PRODUCT SPECIFICATION

VN-200 GNSS/INS

Interface Control Document (Firmware v2.0.0.1)

VECTORNAV

CONTENTS

VectorNav Support

Whether you are looking for details on the VN-200 or assistance with your application, a wealth of information is available to assist you in product design and development. Check out the *Inertial Systems Primer* on our website, and be sure to register for access to a wide range of resources:

PRODUCT SPECIFICATIONS	TECHNICAL NOTES	APPLICATION NOTES
▪ User Manual	▪ Time Synchronization	▪ Gimbal Stabilization & Pointing
▪ Interface Control Document	▪ Hard & Soft Iron Calibration	▪ Satellite Communications
▪ Datasheet	▪ External GNSS Aiding	▪ Lidar Mapping
▪ Quick Start Guide	▪ Firmware Update	▪ Aerial Photogrammetry

All VectorNav products are backed by our customer-focused, robust and responsive support ecosystem. Our team is committed to supporting you through your entire product life-cycle, from concept design to in-field support. Please feel free to contact us by phone or email, our global team of engineers and representatives is ready to work with you through every challenge you know of, and every challenge you don't.

+1.512.772.3615 support@vectornav.com

CHANGE LOGS

FIRMWARE

Changelog for Firmware 2.0.0.1 Since 2.0.0.0

- Add GNSS raw data as an available binary output.

DOCUMENT

REV	DATE	AUTHOR	COMMENTS
R1	2/27/2024	P. Long	▪ Updated the InsStatus binary output type. ▪ Updated the description in the GNSS System Configuration register (Register 99). ▪ Corrected the offset for padding bytes in Registers 58, 59, 72, and 73. ▪ Updated the MagPres binary output type.
R0	6/19/2023	B. Lash	▪ Initial release of Industrial Series ICDs.

1 BASIC COMMUNICATION

The VN-200 module supports a universal asynchronous receiver/transmitter (UART) serial interface, and – on SMD variants – a serial peripheral interface (SPI). On the UART interface, the module communicates using ASCII text for its command and data format (although alternate asynchronous data outputs are also supported). The ASCII protocol is very similar to the widely used NMEA 0183 protocol supported by most GPS receivers, and consists of comma delimited parameters printed in human readable text. On the SPI bus, the VN-200-SMD module communicates as a slave device and uses a binary command and data format.

A detailed description of the command format for each interface is given in the following sections, and formatting specific to each command and associated parameters is provided throughout the remainder of the document.

1.1 UNIVERSAL ASYNCHRONOUS RECEIVE/TRANSMIT (UART) SERIAL INTERFACE

1.1.1 Hardware

The VN-200 incorporates two distinct, bidirectional UART ports, UART-1 and UART-2, that can be independently configured. Registers for configuring a UART port, including baud rate (Register 5) and various asynchronous data outputs (see following sections), default to configuring the port over which the setting is being communicated, but include optional parameters to allow the non-connected port to be configured.

1.1.2 Software

On the serial interface, the VN-200 uses ASCII text for its command format. The ASCII protocol is very similar to the widely used NMEA 0183 protocol supported by most GPS receivers, and consists of comma delimited fields printed in human readable text. All commands start with a dollar sign, followed by a five character command, command specific parameters, an asterisk, a checksum, and a newline character. An example command is shown below.

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

All commands are echoed back by the VN-200 after they have been processed. In the case of commands that reset the microcontroller, the commands are echoed back just before the reset occurs. A `VNERR` message is returned instead if a command is invalid for some reason (see Section 1.5).

In addition to the standard commands described in Section 1.3, the UART serial ports provide several options for asynchronous streaming of data. This includes a binary output option and industry standard outputs (eg. NMEA) that do not adhere to the protocol described above. See the following sections for a brief description of these outputs.

1.1.3 ASCII Asynchronous Data Output

Rather than repeatedly querying the desired data register using the Read Register command, the VN-200 allows the user to select a single ASCII output message to stream asynchronously at a fixed rate. The output message can be selected using Register 6, while the rate can be set using Register 7. Alternatively, the rate can be determined by pulses on the SyncIn line as defined in Register 32.

Note that when setting the output message and/or rate, the VN-200 evaluates the bandwidth required for the desired message size and rate as compared to the baud rate selected in Register 5 and will return an error message if the baud rate is insufficient for the desired data rate. This check is made even if the output rate is being externally defined by the SyncIn line.

When a measurement register is selected as an asynchronous output, the header of that message is unique to each measurement register. This is similar to the NMEA-0183 protocol used by GNSS receivers. For instance, Register 8

- Yaw, Pitch & Roll, uses the header `VNYPR`.

1.1.4 Binary Asynchronous Data Output

Binary output messages provide a means for streaming a user-defined subset of sensor measurements from the sensor at fixed rates. All real-time data either measured or estimated by the sensor is available using binary output messages. The binary output message protocol has been designed to minimize the communication overhead and supports output rates up to 800 Hz. The user can select an arbitrary combination of data to form a message that is output at a fixed rate. Multiple configuration registers exist to allow the user to select different data combinations for different output rates or UART ports (see Register 75-Register 77).

1.1.5 Industry Standard Output

In addition to the ASCII and binary messages available via the VectorNav protocol, the VN-200 provides the option to configure industry standard output messages, specifically the NMEA-0183 protocol. Multiple messages in that protocol can be configured to output simultaneously at a fixed rate. Two independent registers for configuring these industry standard output messages (see Register 101 & Register 102) provide the option to configure different messages at different rates or on separate UART ports.

1.2 SERIAL PERIPHERAL INTERFACE (SPI) (SMD ONLY)

The VN-200-SMD supports the Serial Peripheral Interface (SPI) communication interface specification. The SPI specification consists of synchronous serial communication interface where devices communicate in a master/slave mode. The VN-200-SMD operates as a slave while the device communicating with the VN-200-SMD will act as a master. The master provides a clock to the slave which synchronizes the data transfer to the rising and falling edge of the clock signal. Due to its synchronous communication, high data transfer rates, and master/slave operation, the SPI communication interface is ideal for board-level communication over short distances because it does not require a complex software protocol stack and is fairly straightforward to program on embedded devices.

1.2.1 Hardware Requirements

Four hardware lines are required to implement SPI communication with the VN-200-SMD; a clock (SPI_SCK), two data lines (SPI_MOSI and SPI_MISO), and a chip select pin (SPI_CS). The master is responsible for driving both the clock signal and the chip select lines. The chip select line should be pulled low when the master wants to communicate with that particular device. If multiple slave devices are used on the same bus, then each slave will have its own dedicated chip select line, while sharing the clock and data lines. The VN-200-SMD will leave the SPI_MISO line in a high impedance state while the SPI_CS line is high, enabling communication with other slave devices on the same SPI bus. When the master is finished communicating with the slave the chip select line is pulled high. The clock line should idle high when not in use. The SPI_MISO and SPI_MOSI pins should both transition between logic states on the falling edge of the SPI_SCK clock signal. Data on both the SPI_MISO and SPI_MOSI should be sampled on the rising edge of the SPI_SCK line. The VN-200-SMD uses 3 V digital logic for the SPI interface. If you are interfacing with a 5 V system, it is recommended that you use a logic level translation circuit to ensure reliable communication.

SPI Master Settings

PARAMETER	VALUE
Chip Select	Active Low
Clock Polarity	Idle High (CPOL=1)
Clock Phase	Sample second clock edge (CPHA=1)
Data Format	Most significant bit first (MSB)
Byte Order	Least significant byte first (little-endian)

TABLE 1.1

1.2.2 Software Requirements

Communication with the VN-200-SMD over SPI is conducted with multiple transactions. A transaction for the purpose of this document is defined as a single operation, such as reading or writing to a register on the VN-200-SMD or issuing a command such as requesting a device reset. A single transaction consists to two separate data packets sent to the VN-200-SMD. Each packet consists of a four byte header followed by a data payload (and then a checksum if enabled). The header for the packet differs depending upon whether it is a request packet or a response packet. For each packet sent to the VN-200-SMD the chip select line (SPI_CS) should be pulled low at the beginning of the packet and pulled high at the end. See Appendix A.2 for detailed examples of the SPI implementation.

1.3 COMMANDS

This section provides a listing of the commands available on the VN-200. All commands are available on all UART ports, with the exception of the Firmware Update Mode, which is only supported on UART-1. Most commands are also supported over SPI, though not all.

When a valid command is sent to the device, it is echoed back to the user after it has been fully processed. The exception is commands that cause the sensor to reset, in which case the echoed response is sent immediately prior to the reset. No response is provided for resetting commands sent via SPI. In the case of an invalid command, an error message is sent instead of the echo, as described in Section 1.5.

1.3.1 Read Register

ASCII Command . . . . `VNRRG`
SPI Command . . . . `0x01`
Description . . . . Read the contents of any register.

OFFSET	NAME	FORMAT	DESCRIPTION
0	RegId	u8	Selects the register number to read.

- Example UART Command and Response:

`$VNRRG,5*46`

`$VNRRG,5,115200*6D`

- Example SPI Command and Response:

`01 05 00 00`

`00 01 05 00 00 C2 01 00`

1.3.2 Write Register

ASCII Command . . . . `VNWRG`
SPI Command . . . . `0x02`
Description . . . . Write the contents of any register.

OFFSET	NAME	FORMAT	DESCRIPTION
0	RegId	u8	Selects the register number to write.
1	Args	v8	Variable number of arguments depending on the register being written.

- Example UART Command and Response:

`$VNWRG,5,921600*46`

`$VNWRG,5,921600*6D`

- Example SPI Command and Response:

`02 05 00 00 00 10 0E 00`

`00 02 05 00 00 10 0E 00`

1.3.3 Write Settings

ASCII Command . . . . VNWNV
SPI Command 0x03
Description . . . . Instructs the sensor to save the state of all configuration registers to internal non-volatile memory. The current state of the configuration registers will be used at subsequent startups.

Due to limitations in the flash write speed the write settings command typically takes 500 ms to complete, but may take up to 1 second. Any commands that are sent to the sensor during this time will be responded to after the operation is complete.

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

- Example UART Command and Response:

```
$VNWNV*57
```

```
$VNWNV*57
```

- Example SPI Command and Response:

```
03 00 00 00
```

```
00 03 00 00
```

1.3.4 Restore Factory Settings

ASCII Command . . . . VNRFS
SPI Command 0x04
Description . . . . Restores the sensor's non-volatile configuration register settings to their factory-default state. This does not impact any IMU factory calibrations.

This command performs a reset of the sensor. The echoed response to this command is sent just prior to the reset. Further communication with the device should be paused for 2 seconds or until the asynchronous streaming data has been received from the sensor, whichever comes first.

- Example UART Command and Response:

```
$VNRFS*5F
```

```
$VNRFS*5F
```

- Example SPI Command and Response:

```
04 00 00 00
```

```
00 04 00 00
```

1.3.5 Reset

ASCII Command . . . . VNRST
SPI Command 0x06
Description . . . . Performs a software reset on the device. Any changes to the configuration registers that has not been saved to flash using a Write Settings command will be lost.

> ! This command performs a reset of the sensor. The echoed response to this command is sent just prior to the reset. Further communication with the device should be paused for 2 seconds or until the asynchronous streaming data has been received from the sensor, whichever comes first.

- Example UART Command and Response:

```
$VNRST*4D
```

```
$VNRST*4D
```

- Example SPI Command and Response:

```
06 00 00 00
```

```
00 06 00 00
```

1.3.6 Firmware Update

ASCII Command . . . . VNFWU
Description . . . . Instructs the sensor to stop runtime execution and to enter into its internal bootloader to perform a firmware update. If bootloader does not receive valid commands within 10 seconds it will return to runtime mode and the sensor will begin executing as normal.

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

- Example UART Command and Response:

```
$VNFWU*5C
```

```
$VNFWU*5C
```

1.3.7 Known Magnetic Disturbance

ASCII Command . . . . . . `VNKMD`
SPI Command . . . . . . `0x08`
Description . . . . . . Instructs the AHRS filter (if active) to temporarily stop trusting its internal magnetometer to minimize attitude errors during times when the magnetic environment is known to be poor.

OFFSET	NAME	FORMAT	DESCRIPTION
0	State	u8	Sets the state of the magnetic disturbance logic

> Setting the known magnetic disturbance to Present for more than 60 seconds can lead to filter instability and/or total divergence and should be avoided.

This command notifies the sensor that a magnetic disturbance is present, resulting in the AHRS filter tuning out the magnetometer measurements and pausing the real-time hard/soft iron calibration if it is enabled. Particularly useful for disturbances lasting between 0.5 s and 10 s.

- Example UART Command and Response:

```
$VNKMD,1*47
```

```
$VNKMD,1*47
```

- Example SPI Command and Response:

```
08 01 00 00
```

```
00 08 01 00
```

Enumeration: State

NAME	VALUE	DESCRIPTION
NotPresent	0	No magnetic disturbance is present. Use the magnetometer for heading estimation.
Present	1	Magnetic disturbance is present. Do not use the magnetometer for heading estimation.

TABLE 1.2

1.3.8 Known Acceleration Disturbance

ASCII Command .... `VNKAD`
SPI Command .... `0x09`
Description .... Instructs the AHRS filter (if active) to temporarily stop trusting its internal accelerometer to minimize attitude errors during times when accelerations disturbances are present.

OFFSET	NAME	FORMAT	DESCRIPTION
0	State	u8	Sets the state of the acceleration disturbance logic

> Setting the known acceleration disturbance to Present for more than 60 seconds can lead to filter instability and/or total divergence and should be avoided.

This command notifies the sensor that an acceleration disturbance is present, resulting in the AHRS filter tuning out the accelerometer gravity measurements. Particularly useful for disturbances of 1 g or more that last between 0.5 s and 10 s.

- Example UART Command and Response:

```
$VNKAD,1*4B
```

```
$VNKAD,1*4B
```

- Example SPI Command and Response:

```
09 01 00 00
```

```
00 09 01 00
```

Enumeration: State

NAME	VALUE	DESCRIPTION
NotPresent	0	No acceleration disturbance is present. Use the accelerometer for attitude estimation.
Present	1	Acceleration disturbance is present. Do not use the accelerometer for attitude estimation.

TABLE 1.3

1.3.9 Set Initial Heading

ASCII Command .... `VNSIH`
SPI Command .... `0x14`
Description .... Sets the initial heading of the device to the given heading value. Command accepts a single argument (heading) OR three arguments (yaw-pitch-roll) OR four arguments (quaternion).

OFFSET	NAME	FORMAT	DESCRIPTION
0	Param1	float	Heading angle (deg) OR Yaw (deg) OR First vector component of quaternion.
1	Param2	float	Optional: Pitch (deg) OR Second vector component of quaternion.
2	Param3	float	Optional: Roll (deg) OR Third vector component of quaternion.
3	Param4	float	Optional: Scalar component of quaternion.

This command sets the heading estimate to the angle provided by the user, immediately initializing the INS filter and expediting the startup process. This command has no impact on initial pitch or roll (tilt). Using yaw-pitch-roll or quaternion is recommended in cases where pitch angle is outside of the range $\pm 80°$. This command has no impact on initial pitch or roll (tilt). Not recommended once the INS Mode is reporting as Tracking.

> It is important that the initial heading you provide to the sensor is accurate to within 5 degrees of the true heading of the sensor relative to true north. If the initial heading provided is not within this accuracy window, then the INS may lose tracking and possibly reset.

The Set Initial Heading Command is a variable length command, allowing the user to input a heading value alone, or a yaw-pitch-roll sequence, or a full four-parameter quaternion. One of the latter two methods should be used when pitch or roll exceeds 45 degrees. When using the ASCII form of the command, the sensor automatically detects the number of arguments and interprets it accordingly.

- Example UART Command and Response (Heading Only):

```
$VNSIH,73.2*7E
```

```
$VNSIH,73.2*7E
```

- Example UART Command and Response (Yaw-Pitch-Roll):

```
$VNSIH,73.2,87.1,10.2*73
```

```
$VNSIH,73.2,87.1,10.2*73
```

- Example UART Command and Response (Quaternion):

```
$VNSIH,-0.3574,0.5894,0.3812,0.6161*6A
```

```
$VNSIH,-0.3574,0.5894,0.3812,0.6161*6A
```

When using the SPI form, however, users must specify the mode of the command in the command argument (byte immediately following the command ID).

- Example SPI Command and Response (Heading Only):

```
14 00 00 00 66 66 92 42
```

```
00 14 00 00
```

- Example SPI Command and Response (Yaw-Pitch-Roll):

```
14 01 00 00 66 66 92 42 33 33 AE 42 33 33 23 41
```

```
00 14 01 00
```

- Example SPI Command and Response (Quaternion):

```
14 02 00 00 22 FD B6 BE EB E2 16 3F A5 2C C3 3E BB B8 1D 3F
```

```
00 14 02 00
```

1.3.10 Async Output Enable

ASCII Command .... `VNASY`
Description .... Toggles all asynchronous communication on ASCII, binary, and NMEA output protocols.

OFFSET	NAME	FORMAT	DESCRIPTION
0	OutputState	u8	State of output measurements on the device.

This command allows the user to temporarily pause the asynchronous outputs on the given serial port. When paused, all ASCII asynchronous outputs, NMEA messages, and binary output messages will temporarily stop streaming from the device on the serial port for which this command is received. The configuration registers for all asynchronous outputs remain unchanged and streaming will resume after a reset or power cycle.

This command is useful when you want to send configuration commands to the sensor, but do not want to deal with the additional overhead of having to parse a constant stream of asynchronous output messages while waiting for the response to your configuration commands. It is also useful when you want to manually type commands to the device from a serial terminal.

- Example UART Command and Response:

```
$VNASY,1*4E
```

```
$VNASY,1*4E
```

Enumeration: OutputState

NAME	VALUE	DESCRIPTION
Disable	0	All output messages are temporarily disabled.
Enable	1	All output messages are enabled.

TABLE 1.4

1.3.11 Set Filter Bias

ASCII Command .... `VNSFB`
SPI Command .... `0x11`
Description .... Sets the Filter Startup Bias Register to the current filter bias estimates.

This command copies the current filter bias estimates for the accelerometer, gyro, and pressure sensor (if enabled) into the Filter Startup Bias Register (Register 74). After sending this command and before resetting or power cycling, issue the Write Settings Command to save the state of this register to non-volatile (flash) memory. Once saved, the sensor uses the bias estimates in Register 74 as the initial state at startup.

- Example UART Command and Response:

```
$VNSFB*4F
```

```
$VNSFB*4F
```

- Example SPI Command and Response:

```
11 00 00 00
```

```
00 11 00 00
```

1.3.12 Poll Binary Output Message

ASCII Command . . . . `VNBOM`
SPI Command . . . . `0x12`
Description . . . . Triggers an output of the currently-configured binary output message.

OFFSET	NAME	FORMAT	DESCRIPTION
0	BinMsgNum	u8	Selects the binary output register to poll.

When sent over the UART, this command does not get echoed back. Instead, the response is the binary output message specified.

- Example UART Command:

```
$VNBOM,1*45
```

```
FA 01 29 01 ...
```

- Example SPI Command:

```
12 01 00 00
```

```
00 12 01 00 FA 01 29 01 ...
```

Enumeration: BinMsgNum

NAME	VALUE	DESCRIPTION
BinMsg1	1	Poll binary output message defined in Register 75.
BinMsg2	2	Poll binary output message defined in Register 76.
BinMsg3	3	Poll binary output message defined in Register 77.

TABLE 1.5

1.4 CHECKSUM

All communication interfaces provide the option for utilizing an 8-bit XOR checksum, a 16-bit CRC checksum, or no checksum, as configured in Register 30. By default, the UART serial interface utilizes an 8-bit checksum, whereas the SPI bus utilizes no checksum. The 16-bit CRC checksum is recommended when high reliability is required.

For ASCII messages over the UART interface, the checksum is calculated over all bytes between, but not including, the dollar sign ($) and asterisk (*). All delimiters (commas) are included in the checksum calculation. For messages over the SPI bus, all bytes in the message, including the header but excluding the checksum itself, are included in the checksum calculation.

The binary output messages sent over the UART serial interface utilize a 16-bit checksum regardless of the configuration in Register 30. See Section 1.1.4 for details on which bytes are included when calculating the checksum of binary output messages.

1.4.1 Checksum Bypass

When communicating with the sensor over the UART interface, the checksum can be bypassed by replacing the hexadecimal digits in the checksum with uppercase X characters. This works for both the XOR and CRC checksum and allows for rapid development and testing through a serial terminal without requiring the checksum to be disabled. An example command to read Register 1 over the UART interface using the checksum bypass feature is shown below.

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

1.4.2 8-bit XOR Checksum

The 8-bit XOR checksum is a sequential XOR of each byte. The resultant checksum is an 8-bit number and is represented in the UART ASCII commands as two hexadecimal characters. The C function snippet below calculates the correct checksum.

8-bit XOR Checksum Calculation C-code Snippet

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

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

    return cksum;
}
```

1.4.3 16-bit CRC Checksum

For cases where the 8-bit XOR checksum does not provide sufficient error detection, a full 16-bit CRC checksum is available. The VN-200 uses the CRC16-CCITT algorithm. The resultant CRC checksum is a 16-bit number and is represented in the UART ASCII commands as four hexadecimal characters. The C function snippet below calculates the correct CRC.

16-bit CRC Checksum Calculation C-code Snippet

```c
// Calculates the 16-bit CRC checksum for the given byte sequence.
unsigned short calculateCRC(unsigned char data[], unsigned int length)
{
  unsigned int i;
  unsigned short crc = 0;

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

  return crc;
}
```

1.5 SYSTEM ERROR CODES

In the event of an error, the VN-200 will output `$VNERR`, followed by an error code and the appropriate checksum. The possible error codes are listed in Table 1.6 with a description of the error.

Error Codes

ERROR	CODE	DESCRIPTION
Hard Fault	01	If this error occurs, then the firmware on the VN-200 has experienced a hard fault exception. Any asynchronous streaming is disabled, and requests to read measurement registers return Invalid Register errors. Sensor must be reset or power-cycle to resume operation (if possible).
Serial Buffer Overflow	02	The processor's serial input buffer has experienced an overflow. The processor has a 256 character input buffer.
Invalid Checksum	03	The checksum for the received command was invalid.
Invalid Command	04	The user has requested an invalid command.
Not Enough Parameters	05	The user did not supply the minimum number of required parameters for the requested command.
Too Many Parameters	06	The user supplied too many parameters for the requested command.
Invalid Parameter	07	The user supplied a parameter for the requested command which was invalid.
Invalid Register	08	An invalid register was specified.
Unauthorized Access	09	The user does not have permission to write to this register.
Watchdog Reset	0A	A watchdog reset has occurred. In the event of a non-recoverable error the internal watchdog will reset the processor within 50 ms of the error.
Output Buffer Overflow	0B	The output buffer has experienced an overflow. The processor has a 2048 character output buffer.
Insufficient Baud Rate	0C	The baud rate is not high enough to support the requested asynchronous data output at the requested data rate.
Error Buffer Overflow	FF	An overflow event has occurred on the system error buffer.

TABLE 1.6

2 BINARY OUTPUT MESSAGES

2.1 OVERVIEW

2.1.1 Available Outputs

Particular pieces of data are organized by a three-level hierarchy to enable easy selection:

1. **Binary Field** Binary fields represent individual measurements or estimates, similar to the ASCII message fields. For instance, the accelerometer measurement along the body x-axis is a distinct Binary Field denoted *AccelX*.
2. **Binary Type** Binary types combine one or more Binary Fields into a user-selectable output. For instance, the three body-frame accelerometer measurement fields (*AccelX*, *AccelY*, *AccelZ*) combined form the *Accel* Binary Type. Users may select any combination of Binary Types for output in a particular binary output message.
3. **Binary Group** Multiple Binary Types are grouped together with associated measurements into a *Binary Group*. For instance, the accelerometer, gyro, magnetometer and other direct sensor measurements are part of the IMU Group. Users specify desired outputs by referencing both the Binary Group and the Binary Type within that group of the desired data.

Binary Types by Group

COMMON GROUP	TIME GROUP	IMU GROUP	GNSS GROUP	ATTITUDE GROUP
TimeStartup	TimeStartup	UncompMag	TimeUtc	Ypr
TimeGps	TimeGps	UncompAccel	GpsTow	Quaternion
TimeSyncIn	GpsTow	UncompGyro	GpsWeek	Dcm
Ypr	GpsWeek	Temperature	NumSats	MagNed
Quaternion	TimeSyncIn	Pressure	GnssFix	AccelNed
AngularRate	TimeGpsPps	DeltaTheta	GnssPosLla	LinBodyAcc
PosLla	TimeUtc	DeltaVel	GnssPosEcef	LinAccelNed
VelNed	SyncInCnt	Mag	GnssVelNed	YprU
Accel	SyncOutCnt	Accel	GnssVelEcef	
Imu	TimeStatus	AngularRate	GnssPos Uncertainty	
MagPres		SensSat	GnssVel Uncertainty	
Deltas			GnssTime Uncertainty	
InsStatus			GnssTimeInfo	
SyncInCnt			GnssDop	
TimeGpsPps			GnssSatInfo	
			GnssRawMeas	

INS GROUP

- InsStatus
- PosLla
- PosEcef
- VelBody
- VelNed
- VelEcef
- MagEcef
- AccelEcef
- LinAccelEcef
- PosU
- VelU

2.1.2 Output Configuration

To configure a particular output message, a group byte bitfield is first specified to define which binary output groups are being referenced in the output message. Then 16-bit type words for each of the enabled groups are defined to indicate which output types within that group will be output. For more detail, refer to Register 75 – Register 77, which allows configuring up to three different messages, each with customizable content and output rates. See Appendix A.1 for examples of how to configure the desired binary output.

2.1.3 Message Format

A binary output message consists of four (4) components:

1. **Sync Byte** The sync byte is the first byte of a binary output message and can be used to align to the output messages and has a value of `0xFA`.
2. **Header** The header starts with a binary group byte denoting the Binary Groups included in the message. The group byte is followed by a two-byte type word for each active group. See Table 2.1 for the group byte bit offsets for each Binary Group. The remaining sections of this chapter detail the bit offsets and other details for each Binary Type in each group.
3. **Payload** The payload consists of the output data selected based upon the bits selected in the group byte and type word. All output data in the payload section is ordered sequentially based first on the Binary Group bit offset, then by the Binary Type bit offset. As such, all Binary Types selected for a the lowest numbered Binary Group are output, followed by the Binary Types for the next Binary Group, etc. All data is directly concatenated, with no padding bytes between Binary Types.
4. **Checksum** The checksum on all binary output messages is a 16-bit CRC checksum as defined in Section 1.4.3. The checksum is calculated over the header and the payload. As described in Section 1.4.3, the CRC checksum is output in big-endian format, such that if on receipt of a binary output message, the checksum itself is included in the checksum computation, a valid packet will produce a CRC checksum value of `0x0000`. This provides a simple way of detecting packet corruption by simply checking to see if the CRC calculation of the entire message (not including the sync byte) results in zero.

Figure 2.1 shows the format of a generic binary output message that contains data from N different groups with a total payload length of M bytes. In most cases, total payload length M can be readily computed by summing the fixed length of each configured binary output type. A table of binary output type sizes is provided in Table 2.2.

Note that some Binary Types may be defined as variable length. When a type has a variable length, it contains a count or length variable that must be parsed to determine the length of that output type and the output message as a whole. See Appendix A.1 for examples of the binary output message format.

Bit Field: Binary Groups

NAME	OFFSET	DESCRIPTION
Common	0	Binary group 1 contains a wide assortment of commonly used data required for most applications.
Time	1	The time group provides all timing and event counter related outputs.
Imu	2	The IMU group provides all outputs which are dependent upon the measurements collected from the IMU.
Gnss	3	The GNSS group provides all outputs which are dependent upon the measurements collected from the primary GNSS receiver.
Attitude	4	The attitude group provides all estimated outputs which are dependent upon the estimated attitude solution.
Ins	5	The INS group provides all estimated outputs which are dependent upon the onboard INS state solution.

TABLE 2.1

Binary Output Message Format

	Sync	Header				
		Group Bytes	Type Bytes Group A	Type Bytes Group B	...	Type Bytes Group X
Byte Offset	0	1	2 3	4 5	...	2N 2N+1
Type	uint8	uint8	uint16	uint16	...	uint16

	Payload				Checksum
	Outputs Group A	Outputs Group B	...	Outputs Group X	
Byte Offset	2N+2 ...	...	...	... 2N+M+1	2N+M+2 2N+M+3
Type	Variable	Variable	...	Variable	uint16

FIGURE 2.1

Binary Output Type Sizes

OFFSET	COMMON	TIME	IMU	GNSS	ATTITUDE	INS
0	8	8	–	8	–	2
1	8	8	12	8	12	24
2	8	8	12	2	16	24
3	12	2	12	1	36	12
4	16	8	4	1	12	12
5	12	8	4	24	12	12
6	24	8	16	24	12	12
7	12	4	12	12	12	12
8	12	4	12	12	12	12
9	24	1	12	12	–	4
10	20	–	12	4	–	4
11	28	–	2	4	–	–
12	2	–	–	2	–	–
13	4	–	–	28	–	–
14	8	–	–	10	–	–
16	–	–	–	40	–	–

TABLE 2.2

2.2 COMMON GROUP

The Common Group combines a subset of Binary Types from the other Binary Groups so that most users can define their desired output message using only the Common Group. See the individual Binary Group definitions for details on the contents of each Binary Type. The output types available in the Common Group are listed in Table 2.3.

Bit Field: Common OutputFields

NAME	OFFSET	DESCRIPTION
TimeStartup	0	Copy of Time Group TimeStartup.
TimeGps	1	Copy of Time Group TimeGps.
TimeSyncIn	2	Copy of Time Group TimeSyncIn.
Ypr	3	Copy of Attitude Group Ypr.
Quaternion	4	Copy of Attitude Group Quaternion.
AngularRate	5	Copy of IMU Group AngularRate.
PosLla	6	Copy of INS Group PosLla.
VelNed	7	Copy of INS Group VelNed.
Accel	8	Copy of IMU Group Accel.
Imu	9	Concatenation of IMU Group UncompAccel & UncompGyro.
MagPres	10	Concatenation of IMU Group Mag, Temperature & Pressure.
Deltas	11	Concatenation of IMU Group DeltaTheta & DeltaVel.
InsStatus	12	Copy of INS Group InsStatus.
SyncInCnt	13	Copy of Time Group SyncInCnt.
TimeGpsPps	14	Copy of Time Group TimeGpsPps.

TABLE 2.3

2.3 TIME GROUP

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

Bit Field: Time Group Output Types

NAME	OFFSET	DESCRIPTION
TimeStartup	0	The system time since startup measured in nano seconds.
TimeGps	1	The absolute GPS time since start of GPS epoch 1980 expressed in nano seconds.
GpsTow	2	The GPS time of week given in nano seconds.
GpsWeek	3	The current GPS week.
TimeSyncIn	4	The time since the last SyncIn event trigger expressed in nano seconds.
TimeGpsPps	5	The time since the last GPS PPS trigger event expressed in nano seconds.
TimeUtc	6	The current UTC time.
SyncInCnt	7	The number of SyncIn trigger events that have occurred.
SyncOutCnt	8	The number of SyncOut trigger events that have occurred.
TimeStatus	9	Time valid status flags.

TABLE 2.4

2.3.1 TimeStartup

Output Size (Bytes) 8
Output Bit Offset 0
Description . . . . The system time since startup measured in nano seconds. The time since startup is based upon the internal TXCO oscillator for the MCU. The accuracy of the internal TXCO is ±20 ppm (−40 °C to 85 °C).

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	TimeStartup	uint64	ns	Time since start-up, based on internal TCXO of the microcontroller (20 ppm accuracy).

2.3.2 TimeGps

Output Size (Bytes) 8
Output Bit Offset 1
Description . . . . The absolute GPS time since start of GPS epoch 1980 expressed in nano seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	TimeGps	uint64	ns	Absolute Gps time (since 1980).

2.3.3 GpsTow

Output Size (Bytes) 8
Output Bit Offset 2
Description . . . . The time since the start of the current GPS time week expressed in nano seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	uint64	ns	GPS time of week.

2.3.4 GpsWeek

Output Size (Bytes) 2
Output Bit Offset 3
Description . . . . The current GPS week.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsWeek	uint16	–	The current GPS week.

2.3.5 TimeSyncIn

Output Size (Bytes) 8
Output Bit Offset 4
Description . . . . The time since the last SyncIn event trigger expressed in nano seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	TimeSyncIn	uint64	ns	Time since last SyncIn trigger.

2.3.6 TimeGpsPps

Output Size (Bytes) 8
Output Bit Offset 5
Description . . . . The time since the last GPS PPS trigger event expressed in nano seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	TimeGpsPps	uint64	ns	Time since last Gps PPS trigger.

2.3.7 TimeUtc

Output Size (Bytes) 8
Output Bit Offset 6
Description . . . . The current UTC time. The year is given as a signed byte year offset from the year 2000. For example the year 2013 would be given as year 13.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Year	int8	yr	Number of years before or after 2000.
1	Month	uint8	mo	Month of the year [1 to 12].
2	Day	uint8	d	Day of the month [1 to 31].
3	Hour	uint8	hr	Hour of the day [0 to 23].
4	Minute	uint8	min	Minute of the hour [0 to 59].
5	Second	uint8	s	Second of the minute [0 to 59].
6	FracSec	uint16	ms	Fraction of a second [0 to 999].

2.3.8 SyncInCnt

Output Size (Bytes) 4
Output Bit Offset 7
Description . . . . The number of SyncIn trigger events that have occurred.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	SyncInCnt	uint32	–	Number of SyncIn trigger events.

2.3.9 SyncOutCnt

Output Size (Bytes) 4
Output Bit Offset 8
Description . . . . The number of SyncOut trigger events that have occurred.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	SyncOutCnt	uint32	–	Number of SyncOut trigger events.

2.3.10 TimeStatus

Output Size (Bytes) 1
Output Bit Offset 9
Description . . . . Time valid status flags.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	TimeStatus	uint8	BITS	Time validity status.

Bit Field: TimeStatus

NAME	OFFSET	DESCRIPTION
TowValid	0	Time of week valid
DateValid	1	Date valid
UtcValid	2	UTC time valid

TABLE 2.15

2.4 IMU GROUP

The IMU group provides all outputs which are dependent upon the measurements collected from the IMU.

Bit Field: Imu Group Output Types

NAME	OFFSET	DESCRIPTION
UncompMag	1	The IMU magnetic field given in the body-frame.
UncompAccel	2	The IMU acceleration given in the body-frame.
UncompGyro	3	The IMU angular rate given in the body-frame.
Temperature	4	The IMU temperature.
Pressure	5	The IMU pressure.
DeltaTheta	6	The delta theta (dtheta) is the delta rotation angles incurred due to rotation, since the last time the values were output by the device.
DeltaVel	7	The delta velocity (dvel) is the delta velocity incurred due to motion, since the last time the values were output by the device.
Mag	8	The IMU compensated magnetic field given in the body-frame.
Accel	9	The compensated acceleration measured in the body-frame.
AngularRate	10	The compensated angular rate measured in the body-frame.
SensSat	11	This field provides flags identifying whether any of the measurements are currently saturated.

TABLE 2.16

2.4.1 UncompMag

Output Size (Bytes) 12
Output Bit Offset 1
Description .... The IMU magnetic field given in the body-frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), and the user compensation, however it is not compensated by the onboard Hard/Soft Iron estimator.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	UncompMag X	float	G	Magnetometer body-frame x-axis.
4	UncompMag Y	float	G	Magnetometer body-frame y-axis.
8	UncompMag Z	float	G	Magnetometer body-frame z-axis.

2.4.2 UncompAccel

Output Size (Bytes) 12
Output Bit Offset 2
Description . . . . . The IMU acceleration given in the body-frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), however it is not compensated by any dynamic calibration such as bias compensation from the onboard INS Kalman filter.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	UncompAcc X	float	m/s^2	Accelerometer body-frame x-axis.
4	UncompAcc Y	float	m/s^2	Accelerometer body-frame y-axis.
8	UncompAcc Z	float	m/s^2	Accelerometer body-frame z-axis.

2.4.3 UncompGyro

Output Size (Bytes) 12
Output Bit Offset 3
Description . . . . . The IMU angular rate given in the body-frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), however it is not compensated by any dynamic calibration such as the bias compensation from the onboard AHRS/INS Kalman filters.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	UncompGyro X	float	rad/s	Angular rate body-frame x-axis.
4	UncompGyro Y	float	rad/s	Angular rate body-frame y-axis.
8	UncompGyro Z	float	rad/s	Angular rate body-frame z-axis.

2.4.4 Temperature

Output Size (Bytes) 4
Output Bit Offset 4
Description . . . . . The IMU temperature.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Temperature	float	°C	Sensor temperature.

2.4.5 Pressure

Output Size (Bytes) 4
Output Bit Offset 5
Description .... The IMU pressure. This is an absolute pressure measurement. Typical pressure at sea level would be around 100 kPa.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Pressure	float	kPa	Barometric pressure.

2.4.6 DeltaTheta

Output Size (Bytes) 16
Output Bit Offset 6
Description .... The delta theta (dtheta) is the delta rotation angles incurred due to rotation, since the last time the values were output by the device. The delta angles are calculated based upon the onboard conning and sculling integration performed onboard the sensor at the IMU sampling rate (nominally 800 Hz). The delta time (dtime) is the time interval that the delta angle and velocities are integrated over. The integration for the delta angles are reset each time the values are either polled or sent out due to a scheduled asynchronous ASCII or binary output. Time is given in sections. Delta angles are given in degrees.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	DeltaTime	float	s	Duration of integration interval.
4	DeltaThetaX	float	deg	Integrated rotation vector x-axis.
8	DeltaThetaY	float	deg	Integrated rotation vector y-axis.
12	DeltaThetaZ	float	deg	Integrated rotation vector z-axis.

2.4.7 DeltaVel

Output Size (Bytes) 12
Output Bit Offset 7
Description .... The delta velocity (dvel) is the delta velocity incurred due to motion, since the last time the values were output by the device. The delta velocities are calculated based upon the onboard conning and sculling integration performed onboard the sensor at the IMU sampling rate (nominally 800 Hz). The integration for the delta velocities are reset each time the values are either polled or sent out due to a scheduled asynchronous ASCII or binary output. Delta velocity is given in meters per second.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	DeltaVelX	float	m/s	Integrated velocity x-axis.
4	DeltaVelY	float	m/s	Integrated velocity y-axis.
8	DeltaVelZ	float	m/s	Integrated velocity z-axis.

2.4.8 Mag

Output Size (Bytes) 12
Output Bit Offset 8
Description . . . . The IMU compensated magnetic field given in the body-frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), the user compensation, and the dynamic calibration from the onboard Hard/Soft Iron estimator.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	MagX	float	G	Compensated magnetometer measurement in the body-frame x-axis.
4	MagY	float	G	Compensated magnetometer measurement in the body-frame y-axis.
8	MagZ	float	G	Compensated magnetometer measurement in the body-frame z-axis.

2.4.9 Accel

Output Size (Bytes) 12
Output Bit Offset 9
Description . . . . The compensated acceleration measured in the body-frame. This measurement is compensated by the static calibration (individual factory calibration stored in flash), the user compensation, and the dynamic bias compensation from the onboard INS Kalman filter.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
4	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
8	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.

2.4.10 AngularRate

Output Size (Bytes) 12
Output Bit Offset 10
Description . . . . The compensated angular rate measured in the body-frame. This measurement is compensated by the static calibration (individual factor calibration stored in flash), the user compensation, and the dynamic bias compensation from the onboard INS Kalman filter.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
4	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
8	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

2.4.11 SensSat

Output Size (Bytes) 2
Output Bit Offset 11
Description . . . . This field provides flags identifying whether any of the measurements are currently saturated.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	SensSat	uint16	BITS	Sensor saturation flags.

Bit Field: SensSat

NAME	OFFSET	DESCRIPTION
MagX	0	Saturation detected on magnetometer x-axis.
MagY	1	Saturation detected on magnetometer y-axis.
MagZ	2	Saturation detected on magnetometer z-axis.
AccelX	3	Saturation detected on accelerometer x-axis.
AccelY	4	Saturation detected on accelerometer y-axis.
AccelZ	5	Saturation detected on accelerometer z-axis.
GyroX	6	Saturation detected on gyro x-axis.
GyroY	7	Saturation detected on gyro y-axis.
GyroZ	8	Saturation detected on gyro z-axis.
Pres	9	Saturation detected on pressure sensor.

TABLE 2.28

2.5 GNSS GROUP

The GNSS group provides all outputs which are dependent upon the measurements collected from the primary GNSS receiver. All data in this group is updated at the rate of the GPS receiver (nominally 5 Hz for the internal GNSS).

Bit Field: Gnss Group Output Types

NAME	OFFSET	DESCRIPTION
TimeUtc	0	The current UTC time.
GpsTow	1	The GPS time of week given in nano seconds.
GpsWeek	2	The current GPS week.
NumSats	3	The number of tracked GNSS satellites.
GnssFix	4	The current GNSS fix.
GnssPosLla	5	The current GNSS position measurement given as the geodetic latitude, longitude and altitude above the ellipsoid.
GnssPosEcef	6	The current GNSS position given in the Earth centered Earth fixed (ECEF) reference frame.
GnssVelNed	7	The current GNSS velocity in the North East Down (NED) reference frame.
GnssVelEcef	8	The current GNSS velocity in the Earth centered Earth fixed (ECEF) reference frame.
GnssPosUncertainty	9	The current GNSS position uncertainty in the North East Down (NED) reference frame.
GnssVelUncertainty	10	The current GNSS velocity uncertainty.
GnssTimeUncertainty	11	The current GPS time uncertainty.
GnssTimeInfo	12	Flags for valid GPS TOW, week number and UTC and current leap seconds.
GnssDop	13	Dilution of precision.
GnssSatInfo	14	Information and measurements pertaining to each GNSS satellite in view.
GnssRawMeas	16	Raw measurements pertaining to each GNSS satellite in view.

TABLE 2.29

2.5.1 TimeUtc

Output Size (Bytes) 8
Output Bit Offset 0
Description . . . . . The current UTC time. The year is given as a signed byte year offset from the year 2000. For example the year 2013 would be given as year 13.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Year	int8	yr	Number of years before or after 2000.
1	Month	uint8	mo	Month of the year [1 to 12].
2	Day	uint8	d	Day of the month [1 to 31].
3	Hour	uint8	hr	Hour of the day [0 to 23].
4	Minute	uint8	min	Minute of the hour [0 to 59].
5	Second	uint8	s	Second of the minute [0 to 59].
6	FracSec	int16	ms	Fraction of a second. Typically positive, but may be negative.

2.5.2 GpsTow

Output Size (Bytes) 8
Output Bit Offset 1
Description . . . . . The GPS time of week given in nano seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	uint64	ns	GPS time of week.

2.5.3 GpsWeek

Output Size (Bytes) 2
Output Bit Offset 2
Description . . . . . The current GPS week.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsWeek	uint16	–	GPS week.

2.5.4 NumSats

Output Size (Bytes) 1
Output Bit Offset 3
Description . . . . . The number of tracked GNSS satellites.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	NumSats	uint8	–	Number of satellites tracked by GNSS receiver.

2.5.5 GnssFix

Output Size (Bytes) 1
Output Bit Offset 4
Description .... The current GNSS fix.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GnssFix	uint8	ENUM	Type of GNSS fix.

Enumeration: GnssFix

NAME	VALUE	DESCRIPTION
NoFix	0	Not tracking.
TimeFix	1	Time-only fix.
Fix2D	2	Time plus 2D position fix.
Fix3D	3	Time plus full 3D position fix.
SBAS	4	3D Fix using SBAS corrections.
RtkFloat	7	3D Fix using an approximate RTK solution.
RtkFix	8	3D Fix using a fixed RTK solution.

TABLE 2.35

2.5.6 GnssPosLla

Output Size (Bytes) 24
Output Bit Offset 5
Description .... The current GNSS position measurement given as the geodetic latitude, longitude and altitude above the ellipsoid.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Lat	double	deg	GNSS geodetic latitude.
8	Lon	double	deg	GNSS longitude.
16	Alt	double	m	GNSS altitude above WGS84 ellipsoid.

2.5.7 GnssPosEcef

Output Size (Bytes) 24
Output Bit Offset 6
Description .... The current GNSS position given in the Earth centered Earth fixed (ECEF) reference frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	PosX	double	m	GNSS position in ECEF-frame x-axis.
8	PosY	double	m	GNSS position in ECEF-frame y-axis.
16	PosZ	double	m	GNSS position in ECEF-frame z-axis.

2.5.8 GnssVelNed

Output Size (Bytes) 12
Output Bit Offset 7
Description .... The current GNSS velocity in the North East Down (NED) reference frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	VelN	float	m/s	GNSS velocity in North direction.
4	VelE	float	m/s	GNSS velocity in East direction.
8	VelD	float	m/s	GNSS velocity in Down direction.

2.5.9 GnssVelEcef

Output Size (Bytes) 12
Output Bit Offset 8
Description .... The current GNSS velocity in the Earth centered Earth fixed (ECEF) reference frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	VelX	float	m/s	GNSS velocity in ECEF-frame x-axis.
4	VelY	float	m/s	GNSS velocity in ECEF-frame y-axis.
8	VelZ	float	m/s	GNSS velocity in ECEF-frame z-axis.

2.5.10 GnssPosUncertainty

Output Size (Bytes) 12
Output Bit Offset 9
Description .... The current GNSS position uncertainty in the North East Down (NED) reference frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Pos UncertaintyN	float	m	GNSS position uncertainty, North direction.
4	Pos UncertaintyE	float	m	GNSS position uncertainty, East direction.
8	Pos UncertaintyD	float	m	GNSS position uncertainty, Down direction.

2.5.11 GnssVelUncertainty

Output Size (Bytes) 4
Output Bit Offset 10
Description . . . . . The current GNSS velocity uncertainty.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GnssVel Uncertainty	float	m/s	GNSS velocity uncertainty (scalar).

2.5.12 GnssTimeUncertainty

Output Size (Bytes) 4
Output Bit Offset 11
Description . . . . . The current GPS time uncertainty.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GnssTime Uncertainty	float	s	GNSS time uncertainty.

2.5.13 GnssTimeInfo

Output Size (Bytes) 2
Output Bit Offset 12
Description . . . . . Flags for valid GPS TOW, week number and UTC and current leap seconds.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GnssTime Status	uint8	–	GNSS time validity status. Same as TimeStatus in Time Group
1	Leap Seconds	int8	s	Current number of leap seconds.

2.5.14 GnssDop

Output Size (Bytes) 28
Output Bit Offset 13
Description . . . . . Dilution of precision.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Gdop	float	–	Dilution of precision.
4	Pdop	float	–	Position dilution of precision.
8	Tdop	float	–	Time dilution of precision.
12	Vdop	float	–	Vertical dilution of precision.
16	Hdop	float	–	Horizontal dilution of precision.
20	Ndop	float	–	North dilution of precision.
24	Edop	float	–	East dilution of precision.

2.5.15 GnssSatInfo

Output Size (Bytes) 10
Output Bit Offset 14
Description . . . . . . Information and measurements pertaining to each GNSS satellite in view.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Count	uint8	–	Number of satellites.
1	Resv	uint8	–	Reserved.
2+8k	Sys	uint8	ENUM	GNSS constellation indicator.
3+8k	SvId	uint8	–	Space vehicle Id.
4+8k	Flags	uint8	BITS	Satellite status flags.
5+8k	Cno	uint8	dB Hz	Signal strength (CN0)
6+8k	Qi	uint8	ENUM	Quality indicator.
7+8k	El	int8	deg	Elevation angle
8+8k	Az	int16	deg	Azimuth angle

Variable Length Output The size of this measurement output depends on the number of satellites or signals being tracked by the GNSS receiver, and may change every time the measurement is output.

The first 2 bytes of this measurement should be treated as a measurement header that applies to the rest of the measurement output. The remaining fields are output for each satellite tracked by the GNSS receiver, repeating over k=0 to Count-1.

Enumeration: Sys

NAME	VALUE	DESCRIPTION
GPS	0	Global Positioning System (USA)
SBAS	1	Satellite Based Augmentation System
Galileo	2	Galileo (European Union)
BeiDou	3	BeiDou Navigation Satellite System (China)
IMES	4	Indoor Messaging System (QZSS extension)
QZSS	5	Quasi-Zenith Satellite System (Japan)
GLONASS	6	Globalnaya Navigatsionnaya Sputnikovaya Sistema (Russia)
IRNSS	7	Indian Regional Navigation Satellite System
LBand	8	L-Band correction satellite
Unknown	255	Unknown satellite system

TABLE 2.46

Bit Field: Flags

NAME	OFFSET	DESCRIPTION
Healthy	0	Satellite is healthy.
AlmanacValid	1	Receiver has a valid almanac for this satellite.
EphemerisValid	2	Receiver has a valid ephemeris for this satellite.
DifferentialValid	3	Receiver has a valid differential corrections for this satellite.
UsedForPVT	4	Satellite is being used by the receiver in the PVT solution.
AzimuthElevationValid	5	The azimuth and elevation of this satellite are valid.
UsedForRTK	6	Satellite is being used by the receiver in the RTK solution.

TABLE 2.47

Enumeration: Qi

NAME	VALUE	DESCRIPTION
None	0	No signal
Searching	1	Searching signal
Acquired	2	Signal acquired‘
Unstable	3	Signal detected but unstable
CodeLocked	4	Code locked and time synchronized
CarrierLocked5	5	Code & carrier locked and time synchronized
CarrierLocked6	6	Code & carrier locked and time synchronized
CarrierLocked7	7	Code & carrier locked and time synchronized

TABLE 2.48

2.5.16 GnssRawMeas

Output Size (Bytes) 40
Output Bit Offset 16
Description . . . . . Raw measurements pertaining to each GNSS satellite in view.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Tow	double	s	Time of week in seconds.
8	Week	uint16	–	GPS week
10	NumSats	uint8	–	Number of satellite measurements
11	Resv	uint8	–	Reserved.
12+28k	Sys	uint8	ENUM	GNSS constellation indicator.
13+28k	SvId	uint8	–	Space vehicle Id.
14+28k	Freq	uint8	ENUM	Frequency indicator.
15+28k	Chan	uint8	ENUM	Channel indicator.
16+28k	Slot	int8	–	Slot ID.
17+28k	Cno	uint8	dB Hz	Carrier-to-noise density ratio.
18+28k	Flags	uint16	BITS	Tracking info flags.
20+28k	Pr	double	–	Pseudorange measurement in cycles.
28+28k	Cp	double	–	Carrier phase measurement in cycles.
36+28k	Dp	float	Hz	Doppler measurement. Positive sign for approaching satellites.

Variable Length Output The size of this measurement output depends on the number of satellites or signals being tracked by the GNSS receiver, and may change every time the measurement is output.

The first 12 bytes of this measurement should be treated as a measurement header that applies to the rest of the measurement output. The remaining fields are output for each signal tracked by the GNSS receiver, repeating over k=0 to NumSats-1.

Enumeration: Sys

NAME	VALUE	DESCRIPTION
GPS	0	Global Positioning System (USA)
SBAS	1	Satellite Based Augmentation System
Galileo	2	Galileo (European Union)
BeiDou	3	BeiDou Navigation Satellite System (China)
IMES	4	Indoor Messaging System (QZSS extension)
QZSS	5	Quasi-Zenith Satellite System (Japan)
GLONASS	6	Globalnaya Navigatsionnaya Sputnikovaya Sistema (Russia)
IRNSS	7	Indian Regional Navigation Satellite System
LBand	8	L-Band correction satellite
Unknown	255	Unknown satellite system

TABLE 2.50

Enumeration: Freq

NAME	VALUE	DESCRIPTION
L1	1	L1(GPS,QZSS,SBAS), G1(GLO), E1(GAL), B1C/B1A(BDS).
Unknown	255	Unknown.

TABLE 2.51

Enumeration: Chan

NAME	VALUE	DESCRIPTION
C	1	C/A-code (GPS,GLO,SBAS,QZSS), C-chan (GAL).
Unknown	255	Unknown.

TABLE 2.52

Bit Field: Flags

NAME	OFFSET	DESCRIPTION
Searching	0	Searching signal.
Tracking	1	Tracking signal.
TimeValid	2	Time synchronized.
CodeLock	3	Code locked and time synchronized.
PhaseLock	4	Carrier phase valid.
PhaseHalfAmbiguity	5	Half cycle valid.
PhaseHalfSub	6	Half cycle substracted from phase.
PhaseSlip	7	Carrier phase slip detected.
PseudorangeSmoothed	8	Carrier smoothed pseudorange used.

TABLE 2.53

2.6 ATTITUDE GROUP

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

Bit Field: Attitude Group Output Types

NAME	OFFSET	DESCRIPTION
Ypr	1	The estimated attitude describing the body frame with respect to the local North-East-Down (NED) frame given as the (3-2-1) set of Euler angles corresponding to Yaw-Pitch-Roll.
Quaternion	2	The estimated attitude describing the body frame with respect to the local North-East-Down (NED) frame given as the quaternion.
Dcm	3	The estimated attitude given as the directional cosine matrix (DCM) in column major order mapping the local North-East-Down (NED) frame into the body frame.
MagNed	4	The current estimated magnetic field given in the local North-East-Down (NED) frame.
AccelNed	5	The estimated acceleration (with gravity) given in the local North-East-Down (NED) frame.
LinBodyAcc	6	The estimated linear acceleration (without gravity) given in the body frame.
LinAccelNed	7	The estimated linear acceleration (without gravity) given in the local North-East-Down (NED) frame.
YprU	8	The estimated attitude (Yaw, Pitch, Roll) uncertainty (1 Sigma), reported in degrees.

TABLE 2.54

2.6.1 Ypr

Output Size (Bytes) 12
Output Bit Offset 1
Description . . . . The estimated attitude describing the body frame with respect to the local North-East-Down (NED) frame given as the (3-2-1) set of Euler angles corresponding to Yaw-Pitch-Roll.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.

2.6.2 Quaternion

Output Size (Bytes) 16
Output Bit Offset 2
Description .... The estimated attitude describing the body frame with respect to the local North-East-Down (NED) frame given as the quaternion.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	QuatX	float	–	First vector component of quaternion.
4	QuatY	float	–	Second vector component of quaternion.
8	QuatZ	float	–	Third vector component of quaternion.
12	QuatS	float	–	Scalar component of quaternion.

2.6.3 Dcm

Output Size (Bytes) 36
Output Bit Offset 3
Description .... The estimated attitude given as the directional cosine matrix (DCM) in column major order mapping the local North-East-Down (NED) frame into the body frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	DCM00	float	–	Component of Dcm.
4	DCM01	float	–	Component of Dcm.
8	DCM02	float	–	Component of Dcm.
12	DCM10	float	–	Component of Dcm.
16	DCM11	float	–	Component of Dcm.
20	DCM12	float	–	Component of Dcm.
24	DCM20	float	–	Component of Dcm.
28	DCM21	float	–	Component of Dcm.
32	DCM22	float	–	Component of Dcm.

This measurement 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,

$$DCM = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \tag{2.1}$$

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

2.6.4 MagNed

Output Size (Bytes) 12
Output Bit Offset 4
Description . . . . . . The current estimated magnetic field given in the local North-East-Down (NED) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	MagN	float	G	Compensated magnetic measurement in North direction.
4	MagE	float	G	Compensated magnetic measurement in East direction.
8	MagD	float	G	Compensated magnetic measurement in Down direction.

The current attitude solution is used to map the magnetic measurement from the body frame into the inertial (NED) frame. This measurement is compensated by the factory calibration stored in flash, the user-applied calibration, and the Real-Time HSI compensation.

2.6.5 AccelNed

Output Size (Bytes) 12
Output Bit Offset 5
Description . . . . . . The estimated acceleration (with gravity) given in the local North-East-Down (NED) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	AccelN	float	m/s^2	Compensated acceleration measurement in North direction.
4	AccelE	float	m/s^2	Compensated acceleration measurement in East direction.
8	AccelD	float	m/s^2	Compensated acceleration measurement in Down direction.

The current attitude solution is used to map the acceleration measurement from the body frame into the inertial (NED) frame. This measurement is compensated by the factory calibration stored in flash, the user-applied calibration, and the real-time accelerometer bias estimated by the attitude filter. If the device is stationary and the attitude filter is tracking, the measurement should nominally be equivalent to the gravity reference vector in the inertial (NED) frame.

2.6.6 LinBodyAcc

Output Size (Bytes) 12
Output Bit Offset 6
Description . . . . . . The estimated linear acceleration (without gravity) given in the body frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	LinAccelX	float	m/s^2	Linear acceleration in body-frame x-axis.
4	LinAccelY	float	m/s^2	Linear acceleration in body-frame y-axis.
8	LinAccelZ	float	m/s^2	Linear acceleration in body-frame z-axis.

The current attitude solution is used to map the gravity reference vector from the inertial (NED) frame into the body frame and remove the gravity component from the acceleration measurement. This measurement is also compensated by the factory calibration stored in flash, the user-applied calibration, and the real-time accelerometer bias estimated by the attitude filter. If the device is stationary and the attitude filter is tracking, the measurement should nominally be equivalent to zero in all three axes.

2.6.7 LinAccelNed

Output Size (Bytes) 12
Output Bit Offset 7
Description .... The estimated linear acceleration (without gravity) given in the local North-East-Down (NED) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	LinAccelN	float	m/s^2	Linear acceleration in North direction.
4	LinAccelE	float	m/s^2	Linear acceleration in East direction.
8	LinAccelD	float	m/s^2	Linear acceleration in Down direction.

The current attitude solution is used to map the acceleration measurement from the body frame into the inertial (NED) frame. The gravity component is then removed from the acceleration measurement using the current gravity reference vector. This acceleration measurement is also compensated by the factory calibration stored in flash, the user-applied calibration, and the real-time accelerometer bias estimated by the attitude filter. If the device is stationary and the attitude filter is tracking, the measurement should nominally be equivalent to zero in all three axes.

2.6.8 YprU

Output Size (Bytes) 12
Output Bit Offset 8
Description .... The estimated attitude (Yaw, Pitch, Roll) uncertainty (1 Sigma), reported in degrees.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	YawU	float	deg	Filter estimated yaw uncertainty.
4	PitchU	float	deg	Filter estimated pitch uncertainty.
8	RollU	float	deg	Filter estimated roll uncertainty.

2.7 INS GROUP

The INS group provides all estimated outputs which are dependent upon the onboard INS state solution. All of the fields in this group will only be valid if the INS filter is currently enabled and tracking.

Bit Field: Ins Group Output Types

NAME	OFFSET	DESCRIPTION
InsStatus	0	The INS status bitfield.
PosLla	1	The estimated position given as latitude, longitude, and altitude.
PosEcef	2	The estimate position given in the Earth centered Earth fixed (ECEF) frame.
VelBody	3	The estimated velocity in the body-frame.
VelNed	4	The estimated velocity in the North East Down (NED) frame.
VelEcef	5	The estimated velocity in the Earth centered Earth fixed (ECEF) frame.
MagEcef	6	The compensated magnetic measurement in the Earth centered Earth fixed (ECEF) frame.
AccelEcef	7	The estimated acceleration (with gravity) given in the Earth centered Earth fixed (ECEF) frame.
LinAccelEcef	8	The estimated linear acceleration (without gravity) and given in the Earth centered Earth fixed (ECEF) frame.
PosU	9	The estimated uncertainty (1 Sigma) in the current position estimate.
VelU	10	The estimated uncertainty (1 Sigma) in the current velocity estimate.

TABLE 2.63

2.7.1 InsStatus

Output Size (Bytes) 2
Output Bit Offset 0
Description . . . . The INS status bitfield.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	InsStatus	uint16	BITS	Ins status bitfield.

Heading Data Source

- Mode 0: The heading is entirely magnetometer-derived.
- Mode 1: The heading is either entirely magnetometer-derived, or is in the process of switching from magnetometer-derived to GNSS-derived.
- Mode 2: The heading is entirely GNSS-derived. The magnetometer is entirely ignored.

Bit Field: InsStatus

NAME	OFFSET	DESCRIPTION
Mode	0	Two-bit enumeration that indicates the current mode of the INS filter.
GnssFix	2	Indicates whether the GNSS has a valid fix.
Resv1	3	Reserved.
ImuErr	4	High if gyro or accelerometer subsystem error is detected.
MagPresErr	5	High if magnetometer or pressure subsystem error is detected.
GnssErr	6	High if GNSS communication error is detected or if no valid PPS signal is received.
Resv2	7	Reserved for internal use. May toggle state during runtime and should be ignored.
Resv3	8	Reserved.

TABLE 2.65

Enumeration: Mode

NAME	VALUE	DESCRIPTION
NotTracking	0	The INS filter is non-operational and outputs are invalid. Attitude outputs are provided by the AHRS filter, if AHRS Aiding is enabled in Register 67.
Aligning	1	The INS filter has initialized. Position and velocity outputs are valid, but attitude outputs are not.
Tracking	2	The INS filter is tracking and all outputs are valid.
GnssLost	3	An extended GNSS outage has occurred. Position and velocity outputs are invalid, but the attitude outputs remain valid.

TABLE 2.66

2.7.2 PosLla

Output Size (Bytes) 24
Output Bit Offset 1
Description . . . . The estimated position given as latitude, longitude, and altitude.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	PosLat	double	deg	Geodetic latitude.
8	PosLon	double	deg	Longitude.
16	PosAlt	double	m	Altitude above WGS84 ellipsoid.

2.7.3 PosEcef

Output Size (Bytes) 24
Output Bit Offset 2
Description . . . . The estimate position given in the Earth centered Earth fixed (ECEF) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	PosEX	double	m	Position in ECEF-frame x-axis.
8	PosEY	double	m	Position in ECEF-frame y-axis.
16	PosEZ	double	m	Position in ECEF-frame z-axis.

2.7.4 VelBody

Output Size (Bytes) 12
Output Bit Offset 3
Description . . . . The estimated velocity in the body-frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	VelX	float	m/s	Velocity in body-frame x-axis.
4	VelY	float	m/s	Velocity in body-frame y-axis.
8	VelZ	float	m/s	Velocity in body-frame z-axis.

2.7.5 VelNed

Output Size (Bytes) 12
Output Bit Offset 4
Description . . . . The estimated velocity in the North East Down (NED) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	VelN	float	m/s	Velocity in North direction.
4	VelE	float	m/s	Velocity in East direction.
8	VelD	float	m/s	Velocity in Down direction.

2.7.6 VelEcef

Output Size (Bytes) 12
Output Bit Offset 5
Description .... The estimated velocity in the Earth centered Earth fixed (ECEF) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	VelEX	float	m/s	Velocity in ECEF-frame x-axis.
4	VelEY	float	m/s	Velocity in ECEF-frame y-axis.
8	VelEZ	float	m/s	Velocity in ECEF-frame z-axis.

2.7.7 MagEcef

Output Size (Bytes) 12
Output Bit Offset 6
Description .... The compensated magnetic measurement in the Earth centered Earth fixed (ECEF) frame.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	MagEX	float	G	Compensated magnetic measurement in ECEF-frame x-axis.
4	MagEY	float	G	Compensated magnetic measurement in ECEF-frame y-axis.
8	MagEZ	float	G	Compensated magnetic measurement in ECEF-frame z-axis.

2.7.8 AccelEcef

Output Size (Bytes) 12
Output Bit Offset 7
Description .... The estimated acceleration (with gravity) given in the Earth centered Earth fixed (ECEF) frame. The acceleration measurement has been bias compensated by the onboard INS filter. This measurement is attitude dependent because the attitude is used to map the measurement from the body-frame into the inertial (ECEF) frame. If the device is stationary and the INS filter is tracking, the measurement should be nominally equivalent to the gravity reference vector in the inertial frame (ECEF).

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	AccelEX	float	m/s^2	Compensated acceleration measurement in ECEF-frame x-axis.
4	AccelEY	float	m/s^2	Compensated acceleration measurement in ECEF-frame y-axis.
8	AccelEZ	float	m/s^2	Compensated acceleration measurement in ECEF-frame z-axis.

2.7.9 LinAccelEcef

Output Size (Bytes) 12
Output Bit Offset 8
Description . . . . The estimated linear acceleration (without gravity) and given in the Earth centered Earth fixed (ECEF) frame. This measurement is attitude dependent as the attitude solution is used to map the measurement from the body-frame into the inertial (ECEF) frame. This acceleration measurement has been bias compensated by the onboard INS filter, and the gravity component has been removed using the current gravity reference vector estimate. If the device is stationary and the onboard INS filter is tracking, the measurement will nominally read 0 on all three axes.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	LinAccelEX	float	m/s^2	Linear acceleration in ECEF-frame x-axis.
4	LinAccelEY	float	m/s^2	Linear acceleration in ECEF-frame y-axis.
8	LinAccelEZ	float	m/s^2	Linear acceleration in ECEF-frame z-axis.

2.7.10 PosU

Output Size (Bytes) 4
Output Bit Offset 9
Description . . . . The estimated uncertainty (1 Sigma) in the current position estimate.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Pos Uncertainty	float	m	Filter estimated position uncertainty.

2.7.11 VelU

Output Size (Bytes) 4
Output Bit Offset 10
Description . . . . The estimated uncertainty (1 Sigma) in the current velocity estimate.

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Vel Uncertainty	float	m/s	Filter estimated velocity uncertainty.

3 CONFIGURATION REGISTERS

3.1 OVERVIEW

Sensor configuration registers can be queried using the Read RegisterCommand and modified using the Write RegisterCommand. Settings that have been modified but not saved to non-volatile memory using the Write SettingsCommand are cleared by a reset or power cycle.

Most settings take effect immediately, but some registers, labelled *Static Registers*, only take effect at startup (after having been saved to non-volatile memory, followed by a reset or power-cycle).

3.2 SYSTEM

3.2.1 User Tag

Register ID . . . 0
Size (Bytes) . . . 20
Description . . . User assigned tag register. Any values can be assigned to this register. They will be stored to flash upon issuing a write settings command.
Example Read Response . . . `$VNRRG,00,*5F`
Example Write Command . . . `$VNWRG,00,ABCDEF123*6D`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Tag	string[20]	–		User defined tag register. Up to 20 bytes or characters. If a string with more than 20 characters is given, then the string will be truncated to the first 20.

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

3.2.2 Baud Rate

Register ID 5
Size (Bytes) 4
Description . . . . . Serial Baud Rate
Example Read Response . . . . . `$VNRRG,05,115200*5D`
Example Write Command . . . . . `$VNWRG,05,115200*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	BaudRate	uint32	ENUM	115200	Serial baud rate
4	SerialPort	uint8	ENUM	–	Optional. Baud rate on serial port. If this parameter is not provided then the baud rate will be changed for the active serial port.

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

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

Enumeration: BaudRate

NAME	VALUE	DESCRIPTION
Baud9600	9600	Serial baud rate of 9600.
Baud19200	19200	Serial baud rate of 19200.
Baud38400	38400	Serial baud rate of 38400.
Baud57600	57600	Serial baud rate of 57600.
Baud115200	115200	Serial baud rate of 115200.
Baud128000	128000	Serial baud rate of 128000.
Baud230400	230400	Serial baud rate of 230400.
Baud460800	460800	Serial baud rate of 460800.
Baud921600	921600	Serial baud rate of 921600.

TABLE 3.3

Enumeration: SerialPort

NAME	VALUE	DESCRIPTION
ActiveSerial	0	Set baud rate on the serial port which is communicating the Write Register command.
Serial1	1	Set baud rate on serial port 1.
Serial2	2	Set baud rate on serial port 2.

TABLE 3.4

3.2.3 Async Data Output Type

Register ID 6
Size (Bytes) 4
Description .... Sets which ASCII measurement register is output as asynchronously.
Example Read Response .... `$VNRRG,06,0*69`
Example Write Command .... `$VNWRG,06,1,1*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Ador	uint32	ENUM	22	ASCII asynchronous data output register.
4	SerialPort	uint8	ENUM	–	Optional. Asynchronous data type on serial port. If this parameter is not provided then the ADOR will be changed for the active serial port.

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

Enumeration: Ador

NAME	VALUE	DESCRIPTION
OFF	0	Asynchronous output turned off
YPR	1	Yaw, Pitch, Roll: Register 8
QTN	2	Quaternion: Register 9
QMR	8	Quaternion, Magnetic, Acceleration and Angular Rates: Register 15
MAG	10	Magnetic Measurements: Register 17
ACC	11	Acceleration Measurements: Register 18
GYR	12	Angular Rate Measurements: Register 19
MAR	13	Magnetic, Acceleration and Angular Rate Measurements: Register 20
YMR	14	Yaw, Pitch, Roll, Magnetic, Acceleration, Angular Rate Measurements: Register 27
YBA	16	Yaw, Pitch, Roll, Body True Acceleration and Angular Rate Measurements: Register 239
YIA	17	Yaw, Pitch, Roll, Inertial True Acceleration and Angular Rate Measurements: Register 240
IMU	19	IMU Measurements: Register 54
GPS	20	GNSS Solution LLA: Register 58
GPE	21	GNSS Solution ECEF: Register 59
INS	22	INS LLA: Register 63
INE	23	INS ECEF: Register 64
ISL	28	INS LLA 2: Register 72
ISE	29	INS ECEF 2: Register 73
DTV	30	Delta Theta and Delta Velocity: Register 80
HVE	34	Heave: Register 115

TABLE 3.6

Enumeration: SerialPort

NAME	VALUE	DESCRIPTION
ActiveSerial	0	Set ADOR rate on the serial port which is communicating the Write Register command.
Serial1	1	Set ADOR on serial port 1.
Serial2	2	Set ADOR on serial port 2.

TABLE 3.7

3.2.4 Async Data Output Freq

Register ID 7
Size (Bytes) 4
Description . . . . Sets the output frequency of the ASCII message configured in the Async Data Output Type register
Example Read Response . . . . `$VNRRG,07,10*59`
Example Write Command . . . . `$VNWRG,07,10*5C`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Adof	uint32	Hz	40	ASCII asynchronous data output frequency
4	SerialPort	uint8	ENUM	–	Optional. Asynchronous data type frequency on serial port. If this parameter is not provided then the ADOF will be changed for the active serial port.

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

Enumeration: Adof

NAME	VALUE	DESCRIPTION
Rate0Hz	0	ASCII output at 0 Hz
Rate1Hz	1	ASCII output at 1 Hz
Rate2Hz	2	ASCII output at 2 Hz
Rate4Hz	4	ASCII output at 4 Hz
Rate5Hz	5	ASCII output at 5 Hz
Rate10Hz	10	ASCII output at 10 Hz
Rate20Hz	20	ASCII output at 20 Hz
Rate25Hz	25	ASCII output at 25 Hz
Rate40Hz	40	ASCII output at 40 Hz
Rate50Hz	50	ASCII output at 50 Hz
Rate100Hz	100	ASCII output at 100 Hz
Rate200Hz	200	ASCII output at 200 Hz

TABLE 3.9

Enumeration: SerialPort

NAME	VALUE	DESCRIPTION
ActiveSerial	0	Set ADOF rate on the serial port which is communicating the Write Register command.
Serial1	1	Set ADOF on serial port 1.
Serial2	2	Set ADOF on serial port 2.

TABLE 3.10

3.2.5 Communication Protocol Control

Register ID 30
Size (Bytes) 7
Description . . . . Contains parameters that controls the communication protocol used by the sensor.
Example Read Response . . . . `$VNRRG,30,0,0,0,0,1,0,1*6C`
Example Write Command . . . . `$VNWRG,30,0,0,0,0,1,3,1*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	AsciiAppend Count	uint8	ENUM	0	Selects which counter or timer to append to all ASCII messages.
1	AsciiAppend Status	uint8	ENUM	0	Selects which status to append to all ASCII messages.
2	SpiAppend Count	uint8	ENUM	0	Selects which counter or timer to append to all SPI messages.
3	SpiAppend Status	uint8	ENUM	0	Selects which status to append to all SPI messages.
4	Ascii Checksum	uint8	ENUM	1	Choose the type of checksum used for ASCII communications.
5	SpiChecksum	uint8	ENUM	0	Choose the type of checksum used for the SPI communications.
6	ErrorMode	uint8	ENUM	1	Choose the action taken when errors are generated.

Example Async Messages

- Async Message with AsyncCount Enabled

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

- Async Message with AsyncStatus Enabled

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

- Async Message with AsyncCount and AsyncStatus Enabled

```
$VNYPR,+010.071,+000.278,-002.026,T1162704,S0000*50
```

ASCII Append Count

The *AsciiAppendCount* field provides a means of appending a time or counter to the end of all ASCII output messages.

With the *AsciiAppendCount* field set to NONE a typical serial asynchronous message could appear as:

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

With the *AsciiAppendCount* field set to a non-zero value the same asynchronous message would appear as:

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

When the *AsciiAppendCount* field is enabled the counter will always be appended between the payload and the checksum, subsequent to *AsciiAppendStatus* (if selected). The counter will be preceded by the T character to distinguish it from an appended Status message, and is an unsigned 32-bit integer.

ASCII Append Status

The *AsciiAppendStatus* field provides a means of tracking real-time status information pertaining to the overall state of the sensor measurements and onboard filtering algorithm. With *AsciiAppendStatus* set to None, an ASCII message may appear as:

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

With the *AsciiAppendStatus* field set to a non-zero values, the same asynchronous message may appear as:

Enumeration: AsciiAppendCount

NAME	VALUE	DESCRIPTION
None	0	Off
SyncInCount	1	SyncIn Counter
SyncInTime	2	SyncIn Time
SyncOutCount	3	SyncOut Counter
GpsPps	4	GPS PPS Time
GpsTow	5	GPS Time of Week

TABLE 3.12

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

When the *AsciiAppendStatus* field is not None, the status will always be appended between the payload and checksum, prior to a possibly appended *AsciiAppendCount*. The counter will be preceded by the S character to distinguish it from the counter field. The status consists of 4 hexadecimal characters.

Enumeration: AsciiAppendStatus

NAME	VALUE	DESCRIPTION
None	0	Do not append any statuses to the end of configured ASCII messages.
Ahrs	1	Append AHRS Status to the end of ASCII messages.
Ins	2	Append INS Status to the end of ASCII messages.

TABLE 3.13

Enumeration: SpiAppendCount

NAME	VALUE	DESCRIPTION
None	0	None
SyncInCount	1	SyncIn Counter
SyncInTime	2	SyncIn Time
SyncOutCount	3	SyncOut Counter
GpsPps	4	GPS PPS Time
GpsTow	5	GPS Time of Week

TABLE 3.14

Enumeration: SpiAppendStatus

NAME	VALUE	DESCRIPTION
None	0	None
Ahrs	1	AHRS Status
Ins	2	INS Status

TABLE 3.15

ASCII Checksum

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

SPI Checksum

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

Enumeration: AsciiChecksum

NAME	VALUE	DESCRIPTION
Checksum8bit	1	8-Bit Checksum
Crc16bit	3	16-Bit CRC

TABLE 3.16

Enumeration: SpiChecksum

NAME	VALUE	DESCRIPTION
Off	0	OFF - No checksum
Checksum8bit	1	8-Bit Checksum
Crc16bit	3	16-Bit CRC

TABLE 3.17

Error Mode

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

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

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

Enumeration: ErrorMode

NAME	VALUE	DESCRIPTION
Ignore	0	Ignore Error
SendError	1	Send Error
AdorOff	2	Send Error and set ADOR register to OFF

TABLE 3.18

3.2.6 Synchronization Control

Register ID 32
Size (Bytes) 20
Description . . . . . Controls the conditions and behavior on which SyncIn Events and SyncOut Events occur, which allow the timing of the sensor to be synchronized with external devices.
Example Read Response . . . . . `$VNRRG,32,3,0,0,0,6,1,0,100000000,0*6B`
Example Write Command . . . . . `$VNWRG,32,5,1,1,0,1,0,1,100000,0*5F`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	SyncInMode	uint8	ENUM	3	Controls sensor behavior on each SyncIn Event.
1	SyncInEdge	uint8	ENUM	0	Controls the signal edge on which a SyncIn pulse is detected.
2	SyncInSkip Factor	uint16	–	0	Controls the skip factor applied to the input signal required for a SyncIn Event to occur.
4	Resv1	uint32	–	0	Reserved. Must be set to default value.
8	SyncOutMode	uint8	ENUM	6	Controls when the SyncOut counter increments.
9	SyncOut Polarity	uint8	ENUM	1	Controls the polarity of the SyncOut signal.
10	SyncOutSkip Factor	uint16	–	0	Controls the number of SyncOut counter triggers required for each SyncOut Event.
12	SyncOutPulse Width	uint32	ns	100000000	Controls the width of the SyncOut signal pulse.
16	Resv2	uint32	–	0	Reserved. Must be set to default value.

SyncIn Event

A SyncIn Event occurs each time the number of pulses on the SyncIn pin exceeds the *SyncInSkipFactor*. This allows SyncIn Events to occur at some multiple of the input signal such that a high-frequency input signal can be provided, then divided to the desired rate (e.g. providing a 10 kHz signal with a *SyncInSkipFactor* of 100 will yield a 100 Hz response). Upon each SyncIn Event, the *SyncInMode* behavior is executed, and Register 33 SyncInTime and SyncInCount are reset and incremented, respectively.

SyncOut Event

A SyncOut Event occurs each time the SyncOut counter exceeds the *SyncOutSkipFactor*. The SyncOut counter increments whenever the event specified in *SyncOutMode* occurs. The skip factor allows the SyncOut Events to occur at some multiple of the sensor event (e.g. a 400 Hz NavReady signal with a *SyncOutSkipFactor* of 2 will yield a 200 Hz SyncOut pulse aligned with the NavFilter outputs). Upon each SyncOut Event, a SyncOut signal pulse will occur and Register 33 SyncOutCount is incremented.

Sync In Mode

> ! If *SyncInMode* is set to ImuSample, the IMU will not be sampled without a SyncIn Event. At pulse rates other than the IMU rate this may cause unexpected behavior. This pulse should never occur below 200 Hz.

Enumeration: SyncInMode

NAME	VALUE	DESCRIPTION
Count	3	Count the number of SyncIn Events only.
ImuSample	4	Initiate an IMU sample on each SyncIn Event. The IMU will not be sampled without a SyncIn Event.
AsyncAll	5	Output all configured asynchronous ASCII, binary, or NMEA (if enabled) messages on each SyncIn Events. No asynchronous messages will output without a SyncIn Event.
Async0	6	Output asynchronous ASCII, binary, or NMEA (if enabled) messages that are configured to a rate of 0 on each SyncIn Event.

TABLE 3.20

Enumeration: SyncInEdge

NAME	VALUE	DESCRIPTION
RisingEdge	0	Detect a SyncIn pulse on a rising edge.
FallingEdge	1	Detect a SyncIn pulse on a falling edge.

TABLE 3.21

Enumeration: SyncOutMode

NAME	VALUE	DESCRIPTION
None	0	None
ImuStart	1	Trigger at start of IMU sampling.
ImuReady	2	Trigger when IMU measurements are available.
NavReady	3	Trigger when NavFilter outputs are available (e.g. attitude).
GpsPps	6	Trigger on a GPS PPS event (1 Hz) when a 3D fix is valid.

TABLE 3.22

Enumeration: SyncOutPolarity

NAME	VALUE	DESCRIPTION
NegativePulse	0	Idles high, outputs a negative pulse.
PositivePulse	1	Idles low, outputs a positive pulse.

TABLE 3.23

3.2.7 Binary Output Message Configuration #1

Register ID 75
Size (Bytes) 7
Description . . . . Allow for the construction of custom binary output messages with user specified measurements at a specified rate.
Example Read Response . . . . `$VNRRG,75,0,0,00*5D`
Example Write Command . . . . `$VNWRG,75,1,160,0*6E`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	AsyncMode	uint16	BITS	0	Selects whether the output message should be sent on the serial port(s) at a fixed rate.
2	RateDivisor	uint16	–	0	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ImuRate.
4	OutputGroups	uint8[n]	–	0	Hex value bitfield to select which output groups are active in the message. Typically a single byte.
5	OutputTypes	uint16[m]	–	0	Hex value bitfield to selects which output data types are active within the selected output groups. Typically one 16-bit word per active output group.

This register is variable length, depending on the desired number of binary output types and which output groups they belong to. Typically, there will be a single output group byte (n=1), followed by 16-bit words for each active group (m active groups). See the appendix for detailed examples of using this register.

In the event that the desired binary output has a output group bit offset greater than 7 or a output type bit offset greater than 15, then additional group bytes or type words are required. This is described in detail in the Binary Extensions appendix.

The number of *OutputTypes* present must be consistent with the number of output groups selected in the *OutputGroups* byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two *OutputTypes* parameters present (m = 2). If the number of OutputFields is inconsistent with the number of *OutputGroups* selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

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

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

Bit Field: AsyncMode

NAME	OFFSET	DESCRIPTION
Serial1	0	Message is sent out serial port 1 at a fixed rate.
Serial2	1	Message is sent out serial port 2 at a fixed rate.

TABLE 3.25

3.2.8 Binary Output Message Configuration #2

Register ID . . . 76
Size (Bytes) . . . 7
Description . . . Allow for the construction of custom binary output messages with user specified measurements at a specified rate.
Example Read Response . . . `$VNRRG,76,0,0,00*5E`
Example Write Command . . . `$VNWRG,76,1,160,0*6D`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	AsyncMode	uint16	BITS	0	Selects whether the output message should be sent on the serial port(s) at a fixed rate.
2	RateDivisor	uint16	–	0	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ImuRate.
4	OutputGroups	uint8[n]	–	0	Hex value bitfield to select which output groups are active in the message. Typically a single byte.
5	OutputTypes	uint16[m]	–	0	Hex value bitfield to selects which output data types are active within the selected output groups. Typically one 16-bit word per active output group.

This register is variable length, depending on the desired number of binary output types and which output groups they belong to. Typically, there will be a single output group byte (n=1), followed by 16-bit words for each active group (m active groups). See the appendix for detailed examples of using this register.

In the event that the desired binary output has a output group bit offset greater than 7 or a output type bit offset greater than 15, then additional group bytes or type words are required. This is described in detail in the Binary Extensions appendix.

> The number of *OutputTypes* present must be consistent with the number of output groups selected in the *OutputGroups* byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two *OutputTypes* parameters present (m = 2). If the number of OutputFields is inconsistent with the number of *OutputGroups* selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

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

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

Bit Field: AsyncMode

NAME	OFFSET	DESCRIPTION
Serial1	0	Message is sent out serial port 1 at a fixed rate.
Serial2	1	Message is sent out serial port 2 at a fixed rate.

TABLE 3.27

3.2.9 Binary Output Message Configuration #3

Register ID 77
Size (Bytes) 7
Description ...... Allow for the construction of custom binary output messages with user specified measurements at a specified rate.
Example Read Response .... `$VNRRG,77,0,0,00*5F`
Example Write Command .... `$VNWRG,77,1,160,0*6C`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	AsyncMode	uint16	BITS	0	Selects whether the output message should be sent on the serial port(s) at a fixed rate.
2	RateDivisor	uint16	–	0	Sets the fixed rate at which the message is sent out the selected serial port(s). The number given is a divisor of the ImuRate.
4	OutputGroups	uint8[n]	–	0	Hex value bitfield to select which output groups are active in the message. Typically a single byte.
5	OutputTypes	uint16[m]	–	0	Hex value bitfield to selects which output data types are active within the selected output groups. Typically one 16-bit word per active output group.

This register is variable length, depending on the desired number of binary output types and which output groups they belong to. Typically, there will be a single output group byte (n=1), followed by 16-bit words for each active group (m active groups). See the appendix for detailed examples of using this register.

In the event that the desired binary output has a output group bit offset greater than 7 or a output type bit offset greater than 15, then additional group bytes or type words are required. This is described in detail in the Binary Extensions appendix.

> The number of *OutputTypes* present must be consistent with the number of output groups selected in the *OutputGroups* byte(s). For example if groups 1 and 3 are selected (OutputGroup = 0x05 or 0b00000101), then there must be two *OutputTypes* parameters present (m = 2). If the number of OutputFields is inconsistent with the number of *OutputGroups* selected, then the unit will respond with an invalid parameter error when attempting to write to this register.

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

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

Bit Field: AsyncMode

NAME	OFFSET	DESCRIPTION
Serial1	0	Message is sent out serial port 1 at a fixed rate.
Serial2	1	Message is sent out serial port 2 at a fixed rate.

TABLE 3.29

3.2.10 NMEA Output 1

Register ID . . . 101
Size (Bytes) . . . 8
Description . . . Allows the user to select a set of NMEA messages to output to the configured serial port.
Example Read Response . . . `$VNRRG,101,0,0,0,0,00000000*6F`
Example Write Command . . . `$VNWRG,101,1,1,0,0,1*5B`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Port	uint8	ENUM	0	Sets which serial port(s) will output the configured NMEA message(s)
1	Rate	uint8	ENUM	0	Sets the output frequency the NMEA message(s) are sent out the selected serial port(s). The number given is the output rate in Hz. Messages derived from GNSS data can be configured for 1 or 5 Hz. Messages derived from INS solution can be configured for 1, 5, 10 or 20 Hz.
2	Mode	uint8	ENUM	0	Sets NMEA output mode.
3	Resv	uint8	–	0	Reserved. Must be set to default value.
4	MsgSelection	uint32	BITS	0	Hex value bitfield to select individual message types.

Enumeration: Port

NAME	VALUE	DESCRIPTION
None	0	None. NMEA messages are not automatically sent out either serial port.
Serial1	1	Messages are sent out serial port 1 at a fixed rate.
Serial2	2	Messages are sent out serial port 2 at a fixed rate.
Serial1And2	3	Messages are sent out both serial port 1 and 2 at a fixed rate.

TABLE 3.31

Enumeration: Rate

NAME	VALUE	DESCRIPTION
Rate0Hz	0	0 Hz.
Rate1Hz	1	1 Hz.
Rate5Hz	5	5 Hz.
Rate10Hz	10	10 Hz.
Rate20Hz	20	20 Hz.

TABLE 3.32

Enumeration: Mode

NAME	VALUE	DESCRIPTION
V41GPID	0	NMEA version 4.1 with GP Talker IDs.
V23GPID	1	NMEA version 2.3 with GP Talker IDs.
V41SYSID	2	NMEA version 4.1 with System-Dependent Talker IDs.

TABLE 3.33

Bit Field: MsgSelection

NAME	OFFSET	DESCRIPTION
RMC_GNSS	0	Recommended Minimum Sentence (GNSS).
RMC_INS	1	Recommended Minimum Sentence (INS).
GGA_GNSS	2	GNSS fix data and undulation (GNSS).
GGA_INS	3	GNSS fix data and undulation (INS).
GLL_GNSS	4	Geographic Position (GNSS).
GLL_INS	5	Geographic Position (INS).
GSA_GNSS	6	GNSS DOP and active satellites.
GSV_GNSS	7	GNSS satellites in view.
HDG_INS	8	Heading.
HDT_INS	9	Heading, true.
THS_INS	10	True heading and status.
VTG_GNSS	11	Course over ground and ground speed (GNSS).
VTG_INS	12	Course over ground and ground speed (INS).
ZDA_GNSS	13	UTC time and date (GNSS).
ZDA_INS	14	UTC time and date (INS).
PASHR_INS	15	Inertial attitude data.
TSS1_INS	16	Acceleration, heave, pitch, and attitude in TSS format.

TABLE 3.34

3.2.11 NMEA Output 2

Register ID . . . 102
Size (Bytes) . . . 8
Description . . . Allows the user to select a set of NMEA messages to output to the configured serial port.
Example Read Response . . . `$VNRRG,102,0,0,0,0,00000000*6C`
Example Write Command . . . `$VNWRG,102,1,1,0,0,1*58`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Port	uint8	ENUM	0	Sets which serial port(s) will output the configured NMEA message(s)
1	Rate	uint8	ENUM	0	Sets the output frequency the NMEA message(s) are sent out the selected serial port(s). The number given is the output rate in Hz. Messages derived from GNSS data can be configured for 1 or 5 Hz. Messages derived from INS solution can be configured for 1, 5, 10 or 20 Hz.
2	Mode	uint8	ENUM	0	Sets NMEA output mode.
3	Resv	uint8	–	0	Reserved. Must be set to default value.
4	MsgSelection	uint32	BITS	0	Hex value bitfield to select individual message types.

Enumeration: Port

NAME	VALUE	DESCRIPTION
None	0	None. NMEA messages are not automatically sent out either serial port.
Serial1	1	Messages are sent out serial port 1 at a fixed rate.
Serial2	2	Messages are sent out serial port 2 at a fixed rate.
Serial1And2	3	Messages are sent out both serial port 1 and 2 at a fixed rate.

TABLE 3.36

Enumeration: Rate

NAME	VALUE	DESCRIPTION
Rate0Hz	0	0 Hz.
Rate1Hz	1	1 Hz.
Rate5Hz	5	5 Hz.
Rate10Hz	10	10 Hz.
Rate20Hz	20	20 Hz.

TABLE 3.37

Enumeration: Mode

NAME	VALUE	DESCRIPTION
V41GPID	0	NMEA version 4.1 with GP Talker IDs.
V23GPID	1	NMEA version 2.3 with GP Talker IDs.
V41SYSID	2	NMEA version 4.1 with System-Dependent Talker IDs.

TABLE 3.38

Bit Field: MsgSelection

NAME	OFFSET	DESCRIPTION
RMC_GNSS	0	Recommended Minimum Sentence (GNSS).
RMC_INS	1	Recommended Minimum Sentence (INS).
GGA_GNSS	2	GNSS fix data and undulation (GNSS).
GGA_INS	3	GNSS fix data and undulation (INS).
GLL_GNSS	4	Geographic Position (GNSS).
GLL_INS	5	Geographic Position (INS).
GSA_GNSS	6	GNSS DOP and active satellites.
GSV_GNSS	7	GNSS satellites in view.
HDG_INS	8	Heading.
HDT_INS	9	Heading, true.
THS_INS	10	True heading and status.
VTG_GNSS	11	Course over ground and ground speed (GNSS).
VTG_INS	12	Course over ground and ground speed (INS).
ZDA_GNSS	13	UTC time and date (GNSS).
ZDA_INS	14	UTC time and date (INS).
PASHR_INS	15	Inertial attitude data.
TSS1_INS	16	Acceleration, heave, pitch, and attitude in TSS format.

TABLE 3.39

3.2.12 Legacy Compatibility Settings

Register ID 206
Size (Bytes) 4
Description . . . . . Configuration of settings for compatibility with legacy firmware. This register contains flags used to configure operation of certain features to operate in a backwards compatible way with prior firmware.
Example Read Response . . . . . `$VNRRG,206,0,0,0,0*47`
Example Write Command . . . . . `$VNWRG,206,1,1,1,1*42`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	InsLegacy	uint8	ENUM	0	Bitfield to control setting multiple INS-related legacy outputs.
1	GnssLegacy	uint8	BITS	0	Bitfield to control setting multiple GNSS-related legacy outputs.
2	ImuLegacy	uint8	ENUM	0	Bitfield to control setting multiple IMU-related legacy outputs.
3	Resv	uint8	–	0	Reserved. Must be set to default value.

Enumeration: InsLegacy

NAME	VALUE	DESCRIPTION
Current	0	Use current behavior in the INS.
Legacy	1	Use legacy behavior in INS messages.

TABLE 3.41

Bit Field: GnssLegacy

NAME	OFFSET	DESCRIPTION
LegacyGnssFix	0	0 to output GNSS fix value as given, 1 to limit GNSS fix value output to max of 3.
RequireReg55Reset	1	0 to apply GNSS Basic Configuration register (Register 55) dynamically, 1 to require Write Settings and reset.

TABLE 3.42

Enumeration: ImuLegacy

NAME	VALUE	DESCRIPTION
Current	0	Calibrated temperature from pressure sensor.
Legacy	1	Uncalibrated temperature from microcontroller.

TABLE 3.43

3.3 ATTITUDE

3.3.1 Magnetic and Gravity Reference Vectors

Register ID 21
Size (Bytes) 24
Description .... Calculated reference vectors derived from the World Magnetic model and Earth Gravity model or user defined reference vectors
Example Read Response .... `$VNRRG,21,0.234,0,0.4212,0,0,-9.79375*75`
Example Write Command .... `$VNWRG,21,0.234,0,0.4212,0,0,-9.79375*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	MagRefN	float	G	0.234	North magnetic reference in the inertial frame.
4	MagRefE	float	G	0.0	East magnetic reference in the inertial frame.
8	MagRefD	float	G	0.4212	Down magnetic reference in the inertial frame.
12	GravRefN	float	m/s^2	0	North gravity reference in the inertial frame.
16	GravRefE	float	m/s^2	0	East gravity reference in the inertial frame.
20	GravRefD	float	m/s^2	-9.79375	Down gravity reference in the inertial frame.

This register contains the reference vectors for the magnetic and gravitational fields as used by the onboard filter. The values map to either the user-set values or the results of calculations of the onboard reference models (see the Reference Vector Configuration Register, Register 83). When the reference values come from the onboard model(s), those values are read-only. When the reference models are disabled, the values reflect the user reference vectors and will be writable. For example, if the onboard World Magnetic Model is enabled and the onboard Gravitational Model is disabled, only the gravity reference values will be modified on a register write. Note that the user reference vectors will not be overwritten by the onboard models, but will retain their previous values for when the onboard models are disabled.

3.3.2 VPE Basic Control

Register ID 35
Size (Bytes) 4
Description .... Provides control over various features relating to the onboard attitude filtering algorithm.
Example Read Response .... `$VNRRG,35,1,1,1,1*75`
Example Write Command .... `$VNWRG,35,0,0,0,0*70`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Resv	uint8	–	1	Reserved. Must be set to default value.
1	HeadingMode	uint8	ENUM	1	Heading mode used by the VPE.
2	FilteringMode	uint8	ENUM	1	Enables the VPE to adaptively filter inputs to the Kalman filter.
3	TuningMode	uint8	ENUM	1	Enables the VPE to adaptively tune the Kalman filter.

For more information on the operation and effects of these parameters, see the sensor's User Manual.

Enumeration: HeadingMode

NAME	VALUE	DESCRIPTION
Absolute	0	Absolute heading mode.
Relative	1	Relative heading mode.
Indoor	2	Indoor/Auto heading mode.

TABLE 3.46

Enumeration: FilteringMode

NAME	VALUE	DESCRIPTION
Unfiltered	0	Pass unfiltered measurements to the Kalman filter.
AdaptivelyFiltered	1	Allow the VPE to adaptively filter input measurements to the Kalman filter.

TABLE 3.47

Enumeration: TuningMode

NAME	VALUE	DESCRIPTION
Static	0	Use static, pre-determined tuning parameters in the Kalman filter.
Adaptive	1	Allow the VPE to adaptively tune the Kalman filter.

TABLE 3.48

3.3.3 VPE Magnetometer Basic Tuning

Register ID 36
Size (Bytes) 36
Description . . . . . Provides control of the adaptive filtering and tuning for the magnetometer.
Example Read Response . . . . . `$VNRRG,36,4,4,4,5,5,5,5.5,5.5,5.5*75`
Example Write Command . . . . . `$VNWRG,36,4,3,2,4,3,2,4,3,2*6A`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	BaseTuningX	float	–	4	This sets the level of confidence placed in the magnetometer x-axis when no disturbances are present. A larger number provides better heading accuracy, but with more sensitivity to magnetic interference.
4	BaseTuningY	float	–	4	This sets the level of confidence placed in the magnetometer y-axis when no disturbances are present. A larger number provides better heading accuracy, but with more sensitivity to magnetic interference.
8	BaseTuningZ	float	–	4	This sets the level of confidence placed in the magnetometer z-axis when no disturbances are present. A larger number provides better heading accuracy, but with more sensitivity to magnetic interference.
12	Adaptive TuningX	float	–	5	Level of adaptive tuning for x-axis.
16	Adaptive TuningY	float	–	5	Level of adaptive tuning for y-axis.
20	Adaptive TuningZ	float	–	5	Level of adaptive tuning for z-axis.
24	Adaptive FilteringX	float	–	5.5	Level of adaptive filtering for x-axis.
28	Adaptive FilteringY	float	–	5.5	Level of adaptive filtering for y-axis.
32	Adaptive FilteringZ	float	–	5.5	Level of adaptive filtering for z-axis.

3.3.4 VPE Accelerometer Basic Tuning

Register ID 38
Size (Bytes) 36
Description . . . . Provides control of the adaptive filtering and tuning for the accelerometer.
Example Read Response . . . . `$VNRRG,38,6,6,6,3,3,3,5,5,5*64`
Example Write Command . . . . `$VNWRG,38,4,3,2,4,3,2,4,3,2*64`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	BaseTuningX	float	–	6	This sets the level of confidence placed in the accelerometer x-axis when no disturbances are present. A larger number provides better pitch/roll accuracy, but with more sensitivity to motion, vibrations, and shocks.
4	BaseTuningY	float	–	6	This sets the level of confidence placed in the accelerometer y-axis when no disturbances are present. A larger number provides better pitch/roll accuracy, but with more sensitivity to motion, vibrations, and shocks.
8	BaseTuningZ	float	–	6	This sets the level of confidence placed in the accelerometer z-axis when no disturbances are present. A larger number provides better pitch/roll accuracy, but with more sensitivity to motion, vibrations, and shocks.
12	Adaptive TuningX	float	–	3	Level of adaptive tuning for x-axis.
16	Adaptive TuningY	float	–	3	Level of adaptive tuning for y-axis.
20	Adaptive TuningZ	float	–	3	Level of adaptive tuning for z-axis.
24	Adaptive FilteringX	float	–	5	Level of adaptive filtering for x-axis.
28	Adaptive FilteringY	float	–	5	Level of adaptive filtering for y-axis.
32	Adaptive FilteringZ	float	–	5	Level of adaptive filtering for z-axis.

3.4 IMU

3.4.1 Magnetometer Calibration

Register ID 23
Size (Bytes) 48
Description . . . . . Allows the magnetometer to be calibrated, in addition the factory calibration, for hard and soft iron effects. All values are in the sensor frame.
Example Read Response . . . . . `$VNRRG,23,1,0,0,0,1,0,0,0,1,0,0,0*73`
Example Write Command . . . . `$VNWRG,23,0.71,0.011,0.012,0.013,0.72,0.014,0.015,0.016,0.73,0.1,0.2,0.3*47`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	MagGain00	float	–	1	Magnetometer calibration gain matrix, row 0, column 0.
4	MagGain01	float	–	0	Magnetometer calibration gain matrix, row 0, column 1.
8	MagGain02	float	–	0	Magnetometer calibration gain matrix, row 0, column 2.
12	MagGain10	float	–	0	Magnetometer calibration gain matrix, row 1, column 0.
16	MagGain11	float	–	1	Magnetometer calibration gain matrix, row 1, column 1.
20	MagGain12	float	–	0	Magnetometer calibration gain matrix, row 1, column 2.
24	MagGain20	float	–	0	Magnetometer calibration gain matrix, row 2, column 0.
28	MagGain21	float	–	0	Magnetometer calibration gain matrix, row 2, column 1.
32	MagGain22	float	–	1	Magnetometer calibration gain matrix, row 2, column 2.
36	MagBiasX	float	G	0	Magnetometer bias calibration, sensor-frame x-axis.
40	MagBiasY	float	G	0	Magnetometer bias calibration, sensor-frame y-axis.
44	MagBiasZ	float	G	0	Magnetometer bias calibration, sensor-frame z-axis.

> The calibration parameters defined in this register are applied in addition to the factory calibration saved on this device. As such, this register will neither have any impact on, nor eliminate, the parameters saved on the sensor during factory calibration.

This register contains twelve values representing the Hard and Soft Iron calibration parameters. The magnetic measurements are calibrated 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 this register allows for calibration. These values only need to be changed from their defaults in the event that hard/soft iron calibration needs to be performed, or changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} * \begin{Bmatrix} MX - B0 \\ MY - B1 \\ MZ - B2 \end{Bmatrix} \tag{3.1}$$

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

3.4.2 Accel Calibration

Register ID 25
Size (Bytes) 48
Description . . . . Allows the accelerometer to be calibrated, in addition the factory calibration, for scale factor, misalignment, and bias errors. All values are in the sensor frame.
Example Read Response . . . . `$VNRRG,25,1,0,0,0,1,0,0,0,1,0,0,0*75`
Example Write Command . . . . `$VNWRG,25,0.71,0.011,0.012,0.013,0.72,0.014,0.015,0.016,0.73,0.1,0.2,0.3*41`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	AccelGain00	float	–	1	Accelerometer calibration gain matrix, Row 0, Column 0.
4	AccelGain01	float	–	0	Accelerometer calibration gain matrix, Row 0, Column 1.
8	AccelGain02	float	–	0	Accelerometer calibration gain matrix, Row 0, Column 2.
12	AccelGain10	float	–	0	Accelerometer calibration gain matrix, Row 1, Column 0.
16	AccelGain11	float	–	1	Accelerometer calibration gain matrix, Row 1, Column 1.
20	AccelGain12	float	–	0	Accelerometer calibration gain matrix, Row 1, Column 2.
24	AccelGain20	float	–	0	Accelerometer calibration gain matrix, Row 2, Column 0.
28	AccelGain21	float	–	0	Accelerometer calibration gain matrix, Row 2, Column 1.
32	AccelGain22	float	–	1	Accelerometer calibration gain matrix, Row 2, Column 2.
36	AccelBiasX	float	m/s^2	0	Accelerometer bias calibration, sensor-frame x-axis.
40	AccelBiasY	float	m/s^2	0	Accelerometer bias calibration, sensor-frame y-axis.
44	AccelBiasZ	float	m/s^2	0	Accelerometer bias calibration, sensor-frame z-axis.

User calibration of biases is only recommended in certain circumstances, and calibrating scale factors and cross-axis sensitivities via the gain matrix is never recommended or required.

The calibration parameters defined in this register are applied in addition to the factory calibration saved on this device. As such, this register will neither have any impact on, nor eliminate, the parameters saved on the sensor during factory calibration.

This register contains twelve values representing the accelerometer calibration parameters. The accelerometer measurements are calibrated 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, this register allows for calibration. These values only need to be changed from their defaults in the event that changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} * \begin{Bmatrix} AX - B0 \\ AY - B1 \\ AZ - B2 \end{Bmatrix} \tag{3.2}$$

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.

3.4.3 Reference Frame Rotation

Static Register Changes to this register do not take effect in real-time. Changes must be saved with a Write Settings Command and the device reset or power-cycled to take effect.

Register ID 26
Size (Bytes) 36
Description . . . . . Allows for measurements to be rotated into a different reference frame. Defines the transformation from sensor-frame to body-frame.
Example Read Response . . . . . `$VNRRG,26,1,0,0,0,1,0,0,0,1*6A`
Example Write Command . . . . . `$VNWRG,26,1,0,0,0,-1,0,0,0,-1*6F`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	RFR00	float	–	1	Direction cosine matrix, Row 0, Column 0
4	RFR01	float	–	0	Direction cosine matrix, Row 0, Column 1
8	RFR02	float	–	0	Direction cosine matrix, Row 0, Column 2
12	RFR10	float	–	0	Direction cosine matrix, Row 1, Column 0
16	RFR11	float	–	1	Direction cosine matrix, Row 1, Column 1
20	RFR12	float	–	0	Direction cosine matrix, Row 1, Column 2
24	RFR20	float	–	0	Direction cosine matrix, Row 2, Column 0
28	RFR21	float	–	0	Direction cosine matrix, Row 2, Column 1
32	RFR22	float	–	1	Direction cosine matrix, Row 2, Column 2

The matrix RFR must be an orthonormal, right-handed matrix. The sensor will output an error if the tolerance is not within 1e-5. The sensor will also report an error if any of the parameters are greater than 1 or less than -1.

Register Dependency Register 57 and Register 105 are defined in the body-frame and, as such, must be updated after updating this register.

This register contains a matrix that transforms outputs from the sensor-frame to a user defined body-frame. The use of this register allows for the sensor to be placed in any arbitrary orientation with respect to the user's desired body reference frame. This register also allows to correct for any orientation errors due to mounting the sensor on the user's vehicle or platform.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_B = \begin{bmatrix} RFR00 & RFR01 & RFR02 \\ RFR10 & RFR11 & RFR12 \\ RFR20 & RFR21 & RFR22 \end{bmatrix} * \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_S \tag{3.3}$$

The variables {X,Y,Z}_B are a measured parameter (e.g. acceleration) in the body reference frame. The variables {X,Y,Z}_S are a measured parameter (e.g. acceleration) in the sensor reference frame. The reference frame rotation register thus needs to be loaded with the matrix that will transform measurements from the sensor reference frame to the user-defined body reference frame.

3.4.4 Delta Theta and Delta Velocity Configuration

Register ID 82
Size (Bytes) 6
Description . . . . . Configures the Delta outputs. This does not impact coning and sculling calculations of the internal EKF.
Example Read Response . . . . . `$VNRRG,82,0,0,0,0,0*65`
Example Write Command . . . . . `$VNWRG,82,1,1,1,0,0*61`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Integration Frame	uint8	ENUM	0	Output reference frame for delta velocity.
1	Gyro Compensation	uint8	ENUM	0	Selects which angular rate measurements to use for delta computation.
2	Accel Compensation	uint8	ENUM	0	Selects which acceleration measurements to use for delta computation.
3	EarthRate Compensation	uint8	ENUM	0	Selects which measurements to compensate for Earth's rotation rate of 15 °/hr before integration.
4	Resv	uint16	–	0	Reserved. Must be set to default value.

The Delta Theta and Delta Velocity Configuration register allows configuration of the onboard coning and sculling used to generate integrated motion values from the angular rate and acceleration measurements. The fully coupled coning and sculling integrals are computed at the IMU Rate, ensuring maximum accuracy. These settings only impact the deltas output to the user and do not have any impact on the internal Kalman filter.

Enumeration: IntegrationFrame

NAME	VALUE	DESCRIPTION
Body	0	Body frame.
NED	1	NED (North, East, Down) frame. Relies on the internal EKF.

TABLE 3.55

Enumeration: GyroCompensation

NAME	VALUE	DESCRIPTION
None	0	Use uncompensated gyroscope measurements.
Bias	1	Use bias-compensated gyroscope measurements. Relies on the internal EKF.

TABLE 3.56

Enumeration: AccelCompensation

NAME	VALUE	DESCRIPTION
None	0	Use uncompensated acceleration measurements.
Gravity	1	Use gravity-removed acceleration measurements. Relies on the internal EKF.
Bias	2	Use bias-removed acceleration measurements. Relies on the internal EKF.
BiasAndGravity	3	Use gravity- and bias-removed acceleration measurements. Relies on the internal EKF.

TABLE 3.57

Enumeration: EarthRateCompensation

NAME	VALUE	DESCRIPTION
None	0	No compensation for Earth's angular rate.
GyroRate	1	Gyro measurements compensated for Earth's angular rate.
CoriolisAccel	2	Acceleration measurements compensated for Coriolis effect.
RateAndCoriolis	3	Both gyro and accelerometer measurements compensated for Earth angular rate and Coriolis effect.

TABLE 3.58

3.4.5 Gyro Calibration

Register ID 84
Size (Bytes) 48
Description . . . . . . Allows the gyro to be calibrated, in addition the factory calibration, for scale factor, misalignment, and bias errors. All values are in the sensor frame.
Example Read Response . . . . . . `$VNRRG,84,1,0,0,0,1,0,0,0,1,0,0,0*7E`
Example Write Command . . . . `$VNWRG,84,0.71,0.011,0.012,0.013,0.72,0.014,0.015,0.016,0.73,0.1,0.2,0.3*4A`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	GyroGain00	float	–	1	Gyroscope calibration gain matrix, Row 0, Column 0.
4	GyroGain01	float	–	0	Gyroscope calibration gain matrix, Row 0, Column 1.
8	GyroGain02	float	–	0	Gyroscope calibration gain matrix, Row 0, Column 2.
12	GyroGain10	float	–	0	Gyroscope calibration gain matrix, Row 1, Column 0.
16	GyroGain11	float	–	1	Gyroscope calibration gain matrix, Row 1, Column 1.
20	GyroGain12	float	–	0	Gyroscope calibration gain matrix, Row 1, Column 2.
24	GyroGain20	float	–	0	Gyroscope calibration gain matrix, Row 2, Column 0.
28	GyroGain21	float	–	0	Gyroscope calibration gain matrix, Row 2, Column 1.
32	GyroGain22	float	–	1	Gyroscope calibration gain matrix, Row 2, Column 2.
36	GyroBiasX	float	rad/s	0	Gyroscope bias calibration, sensor-frame x-axis.
40	GyroBiasY	float	rad/s	0	Gyroscope bias calibration, sensor-frame y-axis.
44	GyroBiasZ	float	rad/s	0	Gyroscope bias calibration, sensor-frame z-axis.

> **!** User calibration of biases is only recommended in certain circumstances, and calibrating scale factors and cross-axis sensitivities via the gain matrix is never recommended or required.

> **i** The calibration parameters defined in this register are applied in addition to the factory calibration saved on this device. As such, this register will neither have any impact on, nor eliminate, the parameters saved on the sensor during factory calibration.

This register contains twelve values representing the gyro calibration parameters. The gyro measurements are calibrated for changes in bias, gain, and axis alignment that can occur during the installation of the chip on the customer's board using the following model. Under normal circumstances this register can be left in its factory default state. In the event that there are significant changes to the gyro bias, gain, and axis alignment during installation or during the life of the part; this register allow for calibration. These values only need to be changed from their defaults in the event that changes in bias, gain, and axis alignment have occurred at some point between the times the chip was calibrated at the factory and when it is used in the field.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} * \begin{Bmatrix} GX - B0 \\ GY - B1 \\ GZ - B2 \end{Bmatrix} \tag{3.4}$$

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

3.4.6 IMU Filtering Configuration

Register ID 85
Size (Bytes) 15
Description . . . . Controls the level of filtering performed on the output IMU measurements.
Example Read Response . . . . `$VNRRG,85,0,4,4,4,0,0,3,3,3,0*79`
Example Write Command . . . . `$VNWRG,85,4,0,0,0,4,0,0,0,0,0*7B`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	MagWindow Size	uint16	–	0	Number of previous measurements averaged for output magnetic measurements.
2	AccelWindow Size	uint16	–	4	Number of previous measurements averaged for output acceleration measurements.
4	GyroWindow Size	uint16	–	4	Number of previous measurements averaged for output gyro measurements.
6	TempWindow Size	uint16	–	4	Number of previous measurements averaged for output temperature measurements.
8	PresWindow Size	uint16	–	0	Number of previous measurements averaged for output pressure measurements.
10	MagFilterMode	uint8	BITS	0	Filtering mode for magnetic measurements.
11	AccelFilter Mode	uint8	BITS	3	Filtering mode for acceleration measurements.
12	GyroFilterMode	uint8	BITS	3	Filtering mode for gyro measurements.
13	TempFilter Mode	uint8	BITS	3	Filtering mode for temperature measurements.
14	PresFilterMode	uint8	BITS	0	Filtering mode for pressure measurements.

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

Bit Field: MagFilterMode

NAME	OFFSET	DESCRIPTION
Uncomp	0	Perform filtering on uncompensated measurement outputs.
Comp	1	Perform filtering on compensated measurement outputs.

TABLE 3.61

Bit Field: AccelFilterMode

NAME	OFFSET	DESCRIPTION
Uncomp	0	Perform filtering on uncompensated measurement outputs.
Comp	1	Perform filtering on compensated measurement outputs.

TABLE 3.62

Bit Field: GyroFilterMode

NAME	OFFSET	DESCRIPTION
Uncomp	0	Perform filtering on uncompensated measurement outputs.
Comp	1	Perform filtering on compensated measurement outputs.

TABLE 3.63

Bit Field: TempFilterMode

NAME	OFFSET	DESCRIPTION
Uncomp	0	Perform filtering on uncompensated measurement outputs.
Comp	1	Perform filtering on compensated measurement outputs.

TABLE 3.64

Bit Field: PresFilterMode

NAME	OFFSET	DESCRIPTION
Uncomp	0	Perform filtering on uncompensated measurement outputs.
Comp	1	Perform filtering on compensated measurement outputs.

TABLE 3.65

3.5 HARD/SOFT IRON ESTIMATOR

3.5.1 Real-Time HSI Control

Register ID 44
Size (Bytes) 3
Description . . . . . Controls basic parameters for the onboard Real-Time Hard and Soft Iron (HSI) Estimator.
Example Read Response . . . . . `$VNRRG,44,0,1,5*6B`
Example Write Command . . . . . `$VNWRG,44,1,1,5*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Mode	uint8	ENUM	0	Controls the mode of operation for the Real-Time HSI Estimator.
1	Apply Compensation	uint8	ENUM	1	Controls whether the latest Real-Time HSI Estimator solution is applied to the magnetic measurements. These measurements are used as input to the attitude filter and are available to output as the compensated magnetometer outputs.
2	ConvergeRate	uint8	–	5	Controls how quickly the Real-Time HSI Estimator is allowed to converge onto a new solution.

The magnetometer is only used to provide a coarse heading estimate when GNSS is unavailable and is completely ignored by the INS filter during normal operation. A Hard/Soft Iron calibration may be performed to try and improve the startup magnetic heading, but is not required for normal operaiton.

The Real-Time HSI Estimator solution is stored in Register 47. This solution is separate from, and applied in addition to, any calibration parameters stored in Register 23.

Mode

When the Real-Time Estimator's mode switches from Run to Off, the solution does not clear. As such, when the the estimator switches from Off to Run, it will continue using its last-calculated solution. The estimator can be started and stopped at any time by switching between the Run and Off modes, and will only clear the solution when the mode is set to Reset.

Enumeration: Mode

NAME	VALUE	DESCRIPTION
Off	0	Real-Time HSI Estimator is turned off.
Run	1	Runs the Real-Time HSI Estimator.
Reset	2	Resets the Real-Time HSI Estimator solution.

TABLE 3.67

Enumeration: ApplyCompensation

NAME	VALUE	DESCRIPTION
Disable	1	Real-time HSI estimate is not applied to the magnetic measurements.
Enable	3	Real-time HSI estimate is applied to the magnetic measurements.

TABLE 3.68

Convergence Rate

The slower the convergence the more accurate the estimate of the hard/soft iron solution. A quicker convergence will provide a less accurate estimate of the hard/soft iron parameters, but for applications where the hard/soft iron changes rapidly may provide a more accurate attitude estimate.

Range: 1 to 5

1: Solution converges slowly over approximately 60-90 seconds.

5: Solution converges rapidly over approximately 15-20 seconds.

3.6 VELOCITY AIDING

3.6.1 Velocity Aiding Measurement

Register ID 50
Size (Bytes) 12
Description .... Input register for a velocity measurement to be used by the AHRS to compensate for acceleration disturbances.
Example Read Response .... `$VNRRG,50,0,0,0*6A`
Example Write Command .... `$VNWRG,50,1,2,3*6F`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	VelocityX	float	m/s	0	Velocity in the body-frame x-axis.
4	VelocityY	float	m/s	0	Velocity in the body-frame y-axis.
8	VelocityZ	float	m/s	0	Velocity in the body-frame z-axis.

If you stop sending velocity measurement updates for any reason, the velocity compensation will continue indefinitely using the last received velocity measurement. If you want to stop using while the vehicle is still in motion, be sure to turn off the velocity compensation using the Mode field in the Velocity Aiding Control register (Register 51).

This register is unused whenever there is GNSS available. When GNSS is available the INS is utilized for attitude, which is capable of tracking the acceleration due to motion based upon the measurements provided by the GNSS. This register is only utilized in the AHRS, which provides attitude when GNSS is unavailable for a prolonged period.

The sensor will compute the vector length of the provided 3D velocity vector and use this for velocity compensation in the body-frame x-axis. If you have a scalar measurement you can set only the x-axis and set the y- & z-axes to zero.

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

3.6.2 Velocity Aiding Control

Register ID 51
Size (Bytes) 12
Description . . . . . Provides control over the velocity aiding feature for the AHRS filter.
Example Read Response . . . . . `$VNRRG,51,1,0.1,0.01*5A`
Example Write Command . . . . . `$VNWRG,51,0,0.5,0.1*6A`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	VelAidEnable	uint8	ENUM	1	Selects whether to enable velocity aiding.
1	–PAD–	uint8[3]	–	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
4	VelUncert Tuning	float	–	0.1	Tuning parameter for the uncertainty of the velocity measurement.
8	Resv	float	–	0.01	Reserved. Must be default value.

This register is unused whenever there is GNSS available. When GNSS is available the INS is utilized for attitude, which is capable of tracking the acceleration due to motion based upon the measurements provided by the GNSS. This register is only utilized in the AHRS, which provides attitude when GNSS is unavailable for a prolonged period.

Enumeration: VelAidEnable

NAME	VALUE	DESCRIPTION
Disable	0	Disable velocity aiding.
Enable	1	Enable velocity aiding.

TABLE 3.71

Velocity Uncertainty Tuning

The Velocity Uncertainty Tuning provides a means to adjust the uncertainty level used for the velocity measurement. A larger value places less trust in the velocity measurements, which works best for noisy or unreliable velocity sources; a smaller number places more trust in the velocity measurement, which works best for an accurate velocity measurement.

3.7 GNSS

3.7.1 GNSS Basic Configuration

Register ID 55
Size (Bytes) 5
Description . . . . . Configures basic GNSS receiver parameters.
Example Read Response . . . . . `$VNRRG,55,0,0,5,0,1*6B`
Example Write Command . . . . . `$VNWRG,55,0,0,5,0,1*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	ReceiverEnable	uint8	ENUM	0	Enables GNSS receivers. If an external receiver is enabled, it becomes the Primary Receiver. If an internal receiver is not enabled, it is powered off.
1	PpsSource	uint8	ENUM	0	Source for the PPS hardware interrupt supplied by the Primary Receiver. If no external GNSS is configured in ReceiverEnable, PpsSource must be set to the default value.
2	Rate	uint8	Hz	5	GNSS receiver measurement rate.
3	Resv4	uint8	–	0	Reserved. Must be set to default value.
4	Resv5	uint8	–	1	Reserved. Must be set to default value.

Enumeration: ReceiverEnable

NAME	VALUE	DESCRIPTION
Internal	0	Enable internal GNSS receiver(s).
VnWrite	1	Accept external GNSS data by writing measurements to Register 58 or Register 59.
VnAdor	2	Accept external GNSS data from a VectorNav sensor via the Asynchronous Output Data Register.

TABLE 3.73

Enumeration: PpsSource

NAME	VALUE	DESCRIPTION
PpsPinRising	0	GPS PPS signal is supplied to the GPS PPS pin and should trigger on a rising edge.
PpsPinFalling	1	GPS PPS signal is supplied to the GPS PPS pin and should trigger on a falling edge.
SyncInRising	2	GPS PPS signal is supplied to the SyncIn pin and should trigger on a rising edge.
SyncInFalling	3	GPS PPS signal is supplied to the SyncIn pin and should trigger on a falling edge.

TABLE 3.74

Enumeration: Rate

NAME	VALUE	DESCRIPTION
Rate1Hz	1	Expect GNSS measurements at 1 Hz.
Rate5Hz	5	Expect GNSS measurements at 5 Hz.

TABLE 3.75

3.7.2 GNSS Internal A Antenna Offset

Register ID 57
Size (Bytes) 12
Description . . . . . Configures the position offset of GNSS Internal A (GnssA) antenna from the INS Reference Point in the body reference frame.
Example Read Response . . . . . `$VNRRG,57,0,0,0*6D`
Example Write Command . . . . . `$VNWRG,57,1,1,1*69`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	PositionX	float	m	0	Relative position of GnssA antenna, body-frame x-axis.
4	PositionY	float	m	0	Relative position of GnssA antenna, body-frame y-axis.
8	PositionZ	float	m	0	Relative position of GnssA antenna, body-frame z-axis.

Register Dependency The GNSS A Antenna Offset must be measured with respect to the body-frame and INS Reference Point as defined by the user in Register 26 and Register 105, respectively.

The position of the GnssA antenna relative to the INS Reference Point in the body reference frame, also referred to as the GnssA antenna lever arm. In the example scenario shown in the figure below, the GnssA antenna offset is X= 0.75 m, Y= −0.5 m, Z= 0 m.

GNSS Antenna Measurements

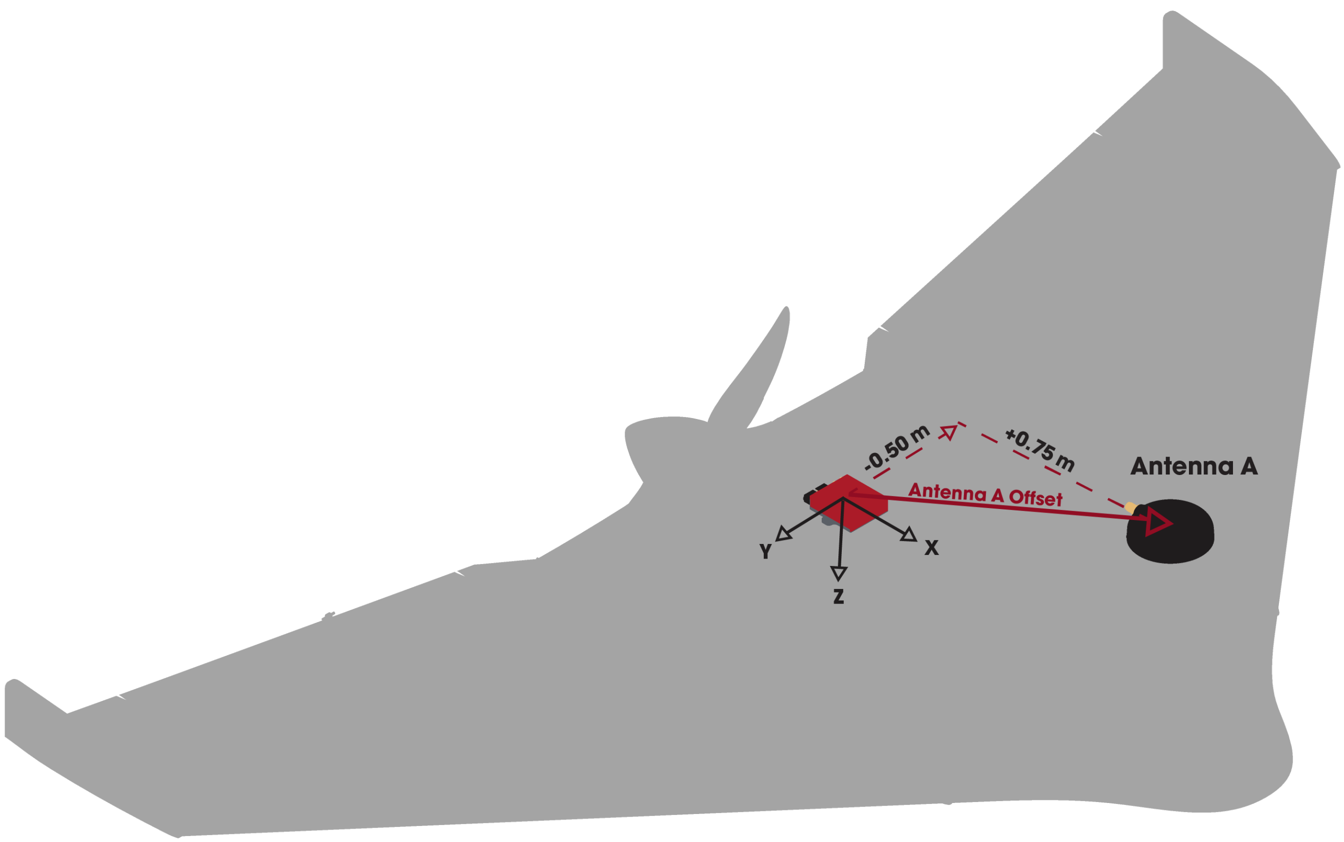

FIGURE 3.1

3.7.3 GNSS System Configuration

Register ID 99
Size (Bytes) 12
Description . . . . . Configuration of the GNSS constellations and satellites.
Example Read Response . . . . . `$VNRRG,99,0013,10,5,16,03,EBF9,003F,0000*4C`
Example Write Command . . . . . `$VNWRG,99,0013,10,10,16,02,EBF9,003F,0000*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Systems	uint16	BITS	0013	Hex value bitfield of enabled GNSS systems.
2	MinCno	uint8	dB	10	Minimum satellite CN0 to use for GNSS navigation solution. (Cannot be set greater than 50 dB)
3	MinElev	uint8	deg	5	Minimum satellite elevation angle to use for GNSS navigation solution. (Cannot be set greater than 90°)
4	MaxSats	uint8	–	16	Maximum number of satellites to use in GNSS navigation solution. (Cannot be set less than 4 or greater than 32)
5	SbasMode	uint8	BITS	03	Hex value bitfield configuring SBAS usage.
6	SbasSelect1	uint16	BITS	EBF9	Hex value bitfield masking individual SBAS satellites for use.
8	SbasSelect2	uint16	BITS	003F	Hex value bitfield masking individual SBAS satellites for use.
10	SbasSelect3	uint16	BITS	0000	Hex value bitfield masking individual SBAS satellites for use.

Systems

While the systems used by the GNSS receiver are configurable, it is recommended that GPS, SBAS, and only one other constellation is configured. Please reach out to VectorNav Support if a different system configuraiton is desired.

Bit Field: Systems

NAME	OFFSET	DESCRIPTION
GPS	0	GPS
SBAS	1	SBAS
GLONASS	2	GLONASS
Beidou	3	Beidou
Galileo	4	Galileo
IMES	5	IMES
QZSS_L1_CA	6	QZSS-L1 C/A
QZSS_L1_SAIF	7	QZSS-L1 SAIF

TABLE 3.78

Bit Field: SbasMode

NAME	OFFSET	DESCRIPTION
Ranging	0	Use ranging in navigation solution.
DiffCorr	1	Apply SBAS differential corrections.
Integrity	2	Use SBAS integrity information.
TestMode	3	Utilize signals from SBAS satellites broadcasting in test mode.

TABLE 3.79

Bit Field: SbasSelect1

NAME	OFFSET	DESCRIPTION
Sbas120	0	Satellite PRN 120
Sbas121	1	Satellite PRN 121
Sbas122	2	Satellite PRN 122
Sbas123	3	Satellite PRN 123
Sbas124	4	Satellite PRN 124
Sbas125	5	Satellite PRN 125
Sbas126	6	Satellite PRN 126
Sbas127	7	Satellite PRN 127
Sbas128	8	Satellite PRN 128
Sbas129	9	Satellite PRN 129
Sbas130	10	Satellite PRN 130
Sbas131	11	Satellite PRN 131
Sbas132	12	Satellite PRN 132
Sbas133	13	Satellite PRN 133
Sbas134	14	Satellite PRN 134
Sbas135	15	Satellite PRN 135

TABLE 3.80

Bit Field: SbasSelect2

NAME	OFFSET	DESCRIPTION
Sbas136	0	Satellite PRN 136
Sbas137	1	Satellite PRN 137
Sbas138	2	Satellite PRN 138
Sbas139	3	Satellite PRN 139
Sbas140	4	Satellite PRN 140
Sbas141	5	Satellite PRN 141
Sbas142	6	Satellite PRN 142
Sbas143	7	Satellite PRN 143
Sbas144	8	Satellite PRN 144
Sbas145	9	Satellite PRN 145
Sbas146	10	Satellite PRN 146
Sbas147	11	Satellite PRN 147
Sbas148	12	Satellite PRN 148
Sbas149	13	Satellite PRN 149
Sbas150	14	Satellite PRN 150
Sbas151	15	Satellite PRN 151

TABLE 3.81

Bit Field: SbasSelect3

NAME	OFFSET	DESCRIPTION
Sbas152	0	Satellite PRN 152
Sbas153	1	Satellite PRN 153
Sbas154	2	Satellite PRN 154
Sbas155	3	Satellite PRN 155
Sbas156	4	Satellite PRN 156
Sbas157	5	Satellite PRN 157
Sbas158	6	Satellite PRN 158

TABLE 3.82

3.8 INS

3.8.1 INS Basic Configuration

Register ID 67
Size (Bytes) 4
Description . . . . Configures basic parameters of the INS.
Example Read Response . . . . `$VNRRG,67,2,1,0,0*71`
Example Write Command . . . . `$VNWRG,67,2,1,0,0*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	Scenario	uint8	ENUM	2	Active INS scenario.
1	AhrsAiding	uint8	ENUM	1	Selects whether the INS should use an AHRS-based attitude solution during periods when there is insufficient GNSS-based attitude observability.
2	Resv1	uint8	–	0	Reserved. Must be set to default value.
3	Resv2	uint8	–	0	Reserved. Must be set to default value.

Enumeration: Scenario

NAME	VALUE	DESCRIPTION
Ahrs	0	AHRS (Attitude Only - no GNSS/INS)
GnssInsWithPressure	1	GNSS/INS - with barometric pressure sensor.
GnssInsNoPressure	2	GNSS/INS - without barometric pressure sensor.

TABLE 3.84

AhrsAiding

AHRS aiding provides the ability to switch to using the magnetometer to stabilize heading during times when GNSS-based heading is unavailable. Disabling AHRS aiding may reduce time to converge to a GNSS-based heading solution, as it removes the transition time from magnetometer-based heading; however if AHRS aiding is disabled and the INS is in Mode 0, attitude outputs will be zero.

Enumeration: AhrsAiding

NAME	VALUE	DESCRIPTION
Disable	0	Disable AHRS aiding.
Enable	1	Enable AHRS aiding.

TABLE 3.85

3.8.2 Filter Startup Bias

Register ID 74
Size (Bytes) 28
Description .... Sets the initial estimate for the INS filter biases in the Sensor Frame.
Example Read Response .... `$VNRRG,74,+00.000000,+00.000000,+00.000000,+00.000,+00.000,+00.000,+00000.000*69`
Example Write Command .... `$VNWRG,74,0.01,0.01,0.01,0.01,0.01,0.01,0.01*46`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	GyroBiasX	float	rad/s	0	X-axis gyro bias.
4	GyroBiasY	float	rad/s	0	Y-axis gyro bias.
8	GyroBiasZ	float	rad/s	0	Z-axis gyro bias.
12	AccelBiasX	float	m/s^2	0	X-axis accelerometer bias
16	AccelBiasY	float	m/s^2	0	Y-axis accelerometer bias
20	AccelBiasZ	float	m/s^2	0	Z-axis accelerometer bias
24	PresBias	float	kPa	0	Pressure bias

Static Register Changes to this register do not take effect in real-time. Changes must be save with a Write Settings Command and the device reset or power-cycled to take effect.

3.8.3 INS Reference Point Offset

Static Register Changes to this register do not take effect in real-time. Changes must be saved with a Write Settings Command and the device reset or power-cycled to take effect.

Register ID . . . 105
Size (Bytes) . . . 24
Description . . . Configures the position offset and measurement uncertainty of the INS Reference Point relative to the IMU sensor installation location in the body-frame.
Example Read Response . . . `$VNRRG,105,+00.000,+00.000,+00.000,00.0000,00.0000,00.0000*5C`
Example Write Command . . . `$VNWRG,105,1,1,1,1,1,1*42`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	RefOffsetX	float	m	0	Position of INS Reference Point relative to IMU location, body-frame x-axis.
4	RefOffsetY	float	m	0	Position of INS Reference Point relative to IMU location, body-frame y-axis.
8	RefOffsetZ	float	m	0	Position of INS Reference Point relative to IMU location, body-frame z-axis.
12	RefUncertX	float	m	0.0	Uncertainty in the x-axis offset measurement.
16	RefUncertY	float	m	0.0	Uncertainty in the y-axis offset measurement.
20	RefUncertZ	float	m	0.0	Uncertainty in the z-axis offset measurement.

Register Dependency The INS Reference Point Offset is measured in the user defined body-frame defined in Register 26. And it is applied prior to the antenna offset defined in Register 57; the antenna offset in Register 57 is defined relative to the new reference point, not the inertial sensor location.

By default, the INS Reference Point is colocated with the inertial sensor. The INS Reference Point allows the user to move the refence point to a different location, thus moving the output location of the position and velocity data. This offset does not impact the inertial data output (e.g. accelerations).

3.9 WORLD MAG/GRAVITY MODEL

3.9.1 Reference Model Configuration

Register ID	83
Size (Bytes)	40
Description	Control register for both the onboard world magnetic and gravity model corrections.
Example Read Response	`$VNRRG,83,1,1,0,0,1000,0.000,+00.00000000,+000.00000000,+00000.000*4E`
Example Write Command	`$VNWRG,83,0,0,0,0,10,2014,30,-94,100*45`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	EnableMag Model	uint8	ENUM	1	Sets whether to use the World Magnetic Model as the source for the magnetic reference vector in Register 21.
1	EnableGravity Model	uint8	ENUM	1	Sets whether to use the World Magnetic Model as the source for the gravity reference vector in Register 21.
2	Resv1	uint8	–	0	Reserved. Must be set to default value.
3	Resv2	uint8	–	0	Reserved. Must be set to default value.
4	Recalc Threshold	uint32	–	1000	Maximum distance traveled before the reference vectors are recalculated from the magnetic and gravity models.
8	Year	float	yr	0	The reference date expressed as a decimal year.
12	–PAD–	uint8[4]	–	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
16	Latitude	double	deg	0	Reference latitude.
24	Longitude	double	deg	0	Reference longitude.
32	Altitude	double	m	0	Reference altitude above the reference ellipsoid.

This register allows configuration of the onboard spherical harmonic models used to calculate the local magnetic and gravitational reference values. Having accurate magnetic reference values improves the accuracy of heading when using the magnetometer and accounts for magnetic declination. Having accurate gravitational reference values improves accuracy by allowing the INS filter to more accurately estimate the accelerometer biases. The sensor currently includes the EGM96 gravitational model and the World Magnetic Model. The models are upgradable to allow updating to future models when available.

When each model is enabled, the sensor will automatically update the respective reference vectors in Register 21 using the last valid GNSS position and time estimate. If there has not been a valid GNSS 3D fix since startup and the values entered in this register are non-default, the Magnetic and Gravity Reference Vectors register (Register 21) vectors will be updated with the values entered in this register. When a model is disabled, the respective vector in Register 21 reverts back to its latest user-entered values. The vector values will be recalculated whenever the current position has changed by the *RecalcThreshold* or the date has changed by more than approximately 8 hours, whichever comes first.

Enumeration: EnableMagModel

NAME	VALUE	DESCRIPTION
Disabled	0	World magnetic model disabled.
Enabled	1	World magnetic model enabled.

TABLE 3.89

Enumeration: EnableGravityModel

NAME	VALUE	DESCRIPTION
Disabled	0	World gravity model disabled.
Enabled	1	World gravity model enabled.

TABLE 3.90

3.10 HEAVE

3.10.1 Heave Basic Configuration

Register ID . . . 116
Size (Bytes) . . . 28
Description . . . Provides basic control over the heave estimation alogorithm.
Example Read Response . . . `$VNRRG,116,0,0.05,20,0.05,0.08,0,0*4D`
Example Write Command . . . `$VNWRG,116,5,0.05,20,0.05,0.08,0,0*XX`

OFFSET	NAME	FORMAT	UNIT	DEFAULT	DESCRIPTION
0	InitialWave Period	float	s	0	Initial value of wave period used by the real-time heave filter. Heave filter is disabled when set to 0. Recommended value is 5 s.
4	InitialWave Amplitude	float	m	0.05	Initial value of wave amplitude used by the real-time heave filter.
8	MaxWave Period	float	s	20	Maximum wave period used by the real-time heave filter.
12	MinWave Amplitude	float	m	0.05	Minimum wave amplitude used by the real-time heave filter.
16	DelayedHeave CutoffFreq	float	rad/s	0.08	Constant delayed heave cutoff frequency.
20	HeaveCutoff Freq	float	rad/s	0	Constant real-time heave cutoff frequency. If non-zero, this value is used instead of the adaptive cutoff frequency.
24	HeaveRate CutoffFreq	float	rad/s	0	Constant real-time heave rate cutoff frequency. If non-zero, this value is used instead of the adaptive cutoff frequency.

4 MEASUREMENT REGISTERS

4.1 OVERVIEW

Measurement registers are defined as those that are read-only through the Read RegisterCommand. Most contain combinations of measurements consistent with those defined in the binary output messages (see Section 2), but some contain device information or device status that is unavailable in the binary outputs.

4.2 SYSTEM

4.2.1 Model

Register ID 1
Size (Bytes) 24
Description . . . . Product model string.
Example Read Response . . . . `$VNRRG,01,VN-200T-DEV*77`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Model	string[24]	–	Product model number, maximum length 24 characters.

4.2.2 Hardware Version

Register ID 2
Size (Bytes) 4
Description . . . . Hardware version number.
Example Read Response . . . . `$VNRRG,02,2*6F`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	HwVer	uint32	–	Hardware version number.

4.2.3 Serial Number

Register ID 3
Size (Bytes) 4
Description . . . . . Device serial number.
Example Read Response . . . . . `$VNRRG,03,0100027769*50`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	SerialNum	uint32	–	The unit's serial number.

4.2.4 Firmware Version

Register ID 4
Size (Bytes) 20
Description . . . . . Firmware version number.
Example Read Response . . . . . `$VNRRG,04,2.0.0.1*76`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	FwVer	string[20]	–	Firmware version.

4.2.5 Synchronization Status

Register ID . . . 33
Size (Bytes) . . . 12
Description . . . Contains counters based on the SyncIn and SyncOut events.
Example Read Response . . . `$VNRRG,33,0,116358757,205*65`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	SyncInCount	uint32	–	Amount of SyncIn Events that have occured.
4	SyncInTime	uint32	μs	The amount of time that has elapsed since the last SyncIn Event.
8	SyncOutCount	uint32	–	Keeps track of the number of times that a SyncOut pulse has occurred.

Writing zero to the *SyncInCount* or the *SyncOutCount* will reset their counter. Any value other than zero will not have an effect. The *SyncInTime* is read only and can only be reset by a SyncIn Event.

When a SyncIn Event and SyncOut pulse occur are determined by the Synchronization Control register (Register 32).

SyncInCount
This field is helpful to correlate a sensor reading to an event on an external system, such as a camera. It can be appended to each asynchronous ASCII or SPI data packet using the Communication Protocol Control register (Register 30).

SyncInTime
This field is helpful to correlate a sensor reading to an event on an external system, such as a camera. It can be appended to each asynchronous ASCII or SPI data packet using the Communication Protocol Control register (Register 30).

SyncOutCount
This field is helpful to index subsequent measurement outputs, which is particularly useful when logging sensor data. It can be appended to each asynchronous ASCII or SPI data packet using the Communication Protocol Control register (Register 30).

4.3 ATTITUDE

4.3.1 Yaw Pitch Roll

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the YPR header.

Register ID 8
Size (Bytes) 12
Description . . . . Attitude solution as yaw, pitch, and roll in degrees. The yaw, pitch, and roll is given as a 3,2,1 Euler angle rotation sequence describing the orientation of the sensor with respect to the inertial North East Down (NED) frame.
Example Read Response . . . . `$VNRRG,08,-122.856,+021.520,-005.127*5D`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.

4.3.2 Quaternion

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the QTN header.

Register ID 9
Size (Bytes) 16
Description . . . . Attitude solution as a quaternion.
Example Read Response . . . . `$VNRRG,09,-0.153574,-0.131056,+0.852509,-0.482150*74`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	QuatX	float	–	First vector component of quaternion.
4	QuatY	float	–	Second vector component of quaternion.
8	QuatZ	float	–	Third vector component of quaternion.
12	QuatS	float	–	Scalar component of quaternion.

4.3.3 Quaternion & Compensated IMU

> **ASCII Async Register** This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the QMR header.

Register ID 15
Size (Bytes) 52
Description .... Quaternion attitude solution, and compensated (Magnetic, Acceleration, Angular Rate) values.
Example Read Response .... `$VNRRG,15,-0.164310,-0.133652,+0.846798,-0.487927,-00.0751,+00.0889,+00.2273,-00.291,+00.121,-09.660,-00.000734,+00.008341,+00.006674*60`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	QuatX	float	–	First vector component of quaternion.
4	QuatY	float	–	Second vector component of quaternion.
8	QuatZ	float	–	Third vector component of quaternion.
12	QuatS	float	–	Scalar component of quaternion.
16	MagX	float	G	Compensated magnetometer measurement in the body-frame x-axis.
20	MagY	float	G	Compensated magnetometer measurement in the body-frame y-axis.
24	MagZ	float	G	Compensated magnetometer measurement in the body-frame z-axis.
28	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
32	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
36	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.
40	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
44	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
48	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.3.4 Yaw-Pitch-Roll & Compensated IMU

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the YMR header.

Register ID 27
Size (Bytes) 48
Description . . . . . Yaw, Pitch, Roll, Accel, and Angular Rates
Example Read Response . . . . `$VNRRG,27,-114.428,+028.727,-000.386,-00.0763,+00.0857,+00.2308,-00.296,+00.106,-09.645,-00.000857,+00.009152,+00.006914*43`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.
12	MagX	float	G	Compensated magnetometer measurement in the body-frame x-axis.
16	MagY	float	G	Compensated magnetometer measurement in the body-frame y-axis.
20	MagZ	float	G	Compensated magnetometer measurement in the body-frame z-axis.
24	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
28	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
32	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.
36	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
40	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
44	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.3.5 Yaw-Pitch-Roll, Linear Acceleration & Gyro

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the YBA header.

Register ID 239
Size (Bytes) 36
Description . . . . Yaw, Pitch, Roll, Linear Body Accel, and Angular Rates.
Example Read Response . . . . `$VNRRG,239,-100.469,+033.473,+002.072,-05.692,+00.387,-01.492,-00.001860,+00.007604,+00.005330*5A`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.
12	LinAccelX	float	m/s^2	Linear acceleration in body-frame x-axis.
16	LinAccelY	float	m/s^2	Linear acceleration in body-frame y-axis.
20	LinAccelZ	float	m/s^2	Linear acceleration in body-frame z-axis.
24	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
28	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
32	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

This register contains the true measured acceleration. The accelerometer measures both acceleration and the effect of static gravity in the body-frame. This register contains the true acceleration which does not contain gravity and should measure 0 when the device is stationary.

The current attitude solution is used to map the gravity reference vector from the inertial (NED) frame into the body frame and remove the gravity component from the acceleration measurement. This measurement is also compensated by the factory calibration stored in flash, the user-applied calibration, and the real-time accelerometer bias estimated by the attitude filter. If the device is stationary and the attitude filter is tracking, the measurement should nominally be equivalent to zero in all three axes.

4.3.6 Yaw-Pitch-Roll, Inertial Linear Acceleration & Gyro

> **ASCII Async Register** This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the YIA header.

Register ID 240
Size (Bytes) 36
Description . . . . Yaw, Pitch, Roll, Linear Inertial Accel, and Angular Rates.
Example Read Response . . . . `$VNRRG,240,-099.493,+033.481,+002.182,+01.324,+05.408,+01.935,+00.000212,+00.005921,+00.007223*59`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.
12	LinAccelN	float	m/s^2	Linear acceleration in North direction.
16	LinAccelE	float	m/s^2	Linear acceleration in East direction.
20	LinAccelD	float	m/s^2	Linear acceleration in Down direction.
24	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
28	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
32	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

> This register contains the true measured acceleration. The accelerometer measures both acceleration and the effect of static gravity in the body-frame. This register contains the true acceleration which does not contain gravity and should measure 0 when the device is stationary. The true acceleration provided in this register is measured in the inertial frame. This means that an up/down movement will always appear as an acceleration in the z-axis on this register regardless of the orientation of the sensor.

The current attitude solution is used to map the acceleration measurement from the body frame into the inertial (NED) frame. The gravity component is then removed from the acceleration measurement using the current gravity reference vector. This acceleration measurement is also compensated by the factory calibration stored in flash, the user-applied calibration, and the real-time accelerometer bias estimated by the attitude filter. If the device is stationary and the attitude filter is tracking, the measurement should nominally be equivalent to zero in all three axes.

4.4 IMU

4.4.1 Compensated Magnetometer

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the MAG header.

Register ID 17
Size (Bytes) 12
Description . . . . . . Compensated magnetometer measurements.
Example Read Response . . . . . . `$VNRRG,17,-00.0751,+00.0878,+00.2297*50`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	MagX	float	G	Compensated magnetometer measurement in the body-frame x-axis.
4	MagY	float	G	Compensated magnetometer measurement in the body-frame y-axis.
8	MagZ	float	G	Compensated magnetometer measurement in the body-frame z-axis.

4.4.2 Compensated Accelerometer

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the ACC header.

Register ID 18
Size (Bytes) 12
Description . . . . Compensated acceleration measurements
Example Read Response . . . . `$VNRRG,18,-00.300,+00.096,-09.604*64`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
4	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
8	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.

4.4.3 Compensated Gyro

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the GYR header.

Register ID 19
Size (Bytes) 12
Description . . . . . . Compensated angular rate measurements.
Example Read Response . . . . . . `$VNRRG,19,-00.000337,+00.010093,+00.008537*51`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
4	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
8	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.4.4 Compensated IMU

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the MAR header.

Register ID 20
Size (Bytes) 36
Description .... Compensated magnetic, acceleration, and angular rate measurements.
Example Read Response .... `$VNRRG,20,-00.0763,+00.0878,+00.2285,-00.307,+00.081,-09.627,+00.000268,+00.009487,+00.007432*6D`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	MagX	float	G	Compensated magnetometer measurement in the body-frame x-axis.
4	MagY	float	G	Compensated magnetometer measurement in the body-frame y-axis.
8	MagZ	float	G	Compensated magnetometer measurement in the body-frame z-axis.
12	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
16	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
20	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.
24	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
28	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
32	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.4.5 IMU Measurements

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the IMU header.

Register ID 54
Size (Bytes) 44
Description . . . . Provides the calibrated IMU measurements including barometric pressure.
Example Read Response . . . . `$VNRRG,54,-00.0774,+00.0900,+00.2250,-00.303,+00.090,-09.615,+00.001916,-00.005746,-00.030280,+30.6,+099.545*50`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	UncompMagX	float	G	Magnetometer body-frame x-axis.
4	UncompMagY	float	G	Magnetometer body-frame y-axis.
8	UncompMagZ	float	G	Magnetometer body-frame z-axis.
12	UncompAccX	float	m/s^2	Accelerometer body-frame x-axis.
16	UncompAccY	float	m/s^2	Accelerometer body-frame y-axis.
20	UncompAccZ	float	m/s^2	Accelerometer body-frame z-axis.
24	UncompGyroX	float	rad/s	Angular rate body-frame x-axis.
28	UncompGyroY	float	rad/s	Angular rate body-frame y-axis.
32	UncompGyroZ	float	rad/s	Angular rate body-frame z-axis.
36	Temperature	float	°C	Sensor temperature.
40	Pressure	float	kPa	Barometric pressure.

4.4.6 Delta Theta and Delta Velocity

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the DTV header.

Register ID	80
Size (Bytes)	28
Description	This register contains the output values of the onboard coning and sculling algorithm.
Example Read Response	`$VNRRG,80,+65.767502,+006.503,-006.963,-121.662,+043.425, +001.734,-634.325*5C`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	DeltaTime	float	s	Duration of integration interval.
4	DeltaThetaX	float	deg	Integrated rotation vector x-axis.
8	DeltaThetaY	float	deg	Integrated rotation vector y-axis.
12	DeltaThetaZ	float	deg	Integrated rotation vector z-axis.
16	DeltaVelX	float	m/s	Integrated velocity x-axis.
20	DeltaVelY	float	m/s	Integrated velocity y-axis.
24	DeltaVelZ	float	m/s	Integrated velocity z-axis.

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

The coning and sculling algorithm can be configured to operate in multiple frames and with a variety of compensations applied. See the Delta Theta and Delta Velocity Configuration register (Register 82) for further details.

Delta Time

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

Delta Theta

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

Delta Velocity

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

4.5 HARD/SOFT IRON ESTIMATOR

4.5.1 Real-Time HSI Results

Register ID 47
Size (Bytes) 48
Description . . . . . Magnetometer calibration values calculated by the real-time HSI calibration filter.
Example Read Response . . . . . `$VNRRG,47,1,0,0,0,1,0,0,0,1,0,0,0*71`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Gain	float[9]	–	The gain values from the real-time calibration in row-major order.
36	Bias	float[3]	–	The bias values from the real-time calibration.

This register contains twelve values representing the calculated hard and soft iron compensation parameters from the real-time HSI calibration filter. When this compensation is enabled via Register 44, the magnetic measurements are compensated for both hard and soft iron using the following model.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} * \begin{Bmatrix} MX - B0 \\ MY - B1 \\ MZ - B2 \end{Bmatrix} \tag{4.1}$$

The variables {MX,MY,MZ} are components of the measured magnetic field after the application of the factory calibration and any offline HSI calibration stored in Register 23. The {X, Y, Z} variables are the new magnetic field measurements outputted after compensation for hard/soft iron effects.

4.6 GNSS

4.6.1 GNSS Solution - LLA

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the GPS header.

Register ID 58
Size (Bytes) 72
Description . . . . Primary GNSS receiver measurement with lat/lon/alt position and velocity in NED frame.
Example Read Response . . . . `$VNRRG,58,000122.000000,0000,0,00,+00.00000000, +000.00000000,+00000.000,+000.000,+000.000,+000.000, +25769804.000,+25769804.000,+22500032.000,+079.584, 4.29E+00*31`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	double	s	GPS time of week.
8	GpsWeek	uint16	–	GPS week.
10	GnssFix	uint8	ENUM	Type of GNSS fix.
11	NumSats	uint8	–	Number of satellites tracked by GNSS receiver.
12	–PAD–	uint8[4]	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
16	Lat	double	deg	GNSS geodetic latitude.
24	Lon	double	deg	GNSS longitude.
32	Alt	double	m	GNSS altitude above WGS84 ellipsoid.
40	VelN	float	m/s	GNSS velocity in North direction.
44	VelE	float	m/s	GNSS velocity in East direction.
48	VelD	float	m/s	GNSS velocity in Down direction.
52	PosUncertainty N	float	m	GNSS position uncertainty, North direction.
56	PosUncertainty E	float	m	GNSS position uncertainty, East direction.
60	PosUncertainty D	float	m	GNSS position uncertainty, Down direction.
64	GnssVel Uncertainty	float	m/s	GNSS velocity uncertainty (scalar).
68	GnssTime Uncertainty	float	s	GNSS time uncertainty.

This register provides the position, velocity, and time (PVT) solution from the primary GNSS receiver (GNSS1) used by the INS Kalman filter for position and velocity inputs.

Enumeration: GnssFix

NAME	VALUE	DESCRIPTION
NoFix	0	Not tracking.
TimeFix	1	Time-only fix.
Fix2D	2	Time plus 2D position fix.
Fix3D	3	Time plus full 3D position fix.
SBAS	4	3D Fix using SBAS corrections.
RtkFloat	7	3D Fix using an approximate RTK solution.
RtkFix	8	3D Fix using a fixed RTK solution.

TABLE 4.20

4.6.2 GNSS Solution - ECEF

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the GPE header.

Register ID 59
Size (Bytes) 72
Description . . . . . . Primary GNSS receiver measurement in ECEF frame.
Example Read Response . . . . `$VNRRG,59,000125.400000,0000,0,00,+6378137.000,+0000000.000,+0000000.000,+000.000,+000.000,+000.000,+25769804.000,+25769804.000,+22500034.000,+080.672,4.29E+00*06`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	double	s	GPS time of week.
8	GpsWeek	uint16	–	GPS week.
10	GnssFix	uint8	ENUM	Type of GNSS fix.
11	NumSats	uint8	–	Number of satellites tracked by GNSS receiver.
12	–PAD–	uint8[4]	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
16	PosX	double	m	GNSS position in ECEF-frame x-axis.
24	PosY	double	m	GNSS position in ECEF-frame y-axis.
32	PosZ	double	m	GNSS position in ECEF-frame z-axis.
40	VelX	float	m/s	GNSS velocity in ECEF-frame x-axis.
44	VelY	float	m/s	GNSS velocity in ECEF-frame y-axis.
48	VelZ	float	m/s	GNSS velocity in ECEF-frame z-axis.
52	PosUncertainty X	float	m	GNSS position uncertainty ECEF X.
56	PosUncertainty Y	float	m	GNSS position uncertainty ECEF Y.
60	PosUncertainty Z	float	m	GNSS position uncertainty ECEF Z.
64	GnssVel Uncertainty	float	m/s	GNSS velocity uncertainty (scalar).
68	GnssTime Uncertainty	float	s	GNSS time uncertainty.

This register provides the position, velocity, and time (PVT) solution in the ECEF frame from the primary GNSS receiver (GNSS1) used by the INS Kalman filter for position and velocity inputs.

Enumeration: GnssFix

NAME	VALUE	DESCRIPTION
NoFix	0	Not tracking.
TimeFix	1	Time-only fix.
Fix2D	2	Time plus 2D position fix.
Fix3D	3	Time plus full 3D position fix.
SBAS	4	3D Fix using SBAS corrections.
RtkFloat	7	3D Fix using an approximate RTK solution.
RtkFix	8	3D Fix using a fixed RTK solution.

TABLE 4.22

4.7 INS

4.7.1 INS Solution - LLA

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the INS header.

Register ID 63
Size (Bytes) 72
Description . . . . Estimated INS solution with lat/lon/alt position.
Example Read Response . . . . `$VNRRG,63,000129.373757,0000,0080,-107.709,+031.740,+000.823,+00.00000000,+000.00000000,+00000.000,+000.000,+000.000,+000.000,99.99,00.01,0.001*69`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	double	s	GPS time of week.
8	GpsWeek	uint16	–	The current GPS week.
10	InsStatus	uint16	BITS	Ins status bitfield.
12	Yaw	float	deg	Yaw angle.
16	Pitch	float	deg	Pitch angle.
20	Roll	float	deg	Roll angle.
24	PosLat	double	deg	Geodetic latitude.
32	PosLon	double	deg	Longitude.
40	PosAlt	double	m	Altitude above WGS84 ellipsoid.
48	VelN	float	m/s	Velocity in North direction.
52	VelE	float	m/s	Velocity in East direction.
56	VelD	float	m/s	Velocity in Down direction.
60	AttUncertainty	float	deg	Filter estimated attitude uncertainty.
64	PosUncertainty	float	m	Filter estimated position uncertainty.
68	VelUncertainty	float	m/s	Filter estimated velocity uncertainty.

Ins Status

Heading Data Source

- Mode 0: The heading is entirely magnetometer-derived.
- Mode 1: The heading is either entirely magnetometer-derived, or is in the process of switching from magnetometer-derived to GNSS-derived.
- Mode 2: The heading is entirely GNSS-derived. The magnetometer is entirely ignored.

Bit Field: InsStatus

NAME	OFFSET	DESCRIPTION
Mode	0	Two-bit enumeration that indicates the current mode of the INS filter.
GnssFix	2	Indicates whether the GNSS has a valid fix.
Resv1	3	Reserved.
ImuErr	4	High if gyro or accelerometer subsystem error is detected.
MagPresErr	5	High if magnetometer or pressure subsystem error is detected.
GnssErr	6	High if GNSS communication error is detected or if no valid PPS signal is received.
Resv2	7	Reserved for internal use. May toggle state during runtime and should be ignored.
Resv3	8	Reserved.

TABLE 4.24

Enumeration: Mode

NAME	VALUE	DESCRIPTION
NotTracking	0	The INS filter is non-operational and outputs are invalid. Attitude outputs are provided by the AHRS filter, if AHRS Aiding is enabled in Register 67.
Aligning	1	The INS filter has initialized. Position and velocity outputs are valid, but attitude outputs are not.
Tracking	2	The INS filter is tracking and all outputs are valid.
GnssLost	3	An extended GNSS outage has occurred. Position and velocity outputs are invalid, but the attitude outputs remain valid.

TABLE 4.25

4.7.2 INS Solution - ECEF

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the INE header.

Register ID . . . 64
Size (Bytes) . . . 72
Description . . . Estimated INS Solution with ECEF position
Example Read Response . . . `$VNRRG,64,000132.681257,0000,8080,-106.347,+032.188, +001.047,+0000000.000,+0000000.000,+0000000.000,+000.000, +000.000,+000.000,180.00,00.00,0.000*6D`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	GpsTow	double	s	GPS time of week.
8	GpsWeek	uint16	–	The current GPS week.
10	InsStatus	uint16	BITS	Ins status bitfield.
12	Yaw	float	deg	Yaw angle.
16	Pitch	float	deg	Pitch angle.
20	Roll	float	deg	Roll angle.
24	PosEX	double	m	Position in ECEF-frame x-axis.
32	PosEY	double	m	Position in ECEF-frame y-axis.
40	PosEZ	double	m	Position in ECEF-frame z-axis.
48	VelEX	float	m/s	Velocity in ECEF-frame x-axis.
52	VelEY	float	m/s	Velocity in ECEF-frame y-axis.
56	VelEZ	float	m/s	Velocity in ECEF-frame z-axis.
60	AttUncertainty	float	deg	Filter estimated attitude uncertainty.
64	PosUncertainty	float	m	Filter estimated position uncertainty.
68	VelUncertainty	float	m/s	Filter estimated velocity uncertainty.

Ins Status

Heading Data Source

- Mode 0: The heading is entirely magnetometer-derived.
- Mode 1: The heading is either entirely magnetometer-derived, or is in the process of switching from magnetometer-derived to GNSS-derived.
- Mode 2: The heading is entirely GNSS-derived. The magnetometer is entirely ignored.

Bit Field: InsStatus

NAME	OFFSET	DESCRIPTION
Mode	0	Two-bit enumeration that indicates the current mode of the INS filter.
GnssFix	2	Indicates whether the GNSS has a valid fix.
Resv1	3	Reserved.
ImuErr	4	High if gyro or accelerometer subsystem error is detected.
MagPresErr	5	High if magnetometer or pressure subsystem error is detected.
GnssErr	6	High if GNSS communication error is detected or if no valid PPS signal is received.
Resv2	7	Reserved for internal use. May toggle state during runtime and should be ignored.
Resv3	8	Reserved.

TABLE 4.27

Enumeration: Mode

NAME	VALUE	DESCRIPTION
NotTracking	0	The INS filter is non-operational and outputs are invalid. Attitude outputs are provided by the AHRS filter, if AHRS Aiding is enabled in Register 67.
Aligning	1	The INS filter has initialized. Position and velocity outputs are valid, but attitude outputs are not.
Tracking	2	The INS filter is tracking and all outputs are valid.
GnssLost	3	An extended GNSS outage has occurred. Position and velocity outputs are invalid, but the attitude outputs remain valid.

TABLE 4.28

4.7.3 INS State - LLA

> **ASCII Async Register** This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the ISL header.

Register ID 72
Size (Bytes) 76
Description . . . . . Estimated INS state with lat/lon/alt position.
Example Read Response . . . . `$VNRRG,72,-103.913,+032.745,+001.402,+00.00000000,+000.00000000,+00000.000,+000.000,+000.000,+000.000,-00.310,+00.098,-09.640,-00.001128,+00.008452,+00.007214*5F`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.
12	–PAD–	uint8[4]	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
16	PosLat	double	deg	Geodetic latitude.
24	PosLon	double	deg	Longitude.
32	PosAlt	double	m	Altitude above WGS84 ellipsoid.
40	VelN	float	m/s	Velocity in North direction.
44	VelE	float	m/s	Velocity in East direction.
48	VelD	float	m/s	Velocity in Down direction.
52	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
56	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
60	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.
64	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
68	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
72	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.7.4 INS State - ECEF

> **ASCII Async Register** This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the ISE header.

Register ID 73
Size (Bytes) 76
Description . . . . Estimated INS state with ECEF position.
Example Read Response . . . . `$VNRRG,73,-102.549,+033.011,+001.622,+00.00000000, +000.00000000,+00000.000,+000.000,+000.000,+000.000,-00.318,+00.087,-09.639,+00.000233,+00.003212,+00.007074*50`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Yaw	float	deg	Yaw angle.
4	Pitch	float	deg	Pitch angle.
8	Roll	float	deg	Roll angle.
12	−PAD−	uint8[4]	–	Padding bytes used for SPI only. Does not appear as a field in the UART serial protocol.
16	PosEX	double	m	Position in ECEF-frame x-axis.
24	PosEY	double	m	Position in ECEF-frame y-axis.
32	PosEZ	double	m	Position in ECEF-frame z-axis.
40	VelEX	float	m/s	Velocity in ECEF-frame x-axis.
44	VelEY	float	m/s	Velocity in ECEF-frame y-axis.
48	VelEZ	float	m/s	Velocity in ECEF-frame z-axis.
52	AccelX	float	m/s^2	Compensated accelerometer measurement in the body-frame x-axis.
56	AccelY	float	m/s^2	Compensated accelerometer measurement in the body-frame y-axis.
60	AccelZ	float	m/s^2	Compensated accelerometer measurement in the body-frame z-axis.
64	GyroX	float	rad/s	Compensated angular rate measurement in the body-frame x-axis.
68	GyroY	float	rad/s	Compensated angular rate measurement in the body-frame y-axis.
72	GyroZ	float	rad/s	Compensated angular rate measurement in the body-frame z-axis.

4.8 HEAVE

4.8.1 Heave and Heave Rate

ASCII Async Register This register can be output automatically at a fixed rate through appropriate configuration of the Async Data Output Type Register (Register 6). When configured as Async Data, this register will be sent out with the HVE header.

Register ID 115
Size (Bytes) 12
Description . . . . Real-time heave and heave-rate estimates, plus a delayed-heave estimate.
Example Read Response . . . . `$VNRRG,115,+00.000,+00.000,+00.000*5F`

OFFSET	NAME	FORMAT	UNIT	DESCRIPTION
0	Heave	float	m	Real-time heave estimate.
4	HeaveRate	float	m/s	Real-time heave rate estimate.
8	DelayedHeave	float	m	Delayed heave. Higher accuracy than real-time heave, but 50 seconds old.

A EXAMPLES

A.1 BINARY OUTPUT MESSAGES

The sections that follow provide some examples of configuring and parsing binary output messages.

A.1.1 Configuration

Binary output messages are configured using Register 75 – Register 77. As an example, assume you want to configure the following: a 40 Hz output on UART-1 consisting of TimeStartup and Ypr from the Common group and YprU from the Attitude group.

Per the definition of Register 75, the first field denotes the port over which to output the message; in this case a value of 1. The second field denotes the *RateDivisor* which sets the output rate as a function of the sensor ImuRate (800 Hz for the VN-200). So to achieve a 40 Hz output rate, the *RateDivisor* is set to 20.

The next fields in the configuration register select the desired outputs through the 8-bit *OutputGroup* byte and 16-bit *OutputType* word. The bit offsets for the Common group and Attitude group can be found in Table 2.1, while the bit offsets for the particular Binary Types for those groups are found in Tables 2.3 and 2.54. Figure A.1 shows how those bit offsets map to the hex representations needed to complete the command. Keep in mind that VectorNav sensors use little-endian format, so the bytes in the 16-bit *OutputType* word are ordered with the low byte first when output as part of the binary output message itself.

- Configuration: `$VNWRG,75,1,20,11,0009,0100*63`
- Output Message Bytes: `FA 11 09 00 00 01 ...`

See Appendix B.1.1 for details on configuring messages that contain binary output groups or types beyond those that can be defined by a single byte for groups or a single 16-bit word for types.

Binary Group Byte and Type Words Example

	Group Byte							
Offset	7	6	5	4	3	2	1	0
Binary	0	0	0	1	0	0	0	1
Hex	0x11							

	Type Word 1															
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Hex	0x00								0x09							

	Type Word 2															
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Hex	0x01								0x00							

FIGURE A.1

A.1.2 Output Messages

As described in Section 2.1.3, binary output messages consist of a sync byte (`0xFA`), header bytes (group and type bytes), a payload, and a 16-bit CRC checksum. The header bytes are identical to those used when configuring the message in Register 75 – Register 77. The number of payload bytes depends on the selected Binary Types. See Appendix B.1.2 for details on parsing binary output messages that exceed the internal buffer size of the sensor (600 bytes) and are split into multiple packets prior to outputting.

Single Binary Group

In the first example, consider outputting only Ypr (Yaw-Pitch-Roll) from the Common group. An example of the output message you might expect is given in Figure A.2. Since only one group is being used, there is only one 16-bit type word in the header. The payload consists of the three floats that make up Ypr. Note that the type word and floats are all output in little-endian format. The final 16-bit checksum, however, is output big-endian, such that if you compute the CRC across the entire output message (excluding the sync byte, but including the checksum), the resultant CRC evaluates to `0x0000`.

Single Binary Group Output Example

	Sync	Header			Payload												Checksum	
		Group	Types		Common Group: YawPitchRoll													
			Common		Yaw				Pitch				Roll					
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Raw Byte	0xFA	0x01	0x08	0x00	0x93	0x50	0x2E	0x42	0x83	0x3E	0xF1	0x3F	0x48	0xB5	0x04	0xBB	0x92	0x88
Type	uint8	uint8	uint16		float				float				float				uint16	
Value	0xFA	0x01	0x0008		+43.578686				+1.884720				-0.002025				0x9288	

FIGURE A.2

Multiple Binary Groups

In the second example, consider outputting Temperature from the IMU group in addition to Ypr from the Common group. Since there are now two active groups in the message, there are two 16-bit type words in the header. In the payload, the enabled outputs of a given group our output together in the order of their bit offsets. The example output message is given in Figure A.3.

Multiple Binary Group Output Example

	Sync	Header					Payload											
			Types		Types		Common Group: YawPitchRoll											
		Group	Common		IMU		Yaw				Pitch				Roll			
Byte Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Raw Byte	0xFA	0x05	0x08	0x00	0x10	0x00	0x42	0x8E	0xE7	0xC2	0x1E	0x12	0x11	0xC1	0xFF	0x49	0x9C	0x40
Type	uint8	uint8	uint16		uint16		float				float				float			
Value	0xFA	0x05	0x0008		0x0010		-115.777853				-9.066923				+4.884033			

	Payload				Checksum	
	IMU Group: Temp					
	Temperature					
Byte Offset	18	19	20	21	22	23
Raw Byte	0xE3	0x27	0xC4	0x41	0x6A	0x4E
Type	float				uint16	
Value	+24.519476				0x6A4E	

FIGURE A.3

Variable Length Types

Certain output types in the GNSS groups have a variable length payload, depending on the number of satellites or signals the GNSS receiver is tracking. When one of those Binary Types is enabled, it is impossible to pre-compute the length of the output message, and instead the message must be partially parsed to identify the total message length and then compute the checksum and parse out the payload.

For instance, consider enabling GpsTow, GpsWeek, GnssPosLla, GnssVelNed, and GnssSatInfo from the GNSS group. The output message then consists of the sync byte (`0xFA`), followed by the group byte (`0x08`) and 16-bit type word

`0xA640`. These are followed by the 46 bytes that make up the fixed length GpsTow, GpsWeek, GnssPosLla, and GnssVelNed. As defined in Section 2.5.15, the next byte is *Count*, followed by reserved byte. Parsing *Count* provides the payload length for GnssSatInfo in that particular message of *Count* $\times 8 + 2$.

A.2 SPI COMMUNICATION

The sections that follow provided some example SPI transactions for the various types of commands available on the VN-200.

A.2.1 Read Register Example

Figure A.4 shows an example of a single transaction with the VN-200 to read Register 5 (serial baud rate). The first packet is the request packet and consists of the master sending out the MOSI line a four byte header with no payload. The first byte in the header has the command ID of 1, which corresponds to a Read Register Command. The second byte is the argument. In the case of the Read Register Command, this corresponds to the register ID, which in this case is Register 5. The next two bytes are always zero in the header.

After this packet is sent the master should raise the chip select line (SPI_CS) and wait at least 100 microseconds before issuing the respond packet. During this time the VN-200 will process the read register request and place the requested data in its SPI output buffer. During the response packet, the master should clock in N bytes of zeroes on the MOSI line, where N is equal to 4 plus the size of the register being read, which in this example is Register 5 (4 bytes).

The header for packets being received from the VN-200 has a different structure with the first byte always being zero. The second and third byte in the header is the command ID and the argument (register ID) of the response. The fourth byte in the header is the error code. If an error occurred while attempting to service the request the VN-200 will issue a non-zero error code in this byte with no payload. In the payload of the response packet the four bytes received correspond to the value of Register 5 which in this case is 115200. As you can see from the example multi-byte values are sent in little endian format with the least significant byte sent first (0x01C200 = 115200).

SPI Read Example: Register 5 (Serial Baud Rate)

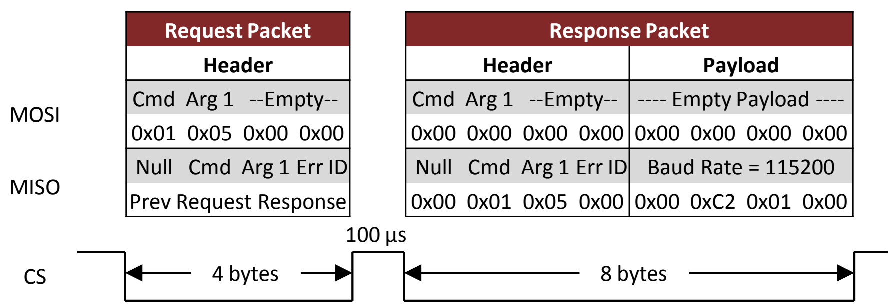

FIGURE A.4

A.2.2 Write Register Example

Figure A.5 shows an example of a write register transaction. In this example the baud rate is being set to 921600 baud in Register 5. In the case of writing to a register, the values to be loaded into the register are in the payload of the request packet. The payload of the response packet contains the contents of the register after the write register command has been processed. In the case that no error occurred the payload of the response packet should be the same as the request. Conveniently, it is sufficient to clock in only four bytes on the response packet to verify that the write register took effect, which is indicated by a zero error code.

A.2.3 Read Register Example - Floats

Figure A.6 shows a transaction involving reading a register with floating point values. In this case Register 8 is read which contains the sensor attitude (*Yaw*, *Pitch* & *Roll*). The floating point values are stored as 32-bit IEEE floating point numbers in little endian byte order.

A.2.4 Transaction Error Example

Figure A.7 demonstrates what will happen when an error occurs during a transaction. In this case, the user attempted to write to a read-only register. The fourth byte of the response packet header shows an Error ID of 8 was returned, which corresponds to an Invalid Register. The different error codes are listed in Table 1.6.

SPI Write Example: Register 5 (Serial Baud Rate)

	Request Packet		Response Packet	
	Header	Payload	Header	Payload
MOSI	Cmd Arg 1 --Empty--	Baud Rate = 921600	Cmd Arg 1 --Empty--	---- Empty Payload ----
	0x02 0x05 0x00 0x00	0x00 0x10 0x0E 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00
MISO	Null Cmd Arg 1 Err ID	Prev Packet Payload	Null Cmd Arg 1 Err ID	Baud Rate = 921600
	Prev Request Response	Prev Request Response	0x00 0x02 0x05 0x00	0x00 0x10 0x0E 0x00

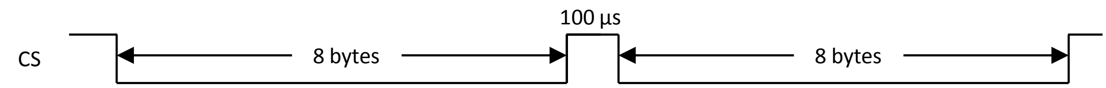

FIGURE A.5

SPI Read Example: Register 8 ((Yaw, Pitch & Roll))

	Request Packet	Response Packet			
	Header	Header	Payload		
MOSI	Cmd Arg 1 --Empty--	Cmd Arg 1 --Empty--	---- Empty Payload ----	---- Empty Payload ----	---- Empty Payload ----
	0x01 0x08 0x00 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00
MISO	Null Cmd Arg 1 Err ID	Null Cmd Arg 1 Err ID	Yaw = -161.69768	Pitch = +1.2750288	Roll = +0.60307616
	Prev Request Response	0x00 0x01 0x08 0x00	0x9B 0xB2 0x21 0xC3	0x25 0x34 0xA3 0x3F	0x33 0x63 0x1A 0x3F

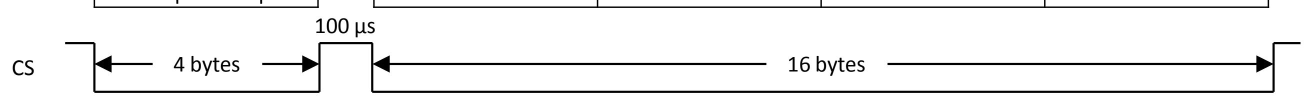

FIGURE A.6

SPI Error Example: Invalid Register

	Request Packet				Response Packet
	Header	Payload			Header
MOSI	Cmd Arg 1 --Empty--	MagX = +1.0	MagY = +2.0	MagZ = +3.0	Cmd Arg 1 --Empty--
	0x02 0x12 0x00 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00	0x00 0x00 0x00 0x00	0x02 0x12 0x00 0x00
MISO	Null Cmd Arg 1 Err ID	Previous Packet Payload			Null Cmd Arg 1 Err ID
	Prev Request Response	Empty or Previous Request Response Payload			0x00 0x02 0x12 0x08

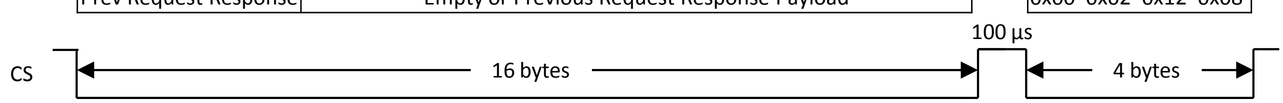

FIGURE A.7

B ADVANCED COMMUNICATIONS

B.1 BINARY EXTENSIONS

Two extensions to the standard binary output messages described in Section 2 have been implemented to provide additional data collection capabilities. First, a method for extending the group and type bytes to allow for more than groups than can be defined by a single byte and more types per group than can be defined by a single word has been developed. Second, a means for splitting up large binary output messages into multiple smaller packets has been defined.

B.1.1 Extended Group Bytes and Type Words

The method for extending the number of binary groups and binary types beyond the limits of the bitfields defined by the standard protocol is designed to be backwards compatible with the standard definition and to minimize communication overhead in defining the groups and types in output messages.

Group Bytes

Based on the standard binary output group message format, the selection of active binary groups is performed by a single byte bitfield, which limits the number of possible groups to eight (8). To allow for additional output groups, the high-bit of the group byte is reserved as an extension bit. If that extension bit is set high, then an additional group byte follows the previous group byte, up to a total of four (4) group bytes. In both the binary configuration registers (Register 75) and the binary output messages, only the minimum number of group bytes are used to minimize communication overhead. The last group byte must have the extension bit low, signifying there are no additional group bytes.

Because the high-bit is reserved for this extension, each group byte can therefore designate seven (7) groups, allowing up to 28 binary groups. The bit offsets in Table 2.1 account for the presence of extension bits, and can be used directly to determine the bit offset required to enable a group in a set of multiple group bytes.

Type Words

The extension for type words operates identically to that of group bytes, with the high-bit of the word reserved as an extension bit. Type definitions are limited to two (2) type words, allowing up to 30 types per group. As with groups, the bit offsets in the binary output type tables (eg. Table 2.3) account for the presence of extension bits, and can be used directly to determine the bit offset required to enable a type relative to the two type words.

Example

For this example, consider a hypothetical output at 40 Hz on UART-1:

1. Binary Type Offset 8 from Group Offset 4
2. Binary Type Offsets 0 & 16 from Group Offset 11

The first group byte can only designate group offsets 0-6, so in addition to setting the correct bit for group offset 4, the extension bit must also be set high, yielding a first group byte of `0x90`. The second group byte can be used to designate group offset 11 and no additional bytes are needed, so the extension bit must be low (designating the last group byte), yielding a second group byte of `0x10`.

Group offset 4 requires only a single type word of `0x0100`. Group offset 11 requires two type words to specify type offsets 0 & 16, so the first type word of that group must both select bit 0 and set the extension bit, yielding a first type word of `0x8001`. The second type word designates type offset 16 with a low extension bit, yielding a second type word of `0x0002`.

- Configuration: `$VNWRG,75,1,20,90,10,0100,8001,0002*69`
- Output Message Bytes: `FA 90 10 00 01 01 80 02 00 ...`

Extended Binary Groups & Types

	Group Bytes															
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
Hex	0x10								0x90							

	Type Word 1															
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Hex	0x01								0x00							

	Type Words 2 & 3																															
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Hex	0x00								0x02								0x80								0x01							

FIGURE B.1

B.1.2 Split Packets

Split packets provide a means for large binary output messages to be split into smaller packets to ease the memory requirements of the host system. A binary output message is automatically output as a series of split packets if that message exceeds 600 bytes in total size (including sync byte, header, payload, and checksum).

Message Format

A split packet consists of the same four (4) components as a standard binary output message (see Section 2.1.3), but with slight variations in the individual component definitions:

1. **Sync Byte** The sync byte is the first byte of a split packet and has a value of `0xFB`.
2. **Header** The header of a split packet is 5 bytes as defined in Table B.1.
3. **Payload** The concatenation of all payloads of a series of split packets with the same MessageID is equivalent to the header and payload of a single standard binary output message.
4. **Checksum** The checksum of a split packet is computed identically to those on a standard binary output message: a CRC calculated over the packet header and payload.

Split Packet Header Format

OFFSET	NAME	FORMAT	DESCRIPTION
0	MsgType	uint8	The message type for split packets is always `0x00`.
1	MessageID	uint8	Message identifier for associating multiple split packets with a single standard binary output message.
2	PacketCount	uint8	Total number of packets for this message (high nibble), and current packet count (low nibble).
3	PayloadLength	uint16	Number of bytes in packet payload.

TABLE B.1

Register Index

VectorNav Technologies, LLC
10501 Markison Rd
Dallas, TX 75238, USA

Tel: +1.512.772.3615
Email: sales@vectornav.com
Web: vectornav.com

Version 24-20005-R1