bit CRC algorithm.

6.3.6 Messaging Overview

The following table summarizes the messages available by 440 Series model. Packet types are assigned mostly using the ASCII mnemonics defined above and are indicated in the summary table below and in the detailed sections for each command. The payload byte-length is often related to other data elements in the packet as defined in the table below. The referenced variables are defined in the detailed sections following. Output messages are sent from the 440 Series inertial system to the user system as a result of a poll request or a continuous packet output setting. Input messages are sent from the user system to the 440 Series inertial system and will result in an associated Reply Message or NAK message. Note that reply messages typically have the same ***<2-byte packet type (U2)>*** as the input message that evoked it but with a different payload.

Message Table

ASCII Mnemonic	***<2-byte packet type (U2)>***	***<payload byte-length (U1)>***	**Description**	**Type**	**Products Available**
Link Test					
PK	0x504B	0	Ping Command and Response	Input/Reply Message	ALL
CH	0x4348	N	Echo Command and Response	Input/Reply Message	ALL
Interactive Commands					
GP	0x4750	2	Get Packet Request	Input Message	ALL
AR	0x4152	0	Algorithm Reset	Input/Reply Message	VG,AHRS, NAV
SR	0x5352	0	Software Reset	Input/Reply Message	ALL
NAK	0x1515	2	Error Response	Reply Message	ALL
WC	0x5743	2	Calibrate Command and Response	Input/Reply Message	AHRS,NAV
CC	0x4343	8	Calibration Completed	Reply Message	AHRS,NAV
Output Messages: Status & Other, (Polled Only)					
ID	0x4944	5+N	Identification Data	Output Message	ALL
VR	0x5652	5	Version Data	Output Message	ALL
T0	0x5430	28	Test 0 (Detailed BIT and Status)	Output Message	ALL
Output Messages: **Measurement Data** **(Continuous or Polled)**					
S0	0x5330	30	Scaled Sensor 0 Data	Output Message	AHRS, NAV
S1	0x5331	24	Scaled Sensor 1 Data	Output Message	ALL
S2	0x5332	28	Scaled Sensor 2	Output	ALL

			Data	Message	
A0	0x4130	30	Angle 0 Data	Output Message	AHRS, NAV
A1	0x4131	32	Angle 1 Data	Output Message	AHRS, NAV
A2	0x4132	30	Angle 2 Data	Output Message	VG, AHRS, NAV
N0	0x4E30	32	Nav 0 Data	Output Message	VG, AHRS, NAV
N1	0x4E31	42	Nav 1 Data	Output Message	VG,,AHRS, NAV
B1	0x4231	18	Short Packet Angle B1 Data	Output Message	OEM Only
B2	0x4232	10	Short Packet Angle B2 Data	Output Message	OEM Only
Advanced Commands					
WF	0x5746	numFields*4 +1	Write Fields Request	Input Message	ALL
WF	0x5746	numFields*2 +1	Write Fields Response	Reply Message	ALL
SF	0x5346	numFields*4 +1	Set Fields Request	Input Message	ALL
SF	0x5346	numFields*2 +1	Set Fields Response	Reply Message	ALL
RF	0x5246	numFields*2 +1	Read Fields Request	Input Message	ALL
RF	0x5246	numFields*4 +1	Read Fields Response	Reply Message	ALL
GF	0x4746	numFields*2 +1	Get Fields Request	Input Message	ALL
GF	0x4746	numFields*4 +1	Get Fields Response	Reply Message	ALL

7 Communicating with the 440 Series

7.1 Link Test.

7.1.1 Ping Command

Ping ('PK' = 0x504B)			
Preamble	*Packet Type*	*Length*	*Termination*
0x5555	0x504B	-	-

The ping command has no payload. Sending the ping command will cause the unit to send a ping response. To facilitate human input from a terminal, the length and CRC fields are not required. (Example: 0x5555504B009ef4 or 0x5555504B))

7.1.2 Ping Response

Ping ('PK' = 0x504B)			
Preamble	*Packet Type*	*Length*	*Termination*
0x5555	0x504B	*0x00*	*<CRC (U2)>*

The unit will send this packet in response to a ping command.

7.1.3 Echo Command

Echo ('CH' = 0x4348)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4348	N	*<echo payload>*	*<CRC (U2)>*

The echo command allows testing and verification of the communication link. The unit will respond with an echo response containing the *echo data*. The *echo data* is N bytes long.

7.1.4 Echo Response

Echo Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	echoData0	U1	-	-	first byte of echo data
1	echoData1	U1	-	-	Second byte of echo data
…	…	U1	-	-	Echo data
N-2	echoData...	U1	-	-	Second to last byte of echo data
N-1	echoData…	U1	-	-	Last byte of echo data

7.2 Interactive Commands

Interactive commands are used to interactively request data from the 440 Series, and to calibrate or reset the 440 Series.

7.2.1 Get Packet Request

Get Packet ('GP' = 0x4750)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4750	0x02	*<GP payload>*	*<CRC (U2)>*

This command allows the user to poll for both measurement packets and special purpose output packets including 'T0', 'VR', and 'ID'.

GP Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	requestedPacketType	U2	-	-	The requested packet type

Refer to the sections below for Packet Definitions sent in response to the 'GP' command

7.2.2 Algorithm Reset Command

Algorithm Reset ('AR' = 0x4152)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4152	0x00	-	*<CRC (U2)>*

This command resets the state estimation algorithm without reloading fields from EEPROM. All current field values will remain in affect. The unit will respond with an algorithm reset response.

7.2.3 Algorithm Reset Response

Algorithm Reset ('AR' = 0x4152)			
Preamble	*Packet Type*	*Length*	*Termination*
0x5555	0x4152	0x00	*<CRC (U2)>*

The unit will send this packet in response to an algorithm reset command.

7.2.4 Software Reset Command

Software Reset ('SR' = 0x5352)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5352	0x00	-	*<CRC (U2)>*

This command performs a core CPU reset, functionally equivalent to a power cycle. All default power-up field settings will apply. The unit will respond with a software reset response before the system goes down.

7.2.5 Software Reset Response

Software Reset ('SR' = 0x5352)			
Preamble	*Packet Type*	*Length*	*Termination*
0x5555	0x5352	0x00	*<CRC (U2)>*

The unit will send this packet in response to a software reset command.

7.2.6 Calibrate Command

Calibrate ('WC' = 0x5743)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5743	0x02	*<WC payload>*	*<CRC (U2)>*

This command allows the user to perform various calibration tasks with the 440 Series. See the calibration command table below for details. The unit will respond immediately with a calibrate response containing the *calibrationRequest* received or an error response if the command cannot be performed.

WC Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	calibrationRequest	U2	-	-	The requested calibration task

Currently, magnetic alignment is the only function supported by the calibrate command. There are two magnetic alignment procedures supported; (1) magnetic alignment with automatic yaw tracking termination, and magnetic alignment without automatic termination.

calibrationRequest	*Description*
0x0009	**Begin magnetic alignment without automatic termination.** Rotate vehicle through >360 degrees yaw and then send 0x000B calibration request to terminate.
0x000B	**Terminate magnetic alignment.** The unit will send a CC response containing the hard-iron and soft-iron values. To accept the parameters, store them using the write magnetic calibration command.
0x000C	**Begin magnetic calibration with automatic termination.** Rotate the unit through 380 degrees in yaw. The unit will send a CC response containing the hard-iron and soft-iron values upon completion of the turn. To accept the parameters, store them using the write magnetic calibration command.
0x000E	**Write magnetic calibration.** The unit will write the parameters to EEPROM and then send a calibration response.

7.2.7 Calibrate Acknowledgement Response

Calibrate ('WC' = 0x5743)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5743	0x02	*<WC payload>*	*<CRC (U2)>*

The unit will send this packet in response to a calibrate request if the procedure can be performed or initiated.

WC Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	calibrationRequest	U2	-	-	The requested calibration task

7.2.8 Calibration Completed Parameters Response

Calibrate Completed ('CC' = 0x4343)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4343	0x08	*<CC payload>*	*<CRC (U2)>*

The unit will send this packet after a calibration has been completed. Currently, there is only one message of this type sent after a magnetic calibration has been completed (with or without automatic termination) and the parameters have been calculated. Thus, the calibrationRequest field will be 0x000B or 0x000C.

CC Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	calibrationRequest	U2	-	-	The requested calibration task
2	xHardIron	I2	2/2^16	G	The x hard iron bias
4	yHardIron	I2	2/2^16	G	The y hard iron bias
6	softIronScaleRatio	U2	2/2^16	-	The scaling ratio between the x and y axis

7.2.9 Error Response

Error Response (ASCII NAK, NAK = 0x1515)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x1515	0x02	*<NAK payload>*	*<CRC (U2)>*

The unit will send this packet in place of a normal response to a *failedInputPacketType* request if it could not be completed successfully.

NAK Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	failedInputPacketType	U2	-	-	the failed request

7.3 Output Packets (Polled)

The following packet formats are special informational packets which can be requested using the 'GP' command.

7.3.1 Identification Data Packet

Identification Data ('ID' = 0x4944)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4944	5+N	*<ID payload>*	*<CRC (U2)>*

This packet contains the unit *serialNumber* and *modelString*. The model string is terminated with 0x00. The model string contains the programmed versionString (8-bit Ascii values) followed by the firmware part number string delimited by a whitespace.

ID Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	serialNumber	U4	-	-	Unit serial number
4	modelString	SN	-	-	Unit Version String
4+N	0x00	U1	-	-	Zero Delimiter

7.3.2 Version Data Packet

Version Data ('VR' = 0x5652)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5652	5	*<VR payload>*	*<CRC (U2)>*

This packet contains firmware version information. *majorVersion* changes may introduce serious incompatibilities. *minorVersion* changes may add or modify functionality, but maintain backward compatibility with previous minor versions. *patch* level changes reflect bug fixes and internal modifications with little effect on the user. The build *stage* is one of the following: 0=release candidate, 1=development, 2=alpha, 3=beta. The *buildNumber* is incremented with each engineering firmware build. The *buildNumber* and *stage* for released firmware are both zero. The final beta candidate is v.w.x.3.y, which is then changed to v.w.x.0.1 to create the first release candidate. The last release candidate is v.w.x.0.z, which is then changed to v.w.x.0.0 for release.

VR Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	majorVersion	U1	-	-	Major firmware version
1	minorVersion	U1	-	-	Minor firmware version

2	patch	U1	-	-	Patch level
3	stage	-	-	-	Development Stage (0=release candidate, 1=development, 2=alpha, 3=beta)
4	buildNumber	U1	-	-	Build number

7.3.3 Test 0 (Detailed BIT and Status) Packet

Test ('T0' = 0x5430)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
03.3x5555	0x5430	0x1C	*<T0 payload>*	*<CRC (U2)>*

This packet contains detailed BIT and status information. The full BIT Status details are described in Section 9 of this manual.

T0 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	BITstatus	U2	-	-	Master BIT and Status Field
2	hardwareBIT	U2	-	-	Hardware BIT Field
4	hardwarePowerBIT	U2	-	-	Hardware Power BIT Field
6	hardwareEnvironmentalBIT	U2	-	-	Hardware Environmental BIT Field
8	comBIT	U2	-	-	communication BIT Field
10	comSerialABIT	U2	-	-	Communication Serial A BIT Field
12	comSerialBBIT	U2	-	-	Communication Serial B BIT Field
14	softwareBIT	U2	-	-	Software BIT Field
16	softwareAlgorithmBIT	U2	-	-	Software Algorithm BIT Field
18	softwareDataBIT	U2	-	-	Software Data BIT Field
20	hardwareStatus	U2	-	-	Hardware Status Field
22	comStatus	U2	-	-	Communication Status Field
24	softwareStatus	U2	-	-	Software Status Field
26	sensorStatus	U2	-	-	Sensor Status Field

7.4 Output Packets (Polled or Continuous)

7.4.1 Scaled Sensor Data Packet 0

Scaled Sensor Data ('S0' = 0x5330)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5330	0x1E	*<S0 payload>*	*<CRC (U2)>*

This packet contains scaled sensor data. The scaled sensor data is fixed point, 2 bytes per sensor, MSB first, for 13 sensors in the following order: accels(x,y,z); gyros(x,y,z); mags(x,y,z); temps(x,y,z,board). Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100)°C

S0 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	xAccel	I2	20/2^16	g	X accelerometer
2	yAccel	I2	20/2^16	g	Y accelerometer
4	zAccel	I2	20/2^16	g	Z accelerometer
6	xRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate
8	yRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate
10	zRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate
12	xMag	I2	2/2^16	Gauss	X magnetometer
14	yMag	I2	2/2^16	Gauss	Y magnetometer
16	zMag	I2	2/2^16	Gauss	Z magnetometer
18	xRateTemp	I2	200/2^16	deg. C	X rate temperature
20	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
22	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
24	boardTemp	I2	200/2^16	deg. C	CPU board temperature
26	GPSITOW	U2	*truncated*	ms	GPS ITOW (lower 2 bytes)
28	BITstatus	U2	-	-	Master BIT and Status

7.4.2 Scaled Sensor Data Packet 1 (Default IMU Data)

Scaled Sensor Data ('S1' = 0x5331)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5331	0x18	*<S1 payload>*	*<CRC (U2)>*

This packet contains scaled sensor data. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10)g

Temperature: scaled to a range of [-100, +100)°C

S1 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	xAccel	I2	20/2^16	g	X accelerometer
2	yAccel	I2	20/2^16	g	Y accelerometer
4	zAccel	I2	20/2^16	g	Z accelerometer
6	xRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate
8	yRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate
10	zRate	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate

12	xRateTemp	I2	200/2^16	deg. C	X rate temperature
14	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
16	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
18	boardTemp	I2	200/2^16	deg. C	CPU board temperature
20	Counter	U2	-	packets	Output packet counter
22	BITstatus	U2	-	-	Master BIT and Status

7.4.3 Scaled Sensor Data Packet 2 (Delta-Theta, Delta-V)

Scaled Sensor Data (‘S2’ = 0x5332)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5332	0x1C	*<S2 payload>*	*<CRC (U2)>*

This packet contains scaled sensor data in the traditional delta-theta and delta-velocity format with integration time equivalent to the packet rate. Changes in body axis angles and velocities are accumulated during the interval between successive packets as determined by the packet rate. Polled requests for this packet will produce values accumulated since the last poll request, and thus, are subject to overflow (data type wrap around).

Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Delta Angle: scaled to range of 3.5* [-pi,+pi) Δ radians or [-630, +630) Δ degrees.

Delta Velocity: scaled to a range of [-100,+100) Δ m/s.

S2 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	xDeltaVel	I4	200/2^32	Δ m/s	X delta velocity
4	yDeltaVel	I4	200/2^32	Δ m/s	Y delta velocity
8	zDeltaVel	I4	200/2^32	Δ m/s	Z delta velocity
12	xDeltaAngle	I4	7*pi/2^32 [1260°/2^32]	Δ rad [Δ °]	X delta angle
16	yDeltaAngle	I4	7*pi/2^32 [1260°/2^32]	Δ rad [Δ °]	Y delta angle
20	zDeltaAngle	I4	7*pi/2^32 [1260°/2^32]	Δ rad [Δ °]	Z delta angle
24	Counter	U2	-	packets	Output packet counter
26	BITstatus	U2	-	-	Master BIT and Status

7.4.4 Angle Data Packet 0

Angle Data (‘A0’ = 0x4130)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4130	0x1E	*<A0 payload>*	*<CRC (U2)>*

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2’s complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10)g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100) °C

A0 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleMag	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (magnetic north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular RateCorrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular RateCorrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular RateCorrected
12	xAccelCorrected	I2	20/2^16	g	X AccelCorrected
14	yAccelCorrected	I2	20/2^16	g	Y AccelCorrected
16	zAccelCorrected	I2	20/2^16	g	Z AccelCorrected
18	xMag	I2	2/2^16	Gauss	X magnetometer
20	yMag	I2	2/2^16	Gauss	Y magnetometer
22	zMag	I2	2/2^16	Gauss	Z magnetometer
24	xRateTemp	I2	200/2^16	deg C	X rate temperature
26	GPSITOW	U2	*truncated*	ms	GPS ITOW (lower 2 bytes)
28	BITstatus	U2	-	-	Master BIT and Status

7.4.5 Angle Data Packet 1 (Default AHRS Data)

Angle Data ('A1' = 0x4131)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4131	0x20	*<A1 payload>*	*<CRC (U2)>*

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100) °C

A1 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle

4	yawAngleMag	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (magnetic north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate Corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate Corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate Corrected
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	xMag	I2	2/2^16	Gauss	X magnetometer
20	yMag	I2	2/2^16	Gauss	Y magnetometer
22	zMag	I2	2/2^16	Gauss	Z magnetometer
24	xRateTemp	I2	*200/2^16*	Deg C	X rate temperature
26	timeITOW	U4	1	ms	DMU ITOW (sync to GPS)
30	BITstatus	U2	-	-	Master BIT and Status

7.4.6 Angle Data Packet 2 (Default VG Data)

Angle Data ('A2' = 0x4132)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4132	0x1E	*<A2 payload>*	*<CRC (U2)>*

This packet contains angle data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100) °C

A2 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (free)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	xAccel	I2	20/2^16	g	X accelerometer

14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	xRateTemp	I2	200/2^16	deg. C	X rate temperature
20	yRateTemp	I2	200/2^16	deg. C	Y rate temperature
22	zRateTemp	I2	200/2^16	deg. C	Z rate temperature
24	timeITOW	U4	1	ms	DMU ITOW (sync to GPS)
28	BITstatus	U2	-	-	Master BIT and Status

7.4.7 Nav Data Packet 0

Nav Data ('N0' = 0x4E30)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4E30	0x20	*<N0 payload>*	*<CRC (U2)>*

This packet contains navigation data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100) °C

Velocities are scaled to a range of [-256,256) m/s

Altitude is scaled to a range of [-100,16284) m using a shifted 2's complement representation.

Longitude and latitude are scaled to a range of [-pi,pi) or [-180 deg to +180 deg).

N0 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (true north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	nVel	I2	512/2^16	m/s	North velocity
14	eVel	I2	512/2^16	m/s	East velocity
16	dVel	I2	512/2^16	m/s	Down velocity
18	longitudeGPS	I4	2*pi/2^32 [360°/2^32]	Radians [°]	GPS Longitude
22	latitudeGPS	I4	2*pi/2^32 [360°/2^32]	Radians [°]	GPS Latitude
26	altitudeGPS	I2*	2^14/2^16	m	GPS altitude [-100,16284)

28	GPSITOW	U2	*truncated*	ms	GPS ITOW (lower 2 bytes)
30	BITstatus	U2	-	-	Master BIT and Status

7.4.8 Nav Data Packet 1 (Default NAV)

Nav Data ('N1' = 0x4E31)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4E31	0x2A	*<N1 payload>*	*<CRC (U2)>*

This packet contains navigation data and selected sensor data scaled in most cases to a signed 2^16 2's complement number. Data involving angular measurements include the factor pi in the scaling and can be interpreted in either radians or degrees.

Angles: scaled to a range of [-pi,+pi) or [-180 deg to +180 deg).

Angular rates: scaled to range of 3.5* [-pi,+pi) or [-630 deg/sec to +630 deg/sec)

Accelerometers: scaled to a range of [-10,+10) g

Magnetometers: scaled to a range of [-1,+1) Gauss

Temperature: scaled to a range of [-100, +100) °C

Velocities are scaled to a range of [-256,256) m/s

Altitude is scaled to a range of [-100,16284) m using a shifted 2's complement representation.

Longitude and latitude are scaled to a range of [-pi,pi) or [-180 deg to +180 deg).

N1 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (true north)
6	xRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	X angular rate corrected
8	yRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Y angular rate corrected
10	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
12	xAccel	I2	20/2^16	g	X accelerometer
14	yAccel	I2	20/2^16	g	Y accelerometer
16	zAccel	I2	20/2^16	g	Z accelerometer
18	nVel	I2	512/2^16	m/s	North velocity
20	eVel	I2	512/2^16	m/s	East velocity
22	dVel	I2	512/2^16	m/s	Down velocity
24	longitudeGPS	I4	2*pi/2^32 [360°/2^32]	Radians [°]	GPS Longitude
28	latitudeGPS	I4	2*pi/2^32 [360°/2^32]	Radians [°]	GPS Latitude
32	altitudeGPS	I2*	2^14/2^16	m	GPS altitude [-100,16284)
34	xRateTemp	I2	200/2^16	deg C	X rate sensor temperature
36	timeITOW	U4	1	ms	DMU ITOW (sync to GPS)

40	BITstatus	U2	-	-	Master BIT and Status

7.4.9 Angle Data Packet B1 (Custom VG Data)

Angle Data ('B1' = 0x4231)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4231	0x12	*<B1 payload>*	*<CRC (U2)>*

This packet contains selected angle and sensor data.

B1 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	yawAngleTrue	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Yaw angle (free)
6	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
8	xAccel	I2	20/2^16	g	X accelerometer
10	yAccel	I2	20/2^16	g	Y accelerometer
12	timeITOW	U4	1	ms	DMU ITOW (sync to GPS)
16	BITstatus	U2	-	-	Master BIT and Status

7.4.10 Angle Data Packet B2 (Custom VG Data)

Angle Data ('B2' = 0x4232)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4232	0x0A	*<B2 payload>*	*<CRC (U2)>*

This packet contains selected angle and sensor data.

B2 Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	rollAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Roll angle
2	pitchAngle	I2	2*pi/2^16 [360°/2^16]	Radians [°]	Pitch angle
4	zRateCorrected	I2	7*pi/2^16 [1260°/2^16]	rad/s [°/sec]	Z angular rate corrected
6	xAccel	I2	20/2^16	g	X accelerometer
8	timeITOWtruncated	U2	1	ms	DMU ITOW (sync to GPS) truncated to two bytes

8 Advanced Commands

The advanced commands allow users to programmatically change the 440 Series settings. This section of the manual documents all of the settings and options contained under the Unit Configuration tab within NAV-VIEW 2.X. Using these advanced commands, a user's system can change or modify the settings without the need for NAV-VIEW 2.X.

8.1 Configuration Fields

Configuration fields determine various behaviors of the unit that can be modified by the user. These include settings like baud rate, packet output rate and type, algorithm type, etc. These fields are stored in EEPROM and loaded on power up. These fields can be read from the EEPROM using the 'RF' command. These fields can be written to the EEPROM affecting the default power up behavior using the 'WF' command. The current value of these fields (which may be different from the value stored in the EEPROM) can also be accessed using the 'GF' command. All of these fields can also be modified immediately for the duration of the current power cycle using the 'SF' command. The unit will always power up in the configuration stored in the EEPROM. Configuration fields can only be set or written with valid data from the table below.

configuration fields	***field ID***	***Valid Values***	***description***
Packet rate divider	0x0001	0,1,2,5,10, 20, 25, 50	quiet, 100Hz, 50Hz, 25Hz, 20Hz, 10Hz, 5Hz, 2Hz
Unit BAUD rate	0x0002	0,1,2,3	9600, 19200, 38400, 57600
Continuous packet type	0x0003	Any output packet type	Not all output packets available for all products. See detailed product descriptions.
Analog Filter Clock 1	0x0004	1338-26785 [40-2 Hz]	Sets low pass cutoff for rate sensors. Cutoff Freq. = 15e5/(28*(value+1))
Analog Filter Clock 2	0x0005	1338-26785 [40-2 Hz]	Sets low pass cutoff for z accel. Cutoff Freq. = 15e5/(28*(value+1))
Analog Filter Clock 3	0x0006	1338-26785 [40-2 Hz]	Sets low pass cutoff for x and y accel. Cutoff Freq. = 15e5/(28*(value+1))
Orientation	0x0007	See below	Determine forward, rightward, and downward facing sides
User Behavior Switches	0x0008	Any	Free Integrate, Use Mags, Use GPS, Stationary Yaw Lock, …
X Hard Iron Bias	0x0009	Any	I2 scaled from [-1,1)
Y Hard Iron Bias	0x000A	Any	I2 scaled from [-1,1)
Soft Iron Scale Ratio	0x000B	Any	U2 scaled from [0,2)
Heading Track Offset	0x000C	Any	Heading-Track Offset to use in NAV filter track update mode.

Note: BAUD rate SF has immediate affect. Some output data may be lost. Response will be received at new BAUD rate.

8.2 Continuous Packet Type Field

This is the packet type that is being continually output. The supported packet depends on the model number. Please refer to Section 6.4 for a complete list of the available packet types.

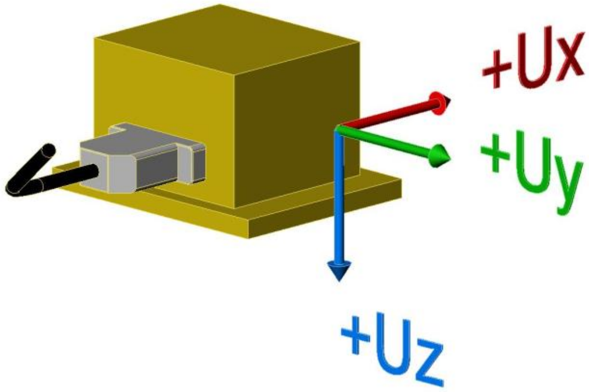
8.3 Analog Filter Clocks 1,2,3

These three fields set hardware low pass filter cutoff frequencies. Each sensor listed is defined in the default factory orientation. Users must consider any additional rotation to their intended orientation.

Filter Clock	**Sensor**
analogFilterClock1	Ux,Uz Accel
analogFilterClock2	Uy Accel
analogFilterClock3	Ux,Uy,Uz rate

8.4 Orientation Field

This field defines the rotation from the factory to user axis sets. This rotation is relative to the default factory orientation (connector aft, baseplate down). The default factory axis set is (Ux,Uy,Uz) defined by the connector pointing in the –Ux direction and the baseplate pointing in the +Uz direction. The user axis set is (X,Y,Z) as defined by this field. A depiction of the factory axis set is shown below:



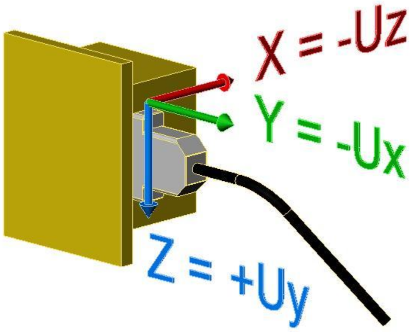
Description	*Bits*	*Meaning*
X Axis Sign	0	0 = positive, 1 = negative
X Axis	1:2	0 = Ux, 1 = Uy, 2 = Uz, 3 = N/A
Y Axis Sign	3	0 = positive, 1 = negative
Y Axis	4:5	0 = Uy, 1 = Uz, 2 = Ux, 3 = N/A
Z Axis Sign	6	0 = positive, 1 = negative
Z Axis	7:8	0 = Uz, 1 = Ux, 2 = Uy, 3 = N/A
Reserved	9:15	N/A

There are 24 possible orientation configurations. Setting/Writing the field to anything else generates a NAK and has no effect.

Orientation Field Value	*X Axis*	*Y Axis*	*Z Axis*
0x0000	+Ux	+Uy	+Uz
0x0009	-Ux	-Uy	+Uz
0x0023	-Uy	+Ux	+Uz
0x002A	+Uy	-Ux	+Uz
0x0041	-Ux	+Uy	-Uz
0x0048	+Ux	-Uy	-Uz
0x0062	+Uy	+Ux	-Uz
0x006B	-Uy	-Ux	-Uz
0x0085	-Uz	+Uy	+Ux
0x008C	+Uz	-Uy	+Ux
0x0092	+Uy	+Uz	+Ux
0x009B	-Uy	-Uz	+Ux
0x00C4	+Uz	+Uy	-Ux
0x00CD	-Uz	-Uy	-Ux

0x00D3	-Uy	+Uz	-Ux
0x00DA	+Uy	-Uz	-Ux
0x0111	-Ux	+Uz	+Uy
0x0118	+Ux	-Uz	+Uy
0x0124	+Uz	+Ux	+Uy
0x012D	-Uz	-Ux	+Uy
0x0150	+Ux	+Uz	-Uy
0x0159	-Ux	-Uz	-Uy
0x0165	-Uz	+Ux	-Uy
0x016C	+Uz	-Ux	-Uy

An example of orientation field value 0x12D is shown below.



8.5 User Behavior Switches

This field allows on the fly user interaction with behavioral aspects of the algorithm.

Description	***Bits***	***Meaning***
Free Integrate	0	0 = use feedback to stabilize the algorithm, 1 = 6DOF inertial integration without stabilized feedback
Use Mags	1	0 = Do not use mags to stabilize heading (heading will run open loop or be stabilized by GPS track), 1 = Use mags to stabilize heading
Use GPS	2	0 = Do not use GPS to stabilize the system, 1 = Use GPS when available
Stationary Yaw Lock	3	0 = Do not lock yaw when GPS speed is near zero (<0.75 m/s), 1 = Lock yaw when GPS speed is near zero

Restart on Over-range	4	0 = Do not restart the system after a sensor over-range, 1 = restart the system after a sensor over-range
Dynamic Motion	5	0=vehicle is static, force high gain corrections, 1= vehicle is dynamic, use nominal corrections
Reserved	6:15	N/A

8.6 Hard and Soft Iron Values

These fields allow access to hard iron bias and soft iron scale ratio values for magnetometer alignment:

Field Name	***Field ID***	***Format***	***Scaling***	***Units***
X Hard Iron Bias	0x0009	I2	2/2^16	Gauss
Y Hard Iron Bias	0x000A	I2	2/2^16	Gauss
Soft Iron Scale Ratio	0x000B	U2	2/2^16	-

The hard iron bias values are scaled from [-1,1] Gauss. These values are subtracted from the tangent plane magnetometer vector before heading is calculated. The soft iron scale ratio is scaled from [0,2) and is multiplied by the tangent plane x magnetometer value before heading is calculated.

8.7 Heading Track Offset

This field is used to set the offset between vehicle heading and vehicle track to be used by the navigation mode filter when no magnetometer heading measurements are available.

Field Name	***Field ID***	***Format***	***Scaling***	***Units***
Heading Track Offset	0x000C	I2	2*pi/2^16 [360°/2^16]	Radians (heading-track) [°]

8.8 Commands to Program Configuration

8.8.1 Write Fields Command

Write Fields ('WF' = 0x5746)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5746	*1+numFields*4*	*<WF payload>*	*<CRC (U2)>*

This command allows the user to write default power-up configuration fields to the EEPROM. Writing the default configuration will not take affect until the unit is power cycled. *NumFields* is the number of words to be written. The *field0, field1, etc.* are the field IDs that will be written with the *field0Data, field1Data, etc.*, respectively. The unit will not write to calibration or algorithm fields. If at least one field is successfully written, the unit will respond with a write fields response containing the field IDs of the successfully written fields. If any field is unable to be written, the unit will respond with an error response. Note that both a write fields and an error response may be received as a result of a write fields command. Attempts to write a field with an invalid value is one way to generate an error response. A table of field IDs and valid field values is available in Section 8.1.

WF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to write
1	field0	U2	-	-	The first field ID to write
3	field0Data	U2	-	-	The first field ID's data to write
5	field1	U2	-	-	The second field ID to write
7	field1Data	U2	-	-	The second field ID's data

…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID to write
numFields*4 -1	field…Data	U2	-	-	The last field ID's data to write

Write Fields Response

Write Fields ('WF' = 0x5746)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5746	*1+numFields*2*	*<WF payload>*	*<CRC (U2)>*

The unit will send this packet in response to a write fields command if the command has completed without errors.

WF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields written
1	field0	U2	-	-	The first field ID written
3	field1	U2	-	-	The second field ID written
…	…	U2	-	-	More field IDs written
numFields*2 – 1	Field…	U2	-	-	The last field ID written

8.8.2 Set Fields Command

Set Fields ('SF' = 0x5346)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5346	*1+numFields*4*	*<SF payload>*	*<CRC (U2)>*

This command allows the user to set the unit's current configuration (SF) fields immediately which will then be lost on power down. *NumFields* is the number of words to be set. The *field0, field1, etc.* are the field IDs that will be written with the *field0Data, field1Data, etc.*, respectively. This command can be used to set configuration fields. The unit will not set calibration or algorithm fields. If at least one field is successfully set, the unit will respond with a set fields response containing the field IDs of the successfully set fields. If any field is unable to be set, the unit will respond with an error response. Note that both a set fields and an error response may be received as a result of one set fields command. Attempts to set a field with an invalid value is one way to generate an error response. A table of field IDs and valid field values is available in Section 8.1.

SF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to set
1	field0	U2	-	-	The first field ID to set
3	field0Data	U2	-	-	The first field ID's data to set
5	field1	U2	-	-	The second field ID to set
7	field1Data	U2	-	-	The second field ID's data to set
…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID to set
numFields*4 -1	field…Data	U2	-	-	The last field ID's data to set

Write Fields Response

Write Fields ('WF' = 0x5746)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5746	*1+numFields*2*	*<WF payload>*	*<CRC (U2)>*

The unit will send this packet in response to a write fields command if the command has completed without errors.

WF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields written
1	field0	U2	-	-	The first field ID written
3	field1	U2	-	-	The second field ID written
…	…	U2	-	-	More field IDs written
numFields*2 – 1	Field…	U2	-	-	The last field ID written

8.9 Read Fields Command

Read Fields ('RF' = 0x5246)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5246	*1+numFields*2*	*<RF payload>*	*<CRC (U2)>*

This command allows the user to read the default power-up configuration fields from the EEPROM. *NumFields* is the number of fields to read. The *field0, field1, etc.* are the field IDs to read. RF may be used to read configuration and calibration fields from the EEPROM. If at least one field is successfully read, the unit will respond with a read fields response containing the field IDs and data from the successfully read fields. If any field is unable to be read, the unit will respond with an error response. Note that both a read fields and an error response may be received as a result of a read fields command.

RF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to read
1	field0	U2	-	-	The first field ID to read
3	field1	U2	-	-	The second field ID to read
…	…	U2	-	-	More field IDs to read
numFields*2 - 1	Field…	U2	-	-	The last field ID to read

8.10 Read Fields Response

Read Fields ('RF' = 0x5246)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x5246	*1+numFields*4*	*<RF payload>*	*<CRC (U2)>*

The unit will send this packet in response to a read fields request if the command has completed without errors.

RF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields read
1	field0	U2	-	-	The first field ID read
3	field0Data	U2	-	-	The first field ID's data read
5	field1	U2	-	-	The second field ID read
7	field1Data	U2	-	-	The second field ID's data read
…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID read
numFields*4 -1	field…Data	U2	-	-	The last field ID's data read

8.11 Get Fields Command

Get Fields ('GF' = 0x4746)

Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4746	*1+numFields*2*	*<GF Data>*	*<CRC (U2)>*

This command allows the user to get the unit's current configuration fields. *NumFields* is the number of fields to get. The *field0, field1, etc.* are the field IDs to get. GF may be used to get configuration, calibration, and algorithm fields from RAM. Multiple algorithm fields will not necessarily be from the same algorithm iteration. If at least one field is successfully collected, the unit will respond with a get fields response with data containing the field IDs of the successfully received fields. If any field is unable to be received, the unit will respond with an error response. Note that both a get fields and an error response may be received as the result of a get fields command.

GF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields to get
1	field0	U2	-	-	The first field ID to get
3	field1	U2	-	-	The second field ID to get
…	…	U2	-	-	More field IDs to get
numFields*2 – 1	Field…	U2	-	-	The last field ID to get

8.12 Get Fields Response

Get Fields ('GF' = 0x4746)				
Preamble	*Packet Type*	*Length*	*Payload*	*Termination*
0x5555	0x4746	*1+numFields*4*	*<GF Data>*	*<CRC (U2)>*

The unit will send this packet in response to a get fields request if the command has completed without errors.

GF Payload Contents					
Byte Offset	*Name*	*Format*	*Scaling*	*Units*	*Description*
0	numFields	U1	-	-	The number of fields retrieved
1	field0	U2	-	-	The first field ID retrieved
3	field0Data	U2	-	-	The first field ID's data retrieved
5	field1	U2	-	-	The second field ID retrieved
7	field1Data	U2	-	-	The second field ID's data
…	…	U2	-	-	…
numFields*4 -3	field…	U2	-	-	The last field ID retrieved
numFields*4 -1	field…Data	U2	-	-	The last field ID's data retrieved

9 Advanced BIT

9.1 Built In Test (BIT) and Status Fields

Internal health and status are monitored and communicated in both hardware and software. The ultimate indication of a fatal problem is a hardware BIT signal on the user connector which is mirrored in the software BIT field as the masterFail flag. This flag is thrown as a result of a number of instantly fatal conditions (known as a "hard" failure) or a persistent serious problem (known as a "soft" failure). Soft errors are those which must be triggered multiple times within a specified time window to be considered fatal. Soft errors are managed using a digital high-pass error counter with a trigger threshold.

The masterStatus flag is a configurable indication as determined by the user. This flag is asserted as a result of any asserted alert signals which the user has enabled.

The hierarchy of BIT and Status *fields* and signals is depicted here:

- *BITstatus Field*
  - masterFail
    - hardwareError
      - *hardwareBIT Field*
        - powerError
          - *hardwarePowerBIT Field*
            - inpPower
            - inpCurrent
            - inpVoltage
            - fiveVolt
            - threeVolt
            - twoVolt
            - twoFiveRef
            - sixVolt
            - grdRef
        - environmentalError
          - *hardwareEnvironmentalBIT Field*
            - pcbTemp
    - comError
      - *comBIT Field*
        - serialAError
          - *comSerialABIT Field*
            - transmitBufferOverflow
            - receiveBufferOverflow
            - framingError
            - breakDetect
            - parityError
        - serialBError
          - *comSerialBBIT Field*
            - transmitBufferOverflow
            - receiveBufferOverflow
            - framingError
            - breakDetect
            - parityError
    - softwareError
      - *softwareBIT Field*
        - algorithmError
          - *softwareAlgorithmBIT Field*
            - initialization
            - overRange
            - missedIntegrationStep
        - dataError
          - *softwareDataBIT Field*

        - calibrationCRCError
        - magAlignOutOfBounds
- ➢ masterStatus
    - hardwareStatus
        - *hardwareStatus Field*
            - ♦ unlocked1PPS (enabled by default on NAV)
            - ♦ unlockedInternalGPS (enabled by default on NAV)
            - ♦ noDGPS
            - ♦ unlockedEEPROM
    - comStatus
        - *comStatus Field*
            - ♦ noExternalGPS (enabled by default on VG and AHRS)
    - softwareStatus
        - *softwareStatus Field*
            - ♦ algorithmInitialization (enabled by default)
            - ♦ highGain (enabled by default)
            - ♦ attitudeOnlyAlgorithm
            - ♦ turnSwitch
    - sensorStatus
        - *sensorStatus Field*
            - ♦ overRange (enabled by default)

9.2 Master BIT and Status (BITstatus) Field

The BITstatus field is the global indication of health and status of the 440 Series product. The LSB contains BIT information and the MSB contains status information.

There are four intermediate signals that are used to determine when masterFail and the hardware BIT signal are asserted. These signals are controlled by various systems checks in software that are classified into three categories: hardware, communication, and software. Instantaneous soft failures in each of these four categories will trigger these intermediate signals, but will not trigger the masterFail until the persistency conditions are met.

There are four intermediate signals that are used to determine when the masterStatus flag is asserted: hardwareStatus, sensorStatus, comStatus, and softwareStatus. masterStatus is the logical OR of these intermediate signals. Each of these intermediate signals has a separate field with individual indication flags. Each of these indication flags can be enabled or disabled by the user. Any enabled indication flag will trigger the associated intermediate signal and masterStatus flag.

BITstatus Field	***Bits***	***Meaning***	***Category***
masterFail	0	0 = normal, 1 = fatal error has occurred	BIT
HardwareError	1	0 = normal, 1= internal hardware error	BIT
comError	2	0 = normal, 1 = communication error	BIT
softwareError	3	0 = normal, 1 = internal software error	BIT
Reserved	4:7	N/A	
masterStatus	8	0 = nominal, 1 = hardware, sensor, com, or software alert	Status
hardwareStatus	9	0 = nominal, 1 = programmable alert	Status
comStatus	10	0 = nominal, 1 = programmable alert	Status
softwareStatus	11	0 = nominal, 1 = programmable alert	Status
sensorStatus	12	0 = nominal, 1 = programmable alert	Status
Reserved	13:15	N/A	

9.3 hardwareBIT Field

The hardwareBIT field contains flags that indicate various types of internal hardware errors. Each of these types has an associated message with low level error signals. The hardwareError flag in the BITstatus field is the bit-wise OR of this hardwareBIT field.

hardwareBIT Field	*Bits*	*Meaning*	*Category*
powerError	0	0 = normal, 1 = error	Soft
environmentalError	1	0 = normal, 1 = error	Soft
reserved	2:15	N/A	

9.4 hardwarePowerBIT Field

The hardwarePowerBIT field contains flags that indicate low level power system errors. The powerError flag in the hardwareBIT field is the bit-wise OR of this hardwarePowerBIT field.

hardwarePowerBIT Field	*Bits*	*Meaning*	*Category*
inpPower	0	0 = normal, 1 = out of bounds	Soft
inpCurrent	1	0 = normal, 1 = out of bounds	Soft
inpVoltage	2	0 = normal, 1 = out of bounds	Soft
fiveVolt	3	0 = normal, 1 = out of bounds	Soft
threeVolt	4	0 = normal, 1 = out of bounds	Soft
twoVolt	5	0 = normal, 1 = out of bounds	Soft
twoFiveRef	6	0 = normal, 1 = out of bounds	Soft
sixVolt	7	0 = normal, 1 = out of bounds	Soft
grdRef	8	0 = normal, 1 = out of bounds	Soft
Reserved	9:15	N/A	

9.5 hardwareEnvironmentalBIT Field

The hardwareEnvironmentalBIT field contains flags that indicate low level hardware environmental errors. The environmentalError flag in the hardwareBIT field is the bit-wise OR of this hardwareEnvironmentalBIT field.

hardwareEnvironmentalBIT Field	*Bits*	*Meaning*	*Category*
pcbTemp	0	0 = normal, 1 = out of bounds	Soft
Reserved	9:15	N/A	

9.6 comBIT Field

The comBIT field contains flags that indicate communication errors with external devices. Each external device has an associated message with low level error signals. The comError flag in the BITstatus field is the bit-wise OR of this comBIT field.

comBIT Field	*Bits*	*Meaning*	*Category*
serialAError	0	0 = normal, 1 = error	Soft
serialBError	1	0 = normal, 1 = error	Soft
Reserved	2:15	N/A	

9.7 comSerialABIT Field

The comSerialABIT field contains flags that indicate low level errors with external serial port A (the user serial port). The serialAError flag in the comBIT field is the bit-wise OR of this comSerialABIT field.

comSerialABIT Field	*Bits*	*Meaning*	*Category*
transmitBufferOverflow	0	0 = normal, 1 = overflow	Soft
receiveBufferOverflow	1	0 = normal, 1 = overflow	Soft
framingError	2	0 = normal, 1 = error	Soft
breakDetect	3	0 = normal, 1 = error	Soft
parityError	4	0 = normal, 1 = error	Soft
Reserved	5:15	N/A	

9.8 comSerialBBIT Field

The comSerialBBIT field contains flags that indicate low level errors with external serial port B (the aiding serial port). The serialBError flag in the comBIT field is the bit-wise OR of this comSerialBBIT field.

comSerialBBIT Field	*Bits*	*Meaning*	*Category*
transmitBufferOverflow	0	0 = normal, 1 = overflow	Soft
receiveBufferOverflow	1	0 = normal, 1 = overflow	Soft
framingError	2	0 = normal, 1 = error	Soft
breakDetect	3	0 = normal, 1 = error	Soft
parityError	4	0 = normal, 1 = error	Soft
Reserved	5:15	N/A	

9.9 softwareBIT Field

The softwareBIT field contains flags that indicate various types of software errors. Each type has an associated message with low level error signals. The softwareError flag in the BITstatus field is the bit-wise OR of this softwareBIT field.

softwareBIT Field	*Bits*	*Meaning*	*Category*
algorithmError	0	0 = normal, 1 = error	Soft
dataError	1	0 = normal, 1 = error	Soft
Reserved	2:15	N/A	

9.10 softwareAlgorithmBIT Field

The softwareAlgorithmBIT field contains flags that indicate low level software algorithm errors. The algorithmError flag in the softwareBIT field is the bit-wise OR of this softwareAlgorithmBIT field.

SoftwareAlgorithmBIT Field	*Bits*	*Meaning*	*Category*
initialization	0	0 = normal, 1 = error during algorithm initialization	Hard
overRange	1	0 = normal, 1 = fatal sensor over-range	Hard
missedNavigationStep	2	0 = normal, 1 = fatal hard deadline missed for navigation	Hard
Reserved	3:15	N/A	

9.11 softwareDataBIT Field

The softwareDataBIT field contains flags that indicate low level software data errors. The dataError flag in the softwareBIT field is the bit-wise OR of this softwareDataBIT field.

SoftwareDataBIT Field	*Bits*	*Meaning*	*Category*
calibrationCRCError	0	0 = normal, 1 = incorrect CRC on calibration EEPROM data or data has been compromised by a WE command.	Hard
magAlignOutOfBounds	1	0 = normal, 1 = hard and soft iron parameters are out of bounds	Hard
Reserved	2:15	N/A	

9.12 hardwareStatus Field

The hardwareStatus field contains flags that indicate various internal hardware conditions and alerts that are not errors or problems. The hardwareStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the hardwareStatus field and the hardwareStatusEnable field. The hardwareStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

hardwareStatus Field	*Bits*	*Meaning*
unlocked1PPS	0	0 = not asserted, 1 = asserted
unlockedInternalGPS	1	0 = not asserted, 1 = asserted
noDGPS	2	0 = DGPS lock, 1 = no DGPS
unlockedEEPROM	3	0=locked, WE disabled, 1=unlocked, WE enabled
Reserved	4:15	N/A

9.13 comStatus Field

The comStatus field contains flags that indicate various external communication conditions and alerts that are not errors or problems. The comStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the comStatus field and the comStatusEnable field. The comStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

comStatus Field	*Bits*	*Meaning*
noExternalGPS	0	0 = external GPS data is being received, 1 = no external GPS data is available
Reserved	1:15	N/A

9.14 softwareStatus Field

The softwareStatus field contains flags that indicate various software conditions and alerts that are not errors or problems. The softwareStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the softwareStatus field and the softwareStatusEnable field. The softwareStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

softwareStatus Field	*Bits*	*Meaning*
algorithmInit	0	0 = normal, 1 = the algorithm is in initialization mode
highGain	1	0 = low gain mode, 1 high gain mode
attitudeOnlyAlgorithm	2	0 = navigation state tracking, 1 = attitude only state tracking
turnSwitch	3	0 = off, 1 = yaw rate greater than turnSwitch

		threshold
Reserved	4:15	N/A

9.15 sensorStatus Field

The sensorStatus field contains flags that indicate various internal sensor conditions and alerts that are not errors or problems. The sensorStatus flag in the BITstatus field is the bit-wise OR of the logical AND of the sensorStatus field and the sensorStatusEnable field. The sensorStatusEnable field is a bit mask that allows the user to select items of interest that will logically flow up to the masterStatus flag.

sensorStatus Field	*Bits*	*Meaning*
overRange	0	0 = not asserted, 1 = asserted
Reserved	1:15	N/A

9.16 Configuring the masterStatus.

The masterStatus byte and its associated programmable alerts are configured using the Read Field and Write Field command as described in Section 8, Advanced Commands. The Table below shows the definition of the bit mask for configuring the status signals.

configuration fields	*field ID*	*Valid Values*	*Description*
hardwareStatusEnable	0x0010	Any	Bit mask of enabled hardware status signals
comStatusEnable	0x0011	Any	Bit mask of enabled communication status signals
softwareStatusEnable	0x0012	Any	Bit mask of enabled software status signals
sensorStatusEnable	0x0013	Any	Bit mask of enabled sensor status signals

9.16.1 hardwareStatusEnable Field

This field is a bit mask of the hardwareStatus field (see BIT and status definitions). This field allows the user to determine which low level hardwareStatus field signals will flag the hardwareStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding hardwareStatus field signal, if asserted, will cause the hardwareStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.2 comStatusEnable Field

This field is a bit mask of the comStatus field (see BIT and status definitions). This field allows the user to determine which low level comStatus field signals will flag the comStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding comStatus field signal, if asserted, will cause the comStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.3 softwareStatusEnable Field

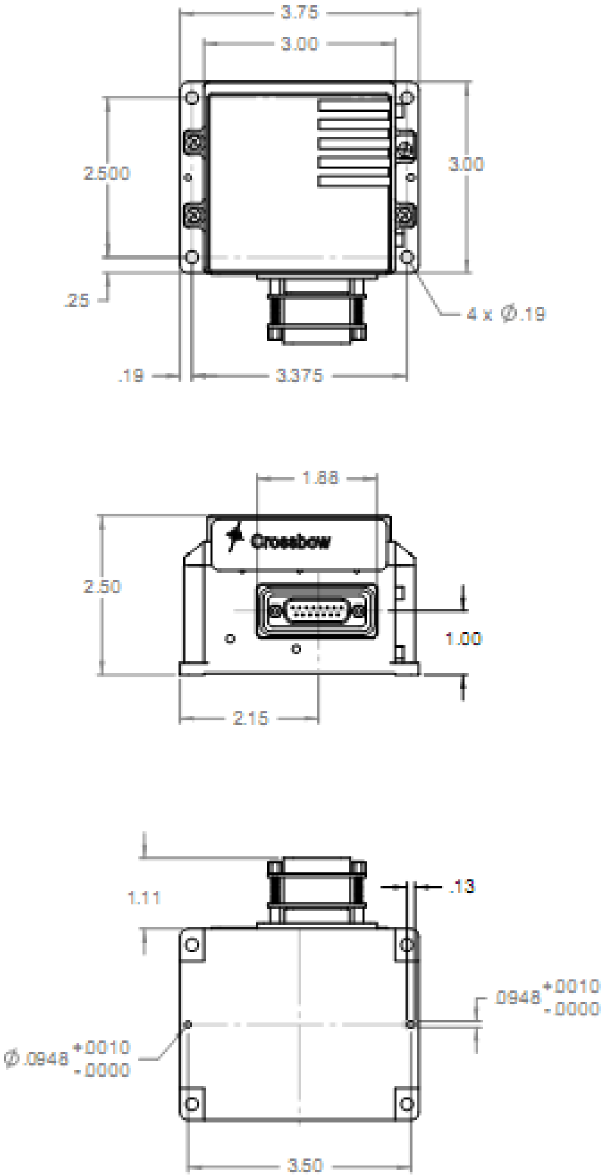
This field is a bit mask of the softwareStatus field (see BIT and status definitions). This field allows the user to determine which low level softwareStatus field signals will flag the softwareStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding softwareStatus field signal, if asserted, will cause the softwareStatus and masterStatus flags to be asserted in the BITstatus field.

9.16.4 sensorStatusEnable Field

This field is a bit mask of the sensorStatus field (see BIT and status definitions). This field allows the user to determine which low level sensorStatus field signals will flag the sensorStatus and masterStatus flags in the BITstatus field. Any asserted bits in this field imply that the corresponding sensorStatus field signal, if asserted, will cause the sensorStatus and masterStatus flags to be asserted in the BITstatus field.

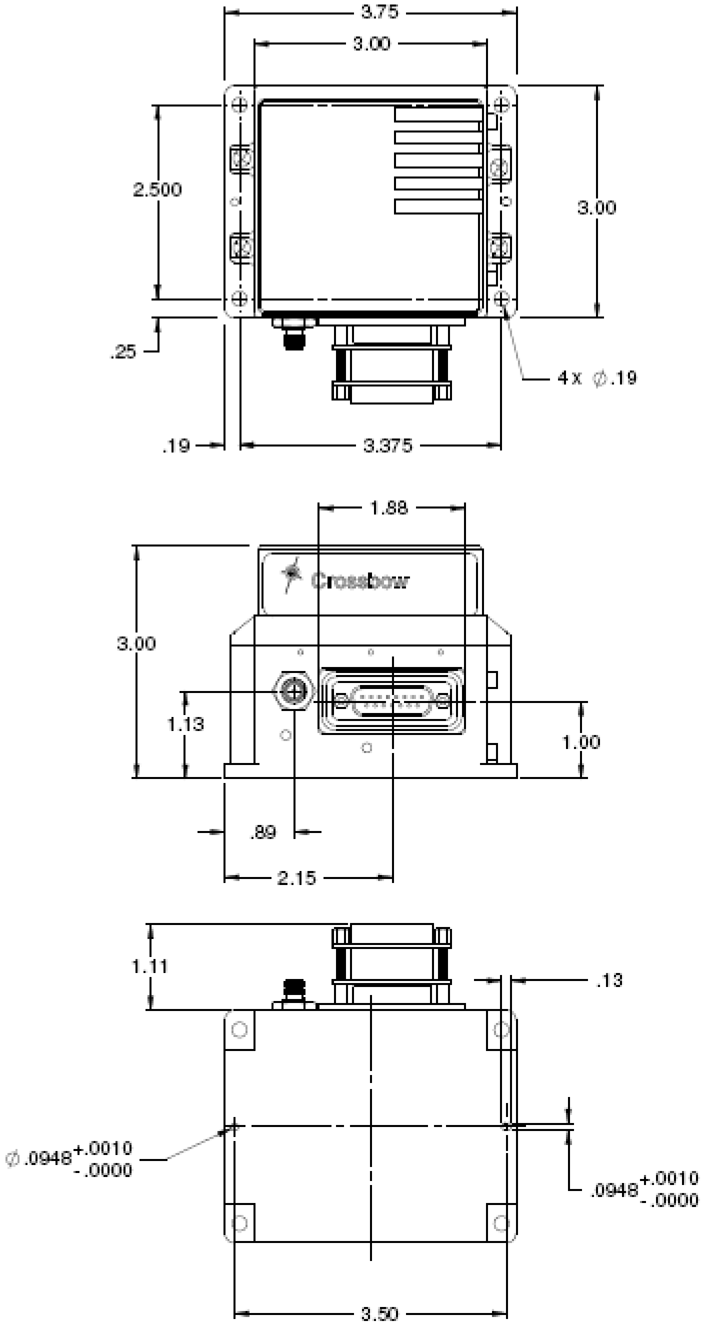
10 Appendix A. Mechanical Specifications

10.1 440 Series Outline Drawing (IMU, VG)



10.2

440 Series Outline Drawing (AHRS, NAV)



11 Appendix B. NMEA Message Format

The GPS receiver outputs data in NMEA-0183 format at 9600 Baud, 8 bits, no parity bit, and 1 stop bit. The GGA and RMC message packet formats are explained in this section.

11.1 GGA - GPS fix data

Time and position, together with GPS fixing related data (number of satellites in use, and the resulting HDOP, age of differential data if in use, etc.).

```
$GPGGA,hhmmss.ss,Latitude,N,Longitude,E,FS,NoSV,HDOP,msl,m,Altref,m,DiffAge,DiffStatio
n*cs<CR><LF>
```

Name	ASCII String		Description
	Format	Example	
$GPGGA	string	$GPGGA	Message ID: GGA protocol header
hhmmss.ss	hhmmss.sss	092725.00	UTC Time: Current time
Latitude	dddmm.mmmm	4717.11399	Latitude: Degrees + minutes
N	character	N	N/S Indicator: N=north or S=south
Longitude	dddmm.mmmm	00833.91590	Longitude: Degrees + minutes
E	character	E	E/W indicator: E=east or W=west
FS	1 digit	1	Position Fix Indicator (See Table below)
NoSV	numeric	8	Satellites Used: Range 0 to 12
HDOP	numeric	1.01	HDOP: Horizontal Dilution of Precision
msl	numeric	499.6	MSL Altitude (m)
m	character	M	Units: Meters (fixed field)
Altref	blank	48.0	Geoid Separation (m)
m	blank	M	Units: Meters (fixed field)
DiffAge	numeric		Age of Differential Corrections (sec): Blank (Null) fields when DGPS is not used
DiffStation	numeric	0	Diff. Reference Station ID
cs	hexadecimal	*5B	Checksum
<CR> <LF>			End of message

Fix Status	Description
0	No fix / Invalid
1	Standard GPS (2D/3D)
2	Differential GPS
6	Estimated (DR) Fix

12 Appendix C. Sample Packet-Parser Code

12.1 Overview

This appendix includes sample code written in ANSI C for parsing packets from data sent by the 440 Series Inertial Systems. This code can be used by a user application reading data directly from the 440 Series product, or perhaps from a log file.

The sample code contains the actual parser, but also several support functions for CRC calculation and circular queue access.:

- **process_xbow_packet** – for parsing out packets from a queue. Returns these fields in structure XBOW_PACKET (see below). Checks for CRC errors
- **calcCRC** – for calculating CRC on packets.
- **Initialize** - initialize the queue
- **AddQueue** - add item in front of queue
- **DeleteQueue** - return an item from the queue
- **peekWord** - for retrieving 2-bytes from the queue, without popping
- **peekByte** – for retrieving a byte from the queue without popping
- **Pop** - discard item(s) from queue
- **Size** – returns number of items in queue
- **Empty** – return 1 if queue is empty, 0 if not
- **Full** - return 1 if full, 0 if not full

The parser will parse the queue looking for packets. Once a packet is found and the CRC checks out, the packet's fields are placed in the XBOW_PACKET structure. The parser will then return to the caller. When no packets are found the parser will simply return to the caller with return value 0.

The XBOW_PACKET stucture is defined as follows:

```
typedef struct xbow_packet
{
    unsigned short packet_type;
    char           length;
    unsigned short crc;
    char           data[256];
} XBOW_PACKET;
```

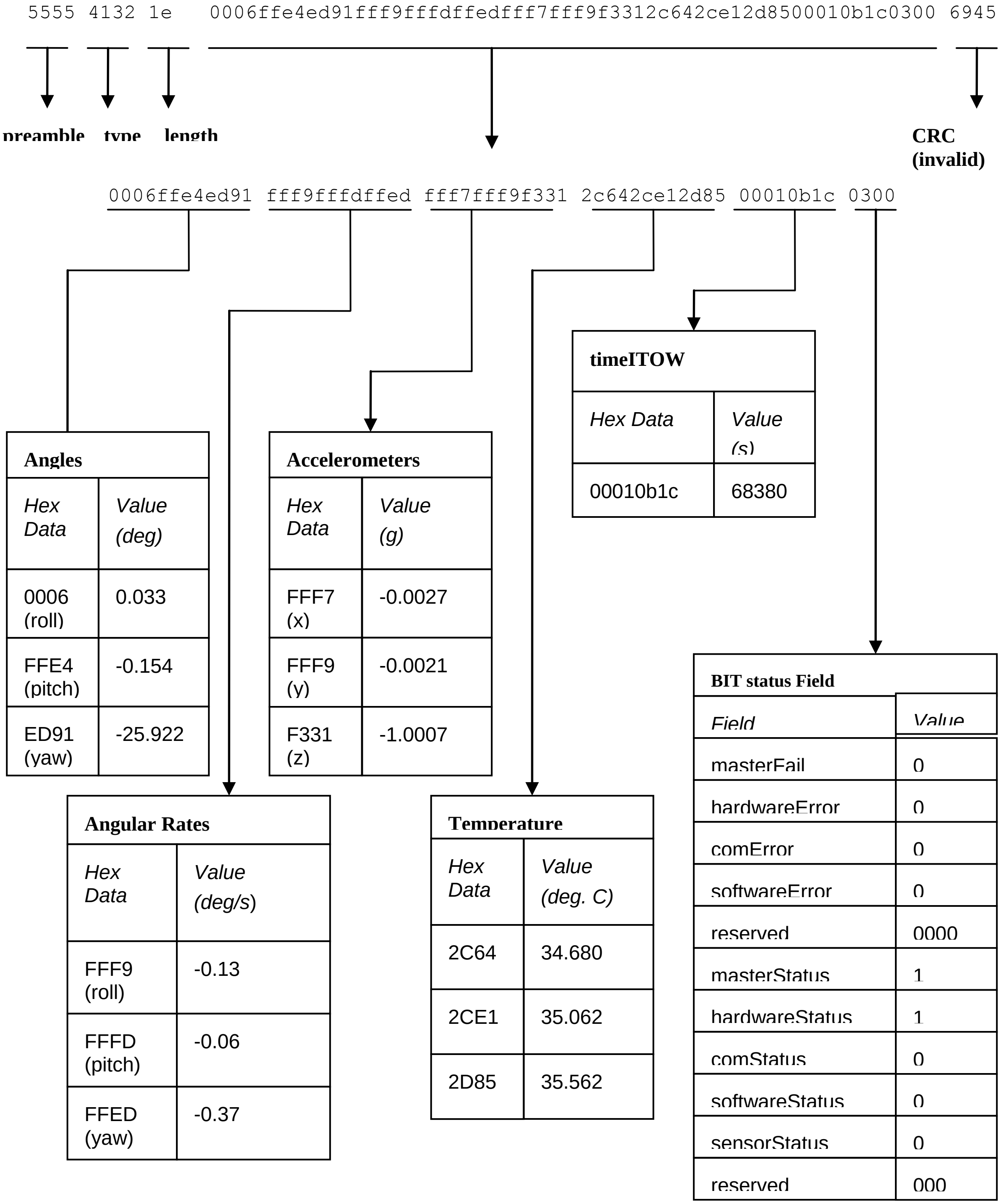
Typically, the parser would be called within a loop in a separate process, or in some time triggered environment, reading the queue looking for packets. A separate process might add data to this queue when it arrives. It is up to the user to ensure circular-queue integrity by using some sort of mutual exclusion mechanism withing the queue access funtions.

12.2 Code listing

```
#include <stdio.h>

/* buffer size */
#define MAXQUEUE 500

/*
 * circular queue
 */
typedef struct queue_tag
{
    int count;
    int front;
    int rear;
    char entry[MAXQUEUE];
} QUEUE_TYPE;

/*
 * crossbow packet
 */
typedef struct xbow_packet
{
    unsigned short packet_type;
    char                      length;
    unsigned short crc;
    char                      data[256];
} XBOW_PACKET;

QUEUE_TYPE circ_buf;

/*******************************************************************************
 * FUNCTION:  process_xbow_packet looks for packets in a queue
 * ARGUMENTS: queue_ptr: is pointer to queue to process
 *                     result: will contain the parsed info when return value is 1
 * RETURNS:    0 when failed.
 *                     1 when successful
 *******************************************************************************/
int process_xbow_packet(QUEUE_TYPE *queue_ptr, XBOW_PACKET *result)
{
    unsigned short myCRC = 0, packetCRC = 0, packet_type = 0, numToPop=0, counter=0;
    char packet[100], tempchar, dataLength;

    if(Empty(queue_ptr))
    {
        return 0;  /* empty buffer */
    }

    /* find header */
    for(numToPop=0; numToPop+1<Size(queue_ptr) ;numToPop+=1)
    {
        if(0x5555==peekWord(queue_ptr, numToPop)) break;
    }

    Pop(queue_ptr, numToPop);
    if(Size(queue_ptr) <= 0)
    {
        /* header was not found */
        return 0;
    }

  /* make sure we can read through minimum length packet */
  if(Size(queue_ptr)<7)
  {
    return 0;
  }

  /* get data length (5th byte of packet) */
  dataLength = peekByte(queue_ptr, 4);

  /* make sure we can read through entire packet */
  if(Size(queue_ptr) < 7+dataLength)
  {
    return 0;
    }



    /* check CRC */
    myCRC = calcCRC(queue_ptr, 2,dataLength+3);
    packetCRC = peekWord(queue_ptr, dataLength+5);

    if(myCRC != packetCRC)
    {
        /* bad CRC on packet – remove the bad packet from the queue and return */
        Pop(queue_ptr, dataLength+7);
        return 0;
    }

    /* fill out result of parsing in structure */
    result->packet_type = peekWord(queue_ptr, 2);
    result->length      = peekByte(queue_ptr, 4);
    result->crc         = packetCRC;
    for(counter=0; counter < result->length; counter++)
    {
        result->data[counter] = peekByte(queue_ptr, 5+counter);
    }

    Pop(queue_ptr, dataLength+7);

    return 1;
}

/********************************************************************************
 * FUNCTION:  calcCRC calculates a 2-byte CRC on serial data using
 *            CRC-CCITT 16-bit standard maintained by the ITU
 *                     (International Telecommunications Union).
 * ARGUMENTS: queue_ptr is pointer to queue holding area to be CRCed
 *                     startIndex is offset into buffer where to begin CRC calculation
 *                     num is offset into buffer where to stop CRC calculation
 * RETURNS:    2-byte CRC
 ********************************************************************************/
unsigned short calcCRC(QUEUE_TYPE *queue_ptr, unsigned int startIndex, unsigned int num) {
    unsigned int i=0, j=0;
    unsigned short crc=0x1D0F; //non-augmented inital value equivalent to augmented initial value 0xFFFF

    for (i=0; i<num; i+=1) {
        crc ^= peekByte(queue_ptr, startIndex+i) << 8;

        for(j=0;j<8;j+=1) {
            if(crc & 0x8000) crc = (crc << 1) ^ 0x1021;
            else crc = crc << 1;
        }
    }
    return crc;
}


/********************************************************************************
 * FUNCTION:  Initialize - initialize the queue
 * ARGUMENTS: queue_ptr is pointer to the queue
 ********************************************************************************/
void Initialize(QUEUE_TYPE *queue_ptr)
{
    queue_ptr->count =  0;
    queue_ptr->front =  0;
    queue_ptr->rear  = -1;
}

/********************************************************************************
 * FUNCTION:  AddQueue - add item in front of queue
 * ARGUMENTS: item holds item to be added to queue
 *                     queue_ptr is pointer to the queue
 * RETURNS:    returns 0 if queue is full. 1 if successful
 ********************************************************************************/
int AddQueue(char item, QUEUE_TYPE *queue_ptr)
{
    int retval = 0;
    if(queue_ptr->count >= MAXQUEUE)
    {
        retval = 0;/* queue is full */
    }
    else
    {
        queue_ptr->count++;
        queue_ptr->rear = (queue_ptr->rear + 1) % MAXQUEUE;
        queue_ptr->entry[queue_ptr->rear] = item;
        retval = 1;
    }
    return retval;
}

/********************************************************************************
 * FUNCTION:  DeleteQeue - return an item from the queue
 * ARGUMENTS: item will hold item popped from queue
 *                     queue_ptr is pointer to the queue
 * RETURNS:    returns 0 if queue is empty. 1 if successful
 *****************************************************************************/
int DeleteQueue(char *item, QUEUE_TYPE *queue_ptr)
{
    int retval = 0;
    if(queue_ptr->count <= 0)
    {
        retval = 0; /* queue is empty */
    }
    else
    {
        queue_ptr -> count--;
        *item = queue_ptr->entry[queue_ptr->front];
        queue_ptr->front = (queue_ptr->front+1) % MAXQUEUE;
        retval=1;
    }
    return retval;
}

/*****************************************************************************
 * FUNCTION:  peekByte returns 1 byte from buffer without popping
 * ARGUMENTS: queue_ptr is pointer to the queue to return byte from
 *                    index is offset into buffer to which byte to return
 * RETURNS:    1 byte
 * REMARKS:    does not do boundary checking. please do this first
 *****************************************************************************/
char peekByte(QUEUE_TYPE *queue_ptr, unsigned int index) {
    char byte;
    int firstIndex;

    firstIndex = (queue_ptr->front + index) % MAXQUEUE;

    byte = queue_ptr->entry[firstIndex];
    return byte;
}


/*****************************************************************************
 * FUNCTION:  peekWord returns 2-byte word from buffer without popping
 * ARGUMENTS: queue_ptr is pointer to the queue to return word from
 *                    index is offset into buffer to which word to return
 * RETURNS:    2-byte word
 * REMARKS:    does not do boundary checking. please do this first
 *****************************************************************************/
unsigned short peekWord(QUEUE_TYPE *queue_ptr, unsigned int index) {
    unsigned short word, firstIndex, secondIndex;

    firstIndex = (queue_ptr->front + index) % MAXQUEUE;
    secondIndex = (queue_ptr->front + index + 1) % MAXQUEUE;

    word = (queue_ptr->entry[firstIndex] << 8) & 0xFF00;
    word |= (0x00FF & queue_ptr->entry[secondIndex]);
    return word;
}

/*****************************************************************************
 * FUNCTION:  Pop - discard item(s) from queue
 * ARGUMENTS: queue_ptr is pointer to the queue
 *                      numToPop is number of items to discard
 * RETURNS:    return the number of items discarded
 ********************************************************************************/
int Pop(QUEUE_TYPE *queue_ptr, int numToPop)
{
    int i=0;
    char tempchar;
    for(i=0; i<numToPop; i++)
    {
        if(!DeleteQueue(&tempchar, queue_ptr))
        {
            break;
        }
    }
    return i;
}

/********************************************************************************
 * FUNCTION:  Size
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return the number of items in the queue
 ********************************************************************************/
int Size(QUEUE_TYPE *queue_ptr)
{
    return queue_ptr->count;
}

/********************************************************************************
 * FUNCTION:  Empty
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return 1 if empty, 0 if not
 ********************************************************************************/
int Empty(QUEUE_TYPE *queue_ptr)
{
    return queue_ptr->count <= 0;
}


/********************************************************************************
 * FUNCTION:  Full
 * ARGUMENTS: queue_ptr is pointer to the queue
 * RETURNS:    return 1 if full, 0 if not full
 ********************************************************************************/
int Full(QUEUE_TYPE *queue_ptr)
{
    return queue_ptr->count >= MAXQUEUE;
}
```

13 Appendix D. Sample Packet Decoding

Example payload from Angle Data Packet 2 (A2)



```
5555 4132 1e   0006ffe4ed91fff9fffdffedfff7fff9f3312c642ce12d8500010b1c0300 6945
```

```
0006ffe4ed91  fff9fffdffed  fff7fff9f331  2c642ce12d85  00010b1c  0300
```

Angles

Hex Data	*Value (deg)*
0006 (roll)	0.033
FFE4 (pitch)	-0.154
ED91 (yaw)	-25.922

Accelerometers

Hex Data	*Value (g)*
FFF7 (x)	-0.0027
FFF9 (y)	-0.0021
F331 (z)	-1.0007

timeITOW

Hex Data	*Value (s)*
00010b1c	68380

Angular Rates

Hex Data	*Value (deg/s)*
FFF9 (roll)	-0.13
FFFD (pitch)	-0.06
FFED (yaw)	-0.37

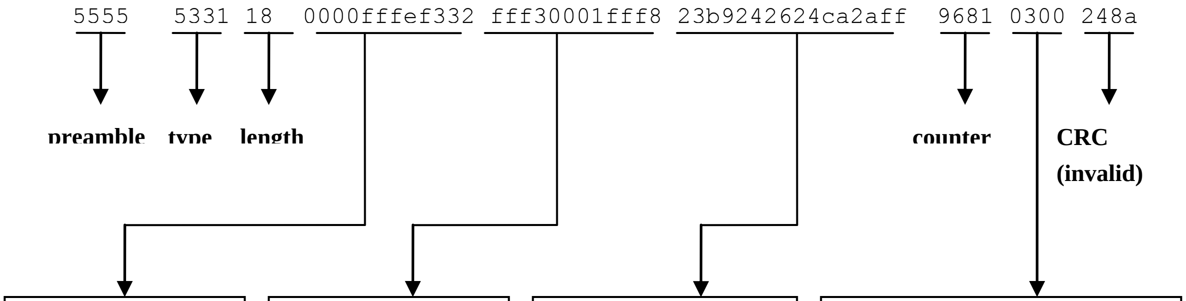
Temperature

Hex Data	*Value (deg. C)*
2C64	34.680
2CE1	35.062
2D85	35.562

BIT status Field

Field	*Value*
masterFail	0
hardwareError	0
comError	0
softwareError	0
reserved	0000
masterStatus	1
hardwareStatus	1
comStatus	0
softwareStatus	0
sensorStatus	0
reserved	000

Example payload from Scaled Sensor 1 data packet (S1)


5555 5331 18 0000fffef332 fff30001fff8 23b9242624ca2aff 9681 0300 248a
preamble
type
length
counter
CRC
(invalid)

Accelerometers

Hex Data	*Value (g)*
0000	0
FFFE	-0.001
F332	-1

Angular Rates

Hex Data	*Value (deg/s)*
FFF3	-0.25
0001	0.02
FFF8	-0.15

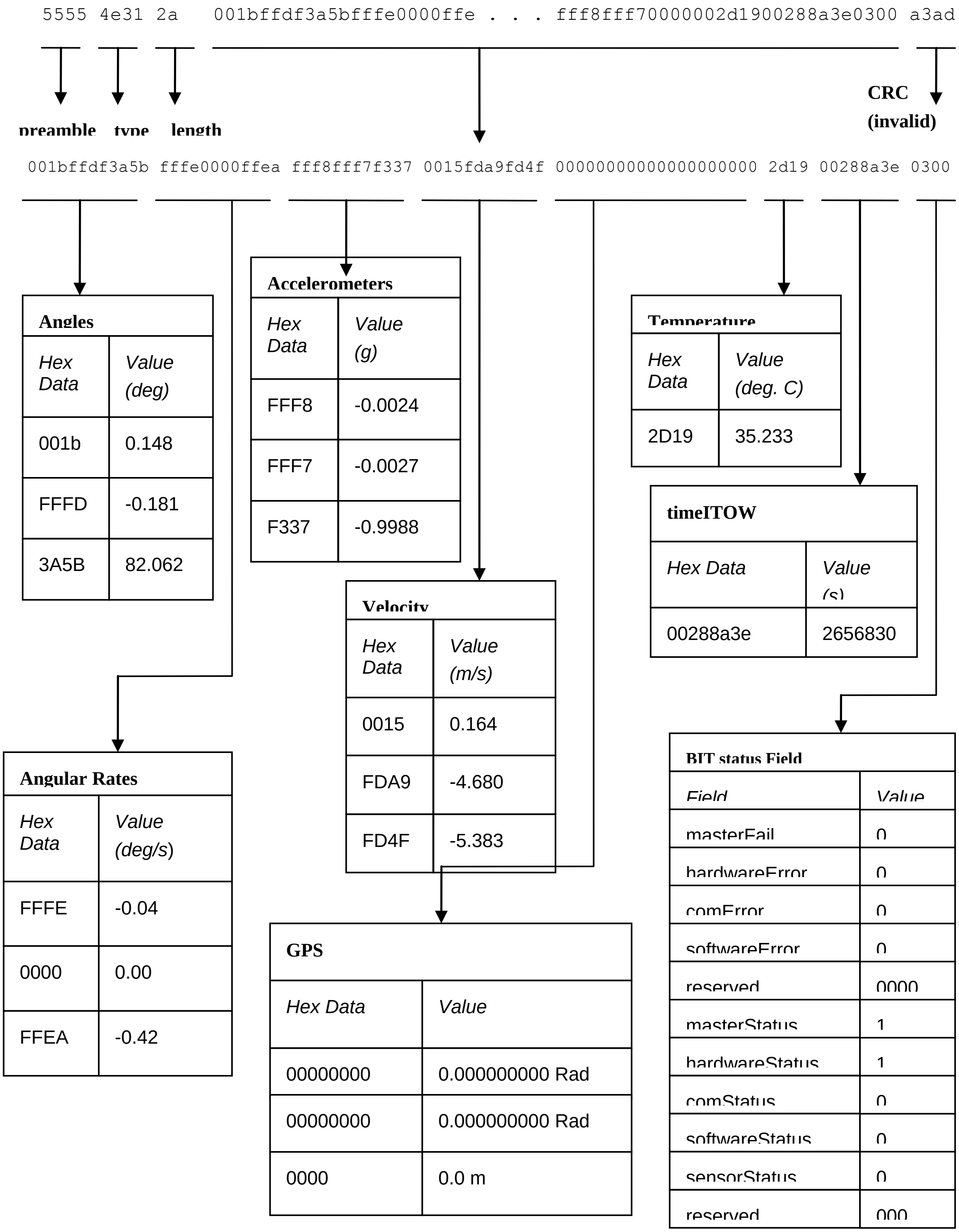
Temperature

Hex Data	*Value (deg. C)*
23B9	28.241
2426	28.741
24CA	33.591
2AFF	38.968

BIT status Field

Field	*Value*
masterFail	0
hardwareError	0
comError	0
softwareError	0
reserved	0000
masterStatus	1
hardwareStatus	1
comStatus	0
softwareStatus	0
sensorStatus	0
reserved	000

Example payload from Nav Data Packet 1 (N1)


5555 4e31 2a 001bffdf3a5bfffe0000ffe . . . fff8fff70000002d1900288a3e0300 a3ad
preamble
type
length
CRC
(invalid)
001bffdf3a5b fffe0000ffea fff8fff7f337 0015fda9fd4f 00000000000000000000 2d19 00288a3e 0300

Angles

Hex Data	*Value (deg)*
001b	0.148
FFFD	-0.181
3A5B	82.062

Accelerometers

Hex Data	*Value (g)*
FFF8	-0.0024
FFF7	-0.0027
F337	-0.9988

Temperature

Hex Data	*Value (deg. C)*
2D19	35.233

timeITOW

Hex Data	*Value (s)*
00288a3e	2656830

Velocity

Hex Data	*Value (m/s)*
0015	0.164
FDA9	-4.680
FD4F	-5.383

Angular Rates

Hex Data	*Value (deg/s)*
FFFE	-0.04
0000	0.00
FFEA	-0.42

GPS

Hex Data	*Value*
00000000	0.000000000 Rad
00000000	0.000000000 Rad
0000	0.0 m

BIT status Field

Field	*Value*
masterFail	0
hardwareError	0
comError	0
softwareError	0
reserved	0000
masterStatus	1
hardwareStatus	1
comStatus	0
softwareStatus	0
sensorStatus	0
reserved	000

14 Warranty and Support Information

14.1 Customer Service

As a Crossbow Technology customer you have access to product support services, which include:

- Single-point return service
- Web-based support service
- Same day troubleshooting assistance
- Worldwide Crossbow representation
- Onsite and factory training available
- Preventative maintenance and repair programs
- Installation assistance available

14.2 Contact Directory

United States:	Phone: 1-408-965-3300 (8 AM to 5 PM PST)
	Fax:1-408-324-4840 (24 hours)
	Email: techsupport@xbow.com
Non-U.S.:	Refer to website www.xbow.com

14.3 Return Procedure

14.3.1 Authorization

Before returning any equipment, please contact Crossbow to obtain a Returned Material Authorization number (RMA).

Be ready to provide the following information when requesting a RMA:

- Name
- Address
- Telephone, Fax, Email
- Equipment Model Number
- Equipment Serial Number
- Installation Date
- Failure Date
- Fault Description
- Will it connect to NAV-VIEW 2.X?

14.3.2 Identification and Protection

If the equipment is to be shipped to Crossbow for service or repair, please attach a tag TO THE EQUIPMENT, as well as the shipping container(s), identifying the owner. Also indicate the service or repair required, the problems encountered, and other information considered valuable to the service facility such as the list of information provided to request the RMA number.

Place the equipment in the original shipping container(s), making sure there is adequate packing around all sides of the equipment. If the original shipping containers were discarded, use heavy boxes with adequate padding and protection.

14.3.3 Sealing the Container

Seal the shipping container(s) with heavy tape or metal bands strong enough to handle the weight of the equipment and the container.

14.3.4 Marking

Please write the words, "***FRAGILE, DELICATE INSTRUMENT***" in several places on the outside of the shipping container(s). In all correspondence, please refer to the equipment by the model number, the serial number, and the RMA number.

14.3.5 Return Shipping Address

Use the following address for all returned products:

Crossbow Technology, Inc.
1421 McCarthy Blvd.
Milpitas, CA 95035
Attn: RMA Number (XXXXXX)

14.4 Warranty

The Crossbow product warranty is one year from date of shipment.

Crossbow Technology, Inc.
1421 McCarthy Blvd.
Milpitas, CA 95035
Phone: 408.965.3300
Fax: 408.324.4840
Email: info@xbow.com
Website: www.xbow.com