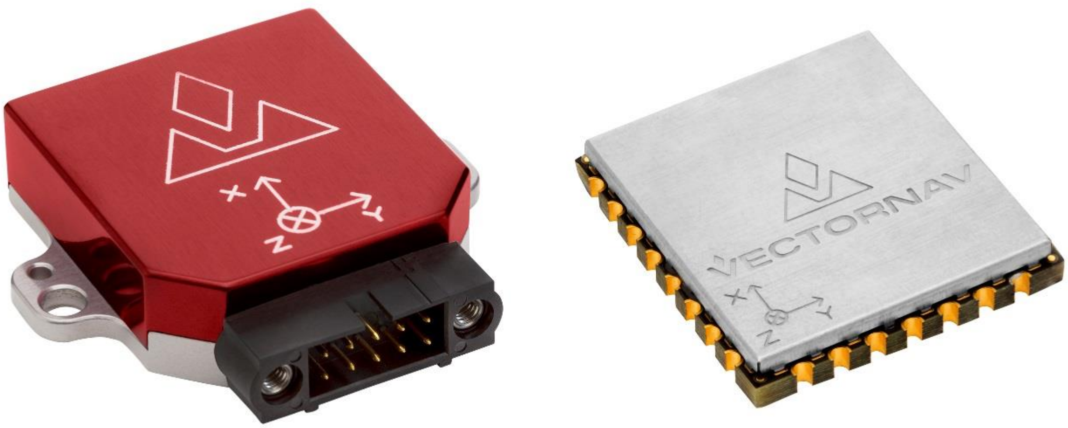
Embedded Navigation Solutions

VN-100 User Manual



Firmware v2.1.0.0

Document Revision 2.22

Document Information

Title	VN-100 User Manual
Subtitle	Inertial Navigation Modules
Document Type	User Manual
Document Number	UM001 v2.22
Document Status	Released

VectorNav Technical Documentation

In addition to our product-specific technical data sheets, the following manuals are available to assist VectorNav customers in product design and development.

- **VN-100 User Manual:** The user manual provides a high-level overview of product specific information for each of our inertial sensors. Further detailed information regarding hardware integration and application specific use can be found in the separate documentation listed below.
- **Hardware Integration Manual:** This manual provides hardware design instructions and recommendations on how to integrate our inertial sensors into your product.
- **Application Notes:** This set of documents provides a more detailed overview of how to utilize many different features and capabilities offered by our products, designed to enhance performance and usability in a wide range of application-specific scenarios.

Document Symbols

The following symbols are used to highlight important information within the manual:

The information symbol points to important information within the manual.

The warning symbol points to crucial information or actions that should be followed to avoid reduced performance or damage to the navigation module.

Technical Support

Our website provides a large repository of technical information regarding our navigation sensors. A list of the available documents can be found at the following address:

http://www.vectornav.com/support

If you have technical problems or cannot find the information that you need in the provided documents, please contact our support team by email or phone. Our engineering team is committed to providing the required support necessary to ensure that you are successful with the design, integration, and operation of our embedded navigation sensors.

Technical Support Contact Info

Email: support@vectornav.com **Phone:** +1.512.772.3615

Table of Contents

1 Introduction

1.1 Product Description

The VN-100 is a miniature surface mount high-performance Inertial Measurement Unit (IMU) and Attitude Heading Reference System (AHRS). Incorporating the latest solid-state MEMS sensor technology, the VN-100 combines a set of 3-axis accelerometers, 3-axis gyroscopes, 3-axis magnetometers, a barometric pressure sensor and a 32-bit processor. The VN-100 is considered both an IMU in that it can output acceleration, angular rate, and magnetic measurements along the X, Y, & Z axes of the sensor as well as an AHRS in that it can output filtered attitude estimates of the sensor with respect to a local coordinate frame.

1.2 Factory Calibration

MEMS inertial sensors are subject to several common sources of error: bias, scale factor, misalignments, temperature dependencies, and gyro g-sensitivity. All VN-100 sensors undergo a rigorous calibration process at the VectorNav factory to minimize these error sources. Compensation parameters calculated during these calibrations are stored on each individual sensor and digitally applied to the real-time measurements.

- Thermal Calibration – this option extends the calibration process over multiple temperatures to ensure performance specifications are met over the full operating temperature range of -40 C to +85 C.

1.3 Operation Overview

The VN-100 has a built-in microcontroller that runs a quaternion based Extended Kalman Filter (EKF), which provides estimates of both the attitude of the sensor as well as the real-time gyro biases. VectorNav uses a quaternion based attitude filter because it is continuous over a full 360 degree range of motion such that there are no limitations on the angles it can compute. However, the VN-100 also has a built-in capability to output yaw, pitch, and roll angles from the VN-100, in which the sensor automatically converts from quaternions to the desired attitude parameter. Outputs from the VN-100 include:

- **Attitude:**
    - Yaw, Pitch, & Roll
    - Quaternions
    - Direction Cosine Matrix
- **Angular Rates:**
    - Bias-Compensated
    - Calibrated X, Y, & Z Gyro Measurements
- **Acceleration:**
    - Calibrated X, Y, & Z Measurements
- **Magnetic:**
    - Calibrated X, Y, & Z Measurements
- Barometric Pressure

The VN-100 EKF relies on comparing measurements from the onboard inertial sensors to two reference vectors in calculating the attitude estimates: gravity down and magnetic North. Measurements from the three-axis accelerometer are compared to the expected magnitude and direction of gravity in determining the pitch and roll angles while measurements from the three-axis magnetometer are compared to the expected magnitude and direction of Earth's background magnetic field in determining the heading angle (i.e. yaw angle with respect to Magnetic North).

> The VN-100 Kalman Filter is based on the assumption that the accelerometer measurements should only be measuring gravity down. If the sensor is subject to dynamic motion that induces accelerations, the pitch and roll estimates will be subject to increased errors. These measurements can be accounted and compensated for by using the VN-100 Velocity Aiding Feature (See Section 10 for more information).

> The VN-100 filter relies on comparing the onboard magnetic measurements to Earth's background magnetic field in determining its heading angle. Common objects such as batteries, electronics, cars, rebar in concrete, and other ferrous materials can bias and distort the background magnetic field leading to increased errors. These measurements can be accounted and compensated for by using the VN-100 Hard/Soft Iron Algorithms (See Section 9 for more information).

> VectorNav has developed a suite of tools called the Vector Processing Engine (VPE™), which are built-into the VN-100 and minimize the effects of these disturbances; however, it is not possible to obtain absolute heading accuracies better than 2 degrees over any extended period of time when relying on magnetometer measurements.

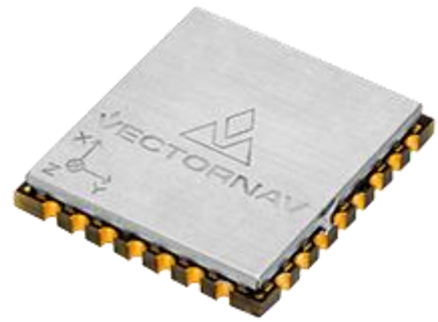
The VN-100 EKF also integrates measurements from the three-axis gyroscopes to provide faster and smoother attitude estimates as well as angular rate measurements. Gyroscopes of all kinds are subject to bias instabilities, in which the zero readings of the gyro will drift over time to due to inherent noise properties of the gyro itself. The VN-100 EKF uses the accelerometer and magnetometer measurements to continuously estimate the gyro bias, such that the report angular rates are compensated for this drift.

1.4 Packaging Options

The VN-100 is available in two different configurations; a 30-pin surface mount package (VN-100 SMD) and an aluminum encased module (VN-100 Rugged). The VN-100 surface mount package is well suited for customers looking to integrate the VN-100 sensor at the electronics level while the VN-100 Rugged provides a precision enclosure with mounting tabs and alignment holes for a more off-the-shelf solution.

1.4.1 Surface-Mount Package

For embedded applications, the VN-100 is available in a miniature surface-mount package.



Features

- Small Size: 22 x 24 x 3 mm
- Single Power Supply: 3.2 to 5.5 V
- Communication Interface: Serial TTL & SPI
- Low Power Requirement: < 105 mW @ 3.3V

1.4.2 Rugged Package

The VN-100 Rugged consists of the VN-100 sensor installed and calibrated in a robust precision aluminum enclosure.

Features

- Precision aluminum enclosure
- Locking 10-pin connector
- Mounting tabs with alignment holes
- Compact Size: 36 x 33 x 9 mm
- Single Power Supply: 4.5 to 5.5 V
- Communication Interface: Serial RS-232 & TTL

1.4.3 Surface Mount Development Kit

The VN-100 Development Kit provides the VN-100 surface-mount sensor installed onto a small PCB, providing easy access to all of the features and pins on the VN-100. Communication with the VN-100 is provided by USB and RS-232 serial communication ports. A 30-pin header provides easy access to each of the critical pins. The VN-100 Development Kit also includes all of the necessary cabling, documentation, and support software.

Features

- Pre-installed VN-100 Sensor
- Onboard USB->Serial converter
- Onboard TTL->RS-232 converter
- 30-pin 0.1” header for access to VN-100 pins
- Power supply jack – 5V (Can be powered from USB)

- Board Size: 76 x 76 x 14 mm

1.4.4 VN-100 Rugged Development Kit

The VN-100 Rugged Development Kit includes the VN-100 Rugged sensor along with all of the necessary cabling required for operation. Two cables are provided in each Development Kit: one custom cable for RS-232 communication and a second custom cable with a built in USB converter. The Development Kit also includes all of the relevant documentation and support software.

Features

- VN-100 Rugged Sensor
- 10 ft RS-232 cable
- 10 ft USB connector cable
- Cable Connection Tool
- CD w/Software Development Kit
- User Manual, Quick Start Guide & Documentation
- Carrying Case

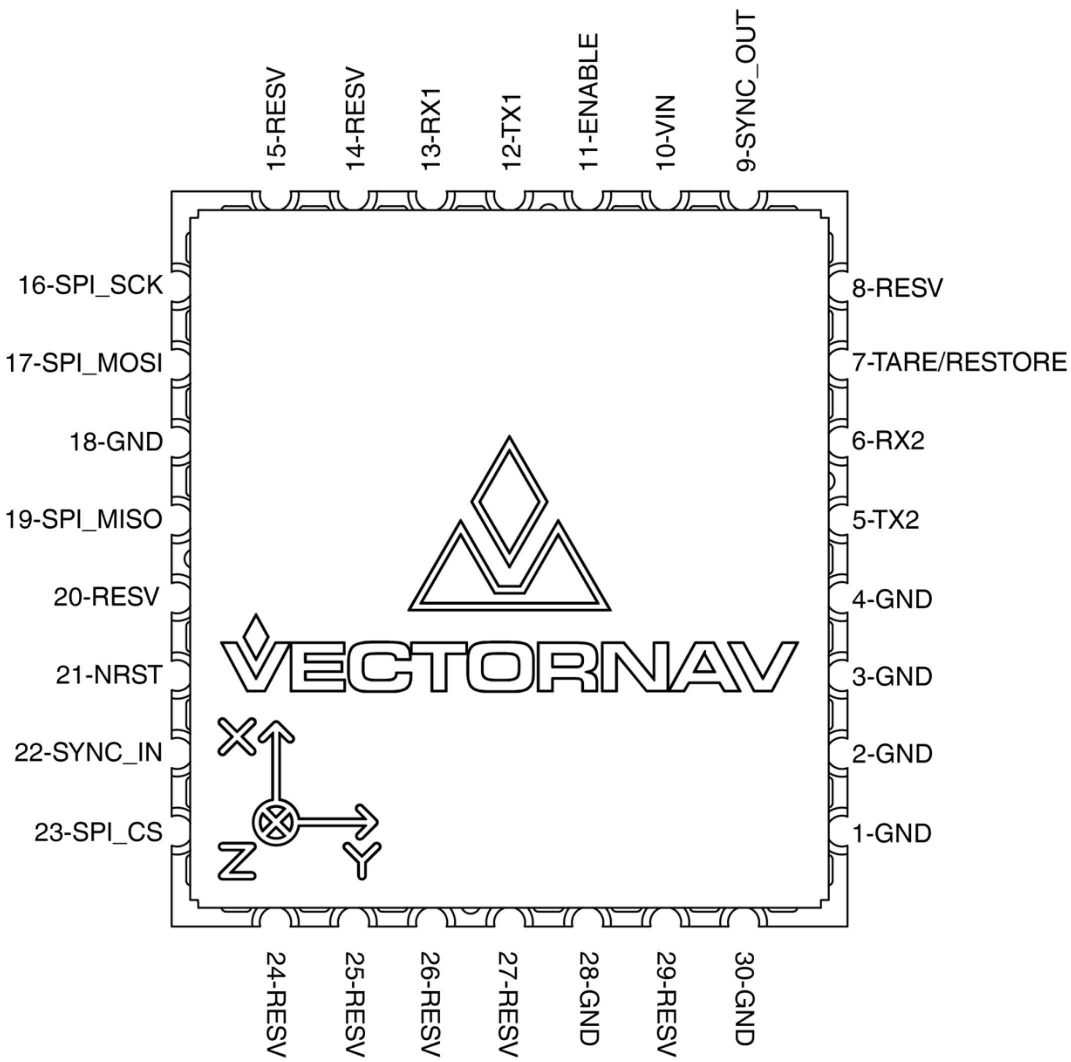
1.5 VN-100 Product Codes

VN-100 Options			
Item Code	**Sensor Packaging**	**Calibration Option**	**Product Type**
VN-100S	Surface Mount Device	Standard at 25C	IMU/AHRS
VN-100T	Surface Mount Device	Thermal -40C to +85C	IMU/AHRS
VN-100S-DEV	Surface Mount Development Kit	Standard at 25C	IMU/AHRS
VN-100T-DEV	Surface Mount Development Kit	Thermal -40C to +85C	IMU/AHRS
VN-100S-CR	Rugged Module	Standard at 25C	IMU/AHRS
VN-100T-CR	Rugged Module	Thermal -40C to +85C	IMU/AHRS
VN-100S-CR-DEV	Rugged Development Kit	Standard at 25C	IMU/AHRS
VN-100T-CR-DEV	Rugged Development Kit	Thermal -40C to +85C	IMU/AHRS
VN-C100-0310	VN-100 Rugged USB Adapter Cable	N/A	Cable
VN-C100-0410	VN-100 Rugged Serial Adapter Cable	N/A	Cable

2 Specifications

2.1 VN-100 Surface-Mount Sensor (SMD) Electrical

Pin assignments (top down view)



VN-100 SMD Pin Assignments

Pin	Pin Name	Type	Description
1	GND	Supply	Ground.
2	GND	Supply	Ground.
3	GND	Supply	Ground.
4	GND	Supply	Ground.
5	TX2	Output	Serial UART #2 data output. (sensor)
6	RX2	Input	Serial UART #2 data input. (sensor)
7	RESTORE	Input	Normally used to zero (tare) the attitude. To tare, pulse high for at least 1 µs. During power on or device reset, holding this pin high will cause the module to restore the default factory settings. As a result, the pin cannot be used for tare until at least 5 ms after a power on or reset. Internally held low with 10k resistor.
8	RESV	N/A	Reserved for internal use. Do not connect.
9	SYNC_OUT	Output	Time synchronization output signal.
10	VIN	Supply	3.2 - 5.5 V input.
11	ENABLE	Input	Leave high for normal operation. Pull low to enter sleep mode. Internally pulled high with pull-up resistor.
12	TX1	Output	Serial UART #1 data output. (sensor)
13	RX1	Input	Serial UART #1 data input. (sensor)
14	RESV	N/A	Reserved for internal use. Do not connect.
15	RESV	N/A	Reserved for internal use. Do not connect.
16	SPI_SCK	Input	SPI clock.
17	SPI_MOSI	Input	SPI input.
18	GND	Supply	Ground.
19	SPI_MISO	Output	SPI output.
20	RESV	N/A	Reserved for internal use. Do not connect.
21	NRST	Input	Microcontroller reset line. Pull low for > 20 µs to reset MCU. Internally pulled high with 10k.
22	SYNC_IN	Input	Time synchronization input signal.
23	SPI_CS	Input	SPI slave select.
24	RESV	N/A	Reserved for internal use. Do not connect.
25	RESV	N/A	Reserved for internal use. Do not connect.
26	RESV	N/A	Reserved for internal use. Do not connect.
26	RESV	N/A	Reserved for internal use. Do not connect.
28	GND	Supply	Ground.
29	RESV	N/A	Reserved for internal use. Do not connect.
30	GND	Supply	Ground.

2.1.1 VN-100 SMD Power Supply

The minimum operating supply voltage is 3.2V and the absolute maximum is 5.5V.

2.1.2 VN-100 SMD Serial (UART) Interface

The serial interface on the VN-100 operates with 3V TTL logic.

Serial I/O Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5 V		0.8 V
Input high level voltage	2 V		5.5 V
Output low voltage	0 V		0.4 V
Output high voltage	2.4 V		3.0 V

2.1.3 VN-100 SMD Serial Peripheral Interface (SPI)

Serial I/O Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5 V		0.8 V
Input high level voltage	2 V		5.5 V
Output low voltage	0 V		0.4 V
Output high voltage	2.4 V		3.0 V
Clock Frequency		8 MHz	16 MHz
Close Rise/Fall Time			8 ns

2.1.4 VN-100 SMD Reset, SyncIn/Out, and Other General I/O Pins

NRST Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5 V		0.8 V
Input high level voltage	2 V		5.5 V
Weak pull-up equivalent resistor	30 kΩ	40 kΩ	50 kΩ
NRST pulse width	20 µs		

SyncIn Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5 V		0.8 V
Input high level voltage	2 V		5.5 V
Pulse Width	100 ns		

SyncOut Specifications

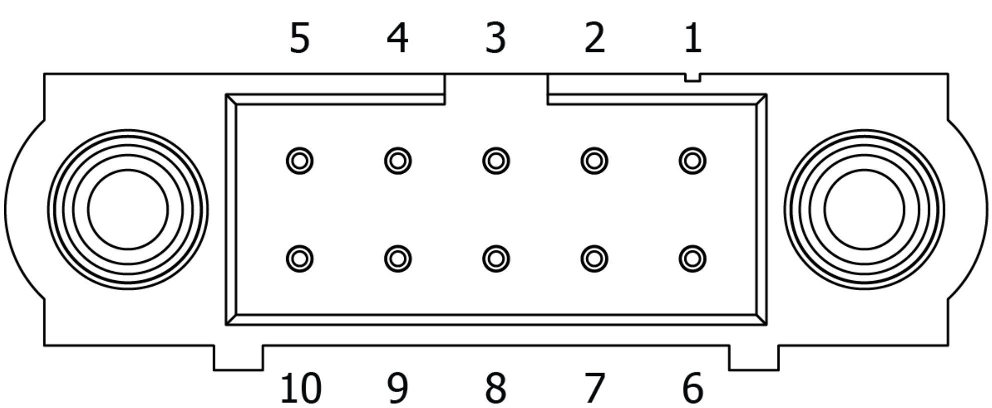
Specification	Min	Typical	Max
Output low voltage	0 V		0.4 V
Output high voltage	2.4 V		3.0 V
Output high to low fall time			125 ns
Output low to high rise time			125 ns
Output Frequency	1 Hz		1 kHz

2.2 VN-100 Rugged Electrical

VN-100 Rugged Pin Assignments

Pin	Pin Name	Description
1	VCC	+4.5V to +5.5V
2	TX1	RS-232 voltage levels data output from the sensor. (Serial UART #1)
3	RX1	RS-232 voltage levels data input to the sensor. (Serial UART #1)
4	SYNC_OUT	Output signal used for synchronization purposes. Software configurable to pulse when ADC, IMU, or attitude measurements are available.
5	GND	Ground
6	TARE/RESTORE	Input signal used to zero the attitude of the sensor. If high at reset, the device will restore to factory default state. Internally held low with 10k resistor.
7	SYNC_IN	Input signal for synchronization purposes. Software configurable to either synchronize the measurements or the output with an external device.
8	TX2_TTL	Serial UART #2 data output from the device at TTL voltage level (3V).
9	RX2_TTL	Serial UART #2 data into the device at TTL voltage level (3V).
10	RESV	This pin should be left unconnected.

VN-100 Rugged External Connector



2.2.1 VN-100 Rugged Power Supply

The power supply input for the VN-100 Rugged is 4.5 to 5.5 V DC.

2.2.2 VN-100 Rugged Serial UART Interface

Serial I/O Specifications

Specification	Min	Typical	Max
Input low level voltage	-25 V		
Input high level voltage			25 V
Output low voltage	-5.0 V	-5.4 V	
Output high voltage	5.0 V	5.5 V	
Output resistance	300 Ω	10 MΩ	
Data rate			1 Mbps
Pulse slew		300 ns	

2.2.3 VN-100 Rugged Reset, SyncIn/Out, and Other General I/O Pins

NRST Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5 V		0.8 V
Input high level voltage	2 V		5.5 V
Weak pull-up equivalent resistor	30 kΩ	40 kΩ	50 kΩ
NRST pulse width	20 μs		

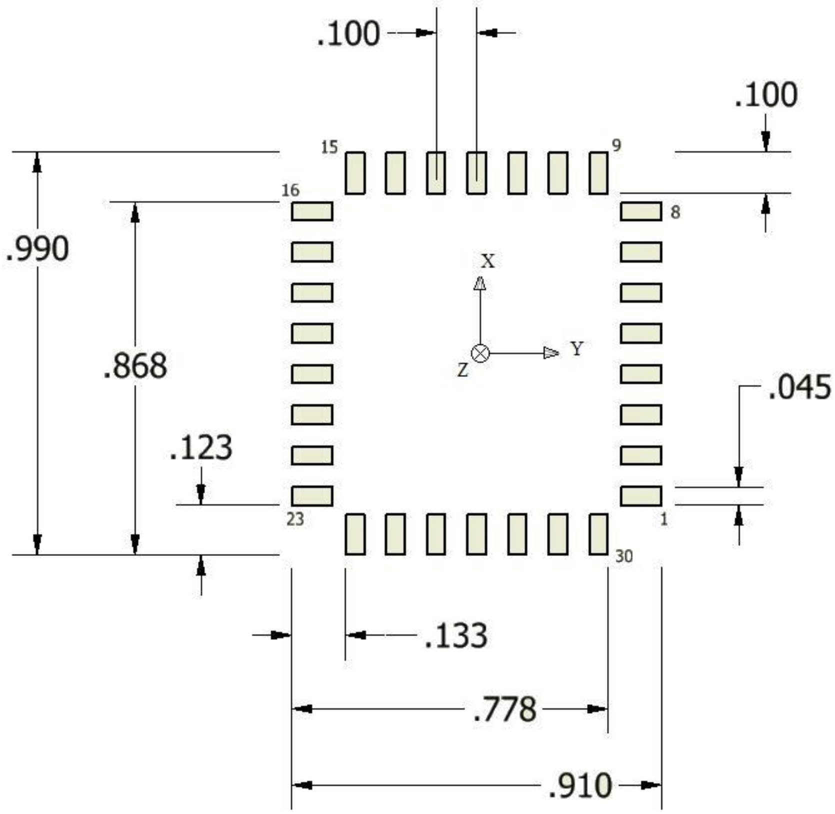
SyncIn Specifications

Specification	Min	Typical	Max
Input low level voltage	-0.5V		0.8V
Input high level voltage	2V		5.5V
Pulse Width	100 ns		

SyncOut Specifications

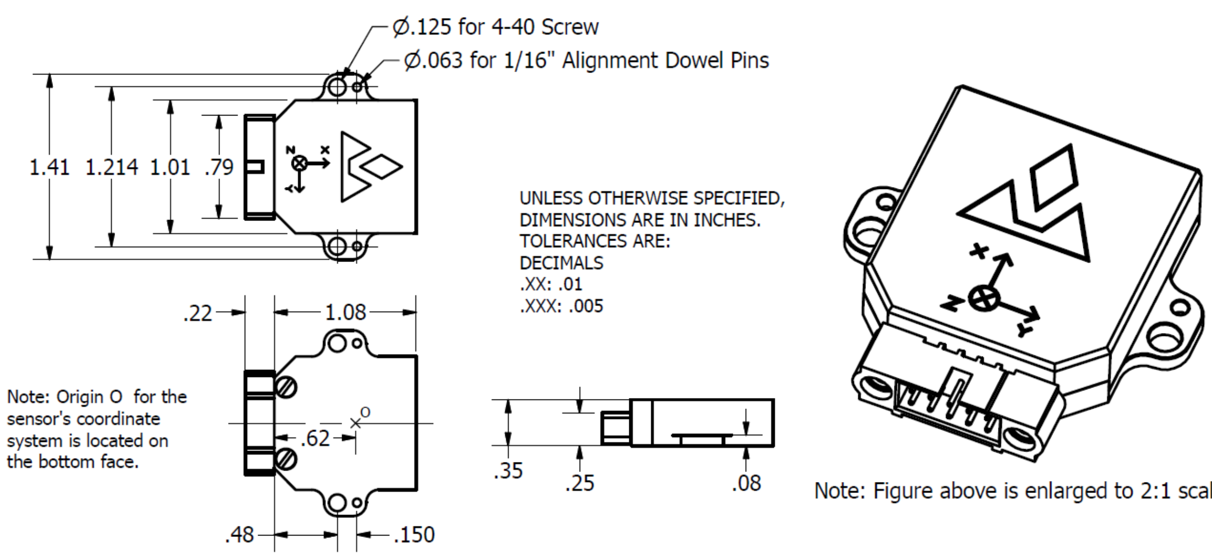
Specification	Min	Typical	Max
Output low voltage	0 V		0.4 V
Output high voltage	2.4 V		3.0 V
Output high to low fall time			125 ns
Output low to high rise time			125 ns
Output Frequency	1 Hz		1 kHz

2.3 VN-100 Surface-Mount Sensor (SMD) Dimensions



* Measurements are in inches

2.4 VN-100 Rugged Dimensions



Note: Figure above is enlarged to 2:1 scale.

2.4.1 Rugged Connector Type

The main connector used on the VN-100 Rugged is a 10-pin Harwin M80-5001042. The mating connector used on the cable assemblies provided by VectorNav for use with the VN-100 Rugged is a Harwin M80-4861005.

2.5 Absolute Maximum Ratings

SMD Absolute Maximum Ratings

Specification	Min	Max
Input Voltage	-0.3 V	5.5 V
Operating Temperature	-40 C	85 C
Storage Temperature	-40 C	85 C

Rugged Absolute Maximum Ratings

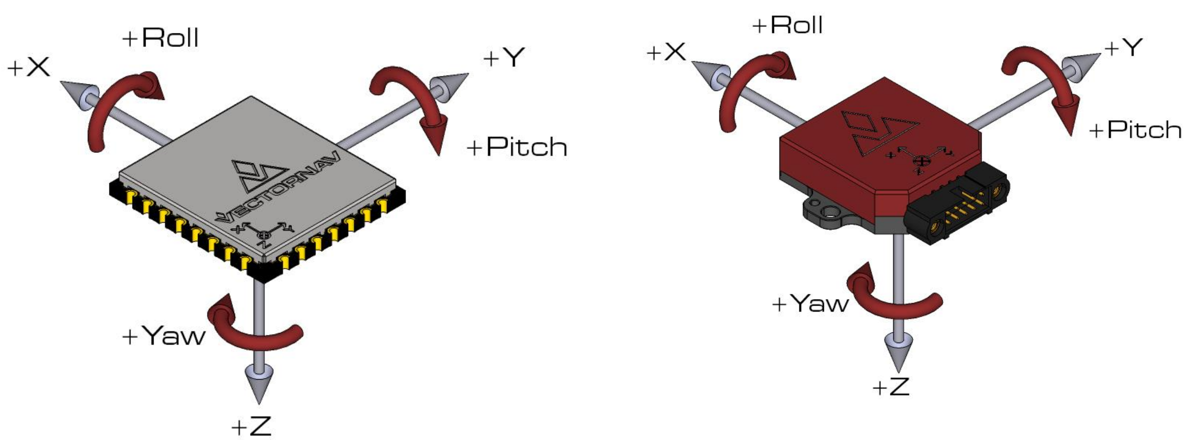
Specification	Min	Max
Input Voltage	-0.3 V	5.5 V
Operating Temperature	-40 C	85 C
Storage Temperature	-40 C	85 C

2.6 Sensor Coordinate System

2.6.1 Sensor Coordinate Frame

The VN-100 uses a right-handed coordinate system. A positive yaw angle is defined as a positive right-handed rotation around the Z-axis. A positive pitch angle is defined as a positive right-handed rotation around the Y-axis. A positive roll angle is defined as a positive right-handed rotation around the X-axis. The axes direction with respect to the VN-100 module is shown in the figure below.

VN-100 Coordinate System



2.6.2 North-East-Down Frame

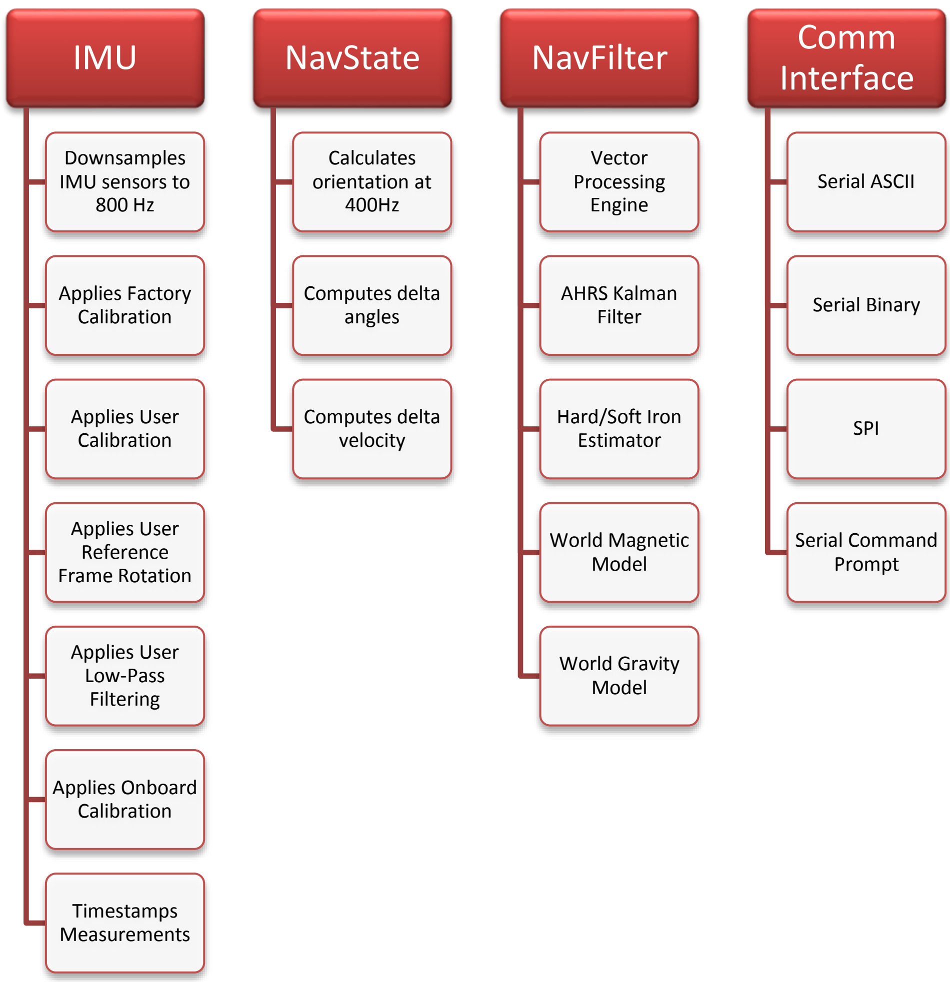
The VN-100 velocity estimates can be output in the North-East-Down (NED) coordinate frame defined as follows (**N_X, N_Y, N_Z**):

- Right-handed, Cartesian, non-inertial, geodetic frame with origin located at the surface of Earth (WGS84 ellipsoid);
- Positive X-axis points towards North, tangent to WGS84 ellipsoid;
- Positive Y-axis points towards East, tangent to WGS84 ellipsoid;
- Positive Z-axis points down into the ground completing the right-handed system.

3 VN-100 Software Architecture

The software architecture internal to the VN-100 includes four separate subsystems. These subsystems are the IMU, the NavState, the NavFilter, and the Communication Interface. The high-level functions performed by these subsystems are outlined below. This chapter describes these functions performed by these subsystems in more detail and describes which of the various measurement outputs originate from each of these corresponding subsystems.

VN-100 Software Architecture

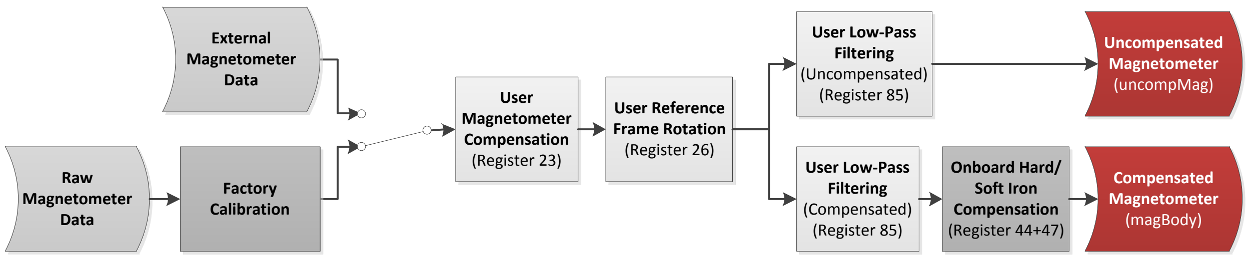


3.1 IMU Subsystem

The IMU subsystem runs at the highest system rate, described from this point forward as the IMU Rate (defaults to 800 Hz). It is responsible for collecting the raw IMU measurements, applying a factory, user, and dynamic calibration to these measurements, and optionally filtering the individual sensor measurements for output. The coning and sculling integrals also are calculated by the IMU subsystem at the full IMU Rate. The IMU subsystem is also responsible for time stamping the IMU measurements to internal system time, and relative to the SyncIn signal.

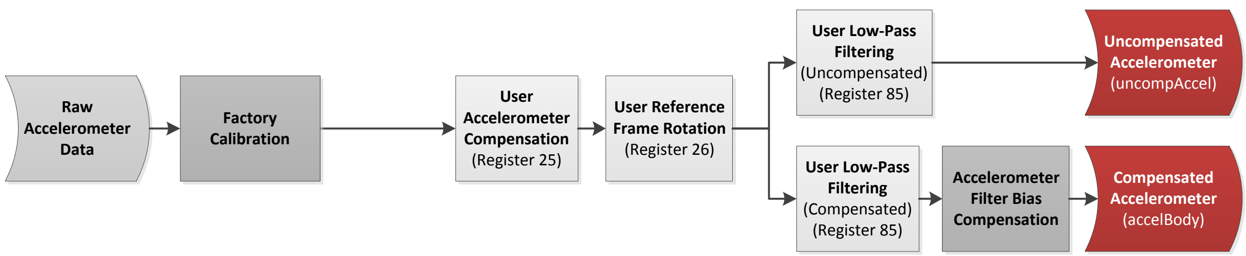
3.1.1 Magnetometer

Magnetometer IMU Measurements



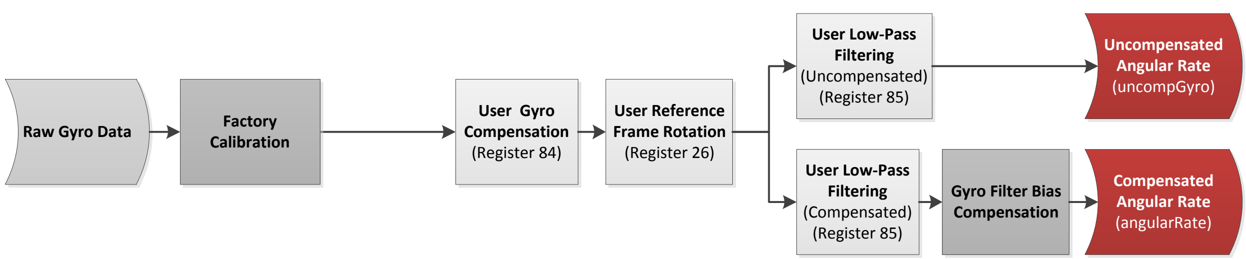
3.1.2 Accelerometer

Accelerometer IMU Measurements



3.1.3 Gyro

Gyro IMU Measurements



3.1.4 Raw IMU Measurements

The raw IMU measurements are collected from the internal MEMS at the highest rate available for each individual sensor. For the gyro and accelerometer, the measurements are down-sampled to the IMU Rate.

3.1.5 Factory Calibration

Each VN-100 sensor is tested at the factory at multiple known angular rates, accelerations, and magnetic field strengths to determine each sensor's unique bias, scale factor, axis alignment, and temperature dependence. The calibration coefficients required to remove these unwanted errors are permanently stored in flash memory on each sensor. At the IMU Rate, these calibration coefficients are applied to the raw IMU measurements, to correct for and remove these known measurement errors. For thermally calibrated units the onboard temperature sensor is used to remove the measurement temperature dependence. The output of the factory calibration stage is referred to as the calibrated (but un-compensated) IMU measurements.

3.1.6 User Calibration

The VN-100 provides the user with the ability to apply a separate user calibration to remove additional bias, scale factor, and axis misalignments. The user calibration is applied after the factory calibration, and can be used to optionally fine tune the calibration for each of the individual sensors. The user calibration is optional and in most cases not required for normal operation.

3.1.7 User Reference Frame Rotation

The user reference frame rotation provides the user with the ability to apply a rigid body rotation to each of the sensor outputs. This can be used to transform the coordinate system of the onboard sensors into any other coordinate frame of the user's choice. Since this transformation is applied to the IMU measurements prior to their use in the onboard attitude estimation algorithms, applying a user reference frame rotation will not only change the output coordinates for the IMU measurements, it will also change the IMU body frame for all subsequent attitude estimation calculations.

> A write settings and reset command must be issued after setting the Reference Frame Rotation Register before coordinate transformation will be applied.

3.1.8 User Low-Pass Filtering

The VN-100 also provides a means (see Register 85) to apply low-pass filtering to the output compensated IMU measurements. It is important to note that the user low-pass filtering only applies to the output compensated IMU measurements. All onboard Kalman filters in the NavFilter subsystem always use the unfiltered IMU measurements after the User Reference Frame Rotation (Register 26) has been applied. As such the onboard Kalman filtering will not be affected by the user low-pass filter settings. The user low-pass filtering can be used to down-sample the output IMU measurements to ensure that information is not lost when the IMU measurements are sampled by the user at a lower rate than the internal IMU Rate.

3.1.9 Timestamp Measurements

All onboard measurements captured by the IMU subsystem are time stamped relative to several internal timing events. These events include the monotonically increasing system time (time since startup), the time since the last SyncIn event, and the time since the last GPS PPS pulse. These timestamps are recorded with microsecond resolution and ~10 microsecond accuracy relative to the onboard temperature compensated crystal oscillator. The onboard oscillator has a timing accuracy of ~20ppm over the temperature range of -40C to 80C.

3.1.10 Coning & Sculling

The IMU subsystem is also responsible for computing and accumulating the coning and sculling integrals. These integrals track the delta angle and delta velocity accumulated from one time step to another. The coning and sculling integrals are reset each time the delta angle and/or delta velocity are outputted (asynchronously) or polled from the delta theta and velocity register (Register 80). Between output and polling events, the coning and sculling integration are performed by the IMU subsystem at the IMU Rate.

3.2 NavState Subsystem

The NavState subsystem generates a continuous reliable stream of low-latency, low-jitter state outputs at a rate fixed to the IMU sample rate. The state outputs include any output such as attitude, position, and velocity, which are not directly measureable by the IMU and hence must be estimated by the onboard Kalman filters. The NavState runs immediately after, and in sync with the IMU subsystem, at a rate divisible into the IMU Rate. This rate is referred to as the NavState Rate (default 800 Hz). The NavState decouples the rate at which the state outputs are made available to the user from the rate at which they are being estimated by the onboard Kalman filters. This is very important for many applications which depend on low-latency, low-jitter attitude, position, and velocity measurements as inputs to their control loops. The NavState guarantees the output of new updated state information at a rate fixed to the IMU Rate with very low latency and output jitter. The NavState also provides the ability for the VN-100 to output estimated states at rates faster than the rate of the onboard Kalman filters, which may be affected by system load and input measurements availability.

3.2.1 NavState Measurements

The measurements shown below are calculated by the NavState subsystem and are made available at the NavState Rate (default 800 Hz).

NavState Outputs
Attitude (Yaw, Pitch, Roll, Quaternion, DCM)
Position (LLA, ECEF)
Velocity (NED, ECEF, Body)
Delta Angle (Available at full IMU rate)
Delta Velocity (Available at full IMU rate)

3.3 NavFilter Subsystem

The NavFilter subsystem consists of the INS Kalman filter, the Vector Processing Engine (VPE), and its collection of other Kalman filters and calculations that run at a lower rate than the NavState. Most high level states such as the estimated attitude, position, and velocity are passed from the NavFilter to the NavState, and as such are made available to the user at the NavState rate. There are a handful of outputs however that will only update at the rate of the NavFilter, some of which are listed below.

NavFilter Outputs
Attitude Uncertainty
Position & Velocity Uncertainty
Gyro & Accel Filter Biases
Mag & Accel Disturbance Estimation
Onboard Magnetic Hard & Soft Iron Estimation
World Magnetic & Gravity Model

3.3.1 INS Kalman Filter

The INS Kalman filter consists of an Extended Kalman filter which nominally runs at the NavFilter rate (default 200 Hz). The INS Kalman filter uses the accelerometer, gyro, GPS, and (at startup) the magnetometer to simultaneously estimate the full quaternion based attitude solution, the position and velocity, as well as the time varying gyro, accelerometer, and barometric pressure sensor biases. The output of the INS Kalman filter is passed to the NavState, allowing for the attitude, position, and velocity to be made available at the higher fixed rate of the NavState.

3.3.2 Vector Processing Engine

The Vector Processing Engine (VPE) is a collection of sophisticated algorithms which provide real-time monitoring and simultaneous estimation of the attitude as well as the uncertainty of the input measurements used by the attitude estimation algorithm. By estimating its own input measurement uncertainty the VPE is capable of providing significantly improved performance when compared to traditional statically tuned Kalman Filters. The estimated measurement uncertainty is used to in real-time adaptively tune the onboard Kalman filters. This adaptive tuning eliminates the need in most cases for the user to perform any custom filter tuning for different applications.

3.3.3 AHRS Kalman Filter

The AHRS Kalman filter consists of an EKF which nominally runs at the NavFilter Rate (default 200 Hz). The AHRS Kalman filter simultaneously estimates the full quaternion based attitude as well as the time varying gyro bias. The quaternion based attitude estimation eliminates any potential gimbal lock issues incurred at high pitch angles, which can be problematic for Euler-angle based AHRS algorithms. The real-time estimation of the gyro bias allows for the removal of small perturbations in the gyro bias which occur over time due to random walk.

3.3.4 Hard/Soft Iron Estimator

The NavFilter subsystem also includes a separate EKF which provides real-time estimation of the local magnetic hard and soft iron distortions. Hard and soft iron distortions are local magnetic field distortions created by nearby ferrous material which moves with the sensor (attached to the same vehicle or rigid-body as the sensor). These ferrous materials distort the direction and magnitude of the local measured magnetic field, thus negatively impacting the ability of an AHRS to reliably and accurately estimate heading based on the magnetometer measurements. To remove the unwanted effect of these materials, a hard & soft iron calibration needs to be performed which requires rotating the sensor around in multiple circles while collecting magnetic data for off-line calculation of the magnetic hard & soft iron calibration coefficients. This calibration can be very time consuming, and might not be possible for some applications. The onboard hard/soft iron estimator runs in the background without requiring any user intervention. For many applications this simplifies the process for the end user, and allows for operation in environments where the hard/soft iron may change slowly over time. While the onboard hard/soft iron estimator runs in the background by default, it can be turned off by the user if desired in the Magnetic Calibration Control Register.

3.3.5 World Magnetic Model

The world magnetic model (WMM) is a large spatial-scale representation of the Earth's magnetic field. The internal model used on the VN-100 is consistent with the current WMM2016 model which consist of a spherical-harmonic expansion of the magnetic potential of the geomagnetic field generated in the Earth's core. By default the world magnetic model on the VN-100 is turned off, allowing the user to directly set the reference magnetic field strength. Alternatively the world magnetic model can be manually used to calculate the magnetic field strength for a given latitude, longitude, altitude, and date which is then subsequently used as the fixed magnetic field reference strength. Control of the world magnetic model is performed using the Reference Vector Configuration Register.

3.3.6 World Gravity Model

The world gravity model (WGM) is a large spatial-scale representation of the Earth's gravity potential as a function of position on the globe. The internal model used on the VN-100 is consistent with the Earth Gravity Model (EGM96), which consist of a spherical-harmonic expansion of the Earth's geopotential. By default the world gravity model on the VN-100 is turned off, allowing the user to directly set the reference gravity vector. Control of the world gravity model is performed using the Reference Vector Configuration Register.

3.4 Vector Processing Engine

The Vector Processing Engine (VPE) is a collection of sophisticated algorithms which provide real-time monitoring and simultaneous estimation of the attitude as well as the uncertainty of the input measurements used by the attitude estimation algorithm. By estimating its own input measurement uncertainty the VPE is capable of providing significantly improved performance when compared to a traditional statically tuned EKF AHRS attitude estimation algorithm. The estimated measurement uncertainty is used too in real-time at the NavFilter rate (default 200 Hz) adaptively tune the attitude estimation Kalman filter. This adaptive tuning eliminates the need in most cases for the user to perform any custom filter tuning for different applications. It also provides extremely good disturbance rejection capabilities, enabling the VN-100 in most cases to reliably estimate attitude even in the presence of vibration, short-term accelerations, and some forms of magnetic disturbances.

3.4.1 Adaptive Filtering

The VPE employs adaptive filtering techniques to significantly reduce the effect of high frequency disturbances in both magnetic and acceleration. Prior to entering the attitude filter, the magnetic and acceleration measurements are digitally filtered to reduce high frequency components typically caused by electromagnetic interference and vibration. The level of filtering applied to the inputs is dynamically altered by the VPE in real-time. The VPE calculates the minimal amount of digital filtering required in order to achieve specified orientation accuracy and stability requirements. By applying only the minimal amount of filtering necessary, the VPE reduces the amount of delay added to the input signals. For applications that have very strict latency requirements, the VPE provides the ability to limit the amount of adaptive filtering performed on each of the input signals.

3.4.2 Adaptive Tuning

Kalman filters employ coefficients that specify the uncertainty in the input measurements which are typically used as "tuning parameters" to adjust the behavior of the filter. Normally these tuning parameters have to be adjusted by the engineer to provide adequate performance for a given application. This tuning process can be ad-hoc, time consuming, and application dependent. The VPE employs adaptive tuning logic which provides on-line estimation of the uncertainty of each of the input signals during operation. This uncertainty is then applied directly to the onboard attitude estimation Kalman filter to correctly account for the uncertainty of the inputs. The adaptive tuning reduces the need for manual filter tuning.

3.4.3 VPE Heading Modes

The VectorNav VPU provides three separate heading modes. Each mode controls how the VPE interprets the magnetic measurements to estimate the heading angle. The three modes are described in detail in the following sections.

Absolute Heading Mode

In Absolute Heading Mode the VPE will assume that the principal long-term DC component of the measured magnetic field is directly related to the earth's magnetic field. As such only short term magnetic disturbances will be tuned out. This mode is ideal for applications that are free from low frequency (less than ~ 1Hz) magnetic disturbances and/or require tracking of an absolute heading. Since this mode assumes that the Earth's magnetic field is the only long-term magnetic field present, it cannot handle constant long-term magnetic disturbances which are of the same order of magnitude as the Earth's magnetic field and cannot be compensated for by performing a hard/soft iron calibration. From the sensor's perspective a constant long-term magnetic disturbance will be indistinguishable from the contribution due to the Earth's magnetic field, and as such if present it will inevitably result in a loss of heading accuracy.

> If a magnetic disturbance occurs due to an event controlled by the user, such as the switching on/off of an electric motor, an absolute heading can still be maintained if the device is notified of the presence of the disturbance.

> To correctly track an absolute heading you will need to ensure that the hard/soft iron distortions remains well characterized.

Absolute Heading Mode Advantages

- Provides short-term magnetic disturbance rejection while maintaining absolute tracking of the heading relative to the fixed Earth.

Absolute Heading Mode Disadvantages

- If the magnetic field changes direction relative to the fixed Earth, then its direction will need to be updated using the reference vector register in order to maintain an accurate heading reference.
- Hard/Soft iron distortions that are not properly accounted for will induce heading errors proportional to the magnitude of the hard/soft iron distortion. In some cases this could be as high as 30-40 degrees.

Relative Heading Mode

In Relative Heading mode the VPE makes no assumptions as to the long term stability of the magnetic field present. In this mode the VPE will attempt to extract what information it reasonably can from the magnetic measurements in order to maintain an accurate estimate of the gyro bias. The VPE will constantly monitor the stability of the magnetic field and when it sees that its direction is reasonably stable, the VPE will maintain a stable heading estimate. Over long periods of time under conditions where the magnetic field direction changes frequently, in Relative Heading mode it is possible for the VN-100 to accumulate some error in its reported heading relative to true North. In this mode the VPE will not attempt to correct for this accumulated heading error.

Relative Heading mode does not assume that the Earth's magnetic field is the only long-term magnetic field present. As such this mode is capable of handling a much wider range of magnetic field disturbances while still maintaining a stable attitude solution. Relative Heading mode should be used in situations where the most important requirement is for the attitude sensor is to maintain a stable attitude solution which minimizes the effect of gyro drift while maintaining a stable and accurate pitch and roll solution. Since the Relative Heading mode assumes that other magnetic disturbances can be present which are indistinguishable from the Earth's field, Relative Heading mode cannot always ensure that the calculated heading is always referenced to Earth's magnetic north.

> Use the Relative Heading mode for applications where the stability of the estimated heading is more important than the long-term accuracy relative to true magnetic North. In general, the Relative Heading mode provides better magnetic disturbance rejection that the Absolute Heading mode.

Relative Heading Mode Advantages

- Capable of handling short-term and long-term magnetic interference.
- Can handle significant errors in the hard/soft iron while still maintaining a stable heading and gyro bias estimate.

Relative Heading Mode Disadvantages

- Unable to maintain heading estimate relative to true North in environments with frequent long-term magnetic field disturbances.

Indoor Heading Mode

The Indoor Heading mode was designed to meet the needs of applications that require the enhanced magnetic disturbance rejection capability of the Relative Heading mode, yet desire to maintain an absolute heading reference over long periods of time. The Indoor Heading mode extends upon the capabilities of the Relative Heading mode by making certain assumptions as to the origin of the measured magnetic fields consistent with typical indoor environments.

In any environment the measured magnetic field in 3D space is actually the combination of the Earth's magnetic field plus the contribution of other local magnetic fields created by nearby objects containing ferromagnetic materials. For indoor environments this becomes problematic due to the potential close proximity to objects such as metal desk and chairs, speakers, rebar in the concrete floor, and other items which either distort or produce their own magnetic field. The strength of these local magnetic fields are position dependent, and if the strength is on the same order of magnitude as that of the Earth's magnetic field, directly trusting the magnetic measurements to determine heading can lead to inaccurate heading estimates.

While in Indoor Heading mode the VPE inspects the magnetic measurements over long periods of time, performing several different tests on each measurement to quantify the likelihood that the measured field is free of the influence of any position dependent local magnetic fields which would distort the magnetic field direction. Using this probability the VPE then estimates the most likely direction of the Earth's magnetic field and uses this information to correct for the heading error while the device is in motion.

Indoor Heading Mode Advantages

- Capable of handling short-term and long-term magnetic interference
- Can handle significant errors in the hard/soft iron while still maintaining a stable heading and gyro bias estimate.
- Capable of maintaining an accurate absolute heading over extended periods of time.

Indoor Heading Mode Disadvantages

- Measurement repeatability may be worse than Relative Mode during periods when the VPE corrects for known errors in absolute heading.

Overview of Heading Modes

A summary of the different types of disturbances handled by each magnetic mode is summarized in the table below.

Capabilities	Absolute Heading	Relative Heading	Indoor Mode	Capabilities
Handle high frequency magnetic disturbances greater than 1Hz?	Yes	Yes	Yes	Handle high frequency magnetic disturbances greater than 1Hz?
Handle constant disturbances lasting less than a few seconds?	Yes	Yes	Yes	Handle constant disturbances lasting less than a few seconds?
Handle constant disturbances lasting longer than a few seconds?	No	Yes	Yes	Handle constant disturbances lasting longer than a few seconds?

3.4.4 VPE Adaptive Filtering and Tuning Settings

The VPE actively employs both adaptive filtering and adaptive tuning techniques to enhance performance in conditions of dynamic motion and magnetic and acceleration disturbances. The VPE provides the ability to modify the amount of adaptive filtering and tuning applied on both the magnetometer and the accelerometer. In many cases the VPE can be used as is without any need to adjust these settings. For some applications higher performance can be obtained by adjusting the amount of adaptive filtering and tuning performed on the inputs. For both the magnetometer and the accelerometer the following settings are provided.

Static Measurement Uncertainty

The static gain adjusts the level of uncertainty associated with either the magnetic or acceleration measurement when no disturbances are present. The level of uncertainty associated with the measurement will directly influence the accuracy of the estimated attitude solution. The level of uncertainty in the measurement will also determine how quickly the attitude filter will correct for errors in the attitude when they are observed. The lower the uncertainty, the quicker it will correct for observed errors.

- This parameter can be adjusted from 0 to 10.
- Zero places no confidence (or infinite uncertainty) in the sensor, thus eliminating its effect on the attitude solution.
- Ten places full confidence (minimal uncertainty) in the sensor and assume that its measurements are always 100% correct.

Adaptive Tuning Gain

The adaptive tuning stage of the VPE monitors both the magnetic and acceleration measurements over an extended period of time to estimate the time-varying level of uncertainty in the measurement. The adaptive tuning gain directly scales either up or down this calculated uncertainty.

- This parameter can be adjusted from 0 to 10.

- The minimum value of zero turns off all adaptive tuning.
- The maximum value of 10 applies several times the estimated level of uncertainty.

Adaptive Filtering Gain

The adaptive filtering stage of the VPE monitors both the magnetic and acceleration measurements to determine if large amplitude high frequency disturbances are present. If so then a variable level of filtering is applied to the inputs in order to reduce the amplitude of the disturbance down to acceptable levels prior to inputting the measurement into the attitude filter. The advantage of the adaptive filtering is that it can improve accuracy and eliminate jitter in the output attitude when large amplitude AC disturbances are present. The disadvantage to filtering is that it will inherently add some delay to the input measurement. The adaptive filtering gain adjusts the maximum allowed AC disturbance amplitude for the measurement prior to entering the attitude filter. The larger the allowed disturbance, the less filtering that will be applied. The smaller the allowed disturbance, the more filtering will be applied.

- This parameter can be adjusted from 0 to 10.
- The minimum value of zero turns off all adaptive filtering.
- The maximum value of 10 will apply maximum filtering.

Keep in mind that regardless of this setting, the adaptive filtering stage will apply only the minimal amount of filtering necessary to get the job done. As such this parameter provides you with the ability to set the maximum amount of delay that you are willing to accept in the input measurement.

3.5 Communication Interface

The VN-100 provides two separate communication interfaces on two separate serial ports.

3.5.1 Serial Interface

The serial interface consists of two physically separate bi-directional UARTs. Each UART supports baud rates from 9600 bps up to a maximum of 921600 bps.

The rugged version includes an onboard TTL to RS-232 level shifter, thus at the 10-pin connector one serial port is offered with RS-232 voltages levels (Serial 1), while the other serial port (Serial 2) remains at 3V TTL logic levels.

> ! It is important to note that the ability to update the firmware using the onboard bootloader is only supported on the serial port 1 interface. It is highly recommended that if serial port 1 is not used for normal operation, a means of accessing it is designed into the product to support future firmware updates.

3.5.2 SPI Interface

The SPI interface consists of a standard 4-wire synchronous serial data link which is capable of high data rates up to 16 Mbps. The VN-100 operates as slave on the bus enabled by the master using the slave select (SPI_CS) line. See the Basic Communication chapter for more information on the operation of the SPI interface.

3.6 Communication Protocol

The VN-100 utilizes a simple command based communication protocol for the serial interface. An ASCII protocol is used for command and register polling, and an optional binary interface is provided for streaming high speed real-time sensor measurements.

3.6.1 Serial ASCII

On the serial interface a full ASCII protocol provides support for all commands, and register polling. The ASCII protocol is very similar to the widely used NMEA 0183 protocol supported by most GPS receivers, and consists of comma delimited parameters printed in human readable text. Below is an example command request and response on the VN-100 used to poll the attitude (Yaw Pitch Roll Register in the Attitude subsystem) using the ASCII protocol.

Example Serial Request

```
$VNRRG,8*4B
```

Example Serial Response

```
$VNRRG,08,-114.314,+000.058,-001.773*5F
```

At the end of this user manual each software subsystem is documented providing a list of all the commands and registers suported by the subsystem on the VN-100. For each command and register an example ASCII response is given to demonstrating the ASCII formatting.

3.6.2 Serial Binary

The serial interface offers support for streaming sensor measurements from the sensor at fixed rates using user configurable binary output packets. These binary output packets provide a low-overhead means of streaming high-speed sensor measurements from the device minimizing both the required bandwidth and the necessary overhead required to parse the incoming measurements for the host system.

3.6.3 Serial Command Prompt

A simple command prompt is also provided on the serial interface, which provides support for advanced device configuration and diagnostics. The serial command prompt is an optional feature that is designed to provide more detailed diagnostic view of overall system performance than is possible using normal command & register structure. It is strictly intended to be used by a human operator, who can type commands to the device using a simple serial terminal, and is not designed to be used programmatically. Each software subsystem described in the software module chapters provides information on the diagnostic commands supported by the serial command prompt at the end of each subsystem section.

3.7 System Error Codes

In the event of an error, the VN-100 will output $VNERR, followed by an error code. The possible error codes are listed in the table below with a description of the error.

Error Codes

Error Name	Code	Description
Hard Fault	1	If this error occurs, then the firmware on the VN-100 has experienced a hard fault exception. To recover from this error the processor will force a restart, and a discontinuity will occur in the serial output. The processor will restart within 50 ms of a hard fault error.
Serial Buffer Overflow	2	The processor's serial input buffer has experienced an overflow. The processor has a 256 character input buffer.
Invalid Checksum	3	The checksum for the received command was invalid.
Invalid Command	4	The user has requested an invalid command.
Not Enough Parameters	5	The user did not supply the minimum number of required parameters for the requested command.
Too Many Parameters	6	The user supplied too many parameters for the requested command.
Invalid Parameter	7	The user supplied a parameter for the requested command which was invalid.
Invalid Register	8	An invalid register was specified.
Unauthorized Access	9	The user does not have permission to write to this register.
Watchdog Reset	10	A watchdog reset has occurred. In the event of a non-recoverable error the internal watchdog will reset the processor within 50 ms of the error.
Output Buffer Overflow	11	The output buffer has experienced an overflow. The processor has a 2048 character output buffer.
Insufficient Baud Rate	12	The baud rate is not high enough to support the requested asynchronous data output at the requested data rate.
Error Buffer Overflow	255	An overflow event has occurred on the system error buffer.

3.8 Checksum / CRC

The serial interface provides the option for either an 8-bit checksum or a 16-bit CRC. In the event neither the checksum nor the CRC is needed, both can be turned off by the user. Refer to the Communication Protocol Control Register for details on disabling the checksum/CRC.

3.8.1 Checksum Bypass

When communicating with the sensor using a serial terminal, the checksum calculation can be bypassed by replacing the hexadecimal digits in the checksum with uppercase X characters. This works for both the 8-bit and 16-bit checksum. An example command to read register 1 is shown below using the checksum bypass feature.

```
$VNRRG,1*XX
```

3.8.2 8-bit Checksum

The 8-bit checksum is an XOR of all bytes between, but not including, the dollar sign ($) and asterisk (*). All comma delimiters are included in the checksum calculation. The resultant checksum is an 8-bit number and is represented in the command as two hexadecimal characters. The C function snippet below calculates the correct checksum.

Example C Code

```c
// Calculates the 8-bit checksum for the given byte sequence.
unsigned char calculateChecksum(unsigned char data[], unsigned int length)
{
    unsigned int i;
    unsigned char cksum = 0;

    for(i=0; i<length; i++){
        cksum ^= data[i];
    }

    return cksum;
}
```

3.8.3 16-bit CRC

For cases where the 8-bit checksum doesn't provide enough error detection, a full 16-bit CRC is available. The VN-100 uses the CRC16-CCITT algorithm. The resultant CRC is a 16-bit number and is represented in the command as four hexadecimal characters. The C function snippet below calculates the correct CRC.

Example C Code

```
// Calculates the 16-bit CRC for the given ASCII or binary message.
unsigned short calculateCRC(unsigned char data[], unsigned int length)
{
  unsigned int i;
  unsigned short crc = 0;

  for(i=0; i<length; i++){
    crc  = (unsigned char)(crc >> 8) | (crc << 8);
    crc ^= data[i];
    crc ^= (unsigned char)(crc & 0xff) >> 4;
    crc ^= crc << 12;
    crc ^= (crc & 0x00ff) << 5;
  }

  return crc;
}
```

4 User Configurable Binary Output Messages

The VN-100 supports 3 separate user configurable binary output messages available on the serial interface. Each message can be configured by the user to contain any of the available output measurement types from the IMU, NavState, or NavFilter subsystems. The device can be configured to asynchronously output each message at a fixed rate based upon a divisor of the IMU internal sampling rate (IMU Rate).

4.1 Available Output Types

All real-time measurements either measured or estimated by the VN-100 are available using the user output messages. The different output types are organized into 6 separate output groups. The first group is a combination of the most common outputs from the remaining groups. The other groups are shown below.

Binary Outputs

Time	IMU	Attitude
•TimeStartup	•Status	•Status
•TimeGps	•UncompMag	•YawPitchRoll
•GpsTow	•UncompAccel	•Quaternion
•GpsWeek	•UncompAngularRate	•DCM
•TimeSyncIn	•Temp	•MagNed
•TimeGpsPps	•Pres	•AccelNed
•TimeUTC	•DeltaTheta	•LinearAccelBody
•SyncInCnt	•DeltaVel	•LinearAccelNed
•SyncOutCnt	•Mag	•YprU
•TimeStatus	•Accel	
	•AngularRate	
	•SatFlags	

4.2 Configuring the Output Types

Configuration of the 3 output messages is performed using the User Output Configuration Registers (Register 75-77). There are 3 separate configuration registers, one for each available output message. The Binary Output Register 1-3 in the System subsystem section describes in more detail the format for these registers. In each of these configuration registers the user can select which output types they want the message to include by specifying the OutputGroup and the OutputFields parameters.

4.2.1 OutputGroup

The OutputGroup and OutputFields parameters consist of variable length arguments to allow conciseness where possible and expandability where necessary.

The OutputGroup parameter consists of one or more bytes which are used to identify the Binary Output Groups from which data will be selected for output (see OutputField parameter). Each 8-bit byte consists of seven group selection bits (Bit 0 through Bit 6) and an extension bit (Bit 7). The extension bit in each byte is used to indicate the presence of a following continuation byte to select additional (higher-numbered) groups. The first byte selects Groups 1-7 (with bit offsets 0-6, respectively), the second byte (if present) selects Groups 8-14, and so on. The sequence of group selection bytes will always end with a byte whose extension bit is not set.

Name	Bit Offset	Description
Output Group 1	0	Common Group
Output Group 2	1	Time Group
Output Group 3	2	IMU Group
Output Group 5	4	Attitude Group

Output group 4, 6, & 7 are not used on the VN-100. The bits for these unused output groups must be set to zero.

Groups 8-14 are not used, however they are reserved for use in future firmware versions.

4.2.2 OutputFields

The OutputField parameter consists of a series of one or more 16-bit words per selected output group (see OutputGroup parameter) which are used to identify the selected output fields for that group. The first series of one or more words corresponds to the fields for the first selected group, followed by a series of word(s) for the next selected group, and so on. Each 16-bit word consists of 15 group selection bits (Bit 0 through Bit 14) and an extension bit (Bit 15). The extension bit in each word is used to indicate the presence of a following continuation word to select additional (higher-numbered) output fields for the current group. The first word corresponding to a specific group selects fields 1-15 (with bit offsets 0-14, respectively), the second word (if present) selects fields 16-30, and so on. Each sequence of field selection words corresponding to a selected output group ends with a word whose extension bit is not set, and is then followed by a sequence of words for the next selected group (if any).

Below is a list of the available output fields for each output group.

Bit Offset	Group 1 Common	Group 2 Time	Group 3 IMU	Group 5 Attitude
0	TimeStartup	TimeStartup	ImuStatus	VpeStatus
1	Reserved		UncompMag	YawPitchRoll
2	TimeSyncIn		UncompAccel	Quaternion
3	YawPitchRoll		UncompGyro	DCM
4	Quaternion	TimeSyncIn	Temp	MagNed
5	AngularRate		Pres	AccelNed
6	Reserved		DeltaTheta	LinearAccelBody
7	Reserved	SyncInCnt	DeltaVel	LinearAccelNed
8	Accel	SyncOutCnt	Mag	YprU
9	Imu	TimeStatus	Accel	
10	MagPres		Gyro	
11	DeltaTheta			
12	VpeStatus			
13	SyncInCnt			
14				
15				

4.2.3 Setup the Configuration Register

Once you have determined the desired outputs for your output messages, you will need to configure the User Output Message Configuration Registers (Register 75 – 77). These registers are described in detail under the Binary Output Register 1-3 in the System subsystem section, however for reference the format of the register is shown below.

Binary Output Register 1-3

Register ID :	75-77 **Access :** Read / Write
Comment :	These registers allow the user to construct a custom output message that contains a collection of desired estimated states and sensor measurements.
Size (Bytes):	6-22
Example Response:	$VNWRG,75,2,4,1,8*XX

Offset	Name	Format	Unit	Description
0	AsyncMode	uint16	-	Selects whether the output message should be sent out on the serial port(s) at a fixed rate. 0 = None. User message is not automatically sent out either serial port. 1 = Message is sent out serial port 1 at a fixed rate. 2 = Message is sent out serial port 2 at a fixed rate. 3 = Message is sent out both serial ports at a fixed rate.
2	RateDivisor	uint16	-	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ***ImuRate*** which is nominally 800Hz. For example to have the sensor output at 50Hz you would set the Divisor equal to 16.
4+N	OutputGroup(N)	uint8	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N+2*M	OutputField(1)	uint16	-	Selects which output data fields are active within the selected output groups.

In the offset column above the variable N is the number of output group bytes. If data is requested from only groups 1-7, there will be only one output group present (N=1). If data is requested from an output group of 9-14, then two output groups bytes will be present.

The number of OutputFields present must be equal to the number of output groups selected in the OutputGroup byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two OutputField parameters present (M = 2).

If the number of OutputFields is inconsistent with the number of OutputGroups selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

If the user attempts to turn on more data than it is possible to send out at the current baud rate, the unit will resond with a insufficient baud rate error.

To turn off the binary output it is recommended to set the AsyncMode = 0.

4.2.4 Example Case 1 – Selecting outputs from only the Common Group

For many applications you might be able to get by with only the output types available in the common group. For these situations the configuration of the output message is simple. Suppose only the following information shown below is desired.

Bit Offset	Group 1 Common
0	TimeStartup
3	YawPitchRoll
5	AngularRate

For this example we will assume that the data will be polled using serial port 2 at 50 Hz.

To configure this output message you would send the following command to the VN-100.

```
$VNWRG,75,2,16,01,0029*XX
```

Now let's dissect this command to see what is actually being set:

Field	Value	Description
Header	$VN	ASCII message header
Command	WRG	Write register command
Register ID	75	Register 75 (Config register for first output message)
AsyncMode	2	Message set to output on serial port 2.
RateDivisor	16	Divisor = 16. If the *ImuRate* = 800Hz then, the message output rate will be (800 / 16 = 50 Hz).
OutputGroup	01	Groups = 0x01. (Binary group 1 enabled)
GroupField 1	0029	Group 1 Field = 0x0029. In binary 0x0029 = 0b00101001. The active bits correspond to the following active output fields: Bit 0 – TimeStartup Bit 3 – YawPitchRoll Bit 5 - AngularRate
Checksum	XX	Payload terminator and checksum. XX instructs the VN-100 to bypass the checksum evaluation. This allows us to manually type messages in a serial terminal without needing to calculate a valid checksum.
End Line	\r\n	Carriage return and line feed. Terminates the ASCII message.

4.2.5 Example Case 2 – Outputs from multiple Output Groups without extention bits

This example case demonstrates how to select multiple output fields from more than one output group. Assume that the following bold output types are desired:

Bit Offset	Group 1 Common	Group 3 IMU	Group 5 Attitude
0	TimeStartup		
1			
2		UncompAccel	Quaternion
3		UncompAngularRate	

4 MagNed

Also assume that you want the message to stream at 50 Hz over serial port 1.

To configure this output message you would send the following command to the VN-100.

```
$VNWRG,75,1,16,15,0001,000C,0014*XX
```

Now let's dissect this command to see what is actually being set:

Field	Value	Description
Header	$VN	ASCII message header
Command	WRG	Write register command
Register ID	75	Register 75 (Config register for first output message)
AsyncMode	1	Message sent on serial port 1.
RateDivisor	16	Divisor = 16. If the ***ImuRate*** = 800Hz then, the message output rate will be (800 / 16 = 50 Hz).
OutputGroup	15	Groups = 0x15. In binary 0x15 = 0x00010101. The active bits correspond to the following active output groups: Bit 0 – Common Bit 2 – Imu Bit 4 - Attitude
GroupField 1	0001	Group 1 Field = 0x0001. In binary 0x0001 = 0b00000001. The active bits correspond to the following active output fields: Bit 0 – TimeStartup
GroupField 2	000C	Group 2 Field = 0x000C. In binary 0x000C = 0b00001100. The active bits correspond to the following active output fields: Bit 2 – UncompAccel Bit 3 – UncompGyro
GroupField 3	0014	Group 3 Field = 0x0014. In binary 0x0014 = 0b00010100. The active bits correspond to the following active output fields: Bit 2 – Qtn Bit 4 – MagNed
Checksum	XX	Payload terminator and checksum. XX instructs the VN-200 to bypass the checksum evaluation. This allows us to manually type messages in a serial terminal without needing to calculate a valid checksum.
End Line	\r\n	Carriage return and line feed. Terminates the ASCII message.

4.3 Serial Output Message Format

The binary output message packets on the serial interface consist of a simple message header, payload, and a 16-bit CRC. An example packet is shown below for reference. The header is variable length depending upon the number of groups active in the message.

	Header						Payload				CRC	
Field	Sync	Groups	Group Field 1		Group Field 2		Payload				CRC	
Byte Offset	0	1	2	3	4	5	6	7	...	N	N+1	N+2
Type	u8	u8	u16		u16		Variable				u16	

4.3.1 Sync Byte

The sync byte is the first byte in the header. Its value will always be equal to 0xFA.

4.3.2 Groups

The Group and Group Field parameters consist of variable length arguments to allow conciseness where possible and expandability where necessary.

The Group parameter consists of one or more bytes which are used to identify the Binary Output Groups from which data will be selected for output (see OutputField parameter). Each 8-bit byte consists of seven group selection bits (Bit 0 through Bit 6) and an extension bit (Bit 7). The extension bit in each byte is used to indicate the presence of a following continuation byte to select additional (higher-numbered) groups. The first byte selects Groups 1-7 (with bit offsets 0-6, respectively), the second byte (if present) selects Groups 8-14, and so on. The sequence of group selection bytes will always end with a byte whose extension bit is not set. The various groups are shown below.

Name	Bit Offset	Description
Binary Group 1	0	General Purpose Group.
Binary Group 2	1	Time and Event Count Group.
Binary Group 3	2	Inertial Measurement Unit Group.
Binary Group 4	3	Not used. Must be set to zero.
Binary Group 5	4	AHRS Group.
Binary Group 6	5	Not used. Must be set to zero.
Binary Group 7	6	Not used. Must be set to zero.
Binary Group 8	7	Not used. Must be set to zero.

Groups 8-14 are not used, however they are reserved for use in future firmware versions.

4.3.3 Group Fields

The Group Field parameter consists of a series of one or more 16-bit words per selected output group which are used to identify the selected output fields for that group. The first series of one or more words corresponds to the fields for the first selected group, followed by a series of word(s) for the next selected group, and so on. Each 16-bit word consists of 15 group selection bits (Bit 0 through Bit 14) and an extension bit (Bit 15). The extension bit in each word is used to indicate the presence of a following continuation word to select additional (higher-numbered) output fields for the current group. The first word corresponding to a specific group selects fields 1-15 (with bit offsets 0-14, respectively), the second word (if present) selects fields 16-30, and so on. Each sequence of field selection words corresponding to a selected output group ends with a word whose extension bit is not set, and is then followed by a sequence of words for the next selected group (if any).

The group fields represent which output types have been selected in the active binary groups. The number of group fields in the header will depend upon how many groups are active in the message. The number of group fields present in the header will always be equal to the number of active bits in the group byte. When parsing the binary packet you can count the number of active bits present in the group byte, and then you can assume that this number of group fields will be present in the header. For example if only binary group 1 is selected (Group Byte = 0x01), then only one Group field will be present in the header, thus the header will be 4 bytes in length. If both binary group 1 and 3 are active (Group Byte = 0x05), then two Group field elements will be present in the header (4 bytes), thus the header in this case will be 6 bytes in length.

4.3.4 Payload

The payload will consist of the output data selected based upon the bits selected in the group byte and the group field bytes. All output data in the payload section consist of the active outputs selected for binary group 1, followed by the active outputs selected for binary group 2, and so forth. No padding bytes are used between output fields.

4.3.5 CRC

The CRC consists of a 16-bit CRC of the packet. The CRC is calculated over the packet starting just after the sync byte in the header (not including the sync byte) and ending at the end of the payload. More information about the CRC algorithm and example code for how to perform the calculation is shown in the Checksum/CRC section of the Software Architecture chapter. The CRC is selected such that if you compute the 16-bit CRC starting with the group byte and include the CRC itself, a valid packet will result in 0x0000 computed by the running CRC calculation over the entire packet. This provides a simple way of detecting packet corruption by simply checking to see if the CRC calculation of the entire packet (not including the sync byte) results in zero.

4.3.6 Payload Length

When parsing the packet you will need to know the length of the payload (in bytes) in order to know where the packet ends in the data stream. In order to reduce the overhead of the packet header length, the length of the payload is not included in the header. Instead it should be derived based upon determining the type of data present in the packet. All output data types are fixed length, thus the total length of the payload can be determined based upon inspection of the group byte and the group field bytes. In most applications you will likely only use a few binary output types, thus hard coding the payload length in your parser is the easiest approach. If you want to develop a more generic parser that can handle all available data output types supported by the VN-100, the easiest approach is to use a table lookup. Below is a table with the payload size (in bytes) for all available output types.

Binary Output Payload Length In Bytes

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7
Field 1	8	8	2	8	2	2	8
Field 2	8	8	12	8	12	24	8
Field 3	8	8	12	2	16	24	2
Field 4	12	2	12	1	36	12	1
Field 5	16	8	4	1	12	12	1
Field 6	12	8	4	24	12	12	24
Field 7	24	8	16	24	12	12	24
Field 8	12	4	12	12	12	12	12
Field 9	12	4	12	12	12	12	12
Field 10	24	1	12	12	12	4	12
Field 11	20	1	12	4	28	4	4
Field 12	28	0	2	4	24	68	4
Field 13	2	0	40	2	0	64	2
Field 14	4	0	0	28	0	0	28
Field 15	8	0	0	2+(N*8)	0	0	2+(N*8)
Field 16	0	0	0	12+(N*28)	0	0	12+(N*28)

The messages highlighted in red in the above table are variable length messages. The size of these messages will be dependent upon the number of packets present. See the description of the fields in the appropriate group section below for more information on how to determine the length of these packets.

The above lookup table can be implemented in C as shown below using a simple 2D array. Assuming you are only using group 1 through 7, with support for 16 fields per group, then this lookup table could be implimented using an 8x16 array of bytes consuming only 128 bytes of memory. With the exception of the SatInfo and RawMeas fields in the GPS group, all other fields have a fixed length. The two variable length fields can be handled seperately with a case statement. For these two fields the lookup table contains the length of the fixed portion of the variable length packet (2 for the SatInfo and 12 for the RawMeas fields).

Example Code

```
// 2D array to determine the payload length for a binary output packet. The first
// index of the array is the group number, and the second index
// is the group field index. Both indices are assumed to be zero based.

const unsigned char groupLen[7][16] =
{
{8,  8,  8,  12, 16, 12, 24, 12, 12, 24, 20, 28, 2,  4,   8,  0},  //Group 1
{8,  8,  8,  2,  8,  8,  8,  4,  4,  1,  1,  0,  0,  0,   0,  0},  //Group 2
{2,  12, 12, 12, 4,  4,  16, 12, 12, 12, 12, 2,  40, 0,   0,  0},  //Group 3
{8,  8,  2,  1,  1,  24, 24, 12, 12, 12, 4,  4,  2,  28,  2,  12}, //Group 4
{2,  12, 16, 36, 12, 12, 12, 12, 12, 12, 28, 24, 0,  0,   0,  0},  //Group 5
{2,  24, 24, 12, 12, 12, 12, 12, 12, 4,  4,  68, 64, 0,   0,  0},  //Group 6
{8,  8,  2,  1,  1,  24, 24, 12, 12, 12, 4,  4,  2,  28,  2,  12}, //Group 7
};
```

4.3.7 Example Cases

To help you better understand how the binary protocol works, the next two sections provide an overview of how the binary output packets are formed for two separate example cases.

Example Case 1

For example 1 we will assume that only binary group 1 is active, and only the yaw, pitch, and roll output is active within this binary group. In this case the header will have the following form.

	Header				Payload												CRC	
Field	Sync	Group	Group 1 Fields		YawPitchRoll												CRC	
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Byte Value (Hex)	FA	01	08	00	93	50	2E	42	83	3E	F1	3F	48	B5	04	BB	92	88
Type	u8	u8	u16		float				float				float				u16	
Value	0xFA	1	8		0x422E5093 +43.578686 (Yaw)				0x3FF13E83 +1.8847202 (Pitch)				0xBB04B548 -2.0249654e-3 (Roll)				0x9288	

Example Case 2

For the second example case we will assume that both binary group 1 and 3 are active. In binary group 1, the Ypr output is selected, and in binary group 3, the Temp output is selected.

Header						
Field	Sync	Group	Group 1 Fields		Group 3 Fields	
Byte Offset	0	1	2	3	4	5
Byte Value (Hex)	FA	05	08	00	01	00
Type	u8	u8	u16		u16	
Value	0xFA	0x05	0x08		0x01	

	Payload																CRC	
Field	YawPitchRoll												Temp				CRC	
Byte Offset	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Byte Value (Hex)	A4	15	02	42	4D	DF	EB	3F	F6	1A	36	BE	BF	2D	A4	41	A8	3A
Type	float				float				float				float				u16	
Value	0x420215A4 +32.521133 (Yaw)				0X3FEBDF4D +1.8427521 (Pitch)				0XBE361AF6 -1.7783722e-1 (Roll)				0X41A42DBF +20.522337 (Temp)				0XA83A	

4.4 Binary Group 1 – Common Outputs

Binary group 1 contains a wide assortment of commonly used data required for most applications. All of the outputs found in group 1 are also present in the other groups. In this sense, group 1 is a subset of commonly used outputs from the other groups. This simplifies the configuration of binary output messages for applications that only require access to the commonly used data found in group 1. For these applications you can hard code the group field to 1, and not worry about implemented support for the other binary groups. Using group 1 for commonly used outputs also has the advantage of reducing the overall packet size, since the packet length is dependent upon the number of binary groups active.

Binary Group 1

Name	Bit Offset	Description
TimeStartup	0	Time since startup.
Reserved	1	Reserved. Not used on this product.
TimeSyncIn	2	Time since last SyncIn trigger.
Ypr	3	Estimated attitude as yaw pitch and roll angles.
Qtn	4	Estimated attitude as a quaternion.
AngularRate	5	Compensated angular rate.
Reserved	6	Reserved. Not used on this product.
Reserved	7	Reserved. Not used on this product.
Accel	8	Estimated acceleration (compensated). (Body)
Imu	9	Calibrated uncompensated gyro and accelerometer measurements.
MagPres	10	Calibrated magnetic (compensated), temperature, and pressure measurements.
DeltaTheta	11	Delta time, theta, and velocity.
InsStatus	12	INS status.
SyncInCnt	13	SyncIn count.
Reserved	14	Reserved. Not used on this product.
Resv	15	Reserved for future use. Should be set to zero.

4.4.1 Time Startup

The system time since startup measured in nano seconds. The time since startup is based upon the internal TXCO oscillator for the MCU. The accuracy of the internal TXCO is +/- 20ppm (-40C to 85C). This field is equivalent to the TimeStartup field in group 2.

	TimeStartup							
Byte Offset	0	1	2	3	4	5	6	7
Type	uint64							

4.4.2 TimeSyncIn

The time since the last SyncIn trigger event expressed in nano seconds. This field is equivalent to the TimeSyncIn field in group 2.

	TimeSyncIn							
Byte Offset	0	1	2	3	4	5	6	7
Type	uint64							

4.4.3 YawPitchRoll

The estimated attitude Yaw, Pitch, and Roll angles measured in degrees. The attitude is given as a 3,2,1 Euler angle sequence describing the body frame with respect to the local North East Down (NED) frame. This field is equivalent to the YawPitchRoll field in group 5.

	YawPitchRoll											
	yaw				pitch				roll			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.4.4 Quaternion

The estimated attitude quaternion. The last term is the scalar value. The attitude is given as the body frame with respect to the local North East Down (NED) frame. This field is equivalent to the Quaternion field in group 5.

	Quaternion															
	qtn[0]				qtn[1]				qtn[2]				qtn[3]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Type	float				float				float				float			

4.4.5 AngularRate

The estimated angular rate measured in rad/s. The angular rates are compensated by the onboard filter bias estimates. The angular rate is expressed in the body frame. This field is equivalent to the AngularRate field in group 3.

	AngularRate											
	rate[0]				rate[1]				rate[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.4.6 Accel

The estimated acceleration in the body frame, given in m/s^2. This acceleration includes gravity and has been bias compensated by the onboard INS Kalman filter. This field is equivalent to the Accel field in group 3.

	Accel											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.4.7 Imu

The uncompensated IMU acceleration and angular rate measurements. The acceleration is given in m/s^2, and the angular rate is given in rad/s. These measurements correspond to the calibrated angular rate and acceleration measurements straight from the IMU. The measurements have not been corrected for bias offset by the onboard Kalman filter. These are equivalent to the UncompAccel and UncompGyro fields in group 3.

	Imu																							
	accel[0]				accel[1]				accel[2]				rate[0]				rate[1]				rate[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Type	float				float				float				float				float				float			

4.4.8 MagPres

The compensated magnetic, temperature, and pressure measurements from the IMU. The magnetic measurement is given in Gauss, and has been corrected for hard/soft iron corrections (if enabled). The temperature measurement is given in Celsius. The pressure measurement is given in kPa. This field is equivalent to the Mag, Temp, and Pres fields in group 3.

	MagPres																			
	mag[0]				mag[1]				mag[2]				temp				pres			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Type	float				float				float				float				float			

4.4.9 DeltaThetaVel

The delta time, angle, and velocity measurements. The delta time (dtime) is the time interval that the delta angle and velocities are integrated over. The delta theta (dtheta) is the delta rotation angles incurred due to rotation, since the last time the values were outputted by the device. The delta velocity (dvel) is the delta velocity incurred due to motion, since the last time the values were outputted by the device. These delta angles and delta velocities are calculated based upon the onboard conning and sculling integration performed onboard the sensor at the IMU rate (default 800Hz). The integration for both the delta angles and velocities are reset each time either of the values are either polled or sent out due to a scheduled asynchronous ASCII or binary output. This field is equivalent to the DeltaTheta and DeltaVel fields in group 3 with the inclusion of the additional delta time parameter.

	DeltaThetaVel															
	dtime				dtheta[0]				dtheta[1]				dtheta[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Type	float				float				float				float			

	DeltaThetaVel (continued)											
	dvel[0]				dvel[1]				dvel[2]			
Byte Offset	16	17	18	19	20	21	22	23	24	25	26	27
Type	float				float				float			

4.4.10 SyncInCnt

The number of SyncIn trigger events that have occurred. This field is equivalent to the SyncInCnt field in group 2.

	SyncInCnt			
Byte Offset	0	1	2	3
Type	u32			

4.5 Binary Group 2 – Time Outputs

Binary group 2 provides all timing and event counter related outputs. Some of these outputs (such as the TimeGps, TimePps, and TimeUtc), require either that the internal GPS to be enabled, or an external GPS must be present.

Binary Group 2

Name	Bit Offset	Description
TimeStartup	0	Time since startup.
Reserved	1	Reserved. Not used on this product.
Reserved	2	Reserved. Not used on this product.
Reserved	3	Reserved. Not used on this product.
TimeSyncIn	4	Time since last SyncIn trigger.
Reserved	5	Reserved. Not used on this product.
Reserved	6	Reserved. Not used on this product.
SyncInCnt	7	SyncIn trigger count.
SyncOutCnt	9	SyncOut trigger count.
TimeStatus	10	Time valid status flags.
Resv	11-15	Reserved for future use. Should be set to zero.

4.5.1 TimeStartup

The system time since startup measured in nano seconds. The time since startup is based upon the internal TXCO oscillator for the MCU. The accuracy of the internal TXCO is +/- 20ppm (-40C to 85C).

	TimeStartup							
Byte Offset	0	1	2	3	4	5	6	7
Type	uint64							

4.5.2 TimeSyncIn

The time since the last SyncIn event trigger expressed in nano seconds.

	TimeSyncIn							
Byte Offset	0	1	2	3	4	5	6	7
Type	uint64							

4.5.3 SyncInCnt

The number of SyncIn trigger events that have occurred.

	SyncInCnt			
Byte Offset	0	1	2	3
Type	u32			

4.5.4 SyncOutCnt

The number of SyncOut trigger events that have occurred.

	SyncOutCnt			
Byte Offset	0	1	2	3
Type	u32			

4.6 Binary Group 3 – IMU Outputs

Binary group 3 provides all outputs which are dependent upon the measurements collected from the onboard IMU, or an external IMU (if enabled).

Binary Group 3

Name	Bit Offset	Description
ImuStatus	0	Reserved for future use.
UncompMag	1	Uncompensated magnetic measurement.
UncompAccel	2	Uncompensated acceleration measurement.
UncompGyro	3	Uncompensated angular rate measurement.
Temp	4	Temperature measurement.
Pres	5	Pressure measurement.
DeltaTheta	6	Delta theta angles.
DeltaV	7	Delta velocity.
Mag	8	Compensated magnetic measurement.
Accel	9	Compensated acceleration measurement.
AngularRate	10	Compensated angular rate measurement.
Resv	11-15	Reserved for future use. Should be set to zero.

4.6.1 ImuStatus

Status is reserved for future use. Not currently used in the current code, as such will always report 0.

	ImuStatus	
Byte Offset	0	1
Type	u16	

4.6.2 UncompMag

The IMU magnetic field measured in units of Gauss, given in the body frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), and the user compensation, however it is not compensated by the onboard Hard/Soft Iron estimator.

	UncompMag											
	mag[0]				mag[1]				mag[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.3 UncompAccel

The IMU acceleration measured in units of m/s^2, given in the body frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), however it is not compensated by any dynamic calibration such as bias compensation from the onboard INS Kalman filter.

	UncompAccel											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.4 UncompGyro

The IMU angular rate measured in units of rad/s, given in the body frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), however it is not compensated by any dynamic calibration such as the bias compensation from the onboard AHRS/INS Kalman filters.

	UncompGyro											
	gyro[0]				gyro[1]				gyro[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.5 Temp

The IMU temperature measured in units of Celsius.

	Temp			
Byte Offset	0	1	2	3
Type	float			

4.6.6 Pres

The IMU pressure measured in kilopascals. This is an absolute pressure measurement. Typical pressure at sea level would be around 100 kPa.

	Pres			
Byte Offset	0	1	2	3
Type	float			

4.6.7 DeltaTheta

The delta theta (dtheta) is the delta rotation angles incurred due to rotation, since the last time the values were output by the device. The delta angles are calculated based upon the onboard conning and sculling integration performed onboard the sensor at the IMU sampling rate (nominally 800Hz). The delta time (dtime) is the time interval that the delta angle and velocities are integrated over. The integration for the delta angles are reset each time the values are either polled or sent out due to a scheduled asynchronous ASCII or binary output. Time is given in sections. Delta angles are given in degrees.

	DeltaTheta															
Fields	dtime				dtheta[0]				dtheta[1]				dtheta[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Type	float				float				float				float			

4.6.8 DeltaV

The delta velocity (dvel) is the delta velocity incurred due to motion, since the last time the values were output by the device. The delta velocities are calculated based upon the onboard conning and sculling integration performed onboard the sensor at the IMU sampling rate (nominally 800Hz). The integration for the delta velocities are reset each time the values are either polled or sent out due to a scheduled asynchronous ASCII or binary output. Delta velocity is given in meters per second.

	DeltaVel											
Fields	dvel[0]				dvel[1]				dvel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.9 Mag

The IMU compensated magnetic field measured units of Gauss, and given in the body frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), the user compensation, and the dynamic calibration from the onboard Hard/Soft Iron estimator.

	Mag											
	mag[0]				mag[1]				mag[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.10 Accel

The compensated acceleration measured in units of m/s^2, and given in the body frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), the user compensation, and the dynamic bias compensation from the onboard INS Kalman filter.

	Accel											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.6.11 AngularRate

The compensated angular rate measured in units of rad/s, and given in the body frame. This measurement is compensated by the static calibration (individual factor calibration stored in flash), the user compensation, and the dynamic bias compensation from the onboard INS Kalman filter.

	AngularRate											
	gyro[0]				gyro[1]				gyro[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7 Binary Group 5 – Attitude Outputs

Binary group 5 provides all estimated outputs which are dependent upon the estimated attitude solution. The attitude will be derived from either the AHRS or the INS, depending upon which filter is currently active and tracking. All of the fields in this group will only be valid if the AHRS/INS filter is currently enabled and tracking.

Binary Group 5

Name	Bit Offset	Description
VpeStatus	0	VPE Status
Ypr	1	Yaw Pitch Roll
Qtn	2	Quaternion
DCM	3	Directional Cosine Matrix
MagNed	4	Compensated magnetic (NED)
AccelNed	5	Compensated acceleration (NED)
LinearAccelBody	6	Compensated linear acceleration (no gravity)
LinearAccelNed	7	Compensated linear acceleration (no gravity) (NED)
YprU	8	Yaw Pitch Roll uncertainty
Resv	9-15	Reserved for future use. Should be set to zero.

4.7.1 VpeStatus

The VPE status bitfield.

	VpeStatus	
Byte Offset	0	1
Type	u16	

VpeStatus BitField

Name	Bit Offset	Format	Unit	Description
AttitudeQuality	0	2 bits	-	Provides an indication of the quality of the attitude solution.
GyroSaturation	2	1 bit	-	At least one gyro axis is currently saturated.
GyroSaturationRecovery	3	1 bit	-	Filter is in the process of recovering from a gyro saturation event.
MagDisturbance	4	2 bit	-	A magnetic DC disturbance has been detected. 0 – No magnetic disturbance 1 to 3 – Magnetic disturbance is present.
MagSaturation	6	1 bit	-	At least one magnetometer axis is currently saturated.
AccDisturbance	7	2 bit	-	A strong acceleration disturbance has been detected. 0 – No acceleration disturbance. 1 to 3 – Acceleration disturbance has been detected.
AccSaturation	9	1 bit	-	At least one accelerometer axis is currently saturated.
Reserved	10	1 bit	-	Reserved for internal use. May change state at run-time.
KnownMagDisturbance	11	1 bit	-	A known magnetic disturbance has been reported by the user and the magnetometer is currently tuned out.
KnownAccelDisturbance	12	1 bit	-	A known acceleration disturbance has been reported by the user and the accelerometer is currently tuned out.
Reserved	13	3 bits	-	Reserved for future use.

Table 1 - AttitudeQuality Field

Value	Description
0	Excellent
1	Good
2	Bad
3	Not tracking

4.7.2 YawPitchRoll

The estimated attitude Yaw, Pitch, and Roll angles measured in degrees. The attitude is given as a 3,2,1 Euler angle sequence describing the body frame with respect to the local North East Down (NED) frame.

	YawPitchRoll											
	yaw				pitch				roll			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7.3 Quaternion

The estimated attitude quaternion. The last term is the scalar value. The attitude is given as the body frame with respect to the local North East Down (NED) frame.

	Quaternion															
	qtn[0]				qtn[1]				qtn[2]				qtn[3]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Type	float				float				float				float			

4.7.4 DCM

The estimated attitude directional cosine matrix given in column major order. The DCM maps vectors from the North East Down (NED) frame into the body frame.

	Dcm																							
Fields	dcm[0]				dcm[1]				dcm[2]				dcm[3]				dcm[4]				dcm[5]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Type	float				float				float				float				float				float			

	Dcm (continued)											
Fields	dcm[6]				dcm[7]				dcm[8]			
Byte Offset	24	25	26	27	28	29	30	31	32	33	34	35
Type	float				float				float			

4.7.5 MagNed

The current estimated magnetic field (Gauss), given in the North East Down (NED) frame. The current attitude solution is used to map the measurement from the measured body frame to the inertial (NED) frame. This measurement is compensated by both the static calibration (individual factory calibration stored in flash), and the dynamic calibration such as the user or onboard Hard/Soft Iron compensation registers.

	MagNed											
	mag[0]				mag[1]				mag[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7.6 AccelNed

The estimated acceleration (with gravity) reported in m/s^2, given in the North East Down (NED) frame. The acceleration measurement has been bias compensated by the onboard INS filter. This measurement is attitude dependent, since the attitude is used to map the measurement from the body frame into the inertial (NED) frame. If the device is stationary and the INS filter is tracking, the measurement should be nominally equivalent to the gravity reference vector in the inertial frame (NED).

	AccelNed											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7.7 LinearAccelBody

The estimated linear acceleration (without gravity) reported in m/s^2, and given in the body frame. The acceleration measurement has been bias compensated by the onboard INS filter, and the gravity component has been removed using the current gravity reference vector model. This measurement is attitude dependent, since the attitude solution is required to map the gravity reference vector (known in the inertial NED frame), into the body frame so that it can be removed from the measurement. If the device is stationary and the onboard INS filter is tracking, the measurement nominally will read 0 in all three axes.

	LinearAccelBody											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7.8 LinearAccelNed

The estimated linear acceleration (without gravity) reported in m/s^2, and given in the North East Down (NED) frame. This measurement is attitude dependent as the attitude solution is used to map the measurement from the body frame into the inertial (NED) frame. This acceleration measurement has been bias compensated by the onboard INS filter, and the gravity component has been removed using the current gravity reference vector estimate. If the device is stationary and the onboard INS filter is tracking, the measurement nominally will read 0 in all three axes.

	LinearAccelNed											
	accel[0]				accel[1]				accel[2]			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

4.7.9 YprU

The estimated attitude (Yaw, Pitch, Roll) uncertainty (1 Sigma), reported in degrees.

	YprU											
	yaw				pitch				roll			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11
Type	float				float				float			

The estimated attitude (YprU) field is not valid when the INS Scenario mode in the INS Basic Configuration register is set to AHRS mode. See the INS Basic Configuration Register in the INS section for more details.

5 System Module

5.1 Commands

5.1.1 Read Register Command

This command allows the user to read any of the registers on the VN-100. The only required parameter is the ID of the register to be read. The first parameter of the response will contain the same register ID followed by a variable number of parameters. The number of parameters and their formatting is specific to the requested register. Refer to the appropriate register listed in the subsystem sections for details on this formatting. If an invalid register is requested, an error code will be returned.

Example Read Register Command

Example Command	Message
UART Command	$VNRRG,5*46
UART Response	$VNRRG,5,9600*65

5.1.2 Write Register Command

This command is used to write data values to a specified register on the VN-100 module. The ID of the register to be written to is the first parameter. This is followed by the data values specific to that register. Refer to the appropriate register listed in the subsystem sections for details on this formatting. If an invalid register is requested, an error code will be returned.

Example Write Register Command

Example Command	Message
UART Command	$VNWRG,5,9600*60
UART Response	$VNWRG,5,9600*60

5.1.3 Write Settings Command

This command will write the current register settings into non-volatile memory. Once the settings are stored in non-volatile (Flash) memory, the VN-100 module can be power cycled or reset, and the register will be reloaded from non-volatile memory.

Example Write Settings Command

Example Command	Message
UART Command	$VNWNV*57
UART Response	$VNWNV*57

Due to limitations in the flash write speed the write settings command takes ~ 500ms to complete. Any commands that are sent to the sensor during this time will be responded to after the operation is complete.

The sensor must be stationary when issuing a Write Settings Command otherwise a Reset command must also be issued to prevent the Kalman Filter from diverging during the write settings process.

5.1.4 Restore Factory Settings Command

This command will restore the VN-100 module's factory default settings and will reset the module. There are no parameters for this command. The module will respond to this command before restoring the factory settings.

Example Restore Factory Settings Command

Example Command	Message
UART Command	$VNRFS*5F
UART Response	$VNRFS*5F

5.1.5 Reset Command

This command will reset the module. There are no parameters required for this command. The module will first respond to the command and will then perform a reset. Upon a reset all registers will be reloaded with the values saved in non-volatile memory. If no values are stored in non-volatile memory, the device will default to factory settings. Also upon reset the VN-100 will re-initialize its Kalman filter, thus the filter will take a few seconds to completely converge on the correct attitude and correct for gyro bias.

Example Reset Command

Example Command	Message
UART Command	$VNRST*4D
UART Response	$VNRST*4D

5.1.6 Firmware Update Command

This command is used to enter the boot loader for performing firmware updates. Upon receiving this command on serial port 1, the VN-100 will enter into firmware reprogramming mode. The easiest method of updating firmware is to use one of the VectorNav Firmware Update Tools. If you wish however to incorporate the ability to update the firmware into your own system, the protocol and procedure for updating the firmware is outlined in the **AN013 Firmware Update Protocol** application note.

Example Firmware Update Command

Example Command	Message
UART Command	$VNFWU*XX
UART Response	$VNFWU*XX

Firmware updates are only supported on serial port 1. If you plan on using either serial port 2 as your primary means of communicating with the sensor, it is recommended that you also provide support in your design to communicate with the sensor using serial port 1 to facilitate firmware updates.

5.1.7 Serial Command Prompt Command

This command allows you to enter into the command prompt mode on either serial port. The command mode supports a wide range of diagnostics and configuration options that go beyond the abilities of the normal read/write configuration register interface.

Example Command Prompt Command

Example Command	Message
UART Command	$VNCMD*XX
UART Response	$VNCMD*XX

5.1.8 Asynchronous Output Pause Command

This command allows the user to temporarily pause the asynchronous outputs on the given serial port. When paused, both the ASCII and the 3 binary asynchronous output messages will temporarily stop outputting from the device on the serial port for which this command is received. The state of the asynchronous output register and the binary output configuration registers will not be changed when the asynchronous outputs are paused. This command is useful when you want to send configuration commands to the VN-100, but do not want to deal with the additional overhead of having to parse a constant stream of asynchronous output messages while waiting for the response to your configuration commands. It is also useful when you want to type commands to the device from a serial command prompt. The below example commands demonstrate how to pause and resume asynchronous outputs.

Example Asynchronous Pause/Resume Commands

Example Command	Message
Pause Async Outputs	$VNASY,0*XX
Resume Async Outputs	$VNASY,1*XX

5.1.9 Binary Output Poll Command

This command allows you to poll the sensor measurements available in the binary output protocol.

Example Command Prompt Command

Example Command	Message
UART Command	$VNBOM,N*XX Where N is 1-3 to selecte the appropriate binary output register.
UART Response	Responds with requested binary packet.

To use the Binary Output Poll command you will first need to configure the desired output packet using the Binary Output Register 1-3. If you wish only to poll this output, set the rate in the Binary Output Register to 0. When you wish to poll the measurement send the command $VNBOM,N*XX where the N is the number of the appropriate binary output register.

5.2 Configuration Registers

5.2.1 User Tag Register

User Tag				
Register ID :	0		**Access :**	Read / Write
Comment :	User assigned tag register. Any values can be assigned to this register. They will be stored to flash upon issuing a write settings command.			
Size (Bytes):	20			
Example Response:	$VNRRG,00,SENSOR_A14*52			

Offset	Name	Format	Unit	Description
0	Tag	char	-	User defined tag register. Up to 20 bytes or characters. If a string with more than 20 characters is given, then the string will be truncated to the first 20.

Only printable ASCII characters are allowed for the user tag register.

Allowable characters include any character in the hexadecimal range of 0x20 to 0x7E, excluding 0x24 (‘$’), 0x2C (‘,’), and 0x2A (‘*’). The use any other character will result in an invalid parameter error code returned. This restriction is required to ensure that the value set in the user tag register remains accessible using the serial ASCII protocol.

5.2.2 Model Number Register

Model Number				
Register ID :	1		**Access :**	Read Only
Comment :	Model Number			
Size (Bytes):	24			
Example Response:	$VNRRG,01,VN-310*58			

Offset	Name	Format	Unit	Description
0	Product Name	char	-	Product name. Max 24 characters.

5.2.3 Hardware Revision Register

Register ID :	2	**Access :**	Read Only
Comment :	Hardware revision.		
Size (Bytes):	4		
Example Response:	$VNRRG,02,1*6C		

Offset	Name	Format	Unit	Description
0	Revision	uint32	-	Hardware revision.

5.2.4 Serial Number Register

Serial Number				
Register ID :	3		**Access :**	Read Only
Comment :	Serial Number			
Size (Bytes):	4			
Example Response:	$VNRRG,03,0100011981*5D			

Offset	Name	Format	Unit	Description
0	SerialNum	uint32	-	Serial Number (32-bit unsigned integer)

5.2.5 Firmware Version Register

Firmware Version Register				
Register ID :	4		**Access :**	Read Only
Comment :	Firmware version.			
Size (Bytes):	4			
Example Response:	$VNRRG,04,0.4.0.0*71			

Offset	Name	Format	Unit	Description
0	Major Version	uint8	-	Major release version of firmware.
1	Minor Version	uint8	-	Minor release version of firmware
2	Feature Version	uint8	-	Feature release version of the firmware.
3	HotFix	uint8	-	Hot fix number. Numbers above 100 are reserved for custom firmware versions.

5.2.6 Serial Baud Rate Register

Serial Baud Rate			
Register ID :	5	**Access :**	Read / Write
Comment :	Serial baud rate.		
Size (Bytes):	4		
Example Command:	$VNWRG,05,115200*58		

Offset	Name	Format	Unit	Description
0	Baud Rate	uint32	-	Serial baud rate.
4	Serial Port	uint8	-	Optional. The serial port to change the baud rate on. If this parameter is not provided then the baud rate will be changed for the active serial port. 1 – Serial Port 1 2 – Serial Port 2

Baud Rate Settings

Acceptable Baud Rates
9600
19200
38400
57600
115200
128000
230400
460800
921600

The serial port parameter in this register is optional. If it is not provided, the baud rate will be changed on the active serial port. The response to this register will include the serial port parameter if the optional parameter is provided. If the second parameter is not provided then the response will not include this parameter.

Upon receiving a baud rate change request, the VN-100 will send the response prior to changing the baud rate.

5.2.7 Async Data Output Type Register

Asynchronous Data Output Type				
	Register ID :	6	**Access :**	Read / Write
	Comment :	Asynchronous data output type.		
	Size (Bytes):	4		
	Example Command:	$VNWRG,06,0*6C		

Offset	Name	Format	Unit	Description
0	ADOR	uint32	-	Output register.
4	Serial Port	uint8	-	Optional. The serial port to change the asynchronous data type on. If this parameter is not provided then the ADOR will be changed for the active serial port. 1 – Serial Port 1 2 – Serial Port 2

This register controls the type of data that will be asynchronously outputted by the module. With this register, the user can specify which data register will be automatically outputted when it gets updated with a new reading. The table below lists which registers can be set to asynchronously output, the value to specify which register to output, and the header of the asynchronous data packet. Asynchronous data output can be disabled by setting this register to zero. The asynchronous data output will be sent out automatically at a frequency specified by the Async Data Output Frequency Register.

The serial port parameter in this register is optional. If it is not provided, the ADOF will be changed on the active serial port. The response to this register will include the serial port parameter if the optional parameter is provided. If the second parameter is not provided, the response will not include this parameter.

Asynchronous Solution Output Settings

Setting	Asynchronous Solution Output Type	Header
0	Asynchronous output turned off	N/A
1	Yaw, Pitch, Roll	VNYPR
2	Quaternion	VNQTN
8	Quaternion, Magnetic, Acceleration and Angular Rates	VNQMR
9	Directional Cosine Orientation Matrix	VNDCM
10	Magnetic Measurements	VNMAG
11	Acceleration Measurements	VNACC
12	Angular Rate Measurements	VNGYR
13	Magnetic, Acceleration, and Angular Rate Measurements	VNMAR
14	Yaw, Pitch, Roll, Magnetic, Acceleration, and Angular Rate Measurements	VNYMR
16	Yaw, Pitch, Roll, Body True Acceleration, and Angular Rates	VNYBA
17	Yaw, Pitch, Roll, Inertial True Acceleration, and Angular Rates	VNYIA
19	IMU Measurements	VNIMU
20	GPS LLA	VNGPS
21	GPS ECEF	VNGPE
22	INS LLA	VNINS
23	INS ECEF	VNINE
30	Delta theta and delta velocity	VNDTV

5.2.8 Async Data Output Frequency Register

Asynchronous Data Output Frequency			
Register ID :	7	**Access :**	Read / Write
Comment :	Asynchronous data output frequency.		
Size (Bytes):	4		
Example Command:	$VNWRG,07,40*59		

Offset	Name	Format	Unit	Description
0	ADOF	uint32	Hz	Output frequency.
4	Serial Port	uint8	-	Optional. The serial port to change the asynchronous data type frequency on. If this parameter is not provided then the ADOF will be changed for the active serial port. 1 – Serial Port 1 2 – Serial Port 2

ADOR Data Rates

Acceptable Data Rates (Hz)
1
2
4
5
10
20
25
40
50
100
200

The serial port parameter in this register is optional. If it is not provided, the ADOF will be changed on the active serial port. The response to this register will include the serial port parameter if the optional parameter is provided. If the second parameter is not provided, the response will not include this parameter.

5.2.9 Synchronization Control

Synchronization Control			
Register ID :	32	**Access :**	Read / Write
Comment :	Contains parameters which allow the timing of the VN-100 to be synchronized with external devices.		
Size (Bytes):	20		
Example Response:	$VNRRG,32,3,0,0,0,6,1,0,100000000,0*6B		

Offset	Name	Format	Unit	Description
0	SyncInMode	uint8	-	Input signal synchronization mode
1	SyncInEdge	uint8	-	Input signal synchronization edge selection
2	SyncInSkipFactor	uint16	-	Input signal trigger skip factor
4	RESERVED	uint32	-	Reserved for future use. Defaults to 0.
8	SyncOutMode	uint8	-	Output synchronization signal mode
9	SyncOutPolarity	uint8	-	Output synchronization signal polarity
10	SyncOutSkipFactor	uint16	-	Output synchronization signal skip factor
12	SyncOutPulseWidth	uint32	ns	Output synchronization signal pulse width
16	RESERVED	uint32	-	Reserved for future use. Defaults to 0.

SyncInMode

The SyncInMode register controls the behavior of the SyncIn event. If the mode is set to COUNT then the internal clock will be used to control the IMU sampling. If SyncInMode is set to IMU then the IMU sampling loop will run on a SyncIn event. The relationship between the SyncIn event and a SyncIn trigger is defined by the SyncInEdge and SyncInSkipFactor parameters. If set to ASYNC then the VN-100 will output asynchronous serial messages upon each trigger event.

SyncIn Mode

Mode	Pin	Value	Description
COUNT	SYNC_IN	3	Count number of trigger events on SYNC_IN.
IMU	SYNC_IN	4	Start IMU sampling on trigger of SYNC_IN.
ASYNC	SYNC_IN	5	Output asynchronous message on trigger of SYNC_IN.
ASYNC3	SYNC_IN	6	Output asynchronous or binary messages configured with a rate of 0 to output on trigger of SYNC_IN.

In ASYNC3 mode messages configured with an output rate = 0 are output each time the appropriate transistion edge of the SyncIn pin is captured according to the edge settings in the SyncInEdge field. Messages configured with output rate > 0 are not affected by the SyncIn puse. This applies to the ASCII Async message set by the Async Data Output Register, the user configurate binary output messages set by the Binary Output Registers, as well as the NMEA messages configured by the NMEA Output Registers.

SyncInEdge

The SyncInEdge register controls the type of edge the signal is set to trigger on. The factory default state is to trigger on a rising edge.

SyncInEdge Mode

Value	Description
0	Trigger on rising edge
1	Trigger on falling edge

SyncInSkipFactor

The SyncInSkipFactor defines how many times trigger edges defined by SyncInEdge should occur prior to triggering a SyncIn event. The action performed on a SyncIn event is determined by the SyncIn mode. As an example if the SyncInSkipFactor was set to 4 and a 1 kHz signal was attached to the SyncIn pin, then the SyncIn event would only occur at 200 Hz.

SyncOutMode

The SyncOutMode register controls the behavior of the SyncOut pin. If this is set to IMU then the SyncOut will start the pulse when the internal IMU sample loop starts. This mode is used to make a sensor the Master in a multi-sensor network array. If this is set to IMU_READY mode then the pulse will start when IMU measurements become available. If this is set to INS mode then the pulse will start when attitude measurements are made available. Changes to this register take effect immediately.

SyncOutMode

Mode	Value	Description
NONE	0	None
IMU_START	1	Trigger at start of IMU sampling
IMU_READY	2	Trigger when IMU measurements are available
INS	3	Trigger when attitude measurements are available

SyncOutPolarity

The SyncOutPolarity register controls the polarity of the output pulse on the SyncOut pin. Changes to this register take effect immediately.

SyncOutPolarity

Value	Description
0	Negative Pulse
1	Positive Pulse

SyncOutSkipFactor

The SyncOutSkipFactor defines how many times the sync out event should be skipped before actually triggering the SyncOut pin.

SyncOutPulseWidth

The SyncOutPulseWidth field controls the desired width of the SyncOut pulse. The default value is 100,000,000 (100 ms).

5.2.10 Communication Protocol Control

Communication Protocol Control			
Register ID :	30	**Access :**	Read / Write
Comment :	Contains parameters that controls the communication protocol used by the sensor.		
Size (Bytes):	7		
Example Response:	$VNRRG,30,0,0,0,0,1,0,1*6C		

Offset	Name	Format	Unit	Description
0	SerialCount	uint8	-	Provides the ability to append a counter or time to the end of the serial asynchronous messages.
1	SerialStatus	uint8	-	Provides the ability to append the status to the end of the serial asynchronous messages.
2	SPICount	uint8	-	Provides the ability to append a counter to the end of the SPI packets.
3	SPIStatus	uint8	-	Provides the ability to append the status to the end of the SPI packets.
4	SerialChecksum	uint8	-	Choose the type of checksum used for serial communications.
5	SPIChecksum	uint8	-	Choose the type of checksum used for the SPI communications.
6	ErrorMode	uint8	-	Choose the action taken when errors are generated.

Serial Count

The SerialCount field provides a means of appending a time or counter to the end of all asynchronous communication messages transmitted on the serial interface. The values for each of these counters come directly from the Synchronization Status Register in the System subsystem.

With the SerialCount field set to OFF a typical serial asynchronous message would appear as the following:

```
$VNYPR,+010.071,+000.278,-002.026*60
```

With the SerialCount field set to one of the non-zero values the same asynchronous message would appear instead as:

```
$VNYPR,+010.071,+000.278,-002.026,T1162704*2F
```

When the SerialCount field is enabled the counter will always be appended to the end of the message just prior to the checksum. The counter will be preceded by the T character to distinguish it from the status field.

SerialCount Field

Mode	Value	Description
NONE	0	OFF
SYNCIN_COUNT	1	SyncIn Counter
SYNCIN_TIME	2	SyncIn Time
SYNCOUT_COUNT	3	SyncOut Counter
GPS_PPS	4	Gps Pps Time

SerialStatus

The SerialStatus field provides a means of tracking real-time status information pertaining to the overall state of the sensor measurements and onboard filtering algorithm. As with the SerialCount, a typical serial asynchronous message would appear as the following:

```
$VNYPR,+010.071,+000.278,-002.026*60
```

With the SerialStatus field set to one of the non-zero values, the same asynchronous message would appear instead as:

```
$VNYPR,+010.071,+000.278,-002.026,S0000*1F
```

When the SerialStatus field is enabled the status will always be appended to the end of the message just prior to the checksum. If both the SerialCount and SerialStatus are enabled then the SerialStatus will be displayed first. The counter will be preceded by the S character to distinguish it from the counter field. The status consists of 4 hexadecimal characters.

SerialStatus

Value	Description
0	OFF
1	VPE Status
2	INS Status

SPICount

The SPICount field provides a means of appending a time or counter to the end of all SPI packets. The values for each of these counters come directly from the Synchronization Status Register.

SPICount Field

Mode	Value	Description
NONE	0	OFF
SYNCIN_COUNT	1	SyncIn Counter
SYNCIN_TIME	2	SyncIn Time
SYNCOUT_COUNT	3	SyncOut Counter
GPS_PPS	4	Gps Pps Time

SPIStatus

The AsyncStatus field provides a means of tracking real-time status information pertaining to the overall state of the sensor measurements and onboard filtering algorithm. This information is very useful in situations where action must be taken when certain crucial events happen such as the detection of gyro saturation or magnetic interference.

SPIStatus

Value	Description
0	OFF
1	VPE Status
2	INS Status

SerialChecksum

This field controls the type of checksum used for the serial communications. Normally the VN-100 uses an 8-bit checksum identical to the type used for normal GPS NMEA packets. This form of checksum however offers only a limited means of error checking. As an alternative a full 16-bit CRC (CRC16-CCITT with polynomial = 0x07) is also offered. The 2-byte CRC value is printed using 4 hexadecimal digits.

SerialChecksum

Value	Description
1	8-Bit Checksum
3	16-Bit CRC

SPIChecksum

This field controls the type of checksum used for the SPI communications. The checksum is appended to the end of the binary data packet. The 16-bit CRC is identical to the one described above for the SerialChecksum.

SPIChecksum

Value	Description
0	OFF
1	8-Bit Checksum
3	16-Bit CRC

ErrorMode

This field controls the type of action taken by the VN-100 when an error event occurs. If the send error mode is enabled then a message similar to the one shown below will be sent on the serial bus when an error event occurs.

```
$VNERR,03*72
```

Regardless of the state of the ErrorMode, the number of error events is always recorded and is made available in the SysErrors field of the Communication Protocol Status Register in the System subsystem.

ErrorMode

Value	Description
0	Ignore Error
1	Send Error
2	Send Error and set ADOR register to OFF

Example Async Messages

The following table shows example asynchronous messages with the AsyncCount and the AsyncStatus values appended to the end.

Example Type	Message
Async Message with AsyncCount Enabled	$VNYPR,+010.071,+000.278,-002.026,T1162704*2F
Async Message with AsyncStatus Enabled	$VNYPR,+010.071,+000.278,-002.026,S0000*1F
Async Message with AsyncCount and AsyncStatus Enabled	$VNYPR,+010.071,+000.278,-002.026,T1162704,S0000*50

5.2.11 Binary Output Register 1

Binary Output Register 1				
Register ID :	75		**Access :**	Read / Write
Comment :	This register allows the user to construct a custom binary output message that contains a collection of desired estimated states and sensor measurements.			
Size (Bytes):	6-22			
Example Response:	$VNWRG,75,2,4,1,8*XX			

Offset	Name	Format	Unit	Description
0	AsyncMode	uint16	-	Selects whether the output message should be sent out on the serial port(s) at a fixed rate. 0 = None. User message is not automatically sent out either serial port. 1 = Message is sent out serial port 1 at a fixed rate. 2 = Message is sent out serial port 2 at a fixed rate. 3 = Message is sent out both serial ports at a fixed rate.
2	RateDivisor	uint16	-	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ***ImuRate*** which is nominally 800Hz. For example to have the sensor output at 50Hz you would set the Divisor equal to 16.
4	OutputGroup	uint16	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N	OutputGroup(N)	uint8	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N+2*M	OutputField(1)	uint16	-	Selects which output data fields are active within the selected output groups.

See the User Configurable Binary Output Messages section for information on the format for the Groups and Group Fields.

In the offset column above the variable N is the number of output group bytes. If data is requested from only groups 1-7, there will be only one output group present (N=1). If data is requested from an output group of 9-14, then two output groups bytes will be present.

The number of OutputFields present must be equal to the number of output groups selected in the OutputGroup byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two OutputField parameters present (M = 2).

If the number of OutputFields is inconsistent with the number of OutputGroups selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

If the user attempts to turn on more data than it is possible to send out at the current baud rate, the unit will resond with a insufficient baud rate error.

To turn off the binary output it is recommended to set the AsyncMode = 0.

5.2.12 Binary Output Register 2

Binary Output Register 2			
Register ID :	76	**Access :**	Read / Write
Comment :	This register allows the user to construct a custom binary output message that contains a collection of desired estimated states and sensor measurements.		
Size (Bytes):	6-22		
Example Response:	$VNWRG,76,2,4,1,8*XX		

Offset	Name	Format	Unit	Description
0	AsyncMode	uint16	-	Selects whether the output message should be sent out on the serial port(s) at a fixed rate. 0 = None. User message is not automatically sent out either serial port. 1 = Message is sent out serial port 1 at a fixed rate. 2 = Message is sent out serial port 2 at a fixed rate. 3 = Message is sent out both serial ports at a fixed rate.
2	RateDivisor	uint16	-	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ***ImuRate*** which is nominally 800Hz. For example to have the sensor output at 50Hz you would set the Divisor equal to 16.
4	OutputGroup	uint16	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N	OutputGroup(N)	uint8	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N+2*M	OutputField(1)	uint16	-	Selects which output data fields are active within the selected output groups.

See the User Configurable Binary Output Messages section for information on the format for the Groups and Group Fields.

In the offset column above the variable N is the number of output group bytes. If data is requested from only groups 1-7, there will be only one output group present (N=1). If data is requested from an output group of 9-14, then two output groups bytes will be present.

The number of OutputFields present must be equal to the number of output groups selected in the OutputGroup byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two OutputField parameters present (M = 2).

If the number of OutputFields is inconsistent with the number of OutputGroups selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

If the user attempts to turn on more data than it is possible to send out at the current baud rate, the unit will resond with a insufficient baud rate error.

To turn off the binary output it is recommended to set the AsyncMode = 0.

5.2.13 Binary Output Register 3

Binary Output Register 3			
Register ID :	77	**Access :**	Read / Write
Comment :	This register allows the user to construct a custom binary output message that contains a collection of desired estimated states and sensor measurements.		
Size (Bytes):	6-22		
Example Response:	$VNWRG,77,2,4,1,8*XX		

Offset	Name	Format	Unit	Description
0	AsyncMode	uint16	-	Selects whether the output message should be sent out on the serial port(s) at a fixed rate. 0 = None. User message is not automatically sent out either serial port. 1 = Message is sent out serial port 1 at a fixed rate. 2 = Message is sent out serial port 2 at a fixed rate. 3 = Message is sent out both serial ports at a fixed rate.
2	RateDivisor	uint16	-	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ***ImuRate*** which is nominally 800Hz. For example to have the sensor output at 50Hz you would set the Divisor equal to 16.
4	OutputGroup	uint16	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N	OutputGroup(N)	uint8	-	Selects which output groups are active in the message. The number of **OutputFields** in this message should equal the number of active bits in the **OutputGroup**.
4+N+2*M	OutputField(1)	uint16	-	Selects which output data fields are active within the selected output groups.

See the User Configurable Binary Output Messages section for information on the format for the Groups and Group Fields.

In the offset column above the variable N is the number of output group bytes. If data is requested from only groups 1-7, there will be only one output group present (N=1). If data is requested from an output group of 9-14, then two output groups bytes will be present.

The number of OutputFields present must be equal to the number of output groups selected in the OutputGroup byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two OutputField parameters present (M = 2).

If the number of OutputFields is inconsistent with the number of OutputGroups selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

If the user attempts to turn on more data than it is possible to send out at the current baud rate, the unit will resond with a insufficient baud rate error.

To turn off the binary output it is recommended to set the AsyncMode = 0.

5.3 Status Registers

5.3.1 Synchronization Status

Synchronization Status				
Register ID :	33		**Access :**	Read / Write
Comment :	Contains status parameters that pertaining to the communication synchronization features.			
Size (Bytes):	12			
Example Response:	$VNRRG,33,2552498,0,0*6A			
Offset	**Name**	**Format**	**Unit**	**Description**
0	SyncInCount	uint32	-	Keeps track of the number of times that the SyncIn trigger even has occured. This register can be used to correlate the attitude to an event on an external system such as a camera or GPS. It is also possible to have the value of this register appended to each asynchronous data packet on the serial bus. This can be done by setting the AsyncStatus field in the Communication Protocol register to 1.
4	SyncInTime	uint32	µs	Keeps track of the amount of time that has elapsed since the last SyncIn trigger event. If the SyncIn pin is connected to the PPS (Pulse Per Second) line on a GPS and the AsyncStatus field in the Communication Protocol Register is set to 1, then each asynchronous measurement will be time stamped relative to the last received GPS measurement.
8	SyncOutCount	uint32	-	Keeps track of the number of times that the SyncOut trigger event has occurred. This register can be used to index subsequent measurement outputs, which is particularly useful when logging sensor data.

Writing zero to the SyncInCount or the SyncOutCount will reset the status counter. Any other value other than zero will not have an effect. The SyncInTime is read only and cannot be reset to zero.

5.4 Factory Defaults

Settings Name	Default Factory Value
User Tag	NULL (Empty string)
Serial Baud Rate	115200
Async Data Output Frequency	40 Hz
Async Data Output Type	INS_LLA
Synchronization Control	3,0,0,0,6,1,0,100000000,0
Communication Protocol Control	0,0,0,0,1,0,1
Binary Output Register 1	0, 0, 0
Binary Output Register 2	0, 0, 0
Binary Output Register 3	0, 0, 0

5.5 Command Prompt

The command prompt provides a fast and simple means of configuring and monitoring the status of the sensor by typing commands to the unit using the serial port.

5.5.1 List Available Commands

Commands for the System subsystem can be accessed by typing in 'system' at the command prompt. To view all available commands, type 'system ?'. Below is a view of a terminal window showing a list of the available commands.

```
system ?

System Module Commands:

Command:    Description:
--------    -----------------------------------------
info        Device specific information such as serial number and firmware version.
comm        Information on the communication interfaces.
errors      Overview of the logged system errors.
reset       Perform a software reset on the unit.
save        Save register settings to flash memory.
restore     Restore register settings to their factory default state.
```

5.5.2 System Info

```
system info

--------------------------------  System Info  ---------------------------------

Hardware:
      Product Model:  VN-310
      Serial Number:  100013003
  MCU Serial Number:  34323439044731322F002100
  Hardware Revision:  2
      Form Revision:  1

Software:
   Firmware Version:  0.3.0.0
           Revision:  691
       Build Number:  2813

--------------------------------------------------------------------------------
```

5.5.3 System Comm

```
system comm

----------------------  System Communication Interfaces  -----------------------

Communication Stats:
  Serial Messages Parsed              : 29
  Spi Messages Parsed                 : 0
  Max Serial RX Buffer Usage          : 0
  Max Serial TX Buffer Usage          : 4
  Max Spi RX Buffer Usage             : 0
  Max Spi TX Buffer Usage             : 0
```

```
  Current Serial 1 TX Bandwidth Usage : 00.0
  Current Serial 2 TX Bandwidth Usage : 49.3

  Max Serial 1 TX Bandwidth Usage : 49.3
  Max Serial 2 TX Bandwidth Usage : 50.5

  Min Serial 1 TX Bandwidth Usage : 00.0
  Min Serial 2 TX Bandwidth Usage : 48.1

--------------------------------------------------------------------------------
```

5.5.4 System Errors

```
system errors

-------------------------------  System Errors  --------------------------------

  Hard Fault Exceptions             : 0
  Serial Input Buffer Overflow      : 0
  Serial Output Buffer Overflow     : 0
  Serial Insufficient Bandwidth     : 0
  Invalid Checksums                 : 6
  Invalid Commands                  : 2
  Input Error - Too Few Parameters  : 0
  Input Error - Too Many Parameters : 0
  Input Error - Invalid Parameter   : 0
  Input Error - Invalid Register    : 0
  Input Error - Unauthorized Access : 2
  Input Error - Watchdog Reset      : 0

--------------------------------------------------------------------------------
```

5.5.5 System Reset

```
system reset
```

5.5.6 System Save

```
system save
```

6 IMU Subsystem

6.1 IMU Measurement Registers

6.1.1 IMU Measurements

This register provides direct access to the calibrated magnetometer, accelerometer, gyro, barometric pressure, and temperature measurements available from the onboard IMU.

IMU Measurements					
Register ID :	54	**Async Header :**	IMU	**Access :**	Read Only
Comment :	Provides the calibrated IMU measurements including barometric pressure.				
Size (Bytes):	44				
Example Read Response:	$VNRRG,54,-02.0841,+00.6045,+02.8911,+00.381,-00.154,-09.657,-00.005683, +00.000262,+00.001475,+21.6,+00099.761*5B				

Offset	Name	Format	Unit	Description
0	MagX	float	Gauss	Uncompensated Magnetic X-axis.
4	MagY	float	Gauss	Uncompensated Magnetic Y-axis.
8	MagZ	float	Gauss	Uncompensated Magnetic Z-axis.
12	AccelX	float	m/s^2	Uncompensated Acceleration X-axis.
16	AccelY	float	m/s^2	Uncompensated Acceleration Y-axis.
20	AccelZ	float	m/s^2	Uncompensated Acceleration Z-axis.
24	GyroX	float	rad/s	Uncompensated Angular rate X-axis.
28	GyroY	float	rad/s	Uncompensated Angular rate Y-axis.
32	GyroZ	float	rad/s	Uncompensated Angular rate Z-axis.
36	Temp	float	C	IMU Temperature.
40	Pressure	float	kPa	Barometric pressure.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNIMU header.

6.1.2 Delta Theta and Delta Velocity

Delta Theta and Delta Velocity				
Register ID :	80	**Async Header:** DTV		**Access :** Read
Comment :	This register contains the output values of the onboard coning and sculling algorithm.			
Size (Bytes):	28			
Example Response:	$VNRRG,80,+0.665016,-000.119,-000.409,-000.025,+000.011,-000.084,-006.702*6A			
Offset	**Name**	**Format**	**Unit**	**Description**
0	DeltaTime	float	sec	Delta time for the integration interval
4	DeltaThetaX	float	deg	Delta rotation vector component in the x-axis.
8	DeltaThetaY	float	deg	Delta rotation vector component in the y-axis.
12	DeltaThetaZ	float	deg	Delta rotation vector component in the z-axis.
16	DeltaVelocityX	float	m/s	Delta velocity vector component in the x-axis.
20	DeltaVelocityY	float	m/s	Delta velocity vector component in the y-axis.
24	DeltaVelocityZ	float	m/s	Delta velocity vector component in the z-axis.

The Delta Theta and Delta Velocity register contains the computed outputs from the onboard coning and sculling algorithm. The coning and sculling integrations are performed at the IMU sample rate (nominally at 800Hz) and reset when the register data is output. If polling this register, the values will represent the delta time, angles, and velocity since the register was last polled. If the Delta Theta/Velocity data is selected for asynchronous output via the Async Data Output Type register (Register 6, type 30), the integrals will be reset each time the data is asynchronously output at the configured rate.

The delta time output contains the length of the time interval over which the deltas were calculated. This can be used to check the interval time or to compute nonlinear “average” rates and accelerations from the integrated values.

The delta theta is output as a principal rotation vector, defined as the product of the unit vector of the principal rotation axis and the principal rotation angle in degrees. For small rotations, a typical use case for delta angles, the principal rotation vector elements may be treated individually as rotations in degrees about the individual sensor axes (in any Euler rotation sequence) with little error.

The delta velocity output provides the integration of the acceleration in the chosen frame, taking into account the coupling effects of any simultaneous rotation experienced.

The coning and sculling algorithm can be configured to operate in multiple frames and with a variety of compensations applied. See the Delta Theta and Delta Velocity Configuration Register in the IMU subsystem for further details.

> You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNDTV header.

6.2 IMU Configuration Registers

6.2.1 Magnetometer Compensation

Magnetometer Compensation				
Register ID :	23		**Access:**	Read / Write
Comment :	Allows the magnetometer to be compensated for hard/soft iron effects.			
Size (Bytes):	48			
Example Command:	$VNRRG,23,1,0,0,0,1,0,0,0,1,0,0,0*73			

Offset	Name	Format	Unit	Description
0	C[0,0]	float	-	
4	C[0,1]	float	-	
8	C[0,2]	float	-	
12	C[1,0]	float	-	
16	C[1,1]	float	-	
20	C[1,2]	float	-	
24	C[2,0]	float	-	
28	C[2,1]	float	-	
32	C[2,2]	float	-	
36	B[0]	float	Gauss	
40	B[1]	float	Gauss	
44	B[2]	float	Gauss	

This register contains twelve values representing the hard and soft iron compensation parameters. The magnetic measurements are compensated for both hard and soft iron using the following model. Under normal circumstances this register can be left in its factory default state. In the event that there are disturbances in the magnetic field due to hard or soft iron effects, then these registers allow for further compensation. These registers can also be used to compensate for significant changes to the magnetometer bias, gain, and axis alignment during installation. Note that this magnetometer compensation is separate from the compensation that occurs during the calibration process at the factory. Setting this register to the default state of an identity matrix and zero offset will not eliminate the magnetometer gain, bias, and axis alignment that occur during factory calibration. These registers only need to be changed from their default values in the event that hard/soft iron compensation needs to be performed, or changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} MX - B0 \\ MY - B1 \\ MZ - B2 \end{Bmatrix}$$

The variables $\{MX, MY, MZ\}$ are components of the measured magnetic field. The {X, Y, Z} variables are the new magnetic field measurements outputted after compensation for hard/soft iron effects. All twelve numbers are represented by single-precision floating points.

6.2.2 Acceleration Compensation

Accelerometer Compensation				
Register ID :	25		**Access :**	Read / Write
Comment :	Allows the accelerometer to be further compensated for scale factor, misalignment, and bias errors.			
Size (Bytes):	48			
Example Command:	$VNRRG,25,1,0,0,0,1,0,0,0,1,0,0,0*75			

Offset	Name	Format	Unit	Description
0	C[0,0]	float	-	
4	C[0,1]	float	-	
8	C[0,2]	float	-	
12	C[1,0]	float	-	
16	C[1,1]	float	-	
20	C[1,2]	float	-	
24	C[2,0]	float	-	
28	C[2,1]	float	-	
32	C[2,2]	float	-	
36	B[0]	float	m/s^2	
40	B[1]	float	m/s^2	
44	B[2]	float	m/s^2	

This register contains twelve values representing the accelerometer compensation parameters. The accelerometer measurements are compensated for changes in bias, gain, and axis alignment that can occur during the installation of the chip on the customer's board using the following model. Under normal circumstances this register can be left in its factory default state. In the event that there are significant changes to the accelerometer bias, gain, and axis alignment during installation, then these registers allow for further compensation. Note that this accelerometer compensation is separate from the compensation that occurs during the calibration process at the factory. Setting this register to the default state of an identity matrix and zero offset will not eliminate the accelerometer gain, bias, and axis alignment that occur during factory calibration. These registers only need to be changed from their default values in the event that changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} AX - B0 \\ AY - B1 \\ AZ - B2 \end{Bmatrix}$$

The variables {AX,AY,AZ} are components of the measured acceleration. The {X, Y, Z} variables are the new acceleration measurements outputted after compensation for changes during sensor mounting. All twelve numbers are represented by single-precision floating points.

6.2.3 Gyro Compensation

Gyro Compensation				
Register ID :	84		**Access :**	Read / Write
Comment :	Allows the gyro to be further compensated for scale factor, misalignment, and bias errors.			
Size (Bytes):	48			
Example Command:	$VNRRG,84,1,0,0,0,1,0,0,0,1,0,0,0*7E			

Offset	Name	Format	Unit	Description
0	C[0,0]	float	-	
4	C[0,1]	float	-	
8	C[0,2]	float	-	
12	C[1,0]	float	-	
16	C[1,1]	float	-	
20	C[1,2]	float	-	
24	C[2,0]	float	-	
28	C[2,1]	float	-	
32	C[2,2]	float	-	
36	B[0]	float	rad/s	
40	B[1]	float	rad/s	
44	B[2]	float	rad/s	

This register contains twelve values representing the gyro compensation parameters. The gyro measurements are compensated for changes in bias, gain, and axis alignment that can occur during the installation of the chip on the customer's board using the following model. Under normal circumstances this register can be left in its factory default state. In the event that there are significant changes to the gyro bias, gain, and axis alignment during installation or during the life of the part; these registers allow for further compensation. Note that this gyro compensation is separate from the compensation that occurs during the calibration process at the factory. Setting this register to the default state of an identity matrix and zero offset will not eliminate the gyro gain, bias, and axis alignment that occur during factory calibration. These registers only need to be changed from their default values in the event that changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} GX - B0 \\ GY - B1 \\ GZ - B2 \end{Bmatrix}$$

The variables $\{GX, GY, GZ\}_{IMU}$ are components of the measured angular rate. The $\{GX, GY, GZ\}_{Comp}$ variables are the new acceleration measurements outputted after compensation for changes during sensor mounting. All twelve numbers are represented by single-precision floating points.

6.2.4 Reference Frame Rotation

Reference Frame Rotation				
Register ID :	26		**Access :**	Read / Write
Comment :	Allows the measurements of the VN-100 to be rotated into a different reference frame.			
Size (Bytes):	36			
Example Response:	$VNRRG,26,1,0,0,0,1,0,0,0,1*6A			

Offset	Name	Format	Unit	Description
0	C[0,0]	float	-	
4	C[0,1]	float	-	
8	C[0,2]	float	-	
12	C[1,0]	float	-	
16	C[1,1]	float	-	
20	C[1,2]	float	-	
24	C[2,0]	float	-	
28	C[2,1]	float	-	
32	C[2,2]	float	-	

This register contains a transformation matrix that allows for the transformation of measured acceleration, magnetic, and angular rates from the body frame of the VN-100 to any other arbitrary frame of reference. The use of this register allows for the sensor to be placed in any arbitrary orientation with respect to the user's desired body coordinate frame. This register can also be used to correct for any orientation errors due to mounting the VN-100 on the user's vehicle or platform.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$$

The variables $\{X, Y, Z\}_B$ are a measured parameter such as acceleration in the body reference frame with respect to the VN-100. The variables $\{X, Y, Z\}_U$ are a measured parameter such as acceleration in the user's frame of reference. The reference frame rotation register thus needs to be loaded with the transformation matrix that will transform measurements from the body reference frame of the VN-100 to the desired user frame of reference.

> ! The matrix C in the Reference Frame Rotation Register must be an orthonormal, right-handed matrix. The sensor will output an error if the tolerance is not within 1e-5. The sensor will also report an error if any of the parameters are greater than 1 or less than -1.

6.2.5 IMU Filtering Configuration

IMU Filtering Configuration			
Register ID :	85	**Access :**	Read / Write
Comment :	Controls the level of filtering performed on the raw IMU measurements.		
Size (Bytes):	15		
Example Response:	$VNRRG,85,0,5,5,5,0,0,3,3,3,0*78		

Offset	Name	Format	Unit	Description
0	MagWindowSize	uint16	-	Number of previous measurements averaged for magnetic measurements.
2	AccelWindowSize	uint16	-	Number of previous measurements averaged for acceleration measurements.
4	GyroWindowSize	uint16	-	Number of previous measurements averaged for gyro measurements.
6	TempWindowSize	uint16	-	Number of previous measurements averaged for temperature measurements.
8	PresWindowSize	uint16	-	Number of previous measurements averaged for pressure measurements.
10	MagFilterMode	uint8	-	Filtering mode for magnetic measurements. See table below for options.
11	AccelFilterMode	uint8	-	Filtering mode for acceleration measurements. See table below for options.
12	GyroFilterMode	uint8	-	Filtering mode for gyro measurements. See table below for options.
13	TempFilterMode	uint8	-	Filtering mode for temperature measurements. See table below for options.
14	PresFilterMode	uint8	-	Filtering mode for pressure measurements. See table below for options.

This register allows the user to configure the FIR filtering what is applied to the IMU measurements. The filter is a uniformly-weighted moving window (boxcar) filter of configurable size. The filtering does not affect the values used by the internal filter, but only the output values.

WindowSize

The WindowSize parameters for each sensor define the number of samples at the IMU rate (default 400Hz) which will be averaged for each output measurement.

FilterMode

The FilterMode parameters for each sensor select which output quantities the filtering should be applied to. Filtering can be applied to either the uncompensated IMU measurements, compensated (HSI and biases compensated by onboard filters, if applicable), or both.

IMU Filtering Modes

Value	Description
0	No Filtering
1	Filtering performed only on raw uncompensated IMU measurements.
2	Filtering performed only on compensated IMU measurements.
3	Filtering performed on both uncompensated and compensated IMU measurements.

6.2.6 Delta Theta and Delta Velocity Configuration

Delta Theta and Delta Velocity Configuration				
Register ID :	82		**Access :**	Read / Write
Comment :	This register contains configuration options for the internal coning/sculling calculations			
Size (Bytes):	6			
Example Response:	$VNRRG,82,0,0,0,0,0*65			

Offset	Name	Format	Unit	Description
0	IntegrationFrame	uint8	-	Output frame for delta velocity quantities
1	GyroCompensation	uint8	-	Compensation to apply to angular rate
2	AccelCompensation	uint8	-	Compensation(s) to apply to accelerations
3	Reserved	uint8	-	Reserved for future use. Should be set to 0.
4	Reserved	uint16	-	Reserved for future use. Should be set to 0.

The Delta Theta and Delta Velocity Configuration register allows configuration of the onboard coning and sculling used to generate integrated motion values from the angular rate and acceleration IMU quantities. The fully-coupled coning and sculling integrals are computed at the IMU sample rate (nominal 400 Hz).

IntegrationFrame

The IntegrationFrame register setting selects the reference frame used for coning and sculling. Note that using any frame other than the body frame will rely on the onboard Kalman filter's attitude estimate. The factory default state is to integrate in the sensor body frame.

IntegrationFrame

Value	Description
0	Body frame
1	NED frame
2	ECEF frame

GyroCompensation

The GyroCompensation register setting selects the compensation to be applied to the angular rate measurements before integration. If bias compensation is selected, the onboard Kalman filter's real-time estimate of the gyro biases will be used to compensate the IMU measurements before integration. The factory default state is to integrate the uncompensated angular rates from the IMU.

GyroCompensation

Value	Description
0	None
1	Bias

AccelCompensation

The AccelCompensation register setting selects the compensation to be applied to the acceleration measurements before integration. If bias compensation is selected, the onboard Kalman filter's real-time estimate of the accel biases will be used to compensate the IMU measurements before integration. The factory default state is to integrate the uncompensated acceleration from the IMU.

AccelCompensation

Value	Description
0	None
1	Bias

6.3 Factory Defaults

Settings Name	Default Factory Value
Magnetometer Compensation	1,0,0,0,1,0,0,0,1,0,0,0
Accelerometer Compensation	1,0,0,0,1,0,0,0,1,0,0,0
Gyro Compensation	1,0,0,0,1,0,0,0,1,0,0,0
Reference Frame Rotation	1,0,0,0,1,0,0,0,1
IMU Filtering Configuration	0,4,4,4,0,0,3,3,3,0
Delta Theta and Delta Velocity Configuration	0,0,0,0,0

6.4 Command Prompt

The command prompt provides a fast and simple means of configuring and monitoring the status of the sensor by typing commands to the unit using the serial port.

6.4.1 List Available Commands

Commands for the System subsystem can be accessed by typing in 'imu' at the command prompt. To view all available commands, type 'imu ?'. Below is a view of a terminal window showing a list of the available commands.

```
imu ?

Imu Module Commands:

Command:    Description:
--------    ----------------------------------------------------------------
info        Imu specific information such as serial number and firmware version.
meas        Current Imu measurement, and run-time statistics.
```

6.4.2 IMU Info

```
imu info

------------------------------  Imu Information  -------------------------------

Magnetometer - HSI Settings (Register 44)
  Mode : Using Onboard

Magnetometer - User HSI Calibration (Register 23)
  +01.000  +00.000  +00.000  +00.000
  +00.000  +01.000  +00.000  +00.000
  +00.000  +00.000  +01.000  +00.000

Magnetometer - Onboard HSI Calibration (Register 47)
  +01.000  +00.000  +00.000  -00.000
  +00.000  +01.000  +00.000  -00.000
  +00.000  +00.000  +01.000  -00.000

Accelerometer - User Calibration (Register 25)
  +01.000  +00.000  +00.000  +00.000
  +00.000  +01.000  +00.000  +00.000
  +00.000  +00.000  +01.000  +00.000

Sensor Self Test: (performed at startup)
  Mag   : Passed
  Accel : Passed
  Gyro  : Passed
  Pres  : Passed

--------------------------------------------------------------------------------
```

6.4.3 IMU Meas

```
imu meas

------------------------------  Imu Measurement  -------------------------------
Current Sensor Measurements:
  Mag X    : -000.866 [Gauss]
  Mag Y    : +001.016 [Gauss]
  Mag Z    : +002.365 [Gauss]
  Acel X   : +004.178 [m/s]
  Acel Y   : -000.637 [m/s]
  Acel Z   : -008.927 [m/s]
  Gyro X   : -000.417 [deg/s]
  Gyro Y   : +000.668 [deg/s]
  Gyro Z   : -001.102 [deg/s]
  Temp     : +027.94 [C]
  Temp Rate: +0.04 [C/min]
  Pres     : +101.36 [kPa]

Current Sensor Noise: (measured over last 5 seconds)
  Sensor    Units     X-Axis    Y-Axis    Z-Axis
  Mag       mGauss    +03.228   +02.934   +04.159
  Accel     mg        +01.854   +02.115   +02.872
  Gyro      deg/s     +0.0631   +0.0544   +0.0580
  Temp      C         +0.0026
  Pres      Pa        +007.36

Minimum Sensor Noise: (since startup)
  Sensor    Units     X-Axis    Y-Axis    Z-Axis
  Mag       mGauss    +02.877   +02.659   +03.673
  Accel     mg        +01.785   +01.966   +02.599
  Gyro      deg/s     +0.0587   +0.0487   +0.0537
  Temp      C         +0.0011
  Pres      Pa        +006.13

Minimum Sensor Measurement: (since startup)
  Sensor    Units     X-Axis    Y-Axis    Z-Axis
  Mag       Gauss     -00.236   +00.244   +00.577
  Accel     g         +00.414   -00.077   -00.949
  Gyro      deg/s     -002.92   -005.33   -002.03
  Temp      C         +27.83
  Pres      kPa       +101.30

Maximum Sensor Measurement: (since startup)
  Sensor    Units     X-Axis    Y-Axis    Z-Axis
  Mag       Gauss     +00.000   +00.271   +00.611
  Accel     g         +00.439   +00.000   +00.000
  Gyro      deg/s     +002.02   +006.44   +000.00
  Temp      C         +28.01
  Pres      kPa       +101.38

Sensor Saturation Events: (since startup)
  Sensor    X-Axis    Y-Axis    Z-Axis
  Mag       0         0         0
  Accel     0         0         0
  Gyro      0         0         0
  Pressure  0
  Temp      0
--------------------------------------------------------------------------------
```

7 Attitude Subsystem

7.1 Commands

7.1.1 Known Magnetic Disturbance Command

This command is used to notify the VN-100 that a magnetic disturbance is present. When the VN-100 receives this command it will tune out the magnetometer and will pause the current hard/soft iron calibration if it is enabled. A single parameter is provided to tell the VN-100 whether the disturbance is present or not.

0 – No Disturbance is present
1 – Disturbance is present

Example Magnetic Disturbance Command

Example Command	Message
UART Command	$VNKMD,1*47
UART Response	$VNKMD,1*47
SPI Command (8 bytes)	08 01 00 00 (shown as hex)
SPI Response (8 bytes)	00 08 01 00 (shown as hex)

7.1.2 Known Acceleration Disturbance Command

This command is used to notify the VN-100 that an acceleration disturbance is present. When the VN-100 receives this command it will tune out the accelerometer. A single parameter is provided to tell the VN-100 whether the disturbance is present or not.

0 – No Disturbance is present
1 – Disturbance is present

Example Acceleration Disturbance Command

Example Command	Message
UART Command	$VNKAD,1*4B
UART Response	$VNKAD,1*4B
SPI Command (8 bytes)	09 01 00 00 (shown as hex)
SPI Response (8 bytes)	00 09 01 00 (shown as hex)

7.1.3 Set Gyro Bias Command

This command will instruct the VN-100 to copy the current gyro bias estimates into register 74. After sending this command you will need to issue the write settings command in the System subsystem to save the state of this register to flash memory. Once saved the VN-100 will use these bias estimates as the initial state at startup.

Example Gyro Bias Command

Example Command	Message

UART Command	$VNSGB*XX
UART Response	$VNSGB*XX
SPI Command (8 bytes)	0C 00 00 00 (shown as hex)
SPI Response (8 bytes)	00 0C 00 00 (shown as hex)

7.2 Measurement Registers

7.2.1 Yaw Pitch Roll

Yaw, Pitch, and Roll

Register ID :	8	**Async Header :**	YPR	**Access :**	Read Only

Comment : Attitude solution as yaw, pitch, and roll in degrees. The yaw, pitch, and roll is given as a 3,2,1 Euler angle rotation sequence describing the orientation of the sensor with respect to the inertial North East Down (NED) frame.

Size (Bytes): 12

Example Response: $VNRRG,8,+006.271,+000.031,-002.000*66

Offset	Name	Format	Unit	Description
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNYPR header.

7.2.2 Attitude Quaternion

Quaternion					
Register ID :	9	**Async Header :**	QTN	**Access :**	Read Only
Comment :	Attitude solution as a quaternion.				
Size (Bytes):	16				
Example Response:	$VNRRG,9,-0.017386,-0.000303,+0.055490,+0.998308*4F				

Offset	Name	Format	Unit	Description
0	Quat[0]	float	-	Calculated attitude as quaternion.
4	Quat[1]	float	-	Calculated attitude as quaternion.
8	Quat[2]	float	-	Calculated attitude as quaternion.
12	Quat[3]	float	-	Calculated attitude as quaternion. Scalar component.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNQTN header.

7.2.3 Yaw, Pitch, Roll, Magnetic, Acceleration, and Angular Rates

Yaw, Pitch, Roll, Magnetic, Acceleration, and Angular Rates					
Register ID :	27	**Async Header :**	YMR	**Access :**	Read Only
Comment :	Attitude solution, magnetic, acceleration, and compensated angular rates.				
Size (Bytes):	48				
Example Response:	$VNRRG,27,+006.380,+000.023,-001.953,+1.0640,-0.2531,+3.0614,+00.005,+00.344,-09.758,-0.001222,-0.000450,-0.001218*4F				

Offset	Name	Format	Unit	Description
0	Yaw	float	deg	Calculated attitude heading angle in degrees.
4	Pitch	float	deg	Calculated attitude pitch angle in degrees.
8	Roll	float	deg	Calculated attitude roll angle in degrees.
12	MagX	float	Gauss	Compensated magnetometer measurement in x-axis.
16	MagY	float	Gauss	Compensated magnetometer measurement in y-axis.
20	MagZ	float	Gauss	Compensated magnetometer measurement in z-axis.
24	AccelX	float	m/s^2	Compensated accelerometer measurement in x-axis.
28	AccelY	float	m/s^2	Compensated accelerometer measurement in y-axis.
32	AccelZ	float	m/s^2	Compensated accelerometer measurement in z-axis.
36	GyroX	float	rad/s	Compensated angular rate in x-axis.
40	GyroY	float	rad/s	Compensated angular rate in y-axis.
44	GyroZ	float	rad/s	Compensated angular rate in z-axis.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNYMR header.

7.2.4 Quaternion, Magnetic, Acceleration and Angular Rates

Quaternion, Magnetic, Acceleration, and Angular Rates					
Register ID :	15	**Async Header :**	QMR	**Access :**	Read Only
Comment :	Attitude solution, magnetic, acceleration, and compensated angular rates.				
Size (Bytes):	52				
Example Response:	$VNRRG,15,-0.017057,-0.000767,+0.056534,+0.998255,+1.0670,-0.2568,+3.0696,-00.019,+00.320,-09.802,-0.002801,-0.001186,-0.001582*65				

Offset	Name	Format	Unit	Description
0	Quat[0]	float	-	Calculated attitude as quaternion.
4	Quat[1]	float	-	Calculated attitude as quaternion.
8	Quat[2]	float	-	Calculated attitude as quaternion.
12	Quat[3]	float	-	Calculated attitude as quaternion. Scalar component.
16	MagX	float	Gauss	Compensated magnetometer measurement in x-axis.
20	MagY	float	Gauss	Compensated magnetometer measurement in y-axis.
24	MagZ	float	Gauss	Compensated magnetometer measurement in z-axis.
28	AccelX	float	m/s^2	Compensated accelerometer measurement in x-axis.
32	AccelY	float	m/s^2	Compensated accelerometer measurement in y-axis.
36	AccelZ	float	m/s^2	Compensated accelerometer measurement in z-axis.
40	GyroX	float	rad/s	Compensated angular rate in x-axis.
44	GyroY	float	rad/s	Compensated angular rate in y-axis.
48	GyroZ	float	rad/s	Compensated angular rate in z-axis.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNQMR header.

7.2.5 Magnetic Measurements

Magnetic Measurements					
Register ID :	17	**Async Header :**	MAG	**Access :**	Read Only
Comment :	Magnetometer measurements.				
Size (Bytes):	12				
Example Response:	$VNRRG,17,+1.0647,-0.2498,+3.0628*66				

Offset	Name	Format	Unit	Description
0	MagX	float	Gauss	Compensated magnetometer measurement in x-axis.
4	MagY	float	Gauss	Compensated magnetometer measurement in y-axis.
8	MagZ	float	Gauss	Compensated magnetometer measurement in z-axis.

> You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNMAG header.

7.2.6 Acceleration Measurements

Acceleration Measurements				
Register ID :	18	**Async Header :** ACC		**Access :** Read Only
Comment :	Acceleration measurements.			
Size (Bytes):	12			
Example Response:	$VNRRG,18,+00.013,+00.354,-09.801*65			

Offset	Name	Format	Unit	Description
0	AccelX	float	m/s^2	Compensated accelerometer measurement in x-axis.
4	AccelY	float	m/s^2	Compensated accelerometer measurement in y-axis.
8	AccelZ	float	m/s^2	Compensated accelerometer measurement in z-axis.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNACC header.

7.2.7 Angular Rate Measurements

Angular Rate Measurements					
Register ID :	19	**Async Header :**	GYR	**Access :**	Read Only
Comment :	Compensated angular rates.				
Size (Bytes):	12				
Example Response:	$VNRRG,19,+0.002112,-0.000362,-0.000876*6C				

Offset	Name	Format	Unit	Description
0	GyroX	float	rad/s	Compensated angular rate in x-axis.
4	GyroY	float	rad/s	Compensated angular rate in y-axis.
8	GyroZ	float	rad/s	Compensated angular rate in z-axis.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNGYR header.

7.2.8 Magnetic, Acceleration and Angular Rates

Magnetic, Acceleration, and Angular Rates					
Register ID :	20	**Async Header :**	MAR	**Access :**	Read Only
Comment :	Magnetic, acceleration, and compensated angular rates.				
Size (Bytes):	36				
Example Response:	$VNRRG,20,+1.0684,-0.2578,+3.0649,-00.005,+00.341,-09.780,-0.000963,+0.000840,-0.000466*64				

Offset	Name	Format	Unit	Description
0	MagX	float	Gauss	Compensated magnetometer measurement in x-axis.
4	MagY	float	Gauss	Compensated magnetometer measurement in y-axis.
8	MagZ	float	Gauss	Compensated magnetometer measurement in z-axis.
12	AccelX	float	m/s^2	Compensated accelerometer measurement in x-axis.
16	AccelY	float	m/s^2	Compensated accelerometer measurement in y-axis.
20	AccelZ	float	m/s^2	Compensated accelerometer measurement in z-axis.
24	GyroX	float	rad/s	Compensated angular rate in x-axis.
28	GyroY	float	rad/s	Compensated angular rate in y-axis.
32	GyroZ	float	rad/s	Compensated angular rate in z-axis.

You can configure the device to output this register at a fixed rate using the Async Data Output Type Register in the System subsystem. Once configured the data in this register will be sent out with the $VNMAR header.

8 Hard/Soft Iron Estimator Subsystem

8.1 Configuration Registers

8.1.1 Magnetometer Calibration Control

Magnetometer Calibration Control				
Register ID :	44		**Access :**	Read / Write
Comment :	Controls the magnetometer real-time calibration algorithm.			
Size (Bytes):	4			
Example Response:	$VNRRG,44,1,2,5*69			

Offset	Name	Format	Unit	Description
0	HSIMode	uint8	-	Controls the mode of operation for the onboard real-time magnetometer hard/soft iron compensation algorithm.
1	HSIOutput	uint8	-	Controls the type of measurements that are provided as outputs from the magnetometer sensor and also subsequently used in the attitude filter.
2	ConvergeRate	uint8	-	Controls how quickly the hard/soft iron solution is allowed to converge onto a new solution. The slower the convergence the more accurate the estimate of the hard/soft iron solution. A quicker convergence will provide a less accurate estimate of the hard/soft iron parameters, but for applications where the hard/soft iron changes rapidly may provide a more accurate attitude estimate. Range: 1 to 5 1 = Solution converges slowly over approximately 60-90 seconds. 5 = Solution converges rapidly over approximately 15-20 seconds.

Table 2 – HSI_Mode Field

Mode	Value	Description
HSI_OFF	0	Real-time hard/soft iron calibration algorithm is turned off.
HSI_RUN	1	Runs the real-time hard/soft iron calibration. The algorithm will continue using its existing solution. The algorithm can be started and stopped at any time by switching between the HSI_OFF and HSI_RUN state.
HSI_RESET	2	Resets the real-time hard/soft iron solution.

Table 3 – HSI_Output Field

Mode	Value	Description
NO_ONBOARD	1	Onboard HSI is not applied to the magnetic measurements.
USE_ONBOARD	3	Onboard HSI is applied to the magnetic measurements.

8.2 Status Registers

8.2.1 Calculated Magnetometer Calibration

Calculated Magnetometer Calibration			
Register ID :	47	**Access :**	Read Only
Comment :	Calculated magnetometer calibration values.		
Size (Bytes):	48		
Example Response:	$VNRRG,46,1,0,0,0,1,0,0,0,1,0,0,0*70		

Offset	Name	Format	Unit	Description
0	C[0,0]	float	-	
4	C[0,1]	float	-	
8	C[0,2]	float	-	
12	C[1,0]	float	-	
16	C[1,1]	float	-	
20	C[1,2]	float	-	
24	C[2,0]	float	-	
28	C[2,1]	float	-	
32	C[2,2]	float	-	
36	B[0]	float	-	
40	B[1]	float	-	
44	B[2]	float	-	

This register contains twelve values representing the calculated hard and soft iron compensation parameters. The magnetic measurements are compensated for both hard and soft iron using the following model.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \cdot \begin{Bmatrix} MX - B0 \\ MY - B1 \\ MZ - B2 \end{Bmatrix}$$

The variables $\{MX, MY, MZ\}$ are components of the measured magnetic field. The {X, Y, Z} variables are the new magnetic field measurements outputted after compensation for hard/soft iron effects.

8.3 Factory Defaults

Settings Name	Default Factory Value
Magnetometer Calibration Control	1,3,5

8.4 Command Prompt

The command prompt provides a fast and simple means of configuring and monitoring the status of the sensor by typing commands to the unit using the serial port.

8.4.1 List Available Commands

Commands for the System subsystem can be accessed by typing in 'hsi' at the command prompt. To view all available commands, type 'hsi ?'. Below is a view of a terminal window showing a list of the available commands.

```
hsi ?

Hard/Soft Iron Estimator Module Commands:

Command:    Description:
--------    ------------------------------------------------------------------
info        Estimator state information and configuration settings.
plotInput   Plot onboard HSI Input.
plotOutput  Plot onboard HSI Output.
```

8.4.2 Info

```
hsi info
-----------------  Hard/Soft Iron Estimator State Information  -----------------
Magnetometer Calibration Control (Register 44):

  HsiMode: Run
  OutMode: Use Onboard
  ConvergeRate: 5

Magnetometer Calibration Status (Register 46):

  LastBin: 0
  NumMeas: 102
  AvgResidual: 0.014
  LastMeas: +0.599  +0.538  +2.910
  Bins[0]: 215
  Bins[1]: 188
  Bins[2]: 135
  Bins[3]: 47
  Bins[4]: 198
  Bins[5]: 231
  Bins[6]: 202

Calculated Magnetometer Calibration (Register 47):

  +00.966  +00.000  +00.000  -00.215
  +00.000  +00.966  +00.000  -00.179
  +00.000  +00.000  +00.966  -00.077


Num Measurements: 358
Filter Run Count: 358
Mag Uncertainty : 0.00
--------------------------------------------------------------------------------
```

8.4.3 PlotInput

```
hsi plotinput

---------------------  HSI Estimator Magnetic Input Plot  ----------------------

                                 Uncalibrated XY

+-------------------+-------------------+-------------------+-------------------+
|                   |          *  *  *  | *   *             |                   |
|                   |      **** *  *  * *      ***          |                   |
|                   | *  *   *   *      |               **  |                   |
|                   *  * *  *    *      |* *      *         |                   |
|                  *** ** *      *   *   **     *            |                   |
|              ***  |   *     * * *   * |                   **   *              |
|               *   |*  *   ** * *      |    *   *  *       |*   *              |
|             *     |     *            *|    *              |                   |
|           *      *|      *  *       * |    *             **       *           |
+---------***-------+-----------*-------*----*--*-----*-----+----------*--------+
|        *          | *            *    *     *             |          **       |
|       * *         *                   | *                 *          ***      |
|     * **   *      |                *  |           *  *    |                   |
|      **           |        *   **     |                   |  *          *     |
|    *            * |      * * *        |               *   |                   |
|    ***            |                  *|      *   * * *    |              *    |
|     *             |      *     *      |            *      |    *     *  *     |
|    *              |   *               |    *              | *             *   |
|     *     **      |                   | *             *   |               *   |
+---*-*-*-----------+---*---------------+--------*-----*----+*------*-------*---+
|   *               | *                 |         *         | *            *    |
|  *   *         *  |                   |                **  |  *                |
|                   |   * *   ** *     *|      *            |        *          |
|   *   ** *        |     *   *        ** *    *  *   *     **    **   *        |
|     *  *          |        *        * |                   |            **     |
|     **            |           *       |         *         |    *              |
|         *     *   *       *           |                 * |           *       |
|     **            |                   |   *             * |          **       |
|      ***   * *    |                   |        *          |         ***       |
+--------**--------*+-------------------+-------------------+-----***-----------+
|        ****  *    |          *    *   * *   *             |*      **          |
|           *  * *  |                   *                   |     **            |
|            ***    |                   |                   |   * *             |
|                *  |   *               |                   * *                 |
|                   |            *      |  *       *        |                   |
|                   |                   |                   |                   |
|                   |  *                |                   |                   |
|                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+

  Plot Center :     +0.000,     +0.000
  Plot Scale  :     +1.042,     +1.042

--------------------------------------------------------------------------------
```

8.4.4 PlotOutput

```
hsi plotoutput

---------------------  HSI Estimator Magnetic Output Plot  ---------------------

                                  Calibrated XY

+-------------------+-------------------+-------------------+-------------------+
|                   |                   |                   |                   |
|                   |                   |                   |                   |
|                   |             **  * *   *    *          |                   |
|                   |          *** *  * * *       **        |                   |
|                   |     * *    *  *   |                * *|                   |
|                   |   *  * * *    *   |  * *      *       |                   |
|                   |  *** ***      *   * ***    *          |                   |
|                  ***      *    ** **  |*                  |**    *            |
|                   *    * *   **  * *  | *    **  *  *     | *    *            |
+-----------------*-+--*------*---------+-------------------+-------------------+
|               *   |         *  *      *      *  *         |*       *          |
|             **    |              *  * | **    *       *   |           **      |
|           * *     |     *             | *  *   *          |*          **      |
|           * *  *  |  *                *             *  *  |             *     |
|          **       |            *  *   |                   |   *               |
|                   |         *   *  *  |                 * |              *    |
|        ***        | *         *       |            * * ** |                   |
|         **        |                   |*        *         |     *     *   *   |
|         *         |      *  *     *   |              *    |   *         *     |
+----------*----**--+-------------------+---*---*---------*-+---------------**--+
|       ***  *      |       *           |          *     *  | *      *      *   |
|       **          |     *             |           *       |  *            *   |
|       *   *       |*                  |                  **   *               |
|        *          |       * *  *** *  | *       *         |         *         |
|         * ****    |        *          |* **    *  *  *    |**     **  *       |
|             *     |            *      *                   |             **    |
|         **        |          *    *   |           *       |      *            |
|          *   *    *  *                |                   *            *      |
|          *       *|                   |     *             *           **      |
+-----------***--*--+--*----------------+----------*--------+-------------------+
|             ***   |             *     |   *               |  *    ***         |
|              *   *|*                 *| *     *           |       **          |
|               ****|                   |                   |      **           |
|                *  |                   |                   | **  *             |
|                   |*      *           |                   |                   |
|                   |                *  |     *      *      |                   |
|                   |      *            |                   |                   |
|                   |                   |                   |                   |
|                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+

  Plot Center :    +0.000,      +0.000
  Plot Scale  :    +0.946,      +0.946

--------------------------------------------------------------------------------
```

9 Velocity Aiding

Velocity aiding provides a method to increase performance of an AHRS sensor for applications where the sensor is subjected to constant accelerations.

9.1 Overview

AHRS Fundamentals

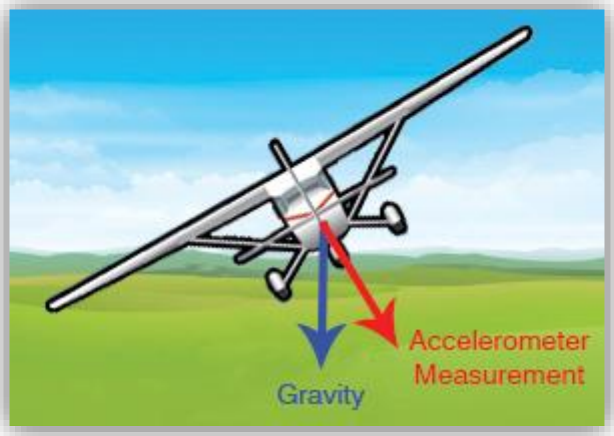
An Attitude Heading Reference System (AHRS) is a sensor system that estimates the attitude of a vehicle based upon the combined measurements provided by a 3-axis gyroscope, accelerometer, and magnetometer. An AHRS sensor typically utilizes a Kalman filter to compute the 3D orientation of the vehicle based upon the vector measurements provided from the accelerometer and the magnetometer. The accelerometer measures the effect of both gravity and any acceleration due to body motion. The magnetometer measures the influence of both the earth's magnetic field and the influence of any nearby magnetic fields created by nearby ferromagnetic objects. The gyroscope provides an accurate short term measurement of the relative change in the orientation of the sensor however it is not capable of providing a measurement of the orientation itself. The absolute accuracy of the heading, pitch and roll solution for an AHRS is ultimately derived from the accuracy of the vector measurements provided by the accelerometer and magnetometer.

AHRS Assumptions

Without any form of external compensation an AHRS does not have by itself any means of knowing how it is moving relative to the fixed Earth. As such it does not have any means of knowing what the actual acceleration of the body is. Since the accelerometer measures the effect of both gravity and the acceleration due to motion, the standard AHRS algorithm has to make the assumption that the long-term acceleration due to motion is zero. With this assumption in place the AHRS know has sufficient information to estimate the pitch and roll based upon the measurement of gravity provided by the accelerometer. This assumption works very well for applications where the sensor does not experience any long-term acceleration such as when it is used indoors or when used on a large marine vessel. Applications that do experience long-term accelerations due to motion however will experience a significant error in the pitch and roll solution due to the fact that the assumption of zero body acceleration in the AHRS algorithm is constantly being violated.

The most common case where this acceleration becomes a significant problem for an AHRS is when it is used on an aircraft operating in a banked turn. In straight and level flight the AHRS will provide an accurate measurement of attitude as long as the long-term accelerations are nominally zero. When the aircraft banks and enters a coordinated turn however, a long-term acceleration is present which due to the centripetal force created by traveling along a curved path. This apparent force is what makes you feel as if you are being pushed to the side when you drive around a corner in a car.

Figure 1 - Measured Acceleration in Coordinated Turn



When an aircraft is in a banked turn the accelerometer will measure gravity plus this centripetal acceleration which will result in a measurement vector that acts perpendicular to the wings of the aircraft as shown in the above figure. This will result in the AHRS estimating a roll angle of zero while the aircraft is in fact in a banked turn and thus has a significant actual roll angle relative to the horizon.

If the AHRS however can obtain some knowledge of this actual motion relative to the fixed Earth then it is possible for it to subtract out the effect of the centripetal acceleration, resulting in an accurate estimate of attitude. By providing the AHRS with the known velocity or airspeed it is possible for the AHRS to estimate the centripetal acceleration term based upon this velocity and the known body angular rates.

Figure 2 - AHRS with Velocity Compensation

The above figure accurately depicts quality of attitude solution provided by three separate types of attitude estimators while operating in a coordinated turn. The flight display on the far left represents the actual attitude which is derived from the flight simulator. Moving from left to right are three separate types of attitude estimators shown in order based upon the accuracy of their derived solution. The most accurate solution is proved by the Inertial Navigation System (INS). This type of estimator incorporates the position and velocity measurements from a GPS along with the accelerometer, and gyroscope in an optimal fashion to simultaneously estimate attitude and the position and velocity of the vehicle. It provides the most accurate attitude estimate since it makes no assumptions regarding the accelerometer measurements.

Measurement Sources for Velocity Aiding

Below are three common sources used for velocity aiding.

Airspeed Sensor

When an airspeed sensor is used for velocity aiding it is important to note which type of airspeed is being used. Since the airspeed input is being used by the AHRS to estimate the centripical acceleration, the airspeed used should be ideally close to the actual speed relative to the fixed earth. Normally airspeed sensors measure the speed of the aircraft relative to the atmosphere, thus there will be a difference between the speed relative to the fixed Earth and the speed given by the airspeed indicator, equal to the speed of the atmosphere relative to the ground (wind speed). In high wind conditions this can cause some increased error in the velocity aiding algorithm.

When using airspeed

Speedometer

For automotive applications the speedometer measurement can be used to perform velocity aiding. The speedometer measurement will provide the ground speed of the vehicle. There will be some small loss due to fact that vertical speed is not included, however the effect will be minimal.

GPS

For most applications GPS provides an excellent source of velocity aiding for an AHRS. It is recommended that you use a GPS receiver with at least a 5Hz update rate.

9.1.1 Tuning for Higher Performance

In most situations the default tuning parameters for the velocity compensation will provide adequate results without the need for manual adjustment. In the event that you have a case where you need improved performance, there are tuning parameters provided in the Velocity Compensation Control Register (Register 50) that provide a means to adjust the behavior of the compensation algorithm.

Velocity Tuning

The velocity tuning field in the Velocity Compensation Control Register in the Velocity Aiding subsystem provides a means to adjust the uncertainty level used for the velocity measurement in the compensation estimation filter. The default value is 0.1. A larger value places less trust in the velocity measurements, while a smaller number will place more trust in the velocity measurement. If your velocity measurement is noisy or unreliable increasing this number may provide better results. If you have a very accurate velocity measurement then lowering this number will likely produce better results.

Velocity Measurement Rate

The performance of the velocity compensation will be affected by both the accuracy of the velocity measurements and the rate at which they are applied. To ensure adequate performance the velocity should be provided at a rate higher than 1Hz. Best performance will be achieved with update rates of 10Hz or higher.

> If you stop sending velocity measurement updates for any reason, the velocity compensation will continue indefinitely using the last received velocity measurement. If you want to stop using while the vehicle is still in motion, be sure to turn off the velocity compensation using the Mode field in the Velocity Compensation Control Register in the Velocity Ading subsystem.

9.2 Configuration Registers

9.2.1 Velocity Compensation Control

Velocity Compensation Control			
Register ID :	51	**Access :**	Read / Write
Comment :	Provides control over the velocity compensation feature for the attitude filter.		
Size (Bytes):	8		
Example Response:	$VNRRG,51,1,0.1,0.01*5A		

Offset	Name	Format	Unit	Description
0	Mode	uint8	-	Selects the type of velocity compensation performed by the VPE. See the table below for available options.
4	VelocityTuning	float	-	Tuning parameter for the velocity measurement.
8	RateTuning	float	-	Tuning parameter for the angular rate measurement.

Table 4 - Velocity Compensation Modes

Value	Description
0	Disabled.
1	Body Measurement.

9.3 Status Registers

9.3.1 Velocity Compensation Status

INTERNAL REGISTER

This register is not listed in the public User Manual. It is not recommended to supply this register to customers unless there is a specific reason to do so.

Velocity Compensation Status				
Register ID :	52		**Access :**	Read
Comment :	Provides diagnostic status information for the velocity compensation algorithm.			
Size (Bytes):	8			
Example Response:	$VNRRG,51,1,0.1,0.01*5A			

Offset	Name	Format	Unit	Description
0	x	float	m/s	Estimated velocity magnitude.
4	xDot	float	m/s^2	Estimated acceleration magnitude.
8	accelOffset	float[3]	m/s^2	Estimated acceleration offset.
20	omega	float[3]	rad/s	Filtered angular rate.

9.4 Input Measurements

9.4.1 Velocity Compensation Measurement

Velocity Compensation Measurement			
Register ID :	50	**Access :**	Read / Write
Comment :	Input register for a velocity measurement to be used by the filter to compensate for acceleration disturbances.		
Size (Bytes):	12		
Example Response:	$VNRRG,50,37.2,0,0*42		

Offset	Name	Format	Unit	Description
0	VelocityX	float	m/s	Velocity in the X-Axis measured in the sensor frame.
4	VelocityY	float	m/s	Velocity in the Y-Axis measured in the sensor frame.
8	VelocityZ	float	m/s	Velocity in the Z-Axis measured in the sensor frame.

For Mode 1 (body measurement mode) the VN-100 will compute the vector length of the provided 3D velocity vector and use this for velocity compensation. If you have a scalar measurement you can set only the X-axis and set the Y & Z to zero.

9.5 Factory Defaults

Settings Name	Default Factory Value
Velocity Compensation Control	1,0.1,0.01