


Hemisphere GPS®

GPS Technical Reference

Part No. 875-0175-000 Rev. D1

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.

Copyright Notice

Hemisphere GPS Precision GPS Applications

Trademarks

Notice to Customers

Contact your local dealer for technical assistance. To find the authorized dealer near you, call or write us at:

Hemisphere GPS

4110 9 Street S.E.	Telephone number:	(403) 259-3311
Calgary, AB, Canada	Fax number:	(403) 259-8866
T2G 3C4	E-mail address:	sales@hemispheregps.com

Table of Contents

List of Figures

List of Tables

1: Quick Start

Quick Start

Receiver Quick Start

The purpose of this chapter is to help get the Crescent A100®, R100 R110 receiver running quickly. This chapter is not intended to replace the balance of this reference manual and it assumes a reasonable amount of knowledge with installing and operating of GPS navigation systems.

We recommend that consulting Appendix A for further information on these services and technology GPS, SBAS, or other GNSS systems.

Note: In this reference guide, “receiver” refers to the Crescent OEM board, A100, R100, R110 and Crescent Vector.

NMEA 0183 Message Interface

The receiver uses the NMEA 0183 interface for interfacing, which allows easy configuration changes by sending text-type commands to the receiver.

Where appropriate, relevant commands for making the configuration changes are discussed in the following chapters. Chapter 5, however, is devoted to describing the NMEA 0183 interface in detail.

Binary Message Interface

In addition to the NMEA 0183 interface, the receiver also supports a selection of binary messages. There is a wider array of information available through the binary messages, plus binary messages are inherently more efficient with data. If the application has a requirement for raw measurement data, for instance, this information is available only in a binary format. Consult Chapter 6 for more information on binary messages.

PocketMAX PC

Chapter 4 of this manual provides sufficient information on how to communicate to the Receiver Evaluation system using our PocketMAX PC utility. This program allows the status and function of the receiver to be graphically monitored, in addition to providing an interface for its control.

We recommend gaining initial experience with the receiver using this utility and then migrate work either to a dumb terminal or begin the integration of appropriate commands and messages within the application software.

2: Introduction

Introduction

This chapter provides a brief introduction to the receiver and some of its high-level features. The remaining chapters provide more detailed information on the workings of the product and the integration requirements.

As mentioned in the previous chapter, we recommend consulting Appendix A for further information on these GPS, SBAS and other GNSS system services and technologies.

Both the GPS and SBAS operation of the receiver module features automatic operational algorithms. When powered for the first time, the receiver system will perform a "cold start," which involves acquiring the available GPS satellites in view and the SBAS differential service.

If SBAS is not available in a particular area, an external source of RTCM SC-104 differential correction may be used. If an external source of correction data is needed, the external source needs to support an eight data bit, no parity and one stop bit configuration (8-N-1).

This chapter describes the various modes of operation and features of the receiver.

GPS

The GPS engine is always operating, regardless of the DGPS mode of operation. The following sections describe the general operation of the receiver.

Satellite Tracking

The receiver automatically searches for GPS satellites, acquires the signal, and manages the associated navigation information required for positioning and tracking. This is a hands-free mode of operation. Satellite acquisition quality is described as a Signal-to-Noise Ratio (SNR). A higher SNR is indicative of better quality signal reception. SNR information is provided by the receiver through the use of NMEA 0183 data messages available via its multiple serial ports.

Positioning Accuracy

The receiver is a sub-meter, 95 percent accurate product under ideal conditions horizontally (minimal error). To determine the positioning performance of the receiver, Hemisphere GPS gathers a 24-hour data set of positions in order to log the diurnal environmental effects and also to log full GPS constellation changes. Shorter data sets than 24 hours tend to provide more optimistic results.

Keeping in mind that this horizontal performance specification is a real world, but ideal scenario test. Obstruction of satellites, multipath signals from reflective objects, and operating with poor corrections will detract from the receiver's ability to provide accurate and reliable positions. Differential performance can also be compromised if the receiver module is used in a region without sufficient ionospheric coverage. If external corrections are used, the baseline separation between the remote base station antennas can affect performance.

Since the receiver will be used in the real world, blockage of the line of sight to SBAS satellites is often inevitable. The COAST function provides solace from obstruction of SBAS service for up to 30 to 40 minutes, depending on the amount of tolerable performance drift.

The estimated positioning precision is accessible through the use of NMEA 0183 command responses as described in Chapter 5 (The GST NMEA data message). As the receiver is not able to determine accuracy with respect to a known location in real time (this is traditionally performed in post-mission analyses), the precision numbers are relative in nature and are only approximates.

Update Rates

The update rate of each NMEA 0183 and binary message of the receiver can be set independently with a maximum that is dependant upon the message type. Some messages have a 1 Hz maximum, for example, while other messages have a 20 Hz. The higher update rates, such as 20 Hz, are an option and can be obtained at an additional cost.

Higher update rates are valuable for applications where higher speeds are present, such as aviation, or more frequent updates are required for manual navigational tasks, such as agricultural guidance.

SBAS

The following section describes the general operation and performance monitoring of the Space-Based Augmentation System (SBAS) demodulator within the receiver module.

Automatic Tracking

The SBAS demodulator featured within the receiver will automatically scan and track two SBAS satellite signals, specified by the $JWAASPRN command (defaulted to WAAS PRN 135 and 138). The receiver will automatically track different satellites if the two default satellites become disabled. This automatic tracking allows the user to focus on other aspects of their application rather than ensuring the receiver is tracking SBAS correctly.

The SBAS demodulator features two-channel tracking that provides an enhanced ability to maintain acquisition on a SBAS signal satellite in regions where more than one satellite is in view. This redundant tracking approach will result in more consistent acquisition of a signal when in an area where signal blockage of either satellite is possible.

Performance

The performance of SBAS is described in terms of Bit Error Rate (BER). The SBAS receiver requires a line of sight to the SBAS satellite in order to acquire a signal.

The BER number indicates the number of unsuccessfully decoded symbols in a moving window of 2048 symbols. Due to the use of forward error correction algorithms, one symbol is composed of two bits. The BER value for both SBAS receiver channels is available in the RD1 NMEA 0183 data message described in Chapter 5.

A lower BER indicates that data is being successfully decoded with fewer errors, providing more consistent throughput. The BER has a default, no-lock of 500 or more. As the receiver begins to successfully acquire a signal, it will result in a lower BER. For best operation, this value should be less than 150 and ideally less than 20.

SBAS broadcasts an ionospheric map on a periodic basis that may take up to five minutes to receive upon startup. The receiver uses the broadcast ionosphere model until it downloads the SBAS map, which can result in a lower performance compared to when the map has been downloaded. This will be the case for any GPS product supporting SBAS services.

Warning!

When the map has been downloaded, a position jump due to the potential difference between the GPS ionospheric model and the ionosphere SBAS map may be observed. To minimize the impact of this issue on the use of the receiver, wait up to five minutes before using the receiver or issue the $JQUERY,GUIDE<CR><LF> message to “ask” the receiver if it feels the performance will be sufficient for operation.

WAAS

The US Federal Aviation Administration is in the process of developing a Wide Area Augmentation System (WAAS) for the purpose of providing accurate positioning to the aviation industry. In addition to providing a high quality and accurate service for this industry, this service is available free of charge to civilians and markets in North America.

Other Government agencies are in the process of developing similar WAAS-compatible systems for their respective geographic regions. In Europe, the European Space Agency, the European Commission and EUROCONTROL are jointly developing the European Geostationary Overlay System (EGNOS). In Japan, the MTSAT Satellite-based Augmentation System (MSAS) is in development by the Japan Civil Aviation Bureau (JCAB). These compatible augmentation systems fall into a broader category often referred to as Space Based Augmentation System (SBAS). The receiver is capable of receiving correction data from all WAAS-compatible SBAS.

WAAS DGPS

WAAS differential, and other compatible SBAS, use a state-based approach in their software architecture. These services take in reference data from a network of base stations and endeavor to model the sources of error directly, rather than computing the sum impact of errors upon observed ranges. The advantage of this approach is that the error source can be more specifically accounted during the correction process.

Specifically, WAAS calculates separate errors for the following:

- Ionospheric error
- GPS satellite timing errors
- GPS satellite orbit errors

Provided that a GPS satellite is available to the WAAS reference station network for tracking purposes, orbit and timing error corrections will be available for that satellite. Ionospheric corrections for that satellite are only available if the signal passes through the ionospheric map provided by WAAS, which covers the majority of North America.

To improve upon the ionospheric map provided by WAAS, the receiver extrapolates information from the broadcast ionospheric coverage map, extending its effective coverage. This allows the receiver to be used successfully in regions that competitive products may not. This is especially important in Canada for regions north of approximately 54° N latitude and est of 110° W longitude. Please note that the process of estimating ionospheric corrections beyond the WAAS broadcast map would not be as good as having an extended WAAS map in the first place. This difference may lead to minor accuracy degradation.

Figure 2-1, on page 13, and Figure 2-2, on page 14, depict the broadcast WAAS ionspheric map extent and the Hemisphere GPS extrapolated version, respectively. As can be seen from Figure 2-2, on page 14, the coverage compared to Figure 2-1, on page 13, extends further in all directions, enhancing usable coverage.

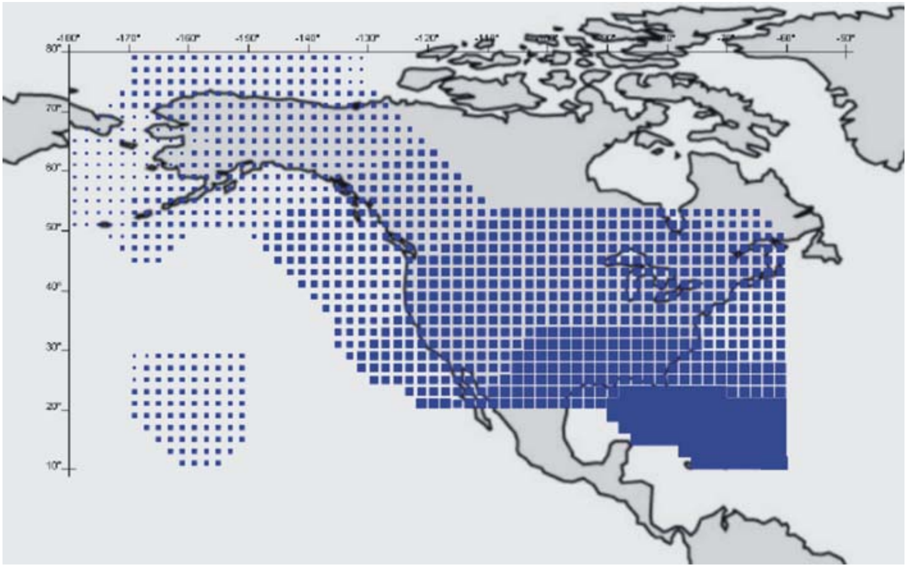


Figure 2-1. Broadcast WAAS ionspheric correction map

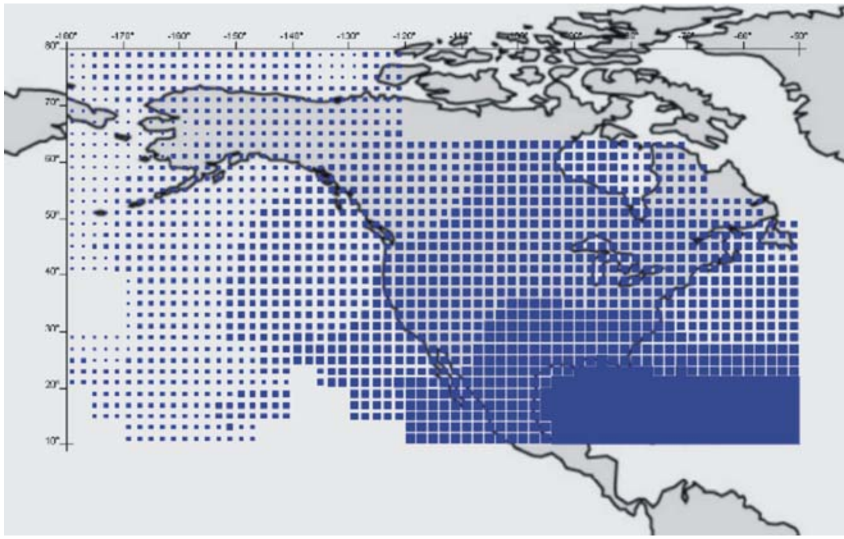


Figure 2-2. Extrapolated WAAS ionspheric correction map

WAAS Signal Information

WAAS and other SBAS systems transmit correction data on the same frequency as GPS, allowing the use of the same receiver equipment used for GPS. Another advantage of having WAAS transmit on the same frequency as GPS is that only one antenna element is required.

WAAS Reception

Since WAAS broadcasts on the L-band, the signal requires a line of site in the same manner as GPS to maintain signal acquisition.

Currently, two commercial satellites are transmitting WAAS data for public use and two additional satellites are in test mode. Due to their location, these satellites may appear lower on the horizon, depending on the geographic position on land and on which satellite that is tuned to. When using WAAS correction data, the receiver is able to provide the azimuth and elevation of all satellites to aid in determining their position with respect to the antenna.

WAAS Coverage

Figure 2-3, on page 16, depicts the current WAAS coverage as provided by the currently leased geostationary satellites. The WAAS satellites are identified by their Pseudo-Range Number (PRN). PRN satellites 135 and 138 are scheduled to from testing mode into operation in the Fall of 2006. In some areas, two or more satellites may be visible. Please note that signal coverage may be present in some areas without either sufficient ionospheric map coverage or satellites with valid orbit and clock corrections. In such a case, differential positioning with WAAS may not be possible, as four or more satellites with correctors must be available to compute a DGPS position.

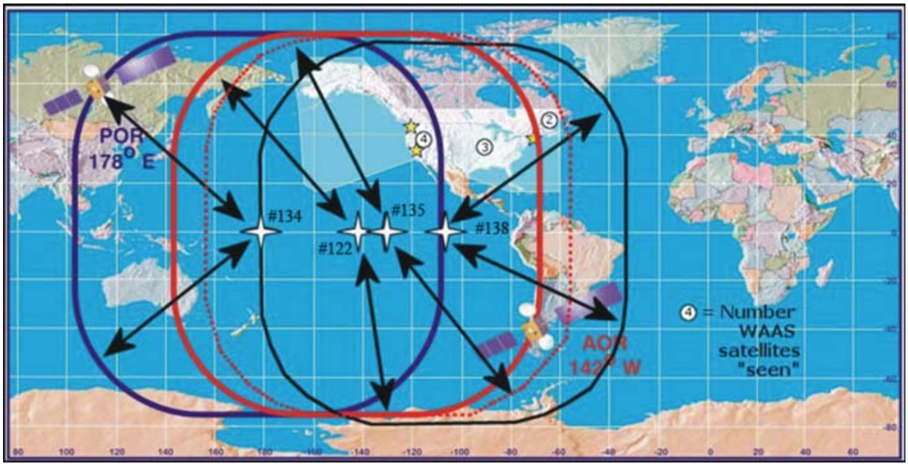


Figure 2-3. WAAS coverage

EGNOS

The European Geostationary Navigation Overlay Station (EGNOS) is one of two commercial satellites transmitting differential correction data for public use. EGNOS is currently located over the Atlantic and Pacific oceans. Due to its location over the oceans, these satellites may appear lower over the horizon, depending on the geographic position on the land. In regions where the satellites appear lower on the horizon, they may be more prone to being masked by terrain, foliage, buildings or other objects, resulting in signal loss. The farther away from the equator, and the satellite's longitude, will cause the satellite to appear lower on the horizon. Hemisphere GPS' COAST technology helps alleviate this problem by maintaining system performance when EGNOS signal loss occurs for extended periods of time. More information on COAST technology is provided later in this chapter.

ESTB Coverage

Figure 2-4, on page 18, presents approximate EGNOS cover provided by the leased Inmarsat Atlantic Ocean Region-East (AORE) and Indian Ocean region (IOR) satellites.

Although EGNOS is not yet broadcasting an official signal, Figure 2-4, on page 18, presents approximate EGNOS test-bed coverage provided by the leased geostationary satellites. Figure 2-4, on page 18, approximates coverage with white shading. Virtually all of Europe, part of Northern Africa and into the Middle East is covered with at least one signal. Most of Europe is covered by two signals.

Note: The satellite elevation angle lowers with increasing distance away from the equator and from the satellite's longitude. Although a good amount of signal coverage is shown in northern latitudes for EGNOS, it may not be usable due to its low elevation angle and the potential for it to be obstructed. ideally, testing of the system in the area of its use is recommended to ensure that the signal is sufficiently available.

Figure 2-4. EGNOS coverage

MSAS

The MTSAT Satellite-based Augmentation System (MSAS) is currently run by the Japan Civil Aviation Bureau (JCAB). The MSAS signal is being broadcast in test mode and no coverage map is available. Further information on the system can is posted at the web site below:

http://www.kasc.go.jp/MSAS/index_e.html

COAST Technology

Crescent OEM and Crescent Vector OEM boards also feature Hemisphere GPS' exclusive COAST software that enables Hemisphere GPS' receivers to utilize old differential GPS correction data for 40 minutes or more without significantly affecting the quality of positioning. When using COAST, the Crescent OEM is less likely to be affected by differential signal outages dues to signal blockages, weak signals or interference.

Note: In order to obtain a full set of SBAS corrections, the Crescent receiver must receive the ionospheric map over a period of a few minutes. After this, the receiver can "coast" until the next set of corrections has been received.

COAST technology provides the following benefits:

- Accurate and minimal position drift during temporary loss of differential signal corrections
- Maintain sub-meter accuracy up to 40 minutes after differential signal loss
- Provides outstanding performance in environments where maintaining a consistent differential link is difficult
- Standard with Crescent GPS receiver technology

e-Dif - Extended Differential Option

The receiver module is designed to work with Hemisphere GPS' patented Extended Differential (e-Dif) software. e-Dif is an optional mode where the receiver can perform with differential-like accuracy for extended periods of time without the use of a differential service. It models the effects of ionosphere, troposphere, and timing errors for extended periods by computing its own set of pseudo-corrections.

e-Dif may be used anywhere geographically and is especially useful where SBAS networks have not yet been installed, such as South America, Africa, Australia, and Asia. An evaluation software key for the receiver is needed to use e-Dif. It can be easily installed in the field using a PC computer and through the issue of a $JK NMEA 0183 command.

Positioning with e-Dif is relatively jump-free, provided that the receiver consistently maintains a lock on at least four satellites at one time. The accuracy of positioning will have a slow drift that limits use of the e-Dif for approximately 30 to 40 minutes, however, it depends on how tolerable the application is to drift, as e-Dif can be used for longer periods.

This mode of operation should be tested to determine if it is suitable for the application and for how long the user is comfortable with its use. As accuracy will slowly drift, the point at which to recalibrate eD-f in order to maintain a certain level of accuracy must be determined.

Figure 2-5, on page 22, displays the static positioning error of e-Dif while it is allowed to age for 14 consecutive cycles of 30 minutes. The top line indicates the age of the differential corrections. The receiver computes a new set of corrections using e-Dif during the calibration at the beginning of each hour and modifies these corrections according to its models. After the initialization, the age correspondingly increases from zero until the next calibration.

The position excursion from the true position (the lines centered on the zero axis are Northing (dark line) and Easting (light line)) with increasing correction age is smooth from position to position, however there is a slow drift to the position. The amount of drift will depend on the rate of change of the environmental errors relative to the models used inside the e-Dif software engine.

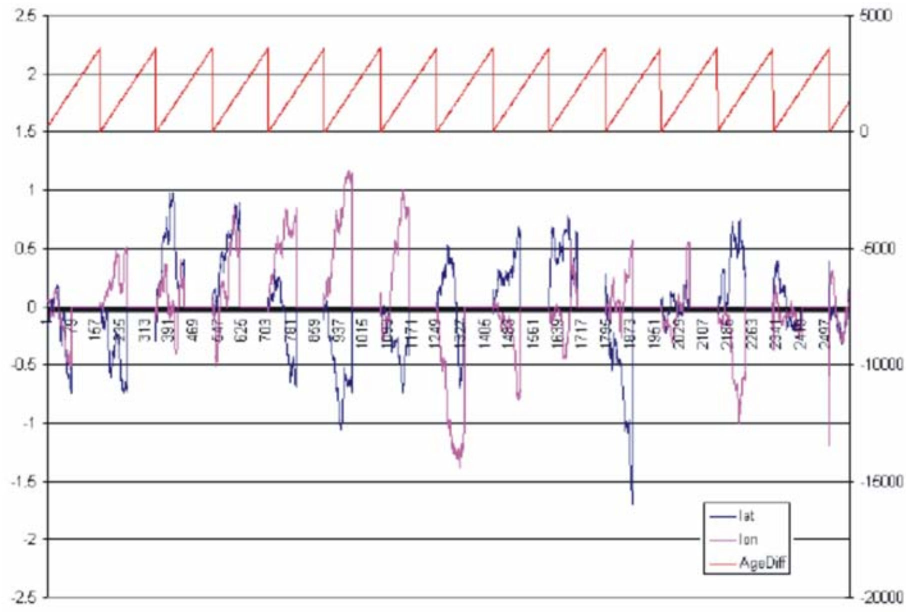


Figure 2-5. e-Dif error drift

As mentioned, it is up to the user for how long e-Dif is to function before performing another calibration. We recommend to test this operation mode to determine the level of performance that is acceptable.

e-Dif Operation

Operation of the receiver unit with the optional e-Dif application requires the sending of NMEA 0183 commands. These commands may be automatically issued through customized software or a simple terminal interface running on a PC, PDA or data logger.
Chapter 5 provides detailed information on the commands supported by the e-Dif feature.

Start-Up

When turning on the receiver on with the e-Dif application running, it will require a minimum of a few minutes to gather enough satellite tracking information to model the errors for the future (up to 10 minutes may be required depending on the environment). The receiver does not have to stay stationary for this process, but should be ensured that the receiver maintains acquisition on the satellites available. This process of gathering information and the subsequent initialization of e-Dif is "calibration."

Calibration

Calibration is the process of zeroing the increasing errors in the e-Dif modeling process. Calibration can be performed either in a relative or absolute sense, depending on positioning needs. Relative positioning will provide positions that are accurate to one another. However, there may be some offset compared to truth. Additionally, unless the same arbitrary point is used for all calibrations, and its assume position stored, it is possible for different cycles of e-Dif to have an offset.

Calibrating for relative positioning is easier than for absolute positioning, since any arbitrary position can be used. Calibrating for absolute positioning mode requires that this task is performed with the antenna at a known reference location. Use this point for subsequent calibrations.

e-Dif Performance

The receiver's positioning performance is dependant upon the rate at which the environmental modeling of e-Dif and the environmental errors diverge. The more that e-Dif is able to model the errors correctly, the longer that e-Dif will provide reliable and accurate positioning. As there is no way in real-time to know the rate of divergence, a rule of thumb is to set the maximum age of differential to either 30 or 40 minutes, depending on how much error the application is able to tolerate (or simply recalibrate before 30 to 40 minutes goes by). Our testing has shown that accuracy will often be better than 1.0 m virtually 95 percent of the time after 30 minutes of e-Dif operation.

We suggest that the user performs testing at their location to determine the level of performance that would be expected to be seen on average. When testing this feature, it is a good idea to look at a number of e-Dif cycles per day, and monitor performance against a known coordinate and possibly other receivers in autonomous and differential mode. This should be done over a number of days with different states of the ionoshpere. The energy level of the ionosphere based upon the amount of solar flare activity can be monitored at the following web sites:

http://iono.jpl.nasa.gov

http://www.spaceweather.com

Base Station Operation

Operation of the receiver with the optional base station application requires the sending of NMEA 0183 commands. These commands may be automatically issued through customized software or through a simple terminal interface running on a PC, PDA or data logger. Chapter 5 provides detailed information on the commands supported by the base station feature.

Start up

When turning on the receiver on with the base station application running (the e-Dif application is used, but different commands must be issued), it will require a minimum of a few minutes to gather enough satellite tracking information to model the errors for the future. Up to 10 minutes may be required depending on the environment. The receiver needs to be kept stationary for this process and it is important to secure the antenna for the base station in a stable location. We refer to this process of gathering information and the subsequent initialization of base station as "calibration."

Calibration

Calibration is the process of zeroing the increasing errors in the base station modeling process. Calibration can be performed either in a relative or absolute sense, depending on the positioning needs. Relative positioning will provide positions that are accurate to one another, however, there may be some offset compared to truth.

Calibrating for relative positioning is easier than for absolute position, since any arbitrary position can be used. Calibrating for absolute positioning mode requires that this task is performed with the antenna at a known reference location.

Base Station Performance

The positioning performance of the receiver unit is dependant upon the rate at which the environmental modeling of the base station feature and the environmental errors diverge. The more that the base station is able to model the errors correctly, the longer that base station will provide reliable and accurate positioning.

We suggest that users perform their own testing at their location to determine the level of performance that they would expect to see on average. When testing this feature, it is a good idea to look at a number of lengths of tests, and monitor performance against a known coordinate and possibly other receivers in autonomous and differential mode. This should be done over a number of days with different states of the ionosphere. The energy level of the ionosphere based upon the amount of solar flare activity can be monitored at the following web sites:

http://iono.jpl.nasa.gov

http://www.spaceweather.com

L-Dif - Local Differential Option

Local Differential (L-Dif) is a specialized message type that can only be sent between two Crescent-based receivers. One receiver is used as the base station and must remain stationary. It is extremely useful to know the coordinates of the base station position, but averaging the position over several days will also suffice. The second receiver is used as a rover and the messages must be sent either through a cable or over a radio link.

Start-up

When turning on the receiver with the L-Dif running, it will require several commands to initialize the proprietary messages that are sent over the air. These commands are outlined in Chapter 5.

L-Dif Performance

The positioning performance of the receiver in L-Dif mode is dependant upon the environment of the base and rover receivers, the distance between them and the accuracy of the entered coordinates of the base station. We suggest that the user perform their own testing at their location to determine the level of performance that they would expect to see on average. When testing this feature, it is a good idea to look at a lengthy test of 12-24 hours, in different environments, and monitor performance against a known coordinate. This should be done over a number of days with different states of the ionosphere. The energy level of the ionosphere based upon the amount of solar flare activity can be monitored at the following web sites:

http://iono.jpl.nasa.gov

http://www.spaceweather.com

OmniSTAR

OmniSTAR is a worldwide terrestrial DGPS service that provides correction data to subscribers of the system with the use of a geostationary transponder.

OmniSTAR DGPS

OmniSTAR is a wide area DGPS service. The information broadcast by this service is based upon a network of reference stations placed at geographically strategic locations. The network stations communicate GPS correction data to control centers where it is decoded, checked, and repackaged into a proprietary format for transmission to a geostationary L-band communications satellite. The satellite rebroadcasts the correction information back to earth over a large signal footprint where the DGPS MAX's L-band differential satellite receiver demodulates the data.

The OmniSTAR signal content is not RTCM SC-104, but a proprietary wide-area signal that is geographically independent. With this service, the positioning accuracy does not degrade as a function of distance to a base station, as the data content is not composed of a single base station's information, but an entire network's information. When the DGPS MAX L-band DGPS receiver demodulates the proprietary signal, it converts it into a local-area format for input to the GPS receiver (standard RTCM SC-104, message Type 1).

The L-band DGPS receiver interpolates corrections from the wide-area signal, specific to the location using Virtual Base Station (VBS) processing algorithms. The resulting RTCM corrections are those that would be calculated if a reference station were set up at the present location. This type of solution ensures a consistent level of accuracy across the entire coverage area. The GPS receiver provides position information to the L-band DGPS receiver for VBS calculations.

OmniSTAR Signal Information

The OmniSTAR L-band signal is a line-of-sight UHF signal that is similar to GPS. There must be a line of sight between the DGPS MAX's antenna and the geostationary communications satellite in order for the L-band differential receiver inside the DGPS MAX to acquire the signal.

Various L-band communications satellites are used for transmitting the correction data to OmniSTAR users around the world. When the DGPS MAX has acquired an OmniSTAR signal, the elevation and azimuth are available in the menu system in order to troubleshoot line of sight problems. Contact OmniSTAR for further information on this service. OmniSTAR contact information is provided in Appendix A of this manual.

OmniSTAR Reception

The OmniSTAR service broadcasts at a similar frequency to GPS, and as a result, is a line-of-sight system. There must be a line of sight between the antenna and the OmniSTAR satellite for reception of the service.

The OmniSTAR service uses geostationary satellites for communication. The elevation angle to these satellites is dependent upon latitude. For latitudes higher than approximately 55° north or south, the OmniSTAR signal may be blocked more easily by obstructions such as trees, buildings, terrain, or other objects.

OmniSTAR Coverage

Figure 2-6, on page 30, shows approximate OmniSTAR service coverage. Regions without coverage, or with poor coverage, are shown with dark shading.

Figure 2-6. Worldwide OmniSTAR coverage

Note: Signal coverage may be present in some areas without reference stations within the region. Operating outside of the reference station network may cause the applicability of the correction data to be less, resulting in a lower degree of positioning accuracy due to spatial decorrelation.

Note: OmniSTAR is a terrestrial-only service.

Automatic Tracking

The receiver features an automatic mode that allows the receiver to locate the best spot beam if more than one is available in a particular region. The L-band DGPS receiver's frequency does not need to be adjusted with this function. The OmniSTAR receiver also features a manual tune mode for flexibility.

Receiver Performance

The OmniSTAR receiver provides both a lock icon and a BER to describe the lock status and reception quality. Both of these features depend on a line-of-sight between the A20/A30 antenna and the geostationary communications satellite broadcasting OmniSTAR correction information.

The A20/A/30 antenna is designed with sufficient gain at low elevation angles to perform well at higher latitudes where the signal power is lower and the satellite appears lower on the horizon.

The BER number indicates the number of unsuccessfully decoded symbols in a moving window of 2048 symbols. Due to the use of forward error correction algorithms, one symbol is composed of two bits.

The BER has a default, no-lock value of 500. As the receiver begins to successfully acquire the signal, it will result in a lower BER. For best operation, this value should be less than 150 and ideally less than 20.

Radiobeacon Range

Many marine authorities, such as Coast Guards, have installed networks of radiobeacons that broadcast DGPS corrections to users of this system. With the increasing utility of these networks for terrestrial applications, there is an increasing trend towards densification of these networks inland.

Radiobeacon range

The broadcasting range of a 300 kHz beacon is dependent upon a number of factors, including transmission power, free space loss, ionospheric state, surface conductivity, ambient noise, and atmospheric losses.

The strength of a signal decreases with distance from the transmitting station, due in large part to spreading loss. This loss is a result of the signal's power being distributed over an increasing surface area as the signal radiates away from the transmitting antenna.

The expected range of a broadcast also depends upon the conductivity of the surface over which it travels. A signal will propagate further over a surface area with high conductivity than a surface with low conductivity. Lower conductivity surfaces, such as dry, infertile soil, absorb the power of the transmission more than higher conductivity surfaces, such as sea water or arable land.

A radiobeacon transmission has three components:

- Direct line of sight wave
- Ground wave
- Sky wave

The line of sight wave is not significant beyond visual range of the transmitting tower and does not have a substantial impact upon signal reception.

The ground wave portion of the signal propagates along the surface of the earth, losing strength due to spreading loss, atmospheric refraction and diffraction, and attenuation by the surface over which it travels (dependent upon conductivity).

The portion of the beacon signal broadcast skyward is known as the "sky wave." Depending on its reflectance, the sky wave may bounce off the ionosphere and back to Earth, causing reception of the ground wave to fade. Fading occurs when the ground and sky waves interfere with each other. The effect of fading is that reception may fade in and out. However, this problem usually occurs in the evening when the ionosphere becomes more reflective and usually on the edge of coverage areas. Fading is not usually an issue with overlapping coverage areas of beacons and their large overall range.

Atmospheric attenuation plays a minor part in signal transmission range, as it absorbs and scatters the signal. This type of loss is the least significant of those described.

Radiobeacon Reception

Various sources of noise affect beacon reception and include:

- Engine noise
- Alternator noise
- Noise from power lines
- DC to AC inverting equipment
- Electric devices such as CRT's, electric motors and solenoids

Noise generated by this type of equipment can mask the beacon signal, reducing or impairing reception.

Antenna Placement

When using the internal beacon receiver as the correction source, selecting an appropriate location for installation of the antenna will influence the performance of the internal beacon receiver. The following list provides some general guidelines for deciding upon an antenna location:

- Choose a location with a clear view of the sky. This is important for GPS, WAAS, and OmniSTAR signal reception.
- Choose a location that is at least three feet away from all forms of transmitting antennas, communications, and electrical equipment. This will reduce the amount of noise present at the antenna, improving beacon receiver performance.
- Install the antenna in the best location for the application, such as the center line of the vehicle or vessel. The position calculated by the beacon receiver is measured to the center of the antenna.
- Do not locate the antenna in areas that exceed environmental conditions that are specified.

Radiobeacon Coverage

Figure 2-7, on page 35, shows the approximate radiobeacon coverage throughout the world. In Figure 2-7, on page 35, light shaded regions note current coverage, with beacon stations symbolized as white circles. The world beacon networks continue to expand. For more current coverage, consult the Hemisphere GPS web site at www.hemispheregps.com.

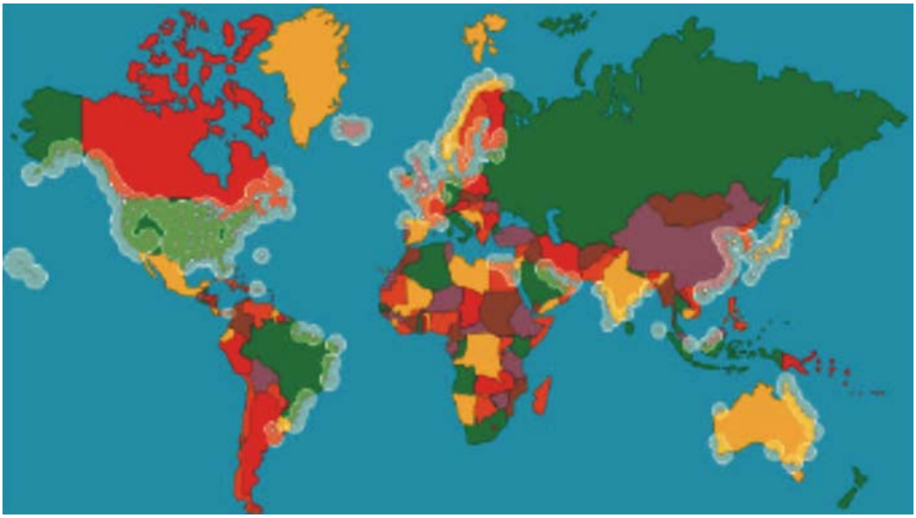


Figure 2-7. World DGPS radiobeacon coverage

Crescent Vector OEM Development Kit

The purpose of a Crescent Vector OEM Development Kit is to provided accurate and reliable heading and position information at high update rates. To accomplish this task, the unit uses one high performance GPS engine and two multipath resistant antennas for GPS signal processing. One antenna is designated as the primary GPS. The other antenna is designated as the secondary GPS. Positions computed by the unit are referenced to the phase center of the primary GPS antenna. Heading data references the vector base case formed from the primary GPS antenna phase center to the secondary GPS antenna phase center.

Moving Base Station RTK

The Crescent Vector's GPS engine uses both the L1 GPS C/A code and phase data to compute the location of the secondary GPS antenna in relation to the primary GPS antenna with a very high sub-centimeter level of precision. The technique of computing the location of the secondary GPS antenna with respect to the primary antenna, when the primary antenna is moving is often referred to as moving base station Real-Time Kinematic, or moving base station RTK.

RTK technology, generally, is very sophisticated and requires a significant number of possible solutions to be analyzed where various combinations of integer numbers of L1 wavelengths to each satellite intersect within a certain search volume. The integer number of wavelengths is often referred to as the "Ambiguity," as they are initially ambiguous at the start of the RTK solution.

The Crescent Vector places a constraint on the RTK solution with the prior knowledge of the fact that the secondary GPS antenna has a fixed separation usually of 0.50 meters (1.6 feet) (this can vary based on setup) from the primary GPS antenna. This reduces the search volume considerably, thus the startup times, since the location of the secondary antenna can theoretically fall only on the surface of a sphere with a radius of 0.50 meters (1.6 feet) centered on the location of the primary antenna, versus a normal search volume that is greater than a cubic meter.

Supplemental Sensors - Reduced Time Search

In addition to incorporating the GPS engine, integrated inside the Crescent Vector are a gyro and a tilt sensor. When used, the tilt sensor aids the rate at which a heading solution is computed on startup and also during reaquisition if the GPS heading is lost due to obstructions. Each supplement sensor may be turned on or off individually, however, the full functionality of the Crescent Vector is realized only when all are used.

The tilt sensor reduces the search volume further beyond the volume associated with just a fixed antenna separation, since the Crescent Vector knows the approximate inclination of the secondary antenna with respect to the primary. The gyro only benefits reacquisition, since it initially requires a GPS heading to self-calibrate. The gyro further reduces the search volume. Reducing the RTK search volume also has the benefit of improving the reliability and accuracy of selecting the correct heading solution by eliminating other possible erroneous solutions.

Note: By default, the tilt aiding and the gyro are turned on.

Supplemental Sensors - Heading System Backup

The gyro is able to operate as secondary source of heading during periods of GPS outages due to obstruction. The Crescent Vector will use the gyro for heading during a short outage. If the outage lasts longer than 3 minutes, the gyro will be deemed to have drifted too far and will stop outputting. There is no user control over the time-out period of the gyro.

Post processing

The receiver module is able to output raw measurement data for post processing applications. The raw measurement and ephemeris data are contained in the Bin 95 and Bin 96 messages documented in this manual. Both messages must be logged in a binary file. Depending on the application, site data can be included within the binary file and the user can perform the translation to RINEX.

We make a Windows-based RINEX translator available, however, RINEX has no facility to store station information. Our translator is available by contacting technical support at Hemisphere GPS.

Evaluating Receiver Performance

As mentioned earlier, Hemisphere GPS evaluates performance of the receiver with the objective of determining best-case performance in a real-world environment. Our static testing has shown that the receiver achieves a performance better than one meter 95 percent of the time.

The qualifier of 95 percent is a statistical probability. Often manufacturers use a probability of “rms,” or “standard deviation.” Performance measures with these probabilities are not directly comparable to a 95 percent measure since they are lower probability (less than 70 percent probability).

Table 2-1, on page 40, summarizes the common horizontal statistical probabilities.

Table 2-1: Horizontal Accuracy Probability Statistics

Accuracy measure	Probability (%)
rms (root mean square)	63 to 68
CEP (circular error probability)	50
R95 (95 percent radius)	95 to 98
2drms (twice the distance root)	95

It is possible to convert from one statistic to another using Table 2-2, on page 41. Using the value where the “From” row meets the “To” column, multiply the accuracy by this conversion value.

Table 2-2: Horizontal Accuracy Statistical Conversions

	To			
From	CEP	rms	R95	2drms
CEP	1	1.2	2.1	2.4
rms	0.83	1	1.7	2.0
R95	0.48	0.59	1	1.2
2drms	0.42	0.5	.83	1

For example, if Product A, after testing, results in an accuracy of 90 cm 95 percent (R95)

To compare this to Product B that has a sub-meter horizontal rms specification

1. Select the value from where the “R95” row and the “rms” column intersect (to convert to rms). This conversion value is 0.59.
2. Multiply the 90 cm accuracy by this conversion factor and the result will be 53 cm rms. By comparing this to Product B’s specification of sub-meter rms, the first Product A would offer better performance.

To properly evaluate one receiver against another statistically, the receivers should be using identical correction input (from an external source) and also share the same antenna using a power splitter (equipped with appropriate DC-blocking of the receivers and a bias-T to externally power the antenna). With this type of setup, the errors in the system are identical with the exception of receiver noise.

Although this is a comparison of the GPS performance qualities of a receiver, it excludes other performance merits of a GPS engine. The dynamic ability of a receiver should always be compared in a similar way with the test subjects sharing the same antenna. Unless a receiver is moving, its software filters are not stressed in a similar manner to the final product application. When testing dynamically, a much more accurate reference would need to be used, such as an RTK system, so that a “truth” position per epoch is available.

Further, there are other performance merits of a GPS engine, such as its ability to maintain a lock on GPS and SBAS satellites. In this case, the same GPS antenna should be shared between the receiver test subjects. For the sake of comparing the tracking availability of one receiver to another, no accurate “truth” system is required, unless performance testing is also to be analyzed. Again, an RTK system would be required, however, it is questionable how its performance will fair with environments where there are numerous obstructions, such as foliage. Other methods of providing a truth reference may need to be provided through observation times on surveyed monuments or traversing well-known routes.

Please contact Hemisphere GPS technical support for further assistance in developing a test setup or procedure for evaluation of the receiver.

3: Receiver Operation

Receiver Operation

This chapter introduces the following topics on the receiver operation:

- General operational features of the receiver system
- Operating modes
- Receiver default operation parameters

Powering the Receiver System

Once appropriate power is connected, the receiver will be immediately powered. Please refer to the receiver specific manual for the power specifications of the product.

With the application of power, the receiver will proceed through an internal start-up sequence, however, it will be ready to communicate immediately.

When installed so the antenna that is being used has an unobstructed view of the sky, the receiver will provide a position quickly, within approximately 60 seconds. SBAS lock requires approximately 30 seconds to acquire.

Note: The receiver can take up to 5 minutes for a full ionospheric map to be received from SBAS. Optimum accuracy will be obtained once the receiver is processing corrected positions using complete ionosphere information.

Communicating with the Receiver Module

The receiver module features three primary serial ports that may be configured independently from each other:

- Port A
- Port B
- Port C

The ports may be configured for any mixture of NMEA 0183, binary, and RTCM SC-104 data. The usual data output is limited to NMEA 0183 data messages, since these are industry standard.

Note: If different data types to be output from the receiver simultaneously are required, such as NMEA 0183 and binary or NMEA 0183 and RTCM, ensure that the software used for logging and processing of the data has been designed to correctly parse the different data types from the single stream of data.

NMEA 0183 Interface

NMEA 0183 is a communications standard established by the National Marine Electronics Association (NMEA). NMEA 0183 provides data definitions for a variety of navigation instruments and related equipment. Such instruments supported include gyrocompasses, Loran receivers, echo sounders, GPS receivers, and more. NMEA 0183 functionality is virtually standard on all GPS equipment available. NMEA 0183 has an ASCII character format that allows the user to read the data via terminal software on the receiving device, if possible. An example of one second of NMEA 0813 data from the receiver is provided on the top of page 47.

$GPGGA,144049.0,5100.1325,N,11402.2729,W,1,07,1.0,1027.4,M,0,
M,,010 *61 $GPVTG,308.88,T,308.88,M,
0,0.04,N,0.08,K*42$GPGSV,3,1,10,02,73,087,54,04,
00,172,39,07,66,202,54,08,23,147,48,*79$GPGSV,3,2,
10,09,23,308,54,11,26,055,54,15,00,017,45,21,02,
353,45*78,GPGSV,3,3,10,26,29,257,51,27,10,147,45
,45,,,,,,,,*74

Depending on each manufacturer's goals for a product, they may have the need to combine data into custom messages. This allows them to improve communication and programming efficiency. The standard NMEA 0183 provides for manufacturers to define their own custom, proprietary messages, as required. Proprietary NMEA 0813 messages are likely to be supported only by the specific manufacturer. In the case of the receiver, it is likely that custom NMEA 0183 commands will need to be supported within the application if the software is to be configured on the unit on-the-fly.

The receiver supports a variety of standard and proprietary NMEA 0813 messages. These messages are used to configure the receiver and also contain the required information from the receiver. A selection of NMEA 0183 data messages on one port can be configured at various update rates. Each message has a maximum update rate, and a different selection of NMEA 0183 messages with different rates on another port.

Chapter 5 presents information relating to the NMEA 0183 interface of the receiver smart antenna. Appendix A - Resources provides contact information should to purchase a copy of the NMEA 0183 standard.

Binary Interface

Binary messages may be output from the receiver simultaneously as NMEA 0183 data. Binary messages have a proprietary definition and would likely require custom software support to be used. Binary messages are inherently more efficient than NMEA 0183 and would be used when maximum communication efficiency is required. Use of the binary messages for most users is not recommended - the NMEA 0183 interface allows control of the operation of the receiver and also receives most types of information regarding status and positioning.

Note: If binary data needs to be logged, please ensure that the logging software has opened the file as a binary file, otherwise data may be lost.

RTCM SC-104 Protocol

RTCM SC-104 is a standard that defines the data structure for differential correction information for a variety of differential correction applications. It has been developed by the Radio Technical Commission for Maritime services (RTCM) and has become an industry standard for communication of correction information. RTCM is a binary data protocol and is not readable with a terminal program. It appears as “garbage” data on-screen, since it is a binary format and not ASCII text. The following is an example of how the RTCM data appears on-screen:

```
mRMP@PJfeUtNsmMFM{nVtlOTDbA^xGh~kDH`_FdW_yqLRryrDuh
cB\@}N`ozbSD@O^}nrGqkeTlpLLrYpDqAsrLRrQN{zW|uW@H`z]~aG
xWYt@I`_FxW_qqLRryrDCikA\@Cj]DE]|E@w_mIroMNjkKOsmMFM{
WDwW@HVEbA^xGhLJQH`_F`W_aNsmMFM[WVLA\@S}amz@iIluP
qx~lZhTCpLLrYpdP@kOsmMFM[kVDHwVGbA^P{WWuNt_SW_yMs
mMnqdrhcC\@sE^ZfC@}vJmNGAHJVhTCqLRryrdviStW@H_GbA^
P{wxu[k
```

RTCM has various levels of detail, however, the highest level is the message. RTCM defines numerous messages that contain specific information. The receiver module processes the C/A code and does not support more advanced methods of differential positioning, such as real-time kinematic (RTK), that uses different RTCM message types. Considering this fact, only certain RTCM messages are important for use with the receiver:

- The Type 1 and Type 9 messages contain similar information. These two messages contain pseudo range corrections and range rate corrections to each satellite.
- The Type 2 message contains delta differential corrections that are used when the remote receiver is using a different satellite navigation message than used by the base station.
- The Type 5 message contains GPS constellation health information used for improving tracking performance of a GPS receiver.
- The Type 6 message contains null information and is broadcast so that a beacon receiver demodulating the data from the broadcast does not lose the lock when the beacon station has no new data to transmit.

Note: RTCM is a local area data standard. This means that when positioning with external connection input to the receiver from an external source or outputting corrections from the receiver to another GPS receiver, performance will degrade as a function of distance from the base station. The additional degradation will depend on the difference in observed orbit and ionospheric errors between the reference station and the remote unit. A general rule of thumb would be an additional 1 m error per 100 miles. This error is often seen as a bias in positioning, resulting in a position offset. The scatter of the receiver is likely to remain close to constant.

The RTCM SC-104 data output by the receiver is converted from the RTCA SC-159 data broadcast by the SBAS network.

Appendix A contains the contact information for purchase a copy of the RTCM SC-104 specifications.

Configuring the Receiver

All aspects of receiver operation may be configured through any serial port with the use of NMEA 0183 commands. These commands are described in the Chapter 5. The following items are user-configurable:

- Selecting one of the two on-board applications (SBAS or e-Dif if present)
- Setting the baud rate of both communication ports
- Choosing NMEA 0183 data messages to output on the serial ports and the update rate of each message
- Setting the maximum differential age cut-off
- Setting the satellite elevation angle cut-off mask

Firmware

The software that runs the receiver is often referred to as "firmware," since it operates at a low level. The type of firmware within the receiver is for the processor. This type of firmware may be upgraded in the field through the serial port A as new revisions become available.

The processor of the receiver's engine supports two simultaneous versions of firmware. Only one of them operates at a given time. These two versions of firmware may have different functionality and are also referred to as "applications."

The receiver currently ships with a SBAS (WAAS) application and the e-Dif application unless Local Differential (L-Dif™) subscription has been purchased. Then the two applications would be SBAS (WAAS) and local-dif rover or base. Chapter 5 describes the $JAPP command used to change between the two receiver applications.

Installing Applications onto the Receiver

This section deals with installing applications (e-Dif, L-Dif, RTK, SBAS (WAAS, EGNOS, MSAS, etc.), LBAND, etc.) onto the receiver.

To install the software onto the receiver:

1. Open the RightARM program with the receiver on.

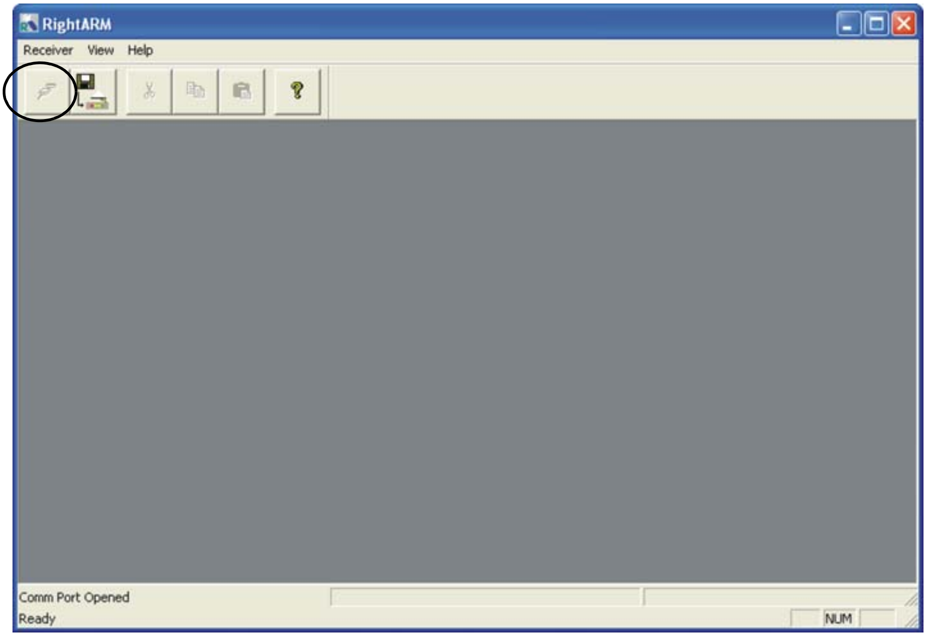
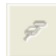
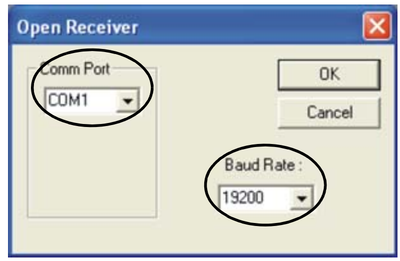


Figure 3-1. RightARM main screen

2. Connect the serial cable from the receiver's data port to computer's serial port.

3. Click  in the tool bar in the tool bar at the top of the main RightARM screen to open the **Open Receiver** window. (See Figure 3-1, on page 52, for the connect button.)



4. Select the appropriate COM Port for the computer.
5. Select the appropriate Baud Rate.

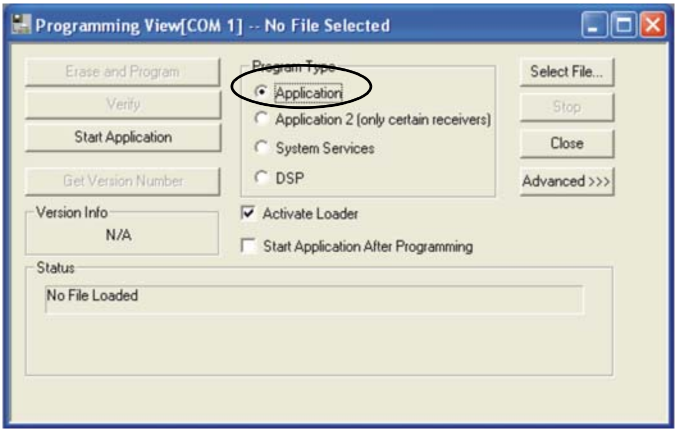
Note: Make sure the sure the receiver's baud rate matches the RightARM's baud rate.

---

6. Click 

.

7. Click  from the tool bar at the top of the main RightARM screen. (See Figure 3-1, on page 52, on page 52, for the connect button.)

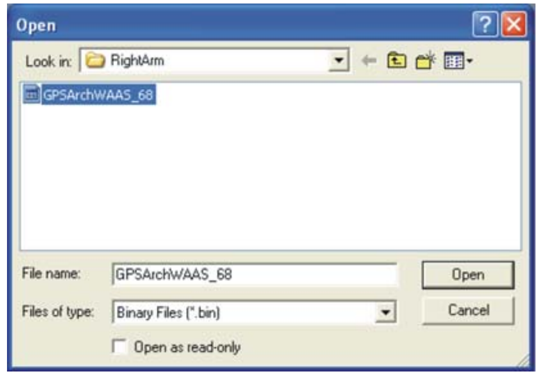


8. Click the Application or Application 2 (only certain receivers) radio button under the Program Type section.

Note: Make sure that the proper application is selected. If the the $JAPP command is given in SLXMon or Pocket Max PC (terminal program) a response will be shown. For example, $JAAP,waas,lband,1,2. This means that WAAS is application 1 and LBAND is application 2.

9. Click 

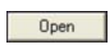
.



10. Choose the desired file. For example, GPSARCHWAAS_68.BIN.

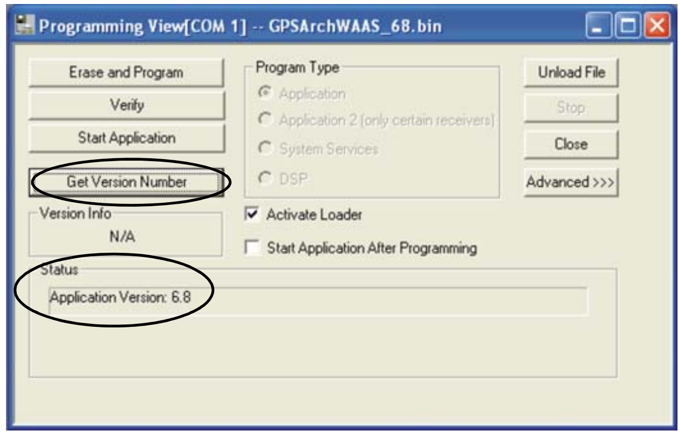
Note: Make sure that the latest file is in the active directory that is choosen.

---

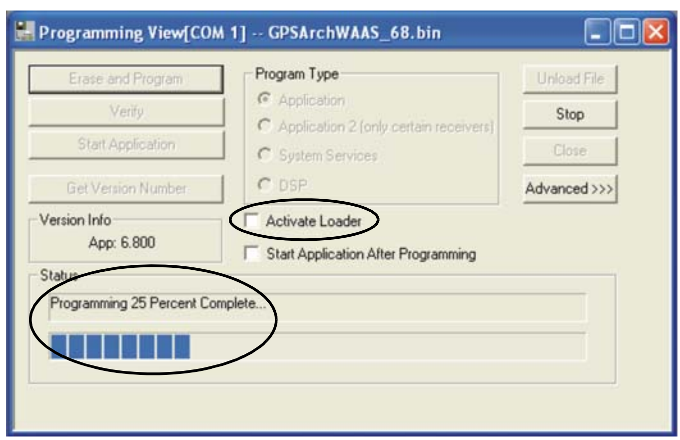
11. Click 

.

12. Click Get Version Number . This will display the version in the status section. For example, 6.8.

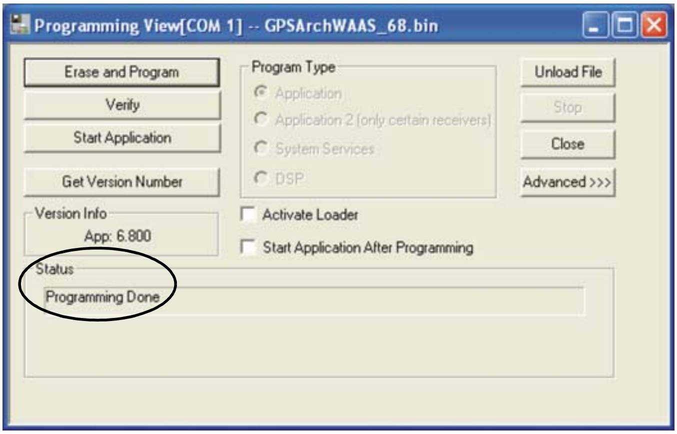


13. Cycle the power on the receiver. The Active Loader check mark will go away. The status window will state “Active Loader Received.”



The receiver is then in receiver mode.

14. The status window will display “Programing Done” when the unit is finished programming.



The application will now be loaded onto receiver.

Subscription Codes

This section covers the following:

- Finding the serial number and inputting a subscription code (e-Dif, L-Dif (base and rover), RTK, 20 Hz or 10Hz, etc.) into a Hemisphere GPS receiver
- Viewing the status and interpreting the $JI subscription date codes

Subscribing to an Application

These instructions explain how to activate an application code on a Hemisphere GPS receiver.

Requirements

- A serial communication cable must be used to connect the Hemisphere GPS receiver to the serial COM port on the computer
- Download SLXmon from the Hemisphere GPS web site or use a generic terminal program such as MS Windows HyperTerminal
- The application in which to subscribe must loaded onto the Hemisphere GPS receiver. See "Installing Applications onto the Receiver" on page 52.
- The application subscription code must be purchased from Hemisphere GPS or an authorized Hemisphere GPS representative

To activate the application on a Hemisphere GPS receiver:

1. Connect the Hemisphere GPS receiver to the serial COM port on the computer.

2. Run the SLXmon program on the computer.

3. Select **File > Connect** to open communication with the receiver (select appropriate COM port and Baud rate).
4. Select **Control > View Command Page** to open the command window.
5. Type the following command in the MESSAGE window:

   $JAPP
6. Confirm what applications are loaded onto the receiver and the order in which they appear. One example of a line in the response list is:

   $>JAPP,WAAS,DIFF

   In that example, WASS (SBAS, EGNOS, MSAS, etc) is the number one application (or application number 1) and DIFF, which is the same as e-Dif, is the “other” application (or application number 2). Use the following command to switch the applications:

   $JAAP,O
7. If DIFF is listed as application number 2 in the $JAPP response then type the following command in the message window:

   $JAPP,O

   “O” stands for “Other” in the example. This will swap the two applications so that DIFF will be the current application
8. Type the following command in the MESSAGE window:

   $JI

9. The first number in the response is the serial number of the receiver. An example of the response is:

   $>JI,810133,1,3,09031998,01/06/1998,12/31/2018,3.5,31

   The serial number is 810133. Write down that serial number and provide it to Hemisphere GPS with the request for an e-Dif subscription code

10. Type the following command in the MESSAGE window after receiving the subscription code from Hemisphere GPS:

    $JK,nnnn

    "nnnn" is the subscription number in the example. The receiver will respond with "subscription accepted."

e-Dif is now loaded as the current application and is ready for use.

Interpreting the $JI Subscription Date Codes

This section provides information on interpreting the $JI subscription date codes. An example of the $JI subscription date code is listed below. The date code is in bold and underlined text.

$>JI,12838,1,7,26022003,01/01/1900,01/01/**<u>3000</u>**,6.8Hx,38

The date code means different things depending on whether an SX-1 or Crescent receiver is queried. See Table 3-1, on page 61, to Table 3-3, on page 62, to determine the receiver type. The date codes can be used to gain a quick understanding of what subscription codes were applied to the receiver.

Table 3-1: $JT Response

Receiver	**$JT response**
SX-1	SX1x
Crescent	SX2x
SLX2	SLXx
SLX3	SX1x

Note: “x” represents the receiver type. For example, in SX2i, “i” represents e-Dif. See Table 3-2, on page 62, for a list of receiver response and the $JT reply.

Table 3-2: $JT Response and Application

$JT response	$JT reply		
	Plessey	GP4020	xScale
RTK rover	None	None	SX2r
RTX base	None	None	SX2b
e-Dif	SLXi	SX1i	SX2i
Attitude slave	None	None	SX2a
Attitude master	None	None	SX2a
OmniSTAR	SLXg	None	SX2g
WAAS	SLXg	SX1g	SX2g
Stand alone	SLXg	SX1g	SX2g
Vector OEM	None	None	SX2a

Table 3-3: SX-1 and SLX HEX Example

Date Code	Hexadecimal	Response
3000	HEX 0	1 Hz SBAS enabled
3001	HEX 1	5 Hz SBAS enabled
3002	HEX 2	1 Hz SBAS/e-Dif enabled
3003	HEX 3	5 Hz SBAS/e-Dif enabled

Table 3-3: Crescent Receiver Data Code

Date Code	Response
3000	10 Hz SBAS enabled
3001	20 Hz SBAS enabled
3002	10 Hz SBAS/e-Dif enabled
3003	20 Hz SBAS/e-Dif enabled
3004	10 Hz SBAS/RTK enabled
3005	20 Hz SBAS/RTK enabled
3006	10 Hz SBAS/e-Dif/RTK enabled
3007	20 Hz SBAS/e-Dif/RTK enabled
3008	10 Hz SBAS/L-Dif enabled
3009	20 Hz SBAS/L-Dif enabled
3010	10 Hz SBAS/e-Dif/L-Dif enabled
3011	20 Hz SBAS/e-Dif/L-Dif enabled
3012	10 Hz SBAS/RTK/L-Dif enabled
3013	20 Hz SBAS/RTK/L-Dif enabled
3014	10 Hz SBAS/e-Dif/RTK/L-Dif enabled
3015	10 Hz SBAS/e-Dif/RTK/L-Dif enabled

The date code can be used to gain a quick understanding of what subscription codes have been applied to the receiver.

How the date code section of the $JI query relates to the date code found in the $JK query

Here are some examples. The date code is in bold and highlighted text.

$JI query date code example:

$>JI,311077,1,7,04102005,01/01/1900,01/01/**3000**,6.8Hx,46

$JK date code example:

$>JK,01/01/**3000,0,(1,2 or no number)**

Note: In the $JK examples, the second to last digit on the right in the date code is the hex value. The last digit to the right acts as the value of output rate in Hertz. If 1 or 2 does not appear, then the output rate is at the default of 10 Hz.

The date codes are identical in either query and are directly related to each other. The last digit in the $JK query is the hexadecimal equivalent of the last two digits in the date code. To better illustrate this, here is another example. The date code is in bold and underlined text.

$JI query date code example:

$>JI,311077,1,7,04102005,01/01/1900,01/01/**3015**,6.8Hx,46

$JK date code example:

$>JK,01/01/**3015,F**

In this example, the date code is showing 15 in the last two digits. Therefore, the Hex number following the date code in the $JK query is F.

Table 3-1, on page 61, and Table 4, on page 65, identifies what codes have been applied to the GPS receiver by looking at one of two queries.

Table 3-4: Crescent HEX Example

Date Code	Hexadecimal	Response
3000	HEX 0	10 Hz SBAS enabled
3001	HEX 1	20 Hz SBAS enabled
3002	HEX 2	10 Hz SBAS/e-Dif enabled
3003	HEX 3	20 Hz SBAS/e-Dif enabled
3004	HEX 4	10 Hz SBAS/RTK enabled
3005	HEX 5	20 Hz SBAS/RTK enabled
3006	HEX 6	10 Hz SBAS/e-Dif/RTK enabled
3007	HEX 7	20 Hz SBAS/e-Dif/RTK enabled
3008	HEX 8	10 Hz SBAS/L-Dif enabled
3009	HEX 9	20 Hz SBAS/L-Dif enabled
3010	HEX A	10 Hz SBAS/e-Dif/L-Dif enabled
3011	HEX B	20 Hz SBAS/e-Dif/L-Dif enabled
3012	HEX C	10 Hz SBAS/RTK/L-Dif enabled
3013	HEX D	20 Hz SBAS/RTK/L-Dif enabled
3014	HEX E	10 Hz SBAS/e-Dif/RTK/L-Dif enabled
3015	HEX F	10 Hz SBAS/e-Dif/RTK/L-Dif enabled

This identifies what codes have been applied to the GPS receiver by looking at one of two queries.

Note: This addition to the end of the $JK response indicates a receiver that has had a code applied to it to downgrade its maximum output capabilities to 1 Hz.

Configuring the Data Message Output

The receiver features three primary bi-directional ports referred to as A, B, and C, in addition to its differential-only Port D. GPS data messages for all three ports are easily configured by sending NMEA 0183 commands to the receiver module through all of its communication ports. The output of Port B can be configured through A, for instance, and vice versa. The $JASC NMEA message, discussed in detail in Chapter 5, allows the messages to be turned on or off as required.

This Port and The Other Port

The NMEA 0183 interface for Port A and B both use “This” and “Other” terminology. When interfacing to either port for the sake of turning data messages on or off, on that same port, the port is referred to as “This” port. To turn a data message on or off, on the opposite port which is being communicated with, the opposite port is referred to as the “Other” port.

For example, when communicating with the receiver Port B, to turn the GPGGA message on at an update rate of 5 Hz on Port A, the following command would be used:

$JASC,GPGGA,5,OTHER<CR><LF>

To turn the GPGGA message on at 5 Hz on Port B, the following command would be issued:

$JASC,GPGGA,5<CR><LF>

When turning a message on or off on “This” port, “This” at the end of the message does not need to be indicated. In contrast, when turning messages on or off on Port C from Port A, or Port B, the following procedure must be used. For example, when communicating with the receiver on Port A and turn on the GLL NMEA 0183 message at 10 Hz on Port C needs to be turned on, the following would be used:

$JASC,GPGLL,10,PORTC<CR><LF>

As with Port A and B, when communicating directly with Port C, nothing needs to be indicated at the end of the message. Consult Chapter 5 for more information on NMEA 0183 messages.

Saving the Receiver Configuration

Each time that the configuration of the receiver is changed, the new configuration should be saved so the receiver does not have to be reconsidered again for the next power cycle.

To save the settings:

1. Issue the $JSAVE command. The receiver will record the current configuration to non-volatile memory. The receiver will indicate when the save process has been completed, which can take approximately five seconds.

Using Port D for RTCM Input

The receiver has a port that has been designed to accommodate externally supplied corrections input according to the RTCM SC-104 protocol. Port D provides this functionality, although it has been fixed to operate at a baud rate of 9600 (8 data bits, no parity, and 1 stop bit – 8-N-1).

To use Port D of the receiver for correction input, receiver must be set to operate in beacon differential mode using the following command:

$JDIFF,BEACON<CR><LF>

This command was designed to “turn on” Port D differential operation in our products, since many use the Hemisphere GPS SBX beacon module, interfaced to Port D.

Although the following RTCM SC-104 message types do not all contain differential data, the receiver is compatible with them.

- Type 1
- Type 2
- Type 3
- Type 5
- Type 6
- Type 7
- Type 9
- Type 16

To return to using SBAS as the correction source, send the following command to the receiver:

$JDIFF,WAAS<CR><LF>

Detailed information on NMEA 0183 messages supported by the receiver is found in Chapter 5.

4: PocketMAX Utility

PocketMAX™

PocketMAX PC and PocketMAX (a PDA version) are freely available utilities designed for use with Hemisphere GPS products. Since these utilities were not designed specifically for one receiver alone, they support features not offered by the receiver, such as tracking of the OmniSTAR differential service and display of our Vector product's true heading. However, the interface may be used for all I/O operations. PocketMAX PC runs on any PC with Windows 95, 98, or NT 4.0+ (Windows 2000 and Windows XP). Screen resolution of 800x600 or greater is recommended. One of the receiver's serial ports must be connected to a COM port on the computer. The current version of PocketMAX PC, or PocketMAX, can be downloaded from the Hemisphere GPS website.

Figure 4-1 is an example screen capture from the PocketMAX.

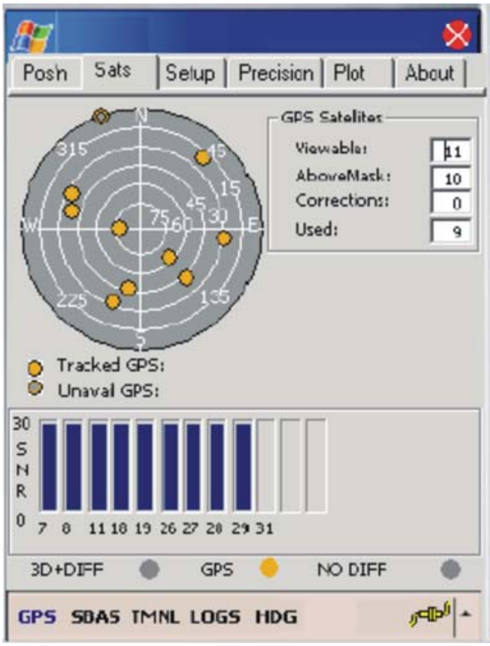


Figure 4-1. PocketMAX PC screen capture

5: NMEA 0183 Commands and Messages

The receiver supports a selection of NMEA 0183 and proprietary binary messages. This chapter identifies the selection of standard and proprietary NMEA 0183 messages for the receiver. Chapter 6 describes the binary software interface in detail.

It is the user choice as a systems designer to choose whether or not to support a NMEA 0183-only software interface or a selection of both NMEA 0183 and binary messages. The receiver is configured only with NMEA 0183 commands. Three NMEA 2000 commands are provided and are described in Chapter 6.

NMEA 0183 Message Elements

NMEA 0183 messages have a common structure, consisting of a message header, data fields, checksum, and carriage return/line feed message terminator. An example of an NMEA 0183 sentence is as follows:

$XXYYY,zzz,zzz,zzz...*XX<CR><LF>

The components of this generic NMEA 0183 message example are displayed in Table 5-1.

Table 5-1: NMEA Message Elements

Element	Description
$	Message header character
XX	NMEA 0183 Talker field. GP indicates a GPS talker
YYY	Type of GPS NMEA 0183 Message
zzz	Variable length message fields
*xx	Checksum
<CR>	Carriage return
<LF>	Line feed

Null, or empty fields, occur when no information is available for that field.

The $JNP command can be used to specify the number of decimal places output in the GGS and GLL messages. See page 105 for more information.

General Commands

This section provides the various commands related to the general operation and configuration of the receiver. Table 5-2 provides the general commands' messages and descriptions.

Table 5-2: General Commands

Message	Description
$JASC,Dx	Command to turn on diagnostic information
$JAIR	Command to place the receiver into "AIR" mode, where the receiver will respond better to the high dynamics associated with airborne applications
$JASC,VIRTUAL	Command used to output RTCM data fed into the other port, through the current port
$JASC,RTCM	Command used to output RTCM data, from the SBAS demodular
$JALT	Command used to set the altitude aiding mode of the receiver
$JAPP	Command used to query the current application and also choose the current application
$JBAUD	Baud rate change or query command
$JCONN	Virtual circuit command used to interface to the internal beacon or communicate with the menu system microprocessor
$JDIFF	Command used to set or query the differential mode
$JK	Command used to subscribe certain features of use of the receiver

Table 5-2: General Commands

Message	Description
$JPOS	Command used to provide the receiver with a seed position to acquire a SBAS signal more quickly upon startup This is not normally needed
$JQUERY,GUIDE	Command used to poll the receiver for it's opinion on whether or not it is providing suitable accuracy after both SBAS and GPS have been acquired (up to 5 min)
$JRESET	Command used to reset the configuration of the receiver
$JSAVE	Command used to save the configuration of the receiver
$JSHOW	Command used to query the receiver for it's configuration
$JT	Command used to poll the receiver for it's receiver type
$JBIN	Command used to turn on the various binary messages supported by the receiver
$JI	Command used to get information from the receiver, such as it's serial number and firmware version information

The following sections provide detailed information relating to the use of each command.

Note: Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JASC,D1

This command allows the output of the RD1 diagnostic information message from the receiver to be adjusted.

This command has the following structure:

$JASC,D1,R[,OTHER]<CR><LF>

Currently, only the RD1 message is currently defined, with x = 1. The message status variable “R” may be one of the following values:

R	Description
0	OFF
1	ON

When the “,OTHER” data field (without the brackets) is specified, this command will enact a change in the RD1 message on the other port.

$JAIR

This command allows the primary GPS engine to be placed within the receiver into AIR mode HIGH, where the receiver is optimized for the high dynamic environment associated with airborne platforms. JAIR defaults to normal (NORM) and this setting is recommended for most applications. The AUTO option allows the receiver to decide when to turn JAIR on high. Turning AIR mode on to HIGH is not recommended for Crescent Vector operation.

$JAIR,NORM ==> normal track and nav filter bandwidth

$JAIR,HIGH ==> highest track and nav filter bandwidth

$JAIR,LOW ==> lowest track and nav filter bandwidth

$JAIR,AUTO ==> default track and nav filter bandwidth, usually the same as normal, but automatically goes to HIGH above 30 m/sec.

On "HIGH" setting, larger "sudden drops in SNR" are tolerated before observation data is discarded from the Navigation solution. This may be beneficial when an aircraft is rapidly banking (e.g., crop-duster) and, hence, the GPS signal rapidly transitions its entry into the antenna from area of high-antenna-gain pattern to that of low antenna gain.

The format of this command follows:

$JAIR,R<CR><LF>

Where feature status variable “R” may be one of the following values:

R	Description
0 or NORM	NORM
1 or HIGH	HIGH
2 or LOW	LOW
3 or AUTO	NORM (AUTO)

The receiver will reply with the following response:

$>JAIR,MAN,NORM

$>JAIR,MAN,HIGH

$>JAIR,MAN,LOW

$>JAIR,AUTO,NORM

$JASC, VIRTUAL

When using an external correction source, this command is used to "daisy chain" RTCM data from being input from one port and output through the other. For example, if RTCM is input on Port B, this data will correct the receiver position and also be output through Port A. The receiver acts as a pass-through for the RTCM data. Either port may be configured to accept RTCM data input. This command then allows the opposite port to output the RTCM data.

To configure the receiver to output RTCM data on the current port from data input on the other port, issue the following command:

$JASC,VIRTUAL,R<CR><LF>

To configure the receiver to output RTCM data on the other port from RTCM data input on the current port, issue the following command:

$JASC,VIRTUAL,R,OTHER<CR><LF>

Where the message status variable "R" may be one of the following:

R	Description
0	OFF
1	ON

The receiver will reply with the following response:

$>

$JALT

This command turns altitude aiding on or off for the receiver module. When set to on, altitude aiding uses a fixed altitude instead of using one satellite's observations to calculate the altitude. The advantage of this feature, when operating in an application where a fixed altitude is acceptable, is that the extra satellite's observations can be used to the betterment of the latitude, longitude, and time offset calculations, resulting in improved accuracy and integrity. Marine markets, for example, may be well suited for use of this feature.

This command has the following layout:

$JALT,c,v[,GEOID] <CR><LF>

Where feature status variable "C" and threshold variable "V" may be one of the following:

c	Description
NEVER	This is the default mode of operation where altitude aiding is not used.
SOMETIMES	Setting this feature to SOMETIMES allows the receiver to use altitude aiding, depending upon the PDOD threshold, specified by "V."
ALWAYS	Setting this feature to ALWAYS allows the receiver to use altitude aiding regardless of a variable. In this case, the ellipsoidal altitude, "V," that the receiver should use may be specified.

The receiver will reply with the following response:

$>

If the antenna is at a constant height, then altitude aiding should help with accuracy. Using a DGPS position, average the height over a period of time. The longer the time period, the more accurate this height value will be. Then take the average height and issue the following command:

$JALT,ALWAYS,h

Where “h” is the ellipsoid height. If the height reported from the GGA message is being used, this is actually geoidal and not ellipsoidal height. In this case, the following command needs to be issued:

$JALT,ALWAYS,h,GEOID

$JLIMIT

This command is used to change the threshold of estimated horizontal performance for which the DGPS position LED is illuminated. The default value for this parameter is a conservative 10.0 meters. This command has the following format:

$JLIMIT,LIMIT<CR><LF>

Where “LIMIT” is the new limit in meters.

The receiver will respond with the following message:

$>

$JAPP

This command requests the receiver for the currently installed applications and to choose which application to use. The receiver, by default, comes pre-installed with WAAS (SBAS) in application slot 1 and a second application, e-Dif, in application slot 2. An activation code from Hemisphere GPS must be purchased use e-Dif.

To poll the receiver for the current applications, send the following message:

$JAPP<CR><LF>

There are no data fields to specify in this message. The receiver will respond with the following message:

$>JAPP,CURRENT,OTHER,[1 OR 2],[2 OR 1]

Where "CURRENT" indicates the current application in use and "OTHER" indicates the secondary application that is not currently in use. 1 and 2 indicate which application slot is currently being used.

Available applications are as follows:

Application
WAAS
AUTODIFF
LOCDIF (local differential rover)
RTKBAS (local differential base)

For the sake of the application names, the SBAS application is referred to as WAAS by the receiver's internal software. For example, if the response to $JAPP<CR><LF> is $>JAPP,WAAS,AUTODIFF,1,2 indicating that WAAS (SBAS) is in application slot 1, e-Dif is in application slot 2, and that WAAS in application slot 1 is currently being used. To change from the current application to the other application, when two applications are present, issue the following command:

$JAPP,OTHER<CR><LF>

Or

$JAPP,APP<CR><LF>

Where “APP” may be one of the following by name:

Application	**Description**
WAAS	This will change to the SBAS application.
AUTODIFF	This will change to the e-Dif application, referred to as “autodiff” in the firmware.
LOCDIF	This will change to the local differential rover application.
RTKBAS	This will change to the local differential base application.

If the $JAPP,OTHER<CR><LF> command is issued on a receiver, continuing with the above example, the response to $JAPP<CR><LF> would then be $>JAPP,AUTODIFF,WAAS,2,1, indicating that application slot 2, containing e-Dif, is currently being used.

Note: Other derivatives of the $JAPP command are the $JAPP,1<CR><LF> and $JAPP,2<CR><LF> commands that can be used to set the receiver to use the first and second application. It is best to follow up the sending of these commands with a $JAPP query to see which application is 1 or 2. These two commands are best used when upgrading the firmware inside the receiver, as the firmware upgrading utility uses the application number to designate which application to overwrite.

$JBAUD

This command is used to configure the baud rates of the receiver.

This command has the following structure:

$JBAUD,R[,OTHER] <CR><LF>

Where "R" may be one of the following baud rates:

Baud rate
4800
9600
19200
38400
57600

When this command has been issued without the ",OTHER" data field (without the brackets), the baud rate of the current port will be changed accordingly. When the ",OTHER" data field is specified (without the brackets), a baud rate change will occur for the other port.

The receiver will reply with the following response:

$>

$JCONN

This command is used to create a virtual circuit between the A and B port, if needed. This allows communication through the receiver to the device on the opposite port.

The virtual circuit command has the following form:

$JCONN,P<CR><LF>

Where the connection type, "P," may be on o f the following:

P	Description
AB	Specify "AB" in order to connect the A port to the B port
X	Once a virtual circuit has been established, to remove the virtual circuit, specify "X" in this command to return the current port to normal
C	Specify "C" in order to communicate directly to the optional SBX beacon receiver

$JDIFF

This command is used to change the differential mode of the receiver module.

The structure of this command is as follows:

$JDIFF,DIFF<CR><LF>

Where the differential mode variable “DIFF” has one of the following values:

DIFF	Description
OTHER	Specifying OTHER instructs the receiver to use external corrections input through the opposite port that is communicating.
THIS	THIS instructs the receiver to use external corrections input through the same port that is communicating.
BEACON	Specifying BEACON instructs the receiver to use RTCM corrections entering Port C at a fixed rate of 9600 baud. This input does not have to be from a beacon receiver, such as SBX. However, this is a common source of corrections.
WAAS	Specifying WAAS instructs the receiver to use SBAS. This is also the response when running the local dif application as the base.
RTK	This is the response when running the local dif application for the rover.
X	Specifying X instructs the receiver to use e-Dif mode (the receiver will respond back with $JDIFF,AUTO to a $JDIFF query).
NONE	In order for the receiver to operate in autonomous mode, the NONE argument may be specified in this command.

$JK

This command is used to subscribe the receiver to various options, such as higher update rates, e-Dif (or base station capability) or L-Dif. This command will have the following format:

$JK,X…<CR><LF>

Where "X..." is the subscription key provided by Hemisphere GPS and is 10 characters in length.

If the $JK command is sent without a subscription key, as follows, it will return the expiry date of the subscription.

$JK<CR><LF>

Reply:

$>JK,12/31/2003,1

$JPOS

This command is used to speed up the initial acquisition when changing continents with the receiver. For example, powering the receiver for the first time in Europe after it has been tested in Canada. This will allow the receiver to begin the acquisition process for the closest SBAS spot beams. This will save some time with acquisition of the SBAS service. However, use of this message is typically not required due to the quick overall startup time of the receiver module.

This command has the following layout:

$JPOS,LAT,LON<CR><LF>

Where “LAT” and “LON” have the following requirements:

Position component	description
LAT	Latitude component must be entered in decimal degrees. This component does not have to be more accurate than half a degree.
LON	Longitude component must be entered in decimal degrees. This component does not have to be more accurate than half a degree.

Note: This command is not normally required for operation of the receiver module.

$JQUERY,GUIDE

This command is used to poll the receiver for its opinion on whether or not it is providing suitable accuracy after the both SBAS and GPS have been acquired (up to 5 min). This feature takes into consideration the download status of the SBAS ionospheric map and also the carrier phase smoothing of the unit.

This command has the following format:

$JQUERY,GUIDE<CR><LF>

If the receiver is ready for use with navigation, or positioning with optimum performance, it will return the following message:

$>JQUERY,GUIDE,YES<CR><LF>

Otherwise, it will return the following message:

$>JQUERY,GUIDE,NO<CR><LF>

$JRESET

This command is used to reset the receiver to its default operating parameters. The $JRESET command does the following:

- Turn off outputs on all ports
- Save the configuration
- Set the configuration to its defaults (refer to Table 5-3)

Table 5-3: Default Configuration

Configuration	Setting
Elev Mask	5
Residual limit	10
Alt aiding	None
Age of Diff	45 minutes
Air mode	Auto
Diff type	Default for app
NMEA precision	5 decimals
COG smoothing	None
speed smoothing	None
WAAS	UERE thresholds

This message has the following format:

$JRESET<CR><LF>

$JRESET,ALL does everything $JRESET does, plus it clears almanacs.

$JRESET,BOOT does everything $JRESET,ALL does, plus it does the following:

- Clears use of the Real-Time clock at startup
- Clears use of backed-up ephemeris and almanacs
- Re-boots the receiver when done

$JSAVE

Sending this command is required after making changes to the operating mode of the receiver module.

This command has the following structure:

$JSAVE<CR><LF>

The receiver will reply with the following two messages. Ensure that the receiver indicates that the save process is complete before turning the receiver off or changing the configuration further.

$> SAVING CONFIGURATION. PLEASE WAIT...
$> Save Complete

No data fields are required. The receiver will indicate that the configuration is being saved and will indicate when the save is complete.

$JSHOW

This command is used to poll the receiver for its current operating configuration.

This command has the following structure:

$JSHOW[,SUBSET] <CR><LF>

Using the $JSHOW command without the optional "subset" field will provide a complete response from the receiver. An example of this response follows:

```
$>JSHOW,BAUD,9600            (1)
$>JSHOW,BAUD,9600,OTHER      (2)
$>JSHOW,BAUD,9600,PORTC      (3)
$>JSHOW,ASC,GPGGA,1.0,OTHER  (4)
$>JSHOW,ASC,GPVTG,1.0,OTHER  (5)
$>JSHOW,ASC,GPGSV,1.0,OTHER  (6)
$>JSHOW,ASC,GPGST,1.0,OTHER  (7)
$>JSHOW,ASC,D1,1,OTHER       (8)
$>JSHOW,DIFF,WAAS            (9)
$>JSHOW,ALT,NEVER            (10)
$>JSHOW,LIMIT,10.0           (11)
$>JSHOW,MASK,5               (12)
$>JSHOW,POS,51.0,-114.0      (13)
$>JSHOW,AIR,AUTO,OFF         (14)
$>JSHOW,FREQ,1575.4200,250   (15)
$>JSHOW,AGE,1800             (16)
```

This example response is summarized in the following table:

Line	Description
1	This line indicates that the current port is set to a baud rate of 9600.
2	This line indicates that the other port is set to a baud rate of 9600.

Line	Description
3	This line indicates that Port C is set to a baud rate of 9600. (Port C is not usually connected externally on the finished product.)
4	This line indicates that GPGGA is output at a rate of 1 Hz from the other port.
5	This line indicates that GPVTG is output at a rate of 1 Hz from the other port.
6	This line indicates that GPGSV is output at a rate of 1 Hz from the other port.
7	This line indicates that GPGST is output at a rate of 1 Hz from the other port.
8	This line indicates that D1 is output at a rate of 1 Hz from the other.
9	This line indicates that the current differential mode is WAAS.
10	This line indicates the status of the altitude aiding feature.
11	The receiver does not support this feature.
12	This line indicates the elevation mask cutoff angle, in degrees.
13	This line indicates the current send position used for startup, in decimal degrees.
14	This line indicates the current status of the AIR mode.
15	This line indicates the current frequency of the L-band receiver.
16	This line indicates the current maximum acceptable differential age in seconds.

When issuing this command with the optional “subset” data field (without the brackets), a one-line response is provided. The subset field may be either CONF or GP.

When CONF is specified for “subset” (without the brackets), the following response is provided:

$>JSHOW,CONF,N,0.0,10.0,5,A,60W

This response is summarized in the following table:

Message component	**Description**
$JSHOW,CONF	Message header
N	“N” indicates no altitude aiding
0.0	“0.0” indicates the aiding value, if specified (either height or PDOP threshold)
10.0	Residual limit for the $JLIMIT command
5	Elevation mask cutoff angle, in degrees
A	AIR mode indication
60	Maximum acceptable age of correction data in seconds
W	Current differential mode, “W” indicates WAAS mode

When GP is specified for “subset,” the following is an example response provided:

$>JSHOW,GPGGA,1.0

This response will provide the >$JSHOW,GP message header, followed by each message currently being output through the current port and also the update rate for that message.

$JT

This command displays the type of receiver engine within the receiver and has the following format:

$JT<CR><LF>The receiver will return the following response, indicating that the receiver is an SX2g ("g" for global differential operation) when in SBAS mode and SX2i ("i" for internal differential operation) when in e-Dif mode:

$>JT,SX12

$JI

This command displays receiver information. It has the following format:

$JI<CR><LF>

The receiver will reply with the following message:

$>JI,11577,1,5,11102002,01/01/1900,01/01/3003,6.3,46

This command is summarized in the following table:

Message component	**Description**
11577	This field provides the serial number of the GPS engine
1	This field is the fleet number
5	This is the hardware version
11102002	This field is the production date code
01/01/1900	This field is the subscription begin date
1/01/3003	This field is the subscription expiration date
1.1	This field is the application software version number
46	This field is a place holder

$JBIN

This command allows the output of the various binary messages, most notably, the Bin95 and Bin96 messages to be requested. The latter two messages contain all information required for post processing.

This message has the following structure:

$JBIN,MSG,R

Where “MSG” is the message name and “R” is the message rate as shown in the following table:

MSG	R (Hz)	Description
Bin1	10, 2, 1, 0 or .2	Binary GPS position message
Bin2	10, 2, 1, 0 or .2	Binary message containing GPS DOP’s
Bin80	1 or 0	Binary message containing SBAS information
Bin95	1 or 0	Binary message containing ephemeris information
Bin96	10, 2, 1 or 0	Binary message containing code and carrier phase information
Bin97	10, 2, 1, 0 or .2	Binary message containing process statistics
Bin98	1 or 0	Binary message containing satellite and almanac information
Bin99	10, 2, 1, 0 or .2	Binary message containing GPS diagnostic information

The receiver will reply with the following information:

$>

Note: Higher update rates may be available with a subscription on Bin 1, 2, 96, 97 and 99.

GPS Commands

This section describes the selection of commands specific to the configuration and operation of the receiver's internal GPS engine.

Table 5-4 provides a brief description of the commands supported by the GPS engine for its configuration and operation.

Table 5-4: GPS Commands

Message	Description
$JASC,GP	This command is used to configure the NMEA 0183 message output of the GPS engine
$JAGE	A command used to configure the maximum age of DGPS corrections
$JOFF	This command is used to turn off all data output by the GPS engine
$JMASK	This command allows the cut-off angle for tracking of GPS satellites to be modified
$J4STRING	This command allows the GPS for output of the GPPGA, GPGSA, GPVTG, and GPZDA messages at a specific baud rate to be modified

The following subsections provide detailed information relating to the use of each command.

Note: Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JASC

This command allows the GPS data messages on at a particular update rate, to be turned on or off. When turning messages on, various update rates are available, depending on what the requirements are.

This command has the following layout:

$JASC,MSG,R[,OTHER]<CR><LF>

Where “MSG” is the name of the data message and “R” is the message rate, as shown in the table below. Sending the command without the optional ‚“OTHER” data field (without the brackets) will enact a change on the current port.

Sending a command with a zero value for the “R” field turns off a message.

MSG	R (Hz)	Description
GPGBS	20, 10, 2, 1, 0 or .2	Satellite fault detection used for RAIM
GPGGA	20, 10, 2, 1, 0 or .2	GPS fix data
GPGLL	20, 10, 2, 1, 0 or .2	Geographic position - latitude/longitude
GPGNS	20, 10, 2, 1, 0 or .2	GNSS fix data
GPGRS	20, 10, 2, 1, 0 or .2	GNSS range residuals
GPGSA	1 or 0	GNSS (Global Navigation Satellite System DOP (and active satellites))
GPGST	1 or 0	GNSS pseudorange error statistics
GPGSV	1 or 0	GNSS satellite in view
GPRMC	10, 2, 1, 0 or .2	Recommended minimum specific GNSS data
GPRRE	1 or 0	Range residual message

MSG	R (Hz)	Description
GPVTG	10, 2, 1, 0 or .2	Course over ground and ground speed
GPZDA	10, 2, 1, 0 or .2	Time and date

When the ",OTHER" data field (without the brackets) is specified, this command will enact a change on the other port.

The receiver will reply with the following response:

$>

$JAGE,AGE

This command allows the maximum allowable age for correction data to be chosen. The default setting for the receiver is 2700 seconds, however, this value may be changed if appropriate. Using COAST technology, the receiver is able to use old correction data for extended periods of time. If a maximum correction age older than 1800 seconds (30 minutes) is chosen, we recommend testing the receiver to ensure that the new setting meets the requirements as accuracy will slowly drift with increasing time.

This command has the following structure:

$JAGE,AGE<CR><LF>

Where maximum differential age time-out variable, "age" may be a value from 6 to 259200 seconds (6 seconds to 3 days).

The receiver will reply with the following response:

$>

$JOFF

This command allows all data messages being output through the current or other port, including any binary messages, such as Bin95 and Bin96 to be turned off.

This command has the following definition:

$JOFF[,OTHER]<CR><LF>

When the ",OTHER" data field (without the brackets) is specified, this command will turn off all messages on the other port. There are no variable data fields for this message. The receiver will reply with the following response:

$>

$JMASK

This command allows the elevation cutoff mask angle for the GPS engine to be changed. Any satellites below this mask angle will be ignored, even if available. The default angle is 5 degrees, as satellites available below this angle will have significant tropospheric refraction errors.

This message has the following format:

$JMASK,E<CR><LF>

Where the elevation mask cutoff angle "E" may be a value from 0 to 60 degrees.

The receiver will reply with the following response:

$>

$JNP

This command allows the user to specify the number of decimal places output in the GGA and GLL messages.

This command has the following definition:

$JNP,X<CR><LF>

Where "x" specifies the number of decimal places from 1 to 8. This command will affect both the GGA and the GLL messages.

$J4STRING

This command allows the GPGGA, GPVTG, GPGSA, and GPZDA messages to all be output with the issue of a single command. The output rate of each message is limited to 1 Hz. However, the baud rate of the current or other port may be changed at the same time.

This command has the following definition:

$J4STRING[,R][,OTHER] <CR><LF>

Where "R" may be one of the following baud rates:

Baud rate
4800
9600
19200
38400
57600

When the "OTHER" data field (without the brackets) is specified, this command will turn on the four NMEA 0183 messages on the other port.

The receiver will reply with the following response:

$>

$JSMOOTH

There is a new command, $JSMOOTH, that enables the user to change the carrier smoothing interval. This command was designed to offer the user flexibility for tuning in different environments. A slight improvement in positioning performance using either the short or long smoothing interval, depending on the multipath environment, may occur. The default for this command is 900 seconds (15 minutes) or LONG.

To change the smoothing interval to 300 seconds (5 minutes), or SHORT, use the following command:

$JSMOOTH,SHORT<CR><LF>

To change the smoothing interval to 900 seconds (15 minutes), or LONG, use the following command:

$JSMOOTH,LONG<CR><LF>

This command can also be entered using the number of seconds desired for smoothing. The limits are from 15 seconds to 6000 seconds (100 minutes). Use the following command, where "X" is the number of seconds used for the carrier smoothing interval:

$JSMOOTH,x<CR><LF>

To request the status of this message, send the following command. It will return the word SHORT or LONG as well as the number of seconds used. The status of this command is also output in the $JSHOW message:

$JSMOOTH<CR><LF>

Note: It is best to be conservative and leave it at the default setting of LONG (900 seconds) if unsure of the best value for this setting.

$JTAU,SPEED

The speed time constant allows the user to adjust the level of responsiveness of the speed measurement provided in the $GPVTG message. The default value of this parameter is 0.0 seconds of smoothing. Increasing the time constant will increase the level of speed measurement smoothing. The following command is used to adjust the speed time constant.

$JTAU,SPEED,TAU<CR><LF>

Where “TAU” is the new time constant that falls within the range of 0.0 to 200.0 seconds. The setting of this value depends upon the expected dynamics of the receiver. If the receiver will be in a highly dynamic environment, this value should be set to a lower value, since the filtering window would be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, this value can be increased to reduce measurement noise. The following formula provides some guidance on how to set this value initially, however, we recommend testing how the revised value works in practice. It is best to be conservative and leave it at the default setting of LONG (900 seconds) if unsure of the best value for this setting.

TAU (IN SECONDS) = 10 / MAXIMUM ACCELERATION (IN M/S$_2$)

The receiver may be queried for the current speed time constant by issuing the same command without an argument:

$JTAU,SPEED<CR><LF>

Note: It is best to be conservative and leave it at the default setting of LONG (900 seconds) if unsure of the best value for this setting.

SBAS Commands

This section details the NMEA 0183 messages accepted by the internal SBAS engine of the receiver.

Table 5-5 provides a brief description of the command supported by the SBAS demodulator for its control and operation.

Table 5-5: SBAS Commands

Message	Description
$JWAASPRN	This message is used to reconfigure the SBAS PRN numbers for use with different WAAS PRNs as well as other Space Based Augmentation Systems (such as EGNOS and MSAS)
$JGEO	This command is used to poll the SBAS demodulator for information relating to the current location and SBAS satellites
$JRD1	This command is used to poll the receiver for the SBAS diagnostic information
$JASC,RTCM	This feature allows the receiver to be configured to output RTCM data from the SBAS demodulator

The following subsections provide detailed information relating to the use of each command.

Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JWAASPRN

This command allows the receiver to be polled for the SBAS PRN's in memory, and change them, if desired.

To poll the receiver for the current SBAS PRN's, send the following message:

$JWAASPRN<CR><LF>

There are no data fields to specify in this message. The receiver will respond with the following message:

$>JWAASPRN,PRN1,PRN2

Where "PRN1" indicates the first PRN number and "PRN2" indicates the second PRN number. The PRN numbers for WAAS are 122, 134, 135 and 138. WAAS PRNs 135 and 138 are currently in test mode. EGNOS is currently using PRN 120, but also has PRN 131. To manually change the current PRN numbers, the following message should be used:

$JWAASPRN[,SV1[,SV2]] <CR><LF>

Where "SV1" is the PRN number of the first SBAS satellite and "SV2" is the PRN number of the second SBAS satellite. "sv1" or both "SV1" and "SV2" may be specified. The receiver will reply with the following response:

$>

To return the unit to automated SBAS tracking, the following command should be sent to the receiver:

$JWAASPRN,,,AUTO <CR><LF>

The receiver will reply with the following response:

$>

$JGEO

This message is used to display information related to the current frequency of SBAS and its location in relation to the receiver's antenna. To query the receiver for the currently used SBAS satellite information, use the following query:

$JGEO<CR><LF>

The receiver will respond with the following data message:

$>JGEO,Sent=1575.4200,Used=1575.4200,PRN=prn,Lon=lon,El=ele,AZ=az

This message response is summarized in the following table:

Data field	**Description**
$>JGEO	Message header
Sent=1575.4200	Frequency sent to the digital signal processor
Used=1575.4200	Frequency currently used by the digital signal processor
PRN=prn	WAAS satellite PRN number
Lon=-lon	Longitude of the satellite
El=ele	Elevation angle from the receiver antenna to the WAAS satellite, reference to the horizon
AZ=az	Azimuth from the receiver antenna to the WAAS satellite, reference to the horizon

To monitor this information for dual SBAS satellites, add the ",ALL" variable to the $JGEO message as follows:

$JGEO[,ALL] <CR><LF>

This will result in the following output message:

$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=122,LON=-54,EL=9.7,AZ=114.0

$>JGEO,SENT=1575.4200,USED=1575.4200,PRN=134,LON=178,EL=5.0,AZ=252.6

As can be seen from this output, the first message is identical to the output from the $JGEO query. However, the second message provides information on the WAAS satellite not being currently used. Both outputs follow the format in the previous table for the $JGEO query.

$JRD1

This command is used to request diagnostic information from the receiver module.

To command the receiver to output the diagnostic information message for the currently used SBAS satellites at a rate of 1 Hz, use the following query:

$JASC,D1,1[,OTHER]<CR><LF>

The receiver will respond with the following data message:

$>

Setting the update rate to zero, as follows, will turn off this message:

$JASC,D1,0<CR><LF>

$JASC,RTCM

This command allows for configuration of the receiver to output RTCM corrections from SBAS, or beacon, through either receiver serial port. The correction data output is RTCM SC-104, even though SBAS uses a different over-the-air protocol (RTCA).

To have the receiver unit output RTCM corrections, send the following command to the smart antenna:

$JASC,RTCM,R[,OTHER]<CR><LF>

The message status variable “R” may be one of the following values:

R	Description
0	OFF
1	ON

When the ‘,OTHER” data field (without the brackets) is specified, this command will turn RTCM data on or off on the other port.

The receiver will reply with the following response:

$>

e-Dif Commands

This section provides information related to the NMEA 0183 messages accepted by the receiver's e-Dif application.

Table 5-6 provides a brief description of the commands supported by the e-Dif application for its control and operation.

Table 5-6: e-Dif Commands

Message	Description
$JRAD,1	This command is used to display the current reference position
$JRAD,1,P	Store present position as reference
$JRAD,1,lat,lon, height	Store entered position as reference
$JRAD,2	Use reference position as base
$JRAD,3	Use current position as base

Note: Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JRAD,1

This command is used to display the current reference position.

This command has the following format:

$JRAD,1<CR><LF>

The receiver will reply with a response similar to the following:

$>JRAD,1,51.00233513,-114.08232345,1050.212

Upon startup of the receiver with the e-Dif application running, as opposed to the SBAS application, no reference position will be present in memory. If attempting to query for the reference position, the receiver will respond with the following message:

$>JRAD,1,FAILED,PRESENT LOCATION NOT STABLE

$JRAD,1,P

This command records the current position as the reference with which to compute e-Dif corrections. This would be used in relative mode, as no absolute point information is specified.

This command has the following format:

$JRAD,1,P<CR><LF>

The receiver will reply with the following response:

$>JRAD,1,OK

$JRAD,1,LAT,LON,HEIGHT

This command is a derivative of the $JRAD,1,P command and is used when absolute positioning is desired.

This command has the following layout:

$JRAD,1,LAT,LON,HEIGHT<CR><LF>

The data fields in this command are described in the following table:

Data field	Description
lat	This is the latitude of the reference point in degrees decimal degrees.
lon	This is the longitude of the reference point in degrees decimal degree.
height	This is the ellipsoidal height of the reference point in meters. Ellipsoidal height can be calculated by adding the altitude and the geiodal separation, both available from the GGA sentence. (See example below.)

Example of ellipsoidal height calculation -

$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4,1071.0,M,-17.8,M,6.0, 0122*48

ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

The receiver will reply with the following response:

$>JRAD,LAT,LON,HEIGHT

Note: Both latitude and longitude must be entered as decimal degrees. The receiver will not accept the command if there are no decimal places

$JRAD,2

This command is used to force the receiver to used the new reference point. This command is normally used following a $JRAD,1 type command.

This command has the following format:

$JRAD,2<CR><LF>

The receiver will reply with the following response:

$>JRAD,2,OK

$JRAD,3

This command is used for two primary purposes. The first purpose is to invoke the e-Dif function once the unit has started up with the e-Dif application active. The second purpose is to update the e-Dif solution (calibration) using the current position as opposed to the reference position used by the $JRAD,2 command.

This command has the following format:

$JRAD,3<CR><LF>

The receiver will respond with the following command if it has tracked enough satellites for a long enough period before sending the command. This period of time can be from 3 to 10 minutes long and is used for modeling errors going forward.

$>JRAD,3,OK<CR><LF>

If the e-Dif algorithms do not find that there has been sufficient data collected, the receiver will send the following response:

$>JRAD,3,FAILED,NOT ENOUGH STABLE SATELLITE TRACKS

If the failure message is received after a few minutes of operation, try again shortly later until the “OK” acknowledgement message is sent. The e-Dif application will begin operating as soon as the $JRAD,3,OK message has been sent, however, a reference position for e-Dif will still need to be defined, unless relative positioning is sufficient for any needs.

Crescent Vector Commands

This section details the various settings that relate to the GPS heading aspect of the Crescent Vector OEM heading system. For a comprehensive list of all commands that can be used with the Crescent Vector, please refer to Hemisphere GPS' Programming Manual, available for download from our website at:

www.hemispheregps.com

Table 5-7 summarizes the commands detailed in this section.

Table 5-7: GPS heading Commands

Message	Description
TILTAID	Command to turn on tilt aiding and query the current feature status
TILTCAL	Command to calibrate tilt aiding and query the current feature status
GYROAID	Command to turn on gyro aiding and query the current feature status and query the current feature status
LEVEL	Command to turn on level operation and query the current feature status
NMEA	This command instructs the Crescent Vector on how to preface the HDT and HDR messages,
CSEP	Query to retrieve the current separation between GPS antennas
MSEP	Command to manually set the GPS antenna separation and query the current setting
HTAU	Command to set the heading time constant and to query the current setting
PTAU	Command to set the pitch time constant and to query the current setting

Table 5-7: GPS heading Commands

Message	Description
HRTAU	Command to set the rate of turn time constant and to query the current setting
JTAU,COG	Command to set the course over ground time constant and to query the current setting
JTAU,SPEED	Command to set the speed time constant and to query the current setting
HBIAS	Command to set the heading bias and to query the current setting
PBIAS	Command to set the pitch bias and to query the current setting
NEGTILT	Command to turn on the negative tilt feature and to query the current setting
ROLL	Command to configure the Crescent Vector for roll or pitch output
SEARCH	Command to force a new RTK heading search
FLIPBRD	Command to allow upside down installation
SUMMARY	Query to show the current configuration of the Crescent Vector
HELP	Query to show the available commands for GPS heading operation and status
JASC	Command to turn on different messages
HEHDG	Command to provide magnetic deviation and variation for calculating magnetic or true heading
HEHDM	Command provides magnetic heading of the vessel derived from the true heading calculated
HEHDT	Command to provide true heading of the vessel

Table 5-7: GPS heading Commands

Message	Description
HPR	Proprietary NMEA sentence that provides the heading, pitch/roll information and time in a single message
INTLT	Proprietary NMEA sentence that provides the title measurement from the internal inclinometer, in degrees
HEROT	Command that contains the vessel's rate of turn information
JWCONF	Command that allows the secondary antenna's SNR within a Vector unit

$JATT,TILTAID

The Crescent Vector's internal tilt sensor (accelerometer) is enabled by default and constrains the RTK heading solution to reduce startup and re acquisition times. Since this sensor resides inside the Crescent Vector, the receiver enclosure must be installed in a horizontal plane, as must the Antenna Array.

To turn the tilt-aiding feature off, use the following command:

$JATT,TILTAID,NO<CR><LF>

This feature may be turned back on with the following command:

$JATT,TILTAID,YES,<CR><LF>

To query the Crescent Vector for the current status of this feature, issue the following command:

$JATT,TILTAID<CR><LF>

Note: When choosing to increase the antenna separation of the Crescent Vector OEM beyond the default 0.5 m length, use of tilt aiding is required.

$JATT,TILTCAL

The tilt sensor of the Crescent Vector can be calibrated in the field; however the Crescent Vector enclosure must be horizontal when performing the calibration. To calibrate the Crescent Vector's internal tilt sensor, issue the following command.

$JATT,TILTCAL<CR><LF>

The calibration process takes about two seconds to complete. The calibration is automatically saved to memory for subsequent power cycles.

$JATT,GYROAID

The Crescent Vector's internal gyro is shipped off by default, and it offers two benefits. It will shorten re acquisition times when a GPS heading is lost, due to obstruction of satellite signals, by reducing the search volume required for solution of the RTK. It will also provide an accurate substitute heading for a short period (depending on the roll and pitch of the vessel) ideally seeing the system through to re acquisition. This is why we highly recommend turning the gyro aiding on.

Exceeding rates of 90 degrees per second is not recommended since the gyro cannot measure rates beyond this point. This is a new recommendation since we now use gyro measurements to get a heading rate measurement.

To turn on the gyro-aiding feature, use the following command:

$JATT,GYROAID,YES<CR><LF>

To turn this feature off, use the following command:

$JATT,GYROAID,NO<CR><LF>

To request the status of this message, send the following command:

$JATT,GYROAID<CR><LF>

Every time the Crescent Vector is powered, the gyro goes through a 'warm-up' procedure. This warm up calibrates the gyro to a point where it is operational to its fullest potential. The gyro will automatically warm up by itself over the span of several minutes. This 'self-calibration' is the equivalent to performing the procedure below. This procedure should be followed if the gyro needs to be fully calibrated at a certain time.

When the Crescent Vector unit is installed, apply power and wait several minutes until it has acquired a GPS signal and it is computing heading. Ensure that the gyroaiding feature is on by issuing a $JATT,GYROAID<CR><LF> command. Then, slowly spin the unit for one minute at a rate of no more than 15 degrees per second. Then, let it sit stationary for four minutes. The Crescent Vector's gyro is now fully calibrated. Since this setting cannot be saved, this procedure must be performed every time the Crescent Vector's power is cycled.

$JATT,LEVEL

This command is used to invoke the level operation mode of the Crescent Vector. If the application will not involve the system tilting more than ±10° maximum, then this mode of operation may be used. The benefit of using level operation is increased robustness and faster acquisition times of the RTK heading solution. By default, this feature is turned off. The command to turn this feature on follows:

$JATT,LEVEL,YES<CR><LF>

To turn this feature off, issue the following command:

$JATT,LEVEL,NO<CR><LF>

To determine the current status of this message, issue the following command:

$JATT,LEVEL<CR><LF>

$JATT,NMEAHE,X

This command instructs the Crescent Vector on how to preface the JDT and HDR messages. They can be prefaced with HE or GP. It has the following format:

$JATT,NMEAHE,x<CR><LF>

Where “x” is either 1 for HE or 0 for GP.

For example, the following command would allow the $GPHDT message to be logged:

$JATT,NMEAHE,0<CR><LF>

The following command would allow the $HEHDT message to be logged:

$JATT,NMEAHE,1<CR><LF>

$JATT,CSEP

This command polls the Crescent Vector for the current separation between antennas, as solved for by the attitude algorithms. It has the following format:

$JATT,CSEP<CR><LF>

The Crescent Vector will reply with the following:

$JATT,x,CSEP>

Where “x” is the antenna separation in m.

$JATT,MSEP

This command is used to manually enter a custom separation between antennas (must be accurate to within one to two centimeters). Using the new center-to-center measurement, send the following command to the Crescent Vector:

$JATT,MSEP,sep<CR><LF>

Where “sep” is the measured antenna separation entered in meters.

To show the current antenna separation, issue the following command:

$JATT,MSEP<CR><LF>

$JATT,HTAU

The heading time constant allows for the adjustment of the level of responsiveness of the true heading measurement provided in the $HEHDT message. The default value of this constant is 2.0 seconds of smoothing when the gyro is enabled. The gyro by default is enabled, but can be turned off. By turning the gyro off, the equivalent default value of the heading time constant would be 0.5 seconds of smoothing.

This is not done automatically, and therefore must be entered manually by the user. Increasing the time constant will increase the level of heading smoothing.

The following command is used to adjust the heading time constant:

$JATT,HTAU,htau<CR><LF>

Where “htau” is the new time constant that falls within the range of 0.0 to 3600.0 seconds.

Depending on the expected dynamics of the vessel, this parameter may need to be adjusted. For instance, if the vessel is very large and is not able to turn quickly, increasing this time is reasonable. The resulting heading would have reduced ‘noise’, resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the heading measurement with higher rates of turn. A convenient formula for determining what the level of smoothing follows for when the gyro is in use. It is best to be conservative and leave it at the default setting if unsure about how to set the value.

htau (in seconds)
=
40 / maximum rate of turn (in °/s) – gyro ON

htau (in seconds)
=
10 / maximum rate of turn (in °/s) – gyro OFF

The Crescent Vector may be queried for the current heading time constant by issuing the same command without an argument:

$JATT,HTAU<CR><LF>

Note: It is best to be conservative and leave it at the default setting of 2.0 seconds when the gyro is on and at 0.5 seconds when the gyro is off if unsure about the best value for the setting.

$JATT,PTAU

The pitch time constant allows for the adjustment of the level of responsiveness of the pitch measurement provided in the $PSAT,HPR message. The default value of this constant is 0.5 seconds of smoothing. Increasing the time constant will increase the level of pitch smoothing.

The following command is used to adjust the pitch time constant:

$JATT,PTAU,ptau<CR><LF>

Where ‘ptau’ is the new time constant that falls within the range of 0.0 to 3600.0 seconds.

Depending on the expected dynamics of the vessel, this parameter may need adjusting. For instance, if the vessel is very large and is not able to pitch quickly, increasing this time is reasonable. The resulting pitch would have reduced ‘noise’, resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the pitch measurement. A convenient formula for determining what the level of smoothing follows. It is best to be conservative and leave it at the default setting if unsure about how to set this value. (See the formula at the top of page 128.)

ptau (in seconds)
=
10 / maximum rate of pitch (in °/s)

The Crescent Vector OEM may be queried for the current pitch time constant by issuing the same command without an argument:

$JATT,PTAU<CR><LF>

Note: It is best to be conservative and leave it at the default setting of 0.5 seconds if unsure about the best value for this setting.

$JATT,HRTAU

The heading rate time constant allows for the adjustment of the level of responsiveness of the rate of heading change measurement provided in the $HEROT message. The default value of this constant is 2.0 seconds of smoothing. Increasing the time constant will increase the level of heading smoothing.

The following command is used to adjust the heading time constant:

$JATT,HRTAU,hrtau<CR><LF>

Where “hrtau” is the new time constant that falls within the range of 0.0 to 3600.0 seconds.

Depending on the expected dynamics of the vessel, this parameter may be adjusted. For instance, if the vessel is very large and is not able to turn quickly, increasing this time is reasonable. The resulting heading would have reduced ‘noise’, resulting in consistent values with time. However, artificially increasing this value such that it does not agree with a more dynamic vessel could create a lag in the rate of heading change measurement with higher rates of turn. A convenient formula for determining what the level of smoothing follows. It is best to be conservative and leave it at the default setting if unsure about how to set this value.

hrtau (in seconds)
=
10 / maximum rate of the rate of turn (in °/s2)

The Crescent Vector may be queried for the current heading rate time constant by issuing the same command without an argument:

$JATT,HRTAU<CR><LF>

Note: It is best to be conservative and leave it at the default setting of 2.0 seconds if unsure about the best value for this setting.

$JTAU,COG

The course over ground (COG) time constant allows the level of responsiveness of the COG measurement provided in the $GPVTG message to be adjusted. The default value of this constant is 0.0 seconds of smoothing. Increasing the time constant will increase the level of COG smoothing.

The following command is used to adjust the COG time constant:

$JTAU,COG,tau<CR><LF>

Where “tau” is the new time constant that falls within the range of 0.0 to 200.0 seconds.

The setting of this value depends upon the expected dynamics of the Crescent. If the Crescent will be in a highly dynamic environment, this value should be set to a lower value since the filtering window would be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, this value can be increased to reduce measurement noise. The following formula provides some guidance on how to set this value. It is best to be conservative and leave it at the default setting if unsure about best value for this setting.

tau (in seconds)
=
10 / maximum rate of change of course (in °/s)

The Crescent may be queried for the current course over ground time constant by issuing the same command without an argument:

$JTAU,COG<CR><LF>

Note: It is best to be conservative and leave it at the default setting of 0.0 seconds if unsure about the best value for this setting.

$JTAU,SPEED

The speed time constant allows for the adjustment of the level of responsiveness of the speed measurement provided in the $GPVTG message. The default value of this parameter is 0.0 seconds of smoothing. Increasing the time constant will increase the level of

speed measurement smoothing. The following command is used to adjust the speed time constant:

$JTAU,SPEED,tau<CR><LF>

Where “tau” is the new time constant that falls within the range of 0.0 to 200.0 seconds.

The setting of this value depends upon the expected dynamics of the receiver. If the Crescent will be in a highly dynamic environment, this value should be set to a lower value since the filtering window would be shorter, resulting in a more responsive measurement. However, if the receiver will be in a largely static environment, this value can be increased to reduce measurement noise. The following formula provides some guidance on how to set this value initially, however, we recommend to test how the revised value works in practice. It is best to be conservative and leave it at the default setting if unsure what is the best value for this setting.

tau (in seconds)
=
10 / maximum acceleration (in m/s2)

The Crescent may be queried for the current speed time constant by issuing the same command without an argument.

$JTAU,SPEED<CR><LF>

Note: It is best to be conservative and leave it at the default setting of 0.0 seconds if unsure of the best value for this setting.

$JATT,HBIAS

The heading output from the Crescent Vector may be adjusted in order to calibrate the true heading of the Antenna Array to reflect the true heading of the vessel using the following command:

$JATT,HBIAS,x<CR><LF>

Where “x” is a bias that will be added to the Crescent Vector’s heading, in degrees. The acceptable range for the heading bias is -180.0° to 180.0°. The default value of this feature is 0.0°.

To determine what the current heading compensation angle is, send the following message to the Crescent Vector:

$JATT,HBIAS<CR><LF>

$JATT,PBIAS

The pitch/roll output from the Crescent Vector may be adjusted in order to calibrate the measurement if the Antenna Array is not installed in a horizontal plane. The following NMEA message allows for calibration of the pitch/roll reading from the Crescent Vector:

$JATT,PBIAS,x<CR><LF>

Where “x” is a bias that will be added to the Crescent Vector’s pitchroll measure, in degrees. The acceptable range for the pitch bias is -15.0° to 15.0°. The default value of this feature is 0.0°.

To determine what the current pitch compensation angle is, send the following message to the Crescent Vector.

Note: The pitch/roll bias is added after the negation of the pitch/roll measurement (if so invoked with the $JATT,NEGTILT command).

$JATT,NEGTILT

When the secondary GPS antenna is below the primary GPS antenna, the angle from the horizon at the primary GPS antenna to the secondary GPS antenna is considered negative.

Depending on the convention for positive and negative pitch/roll, it may be good to change the sign (either positive or negative) of the pitch/roll. To do this, issue the following command:

$JATT,NEGTILT,YES<CR><LF>

To return the sign of the pitch/roll measurement to its original value, issue the following command:

$JATT,NEGTILT,NO<CR><LF>

To query the Crescent Vector for the current state of this feature, issue the following command:

$JATT,NEGTILT<CR><LF>

$JATT,ROLL

To get the roll measurement, he Antenna Array will need to be installed perpendicular to the vessel's axis, and send the following command to the Crescent Vector:

$JATT,ROLL,YES<CR><LF>

To return the Crescent Vector to its default mode of producing the pitch measurement, issue the following command:

$JATT,ROLL,NO<CR><LF>

The Crescent Vector may be queried for the current roll/pitch status with the following command:

$JATT,ROLL<CR><LF>

$JATT,SEARCH

The Crescent Vector may be forced to reject the current RTK heading solution and have it begin a new search with the following command:

$JATT,SEARCH<CR><LF>

Note: The SEARCH function will not work if the GYROAID feature has been enabled. In this case power must be cycled to the receiver to have a new RTK solution computed.

$JATT,FLIPBRD

This new command was added to allow for the Crescent Vector OEM board to be installed upside down. This command should only be used with the Vector Sensor and the Crescent Vector OEM board, since flipping the OEM board doesn't affect the antenna array, which needs to remain facing upwards. When using this command, the board needs to be flipped about roll, so that the front still faces the front of the vessel. To turn this 'upside down' feature on, use the following command:

$JATT,FLIPBRD,YES<CR><LF>

To return the Crescent Vector to its default mode of being right side up, issue the following command:

$JATT,FLIPBRD,NO<CR><LF>

To query the Crescent Vector for the current flip status with the following command:

$JATT,FLIPBRD<CR><LF>

$JATT,SUMMARY

This command is used to receive a summary of the current Crescent Vector settings.

This command has the following format:

$JATT,SUMMARY<CR><LF>

The response has the following format:

$>JATT,SUMMARY,htau,hrtau,ptau,ctau,spdtau,hbias,pbias,hexfla
g<CR><LF>

An example of the response to this message follows:

$>JATT,SUMMARY,TAU:H=0.50,HR=2.00,COG=0.00,SPD=0.00,BIAS
:H=0.00,P=0.00,FLAG_HEX:HF-RMTL=01

Field	**Description**
htau	This data field provides the current heading time constant in seconds
hrtau	This data field provides the current heading rate time constant in seconds
ptau	This data field provides the current pitch time constant in seconds
cogtau	This data field provides the current course over ground time constant in seconds
spdtau	This data field provides the current speed time constant in seconds
hbias	This data field gives the current heading bias in degrees
pbias	This data field gives the current pitch/roll bias in degrees
hexflag	This field is a hex code that summarizes the heading feature status and is described in the following table

Flag	**Value**	
	Feature on	**Feature off**
Gyro aiding	02	0
Negative tilt	01	0
Roll	08	0
Tilt aiding	02	0

Flag		Value
Level	01	0

The “GN- RMTL” field is two separate hex flags, “GN” and “RMTL.” The “GN” value is determined by computing the sum of the gyro aiding and negative tilt values, depending if they are on or off. If the feature is on, their value is included in the sum. If the feature is off, it has a value of zero when computing the sum. The value

of RMTL is computed in the same fashion but by adding the values of roll, tilt aiding, and level operation.

For example, if gyro aiding, roll, and tilt aiding features were each on, the values of ‘GN’ and ‘RMTL’ would be the following:

GN = hex (02 + 0) = hex (02) = 2
RMTL = hex (08 + 02) = hex (10) = A
‘GN-RMTL’ = 2A

The following tables summarize the possible feature configurations for the first GN character and the second RMTL character.

GN value	Gyro value	Negative tilt
0	Off	Off
1	Off	**On**
2	**On**	Off
3	**On**	**On**

RMTL value	Roll	Tilt aiding	Level
0	Off	Off	Off
1	Off	Off	**On**
2	Off	On	Off
3	Off	On	On
8	On	Off	Off
9	On	Off	On
A	On	On	Off
B	On	On	On

$JATT,HELP

The Crescent Vector supports a command that provides a short list of the supported commands if in case they are needed in the field and no documentation is available.

This command has the following format:

$JATT,HELP<CR><LF>

The response to this command will be the following:

$>JATT,HELP,CSEP,MSEP,EXACT,LEVEL,HTAU,HRTAU,HBIASPBIAS,NEGTILT,ROLL,TILTAID,TILTCAL,MAGAID,MAGCAL,MAGCLR,GYROAID,COGTAU,SPDTAU,SEARCH,SUMMARY

$JASC

This command allows the GPS data messages on to be turned on at a particular update rate, or to be turned off. When turning messages on, various update rates available to choose from, depending on what the requirements are.

This command has the following layout:

$JASC,MSG,R[,OTHER]<CR><LF>

Where “MSG” is the name of the data message and “R” is the message rate, as shown in the table below. Sending the command without the optional “,OTHER” data field (without the brackets) will enact a change on the current port.

Sending a command with a zero value for the “R” field turns off a message.

When the “,OTHER” data field (without the brackets) is specified, this command will enact a change on the other port.

The receiver will reply with the following response:

$>

Table 5-8: HDT, ROT, INLT, HPR, HDG, HDM

Field	R (Hz)	Description
HDG	20, 10, 1, 0 or .2	Magnetic deviation and variation
HDM	20, 10, 1, 0 or .2	Magnetic heading
HDT	20, 10, 1, 0 or .2	RTK-derived GPS heading
HPR	20, 10, 1, 0 or .2	Proprietary message containing heading and roll or pitch
INTLT	1 or 0	Internal tilt sensor measurement
ROT	20, 10, 1, 0 or .2	RTC-derived GPS rate of turn

HEHDG Data

This message provides magnetic deviation and variation for calculating magnetic or true heading. This message simulates data from a magnetic sensor, although it does not actually contain one. The purpose of this message is to support older systems which may not be able to accept the HDT message, which is recommended for use. The HDG data message has the following format:

$HEHDG,s.s,d.d,D,v.v,V*cc<CR><LF>

Where “s.s” is the magnetic sensor reading in degrees. Where “d.d” is the magnetic deviation in degrees. Where “D” is either ‘E’ for Easterly deviation or ‘W’ for Westerly deviation. Where “v.v” is the magnetic variation in degrees. Where “V” is either ‘E’ for Easterly deviation or ‘W’ for Westerly deviation.

Note: The HEHDG message header can be changed to GP by using the $JATT,NMEAHE,X command.

HEHDM Message

This message provides magnetic heading of the vessel derived from the true heading calculated. The HDM data message has the following format:

$HEHDM,x.x,M*cc<CR><LF>

Where "x.x" is the current heading in degrees and "M" indicates magnetic heading.

Note: The HEHDM message header can be changed to GP by using the $JATT,NMEAHE,X command.

HEHDT Data

This message provides true heading of the vessel. This is the direction that the vessel (antennas) is pointing and is not necessarily the direction of vessel motion (the course over ground). The HDT data message has the following format:

$HEHDT,x.x,T*cc<CR><LF>

Where "x.x" is the current heading in degrees and "T" indicates true heading.,HPR Data

Note: The HEHTG message header can be changed to GP by using the $JATT,NMEAHE,X command.

The $PSAT,HPR message is a proprietary NMEA sentence that provides the heading, pitch/roll information and time in a single message. This message has the following format:

$PSAT,HPR,time,heading,pitch,roll,type*7B<CR><LF>

Field	Description
Time	UTC time (HHMMSS.SS)
Heading	Heading (degrees)
Pitch	Pitch (degrees)
roll	Roll (degrees)
Type	"N" for GPS derived heading "G" for gyro heading

INTLT Data

The $PSAT,INTLT data message is a proprietary NMEA sentence that provides the title measurement from the internal inclinometer, in degrees. It delivers an output of crude accelerometer measurements of pitch and roll with no temperature compensation or calibration for GPS heading/pitch/roll. This message has the following format:

$PSAT,INTLT,pitch,roll*7B

Where "pitch" and "roll" are in degrees.

ROT Data

The HEROT data message contains the vessel's rate of turn information. It has the following format:

$HEROT,x.x,A*cc<CR><LF>

Where "x.x" is the rate of turn in degrees per minute and "A" is a flag indicating that the data is valid. The "x.x" field is negative when the vessel bow turns to port.

$JWCONF

The JWCONF command allows the secondary antenna within a Vector unit's SNR to be viewed. It has the following format:

$JWCONF,12,0

Where "0" is used to have SLXMon view the primary antennas SNR (factory default)

$JWCONF,12,1

Where "1" is used only for SLXMon.

DGPS Base Station Commands

This section provides information related to the NMEA 0183 messages accepted by the receiver's **e-Dif application Base Station feature**.

Table 5-9 provides a brief description of the commands supported by the Base Station feature for its control and operation.

Table 5-9: Base station Commands

Message	Description
$JRAD,1	This command is used to display the current reference position
$JRAD,1,P	Store present position as reference
$JRAD,lat,lon,height	Store entered position as reference
$JRAD,9,1,1	Initialize Base Station feature

The following subsections provide detailed information relating to the use of each command.

Note: Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JRAD,1

This command is used to display the current reference position in e-Dif applications only.

This command has the following format:

$JRAD,1<CR><LF>

The receiver will reply with a response similar to the following:

$>JRAD,1,51.00233513,-114.08232345,1050.212

Upon startup of the receiver with the e-Dif application running, as opposed to the SBAS application, no reference position will be present in memory. When attempting to query for the reference position, the receiver will respond with the following message:

$JRAD,1,FAILED,PRESENT LOCATION NOT STABLE

$JRAD,1,P

This command records the current position as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used in relative mode, as no absolute point information is specified.

This command has the following format:

$JRAD,1,P<CR><LF>

The receiver will reply with the following response:

$>JRAD,1,OK

$JRAD,1,LAT,LON,HEIGHT

This command is a derivative of the $JRAD,1,P command and is used when absolute positioning is desired in e-Dif applications only.

This command has the following layout:

$JRAD,1,LAT,LON,HEIGHT<CR><LF>

Where the data fields in this command are described in the following table.

Data field	Description
lat	This is the latitude of the reference point in degrees decimal degrees.
lon	This is the longitude of the reference point in degrees decimal degrees.
height	This is the ellipsoidal height of the reference point in meters. Ellipsoidal height can be calculated by adding the altitude and the geiodal separation, both available from the GGA sentence. See example below.

Example of ellipoidal height calculation -

$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4,1071.0,M,
-17.8,M,6.0, 0122*48

ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

The receiver will reply with the following response:

$>JRAD,LAT,LON,HEIGHT

Note: Both latitude and longitude must be entered as decimal degrees. The receiver will not accept the command if there are no decimal places.

$JRAD,9,1,1

This command initializes the Base Station feature and uses the previously entered point, either with $JRAD,1,P or $JRAD,1,lat,long,height, as the reference with which to compute Base Station corrections in e-Dif applications only. This would be used for both relative mode and absolute mode.

This command has the following format:

$JRAD,9,1,1<CR><LF>

The receiver will reply with the following response:

$>JRAD,9,OK

Note: The $JASC,RTCM,1 command must be sent to the receiver to start outputting standard RTCM corrections.

Note: Please refer to the Base Station instructions document for more detailed setup steps.

Local Differential and RTK Commands

Table 5-10 provides a brief description of the commands supported by Local Differential (L-Dif) and RTK feature for its control and operation.

Table 5-10: L-Dif and RTK Commands

$JRTK,1	Shows the receiver's reference position (base station and rover)
$JRTK,1,P	Sets the receiver's reference position to the current nav position (base station and rover)
$JRK,1,lat,lon,height	Sets the receiver's reference position to the command position
$JRTK,5	Shows transmission status
$JRTK,5Transmit	Suspends or resumes RTK transmission
$JRTK,6	View base station progress
$JRTK,12,Allow	Disable or enable the receiver to go into fixer integer more (i.e. RTK mode)
$JRTK,17	Display lat and lon height that is currently being used
$JRTK,18	Display distance to base station
$JASC,DFX,r[,OTHER]	Single frequency only (only for Crescent)
$JRTK,ROX,r[,OTHER]	Dual Frequency only (only for Eclipse)

$JRTK,1

ThE $JRTK,1 command shows the receiver's reference position (base station and rover).

$JRTK,1,P

The $JRTK,1,P command sets the receiver's reference position to the current nav position (base station and rover).

$JRTK,1,lat,lon,height

The $JRTK,1,lat,lon,height command initializes the L-Dif feature and uses the entered point coordinates as the reference with which to compute L-Dif corrections in L-Dif.

This command has the following format:

$JRTK,1,LAT,LON,HEIGHT<CR><LF>

Where the data fields in this command are described in the following table.

Data field	Description
lat	This is the latitude of the reference point in degrees decimal degrees.
lon	This is the longitude of the reference point in degrees decimal degrees.
height	This is the ellipsoidal height of the reference point in meters. Ellipsoidal height can be calculated by adding the altitude and the geiodal separation, both available from the GGA sentence. See example below.

Example of ellipsoidal height calculation -

$GPGGA,173309.00,5101.04028,N,11402.38289,W,2,07,1.4,1071.0,M,
-17.8,M,6.0, 0122*48
ellipsoidal height = 1071.0 + (-17.8) = 1053.2 meters

Note: Both latitude and longitude must be entered as decimal degrees. The receiver will not accept the command if there are no decimals.

$JRTK,5

The $JRTK,5 command shows the base's transmission status for RTK applications. If suspended, respond with:

$>JKRTK,6

Otherwise:

$>JRTK,5,1

Also see the $JRTK,6 command on page 150.

$JRTK,5,Transmit

THe $JRTK,5,Transmit command suspends or resumes the transmission of RTK.

Field	Description
0	Suspend
1	Resume

$JRTK,6

The $JRTK,6 command views the progress of the base station. The reply is:

$JRTK,6,TimeToGo,ReadyTransmit,Transmitting

Field	Description
TimeToGo	Seconds left until ready to transmit RTK
ReadyTransmit	Non zero when configured to transmit and ready to transmit RTK on at least one port. It is a BitMask of the transmitting port, with bit 0 being port A, bit 1 being port B and bit 2 being port C. It will be equal to “Transmitting” unless transmission has be suspended with $JRTK,5,0.
Transmitting	Non zero when actually transmitting RTK on at least one port. It is a BitMask of the transmitting port, with bit 0 being port A, bit 1 being port B and bit 2 being port C.

$JRTK,12,Allow Rover

The $JRTK,12, Allow Rover disables or disables the receiver to go into fixed integer more (i.e. RTK mode).

Field	Description
Default	Enabled
Allow = 1	Allow RTK (recommended)
Allow = 0	Stay in L-Dif mode

$JRTK,17

The $JRTK,17 command displays the lat and log height that is currently used as a reference for the base station (base station and rover.

$JRTK,18

The $JRTK,18 command shows the distance from the rover to the base station the rover in meters (rover only).

$JASC,DFX,r[,OTHER]

The $JASC,DFX,r[,OTHER] command is used only on the base receiver when using L-Dif or RTK mode on the Crescent to turn on the proprietary messages that are sent to the rover to correct its position. Where “r” is “0” or “1.” Differential is relative to the reference position (base only). “0” turns the corrections on. “1” turns the corrections on.

Note: The $JASC,DFX,1 command must be sent to the receiver to start outputting proprietary L-Dif corrections.

Note: Please refer to the L-Dif user guide and quick reference for more detailed setup steps.

JASC,ROX,r[,OTHER]

The $JASC,DFX,r[,OTHER] command is used only on the Eclipse base receiver when using GPS dual frequency RTK mode to turn on the proprietary messages that are sent to the rover to correct its position for Eclipse only. Where “r” is “0” or “1.” RTK is relative to the reference position (base only). “0” turns the corrections off. “1” turns the corrections on.

Note: The $JASC,ROX,1 command must be sent to the receiver to start outputting proprietary corrections.

Data Messages

The following subsections describe the NMEA 0183 data messages listed in Table 5-11 in detail.

Table 5-11: Data Messages

Message	**Max rate**	**Description**
GPGNS	20 Hz	Fixes data for single or combined satellite navigation systems
GPGGA	20 Hz	GPS fix data
GPGLL	20 Hz	Geographic position - latitude/longitude
GPGSA	1 Hz	GNSS (Global Navigation Satellite System) DOP and active satellites
GPGST	1 Hz	GNSS pseudorange error statistics
GPGSV	1 Hz	GNSS satellite in view
GPRMC	20 Hz	Recommended minimum specific GNSS data
GPRRE	1 Hz	Range residual message
GPVTG	20 Hz	Course over ground and ground speed
GPZDA	20 Hz	Time and date
GRS	20 Hz	Supports the Receiver Autonomous Integrity Monitoring (RAIM)
RD1	1 Hz	SBAS diagnostic information (proprietary NMEA 0183 message)

Note: For clarity, each data message will be presented on a new page.

Note: 20 Hz output is only available with a 20 Hz subscription.

GPGNS Data Message

The GPGNS message fixes data for GPS, GLONASS, possible future satellite systems and system combining these. The GPGNS data message is broken down into its components in Table 5-12. This message follows the following form:

$GPGNS,hhmmss.ss,llll.ll,a,yyyyy.yy,a,mm,ss,h.h,a.a,
g.g,d.d,r.r*hh<CR><LF>

Table 5-12: GPGNS Data Message Defined

Field	Description
GNS,hhmmss.ss	UTC of Position
llll.ll	Latitude, N/S
a	Latitude, N/S
yyyyy.yy	Longitude, E/W
a	Longitude, E/W
mm	Mode indicator
ss	Total number of satellites in use, 00-99
h.h	HDOP
a.a	Antenna altitude, meters, re: mean-sea-level (geoid)
g.g	Geoidal seperation, meters
d.d	Age of differential data
r.r	Differential reference station ID

GPGGA Data Message

The GPGGA message contains detailed GPS position information and is the most frequently used NMEA 0183 data message. The GGA data message is broken down into its components in Table 5-13. This message takes the following form:

$GPGGA,HHMMSS.SS,DDMM.MMMM,S,DDDMM.MMMM,S,N,QQ,PP.P,
S

AAAAA.AA,M,±XXXX.XX,M,SSS,AAAA*CC<CR><LF>

Table 5-13: GPGGA Data Message Defined

Field	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS position
DDMM.MMMMMS	Latitude in degrees, minutes, and decimal minutes
S	S = N or S = S, for North or South latitude
DDDMM.MMMMMM	Longitude in degrees, minutes, and decimal minutes
S	S = E or S = W, for East or West longitude
N	Quality indicator, 0 = no position, 1 = un-differentially corrected position (autonomous) 2 = differentially corrected position (SBAS, DGPS, OmniStar, l-Dif and e-Dif) 4 = RTK fixed integer (Crescent RTK, Eclipse RTK)
QQ	Number of satellites used in position computation
PP.P	HDOP = 0.0 to 9.9

Table 5-13: GPGGA Data Message Defined

Field	Description
SAAAA.AA	Antenna altitude
M	Altitude units, M = meters
+/-XXXXX.XX	Geoidal separation (needs geoidal height option)
M	Geoidal seperation units, M = meters
SSS	Age of differential corrections in seconds
AAA	Reference station identification
*CC	Checksum
<CR><LF>	Carriage return and line feed

GPGLL Data Message

The GPGLL message contains latitude and longitude. The GLL data message is broken down into its components in Table 5-14. This message has the following format:

$GPGLL,DDMM.MMMM,S,DDDMM.MMMM,S,HHMMSS.SS,S*CC<CR><LF>

Table 5-14: GPGLL Data Message Defined

Field	Description
DDMM.MMMMM	Latitude in degrees, minutes, and decimal minutes
S	S = N or S = S, for North or South latitude
DDDMM.MMMMM	Longitude in degrees, minutes, and decimal minutes

Table 5-14: GPGLL Data Message Defined

Field	Description
S	S = E or S = W, for East or West longitude
HHMMSS.SS	UTC time in hours, minutes, and seconds of GPS position
S	Status, S = A = valid, S = V = invalid
*SS	Checksum
<CR><LF>	Carriage return system

GPGSA Data Message

The GPGSA message contains GPS DOP and active satellite information. Only satellites used in the position computation are present in this message. Null fields are present when data is unavailable due to the number of satellites tracked. Table 5-15 breaks down the GSA message into its components. This message has the following format:

$GPGSA,A,B,CC,DD,EE,FF,GG,HH,II,JJ,KK,MM,NN,OO,P.P,Q.Q,R.R*CC <CR><LF>

Table 5-15: GPGSA Data Message Defined

Field	Description
A	Satellite acquisition mode M = manually forced to 2D or 3D, A = automatic swap between 2D and 3D
B	Position mode, 1 = fix not available, 2 = 2D fix, 3 = 3D fix
CC to OO	Satellites used in the position solution, a null field occurs if a channel is unused
P.P	Position Dilution of Precision (PDOP) = 1.0 to 9.9

Table 5-15: GPGSA Data Message Defined

Field	Description
Q.Q	Horizontal Dilution of Precision (HDOP) 1.0 to 9.9
R.R	Vertical Dilution of Precision (VDOP) = 1.0 to 9.9
*CC	Checksum
<CR><LF>	Carriage return and line feed

GPGST Data Message

The GPGST message contains Global Navigation Satellite System (GNSS) pseudorange error statistics. Table 5-16, breaks down the GST message into its components. This message has the following format:

$GPGST,HHMMSS.SS,A.A,B.B,C.C,D.D,E.E,F.F,G.G *CC<CR><LF>

Table 5-16: GPGTS Data Message Defined

Field	Description
HHMMSS.SSS	UTC time in hours, minutes, and seconds of the GPS position
A.A	Root mean square (rms) value of the standard deviation of the range inputs to the navigation process. Range inputs include pseudoranges and differential GNSS (DGNSS) corrections
B.B	Standard deviation of semi-major axis of error ellipse (meters)
C.C	Standard deviation of semi-minor axis of error ellipse (meters)
D.D.	Error in Eclipse's semi major axis origination, in decimal degrees, true north.

Table 5-16: GPGTS Data Message Defined

Field	Description
E.E	Standard deviation of latitude error (meters)
F.F	Standard deviation of longitude error (meters)
G.G	Standard deviation of altitude error (meters)
*CC	Checksum
<CR><LF>	Carriage return and line feed

(NMEA, NMEA 0183: Standard for Interfacing Electronic Devices, version 3.00, 2006)

GPGSV Data Message

The GPGSV message contains GPS satellite information. Null fields occur where data is not available due to the number of satellites tracked. Table 5-17 breaks down the GSV data message into it components. This message has the following format:

$GPGSV,T,M,N,II,EE,AAA,SS,…II,EE,AAA,SS,*CC<CR><LF>

Table 5-17: GPGSV Data Message Defined

Field	Description
T	Total number of messages
M	Message number, m = 1 to 3
N	Total number of satellites in view
II	Satellite number
EE	elevation in degrees, ee = 0 to 90
AAA	Azimuth (true) in degrees, aaa = 0 to 359

Table 5-17: GPGSV Data Message Defined

Field	Description
SS	SNR (dB) + 30, ss = 99
*CC	Checksum
<CR><LF>	Carriage return and line feed

GPRMC Data Message

The GPRMC message contains recommended minimum specific GPS data. Table 5-18 breaks down the RMC data message into its components. This message has the following format:

$GPRMC,HHMMSS.SS,A,DDMM.MMM,N,DDDMM.MMM,W,Z.Z,Y.Y,DDMMYY,D.D,V *CC<CR><LF>

Table 5-18: GPRMC Data Message Defined

Field	Description
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS position
A	Status is valid if A = A, status is invalid if A = V
DDMM.MMMMM	Latitude in degrees, minutes, and decimal minutes
N	S = N or S = S, for North or South latitude
DDDMM.MMM	Longitude in degrees, minutes, and decimal minutes
W	S = E or S = W, for East or West longitude
Z.Z	Ground speed in knots
Y.Y	Track made good, reference to true north

Table 5-18: GPRMC Data Message Defined

Field	Description
DDMMYY	UTC date of position fix in day, month and year
D.D	Magnetic Variation in degrees
V	Variation sense v = E = East, V = W = West
*CC	Checksum
<CR><LR>	Carriage return and line feed

GPRRE Data message

The GPRRE message contains the satellite range residuals and estimated position error. Table 5-19 breaks down the RRE data message into its components. This message has the following format:

$GPRRE,N,II,RR…II,RR,HHH.H,VVV.V *CC<CR><LF>

Table 5-19: GPRRE Data Message Defined

Field	Description
N	Number of satellites used in position computation
II	Satellite number
RR	Range residual in meters
HHH.H	Horizontal position error estimate in meters
VVV.V	Vertical position error estimate in meters
*CC	Checksum
<CR><LF>	Carriage return and line feed

GPVTG Data Message

The VTG message contains velocity and course information. Table 5-20 breaks down the VTG data message into its components. This message has the following format:

$GPVTG,TTT,C,TTT,C,GGG.GG,U,GGG.GG,U*CC<CR><LF>

Table 5-20: GPVTG Data Message Defined

Field	Description
TTT	True course over ground, TTT = 000 to 359, in degrees
C	True course over ground indicator, C = T always
TTT	Magnetic course over ground, TTT = 000 to 359, in degrees
C	Magnetic course over ground indicator, always C = M
GGG.GG	Speed over ground, 000 to 999 km/h
U	Speed over ground units, U = N = Nautical mile/hour
GGG.GG	Speed over ground, 000 to 999 km/h
U	A = Autonomous mode D = Differential mode E = Estimated (dead reckoning) mode M = Manual input mode S = Simulator mode N = Data not valid
*CC	Checksum
<CR><LF>	Carriage return and line feed

GPZDA Data Message

The GPZDA message contains Universal Time information. Table 5-21 breaks down the GPZDA data message into its components. This message has the following format:

$GPZDA,HHMMSS.SS,DD,MM,YYYY,XX,YY*CC<CR><LF>

Table 5-21: GPZDA Data Message Defined

Field	Message
HHMMSS.SS	UTC time in hours, minutes, and seconds of the GPS unit
DD	Day, DD = 0 to 31
MM	Month, MM = 1 to 12
YYYY	Year
XX	Local zone description in hours, XX = -13 to 13
YY	Local zone description in minutes, YY = 0 to 59
*CC	Checksum
<CR><LF>	Carriage return and line feed

GRS Data Message

The GRS message supports the Receiver Autonomous Intergrity Monitoring (RAIM).The GGA data message is broken down into its components in Table 5-22. This message takes the following form:

$--GRS.hhmmss.ss,x,x.x,x.x,x.x,x.x,x.x,
x.x,x.x,x.x,x.x,x.x,x.x,x.x*hh<CR><LF>

Table 5-22: GRS Data Message Defined

Field	Description
hhmmss.ss	UTC time of the GGA or GNS fix associated with this sentence
x	Mode: 0=residuals were used to calculate the position given in the matching GGA or GNS sentence 1=residuals were recomputed after the GGA or GNS position was computed
x.x,x.x,x.x,x.x, x.x,x.x,x.x,x.x, x.x,x.x,x.x,x.x	Range residuals, in meters, for satellites used in the navigation solution. Order must match order of satellite ID numbers in GSA. When GRS is used, GSA and GSV are generally required

(NMEA, NMEA 0183: Standard for Interfacing Electronic Devices, version 3.00, 2006)

RD1 Data Message

The RD1 message contains diagnostic information for SBAS operation. Table 5-23 breaks down the RD1 data message into its components.

This message has the following format:

$RD1,SECOFWEEK,WEEKNUM,FREQMHZ,DSPLOCKED,BERBER2,
AGC,DDS,DOPPLER,DSPSTAT,ARMSTAT,DIFFSTATUS,NAVCON
DITION *CC<CR><LF>

Table 5-23: RD1 Data Message Defined

Field	Description
SecOfWeek	The Second of GPS Week (may be a couple of seconds old)
WeekNum	The GPS week number
FeqMHz	The L-band frequency in MHz (1475.4200 is used for SBAS)
DSPLocked	N/A
BER-BER2	BER - BERs are given for both SBAS satellites being tracked
AGC	L-band signal strength
DDS	0.0 for SBAS
DOPPLER	0 for SBAS
DSPSTAT	A status bit mask for the DSP tracking of SBAS
ARMSTAT	A status bit mask for the ARM GPS solution
DiffSTATUS	The SBAS PRN of the satellite in use

Table 5-23: RD1 Data Message Defined

Field	Description
NAVCONDITION	A series of hex character fields, which is read from right to left, with each field representing the number of GPS satellites satisfying a certain condition, all of which conditions are required if the satellite is to be used in the solution
*CC	Checksum
<CR><LF>	Carriage return and line feed

The following table describes the ARM status.

Field	Description
01	GPS lock
02	DGPS valid data
04	The ARM processor has lock
08	DGPS solution
10	DGPS solution is good
20	Not used
40	Not used

An example of the NavCondition is presented in the following table for the 179889A value.

Field	Description
A	The number of satellites with lock and carrier phase
9	The number of satellites with ephemeris received
8	The number of satellites with healthy ephemeris
8	The number of satellites that are tracked, have an ephemeris, which is healthy, and are above the elevation mask
9	The number of satellites that above the elevation mask
7	The number of satellites with differential
1	The number of satellites with no differential

Beacon Receiver Commands

If integrating a Hemisphere GPS SBX beacon module along with the receiver GPS engine, we recommend that the beacon receiver be interfaced to Port D of the receiver engine. We have implemented some command and message pass-through intelligence for such an integration. In this configuration, the commands in the following table may be issued to the beacon receiver through either Port A, Port B, or Port C of the receiver module. Table 5-24 summarizes the beacon commands passed through from Port A, B, and C to Port D, to which the SBX beacon module would be interfaced.

Table 5-24: SBX Beacon Commands

Message	Description
$GPMSK	Command to tune beacon receiver and turn on diagnostic information
$PCSI,1	This command is used to get beacon status information from the SBX beacon engine inside the receiver
$PCSI,3,2	Command to list the 10 closest stations
$PCSI,3,3	Command to display the contents of the station database

$GPMSK Beacon Tune Command

This command instructs the SBX to tune to a specified frequency and automatically select the correct MSK rate. When this command is sent through either Port A, B, or C, it will automatically be routed to Port D. The resulting confirmation of this message will be returned to the same port from which the command was sent. It has the following form:

$GPMSK,fff.f,F,mmm,M,n<CR><LF

Response:

$PCSI,ACK,GPMSK,fff.f,F,mmm,M,n<CR><LF>

If using database tuning mode the command has the following form:

$GPMSK,,D,,D<CR><LF>

Field	**Description**
fff.f	This field is the beacon frequency in kHz (283.5 to 325) and may be left null if the following field is set to automatic "A" or database "D"
F	This field selects the frequency selection mode, either manual "M" or automatic "A" or Database "D"
mmm	This field is the MSK bit rate. If the following field is set to automatic "A" or database "D," this field can be left null.
M	This field selects the MSK rate selection mode, either manual "M" or automatic "A" or database "D""
n	Period of output of performance status message 0 to 100 seconds ($CRMSS)

When this message is acknowledged by the SBX, it will immediately tune to the frequency specified and demodulate at the rate specified. When the "n" field is set to a non-zero value, the SBX will output the $CRMSS message at that period through the serial port from which the SBX was tuned. When issuing the tune command with a nonzero "n" field through Port B, the periodic output of the $CRMSS message will not impact the output of RTCM on Port A. However, when tuning the SBX with a non-zero "n" field through Port A, the NMEA 0183 status message will be interspersed within the RTCM data. Most GPS engines will not be able to filter the NMEA 0183 message, causing the overall data to fail parity checking. When power to the SBX is removed and reapplied, the status output interval resets to zero (no output).

Note: When tuning the SBX engine, if the “n” field in this message is non-zero, the status data message output by the SBX may interrupt the flow of RTCM data to the GPS receiver. Re-power the SBX to stop the output of the $CRMSS message, or retune the Beacon receiver with the “n” field set to zero.

$PCSI,1 Beacon Status Command

This command is used to obtain $PCSI,CS0 beacon status data from an SBX engine when interfaced to the receiver Port D. When this command is sent through either Port A, B, or C it will automatically be routed to Port D. The resulting $PCSI,CS0 message will be returned to the same port from which the command was sent at the desired rate. It has the following format:

$PCSI,1,1<CR><LF>

Response:

$PCSI,CS0,Pxxx-y.yyy,SN,fff.f,M,ddd,R,SS,SNR,MTP,**WER**,ID,H,T,**G**

Field	Description
ID	1024 (undefined), 0-1023 (valid range)
H	8 (undefined), 0-7 (valid range)

Example:

$PCSI,CS0,P030-0.000,19001,313.0,D,100,D,18,8,80,**0**,63,0,1,**48**

Field	Description
Tune modes	"**A**"uto, "**M**"anual, and "**D**"atabase
WER	The percentage of bad 30-bit RTCM words in the last 25 words
G	The AGC gain in dB (0 to 48 dB)

$PCSI,3,2 Ten Closest Stations

This command displays the ten closest stations. It has the following format:

$PCSI,3,2<CR><LF>

Response:

$PCSI,3,2,StationID,name,freq,status,time,date,
distance,health,WER
$PCSI,3,2, ...
$PCSI,3,2, ...
$PCSI,3,2, ...
$PCSI,3,2, ...
...

Field	Description
Name	Displays time/date of update for a station added by using information from an almanac message (in the format ddmmyy->time)
Status	0 (operational), 1 (undefined), 2 (no information), 3 (do not use)
Time	Have not been implemented. Currently displayed at 0
Date	Have not been implemented. Currently displayed at 0
Distance	Calculated in nautical miles
Health	-1 (not updated), 8 (undefined), 0-7 (valid range)
WER	-1 (not updated), 0-100 (valid range)

Example:

```
$PCSI,3,2, 849,Polson        MT,2870,0,210,0,0,-1,-1
$PCSI,3,2, 848,Spokane       WA,3160,0,250,0,0,-1,-1
$PCSI,3,2, 907,Richmond      BC,3200,0,356,0,0,-1,-1
$PCSI,3,2, 888,Whidbey Is.   WA,3020,0,363,0,0,-1,-1
$PCSI,3,2, 887,Robinson Pt.  WA,3230,0,383,0,0,-1,-1
$PCSI,3,2, 874,Billings      MT,3130,0,389,0,0,-1,-1
$PCSI,3,2, 871,Appleton      WA,3000,0,420,0,0,-1,-1
$PCSI,3,2, 908,Amphitrite Pt BC,3150,0,448,0,0,-1,-1
$PCSI,3,2, 886,Fort Stevens  OR,2870,0,473,0,0,-1,-1
$PCSI,3,2, 909,Alert Bay     BC,3090,0,480,0,0,-1,-1
```

$PCSI,3,3 Station Database

This command displays the contents of the station database. It has the following format:

$PCSI,3,3<CR><LF>

Response:

$PCSI,3,3,IDref1,IDref2,StationID,name,frq,lat,long,
datum,status
$PCSI,3,3, ...
$PCSI,3,3, ...
$PCSI,3,3, ...
$PCSI,3,3, ...
...

Field	Description
Lat	Scaled by 364 (+ve indicates N and -ve indicates S)
Long	The longitude is scaled by 182 (+ve indicates N and -ve indicates S)
Datum	1 (NAD83), 0(WGS84)
Status	0 (operational), 1(undefined), 2 (no information), 3, (do not use)

Example:

```
$PCSI,3,3,0282,0283,0891,Level Island AK,2950,20554,-24221,1,0
$PCSI,3,3,0306,0307,0906,Sandspit BC,3000,19377,-23991,1,0
$PCSI,3,3,0278,0279,0889,Annette Is. AK,3230,20044,-23951,1,0
$PCSI,3,3,0300,0301,0909,Alert Bay BC,3090,18412,-23099,1,0
$PCSI,3,3,0302,0303,0908,Amphitrite Pt BC,3150,17806,-22850,1,0
$PCSI,3,3,0270,0271,0885,C. Mendocino CA,2920,14718,-22641,1,0
$PCSI,3,3,0272,0273,0886,Fort Stevens OR,2870,16817,-22559,1,0
$PCSI,3,3,0304,0305,0907,Richmond BC,3200,17903,-22407,1,0
$PCSI,3,3,0276,0277,0888,Whidbey Is. WA,3020,17587,-22331,1,0
...
```

NMEA 0183 queries

This section discusses the standard and proprietary NMEA 0183 queries accepted by the SBX.

It is important to note that when issued to the SBX primary communications port, the response messages will be output interspersed with RTCM correction information. This may cause conflicts with a GPS receiver's ability to compute differential corrected solutions. To avoid this potential conflict, Section 3.5 describes the use of the SBX secondary communication port for performance and configuration status querying and reporting.

Standard Queries

The following subsections describe the selection of valid standard NMEA-0183 queries, and their responses.

$GPCRQ Receiver Operating Status Query -

This standard NMEA query prompts the SBX for its operational status. It has the following format:

$GPCRQ,MSK<CR><LF>

Response:

$CRMSK,fff.f,X,ddd,Y,n*CS

Field	Description
fff.f	Frequency in kHz (283.5 to 325)
x	Tune mode (M = manual tune mode, A = automatic tune mode)
ddd	MSK bit rate (100, or 200 bps)
y	MSK rate selection (M = manual tune mode, A = automatic tune mode)
n	Period of output of performance status message, 0 to 100 seconds ($CRMSS)

The $GPCRQ,MSK query may be issued through the secondary serial port with a standard response issued to the same port. This will not affect the output of RTCM data from the main serial port, when the receiver has acquired a lock on a beacon station.

$GPCRQ Receiver Performance Status Query -

This standard NMEA query prompts the SBX for its performance status:

$GPCRQ,MSS<CR><LF>

Response:

$CRMSS,xx,yy,fff.f,ddd*CS

Field	Description
xx	Signal Strength (dB µV/m)
yy	Signal to Noise Ratio (dB)
fff.f	Frequency in kHz (283.5 to 325)
ddd	Frequency in kHz (283.5 to 325)

The $GPCRQ,MSS query may be issued through the secondary serial port with a standard response issued to the same port. This will not affect the output of RTCM data from the main serial port, when the receiver has acquired a lock on a beacon station.

Proprietary Queries

The following subsections describe the selection of valid Hemisphere GPS proprietary NMEA-0183 queries, and their responses.

$PCSI,0 Receiver Help Query -

This command queries the SBX for a list of available proprietary $PCSI commands:

$PCSI,0<CR><LF>

Response:

$PCSI,ACK,0
$PCSI,P003-0K,012
$PCSI,0 ->HELP Msg
$PCSI,1 ->Status line A,<T>,<S>
$PCSI,2 ->Status line B,<T>
$PCSI,3 ->Dump Search
$PCSI,4 ->Wipe Search
$PCSI,5 ->Port Rate,<P0>,<P1>
$PCSI,6 ->Reserved
$PCSI,7 ->RTCM Mode

$PCSI,1 Status Line A, Channel 0 -

This query commands the SBX to output a selection of parameters related to the operational status of its primary channel. It has the following format:

$PCSI,1<CR><LF>

Response:

$PCSI,ACK,1$PCSI,CS0,
PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,Q,ID,H,T

Field	Description
CSO	Channel 0
PXXX-Y.YYY	Resident SBX firmware version
S/N	SBX receiver serial number
fff.f	Channel 0 current frequency
M	Frequency Mode (‘A’ – Auto or ‘M’ – Manual)
ddd	MSK bit rate
R	RTCM rate
SS	Signal strength
SNR	Signal to noise ratio
MTP	Message throughput
Q	Quality number {0-25} – number of successive good 30 bit RTCM words received
ID	Beacon ID to which the receiver’s primary channel is tuned
H	Health of the tuned beacon [0-7]
T	$PCSI,1 status output period [0-99]

Optionally, the Status Line A query can be issued requesting the output of the response message at a specified output rate. It has the following format, where T is the output period in seconds:

$PCSI,1,T<CR><LF>

Response:

$PCSI,ACK,1$PCSI,CS0,PXXXY.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,Q,
ID,H,T

Cycling receiver power discontinues the output of this message. Additionally, message output can be halted by issuing the $PCSI,1<CR><LF> query without the output period field.

The response message has the same format as discussed above. In addition to this modified version of the Status Line A command, an additional S field may be placed after the T field, resulting in the following command:

$PCSI,1,T,S<CR><LF>

The S field is not a variable and specifies that the output of the Status Line A message should continue after the power has been cycled. To return the receiver to the default mode (in which message output ceases after receiver power is cycled) a $PCSI,1<CR><LF> query must be sent to the receiver.

The $PCSI,1 query may be sent through the either serial port for reporting of the full status of the primary receiver channel. The response to the query will be returned to the port that the command was issued. When querying the primary receiver channel using the secondary serial port, no interruptions in RTCM data output will occur on the primary port, provided the SBX has acquired a valid beacon.

$PCSI,2 Status Line B, Channel 1 -

This query commands the SBX to output a selection of parameters related to the operational status of its secondary channel. It has the following format:

$PCSI,2<CR><LF>

Response:

$PCSI,ACK,2$PCSI,CS1,
PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,Q,ID,H,T

Field	Description
CS1	Channel 1
PXXX-Y.YYY	Resident SBX firmware version
S/N	SBX receiver serial number
fff.f	Channel 1 current frequency
M	Frequency Mode ('A' – Auto or 'M' – Manual)
ddd	MSK bit rate
R	RTCM rate
SS	Signal strength
SNR	Signal to noise ratio
MTP	Message throughput
Q	Quality number {0-25} – number of successive good 30 bit RTCM words received
ID	Beacon ID to which the receiver's secondary channel is tuned
H	Health of the tuned beacon [0-7]
T	$PCSI,1 status output period [0-99]

Optionally, the Status Line B query can be modified to request the output of the response message once every period. It has the following format, where T is the output period in seconds:

$PCSI,2,T<CR><LF>

Response:

$PCSI,ACK,2$PCSI,CS0,
PXXX-Y.YYY,SN,fff.f,M,ddd,R,SS,SNR,MTP,Q,ID,H,T

The response message has the same format as discussed above. The Status Line B message output cannot be set to remain active after the power of the SBX has been cycled.

The $PCSI,2 query may be sent through the either serial port for reporting of the full status of the secondary receiver channel. The response to the query will be returned to the port that the command was issued. When querying the secondary receiver channel using the secondary serial port, no interruptions in RTCM data output will occur on the primary port, provided that SBX has acquired a valid beacon.

$PCSI,3 Receiver Search Dump -

This query commands the SBX to display the search information used for beacon selection in Automatic Beacon Search mode. The output has three frequencies per line.

$PCSI,3<CR><LF>

Response:

$PCSI,ACK,3

$PCSI,01,2835,209,0E,00,-0009,02,2840,339,0E,00,
-0012,03,2845,006,0E,00,0009$PCSI,04,2850,342,0E,00,-
0010,05,2855,547,0E,00,-0005,06,2860,109,0E,00,-0011
$PCSI,07,2865,188,0E,00,-0007,08,2870,272,0E,00,-
0004,09,2875,682,0E,00,-0006$PCSI,10,2880,645,0E,00,
-0007,11,2885,256,0E,00,-0009,12,2890,000,06,00,-0012
$PCSI,13,2895,132,0E,00,-0009,14,2900,281,0E,00,
-0010,15,2905,634,0E,00,-0008$PCSI,16,2910,172,0E,00,
-0007,17,2915,006,0E,00,-0009,18,2920,546,0E,00,-0014

$PCSI,19,2925,358,0E,00,-0008,20,2930,479,0E,00,-0009,21,2935,358,0E,00,-0011$PCSI,22,2940,853,0E,00,-0005,23,2945,588,0E,00,-0015,24,2950,210,0E,00,-0011 $PCSI,25,2955,000,06,00,-0011,26,2960,663,0E,00,-0010,27,2965,596,0E,00,-0009$PCSI,28,2970,000,06,00,-0009,29,2975,917,0E,00,-0009,30,2980,000,06,00,-0016$PCSI,31,2985,343,0E,00,-0013,32,2990,546,0E,00,-0010,33,2995,546,0E,00,-0010$PCSI,34,3000,172,0E,00,-0014,35,3005,006,0E,00,-0011,36,3010,1006,0E,00,-0009

$PCSI,37,3015,006,0E,00,-0015,38,3020,300,0E,00,-0013,39,3025,277,0E,00,-0100$PCSI,40,3030,479,0E,00,-0010,41,3035,006,0E,00,-0012,42,3040,050,0E,00,-0008$PCSI,43,3045,000,06,00,-0014,44,3050,172,0E,00,-0013,45,3055,000,06,00,-0011$PCSI,46,3060,000,06,00,-0011,47,3065,000,06,00,-0014,48,3070,000,06,00,-0010$PCSI,49,3075,000,06,00,-0012,50,3080,006,0E,00,-0015,51,3085,000,06,00,-0015$PCSI,52,3090,300,0E,00,-0007,53,3095,000,06,00,-0013,54,3100,000,06,00,-0013$PCSI,55,3105,000,06,00,-0012,56,3110,127,0E,00,-0013,57,3115,000,06,00,-0012$PCSI,58,3120,596,0E,00,-0012,59,3125,051,0E,00,-0009,60,3130,000,06,00,-0011$PCSI,61,3135,213,0E,00,-0008,62,3140,000,06,00,-0011,63,3145,000,06,00,-0015$PCSI,64,3150,302,0E,00,-0008,65,3155,000,06,00,-0009,66,3160,000,06,00,-0003$PCSI,67,3165,000,06,00,-0013,68,3170,000,06,00,-0011,69,3175,612,0E,01,0000$PCSI,70,3180,000,06,00,-0015,71,3185,000,06,00,-0008,72,3190,000,06,00,-0009$PCSI,73,3195,000,06,00,0011,74,3200,1002,0E,01,-0002,75,3205,067,0E,00,-0008

$PCSI,76,3210,001,0E,00,-0008,77,3215,000,06,00,-0009,78,3220,132,0E,00,-0009$PCSI,79,3225,000,06,00,-0010,80,3230,339,0E,00,-0013,81,3235,000,06,00,-0011$PCSI,82,3240,000,06,00,-0010,83,3245,202,0E,00,-0007,84,3250,006,0E,00,-0002

RAIM Commands and Messages

RAIM stands for receiver autonomous integrity monitoring. RAIM is a GPS integrity monitoring scheme that uses redundant ranging signals to detect a satellite malfunction that results in a large range error. The Hemisphere GPS products use RAIM to alert users when errors have exceeded a user specified tolerance. RAIM is available for SBAS, Beacon and OmniSTAR applications.

$JRAIM

This command allows the parameters of the RAIM scheme that affects the output of the $PSAT,GBS message to be setup.

This command has the following structure:

$JRAIM,HPR,probHPR,probFALSE<CR><LF>

Where:

Variable	Description
HPR	Horizontal Protection Radius (HPR). Notification in the $PSAT,GBS message that the horizontal error has exceeded this amount will be received. The acceptable range for this value is 1 to 10,000 meters. The default is 10 meters.
probHPR	Maximum allowed probability that the position computed lies outside the HPR. The acceptable range for this value is from 0.001 percent to 50 percent. The default is 5 percent.
probFALSE	Maximum allowed probability that there is a false alarm (That the position error is reported outside the of the HPR, but it is really within the HPR). The acceptable range for this value is from 0.001 percent to 50 percent. The default is 1 percent.

The Purpose of the Probability of False Alarm (probFALSE)

The purpose of the probability of false alarm is to help make a decision as to whether to declare a fault or warning in an uncertain situation. As an example, several satellites which may have failed, all giving similar probabilities of failure. One would put the user outside of the horizontal protection radius (HPR) and the others would not. always choose the one that puts the user outside of the radius to be safe, but in this situation, an error (the failed satellite may actually be another satellite) was likely made. So here, there is a high probability of false alarm. The high probability of false alarm can be used to downgrade a fault to a warning. If the probability of false alarm (probFALSE) is > 5 percent, (assuming this is the chosen threshold) a fault is downgraded to a warning.

There are other situations where the probability of false alarm is high due to geometry. Because of the geometry, get excessive faults may occur, even though the user is actually within the protection radius. Again, downgrading to a warning prevents excessive faults when the conditions are highly uncertain.

The philosophy is to only issue a fault if the user is relatively certain (to within the probability of a false alarm) that the protection radius has been exceeded, else issue a warning.

$PSAT,GBS Data Message

The GBS message is used to support Receiver Autonomous Integrity Monitoring (RAIM). In the table below, the GBS data message is broken down into its components. This message takes the following form:

```
$PSAT,GBS,hhmmss.ss,ll.l,LL.L,aa.a,ID,p.ppppp,b.b,s.s,flag*cc
```

$PSAT,GBS Data Message Defined

Field	Description
hhmmss.ss	UTC time in hours, minutes and seconds of the GGA or GNS fix associated with this sentence
ll.l	Expected error in latitude
LL.L	Expected error in longitude
aa.a	Expected error in altitude
ID	ID number of most likely failed satellite
p.ppppp	Probability of HPR fault
b.b	Estimate of range bias, in meters, on most likely failed satellite
s.s	Standard deviation of range bias estimate
flag	Good (0), Warning (1), Bad or Fault (2) Flag (based on horizontal radius
*cc	Checksum

To start outputting the $PSAT,GBS message once per second (the only output rate available), enter the following:

$JASC,GPGBS,1<CR><LF>

Or, to turn the $PSAT,GBS message off, send the following command.

$JASC,GPGBS,0<CR><LF

OmniSTAR Commands

This section details the commands accepted by the LX-1 OmniSTAR receiver to configure and monitor the OmniSTAR functionality of the receiver. Table 5-25 provides a brief description of the commands supported by the OmniSTAR sensor for its configuration and control.

Table 5-25: OmniSTAR Commands

Message	Description
$JLBEAM	This command requests the current spot beam tables in use by the OmniSTAR receiver.
$JLXBEAM	This command requests debug information for the current spot beam tables.
$JOMS	This command requests the OmniSTAR engine to provide the current subscription information for the OmniSTAR service.
$JOMR	This command requests the OmniSTAR receiver to provide raw OmniSTAR region information.
$JFREQ	This command tunes the OmniSTAR receiver either in automatic mode or manually.

The following subsections provide detailed information relating to the use of each command.

Note: Please save any changes that need to be kept beyond the current power-up by using the $JSAVE command and wait for the $>SAVE COMPLETE response.

$JLBEAM

This command displays the current spot beams used by the OmniSTAR receiver.

This command has the following layout:

$JLBEAM<CR><LF>

The receiver will output the following data:

$>JLBEAM,Sent frequency1,Used frequency2,Baud xxx,Geo xxx (1)
$>JLBEAM,frequency1,longitude1,latitude1,symbol1,satlongitude1 (2)
.
.
.

$>JLBEAM,frequencyn,longituden,latituden,baud,satlongituden

The first line of this output is described in the following table:

Data Field	Description
$JLBEAM	Message Header
Sent frequency	This field provides the frequency sent to the digital signal processor.
Used frequency	This field provides the frequency currently being used by the digital signal processor.
Bad xxxx	This data field provides the currently used baud rate of the acquired signal.
Geo xxx	This field provides the currently used satellites longitude, in degrees.

The second line, and those that follow, are described in the following table:

Data Field	Description
$>JLBEAM	Message Header
frequency	This data field provides the frequency of the spot beam.
longitude	This data field indicates the longitude of the center of the spot beam, in degrees.
latitude	This data field indicates the latitude of the center of the spot beam, in degrees.
baud	This field indicates the baud rate at which this spot beam is modulated.
satlongitude	This data field provides the satellites longitude, in degrees.

An example of this response follows:

```
$>JLBEAM,Sent 1551.4890,Used 1551.4890,Baud 1200,Geo -101
$>JLBEAM,1556.8250,-88,45,1200,(-101)
$>JLBEAM,1554.4970,-98,45,1200,(-101)
$>JLBEAM,1551.4890,-108,45,1200,(-101)
$>JLBEAM,1531.2300,25,50,1200,(16)
$>JLBEAM,1535.1375,-75,0,1200,(-98)
$>JLBEAM,1535.1375,-165,13,1200,(-98)
$>JLBEAM,1535.1525,20,6,1200,(25)
$>JLBEAM,1558.5100,135,-30,1200,(160)
$>JLBEAM,1535.1375,90,15,1200,(109)
$>JLBEAM,1535.1375,179,15,1200,(109)
```

$JLXBEAM

This command displays the current spot beams used by the OmniSTAR receiver.

This command has the following layout:

$JLBEAM<CR><LF>

The receiver will output the following data:

$>JLBEAMEX,0 (1)
$> Table:0 (2)
$> Beam:1,DDSfreq1,long1,lat1,symbol1,satlong1
.
.$> Beam:n,DDSfreqn,longn,symboln,satlongn
$> Table:1

Data Field	Description
DDSfreq	This field provides the DDS frequency
long	This variable is the longitude of the spot beam centroid
lat	This field provides the latitude of the spot beam centroid
symbol	This data field indicates the symbol rate used for that particular spot beam
satlong	This field provides the longitude of the L-band satellite

An example of this response follows:

$>JLBEAMEX,0

$> Table:0
$> Beam:0,1753247034,-88,45,1200,-101
$> Beam:1,1750643210,-98,45,1200,-101
$> Beam:2,1747278819,-108,45,1200,-101
$> Beam:3,1724619511,25,50,1200,16
$> Beam:4,1728989976,-75,0,1200,-98
$> Beam:5,1728989976,-165,13,1200,-98
$> Beam:6,1729006753,20,6,1200,25
$> Beam:7,1755131675,135,-30,1200,160
$> Beam:8,1728989976,90,15,1200,109
$> Beam:9,1728989976,179,15,1200,109
$> Table:1

$JOMS

This command requests the raw OmniSTAR subscription information and has the following form:

$JOMS

The receiver will respond with the following message:

$>JOMS,Opt,Source,Type,AccrReduction,StartDate,EndDate,HourGlass,ExtentionTime,LinkVector,SoftwareVersion

This message is summarized in the following table.

Data Field	Description
Ots	This field indicates a WET or DRY subscription
Source	RTCM source ID or ALL if VBS
AccrReduction	0 is most accurate
StartDate	Subscription end date
HourGlass	Seconds of metered time
Extension Time	Seconds of extension
Link Vector	Hexadecimal mask of links
SoftWareVersion	This item shows the OmniSTAR library version

An example of this response follows:

```
$>JOMS,DRY,ALL,VBS,0,01/06/2000,01/06/2001,0,0,1E00,1.43
```

$JOMR

This command displays raw OmniSTAR region information and has the following structure:

$JOMR

The receiver will respond with the following messages:

$JOMR,1,latitude1,longitude1,radius1<CR><LF>
$JOMR,2,latitude2,longitude2,radius2<CR><LF>
$JOMR,3,latitude3,longitude3,radius3<CR><LF>
$JOMR,4,latitude4,longitude4,radius4<CR><LF>
$JOMR,5,latitude5,longitude5,radius5<CR><LF>

Where latitude and longitude are expressed in radians and the radius is in meters.

If the receiver has an active subscription, the first line should show the inclusion area. The subsequent lines will show additional inclusion and/or exclusion areas. A negative radius indicates that the region is an exclusion zone. An example follows:

$>JOMR,1,.994787,-1.605694,4500000.000
$>JOMR,2,0.000000,0.000000,0.000000
$>JOMR,3,0.000000,0.000000,0.000000
$>JOMR,4,0.000000,0.000000,0.000000
$>JOMR,5,0.000000,0.000000,0.000000

$JFREQ

This message allows the OmniSTAR receiver to be either manually or automatically tuned.

This command has the following structure:

$JFREQ,freq,symb<CR><LF>

Where “freq” is the frequency in kHz and “symb” is the symbol rate (1200 or 2400 baud).

The receiver will reply with the following response:

$>

Entering a frequency of zero with no associated symbol rate will place the OmniSTAR engine into automatic mode. Entering a valid frequency and symbol rate will manually tune the receiver.

The following table provides frequency information for the OmniSTAR satellites.

Coverage Area	**Longitude**	**Frequency**	**Baud Rate**	**Sat. Name**
Eastern U.S.	101 West	1556.825	1200	AMSC-E
Central U.S.	101 West	1554.497	1200	AMSC-C
Western U.S.	101 West	1551.489	1200	AMSC-W
Central America, South America, Caribbean	98 West	1535.1375	1200	Am-SAT

Coverage Area	Longitude	Frequency	Baud Rate	Sat. Name
West Africa, South Africa, Asia, Pacific Islands	109 East	1535.1375	1200	AP-SAT
East Africa, Middle East	25 East	1536.1525	1200	EA-SAT
Australia, Far East	160 East	1535.185	1200	OCSAT
Europe	16 East	1531.230	1200	EMS

Note: Sending this command does not require send a $JSAVE command to be sent to save changes to the tuning of the OmniSTAR egnine.

OmniSTAR HP

For Eclipse receiver's OmniStar HP initialization time can be reduced by supplying the known position. If the current position coordinates are known very accurately, the OmniSTAR algorithm can be sent with the known coordinates.

Warning!

The coordinates accurately should be known to within 2 cm (1 inch) before attempting to seed the position. Any errors entered here will effect the future accuracy of the position solution.

The current position can be queried and stored with the commands. -

Command Option 1: $JHP,POS,P:

This will initialize the OmniSTAR HP algorithm with the receiver's present location. If the current latitude, longitude and altitude standard deviations are cumulatively greater than 0.6m, the current position is not stable and the command is ignored. Under this condition, the system responds with the message, "Present Location Not Stable."

Command Option 2: $JHP,POS:

Returns current user position.

Command Option 3: $JHP,POS,lat,lon,height:

Where "lat" and "lon" are the user's latitude and longitude in degrees; and "height" is in meters.

The user can seed the OmniSTAR algorithm with a position, to speed up initialization, with the following command.

Command: $JHP,SEED, lat,lon,height:

Where "lat" and "lon" are the user's latitude and longitude in degrees; and "height" is in meters.

When the current receiver position is greater than 12 meters (in the horizontal plane) from the seed position, the receiver responds with, "Current Position Too Far From Seed," and aborts the command.

6: Binary Data

Binary Message Structure
Binary Message

Binary Message Structure

The receiver supports a selection of binary data messages that provide improved communication port efficiency.

The binary messages supported by the receiver are in an Intel Little Endian format for direct read in a PC environment. More information on this format at the following web site:
http://www.cs.umass.edu/~verts/cs32/endian.html

Each binary message begins with an 8-byte header and ends with a carriage-return, line-feed pair (0x0D, 0x0A). The first four characters of the header is the ASCII sequence $BIN.

Table 6-1, on page 199, provides the general binary message structure.

Table 6-1: Binary Message Structure

Group	Components	Type	Bytes	Value
Header	Synchronization String	4 byte string	4	$BIN
	Block ID - a number which tells the type of binary message	Unsigned short	2	1, 2, 80, 93, 94, 95, 96, 97, 98, or 99
	DataLength - the length of the binary messages	Unsigned short	2	52, 16, 40, 56, 96, 128, 300, 28, 68, or 304
Data	Binary Data - varying fields of data with a total length of *DataLength* bytes	Mixed fields	52, 16, 40, 56, 96, 128, 300, 28, 68, or 304	Varies - see message tables
Epilogue	Checksup - sum of all bytes of the data (all Data length bytes). The sum is placed in a 2 byte integer	Unsigned short	2	Sum of data bytes
	CR- Carriage return	Byte	1	0D hex
	LF - Line feed	Byte	1	0A hex

The total length of the binary message packet is DataLength plus 12 (8 byte header, 2 byte checksum and 2 bytes for CR, LF).

Bin 1

This message has a BlockID of 1 and is 52 bytes, excluding the header and epilogue. It consists of GPS position and velocity data. It is the only binary message that can be output at a rate of 5 Hz. Table 6-2 describes the content of this message.

Table 6-2: Bin 1 Message

Name	Components	Type	Bytes	Value
AgeOfDiff	Age of differential, seconds. Use Extended AgeOfDiff first. If both = 0, then no differential	Byte	1	0 to 255
NumOfSats	Number of satellites used in the GPS solution	Byte	1	0 to 12
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
GPSTimeOfWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
Latitude	Latitude in degrees north	Double	8	-90.0 to 90.0
Longitude	Longitude in degrees East	Double	8	-180.0 to 180.0
Height	Altitude above the ellipsoid in meters	Float	4	
VNorth	Velocity north in m/s	Float	4	
VEast	Velocity east in n/s	Float	4	
Vup	Velocity up in m/s	Float	4	
StdDevResid	Standard deviation of residuals in meters	Float	4	Positive

Table 6-2: Bin 1 Message

Name	Components	Type	Bytes	Value
NavMode	Navigation mode: 0 = No fix 1 = FIX_2D 2 = FIX_3D (or FIX_3d and solving ambiguities if rover) 3 = FIX_2D and Diff 4 = FIX_3D Diff (not solving ambiguities if rover) 5 = RTK Search 6 = FIX_3D and Diff and RTK solution If bit 7 is set (left-most bit, then this is a manual position	Unsigned short	2	Bits 0 through 6 = Navemode Bit 7 = Manual mark
Extended AgeOfDiff	Extended age of differential, seconds. If 0, use 1 byte AgeOfDiff listed above	Unsigned short	2	0 to 65536

Bin 2

This message has a BlockID of 2 and is 16 bytes, excluding the header and epilogue. This message contains various quantities that are related to the GPS solution. Table 6-3 describes the details of this message in order.

Table 6-3: Bin 2 Message

Name	Components	Type	Bytes	Value
MaskSatsTracked	A mask of satellites tracked by the GPS. Bit 0 corresponds to the GPS satellite with PRN 1.	Unsigned long	4	Individual bits represent satellites
MaskSatsUsed	A mask of satellites used in the GPS solution. Bit 0 corresponds to the GPS satellite with PRN 1.	Unsigned long	4	Individual bits represent satellites
GPSUtcDiff	Whole seconds between UTC and GPS time (GPS minus UTC)	Unsigned short	2	Positive
HDOPTimes10	Horizontal dilution of precision scaled by 10 (0.1 units)	Unsigned short	2	Positive
VDOPTimes10	Vertical dilution of precision scaled by 10 (0.1 units)	Unsigned short	2	Positve
WAAS PRN bitmask	PRN and tracked or used status masks	Unsigned short	2	See page 193

WAAS PRN mask:

- Bit 00 Mask of satellites tracked by first WAAS satellite
- Bit 01 Mask of satellites tracked by second WAAS satellite
- Bit 02 Mask of satellites used by first WAAS satellite
- Bit 03 Mask of satellites used by second WAAS satellite
- Bit 04 Unused
- Bit 05-09 Value used to find PRN of first WAAS satellite (This value + 120 = PRN)
- Bit 10-14 Value used to find PRN of second WAAS satellite (This value + 120 = PRN)
- Bit 15 Unused

Bin 76

Note: “Code” means pseudorange derived from code phase. “Phase means range derived from carrier phase. This will contain cycle ambiguities.

Only the lower 16 bits of L1P code, L2P code and the lower 23 bits of carrier phase are provided. THe upper 19 bits of the L1CA code are found in m_aulCACodeMSBsPRN[]. The upper 19 bits of L1P or L2P must be derived using the fact L1P and L2P are within 128 meters (419.9 feet) of L1CA.

To determine L1P or L2P:

1. Use the lower 16 bits provided in the message.
2. Set the upper buts to that of L1CA.
3. Add or subtract on LSB of the upper bits (256 meters (839.9 feet)) so that L1P or L2P are with in 1/2 LSB (128 meters (419.9 feet))

THe carrier phase is in units of cycles, rather than meters, and is held to within 1023 cycles of the respective code range. Only the lower 16+7 = 23 bits of carrier phase are transmitted in Bin 76.

To determine the remaining bits:

1. Convert the respective code range (determined above) into cycles by dividing by the carrier wavelength. This is the nominal reference phase.
2. Extract the 16 and 7 bit blocks of carrier phase from bin 76 and arrange it to form the lower 23 bits of carrier phase.
3. Set the upper bits (bit 23 and above) equal to those of the nominal reference phase

4. Add or subtract the least significant upper bit (8192 cycles) so that carrier phase most closely agrees with the nominal reference phase (to within 4096 cycles).

Table 6-4: Bin 76 Message Structure (length = 8 = (448) + 2 + 2 = 460)

Name	Components	Type	Bytes	Values
Message Header			8	
TOW	Predicted GPS time in seconds	double	8	
Week	GPS week number	Unsigned short	2	
Spare 1		Unsigned short	2	
Spare 2		Unsigned long	4	
L2PObs[12]	L2 satellite observation data	Structure array	12 x 12 = 144	See Table 6-5
L1CAObs[15]	L1 satellite code observation data	Structure array	15 x 12 = 180	See Table 6-6
L1CACodeMSBsPRN[15]	L1CA code observation	Array of 15 unsigned long	15 x 4 = 60	See Table 6-11
L1PCode[12]	L1(P) code observation data	Array of 12 unsigned long	12 x 4 = 48	See Table 6-12
wCeckSum	Sum of all bytes of header and data	Unsigned short	2	
wCRLF	Carriage return line feed	Unsigned short	2	

Table 6-5: L2PSatObs[12] Structure

Name	Components	Type	Bytes
CS_TT_W3_SNR	See Table 6-7, on page 207	Unsigned long	4
P7_Doppler_FL	See Table 6-9, on page 208	Unsigned long	4
CideAndPhase	See Table 6-10, on page 209	Unsigned long	4

Table 6-6: L1CASatObs[12] Structure

Name	Components	Type	Bytes
CS_TT_W3_SNR	See Table 6-8, on page 208	Unsigned long	4
P7_Doppler_FL	See Table 6-9, on page 208	Unsigned long	4
CideAndPhase	See Table 6-10, on page 209	Unsigned long	4

Table 6-7: CS_TTW3_SNR Structure for L2PSatObs[12]

Name	Description	Bits	Location	Value
SNR		12	0-11	10.0 X log10(0.1164xSNR_value
Cycle Slip Warn	Warning for potential 1/2 cycle	3	14-12	A warning exists if any of these bits are set
Long Track Time		1	15	1 if Track Time > 25.5 sec, 0 otherwise
Track Time	Signal tracking time in seconds	8	23-16	LSB = 0.1 seconds Range = 0 to 25.5 seconds
Cyle Slips		8	31-24	Increments by 1 every cycle slip with natural roll-over after 255

Table 6-8: CS_TT_W3_SNR Structure for L1CASatObs[15]

Name	Description	Bits	Location	Value
SNR		12	0-11	10.0 x Log10(0.1024 x SNR_value)
Cycle Slip Warn		3	14-12	Warning for potential 1/2 cycle slips. Warning exists if any of these bits are set.
Long Track Time		1	15	1 if track time > 25.5 sec, 0 otherwise
Track Time		8	23-16	LSB = 0.1 seconds Range = 0 to 25.5 seconds
Track Time	Signal tracking time in seconds	8	23-16	LSB = 0.1 seconds Range = 0 to 25.5 seconds
Cycle Slips		8	31-24	Increments by 1 every cycle slip with natural roll-over after 255

Table 6-9: Bin 6-9 76 P7_Doppler_FL Structure for Both L2PSatObs[12] and L1CASatObs[15]

Name	Description	Bits	Location	Value
Phase Valid	Boolean	1	0	1 if valid phase 0 otherwise
Doppler	Magnitude of Doppler	23	1-23	LSB = 1/512 cycle/sec Range = 0 to 16384 cycle/sec
Doppler Sign	Sign of Doppler	24	24	1 = negative 0 = positive
Carrier Phase (High port)	Upper 7 bits of the 23 bit carrier phase	7	25-31	LSB = 64 cycles MSB = 4096 cycles

Table 6-10: Bin 76 CodeAndPhase structure for L2PSatObs[12] and L1CASatObs[15]

Name	Description	Bits	Location	Value
Pseudo Range	Lower 16 bits of code pseudorange	16	0-15	LSB = 1/256 meters MSB = 128 meters Note: For CA code, the upper 19 bits are given in L1CACodeMSBsPRN[] in Table 6-4
Carrier Phase	Lower 16 bits of the carrier phase	16	16-31	LSB = 1/1024 cycles MSB = 32 cycles **Note:** The 7 MSBs are given in P7_Doppler_FL_ within Table 6-9

Table 6-11: Bin 76 message for L1CACodeMSBsPRN[15] Structure

Name	Description	Bits	Location	Value
PRN	Space vehicle ID	8	0-7	PRN = 0 if no data
Unused	Unused	5	8-12	
L1CA Range	Upper 19 bits of L1CA	19	31-13	LSB = 256 meters MSB = 67,108,864 meters

Table 6-12: Bin 76 Message for L1PCode[12] Structure

Name	Description	Bits	Location	Value
L1P Range	Lower 16 bits of the L1P code pseudo range	16	0-15	LSB = 1/256 meters MSB = 128 meters
L1P SNR	L1P Signal to Noise Ratio	12	27-16	SNR = 10.0 x log(0.1164 x SNR_value) If 0, then L1P channel not tracked
Unused	Unused	4	31-28	

Bin 80

This message has a BlockID of 80 and is 40 bytes, excluding the header and epilogue. This message contains the WAAS message. Table 6-13 describes the constituents of this message in order.

Table 6-13: Bin 80 Message

Name	Components	Type	Bytes	Value
PRN	Broadcast PRN	Unsigned short	2	
Spare	Not used at this time	Unsigned short	2	Future use
MsgSecOfWeek	Seconds of week for message	Unsigned long	4	
WaasMsg[8]	250 bit WAAS message (RTCA DO0229). 8 unsigned longs, with most significant bit received first	Unsigned long	4 x 8 = 32	

Bin 93

This message has a BlockID of 93 and is 45 bytes excluding the header and epilogue. This message contains information relating to the WAAS ephemeris. Table 6-14 describes the contents of this message in order.

Table 6-14: Bin 93 Message

Name	Components	Type	Bytes	Value
SV	Satellite to which this data belongs	Unsigned short	2	
Spare	Not used at this time	Unsigned short	2	Future use
TOWSecOfWeek	Time at which this arrived (LSB = 1 sec)	Unsigned long	4	
IODE		Unsigned short	2	
URA	Consult the ICD-GPS-200 for definition in Appendix A	Unsigned short	2	
TO	Bit 0 = 1 sec	Long	4	
XG	Bit 0 = 0.08 m	Long	4	
YG	Bit 0 = 0.08 m	Long	4	
ZG	Bit 0 = 0.4 m	Long	4	
XGDot	Bit 0 = 0.000625 m/sec	Long	4	
YXDot	Bit 0 = 0.000625 m/sec	Long	4	
ZGDot	Bit 0 = 0.004 m/sec	Long	4	
XGDotDot	Bit 0 = 0.0000125 m/sec/sec	Long	4	
YGDotDot	Bit 0 = 0.0000125 m/sec/sec	Long	4	
ZGDotDot	Bit 0 = 0.0000625 m/sec/sec	Long	4	

Table 6-14: Bin 93 Message

Name	Components	Type	Bytes	Value
Gf0	Bit 0 = 2**-31 sec	Unsigned short	2	
Gf0Dot	Bit 0 = 2**-40sec/sec	Unsigned short	2	

Bin 94

This message has a BlockID of 94 and is 96 bytes, excluding the header and epilogue. This message contains ionospheric and UTC conversion parameters. Table 6-15 describes the details of this message in order.

Table 6-15: Bin 94 Message

Name	Components	Type	Bytes	Value
a0, a1, a2, a3	AFCRL alpha parameters	Double	8 x 4 = 32	
b0, b1, b2, b3	AFCRL beta parameters	Double	8 x 4 = 32	
A0, A1	Coefficients for determining UTC time	Double	8 x 2 = 16	
tot	Reference time for A0 and A1, second of GPS week	Unsigned long	4	
wnt	Current UTC reference week	Unsighned short	2	
wnlsf	Week number when dtlsf becomes effective	Unsighned short	2	
dn	Day of week (1-7) when dtlsf becomes effective	Unsighned short	2	
dtls	Cumulative past leap	Short	2	
dtlsf	Scheduled future leap	Short	2	
Spare	Not used at this time	Unsigned Short	2	Future use

Bin 95

This message has a BlockID of 95 and is 128 bits, excluding the header and epilogue. This message contains ephemeris data of all 12 channels. Table 6-16 describes the contents of this message in order.

Table 6-16: Bin 95 Message

Name	Components	Type	Bytes	Value
SV	The satellite to which this data belongs	unsigned short	2	
Spare1	Not used at this time	unsigned short	2	Future use
SecOfWeek	Time at which this arrived (LSB = 6)	unsigned long	4	
SF1words[10]	Unparsed SF 1 message	unsigned long	4 x 10 = 40	
SF2words[10]	Unparsed SF 2 message	unsigned long	4 x 10 = 40	
SF3words[10]	Unparsed SF 3 message	unsigned long	4 x 10 = 40	

Bin 96

This message has a BlockID of 96 and is 300 bytes excluding the header and epilogue. This message contains phase and code data. Table 6-17, on page 215, to Table 6-20, on page 217, describes the constituents of this message in order.

Table 6-17: Bin 96 Messages

Name	Components	Type	Bytes	Values
Spare1	Not used at this time	Unsigned short	2	Future use
Week	GPS week number	Unsigned short	2	
TOW	Predicted GPS time in seconds	Double	8	
Sat Obs	Observation data	Structure array	24 x 12= 288	See Table 6-9, on page 208

Table 6-18: Sat Obs Array (From Message Type 96)

Name	Components	Type	Bytes
UNICS_TT_SNR_PRN	See Table 6-10, on page 209	Unsigned long	4
UIDopplet_FL[12]	See Table 6-11, on page 209	Unsigned long	4
PseudoRange[12]	Pseudo range	Double	8
Phase[12]	Phase (m) L1 wave = 0.190293672798365	Double	8

Table 6-19: UNICS_TT_SNR_PRN [12] Structure

Name	Description	Bits	Location	Value
PRN	Pseudorandom noise	8	0-7	PRN is 0 if no data
SNR	Signal-to noise ratio	8	8-15	SNR=10.0 *$\log_{10}$* (0.8192*SNR)
PTT	PhaseTrackTime	8	16-23	In units of 1/10 sec range=0 to 25 sec (see next field)
CSC	CycleSlip Counter	8	24-31	Increments by 1 every cycle with natural rollover after 255

Table 6-20: UIDoppler_FL Structure

Name	Bits	Location	Value
Phase	1	0	1 if valid 0 otherwise
TrackTime	1	1	1 if track time > 25.5 seconds 0 otherwise
Unused	2	2-3	Unused
Doppler	28	4-31	Signed (two's compliment) Doppler in units of m/sec x 4096. (i.e., LSB=1/4096), range = +/- 32768 m/sec. Computed as phase change over 1/10 sec.

Bin 97

This message has a BlockID of 97 and is 28 bytes, excluding the header and epilogue. This message contains statistics for processor utilization. Table 6-21 describes the details of this message in order.

Table 6-21: Bin 97 Messages

Name	Components	Type	Bytes	Value
CPUFactor	CPU utilization factor. Multiply by 450e-06 to get percentage of spare CPU that is available **Note:** This field is only relevant on the old SLX platforms and Eclipse platform. It is not relevant for the Crescent receivers.	Unsigned long	4	Positive
MissedSubFrame	The total number of missed sub frames in the navigation message since power on	Unsigned short	2	Positive
MaxSubFramePnd	Max sub frames queued for processing at any one time	Unsigned short	2	Positive
MissedAccum	The total number of missed code accumulation measurements in the channel-tracking loop	Unsigned short	2	Positive
MissedMeas	The total number missed pseudorange measurements	Unsigned short	2	Positive
Spare 1	Not used at this time	Unsigned long	4	Future Use
Spare 2	Not used at this time	Unsigned long	4	Future use
Spare 3	Not used at this time	Unsigned long	4	Future use

Table 6-21: Bin 97 Messages

Name	Components	Type	Bytes	Value
Spare 4	Not used at this time	Unsigned short	2	Future use
Spare 5	Not used at this time	Unsigned short	2	Future use

Bin 98

This message has a BlockID of 98 and is 68 bytes, excluding the header and epilogue. This message contains data derived from the satellite almanacs. Table 6-22, on page 219, and Table 6-23, on page 220, describes the contents of this message in order.

Table 6-22: Bin 98 Message

Name	Components	Type	Bytes	Value
AlmanData	Almanac-derived-data, 8 satellites at a time	Structure array	8 x 8 = 64	See the Table 6-23, on page 220
LastAlman	Last almanac processed	Byte	1	0 to 31
IonoUTCVFlag	Flag that is set when ionoshpere modeling data is extracted from the GPS sub frame 4	Byte	1	0 = not logged 2 = valid
Spare	Not used at this time	Unsigned short	2	Future use

Table 6-23: AlmanData Structure Array

Name	Components	Type	Byte	Value
DoppHz	Predicted Doppler in HZ for the satellite in question (assuming a stationary satellite)	Short	2	
CountUpdate	Number of times the almanac has changed for this satellite since the receiver was turned on	Byte	1	Positive
Svindex	Channel number (groups of 8)	Byte	1	0 to 7 8 to 15 16 to 23 24 to 31
AlmVFlag	Almanac valid flag	Byte	1	0 = not logged 1 = invalid 2 = valid 3 = has data (not yet validated)
AlmHealth	Almanac health from sub frame 4 of the GPS message	Byte	1	See ICD-GPS-200 in Appendix A
Elev	Elevation angle in degrees	Char	1	-90 to 90
Azimuth	1/2 the azimuth in degrees	Byte	1	0 to 180 represents 360 degrees

Bin 99

This message has a BlockID of 99 and is 304 bytes, excluding the header and epilogue. This message contains quantities related to the tracking of the individual GPS satellites, along with some other relevant data. Table 6-24, on page 221, and Table 6-25, on page 222, describes the constituents of this message in order.

Table 6-24: Bin 99 Message

Name	Components	Type	Bytes	Value
NavMode2	Navigation mode data (lower 3 bits hold the GPS mode, upper bit set if differential is available)	Byte	1	Lower 3 bits take on the values 0 = time not valid 1 = No fix 2 = 2D fix 3 = 3D fix Upper bit (bit 7) is 1 if differential is available
UTCTimeDiff	Whole seconds between UTC and GPS time (GPS minus UTC)	Byte	1	Positive
GPSWeek	GPS week associated with this message	Unsigned short	2	0 to 65536
GPSTimeofWeek	GPS tow (sec) associated with this message	Double	8	0.0 to 604800.0
ChannelData	12 structure (see Table 6-25, on page 222) containing tracking data for each of the 12 receiver channels	Structure array	12 x 24 = 288	See Table 6-25, on page 222

Table 6-24: Bin 99 Message

Name	Components	Type	Bytes	Value
ClockErrAtL1	The clock error of the GPS clock oscillator at L1 frequency in Hz	Short	2	-32768 to 32768
Spare	Not used at this time	Unsigned short	2	Future use

Table 6-25: ChannelData Array

Name	Components	Type	Bytes	Value
Channel	Channel number	Byte	1	0 to 12
SV	Satellite being tracked, 0 = not tracked	Byte	1	0 to 32
Status	Status bit mask (code carrier bit frame...)	Byte	1	Bit 0 = code lock 1 = carrier lock 2 = bit lock 3 = frame sync 4 = frame sync and new epoch 5 = channel reset 6 = phase lock 7 = spare
LastSubframe	Last sub frame processed in the GPS message	Byte	1	1 to 5
EphmvFlag	Ephemeris valid flag	Byte	1	0 = not logged 1 = invalid 2 = valid 3 = has data (not yet validated)

Table 6-25: ChannelData Array

Name	Components	Type	Bytes	Value
EphmHealth	Satellite health from sub frame 1 of the GPS message	Byte	1	See ICD-GPS-200 in Appendix A
AlmVFlag	Almanac valid flag	Byte	1	0 = not logged 1 = invalid 2 = valid 3 = has data (not yet validated)
AlmHealth	Almanac health from sub frame 4 of the GPS message	Byte	1	See ICD-GPS-200 in Appendix A
Elev	elevation angle in degrees	Char	1	-90 to 90
Azimuth	1/2 the azimuth in degrees	Byte	1	0 to 180 degrees represents 0 to 360 degrees
URA	User range error from sub frame 1 of the GPS message	Byte	1	See ICD-GPS-200 in Appendix A
Spare	Not used at this time	Byte	1	Future use
CliForSNR	Code lock indicator for SNR (SNR = 10.0 * log10(4096*CliForSNR/ Nose_Floor = 80000.0)	Unsigned short	2	Positive
DiffCorr	100 times the differential correction for this channel's pseudorange	Short	2	
PosResid	10 times the position residual from the GPS solution for this channel	Short	2	

Table 6-25: ChannelData Array

Name	Components	Type	Bytes	Value
VelResid	10 times the velocity residual from the GPS solution for this channel	Short	2	
DoppHZ	Expected Doppler for this channel in Hz	Short	2	
NCOHz	Carrier track offset for this channel in Hz	Short	2	

Binary Messages

This section provides the code for the binary messages that Hemisphere GPS uses.

Note: Due to the code length, some code on the right side may wrap to a new line. The code is aligned to be on the right side in these cases.

SBinaryMsg Header

```
typedef struct
{
char m_strSOH[4];                /* start of header ($BIN) */
unsigned short m_byBlockID;     /* ID of message (1,2,99,98,97,96,95,
                                94,93 or 80 ) */
unsigned short m_wDataLength; /* 52 16,304,68,28,300,128,96,56, or 40
                                */} SBinaryMsgHeader;
typedef struct
{
unsigned long ulDwordPreamble; /* 0x4E494224 = $BIN */
unsigned long ulDwordInfo;        /* 0x00340001 or 0x00100002 or
                                 0x01300063 */
} SBinaryMsgHeaderDW;          /* or 0x00440062 or 0x001C0061 or
                                0x012C0060 */
                               /* or 0x0080005F or 0x0060005E
                                or0x0038005D */
                               /* or 0x00280050 */
#define BIN_MSG_PREAMBLE 0x4E494224       /* $BIN = 0x4E494224 */
#define BIN_MSG_HEAD_TYPE1 0x00340001    /* 52 = 0x34 */
#define BIN_MSG_HEAD_TYPE2 0x00100002    /* 16 = 0x10 */
#define BIN_MSG_HEAD_TYPE99 0x01300063  /* 99 = 0x63, 304 = 0x130
                                                 */
#define BIN_MSG_HEAD_TYPE100 0x01300064 /* 100 = 0x64, 304 =
                                                   0x130 */
#define BIN_MSG_HEAD_TYPE98 0x00440062 /* 98 = 0x62, 68 = 0x44 */
#define BIN_MSG_HEAD_TYPE97 0x001C0061 /* 97 = 0x61, 28 = 0x1C */
#define BIN_MSG_HEAD_TYPE96 0x012C0060  /* 96 = 0x60, 300 = 0x12C
```

```
                                          */
#define BIN_MSG_HEAD_TYPE95 0x0080005F /* 95 = 0x5F, 128 =
                                          0x80 */
#define BIN_MSG_HEAD_TYPE94 0x0060005E /* 94 = 0x5E, 96 = 0x60 */
#define BIN_MSG_HEAD_TYPE93 0x0038005D /* 93 = 0x5D, 56 = 0x38 */
#define BIN_MSG_HEAD_TYPE91 0x0198005B /* 91 = 0x5B, 408 = 0x198
                                          = total size in bytes
                                    -8 -2 -2*/

#define BIN_MSG_HEAD_TYPE80 0x00280050 /* 80 = 0x50, 40 = 0x28 */
#if !defined(OLD_BIN76)
#define BIN_MSG_HEAD_TYPE76 0x0190004C /* 76 = 0x4C, 400 = 0x190
                                          = total size in bytes
                                         -8 -2 -2*/
#else
#define BIN_MSG_HEAD_TYPE76 0x0198004C /* 76 = 0x4C, 408 = 0x198
                                          = total size in bytes
                                          bytes -8 -2 -2*/
#endif
#define BIN_MSG_HEAD_TYPE71 0x01980047 /* 71 = 0x47, 408 = 0x198
                                          = total size in
                                          bytes -8 -2 -2*/
#define BIN_MSG_CRLF 0x0A0D                 /* CR LF = 0x0D, 0x0A */
#define CHANNELS_12 12
#if MAXCHANNELS > CHANNELS_12
#error Need to increase CHANNELS_12 in Structs.h
#endif
typedef union
{
SBinaryMsgHeader sBytes;
SBinaryMsgHeaderDW sDWord;
} SUnionMsgHeader;
```

SBinaryMsg1

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned char m_byAgeOfDiff;      /* age of differential, seconds
                                     (255 max)*/
unsigned char m_byNumOfSats;      /* number of satellites used (12
                                     max) */
unsigned short m_wGPSWeek;        /* GPS week */
double m_dGPSTimeOfWeek;          /* GPS tow */
double m_dLatitude;               /* Latitude degrees, -90..90 */
double m_dLongitude;              /* Longitude degrees, -180..180 */
float m_fHeight;                  /* (m), Altitude ellipsoid */
float m_fVNorth;                  /* Velocity north m/s */
float m_fVEast;                   /* Velocity east m/s */
float m_fVUp;                     /* Velocity up m/s */
float m_fStdDevResid;             /* (m), Standard Deviation of z
                                     Residuals */
unsigned short m_wNavMode;
unsigned short m_wAgeOfDiff;  /* age of diff using 16 bits */
unsigned short m_wCheckSum; /* sum of all bytes of the data*/
unsigned short m_wCRLF;       /* Carriage Return Line Feed */
} SBinaryMsg1;                /* length = 8 + 52 + 2 + 2 = 64 */
```

SBinaryMsg2

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned long   m_ulMaskSatsTracked;  /* SATS Tracked, bit mapped
                                         0..31 */
unsigned long  m_ulMaskSatsUsed;       /* SATS Used, bit mapped
                                         0..31 */
unsigned short m_wGpsUtcDiff;          /* GPS/UTC time difference
                                         (GPS minus UTC)*/
unsigned short m_wHDOPTimes10;         /* HDOP (0.1 units) */
unsigned short m_wVDOPTimes10;         /* VDOP (0.1 units) */
unsigned short m_wWAASMask;            /* Bits 0-1: tracked sats,
                                        Bits 2-3:used sats, Bits
                                        5-9 WAAS PRN 1 minus 120,
                                        Bits 10-14 WAAS PRN 1 minus
                                        120 */
unsigned short m_wCheckSum;            /* sum of all bytes of the data*/
unsigned short m_wCRLF;                /* Carriage Return Line Feed */
} SBinaryMsg2; /* length = 8 + 16 + 2 + 2 = 28 */
```

SChannelData

```
typedef struct
{
unsigned char m_byChannel;         /* channel number */
unsigned char m_bySV;              /* satellite being tracked, 0 ==
                                   not tracked */
unsigned char m_byStatus;          /* Status bits (code carrier bit
                                   frame...) */
unsigned char m_byLastSubFrame; /* last subframe processed */
unsigned char m_byEphmVFlag;       /* ephemeris valid flag */
unsigned char m_byEphmHealth;      /* ephemeris health */
unsigned char m_byAlmVFlag;        /* almanac valid flag */
unsigned char m_byAlmHealth;       /* almanac health */
char m_chElev;                     /* elevation angle */
unsigned char m_byAzimuth;         /* 1/2 the Azimuth angle */
unsigned char m_byURA;             /* User Range Error */
unsigned char m_byDum;             /* Place Holder */
unsigned short m_wCliForSNR;       /* code lock indicator for SNR
                                   divided by 32 */
short        m_nDiffCorr;          /* Differential correction *
                                   100 */
short        m_nPosResid;          /* position residual * 10 */
short        m_nVelResid;          /* velocity residual * 10 */
short        m_nDoppHz;            /* expected doppler in HZ */
short        m_nNCOHz;             /* track from NCO in HZ */
} SChannelData;                    /* 24 bytes */
```

SChannelL2Data

```
//#if defined(_DUAL_FREQ_)
typedef struct
{
unsigned char m_byChannel;        /* channel number */
unsigned char m_bySV;             /* satellite being tracked, 0 == not
                                     tracked */
unsigned char m_byStatus;         /* Status bits (code carrier bit frame...)
                                     */
unsigned char m_byLastSubFrame; /* last subframe processed */
unsigned char m_byEphmVFlag;      /* ephemeris valid flag */
unsigned char m_byEphmHealth;     /* ephemeris health */
unsigned char m_byAlmVFlag;       /* almanac valid flag */
unsigned char m_byAlmHealth;      /* almanac health */
char m_chElev;                    /* elevation angle */
unsigned char m_byAzimuth;        /* 1/2 the Azimuth angle */
// unsigned char m_byURA;         /* User Range Error */
// unsigned char m_byDum;         /* Place Holder */
short m_nL1CACodeNoise;
unsigned short m_wCliForSNR;      /* code lock indicator for SNR divided by
                                     32*/
short m_nL1CAL2Piono;             /* L1CA - L2P phase difference in meters
                                     * 100.*/
short m_nL2CodeRange;             /* L2 Code Range * 10 */
short m_nWAASiono;                /* WAAS Iono Estimate in meters * 10 */
short m_nDoppHz;                  /* expected doppler in HZ */
short m_nNCOHz;                   /* track from NCO in HZ */
} SChannelL2Data; /* 24 bytes */
//#endif
```

SBinaryMsg99

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned char m_byNavMode;                    /* Nav Mode FIX_NO, FIX_2D,
                                          FIX_3D (high bit=has_diff) */
char m_cUTCTimeDiff;                        /* whole Seconds between
                                         UTC and GPS */
unsigned short m_wGPSWeek;                 /* GPS week */
double m_dGPSTimeOfWeek;                   /* GPS tow */
SChannelData m_asChannelData[CHANNELS_12]; /* channel data */
                                           short m_nClockErrAtL1; /
                                           * clock error at L1, Hz */
unsigned short m_wSpare;                   /* spare */
unsigned short m_wCheckSum;                /* sum of all bytes of the data*/
unsigned short m_wCRLF;                    /* Carriage Return Line Feed */
} SBinaryMsg99;                            /* length = 8 + 304 + 2 + 2 =
                              316 */
```

SBinaryMsg100

```
//#if defined(_DUAL_FREQ_)
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned char m_byNavMode;   /* Nav Mode FIX_NO, FIX_2D, FIX_3D
                             (high bit=has_diff) */
char m_cUTCTimeDiff;         /* whole Seconds between UTC and GPS */
unsigned short m_wGPSWeek;  /* GPS week */
double m_dGPSTimeOfWeek;    /* GPS tow */
SChannelL2Data m_asChannelData[CHANNELS_12]; /* channel data */
short m_nClockErrAtL1;      /* clock error at L1, Hz */
unsigned short m_wSpare;    /* spare */
unsigned short m_wCheckSum; /* sum of all bytes of the data */
unsigned short m_wCRLF;     /* Carriage Return Line Feed */
} SBinaryMsg100;            /* length = 8 + 304 + 2 + 2 = 316 */
//#endif
```

SSVAlmanData

```
typedef struct
{
short m_nDoppHz;                    /* doppler in HZ for stationary receiver */
unsigned char m_byCountUpdate; /* count of almanac updates */
unsigned char m_bySVindex;       /* 0 through 31 (groups of 8)*/
unsigned char m_byAlmVFlag;      /* almanac valid flag */
unsigned char m_byAlmHealth;     /* almanac health */
char m_chElev;                          /* elevation angle */
unsigned char m_byAzimuth;        /* 1/2 the Azimuth angle */
} SSVAlmanData;                   /* 8 bytes */
```

SBinaryMsg98

```
typedef struct
{
SUnionMsgHeader m_sHead;
SSVAlmanData m_asAlmanData[8]; /* SV data, 8 at a time */
unsigned char m_byLastAlman;       /* last almanac processed */
unsigned char m_byIonoUTCVFlag;  /* iono UTC flag */
unsigned short m_wSpare;            /* spare */
unsigned short m_wCheckSum;      /* sum of all bytes of the data */
unsigned short m_wCRLF;             /* Carriage Return Line Feed */
} SBinaryMsg98;                         /* length = 8 + (64+1+1+2) + 2 + 2 =
                                        80 */
```

SBinaryMsg97

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned long m_ulCPUFactor;           /* CPU utilization Factor
                                       (%=multby 450e-6)*/
unsigned short m_wMissedSubFrame;    /* missed subframes */
unsigned short m_wMaxSubFramePend; /* max subframe pending */
unsigned short m_wMissedAccum;       /* missed accumulations */
unsigned short m_wMissedMeas;        /* missed measurements */
unsigned long m_ulSpare1;            /* spare 1 (zero)*/
unsigned long m_ulSpare2;            /* spare 2 (zero)*/
unsigned long m_ulSpare3;            /* spare 3 (zero)*/
unsigned short m_wSpare4;            /* spare 4 (zero)*/
unsigned short m_wSpare5;            /* spare 5 (zero)*/
unsigned short m_wCheckSum;          /* sum of all bytes of the data */
unsigned short m_wCRLF;              /* Carriage Return Line Feed */
} SBinaryMsg97;
```

SObservations

```
typedef struct
{
unsigned long m_ulCS_TT_SNR_PRN; /* Bits 0-7 PRN (PRN is 0 if no data)
                                    */
                                   /* Bits 8-15 SNR_value
                                    SNR = 10.0*log10(
                                    0.8192*SNR_value) */
                                      /* Bits 16-23 Phase Track Time in units
                                   of 1/10 second (range = 0 to 25.5
                                   seconds (see next word) */
                                  /* Bits 24-31 Cycle Slip Counter
                                   Increments by 1 every cycle slip
                                   with natural roll over after 255 */
unsigned long m_ulDoppler_FL;
                                 /* Bit 0: 1 if Valid Phase, 0 otherwise
                                  Bit 1: 1 if Track Time > 25.5 sec,
                                  0 otherwise
                                  Bits 2-3: unused
                                  Bits 4-32: Signed (two's compliment)
                                  doppler in units of m/sec x 4096.
                                  (i.e., LSB = 1/4096). Range =
                                  +/- 32768 m/sec. Computed as
                                  phase change over 1/10 sec. */
double m_dPseudoRange;             /* pseudo ranges (m) */
double m_dPhase;                  /* phase (m) L1 wave len =
                                  0.190293672798365 */
} SObservations; /* 24 bytes */
```

SBinaryMsg96

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned short m_wSpare1;     /* spare 1 (zero)*/
unsigned short m_wWeek;       /* GPS Week Number */
double m_dTow;                /* Predicted GPS Time in seconds */
SObservations m_asObvs[CHANNELS_12];/* 12 sets of observations */unsigned
short m_wCheckSum; /* sum of all bytes of the data */
unsigned short m_wCRLF;       /* Carriage Return Line Feed */
} SBinaryMsg96;
```

SBinaryMsg95

```
/* sent only upon command or when values change */
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned short m_wSV;                /* The satellite to which this data
                                     belongs. */
unsigned short m_wSpare1;            /* spare 1 (chan number (as zero
                                     9/1/2004)*/
unsigned long m_TOW6SecOfWeek;  /* time at which this arrived (LSB =
                                     6sec)*/
unsigned long m_SF1words[10];     /* Unparsed SF 1 message words. */
unsigned long m_SF2words[10];     /* Unparsed SF 2 message words. */
unsigned long m_SF3words[10];     /* Unparsed SF 3 message words. */
                                     /* Each of the subframe words contains
                                     one 30-bit GPS word in the lower
                                     30 bits, The upper two bits are ignored
                                     Bits are placed in the words from left
                                     to right as they are received */
unsigned short m_wCheckSum;       /* sum of all bytes of the data */
unsigned short m_wCRLF;            /* Carriage Return Line Feed */
} SBinaryMsg95;                     /* length = 8 + (128) + 2 + 2 = 140 */
```

SBinaryMsg94

```
/* sent only upon command or when values change */
typedef struct
{
SUnionMsgHeader m_sHead;
/* Iono parameters. */
double m_a0,m_a1,m_a2,m_a3; /* AFCRL alpha parameters. */
double m_b0,m_b1,m_b2,m_b3; /* AFCRL beta parameters. */
/* UTC conversion parameters. */
double m_A0,m_A1;              /* Coeffs for determining UTC time. */
unsigned long m_tot;           /* Reference time for A0 & A1, sec of GPS
                               week. */
unsigned short m_wnt;          /* Current UTC reference week number. */
unsigned short m_wnlsf;        /* Week number when dtlsf becomes
                               effective. */
unsigned short m_dn;           /* Day of week (1-7) when dtlsf becomes
                               effective. */
short m_dtls;                  /* Cumulative past leap seconds. */
short m_dtlsf;                 /* Scheduled future leap seconds. */
unsigned short m_wSpare1;      /* spare 4 (zero)*/
unsigned short m_wCheckSum; /* sum of all bytes of the data */
unsigned short m_wCRLF;        /* Carriage Return Line Feed */
} SBinaryMsg94; /* length = 8 + (96) + 2 + 2 = 108 */
```

SBinaryMsg93

```
/* sent only upon command or when values change */
/* WAAS ephemeris */
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned short m_wSV;              /* The satellite to which this data
                                      belongs.*/
unsigned short m_wWeek;            /* Week corresponding to m_lTOW*/
unsigned long m_lSecOfWeekArrived; /* time at which this arrived (LSB =
                                      1sec)*/
unsigned short m_wIODE;
unsigned short m_wURA;             /* See 2.5.3 of Global Pos Sys Std Pos
                                      Service Spec */
long m_lTOW;                       /* Sec of WEEK Bit 0 = 1 sec */
long m_lXG;                        /* Bit 0 = 0.08 m */
long m_lYG;                        /* Bit 0 = 0.08 m */
long m_lZG;                        /* Bit 0 = 0.4 m */
long m_lXGDot;                     /* Bit 0 = 0.000625 m/sec */
long m_lYGDot;                     /* Bit 0 = 0.000625 m/sec */
long m_lZGDot;                     /* Bit 0 = 0.004 m/sec */
long m_lXGDotDot;                  /* Bit 0 = 0.0000125 m/sec/sec */
long m_lYGDotDot;                  /* Bit 0 = 0.0000125 m/sec/sec */
long m_lZGDotDot;                  /* Bit 0 = 0.0000625 m/sec/sec */
short m_nGf0;                      /* Bit 0 = 2**-31 sec */
short m_nGf0Dot;                   /* Bit 0 = 2**-40 sec/sec */
unsigned short m_wCheckSum;        /* sum of all bytes of the data*/
unsigned short m_wCRLF;            /* Carriage Return Line Feed */
} SBinaryMsg93; /* length = 8 + (56) + 2 + 2 = 68 */
```

SBinaryMsg80

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned short m_wPRN;                  /* Broadcast PRN */
unsigned short m_wSpare;                /* spare (zero) */
unsigned long m_ulMsgSecOfWeek; /* Seconds of Week For Message */unsigned
long m_aulWaasMsg[8];      /* Actual 250 bit waas message*/
unsigned short m_wCheckSum;          /* sum of all bytes of the data */
unsigned short m_wCRLF;               /* Carriage Return Line Feed */
} SBinaryMsg80;                             /* length = 8 + (40) + 2 + 2 = 52 */
```

SMsg91data

```
typedef struct
{
unsigned char bySV; /* satellite being tracked, 0 == not tracked
*/
unsigned char byStatus;          /* Status bits (code carrier bit frame...) */
unsigned char byStatusSlave;   /* Status bits (code carrier bit frame...) */
unsigned char byChannel;        /* Not used */
unsigned short wEpochSlew;    /* 20*_20MS_EPOCH_SLEW +
                                   _1MS_EPOCH_SLEW */
unsigned short wEpochCount;   /* epoch_count */
unsigned long codeph_SNR;     /* 0-20 = code phase (21 bits), 28-32
                                   = SNR/4096, upper 4 bits */
unsigned long ulCarrierCycles_SNR; /* 0-23 = carrier cycles, 24-32 =
                                   SNR/4096 lower 8 bits */
unsigned short wDCOPhaseB10_HalfWarns; /* 0-11 = DCO phase, 12-14 =
                                          Half Cycle Warn 15 = half
                                          Cycle added */
unsigned short m_wPotentialSlipCount;      /* potential slip count */
/* SLAVE DATA */
unsigned long codeph_SNR_Slave;             /* 0-20 = code phase (21 bits),
                                          28-32 = SNR/4096, upper
                                          4 bits */
unsigned long ulCarrierCycles_SNR_Slave;  /* 0-23 = carrier cycles, 24-32
                                          = SNR/4096 lower 8 bits */
unsigned short wDCOPhaseB10_HalfWarns_Slave; /* 0-11 = DCO phase,
                                                12-14 = Half Cycle Warn
                                             15 = half Cycle added */
unsigned short m_wPotentialSlipCount_Slave; /* potential slip count */
} SMsg91Data; /* 32 bytes */
```

SBinaryMsg91

```
typedef struct
{
SUnionMsgHeader m_sHead;             /* 8 */
double m_sec;                        /* 8 bytes */
int m_iWeek;                         /* 4 bytes */
unsigned long m_Tic;                 /* 4 bytes */
long lTicOfWeek;                     /* 4 bytes */
long lProgTic;                       /* 4 bytes */
SMsg91Data s91Data[CHANNELS_12]; /* 12*32= 384 bytes */
unsigned short m_wCheckSum;          /* sum of all bytes of the data */
unsigned short m_wCRLF;              /* Carriage Return Line Feed */
} SBinaryMsg91;                      /* length = 8 + (408) + 2 + 2 = 420
                                     */
```

SOBsPacket

```
typedef struct
{
unsigned long m_ulCS_TT_W3_SNR; /* Bits 0-11 (12 bits) =SNR_value
                                   For L1 SNR =10.0*
                                   log10(0.1024*SNR_value)
                                   FOR L2 SNR = 10.0*log10(
                                   0.1164*SNR_value) */
                                 /* Bits 12-14 (3 bits) = 3 bits of
                                   warning for potential 1/2 cycle slips.
                                   A warning exists if any of these bits
                                   are set. */
                                /* bit 15: (1 bit) 1 if Track Time > 25.5
                                  sec,0 otherwise */
                                /* Bits 16-23 (8 bits): Track Time in
                                  units of 1/10 second (range = 0 to
                                  25.5 seconds) */
                               /* Bits 24-31 (8 bits) = Cycle Slip
                                 Counter Increments by 1 every cycle
                                 slip with natural roll-over after 255 */
                                     unsigned long m_ulP7_Doppler_FL; /*
                              Bit 0: (1 bit) 1 if Valid Phase, 0 otherwise
                              Bit 1-23: (23 bits) =Magnitude of doppler
                              LSB = 1/512 cycle/sec
                              Range = 0 to 16384 cycle/sec
                                 Bit 24: sign of doppler, 1=negative, 0=pos
                              Bits 25-31 (7 bits) = upper 7 bits of the
                                23 bit carrier phase.
                              LSB = 64 cycles, MSB = 4096 cycles
                               */
unsigned long m_ulCodeAndPhase; /* Bit 0-15 (16 bits) lower 16 bits of code
                              pseudorange
                              LSB = 1/256 meters
                              MSB = 128 meters
                              Note, the upper 19 bits are given in
                              m_aulCACodeMSBsPRN[] for CA code
                                 Bit 16-31 lower 16 bits of the carrier phase,
                              7 more bits are in
                              m_ulP7_Doppler_FL
```

LSB = 1/1024 cycles
MSB = 32 cycles */

SBinaryMsg76

Note: "Code: means the PseudoRange derived from code phase. "Phase" means range derived from carrier phase. This will contain the cycle ambiguities.

Only the lower 16 bits of L1P code, L2P code and the lower 23 bits of carrier phase are provided. The upper 19 bits of the L1CA code are found in m_aulCACodeMSBsPRN[]. The upper 19 bits of L1P or L2P must be derived using the fact that L1P and L2P are within 128 meters of L1CA. To determine L1P or L2P, use the lower 16 bits provided in the message and set the upper bits to that of L1CA. Then add or subtract one LSB of the upper bits (256 meters) so that L1P or L2P are within 1/2 LSB (128 meters) of the L1CA code.
The carrier phase is in units of cycles, rather than meters, and is held to within 1023 cycles of the respective code range. Only the lower 16+7=23 bits of carrier phase are transmitted in Msg 76.

To determine the remaining bits:

1. Convert the respective code range (determined above) into cycles by dividing by the carrier wavelength. Call this the "nominal reference phase."
2. Extract the 16 and 7 bit blocks of carrier phase from Msg 76 and arrange to form the lower 23 bits of carrier phase.
3. Set the upper bits (bit 23 and above) equal to those of the nominal reference phase.
4. Add or subtract the least significant upper bit (8192 cycles) so that carrier phase most closely agrees with the nominal reference phase (to within 4096 cycles). This is similar to what is done with L1P and L2P.

```
ypedef struct
{
SUnionMsgHeader m_sHead;
double m_dTow;              /* GPS Time in seconds */
unsigned short m_wWeek;   /* GPS Week Number */
unsigned short m_wSpare1; /* spare 1 (zero)*/
unsigned long m_ulSpare2;  /* spare 2 (zero)*/
unsigned long m_aulCACodeMSBsPRN[CHANNELS_12]; /* array of 12
                              words. bit 7:0 (8 bits) = satellite
                               PRN, 0 if no satellite bit 12:8 (5 bits) = spare
                              bit 31:13 (19 bits) = upper 19 bits
                              of L1CA LSB = 256 meters
                              MSB = 67108864 meters*/
SObsPacket m_asL1CAObs[CHANNELS_12]; /* 12 sets of L1(CA)
                                          observations */
SObsPacket m_asL2PObs[CHANNELS_12];   /* 12 sets of L2(P)
                                          observations */
unsigned long m_auL1Pword[CHANNELS_12]; /* array of 12 words relating
                                           to L1(P) code. Bit 0-15 (16
                                           bits) lower 16 bits of the
                                           L1P code pseudo range.
                                           LSB = 1/256 meters
                                           MSB = 128 meters
                                           Bits 16-27 (12 bits) = L1P
                                           SNR_value
                                           SNR = 10.0*log10(
                                           0.1164*SNR_value)
                                           Bits 28-31 (4 bits) spare */
unsigned short m_wCheckSum; /* sum of all bytes of the data */
unsigned short m_wCRLF; /* Carriage Return Line Feed */
} SBinaryMsg76; /* length = 8 + (400) + 2 + 2 = 412*/
#else /* OLD_BIN76 */
```

SObsPacket

```
typedef struct
{
unsigned long m_ulCS_TT_R3_SNR; /* Bits 0-11 (12 bits) =SNR_value
                                   For L1 SNR = 10.0*log10(
                                   0.1024*SNR_value)FOR L2 SNR =
                                   10.0*log10(0.1164*SNR_value) */
                                  /* Bits 12-14 (3 bits) = upper 3 bits of
                                   the pseudoRange, m_ulPseudoRange
                                   LSB = 16777216 meters */
                                 /* bit 15: 1 if Phase Track Time > 25.5
                                  sec,0 otherwise */
                                   /* Bits 16-23 (8 bits): Phase Track Time
                                  in units of 1/10 second (range = 0 to
                                    25.5 seconds) */
                                /* Bits 24-31 (8 bits) = Cycle Slip
                                 CounterIncrements by 1 every cycle
                                 slip with natural roll-over after
                                          255 */
                                 unsigned long m_ulS_P6_Doppler_FL;
                                /* Bit 0: (1 bit) 1 if Valid Phase, 0
                                otherwise Bit 1-23: (23 bits) =
                                Magnitude of doppler
                                LSB = 1/512 cycle/sec
                                Range = 0 to 16384 cycle/sec
                                Bit 24: sign of doppler, 1=negative,
                                0=pos Bits 25-30 (6 bits) = upper 6
                                bits of the carrier phase magnitude,
                                m_ulPhase.
                                LSB = 4194304 cycles, MSB =
                                134217728 cycles
                                Bit 31: sign of m_ulPhase,
                                1=negative, 0=pos */
unsigned long m_ulPseudoRange; /* Bit 0-31 (32 bits) lower 32 bits of the
                                pseudo range
                                LSB = 1/256 meters
                                MSB = 8388608 meters*/
unsigned long m_ulPhase;        /* Bit 0-31 (32 bits) lower 32 bits of the
                                magnitude of the carrier phase
```

```
                        LSB = 1/1024 cycles
                      MSB = 2097152 cycles */

} SObsPacket; /* 16 bytes , note: all zero if data not available */
```

SBinaryMsg76

```
typedef struct
{
SUnionMsgHeader m_sHead;
unsigned short m_wSpare1;            /* spare 1 (zero)*/
unsigned short m_wWeek;              /* GPS Week Number */
double m_dTow;                       /* GPS Time in seconds */
unsigned char m_aPRN[CHANNELS_12]; /* 12 PRNS to match data below, 0
                                     if no PRN */
SObsPacket m_asL1CAObs[CHANNELS_12]; /* 12 sets of L1(CA)
                                        observations */
SObsPacket m_asL2PObs[CHANNELS_12]; /* 12 sets of L2(P)
                                       observations */
unsigned short m_wCheckSum;          /* sum of all bytes of the data */
unsigned short m_wCRLF;              /* Carriage Return Line Feed */
} SBinaryMsg76;                      /* length = 8 + (408) + 2 + 2
                                     = 420*/
#endif /* OLD_BIN76 */
```

SMsg71Data

```
typedef struct
{
unsigned char bySV;              /* satellite being tracked, 0 == not tracked*/
unsigned char byStatus;          /* Status bits (code carrier bit frame...) */
unsigned char byStatusL1P;       /* Status bits (code carrier phase ...) */
unsigned char byStatusL2P;       /* Status bits (code carrier phase ...) */
unsigned short wEpochSlew;   /* 20*_20MS_EPOCH_SLEW +
                                   _1MS_EPOCH_SLEW*/
unsigned short wEpochCount; /* epoch_count */
unsigned long codeph_SNR;   /* 0-20 = code phase (21 bits), 28-32
                                 = SNR/4096, upper 4 bits */
unsigned long ulCarrierCycles_SNR; /* 0-23 = carrier cycles, 24-32 =
                                        SNR/4096 lower 8 bits */
unsigned short wDCOPhaseB10_HalfWarns; /* 0-11 = DCO phase, 12-14 =
                                              Half Cycle Warn
                                              15 = half Cycle added */
unsigned short m_wPotentialSlipCount; /* potential slip count */
/* L1P and L2P Data */=
// unsigned long codeph_SNR_L1P; NOT USED YET /* 0-22 = L1 code phase
                                  (23 bits),28-32 = SNR/8192, upper 4 bits */
unsigned long codeph_SNR_L2P; /* 0-22 = L2 code phase (23 bits),
                                     28-32 = SNR/8192, upper 4 bits */
unsigned long ulCarrierCycles_SNR_L2P; /* 0-23 = carrier cycles, 24-32 =
                                            SNR/8192 lower 8 bits */
unsigned short wDCOPhaseB10_L2P; /* 0-11 = DCO phase, 12-15 =
                                         Spare */
unsigned short m_wSNR_L1P; /* 0-11= L1P SNR/ 256 Lower 12
                                   bits, 12-15 = Spare */
} SMsg71Data; /* 32 bytes */
```

SBinaryMsg71

```
typedef struct
{
SUnionMsgHeader m_sHead;             /* 8 */
double m_sec;                        /* 8 bytes */
int m_iWeek;                         /* 4 bytes */
unsigned long m_Tic;                 /* 4 bytes */
long lTicOfWeek;                     /* 4 bytes */
long lProgTic;                       /* 4 bytes */
SMsg71Data s91Data[CHANNELS_12]; /* 12*32= 384 bytes */
unsigned short m_wCheckSum;          /* sum of all bytes of the data */
unsigned short m_wCRLF; /* Carriage Return Line Feed */
} SBinaryMsg71;                      /* length = 8 + (408) + 2 + 2 =
                                     420 */
#if defined(WIN32) || (__ARMCC_VERSION >= 300441)
#pragma pack(pop)
#endif
#endif // __BinaryMsg_H_
```

7: NMEA 2000

NMEA 2000

The receiver supports three NMEA 2000 CAN messages.

- GNSSSPositionData
- GNSSPositionRapidUpdates
- NMEACogData

GNSSSPositionData

The GNSSPositionData command (PGN 0x1F805/129029) has an update rate of 1 Hz and DLC of 43, 47 or 51, dependent on the NumberOfReferenceStations. Table 7-1 provides the startbit, length (Bit), value type, factor and offset for the fields of the GNSSPosition.

Table 7-1: GNSSPosition Data

Field Name	Start bit	Length (Bit)	Byte Order	Value type	Factor
SequenceID	0	8	Intel	Unsigned	1
PositionDate	8	16	Intel	Unsigned	1
PositionTime	24	32	Intel	Unsigned	0.0001
LatitudeLow	56	32	Intel	Unsigned	1.00E-16
LatitudeHigh	88	32	Intel	Signed	4.29E-07
LongitudeLow	120	32	Intel	Unsigned	1.00E-16
LongitudeHigh	152	32	Intel	Signed	4.29E-07
AltitudeLow	184	32	Intel	Unsigned	1.00E-6
AltitudeHigh	216	32	Intel	Signed	4294.97
TypeOfSystem	248	4	Intel	Unsigned	1

Table 7-1: GNSSPosition Data

Field Name	Start bit	Length (Bit)	Byte Order	Value type	Factor
GNSSMethod	252	4	Intel	Unsigned	1
GNSSIntegrity	256	2	Intel	Unsigned	1
GNSS_Reserved1	258	6	Intel	Unsigned	1
NumberOfSVs	264	8	Intel	Unsigned	1
HDOP	272	16	Intel	Signed	0.01
PDOP	288	16	Intel	Signed	0.01
GeodalSeparation	304	32	Intel	Signed	0.01
NumberOfReferenceStations	336	8	Intel	Unsigned	1
ReferenceStationType1	344	4	Intel	Unsigned	1
ReferenceStationID1	348	12	Intel	Unsigned	1
AgeOfDGNSSCorrections1	360	16	Intel	Unsigned	0.01
ReferenceStationType2	376	4	Intel	Unsigned	1
ReferenceStationID2	380	12	Intel	Unsigned	1
AgeOfDGNSSCorrections2	392	16	Intel	Unsigned	0.01

Table 7-2 provides the GNSSPosition data's offset, minimum, maximum, unit and comment for the GNSSPosition fields.

Table 7-2: GNSSPosition Data

Field Name	Offset	Minimum	Maximum	Unit	Comment
SequenceID	0	0	255		An upward counting number used to tie related information together between different PGNS
PositionDate	0	0	65532	day	Days since January 1, 1970. Date is relative to UTC time.
PositionTime	0	0	86401	sec	24 hour clock, 0=midnight, time is in UTC
LatitudeLow	0	0	4.29E-07	deg	Latitude referenced to WGS-84
LatitudeHigh	0	-90	90	deg	Latitude referenced to WGS-84
LongitudeLow	0	0	4.29E-07	deg	Longitude referenced toWGS-84
LongitudeHigh	0	-180	180	deg	Longitude referenced toWGS-84

Table 7-2: GNSSPosition Data

Field Name	Offset	Minimum	Maximum	Unit	Comment
AltitudeLow	0	0	4294	m	Altitude referenced to WGS-84
AltitudeHigh	0	-9.22 E+12	9.22E+12	m	Altitude referenced to WGS-84
TypeOfSystem	0	0	4		0x0 GPS 0x1 GLONASS 0x2 GPS and GLONASS 0x3 GPS and SBAS (WAAS/ EGNOS) 0x4 GPS and SBAS and GLONASS
GNSSMethod	0	0	15		0x0 No GPS 0x1 GNSS fix 0x2 DGNSS fix 0x3 Precise GNSS 0x4 RTK fixed integer 0x5 RTK float 0x6 Estimated (DR) mode 0x7 Manual input 0x8 Simulate mode 0xE Error

Table 7-2: GNSSPosition Data

Field Name	Offset	Minimum	Maximum	Unit	Comment
GNSSIntegrity	0	0	3		0x0 No integrity checking 0x1 Safe 0x2 Caution 0X3 Unsafe
GNSS_Reserved1	0	0	63		
NumberOfSVs	0	0	252		Numeric count, event counter
HDOP	0	-327.64	327.64		Dilution of Precision (DOP) indicates the contribution of satellite configuration geometry to positioning error
PDOP	0	-327.64	327.64		Dilution of Precision (DOP) indicates the contribution of satellite configuration geometry to positioning error

Table 7-2: GNSSPosition Data

Field Name	Offset	Minimum	Maximum	Unit	Comment
GeodalSeparation	0	-2.15 E+07	2.15E+07	m	The difference between the earth ellipsoid and mean-sea-level (period), defined by the reference datum used in the position solution. “-”=mean-sea-level below ellipsoid
NumberOf Reference Stations	0	0	252		Number of reference stations reported
ReferenceStation Type1	0	0	15		0x0 GPS 0x1 GLONASS 0xE Error
ReferenceStation ID1	0	0	4095		Reference station ID
AgeOfDGNSS Corrections1	0	0	655.32	sec	Age of differential corrections
ReferenceStation Type2	0	0	15		0x0 GPS 0x1 GLONASS 0xE Error
ReferenceStation ID2	0	0	4095		Reference station ID
AgeOfDGNSS Corrections2	0	0	655.32	sec	Age of differential corrections

GNSSPositionRapidUpdate

The GNSSPositionRapidUpdate command (PGN 0x1F801/129025) has an update rate equal to the subscribed rate (default of 10 Hz) and DLC of 8. Table 7-3 provides the start bit, length (Bit), value type, factor and offset for fields of the GNSSPositionRapidUpdate.

Table 7-3: GNSSPositionRapid Update

Field Name	Startbit	Length (Bit)	Byte order	Value Type	Factor	Offset
Latitude	0	32	Intel	Signed	0.0000001	0
Longitude	32	32	Intel	Signed	0.0000001	0

Table 7-4 provides the GNSSPositionRapidUpdate's offset, minimum, maximum and unit for the GNSSPositionRapidUpdate fields.

Table 7-4: GNSSPositionRapid Update

Field Name	Minimum	Maximum	Unit
Latitude	-90	90	deg
Longitude	-180	180	deg

NMEACogSogData

The NMEACogSogDaa command (PGN 0x1F802/129026) has an update rate equal to the subscribed rate (default of 10 Hz) and DLC of 8. Table 7-5 provides the startbit, length (Bit) value type, factor, and offset for the fields of the NMEACogSogData.

Table 7-5: NMEACogSogData

Field Name	Startbit	Length (Bit)	Byte order	Value Type	Factor
NMEA_SequenceID	0	8	Intel	Unsigned	1
NMEA_DirectionReferene	8	2	Intel	Unsigned	1
NMEA_Reserved1	10	6	Intel	Unsigned	1
NMEA_CourseOverGround	16	16	Intel	Unsigned	0.0001
NMEA_SpeedOverGround	32	16	Intel	Unsigned	0.01
NMEA_Reserved2	48	16	Intel	Unsigned	1

Table 7-6 provides the GNSSPositionRapidUpdate's offset, minimum, maximum, unit, and comments for the GNSSPositionRapidUpdate fields.

Table 7-6: NMEACogSogData

Field Name	**Minimum**	**Maximum**	**Unit**	**Comment**
NMEA_SequenceID	0	255		An upward counting number used to tie related information together between different PGNs
NMEA_DirectionReference	0	3		0x0 True north 0x1 Magnetic north 0x2 Error 0X3 Null
NMEA_Reserved1	0	63		
NMEA_CourseOverGround	0	6.5535	rad	GPS based travel direction
NMEA_SpeedOverGround	0	655.35	m/s	GPS based travel speed
NMEA_Reserved2	0	65535		

Appendix A: Troubleshooting

Use the checklist in Table A-1 to troubleshoot the anomalous receiver system operation. Table A-1 provides a common problem, followed by a list of possible solutions.

Table A-1: Troubleshooting

Problem	Possible Solutions
Receiver fails to power	• Verify polarity of power leads • Check 1.0 A in-line power cable fuse • Check integrity of power cable connections • Check power input voltage • Check current restrictions imposed by power source (minimum available should be > 1.0 A)
No data from receiver	• Check receiver power status • Verify that receiver is locked to a valid DGPS signal (this can often be done on the receiving device with the use of the PocketMAX PC) • Verify that receiver is locked to GPS satellites (this can often be done on the receiving device with the use of the PocketMAX PC) • Check integrity and connectivity of power and data cable connections

Table A-1: Troubleshooting

Problem	Possible Solutions
Random data from receiver	• Verify that the RTCM or the Bin95 and Bin96 messages are not being output accidentally (send a $JSHOW command) • Verify baud rate settings of receiver and remote device match correctly • Potentially, the volume of data requested to be output by the receiver could be higher than the current rate supports. Try using 19,200 or 38,400 as the baud rate for all devices
No GPS lock	• Check integrity of antenna cable • Verify antenna's unobstructed view of the sky • Verify the lock status of the GPS satellites (this can often be done on the receiving device with the use of the PocketMAX PC)
No SBAS lock	• Check antenna connections • Verify antenna's unobstructed view of the sky • Verify the lock status of SBAS satellites (this can often be done on the receiving device with the use of the PocketMAX PC - monitor BER value)

Table A-1: Troubleshooting

Problem	Possible Solutions
No DGPS position in external RTCM mode	• Verify that the baud rate of the RTCM input port matches the baud rate of the external source • Verify the pin-out between the RTCM source and the RTCM input port (transmit from the source must go to receiver of the RTCM input port and grounds must be connected) • Verify the differential mode for the port which RTCM is being imported on is set to $JDIFF,THIS
Non-differential GPS output	• Verify receiver SBAS and lock status, or external source is locked
Multipath signals	• Operate away from large, reflective structures • Use special antennas and GPS equipment to help reduce impact

Appendix B: Resources

Reference Documents

National Marine Electronics Association, National Marine Electronics Association (NMEA) Standard for Interfacing Marine Electronic Devices, Version 2.1, October 15, NMEA 1995,
PO Box 50040, Mobile, Alabama, 36605 USA,
Tel: +1-205-0473-1793, Fax: +1-205-473-1669

Radio Technical Commission for Marinetime Services, RTCM Recommended Standards for Differential NAVSTAR GPS Service, Version 2.2, Developed by Special Committee
No. 104, RTCM 1998, 1800 Diagonal Rd, Suite 600,
Alexandria, VA, 22134-2840 USA Tel: +1-703-684-4481,
Fax: +1-703-836-4429

Radio Technical Commission for Aeronautics, Minimum Operational Performance Standards (MOPS) for Global Positioning System/Wide Area Augmentation System Airborne Equipment, Document RTCA D0-229A, Special Committee
No. 159, RTCA 1998, 1828 L Street, NW, Suite 805, Washington DC, 20036 USA, Tel: +1-202-833-9339

ARIC Research Corporation, Interface Control Document, Navstar GPS Space Segment/Navigation User Interfaces,
ICD-GPS-200, April 12, 2000, 2250 E. Imperial Highway, Suite 450, El Segundo, CA 90245-3509,
http://www.navcen.uscg.gov/gps/geninfo/default.htm

HemisphereGPS Website

This following address is the Hemisphere GPS web site, which provides detailed information on all products offered by Hemisphere GPS.

www.hemispheregps.com

FAA WAAS Website

This site offers general information on the WAAS service provided by the U.S. FAAS.

http://gps.faa.gov/Library/waas-f-text.htm

ESA EGNOS System Test Bed Website

This site contains information relating to past performance, real-time performance, and broadcast schedule of EGNOS.

http://www.esa.int/esaNA/egnos.html

Solar and Ionosphereic Activity Websites

The following sites are useful in providing details regarding solar and ionospheric activity.

http://iono.jpl.nasa.gov

http://www.spaceweather.com

OmniSTAR Service Activation

The OmniSTAR DGPS service for a DGPS MAX receiver may be activated by contacting the service provider in the user's region. Contact OmniSTAR with the unit number and they will activate the subscription over the air. Please be ready to have the receiver ready to receive the OmniSTAR signal for subscription validation.

For questions regarding the OmniSTAR service, please contact OmniSTAR for further information. Contact information is provided in Table 1 on page 270.

OmniSTAR License Agreement

OmniSTAR requires that the enclosed license agreement be filled out the before subscription activation. Please read the agreement thoroughly before filling in the require information. Be ready to fax the completed agreement when contacting OmniSTAR.

Contacting OmniSTAR

Table 1 provides the contact numbers for the various OmniSTAR offices throughout the world. Please contact the office responsible for subscriptions in the user's area by consulting Figure 1 on page 271.

Table B-1: OmniSTAR Contact Information

Location	**Telephone Number**	**Fax Number**
North America	+1-888-883-8476	+1-713-780-9408
Europe and North Africa	+1-31-70-311-1151	+31-71-581-4719
Asia, Australia, New Zealand, and South Africa	+61-89-322-5295	+61-8-9322-4164
Central America and South America	+1-713-785-5850	+1-713-780-9408

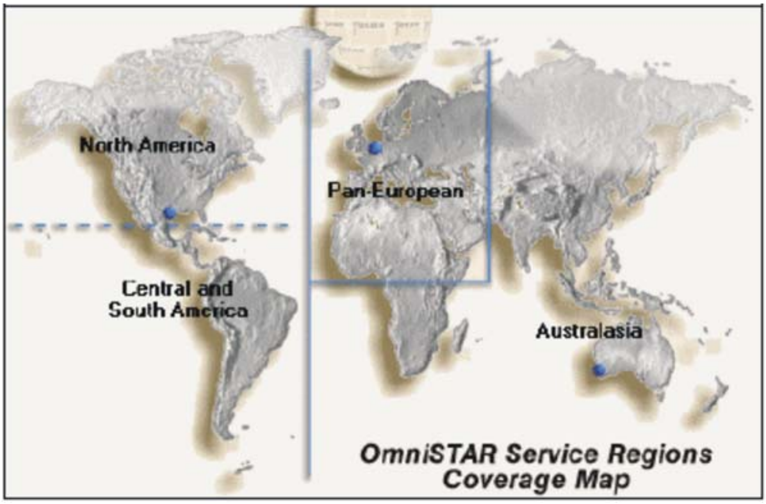


Figure B-1. OmniSTAR regions

Index

Hemisphere GPS®

www.hemispheregps.com
e-mail: info@hemispheregps.com