VN-100

User Manual

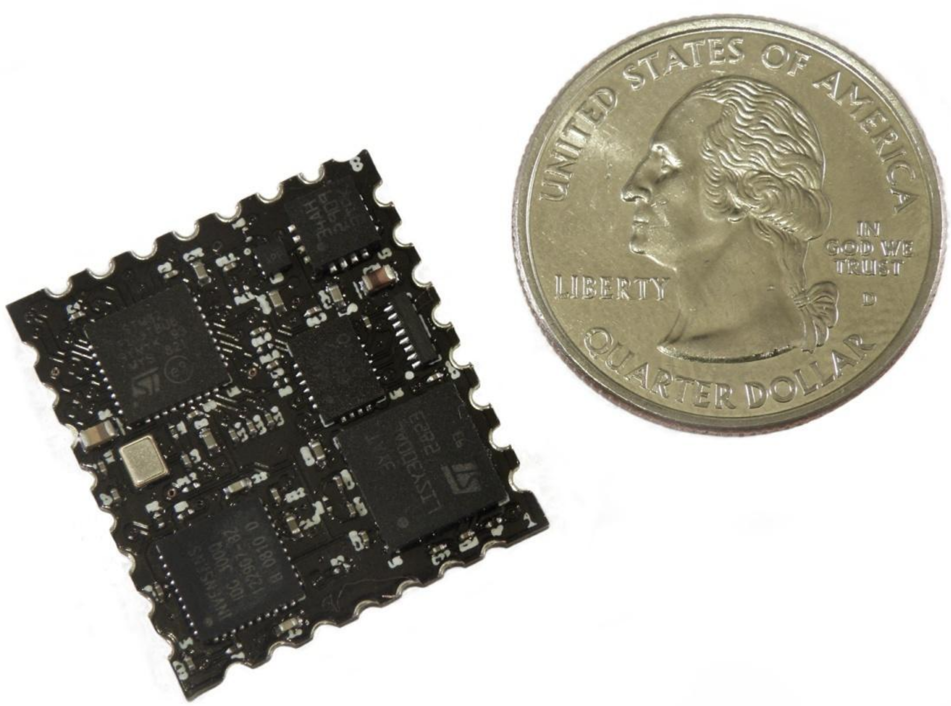


Table of Contents

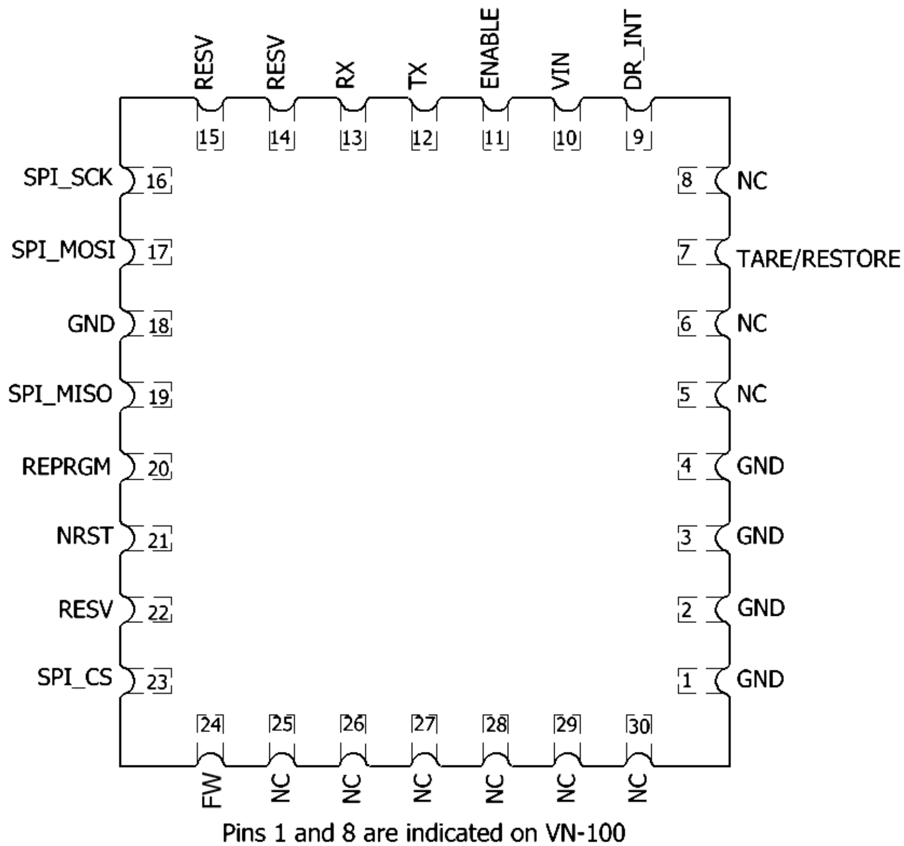
1 Introduction

The VN-100 module is the smallest attitude and heading reference system (AHRS) produced and is also the first available in a surface mount package. These features make the VN-100 module ideal for incorporating accurate and reliable device orientation information in your compact embedded electronic designs. This user manual details the specifics of the VN-100 module and describes how to embed it into your electronic system.

People new to using AHRS devices are recommended to read Section 11. Intro to AHRS and the VN-100 Kalman Filter. It provides an introduction to AHRS technology and some of the inherent properties of using an AHRS device.

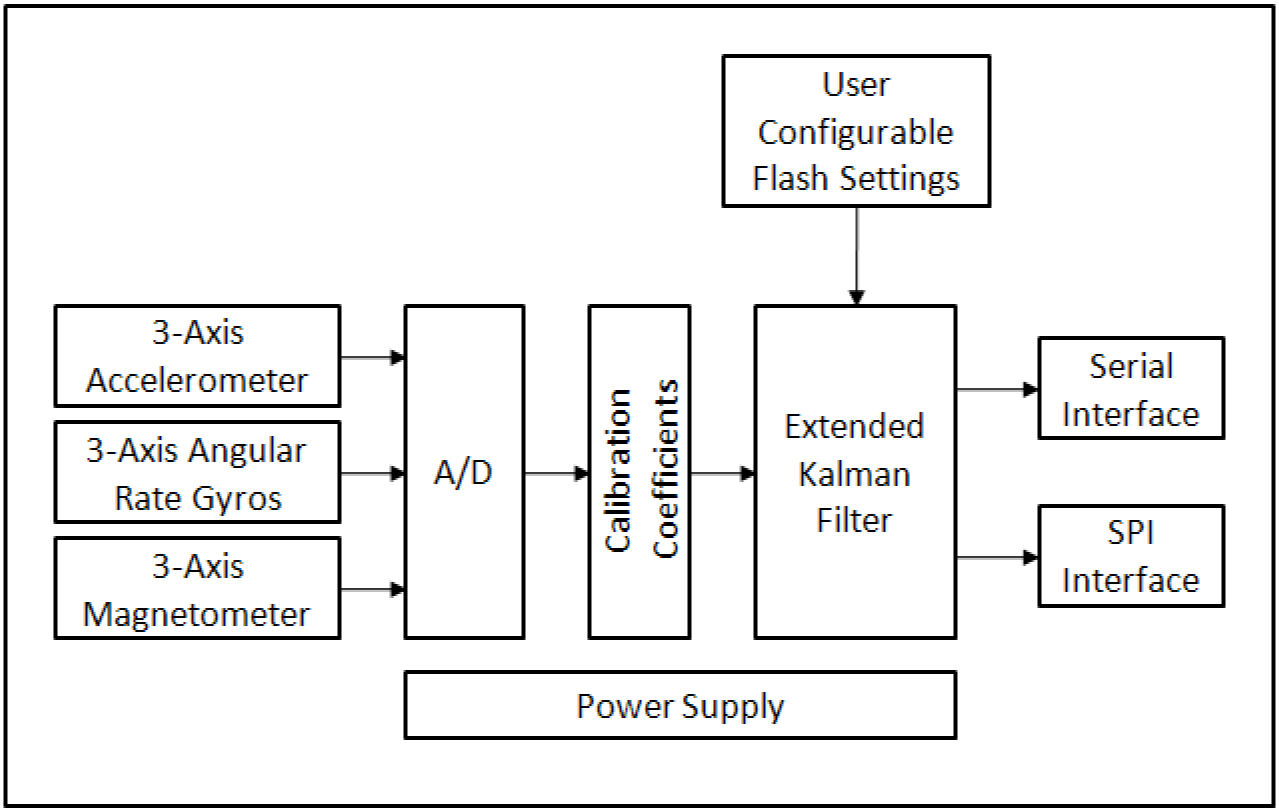
2 Device Overview

Figure 1 – Pin assignments



-

Figure 2 - Simplified Block Diagram



The VN-100 internal architecture combines a full 3-axis set of accelerometers, magnetometers and rate gyros that send analog outputs through low pass filters prior to digital conversion. After the analog voltages have been digitized they run through calibration models with factory calibrated coefficients. This calibrated data is then made available as outputs from the device and also to the embedded filter. Based on the user configurable flash settings, the VN-100 then uses a Kalman filter algorithm to solve for the orientation and bias-free gyro rates.

2.1 Coordinate System

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 Figure 3.

Figure 3 - VN-100 Coordinate System

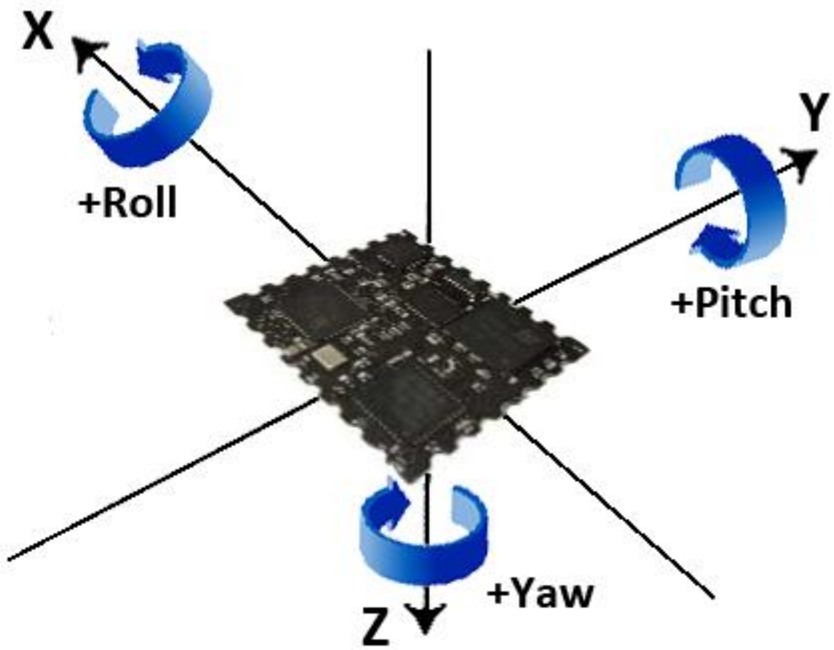


Table 1 – Pin Assignments

Pin #	Pin Name	Description
1	GND	Ground
2	GND	Ground
3	GND	Ground
4	GND	Ground
5	NC	Not connected. Empty pin.
6	NC	Not connected. Empty pin.
7	TARE/RESTORE	Normally used to tare the chip. 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 its default factory settings. Because of this, this pin cannot be used for tare until at least 5 ms after a power on or reset. If the REPRGM pin is pulled High and the TARE/RESTORE pin is held Low on power on or after reset, the chip will enter reprogramming mode.
8	NC	Not used
9	DR_INT	Data Ready Interrupt. Will pulse Low for 500 µs when a new data point is ready.
10	VIN	3.3-5.5V input.
11	ENABLE	Leave high for normal operation. Pull low to enter sleep mode. Internally pulled high with pull-up resistor.
12	TX	Serial UART data output line for AHRS
13	RX	Serial UART data input line for AHRS
14	RESV	Reserved for future use. Leave pin floating.
15	RESV	Reserved for future use. Leave pin floating.
16	SPI_SCK	SPI clock
17	SPI_MOSI	SPI input
18	GND	Ground
19	SPI_MISO	SPI output
20	REPRGM	Used for controlling reprogramming of the module. Must be left floating or set to Low for normal operations. Pull High (along with pulling TARE/RESTORE Low) on startup to set the chip in reprogram mode. Internally pulled Low.
21	NRST	Microcontroller reset line. Pull low for > 20µs to reset MCU. Internally pulled high with 10k.
22	RESV	Reserved for future use. Leave pin floating.
23	SPI_CS	SPI slave select
24	FW	Pull high during reset to put chip into firmware update mode. This pin is supported by firmware v5 and higher.
25	NC	Not connected. Empty pin.
26	NC	Not connected. Empty pin.
27	NC	Not connected. Empty pin.
28	NC	Not connected. Empty pin.
29	NC	Not connected. Empty pin.
30	NC	Not connected. Empty pin.

3 Communication Modes

The VN-100 module supports two communication modes: UART and SPI. In UART mode, the module communicates over a universal asynchronous receiver/transmitter (UART) and uses ASCII text for its command format. In SPI mode, the VN-100 module communicates as a slave device on a Serial Peripheral Interface (SPI) data bus and uses a binary command format. Both modes support the complete command set implemented by the module. A general overview of the command format for each mode is given in the next two sections and formatting specific to each command and associated parameters is provided in the Protocol and Register sections (Section 5 & 6).

3.1 UART Mode

When in UART mode, the module uses ASCII text for its command format. All commands start with a dollar sign, followed by a five character command, a comma, command specific parameters, an asterisk, a checksum, and a newline character. An example command is shown below.

```
$VNRRG,11*73
```

3.2 Checksum Algorithm

The checksum is only used in UART mode. It 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 a two character hexadecimal representation. The C function snippet below calculates the correct checksum.

```
unsigned char calculateChecksum(char* command, int length)
{
    unsigned char xor = 0;

    for(int i = 0; i < length; i++)
        xor ^= (unsigned char)command[i];

    return xor;
}
```

3.3 SPI Mode

SPI mode uses a lightweight binary message format over the SPI data bus. The start of a command is signaled by pulling the VN-100 module's chip select pin (pin 23) low. Both the chip select line and clock are active low. The first byte transmitted to the module should be the command ID and then a variable number of bytes will follow dependent on the type of command specified. A communication transaction can be cancelled at any time by releasing the chip select pin. Pulling the pin low again will start a new communication transaction. All binary data is sent to and from the chip with most significant bit (MSB) first in little-endian byte order with all numbers either represented as 32-bit floating point or 32-bit unsigned integers. For example the serial baud rate register with a value of 9600 (0x2580) would be sent across the SPI as a 0x80, 0x25, 0x00, 0x00. Data is requested from and written to the device using multiple SPI transactions.

Figure 4 – SPI Timing Diagram

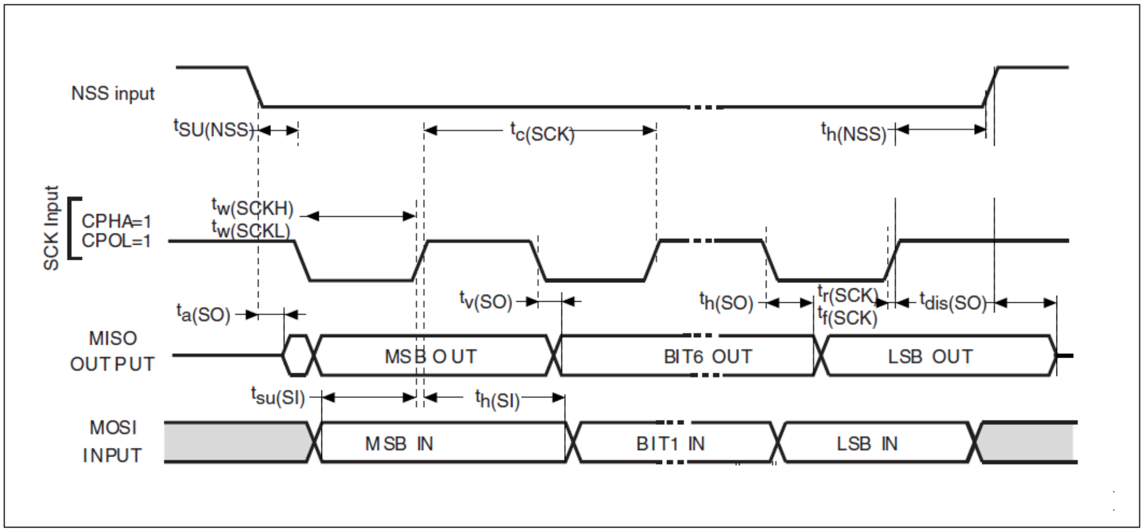
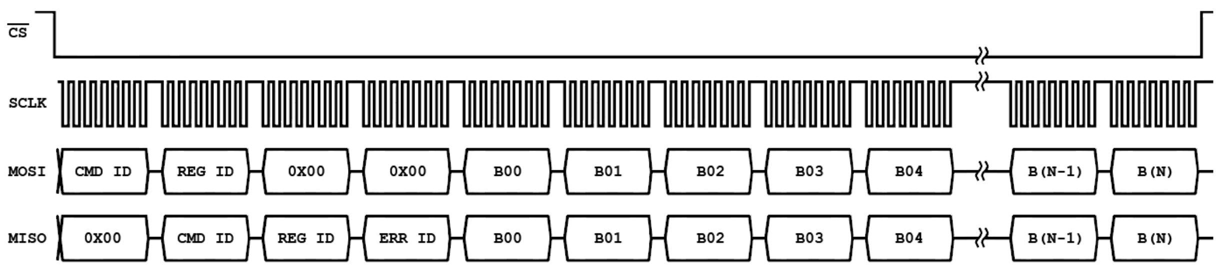


Figure 5 - SPI Data Diagram



A response for a given SPI command will be sent over the MISO line on the next SPI transaction. Thus the data received by the Master on the MISO line will always be the response to the previous transaction. So for example if Yaw, Pitch, Roll and angular rates are desired, then the necessary SPI transactions would proceed as shown below.

SPI Transaction 1		
Line	**Bytes**	**Description**
SCK	8 bytes	
MOSI	01 08 00 00 00 00 00 00 (shown as hex)	Read register 8 (Yaw, Pitch, Roll)
MISO	00 00 00 00 00 00 00 00 (shown as hex)	No response

SPI Transaction 2		
Line	**Bytes**	**Description**
SCK	16 bytes	
MOSI	01 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)	Read register 13 (Angular Rates)
MISO	00 01 08 00 39 8A 02 43 FD 43 97 C1 CD 9D 67 42 (shown as hex)	Yaw, Pitch, Roll = -130.54, -18.91, +57.90

SPI Transaction 3		
Line	**Bytes**	**Description**
SCK	16 bytes	
MOSI	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)	No command
MISO	00 01 13 00 00 F5 BF BA 00 80 12 38 B8 CC 8D 3B (shown as hex)	Rates = -0.001465, +0.000035, +0.004327

During the first transaction the master sends the command to read register 8. The available registers which can be read or written to are listed in Table 5. At the same time zeros are received by the master, assuming no previous SPI command was sent to the chip since reboot. On the second transaction the master sends the command to read register 13. At the same time the response from the previously requested register 8 is received by the master on the MISO line. It consists of four 32-bit words. The first byte of the first word will always be zero. The second byte of the first word is the type of command that this transaction is in response to. In this case it is a 0x01 which means that on the previous transaction a read register command was issued. The third byte of the first word is the register that was requested on the previous transaction. In this case it shows to be 0x08 which is the yaw, pitch, roll register. The fourth byte of the first word is the error code for the previous transaction. Possible error codes are listed in Table 4. The remaining three 4-byte words are the yaw, pitch, and roll respectively given as single precision floating point numbers. The floating point numbers are consistent with the IEEE 754 standard. On the third SPI transaction 16 bytes are clocked on the SCK line, during which zeros are sent by the master since no further data is required from the sensor. These 16 bytes are clocked out the SPI for the sole purpose of reading the response from the previous read register 13 command. The response consists of 4 32-bit words, starting with the zero byte, the requested command byte, register ID, error code, and three single precision floating point numbers. If only one register is required on a regular basis then this can be accomplished by sending the same command twice to the VN-100. The response received on the second transaction will contain the most up to date values for the desired register.

SPI Transaction 1		
Line	**Bytes**	**Description**
SCK	16 bytes	
MOSI	`01 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)`	Read register 8 (Yaw, Pitch, Roll)
MISO	`00 01 08 00 39 8A 02 43 FD 43 97 C1 CD 9D 67 42 (shown as hex)`	Yaw, Pitch, Roll = +130.54, -18.91, +57.90

SPI Transaction 2		
Line	**Bytes**	**Description**
SCK	16 bytes	
MOSI	`01 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)`	Read Register 8 (Yaw, Pitch, Roll)
MISO	`00 01 08 00 C5 9A 02 43 51 50 97 C1 32 9A 67 42 (shown as hex)`	Yaw, Pitch, Roll = +130.60, -18.91, +57.90

At first the device would be initialized by sending the eight bytes 01 08 00 00 00 00 00 00, requesting a read of the yaw, pitch, roll register. The response from the second transaction would be the response to the requested yaw, pitch, roll from the first transaction. The minimum time required between SPI transactions is 50μs.

4 Asynchronous Data Output

The VN-100 module continuously computes its orientation solution and is capable of automatically notifying the user and outputting this data when it becomes immediately available. The frequency of this output is specified by the Async Data Output Frequency (ADOF) register (Section 6.7) and the format is specified by the Async Data Output Register (Section 6.6). In UART mode, the module will output the data. However, when the module is in SPI mode, the module will notify the user by pulling low the DR_INT pin, at which time the user can retrieve the desired data from the chip.

5 Protocol

The following sections describe the serial and SPI data protocol used by the VN-100.

5.1 Numeric Formats

Floating point numbers displayed as ASCII text are presented in two formats: single precision floating point and single precision fixed point. In order to conserve bandwidth each variable in the register has associated with it either a floating or fixed point representation. Any time this variable is accessed using a read/write register command or as Async output, the variable will always use its associated data format.

5.1.1 Single Precision Floating Points

Single precision floating point numbers are represented with 7 significant digits and a 2 digit exponent. Both the sign of the number and exponent are provided. The decimal point will always follow the first significant digit. An 'E' will separate the significant digits from the exponential digits. Below are some samples of correct single precision floating point numbers.

Single Precision Floating Point Number Examples

+9.999999E+99	-7.344409E-05
-1.234567E+01	+4.893203E+00

5.1.2 Fixed-Point Numbers

The fixed-point representation consists of a specified number of digits to the left and right of a fixed decimal point. The registers that use fixed point representation and their associated formatting are listed below.

Table 2 – Floating Point Representation

Variable Type	Fixed/Floating	Register ID(s)	Printf/Scanf	Example
Yaw, Pitch, Roll	Fixed	8, 27	%+07.2f	+082.76
Quaternion	Fixed	9,10,11,12,13,14,15	%+09.6f	+0.053362
Magnetic	Fixed	10,13,15,17,20,27	%+07.4f	-0.3647
Acceleration	Fixed	11,13,14,18,20,27	%+07.3f	-09.091
Angular Rate	Fixed	12,14,15,19,20,27	%+08.4f	+00.0017
Directional Cosine Matrix	Floating	16	%+13.6E	+9.801470E-01
Reference Vectors	Floating	21	%+13.6E	+8.135935E-02
Filter Measurement Variance	Floating	22	%+13.6E	+4.968087E-06
Hard/Soft Iron Compensation (HSI)	Floating	23	%+13.6E	+1.000000E+00
Auto Tuning Parameters	Floating	24	%+13.6E	+9.900000E-01
Accelerometer Compensation	Floating	25	%+13.6E	+1.000000E+00
Reference Frame Rotation	Floating	26	%+13.6E	+1.000000E+00

5.2 Command List

This section describes the list of commands available on the VN-100 module. All commands are available in both ASCII text (UART) and binary (SPI) command formats.

The table below lists the commands available along with some quick information about the commands. The Text ID is used to specify the command when using the text command format and the Binary ID is used to specify the command when using the binary command format. More details about the individual commands can be found in the referenced section.

Table 3 – List of Available Commands

Command Name	Text ID	Binary ID	Section
Read Register	VNRRG	0x01	5.2.1
Write Register	VNWRG	0x02	5.2.2
Write Settings	VNWNV	0x03	5.2.3
Restore Factory Settings	VNRFS	0x04	5.2.4
Tare	VNTAR	0x05	5.2.5
Reset	VNRST	0x06	5.2.6

5.2.1 Read Register Command

This command allows the user to read any of the registers on the VN-100 module (see Section 6 for the list of available registers). 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 section contained in Section 6 for details on this formatting. If an invalid register is requested, an error code will be returned. The error code format is described in Section 5.3.

5.2.1.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the serial baud rate (register 5) and the response from the VN-100 shows that the baud rate is set to 9600.

UART Command	`$VNRRG,5*46`
UART Response	`$VNRRG,5,9600*65`
SPI Command (8 bytes)	`01 05 00 00 80 25 00 00 (shown as hex)`
SPI Response (8 bytes)	`00 01 05 00 80 25 00 00 (shown as hex)`

5.2.2 Write Register Command

This command is used to write data values to a specified register on the VN-100 module (see Section 6 for the list of available registers). 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 section in Section 6 for this formatting. If an invalid register is requested, an error code will be returned. The error code format is described in Section 5.3.

5.2.2.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user instructs the VN-100 to set the serial baud rate to 9600. The VN-100 then responds with the updated baud rate.

UART Command	`$VNWRG,5,9600*60`
UART Response	`$VNWRG,5,9600*60`
SPI Command (8 bytes)	`02 05 00 00 80 25 00 00 (shown as hex)`
SPI Response (8 bytes)	`00 02 05 00 80 25 00 00 (shown as hex)`

5.2.3 Write Settings Command

This command will write the current register settings into non-volatile memory. Once the settings are stored in non-volatile memory, the VN-100 module can be power cycled or reset, and the register will be reloaded from non-volatile memory. The module can always be reset to the factory settings by issuing the Restore Factory Settings command (Section 5.2.4) or by pulling pin 15 high during reset.

5.2.3.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In this example the user simply requests that the current register settings be saved to non-volatile memory. Upon completing the operation the VN-100 responds with the same command to indicate that the operation was successful.

UART Command	$VNWNV*57
UART Response	$VNWNV*57
SPI Command (8 bytes)	03 00 00 00 00 00 00 00 (shown as hex)
SPI Response (8 bytes)	00 03 00 00 00 00 00 00 (shown as hex)

5.2.4 Restore Factory Settings Command

This command will restore the VN-100 module's factory default settings (see Section 7). There are no parameters for this command. The module will respond to this command before restoring the factory settings.

5.2.4.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In this example the user sends a command to restore all of the registers to the factory default state. A response is sent from the VN-100 to indicate that the operation was successful.

UART Command	$VNRFS*5F
UART Response	$VNRFS*5F
SPI Command (8 bytes)	04 00 00 00 00 00 00 00 (shown as hex)
SPI Response (8 bytes)	00 04 00 00 00 00 00 00 (shown as hex)

5.2.5 Tare Command

The Tare command will have the module zero out its current orientation. The effect of this command will be to set the current orientation as zero yaw, zero pitch, and zero roll, or equivalently a quaternion of [0, 0, 0, 1]. Yaw will still be measured as rotation about the current Z-axis, pitch about the current Y-axis, and roll about the current X-axis. See the Reference Frame Rotation Register (Section 6.26) for an explanation of how to determine the orientation of a different coordinate reference frame than that used by the VN-100. There are no parameters for this command. For more information on this command see the Appendix Utilizing the "Tare" command.

5.2.5.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user sends a command to zero out the current orientation. A response is sent from the VN-100 to indicate the operation was successful.

UART Command	$VNTAR*5F
UART Response	$VNTAR*5F
SPI Command (8 bytes)	05 00 00 00 00 00 00 00 (shown as hex)
SPI Response (8 bytes)	00 05 00 00 00 00 00 00 (shown as hex)

5.2.6 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 then 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. This command is equivalent in functionality to the hardware reset performed by pulling pin 21 low.

5.2.6.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user sends a command to initiate a reset of the VN-100. The module will respond to the command and then immediately perform a reset.

UART Command	$VNRST*4D
UART Response	$VNRST*4D
SPI Command (8 bytes)	06 00 00 00 00 00 00 00 (shown as hex)
SPI Response (8 bytes)	00 06 00 00 00 00 00 00 (shown as hex)

5.3 Error Codes

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

Table 4 – Error Codes

Error Name	Code	Description
Hard Fault	1	If this error occurs, then the firmware on the chip 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 50ms 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 has attempted to write to a read-only 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 50ms 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.

6 Registers

The VN-100 module contains a collection of registers used for configuring the module and accessing the data it produces. These registers may be read or written to using the Read Register and Write Register commands (Sections 5.2.1 and 5.2.2). When the module is rebooted or power-cycled, values written to the registers will revert back to their previous values unless a Write Settings command has been issued (Section 5.2.3) to save the registers to non-volatile memory.

Table 5 provides a quick reference for all of the registers and their associated properties. The second column lists the Access ID, which is used to identify a specific register. The third column indicates if the register can be read (R) and/or written to (W). The forth column indicates the width of the register in bytes (relevant only in SPI mode) and the last column provides the section number where a more detailed explanation of the register may be found.

Each register may be written to or read using either Serial or SPI communication modes. The specific register sections that follow describe the format used by each communication mode.

Table 5 –Available Registers

Register Name	Access ID	R/W	Width (bytes)	Section
Model Number	1	R	12	6
Hardware Revision	2	R	4	6.2
Serial Number	3	R	12	6.3
Firmware Version	4	R	4	6.4
Serial Baud Rate	5	R/W	4	6.5
Async Data Output Type	6	R/W	4	6.6
Async Data Output Frequency	7	R/W	4	6.7
Attitude (Yaw, Pitch, Roll)	8	R	3 x 4	6.8
Attitude (Quaternion)	9	R	4 x 4	6.9
Quaternion and Magnetic	10	R	7 x 4	6.10
Quaternion and Acceleration	11	R	7 x 4	6.11
Quaternion and Angular Rates	12	R	7 x 4	6.12
Quaternion, Magnetic and Acceleration	13	R	10 x 4	6.13
Quaternion, Acceleration and Angular Rates	14	R	10 x 4	6.14
Quaternion, Magnetic, Acceleration, and Angular Rates	15	R	13 x 4	6.15
Attitude (Directional Cosine Matrix)	16	R	9 x 4	6.16
Magnetic Measurements	17	R	3 x 4	6.17
Acceleration Measurements	18	R	3 x 4	6.18
Angular Rate Measurements	19	R	3 x 4	6.19
Magnetic, Acceleration, and Angular Rate Measurements	20	R	9 x 4	6.20
Magnetic and Gravity Reference Vectors	21	R/W	6 x 4	6.21
Filter Measurements Variance Parameters	22	R/W	10 x 4	6.22
Magnetic Hard/Soft Iron Compensation Parameters	23	R/W	12 x 4	6.23
Disturbance Tuning Parameters	24	R/W	4 x 4	6.24
Accelerometer Compensation	25	R/W	12 x 4	6.25
Reference Frame Rotation	26	R/W	9 x 4	0
Yaw, Pitch, Roll, Magnetic, Acceleration, and Angular Rates	27	R/W	12 x 4	0
Accelerometer Gain	28	R/W	4	6.28

6.1 Model Number Register

This register contains the device's model number. When read using the UART a six character string "VN-100" is returned, whereas using the SPI mode gives a twelve character string with the NUL character after the "VN-100" and 0xFF thereafter.

6.1.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the model number and the VN-100 responds back with "VN-100".

Example Type	Message
UART Read Request	$VNRRG,01*72
UART Read Response	$VNRRG,01,VN-100*5A
SPI Read Request (16 bytes)	`01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (16 bytes)	`00 01 01 00 56 4E 2D 31 30 30 00 FF FF FF FF FF  (shown as hex)`

6.2 Hardware Revision Register

This register returns the hardware revision of the module.

6.2.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the hardware revision, and the VN-100 responds back with revision 1.

Example Type	Message
UART Read Request	$VNRRG,02*71
UART Read Response	$VNRRG,02,1*6C
SPI Read Request (8 bytes)	`01 02 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (8 bytes)	`00 01 02 00 01 00 00 00  (shown as hex)`

6.3 Serial Number Register

This registers contains the serial number of the module. The serial number is a 12-byte number represented as a 24 hexadecimal digits that is unique to the VN-100 module.

6.3.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. In the example the user requests the device serial number and the VN-100 responds back with 067200383733335843046264.

Example Type	Message
UART Read Request	$VNRRG,03*70
UART Read Response	$VNRRG,03,0672003837333358430462б4*58
SPI Read Request (16 bytes)	01 03 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (16 bytes)	00 01 03 00 45 35 02 43 58 33 37 38 00 6D 06 (shown as hex)

6.4 Firmware Version Register

This register contains the version number of the onboard firmware.

6.4.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the firmware version, and the VN-100 responds back with version 4.

Example Type	Message
Read Request	$VNRRG,04*77
Read Response	$VNRRG,04,1*6A
SPI Read Request (8 bytes)	01 04 00 00 00 00 00 00 (shown as hex)
SPI Read Response (8 bytes)	00 01 04 00 0A 00 00 00 (shown as hex)

6.5 Serial Baud Rate Register

This register specifies the baud rate of the serial data bus. The table below specifies the associated baud rate achieved when the register is set to one of the values listed in Table 6. The response for this command will be sent after the baud rate is changed.

Table 6 – Baud Rate Settings

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

6.5.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the

VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the serial baud rate and the VN-100 responds back with 9600 baud. Then the user sends a write request to change the baud rate to 115200. The VN-100 then responds back with the new enabled baud rate.

Example Type	Message
UART Read Request	$VNRRG,05*76
UART Read Response	$VNRRG,05,9600*55
UART Write Request	$VNWRG,05,115200*58
UART Write Response	$VNWRG,05,115200*58
SPI Read Request (8 bytes)	01 05 00 00 00 00 00 00 (shown as hex)
SPI Read Response (8 bytes)	00 01 05 00 80 25 00 00 (shown as hex)
SPI Write Request (8 bytes)	02 05 00 00 00 00 00 00 (shown as hex)
SPI Write Response (8 bytes)	00 02 05 00 00 C2 01 00 (shown as hex)

6.6 Async Data Output Type Register

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. Table 7 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 (Section 6.7).

Table 7 – Asynchronous Solution Output Settings

Setting	Asynchronous Solution Output Type	Header	Formatting Section
0	Asynchronous output turned off	N/A	N/A
1	Yaw, Pitch, Roll	VNYPR	6.8
2	Quaternion	VNQTN	6.9
3	Quaternion and Magnetic	VNQTM	6.10
4	Quaternion and Acceleration	VNQTA	6.11
5	Quaternion and Angular Rates	VNQTR	6.12
6	Quaternion, Magnetic and Acceleration	VNQMA	6.13
7	Quaternion, Acceleration and Angular Rates	VNQAR	6.14
8	Quaternion, Magnetic, Acceleration and Angular Rates	VNQMR	6.15
9	Directional Cosine Orientation Matrix	VNDCM	6.16
10	Magnetic Measurements	VNMAG	6.17
11	Acceleration Measurements	VNACC	6.18
12	Angular Rate Measurements	VNGYR	6.19
13	Magnetic, Acceleration, and Angular Rate Measurements	VNMAR	6.20
14	Yaw, Pitch, Roll, Magnetic, Acceleration, and Angular Rate Measurements	VNYMR	6.27

6.6.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current Async Data Output Type. The VN-100 then responds back with the Async Data Output Type 1 (Yaw, Pitch, and Roll). The user then sends a write request to set the Async Data Output type to 2 (Quaternion). A response is sent from the VN-100 to indicate the operation was successful.

Example Type	Message
UART Read Request	$VNRRG,06*75
UART Read Response	$VNRRG,06,1*68
UART Write Request	$VNWRG,06,2*6E
UART Write Response	$VNWRG,06,2*6E
SPI Read Request (8 bytes)	`01 06 00 00 00 00 00 00 (shown as hex)`
SPI Read Response (8 bytes)	`00 01 06 00 01 00 00 00 (shown as hex)`
SPI Write Request (8 bytes)	`02 06 00 00 02 00 00 00 (shown as hex)`
SPI Write Response (8 bytes)	`00 02 06 00 02 00 00 00 (shown as hex)`

6.7 Async Data Output Frequency Register

This register determines the frequency for asynchronous data output. This register does not affect the rate at which the internal Kalman filter is run. The Kalman filter is always set at a fixed rate of 200Hz. This register merely determines the number of filter cycles to wait before sending out an asynchronous data output. If asynchronous data output is disabled (see Section 6.6), this register's setting will have no effect until asynchronous data output is enabled again. This register may be set to any of the values listed in Table 8. If the specified frequency is too high to output due to a low baud rate setting, the VN-100 will return error code 12 – Insufficient Baud Rate (see Section 5.3). Table 9 shows the maximum allowed data rates for each possible asynchronous data output type.

Table 8 - Data Rates

Acceptable Data Rates (Hz)
1
2
4
5
10
15
20
25
40
50
100
200

Table 9 - Maximum Data Rates

ADOR Setting	Baud Rates (bps)								
	9600	**19200**	**38400**	**57600**	**115200**	**128000**	**230400**	**460800**	**921600**
YPR	20	40	100	200	200	200	200	200	200
QTN	10	25	50	50	100	200	200	200	200
QTM	5	20	40	50	100	100	200	200	200
QTA	5	20	40	50	100	100	200	200	200
QTR	5	20	40	50	100	100	200	200	200
QMA	5	10	25	40	50	100	100	200	200
QAR	5	10	25	40	50	100	100	200	200
QMR	5	10	20	25	50	50	100	200	200
DCM	4	10	20	25	50	50	100	200	200
MAG	20	40	50	100	200	200	200	200	200
ACC	20	40	50	100	200	200	200	200	200
GYR	20	40	50	100	200	200	200	200	200
MAR	5	10	25	50	100	100	200	200	200
YMR	5	10	20	25	50	50	100	200	200

6.7.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current Async Data Output Frequency and the VN-100 responds back with 10 Hz. Then the user sends a write request to set the Async Data Output Frequency to 100 Hz. A response is sent from the VN-100 to indicate the operation was successful.

Example Type	Message
UART Read Request	$VNRRG,07*74
UART Read Response	$VNRRG,07,10*59
UART Write Request	$VNWRG,07,100*6C
UART Write Response	$VNWRG,07,100*6C
SPI Read Request (8 bytes)	01 07 00 00 00 00 00 00 (shown as hex)
SPI Read Response (8 bytes)	00 01 07 00 0A 00 00 00 (shown as hex)
SPI Write Request (8 bytes)	02 07 00 00 64 00 00 00 (shown as hex)
SPI Write Response (8 bytes)	00 02 07 00 64 00 00 00 (shown as hex)

6.8 Attitude (Yaw, Pitch, Roll Format)

This register contains three values representing the module's yaw, pitch, and roll in Euler angles. The fields of this register are outputted as fixed point precision for the serial protocol and 32-bit floating point for the SPI protocol. The values in this register are updated every 5ms (200Hz) by the onboard Kalman filter. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	Yaw Angle	degrees	%+07.2f
2	Pitch Angle	degrees	%+07.2f
3	Roll Angle	degrees	%+07.2f

6.8.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current yaw, pitch, and roll measurements. The VN-100 then responds back with the current measurements.

Example Type	Message
UART Read Request	$VNRRG,08*7B
UART Read Response	$VNRRG,08,-027.33,-005.33,+002.63*65
SPI Read Request (16 bytes)	01 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (16 bytes)	00 01 08 00 CB A5 DA C1 48 80 AA C0 BB 98 28 40 (shown as hex)

6.9 Quaternion

This register contains four values representing the quaternion vector. The quaternion provides a redundant, nonsingular attitude representation that is well suited for describing arbitrary, large rotations. The quaternion is a non-dimensional 4x1 unit vector with the fourth value as the scalar term. The fields of this register are represented with fixed point precision for the serial protocol and 32-bit floating point precision for the SPI protocol. This is a read-only register. All filtering and other mathematical operations performed by the VN-100 are performed using quaternions. The quaternions used by the VN-100 have the following form.

$$q[0] = e_x * \sin\left(\frac{\vartheta}{2}\right)$$

$$q[1] = e_y * \sin\left(\frac{\vartheta}{2}\right)$$

$$q[2] = e_z * \sin\left(\frac{\vartheta}{2}\right)$$

$$q[3] = \cos\left(\frac{\vartheta}{2}\right)$$

Where $e = \begin{Bmatrix} e_x \\ e_y \\ e_z \end{Bmatrix}$ is the principal axis and ϑ is the principal angle.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2 of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f

6.9.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion measurement. The VN-100 then responds back with the current quaternion measurement.

Example Type	Message
UART Read Request	$VNRRG,09*7A
UART Read Response	$VNRRG,09,+0.011391,-0.050566,-0.235156,+0.970574*7F
SPI Read Request (20 bytes)	01 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (20 bytes)	00 01 09 00 B5 A0 3A 3C 86 1E 4F BD CA CC 70 BE 92 77 78 3F (shown as hex)

6.10 Quaternion and Magnetic

This register contains seven values representing a quaternion vector, which represents the module's orientation, followed by the magnetic measurements. The quaternion is a 4x1 unit vector while the magnetic measurement vector is a non-dimensional 3x1 vector. The ordering of the registers seven values is shown below. All seven numbers are represented as fixed point for the serial protocol and 32-bit floating point for the SPI protocol. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Magnetic	N/A	%+07.4f
6	Y-Axis Magnetic	N/A	%+07.4f
7	Z-Axis Magnetic	N/A	%+07.4f

6.10.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion and magnetic measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,10*72
UART Read Response	$VNRRG,10,+0.011129,-0.050382,-0.235107,+0.970599,+0.5048,+0.3128,+0.8129*67
SPI Read Request (32 bytes)	01 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (32 bytes)	00 01 0A 00 F9 55 36 3C E5 5C 4E BD C6 BF 70 BE 32 79 78 3F 07 3C 01 3F CA 2D A0 3E 33 1C 50 3F (shown as hex)

6.11 Quaternion and Acceleration

This register contains seven values representing a quaternion vector, which represents the module's orientation, followed by the acceleration measurements. The quaternion is a 4x1 unit vector while the acceleration measurement vector is a dimensional 3x1 vector. The ordering of this register's seven values is shown below. All seven numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Acceleration	m/s^2	%+07.3f
6	Y-Axis Acceleration	m/s^2	%+07.3f
7	Z-Axis Acceleration	m/s^2	%+07.3f

6.11.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion and acceleration measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,11*73
UART Read Response	$VNRRG,11,+0.010976,-0.050312,-0.235334,+0.970549,-00.206,-00.017,-09.828*6F
SPI Read Request (32 bytes)	01 0B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (32 bytes)	00 01 0B 00 21 D5 33 3C C5 13 4E BD 7E FB 70 BE EC 75 78 3F 48 84 52 BE DC 38 8D BC C6 40 1D C1 (shown as hex)

6.12 Quaternion and Angular Rates

This register contains seven values representing a quaternion vector, which represents the module's orientation, followed by the angular rate measurements. The quaternion is a 4x1 unit vector while the angular rate measurement vector is a dimensional 3x1 vector. The ordering of this register's seven values is shown below. All seven numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Angular Rate	rad/s	%+08.4f
6	Y-Axis Angular Rate	rad/s	%+08.4f
7	Z-Axis Angular Rate	rad/s	%+08.4f

6.12.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion and angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,12*70
UART Read Response	$VNRRG,12,+0.011106,-0.050376,-0.235626,+0.970474,-00.0027,-00.0066,+00.8842*55
SPI Read Request (32 bytes)	01 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (32 bytes)	00 01 0C 00 B5 F4 35 3C 99 56 4E BD DE 47 71 BE FA 70 78 3F 00 1E 31 BB C0 D9 D6 BB 2C 58 62 3F (shown as hex)

6.13 Quaternion, Magnetic and Acceleration

This register contains ten values representing a quaternion vector, which represents the module's orientation, followed by the magnetic and acceleration measurements. The quaternion is a 4x1 unit vector. The ordering of this register's ten values is shown below. All ten numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Magnetic	N/A	%+07.4f
6	Y-Axis Magnetic	N/A	%+07.4f
7	Z-Axis Magnetic	N/A	%+07.4f
8	X-Axis Acceleration	m/s^2	%+07.3f
9	Y-Axis Acceleration	m/s^2	%+07.3f
10	Z-Axis Acceleration	m/s^2	%+07.3f

6.13.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion, magnetic and acceleration measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,13*71
UART Read Response	$VNRRG,13,+0.011092,-0.050245,-0.235634,+0.970479,+0.5047,+0.3136,+0.8133,-00.179,-00.053,-09.832*7A
SPI Read Request (44 bytes)	01 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (44 bytes)	00 01 0D 00 57 BB 35 3C 2F CE 4D BD 21 4A 71 BE 4B 71 78 3F 28 33 01 3F 2F 8B A0 3E 04 37 50 3F 61 68 37 BE 4B 9B 58 BD E5 4F 1D C1 (shown as hex)

6.14 Quaternion, Acceleration and Angular Rates

This register contains ten values representing a quaternion vector, which represents the module's orientation, followed by the acceleration and angular rate measurements. The quaternion is a 4x1 unit vector. The ordering of this register's ten values is shown below. All ten numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Acceleration	m/s^2	%+07.3f
6	Y-Axis Acceleration	m/s^2	%+07.3f
7	Z-Axis Acceleration	m/s^2	%+07.3f
8	X-Axis Angular Rate	rad/s	%+08.4f
9	Y-Axis Angular Rate	rad/s	%+08.4f
10	Z-Axis Angular Rate	rad/s	%+08.4f

6.14.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion, acceleration, and angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,14*76
UART Read Response	$VNRRG,14, +0.011240,-0.050222,-0.235011,+0.970629,-00.191,-00.082,-09.826,-00.0064,-00.0064,+00.8756*6C
SPI Read Request (42 bytes)	01 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (42 bytes)	00 01 0E 00 03 27 38 3C C2 B5 4D BD AE A6 70 BE 2A 7B 78 3F A5 FA 43 BE F9 B2 A7 BD F8 37 1D C1 00 21 D3 BB 00 5A D0 BB C6 26 60 3F (shown as hex)

6.15 Quaternion, Magnetic, Acceleration and Angular Rates

This register contains thirteen values representing a quaternion vector, which represents the module's orientation, followed by the magnetic, acceleration and angular rate measurements. The quaternion is a 4x1 unit vector. The ordering of this register's thirteen values is shown below. All thirteen numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	q[0] of quaternion	N/A	%+09.6f
2	q[1] of quaternion	N/A	%+09.6f
3	q[2] of quaternion	N/A	%+09.6f
4	q[3] of quaternion (scalar term)	N/A	%+09.6f
5	X-Axis Magnetic	N/A	%+07.4f
6	Y-Axis Magnetic	N/A	%+07.4f
7	Z-Axis Magnetic	N/A	%+07.4f
8	X-Axis Acceleration	m/s^2	%+07.3f
9	Y-Axis Acceleration	m/s^2	%+07.3f
10	Z-Axis Acceleration	m/s^2	%+07.3f
11	X-Axis Angular Rate	rad/s	%+08.4f
12	Y-Axis Angular Rate	rad/s	%+08.4f
13	Z-Axis Angular Rate	rad/s	%+08.4f

6.15.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current quaternion, magnetic, acceleration, and angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,15*77
UART Read Response	$VNRRG,15,+0.010957,-0.050125,-0.235323,+0.970562,+0.5058,+0.3132,+0.8123,-00.210,-00.036,-09.833,+00.0117,-00.0061,+00.8732*54
SPI Read Request (56 bytes)	`01 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)`
SPI Read Response (56 bytes)	`00 01 0F 00 8E 86 33 3C 43 50 4D BD 67 F8 70 BE C1 76 78 3F 74 7A 01 3F DA 5C A0 3E 62 F0 4F 3F 90 19 57 BE E3 D6 14 BD 4B 53 1D C1 40 B1 3F 3C 00 BC C7 BB D0 8C 5F 3F (shown as hex)`

6.16 Attitude (Directional Cosine Orientation Matrix)

This register contains the attitude directional cosine matrix. This matrix is a valid 3x3 rotation matrix. Nine parameters are returned from this command, and the terms are mapped to a 3x3 matrix as follows,

$$DCM = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

The ordering of this register's nine values is shown below. All nine numbers are represented as floating point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	1st Row, 1st Column	N/A	%+13.6E
2	1st Row, 2nd Column	N/A	%+13.6E
3	1st Row, 3rd Column	N/A	%+13.6E
4	2nd Row, 1st Column	N/A	%+13.6E
5	2nd Row, 2nd Column	N/A	%+13.6E
6	2nd Row, 3rd Column	N/A	%+13.6E
7	3rd Row, 1st Column	N/A	%+13.6E
8	3rd Row, 2nd Column	N/A	%+13.6E
9	3rd Row, 3rd Column	N/A	%+13.6E

6.16.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current directional cosine matrix. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,16*74
UART Read Response	$VNRRG,16,+8.839433E-01,-4.584605E-01,+9.196996E-02,+4.562554E-01,+8.887121E-01,+4.496603E-02,-1.023500E-01,+2.214367E-03,+9.947461E-01*03
SPI Read Request (40 bytes)	`01 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)`
SPI Read Response (40 bytes)	`00 01 10 00 1B 4A 62 3F 55 BB EA BE BF 5A BC 3D 4F 9A E9 3E A3 82 63 3F 4E 2E 38 3D DB 9C D1 BD E8 1E 11 3B AE A7 7E 3F (shown as hex)`

6.17 Magnetic Measurements

This register contains three values representing a non-dimensional 3x1 magnetic measurement vector. The ordering of this register's three values is shown below. All three numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	X-Axis Magnetic	N/A	%+07.4f
2	Y-Axis Magnetic	N/A	%+07.4f
3	Z-Axis Magnetic	N/A	%+07.4f

6.17.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current magnetic measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,17*75
UART Read Response	$VNRRG,+0.5051,+0.3146,+0.8139*44
SPI Read Request (16 bytes)	`01 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (16 bytes)	`00 01 11 00 CE 4F 01 3F 2F 0E A1 3E B9 5B 50 3F  (shown as hex)`

6.18 Acceleration Measurements

This register contains three values representing a dimensional 3x1 acceleration measurement vector. The ordering of this register's three values is shown below. All three numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	X-Axis Acceleration	m/s^2	%+07.3f
2	Y-Axis Acceleration	m/s^2	%+07.3f
3	Z-Axis Acceleration	m/s^2	%+07.3f

6.18.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current acceleration measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,18*7A
UART Read Response	$VNRRG,18,-00.203,-00.055,-09.815*61
SPI Read Request (16 bytes)	`01 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (16 bytes)	`00 01 12 00 BF 0D 50 BE CD C4 62 BD DD 09 1D C1  (shown as hex)`

6.19 Angular Rate Measurements

This register contains three values representing a dimensional 3x1 angular rate measurement vector. The ordering of this register's three values is shown below. All three numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	X-Axis Angular Rate	rad/s	%+08.4f
2	Y-Axis Angular Rate	rad/s	%+08.4f
3	Z-Axis Angular Rate	rad/s	%+08.4f

6.19.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the

VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,19*7B
UART Read Response	$VNRRG,19,+00.0035,-00.0060,+00.8632*5B
SPI Read Request (16 bytes)	`01 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (16 bytes)	`00 01 13 00 00 C3 67 3B C0 80 C3 BB 2C FA 5C 3F  (shown as hex)`

6.20 Magnetic, Acceleration and Angular Rates

This register contains nine values representing the non-dimensional magnetic, dimensional acceleration, and the dimensional angular rate measurement vectors. The ordering of this register's nine values is shown below. All nine numbers are represented as fixed point. This is a read-only register.

Field #	Meaning	Units	Printf/Scanf format
1	X-Axis Magnetic	N/A	%+07.4f
2	Y-Axis Magnetic	N/A	%+07.4f
3	Z-Axis Magnetic	N/A	%+07.4f
4	X-Axis Acceleration	m/s^2	%+07.3f
5	Y-Axis Acceleration	m/s^2	%+07.3f
6	Z-Axis Acceleration	m/s^2	%+07.3f
7	X-Axis Angular Rate	rad/s	%+08.4f
8	Y-Axis Angular Rate	rad/s	%+08.4f
9	Z-Axis Angular Rate	rad/s	%+08.4f

6.20.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current magnetic, acceleration, and angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,20*71
UART Read Response	$VNRRG,20,+0.5056,+0.3139,+0.8138,-00.220,-00.043,-09.833,+00.0038,-00.0060,+00.8593*5C
SPI Read Request (40 bytes)	01 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (40 bytes)	00 01 14 00 24 6E 01 3F 9B B7 A0 3E 50 54 50 3F D0 89 61 BE 10 A5 2E BD 53 53 1D C1 00 87 77 3B C0 0C C3 BB 56 F8 5B 3F (shown as hex)

6.21 Magnetic and Gravity Reference Vectors

This register contains six values representing the non-dimensional magnetic reference vector and the dimensional gravity reference vector. The vectors are represented in the NED (North, East, Down) reference frame. The ordering of this register's six values is shown below. All six numbers are represented by single-precision floating points.

Field #	Meaning	Units	Printf/Scanf format
1	X-Axis Magnetic Reference	N/A	%+13.6E
2	Y-Axis Magnetic Reference	N/A	%+13.6E
3	Z-Axis Magnetic Reference	N/A	%+13.6E
4	X-Axis Gravity Reference	m/s^2	%+13.6E
5	Y-Axis Gravity Reference	m/s^2	%+13.6E
6	Z-Axis Gravity Reference	m/s^2	%+13.6E

6.21.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the current magnetic and gravity reference vectors. The VN-100 then responds back with the requested vectors. In the write example the user requests that the magnetic reference vector be set to [0.1, 0.2, 0.3] and the acceleration reference vector be set to [0.4, 0.5, 0.6]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,21*70
UART Read Response	$VNRRG,21,+1.043658E+00,+8.135935E-02,+1.811239E+00,+0.000000E+00,+0.000000E+00,-9.793746E+00*7F
UART Write Request	$VNWRG,21,0.1,0.2,0.3,0.4,0.5,0.6*72
UART Write Response	$VNWRG,21,+1.000000E-01,+2.000000E-01,+3.000000E-01,+4.000000E-01,+5.000000E-01,+6.000000E-01*72
SPI Read Request (28 bytes)	01 15 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (28 bytes)	00 01 15 00 9A 96 85 3F BB 9F A6 3D AE D6 E7 3F 00 00 00 00 00 00 00 00 2F B3 1C C1 (shown as hex)
SPI Write Request (28 bytes)	02 15 00 00 CD CC CC 3D CD CC 4C 3E 9A 99 99 3E CD CC CC 3E 00 00 00 3F 9A 99 19 3F (shown as hex)
SPI Write Response (28 bytes)	00 02 15 00 CD CC CC 3D CD CC 4C 3E 9A 99 99 3E CD CC CC 3E 00 00 00 3F 9A 99 19 3F (shown as hex)

6.22 Filter Measurements Variance Parameters

This register contains ten values representing the statistical variance of various measured terms used by the Extended Kalman Filter. All ten numbers are represented by single-precision floating points.

Field #	Meaning	Units	Printf/Scanf format
1	Variance - Angular Walk	N/A	%+13.6E
2	Variance – X Axis Angular Rate	$\left(\frac{rad}{s}\right)^2$	%+13.6E
3	Variance – Y Axis Angular Rate	$\left(\frac{rad}{s}\right)^2$	%+13.6E
4	Variance – Z Axis Angular Rate	$\left(\frac{rad}{s}\right)^2$	%+13.6E
5	Variance – X Axis Magnetic	N/A	%+13.6E
6	Variance – Y Axis Magnetic	N/A	%+13.6E
7	Variance – Z Axis Magnetic	N/A	%+13.6E
8	Variance – X Axis Acceleration	$\left(\frac{m}{s^2}\right)^2$	%+13.6E
9	Variance – Y Axis Acceleration	$\left(\frac{m}{s^2}\right)^2$	%+13.6E
10	Variance – Z Axis Acceleration	$\left(\frac{m}{s^2}\right)^2$	%+13.6E

6.22.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the filter measurement variance parameters. The VN-100 then responds back with the requested parameters. In the write example the user requests that the filter variance parameters be set to [0.1, 0.2, 0.3] and the acceleration reference vector be set to [0.4, 0.5, 0.6]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,22*73
UART Read Response	$VNRRG,22,+1.000000E-06,+4.968087E-06,+4.112245E-06,+1.775172E-04,+2.396688E-06,+3.198908E-06,+1.389186E-05,+6.620304E-05,+4.869544E-05,+1.560575E-04*52
UART Write Request	$VNWRG,22,1E-12,1E-9,1E-9,1E-9,2E-6,2E-6,2E-6,5E-5,5E-5,5E-5*48
UART Write Response	$VNWRG,22,+1.000000E-12,+1.000000E-09,+1.000000E-09,+1.000000E-09,+2.000000E-06,+2.000000E-06,+2.000000E-06,+5.000000E-05,+5.000000E-05,+5.000000E-05*78
SPI Read Request (44 bytes)	01 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (44 bytes)	00 01 16 00 BD 37 86 35 8B B3 A6 36 EB FB 89 36 EC 23 3A 39 CA D6 20 36 D2 AC 56 36 18 11 69 37 7D D6 8A 38 55 3E 4C 38 5B A3 23 39 (shown as hex)
SPI Write Request (44 bytes)	02 16 00 00 CC BC 8C 2B 5F 70 89 30 5F 70 89 30 5F 70 89 30 BD 37 06 36 BD 37 06 36 BD 37 06 36 17 B7 51 38 17 B7 51 38 17 B7 51 38 (shown as hex)
SPI Write Response (44 bytes)	00 02 16 00 CC BC 8C 2B 5F 70 89 30 5F 70 89 30 5F 70 89 30 BD 37 06 36 BD 37 06 36 BD 37 06 36 17 B7 51 38 17 B7 51 38 17 B7 51 38 (shown as hex)

6.23 Magnetic Hard/Soft Iron Compensation Parameters

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.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} S11 & S12 & S13 \\ S21 & S22 & S23 \\ S31 & S32 & S33 \end{bmatrix} \cdot \begin{Bmatrix} MX - H1 \\ MY - H2 \\ MZ - H3 \end{Bmatrix}$$

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.

Field #	Meaning	Units	Printf/Scanf format
1	S11	N/A	%+13.6E
2	S12	N/A	%+13.6E
3	S13	N/A	%+13.6E
4	S21	N/A	%+13.6E
5	S22	N/A	%+13.6E
6	S23	N/A	%+13.6E
7	S31	N/A	%+13.6E
8	S32	N/A	%+13.6E
9	S33	N/A	%+13.6E
10	H1	N/A	%+13.6E
11	H2	N/A	%+13.6E
12	H3	N/A	%+13.6E

6.23.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the magnetic hard/soft iron parameters. The VN-100 then responds back with the requested parameters. In the write example the user requests that the hard/soft iron parameters be set to [1, 0.01, 0.01, -0.02, 1, 0, -0.1, 0.1, 1, 0.2, 0.1, -0.1]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,23*72
UART Read Response	$VNRRG,23,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00*73
UART Write Request	$VNWRG,23,1,0.01,0.01,-0.02,1,0,-0.1,0.1,1,0.2,0.1,-0.1*6B
UART Write Response	$VNWRG,23,+1.000000E+00,+1.000000E-02,+1.000000E-02,-2.000000E-02,+1.000000E+00,+0.000000E+00,-1.000000E-01,+1.000000E-01,+1.000000E+00,+2.000000E-01,+1.000000E-01,-1.000000E-01*73
SPI Read Request (52 bytes)	01 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (52 bytes)	00 01 17 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Write Request (52 bytes)	02 17 00 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80 3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F CD CC 4C 3E CD CC CC 3D CD CC CC BD (shown as hex)
SPI Write Response (52 bytes)	00 02 17 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80 3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F CD CC 4C 3E CD CC CC 3D CD CC CC BD (shown as hex)

6.24 Filter Active Tuning Parameters

This register contains four values representing the active tuning gains for the Kalman Filter. See section 11.3 for more information on the meaning of these parameters. All four numbers are represented by single-precision floating points.

Field #	Meaning	Units	Range	Printf/Scanf format
1	Magnetic Disturbance Gain	N/A	0 to 10000	%+13.6E
2	Acceleration Disturbance Gain	N/A	0 to 10000	%+13.6E
3	Magnetic Disturbance Memory	N/A	0 to 1	%+13.6E
4	Acceleration Disturbance Memory	N/A	0 to 1	%+13.6E

6.24.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the filter active tuning parameters. The VN-100 then responds back with the requested parameters. In the write example the user requests that the filter active tuning parameters be set to [1, 1, 0.99, 0.99]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,24*75
UART Read Response	$VNRRG,24,+0.000000E+00,+0.000000E+00,+9.900000E-01,+9.900000E-01*75
UART Write Request	$VNWRG,24,1,1,0.99,0.99*70
UART Write Response	$VNWRG,24,+1.000000E+00,+1.000000E+00,+9.900000E-01,+9.900000E-01*70
SPI Read Request (20 bytes)	01 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (20 bytes)	00 01 18 00 00 00 00 00 00 00 00 00 A4 70 7D 3F A4 70 7D 3F (shown as hex)
SPI Write Request (20 bytes)	02 18 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Write Response (20 bytes)	00 02 18 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)

6.25 Accelerometer Compensation

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.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C11 & C12 & C13 \\ C21 & C22 & C23 \\ C31 & C32 & C33 \end{bmatrix} \cdot \begin{Bmatrix} AX - B1 \\ AY - B2 \\ AZ - B3 \end{Bmatrix}$$

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.

Field #	Meaning	Units	Data Format	Printf/Scanf format
1	C11	N/A	Floating-point	%+13.6E
2	C12	N/A	Floating-point	%+13.6E
3	C13	N/A	Floating-point	%+13.6E
4	C21	N/A	Floating-point	%+13.6E
5	C22	N/A	Floating-point	%+13.6E
6	C23	N/A	Floating-point	%+13.6E
7	C31	N/A	Floating-point	%+13.6E
8	C32	N/A	Floating-point	%+13.6E
9	C33	N/A	Floating-point	%+13.6E
10	B1	N/A	Floating-point	%+13.6E
11	B2	N/A	Floating-point	%+13.6E
12	B3	N/A	Floating-point	%+13.6E

6.25.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the accelerometer compensation parameters. The VN-100 then responds back with the requested parameters. In the write example the user requests that the accelerometer compensation parameters be set to [1, 0.01, 0.01, -0.02, 1, 0, -0.1, 0.1, 1, 0.2, 0.1, -0.1]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,25*74
UART Read Response	$VNRRG,25,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00*75
UART Write Request	$VNWRG,25,1,0.01,0.01,-0.02,1,0,-0.1,0.1,1,0.2,0.1,-0.1*6D
UART Write Response	$VNWRG,25,+1.000000E+00,+1.000000E-02,+1.000000E-02,-2.000000E-02,+1.000000E+00,+0.000000E+00,-1.000000E-01,+1.000000E-01,+1.000000E+00,+2.000000E-01,+1.000000E-01,-1.000000E-01*75
SPI Read Request (52 bytes)	01 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Read Response (52 bytes)	00 01 19 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 (shown as hex)
SPI Write Request (52 bytes)	02 19 00 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80 3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F CD CC 4C 3E CD CC CC 3D CD CC CC BD (shown as hex)
SPI Write Response (52 bytes)	00 02 19 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80 3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F CD CC 4C 3E CD CC CC 3D CD CC CC BD (shown as hex)

6.26 Reference Frame Rotation

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 circuit board.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_U = \begin{bmatrix} C11 & C12 & C13 \\ C21 & C22 & C23 \\ C31 & C32 & C33 \end{bmatrix} \cdot \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B$$

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. It is crucial that these two frames of reference be rigidly attached to each other. All nine numbers are represented by single-precision floating points.

Field #	Meaning	Units	Printf/Scanf format
1	C11	N/A	%+13.6E
2	C12	N/A	%+13.6E
3	C13	N/A	%+13.6E
4	C21	N/A	%+13.6E
5	C22	N/A	%+13.6E
6	C23	N/A	%+13.6E
7	C31	N/A	%+13.6E
8	C32	N/A	%+13.6E
9	C33	N/A	%+13.6E

6.26.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the reference frame rotation parameters. The VN-100 then responds back with the requested parameters. In the write example the user requests that the reference frame rotation parameters be set to [1, 0.01, 0.01, -0.02, 1, 0, -0.1, 0.1, 1]. The VN-100 then responds to confirm that the register has been set.

Example Type	Message
UART Read Request	$VNRRG,26*77
UART Read Response	$VNRRG,26,+1.000000E+00,+0.000000E+00,+0.000000E+00,+0.000000E+00,+1.000000E+00,+0.000000E+00, +0.000000E+00,+0.000000E+00,+1.000000E+00*01
UART Write Request	$VNWRG,26,1,0.01,0.01,-0.02,1,0,-0.1,0.1,1*43
UART Write Response	$VNWRG,26,+1.000000E+00,+1.000000E-02,+1.000000E-02,-2.000000E-02,+1.000000E+00,+0.000000E+00,-1.000000E-01,+1.000000E-01,+1.000000E+00*02
SPI Read Request (40 bytes)	`01 1A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00` `00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  (shown as hex)`
SPI Read Response (40 bytes)	`00 01 1A 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80` `3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 3F  (shown as hex)`
SPI Write Request (40 bytes)	`02 1A 00 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80` `3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F  (shown as hex)`
SPI Write Response (40 bytes)	`00 02 1A 00 00 00 80 3F 0A D7 23 3C 0A D7 23 3C 0A D7 A3 BC 00 00 80` `3F 00 00 00 00 CD CC CC BD CD CC CC 3D 00 00 80 3F  (shown as hex)`

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

This register contains twelve values representing the module's yaw, pitch, roll, magnetic, acceleration, and angular rate measurements. The fields of this register are represented as fixed point.

Field #	Meaning	Units	Printf/Scanf format
1	Yaw Angle	degrees	%+07.2f
2	Pitch Angle	degrees	%+07.2f
3	Roll Angle	degrees	%+07.2f
4	X-Axis Magnetic	N/A	%+07.4f
5	Y-Axis Magnetic	N/A	%+07.4f
6	Z-Axis Magnetic	N/A	%+07.4f
7	X-Axis Acceleration	m/s^2	%+07.3f
8	Y-Axis Acceleration	m/s^2	%+07.3f
9	Z-Axis Acceleration	m/s^2	%+07.3f
10	X-Axis Angular Rate	rad/s	%+08.4f
11	Y-Axis Angular Rate	rad/s	%+08.4f
12	Z-Axis Angular Rate	rad/s	%+08.4f

6.27.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the example the user requests the yaw, pitch, roll, magnetic, acceleration, and angular rate measurements. The VN-100 then responds back with the requested measurements.

Example Type	Message
UART Read Request	$VNRRG,27*76
UART Read Response	$VNRRG,27,-027.56,-005.30,+002.62,+0.5045,+0.3158,+0.8140,-00.197,-00.082,-09.830,-00.0182,-00.0059,+00.8308*43
SPI Read Request (52 bytes)	01 1B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
SPI Read Response (52 bytes)	00 01 1B 00 D6 83 DC C1 EA AB A9 C0 0A C7 27 40 BA 26 01 3F 23 B4 A1 3E 3C 5F 50 3F 5D 91 49 BE 57 88 A8 BD 24 47 1D C1 20 C2 94 BC 80 6A C0 BB 36 B0 54 3F

6.28 Accelerometer Gain

This register contains one value that allows for the adjustment of the accelerometer gain. The user can select between either ±2g or ±6g. The setting will take effect immediately; however the accelerometer output is not guaranteed to be within spec until 10ms after the transition due to transient effects of the switching gain and the effects of the low pass filters. After at least 10ms of switching the gain, the accelerometer should read the correct measurement. The accelerometer is independently calibrated at both ±2g and ±6g for maximum accuracy. The field of this register is represented as an unsigned 32-bit integer and can have only the two listed values.

Table 10 – Accelerometer Gain

Register Value	Accelerometer Gain
0	±2g
1	±6g

6.28.1 Example Commands

The following table shows example read and write request commands for the given register and their subsequent responses from the VN-100. A request is defined as a message sent from the user to the VN-100, while a response is a message sent from the VN-100 to the user. Both the UART and SPI examples perform the same action. In the read example the user requests the current accelerometer gain. The VN-100 then responds back that the gain is ±2g. In the write example the user sets the accelerometer gain to ±6g. The VN-100 then responds back with the updated accelerometer gain.

Example Type	Message
UART Read Request	$VNRRG,27*76
UART Read Response	$VNRRG,27,-027.56,-005.30,+002.62,+0.5045,+0.3158,+0.8140,-00.197,-00.082,-09.830,-00.0182,-00.0059,+00.8308*43
SPI Read Request (8 bytes)	01 1C 00 00 00 00 00 00
SPI Read Response (8 bytes)	00 01 1C 00 00 00 00 00
SPI Write Request (8 bytes)	02 1C 00 00 01 00 00 00
SPI Write Response (8 bytes)	00 02 1C 00 01 00 00 00

7 Default Factory Settings

The following table details the VN-100 module's settings as it is delivered from the factory. These settings may be restored by issuing a Restore Factory Settings command (Section 5.2.4) or by using the Restore Factory Settings signal pins.

Table 11 – Factory Default Register Values

Settings Name	Default Factory Value
Serial Baud Rate	9600
Async Data Output Frequency	10 Hz
Async Data Output Type	Attitude (Yaw, Pitch, Roll Format)
Magnetic and Gravity Reference Vectors	+1.0436585e+000 +8.1359350e-002 +1.8112390e+000 +0.00e+0 +0.00e+0 -9.793746e+0
Filter Measurement Variance Parameters	+1.0e-6 +4.968087236542740e-006 +4.112245302664222e-006 +1.775172462044985e-004 +2.396688069825628e-006 +3.198907908235179e-006 +1.389186225936592e-005 +6.620304667228608e-005 +4.869544197003338e-005 +1.560574584667775e-004
Magnetic Hard/Soft Iron Compensation Parameters	1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
Filter Active Tuning Parameters	0.0 0.0 0.99 0.99
Acceleration Compensation Parameters	1.0 0.0 0.0 0.0 1.0

	0.0 0.0 0.0 1.0 0.0 0.0 0.0
Reference Frame Rotation	1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0
Accelerometer Gain	±2g

8 Performance Specifications

Table 12 – Performance Specifications

Heading	
Range (°)	±180
Accuracy[1] (° rms)	< 2.0
Resolution (°)	< 0.2

Attitude	
Range: Pitch, Roll (°)	±180, ±90
Accuracy[1] (° rms)	< 0.5
Resolution (°)	< 0.06

Angular Rate	
Range: Heading (°/sec)	±300
Range: Pitch, Roll (°/sec)	±500
Bias Stability[1]: Heading (°/sec)	< ±0.1 @
Bias Stability[1]: Pitch, Roll	< ±0.06 @
Resolution: Heading (°/sec)	< ±0.2
Resolution: Pitch, Roll (°/sec)	< ±0.06
Bandwidth: Heading (Hz)	80
Bandwidth: Pitch, Roll (Hz)	140

Acceleration	
Input Range[2]: X/Y/Z (g)	±2g, ±6g
Bias Stability[1]: X/Y (mg)	< 0.5 @
Bias Stability[1]: X/Z (mg)	< 1.6 @
Resolution: X/Y (mg)	< 0.4
Resolution: Z (mg)	< 2
Bandwidth (Hz)	50

[1]Specified at room temperature only

[2]User selectable

9 General Specifications

Table 13 - Electrical Characteristics

Parameter	Conditions	Min	Typical	Max	Units
Power Supply					
Operating Voltage Range, V_{cc}		3.1	5.0	5.5	V
Power Supply Current	@ 25°C		65		mA
Start-Up Time					
Valid Data			< 1		sec
Fully Stabilized Data			< 30		sec
Flash Memory	Endurance	10000			cycles
Data Retention		10			years
Input Port Characteristics					
Low level voltage		-0.5		0.8	V
High level voltage		2		5.5	V
Tare signal pulse width		10			ns
NRST Input low level voltage		-0.5		0.8	V
NRST Input high level voltage		2		5.5	V
Output Port Characteristics					
Low level voltage				0.4	V
High level voltage		2.6			V
High to low level fall time				8	ns
Low to high level rise time				8	ns
SPI Interface characteristics					
Clock frequency	f_{SCK}	0		18	MHz
Clock rise and fall time	$T_{r(SCK)}$	0		8	ns
NSS setup time	$t_{SU(NSS)}$	112			ns
NSS hold time	$T_{h(NSS)}$	73			ns
SCK high and low time	$T_{w(SCKH)}$, $T_{w(SCKL)}$	50	60		ns
Data input setup time	$T_{su(SI)}$	1			ns
Data input hold time	$T_{h(SI)}$	3			ns
Data output access time	$T_{a(SO)}$	0		55	ns
Data output disable time	$T_{dis(SO)}$	10			ns
Data output valid time	$T_{v(SO)}$			25	ns
Data output hold time	$T_{h(SO)}$	4			ns

Table 14 – Temperature Ranges

Parameter	Conditions	Min	Typical	Max	Units
Operating Temperature Range		-20		60	C
Storage Temperature Range		-40		85	C

Table 15 – Absolute Maximum Ratings

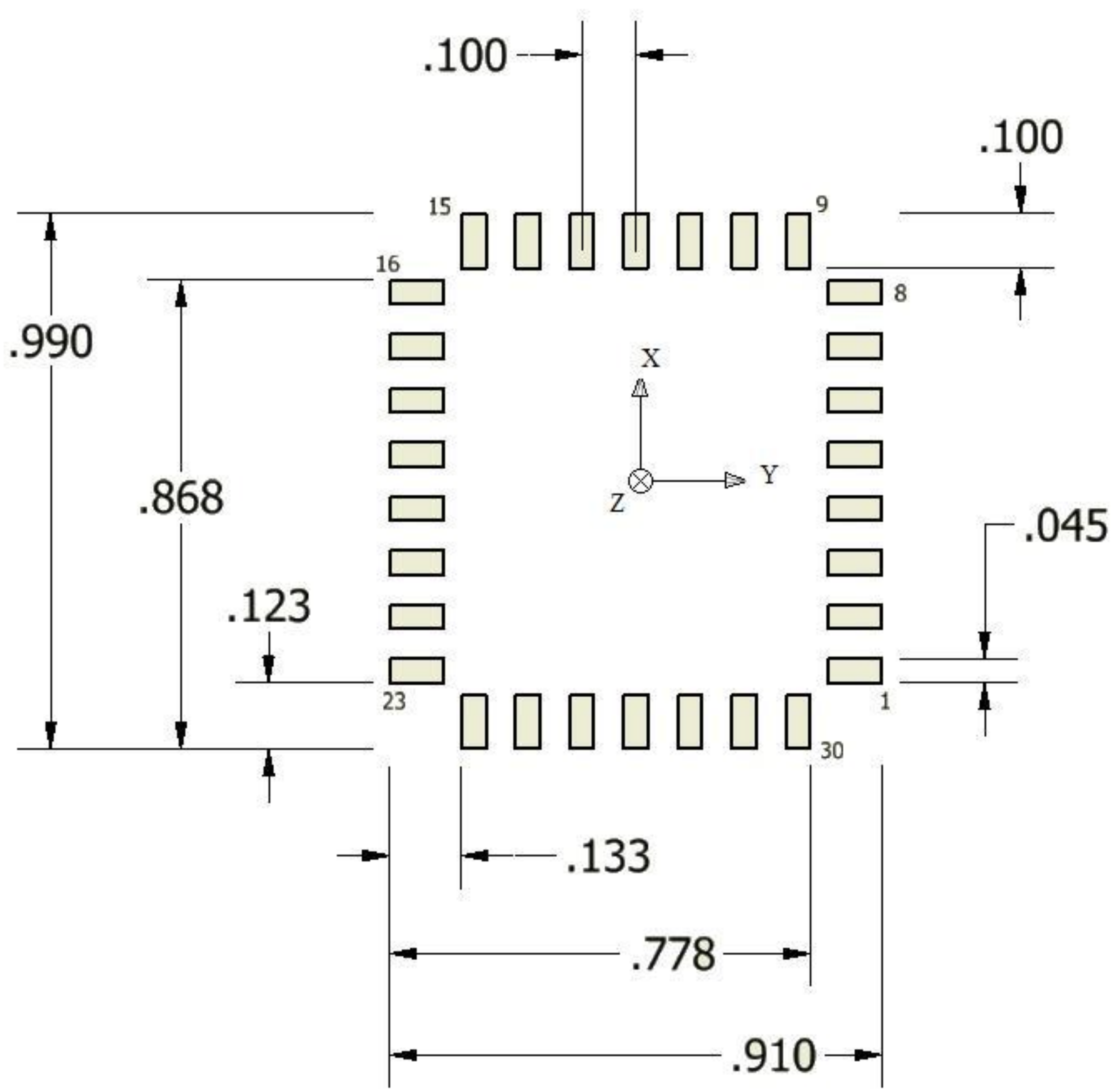
Parameter	Conditions	Min	Typical	Max	Units
Shock limit – Powered				1000	G
Shock limit – Unpowered				500	g
VCC to GND		0		6.0	V
Digital I/O Voltage to GND		-3.3		5.5	V

Notes:

1. Specifications are subject to change at any time without notice.
2. Exceeding the absolute maximum rating may damage the device.
3. The device is not guaranteed to function outside its operating range.
4. Device is ESD sensitive. Handling precautions recommended. Human body model, 1.5k in series with 100pF.
5. Hand soldering recommended.

10 Recommended Land Layout

Figure 6 – PCB Footprint



11 Intro to AHRS and the VN-100 Kalman Filter

11.1 AHRS background

AHRS and IMU technology has effectively been reserved for high end aerospace and military applications until the most recent, last 10 years or so, introduction of MEMS technology. This new technology has by now become commonplace in civilian applications worldwide and the combination of MEMS accelerometers, magnetometers and gyroscopes enable the small and low cost orientation solutions now possible.

Until the most recent advent of MEMS devices the solution to the orientation problem typically relied on plum lines, bubble levels and magnetic compass for heading.

The VN-100 combines measurements of angular rate with the magnetic and gravitational fields to continuously solve for the orientation solution of the body frame. The solution of orientation is possible with the help of a Kalman filter using "known" reference vectors for the gravitational and magnetic fields. These reference vectors can be programmed into solid state memory such that the VN-100 can reference any orientation previously set. Furthermore the Kalman filter has a various parameters that can be used to optimize the filtering performance for specific applications.

11.2 Filter tuning parameters

Tuning of Kalman filters has always been associated with trial and error, and some will argue that this is an art. The VN-100 comes preloaded with a factory set default set of filter parameters. These are designed for generic use and will perform nominally well in many circumstances. Register 22 on the VN-100 holds ten coefficients that do have statistical significance. These represent the standard deviation of the individual sensors output – squared. If one would like to place less emphasis on a specific sensor then one could increase the variance, alternatively if one would like to place significant value on a specific sensor one can accomplish this by reducing the value set for the variance of that sensor. The filter random walk parameter weights how fast the gyro bias estimates converge. Setting this parameter to low will make the filter tracking of the gyro drift extremely slow. While setting this value high will make the filter track even the noise in the gyro and not much benefit is added by having the gyro measurements at all. The default factory settings can be used as a guide for this advanced/low-level tuning of the VN-100 Kalman filter option.

11.2.1 Example tuning parameters

Provided below are example tuning parameters for four separate example cases. A description of each is case is provided along with the parameters that are recommended for this case. Note that the parameters given below will work better for each case than the factory default parameters and are useful as a good starting point for further refinement and tuning. It is ultimately up to the user to tune the filter to be able to get optimum performance for their specific application.

11.2.1.1 Case 1 – Nominal

The nominal case is considered as the baseline state for further filter tuning. In this case all measurements are given equal weight. This case is somewhat similar in performance to the factory default parameters. Some attributes to note about this case is that the performance of the device will be highly sensitive to the accuracies of both the acceleration and magnetic measurements. The gyros are only loosely trusted, while a high confidence is placed in both vector measurements from the magnetometer and accelerometer. As a result the device will experience significant errors in attitude in the presence of dynamic and magnetic disturbances.

Example Type	Message
UART Write Request	$VNWRG,22,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6*76
SPI Write Request (44 bytes)	02 16 00 00 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 (shown as hex)

11.2.1.2 Case 2 – More trust in gyros

This case provides overall better performance than the nominal or factory default for many applications. Essentially this selection of tuning parameters places a higher confident in the gyros, while providing equal trust between the magnetometer and accelerometer measurements. The only downside to this case is that since there is more trust in the gyros, if the gyros become saturated and the attitude solution becomes significantly in error, then it will likely take a few seconds for the filter to converge back on the correct attitude. If you have an application where you know the gyros will not saturate then this selection of tuning parameters is definitely worth trying. It will provide better immunity to both dynamic and magnetic disturbances than the nominal case.

Example Type	Message
UART Write Request	$VNWRG,22,1E-9,1E-9,1E-9,1E-9,1E-6,1E-6,1E-6,1E-6,1E-6,1E-6*76
SPI Write Request (44 bytes)	02 16 00 00 5F 70 89 30 5F 70 89 30 5F 70 89 30 5F 70 89 30 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 BD 37 86 35 (shown as hex)

11.2.1.3 Case 3 – Tune out magnetometers

Many of our customers are looking for ways to tune out the effect of the magnetometer from the attitude solution. An example case where this would be beneficial is for customers that are looking for an accurate gyro-stabilized inclinometer solution where the magnetic field cannot be trusted. For this case the heading angle is not of importance, only the pitch and roll angles need to be accurately measured. Also if there is a need to place the VN-100 in close proximity to a running electric motor, then the magnetic field from the motor will likely affect the readings on the VN-100 in an unpredictable manner. In this case the user would want to tune out the magnetometer from the attitude solution. To do this essentially all you need to do is significantly increase the variance (reduce trust) in the magnetometer to a point where it has little, if any, affect on the attitude solution.

Example Type	Message
UART Write Request	$VNWRG,22,1E-9,1E-9,1E-9,1E-9,1E2,1E2,1E2,1E-6,1E-6,1E-6*5F
SPI Write Request (44 bytes)	02 16 00 00 5F 70 89 30 5F 70 89 30 5F 70 89 30 5F 70 89 30 00 00 C8 42 00 00 C8 42 00 00 C8 42 BD 37 86 35 BD 37 86 35 BD 37 86 35 (shown as hex)

11.2.1.4 Case 3 – Reduce effect of vibration

In many cases you may have measureable vibration induced by nearby motors that wreck havoc on the accelerometer measurements of the VN-100 using the normal factory default settings. Since the default settings place a large confidence on the accelerometers, it is likely that the vibration will produce errors in the attitude that are larger than desired. In order to combat this problem you essentially want to tune the filter to have increased confidence in the gyros, while having less confidence in the accelerometers. As long as you can get good readings from your magnetometer, the accuracies of your attitude solution will likely remain quite good even in the presence of significant vibration.

Example Type	Message
UART Write Request	$VNWRG,22,1E-9,1E-9,1E-9,1E-9,1E-6,1E-6,1E-6,1E-3,1E-3,1E-3*73
SPI Write Request (44 bytes)	02 16 00 00 5F 70 89 30 5F 70 89 30 5F 70 89 30 5F 70 89 30 BD 37 86 35 BD 37 86 35 BD 37 86 35 6F 12 83 3A 6F 12 83 3A 6F 12 83 3A (shown as hex)

11.3 Active filter tuning parameters

The active filter tuning parameters are set in Register 24 on the VN-100 module. These are designed for ease of use, and to present a more intuitive way of tuning the AHRS. The purpose of these parameters is to smooth the solution in the presence of disturbances to the magnetic and gravitational fields. The need for this arise when one consider that the VN-100 is solving for orientation at a rate of 200 Hz. Even holding the module in ones hand will induce disturbances to the sensed acceleration due to the earth. The filter makes the assumption that the sensed acceleration is aligned with the earths' gravity field direction, and clearly this assumption fails when the device is subjected to motion. To compensate for this a dynamic variance term is introduced on the accelerometers, this allows the Kalman filter to disregard the components of the acceleration that is not constant. Similarly for the magnetic field vector one can set the gain on its' dynamic variance term and adjust its sensitivity to non-constant readings.

The register holds four values, the first is the gain on the dynamic variance term on the magnetic reading, and the second holds the gain for the dynamic variance term on the acceleration reading. The third and fourth values in the register emulate a memory effect. E.g. if the filter detects strong disturbances it will remember this for a short time. The third value sets this parameter for the magnetic reading and the fourth sets this for the acceleration reading. These memory coefficients acts by reducing the dynamic variance term by a fraction on each filter cycle, thus the third and fourth value should ALWAYS be set to a number between 0 and 1 (If used this is normally set fairly close to 1). An example value for Register 24 could be [0.50 1.00 0.99 0.99].

11.4 Utilizing the "Tare" command

The "Tare" (command ID 5) command is a special command used to establish a set of reference vectors for the Kalman filter. This function when used will set the VN-100 into a state where it samples and averages the magnetic field and acceleration for approximately one second. These two vectors then get programmed as reference vectors for the filter. During the execution of the "Tare" command the VN-100 also zeros out the orientation and sets the gyro bias. This function furthermore sets the magnitude information for the magnetic field used in the active tuning. If active tuning is used it is recommended that the sensor is "Tared" in order to increase the accuracy of the active tuning.

It is important that the VN-100 is not moving during the execution of the Tare command. Also, note that many tables have relatively strong magnetic fields associated with metal supports, and if "Tare" is used when near a non-uniform magnetic field it will record an incorrect magnetic reference vector. This can lead to poor performance of the attitude solution.

It is recommended to use the "Tare" command in order to achieve the most accurate relative measurement.

11.5 Setting the Reference vectors for Absolute Heading Referencing

The VN-100 uses the North/East/Down Earth Fixed coordinate system. When used in an absolute heading system the Yaw angle (heading) is referenced to true north, pitch and roll are referenced to the local horizontal. The heading accuracy will depend on the quality of the local magnetic field (model). A local magnetic field that is permanently disturbed, invalidating the magnetic reference stored in non-volatile memory will lead to a bias in the heading from true north. It is worth noting that the earth's magnetic field does vary both geographically and also temporally.

In order to use the module as an inertial heading system the local earth magnetic field vector must be found and programmed to non-volatile memory. The magnetic reference for your latitude, longitude and current year can be found at:

www.ngdc.noaa.gov/geomagmodels/IGRFWMM.jsp?defaultModel=WMM

Register 21 holds the reference vectors for the Kalman filter. The first three values are the magnetic reference; the last three are the acceleration reference. When setting the reference for the magnetic field the VN-100 will not utilize the magnitude information, so any units can be used for the magnetic reference vector. However, if active filter tuning is being utilized one should execute a "Tare" command just prior to writing the reference vectors. This allows VN-100 to have an accurate magnitude to assign to the incoming reference vector.

11.6 General Considerations for using the VN-100

11.6.1 Magnetic effects

The VN-100 relies on magnetic and acceleration measurements in order to solve for orientation, however, when the quality of the magnetic and acceleration measurements deteriorates so will the accuracy of the solution. This can be mitigated to some level by using active tuning and especially external to the sensor body field effects can be rejected this way (external magnetic and acceleration disturbances). However, when embedding this module in an electronics assembly one invariably will expose the VN-100 to nearby hard (magnetized metallic components) and soft (e.g. nickel batteries) iron magnetic effects. If one does not compensate for this it will adversely affect the performance of the VN-100.

VN-100 has registers that allow for both hard and soft Iron compensation in order to manage this. Setting these registers properly will remove the constant body disturbance of the magnetic and allow for optimal performance of the module.

In the event that the VN-100 is exposed to strong magnetic fields, bias can be induced in the magnetic readings. This can be removed by executing a reset/power cycle of the VN-100 module. During the startup sequence the VN-100 the module will always execute a strong SET/RESET current pulse to the magnetometers; this will null any such externally induced bias and reset the magnetometers to the factory calibrated state.

11.6.2 Acceleration considerations

If one were to use the AHRS in a moving vehicle one should take care to mount the VN-100 module as close to the cg as possible. This will mitigate the effect of centripetal forces on the solution, in the event of angular rates. If active filter tuning is utilized much of this disturbance can be rejected, however, in cases with persistent angular rates there will be a dc acceleration disturbance that cannot be rejected utilizing active filter tuning parameters.

12 Revision history

Table 16 – Revision History

Date	Revision	Changes
Jun-05-2009	1.0.0	Initial release.
Sep-03-2009	1.0.1	Improved section 3.3 SPI description.
Oct-06-2009	1.1.0	New document format. Added 11.2.1 Example tuning parameters.