SPAN-SE

USER MANUAL

OM-20000124 Rev 2

SPAN-SE Technology for OEMV User Manual

Publication Number: OM-20000124
Revision Level: 2
Revision Date: 2009/09/01

This manual reflects SPAN-SE firmware Version SPPC 1.000 and OEMV firmware Version 3.621.

Proprietary Notice

Manufactured and protected under U.S. Patent:

Narrow Correlator
#5,101,416
#5,390,207
#5,495,499
#5,809,064
PAC Correlator
#6,243,409 B1
#5,414,729
Dual Frequency GPS
#5,736,961
Anti-Jamming Technology
#5,734,674

Position for Velocity Kalman Filter
#6,664,923 B1
#7,193,559 B2
SPAN Technology
#6,721,657 B2
#6,750,816 B1
10/758,363 (pending)
10/932,497 (pending)
#7,346,452

Table of Contents

Appendices

Figures

Tables

Software License

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE PRODUCT, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THESE TERMS OF USE, DO NOT INSTALL, COPY OR USE THIS ELECTRONIC PRODUCT (SOFTWARE, FIRMWARE, SCRIPT FILES, OR OTHER ELECTRONIC PRODUCT WHETHER EMBEDDED IN THE HARDWARE, ON A CD OR AVAILABLE ON THE COMPANY WEB SITE) (hereinafter referred to as "Software").

1. License: NovAtel Inc. ("NovAtel") grants you a non-exclusive, non-transferable license (not a sale) to, where the Software will be used on NovAtel supplied hardware or in conjunction with other NovAtel supplied software, use the Software with the product(s) as supplied by NovAtel. You agree not to use the Software for any purpose other than the due exercise of the rights and licences hereby agreed to be granted to you.

2. Copyright: NovAtel owns, or has the right to sublicense, all copyright, trade secret, patent and other proprietary rights in the Software and the Software is protected by national copyright laws, international treaty provisions and all other applicable national laws. You must treat the Software like any other copyrighted material except that you may make one copy of the Software solely for backup or archival purposes (one copy may be made for each piece of NovAtel hardware on which it is installed or where used in conjunction with other NovAtel supplied software), the media of said copy shall bear labels showing all trademark and copyright notices that appear on the original copy. You may not copy the product manual or written materials accompanying the Software. No right is conveyed by this Agreement for the use, directly, indirectly, by implication or otherwise by Licensee of the name of NovAtel, or of any trade names or nomenclature used by NovAtel, or any other words or combinations of words proprietary to NovAtel, in connection with this Agreement, without the prior written consent of NovAtel.

3. Patent Infringement: NovAtel shall not be liable to indemnify the Licensee against any loss sustained by it as the result of any claim made or action brought by any third party for infringement of any letters patent, registered design or like instrument of privilege by reason of the use or application of the Software by the Licensee or any other information supplied or to be supplied to the Licensee pursuant to the terms of this Agreement. NovAtel shall not be bound to take legal proceedings against any third party in respect of any infringement of letters patent, registered design or like instrument of privilege which may now or at any future time be owned by it. However, should NovAtel elect to take such legal proceedings, at NovAtel's request, Licensee shall co-operate reasonably with NovAtel in all legal actions concerning this license of the Software under this Agreement taken against any third party by NovAtel to protect its rights in the Software. NovAtel shall bear all reasonable costs and expenses incurred by Licensee in the course of co-operating with NovAtel in such legal action.

4. Restrictions: You may not:

(a) copy (other than as provided for in paragraph 2), distribute, transfer, rent, lease, lend, sell or sublicense all or any portion of the Software except in the case of sale of the hardware to a third party;

(b) modify or prepare derivative works of the Software;

(c) use the Software in connection with computer-based services business or publicly display visual output of the Software;

(d) transmit the Software over a network, by telephone or electronically using any means (except when downloading a purchased up[grade from the NovAtel web site); or

(e) reverse engineer, decompile or disassemble the Software.

You agree to keep confidential and use your best efforts to prevent and protect the contents of the Software from unauthorized disclosure or use.

5. <u>Term and Termination</u>: This Agreement and the rights and licences hereby granted shall continue in force in perpetuity unless terminated by NovAtel or Licensee in accordance herewith. In the event that the Licensee shall at any time during the term of this Agreement: i) be in breach of its obligations hereunder where such breach is irremediable or if capable of remedy is not remedied within 30 days of notice from NovAtel requiring its remedy; then and in any event NovAtel may forthwith by notice in writing terminate this Agreement together with the rights and licences hereby granted by NovAtel. Licensee may terminate this Agreement by providing written notice to NovAtel. Upon termination, for any reasons, the Licensee shall promptly, on NovAtel's request, return to NovAtel or at the election of NovAtel destroy all copies of any documents and extracts comprising or containing the Software. The Licensee shall also erase any copies of the Software residing on Licensee's computer equipment. Termination shall be without prejudice to the accrued rights of either party, including payments due to NovAtel. This provision shall survive termination of this Agreement howsoever arising.

6. <u>Warranty</u>: NovAtel does not warrant the contents of the Software or that it will be error free. The Software is furnished "AS IS" and without warranty as to the performance or results you may obtain by using the Software. The entire risk as to the results and performance of the Software is assumed by you. See product enclosure, if any for any additional warranty.

7. <u>Indemnification</u>: NovAtel shall be under no obligation or liability of any kind (in contract, tort or otherwise and whether directly or indirectly or by way of indemnity contribution or otherwise howsoever) to the Licensee and the Licensee will indemnify and hold NovAtel harmless against all or any loss, damage, actions, costs, claims, demands and other liabilities or any kind whatsoever (direct, consequential, special or otherwise) arising directly or indirectly out of or by reason of the use by the Licensee of the Software whether the same shall arise in consequence of any such infringement, deficiency, inaccuracy, error or other defect therein and whether or not involving negligence on the part of any person.

8. <u>Disclaimer and Limitation of Liability:</u>

(a) **THE WARRANTIES IN THIS AGREEMENT REPLACE ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NovAtel DISCLAIMS AND EXCLUDES ALL OTHER WARRANTIES. IN NO EVENT WILL NovAtel's LIABILITY OF ANY KIND INCLUDE ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, EVEN IF NovAtel HAS KNOWLEDGE OF THE POTENTIAL LOSS OR DAMAGE.**

(b) NovAtel will not be liable for any loss or damage caused by delay in furnishing the Software or any other performance under this Agreement.

(c) NovAtel's entire liability and your exclusive remedies for our liability of any kind (including liability for negligence) for the Software covered by this Agreement and all other performance or non-performance by NovAtel under or related to this Agreement are to the remedies specified by this Agreement.

9. <u>Governing Law</u>: This Agreement is governed by the laws of the Province of Alberta, Canada. Each of the parties hereto irrevocably attorns to the jurisdiction of the courts of the Province of Alberta.

10. <u>Customer Support</u>: For Software UPDATES and UPGRADES, and regular customer support, contact the NovAtel GNSS Hotline at 1-800-NOVATEL (U.S. or Canada only), or 403-295-4900, Fax 403-295-4901, e-mail to support@novatel.ca,
Web site: http://www.novatel.com or write to:
NovAtel Inc.
Customer Service Dept.
1120 - 68 Avenue NE,
Calgary, Alberta, Canada T2E 8S5

Standard Terms and Conditions of Sales

1. PRICES: All prices are Firm Fixed Price, FCA 1120 - 68th Avenue N.E., Calgary, Alberta. All prices include standard commercial packing for domestic shipment. All transportation, insurance, special packing costs and expenses, and all Federal, provincial and local excise, duties, sales, and other similar taxes are the responsibility of the Purchaser.

2. PAYMENT: Terms are prepayment unless otherwise agreed in writing. Interest shall be charged on overdue accounts at the rate of 18% per annum (1.5% per month) from due date. To expedite payment by wire transfer to NovAtel Inc.: Bank - HSBC Bank of Canada

Bank:	HSBC Bank of Canada	US Account #	788889-002
	407 - 8 Avenue S.W.	CDN Account #	788889-001
	Calgary, AB, Canada T2P 1E5	EURO Account #	788889-270
		Transit #	10029-016
		Swift	HKBCCATTCAL

3. DELIVERY: Purchaser shall supply shipping instructions with each order. (Ship to and bill to address, NovAtel Quotation #, Preferred carrier and account #, Custom broker/freight forwarder including name and contact #) In the absence of specific instructions, NovAtel may select a carrier and insure Products in transit and charge Purchaser accordingly. NovAtel shall not be responsible for any failure to perform due to unforeseen circumstances or causes beyond its ability to reasonably control. Risk of loss, damage or destruction shall pass to Purchaser upon delivery to carrier. Goods are provided solely for incorporation into the Purchaser’s end product and shall not be onward delivered except as incorporated in the Purchaser’s end product.

4. COPYRIGHT AND CONFIDENTIALITY: Copyright in any specification, drawing, computer software, technical description and other document supplied by NovAtel under or in connection with the Order and all intellectual property rights in the design of any part of the Equipment or provision of services, whether such design be registered or not, shall vest in NovAtel absolutely. The Buyer shall keep confidential any information expressed or confirmed by NovAtel in writing to be confidential and shall not disclose it without NovAtel's prior consent in writing to any third party or use it other than for the operation and maintenance of any Equipment provided.

5. GENERAL PROVISIONS: All Purchase Orders are subject to approval and acceptance by NovAtel. Any Purchase Order or other form from the Purchaser, which purports to expand, alter or amend these terms and conditions, is expressly rejected and is and shall not become a part of any agreement between NovAtel and the Purchaser. This agreement shall be interpreted under the laws of the Province of Alberta.

6. LIMITED WARRANTY AND LIABILITY: Warranty Period: Products - 1 year; Accessories - 90 days (in each case from the date of invoice). NovAtel warrants that during the Warranty Period that (a) the Product will be free from defects in material and workmanship and conform to NovAtel specifications; (b) the software will be free from error which materially affect performance; and (c) if applicable as defined in the User’s Manual, be eligible for access to post contract support and software updates when available. **THESE WARRANTIES ARE EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NOVATEL SHALL IN NO EVENT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE DUE TO ANY CAUSE.**

Purchaser’s exclusive remedy for a claim under this warranty shall be limited to the repair or replacement at NovAtel’s option and at NovAtel’s facility, of defective or nonconforming materials, parts or components or in the case of software, provision of a software revision for implementation by the Buyer. All material returned under warranty shall be returned to NovAtel prepaid by the Buyer and returned to the Buyer, prepaid by NovAtel. The foregoing warranties do not extend to (i) nonconformities, defects or errors in the Products due to accident, abuse, misuse or negligent use of the Products or use in other than a normal and customary manner, environmental conditions not conforming to NovAtel’s specifications, or failure to follow prescribed installation, operating and maintenance procedures, (ii) defects, errors or nonconformities in the Products due to modifications, alterations, additions or changes not made in accordance with NovAtel’s specifications or authorized by NovAtel, (iii) normal wear and tear, (iv) damage caused by force of nature or act of any third person, (v) shipping damage, (vi) service or repair of Product by the Purchaser without prior written consent from NovAtel, (vii) Products designated by NovAtel as beta site test samples, experimental, developmental, preproduction, sample, incomplete or out of specification Products, (viii) returned Products if the original identification marks have been removed or altered or (ix) Services or research activities.

7. EXCLUSION OF LIABILITY: If a Party would, but for this paragraph (7), have concurrent claims in contract and tort (including negligence) such claims in tort (including negligence) shall to the extent permitted by law be wholly barred, unenforceable and excluded.

NovAtel shall not be liable to the Buyer by way of indemnity or by reason of any breach of the Order or of statutory duty or by reason of tort (including but not limited to negligence) for any loss of profit, loss of use, loss of production, loss of contracts or for any financing costs or for any indirect or consequential damage whatsoever that may be suffered by the Buyer.

In the event and to the extent that NovAtel shall have any liability to Buyer pursuant to the terms of the Order, NovAtel shall be liable to Buyer only for those damages which have been foreseen or might have reasonably been foreseen on the date of effectivity of the Order and which are solely an immediate and direct result of any act or omission of NovAtel in performing the work or any portion thereof under the Order and which are not in the aggregate in excess of ten (10%) percent of the total Order price.

Warranty Policy

NovAtel Inc. warrants that its Global Navigational Satellite Systems (GNSS) products are free from defects in materials and workmanship, subject to the conditions set forth below, for the following time periods:

OEMV-3 Receivers including SPAN-SE	One (1) Year
IMU Units (return to manufacturer) [1]	One (1) Year
Antennas	One (1) Year
Cables and Accessories	Ninety (90) Days
Computer Discs	Ninety (90) Days
Software Warranty	One (1) Year

Date of sale shall mean the date of the invoice to the original customer for the product. NovAtel's responsibility respecting this warranty is solely to product replacement or product repair at an authorized NovAtel location only.

Determination of replacement or repair will be made by NovAtel personnel or by technical personnel expressly authorized by NovAtel for this purpose (*continued on page 17*).

WARNING: **Only return an IMU to its manufacturer and not to NovAtel.**

---

1. Litton: Northrop Grumman/Litton Systems, Inc.
Navigation Systems Division (NSD)
21240 Burbank Blvd.
Woodland Hills, CA 91367

iMar: iMAR GmbH
Im Reihersbruch 3
D-66386 St. Ingbert
Germany

Honeywell: Honeywell International Inc.
2600 Ridgway Parkway *(Ridgway is really not spelled with an 'e')*
Minneapolis, MN 55413

When returning a Litton or Honeywell IMU from outside the U.S., follow these steps:

a) Include a copy of the original U.S. export permit with it.

b) Send the unit to Litton or Honeywell, with the following wording on the documentation: "Shipped in accordance with 22 CFR 123.4 (a) (1)", using air transport and not a carrier service. The repaired or replaced device will be returned to you under this same CFR exemption.

c) Identify the paperwork with the value of the hardware ($), the country of origin as U.S. and the Incoterms if applicable (for example, FOB, FAS, CIF Ex-Works).

d) Lastly, please clearly note on the paperwork to notify, upon receipt, Honeywell's customs broker, "EXPIDITORS", or for Litton, "FOR CUSTOMS CLEARANCE BY: FedEx Trade Networks, 19601 Hamilton Ave. Torrance, CA 90502-1309, U.S.A.".

NovAtel warrants that during the Warranty Period that (a) the Product will be free from defects in material and workmanship and conform to NovAtel specifications; and (b) the software will be free from error which materially affect performance. THESE WARRANTIES ARE EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NOVATEL SHALL IN NO EVENT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE DUE TO ANY CAUSE.

Purchaser's exclusive remedy for a claim under this warranty shall be limited to the repair or replacement at NovAtel's option and at NovAtel's facility, of defective or nonconforming materials, parts or components or in the case of software, provision of a software revision for implementation by the Buyer. All material returned under warranty shall be returned to NovAtel prepaid by the Buyer and returned to the Buyer, prepaid by NovAtel.

THE FOREGOING WARRANTIES DO NOT EXTEND TO (I) NONCONFORMITIES, DEFECTS OR ERRORS IN THE PRODUCTS DUE TO ACCIDENT, ABUSE, MISUSE OR NEGLIGENT USE OF THE PRODUCTS OR USE IN OTHER THAN A NORMAL AND CUSTOMARY MANNER, ENVIRONMENTAL CONDITIONS NOT CONFORMING TO NOVATEL'S SPECIFICATIONS, OR FAILURE TO FOLLOW PRESCRIBED INSTALLATION, OPERATING AND MAINTENANCE PROCEDURES, (II) DEFECTS, ERRORS OR NONCONFORMITIES IN THE PRODUCTS DUE TO MODIFICATIONS, ALTERATIONS, ADDITIONS OR CHANGES NOT MADE IN ACCORDANCE WITH NOVATEL'S SPECIFICATIONS OR AUTHORIZED BY NOVATEL, (III) NORMAL WEAR AND TEAR, (IV) DAMAGE CAUSED BY FORCE OF NATURE OR ACT OF ANY THIRD PERSON, (V) SHIPPING DAMAGE; OR (VI) SERVICE OR REPAIR OF PRODUCT BY THE DEALER WITHOUT PRIOR WRITTEN CONSENT FROM NOVATEL. IN ADDITION, THE FOREGOING WARRANTIES SHALL NOT APPLY TO PRODUCTS DESIGNATED BY NOVATEL AS BETA SITE TEST SAMPLES, EXPERIMENTAL, DEVELOPMENTAL, PREPRODUCTION, SAMPLE, INCOMPLETE OR OUT OF SPECIFICATION PRODUCTS OR TO RETURNED PRODUCTS IF THE ORIGINAL IDENTIFICATION MARKS HAVE BEEN REMOVED OR ALTERED. THE WARRANTIES AND REMEDIES ARE EXCLUSIVE AND ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, WRITTEN OR ORAL, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXCLUDED. NOVATEL SHALL NOT BE LIABLE FOR ANY LOSS, DAMAGE, EXPENSE, OR INJURY ARISING DIRECTLY OR INDIRECTLY OUT OF THE PURCHASE, INSTALLATION, OPERATION, USE OR LICENSING OR PRODUCTS OR SERVICES. IN NO EVENT SHALL NOVATEL BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE DUE TO ANY CAUSE.

There are no user serviceable parts in the GNSS receiver and no maintenance is required. When the status code indicates that a unit is faulty, replace with another unit and return the faulty unit to NovAtel Inc.

Before shipping any material to NovAtel or Dealer, please obtain a Return Material Authorization (RMA) number from the point of purchase.

Once you have obtained an RMA number, you will be advised of proper shipping procedures to return any defective product. When returning any product to NovAtel, please return the defective product in the original packaging to avoid ESD and shipping damage.

Customer Service

Firmware Upgrades

Firmware upgrades are firmware releases, which increase basic functionality of the receiver from one model to a higher level model type. When available, upgrades may be purchased at a price, which is the difference between the two model types on the current NovAtel GNSS Price List plus a nominal service charge.

Please refer to the *PC Software and Firmware* chapter in the *OEMV Installation and Operation User Manual*.

Contact Information

Firmware upgrades are accomplished through NovAtel authorized dealers.

Contact your local NovAtel dealer first for more information. To locate a dealer in your area or if the problem is not resolved, contact NovAtel Inc. directly using one of the following methods:

Call the NovAtel GNSS Hotline at 1-800-NOVATEL (North America), or 403-295-4900 (international)

Fax: 403-295-4901 E-mail: support@novatel.ca Web site: http://www.novatel.com

Write: NovAtel Inc., Customer Service Dept., 1120 - 68 Avenue NE, Calgary, AB., Canada, T2E 8S5

✉ Before contacting NovAtel Customer Service regarding software concerns, please do the following**:**

1. Establish communication with the receiver.
2. Send the SETIMUTYPE command to re-establish communication with the IMU, see *Table 30* on *page 139*.
3. Log the following data to a file on your PC for 30 minutes:

   RXSTATUSB once
   RAWEPHEMB onchanged
   RANGECMPB ontime 1
   BESTPOSB ontime 1
   RXCONFIGA once
   VERSIONB once
   RAWIMUSB onnew
   INSPVASB ontime 0.1
   INSCOVSB onchanged
   INSUPDATEB onchanged
   BESTGPSPOSB ontime 1

4. Send the file containing the logs to NovAtel Customer Service using the support@novatel.com e-mail address.

Notices

CAUTION

1. This device incorporates circuitry to absorb most static discharges. However, severe static shock may cause inaccurate operation of the unit. Use anti-static precautions where possible.
2. This device is a precision instrument. It performs best when handled with care.

Foreword

Congratulations!

Congratulations on purchasing your Synchronized Position Attitude Navigation (SPAN) Technology system. SPAN features a tight integration of a NovAtel GNSS receiver and an Inertial Measurement Unit (IMU). SPAN provides continuous navigation information, using an Inertial Navigation System (INS), to bridge short Global Navigational Satellite Systems (GNSS) outages. Designed for dynamic applications, SPAN provides precise position, velocity and attitude information.

By complementing GNSS with inertial measurements, SPAN Technology provides robust positioning in challenging conditions where GNSS alone is less reliable. During short periods of GNSS outage, or when less than four satellites are received, SPAN Technology offers uninterrupted position and attitude output. The tight coupling of inertial technology with GNSS also provides the benefits of faster satellite reacquisition and faster RTK initialization after outages.

SPAN-SE receivers are the processing engines of the SPAN Technology system. Separate GNSS and IMU enclosures provide a simple modular system. This allows the IMU mounting at the most suitable location, while the GNSS receiver is mounted where it is most convenient. SPAN Technology provides a robust GNSS and Inertial solution as well as a portable, high-performance GNSS receiver in one system.

Scope

This manual contains sufficient information on the installation and operation of the SPAN system. It is beyond the scope of this manual to provide details on service or repair. Contact your local NovAtel dealer for any customer-service related inquiries, see *Customer Service* on *page 18*.

After the addition of accessories, an antenna and a power supply, the SPAN system is ready to go.

The receiver utilizes a comprehensive user-interface command structure, which requires communications through its communications (COM) ports. This manual also describes the INS specific commands and logs. Refer to the *OEMV Family Firmware Reference Manual* for information on the logs and commands available for the OEMV-3 that is the GNSS engine of your SPAN-SE. Visit www.novatel.com to download any NovAtel product manual. It is recommended that these documents be kept together for easy reference.

SPAN system output is compatible with post-processing software from NovAtel's Waypoint Products Group. Visit our Web site at www.novatel.com for details.

What's new in Version 2 of this manual?

SPAN3.620 is a feature release that provides users with AdVance RTK which was unavailable in previous SPAN on OEMV releases. Version two of this manual includes the PASHR log on page 239.

Prerequisites

The installation chapters of this document provide information concerning the installation requirements and considerations for the different parts of the SPAN system.

To run the SPAN system software, your personal computer must meet or exceed this minimum configuration:

- Microsoft Windows user interface (Windows 98 or higher)
- Pentium Microprocessor recommended
- VGA Display
- Windows compatible mouse or pointing device

Although previous experience with Windows is not necessary to use the SPAN system software, familiarity with certain actions that are customary in Windows will assist in the usage of the program. This manual has been written with the expectation that you already have a basic familiarity with Windows.

Chapter 1 Introduction

NovAtel's SPAN technology brings together two very different but complementary positioning and navigation systems namely GNSS and an Inertial Navigation System (INS). By combining the best aspects of GNSS and INS into one system, SPAN technology is able to offer a solution that is more accurate and reliable than either GNSS or INS alone could provide. The combined GNSS/INS solution has the advantage of the absolute accuracy available from GNSS and the continuity of INS through traditionally difficult GNSS conditions.

SPAN-SE is the solution engine of NovAtel’s leading-edge SPAN technology. It provides the user interface to SPAN and outputs raw measurement data or solution data over several communication protocols or to a removable SD Card. Multiple GNSS-synchronous strobes and event input lines offer easy integration into a larger system. Combining SPAN-SE with a SPAN-supported IMU creates a complete GNSS/INS system

Figure 1: SPAN-SE Receiver

Figure 2: SPAN System IMUs

The SPAN system consists of the following components:

- A SPAN-capable receiver, such as SPAN-SE. The SPAN-SE is capable of receiving and tracking different combinations of GPS, GLONASS, and L-band (CDGPS and OmniSTAR) signals using a maximum of 72 channels. Patented Pulsed Aperture Correlator (PAC) technology combined with a powerful microprocessor make possible multipath-resistant processing. Excellent acquisition and re-acquisition times allow this receiver to operate in environments where very high dynamics and frequent interruption of signals can be expected. The receiver also supports the timing requirements of the IMU and runs the real-time INS Kalman filter.

  The SPAN-SE also offers on-board data logging with a Secure Digital (SD) card, Ethernet connectivity, wheel sensor input and scalability for future GNSS advances.
- IMU Enclosure - The Inertial Measurement Unit (IMU) is housed in the IMU enclosure that provides a steady power supply to the IMU, and decodes and times the IMU output data. The IMU itself consists of three accelerometers and 3 gyroscopes (gyros) so that accelerations along specific axis and angular rotations can be measured. Several IMU types are supported and are listed in *Table 1, SPAN-SE Compatible Receiver and IMU Models* on *page 24* and *Table 30, IMU Type* on *page 139*.
- PC Software - Real-time data collection, status monitoring and receiver configuration is possible through NovAtel's Control and Display Unit (**CDU**) software utility, see *Section 3.2* on *page 37*.
- A dual-frequency GNSS or GNSS/GLONASS Antenna.

The GNSS receiver is connected to the IMU enclosure with an RS-232 or RS-422 serial link. A NovAtel GNSS antenna must also be connected to the receiver to track GNSS signals. Once the IMU enclosure, GNSS antenna and appropriate power supplies are attached, and a few simple configuration commands are entered, the SPAN system will be up and running and ready to navigate.

1.1 Fundamentals of GNSS/INS

GNSS positioning observes range measurements from orbiting Global Positioning System and GLONASS satellites. From these observations, the receiver can compute position and velocity with high accuracy. NovAtel GNSS positioning systems have been established as highly accurate positioning tools; however GNSS in general has some significant restrictions, which limit its usefulness in some situations. GNSS positioning requires line of site view to at least four satellites simultaneously. If these criteria are met, differential GNSS positioning can be accurate to within a few centimeters. If however, some or all of the satellite signals are blocked, the accuracy of the position reported by GNSS degrades substantially, or may not be available at all.

In general, an inertial navigation system (INS) uses forces and rotations measured by an IMU to calculate position, velocity and attitude. This capability is embedded in the firmware of SPAN capable receivers. Forces are measured by accelerometers in three perpendicular axes within the IMU and the gyros measure angular rotation rates around those axes. Over short periods of time, inertial navigation gives very accurate position, velocity and attitude output. The INS must have prior knowledge of its initial position, initial velocity, initial attitude, Earth rotation rate and gravity field. Since the IMU measures changes in orientation and acceleration, the INS determines changes in position and attitude, but initial values for these parameters must be provided from an external source. Once these parameters are known, an INS is capable of providing an autonomous solution with no external inputs. However, because of errors in the IMU measurements that accumulate over time, an inertial-only solution degrades with time unless external updates such as position, velocity or attitude are supplied.

The SPAN system's combined GNSS/INS solution integrates the raw inertial measurements with all available GNSS information to provide the optimum solution possible in any situation. By using the high accuracy GNSS solution, the IMU errors can be modeled and mitigated. Conversely, the continuity and relative accuracy of the INS solution enables faster GNSS signal reacquisition and RTK solution convergence.

The advantages of using SPAN technology are its ability to:

- Provide a full attitude solution (roll, pitch and azimuth)
- Provide continuous solution output (in situations when a GNSS-only solution is impossible)
- Provide faster signal reacquisition and RTK solution resolution (over stand-alone GNSS because of the tightly integrated GNSS and INS filters)
- Output high-rate (up to 100 or 200 Hz depending on your IMU model and other logging selections) position, velocity and attitude solutions for high-dynamic applications
- Use raw phase observation data (to constrain INS solution drift even when too few satellites are available for a full GNSS solution)

1.2 Models and Features

All SPAN system receivers are factory configurable for L1/L2 RTK capability and are compatible with an IMU. See *Table 1* for firmware model details.

Table 1: SPAN-SE Compatible Receiver and IMU Models

Model Name	Max. Output Rate	Compatible IMUs	SPAN-SE Models
IMU-H58 IMU-H62	100 Hz	HG1700-AG58 HG1700-AG62	SPAN-SE-RT2-G-S-I SPAN-SE-RT2-S-I
IMU-LN200	200 Hz	LN-200 200 and 400 Hz models	SPAN-SE-RT2-G-S-J SPAN-SE-RT2-S-J
IMU-FSAS-EI	200 Hz	iIMU-FSAS	SPAN-SE-RT2-G-S-J SPAN-SE-RT2-S-J

Each model is capable of multiple positioning modes of operation. For a discussion on GNSS positioning, please refer to the *OEMV Family Installation and Operation User Manual.*

Each model has the following standard features:

- Rugged shock, water, and dust-resistant enclosure
- NovAtel's advanced OEMV L1/L2 GNSS/GLONASS and PAC technology
- Four bi-directional COM ports which support data transfer rates of up to 921,600 bits/s [1]
- A removable SD Card slot for on-board data collection
- A USB port for PC communication
- A serial port capable of communication with an IMU. See also *Table 1 on page 24*.
- An Ethernet port for TCP (or UDP) communication with the receiver
- Field-upgradeable firmware (program software).[2] What makes one model different from another is software, not hardware. This unique feature means that the firmware can be updated any time, anywhere, without any mechanical procedures whatsoever. For example, a model with L1/L2-only capabilities can be upgraded to a model with L1/L2 RT-2 in only a few minutes in your office (instead of the days or weeks that would be required if the receiver had to be sent to a service depot). All that is required to unlock the additional features is a special authorization code. Refer to the SPANAUTH command on *page 149* for further details on this topic.

SPAN currently supports Honeywell, iMAR and Litton IMUs. When using an IMU with SPAN, it is housed in an enclosure with a PCB board to handle power, communication and data timing. See *Appendix A, Technical Specifications* starting on *page 62* for details.

1. Rates higher than 115, 200 are not standard on most PCs and may require extra PC hardware
2. You must have a valid Post Contractual Support (PCS) subscription, refer to our Web site at www.novatel.com.

Chapter 2 SPAN-SE Installation

This chapter contains instructions to set up your SPAN-SE system.

SPAN-SE uses NovAtel's powerful OEMV receiver technology as its GNSS engine. The OEMV delivers many enabling features like GNSS/GLONASS capability and AdVance RTK, which are both supported in SPAN-SE. A dedicated CPU, for real-time GNSS/INS processing on these cards, results in fast data rates and low raw data and solution latency for highly dynamic or time-critical applications.

2.1 SPAN-SE Hardware Description

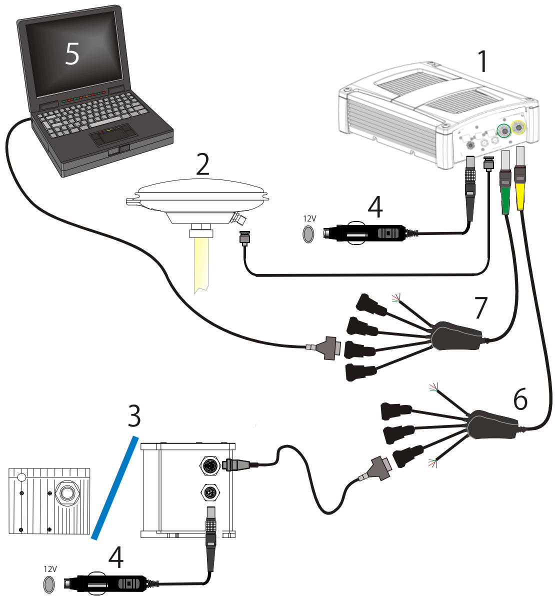
The basic hardware setup consists of a SPAN-SE receiver (see *Figure 1* on *page 22*) connected to an IMU (see *Figure 2* on *page 22*), a GNSS antenna and a power supply.

For real time differential operation, a communication link between the base and rover(s) is necessary. This can be a null-modem cable or a radio link.

Figure 3 on *page 27* shows a basic setup. *Figure 4* on *page 28* shows a setup with a radio link on the base and the rover using the LN-200 IMU and the iIMU as an option. For more details on the connections between the SPAN-SE receiver and the iIMU, see *Figure 26, iIMU Interface Cable Connections with a SPAN-SE* on *page 78*.

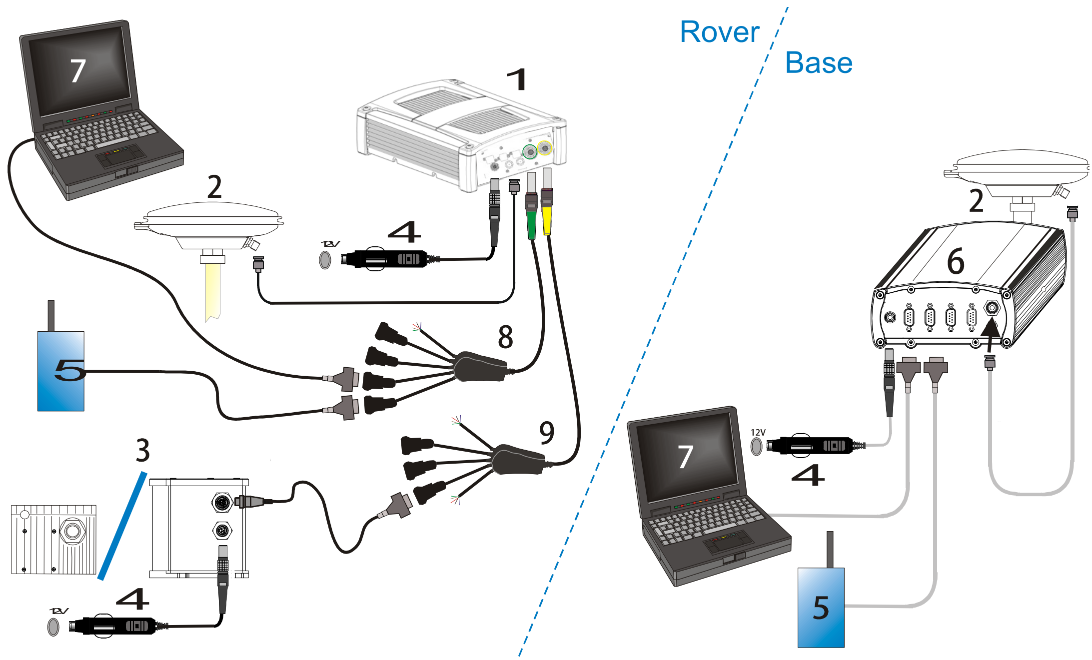
If your IMU enclosure and IMU have come separately, additional installation instructions for installing the IMU can be found in *Appendix E, HG1700 IMU Installation* starting on *page 277* or *Appendix F, LN-200 IMU Installation* starting on *page 282*.

For more information on SPAN-SE cables, please see *Appendix A, Technical Specifications* on *page 62*.



Reference	Description
1	SPAN-SE receiver with an on-board SD Card for data storage
2	User-supplied NovAtel GNSS antenna
3	LN-200, HG-1700 or iIMU FSAS IMU and IMU interface cable to the connector labelled IMU on the SPAN-SE I/O 2 yellow cable. For the other connections, that only apply to the iIMU-FSAS, see *Section A.2.2.1, iIMU-FSAS Interface Cable* starting on *page 78*.
4	User-supplied power supply: SPAN-SE rover (1): +9 to +28 V DC ProPak-V3 base (6): +9 to +18 V DC Separate supply for IMU (3): see *Table 3* on *page 33*
5	User-supplied PC/laptop, for setting up and monitoring, to COM1 on the ProPak-V3, or in the case of the SPAN-SE to one of the four available COM ports, the USB host port or the Ethernet port 6User-supplied base station (ProPak-V3) receiver
6	SPAN-SE I/O 2 yellow cable see *Section A.1.1.3, I/O 2 Yellow Cable (NovAtel part number 01018133)* on *page 68*the ProPak-V3, or in the case of the SPAN-SE to one of the four available COM ports, the USB host port or the Ethernet port
7	SPAN-SE I/O 1 green cable see *Section A.1.1.2, I/O 1 Green Cable (NovAtel part number 01018134)* on *page 66*

Figure 3: Basic SPAN-SE Set-Up



Reference	Description
1	SPAN-SE receiver with an on-board SD Card for data storage
2	User-supplied NovAtel GNSS antenna
3	LN-200, HG-1700 or iIMU FSAS IMU and IMU interface cable to the connector labelled IMU on the SPAN-SE I/O 2 yellow cable. For the other connections, that only apply to the iIMU-FSAS, see *Section A.2.2.1, iIMU-FSAS Interface Cable* starting on *page 78*.
4	User-supplied power supply: SPAN-SE rover (1): +9 to +28 V DC ProPak-V3 base (6): +9 to +18 V DC Separate supply for IMU (3): see *Table 3* on *page 33*
5	User-supplied radio device to the connector labelled OEMV3 on the SPAN-SE I/O green cable.
6	User-supplied base station (ProPak-V3) receiver
7	User-supplied PC/laptop, for setting up and monitoring, to COM1 on the ProPak-V3, or in the case of the SPAN-SE to one of the four available COM ports, the USB host port or the Ethernet port
8	SPAN-SE I/O 1 green cable see *Section A.1.1.2, I/O 1 Green Cable (NovAtel part number 01018134)* on *page 66*
9	SPAN-SE I/O 2 yellow cable see *Section A.1.1.3, I/O 2 Yellow Cable (NovAtel part number 01018133)* on *page 68*

Figure 4: SPAN-SE Set-Up

The sections that follow outline how to set up the system's parts and cables. See the specifications starting on *page 86* for the NovAtel part numbers of SPAN-SE cables and their pinouts.

✉ Data can be collected through any of the peripheral devices: USB, Ethernet, or serial COM ports. Ensure that your peripheral is configured for a suitably high baud rate to handle the size of the logs you request. USB is recommended for logging of high-rate data.

Data storage is via a Secure Digital (SD) memory card that you access in the front of the SPAN-SE. See also *Section 3.8, The SD Card* starting on *page 54*.

Figure 5: SD Memory Card

The back panel of the SPAN-SE is shown in *Figure 6*. The SPAN-SE has multiple COM and I/O connectors. Note that there is more than one interface cable with the SPAN-SE.

Figure 6: Receiver Enclosure Back Panel

Table 2 on *page 30* shows a summary of the receiver's back panel port names.

Table 2: Receiver Enclosure Back Panel Labels

SPAN Enclosure	Port Label	Description
SPAN-SE	**Power 9-28 VDC**	Supply Voltage
	USB Host	USB Host
	USB Device	USB Device
	Ethernet	Ethernet
	GPS1	Antenna 1
	GPS2	Antenna 2 (optional)
	I/O 1	Green multi-pin connector 1 containing SPAN-SE COM ports, OEMV COM port, event inputs, and output strobes
	I/O 2	Yellow multi-pin connector 2 containing SPAN-SE COM ports, IMU COM port, event inputs, and output strobes

Each connector can be inserted in only one way, to prevent damage to both the receiver and the cables. Furthermore, the connectors that are used to mate the cables to the receiver require careful insertion and removal. Observe the following when handling the cables.

- To insert a cable, make certain you are using the appropriate cable for the port - the I/O cable has a different connector (number of pins) than the power cable
- Insert the connector until it is straight on and secure
- To remove a cable, grasp it by the connector and pull

WARNING: **DO NOT PULL DIRECTLY ON THE CABLE.**

Review this section's hardware set-up subsections and follow the numbered steps, in bold, to install your SPAN system. The example graphics, in the sections that follow, show the connections on the back of a SPAN-SE receiver.

2.2 SPAN-SE Hardware Installation

2.2.1 Mount Antenna

For the best possible positioning precision and accuracy, as well as to minimize the risk of damage, ensure that the antenna is securely mounted on a stable structure that will not sway or topple. Where possible, select a location with a clear view of the sky to the horizon so that each satellite above the horizon can be tracked without obstruction. The location should also be one that minimizes the effect of multipath interference. For a discussion on multipath, please refer to the *GNSS Reference Book*.

2.2.2 Mount IMU

Mount the IMU in a fixed location where the distance from the IMU to the GNSS antenna phase center is constant. Ensure that the orientation with respect to the vehicle and antenna is constant and that the distance and relative direction between them is fixed.

The IMU should be mounted in such that the positive Z-axis marked on the enclosure points up and the Y-axis points forward through the front of the vehicle, in the direction of track. The IMU can be mounted in other orientations, see *Section 29, Full Mapping Definitions* on *page 136*, but this can make interpreting the raw IMU and attitude output more difficult.

Also, it is important to measure the distance from the IMU to the antenna (the Antenna Lever Arm), on the first usage, on the axis defined on the IMU enclosure. See *Section 3.4.6, Lever Arm Calibration Routine* starting on *page 48*. See also *Appendix A, Technical Specifications* starting on *page 62*, which gives dimensional drawings of the IMU enclosures.

---

1. The closer the antenna is to the IMU, the more accurate the position solution. Also, your measurements when using the SETIMUTOANTOFFSET command must be as accurate as possible, or at least more accurate than the GNSS positions being used. **For example, a 10 cm error in recording the antenna offset will result in at least a 10 cm error in the output. Millimeter accuracy is preferred.**
2. The offset from the IMU to the antenna, and/or a user point device, must remain constant especially for RTK or DGPS data. Ensure the IMU, antenna and user point device are bolted in one position perhaps by using a custom bracket.

2.2.3 Connect Interface Cables

The SPAN-SE has two circular connectors on the back panel. Each connector has a cable that breaks out the serial ports into DB9 connectors and the input and output event signals to bare wires. Each peripheral signal is identified on the cable with a label.

See *Section A.2.3.3, Electrical and Environmental* on *page 86* for more information on signals, wiring and pin-out information of the SPAN-SE port and its cables.

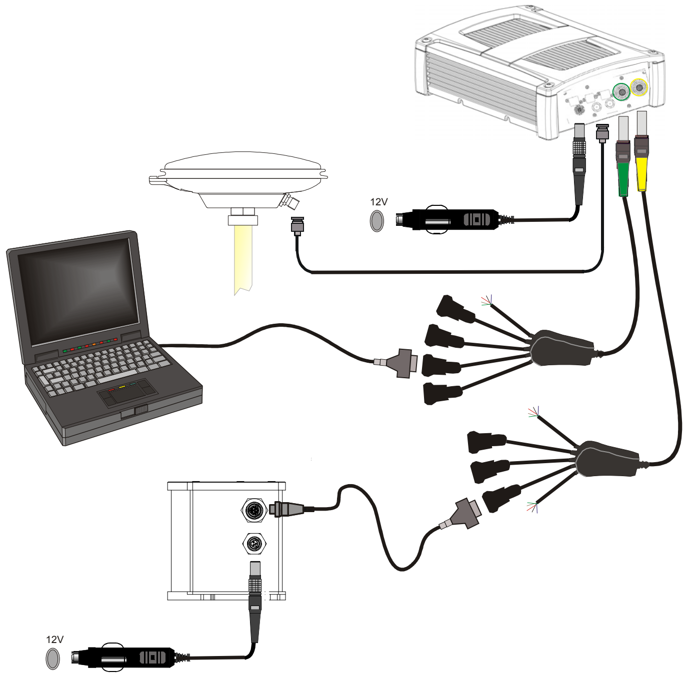
1. **Connect the I/O 1 green cable’s 30-pin connector to the I/O 1 green port on the SPAN-SE.**
2. **Connect the I/O 2 yellow cable’s 30-pin connector to the I/O 2 yellow port on the SPAN-SE.**
3. **Connect a communications cable.**

   If you want to connect via a serial connection, the I/O 1 green cable has a DB9 connectors for COM3 and COM4 and the I/O 2 yellow cable has DB9 connections for COM1 and COM2.

   If a USB connection is required, connect a USB cable to the USB Device port. USB Host support is not available at this time.

   If an Ethernet connection is required, connect a network cable to the Ethernet port.

4. **Connect the I/O 2 yellow cable's IMU connector to an IMU COM port using the IMU's interface cable. The figure below shows the HG-1700 or LN-200 connections. For the iIMU-FSAS connections with a SPAN-SE, see *iIMU-FSAS Interface Cable*** on *page 78*.



5. **Connect the antenna to the antenna port on the receiver using an appropriate coaxial cable.**

2.2.4 Connect Power

The SPAN-SE receiver requires an input supply voltage between +9 VDC and +28 VDC. The power cable supplied has bare leads that can be connected to an appropriate DC power supply. The receiver has an internal power module that does the following:

- filters and regulates the supply voltage
- protects against over-voltage, over-current, and high-temperature conditions
- provides automatic reset circuit protection

There is always a drop in voltage between the power source and the power port due to cable loss. Improper selection of wire gauge can lead to an unacceptable voltage drop at the SPAN system. A paired wire run represents a feed and return line. Therefore, a 2-m wire pair represents a total wire path of 4 m. For a SPAN system operating from a 12 V system, a power cable longer than 2.1 m (7 ft.) should not use a wire diameter smaller than 24 AWG.

Each IMU requires its own power supply, see *Table 3 on page 33*.

Table 3: IMU Power Supply

IMU	Power Requirement
LN-200	+12 to +28 V DC
iIMU-FSAS	+10 to +34 V DC
HG1700	+12 to +28 V DC

For pin-out information on the power connector on the SPAN-SE, see *Section A.2.3.3, Electrical and Environmental* on *page 86*. Details on each IMU's power ports and cables are in the IMU appendices starting on *page 62*.

2.2.5 Power Button

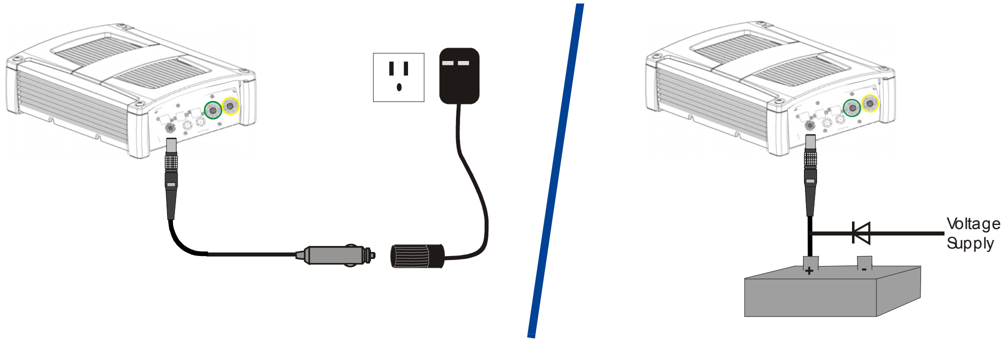
The power button on the front of the SPAN-SE, see *Figure 7*, is managed by software. When the system receives sufficient power, it powers itself on without the need to press the power button. However, the power button is connected directly to the onboard power supply to re-enable the system when it has been automatically shut down and to manually power down the system. The state of the button depends on the amount of time the power button is depressed.

Table 4: Power Button States

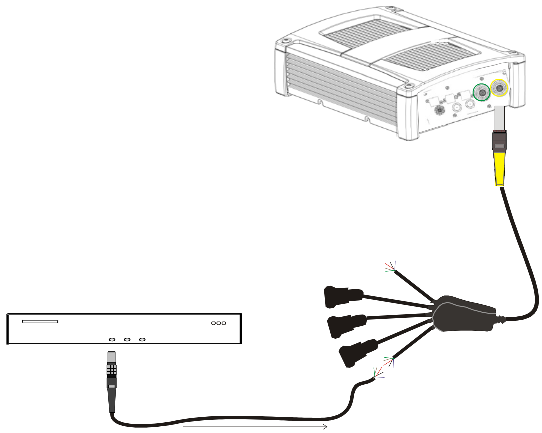
Seconds Button is Held	Button Action
3-10 seconds	SPAN-SE is powered off
> 10 seconds	Factory reset (see the FRESET command on *page 105)*

Figure 7: SPAN-SW Power Button

6. **Apply power to the IMU and to the receiver.** It is recommended that a back-up battery is placed between the receiver and its voltage supply to act as a power buffer if installed in a vehicle. When a vehicle engine is started, power can dip to 9.6 VDC or cut-out to ancillary equipment causing the receiver and IMU to lose lock and calibration settings.



☒ *For advanced users*: If you have additional equipment to connect to your system requiring an output timing pulse, or an input pulse into SPAN-SE, see *Section 3.10, Synchronizing External Equipment* starting on *page 58*. See *Section A.1.1.3, I/O 2 Yellow Cable (NovAtel part number 01018133)* on *page 68* for its bare wire pin-outs. The jacket insulation is cut away slightly from the end but the insulation on each wire is intact. Then, refer to your device's documentation for information on its connectors and cables. The arrow along the cable in the figure indicates a MARKIN pulse from the user device on the right to the SPAN-SE I/O port.



Chapter 3 SPAN-SE Operation

Before operating your SPAN system, ensure that you have followed the installation and setup instructions in *Chapter 2, SPAN-SE Installation* starting on *page 26.*

You can use NovAtel's **CDU** software to monitor data in real-time.

SPAN system output is compatible with post-processing software from NovAtel's Waypoint Products Group. Visit our Web site at www.novatel.com for details.

3.1 Definition of Reference Frames Within SPAN

The reference frames that are most frequently used throughout this manual are the following:

- The Local-Level Frame
- The SPAN Computation Frame
- The Enclosure Frame
- The Vehicle Frame

3.1.1 The Local-Level Frame (ENU)

The definition of the local level coordinate frame is as follows:

- z-axis– pointing up (aligned with gravity)
- y-axis– pointing north
- x-axis – pointing east

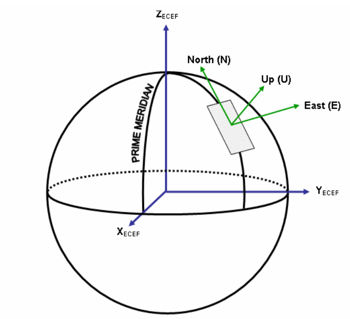


Figure 8: Local-Level Frame (ENU)

3.1.2 The SPAN Computation Frame

The definition of the SPAN computation frame is as follows:

- z-axis– pointing up (aligned approximately with gravity)

- y-axis– defined by how user has mounted the IMU
- x-axis – defined by how user has mounted the IMU

To determine your SPAN x-axis and y-axis, see *Table 29* on *page 136*. This frame is also known as the computation frame and is the default frame that attitude is output in.

3.1.3 The Enclosure Frame

The definition of the enclosure frame is defined on the IMU and represents how the sensors are mounted in the enclosure. If the IMU is mounted with the z-axis (as marked on the IMU enclosure) pointing up, the IMU enclosure frame is the same as the SPAN computation frame.

This origin of this frame is not the enclosure center, but the Center of Navigation (sensor center).

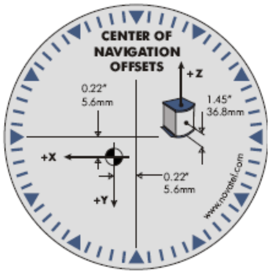


Figure 9: The Enclosure Frame

3.1.4 The Vehicle Frame

The definition of the vehicle frame is as follows:

- z-axis– points up through the roof of the vehicle perpendicular to the ground
- y-axis– points out the front of the vehicle in the direction of travel
- x-axis–completes the right-handed system (out the right-hand side of the vehicle when facing forward)

See the VEHICLEBODYROTATION command on *page 155* for information on entering the rotation into the system and see the RVBCALIBRATE command on *page 128* for information on calculating this rotation.

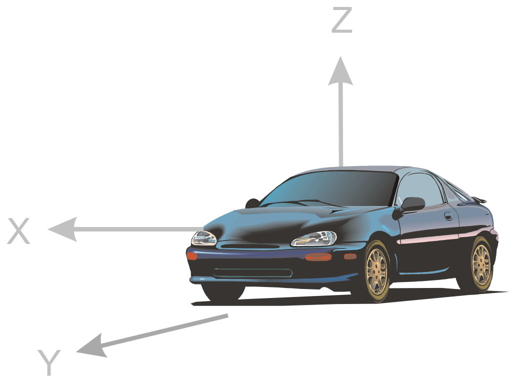


Figure 10: Vehicle Frame

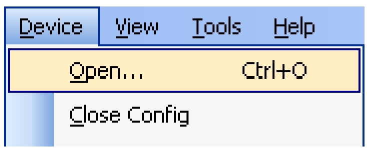
3.2 Communicating with the SPAN System

Once the receiver is connected to the PC, antenna, and power supply, install NovAtel's OEMV PC Utilities (**CDU** and *Convert*). You can find installation instructions in your receiver's *Quick Start Guide*. (Alternatively, you can use a terminal emulator program such as HyperTerminal to communicate with the receiver.) Refer also to the **CDU** Help file for more details on **CDU**. The Help file is accessed by choosing *Help* from the main menu in **CDU**.

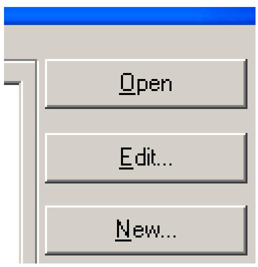
Start **CDU** on your PC to enable communication:

1. Launch **CDU** from the *Start* menu folder specified during the installation process. The default location is *Start* | *Programs* | *NovAtel PC Software* | *NovAtel CDU*.

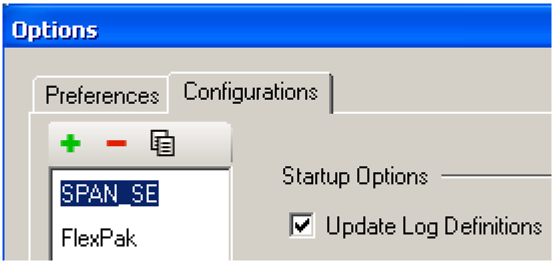
2. Select *Open....* from the *Device* menu.



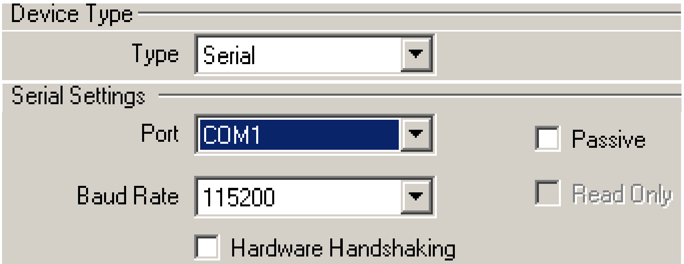
3. Select the *New...* button in the *Open* dialog box. The *Options* | *Configuration* dialog opens.



4. Use the  button at the top of the configurations selection box to add a new configuration. To delete a configuration, select it from the list and click on the button. To duplicate an existing configuration, click on the  button. You can select any name in the list and edit it to change it.



5. Select *Serial,* or *Network,* from the *Type* list and select the PC/laptop port, that the receiver is connected to, from the *Port* list. If selecting a network connection, you should have configured an IP address into the receiver prior to connecting, see *SPAN-SE Ethernet Connection* on *page 60*.



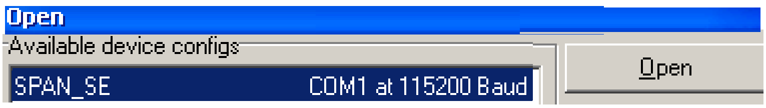
6. Select your desired baud rate from the *Baud Rate* list. If you are logging high-rate data, we recommend using the highest baud rate that your hardware is able to support.

7. Uncheck the *Use hardware handshaking* checkbox.

8. Select *OK* to save the new device settings.

9. Select the new configuration from the *Available device configs* area of the *Open* dialog.

10. Select the *Open* button to open receiver communications.



11. As **CDU** establishes the communication session with the receiver, a progress box is displayed.

12. Select *Tools | Logging Control Window* from the **CDU** main menu to control the receiver's logging to files and serial ports. Refer to **CDU**'s on-line Help for more information.

13. Use the *Console* window to enter commands. See also *Section 3.6, Data Collection for Post Processing* on *page 52*.

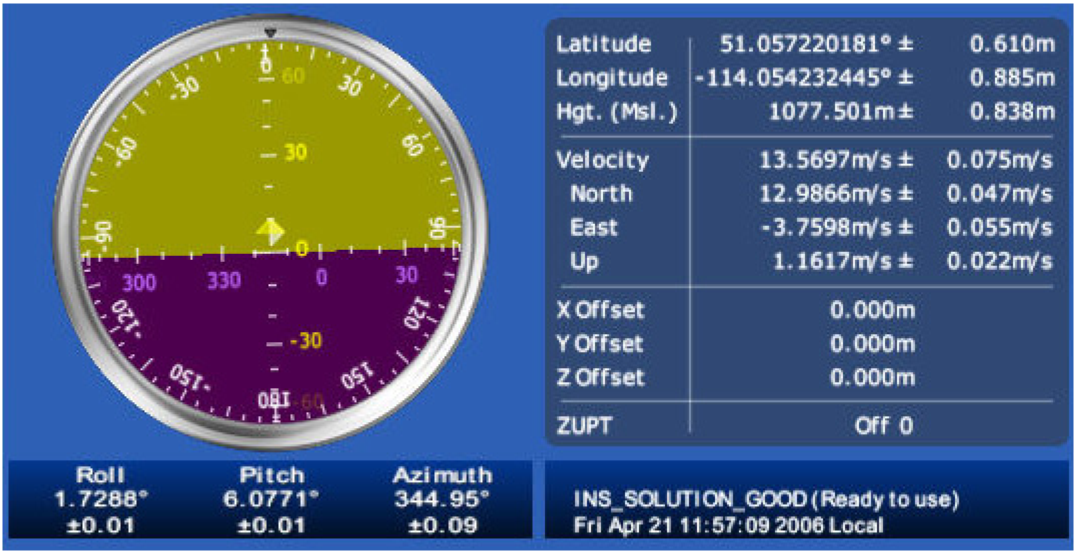
If you have to power down your receiver, ensure that all windows, other than the Console window, are closed in **CDU** and then use the SAVECONFIG command.

3.2.1 INS Window in CDU

CDU is a 32 bit Windows application. The application provides a graphical user interface to allow you to set-up and monitor the operation of the SPAN system by providing a series of windows.

The INS Window in **CDU** is described below. Please refer to the *OEMV Family Installation and Operation User Manual* for more details on **CDU** and other OEMV Family PC software programs.

•***INS Window:*** The Position, Velocity and Attitude (roll, pitch and azimuth) sections display data from the INSPVA log along with standard deviations calculated from the INSCOV log. Information in the ZUPT (Zero Velocity Update) section reflects whether a ZUPT has been applied. The receiver uses the *X,Y* and *Z Offset* fields to specify an offset from the IMU, for the output position and velocity of the INS solution, as specified by the SETINSOFFSET command or **CDU**'s SPAN wizard. The *INS Configuration/Status* section displays the IMU type, IMU Status and local date/time information. The dial is a graphical display of the Roll, Pitch and Azimuth values indicated by an arrow on each axis.



3.3 Software Configuration

3.3.1 GNSS Configuration

The GNSS configuration can be set up for different accuracy levels such as single point, SBAS, DGPS and RTK (RTCA, RTCM, RTCM V3 and CMR). SPAN-SE receivers can also be set up for Omnistar HP, Omnistar VBS or CDGPS.

With no additional configuration, the system operates in single point mode.

Once your base and SPAN rover are set up, you can configure them as shown in the configuration examples starting on *page 41*. The section on *page 41* gives an example of how to set up your base and rover for GNSS + GLONASS RTCA operation. Refer to the *OEMV Family* user manuals for details on DGPS, RTK, L-band or SBAS setup and operation.

The GNSS positioning mode of operation can also be configured using the position mode wizard in NovAtel's Control and Display Unit (**CDU**) software utility. See **CDU**'s *Help* and its wizard screens for more information.

SPAN-SE RTK ROVER CONFIGURATION

☒ Command description brackets [ ] represent optional parameters.

RTK correction data is input to SPAN-SE using the port labelled OEMV3 on the green cable. The port is configured using the GNSSCARDCONFIG command at the rover as follows:

gnsscardconfig [card] [port] rx_type tx_type baud [*com control parameters*]

See *page 106* for a detailed description.

For example:

Via SPAN-SE COM1-COM4, USB, or ethernet

RTCA `gnsscardconfig rtca none 9600`

RTCM `gnsscardconfig rtcm none 9600`

RTCMV3 `gnsscardconfig rtcmv3 none 9600`

CMR+ `gnsscardconfig cmr none 9600`

CMR `gnsscardconfig cmr none 9600` *(same as CMR+)*

☒ The baud rate of the rover must match the baud rate of the RTK correction data source.

3.3.2 SPAN IMU Configuration

3.3.2.1 SPAN Configuration Manually

Follow these steps to enable INS as part of the SPAN system using software commands or see *SPAN Configuration with CDU* on *page 42* to see the preferred method using NovAtel's *Control and Display Unit* (**CDU**) software utility:

1. **Issue the SETIMUTYPE command to specify the type of IMU being used.**

Basic configuration of the SPAN system is now complete. The inertial filter starts once the GNSS solution reaches FINESTEERING status and the IMU is connected.

☒ A GNSS antenna must be connected and tracking satellites for operation.

---

2. **Issue the SETIMUTOANTOFFSET command** to enter the distance from the IMU to the GNSS antenna, see *page 137*.

   The offset between the antenna phase centre and the IMU center of navigation must remain constant and be known accurately (m). The X (pitch), Y (roll) and Z (azimuth) directions are clearly marked on the IMU enclosure. The SETIMUTOANTOFFSET parameters are (where the standard deviation fields are optional and the distances are measured from the IMU to the Antenna):

   x_offset y_offset z_offset [x_stdev] [y_stdev] [z_stdev]

   A typical RTK GNSS solution is accurate to a few centimeters. For the integrated INS/GNSS system to have this level of accuracy, the offset must be measured to within a centimeter. Any offset error between the two systems shows up directly in the output position. For example, a 10 cm error in recording this offset will result in at least a 10 cm error in the output.

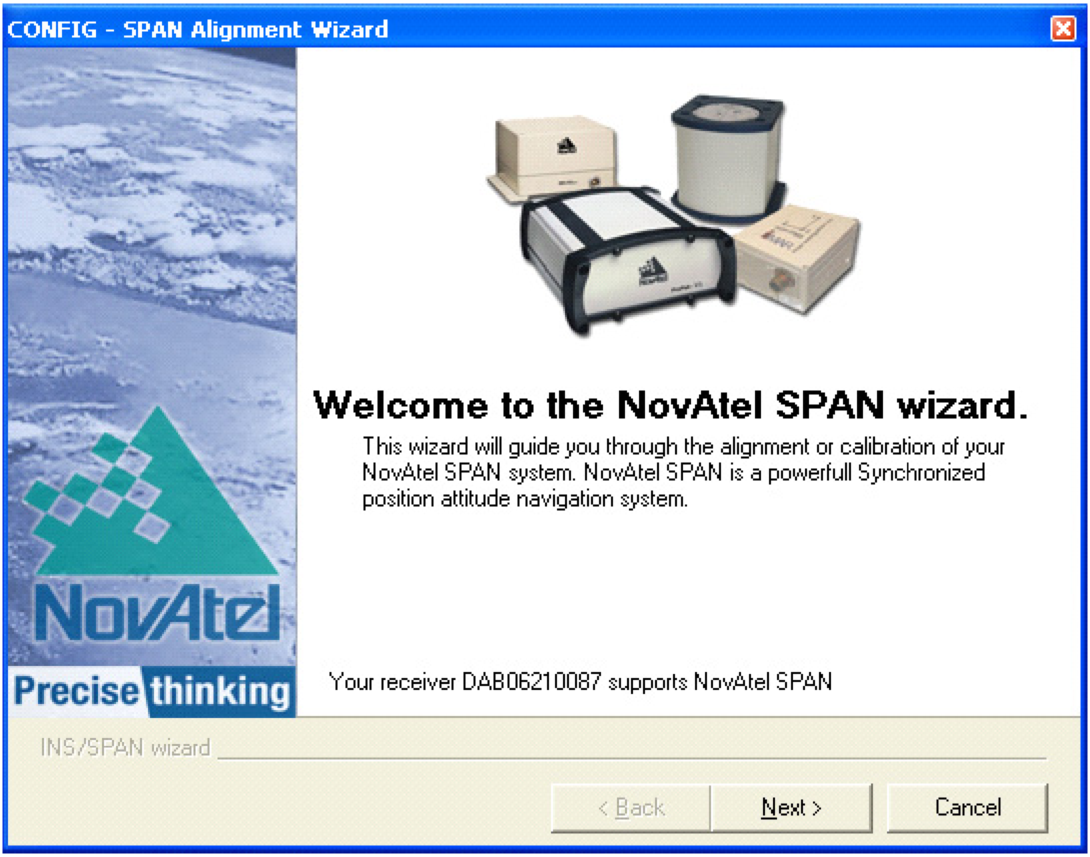
   If it is impossible to measure the IMU to GNSS antenna offset precisely, the offset can be estimated by carrying out the Lever Arm Calibration Routine. See *Section 3.4.7, Vehicle to SPAN frame Angular Offsets Calibration Routine* on *page 49*.

3.3.2.2 SPAN Configuration with CDU

Follow these steps to enable INS as part of the SPAN system using the NovAtel **CDU** software utility:

> ✉ The **CDU** screen shots in this manual are from **CDU** Version 3.3.0.3 and may differ from the current/your **CDU** version.

1. **SPAN basic configuration**: Select *Tools | SPAN Alignment Wizard* from the main menu of **CDU**. This wizard takes you through the steps to complete a coarse or fast alignment, select the type of IMU and configure the receiver port, connected to the IMU, to accept IMU data:



2. **Optional SPAN calibration:** Select *Tools | SPAN Calibration Wizard* from the main menu of **CDU**. The wizard takes you through the steps to calibrate your lever arm and/or vehicle to body rotation, as well as select the type of IMU and configure the receiver port connected to the IMU and to accept data:

✉ You need only run the *Calibration Wizard* if you need to calibrate the lever arm or vehicle to frame angular offsets. It is not required for the SPAN filter to run.

3.4 Real-Time Operation

SPAN operates through the OEMV command and log interface. Commands and logs specifically related to SPAN operation are documented in *Appendices B* and *C* of this manual respectively.

Real-time operation notes:

- Inertial data does not start until time is set and therefore, the SPAN system does not function unless a GNSS antenna is connected with a clear view of the sky.
- The Inertial solution is computed separately from the GNSS solution. The GNSS solution is available from the SPAN system through the GNSS-specific logs even without SPAN running. The integrated INS/GNSS solution is available through special INS logs documented in *Appendix C* of this manual.
- The INS/GNSS solution is available at the maximum rate of output of the IMU (100 or 200 Hz). Because of this high data rate, a shorter header format was created. These shorter header logs are defined with an S (RAWIMUSB rather than RAWIMUB). We recommend you use these logs instead of the standard header logs to save throughput.

Status of the inertial solution can be monitored using the inertial status field in the INS logs, *Table 5* below.

Table 5: Inertial Solution Status

Binary	ASCII	Description
0	INS_INACTIVE	IMU logs are present, but the alignment routine has not started; INS is inactive.
1	INS_ALIGNING	INS is in alignment mode.
2	INS_SOLUTION_NOT_GOOD	The INS solution is still being computed but the azimuth solution uncertainty has exceed 2 degrees. The solution is still valid but you should monitor the solution uncertainty in the INSCOV log. You may encounter this state during times when the GNSS, used to aid the INS, is absent. [1]
3	INS_SOLUTION_GOOD	The INS filter is in navigation mode and the INS solution is good.
6	INS_BAD_GPS_AGREEMENT	The INS filter is in navigation mode, and the GNSS solution is suspected to be in error. This may be due to multipath or limited satellite visibility. The inertial filter has rejected the GNSS position and is waiting for the solution quality to improve.
7	INS_ALIGNMENT_COMPLETE	The INS filter is in navigation mode, but not enough vehicle dynamics have been experienced for the system to be within specifications.

1. See also the *Frequently Asked Question* appendix, question *#8* on *page 296*

3.4.1 Configuration for Alignment

✉ All alignment and calibration activities should be conducted under open sky conditions for maximum system performance.

A coarse alignment routine requires the vehicle to remain stationary for at least 1 minute. If that is not possible, an alternate fast alignment routine is available. The fast or moving alignment is performed by estimating the attitude from the GNSS velocity vector and injecting it into the SPAN filter as the initial system attitude. See also *Section 3.4.1, Configuration for Alignment* starting on *page 45* for more details on coarse and fast alignments.

3.4.2 INS Configuration Command Summary

This section gives a brief recap of the commands necessary to get the SPAN system running.

1. **Issue the SETIMUTYPE command** to specify the type of IMU being used, see the SETIMU-TYPE command on *page 138.*

```
setimutype imu_ln200
```

2. **Issue the SETIMUTOANTOFFSET command** to enter the distance from the IMU to the GNSS antenna, see *page 137*.

```
setimutoantoffset 0.1 0.1 0.1 0.01 0.01 0.01
```

3.4.3 System Start-Up and Alignment Techniques

WARNING: **If logging to a PC, ensure the Control Panel's Power Settings on your PC are not set to go into Hibernate or Standby modes. Data will be lost if one of these modes occurs during a logging session.**

The system requires an initial attitude estimate to start the navigation filter. This is called system alignment. On start-up the system has no position, velocity or attitude information. When the system is first powered up, the following sequence of events happens:

1. The first satellites are tracked and coarse time is solved
2. Enough satellites are tracked to compute a position
3. Receiver "fine time" is solved, meaning the time on board the receiver is accurate enough to begin timing IMU measurements
4. Raw IMU measurements begin to be timed by the receiver and are available to the INS filter. They are also available to you in the RAWIMU or RAWIMUS log, see *page 249* and *page 252*. The INS Status field reports INS_INACTIVE.
5. The inertial alignment routine starts and the INS Status field reports INS_ALIGNING.
6. Alignment is complete and the INS Status field changes to INS_ALIGNMENT_COMPLETE. The system transitions to navigation mode.
7. The solution is refined using updates from GNSS. Once the system is operating within specifica-

tions and after some vehicle movement, the INS Status field changes to INS_SOLUTION_GOOD. This indicates that the estimated azimuth standard deviation is below 2°. If it increases above 2°, the status changes to INS_SOLUTION_NOTGOOD.

3.4.3.1 Coarse Alignment

The coarse alignment is the default alignment routine for SPAN. The alignment starts as soon as a GNSS solution is available, the receiver has computed fine time and the IMU is connected and configured. The vehicle must remain stationary for the alignment to happen. During the coarse alignment, accelerometer and gyro measurements are averaged over a period of time to measure Earth rotation and gravity. From these averaged measurements, initial estimates of roll, pitch and heading are computed. Because the coarse alignment uses averaged sensor output, the vehicle must remain stationary for the duration of the alignment, which is approximately 1 minute. The attitude estimates solved by the alignment are larger than the system specified attitude accuracy and vary upon the characteristics of the sensor and the geographic latitude of the system. Attitude accuracy converges with motion after the coarse alignment is complete (see *Section 3.4.4, Navigation Mode* on *page 47*).

If the system is stationary for less than 1 minute, the coarse alignment finishes early, provided at least 5 stationary seconds were detected. The quality of the coarse alignment is poorer with stationary durations of less than 1 minute.

3.4.3.2 Kinematic Alignment

If the preferred coarse alignment routine cannot be performed because the vehicle cannot remain stationary for the length of time required, an alternate alignment routine is available. The kinematic or moving alignment is performed by estimating the attitude from the GNSS velocity vector and injecting it into the SPAN filter as the initial system attitude.

Currently, this alignment routine is meant only for ground-based vehicles. The assumptions used for the alignment may not hold for marine or airborne applications. For the fast alignment routine to work optimally, the course-over-ground's azimuth and pitch must match the SPAN computation frame's azimuth and pitch. (For example, a plane being blown in the wind has a a large 'crab angle' and the course-over ground trajectory will not match the direction the IMU is pointing.)

Additional configuration parameters are necessary to enable the kinematic alignment. In order to simplify this configuration it is strongly suggested that you mount the IMU in parallel to the vehicle frame. The Y axis marked on the IMU enclosure, should point in the direction of travel.

Specify which IMU axes are most closely aligned with gravity using the SETIMUORIENTATION command. If the IMU is mounted with the Z-axis up and the Y-axis pointing in the direction of travel, then the command would be:

```
SETIMUORIENTATION 5
```

Specify the angular offsets between the SPAN frame and the vehicle frame (known as vehicle/body rotation or RVB) using the VEHICLEBODYROTATION command, see *page 149*. If the IMU is mounted coincidentally with the vehicle frame (defined as z up and y pointing in the direction of travel), then the command would be:

```
VEHICLEBODYROTATION 0 0 0
```

Alternatively, solve the vehicle to IMU frame angular offsets using the RVBCALIBRATE routine. See also *Section 3.4.7, Vehicle to SPAN frame Angular Offsets Calibration Routine* starting on *page 49*.

The kinematic alignment begins when the receiver has a good GNSS position, fine time is solved, the configuration parameters have been set and a GNSS velocity of at least 1.15 (~4 km/h) m/s is observed. During kinematic alignment, keep the vehicle roll at less then 10°. Straight line driving is best.

The accuracy of the initial attitude of the system following the kinematic alignment varies and depends on the dynamics of the vehicle and the accuracy of the RVB estimates. The attitude accuracy will converge to within specifications once some motion is observed by the system. This transition can be observed by monitoring the INS Status field in the INS logs.

3.4.3.3 Manual Alignment

Manually enter the attitude information using the SETINITATTITUDE, or SETINITAZIMUTH, commands. Details of these commands start on *page 140*.

3.4.4 Navigation Mode

Once the alignment routine has successfully completed, SPAN enters navigation mode.

SPAN computes the solution by accumulating velocity and rotation increments from the IMU to generate position, velocity and attitude. SPAN models system errors by using a Kalman filter. The GNSS solution, phase observations and automatic zero velocity updates (ZUPTs) provide updates to the Kalman filter. When a wheel sensor is connected to the system, wheel displacement updates are also used in the filter.

The attitude is coarsely defined from the initial alignment process, especially in heading. Vehicle dynamics, specifically turns, stops and starts, allow the system to observe the heading error and allows the heading accuracy to converge. Three to five changes in heading should be sufficient to resolve the heading accuracy. The INS Status field changes to INS_SOLUTION_GOOD once convergence is complete. If the attitude accuracy decreases, the INS Status field changes to INS_SOLUTION_NOTGOOD. When the accuracy converges again, the INS status continues as INS_SOLUTION_GOOD.

3.4.5 Data Collection

The INS solution is available in the INS-specific logs with either a standard or short header. As shown in *Table 6*:

Table 6: Solution Parameters

Parameter	Log
Position	INSPOS *or* INSPOSS INSPVA *or* INSPVAS
Velocity	INSVEL *or* INSVELS INSSPD *or* INSSPDS INSPVA *or* INSPVAS
Attitude	INSATT *or* INSATTS INSPVA *or* INSPVAS
Solution Uncertainty	INSCOV *or* INSCOVS

Note that the position, velocity and attitude are available together in the INSPVA and INSPVAS logs.

☒ The BESTPOS position log can be logged at rates up to 20 Hz directly from the OEMV port, but is available at 1 Hz or 5 Hz from any SPAN-SE port. Other GNSS logs (RANGE, PSRPOS, and so on) can be logged up to 20 Hz from the SPAN ports. The BESTGPSPOS log is available from SPAN-SE only, at 1 Hz or 5 Hz.

WARNING: Ensure that all windows, other than the Console, are closed in **CDU** and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload your system.

Specific logs need to be collected for post-processing. See *Section 3.6, Data Collection for Post Processing* on *page 52*.

To store data directly to the internal SD Card, see *Section 3.8, The SD Card* starting on *page 54*.

3.4.6 Lever Arm Calibration Routine

Each time the system is re-mounted on a vehicle, or the IMU or antenna is moved on the vehicle, the lever arm must be redefined either through manual measurement or through calibration.

☒ We recommend that you measure the lever arm using survey methodology and equipment, for example, a total station. Only use calibrations when precise measurement of the lever arm is not possible.

The lever arm calibration routine should only be used when the receiver is operating in RTK mode. Initial estimates and uncertainties for the lever arm are entered using the SETIMUTOANTOFFSET command, see *page 137*. The calibration routine uses these values as the starting point for the lever arm computation.

The steps involved in the calibration are:

1. Power the receiver and the IMU, see the IMU choices and their technical specifications starting on *page 62*.
2. Configure the RTK corrections and make sure that the BESTGPSPOS log, see *page 172*, reports a good RTK solution.
3. Configure the IMU, see *Section 3.3.2, SPAN IMU Configuration* starting on *page 41*.
4. Enter the initial estimate for the lever arm using the SETIMUTOANTOFFSET command, see *page 137.*
5. Specify the limits of the calibration through the LEVERARMCALIBRATE command, see *page 111*. The calibration can be limited by time or accuracy of the lever arm. It is recommended that the calibration is limited by a minimum of 300 seconds.
6. Remain stationary long enough for the coarse alignment to finish. The alignment is complete when the INS status changes to INS_ALIGNMENT_COMPLETE, see *Table 5* on *page 44*.

Another indication that the alignment is complete is the availability of INSCOV log on *page 209*.

7. Start to move the system. The lever arm is not observable while the system is stationary. Immediately, drive a series of manoeuvres such as figure eights. The turns should alternate between directions, and you should make an equal number of turns in each direction. Some height variation in the route is also useful for providing observability in the Z-axis. When the calibration is complete, either because the specified time has passed or the accuracy requirement has been met, the BESTLEVERARM log outputs the solved lever arm.

8. Monitor the calibration, log BESTLEVERARM, see *page 178*, using the ONCHANGED trigger.

The lever arm is saved automatically in non-volatile memory. If the IMU or GNSS antenna are re-mounted, the calibration routine should be re-run to compute an accurate lever arm.

3.4.7 Vehicle to SPAN frame Angular Offsets Calibration Routine

Kinematic fast alignment requires that the angular offset between the vehicle and IMU SPAN frame is known approximately. If the angles are simple (that is, a simple rotation about one axis) the values can easily be entered manually through the VEHICLEBODYROTATION command, see *page 149*. If the angular offset is more complex (that is, rotation is about 2 or 3 axis), then the calibration routine provides a more accurate estimation of the values. As with the lever arm calibration, the vehicle to SPAN frame angular offset calibration requires RTK GNSS. The steps for the calibration routine are:

1. Apply power to the receiver and IMU, see the IMU choices and their technical specifications starting on *page 62*.

2. Configure the RTK corrections and make sure that the BESTGPSPOS log, see *page 172*, reports a good RTK solution.

3. Configure the IMU, see *Section 3.3.2, SPAN IMU Configuration* starting on *page 41*.

4. Ensure that an accurate lever arm has been entered into the system either manually or through a lever arm calibration, see *page 48*.

5. Allow the system to complete a coarse alignment, see *page 46.* Remain stationary long enough for the coarse alignment to finish. The alignment is complete when the INS status changes to INS_ALIGNMENT_COMPLETE, see *Table 5* on *page 44*. Another indication that the alignment is complete is the availability of INSCOV log on *page 209.*

6. Enable the vehicle to body calibration using the RVBCALIBRATE ENABLE command, see *page 128*.

7. Start to move the system. As with the lever arm calibration, movement of the system is required for the observation of the angular offsets.

8. Drive a series of manoeuvres such as figure eights if the driving surface is not level, or a straight course if on level ground (remember that most roads have a crown, resulting in a constant roll of a few degrees). Avoid driving on a surface with a constant, non-zero, slope to prevent biases in the computed angles. Vehicle speed must be greater than 5 m/s (18 km/hr) for the calibration to complete.

9. When the uncertainties of the offsets are low enough to be used for a fast alignment, the calibration stops and the VEHICLEBODYROTATION log, see *page 269*, is overwritten with the solved values. To monitor the progress of the calibration, log VEHICLEBODYROTATION using the ONCHANGED trigger.

The rotation parameters are saved in NVM for use on start-up in case a kinematic alignment is required. Each time the IMU is re-mounted this calibration should be performed again. See also *Sections 3.4.3.1* and *3.4.3.2* starting on *page 46* for details on coarse and kinematic alignment.

WARNING: **After the RVBCALIBRATE ENABLE command is entered, there are no vehicle-body rotation parameters present and a kinematic alignment is NOT possible. Therefore this command should only be entered after the system has performed either a static or kinematic alignment and has a valid INS solution.**

☒ The solved rotation values are used only for a rough estimate of the angular offsets between the IMU and vehicle frames. The offsets are used when aligning the system while in motion (see *Section 3.4.1, Configuration for Alignment* starting on *page 45*). The angular offset values are not applied to the attitude output, unless the APPLYVEHICLEBODYROTATION command is enabled, see *page 91*.

3.5 SPAN Wheel Sensor Configuration

The SPAN-SE receiver supports various wheel sensor inputs.

A wheel sensor can be used to measure the distance travelled by counting the number of revolutions of a ground vehicle wheel. Typical wheel sensor hardware outputs a variable frequency pulse that varies linearly with speed. If the pulses are accumulated and the size of the wheel known, a displacement of the wheel over time can be calculated. SPAN-SE takes in a wheel sensor input and applies a displacement update to the GNSS/INS Kalman filter in order to constrain the position error growth during GNSS outages. SPAN also automatically calculates the exact size of the wheel to mitigate small changes in the size of the wheel due to hardware changes or environmental conditions. Information on how the wheel sensor updates are being used is available in the INSUPDATE log, see *page 219*.

Wheel sensor information can be input into the system using one of three separate methods:

1. Directly connecting the wheel sensor to one of the event input lines available on the SPAN-SE
2. Using the wheel sensor interface on the iIMU-FSAS IMU
3. Entering the WHEELVELOCITY commands, see *page 157*, through the user interface

Specific details on the three methods of wheel sensor input are described below.

3.5.1 Wheel Sensor Updates Using the Event Input Lines

The event input lines in SPAN-SE can be configured to accept a wheel sensor signal directly. Any of the four available event input lines can be used, but only one can be used at a time – the system does not support multiple wheel sensors. This method currently only supports A mode (directionless) and not A/B (directional) mode of operation for the wheel sensor. The receiver automatically accumulates the wheel sensor pulses, calculates a distance travelled and applies the constraint information in the SPAN GNSS/INS filter.

To connect your wheel sensor to the SPAN-SE event input line, connect Signal A from the wheel sensor to one of the event input lines available on the I/O 2 yellow cable (see *I/O 2 Yellow Cable* on *page 68*).

The event input line must be configured for wheel sensor input and the size of the wheel, and the number of ticks per revolution must be set using the SETWHEELPARAMETERS command. For example if you have your wheel sensor connected to event input 2 with a 2 m circumference wheel and 2000 pulses per revolution, the configuration command would be:

```
SETWHEELPARAMETERS MARK2 POSITIVE 2000 2.0 0.001
```

3.5.2 Wheel Sensor Updates using the iIMU-FSAS IMU

The FSAS IMU also has a wheel sensor input that can be directly attached to the output of the wheel sensor.

You can use iMAR's iMWS or another wheel sensor that meets the iIMU-FSAS requirements (see http://www.imar-navigation.de/englishside/imar.htm for details). An optical-encoder style wheel sensor such as the Corrsys Datron wheel pulse transducer can also be used.

Information about cabling requirements for wheel sensor input to the FSAS wheel sensor interface is available in *iIMU-FSAS Odometer Cabling* on *page 80.*

The size of the wheel and the number of ticks per revolution must also be set using the SETWHEELPARAMETERS command. For example a 2 m circumference wheel with 2000 pulses per revolution would be configured using the following command:

```
SETWHEELPARAMETERS 2000 2.0 0.001
```

3.5.3 Wheel Sensor Updates using the WHEELVELOCITY Command

If you have wheel sensor hardware that accumulates the pulses from a wheel sensor, you can send the accumulated tick-count to the SPAN-SE at 1 Hz using the WHEELVELOCITY command (see *page 157)*. The command can be sent in ASCII or binary format. The tick count in the WHEELVELOCITY command should reference the number of ticks accumulated at the time of the GNSS second boundary. For reference, the GNSS second boundary is available from the event output lines on SPAN-SE. See also the EVENTOUTCONTROL command on *page 103*.

3.5.4 Logging Wheel Sensor Data from SPAN-SE

The accumulated wheel sensor counts are available by logging the TIMEDWHEELDATA log with the ONNEW trigger:

```
LOG TIMEDWHEELDATAB ONNEW
```

If you wish to use the wheel sensor data in post-processing then ensure that the TIMEDWHEELDATAB log is included in your logging profile.

The computed wheel size is available through the WHEELSIZE log with the ONNEW trigger:

```
LOG WHEELSIZEB ONNEW
```

3.6 Data Collection for Post Processing

Some operations such as aerial measurement systems do not require real-time information from SPAN. These operations are able to generate the position, velocity or attitude solution post-mission in order to generate a more robust and accurate solution than is possible in real-time.

In order to generate a solution in post-processing, data must be simultaneously collected at a base station and each rover. The following logs must be collected in order to successfully post process data:

From a base if not using GLONASS:

- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW

From a base if using GLONASS:

- RANGECMPB ONTIME 1
- GLORAWEPHEMB ONNEW
- GLOEPHEMERISB ONCHANGED

From a rover if not using GLONASS:

- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW
- RAWIMUSB ONNEW
- BESTLEVERARMB ONNEW

From a rover if using GLONASS:

- RANGECMPB ONTIME 1
- GLORAWEPHEMB ONNEW
- GLOEPHEMERISB ONCHANGED
- RAWIMUSB ONNEW
- BESTLEVERARMB ONNEW

Post processing is performed through the Waypoint Inertial Explorer software package available from from NovAtel's Waypoint Products Group. Visit our Web site at www.novatel.com for details.

3.7 Status Indicators

LED indicators on the front of the SPAN-SE, see *Figure 11* below, provide the status of the receiver. *Table 7* details the LED states, which are solid unless otherwise indicated as blinking. They represent these categories: Power, SD Card, OEMV-2 Card (which is not included in every SPAN-SE system), OEMV-3 Card, IMU (which indicates the status of the raw data received from the IMU) and INS (which indicates the status of the GNSS/INS solution computed by the SPAN-SE).

Figure 11: SPAN-SE LED Indicators

Table 7: Positioning Mode LEDs

Label	LED	Off	Green	Orange	Green & Orange Flashing	Red
SD Card	SD Card	No card	Card in (Flashing: file open)	Card in but low on space with <10% space remaining (Flashing: same as above but a file is open)	Card busy (either formatting or mounting)	Card in but has <1% space remaining and logging stops automatically (Flashing: SD card error that can occur at any time regardless of remaining space)
GPS 1	Primary GNSS	No Data	Solution complete and fine steering (Flashing: coarse steering)	Insufficient Observations	N/A	Receiver status error (bits 0: SDRAM 1: Firmware 2: ROM 7: Supply Voltage)
GPS 2	Secondary GNSS	No Data	*(same as Primary GNSS above)*			
INS	INS	GNSS only	INS_SOLUTION_GOOD status (Flashing: INS_ALIGNMENT_COMPLETE status)	Aligning (Flashing: INS_SOLUTION_NOTGOOD status)	N/A	INS_INACTIVE status
IMU	IMU	No IMU detected	RAWIMU packets with good IMU status, as reported by the IMU	No RAWIMU (IMU type not set)	N/A	IMU status error bits
N/A	Power [1]	No power to unit	Unit powered but off and not operational (Flashing: unit powered, on, and operational)	N/A	N/A	N/A

1. The power LED is on the power button, see *SPAN-SW Power Button* on *page 33*.

3.8 The SD Card

Data commands and logs can be recorded from the SPAN-SE to a removable SD Card. The need for a companion handheld data logger is avoided and continuous user interaction is not required, since the SPAN-SE is capable of logging data according to pre-configured parameters without any user intervention.

WARNING: To minimize the possibility of damage, always keep the SD Card cover closed, except when exchanging SD Cards. **Do not change the card while logging is in progress. Data will be lost.** It is not necessary to turn the receiver off before inserting or extracting a SD Card, but the logging session should be closed by pressing the SD Card logging button or issuing the `LOGFILE CLOSE` command before removing the card.

An example of an SD Card is shown in *Figure 5, SD Memory Card on page 29*.

The SD Card access door is shown closed in *Figure 12* below. Move the arrow latch to the left to open the access door. To remove the SD Card, unlock the access door. When the door is open, you can see the card. Push it slightly to partially eject it. Then grasp the card and pull it all the way out.

Figure 12: SD Card Access Door

To insert the card, ensure that it is correctly aligned before gently sliding it into the slot. If you attempt to insert the card incorrectly, it will not go all the way in. In this case, do not force the card! Remove it, orient it properly, and then insert it. After the card is locked in place, close the cover by moving the arrow latch to the right until it clicks in place.

3.9 Logging Data to the SD Card

3.9.1 Insert the SD Card

1. Insert the card into the SD Card slot.
2. Wait for the SD LED to turn solid green.

☒ Large memory-sized cards may take a few minutes to mount. During this time the LED flashes green and orange. Also, if you request a DIR command, the receiver generates an `<ERROR: DISK BUSY` response.

3.9.2 Prepare the Card

To prepare the SD Card in the SPAN-SE for data logging:

1. Connect to the receiver through the serial, USB or Ethernet ports.
2. If necessary, format the card using the command `FORMAT SD`.
   During the format process, the SD LED flashes alternating green and orange.
   The LED turns solid green when formatting is complete.

☒ During the format process, if you request a `DIR` command, the receiver generates a `<ERROR: Disk Busy` response.

WARNING: **Formatting the card deletes any data that is on the SD Card. Ensure that all data is copied to another location before formatting.**

At this stage, if you only need data for post-processing, the logging button (located to the right of the card behind the access door) can be pressed to start logging of a pre-defined list of logs required for post-processing applications to an automatically named file in the root directory of the SD Card, see *Section 3.9.5, Log a Pre-Defined List of Logs* on *page 56*. Otherwise, continue to Step #3.

3. Select the location on the disk to store your data. The default location is in the root directory, but you can modify the directory structure using the following commands:

   a. To view the current working directory, enter the PWD command:

   ```
   [COM1] pwd SD
   \
   [COM1]
   ```
   (Now in the root directory)

   b. To make a directory, enter the MKDIR command:

   ```
   MKDIR SD TEST1
   ```
   (Create a directory called test1 under the root)

   c. The DIRENT log lets you view the contents of the current directory, which now contains a TEST1 directory.

   ```
   [COM1]LOG DIRENT
   <OK
   [COM1]<DIRENT COM1 0 99.0 FINESTEERING 1523
   153428.656 00000000 0000 159
   <     "TEST1" 0 0 20090316 183648
   ```

   The `DIR` command can also be used at the command prompt to return a Disk Operating System (DOS) directory structure response.

   d. To change the directory, enter the CD command:

   ```
   CD SD TEST1
   ```
   (Change current working directory to new TEST1)

   e. To view the current working directory, enter the `PWD` command:

   ```
   [COM1] PWD SD
   ```

```
\TEST1
[COM1]
```

f. To remove a directory, use the RMDIR command

```
CD \                    (Change back to the root directory)
RMDIR SD TEST1          (Remove the TEST1 directory)
```

3.9.3 Select Logs to Send to the SD Card

Use the LOG command, see *page 113*, and its *FILE* designator, to specify which logs to send to the SD Card. For example, a standard logging configuration for GNSS-only post-processing applications would be:

```
LOG FILE RANGECMPB ONTIME 1
LOG FILE RAWEPHEMB ONNEW
LOG FILE RAWIMUSB ONNEW
LOG FILE BESTLEVERARMB ONNEW
```

3.9.4 Start and Stop Logging

To start or stop logging, either use the button next to the SD Card access door, or use the LOGFILE command, see *page 118*.

Once a list of logs has been specified for logging, press the Log button, on the SPAN-SE, once, to start the logging into an auto-named logging file in the current working directory. Press the button a second time to stop the logging and close the file. Press the button a third time to re-start the logging to a new file, and so on.

The LOGFILE command lets you start and stop logging and specify the file name to use. If no file name is entered in the command, a new auto-generated file name is created every time you open a file to write to it.

For example:

`LOGFILE OPEN SD FIRSTFILE.GPS` *(Open a file in the current working directory called `FIRSTFILE.GPS` and start logging)*

If the file name entered already exists on the card, the command returns an error.

When logging is enabled the SD LED flashes green.
When logging is stopped, the SD LED is solid green.
When the card has 10% capacity remaining, the SD LED turns orange.
When the card has less than 1% capacity remaining the SD LED turns red.

3.9.5 Log a Pre-Defined List of Logs

To log a pre-defined list of logs needed for post-processing, follow these steps:

- Insert the SD Card
- Prepare the SD Card by letting it complete its mounting, or format the card if necessary. When the

card is ready for logging, the SD LED turns solid green if the card is empty, or orange if the card has < 10% of free space remaining.

- Press the SD Logging button, located behind the SD Card access door, to open a new file and start logging.

  The SD LED starts blinking, green if the card is empty; orange if the card has < 10% of free space remaining, when the file is opened.

  The list of pre-defined logs include the following:

  - RAWIMUSB ONNEW
  - BESTGPSPOSB ONTIME 1
  - RANGECMPB ONTIME 1
  - RAWEPHEMB ONNEW
  - GLORAWEPHEMB ONNEW

- Press the SD Logging button to stop logging, or use the LOGFILE CLOSE command, see *page 118*, to close the file. Note that this is **not** an UNLOGALL command and if you open a file again, the profile will continue to log.

  Also, you must set the SETIMUTYPE command, see *page 138*, before the receiver logs RAWIMUSB data.

3.9.6 Auto-Logging on Start-Up

After configuring log output using the `LOG` commands, configure the receiver to log the log profile on start-up every time by issuing these two commands:

```
SETAUTOLOGGING ON
SAVECONFIG
```

Every time the receiver powers-up, the SD Card logging configuration you specified starts. See also the SAVECONFIG command on *page 129* and the SETAUTOLOGGING command on *page 132*.

3.9.7 Reading data from the card

You can read data from the SD Card in multiple ways after you stop logging:

1. Remove the card from the receiver and read the data using a PC SD Card reader.
2. Use the File Transfer Protocol (FTP) functionality built into the SPAN-SE:
   - The FTP functionality is available over the Ethernet port on the receiver.
   - The Internet Protocol (IP) address, default mask and gateway settings for the receiver can be set using the IFCONFIG command.
   - Only use FTP on a secure connection as this port has no security settings at this time.

✉ FTP functionality is only available if the receiver is not writing files to the SD Card.

---

3. Use the NovAtel Explorer inside **CDU** to download the files from the SD Card over any of the SPAN-SE ports connected to a PC. While all ports are supported, for the fastest transfer, use the USB connection.

D CARD IMPORTANT INFORMATION

Do not remove the SD Card while data logging to the card is in progress! This may result in damage to the card and loss of data. Stop the logging using the LOG button, or the LOGFILE command, before removing the SD Card.

Do not unplug power to the receiver while data logging to the card is in progress! Stop the logging before removing power, or use the power button to power down.

3.10 Synchronizing External Equipment

The SPAN-SE allows you to synchronize with external equipment in two ways:

1. The receiver has three configurable output strobes. Each strobe is synchronous with GPS time and can be configured for pulse length and polarity.
2. The receiver accepts up to four input pulses (events). Each event signal can be configured for positive or negative polarity. Time, or a solution (position, velocity, attitude), can be generated and output synchronously with each input pulse.

3.10.1 Configuring a Synchronous Output Pulse

The EVENTOUTCONTROL command, see *page 103*, is used to configure an output strobe. There are three output strobe lines in the receiver called MARK1, MARK2 and MARK3 and each of them can be configured independently. The event strobes toggle between 3.3 V and 0 V. Each strobe can supply 24 mA.

The pulse consists of two states: an active state and a not-active state. The start of the active state is synchronized with the top of the GPS time second and the polarity of the signal indicates whether the active period is 3.3 V or 0 V. The not-active period immediately follows the active period and has the alternate voltage.

Each output strobe can be configured in the following ways:

Polarity:	The polarity defines the signal state of the active portion of the signal. A positive polarity dictates that the active portion of the signal is in a high state (3.3 V).
Active Period Width:	The active period starts at the GPS time synchronized edge (rising for negative polarity and falling for positive polarity). The time length of this period is specified in nanoseconds (ns).
Not-Active Period Width:	The not-active period immediately follows the active period. The width of this period is specified in ns.

Rules Governing Period Widths:

- The minimum period is 1000 ns. The maximum period is 999 999 000 ns.
- Periods must be entered as a multiple of 25 ns, that is 1000, 1025, 1050, 1075 and so on.
- The sum of the active and not-active periods must be a factor of 1 s. That is:
  $*$ K (active + not-active) = 1 000 000 000, where K = 1, 2, 3, ..., 500 000

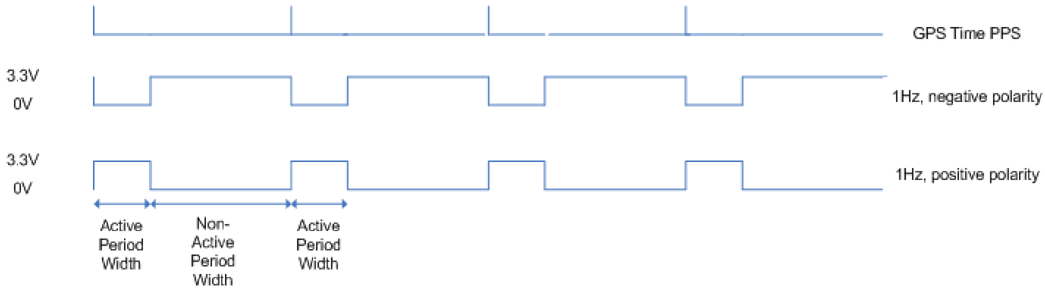


Figure 13: Event Out

3.10.2 Configuring an Input Strobe

SPAN-SE has four available input strobes. The input strobes apply an accurate GPS time to the rising, or falling, edge of an input pulse called an event. For each event, an accurate position, velocity or attitude solution is also available. Each input strobe is usually associated with a separate device, therefore different solution output lever arm offsets can be applied to each strobe.

Each input strobe can be configured using the EVENTINCONTROL command, see *page 102*, for the following parameters:

1. Polarity:	When polarity is set to positive, events trigger on the rising edge. When polarity is set to negative, events trigger on the falling edge.
2. Time Bias:	A constant time bias in ns can be applied to each event pulse. Typically this is used to account for a transmission delay.
3. Time Guard:	The time guard specifies the minimum number of milliseconds between pulses. This is used to coarsely filter the input pulses.

The time of the input pulses is available from the MARK*x*TIME logs, see *page 237*. The solution synchronous with the event pulses is available from the MARK*x*PVA logs, see *page 236*. The logs required for input strobes are:

`LOG MARK1TIMEB ONNEW`	Output time for every pulse received.
`LOG MARK1PVAB ONNEW`	Output time, position, velocity and attitude for every pulse received at the location specified by the SETMARK1OFFSET command.

The input signal levels are 3.75 V to -0.3 V. Signal voltages outside these bounds damage the receiver. The minimum detectable pulse duration must be greater than or equal to 1 microsecond.

3.10.2.1 Using the Input Strobe to Accumulate Counts

You can also use an input strobe line to count the number of pulses over one second and report the total at the top of each second by setting the input event line to `COUNT` mode.

```
EVENTINCONTROL MARK1 COUNT
```

When in count mode, the polarity, time bias and time guard entries in the EVENTINCONTROL log are ignored. The maximum signal frequency for the count mode is 50 kHz.

When an input strobe is configured for `COUNT` mode, the totals are available by logging the MARK*x*COUNT logs, see *page 235*. For example, the following gives the total pulses on event strobe 1 every second:

```
LOG MARK1COUNTA ONNEW
```

3.11 SPAN-SE Ethernet Connection

The SPAN receiver has one 10/100 RJ-45 Ethernet port. The device has a Media Access Control (MAC) address, hard coded into flash, and user-configurable IP information. There is one port available for Ethernet, Port 3000 can be used for both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic, but not simultaneously.

SPAN-SE uses a static IP address. Dynamic Host Configuration Protocol (DHCP) is a protocol for automating the configuration of computers that use TCP/IP. There is no DHCP support at this time.

An FTP port is available for transfer of data files from the data logging SD Card.

The receiver is shipped with a default configuration as follows:

- Default IP: 192.168.0.10
- Default mask: 255.255.255.0
- Default Gateway: 192.168.0.1

A unique MAC address is programmed into the receiver before it is shipped. The MAC address is available to the user through the MAC log.

3.11.1 Configuring for TCP or UDP Operation

The SPAN-SE Ethernet connection can be configured for either TCP or UDP. The default configuration of the Ethernet port is for TCP operation.

The SETETHPROTOCOL command can be used to change the mode, but note that the command must be followed by a receiver reset through the RESET command or cycling the power. See *page 133* and *page 123* respectively.

☒ The Ethernet protocol setting is permanent. The receiver will stay configured as either TCP or UDP until the SETETHPROTOCOL and RESET commands are entered to change the setting again.

3.11.2 Configuring the Ethernet Connection Settings

Use the IFCONFIG command, see *page 108*, to set the static IP Address, the subnet mask and the gateway. An example of the IFCONFIG command is:

```
IFCONFIG 10.1.100.25 255.255.255.0 10.1.100.1
```

3.11.3 Configuring Log Requests Destined for the Ethernet Port.

The COM port identifier for the Ethernet port is ETH1 in ASCII or 20 in binary. A sample log request for the Ethernet port is:

```
LOG ETH1 RANGECMPB ONTIME 1
```

3.11.4 Connecting to the Ethernet Port

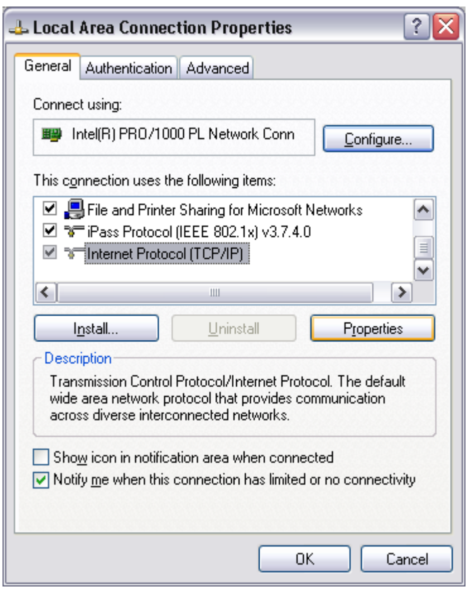
If the port is configured in TCP mode, only one connection to the receiver is allowed at a time. Data automatically streams to the IP address that connects to the port.

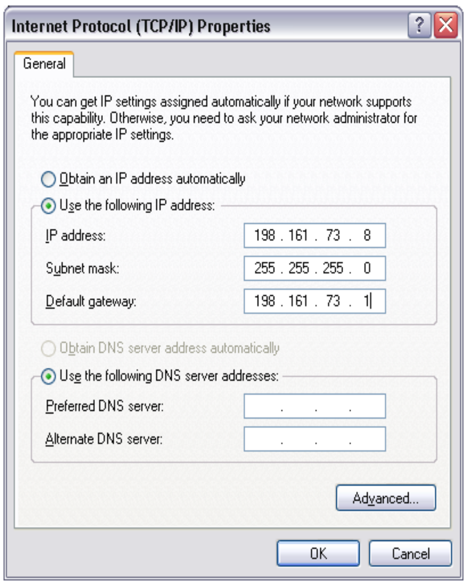
Because UDP is a connectionless protocol, multiple end-points could communicate with the port at one time from multiple IP addresses. Data streams to the last IP Address to communicate with the receiver.

For details on the FTP functionality of the Ethernet port, see the FTP DOS command on *page 90*.

To connect the SPAN-SE directly to your PC Ethernet port (not through a network), follow these steps:

1. Connect you PC Ethernet port to the SPAN-SE Ethernet port using an Ethernet cross-over cable.
2. Set the static IP address on your PC to the following settings in the Local Area Connection Properties dialog box:





Appendix A Technical Specifications

This appendix details the technical specifications of the IMUs and the SPAN-SE receiver. Refer to your SPAN system enclosure (for example, ProPak-V3) manual (*OEMV Family Installation and Operation User Manual)* for more information on its technical specifications, performance and cables.

A.1 SPAN-SE

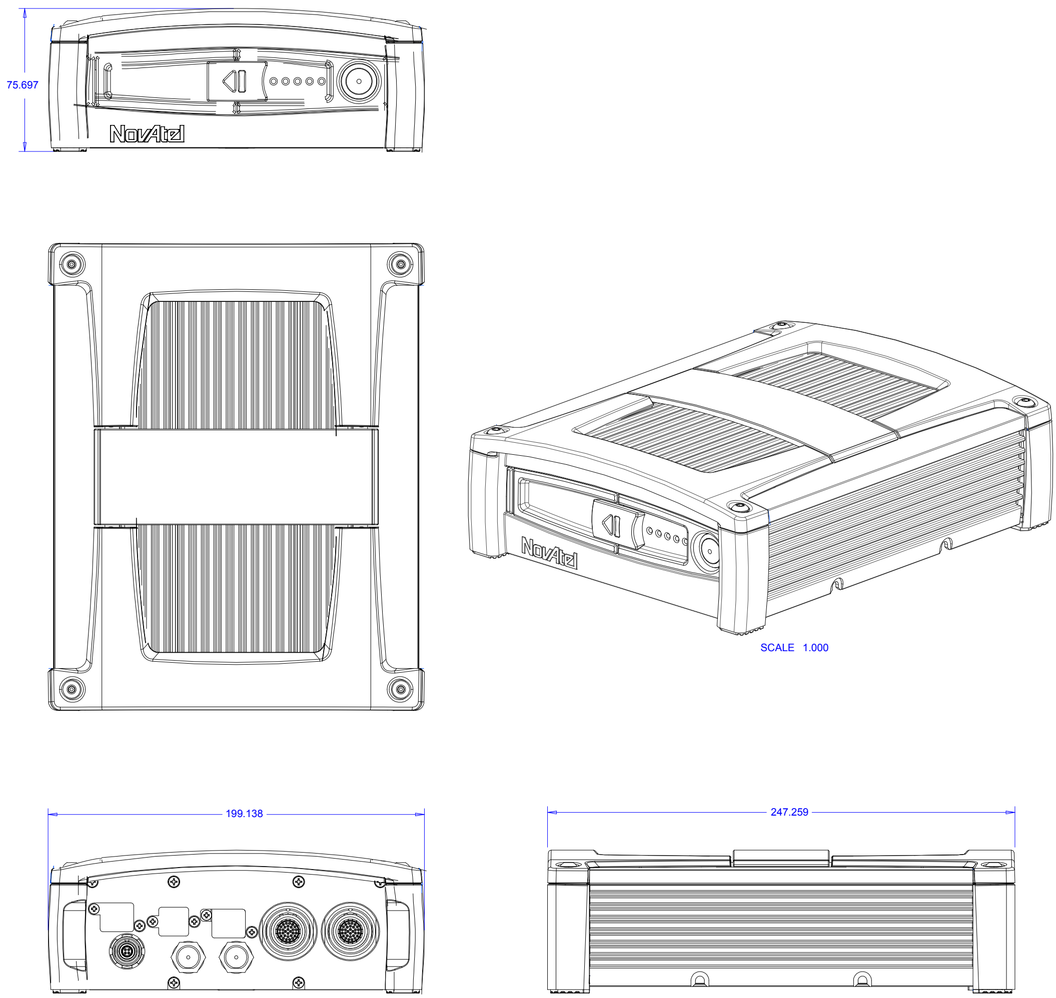
SPAN-SE is a SPAN-capable receiver. The SPAN-SE receiver's technical specifications follow. For the other OEMV-based and SPAN-capable receivers' details, refer to the *OEMV Family Installation and Operation User Manual*.

A.1.1 SPAN-SE Receiver

INPUT/OUTPUT CONNECTORS	
Antenna Input 1 and 2	TNC female jack, 50 Ω nominal impedance +5 V DC, 100 mA max (output from SPAN-SE to antenna/LNA)
Power	ODU Mini Snap, Series K, 4-pin connector +9 to +28 V DC Power Consumption Single Antenna: 10 W (typical) Dual Antenna: 12 W (typical)
USB Host USB Device Ethernet I/O 1 (Green) I/O 2 (Yellow)	USB-A USB-B RJ-45 Ethernet ODU Mini Snap, Series K, 30-pin connector, see *Table 8* on *page 67* ODU Mini Snap, Series K, 30-pin connector, see *Table 9* on *page 69*
NOVATEL PART NUMBER	
SPAN-SE	01018071
PHYSICAL	
Size	247 x 199 x 76 mm
Weight	3.4 kg maximum
ENVIRONMENTAL	
Operating	-40°C to +65°C
Storage Temperature	-50°C to +95°C
Humidity	Not to exceed 95% non-condensing

ENVIRONMENTAL		
Tested to these standards:	IEC 60529 IPX7	Waterproof
	IEC 60529 IPX6	Dust
	IEC 68-2-27, 60 g	Shock (non-operating)
	RTCA D0-160D, curve C	Vibration (random)
	IEC 68-2-6	Vibration (sinusoidal)
	FCC Part 15, Class B	Emissions
	EN 55022, Class B	Emissions
	EN 55024	Immunity
	EN 60950-1	Safety

DIMENSIONS [a]



a. All dimensions are in millimeters, please use the *Unit Conversion* section of the *GNSS Reference Book*, available from our Web site at www.novatel.com for conversion to imperial measurements.

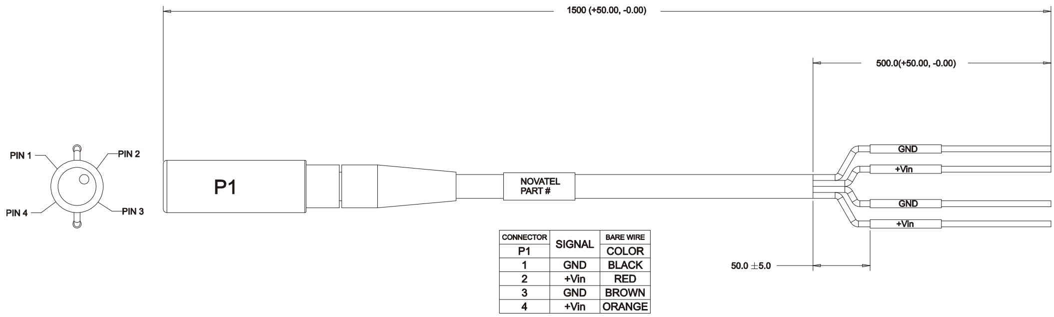
A.1.1.1 Power Adapter Cable (NovAtel part number 01018135)

The power adapter cable supplied with the SPAN-SE, see *Figure 14*, provides a means for supplying +9 to +28 V DC while operating in the field.

Input is provided through the bare wire power outlets. The exposed wires (red and orange for positive, brown and black for negative) can then be tied to a supply capable of at least 10 W for a single antenna or 12 W for dual antennas.

This cable is RoHS compliant.

For alternate power sources, see *Section 2.2.4* on *page 32*.



CONNECTOR P1	SIGNAL	BARE WIRE COLOR
1	GND	BLACK
2	+Vin	RED
3	GND	BROWN
4	+Vin	ORANGE

Reference	Description
P1	ODU 4-pin
+Vin:	+9 to +28 V DC

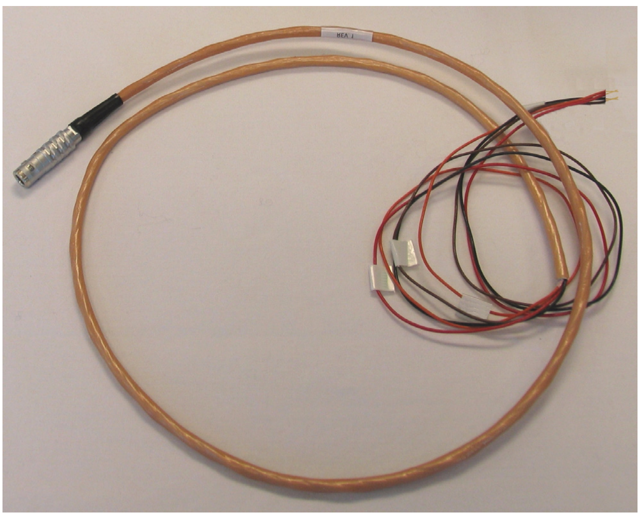
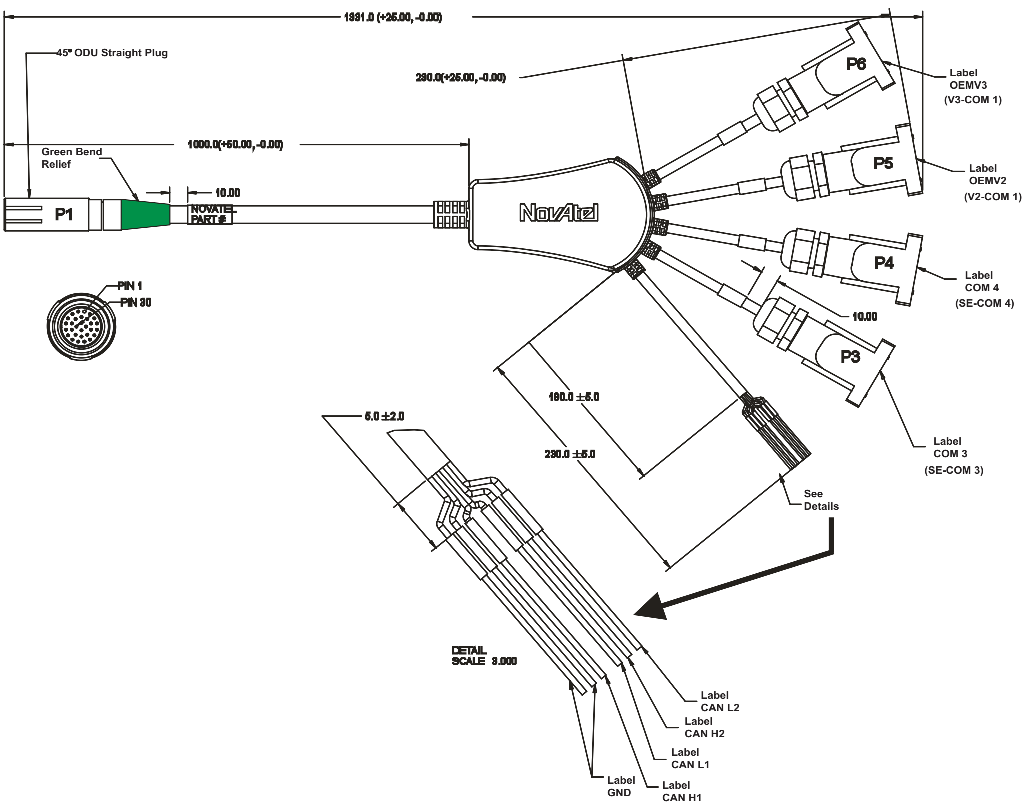


Figure 14: SPAN-SE Power Cable

A.1.1.2 I/O 1 Green Cable (NovAtel part number 01018134)

This cable, supplied with the SPAN-SE, see *Figure 15*, provides a means of connecting with communications and I/O devices. The cable is equipped with a 30-pin connector at the receiver end plus four DB-9 connectors at the other end, one for each serial port. The serial ports available on this cable are COM3, COM4 (both used for command input and data output), a direct connection to the internal OEMV-3 COM1 (used for RTK correction input) and a direct connection to the internal OEMV-2 COM1. There is also an end with six bare cables for CAN configurations. See *Table 8, I/O 1 Green Cable Connector Pin-Outs* on *page 67*.

This cable is RoHS compliant.



Reference	Description
P1	ODU 30-pin
P3 to P6	DB-9

Figure 15: SPAN-SE I/O 1 Green Cable

Table 8: I/O 1 Green Cable Connector Pin-Outs

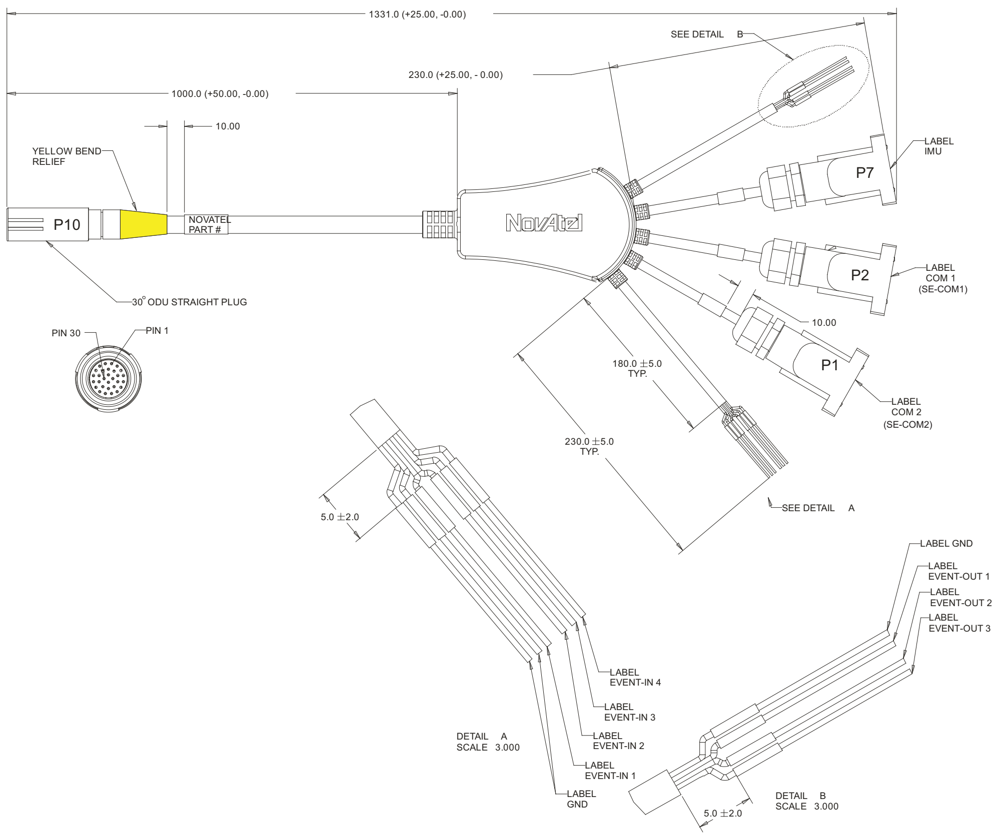
P1 [a]		Remote Connectors	
Pin #	Function	Connector	Pin # [a]
12	CAN H1	Bare Wire (BLACK)	
11	CAN L1	Bare Wire (BLUE)	
22	CAN H2	Bare Wire (RED)	
10	CAN L2	Bare Wire (BROWN)	
29	GND	Bare Wire (GREEN)	
30	GND	Bare Wire (WHITE)	
17	RXD3	(COM3)	3
16	TXD3	(COM3)	2
23	GND	(COM3)	5
15	RTS3	(COM3)	8
1	CTS3	(COM3)	7
25	RXD4	(COM4)	3
13	TXD4	(COM4)	2
9	VDC OUT	(COM4)	4
26	GND	(COM4)	5
24	RTS4	(COM4)	8
14	CTS4	(COM4)	7
20	RXD_V2	OEMV2	3
21	TXD_V2	OEMV2	2
27	GND	OEMV2	5
7	RTS_V2	OEMV2	8
6	CTS_V2	OEMV2	7
4	RXD_V3	OEMV3	3
5	TXD_V3	OEMV3	2
8	VDC OUT	OEMV3	4
28	GND	OEMV3	5
19	RTS_V3	OEMV3	8
3	CTS_V3	OEMV3	7

a. Refer to connector numbers, P1 through P6 in *Figure 15* on *page 66*

A.1.1.3 I/O 2 Yellow Cable (NovAtel part number 01018133)

This cable, supplied with the SPAN-SE, see *Figure 16*, provides a means of connecting with communications and I/O devices. The cable is equipped with a 30-pin connector at the receiver end plus three DB-9 connectors at the other end, each connected to a serial port. On this cable, serial ports COM1, COM2 and the IMU port are available. There are also two ends with bare cables as shown in the figure below. See *Table 9, I/O 2 Yellow Cable Connector Pin-Outs* on *page 69*.

This cable is RoHS compliant.



Reference	Description
P10	ODU 30-pin
P1, P2 & P7	DB-9

Figure 16: SPAN-SE I/O 2 Yellow Cable

Table 9: I/O 2 Yellow Cable Connector Pin-Outs

P1 [a]		Remote Connectors	
Pin #	Function	Connector	Pin # [a]
10	EVENT-OUT 1	Detail B Bare Wire (BLACK)	
23	EVENT-OUT 2	Detail B Bare Wire (BLUE)	
11	EVENT-OUT 3	Detail B Bare Wire (RED)	
27	GND	Detail B Bare Wire (GREEN)	
6	EVENT-IN 1	Detail A Bare Wire (BLACK)	
5	EVENT-IN 2	Detail A Bare Wire (BLUE)	
20	EVENT-IN 3	Detail A Bare Wire (RED)	
19	EVENT-IN 4	Detail A Bare Wire (BROWN)	
28	GND	Detail A Bare Wire (GREEN)	
29	GND	Detail A Bare Wire (WHITE)	
4	TXD2	COM2	2
2	RXD2	COM2	3
9	VDC OUT	COM2	4
30	GND	COM2	5
18	CTS2	COM2	7
3	RTS2	COM2	8
16	RXD1	COM1	3
15	TXD1	COM1	2
8	VDC OUT	COM1	4
17	GND	COM1	5
14	RTS1	COM1	8
1	CTS1	COM1	7
22	EVENT-OUT 4	IMU	1
25	RXD_IMU	IMU	2
12	TXD_IMU	IMU	3
26	GND	IMU	5
24	RTS_IMU	IMU	7
13	CTS_IMU	IMU	8

a. Refer to connectors P1, P2, P7 and P10, and to the bare wires in Detail A and Detail B, in *Figure 16* on *page 68*

A.2 Inertial Measurement Units (IMUs)

A.2.1 LN-200 IMU

Table 10: LN-200 IMU Specifications

PHYSICAL	
IMU Enclosure Size	135 mm x 153 mm x 130 mm (5.315” x 6.024” x 5.118”)
IMU Size	89 mm D x 85 mm H (3.504” D x 3.346” H)
IMU Weight	~3 kg (6.6 lb.)
MECHANICAL DRAWINGS	

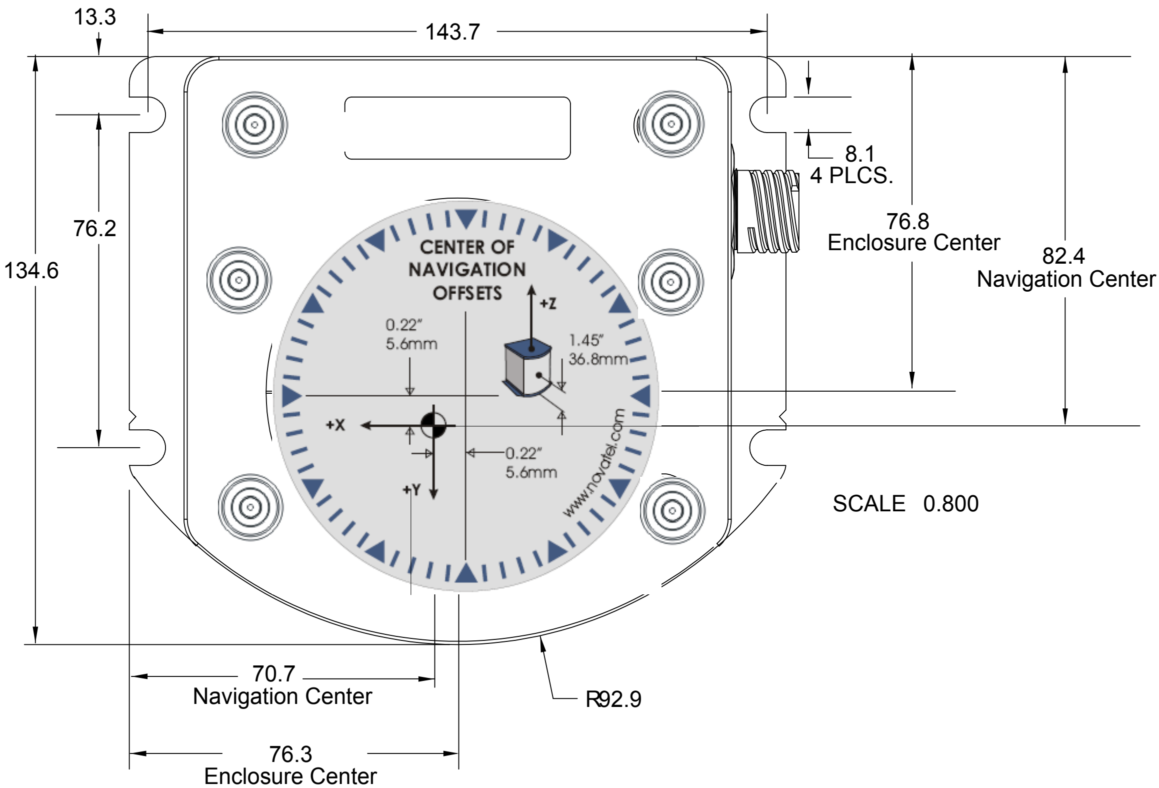


Figure 17: LN-200 IMU Enclosure Top/Bottom Dimensions and Centre of Navigation

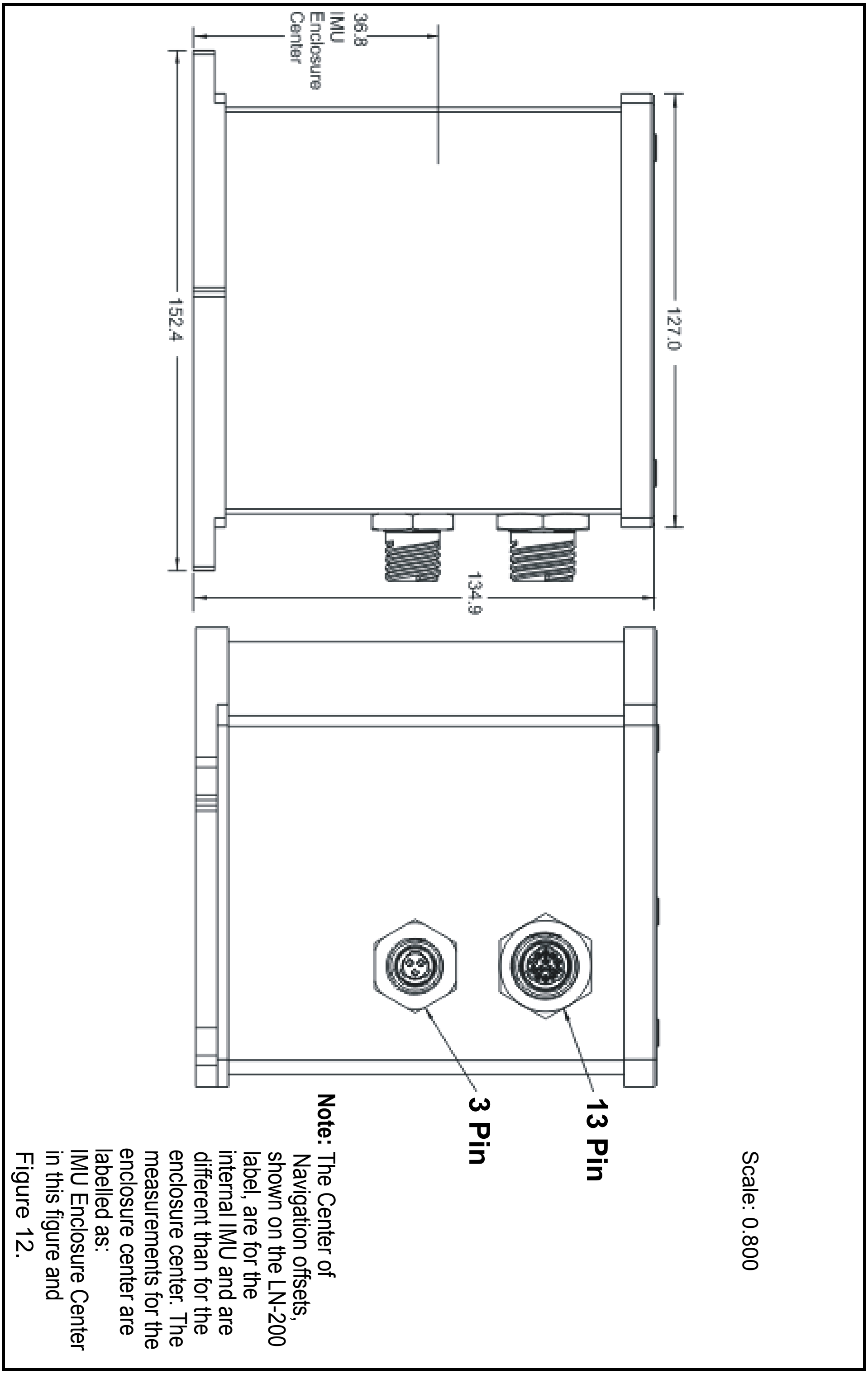


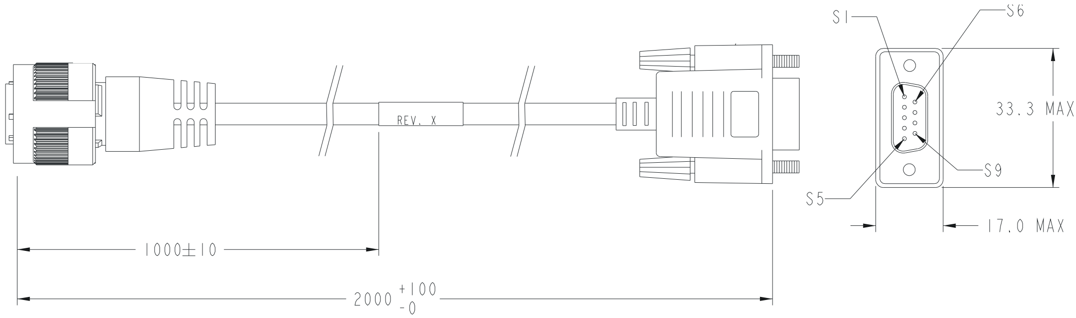
Figure 18: LN-200 Enclosure Side Dimensions

A.2.1.1 LN-200 IMU Interface Cable

NovAtel's part number for the LN-200 IMU interface cable is 01017375 (*Figures 19* and *20 below*).

The IMU interface cable supplied enables input and output between the IMU and the receiver.

Figure 19: LN-200 Interface Cable



Deutsch 13-Pin to IMU		DB-9 Female to Receiver
S1		N/C
S2	PAIRED	S3
S3		S7
S4		N/C
S5		S5
S6		N/C
S7	PAIRED	S8
S8		S2
S9	2 WIRES	S1
S9		S6
S10		N/C
S11	PAIRED	N/C
S12		N/C
S13		N/C

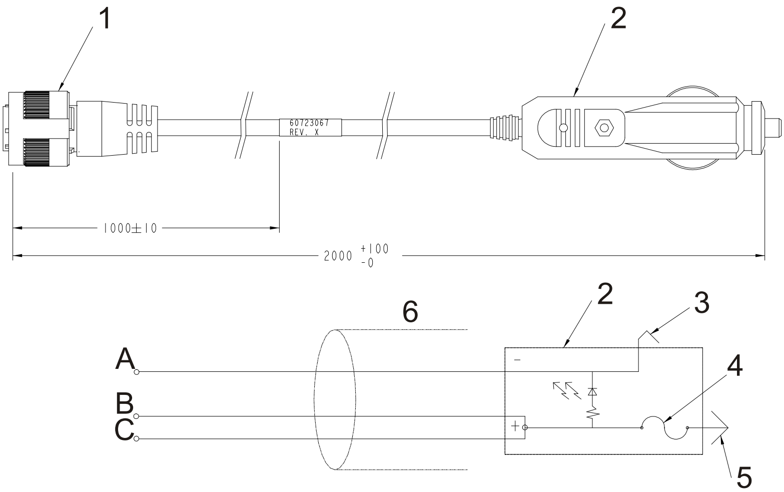
Figure 20: IMU Interface Cable Pin-Out (ProPak-V3)

A.2.1.2 LN-200 IMU Power Adapter Cable

The power adapter cable, NovAtel part number 01017821, supplied with the LN-200 provides a convenient means for supplying +12 VDC while operating from a 12V source. *Figure 21* shows the cable and *Figure 22* the wiring diagram of the 12V adapter.

The output of the power adapter uses a 3-pin Deutsch socket (Deutsch part number: 59064-09-98SN). This cable plugs directly into the 3-pin port on the front of the LN-200 enclosure.

Figure 21: LN-200 Power Cable



Reference	Description	Reference	Description
1	3-pin Deutsch connector	A	Black
2	12V adapter	B	Red
3	Outer contact	C	White/Natural
4	3 amp slow-blow fuse		
5	Center contact		
6	Foil shield		

Figure 22: IMU Power Cable Pin-Out

A.2.1.3 IMU Performance

PERFORMANCE *(IMU)*		
IMU-LN200	Gyro Input Range	± 1000 degrees/s
	Gyro Rate Bias	1°/hr
	Gyro Rate Scale Factor	100 ppm
	Angular Random Walk	0.07 degrees/rt-hr
	Accelerometer Range	± 40 g
	Accelerometer Linearity	-
	Accelerometer Scale Factor	300 ppm
	Accelerometer Bias	0.3 mg

A.2.1.4 Electrical and Environmental

ELECTRICAL		
IMU Power Consumption	16 W (max)	
IMU Input Voltage	+12 to +28 V DC	
Receiver Power Consumption	ProPak-V3	2.8 W (typical)
System Power Consumption	ProPak-V3	14.8 W (typical)
Data Connector on Enclosure	13-pin Deutsch P/N 59065-11-35PF [a]	
Power Connector on Enclosure	3-pin Deutsch P/N 59065-09-98PN [a] +6 to +18 VDC	
IMU Interface	RS-232 or RS-422	
ENVIRONMENTAL *(LN-200 IMU)*		
Temperature	Operating	-30°C to +60°C (-22°F to 140°F)
	Storage	-45°C to +80°C (-49°F to 176°F)
Humidity	95% non-condensing	

a. For replacement connectors on the interface or power cables, see *Section H.3, Manufacturer's Part Numbers* on *page 294*.

A.2.2 iIMU-FSAS

Table 11: iIMU-FSAS Specifications

PHYSICAL	
IMU Size	128 mm x 128 mm x 104 mm (5.04” x 5.04” x 4.09”)
IMU Weight	2.1 kg (4.63 lb.)
MECHANICAL DRAWINGS [a]	

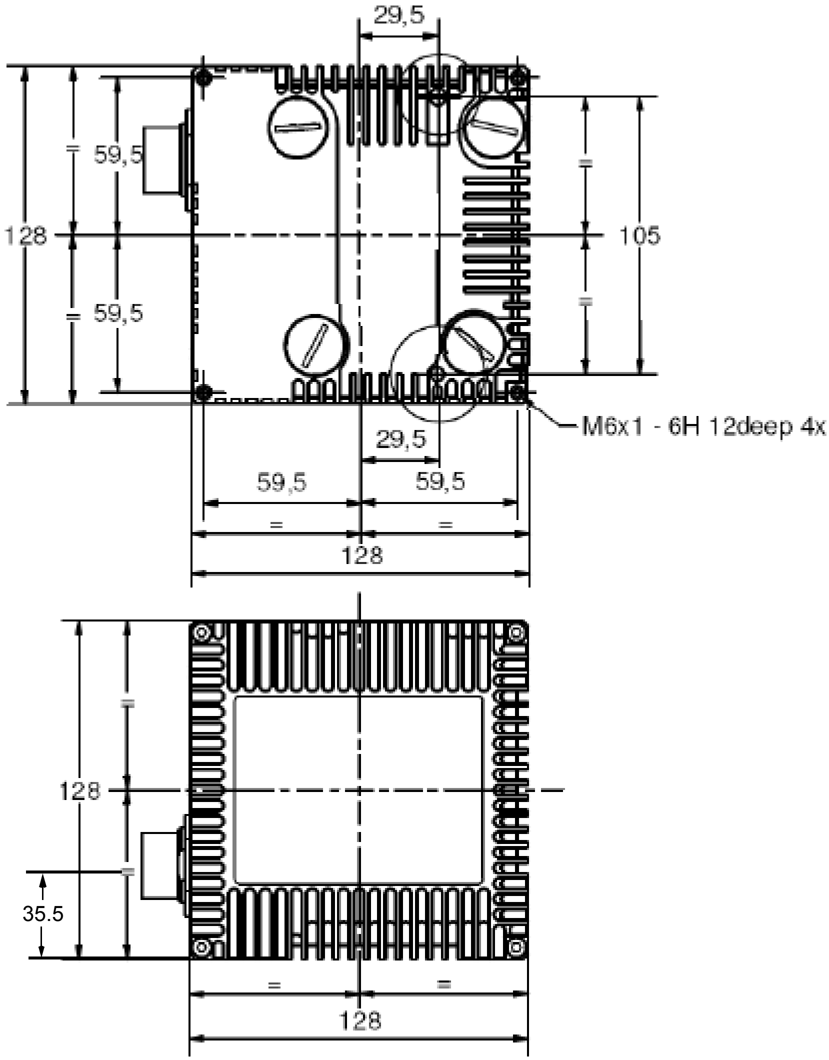


Figure 23: iIMU-FSAS Top/Bottom Dimensions

a. See *Figure 25* on *page 77* for the centre of navigation dimensions
b. Dimensions are in mm.

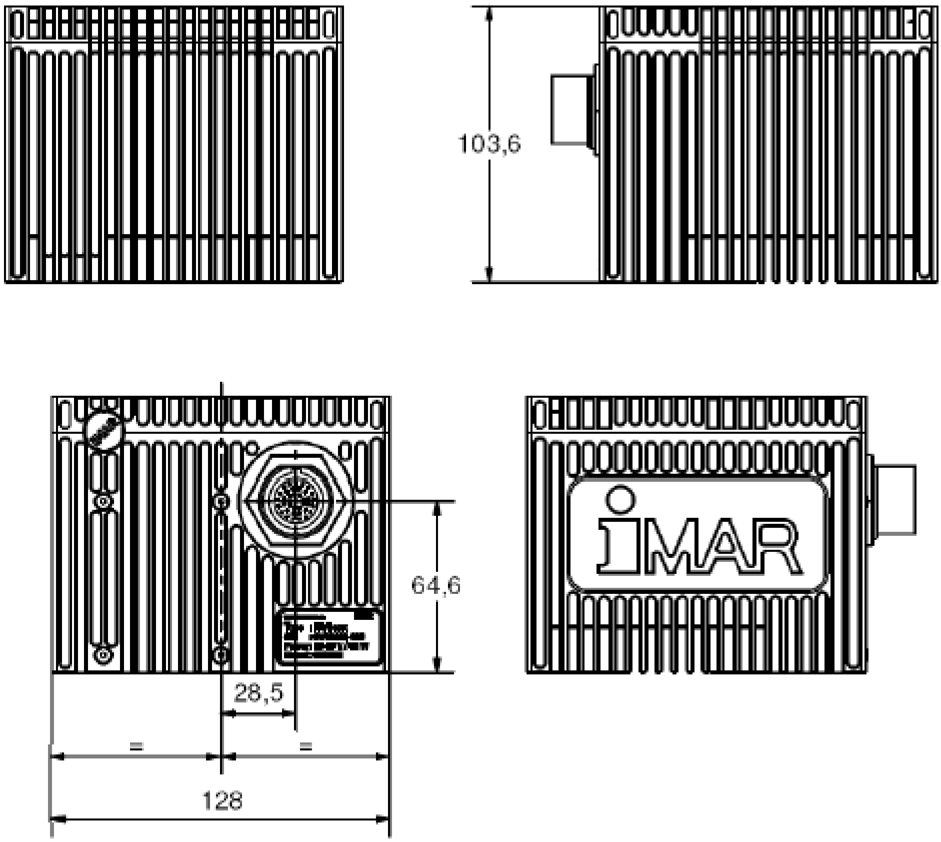


Figure 24: iIMU-FSAS Enclosure Side Dimensions

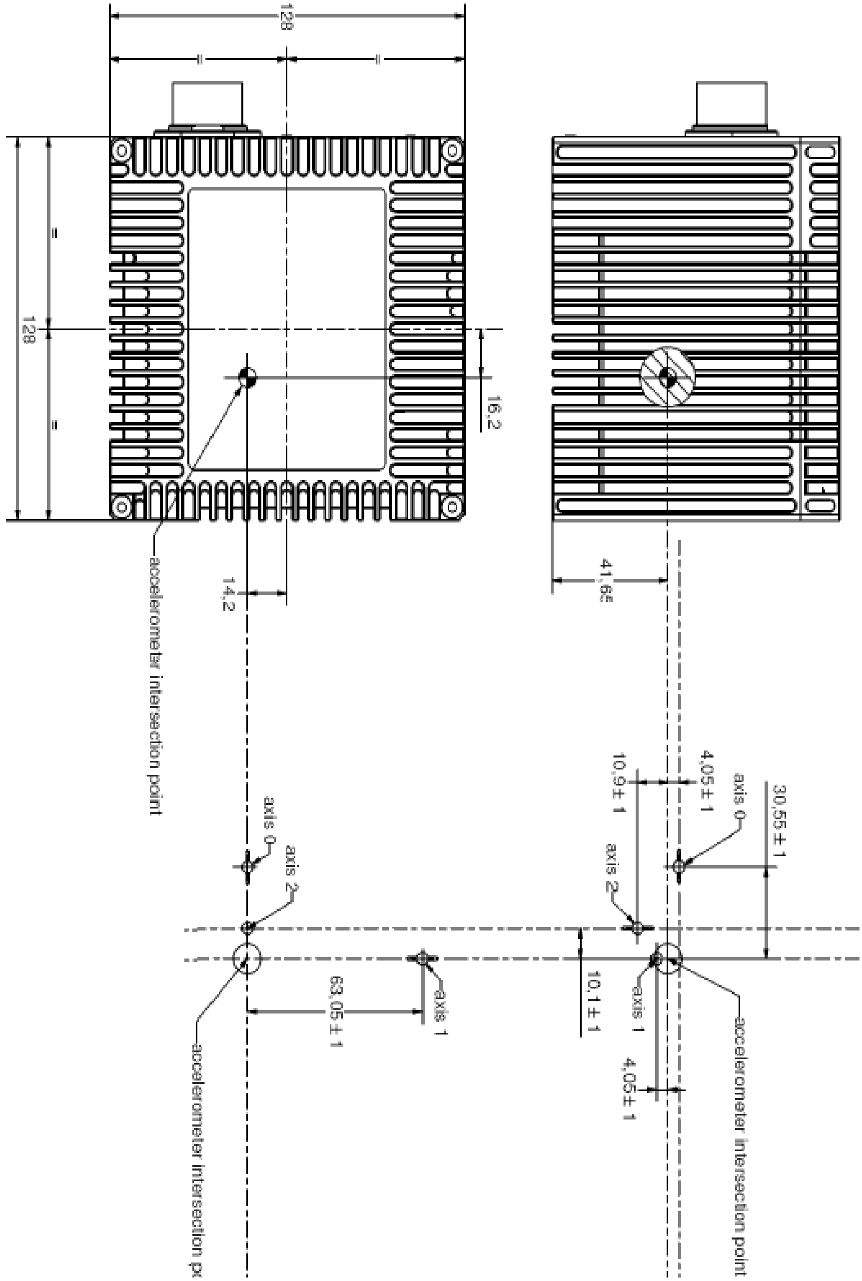


Figure 25: iIMU-FSAS Centre of Navigation

A.2.2.1 iIMU-FSAS Interface Cable

The NovAtel part number for the 1 m iIMU-FSAS interface cable is 01018221 (see *Table 12* on *page 79* and *Figure 30, iIMU-FSAS Interface Cable* on *page 82*). See also *Section A.2.2.2, iIMU-FSAS Odometer Cabling* on *page 80* if applicable.

To talk to the SPAN-SE with the iIMU-FSAS interface cable, a FSAS SPAN-SE Y Adapter cable is needed. Please see *Table 13* on *page 80* for cable pin-out information.

The iIMU interface cable supplied, provides power to the IMU from an external power source, and enables input and output between the receiver and IMU. *Figure 26* below shows the iIMU interface cable connections when used with a SPAN-SE receiver while the rest of the SPAN-SE connections are shown in *Figure 4* on *page 28*.

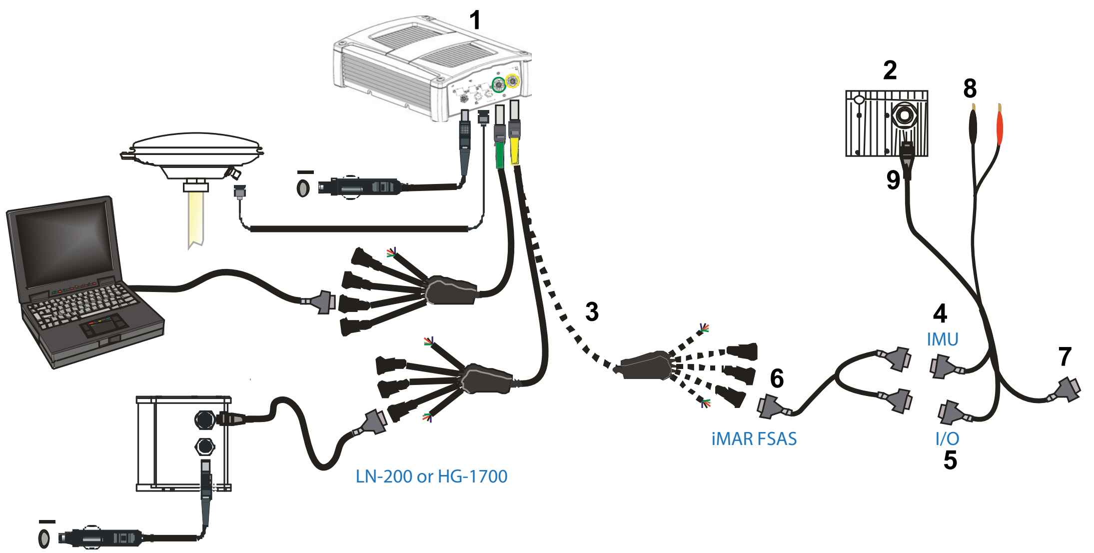


Figure 26: iIMU Interface Cable Connections with a SPAN-SE

Reference	Description
1	SPAN-SE receiver
2	iIMU-FSAS IMU
3	I/O 2 yellow cable's 30-pin connector to I/O 2 port on the SPAN-SE
4	iIMU interface cable's DB-9 IMU connector to iIMU interface Y cable
5	iIMU interface cable's DB-9 I/O connector to iIMU interface Y cable
6	iIMU interface Y cable to I/O 2 yellow cable's DB-9 IMU connector
7	iIMU interface cable's DB-9 ODO connector to (optional) wheel sensor cable
8	iIMU interface cable's (+ve) and (-ve) connectors to user-supplied power source
9	iIMU interface cable's MIL 22-pin connector to the iIMU-FSAS IMU

Table 12: IMU Interface Cable Pin-Out

MIL-C-38999 III Connector Pin	Function	Power 4 mm plugs	Female DB9 to COM3	Male DB9 to I/O	Male DB9 to ODO	Comments
1	PGND	Color: black Label: PGND				Power ground
2	ODO_AN				7	Odometer input A(-), opto-coupler: +2 to +6 V [a]
3	V_{IN}	Color: red Label: 10-34 VDC				+10 to +34 VDC
4	ODO_A				6	Odometer input A(+), opto-coupler: +2 to +6 V [a]
5-6	Reserved					
7	DAS			1 and 6		Shielded data acquisition signal (LVTTL to VARF)
8	Reserved					
9	DAS_ GND			9		Shielded ground reference for data acquisition & control signals
10	Reserved					
11	DON		8			Twisted pair; serial data output signal / RS-422(-)
12	DO		2			Twisted pair; serial data output signal / RS-422(+)
13	Reserved					
14	DGND		5			Digital ground
15	DGND		5			Digital ground
16	ODO_B				3	Odometer input B(+), opto-coupler: +2 to +6 V [a]
17	ODO_BN				1	Odometer input B(-), opto-coupler: +2 to +6 V [a]
18	Reserved					
19	DI		3			Twisted pair; serial data in / RS-422(+)
20	DIN		7			Twisted pair; serial data in / RS-422(-)
21	SW_ON_ SIG					Connected to Pin 3; switch IMU signal ON/OFF (voltage applied = ON) +4 to +34 V
22	SW_ON_ GND					Connected to Pin 1; ground for IMU signal ON

a. RS-422 compatible

Table 13: FSAS SPAN-SE Y Adapter Cable Pin-Out

Function	DB-9 Male to FSAS COM 3 Cable (M1 in *Figure 27*)	DB-9 Female to FSAS I/O Cable (F1 in *Figure 27*)	DB-9 Female to SPAN-SE Cable (F2 *in Figure 27*)	Description
DAS		1	1	Data acquisition and control signals
DO	2		2	Data output signal / RS-422(+)
DI	3		3	Data input signal / RS-422(+)
			4	
DGND	5	9	5	Digital ground
			6	
DIN	7		7	Data input signal / RS-422(-)
DON	8		8	Data output signal / RS-422(-)
			9	

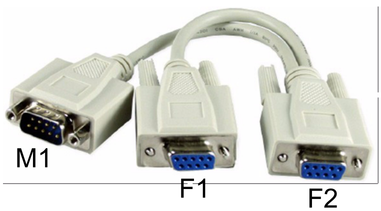


Figure 27: FSAS SPAN-SE Y Adapter Cable

A.2.2.2 iIMU-FSAS Odometer Cabling

The iIMU-FSAS with the –O wheel sensor option provides wheel sensor input from the Distance Measurement Instrument (DMI) through the DB-9 connector labelled "ODO" on the IMU interface cable. The IMU data goes through the IMU and then into the SPAN receiver through the serial communication line.

There are two DMI products that are compatible with the iIMU-FSAS system:

- iMWS-V2 (Magnetic Wheel Sensor) from iMAR
  - A magnetic strip and detector are installed inside the wheel. The signal then goes through a box that translates the magnetic readings into pulses that are then passed through the cable into the ODO connector on the IMU cable. See also *Figure 29* below.
- WPT (Wheel Pulse Transducer) from Corrsys Datron

- A transducer traditionally fits to the outside of a non-drive wheel. A pulse is then generated from the transducer which is fed directly to the ODO connector on the IMU cable.See also *Figure 28* on *page 81*.

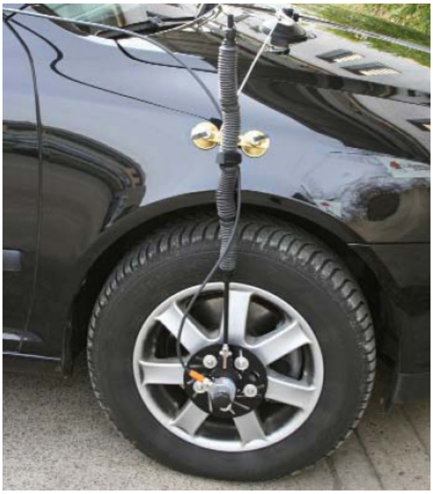


Figure 28: Corrsys Datron WPT

The WPT mounts to the wheel lug nuts via adjustable mounting collets. The torsion protection rod, which maintains rotation around the wheel axis, affixes to the vehicle body with suction cups. Refer to the Corrsys Datron WPT user manual for mounting instructions.

☒ The iMAR iMWS-V2 sensor is on the inside of the wheel so that all you can see in the vehicle is the grey signal converter box.

Figure 29: iMAR iMWS Pre-Installed

iMAR provides a sensor that operates with a magnetic strip glued inside the rim of a non-drive wheel and a special detector (iRS) mounted on the inside of the wheel (the disk of the wheel suspension, brake cover or brake caliper holder). Details are shown in the installation hints delivered with the system.

The NovAtel IMU interface cable, with ODO, is the same as that in *Section A.2.2.1* but with some of the reserved pins having odometer uses. It still provides power to the IMU from an external source, and enables input and output between the receiver and IMU.

See also *Section 3.5* on *page 50*. The cable modification is shown in *Table 14* below.

☒ Connect the female DB9 connector to the male ODO end of the iIMU-FSAS interface cable.

Table 14: Cable Modification for Corrsys Datron WPT

8-pin M12 Connector on the Corrsys Datron Cable [a, b]			Female DB9 Connector
Pin #	**Description**	**Color**	
1	GND	White	No change
2	$+U_B$ (Input Power)	Brown	
3	Signal A	Green	6
4	Signal A inverted	Yellow	7
5	Signal B	Grey	3
6	Signal B inverted	Pink	1
7	Reserved		No change
8			

a. Pin 2 is wired to a red banana plug (Power in) and Pin 1 is wired to a black banana plug (Power return) so the WPT needs power to operate (+10 to +30 V). Solder the shield on the WPT cable to the female DB9 housing.
b. This modification is for the Corrsys Datron WPT 8-pin M12-plug cable number 14865.

Figure 30: iIMU-FSAS Interface Cable

A.2.2.3 IMU Performance

PERFORMANCE *(IMU)*		
iIMU-FSAS	Gyro Input Range	± 500 degrees/s
	Gyro Rate Bias	0.75°/hr
	Gyro Rate Scale Factor	300 ppm
	Angular Random Walk	0.1 degrees/sq rt hr
	Accelerometer Range	± 5 g (± 20 g optional)
	Accelerometer Linearity	-
	Accelerometer Scale Factor	400 ppm
	Accelerometer Bias	1.0 mg

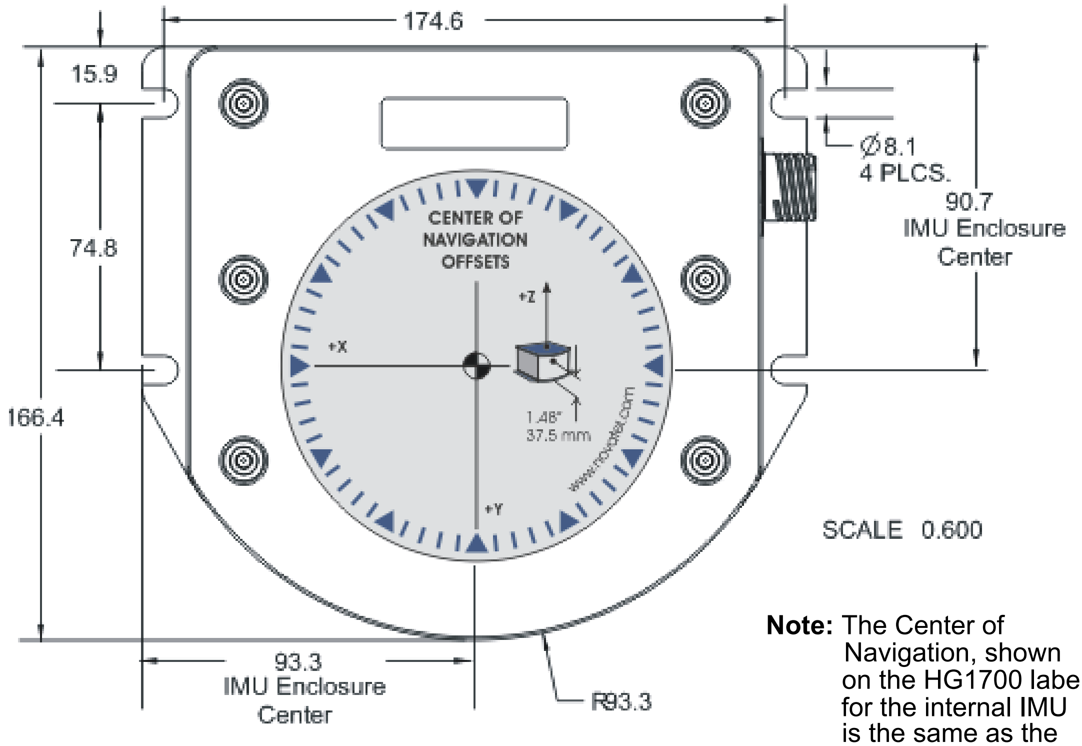
A.2.2.4 Electrical and Environmental

ELECTRICAL		
IMU Power Consumption	16 W (max)	
IMU Input Voltage	+10 to +34 V DC	
Receiver Power Consumption	ProPak-V3	2.8 W (typical)
System Power Consumption	ProPak-V3	14.8 W (typical)
Data Connector	MIL-C-38999-III	
Power Connector	MIL-C-38999-III (same as data connector)	
IMU Interface	RS-422	
ENVIRONMENTAL *(iIMU-FSAS)*		
Temperature	Operating Storage	-40°C to +71°C (-40°F to 160°F) -40°C to +85°C (-40°F to 185°F)
Humidity	95% non-condensing	

A.2.3 HG1700 IMU

Table 15: HG1700 IMU Specifications

PHYSICAL	
IMU Enclosure Size	193 mm x 167 mm x 100 mm (7.6” x 6.6” x 3.9”)
IMU Size	160 mm x 160 mm x 100 mm (6.3” x 6.3” x 3.9”)
IMU Weight	3.4 kg (7.49 lb.)
MECHANICAL DRAWINGS	



Note: The Center of Navigation, shown on the HG1700 label, for the internal IMU is the same as the enclosure’s center. The enclosure center measurements are labelled as: IMU Enclosure Center in this figure.

Figure 31: HG1700 Top/Bottom Dimensions

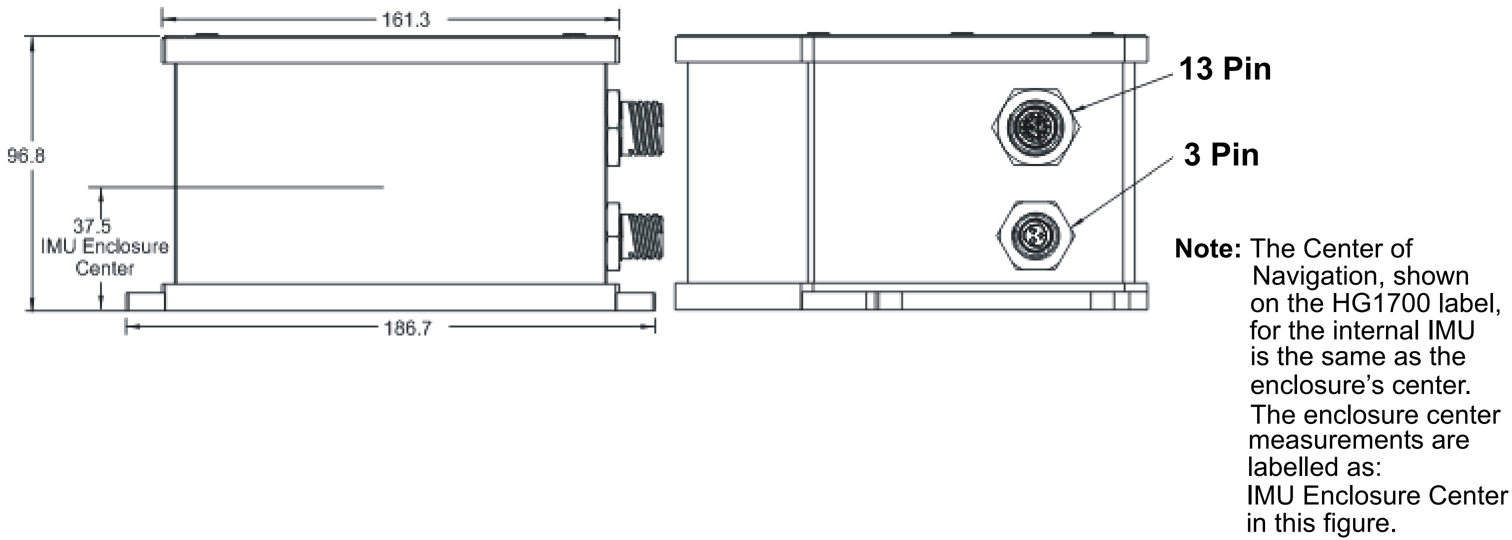


Figure 32: HG1700 Enclosure Side Dimensions

A.2.3.1 HG1700 IMU Interface Cable

The IMU interface cable supplied, the power adapter cable provides power to the IMU from an external power source, and enables input and output between the receiver and IMU. The HG1700 uses the same cable supplied with the LN-200, see *Figure 19* on *page 72*.

A.2.3.2 IMU Performance

PERFORMANCE *(IMU)*		
IMU-H58	Gyro Input Range Gyro Rate Bias Gyro Rate Scale Factor Angular Random Walk Accelerometer Range Accelerometer Linearity Accelerometer Scale Factor Accelerometer Bias	± 1000 degrees/s 1.0 degree/hr 150 ppm 0.125 degrees/rt hr ± 50 g 500 ppm 300 ppm 1.0 mg
IMU-H62	Gyro Input Range Gyro Rate Bias Gyro Rate Scale Factor Angular Random Walk Accelerometer Range Accelerometer Linearity Accelerometer Scale Factor Accelerometer Bias	± 1000 degrees/s 5.0 degrees/hr 150 ppm 0.5 degrees/rt-hr ± 50 g 500 ppm 300 ppm 3.0 mg

A.2.3.3 Electrical and Environmental

ELECTRICAL		
IMU Power Consumption	IMU-H58: 9 W (max) IMU-H62: 8 W (max)	
IMU Input Voltage	+12 to +28 V DC	
Receiver Power Consumption	ProPak-V3	2.8 W (typical)
System Power Consumption	ProPak-V3	14.8 W (typical)
Data Connector on Enclosure	13-pin Deutsch P/N 59065-11-35PF [a]	
Power Connector on Enclosure	3-pin Deutsch P/N 59065-09-98PN [a] +6 to +18 VDC	
IMU Interface	RS-232 or RS-422	
ENVIRONMENTAL *(IMU)*		
Temperature	Operating Storage	-30°C to +60°C (-22°F to 140°F) -45°C to +80°C (-49°F to 176°F)
Humidity	95% non-condensing	

a. For replacement connectors on the interface and power cables, see *Section H.3, Manufacturer's Part Numbers* on *page 294*.

Appendix B Commands

This appendix describes in detail the commands needed to configure the receiver and request the data you need.

For information on other available commands, refer to the *OEMV Family Firmware Reference Manual*.

B.1 Command Formats

The receiver accepts commands in 3 formats:

- Abbreviated ASCII
- ASCII
- Binary

Abbreviated ASCII is the easiest to use for your input. The other two formats include a CRC for error checking and are intended for use when interfacing with other electronic equipment.

Here are examples of the same command in each format:

Abbreviated ASCII Example:

LOG COM1 BESTPOSB ONTIME 1[CR]

ASCII Example:

LOGA,COM2,0,66.0,UNKNOWN,0,15.917,004c0000,5255,32858;COM1,
BESTPOSB,ONTIME,1.000000,0.000000,NOHOLD*F95592DD[CR]

Binary Example:

AA44121C 01000240 20000000 1D1D0000 29160000 00004C00 55525A80
20000000 2A000000 02000000 00000000 0000F03F 00000000 00000000
00000000 2304B3F1

B.2 Using a Command as a Log

All NovAtel commands may be used for data input, as normal, or used to request data output (a unique OEMV Family feature). INS-specific commands may be in Abbreviated ASCII or Binary format.

Consider the *lockout* command (refer to the *OEMV Family Firmware Reference Manual*) with the syntax:

lockout prn

You can put this command into the receiver to de-weight an undesirable satellite in the solution, or you can use the *lockout* command as a log to see if there is a satellite PRN that has already been locked out. In ASCII, this might be:

log com1 lockouta once

Notice the ‘a’ after *lockout* to signify you are looking for ASCII output.

⌧ The BESTPOS position log can be logged at rates up to 20 Hz directly from the OEMV port, but is available at 1 Hz or 5 Hz from any SPAN-SE port. Other GNSS logs (RANGE, PSRPOS, and so on) can be logged up to 20 Hz from the SPAN ports. The BESTGPSPOS log is available from SPAN-SE only, at 1 Hz or 5 Hz.

WARNING: Ensure that all windows, other than the Console, are closed in **CDU** and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload your system.

B.3 DOS Commands

The SPAN-SE receiver accepts many traditional DOS commands for accessing the SD Card. DOS commands that produce output (logs) **do not** conform to traditional NovAtel command/log formats. The resulting "logs" are output as simple ASCII as with normal DOS commands. To display the results to another COM port, the port must be passed as a parameter. The default device, and currently the only option for these commands, is the internal SD Card, see *Table 16* below.

Most commands are acknowledged with an OK or an Error message. However, due to the length of time the FORMAT command can take, it always responds with OK. When the format is taking place, the SD LED flashes green and orange. If the format fails, the LED blinks red indicating an error. Note that mounting a large capacity SD Card can also take extra time as the free space is being calculated. During mounting, the SD LED flashes green and orange to indicate "busy".

Table 16: Mass Storage Device

ASCII	Binary	Description
SD	0	Internal SD Card (default)

B.3.1 DIR - Show Directory

Command: DIR (Message ID = 1055)

Parameter	Values
COM Port Enum, see *Table 18* on *page 96*	(THISPORT = default)
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)

B.3.2 CD - Change Directory

Command: CD (Message ID = 1054)

Parameter	Values
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)
Path	Null terminated string

B.3.3 FORMAT - Format storage medium

Command: FORMAT (Message ID = 1057)

Parameter	Values
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)
Volume Name	Optional string

B.3.4 MKDIR - Make Directory

Command: MKDIR (Message ID = 1060)

Parameter	Values
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)
Path	Null terminated string

B.3.5 RMDIR - Remove Directory

Command: RMDIR (Message ID = 1058)

Parameter	Values
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)
Path	Null terminated string

B.3.6 PWD - Present Working Directory

Command: PWD (Message ID = 1061)

Parameter	Values
COM Port Enum, see *Table 18* on *page 96*	(THISPORT = default)
Mass Storage Device Enum, see *Table 16* on *page 88*	(SD = default)

B.3.7 FTP

The SPAN-SE has a built-in FTP server to simplify retrieving data from the SD Card. After the IP information has been set, using the IFCONFIG command, any FTP client can connect to the SPAN-SE on port 21. The FTP server allows basic file manipulation and directory browsing but files cannot be uploaded to the SD Card at this time.

To ensure high-speed logging is not corrupted, the FTP server reads from the SD Card when it is idle (that is, mounted and no log file open for writing). Attempting to use an FTP command when the card is not idle will result in this error: `425 SD Card not ready`.

Command	Description
GET <filename>	Copy file from SD Card to PC
DIR	Directory listing of SD Card
DEL <filename>	Delete file from SD Card
CWD	Change Working Directory

B.4 SPAN-SE Command Reference

For convenience, some commonly used OEMV commands are included in this manual. All SPAN-specific commands are included in this manual. Please refer to the *OEMV Family Firmware Reference Manual* for a complete list of GNSS-only commands, categorized by function and then detailed in alphabetical order.

B.4.1 APPLYVEHICLEBODYROTATION Enable vehicle to body rotation

This command allows you to apply the vehicle to body rotation to the output attitude (which was entered with the VEHICLEBODYROTATION command, see *page 149*). This rotates the SPAN computation frame output in the INSPVA, INSPVAS and INSATT logs to the vehicle frame. APPLYVEHICLEBODYROTATION is disabled by default.

Abbreviated ASCII Syntax: **Message ID: 1071**

APPLYVEHICLEBODYROTATION [switch]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	switch	Disable	0	Enable/disable vehicle body rotation using values entered in the vehiclebodyrotation command. default = disable	Enum	4	H
		Enable	1				

Abbreviated ASCII Example:

APPLYVEHICLEBODYROTATION ENABLE

B.4.2 ASSIGNLBAND Set L-band satellite communication parameters

You must use this command to ensure that the receiver searches for a specified L-band satellite at a specified frequency with a specified baud rate. The factory parameter default is ASSIGNLBAND IDLE.

☒ 1. In addition to a NovAtel receiver with L-band capability, a subscription to the OmniSTAR, or use of the free CDGPS, service is required. Contact NovAtel for details, see *page 18*.

2. The frequency assignment, field #3 below, can be made in kHz or Hz. For example:

```
ASSIGNLBAND OMNISTAR 1557855 1200
```

A value entered in Hz is rounded to the nearest 500 Hz.

3. OmniSTAR has changed channels (frequencies) on the AMSC Satellite that broadcasts OmniSTAR corrections for North America. NovAtel receivers do not need a firmware change. To change frequencies, connect your receiver and issue an ASSIGNLBAND command. For example, the Western Beam frequency as stated on OmniSTAR's Web site is 1557.8550 MHz. Input into the receiver: assignlband omnistar 1557855 1200.

4. The NAD83 (CSRS) datum is available to CDGPS users. The receiver automatically transforms the CDGPS computed coordinates into WGS84 (the default datum of the receiver). Alternatively, select any datum, including CSRS, for a specified coordinate system output.

Abbreviated ASCII Syntax: **Message ID: 729**

ASSIGNLBAND mode freq baud

Factory Default:

ASSIGNLBAND IDLE

Abbreviated ASCII Example 1:

ASSIGNLBAND CDGPS 1547547 4800

Abbreviated ASCII Example 2:

ASSIGNLBAND IDLE

Table 17: L-band Mode

Binary	ASCII	Description
0	Reserved	
1	OMNISTAR	When you select OmniSTAR, enter a dedicated frequency and baud rate.
2	CDGPS	When you select CDGPS, enter a dedicated frequency and baud rate.
3	IDLE	When you select IDLE, the receiver is configured to stop tracking any L-band satellites. The 'freq' and 'baud' fields are optional so that you may select IDLE without specifying the other fields.
4	OMNISTARAUTO	When you select OMNISTARAUTO, the receiver automatically selects the best OmniSTAR beam to track based on the receiver's position. This requires the receiver to have a downloaded satellite list from an OmniSTAR satellite. Therefore, a manual assignment is necessary the first time an OmniSTAR satellite is assigned on a new receiver. After collection, the satellite list is stored in NVM for subsequent auto assignments. Lists are considered valid for 6 months and are constantly updated while an OmniSTAR signal is tracking. If the receiver has a valid satellite list, it is reported in a status bit in the LBANDSTAT log, see *page 225*. [1]
5	OMNISTARNARROW	When you select OMNISTARNARROW, enter a dedicated frequency and baud rate. For re-acquisitions of the L-band signal, the receiver uses a 1500 Hz search window and the stored TCXO offset information. To remove the TCXO offset information from NVM, use the FRESET LBAND_TCXO_OFFSET command. A standard FRESET command does not do this, see *page 105*. [2]

1. The receiver will always track an available local beam over a global beam. The receiver constantly monitors the satellite list to ensure it is tracking the best one and automatically switches beams if it is not tracking the best one.
2. Refer also to the *L-band Tracking and Data Output with GPS* application note available on our Web site as APN-043 at http://www.novatel.com/support/applicationnotes.htm.

B.4.2.1 Beam Frequencies

You can switch between Omnistar VBS and CDGPS by using the following commands:

Use CDGPS

assignlband cdgps <freq> 4800
psrdiffsource cdgps

Use OmniStar VBS

assignlband omnistar <freq> 1200
psrdiffsource omnistar

Where `<freq>` is determined for CDGPS or OmniStar as follows:

1. CDGPS beam frequency chart:

   - East 1547646 or 1547646000
   - East-Central 1557897 or 1557897000
   - West-Central 1557571 or 1557571000
   - West 1547547 or 1547547000

2. The OmniStar beam frequency chart can be found at http://www.omnistar.com/chart.html.

   For example:

   Eastern US (Coverage is Northern Canada to southern Mexico) 1557845 or 1557845000

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	ASSIGNLBAND header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	mode	See *Table 17*		Set the mode and enter specific frequency and baud rate values	Enum	4	H
3	freq	1525000 to 1560000 or 1525000000 to 1560000000		L-band service beam frequency of satellite (Hz or kHz). See also *Beam Frequencies* on *page 94*. (default = 1536782 if the mode is OMNISTAR)	Ulong	4	H+4
4	baud	300, 600, 1200, 2400 or 4800		Data rate for communication with L-band satellite (default = 1200)	Ulong	4	H+8

B.4.3 COM Port configuration control

This command permits you to configure the SPAN-SE receiver's asynchronous serial port communications drivers.

The current COM port configuration can be reset to its default state at any time by sending it two hardware break signals of 250 milliseconds each, spaced by fifteen hundred milliseconds (1.5 seconds) with a pause of at least 250 milliseconds following the second break. This will:

- Stop the logging of data on the current port (see UNLOGALL on *page 154)*
- Clear the transmit and receive buffers on the current port
- Return the current port to its default settings
- Set the interface mode to NovAtel for both input and output (see the GNSSCARDCONFIG command on *page 106*)

✉ 1. The COMCONTROL command, see *page 98*, may conflict with handshaking of the selected COM port. If handshaking is enabled, then unexpected results may occur.

2. Watch for situations where the COM ports of two receivers are connected together and the baud rates do not match. Data transmitted through a port operating at a slower baud rate may be misinterpreted as break signals by the receiving port if it is operating at a higher baud rate. This is because data transmitted at the lower baud rate is stretched relative to the higher baud rate. In this case, configure the receiving port to have break detection disabled using the COM command.

3. Baud rates higher than 115,200 bps are not supported by standard PC hardware. Special PC hardware may be required for higher rates, including 230400 bps, 460800 bps and 921600 bps. Also, some PC's have trouble with baud rates beyond 57600 bps.

Abbreviated ASCII Syntax: **Message ID: 4**

COM [port] bps [parity[databits[stopbits[handshake[echo[break]]]]]]

Factory Default:

com com1 9600 n 8 1 n off on
com com2 9600 n 8 1 n off on
com com3 9600 n 8 1 n off on
com com4 9600 n 8 1 n off on

Abbreviated ASCII Example:

COM COM1 57600 N 8 1 N OFF ON

Table 18: COM Serial Port Identifiers

Binary	ASCII	Description
1	COM1	COM Port 1
2	COM2	COM Port 2
3	COM3	COM Port 3
7	FILE	SD Card
13	USB1	USB Device
19	COM4	COM Port 4
20	ETH1	10/100 Ethernet

Table 19: Parity

Binary	ASCII	Description
0	N	No parity (default)
1	E	Even parity
2	O	Odd parity

Table 20: Handshaking

Binary	ASCII	Description
0	N	No handshaking (default)
1	XON	XON/XOFF software handshaking
2	CTS	CTS/RTS hardware handshaking

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	COM header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	port	See *Table 18* on *page 96*		Port to configure.	Enum	4	H
3	bps/baud	300, 600, 900, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, or 230400		Communication baud rate (bps). Bauds of 460800 and 921600 are also available on COM1 of OEMV-2-based products.	ULong	4	H+4
4	parity	See *Table 19* on *page 96*		Parity	Enum	4	H+8
5	databits	7 or 8		Number of data bits (default = 8)	ULong	4	H+12
6	stopbits	1 or 2		Number of stop bits (default = 1)	ULong	4	H+16
7	handshake	See *Table 20* on *page 96*		Handshaking	Enum	4	H+20
8	echo	OFF	0	No echo (default)	Enum	4	H+24
		ON	1	Transmit any input characters as they are received			
9	break	OFF	0	Disable break detection	Enum	4	H+28
		ON	1	Enable break detection (default)			

B.4.4 COMCONTROL Control the RS232 hardware control lines

This command is used to control the hardware control lines of the COM ports. On SPAN-SE, the mode of COM1, COM2, COM3 and COM4 can be configured to be RS232 or RS422. On OEMV products, the mode is only hardware configurable. The TOGGLEPPS mode of this command is typically used to supply a timing signal to a host PC computer by using the RTS or DTR lines. The accuracy of controlling the COM control signals is better than 900 μs. As a SPAN-SE user, you have access to 3 event out lines that can provide precise PPS output. The other modes are typically used to control custom peripheral devices. Also, it is possible to communicate with all three serial ports simultaneously using this command.

---

1. If handshaking is disabled, any of these modes can be used without affecting regular RS232 communications through the selected COM port. However, if handshaking is enabled, it may conflict with handshaking of the selected COM port, causing unexpected results.
2. Be aware that RS422 transceiver code and hardware handshaking are mutually exclusive.
3. The PULSEPPSLOW control type cannot be issued for a TX signal.
4. Only PULSEPPSHIGH, FORCEHIGH and FORCELOW control types can be used for a TX signal.
5. The IMU port does not need to be configured by the user. Do not attempt to do so.

Abbreviated ASCII Syntax: **Message ID: 431**

COMCONTROL [port] [signal] [control] mode

Factory Default:

comcontrol com1 rts default rs232

comcontrol com2 rts default rs232

comcontrol com3 rts default rs232

comcontrol com4 rts default rs232

Abbreviated ASCII Example

COMCONTROL COM1 RS422

Table 21: Tx, DTR and RTS Availability

	Tx Available On:	DTR Available On:	RTS Available On:
SE-CARD	COM1, COM2, COM3, COM4	COM1, COM2, COM3, COM4	COM1, COM2, COM3, COM4

Table 22: SPAN-SE COM Port Values

Binary	ASCII
1	COM1
2	COM2
3	COM3
6	THISPORT
7	FILE
8	ALL
13	USB1
19	COM4
20	ETH1

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	COMCONTROL header	-	-	This field contains the command name or message header depending on whether command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	port	See *Table 22* on *page 99*		RS232/RS422 port to control. Valid ports are COM1, COM2, COM3 and COM4	Enum	4	H
3	signal	RTS	0	COM signal to control: RTS, DTR and TX. See *Table 21* on *page 98*	Enum	4	H+4
		DTR	1				
		TX	2				
4	control	DEFAULT	0	Disables this command and returns the COM signal to its default state	Enum	4	H+8
		FORCEHIGH	1	Forces signal high			
		FORCELOW	2	Forces signal low			
		TOGGLE	3	Immediately toggles the current sate of the signal			
		TOGGLEPPS	4	Toggles state of selected signal within 900 µs after each 1PPS event. State change of signal lags 1PPS by an average of 450 µs. Delay of each pulse varies by a uniformly random amount < 900 µs.			
		PULSEPPSLOW	5	Pulses the line low at a 1PPS event and to high 1 ms after it. Not for TX.			
		PULSEPPSHIGH	6	Pulses line high for 1 ms at time of a 1PPS event			
5	mode	RS232	0	RS-232 mode	Enum	4	H+12
		RS422	1	RS-422 mode			
		N/A	2	Used only for ETH1 and USB1 information			

B.4.5 COMVOUT Turn power to the ports on or off

This command allows you turn power to the COM ports on or off (all on or all off).

Power is supplied at the input voltage, out through Pin 4 of COM1, COM2 and COM4.

☒ Power is turned on through Pin 4 of COM1, COM2 and COM4. Ensure the connections are correct before issuing this command, to prevent damage to the electronics.

Abbreviated ASCII Syntax: **Message ID: 779**

COMVOUT switch

Factory Default:

COMVOUT OFF

Abbreviated ASCII Example:

COMVOUT ON

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	switch	0	OFF	The state of the output power lines.	Enum	4	H
		1	ON				

B.4.6 EVENTINCONTROL Control mark input properties

This command controls up to four Event-In input triggers. See also *Section 3.10, Synchronizing External Equipment* starting on *page 58*.

Abbreviated ASCII Syntax: **Message ID: 614**

EVENTINCONTROL mark event [polarity] [t_bias] [t_guard]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	mark	MARK1	0	Choose which Event-In mark to use	Enum	4	H
		MARK2	1				
		MARK3	2				
		MARK4	3				
3	event	DISABLE	0	Disables Event input	Enum	4	H+4
		EVENT	1	Captures a single asynchronous event with the input			
		COUNT	2	Increments a counter with each input (for a wheel sensor, for example). Period of count is from one 1PPS to the next PPS.			
4	polarity	NEGATIVE	0	Negative polarity (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity			
5	t_bias			If Field #3 is EVENT: Time bias in nanoseconds: default = 0 minimum = -999 999 999 maximum = 999 999 999 If Field #3 is COUNT: This field is not used	Long	4	H+12
6	t_guard			If Field #3 is EVENT: Time guard in milliseconds: default = 4 minimum = 4 maximum = 3 599 999 If Field #3 is COUNT: This field is not used	Ulong	4	H+16

Abbreviated ASCII Example:

EVENTINCONTROL MARK1 COUNT

B.4.7 EVENTOUTCONTROL Control PPS signal properties

This command controls up to three Event-Out output triggers (PPS signal properties). See also *Section 3.10, Synchronizing External Equipment* starting on *page 58*.

☒ The `EVENTOUTCONTROL MARK1 ENABLE POSITIVE 10000000 240000000` command will generate a 4 Hz signal. The signal is held high for 10 ms during each cycle and the leading edge of the high signal is aligned to the 1PPS.

Abbreviated ASCII Syntax: **Message ID: 613**

EVENTOUTCONTROL mark switch [polarity] [active period] [non-active period]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	mark	MARK1	0	Choose which Event-Out mark to use	Enum	4	H
		MARK2	1				
		MARK3	2				
3	switch	DISABLE	0	Disables Event output	Enum	4	H+4
		ENABLE	1				
4	polarity	NEGATIVE	0	Negative polarity (default)	Enum	4	H+8
		POSITIVE	1	Positive polarity			
5	active period			Active period of the Event Out signal in nanoseconds: default = 500 000 000 minimum = 1000 maximum = 999 999 000	Ulong	4	H+12
6	not-active period			Not-active period of the Event Out signal in nanoseconds: default = 500 000 000 minimum = 1000 maximum = 999 999 000	Ulong	4	H+16

Abbreviated ASCII Example:

EVENTOUTCONTROL MARK3 ENABLE

B.4.8 FORMAT Format the SD Card

This command allows you to format the SC card in the SPAN-SE.

Abbreviated ASCII Syntax: **Message ID: 1057**

FORMAT device [volume]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	device			Choose a mass storage device, see *Table 16* on *page 88*	Enum	4	H
3	volume			DOS volume label	String[11]	11	H+4

Abbreviated ASCII Example:

FORMAT SD

B.4.9 FRESET Factory reset

This command clears data which is stored in non-volatile memory. Such data includes the almanac, ephemeris, and any user-specific configurations. The receiver is forced to hardware reset.

When the SPAN-SE receives a FRESET command, it is also passed to the OEMV-3 but without any parameters. Therefore the OEMV-3 only does a full reset. SPAN-SE can do a partial reset of some of its fields.

Abbreviated ASCII Syntax: **Message ID: 20**

FRESET [target]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	target	See *Table 23*		What data is to be reset by the receiver.	Enum	4	H

Abbreviated ASCII Example:

FRESET USER_CFG

Table 23: FRESET Target

Binary	ASCII	Description
0	STANDARD	Resets commands and INS data
1	USER_CFG	Resets the stored commands (user configuration)
4	MODEL	Resets the currently selected model
6	INS_LEVER_ARM	Resets the GNSS antenna to IMU lever arm
7	VEHICLE_BODY_R	Resets stored vehicle to body rotations

B.4.10 GNSSCARDCONFIG GNSS port configuration

Use this command to configure both the interface mode and COM port mode on an internal GNSS card from a SPAN-SE receiver port. Do this from a [COM1], [COM2], [COM3], [COM4], [ETH1], or [USB1] prompt. The GNSSCARDCONFIG command is especially useful for configuring RTK because the OEMV3 COM1 port is used for RTK correction input data.

You cannot use this command with the OEMV2 and OEMV3 connectors on the I/O 1 Green cable, as they provide direct access to the OEMV-2 and OEMV-3 GNSS cards respectively within the receiver. Instead, use the standard OEMV family INTERFACEMODE and COM commands.

Abbreviated ASCII Syntax: **Message ID: 1092**

GNSSCARDCONFIG [card] [port] rx_inter tx_inter [response] bps [parity] [data bits] [stop bits] [handshaking] [echo] [break]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	card	CARD1	1	Select a receiver card: CARD1 is OEMV3 (default) and CARD2 is OEMV2	Enum	4	H
		CARD2	2				
3	port	COM1	1	Enter COM1 only for the COM1 port on the GNSS receiver (not a SPAN-SE port) default = COM1	Enum	4	H+4
4	rx_inter	See *Table 24* on *page 107*		Receiver interface mode	Enum	4	H+8
5	tx_inter			Transmit interface mode	Enum	4	H+12
6	response	OFF	0	Response mode default = ON	Enum	4	H+16
		ON	1				
7	bps			Bits per second (or baud rate)	Ulong	4	H+20
8	parity	N	0	No parity (default)	Enum	4	H+24
		E	1	Even parity			
		O	2	Odd parity			
9	data bits	7 or 8		Number of data bits: 7 or 8 (default)	Ulong	4	H+28
10	stop bits	1 or 2		Number of stop bits: 1 (default) or 2	Ulong	4	H+32
11	handshaking	N	0	No handshaking (default)	Enum	4	H+36
		XON	1	XON/XOFF software handshaking			
		CTS	2	CTS/RTS hardware handshaking			
12	echo	OFF	0	No echo (default)	Enum	4	H+40
		ON	1	Transmit any input characters as they are received			
13	break	OFF	0	Disable break detection	Enum	4	H+44
		ON	1	Enable break detection (default)			

Abbreviated ASCII Example:

GNSSCARDCONFIG CARD1 COM1 RTCA NOVATEL ON 57600 N 8 1 N OFF ON

Table 24: Serial Port Interface Modes

Binary Value	ASCII Mode Name	Description
0	NONE	The port accepts/generates nothing
1	NOVATEL	The port accepts/generates NovAtel commands and logs
2	RTCM	The port accepts/generates RTCM corrections
3	RTCA	The port accepts/generates RTCA corrections
4	CMR	The port accepts/generates CMR corrections
5	OMNISTAR	The port accepts/generates OmniSTAR corrections
6-7	Reserved	
8	RTCMNOCR	RTCM with no CR/LF appended [1]
9	CDGPS	The port accepts GPS*C data [2]
10-13	Reserved	
14	RTCMV3	The port accepts/generates RTCM Version 3.0 corrections
15	NOVATELBINARY	The port only accepts/generates binary messages. If an ASCII command is entered when the mode is set to binary only, the command is ignored. Only properly formatted binary messages are responded to and the response is a binary message.
16-17	Reserved	
18	GENERIC	The port accepts/generates nothing. SEND/SENDHEX commands from another port generate data on this port. Any incoming data on this port can be seen with OEMV PASSCOM logs on another port.
20	MRTCA	The port accepts Modified RTCA (MRTCA) data to output CDGPS positions. This is useful on a receiver, such as the OEMV-2, that does not track CDGPS. You must use this feature in combination with a CDGPS-cable receiver, such as an OEMV-3, which can access the CDGPS signals and then rebroadcast them to MRTCA corrections.

1. An output interface mode of RTCMNOCR is identical to RTCM but with the CR/LF appended. An input interface mode of RTCMNOCR is identical to RTCM and functions with or without the CR/LF.
2. CDGPS has three options for output of differential corrections - NMEA, RTCM, and GPS*C. If you have a ProPak-V3 receiver, you do not need to use CDGPS as the argument. The CDGPS argument is for use with obsolete external non-NovAtel CDGPS receivers. These receivers use GPS*C (NavCanada's proprietary format differential corrections from the CDGPS service).

B.4.11 IFCONFIG Set IP information

Use this command to configure Internet Protocol (IP) information. See also *Section 3.11, SPAN-SE Ethernet Connection* on *page 60*.

Abbreviated ASCII Syntax: **Message ID: 1059**

IFCONFIG IP mask gateway

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	IP			IP address	Ulong	4	H
3	mask			IP mask	Ulong	4	H+4
4	gateway			IP gateway	Ulong	4	H+8

Abbreviated ASCII Example:

IFCONFIG 198.161.73.11 255.255.255.0 198.161.73.1

B.4.12 INSCOMMAND INS control command

This command allows you to enable, disable or reset INS positioning. When INS positioning is disabled, no INS position, velocity or attitude is output. Also, INS aiding of RTK initialization and tracking reacquisition is disabled. If the command is used to disable INS and then re-enable it, the INS system has to go through its alignment procedure (equivalent to issuing a RESET command). See also *Section 3.4.1, Configuration for Alignment* starting on *page 45*

Abbreviated ASCII Syntax: **Message ID: 379**

INSCOMMAND action

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	action	RESET	0	Resets the GNSS/INS alignment and restarts the alignment initialization.	Enum	4	H
		DISABLE	1	Disables INS positioning.			
		ENABLE	2	Enables INS positioning where alignment initialization starts again. (default)			

Abbreviated ASCII Example:

INSCOMMAND ENABLE

B.4.13 INSZUPT Request Zero Velocity Update

This command allows you to manually perform a Zero Velocity Update (ZUPT), that is, to update the receiver when the system has stopped.

NovAtel's SPAN Technology System does ZUPTs automatically. It is not necessary to use this command under normal circumstances.

WARNING: **This command should only be used by advanced users of GNSS/INS.**

Abbreviated ASCII Syntax: **Message ID: 382**

INSZUPT

B.4.14 LEVERARMCALIBRATE INS Calibration Command

Use the LEVERARMCALIBRATE command to control the IMU to antenna lever arm calibration.

The IMU to antenna lever arm is the distance from the IMU centre of navigation to the phase centre of the antenna. See also the SETIMUTOANTOFFSET command starting on *page 137* and *Section 3.4.6, Lever Arm Calibration Routine* starting on *page 48*.

The calibration runs for the time specified or until the specified uncertainty is met. The BESTLEVERARM log outputs the lever arm calculations once the calibration is complete, see also *page 178*.

If a SETIMUANTOFFSET command is already entered (or there is a previously saved lever arm in NVM), before the LEVERARMCALIBRATE is sent, the calibration starts using initial values from SETIMUTOANTOFFSET (or NVM). Ensure the initial standard deviations are representative of the initial lever arm values.

Abbreviated ASCII Syntax: **Message ID: 675**

LEVERARMCALIBRATE [switch] maxtime [maxstd]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	switch	OFF	0	Offset along the IMU X axis	Enum	4	H
		ON (default)	1				
3	maxtime	0 - 1000		Maximum calibration time (s)	Double	8	H+4
4	maxstd	0.02 – 0.5		Maximum offset uncertainty (m)	Double	8	H+12

Abbreviated ASCII Example 1:

LEVERARMCALIBRATE 600

Given this command, the lever arm calibration runs for 600 seconds. The final standard deviation of the estimated lever arm is output in the BESTLEVERARM log.

☒ The calibration starts when the SPAN solution reaches `INS_ALIGNMENT_COMPLETE`. The example's 600 s duration is from when calibration begins and not from when you issue the command.

Abbreviated ASCII Example 2:

LEVERARMCALIBRATE 600 0.05

Given this command, the lever arm calibration runs for 600 s or until the estimated lever arm standard deviation is <= 0.05 m in each direction (x, y, z), whichever happens first.

Abbreviated ASCII Example 3:

LEVERARMCALIBRATE OFF 0

This command stops the calibration. The current estimate, when the command was received, is output in the BESTLEVERARM log, and used in the SPAN computations.

B.4.15 LOG Request logs from the receiver

Many different types of data can be logged using several different methods of triggering the log events. The ONTIME trigger option requires the addition of the *period* parameter. See *Section C.1, Log Types* starting on *page 158* for further information and a complete list of data log structures. The *LOG* command tables in this section show the binary format followed by the ASCII command format.

The optional parameter [hold] prevents a log from being removed when the UNLOGALL command, with its defaults, is issued. To remove a log which was invoked using the [hold] parameter requires the specific use of the UNLOG command, see *page 152*. To remove all logs that have the [hold] parameter, use the UNLOGALL command with the *held* field set to 1, see *page 154*.

The [port] parameter is optional. If [port] is not specified, [port] is defaulted to the port that the command was received on.

☒ 1. SPAN-SE users can request up to 25 GNSS only logs (that is, logs generated on the internal OEMV-3), and up to 30 SPAN-specific logs, provided the requested data amount is less than the effective baud rate of the communication port logging the data. If you attempt to log more than 30 logs at a time, the receiver responds with an Insufficient Resources error.

2. Maximum flexibility for logging data is provided to the user by these logs. The user is cautioned, however, to recognize that each log requested requires additional CPU time and memory buffer space. Too many logs may result in lost data. Receiver overload can be monitored using the idle-time field and buffer overload bits of the Receiver Status in any log header.

3. Polled log types do not allow fractional offsets or ONTIME rates faster than 1Hz.

4. Use the ONNEW trigger with the MARK*x*TIME or MARK*x*PVA logs, see *page 236*

5. Only the MARK*x*PVA logs, or MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

6. If you do use the ONTIME trigger with asynchronous logs, the time stamp in the header does not necessarily represent the time the data was generated, but rather the time when the log is being transmitted. If the log contains a time parameter in the message itself, this time will be the time of validity of the data.

Abbreviated ASCII Syntax: **Message ID: 1**

LOG [port] message [trigger [period [offset [hold]]]]

Factory Default:

log com1 rxstatuseventa onnew 0 0 hold

log com2 rxstatuseventa onnew 0 0 hold

log com3 rxstatuseventa onnew 0 0 hold

log com4 rxstatuseventa onnew 0 0 hold

```
log usb1 rxstatuseventa onnew 0 0 hold
```

Abbreviated ASCII Example 1:

```
LOG COM1 PSRPOS ONTIME 1 0.5 HOLD
```

The above example shows BESTPOS logging to COM port 1 at 1 second intervals and offset by 0.5 seconds (output at 0.5, 1.5, 2.5 seconds and so on). The [hold] parameter is set so that logging is not disrupted by the UNLOGALL command.

To send a log only one time, the trigger option can be ignored.

Abbreviated ASCII Example 2:

```
LOG COM1 PSRPOS ONCE NOHOLD
```

See *Section Section B.1, Command Formats* on *page 87* for additional examples.

Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
1	LOG (binary) header	(See *Table 34, Binary Message Header Structure,* on *page 163*)	This field contains the message header.	-	H	0
2	port	See *Table 18, COM Serial Port Identifiers,* on *page 96*	Output port	Enum	4	H
3	message	Any valid message ID	Message ID of log to output	UShort	2	H+4
4	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7
6	trigger	0 = ONNEW	Does not output current message but outputs when the message is updated (not necessarily changed) [1]	Enum	4	H+8
		1 = ONCHANGED	Outputs the current message and then continue to output when the message is changed			
		2 = ONTIME	Output on a time interval			
		3 = ONNEXT	Output only the next message			
		4 = ONCE	Output only the current message			
7	period	Valid values for the high rate logging are 0.05, 0.1, 0.2, 0.25 and 0.5. For logging slower than 1Hz any integer value is accepted.	Log period (for ONTIME trigger) in seconds [2]	Double	8	H+12

Continued on page 116

Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
8	offset	A valid value is any integer smaller than the period. These decimal values, on their own, are also valid: 0.1, 0.2, 0.25 or 0.5	Offset for period (ONTIME trigger) in seconds. If you wished to log data at 1 second after every minute you would set the period to 60 and the offset to 1	Double	8	H+20
9	hold	0 = NOHOLD	Allow log to be removed by the UNLOGALL command	Enum	4	H+28
		1 = HOLD	Prevent log from being removed by the default UNLOGALL command			

1. See also the MARK*x*PVA and MARK*x*TIME logs starting on *page 236*.
2. See *Appendix A* in the *OEMV Family Installation and Operation User Manual* for the maximum raw measurement rate to calculate the minimum period. If the value entered is lower than the minimum measurement period, the value is ignored and the minimum period is used.

Field	Field Name	ASCII Value	Description	Field Type
1	LOG (ASCII) header	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII or ASCII respectively.	-
2	port	See *Table 18, COM Serial Port Identifiers,* on *page 96*	Output port	Enum
3	message	Any valid message name, with an optional A or B suffix.	Message name of log to output	Char [ ]
4	trigger	ONNEW	Output when message is updated (not necessarily changed) (see *Footnote 1* on *page 116*)	Enum
		ONCHANGED	Output when the message is changed	
		ONTIME	Output on a time interval	
		ONNEXT	Output only the next message	
		ONCE	Output only the current message (default)	
5	period	Any positive double value larger than the receiver's minimum raw measurement period	Log period (for ONTIME trigger) in seconds (default = 0) (see *Footnote 2* on *page 116*)	Double
6	offset	Any positive double value smaller than the period.	Offset for period (ONTIME trigger) in seconds. If you wished to log data at 1 second after every minute you would set the period to 60 and the offset to 1 (default = 0)	Double
7	hold	NOHOLD	Allow log to be removed by the UNLOGALL command (default)	Enum
		HOLD	Prevent log from being removed by the UNLOGALL command	

B.4.16 LOGFILE Log Data to a File on the SD Card

This command allows you to log data to a file on the SD Card.

Abbreviated ASCII Syntax: **Message ID: 157**

LOGFILE action [device] [filename]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	action	Open	0	Open a file to log to it or close a file.	Enum	4	H
		Close	1				
3	device			Choose a mass storage device, see *Table 16* on *page 88* default = SD	Enum	4	H+4
4	filename			Filename where filenames have a maximum 12 character limit. default = SPAN_#.log where # is the next number in the list starting at 0	Char[12]	12	H+8

Abbreviated ASCII Example:

LOGFILE OPEN SD SITE1.GPS

B.4.17 NMEATALKER Set the NMEA Talker ID

This command allows you to alter the behavior of the NMEA talker ID. The talker is the first 2 characters after the $ sign in the log header of the GPGGA, GPGLL, GPGRS, GPGSA, GPGST, GPGSV, GPRMB, GPRMC, GPVTG, and GPZDA log outputs.

The default GNSS NMEA message (`nmeatalker GP`) outputs GP as the talker ID regardless of the position type given in position logs such as BESTPOS. The `nmeatalker auto` command switches the talker ID between GP and IN according to the position type given in position logs.

Abbreviated ASCII Syntax: **Message ID: 861**

NMEATALKER [ID]

Factory Default:

nmeatalker gp

Abbreviated ASCII Example:

NMEATALKER AUTO

This command only affects NMEA logs that are capable of an INS position and/or velocity output. For example, GPGSV is for information on GNSS satellites and its output always uses the GP ID. *Table 25* shows the NMEA logs and whether they use GP or GP + IN IDs with `nmeatalker auto`.

Table 25: NMEA Talkers

Log	GPALM	GPGGA	GPGLL	GPGRS	GPGSA	GPGST	GPGSV	GPRMB	GPRMC	GPVTG	GPZDA
Talker IDs	GP	GP	GP/IN	GP	GP	GP	GP	GP	GP	GP/IN	GP

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	NMEA-TALKER header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	ID	GP	0	GNSS (GP) only	Enum	4	H
		AUTO	1	GNSS and/or Inertial (IN)			

B.4.18 PSRDIFFSOURCE Set the pseudorange correction source

This command lets you identify from which base station to accept differential corrections. This is useful when the receiver is receiving corrections from multiple base stations. See also the RTKSOURCE command on *page 126*.

☒ 1. To use L-band differential corrections, an L-band receiver and a subscription to the OmniSTAR, or use of the free CDGPS, service are required. Contact NovAtel for details, see *page 18*.

2. Since several errors affecting signal transmission are nearly the same for two receivers near each other on the ground, a base at a known location can monitor the errors and generate corrections for the rover to use. This method is called Differential GPS, and is used by surveyors to obtain millimeter accuracy. Major factors degrading GPS signals, which can be removed or reduced with differential methods, are the atmosphere, ionosphere, satellite orbit errors and satellite clock errors. Errors not removed include receiver noise and multipath.

Abbreviated ASCII Syntax: **Message ID: 493**

PSRDIFFSOURCE type ID

Factory Default:

psrdiffsource auto "any"

Abbreviated ASCII Examples:

1. Select only SBAS:
   RTKSOURCE NONE
   PSRDIFFSOURCE SBAS
   SBASCONTROL ENABLE AUTO
2. Enable OmniSTAR VBS, and HP or XP:
   RTKSOURCE OMNISTAR
   PSRDIFFSOURCE OMNISTAR
3. Enable RTK and PSRDIFF from RTCM, with a fall-back to SBAS:
   RTKSOURCE RTCM ANY
   PSRDIFFSOURCE RTCM ANY
   SBASCONTROL ENABLE AUTO

Table 26: DGPS Type

Binary	ASCII	Description
0	RTCM [1] [4]	RTCM ID: 0 ≤ RTCM ID ≤ 1023 **or** ANY
1	RTCA [1] [4]	RTCA ID: A four character string containing only alpha (a-z) or numeric characters (0-9) **or** ANY
2	CMR [1] [2] [4]	CMR ID: 0 ≤ CMR ID ≤ 31 **or** ANY
3	OMNISTAR [3] [4]	In the PSRDIFFSOURCE command, OMNISTAR enables OmniSTAR VBS and disables other DGPS types. OmniSTAR VBS produces RTCM-type corrections. In the RTKSOURCE command, OMNISTAR enables OmniSTAR HP/XP (if allowed) and disables other RTK types. OmniSTAR HP/XP has its own filter, which computes corrections in RTK float mode or within about 10 cm accuracy.
4	CDGPS [3] [4]	In the PSRDIFFSOURCE command, CDGPS enables CDGPS and disables other DGPS types. CDGPS produces SBAS-type corrections. **Do not set CDGPS in the RTKSOURCE command as it can not provide carrier phase positioning and disallows all other sources of RTK information**.
5	SBAS [3] [4]	In the PSRDIFFSOURCE command, when enabled, SBAS, such as WAAS, EGNOS and MSAS, forces the use of SBAS as the pseudorange differential source. SBAS is able to simultaneously track two SBAS satellites, and incorporate the SBAS corrections into the position to generate differential-quality position solutions. An SBAS-capable receiver permits anyone within the area of coverage to take advantage of its benefits. Do not set SBAS in the RTKSOURCE command as it can not provide carrier phase positioning and disallows all other sources of RTK information.
10	AUTO [3] [4]	In the PSRDIFFSOURCE command, AUTO means the first received RTCM or RTCA message has preference over an L-band message. In the RTKSOURCE command, AUTO means that both the NovAtel RTK filter and the OmniSTAR HP/XP filter (if authorized) are enabled. The NovAtel RTK filter selects the first received RTCM, RTCA, RTCMV3 or CMR message. The BESTPOS log selects the best solution between NovAtel RTK and OmniSTAR HP/XP.
11	NONE [3] [4]	Disables all the DGPS and OMNISTAR types
12	Reserved	
13	RTCMV3 [2]	RTCM Version 3.0 ID: 0 ≤ RTCMV3 ID ≤ 4095 **or** ANY

1. Disables L-band Virtual Base Stations (VBS)
2. Available only with the RTKSOURCE command, see *page 126*
3. ID parameter is ignored
4. All PSRDIFFSOURCE entries fall back to SBAS (even NONE) for backwards compatibility

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	PSRDIFFSOURCE header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	type	See *Table 26* on *page 121*		ID Type. All types may revert to SBAS (if enabled) or SINGLE position types. See also *Table 38, Position or Velocity Type,* on *page 172.* [1]	Enum	4	H
3	ID	Char [5] or ANY		ID string	Char[5]	8 [2]	H+4

1. If you choose ANY, the receiver ignores the ID string. Specify a Type when you are using base station IDs.
2. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment

B.4.19 RESET Perform a hardware reset

This command performs a hardware reset. Following a RESET command, the receiver initiates a cold-start boot up. Therefore, the receiver configuration reverts either to the factory default, if no user configuration was saved, or the last SAVECONFIG settings. See also the FRESET command on *page 105*.

The optional delay field is used to set the number of seconds the receiver is to wait before resetting.

☒ The RESET command can be used to erase any unsaved changes to the receiver configuration.

Abbreviated ASCII Syntax: **Message ID: 18**

RESET [delay]

Abbreviated ASCII Example

RESET 120

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RESET header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	delay			Seconds to wait before resetting. (default = 0)	Ulong	4	H

B.4.20 RTKCOMMAND Reset or set the RTK filter to its defaults

This command provides the ability to reset the RTK filter and clear any set RTK parameters. The RESET parameter causes the AdVance RTK algorithm to undergo a complete reset, forcing the system to restart the ambiguity resolution calculations. The USE_DEFAULTS command executes the following commands:

```
RTKDYNAMICS DYNAMIC
RTKSVENTRIES 12
```

Abbreviated ASCII Syntax: **Message ID: 97**

RTKCOMMAND action

Factory Default:

rtkcommand use_defaults

Abbreviated ASCII Example:

RTKCOMMAND RESET

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKCOMMAND header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	type	USE_DEFAULTS	0	Reset to defaults	Enum	4	H
		RESET	1	Reset RTK algorithm			

B.4.21 RTKDYNAMICS Set the RTK dynamics mode

This command provides the ability to specify how the receiver looks at the data. There are three modes: STATIC, DYNAMIC, and AUTO. The STATIC mode forces the RTK software to treat the rover station as though it were stationary, regardless of the output of the motion detector.

DYNAMIC forces the software to treat the receiver as though it were in motion. If the receiver is undergoing very slow steady motion (< 2.5 cm/s for more than 5 seconds), you should use DYNAMIC mode (as opposed to AUTO) to prevent inaccurate results and possible resets.

On start-up, the receiver defaults to the DYNAMIC setting.

---

1. For reliable performance, the antenna should not move more than 1-2 cm when in static mode.
2. Use the static option to decrease the time required to fix ambiguities and reduce the amount of noise in the position solution. If you use STATIC mode when the antenna is not static, the receiver will have erroneous solutions and unnecessary RTK resets.

Abbreviated ASCII Syntax: **Message ID: 183**

RTKDYNAMICS mode

Factory Default:

rtkdynamics dynamic

Abbreviated ASCII Example:

RTKDYNAMICS STATIC

Table 27: Dynamics Mode

ASCII	Binary	Description
AUTO	0	Automatically determine dynamics mode
STATIC	1	Static mode
DYNAMIC	2	Dynamic mode

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKDYNAMICS header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	mode	See *Table 27*		Set the dynamics mode	Enum	4	H

B.4.22 RTKSOURCE Set the RTK correction source

This command lets you identify from which base station to accept RTK (RTCM, RTCMV3, RTCA, CMR and OmniSTAR (HP/XP)) differential corrections. This is useful when the receiver is receiving corrections from multiple base stations. See also the PSRDIFFSOURCE command on *page 120*. To set up RTK differential corrections, see the GNSSCARDCONFIG command on *page 106*.

☒ To use OmniSTAR HP/XP differential corrections, a NovAtel receiver with L-band capability and a subscription to the OmniSTAR service are required. Contact NovAtel for details. Contact information may be found on the back of this manual or you can refer to the *Customer Service* section in the *OEMV Family Installation and Operation User Manual*.

Abbreviated ASCII Syntax: **Message ID: 494**

RTKSOURCE type ID

Factory Default:

rtksource auto "any"

Abbreviated ASCII Examples:

1. Specify the format before specifying the base station IDs:
   RTKSOURCE RTCMV3 5
   RTKSOURCE RTCM 6

☒ The RTKSOURCE command supports both RTCM and RTCMV3 while the PSRDIFFSOURCE commands supports only RTCM.

---

2. Select only SBAS:
   RTKSOURCE NONE
   PSRDIFFSOURCE NONE
   SBASCONTROL ENABLE AUTO

3. Enable OmniSTAR HP and VBS:
   RTKSOURCE OMNISTAR
   PSRDIFFSOURCE OMNISTAR

4. Enable RTK and PSRDIFF from RTCM, with a fall-back to SBAS:
   RTKSOURCE RTCM ANY
   PSRDIFFSOURCE RTCM ANY
   SBASCONTROL ENABLE AUTO

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	RTKSOURCE header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	type	See *Table 26, DGPS Type,* on *page 121*		ID Type [1]	Enum	4	H
3	ID	Char [5] or ANY		ID string	Char[5]	8 [2]	H+4

1. If you choose ANY, the receiver ignores the ID string. Specify a Type when you are using base station IDs.
2. In the binary log case, an additional 3 bytes of padding are added to maintain 4-byte alignment.

B.4.23 RVBCALIBRATE Vehicle to Body Rotation Control

The RVBCALIBRATE command is used to enable or disable the calculation of the vehicle frame to the SPAN computation frame angular offset. These angular offsets must be known in the SPAN system before a kinematic alignment can be attempted. The angular offset can be entered with the VEHICLEBODYROTATION command, or solved for with the RVBCALIBRATE command. This command should be entered when the IMU is re-mounted in the vehicle or if the rotation angles available are known to be incorrect.

WARNING: **After the RVBCALIBRATE ENABLE command is entered, there are no vehicle-body rotation parameters present and a kinematic alignment is NOT possible. Therefore this command should only be entered after the system has performed either a static or kinematic alignment and has a valid INS solution.**

A good INS solution and vehicle movement are required for the SPAN system to solve the vehicle-SPAN body offset. The solved vehicle-body rotation parameters are output in the VEHICLEBODYROTATION log when the calibration is complete, see *page 269*. When the calibration is done, the rotation values are fixed until the calibration is re-run by entering the RVBCALIBRATE command again.

☒ The solved rotation values are used only for a rough estimate of the angular offsets between the IMU and vehicle frames. The offsets are used when aligning the system while in motion (see *Section 3.4.1, Configuration for Alignment* starting on *page 45*). The angular offset values are not applied to the attitude output, unless the APPLYVEHICLEBODYROTATION command is disabled.

Abbreviated ASCII Syntax: **Message ID: 641**

RVBCALIBRATE reset

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Log Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	Switch	RESET	0	Control the vehicle/ body rotation computation	ENUM	4	H
		DISABLE	1				
		ENABLE	2				

Abbreviated ASCII Example:

```
RVBCALIBRATE RESET
```

B.4.24 SAVECONFIG Save current configuration in NVM

This command saves the user's present configuration in non-volatile memory. The configuration includes the current log settings, FIX settings, port configurations, and so on. Its output is in the RXCONFIG log, see *page 254*. See also the FRESET command, *page 105*.

WARNING!: **If you are using this command in CDU, ensure that you have all windows other than the Console window closed. Otherwise, log commands used for the various windows are saved as well. This will result in unnecessary data being logged.**

Abbreviated ASCII Syntax: **Message ID: 19**

SAVECONFIG

B.4.25 SBASCONTROL Set SBAS test mode and PRN

This command allows you to dictate how the receiver handles Satellite Based Augmentation System (SBAS) corrections. The receiver automatically switches to Pseudorange Differential (RTCM or RTCA) or RTK if the appropriate corrections are received, regardless of the current setting.

To enable the position solution corrections, you must issue the SBASCONTROL ENABLE command. The receiver does not attempt to track any GEO satellites until you use the SBASCONTROL command to tell it to use either WAAS, EGNOS, or MSAS corrections. DISABLE stops the corrections from being used.

When in AUTO mode, if the receiver is outside the defined satellite system's corrections grid, it reverts to ANY mode and chooses a system based on other criteria.

Once tracking satellites from one system in ANY or AUTO mode, it does not track satellites from other systems. This is because systems such as WAAS, EGNOS and MSAS do not share broadcast information and have no way of knowing each other are there.

The "testmode" parameter in the example is to get around the test mode of these systems. EGNOS at one time used the IGNOREZERO test mode. At the time of printing, ZEROTOTWO is the correct setting for all SBAS, including EGNOS, running in test mode. On a simulator, you may want to leave this parameter off or specify NONE explicitly.

When you use the SBASCONTROL command to direct the GNSS receiver to use a specific correction type, the GNSS receiver begins to search for and track the relevant GEO PRNs for that correction type only. You can force the GNSS receiver to track a specific PRN using the ASSIGN command. You can force the GNSS receiver to use the corrections from a specific SBAS PRN using the SBASCONTROL command.

Abbreviated ASCII Syntax: **Message ID: 652**

SBASCONTROL keyword [system] [prn] [testmode]

Factory Default:

sbascontrol disable auto 0 none

Abbreviated ASCII Example 1:

SBASCONTROL ENABLE WAAS 0 ZEROTOTWO

Table 28: System Types

ASCII	Binary	Description
NONE	0	Don't use any SBAS satellites
AUTO	1	Automatically determine satellite system to use (default)
ANY	2	Use any and all SBAS satellites found
WAAS	3	Use only WAAS satellites
EGNOS	4	Use only EGNOS satellites
MSAS	5	Use only MSAS satellites

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SBASCONTROL header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	keyword	DISABLE	0	Receiver does not use the SBAS corrections it receives	Enum	4	H
		ENABLE	1	Receiver uses the SBAS corrections it receives			
3	system	See *Table 28* on *page 130*		Choose the SBAS the receiver will use	Enum	4	H+4
4	prn	0		Receiver uses any PRN (default)	ULong	4	H+8
		120-138		Receiver uses SBAS corrections only from this PRN			
5	testmode	NONE	0	Receiver interprets Type 0 messages as they are intended (as do not use) (default)	Enum	4	H+12
		ZEROTOTWO	1	Receiver interprets Type 0 messages as Type 2 messages			
		IGNOREZERO	2	Receiver ignores the usual interpretation of Type 0 messages (as do not use) and continues			

B.4.26 SETAUTOLOGGING Start SD Card Logging at Boot-Up

This command is used to enable and disable SD Card auto logging at boot-up. If you have already used the SAVECONFIG command for some logs on the FILE port that you wish to start logging on automatically, this command enables the SD Card and opens a file for writing immediately after the card is mounted and ready for use (even before the rest of the system is ready).

For example, enter LOG FILE RANGEA ONTIME 1 followed by SAVECONFIG. If you also enter SETAUTOLOGGING ON, a file is created and RANGEA logs are recorded automatically after each system boot-up or restart. If the logs are requested but SETAUTOLOGGING is OFF, nothing is written to the card. Similarly, if SETAUTOLOGGING is ON but no logs to the FILE port have been requested, no data is written to the card but a blank file is created.

The user can still type LOGFILE CLOSE at any time to stop logging to the file whether it was opened for writing manually (using LOGFILE OPEN) or automatically (using SETAUTOLOGGING ON). Since data is being recorded immediately at boot-up, some early output will have invalid GPS TIME and other potential error or warning bits (for example indicating invalid position or almanac). When the system is running, this should correct itself.

Abbreviated ASCII Syntax: **Message ID = 1129**

SETAUTOLOGGING switch

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	switch	OFF	0	Enable or disable auto logging on boot-up	Enum	4	H
		ON	1				

Abbreviated ASCII Example:

SETAUTOLOGGING ON

B.4.27 SETETHPROTOCOL Set Eth1 Protocol

The SPAN-SE has a 10/100 RJ-45 Ethernet port, which has a MAC address hard coded into flash and user-configurable IP information. Port 3000 can be used for both TCP and UDP traffic but not simultaneously. You must configure the system for either UDP or TCP communication and the **system must be restarted**. The default is TCP.

To configure the ETH1 transport protocol, use the SETETHPROTOCOL command with its one non-optional parameter.

IMPORTANT!: **You must manually reset the system for this setting to take effect using the RESET command or a power cycle. See also the RESET command on *page 123*.**

Abbreviated ASCII Syntax **Message ID = 1128**

SETETHPROTOCOL IPProtocol

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	IPProtocol	UDP	0	User Datagram Protocol	Enum	4	H
		TCP	1	Transport Control Protocol (default)			

Abbreviated ASCII Example:

```
SETETHPROTOCOL UDP
RESET
```

B.4.28 SETIMUORIENTATION Set IMU Orientation

The SETIMUORIENTATION command is used to specify which of the IMU axis is aligned with gravity. The IMU orientation can be saved using the SAVECONFIG command so that on start-up, the SPAN system does not have to detect the orientation of the IMU with respect to gravity. This is particularly useful for situations where the receiver is powered while in motion.

---

1. The default IMU axis definitions are:

    Y - forward

    Z - up

    X - out the right hand side

    It is strongly recommended that you mount your IMU in this way with respect to the vehicle.

2. You only need to use this command if the system is to be aligned while in motion using the fast alignment routine, see *Section 3.4.3.3, Manual Alignment* on *page 47*.

WARNING: Ensure that all windows, other than the Console, are closed in **CDU** and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload your system.

This orientation command serves to transform the incoming IMU signals in such a way that a 5 mapping is achieved, see *Table 29* on *page 136*. For example, if the IMU is mounted with the X-axis pointing UP and a mapping of 1 is specified then this transformation of the raw IMU data is done:

X⇒Z, Y⇒X, Z⇒Y (where the default is X⇒X, Y⇒Y, Z⇒Z)

Notice that the X-axis observations are transformed into the Z axis, resulting in Z being aligned with gravity and a 5 mapping. The SPAN frame is defined so that Z is always pointing up along the gravity vector. If the IMU mapping is set to 1, the X axis of the IMU enclosure is mapped to the SPAN frame Z axis (pointing up), its Y axis to SPAN frame X and its Z axis to SPAN frame Y.

The X (pitch), Y (roll) and Z (azimuth) directions of the inertial enclosure frame are clearly marked on the IMU, see the IMU choices and their technical specifications starting on *page 62.* The example from the LN-200 is shown in *Figure 33*.

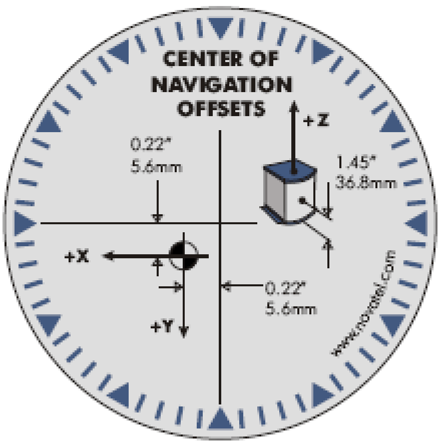


Figure 33: Frame of Reference

1. Azimuth is positive in a clockwise direction while yaw is positive in a counter-clockwise direction when looking toward the axis origin. Yaw follows the right-handed system convention where as azimuth follows the surveying convention.
2. The data in the RAWIMUS log is never mapped. The axes referenced in the RAWIMUS log description form the IMU enclosure frame (as marked on the enclosure).

Abbreviated ASCII Syntax: **Message ID: 567**

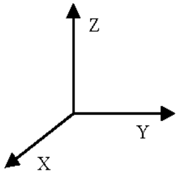
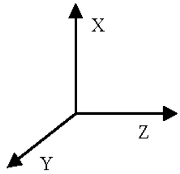
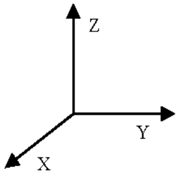
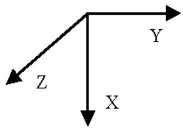
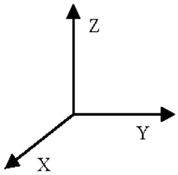
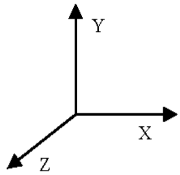
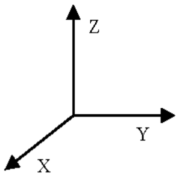
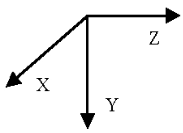
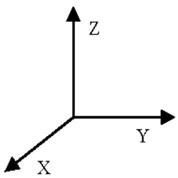
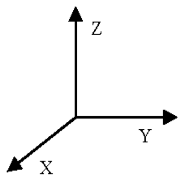
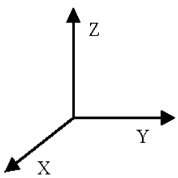
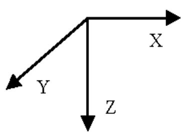
SETIMUORIENTATION switch

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Log Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	Switch	0	0	IMU determines axis orientation automatically during coarse alignment. (default)	ENUM	4	H
		1	1	IMU **X** axis is pointing **UP**			
		2	2	IMU **X** axis is pointing **DOWN**			
		3	3	IMU **Y** axis is pointing **UP**			
		4	4	IMU **Y** axis is pointing **DOWN**			
		5	5	IMU **Z** axis is pointing **UP**			
		6	6	IMU **Z** axis is pointing **DOWN**			

Abbreviated ASCII Example:

SETIMUORIENTATION 1

Table 29: Full Mapping Definitions

Mapping	SPAN Frame Axes	SPAN Frame	IMU Enclosure Frame Axes	IMU Enclosure Frame
1	X		Y	
	Y		Z	
	Z		X	
2	X		Z	
	Y		Y	
	Z		-X	
3	X		Z	
	Y		X	
	Z		Y	
4	X		X	
	Y		Z	
	Z		-Y	
5 (default)	X		X	
	Y		Y	
	Z		Z	
6	X		Y	
	Y		X	
	Z		-Z	

B.4.29 SETIMUTOANTOFFSET Set IMU to antenna offset

It is recommended that you mount the IMU as close as possible to the GNSS antenna, particularly in the horizontal plane. This command is used to enter the offset between the IMU and the GNSS antenna. The measurement should be done as accurately as possible, preferably to within millimeters especially for RTK operation. The x, y and z fields represent the vector from the IMU to the antenna phase center in the IMU enclosure frame. The a, b and c fields allow you to enter any possible errors in your measurements. If you think that your 'x' offset measurement is out by a centimeter for example, enter 0.01 in the 'a' field.

The X (pitch), Y (roll) and Z (azimuth) directions of the inertial frame are clearly marked on the IMU.

This command must be entered before or during the INS alignment mode (not after).

Abbreviated ASCII Syntax: **Message ID: 383**

SETIMUTOANTOFFSET x y z [a] [b] [c]

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	x	± 20		x offset (m)	Double	8	H
3	y	± 20		y offset (m)	Double	8	H+8
4	z	± 20		z offset (m)	Double	8	H+16
5	a	0 to +1		Uncertainty in x (m) (Defaults to 10% of the x offset to a minimum of 0.01 m)	Double	8	H+24
6	b	0 to +1		Uncertainty in y (m) (Defaults to 10% of the y offset to a minimum of 0.01 m)	Double	8	H+32
7	c	0 to +1		Uncertainty in z (m) (Defaults to 10% of the z offset to a minimum of 0.01 m)	Double	8	H+40

Abbreviated ASCII Example:

SETIMUTOANTOFFSET 0.54 0.32 1.20 0.03 0.03 0.05

B.4.30 SETIMUTYPE Set IMU type

The SETIMUTYPE command is used to specify the type of IMU connected to the receiver. The IMU type can be saved using the SAVECONFIG command.

WARNING: Ensure that all windows, other than the Console, are closed in **CDU** and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload your system.

Abbreviated ASCII Syntax: **Message ID: 569**

SETIMUTYPE switch

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Log Header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	Switch	See *Table 30, IMU Type,* on *page 139*		IMU Type	ENUM	4	H

Table 30: IMU Type

Binary	ASCII	Description
0	IMU_UNKNOWN	Unknown IMU type (default)
1	IMU_HG1700_AG11	Honeywell HG1700 AG11/AG58
2-3	Reserved	
4	IMU_HG1700_AG17	Honeywell HG1700 AG17/AG62
5-7	Reserved	
8	IMU_LN200	Litton LN-200 (200 Hz model)
9	IMU_LN200_400HZ	Litton LN-200 (400 Hz model)
10	IMU_IMAR_FSAS	iMAR iIMU-FSAS
11	IMU_HG1700_AG58	Honeywell HG1700 AG58
12	IMU_HG1700_AG62	Honeywell HG1700 AG62

Abbreviated ASCII Example:

SETIMUTYPE IMU_IMAR_FSAS

B.4.31 SETINITATTITUDE Set initial attitude of SPAN in degrees

This command allows you to input a known attitude to start SPAN operation, rather than the usual coarse alignment process. The caveats and special conditions of this command are listed below:

- This alignment is instantaneous based on the user input. This allows for faster system startup; however, the input values must be accurate or SPAN will not perform well.
- If you are uncertain about the standard deviation of the angles you are entering, err on the side of a larger standard deviation.
- Sending SETINITATTITUDE resets the SPAN filter. The alignment is instantaneous, but some time and vehicle dynamics are required for the SPAN filter to converge. Bridging performance is poor before filter convergence.
- The roll (about the y-axis), pitch (about the x-axis), and azimuth (about the z-axis) are with respect to the SPAN frame. If the IMU enclosure is mounted with the z axis pointing upwards, the SPAN frame is the same as the markings on the enclosure. If the IMU is mounted in another way, SPAN transforms the SPAN frame axes such that z points up for SPAN computations. You must enter the angles in SETINITATTITUDE with respect to the transformed axis. See SETIMUORIENTATION for a description of the axes mapping that occurs when the IMU is mounted differently from z up.
- This command is not save configurable (see the SAVECONFIG command on *page 129*) and, if needed, must be entered at startup.

✉ 1. Azimuth is positive in a clockwise direction when looking towards the z-axis origin.

2. You do not have to use the SETIMUORIENTATION command, see *page 134,* unless you have your IMU mounted with the z axis not pointing up. Then use the tables in the SETIMURIENTATION command, on *pages 135-136*, to determine the azimuth axis that SPAN is using.

Abbreviated ASCII Syntax: **Message ID: 862**

SETINITATTITUDE pitch roll azimuth pitchSTD rollSTD azSTD

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	pitch	-360° to +360°		Input pitch angle, about the x-axis, in degrees	Double	8	H
3	roll	-360° to +360°		Input roll angle, about the y-axis, in degrees	Double	8	H+8
4	azimuth	-360° to +360°		Input azimuth angle, about the z-axis, in degrees	Double	8	H+16
5	pitchSTD	0.000278° [1] to 180°		Input pitch standard deviation (STD) angle in degrees	Double	8	H+24
6	rollSTD			Input roll STD angle in degrees	Double	8	H+32
7	azSTD			Input azimuth STD angle in degrees	Double	8	H+40

1. 0.000278° is equal to 1 arc second

Abbreviated ASCII Example:

SETINITATTITUDE 0 0 90 5 5 5

In this example, the initial roll and pitch has been set to zero degrees, with a standard deviation of 5 degrees for both. This means that the SPAN system is very close to level with respect to the local gravity field. The azimuth is 90 degrees (see the SETINITAZIMUTH example on *page 142*), also with a 5 degrees standard deviation.

B.4.32 SETINITAZIMUTH Set initial azimuth and standard deviation

This command allows you to start SPAN operation with a previously known azimuth. Azimuth is the weakest component of a coarse alignment, and is also the easiest to know from an external source (i.e. like the azimuth of roadway). When using this command, SPAN operation through alignment will appear the same as with a usual coarse alignment. Roll and pitch will be determined using averaged gyro and accelerometer measurements. The input azimuth will be used rather than what is computed by the normal coarse alignment routine.

- This alignment takes the same amount of time as the usual coarse alignment (60 s nominally).
- Input azimuth values must be accurate for good system performance.
- Sending SETINITAZIMUTH resets the SPAN filter. The alignment will take approximately 1 minute, but some time and vehicle dynamics are required for the SPAN filter to converge. Bridging performance will be poor before filter convergence.
- The azimuth angle is with respect to the SPAN frame. If the IMU enclosure is mounted with the z axis pointing upwards, the SPAN frame is the same as what is marked on the enclosure. If the IMU is mounted in another way, SPAN transforms the SPAN frame axes such that z points up for SPAN computations. You must enter the azimuth with respect to the transformed axis. See SETIMUORIENTATION on *page 134*, for a description of the axes mapping that occurs when the IMU is mounted differently from z pointing up.
- This command is not save configurable (see the SAVECONFIG command on Page 130) and, if needed, must be entered at startup.

1. Azimuth is positive in a clockwise direction when looking towards the z-axis origin.
2. You do not have to use the SETIMUORIENTATION command, see *page 134,* unless you have your IMU mounted with the z axis not pointing up. Then, use the tables in the SETIMURIENTATION command, on *pages 135-136*, to determine the azimuth axis that SPAN is using.

Abbreviated ASCII Syntax: **Message ID: 863**

SETINITAZIMUTH azimuth azSTD

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	azimuth	-360° to +360°		Input azimuth angle in degrees	Double	8	H
3	azSTD	0.000278° to 180°		Input azimuth standard deviation angle in degrees	Double	8	H+8

Abbreviated ASCII Example:

SETINITAZIMUTH 90 5

In this example, the initial azimuth has been set to 90 degrees. This means that the SPAN system Y axis is pointing due East, within a standard deviation of 5 degrees. Note that if you have mounted your SPAN system with the positive Z axis (as marked on the enclosure) in a direction that is not up, please refer to the SETIMUORIENTATION command to determine the SPAN computation frame axes mapping that SPAN automatically applies.

B.4.33 SETINSOFFSET Set INS offset

The SETINSOFFSET command is used to specify an offset from the IMU for the output position and velocity of the INS solution. This command shifts the position and velocity in the INSPOS, INSPOSS, INSVEL, INSVELS, INSSPD, INSSPDS, INSPVA and INSPVAS logs by the amount specified in metres with respect to the IMU enclosure frame axis.

Abbreviated ASCII Syntax: **Message ID: 676**

SETINSOFFSET xoffset yoffset zoffset

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	X offset	± 100		Offset along the IMU enclosure frame X axis (m)	Double	8	H
3	Y offset	± 100		Offset along the IMU enclosure frame Y axis (m)	Double	8	H+8
4	Z offset	± 100		Offset along the IMU enclosure frame Z axis (m)	Double	8	H+16

Abbreviated ASCII Example:

SETINSOFFSET 0.15 0.15 0.25

B.4.34 SETMARK1OFFSET, SETMARK2OFFSET, SETMARK3OFFSET, SETMARK4OFFSET Set Mark offset

Set the offset to the Mark1, Mark2, Mark3 or Mark4 trigger event. See also the MARK1PVA to MARK4PVA logs on *page 236*. The X, Y, Z offset is measured from the IMU to the asked location, in the IMU enclosure frame.

Abbreviated ASCII Syntax: **Message ID: 1069**

SETMARK1OFFSET xoffset yoffset zoffset αoffset βoffset γoffset

Abbreviated ASCII Syntax: **Message ID: 1070**

SETMARK2OFFSET xoffset yoffset zoffset αoffset βoffset γoffset

Abbreviated ASCII Syntax: **Message ID: 1116**

SETMARK3OFFSET xoffset yoffset zoffset αoffset βoffset γoffset

Abbreviated ASCII Syntax: **Message ID: 1117**

SETMARK4OFFSET xoffset yoffset zoffset αoffset βoffset γoffset

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	x offset	± 360		Offset along the IMU enclosure frame X axis (m) for Mark1, 2, 3 or 4	Double	8	H
3	y offset	± 360		Offset along the IMU enclosure frame Y axis (m) for Mark1, 2, 3 or 4	Double	8	H+8
4	z offset	± 360		Offset along the IMU enclosure frame Z axis (m) for Mark1, 2, 3 or 4	Double	8	H+16
5	αoffset	± 360		Roll offset for Mark (degrees)	Double	8	H+24
6	βoffset	± 360		Pitch offset for Mark (degrees)	Double	8	H+32
7	γoffset	± 360		Azimuth offset for Mark (degrees)	Double	8	H+40

Abbreviated ASCII Example:

SETMARK1OFFSET -0.324 0.106 1.325 0 0 0

B.4.35 SETWHEELPARAMETERS Set wheel parameters

The SETWHEELPARAMETERS command can be used when wheel sensor data is available. It allows you to give the filter a good starting point for the wheel size scale factor. It also gives the SPAN-SE filter an indication of the expected accuracy of the wheel data.

Usage of the SETWHEELPARAMETERS command depends on which method is used to communicate to the wheel sensor (see section 3.3.8)

1. If you have integrated an external wheel sensor, the SETWHEELPARAMETERS command can be used to override the number of ticks per revolution given in the WHEELVELOCITY command. If this command is not entered, the default wheel circumference of 1.96 meters is used. In addition, this command supplies the resolution of the wheel sensor, which allows the filter to weight the wheel sensor data appropriately, as in:

   ```
   SETWHEELPARAMETERS 1000 2.03 0.002
   ```

2. If you have an external wheel sensor that will be connected to an EVENT line on the SPAN-SE, then the SETWHEELPARAMETERS command **must** be sent in order to select which MARK to use The wheel parameters **must** also be specified here as the default values will not be used. The two optional parameters in the command are specifically for this mode of operation. For example, if you had a wheel sensor attached to the first EVENT IN (MARK1) with a tick provided with positive polarity, the command would look like:

   ```
   SETWHEELPARAMETERS MARK1 POSITIVE 1000 2.03 0.002
   ```

3. If you are using a wheel sensor connected directly to the iMAR iIMU-FSAS, the SETWHEELPARAMETERS command allows you to set the number of ticks per revolution that is correct for your wheel installation (the default is 58). The default wheel circumference is 1.96 meters. The input type for this mode should be 'IMU' and the polarity is unused.

   ```
   SETWHEELPARAMETERS IMU 1000 2.03 0.002
   ```

Abbreviated ASCII Syntax: **Message ID: 847**

SETWHEELPARAMETERS [input] [polarity] ticks circ spacing

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	input	See *Table 31* on *page 147*		Optional field to specify to which input the command should be applied. (default = IMU)	Enum	4	H
3	polarity	NEGATIVE	0	Optional field to specify the polarity of the pulse to be received on the mark input. (default = POSITIVE)	Enum	4	H+4
		POSITIVE	1				
4	ticks	1-10 000		Number of ticks per revolution	Ushort	4 [1]	H+8
5	circ	0.1-100		Wheel circumference (m)	Double	8	H+12
6	spacing	0.001-1000		Spacing of ticks, or resolution of the wheel sensor (m)	Double	8	H+20

1. In the binary log case, an additional 2 bytes of padding are added to maintain 4-byte alignment.

Table 31: SETWHEELPARAMETERS Input

Binary	ASCII
0	IMU (default)
1	MARK1
2	MARK2
3	MARK3
4	MARK4

Abbreviated ASCII Example:

SETWHEELPARAMETERS 58 1.96 0.025

Fields 2, 3 and 4 do not have to ‘add up’. Field 4 is used to weight the wheel sensor measurement. Fields 2 and 3 are used with the estimated scale factor to determine the distance travelled.

B.4.36 SOFTPOWER Power down the SPAN-SE

Use the SOFTPOWER command to power down the SPAN-SE. This command is meant for automated setups where the user may not be able to physically touch the SPAN-SE but needs to shut the system down.

Abbreviated ASCII Syntax: **Message ID: 213**

SOFTPOWER priority

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	priority	NOW	1	Power down the SPAN-SE immediately	Enum	4	H

Abbreviated ASCII Example:

SOFTPOWER NOW

B.4.37 SPANAUTH Add an authorization code for a new model

This command is used to add or remove authorization codes from the receiver. Authorization codes are used to authorize models of software for a receiver. The receiver is capable of keeping track of 5 authorization codes at one time. The SPANVALIDMODELS log, see *page 265*, lists the current available models in the receiver. This simplifies the use of multiple software models on the same receiver.

If there is more than one valid model in the receiver, the receiver uses the model of the last `spanauth` code entered via the SPANAUTH command. The SPANAUTH command causes a reset automatically.

To change models on the internal OEMV-3, use the AUTH, MODEL and VALIDMODELS commands defined in the *OEMV Family Firmware Reference Manual*. We recommend that you contact *NovAtel Customer Service* for assistance in doing this, see *page 18* or *Note #2* below.

---

1. Authorization codes are firmware version specific. If the receiver firmware is updated, it is necessary to acquire new SPAN authorization codes for the required models. If you wish to update the firmware in the receiver, please contact *NovAtel Customer Service*.
2. When you want to easily upgrade your SPAN-SE receiver, or its internal OEMV-3, without returning your SPAN-SE to the factory, our unique field-upgradeable feature allows you to buy the equipment that you need today, and upgrade them without facing obsolescence.

   When you are ready to upgrade from one model to another, call 1-800-NOVATEL to speak with our Customer Service/Sales Personnel, who can provide the SPAN authorization code that unlocks the additional features of your SPAN-SE receiver. This procedure can be performed at your work-site and takes only a few minutes.

WARNING!: **Removing a SPAN authorization code will cause the receiver to permanently lose this information.**

Abbreviated ASCII Syntax: **Message ID: 1086**

SPANAUTH [state] part1 part2 part3 part4 part5 model [date]

Abbreviated ASCII Examples:

SPANAUTH ADD 1234 5678 9ABC DEF0 1234 SJ 100131

SPANAUTH 1234 5678 9ABC DEF0 1234 SJ

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SPAN-AUTH header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	state	REMOVE	0	Remove the SPAN authcode from the system.	Enum	4	H
		ADD	1	Add the SPAN authcode to the system. (default)			
3	part1	4 digit hexadecimal (0-FFFF)		Authorization code section 1.	ULong	4	H+4
4	part2	4 digit hexadecimal (0-FFFF)		Authorization code section 2.	ULong	4	H+8
5	part3	4 digit hexadecimal (0-FFFF)		Authorization code section 3.	ULong	4	H+12
6	part4	4 digit hexadecimal (0-FFFF)		Authorization code section 4.	ULong	4	H+16
7	part5	4 digit hexadecimal (0-FFFF)		Authorization code section 5.	ULong	4	H+20
8	model	Alpha numeric	Null terminated	Model name of the receiver	String [max. 16]	Vari-able [1]	Vari-able
9	date	Numeric	Null terminated	Expiry date entered as yymmdd in decimal.	String [max. 7]	Vari-able [a]	Vari-able

1. In the binary log case, additional bytes of padding are added to maintain 4-byte alignment

B.4.38 SPANMODEL Switch to a previously authorized model

This command is used to switch the receiver between models previously added with the SPANAUTH command, see *page 149*. When this command is issued, the receiver saves this model as the active model. The active model is now used on every subsequent start-up. The SPANMODEL command causes an automatic reset.

Use the SPANVALIDMODELS log to output a list of available models for your receiver. The SPANVALIDMODELS log is described on *page 265*. Use the VERSION log to output the active model, see *page 270.*

⌧ If you switch to an expired model, the receiver will reset and enter into an error state. You will need to switch to a valid model to continue.

Abbreviated ASCII Syntax: **Message ID: 1087**

SPANMODEL model

Input Example:

spanmodel sj

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	SPANMODEL header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	model	Max 16 character null-terminated string (including the null)		SPAN model name	String [max. 16]	Variable[1]	Variable
3	Reserved				Ulong	4	Variable

1. In the binary log case, additional bytes of padding are added to maintain 4-byte alignment

B.4.39 UNLOG Remove a log from logging control

This command permits you to remove a specific log request from the system.

The [*port*] parameter is optional. If [*port*] is not specified, it is defaulted to the port on which the command was received. This feature eliminates the need for you to know which port you are communicating on if you want logs to be removed on the same port as this command.

Abbreviated ASCII Syntax: **Message ID: 36**

UNLOG [port] datatype

Abbreviated ASCII Example:

UNLOG COM1 BESTPOSA

UNLOG BESTPOSA

✉ The UNLOG command allows you to remove one or more logs while leaving other logs unchanged.

Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
1	UNLOG (binary) header	(See *Table 34, Binary Message Header Structure,* on *page 163*)	This field contains the message header.	-	H	0
2	port	See *Table 18, COM Serial Port Identifiers,* on *page 96*	Port to which log is being sent (default = THISPORT)	Enum	4	H
3	message	Any valid message ID	Message ID of log to output	UShort	2	H+4
4	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit 0 = Original Message 1 = Response Message	Message type of log	Char	1	H+6
5	Reserved			Char	1	H+7

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOG (ASCII) header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	port	See *Table 18, COM Serial Port Identifiers,* on *page 96*		Port to which log is being sent (default = THISPORT)	Enum	4	H
3	message	Message Name	N/A	Message Name of log to be disabled	ULong	4	H+4

B.4.40 UNLOGALL Remove all logs from logging control

If [*port*] is specified this command disables all logs on the specified port only. All other ports are unaffected. If [*port*] is not specified this command defaults to the ALL_PORTS setting.

Abbreviated ASCII Syntax: **Message ID: 38**

UNLOGALL [port]

Abbreviated ASCII Example:

UNLOGALL COM2

✉ The UNLOGALL command allows you to remove all log requests currently in use.

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	UNLOGALL header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	port	See *Table 18, COM Serial Port Identifiers,* on *page 96*		Port to clear (default = ALL_PORTS)	Enum	4	H
3	held	FALSE	0	Does not remove logs with the HOLD parameter (default)	Enum	4	H+4
		TRUE	1	Removes previously held logs, even those with the HOLD parameter			

B.4.41 VEHICLEBODYROTATION Vehicle to SPAN frame rotation

Use the VEHICLEBODYROTATION command to set angular offsets between the vehicle frame (direction of travel) and the SPAN computation frame. If you estimate the angular offsets using the RVBCALIBRATE command, the VEHICLEBODYROTATION command values are used as the initial values. The uncertainty values are optional (defaults = 0.0). Please see *Section 3.4.7, Vehicle to SPAN frame Angular Offsets Calibration Routine* starting on *page 49* for more details. For more information on reference frames, see *Section 3.1, Definition of Reference Frames Within SPAN* starting on *page 35*. RVBCALIBRATE command information is on *page 128*.

The VEHICLEBODYROTATION message can be requested as a log and will report whatever the user entered as a command, or the results of the RVBCALIBRATE process, whichever is most recent.

The rotation values are used during kinematic alignment. The rotation is used to transform the vehicle frame attitude estimates from GNSS into the SPAN frame of the IMU during the kinematic alignment. If you use the APPLYVEHICLEBODYROTATION command on *page 91*, the reported attitude in INSPVA or INSATT will be in the vehicle frame; otherwise, the reported attitude will be in the SPAN frame.

The uncertainty values report the accuracy of the angular offsets.

The VEHICLEBODYROTATION command sets the initial estimates for the angular offset. The uncertainty values are optional.

Follow these steps:

1. Start with the SPAN computation frame coincident with the vehicle frame.
2. Rotate about the vehicle Z-axis. This angle is the gamma-angle in the command and follows the right-hand rule for sign correction.
3. Rotate about the new X-axis. This angle is the alpha-angle in the command.
4. Finally, rotate about the new Y-axis. This angle is the beta-angle in the command. The IMU should now be in its mounted position.

☒ Enter rotation angles in degrees.

To apply the vehicle to body rotation angles to the output attitude in the INSPVA or INSATT logs, the APPLYVEHICLEBODYROTATION command needs to be enabled, please refer to *Section B.4.1, APPLYVEHICLEBODYROTATION Enable vehicle to body rotation* starting on *page 91*.

Abbreviated ASCII Syntax: **Message ID: 642**

VEHICLEBODYROTATION alpha beta gamma [δalpha] [δbeta] [δgamma]

Structure: **Log Type: Asynch**

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	X Angle	Right hand rotation about vehicle frame X axis, degrees	Double	8	H
3	Y Angle	Right hand rotation about vehicle frame Y axis, degrees	Double	8	H+8
4	Z Angle	Right hand rotation about vehicle frame Z axis, degrees	Double	8	H+16
5	X Uncertainty	Uncertainty of X rotation, degrees (default = 0)	Double	8	H+24
6	Y Uncertainty	Uncertainty of Y rotation, degrees (default = 0)	Double	8	H+32
7	Z Uncertainty	Uncertainty of Z rotation, degrees (default = 0)	Double	8	H+40
8	xxxx	32-bit CRC	Hex	4	H+48
9	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Refer also to our application note on *Vehicle to Body Rotations*, NovAtel part number APN-037 (available on our Web site at http://www.novatel.com/support/applicationnotes.htm).

Abbreviated ASCII Example:

VEHICLEBODYROTATION 0 0 90 0 0 5

B.4.42 WHEELVELOCITY Wheel velocity for INS augmentation

The WHEELVELOCITY command is used to input wheel sensor data into the OEMV receiver.

Abbreviated ASCII Syntax: **Message ID: 504**

WHEELVELOCITY latency ticks/rev wheel vel Rsrvd fwheel vel Rsrvd Rsrvd ticks/s

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary, respectively.	-	H	0
2	latency			A measure of the latency in the velocity time tag in ms.	Ushort	2	H
3	ticks/rev			Number of ticks per revolution	Ushort	2	H+2
4	wheel vel			Short wheel velocity in ticks/s	Ushort	2	H+4
5	Reserved				Ushort	2	H+6
6	fwheel vel			Float wheel velocity in ticks/s	Float	4	H+8
7	Reserved				Ulong	4	H+12
8					Ulong	4	H+16
9	ticks/s			Cumulative number of ticks/s	Ulong	4	H+20

Refer also to our application note on *Using a Wheel Sensor with SPAN*, NovAtel part number APN-036 (available on our Web site at http://www.novatel.com/support/applicationnotes.htm).

Abbreviated ASCII Example:

WHEELVELOCITY 123 8 10 0 0 0 0 40
WHEELVELOCITY 123 8 10 0 0 0 0 80
WHEELVELOCITY 123 8 10 0 0 0 0 120

The above are for a vehicle traveling at a constant velocity with these wheel sensor characteristics:

Wheel Circumference = 2 m
Vehicle Velocity (assumed constant for this example) = 10 m/s
Ticks Per Revolution = 8
Cumulative Ticks Per Second = (10 m/s)*(8 ticks/rev)/(2 m/rev) = 40
Latency between 1PPS and measurement from wheel sensor hardware =123 ms

1. The ticks per second do not need to be computed as shown in the example above. If your hardware provides the tick count directly, it is not necessary to compute wheel velocity.
2. The wheel velocities in Fields #4 and #6 are not currently used in the SPAN filter. In Inertial Explorer post-processing, wheel velocities may be used. If you wish to use wheel velocities in post-processing, fill Fields #4 and #6 with meaningful values, otherwise, leave as zeroes.

Appendix C Data Logs

The INS-specific logs follow the same general logging scheme as normal OEMV Family logs. They are available in ASCII or binary formats and are defined as being either synchronous or asynchronous. Information on both SPAN-only and selected OEMV logs are contained in this appendix. For information on other available logs and output logging, please refer to the *OEMV Family Firmware Reference Manual*.

One difference from the standard OEMV Family logs is that there are two possible headers for the ASCII and binary versions of the logs. Which header is used for a given log is described in the log definitions in this chapter. The reason for having the alternate short headers is that the normal OEMV-3 binary header is quite long at 28 bytes. This is nearly as long as the data portion of many of the INS logs, and creates excess storage and baud rate requirements. Note that the INS-related logs contain a time tag within the data block in addition to the time tag in the header. The time tag in the data block should be considered the exact time of applicability of the data. All the described INS logs except the INSCOV, INSPOSSYNC, and INSUPDATE can be obtained at rates up to 100 or 200 Hz depending on your IMU, subject to the limits of the output baud rate.

---

1. Each log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters, for example, `*1234ABCD[CR][LF]`. This value is a 32-bit CRC of all bytes in the log, excluding the '#' or '%' identifier and the asterisk preceding the four checksum digits. See also *Section C.2, Description of ASCII and Binary Logs with Short Headers* on *page 168*.
2. The BESTPOS position log can be logged at rates up to 20 Hz directly from the OEMV port, but is available at 1 Hz or 5 Hz from any SPAN-SE port. Other GNSS logs (RANGE, PSRPOS, and so on) can be logged up to 20 Hz from the SPAN ports. The BESTGPSPOS log is available from SPAN-SE only, at 1 Hz or 5 Hz.
3. *Table 5, Inertial Solution Status* on *page 44* shows the status values included in the INS position, velocity and attitude output logs. If you think you have an IMU unit hooked up properly, your GNSS time status is FINESTEERING as shown in the log headers, and you are not getting a good status value, something is wrong and the hardware setup must be checked out. Check the IMU status word in the RAWIMU or RAWIMUS log, and verify that the times in the RAWIMU or RAWIMUs logs are changing over time.

Please also refer to the *OEMV Family Firmware Reference Manual* for information on the supplied Convert4 program that lets you change binary to ASCII data, or short binary to short ASCII data, and vice versa. Convert4 is also capable of RINEX conversions to and from ASCII or binary.

C.1 Log Types

Refer to the LOG command, see *page 113*, for details on requesting logs.

The receiver is capable of generating many different logs. These logs are divided into the following three types: Synchronous, asynchronous, and polled. The data for synchronous logs is generated on a regular schedule. Asynchronous data is generated at irregular intervals. If asynchronous logs were collected on a regular schedule, they would not output the most current data as soon as it was available. The data in polled logs is generated on demand. An example would be RXCONFIG. It would be polled because it changes only when commanded to do so. Therefore, it would not make sense to log this kind of data ONCHANGED, or ONNEW. The following table outlines the log types and the valid triggers to use:

Table 32: Log Type Triggers

Type	Recommended Trigger	Illegal Trigger
Synch	ONTIME	ONNEW, ONCHANGED
Asynch	ONCHANGED	-
Polled	ONCE or ONTIME [1]	ONNEW, ONCHANGED

1. Polled log types do not allow fractional offsets and cannot do ontime rates faster than 1Hz.

See *Section C.1.4, Message Time Stamps* on *page 166* for information on how the message time stamp is set for each type of log.

1. A SPAN-SE user can request up to 25 logs from the OEMV-3, in addition to 30 SPAN-specific logs. If you attempt to log more than 30 logs at a time, the receiver responds with an Insufficient Resources error.

2. Asynchronous logs, such as BESTLEVERARM, should only be logged ONNEW. Otherwise, the most current data is not output when it is available. This is especially true of the ONTIME trigger, which may cause inaccurate time tags to result.

3. Use the ONNEW trigger with the MARK*x*TIME or MARK*x*PVA logs.

Before the output of fields for ASCII and Binary logs, there is an ASCII or binary header respectively. See the ASCII and Binary Sections that follow. There is no header information before Abbreviated ASCII output.

C.1.1 ASCII

ASCII messages are readable by both the user and a computer. The structures of all ASCII messages follow the general conventions as noted here:

1. The lead code identifier for each record is '#'.
2. Each log or command is of variable length depending on amount of data and formats.
3. All data fields are delimited by a comma ',' with two exceptions. The first exception is the last header field which is followed by a ';' to denote the start of the data message. The other exception is the last data field, which is followed by a * to indicate end of message data.
4. Each log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters, for example, *1234ABCD[CR][LF]. This value is a 32-bit CRC of all bytes in the log, excluding the '#' identifier and the asterisk preceding the four checksum digits.
5. An ASCII string is one field and is surrounded by double quotation marks, for example, "ASCII string". If separators are surrounded by quotation marks then the string is still one field and the separator will be ignored, for example, "xxx,xxx" is one field. Double quotation marks within a string are not allowed.
6. If the receiver detects an error parsing an input message, it will return an error response message. Please refer to the *OEMV Firmware Reference Manual* found on our Web site at http://www.novatel.com/support/docupdates.htmfor a list of response messages from the receiver.

Message Structure:

header;	data field...,	data field...,	data field...	*xxxxxxxx	[CR][LF]

The ASCII message header is formatted as follows:

Table 33: ASCII Message Header Structure

Field #	Field Name	Field Type	Description	Ignored on Input
1	Sync	Char	Sync character. The ASCII message is always preceded by a single '#' symbol.	N
2	Message	Char	This is the ASCII name of the log or command	N
3	Port	Char	This is the name of the port from which the log was generated. The string is made up of the port name followed by an _x where x is a number from 1 to 31 denoting the virtual address of the port. If no virtual address is indicated, it is assumed to be address 0.	Y
4	Sequence #	Long	This is used for multiple related logs. It is a number that counts down from N-1 to 0 where 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0.	N
5	% Idle Time	Float	The minimum percentage of time that the processor is idle between successive logs with the same Message ID.	Y
6	GPS Time Status	Enum	This value indicates the quality of the GPS time (see *Table 35, GPS Time Status* on *page 165*)	Y
7	Week	Ulong	GPS week number.	Y
8	Seconds	GPSec	Seconds from the beginning of the GPS week accurate to the millisecond level.	Y
9	Receiver Status	Ulong	This is an eight digit hexadecimal number representing the status of various hardware and software components of the receiver between successive logs with the same Message ID (see *Table 63, SPAN Receiver Status* on *page 259*)	Y
10	Reserved	Ulong	Reserved for internal use.	Y
11	Receiver s/w Version	Ulong	This is a value (0 - 65535) that represents the receiver software build number.	Y
12	;	Char	This character indicates the end of the header.	N

Example Log:

```
#RAWEPHEMA,COM1,0,35.0,SATTIME,1364,496230.000,00100000,97b7,2310;
30,1364,496800,8b0550a1892755100275e6a09382232523a9dc04ee6f794a0000090394ee,8b05
50a189aa6ff925386228f97eabf9c8047e34a70ec5a10e486e794a7a,8b0550a18a2effc2f80061c
2fffc267cd09f1d5034d3537affa28b6ff0eb*7a22f279
```

C.1.2 Binary

Binary messages are meant strictly as a machine readable format. They are also ideal for applications where the amount of data being transmitted is fairly high. Because of the inherent compactness of binary as opposed to ASCII data, the messages are much smaller. This allows a larger amount of data to be transmitted and received by the receiver's communication ports. The structure of all Binary messages follows the general conventions as noted here:

1. Basic format of:

   Header — 3 Sync bytes plus 25 bytes of header information. The header length is variable as fields may be appended in the future. Always check the header length.

   Data — variable

   CRC — 4 bytes

2. The 3 Sync bytes will always be:

Byte	Hex	Decimal
First	AA	170
Second	44	68
Third	12	18

3. The CRC is a 32-bit CRC performed on all data including the header.
4. The header is in the format shown in *Table 34*.

Table 34: Binary Message Header Structure

Field #	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
1	Sync	Char	Hexadecimal 0xAA.	1	0	N
2	Sync	Char	Hexadecimal 0x44.	1	1	N
3	Sync	Char	Hexadecimal 0x12.	1	2	N
4	Header Lgth	Uchar	Length of the header.	1	3	N
5	Message ID	Ushort	This is the Message ID number of the log. Each log has its own unique message ID and you can find as part of each log description in this chapter.	2	4	N
6	Message Type	Char	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response bit 0 = Original Message 1 = Response Message	1	6	N
7	Port Address	Uchar	See *Table 18* on *page 96*	1	7	N [1]
8	Message Length	Ushort	The length in bytes of the body of the message. This does not include the header nor the CRC.	2	8	N
9	Sequence	Ushort	This is used for multiple related logs. It is a number that counts down from N-1 to 0 where N is the number of related logs and 0 means it is the last one of the set. Most logs only come out one at a time in which case this number is 0.	2	10	N

Continued on page 164

Field #	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
10	Idle Time	Uchar	The time that the processor is idle in the last second between successive logs with the same Message ID. Take the time (0 - 200) and divide by two to give the percentage of time (0 - 100%).	1	12	Y
11	Time Status	Enum	Indicates the quality of the GPS time (see *Table 35 on page 165*).	1 [2]	13	N [3]
12	Week	Ushort	GPS week number.	2	14	N [d]
13	ms	GPSec	Milliseconds from the beginning of the GPS week.	4	16	N [d]
14	Receiver Status	Ulong	32 bits representing the status of various hardware and software components of the receiver between successive logs with the same Message ID	4	20	Y
15	Reserved	Ushort	Reserved for internal use.	2	24	Y
16	Receiver S/W Version	Ushort	This is a value (0 - 65535) that represents the receiver software build number.	2	26	Y

1. Recommended value is THISPORT (binary 192)
2. This ENUM is not 4 bytes long but, as indicated in the table, is only 1 byte.
3. These time fields are ignored if Field #11, Time Status, is invalid. In this case the current receiver time is used. The recommended values for the three time fields are 0, 0, 0.

C.1.3 GPS Time Status

All reported receiver times are subject to a qualifying time status. This status gives you an indication of how well a time is known, see *Table 35*:

Table 35: GPS Time Status

GPS Time Status (Decimal)	GPS Time Status [1] (ASCII)	Description
20	UNKNOWN	Time validity is unknown.
60	APPROXIMATE	Time is set approximately.
80	COARSEADJUSTING	Time is approaching coarse precision.
100	COARSE	This time is valid to coarse precision.
120	COARSESTEERING	Time is coarse set, and is being steered.
130	FREEWHEELING	Position is lost, and the range bias cannot be calculated.
140	FINEADJUSTING	Time is adjusting to fine precision.
160	FINE	Time has fine precision.
180	FINESTEERING	Time is fine, set and is being steered.
200	SATTIME	Time from satellite. This is only used in logs containing satellite data such as ephemeris and almanac.

1. See also *Section C.1.4, Message Time Stamps* on *page 166*

There are several distinct states that the receiver goes through:

- UNKNOWN
- COARSE
- FREEWHEELING
- FINE
- FINESTEERING

On start up, and before any satellites are being tracked, the receiver can not possibly know the current time. As such, the receiver time starts counting at GPS week 0 and second 0.0. The time status flag is set to UNKNOWN.

After the first ephemeris is decoded, the receiver time is set to a resolution of ±10 milliseconds. This state is qualified by the COARSE or COARSESTEERING time status flag depending on the state of the CLOCKADJUST switch.

Once a position is known and range biases are being calculated, the internal clock model begins modelling the position range biases and the receiver clock offset.

Modelling continues until the model is a good estimation of the actual receiver clock behavior. At this time, the receiver time will adjusts again, this time to an accuracy of ±1 microsecond. This state is qualified by the FINE time status flag.

If for some reason position is lost and the range bias cannot be calculated, the time status degrades to FREEWHEELING.

C.1.4 Message Time Stamps

NovAtel format messages, generated by the OEMV family receivers, have a GPS time stamp in their header. GPS time is referenced to UTC with zero point defined as midnight on the night of January 5 1980. The time stamp consists of the number of weeks since that zero point and the number of seconds since the last week number change (0 to 604,799). GPS time differs from UTC time since leap seconds are occasionally inserted into UTC but GPS time is continuous. In addition a small error (less than 1 microsecond) can exist in synchronization between UTC and GPS time. The TIME log reports both GPS and UTC time and the offset between the two.

The data in synchronous logs (for example, RANGE, BESTPOS, TIME) are based on a periodic measurement of satellite pseudoranges. The time stamp on these logs is the receiver estimate of GPS time at the time of the measurement. When setting time in external equipment, a small synchronous log with a high baud rate will be accurate to a fraction of a second. A synchronous log with trigger ONTIME 1 can be used in conjunction with the 1PPS signal to provide relative accuracy better than 250 ns.

Other log types (asynchronous and polled) are triggered by an external event and the time in the header may not be synchronized to the current GPS time. Logs that contain satellite broadcast data (for example, ALMANAC, GPSEPHEM) have the transmit time of their last subframe in the header. In the header of differential time matched logs (for example, MATCHEDPOS) is the time of the matched reference and local observation that they are based on. Logs triggered by a mark event (for example, MARK1PVA, MARK1TIME) have the estimated GPS time of the mark event in their header. In the header of polled logs (for example, LOGLIST, PORTSTATS, VERSION) is the approximate GPS time when their data was generated. However, when asynchronous logs are triggered ONTIME, the time stamp will represent the time the log was generated, not the time of validity given in the data.

C.1.5 Log Type Examples

For polled logs, the receiver only supports an offset that is:

- smaller than the logging period
- an integer

The following are valid examples for a polled log:

```
LOG COMCONFIG ONTIME 2 1
LOG PORTSTATS ONTIME 4 2
LOG VERSION ONCE
```

For polled logs, the following examples are invalid:

`LOG COMCONFIG ONTIME 1 2`	[offset is larger than the logging period]
`LOG COMCONFIG ONTIME 4 1.5`	[offset is not an integer]

For synchronous and asynchronous logs, the receiver supports any offset that is:

- smaller than the logging period
- a multiple of the minimum logging period

For example, if the receiver supports 20 Hz logging, the minimum logging period is 1/20 Hz or 0.05 s. The following are valid examples for a synchronous, or asynchronous log, on a receiver that can log at rates up to 20 Hz:

`LOG PSRPOS 0.05`	*[20 Hz]*
`LOG PSRPOS 0.1`	*[10 Hz]*
`LOG PSRPOS 0.1 0.05`	
`LOG PSRPOS ONTIME 1`	*[1 Hz]*
`LOG PSRPOS ONTIME 1 0.1`	
`LOG PSRPOS ONTIME 1 0.90`	
`LOG AVEPOS ONTIME 1 0.95`	
`LOG AVEPOS ONTIME 2`	*[0.5 Hz]*
`LOG AVEPOS ONTIME 2 1.35`	
`LOG AVEPOS ONTIME 2 1.75`	

For synchronous and asynchronous logs, the following examples are invalid:

`LOG PSRPOS ONTIME 1 0.08`	[offset is not a multiple of the minimum logging period]
`LOG PSRPOS ONTIME 1 1.05`	[offset is larger than the logging period]

C.2 Description of ASCII and Binary Logs with Short Headers

These logs are set up in the same way normal ASCII or binary logs are, except that a normal ASCII or binary header is replaced with a short header (see *Tables 36 and 37*). For the message header structure of OEMV-3 regular Binary and ASCII logs, please refer to the *OEMV Family Firmware Reference Manual*.

Table 36: Short ASCII Message Header Structure

Field #	Field Type	Field Type	Description
1	%	Char	% symbol
2	Message	Char	This is the name of the log
3	Week Number	Ushort	GPS week number
4	Milliseconds	Ulong	Milliseconds from the beginning of the GPS week

Table 37: Short Binary Message Header Structure

Field #	Field Type	Field Type	Description	Binary Bytes	Binary Offset
1	Sync	Char	Hex 0xAA	1	0
2	Sync	Char	Hex 0x44	1	1
3	Sync	Char	Hex 0x13	1	2
4	Message Length	Uchar	Message length, not including header or CRC	1	3
5	Message ID	Ushort	Message ID number	2	4
6	Week Number	Ushort	GPS week number	2	6
7	Milliseconds	Ulong	Milliseconds from the beginning of the GPS week	4	8

C.3 NMEA Standard Logs

The National Marine Electronic Association (NMEA) logs in this manual are listed below:

GPALM	**ALMANAC DATA**
GPGGA	**GLOBAL POSITION SYSTEM FIX DATA AND UNDULATION**
GPGLL	**GEOGRAPHIC POSITION**
GPGRS	**GPS RANGE RESIDUALS FOR EACH SATELLITE**
GPGSA	**GPS DOP AN ACTIVE SATELLITES**
GPGST	**PSEUDORANGE MEASUREMENT NOISE STATISTICS**
GPGSV	**GPS SATELLITES IN VIEW**
GPZDA	**UTC TIME AND DATE**

The NMEA log structures follow format standards as adopted by the National Marine Electronics Association. The reference document used is "Standard For Interfacing Marine Electronic Devices NMEA 0183 Version 3.01". For further information, see the *Standards and References* section of the *GNSS Reference Book,* available on our Web site at http://www.novatel.com/support/docupdates.htm. The following table contains excerpts from Table 6 of the NMEA Standard which defines the variables for the NMEA logs. The actual format for each parameter is indicated after its description.

The NMEA (National Marine Electronics Association) has defined standards that specify how electronic equipment for marine users communicate. GPS receivers are part of this standard and the NMEA has defined the format for several GPS data logs otherwise known as 'sentences'.

Each NMEA sentence begins with a '$' followed by the prefix 'GP' followed by a sequence of letters that define the type of information contained in the sentence. Data contained within the sentence is separated by commas and the sentence is terminated with a two digit checksum followed by a carriage return/line feed. Here is an example of an NMEA sentence that describes time, position, and fix related data:

```
$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,
-16.27,M,08,AAAA*60
```

This example, and other NMEA logs, are output the same no matter what GPS receiver is used, providing a standard way to communicate and process GPS information.

Field Type	Symbol	Definition
Special Format Fields		
Status	A	Single character field: A = Yes, Data Valid, Warning Flag Clear V = No, Data Invalid, Warning Flag Set
Latitude	llll.ll	Fixed/Variable length field: degrees|minutes.decimal - 2 fixed digits of degrees, 2 fixed digits of mins and a variable number of digits for decimal-fraction of mins. Leading zeros always included for degrees and mins to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required.
Longitude	yyyyy.yy	Fixed/Variable length field: degrees|minutes.decimal - 3 fixed digits of degrees, 2 fixed digits of mins and a variable number of digits for decimal-fraction of mins. Leading zeros always included for degrees and mins to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required
Time	hhmmss.ss	Fixed/Variable length field: hours|minutes|seconds.decimal - 2 fixed digits of hours, 2 fixed digits of mins, 2 fixed digits of seconds and variable number of digits for decimal-fraction of seconds. Leading zeros always included for hours, mins and seconds to maintain fixed length. The decimal point and associated decimal-fraction are optional if full resolution is not required.
Defined field		Some fields are specified to contain pre-defined constants, most often alpha characters. Such a field is indicated in this standard by the presence of one or more valid characters. Excluded from the list of allowable characters are the following which are used to indicate field types within this standard: "A", "a", "c", "hh", "hhmmss.ss", "llll.ll", "x", "yyyyy.yy"
Numeric Value Fields		
Variable numbers	x.x	Variable length integer or floating numeric field. Optional leading and trailing zeros. The decimal point and associated decimal-fraction are optional if full resolution is not required (example: 73.10 = 73.1 = 073.1 = 73)
Fixed HEX	hh___	Fixed length HEX numbers only, MSB on the left
Information Fields		
Variable text	c--c	Variable length valid character field.
Fixed alpha	aa___	Fixed length field of uppercase or lowercase alpha characters
Fixed	xx___	Fixed length field of numeric characters
Fixed text	cc___	Fixed length field of valid characters

NOTES:

1. Spaces may only be used in variable text fields.
2. A negative sign "-" (HEX 2D) is the first character in a Field if the value is negative. The sign is omitted if the value is positive.
3. All data fields are delimited by a comma (,).
4. Null fields are indicated by no data between two commas (,,). Null fields indicate invalid data or no data available.
5. The NMEA Standard requires that message lengths be limited to 82 characters.

C.4 SPAN-SE Logs

The receivers are capable of generating many NovAtel-format output logs, in either Abbreviated ASCII, ASCII or binary format. For convenience, some commonly used OEMV logs are included in this manual. All SPAN-specific logs are included in this manual. Please refer to the *OEMV Family Firmware Reference Manual* for a complete list of logs containing GNSS-only information, categorized by function and then detailed in alphabetical order.

Logging Restriction Important Notice

Please note these 3 rules when configuring your SPAN system:

1. BESTPOS and BESTGPSPOS logs are available at 1 and 5 Hz only on SPAN-SE.
2. When requesting high-rate data over COM1, COM2, COM3 or COM4, be careful not to overrun the baud rate.
3. RAWIMU and RAWIMUS logs are only available with the ONNEW or ONCHANGED trigger. These logs are not valid with the ONTIME trigger. The raw IMU observations contained in these logs are sequential changes in velocity and rotation. As such, you can only use them for navigation if they are logged at their full rate. See details of these log starting on *page 249*.
4. In order to collect wheel sensor information, useful in post-processing, the TIMEDWHEELDATA log should only be used with the ONNEW trigger. See also *page 268* for details on this log.

✉ The periods available when you use the ONTIME trigger are 0.005 (200Hz), 0.01 (100Hz), 0.02 (50 Hz), 0.05, 0.1, 0.2, 0.25, 0.5, 1, 2, 3, 5, 10, 15, 20, 30 or 60 seconds.

C.4.1 BESTPOS Best Position and BESTGPSPOS Best GPS Position

The BESTPOS log contains the best available position from either GNSS only, or GNSS/INS. BESTGPSPOS contains the best available GNSS position (without INS). Both logs have an identical format. In addition, it reports several status indicators, including differential age, which is useful in predicting anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used.

On SPAN-SE, the BESTPOS and BESTGPSPOS logs are available at < 1 Hz, 1 Hz and 5 Hz only. It is a SPAN-only log and is not available directly from the OEMV.

With the system operating in an RTK mode, this log reflects the latest low-latency solution for up to 60 seconds after reception of the last base station observations. After this 60 second period, the position reverts to the best solution available; the degradation in accuracy is reflected in the standard deviation fields. If the system is not operating in an RTK mode, pseudorange differential solutions continue for 300 seconds after loss of the data link, though a different value can be set using the DGPSTIMEOUT command, refer to the *OEMV Family Firmware Reference Manual.*

Structure: **BESTGPSPOS Message ID: 423**

BESTPOS Message ID: 42

Log Type: Synch

Table 38: Position or Velocity Type

Position Type (binary)	Position Type (ASCII)	Description
0	NONE	No solution
1	FIXEDPOS	Position has been fixed by the FIX POSITION command or by position averaging
2	FIXEDHEIGHT	Position has been fixed by the FIX HEIGHT, or FIX AUTO, command or by position averaging
3	Reserved	
4	FLOATCONV	Solution from floating point carrier phase ambiguities
5	WIDELANE	Solution from wide-lane ambiguities
6	NARROWLANE	Solution from narrow-lane ambiguities
7	Reserved	
8	DOPPLER_VELOCITY	Velocity computed using instantaneous Doppler
9-15	Reserved	
16	SINGLE	Single point position
17	PSRDIFF	Pseudorange differential solution
18	WAAS	Solution calculated using corrections from an SBAS
19	PROPOGATED	Propagated by a Kalman filter without new observations

Continued on page 173

Position Type (binary)	Position Type (ASCII)	Description
20	OMNISTAR	OmniSTAR VBS position (L1 sub-meter) [1]
21-31	Reserved	
32	L1_FLOAT	Floating L1 ambiguity solution
33	IONOFREE_FLOAT	Floating ionospheric-free ambiguity solution
34	NARROW_FLOAT	Floating narrow-lane ambiguity solution
48	L1_INT	Integer L1 ambiguity solution
49	WIDE_INT	Integer wide-lane ambiguity solution
50	NARROW_INT	Integer narrow-lane ambiguity solution
51	RTK_DIRECT_INS	RTK status where the RTK filter is directly initialized from the INS filter. [2]
52	INS	INS calculated position corrected for the antenna [2]
53	INS_PSRSP	INS pseudorange single point solution - no DGPS corrections [2]
54	INS_PSRDIFF	INS pseudorange differential solution [2]
55	INS_RTKFLOAT	INS RTK floating point ambiguities solution [2]
56	INS_RTKFIXED	INS RTK fixed ambiguities solution [2]
57	INS_OMNISTAR	INS OmniSTAR VBS position (L1 sub-meter) [1]
58	INS_OMNISTAR_HP	INS OmniSTAR high precision solution [1]
59	INS_OMNISTAR_XP	INS OmniSTAR extra precision solution [1]
64	OMNISTAR_HP	OmniSTAR high precision [1]
65	OMNISTAR_XP	OmniSTAR extra precision [1]
66	CDGPS	Position solution using CDGPS corrections [1]

1. In addition to a NovAtel receiver with L-band capability, a subscription to the OmniSTAR, or use of the free CDGPS, service is required. Contact NovAtel for details.
2. These types appear in position logs such as BESTPOS.

Table 39: Solution Status

Binary	ASCII	Description
0	SOL_COMPUTED	Solution computed
1	INSUFFICIENT_OBS	Insufficient observations
2	NO_CONVERGENCE	No convergence
3	SINGULARITY	Singularity at parameters matrix
4	COV_TRACE	Covariance trace exceeds maximum (trace > 1000 m)
5	TEST_DIST	Test distance exceeded (maximum of 3 rejections if distance > 10 km)
6	COLD_START	Not yet converged from cold start
7	V_H_LIMIT	Height or velocity limits exceeded (in accordance with COCOM export licensing restrictions)
8	VARIANCE	Variance exceeds limits
9	RESIDUALS	Residuals are too large
10	DELTA_POS	Delta position is too large
11	NEGATIVE_VAR	Negative variance
12-17	Reserved	
18	PENDING	When a FIX POSITION command is entered, the receiver computes its own position and determines if the fixed position is valid [1]
19	INVALID_FIX	The fixed position, entered using the FIX POSITION command, is not valid

1. PENDING implies there are not enough satellites being tracked to verify if the FIX POSITION entered into the receiver is valid. The receiver needs to be tracking two or more GPS satellites to perform this check. Under normal conditions you should only see PENDING for a few seconds on power up before the GPS receiver has locked onto its first few satellites. If your antenna is obstructed (or not plugged in) and you have entered a FIX POSITION command, then you may see PENDING indefinitely.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header	-	H	0
2	Sol Status	Solution status, see *Table 39* on *page 174*	Enum	4	H
3	Pos Type	Position type, see *Table 38 on page 172*	Enum	4	H+4
4	Lat	Latitude	Double	8	H+8
5	Lon	Longitude	Double	8	H+16
6	Hgt	Height above mean sea level	Double	8	H+24
7	Undulation	Undulation	Float	4	H+32
8	Datum ID	Datum ID (refer to the DATUM command in the *OEMV Family Firmware Reference Manual*)	Enum	4	H+36
9	Lat σ	Latitude standard deviation	Float	4	H+40
10	Lon σ	Longitude standard deviation	Float	4	H+44
11	Hgt σ	Height standard deviation	Float	4	H+48
12	Stn ID	Base station ID	Char[4]	4	H+52
13	Diff_age	Differential age	Float	4	H+56
14	Sol_age	Solution age in seconds	Float	4	H+60
15	#obs	Number of observations tracked	Uchar	1	H+64
16	#GPSL1	Number of GPS L1 ranges used in computation	Uchar	1	H+65
17	#L1	Number of GPS L1 ranges above the RTK mask angle	Uchar	1	H+66
18	#L2	Number of GPS L2 ranges above the RTK mask angle	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20			Uchar	1	H+69
21			Uchar	1	H+70
22			Uchar	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log bestgpsposa ontime 1

ASCII Example:

```
#BESTGPSPOSA,COM1,0,62.5,FINESTEERING,1036,484878.000,00000028,63e2,0;
SOL_COMPUTED,SINGLE,51.11629893124,-114.03820302746,1052.3434,
-16.271287293,61,19.6934,13.1515,23.8561,"",0.0,60.000,10,10,0,0,
0,0,0,0*1051ada9
```

C.4.2 BESTVEL Best Available Velocity Data and BESTGPSVEL Best Available GPS Velocity Data

The BESTVEL log contains the best available velocity from either GNSS only, or GNSS/INS. BESTGPSVEL contains the best available GNSS velocity (without INS). Both logs have an identical format. In addition, it reports a velocity status indicator, which is useful in indicating whether or not the corresponding data is valid. The velocity measurements sometimes have a latency associated with them. The time of validity is the time tag in the log minus the latency value. A valid solution with a latency of 0.0 indicates that the instantaneous Doppler measurement was used to calculate velocity.

The velocity is typically computed from the average change in pseudorange over the time interval or the RTK Low Latency filter. As such, it is an average velocity based on the time difference between successive position computations and not an instantaneous velocity at the BESTGPSVEL time tag. The velocity latency to be subtracted from the time tag is normally 1/2 the time between filter updates. Under default operation, the positioning filters are updated at a rate of 2 Hz. This translates into a velocity latency of 0.25 second. The latency can be reduced by increasing the update rate of the positioning filter being used by requesting the BESTGPSVEL or BESTGPSPOS messages at a rate higher than 2 Hz. For example, a logging rate of 10 Hz would reduce the velocity latency to 0.005 seconds. For integration purposes, the velocity latency should be applied to the record time tag.

On SPAN-SE, BESTVEL and BESTGPSVEL are available at 1 Hz or 5 Hz. Higher rate velocity information is available in the INSVEL, INSPVA or INSSPD logs.

Structure: **BESTGPSVEL Message ID: 506**

BESTVEL Message ID: 99

Log Type: Synch

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header	-	H	0
2	Sol Status	Solution status, see *Table 39 on page 174*	Enum	4	H
3	Vel Type	Velocity type, see *Table 38 on page 172*	Enum	4	H+4
4	Latency	A measure of the latency in the velocity time tag in seconds. It should be subtracted from the time to give improved results.	Float	4	H+8
5	Age	Differential age	Float	4	H+12
6	Hor Spd	Horizontal speed over ground, in metres per second	Double	8	H+16
7	Trk Gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+24
8	Vert Spd	Vertical speed, in metres per second, where positive values indicate increasing altitude (up) and negative values indicate decreasing altitude (down)	Double	8	H+32
9	Reserved		Float	4	H+40
10	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
11	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log bestgpsvela ontime 1

ASCII Example:

```
#BESTGPSVELA,COM1,0,62.5,FINESTEERING,1049,247755.000,00000128,f7e3,0;
SOL_COMPUTED,SINGLE,0.250,0.000,0.1744,333.002126,0.3070,6.0082*dfdc635c
```

C.4.3 BESTLEVERARM IMU to Antenna Lever Arm

This log contains the distance between the IMU and the GNSS antenna in the IMU enclosure frame and its associated uncertainties. If the you enter the lever arm through the SETIMUTOANTOFFSET command, see *page 137*, these values are reflected in this log. When the lever arm calibration is complete, see the LEVERARMCALIBRATE command on *page 111*, the solved values are also output in this log.

The default X (pitch), Y (roll) and Z (azimuth) directions of the inertial frame are clearly marked on the IMU, see *Figure 33* on *page 134*.

Structure: **Message ID: 674**

Log Type: Asynch

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	X Offset	IMU Enclosure Frame (m)	Double	8	H
3	Y Offset	IMU Enclosure Frame (m)	Double	8	H+8
4	Z Offset	IMU Enclosure Frame (m)	Double	8	H+16
5	X Uncertainty	IMU Enclosure Frame (m)	Double	8	H+24
6	Y Uncertainty	IMU Enclosure Frame (m)	Double	8	H+32
7	Z Uncertainty	IMU Enclosure Frame (m)	Double	8	H+40
8	iMapping	See *Table 29, Full Mapping Definitions* on *page 136*	Integer	4	H+48
9	xxxx	32-bit CRC	Hex	4	H+52
10	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Recommended Input:

log bestleverarma onchanged

ASCII Example:

```
#BESTLEVERARMA,COM1,0,83.5,UNKNOWN,0,2.983,00000008,39e4,35484;
0.3934000000000000,-1.2995000000000001,0.0105500000000000,
0.0300000000000000,0.0300000000000000,0.0300000000000000,4*876c47ad
```

C.4.4 COMCONFIG Current COM Port Configuration

This log outputs the current COM port configuration for each port on your receiver.

Message ID: **317**

Log Type: **Polled**

Recommended Input:

log comconfiga once

ASCII example:

```
#COMCONFIGA,COM1,0,96.5,FINESTEERING,1521,318837.286,00000000,0000,149;
7,
COM1,9600,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,
COM2,230400,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,
COM3,9600,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,
COM4,9600,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,
IMU,115200,N,8,1,N,OFF,OFF,IMU,IMU,OFF,
USB1,12000000,N,0,0,N,OFF,OFF,NOVATEL,NOVATEL,ON,
ETH1,10000000,N,0,0,N,OFF,OFF,NOVATEL,NOVATEL,ON*d32b5437
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	COMCONFIG header	Log header		H	0
2	#port	Number of ports with information to follow	Long	4	H
3	port	Serial port identifier, see *Table 18* on *page 96*	Enum	4	H+4
4	baud	Communication baud rate	Ulong	4	H+8
5	parity	See *Table 19* on *page 96*	Enum	4	H+12
6	databits	Number of data bits	Ulong	4	H+16
7	stopbits	Number of stop bits	Ulong	4	H+20
8	handshake	See *Table 20* on *page 96*	Enum	4	H+24
9	echo	When echo is on, the port is transmitting any input characters as they are received. 0 = OFF 1 = ON	Enum	4	H+28
10	breaks	Breaks are turned on or off 0 = OFF 1 = ON	Enum	4	H+32
11	rx type	The status of the receive interface mode, see *Table 24, Serial Port Interface Modes* on *page 107*	Enum	4	H+36
12	tx type	The status of the transmit interface mode, see *Table 24* on *page 107*	Enum	4	H+40
13	response	Responses are turned on or off 0 = OFF 1 = ON	Enum	4	H+44
14	next port offset = H + 4 + (#port x 44)				
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(#port x44)
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.5 COMPROTOCOL COM Port Protocol

This log reports what the current protocol settings are on each SPAN-SE COM port. The protocol can be set with the COMCONTROL command, described on *page 98* of this manual.

Message ID: **1145**

Log Type: **Polled**

Recommended Input:

log comprotocola once

ASCII Example:

```
#COMPROTOCOLA,COM1,0,95.0,FINESTEERING,1521,319232.645,00000000,0000,149;
5,
COM1,RS232,
COM2,RS232,
COM3,RS232,
COM4,RS232,
IMU,RS232*de92c2fb
```

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log Header	-	H	0
2	#recs	Number of records to follow	Ulong	4	H
3	port	COM port, see *Table 18, COM Serial Port Identifiers* on *page 96*	Enum	4	H+4
4	protocol	Port protocol, see *Table 40* below	Enum	4	H+8
5	next record offset = H + 4 + (#recs x 8)				
6	xxxx	32-bit CRC	Hex	4	H+4+ (#recs x 8)
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Table 40: Port Protocol

ASCII	Binary	Description
RS232	0	RS-232 mode
RS422	1	RS-422 mode

C.4.6 DIRENT SD Card File List

The DIRENT log contains the current file contents of the receiver's SD Card. Up to 1024 files can be listed using this message.

The date and time for the DIRENT log is in UTC (Universal Coordinated Time). If the SPAN-SE receiver has no almanac, UTC is unavailable.

The *Date of Last Change* field has 4 decimal digits reserved for the year, followed by 2 decimal digits for the month, and 2 decimal digits for the day.

Example: Date of September 5, 2001

Date of Last Change field for this date when converted to a Ulong has a value of 20010905. The following steps can be used to obtain the various parts of the *Date of Last Change* field:

Ulong Year = (Ulong)(DateOfLastChange / 10000)
Ulong Month = (Ulong)((DateOfLastChange - (Year * 10000)) / 100)
Ulong Day = (Ulong)(DateOfLastChange - (Year * 10000) - (Month * 100))

The *Time of Last Change* field is similar to the *Date of Last Change* field, in that the value of the field has 2 decimal digits reserved for the hour, followed by 2 decimal digits for the minutes, and 2 decimal digits for the seconds.

Example: Time of 16:01:25

Time of Last Change field when converted to a Ulong has a value of 160125. The following steps can be used to obtain the various parts of the *Time of Last Change* field:

Ulong Hour = (Ulong)(TimeOfLastChange / 10000)
Ulong Minutes = (Ulong)((TimeOfLastChange - (Hour * 10000)) / 100)
Ulong Seconds = (Ulong)(TimeOfLastChange - (Hour * 10000) - (Minutes * 100))

Structure: **Message ID = 159**

Log Type: Polled

Field	Data	Bytes	Format	Units	Offset
1	Log Header		-	-	0
2	Filename	12	Char[ ]	none	H
3	Size (bytes)	4	Ulong	bytes	H+12
4	Size (packets)	4	Ulong	packets	H+16
5	Date of Last Change	4	Ulong	yyyymmdd	H+20
6	Time of Last Change	4	Ulong	hhmmss	H+24

C.4.7 GLOCLOCK GLONASS Clock Information

This log contains the time difference information between GNSS and GLONASS time as well as status flags. The status flags are used to indicate the type of time processing used in the least squares adjustment. GNSS and GLONASS time are both based on the Universal Time Coordinated (UTC) time scale with some adjustments. GPS time is continuous and does not include any of the leap second adjustments to UTC applied since 1980. The result is that GPS time currently leads UTC time by 14 seconds.

GLONASS time applies leap seconds but is also three hours ahead to represent Moscow time. The nominal offset between GPS and GLONASS time is therefore due to the three hour offset minus the leap second offset. Currently this value is at 10787 seconds with GLONASS leading. As well as the nominal offset, there is a residual offset on the order of nanoseconds which must be estimated in the least squares adjustment. The GLONASS-M satellites broadcasts this difference in the navigation message.

This log also contains information from the GLONASS navigation data relating GLONASS time to UTC.

Message ID: **719**

Log Type: **Asynch**

Recommended Input:

log gloclocka onchanged

ASCII Example:

```
#GLOCLOCKA,COM1,0,54.5,SATTIME,1364,411884.000,00000000,1d44,2310;
0,0.000000000,0.000000000,0,0,-0.000000275,792,-0.000001207,
0.000000000,0.000000000,0*437e9afaf
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLOCLOCK header	Log header		H	0
2	Reserved		Ulong	4	H
3			Double	8	H+4
4			Double	8	H+12
5	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (new M type)	Uchar	1	H+20
6	N^4	Four-year interval number starting from 1996	Uchar	1 [1]	H+21 [a]
7	τ_{GPS}	GPS time scale correction to UTC(SU) given at beginning of day N^4, in seconds	Double	8	H+24
8	N^A	GLONASS calendar day number within a four year period beginning since the leap year, in days	Ushort	2 [a]	H+32 [a]
9	τ_C	GLONASS time scale correction to UTC time, in seconds	Double	8	H+36
10	b1	Beta parameter 1st order term	Double	8	H+44
11	b2	Beta parameter 2nd order term	Double	8	H+52
12	Kp	The Kp scale summarizes the global level of geomagnetic activity. A Kp of 0 to 4 is below storm levels (5 to 9).	Uchar	1	H+60
13	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+61
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. In the binary log case, an additional bytes of padding are added to maintain 4-byte alignment

C.4.8 GLOEPHEMERIS GLONASS Ephemeris Data

GLONASS ephemeris information is available through the GLOEPHEMERIS log. GLONASS ephemerides are referenced to the SGS-90 geodetic datum, and GLONASS coordinates are reconciled internally through a position filter and output to WGS84.

✉ GLONASS measurements can be used for post-processed positioning solutions or in user-designed programs. NovAtel plans to offer GLONASS positioning in the future. In the meantime, OEMV-based output is compatible with post-processing software from the Waypoint Products Group, NovAtel Inc. See also www.novatel.com for details.

Message ID: **723**
Log Type: **Asynch**

Recommended Input:

log gloephemerisa onchanged

Example:

```
#GLOEPHEMERISA,COM1,3,49.0,SATTIME,1364,413624.000,00000000,6b64,2310;
43,8,1,0,1364,413114000,10786,792,0,0,87,0,9.0260864257812500e+06,
-6.1145468750000000e+06,2.2926090820312500e+07,1.4208841323852539e+03,
2.8421249389648438e+03,1.9398689270019531e+02,0.00000000000000000,
-2.79396772384643555e-06,-2.79396772384643555e-06,2.12404876947402954e-04,
-1.396983862e-08,-3.63797880709171295e-12,78810,3,15,0,12*a02ce18b
#GLOEPHEMERISA,COM1,2,49.0,SATTIME,1364,413626.000,00000000,6b64,2310;
44,11,1,0,1364,413116000,10784,792,0,0,87,13,-1.2882617187500000e+06,
-1.9318657714843750e+07,1.6598909179687500e+07,9.5813846588134766e+02,
2.0675134658813477e+03,2.4769935607910156e+03,2.79396772384643555e-06,
-3.72529029846191406e-06,-1.86264514923095703e-06,6.48368149995803833e-05,
-4.656612873e-09,3.63797880709171295e-12,78810,3,15,3,28*e2d5ef15
#GLOEPHEMERISA,COM1,1,49.0,SATTIME,1364,413624.000,00000000,6b64,2310;
45,13,0,0,1364,413114000,10786,0,0,0,87,0,-1.1672664062500000e+07,
-2.2678505371093750e+07,4.8702343750000000e+05,-1.1733341217041016e+02,
1.3844585418701172e+02,3.5714883804321289e+03,2.79396772384643555e-06,
-2.79396772384643555e-06,0.00000000000000000,-4.53162938356399536e-05,
5.587935448e-09,-2.36468622460961342e-11,78810,0,0,0,8*c15abfeb
#GLOEPHEMERISA,COM1,0,49.0,SATTIME,1364,413624.000,00000000,6b64,2310;
59,17,0,0,1364,413114000,10786,0,0,0,87,0,-2.3824853515625000e+05,
-1.6590188964843750e+07,1.9363733398437500e+07,1.3517074584960938e+03,
-2.2859592437744141e+03,-1.9414072036743164e+03,1.86264514923095703e-06,
-3.72529029846191406e-06,-1.86264514923095703e-06,7.92574137449264526e-05,
4.656612873e-09,2.72848410531878471e-12,78810,0,0,0,12*ed7675f
```

Table 41: GLONASS Ephemeris Flags Coding

Nibble Number	Bit	Description	Range Values	Hex Value
N 0	lsb = 0	P1 FLAG - TIME INTERVAL BETWEEN ADJACENT iSSUE(tb) VALUES	See Table below (*Table 42*)	00000001
	1			00000002
	2	P2 FLAG - ODDNESS OR EVENNESS OF iSSUE(tb) VALUE	0 = even, 1 = odd	00000004
	3	P3 FLAG - NUMBER OF SATELLITES WITH ALMANAC INFORMATION WITHIN CURRENT SUBFRAME	0 = four, 1 = five	00000008
	4 : 31	RESERVED (N-1 through N-7)		

Table 42: *Bits 0 - 1: P1 Flag Range Values*

State	Description
00	0 minutes
01	30 minutes
10	45 minutes
11	60 minutes

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLO-EPHEMERIS header	Log header		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). This is also called SLOTO in CDU	Ushort	2	H
3	freqo	Frequency channel offset for satellite in the range 0 to 20	Ushort	2	H+2
4	sat type	Satellite type where 0 = GLO_SAT 1 = GLO_SAT_M (new M type)	Uchar	1	H+4
5	Reserved			1	H+5
6	e week	Reference week of ephemeris (GPS time)	Ushort	2	H+6
7	e time	Reference time of ephemeris (GPS time) in ms	Ulong	4	H+8
8	t offset	Integer seconds between GPS and GLONASS time. A positive value implies GLONASS is ahead of GPS time.	Ulong	4	H+12
9	Nt	Current data number. This field is only output for the new M type satellites. See example output from both satellite types (field 4) on *page 185*.	Ushort	2	H+16
10	Reserved			1	H+18
11	Reserved			1	H+19
12	issue	15-minute interval number corresponding to ephemeris reference time	Ulong	4	H+20
13	health	Ephemeris health where 0 = GOOD 1 = BAD	Ulong	4	H+24
14	pos x	X coordinate for satellite at reference time (PZ-90.02), in meters	Double	8	H+28
15	pos y	Y coordinate for satellite at reference time (PZ-90.02), in meters	Double	8	H+36
16	pos z	Z coordinate for satellite at reference time (PZ-90.02), in meters	Double	8	H+44
17	vel x	X coordinate for satellite velocity at reference time (PZ-90.02), in meters/s	Double	8	H+52
18	vel y	Y coordinate for satellite velocity at reference time (PZ-90.02), in meters/s	Double	8	H+60

Continued on page 188

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
19	vel z	Z coordinate for satellite velocity at reference time (PZ-90.02), in meters/s	Double	8	H+68
20	LS acc x	X coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters/s/s	Double	8	H+76
21	LS acc y	Y coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters/s/s	Double	8	H+84
22	LS acc z	Z coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters/s/s	Double	8	H+92
23	tau_n	Correction to the nth satellite time t_n relative to GLONASS time t_c, in seconds	Double	8	H+100
24	delta_tau_n	Time difference between navigation RF signal transmitted in L2 sub-band and navigation RF signal transmitted in L1 sub-band by nth satellite, in seconds	Double	8	H+108
25	gamma	Frequency correction, in seconds/second	Double	8	H+116
26	Tk	Time of frame start (since start of GLONASS day), in seconds	Ulong	4	H+124
27	P	Technological parameter	Ulong	4	H+128
28	Ft	User range	Ulong	4	H+132
29	age	Age of data, in days	Ulong	4	H+136
30	Flags	Information flags, see *Table 41, GLONASS Ephemeris Flags Coding* on *page 186*	Ulong	4	H+140
31	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+144
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.9 GLORAWEPHEM Raw GLONASS Ephemeris Data

This log contains the raw ephemeris frame data as received from the GLONASS satellite.

Message ID: **792**
Log Type: **Asynch**

Recommended Input:

log glorawephema onchanged

Example:

```
#GLORAWEPHEMA,COM1,3,47.0,SATTIME,1340,398653.000,00000000,332d,2020;
38,9,0,1340,398653.080,4,
0148d88460fc115dbdaf78,0,0218e0033667aec83af2a5,0,
038000b9031e14439c75ee,0,0404f22660000000000065,0*17f3dd17
…
#GLORAWEPHEMA,COM1,0,47.0,SATTIME,1340,398653.000,00000000,332d,2020;
41,13,0,1340,398653.078,4,
0108d812532805bfa1cd2c,0,0208e0a36e8e0952b111da,0,
03c02023b68c9a32410958,0,0401fda44000000000002a,0*0b237405
```

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	GLORAWEPHEM header	Log header		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). Ephemeris relates to this slot and is also called SLOTO in CDU.	Ushort	2	H
3	freqo	Frequency channel offset in the range 0 to 20	Ushort	2	H+2
4	sigchan	Signal channel number	Ulong	4	H+4
5	week	GPS Week, in weeks	Ulong	4	8
6	time	GPS Time, in milliseconds (binary data) or seconds (ASCII data)	Ulong	4	12
7	#recs	Number of records to follow	Ulong	4	H+16
8	string	GLONASS data string	Uchar [string size] [1]	variable	H+20
9	Reserved		Uchar	1	variable
10...	Next record offset = H + 20 + (#recs x [string size + 1])				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	variable
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. In the binary log case, additional bytes of padding are added to maintain 4-byte alignment.

C.4.10 GPALM Almanac Data

This National Marine Electronics Association (NMEA) log, see also *Section C.3, NMEA Standard Logs* on *page 169*, outputs raw almanac data for each satellite PRN contained in the broadcast message. A separate record is logged for each PRN, up to a maximum of 32 records. GPALM outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then UTC time is then set to VALID. It takes a minimum of 12.5 minutes to collect a complete almanac following receiver boot-up. If an almanac was stored in NVM, the stored values are reported in the GPALM log once time is set on the receiver.

☒ To obtain copies of ICD-GPS-200, seen in the GPALM table footnotes, refer to ARINC in the *Standards and References* section of the *GNSS Reference Book*, available on our Web site. Refer also to NMEA contact information there.

Message ID: **217**

Log Type: **Asynch**

Recommended Input:

log gpalm onchanged

Example:

```
$GPALM,28,01,01,1337,00,305a,90,1b9d,fd5b,a10ce9,ba0a5e,2f48f1,cccb76,006,001*27
$GPALM,28,02,02,1337,00,4aa6,90,0720,fd50,a10c5a,4dc146,d89bab,0790b6,fe4,000*70
.
.
.
$GPALM,28,24,26,1337,00,878c,90,1d32,fd5c,a10c90,1db6b6,2eb7f5,ce95c8,00d,000*23
$GPALM,28,25,27,1337,00,9cde,90,07f2,fd54,a10da5,adc097,562da3,6488dd,00e,000*2F
$GPALM,28,26,28,1337,00,5509,90,0b7c,fd59,a10cc4,a1d262,83e2c0,3003bd,02d,000*78
$GPALM,28,27,29,1337,00,47f7,90,1b20,fd58,a10ce0,d40a0b,2d570e,221641,122,006*7D
$GPALM,28,28,30,1337,00,4490,90,0112,fd4a,a10cc1,33d10a,81dfc5,3bdb0f,178,004*28
```

Field	Structure	Field Description	Symbol	Example
1	$GPALM	Log header		$GPALM
2	# msg	Total number of messages logged. Set to zero until almanac data is available.	x.x	17
3	msg #	Current message number	x.x	17
4	PRN	Satellite PRN number: GPS = 1 to 32	xx	28
5	GPS wk	GPS reference week number [a].	x.x	653
6	SV hlth	SV health, bits 17-24 of each almanac page [b]	hh	00
7	ecc	e, eccentricity [c d]	hhhh	3EAF
8	alm ref time	toa, almanac reference time [c]	hh	87
9	incl angle	$(sigma)_i$, inclination angle [c]	hhhh	OD68
10	omegadot	OMEGADOT, rate of right ascension [c]	hhhh	FD30
11	rt axis	$(A)^{1/2}$, root of semi-major axis [c]	hhhhhh	A10CAB
12	omega	omega, argument of perigee [c e]	hhhhhh	6EE732
13	long asc node	$(OMEGA)_o$,longitude of ascension node [c]	hhhhhh	525880
14	M_o	Mo, mean anomaly [c]	hhhhhh	6DC5A8
15	a_{f0}	af0, clock parameter [c]	hhh	009
16	a_{f1}	af1, clock parameter [c]	hhh	005
17	*xx	Checksum	*hh	*37
18	[CR][LF]	Sentence terminator		[CR][LF]

a Variable length integer, 4-digits maximum from (2) most significant binary bits of Subframe 1, Word 3 reference Table 20-I, ICD-GPS-200, Rev. B, and (8) least significant bits from subframe 5, page 25, word 3 reference Table 20-I, ICD-GPS-200

b Reference paragraph 20.3.3.5.1.3, Table 20-VII and Table 20-VIII, ICD-GPS-200, Rev. B

c Reference Table 20-VI, ICD-GPS-200, Rev. B for scaling factors and units.

d A quantity defined for a conic section where e= 0 is a circle, e = 1 is an ellipse, 0<e<1 is a parabola and e>1 is a hyperbola.

e A measurement along the orbital path from the ascending node to the point where the SV is closest to the Earth, in the direction of the SV's motion

C.4.11 GPGGA GPS Fix Data and Undulation

This NMEA log provides time, position and fix-related data of the GNSS receiver. See also *Section C.3, NMEA Standard Logs* on *page 169*. For more on precision of NMEA logs, see *Table 46, Position Precision of NMEA Logs* on *page 196*.

Below are tables that show how many GNSS and/or GLONASS satellites you need to obtain a fixed ambiguity solution, *Table 43* below, and how many you need to keep a fixed ambiguity solution, see *Table 44* on *page 193*.

The GPGGA log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

Message ID: **218**

Log Type **Synch**

Recommended Input:

log gpgga ontime 1

Example:

```
$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,-16.27,M,
08,AAAA*60
```

Table 43: To Obtain a Fixed Ambiguity Solution

	#GNSS Satellites						
#GLO Satellites	**2**	**3**	**4**	**5**	**6**	**7**	**8**
2	No	Float	Fix	Fix	Fix	Fix	Fix
3	Float	Float	Fix	Fix	Fix	Fix	Fix
4	Float	Float	Fix	Fix	Fix	Fix	Fix
5	Float	Float	Fix	Fix	Fix	Fix	Fix
6	Float	Float	Fix	Fix	Fix	Fix	Fix
7	Float	Float	Fix	Fix	Fix	Fix	Fix
8	Float	Float	Fix	Fix	Fix	Fix	Fix

Table 44: To Maintain a Fixed Ambiguity Solution

	#GNSS Satellites						
#GLO Satellites	2	3	4	5	6	7	8
2	No	Fix	Fix	Fix	Fix	Fix	Fix
3	Fix	Fix	Fix	Fix	Fix	Fix	Fix
4	Fix	Fix	Fix	Fix	Fix	Fix	Fix
5	Fix	Fix	Fix	Fix	Fix	Fix	Fix
6	Fix	Fix	Fix	Fix	Fix	Fix	Fix
7	Fix	Fix	Fix	Fix	Fix	Fix	Fix
8	Fix	Fix	Fix	Fix	Fix	Fix	Fix

Field	Structure	Field Description	Symbol	Example
1	$GPGGA	Log header		$GPGGA
2	utc	UTC time of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	202134.00
3	lat	Latitude (DDmm.mm)	llll.ll	5106.9847
4	lat dir	Latitude direction (N = North, S = South)	a	N
5	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	lon dir	Longitude direction (E = East, W = West)	a	W
7	GNSS qual	GNSS Quality indicator 0 = fix not available or invalid 1 = GPS fix 2 = C/A differential GPS, OmniSTAR HP, OmniSTAR XP, OmniSTAR VBS, or CDGPS 4 = RTK fixed ambiguity solution (RT2), see also *Table 44* on *page 193* 5 = RTK floating ambiguity solution (RT20), OmniSTAR HP or OmniSTAR XP 6 = Dead reckoning mode 7 = Manual input mode (fixed position) 8 = Simulator mode 9 = WAAS [1]	x	1
8	# sats	Number of satellites in use. May be different to the number in view	xx	10
9	hdop	Horizontal dilution of precision	x.x	1.0
10	alt	Antenna altitude above/below mean sea level	x.x	1062.22
11	a-units	Units of antenna altitude (M = meters)	M	M
12	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	u-units	Units of undulation (M = meters)	M	M
14	age	Age of Differential GPS data (in seconds) [2]	xx	(empty when no differential data is present)
15	stn ID	Differential base station ID, 0000-1023	xxxx	(empty when no differential data is present)
16	*xx	Checksum	*hh	*48
17	[CR][LF]	Sentence terminator		[CR][LF]

1. An indicator of 9 has been temporarily set for WAAS (NMEA standard for WAAS not decided yet).
2. The maximum age reported here is limited to 99 seconds.

C.4.12 GPGLL Geographic Position

This NMEA log provides altitude and longitude of the present vessel position, time of position fix, and status. See also *Section C.3, NMEA Standard Logs* on *page 169*.

Table 46 on page 196 compares the position precision of selected NMEA logs.

The GPGLL log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

> ✉ If the NMEATALKER command, see *page 119*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only) or IN (GNSS+INS solution).

Message ID: **219**

Log Type: **Synch**

Recommended Input:

log gpgll ontime 1

Example1 (GPS only):

```
$GPGLL,5107.0013414,N,11402.3279144,W,205412.00,A,A*73
```

Example 2 (Combined GPS and INS):

```
$INGLL,5106.9812620,N,11402.2906137,W,193052.00,A,A*6D
```

Table 45: NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

Table 46: Position Precision of NMEA Logs

NMEA Log	Latitude (# of decimal places)	Longitude (# of decimal places)	Altitude (# of decimal places)
GPGGA	4	4	2
GPGLL	7	7	N/A
GPRMC	7	7	N/A

Field	Structure	Field Description	Symbol	Example
1	$GPGLL	Log header		$GPGLL
2	lat	Latitude (DDmm.mm)	llll.ll	5106.7198674
3	lat dir	Latitude direction (N = North, S = South)	a	N
4	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.3587526
5	lon dir	Longitude direction (E = East, W = West)	a	W
6	utc	UTC time of position (hours/minutes/ seconds/decimal seconds)	hhmmss.ss	220152.50
7	data status	Data status: A = Data valid, V = Data invalid	A	A
8	mode ind	Positioning system mode indicator, see *Table 45* on *page 195*	a	A
9	*xx	Checksum	*hh	*1B
10	[CR][LF]	Sentence terminator		[CR][LF]

C.4.13 GPGRS GPS Range Residuals for Each Satellite

Range residuals can be computed in two ways, and this NMEA log reports those residuals. See also *Section C.3, NMEA Standard Logs* on *page 169*.

Under mode 0, residuals output in this log are used to update the position solution output in the GPGGA message. Under mode 1, the residuals are re-computed after the position solution in the GPGGA message is computed. The receiver computes range residuals in mode 1. An integrity process using GPGRS would also require GPGGA (for position fix data), GPGSA (for DOP figures), and GPGSV (for PRN numbers) for comparative purposes.

The GPGRS log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

---

1. If the range residual exceeds ± 99.9, then the decimal part is dropped. Maximum value for this field is ± 999. The sign of the range residual is determined by the order of parameters used in the calculation as follows:

   range residual = calculated range - measured range

2. There is no residual information available from the OmniSTAR HP/XP service, so the GPGRS contains the pseudorange position values when using it. For the OmniSTAR VBS or CDGPS service, residual information is available.

Message ID: **220**

Log Type: **Synch**

Recommended Input:

```
log gpgrs ontime 1
```

Example 1 (GPS only):

```
$GPGRS,142406.00,1,-1.1,-0.1,1.7,1.2,-2.0,-0.5,1.2,-1.2,-0.1,,,*67
```

Field	Structure	Field Description	Symbol	Example
1	$GPGRS	Log header		$GPGRS
2	utc	UTC time of position (hours/minutes/ seconds/ decimal seconds)	hhmmss.ss	192911.0
3	mode	Mode 0 = Residuals were used to calculate the position given in the matching GGA line (apriori) (not used by OEMV family receiver) Mode 1 = Residuals were recomputed after the GGA position was computed (preferred mode)	x	1
4 - 15	res	Range residuals for satellites used in the navigation solution. Order matches order of PRN numbers in GPGSA.	x.x,x.x,.....	-13.8,-1.9,11.4,-33.6,0.9, 6.9,-12.6,0.3,0.6, -22.3
16	*xx	Checksum	*hh	*65
17	[CR][LF]	Sentence terminator		[CR][LF]

C.4.14 GPGSA GPS DOP and Active Satellites

This NMEA log provides GPS receiver operating mode, satellites used for navigation and DOP values. See also *Section C.3, NMEA Standard Logs* on *page 169*.

The GPGSA log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

The DOPs provide a simple characterization of the user-satellite geometry. DOP is related to the volume formed by the intersection points of the user-satellite vectors, with the unit sphere centered on the user. Larger volumes give smaller DOPs. Lower DOP values generally represent better position accuracy. The role of DOP in GPS positioning, however, is often misunderstood. A lower DOP value does not automatically mean a low position error. The quality of a GPS-derived position estimate depends upon both the measurement geometry as represented by DOP values, and range errors caused by signal strength, ionospheric effects, multipath and so on.

☒ If the DOP values exceed 9999.0, or there is an insufficient number of satellites to calculate a DOP value, 9999.0 is reported for PDOP and HDOP. VDOP is reported as 0.0 in this case.

Message ID: **221**

Log Type: **Synch**

Recommended Input:

log gpgsa ontime 1

Example 1 (GPS only):

```
$GPGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*35
```

Field	Structure	Field Description	Symbol	Example
1	$GPGSA	Log header		$GPGSA
2	mode MA	A = Automatic 2D/3D M = Manual, forced to operate in 2D or 3D	M	M
3	mode 123	Mode: 1 = Fix not available; 2 = 2D; 3 = 3D	x	3
4 - 15	prn	PRN numbers of satellites used in solution (null for unused fields), total of 12 fields GPS = 1 to 32 SBAS = 33 to 64 (add 87 for PRN number) GLO = 65 to 96 [1]	xx,xx,.....	18,03,13, 25,16, 24,12, 20,,,,
16	pdop	Position dilution of precision	x.x	1.5
17	hdop	Horizontal dilution of precision	x.x	0.9
18	vdop	Vertical dilution of precision	x.x	1.2
19	*xx	Checksum	*hh	*3F
20	[CR][LF]	Sentence terminator		[CR][LF]

1. The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

C.4.15 GPGST Pseudorange Measurement Noise Statistics

This NMEA log provides pseudorange measurement noise statistics. Pseudorange measurement noise statistics are translated in the position domain in order to give statistical measures of the quality of the position solution. See also *Section C.3, NMEA Standard Logs* on *page 169*.

This log reflects the accuracy of the solution type used in BESTGPSPOS, see *page 172*, and GPGGA, see *page 192*, logs except for the RMS field. The RMS field, since it specifically relates to pseudorange inputs, does not represent carrier-phase-based positions. Instead it reflects the accuracy of the pseudorange position.

The GPGST log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

✉ Accuracy is based on statistics, reliability is measured in percent. When a receiver can measure height to one meter, this is an accuracy. Usually this is a one sigma value (one SD). A one sigma value for height has a reliability of 68%, that is, the error is less than one meter 68% of the time. For a more realistic accuracy, double the one sigma value (1 m) and the result is 95% reliability (error is less than 2 m 95% of the time). Generally, GPS heights are 1.5 times poorer than horizontal positions.

As examples of statistics, the GPSGST message and NovAtel performance specifications use root mean square RMS. Specifications may be quoted in CEP:

RMS: Root mean square (a probability level of 68%)

CEP: Circular error probable (the radius of a circle such that 50% of a set of events occur inside the boundary)

Message ID: **222**

Log Type: **Synch**

Recommended Input:

log gpgst ontime 1

Example 1 (GPS only):

```
$GPGST,141451.00,1.18,0.00,0.00,0.0000,0.00,0.00,0.00*6B
```

Field	Structure	Field Description	Symbol	Example
1	$GPGST	Log header		$GPGST
2	utc	UTC time of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	173653.00
3	rms	RMS value of the standard deviation of the range inputs to the navigation process. Range inputs include pseudoranges and DGPS corrections.	x.x	2.73
4	smjr std	Standard deviation of semi-major axis of error ellipse (m)	x.x	2.55
5	smnr std	Standard deviation of semi-minor axis of error ellipse (m)	x.x	1.88
6	orient	Orientation of semi-major axis of error ellipse (degrees from true north)	x.x	15.2525
7	lat std	Standard deviation of latitude error (m)	x.x	2.51
8	lon std	Standard deviation of longitude error (m)	x.x	1.94
9	alt std	Standard deviation of altitude error (m)	x.x	4.30
10	*xx	Checksum	*hh	*6E
11	[CR][LF]	Sentence terminator		[CR][LF]

C.4.16 GPGSV GPS Satellites in View

This NMEA log provides the number of SVs in view, PRN numbers, elevation, azimuth and SNR value. See also *Section C.3, NMEA Standard Logs* on *page 169.*

There are four satellites maximum per message. When required, additional satellite data sent in 2 or more messages (a maximum of 9). The total number of messages being transmitted and the current message being transmitted are indicated in the first two fields.

The GPGSV log outputs these messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

The GPGSV log can be used to determine which satellites are currently available to the receiver. Comparing the information from this log to that in the GPGSA log shows you if the receiver is tracking all available satellites.

---

1. Satellite information may require the transmission of multiple messages. The first field specifies the total number of messages, minimum value 1. The second field identifies the order of this message (message number), minimum value 1.

2. A variable number of 'PRN-Elevation-Azimuth-SNR' sets are allowed up to a maximum of four sets per message. Null fields are not required for unused sets when less than four sets are transmitted.

Message ID: **223**

Log Type: **Synch**

Recommended Input:

log gpgsv ontime 1

Example (Including GPS and GLONASS sentences):

```
$GPGSV,3,1,11,18,87,050,48,22,56,250,49,21,55,122,49,03,40,284,47*78
$GPGSV,3,2,11,19,25,314,42,26,24,044,42,24,16,118,43,29,15,039,42*7E
$GPGSV,3,3,11,09,15,107,44,14,11,196,41,07,03,173,*4D
```

Field	Structure	Field Description	Symbol	Example
1	$GPGSV	Log header		$GPGSV
2	# msgs	Total number of messages (1-9)	x	3
3	msg #	Message number (1-9)	x	1
4	# sats	Total number of satellites in view. May be different than the number of satellites in use (see also the GPSGSA log on *page 199*).	xx	09
5	prn	Satellite PRN number GPS = 1 to 32 SBAS = 33 to 64 (add 87 for PRN#s) GLO = 65 to 96 [1]	xx	03
6	elev	Elevation, degrees, 90 maximum	xx	51
7	azimuth	Azimuth, degrees True, 000 to 359	xxx	140
8	SNR	SNR (C/No) 00-99 dB, null when not tracking	xx	42
... ...	... ...	Next satellite PRN number, elev, azimuth, SNR, ... Last satellite PRN number, elev, azimuth, SNR,		
variable	*xx	Checksum	*hh	*72
variable	[CR][LF]	Sentence terminator		[CR][LF]

1. The NMEA GLONASS PRN numbers are 64 plus the GLONASS slot number. Current slot numbers are 1 to 24 which give the range 65 to 88. PRN numbers 89 to 96 are available if slot numbers above 24 are allocated to on-orbit spares.

C.4.17 GPVTG Track Made Good And Ground Speed

The GPVTG log outputs these messages without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, UTC status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID.

Message ID: **226**
Log Type: **Synch**

Recommended Input:

log gpvtg ontime 1

Example 1 (GPS only):

```
$GPVTG,172.516,T,155.295,M,0.049,N,0.090,K,D*2B
```

Example 2 (Combined GPS and INS):

```
$INVTG,190.919,T,190.919,M,0.856,N,1.585,K,A*31
```

☒ If the NMEATALKER command, see *page 119*, is set to AUTO, the talker (the first 2 characters after the $ sign in the log header) is set to GP (GPS satellites only) or IN (GNSS+INS solution).

Field	Structure	Field Description	Symbol	Example
1	$GPVTG	Log header		$GPVTG
2	track true	Track made good, degrees True	x.x	24.168
3	T	True track indicator	T	T
4	track mag	Track made good, degrees Magnetic;	x.x	24.168
5	M	Magnetic track indicator	M	M
6	speed Kn	Speed over ground, knots	x.x	0.4220347
7	N	Nautical speed indicator (N = Knots)	N	N
8	speed Km	Speed, kilometers/hour	x.x	0.781608
9	K	Speed indicator (K = km/hr)	K	K
10	mode ind	Positioning system mode indicator, see *Table 47* below	a	A
11	*xx	Checksum	*hh	*7A
12	[CR][LF]	Sentence terminator		[CR][LF]

Table 47: NMEA Positioning System Mode Indicator

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

C.4.18 GPZDA UTC Time and Date

This NMEA log outputs messages with contents without waiting for a valid almanac. Instead, it uses a UTC time, calculated with default parameters. In this case, the UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters. Then the UTC time is set to VALID. See also *Section C.3, NMEA Standard Logs* on *page 169*.

Message ID: **227**

Log Type: **Synch**

Recommended Input:

log gpzda ontime 1

Example:

```
$GPZDA,143042.00,25,08,2005,,*6E
```

Field	Structure	Field Description	Symbol	Example
1	$GPZDA	Log header		$GPZDA
2	utc	UTC time	hhmmss.ss	220238.00
3	day	Day, 01 to 31	xx	15
4	month	Month, 01 to 12	xx	07
5	year	Year	xxxx	1992
6	null	Local zone description - not available	xx	(empty when no data is present)
7	null	Local zone minutes description - not available [1]	xx	(empty when no data is present)
8	*xx	Checksum	*hh	*6F
9	[CR][LF]	Sentence terminator		[CR][LF]

1. Local time zones are not supported by OEMV family receivers. Fields 6 and 7 are always null.

C.4.19 INSATT INS Attitude

This log, and the INSATTS log, contains the attitude measurements corresponding to the SPAN computation frame axis. See *Section 3.1, Definition of Reference Frames Within SPAN* on *page 35* for definitions of the frames used in SPAN. The attitude measurements provided by SPAN may not correspond to other definitions of the terms pitch, roll and azimuth. If your IMU's z-axis (as marked on the enclosure) is not pointing up, the output attitude will be with respect to the SPAN computational frame, and not the frame marked on the enclosure. See the SETIMUORIENTATION command, on *page 134,* to determine what the SPAN computation frame will be, given how your IMU is mounted. To output the attitude in the vehicle frame, see *page 91* for information on the APPLYVEHICLEBODYROTATION command.

Structure: **Message ID: 263**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right handed rotation from local level around y-axis[1] in degrees.	Double	8	H+12
5	Pitch	Right handed rotation from local level around x-axis in degrees.	Double	8	H+20
6	Azimuth	Left handed rotation around z-axis. Degrees clockwise from North.	Double	8	H+28
7	Status	INS status, see *Table 5* on *page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. Axis of the SPAN computation frame. If the APPLYVEHICLEBODYROTATION command has been invoked, it will be the axis of the vehicle frame. See *Section 3.1, Definition of Reference Frames Within SPAN* on *page 35* for frame definitions.

Recommended Input:

log insatta ontime 1

ASCII Example:

```
#INSATTA,COM3,0,0.0,EXACT,1105,425385.000,00040000,0638,0;
1105,425384.996167250,4.822147742,0.035766158,123.262113519,
INSSolutionGood*3563a760
```

C.4.20 INSATTS Short INS Attitude

This is a short header version of the *INSATT* log on *page 207*.

Structure: **Message ID: 319**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right handed rotation from local level around y-axis[1] in degrees.	Double	8	H+12
5	Pitch	Right handed rotation from local level around x-axis in degrees.	Double	8	H+20
6	Azimuth	Left handed rotation around z-axis. Degrees clockwise from North.	Double	8	H+28
7	Status	INS status, see *Table 5* on *page 44*.	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. Axis of the SPAN computation frame. If the APPLYVEHICLEBODYROTATION command has been invoked, it will be the axis of the vehicle frame. See *Section 3.1, Definition of Reference Frames Within SPAN* on *page 35* for frame definitions.

Recommended Input:

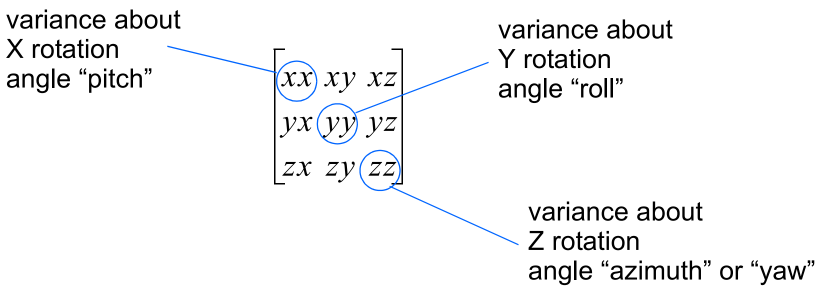
log insattsa ontime 1

ASCII Example:

```
%INSATTSA,1105,425385.000;
1105,425384.996167250,4.822147742,0.035766158,123.262113519,
INSSolutionGood*3563a760
```

C.4.21 INSCOV INS Covariance Matrices

The position, attitude, and velocity matrices in this log each contain 9 covariance values, with respect to the local level frame. For the attitude, the x, y, z axis are of the SPAN Computational Frame. See below for the format of the variance output:



and are displayed within the log output as:

...,xx,xy,xz,yx,yy,yz,zx,zy,zz,...

These values are computed once per second and are only available after alignment. See also *Section 3.4.1, Configuration for Alignment* starting on *page 45* and *Section 3.1, Definition of Reference Frames Within SPAN* on *page 35*.

Structure: **Message ID: 264**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Position Covariance	Position covariance matrix in local level frame (Meters squared)	List of 9 Doubles	72	H+12
5	Attitude Covariance	Attitude covariance matrix in local level frame. (Degrees squared - rotation around the given axis)	List of 9 Doubles	72	H+84
6	Velocity Covariance	Velocity covariance matrix in local level frame. (Meters/second squared)	List of 9 Doubles	72	H+156
7	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+228
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log inscova onchanged

ASCII Example:

```
#INSCOVA,COM3,0,0.0,EXACT,1105,425385.020,00040000,c45c,0;
1105,425385.000000000,
0.0997319969301073,-0.0240959791179416,-0.0133921499963209,
-0.0240959791179416,0.1538605784734939,0.0440068023663888,
-0.0133921499963210,0.0440068023663887,0.4392033415009359,
0.0034190251365443,0.0000759398593357,-0.1362852812808768,
0.0000759398593363,0.0032413999569636,-0.0468473344270137,
-0.1362852812808786,-0.0468473344270131,117.5206493841025100,
0.0004024901765302,-0.0000194916086028,0.0000036582459112,
-0.0000194916086028,0.0004518869575566,0.0000204616202028,
0.0000036582459112,0.0000204616202028,0.0005095575483948*1fc92787
```

C.4.22 INSCOVS Short INS Covariance Log

This is a short header version of the *INCOV* log on *page 209*. These values are also computed once per second.

Structure: **Message ID: 320**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Position Covariance	Position covariance matrix in local level frame. (Meters squared) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+12
5	Attitude Covariance	Attitude covariance matrix in local level frame. (Degrees squared - rotation around the given axis) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+84
6	Velocity Covariance	Velocity covariance matrix in local level frame. (Meters/second squared) xx,xy,xz,yx,yy,yz,zx,zy,zz	List of 9 Doubles	72	H+156
7	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+228
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log inscovsa onchanged

ASCII Example:

```
%INSCOVSA,1105,425385.020;
1105,425385.000000000,
0.0997319969301073,-0.0240959791179416,-0.0133921499963209,
-0.0240959791179416,0.1538605784734939,0.0440068023663888,
-0.0133921499963210,0.0440068023663887,0.4392033415009359,
0.0034190251365443,0.0000759398593357,-0.1362852812808768,
0.0000759398593363,0.0032413999569636,-0.0468473344270137,
-0.1362852812808786,-0.0468473344270131,117.5206493841025100,
0.0004024901765302,-0.0000194916086028,0.0000036582459112,
-0.0000194916086028,0.0004518869575566,0.0000204616202028,
0.0000036582459112,0.0000204616202028,0.0005095575483948*1fc92787
```

C.4.23 INSPOS INS Position

This log contains the most recent position measurements in WGS84 coordinates and includes an INS status indicator. The log reports the position at the IMU centre, unless you issue the SETINSOFFSET command, see *page 144*.

Structure: **Message ID: 265**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84)	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insposa ontime 1

ASCII Example:

```
#INSPOSA,COM3,0,0.0,EXACT,1105,425385.000,00040000,323a,0;
1105,425384.996167250,51.058410364,-114.065465722,
1067.791685696,INSSolutionGood*9bfd5a12
```

C.4.24 INSPOSS Short INS Position

This is a short header version of the *INSPOS* log on *page 212*.

Structure: **Message ID: 321**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84)	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log inspossa ontime 1

ASCII Example:

```
%INSPOSSA,1105,425385.000;
1105,425384.996167250,51.058410364,-114.065465722,
1067.791685696,INSSolutionGood*9bfd5a12
```

C.4.25 INSPOSSYNC Time Synchronised INS Position

This log contains the time synchonised INS position. It is synchronised with GPS each second.

Structure: **Message ID: 322**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Sec	Age of synchronised INS solution (s)	Double	8	H
3	X	ECEF X coordinate	Double	8	H+8
4	Y	ECEF Y coordinate	Double	8	H+16
5	Z	ECEF Z coordinate	Double	8	H+24
6	Cov	ECEF covariance matrix (a 3 x 3 array of length 9).	Double[9]	72	H+32
7	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+104
8	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log inspossynca onchanged

ASCII Example:

```
#INSPOSSYNCA,COM1,0,47.5,FINESTEERING,1332,484154.042,00000000,c98c,34492;
484154.000000000,-1634523.2463,-3664620.7609,4942494.6795,
1.8091616236414247,0.0452272887760925,-0.7438098675219428,
0.0452272887760925,2.9022554471257266,-1.5254793710104819,
-0.7438098675219428,-1.5254793710104819,4.3572293495804546*9fcd6ce1
```

C.4.26 INSPVA INS Position, Velocity and Attitude

This log allows INS position, velocity and attitude to be collected in one log, instead of using three separate logs. The attitude is of the SPAN computation frame by default. See the INSATT log, on *page 207*, for an explanation of how the SPAN frame may differ from the IMU enclosure frame. The attitude can be output in the vehicle frame as well. See the APPLYVEHICLEBODYROTATION command on *page 91*.

Structure: **Message ID: 507**

Log Type: Synch

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84)	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction)	Double	8	H+44
9	Up Velocity	Velocity in an up direction	Double	8	H+52
10	Roll	Right handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left handed rotation around z-axis Degrees clockwise from North	Double	8	H+76
13	Status	INS Status, see *Table 5 on page 44*	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Recommended Input:

log inspvaa ontime 1

ASCII Example:

```
#INSPVAA,COM1,0,31.0,FINESTEERING,1264,144088.000,00040000,5615,1541;
1264,144088.002284950,51.116827527,-114.037738908,401.191547167,
354.846489850,108.429407241,-10.837482850,1.116219952,-3.476059035,
7.372686190,INS_ALIGNMENT_COMPLETE*af719fd9
```

C.4.27 INSPVAS Short INS Position, Velocity and Attitude

This log allows INS position, velocity and attitude to be collected in one log, instead of using three separate logs. The attitude is of the SPAN computation frame by default. See the INSATT log, on *page 207*, for an explanation of how the SPAN frame may differ from the IMU enclosure frame. The attitude can be output in the vehicle frame as well. See the APPLYVEHICLEBODYROTATION command on *page 91*.

Structure: **Message ID: 508**

Log Type: Synch

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds	Seconds from week start	Double	8	H+4
4	Latitude	Latitude (WGS84)	Double	8	H+12
5	Longitude	Longitude (WGS84)	Double	8	H+20
6	Height	Ellipsoidal Height (WGS84)	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (a -ve value implies a southerly direction)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a -ve value implies a westerly direction)	Double	8	H+44
9	Up Velocity	Velocity in an up direction	Double	8	H+52
10	Roll	Right handed rotation from local level around y-axis in degrees	Double	8	H+60
11	Pitch	Right handed rotation from local level around x-axis in degrees	Double	8	H+68
12	Azimuth	Left handed rotation around z-axis Degrees clockwise from North	Double	8	H+76
13	Status	INS Status, see *Table 5 on page 44*	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Recommended Input:

log inspvasa ontime 1

ASCII Example:

```
%INSPVASA,1264,144059.000;
1264,144059.002135700,51.116680071,-114.037929194,515.286704183,
277.896368884,84.915188605,-8.488207941,0.759619515,-2.892414901,
6.179554750,INS_ALIGNMENT_COMPLETE*855d6f76
```

C.4.28 INSSPD INS Speed

This log contains the most recent speed measurements in the horizontal and vertical directions, and includes an INS status indicator.

Structure: **Message ID: 266**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Actual direction of motion over ground (track over ground) with respect to True North, in degrees	Double	8	H+12
5	Horizontal Speed	Magnitude of horizontal speed in m/s where a positive value indicates you are moving forward and a negative value indicates you are reversing.	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in m/s where a positive value indicates speed upward and a negative value indicates speed downward.	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insspda ontime 1

ASCII Example:

```
#INSSPDA,COM3,0,0.0,EXACT,1105,425385.000,00040000,efce,0;
1105,425384.996167250,223.766800423,0.019769837,
-0.024795257,INSSolutionGood*15b864f4
```

C.4.29 INSSPDS Short INS Speed

This is a short header version of the *INSSPD* log on *page 217*.

Structure: **Message ID: 323**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Trk gnd	Track over ground	Double	8	H+12
5	Horizontal Speed	Horizontal speed in m/s	Double	8	H+20
6	Vertical Speed	Vertical speed in m/s	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insspdsa ontime 1

ASCII Example:

```
%INSSPDSA,1105,425385.000;
1105,425384.996167250,223.766800423,0.019769837,
-0.024795257,INSSolutionGood*15b864f4
```

C.4.30 INSUPDATE INS Update

This log contains the most recent INS update information. It gives you information about what updates were performed in the INS filter at the previous update epoch and a wheel sensor status indicator.

Structure: **Message ID: 757**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Solution Type	Type of GPS solution used for the last update, see *Table 38* on *page 172*	Enum	4	H
3	Reserved		Integer	4	H+4
4	#Phase	Number of raw phase observations used in the last INS filter update	Integer	4	H+8
5	Reserved		Integer	4	H+12
6	Zupt Flag	A zero velocity update was performed during the last INS filter update: 0 = False 1 = True	Boolean	2	H+16
7	Wheel Status	Wheel status, see *Table 48* on *page 220*	Ulong	4	H+18
8	Reserved		Ulong	4	H+22
9	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+26
10	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insupdate onchanged

ASCII Example:

```
#INSUPDATEA,UNKNOWN,0,32.5,FINESTEERING,1379,339642.042,00040040,3670,2431;
SINGLE,0,6,0,FALSE,WHEEL_SENSOR_UNSYNCED,0*fb5df08b
```

In this example, the header time is 339642.042. This means the updates (a single point position update and 6 phase updates) were applied at 339641.000.

Table 48: Wheel Status

Binary	ASCII
0	WHEEL_SENSOR_INACTIVE
1	WHEEL_SENSOR_ACTIVE
2	WHEEL_SENSOR_USED
3	WHEEL_SENSOR_UNSYNCED[1]
4	WHEEL_SENSOR_BAD_MISC
5	WHEEL_SENSOR_HIGH_ROTATION

1. WHEEL_SENSOR_USED means the wheel sensor data was applied as an update in the SPAN filter.

C.4.31 INSVEL INS Velocity

This log contains the most recent North, East, and Up velocity vector values, with respect to the local level frame, and also includes an INS status indicator.

Structure: **Message ID: 267**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North in m/s	Double	8	H+12
5	East Velocity	Velocity East in m/s	Double	8	H+20
6	Up Velocity	Velocity Up in m/s	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insvela ontime 1

ASCII Example:

```
#INSVELA,COM3,0,0.0,EXACT,1105,425385.000,00040000,7d4a,0;
1105,425384.996167250,-0.014277009,-0.013675287,
-0.024795257,INSSolutionGood*2f3fe011
```

C.4.32 INSVELS Short INS Velocity

This is a short header version of the *INSVEL* log on *page 221*.

Structure: **Message ID: 324**

Log Type: Synch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	North Velocity	Velocity North m/s	Double	8	H+12
5	East Velocity	Velocity East m/s	Double	8	H+20
6	Up Velocity	Velocity Up m/s	Double	8	H+28
7	Status	INS status, see *Table 5 on page 44*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log insvelsa ontime 1

ASCII Example:

```
%INSVELSA,1105,425385.000;
1105,425384.996167250,-0.014277009,-0.013675287,
-0.024795257,INSSolutionGood*2f3fe011
```

C.4.33 LBANDINFO L-band Configuration Information

This log outputs configuration information for an L-band service. In the case of using the free CDGPS service, no subscription is required and therefore the subscription fields report an UNKNOWN subscription status. See also the examples below.

☒ In addition to a NovAtel receiver with L-band capability, a subscription to the OmniSTAR, or use of the free CDGPS, service is required. Contact NovAtel for details. Contact information may be found on the back of this manual or you can refer to the *Customer Service* section in the *OEMV Family Installation and Operation User Manual*.

Message ID: **730**

Log Type: **Asynch**

Recommended Input:

log lbandinfoa ontime 1

ASCII Example 1 (OmniSTAR HP):

```
#LBANDINFOA,COM2,0,81.5,FINESTEERING,1295,152639.184,00000240,c51d,34461;
1547547,4800,c685,0,762640,EXPIRED,0,0,FIXEDTIME,1199,259199,0*8cc5e573
```

Abbreviated ASCII Example 2 (CDGPS):

```
LBANDINFO COM1 0 45.5 FINESTEERING 1297 498512.389 00000000 c51d 34486
1547547 4800 0 0 762640 UNKNOWN 0 0 UNKNOWN 0 0 0
```

Table 49: L-band Subscription Type

Binary	ASCII	Description
0	EXPIRED	The L-band subscription has expired or does not exist.
1	FIXEDTIME	The L-band subscription expires at a fixed date and time.
2	COUNTDOWN	The L-band subscription expires after the specified amount of running time.
3	COUNTDOWNOVERRUN	The COUNTDOWN subscription has expired but has entered a brief grace period. Resubscribe immediately.
16	UNKNOWN	Unknown subscription

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LBANDINFO header	Log header		H	0
2	freq	Selected frequency for L-band service (kHz)	Ulong	4	H
3	baud	Communication baud rate from L-band satellite	Ulong	4	H+4
4	ID	L-band signal service ID	Ushort	2	H+8
5	Reserved		Ushort	2	H+10
6	OSN	L-band serial number	Ulong	4	H+12
7	vbs sub	L-band VBS subscription type (see *Table 49* on *page 223*)	Enum	4	H+16
8	vbs exp week	GPS week number of L-band VBS expiration date [1]	Ulong	4	H+20
9	vbs exp secs	Number of seconds into the GPS week of L-band VBS expiration date [a]	Ulong	4	H+24
10	hp sub	OmniSTAR HP or XP subscription type (see *Table 49* on *page 223*)	Enum	4	H+28
11	hp exp week	GPS week number of OmniSTAR HP or XP expiration date [a]	Ulong	4	H+32
12	hp exp secs	Number of seconds into the GPS week of OmniSTAR HP or XP expiration date [a]	Ulong	4	H+36
13	hp sub mode	HP or XP subscription mode if the subscription is valid: 0 = HP 1 = XP	Ulong	4	H+40
14	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. If the subscription type is COUNTDOWN, see Field #7 above, the expiration week and expiration seconds into the GPS week contain the amount of running time remaining in the subscription.

   If the subscription type is COUNTDOWNOVERRUN, the expiration week and expiration seconds into GPS week count the amount of the overrun time.

C.4.34 LBANDSTAT L-band Status Information

This log outputs status information for a standard L-band, OmniSTAR XP (Extra Precision) or OmniSTAR HP (High Performance) service.

1. In addition to a NovAtel receiver with L-band capability, a subscription to the OmniSTAR, or use of the free CDGPS, service is required. Contact NovAtel for details.
2. In binary, the receiver outputs 48 bytes without the checksum when the LBANDSTATB log is requested.

Message ID: 731
Log Type: Asynch

Recommended Input:

log lbandstata ontime 1

ASCII Example:

```
#LBANDSTATA,COM1,0,73.5,FINESTEERING,1314,494510.000,00000000,c797,1846;
1551488896,43.19,62.3,0.00,0082,0000,7235,11,0,0000,0001,7762,04000000,0
*93f7d2af
```

Table 50: L-band Signal Tracking Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x0001	Tracking State	0 = Searching, 1 = Pull-in, 2 = Tracking, 3 = Idle
	1	0x0002		
	2	0x0004	Reserved	
	3	0x0008		
N1	4	0x0010		
	5	0x0020		
	6	0x0040	Bit Timing Lock	0 = Not Locked, 1 = Locked
	7	0x0080	Phase Locked	0 = Not Locked, 1 = Locked
N2	8	0x0100	DC Offset Unlocked	0 = Good, 1 = Warning
	9	0x0200	AGC Unlocked	0 = Good, 1 = Warning
	10	0x0400	Reserved	
	11	0x0800		
N3	12	0x1000		
	13	0x2000		
	14	0x4000		
	15	0x8000	Error	0 = Good, 1 = Error

Table 51: OmniSTAR VBS Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0001	Subscription Expired [1]	False	True
	1	0x0002	Out of Region [1]	False	True
	2	0x0004	Wet Error [1]	False	True
	3	0x0008	Link Error [1]	False	True
N1	4	0x0010	No Remote Sites	False	True
	5	0x0020	No Almanac	False	True
	6	0x0040	No Position	False	True
	7	0x0080	No Time	False	True
N2	8	0x0100	Reserved		
	9	0x0200			
	10	0x0400			
	11	0x0800			
N3	12	0x1000			
	13	0x2000			
	14	0x4000			
	15	0x8000	Updating Data	False	True

1. Contact OmniSTAR for subscription support. All other status values are updated by collecting OmniSTAR data for 20-35 minutes.

Table 52: OmniSTAR HP/XP Additional Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0001	Solution not fully converged	False	True
	1	0x0002	OmniStar satellite list available	False	True
	2	0x0004	Reserved		
	3	0x0008			
N1	4	0x0010	HP not authorized [1]	Authorized	Unauthorized
	5	0x0020	XP not authorized [1]	Authorized	Unauthorized
	6	0x0040	Reserved		
	7	0x0080			
N2	8	0x0100			
	9	0x0200			
	10	0x0400			
	11	0x0800			
N3	12	0x1000			
	13	0x2000			
	14	0x4000			
	15	0x8000			

1. This authorization is related to the receiver model and not the OmniStar subscription. To view OmniSTAR subscription information use the LBANDINFO log, see *page 223*.

Table 53: OmniSTAR HP/XP Status Word

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Subscription Expired [1]	False	True
	1	0x00000002	Out of Region [1]	False	True
	2	0x00000004	Wet Error [1]	False	True
	3	0x00000008	Link Error [1]	False	True
N1	4	0x00000010	No Measurements	False	True
	5	0x00000020	No Ephemeris	False	True
	6	0x00000040	No Initial Position	False	True
	7	0x00000080	No Time Set	False	True
N2	8	0x00000100	Velocity Error	False	True
	9	0x00000200	No base stations	False	True
	10	0x00000400	No Mapping Message	False	True
	11	Reserved			
N3-N5	12-23				
N6	24-25				
	26	0x04000000	Static Initialization Mode	False	True
	27	Reserved			
N7	28-30				
	31	0x80000000	Updating Data	False	True

1. Contact OmniSTAR for subscription support. All other status values are updated by collecting the OmniSTAR data for 20-35 minutes.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	LBANDSTAT header	Log header		H	0
2	freq	Measured frequency of L-band signal (Hz)	Ulong	4	H
3	C/No	Carrier to noise density ratio C/No = 10[$\log_{10}(S/N_0)$] (dB-Hz)	Float	4	H+4
4	locktime	Number of seconds of continuous tracking (no cycle slipping)	Float	4	H+8
5	Reserved		Float	4	H+12
6	tracking	Tracking status of L-band signal (see *Table 50* on *page 226*)	Hex	2	H+16
7	VBS status	Status word for OmniSTAR VBS (see *Table 51* on *page 227*)	Hex	2	H+18
8	#bytes	Number of bytes fed to the standard process	Ulong	4	H+20
9	#good dgps	Number of standard updates	Ulong	4	H+24
10	#bad data	Number of missing standard updates	Ulong	4	H+28
11	Reserved (the *hp status 1* field is obsolete and has been replaced by the longer OmniSTAR HP Status field. The shorter legacy status here is maintained for backward compatibility)		Hex	2	H+32
12	hp status 2	Additional status pertaining to the HP or XP process (see *Table 52* on *page 228*)	Hex	2	H+34
13	#bytes hp	Number of bytes fed to the HP or XP process	Ulong	4	H+36
14	hp status	Status from the HP or XP process (see *Table 53* on *page 229*)	Hex	4	H+40
15	Reserved		Hex	4	H+44
16	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+48
17	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.35 LOGLIST List of System Logs

Outputs a list of log entries in the system. The following tables show the binary ASCII output. See also the RXCONFIG log on *page 254* for a list of current command settings.

Message ID: 5

Log Type: Polled

Recommended Input:

log loglista once

ASCII Example:

```
#LOGLISTA,COM1,0,93.5,FINESTEERING,1521,319135.030,00000000,0000,149;
7,
COM1,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM2,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM3,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM4,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD,
COM1,LOGLISTA,ONCE,0.000000,0.000000,NOHOLD,
COM2,RAWIMUSB,ONNEW,0.000000,0.000000,NOHOLD,
COM2,INSPVASB,ONTIME,0.020000,0.000000,NOHOLD*21ed4ccd
```

WARNING!: **Do not use undocumented logs or commands! Doing so may produce errors and void your warranty.**

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	LOGLIST (binary) header	Log header		H	0
2	#logs	Number of messages to follow, maximum = 20	Long	4	H
3	port	Output port, see *Table 18, COM Serial Port Identifiers* on *page 96*	Enum	4	H+4
4	message	Message ID of log	Ushort	2	H+8
5	message type	Bits 0-4 = Reserved Bits 5-6 = Format 00 = Binary 01 = ASCII 10 = Abbreviated ASCII, NMEA 11 = Reserved Bit 7 = Response Bit 0 = Original Message 1 = Response Message	Char	1	H+10
6	reserved		Char	3[1]	H+11
7	trigger	0 = ONNEW 1 = ONCHANGED 2 = ONTIME 3 = ONNEXT 4 = ONCE	Enum	4	H+14
8	period	Log period for ONTIME	Double	8	H+18
9	offset	Offset for period (ONTIME trigger)	Double	8	H+26
10	hold	0 = NOHOLD 1 = HOLD	Enum	4	H+34
11...	Next log offset = H + 4 + (#logs x 34)				
variable	xxxx	32-bit CRC	Hex	4	H+4+(#logs x 34)

1. In the binary log case, an additional 2 bytes of padding are added to maintain 4-byte alignment

Field #	Field type	Data Description	Format
1	LOGLIST (ASCII) header	Log header	
2	#port	Number of messages to follow, maximum = 20	Long
3	port	Output port, see *Table 18, COM Serial Port Identifiers* on *page 96*	Enum
4	message	Message name of log with no suffix for abbreviated ascii, an A suffix for ascii and a B suffix for binary.	Char [ ]
5	trigger	ONNEW ONCHANGED ONTIME ONNEXT ONCE	Enum
6	period	Log period for ONTIME	Double
7	offset	Offset for period (ONTIME trigger)	Double
8	hold	NOHOLD HOLD	Enum
9...	Next port		
variable	xxxx	32-bit CRC	Hex
variable	[CR][LF]	Sentence terminator	-

C.4.36 MAC MAC Address

This log displays the SPAN-SE's Media Access Control (MAC) address. See also *Section 3.11, SPAN-SE Ethernet Connection* on *page 60*.

The 6-byte MAC address is typically spaced with colons. The first 3 bytes are the same numbers for every SPAN-SE and are registered to NovAtel. The second three bytes are specific to each SPAN-SE.

Structure: **Message ID: 1100**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	MAC address	6 MAC address numbers separated by colons	Uchar	6	H
3	Protocol	Protocol 0 = UDP 1 = TCP	Enum	4	H+4
4	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+8
5	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log maca once

Abbreviated ASCII Example:

```
#MACA,COM4,0,98.0,FINESTEERING,1522,327807.461,40000020,0000,159;
"00:21:66:00:01:91",TCP*35b51c40
```

C.4.37 MARK1COUNT, MARK2COUNT, MARK3COUNT, MARK4COUNT Mark Count

When the input mode is set to COUNT using the EVENTINCONTROL command, see *page 102*, the MARKxCOUNT logs become available.

---

1. Use the ONNEW trigger with this, the MARK*x*TIME, or the MARK*x*PVA logs.

2. Only the MARK*x*COUNT, MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

MARK1COUNT Message ID: 1093
MARK2COUNT Message ID: 1094
MARK3COUNT Message ID: 1095
MARK4COUNT Message ID: 1096 **Log Type: Asynch**

Recommended Input:

log mark1counta onnew

Example:

```
#MARK1COUNTA,COM1,0,98.5,FINESTEERING,1520,515353.000,00000000,0000,137;
1000000,1*1786750b
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MARK*x*COUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4

C.4.38 MARK1PVA, MARK2PVA, MARK3PVA, MARK4PVA Position, Velocity and Attitude at Mark

This log outputs position, velocity and attitude information received on a Mark input. By default, the MARK*x*PVA logs contain the solution at the IMU centre in the SPAN computation frame. If the SETMARK*x*OFFSET command has been entered, the MARK*x*PVA log will contain the solution translated, and then rotated, by the values provided in the command. See also the SETMARK*x*OFFSET commands, valid at the time, on *page 145*.

MARK1PVA Message ID: 1067
MARK2PVA Message ID: 1068
MARK3PVA Message ID: 1118
MARK4PVA Message ID: 1119 **Log Type: Synch**

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week at Mark1, 2, 3 or 4 request	Ulong	4	H
3	Seconds	Seconds from week at Mark1, 2, 3 or 4	Double	8	H+4
4	Latitude	Latitude (WGS84) at Mark1, 2, 3 or 4	Double	8	H+12
5	Longitude	Longitude (WGS84) at Mark1, 2, 3 or 4	Double	8	H+20
6	Height	Height (WGS84) at Mark1, 2, 3 or 4	Double	8	H+28
7	North Velocity	Velocity in a northerly direction (-ve implies a southerly direction) at Mark1, 2, 3 or 4	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (-ve implies a westerly direction) at Mark1, 2, 3 or 4	Double	8	H+44
9	Up Velocity	Velocity in an up direction at Mark1, 2, 3 or 4	Double	8	H+52
10	Roll	Right handed rotation from local level around y-axis in degrees at Mark1, 2, 3 or 4	Double	8	H+60
11	Pitch	Right handed rotation from local level around x-axis in degrees at Mark1, 2, 3 or 4	Double	8	H+68
12	Azimuth	Left handed rotation around z-axis Degrees clockwise from North at Mark1, 2, 3 or 4	Double	8	H+76
13	Status	INS Status, see *Table 5 on page 44* at Mark	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

Recommended Input:

log mark1pva onnew 1

Abbreviated ASCII Example:

```
MARK1PVA USB1 0 51.5 EXACT 1481 251850.001 00040000 46f4 3388
1481 251850.001000000 51.116573435 -114.037237211 1040.805671970 0.000257666
-0.003030102 -0.000089758 3.082229474 -1.019023628 89.253955744
INS_SOLUTION_GOOD
```

C.4.39 MARK1TIME, MARK2TIME, MARK3TIME, MARK4TIME Time of Mark Input Event

This log contains the time of the leading edge of the detected mark input pulse. MARK1TIME gives the time when a pulse occurs on the MK1I input, MARK2POS is generated when a pulse occurs on a MK2I input and so on.

These logs allow you to measure the time when events are occurring in other devices (such as a video recorder). See also the SETMARK*x*OFFSET commands starting on *page 145*.

1. Use the ONNEW trigger with this or the MARK*x*PVA logs.

2. Only the MARK*x*PVA logs, the MARK*x*TIME logs, and 'polled' log types are generated 'on the fly' at the exact time of the mark. Synchronous and asynchronous logs output the most recently available data.

MARK1TIME Message ID: 1130
MARK2TIME Message ID: 616
MARK3TIME Message ID: 1075
MARK4TIME Message ID: 1076 **Log Type: Asynch**

Recommended Input:

log mark1timea onnew

Example:

```
#MARK1TIMEA,COM1,0,98.0,FINESTEERING,1521,336487.000,00000000,0000,149;1521,3
36487.000000025,0.000000000,0.000000000,-14.999999992,VALID*7597ecee

#MARK2TIMEA,COM1,0,98.5,FINESTEERING,1521,336487.000,00000000,0000,149;1521,3
36487.000000025,0.000000000,0.000000000,-14.999999992,VALID*8fd08ef6

#MARK3TIMEA,COM1,0,98.5,FINESTEERING,1521,336487.000,00000000,0000,149;1521,3
36487.000000025,0.000000000,0.000000000,-14.999999992,VALID*ed342f79
```

Table 54: Clock Model Status

Clock Status (Binary)	Clock Status (ASCII)	Description
0	VALID	The clock model is valid
1	CONVERGING	The clock model is near validity
2	ITERATING	The clock model is iterating towards validity
3	INVALID	The clock model is not valid
4	ERROR	Clock model error

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	MARK*x*TIME header	Log header		H	0
2	week	GPS week number	Long	4	H
3	seconds	Seconds into the week as measured from the receiver clock, coincident with the time of electrical closure on the Mark Input port.	Double	8	H+4
4	offset	Receiver clock offset, in seconds. A positive offset implies that the receiver clock is ahead of GPS Time. To derive GPS time, use the following formula: GPS time = receiver time - (offset)	Double	8	H+12
5	offset std	Standard deviation of receiver clock offset (s)	Double	8	H+20
6	utc offset	This field represents the offset of GPS time from UTC time, computed using almanac parameters. UTC time is GPS time plus the current UTC offset plus the receiver clock offset. UTC time = GPS time + offset + UTC offset[1]	Double	8	H+28
7	status	Clock model status, see *Table 54, Clock Model Status* on *page 237*	Enum	4	H+36
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+40
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. 0 indicates that UTC time is unknown because there is no almanac available in order to acquire the UTC offset.

C.4.40 PASHR NMEA, inertial attitude data

The PASHR log uses a UTC time, calculated with default parameters, to output NMEA messages without waiting for a valid almanac. The UTC time status is set to WARNING since it may not be 100% accurate. When a valid almanac is available, the receiver uses the real parameters and sets the UTC time to VALID. For more information about NMEA, refer to the *OEMV Firmware Reference Manual* found on our Web site. The PASHR log contains only INS derived attitude information and is only filled when an inertial solution is available.

Structure: **Message ID: 1177**

Log TypeSynch

Field	Structure	Field Description	Symbol	Example
1	$PASHR	Log Header	---	$PASHR
2	Time	UTC Time	hhmmss.ss	195124.00
3	Heading	Heading value in decimal degrees	HHH.HH	305.30
4	True Heading	T displayed if heading is relative to true north.	T	T
5	Roll	Roll in decimal degrees. The +/- sign will always be displayed.	RRR.RR	+0.05
6	Pitch	Pitch in decimal degrees. The +/- sign will always be displayed.	PPP.PP	-0.13
7	Reserved	------	----	----
8	Roll Accuracy	Roll standard deviation in decimal degrees.	rr.rrr	0.180
9	Pitch Accuracy	Pitch standard deviation in decimal degrees.	pp.ppp	0.185
10	Heading Accuracy	Heading standard deviation in decimal degrees.	hh.hhh	4.986
11	GPS Update Quality Flag	0 = No position 1 = All non-RTK fixed integer positions 2 = RTK fixed integer position	1	1
12	Checksum	Checksum	*XX	*2B
13	[CR][LF]	Sentence terminator		[CR][LF]

Recommended Input:

log pashr ontime 1

Example:

```
$PASHR,,,,,,,,,,0*68 (empty)
$PASHR,195124.00,305.30,T,+0.05,-0.13,,0.180,0.185,4.986,1*2B
```

C.4.41 PORTSTATS Port Statistic

This log conveys various status parameters of the receiver's communication ports. The receiver maintains a running count of a variety of status indicators of the data link. This log outputs a report of those indicators.

Message ID: **72**

Log Type: **Polled**

Recommended Input:

log portstatsa once

ASCII Example:

```
#PORTSTATSA,COM1,0,94.5,FINESTEERING,1521,319328.143,00000000,0000,149;
7,
COM1,101688,552,552,0,0,1074394,0,0,0,
COM2,155749,331,331,0,0,2712888,0,0,0,
COM3,1213,34,34,0,0,28728,0,0,0,
COM4,936,36,36,0,0,22784,0,0,0,
IMU,1194365,13190643,13190643,179423,0,0,0,0,0,
USB1,0,0,0,0,0,0,0,0,0,
ETH1,0,0,0,0,0,0,0,0,0*a54c453f
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	PORTSTATS header	Log header		H	0
2	#port	Number of ports with information to follow	Long	4	H
3	port	Serial port identifier, see *Table 18, COM Serial Port Identifiers* on *page 96*	Enum	4	H+4
4	rx chars	Total number of characters received through this port	Ulong	4	H+8
5	tx chars	Total number of characters transmitted through this port	Ulong	4	H+12
6	acc rx chars	Total number of accepted characters received through this port	Ulong	4	H+16
7	dropped chars	Number of software overruns	Ulong	4	H+20
8	interrupts	Number of interrupts on this port	Ulong	4	H+24
9	breaks	Number of breaks (This field does not apply for a USB port and is always set to 0 for USB.)	Ulong	4	H+28
10	par err	Number of parity errors (This field does not apply for a USB port and is always set to 0 for USB.)	Ulong	4	H+32
11	fram err	Number of framing errors (This field does not apply for a USB port and is always set to 0 for USB.)	Ulong	4	H+36
12	overruns	Number of hardware overruns	Ulong	4	H+40
13	Next port offset = H + 4 + (#port x 40)				
14	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#port x 40)
15	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.42 RANGE Satellite Range Information

RANGE contains the channel measurements for the currently tracked satellites. When using this log, please keep in mind the constraints noted along with the description.

It is important to ensure that the receiver clock has been set. This can be monitored by the bits in the *Receiver Status* field of the log header. Large jumps in pseudorange as well as accumulated Doppler range (ADR) occur as the clock is being adjusted. If the ADR measurement is being used in precise phase processing, it is important not to use the ADR if the "parity known" flag in the *ch-tr-status* field is not set as there may exist a half (1/2) cycle ambiguity on the measurement. The tracking error estimate of the pseudorange and carrier phase (ADR) is the thermal noise of the receiver tracking loops only. It does not account for possible multipath errors or atmospheric delays.

If both the L1 and L2 signals are being tracked for a given PRN, two entries with the same PRN appear in the range logs. As shown in *Table 58, Channel Tracking Status* on *page 244*, these entries can be differentiated by bit 20, which is set if there are multiple observables for a given PRN, and bits 21-22, which denotes whether the observation is for L1 or L2. This is to aid in parsing the data.

Message ID: **43**

Log Type: **Synch**

Recommended Input:

log rangea ontime 30

ASCII Example:

```
#RANGEA,COM1,0,63.5,FINESTEERING,1429,226979.000,00000000,5103,2748;
26,
6,0,23359924.081,0.078,-122757217.106875,0.015,-3538.602,43.3,19967.080,
08109c04,
6,0,23359926.375,0.167,-95654966.812027,0.019,-2757.355,36.7,19960.461,
01309c0b,
21,0,20200269.147,0.038,-106153137.954409,0.008,-86.289,49.5,13397.470,
08109c44,
21,0,20200268.815,0.056,-82716721.366921,0.008,-67.242,46.1,13391.980,
01309c4b,
16,0,23945650.428,0.091,-125835245.287192,0.024,-2385.422,41.9,10864.640,
08109c64,
16,0,23945651.399,0.148,-98053428.283142,0.028,-1858.773,37.7,10859.980,
01309c6b,
.
.
.
44,12,19388129.378,0.335,-103786179.553598,0.012,975.676,36.6,3726.656,
18119e24,
44,12,19388136.659,0.167,-80722615.862096,0.000,758.859,42.7,3714.860,
10b19e2b,
43,8,20375687.399,0.253,-108919708.904476,0.012,-2781.090,39.1,10629.934,
18119e84,
43,8,20375689.555,0.177,-84715349.232514,0.000,-2163.074,42.2,10619.916,
10b19e8b*fd2d3125
```

✉ On SPAN-SE, it is recommended the RANGE log be requested in binary only, especially if high rates are desired. An ASCII example is shown above for clarity and consistency.

Table 55: Tracking State

State	Description	State	Description
0	L1 Idle	7	L1 Frequency-lock loop
1	L1 Sky search	8	L2 Idle
2	L1 Wide frequency band pull-in	9	L2 P-code alignment
3	L1 Narrow frequency band pull-in	10	L2 Search
4	L1 Phase lock loop	11	L2 Phase lock loop
5	L1 Reacquisition	19	L2 Steering
6	L1 Steering		

Table 56: Correlator Type

State	Description
0	N/A
1	Standard correlator: spacing = 1 chip
2	Narrow Correlator: spacing < 1 chip
3	Reserved
4	Pulse Aperture Correlator (PAC)
5-6	Reserved

Table 57: Channel Tracking Example

	0x	Bit #	Binary[1]	Data	Value
N7	0	31	0	Chan. Assignment	Automatic
		30	0	Reserved (R)	
		29	0		
		28	0		
N6	8	27	1	Primary L1	Primary
		26	0	R	
		25	0	Signal Type	L1 C/A
		24	0		
N5	1	23	0		
		22	0		
		21	0		
		20	1	Grouping	Grouped
N4	0	19	0	R	
		18	0	Satellite System	GPS
		17	0		
		16	0		
N3	9	15	1	Correlator Spacing	PAC
		14	0		
		13	0		
		12	1	Code locked flag	Locked
N2	C	11	1	Parity flag	Known
		10	1	Phase lock flag	Locked
		9	0	Channel Number	Channel 0
		8	0		
N1	0	7	0		
		6	0		
		5	0		
		4	0	Tracking State	L1 Phase Lock Loop
N0	4	3	0		
		2	1		
		1	0		
		0	0		

1. For a complete list of hexadecimal and binary equivalents please refer to the conversions section of the *GNSS Reference Book*, available on our Web site at http://www.novatel.com/support/docupdates.htm.

Table 58: Channel Tracking Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Tracking state	0-11, see *Table 55, Tracking State* on *page 243*
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010		
	5	0x00000020	SV channel number	0-n (0 = first, n = last) n depends on the receiver
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100		
	9	0x00000200		
	10	0x00000400	Phase lock flag	0 = Not locked, 1 = Locked
	11	0x00000800	Parity known flag	0 = Not known, 1 = Known
N3	12	0x00001000	Code locked flag	0 = Not locked, 1 = Locked
	13	0x00002000	Correlator type	0-7, see *Table 56, Correlator Type* on *page 243*
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Satellite system	0 = GPS 1= GLONASS 2 = WAAS 3-6 = Reserved 7 = Other
	17	0x00020000		
	18	0x00040000		
	19	0x00080000	Reserved	
N5	20	0x00100000	Grouping [1]	0 = Not grouped, 1 = Grouped
	21	0x00200000	Signal type	Dependant on satellite system above: GPS: 0 = L1 C/A, 5 = L2 P, 9 = L2 P codeless, 17 = L2C GLONASS: 0 = L1 C/A, 5 = L2 P SBAS: 0 = L1 C/A Other: 19 = OmniSTAR, 23 = CDGPS
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000	Forward Error Correction	0 = Not FEC, 1 = FEC
	27	0x08000000	Primary L1 channel	0 = Not primary, 1 = Primary

Continued on page 245

Nibble #	Bit #	Mask	Description	Range Value
N7	28	0x10000000	Carrier phase measurement [2]	0 = Half Cycle Not Added, 1 = Half Cycle Added
	29	Reserved		
	30	0x40000000	PRN lock flag	0 = PRN Not Locked Out, 1 = PRN Locked Out
	31	0x80000000	Channel assignment	0 = Automatic, 1 = Forced

1. Grouped: Channel has an associated channel (L1/L2 pairs)
2. This bit is zero until the parity is known and the parity known flag (bit 11) is set to 1.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RANGE header	Log header		H	0
2	# obs	Number of observations with information to follow [1]	Long	4	H
3	PRN/slot	Satellite PRN number of range measurement (GPS: 1 to 32, SBAS: 120 to 138, and GLONASS: 38 to 61)	UShort	2	H+4
4	glofreq	(GLONASS Frequency + 7)	UShort	2	H+6
5	psr	Pseudorange measurement (m)	Double	8	H+8
6	psr std	Pseudorange measurement standard deviation (m)	Float	4	H+16
7	adr	Carrier phase, in cycles (accumulated Doppler range)	Double	8	H+20
8	adr std	Estimated carrier phase standard deviation (cycles)	Float	4	H+28
9	dopp	Instantaneous carrier Doppler frequency (Hz)	Float	4	H+32
10	C/No	Carrier to noise density ratio $C/No = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	# of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see *Table 58, Channel Tracking Status* on *page 244* and the example in *Table C.4.46*)	ULong	4	H+44
13...	Next PRN offset = H + 4 + (#obs x 44)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#obs x 44)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. Satellite PRNs may have two lines of observations, one for the L1 frequency and the other for L2.

C.4.43 RANGECMP Compressed Version of the RANGE Log

Message ID: **140**

Log Type: **Synch**

Recommended Input:

log rangecmpa ontime 10

Example:

```
#RANGECMPA,COM1,0,63.5,FINESTEERING,1429,226780.000,00000000,9691,2748;
26,
049c10081857f2df1f4a130ba2888eb9600603a709030000,
0b9c3001225bf58f334a130bb1e2bed473062fa609020000,
449c1008340400e0aaa9a109a7535bac2015cf71c6030000,
4b9c300145030010a6a9a10959c2f09120151f7166030000,
...
0b9d301113c8ffefc284000c6ea051dbf3089da1a0010000,
249d1018c6b7f67fa228820af2e5e39830180ae1a8030000,
2b9d301165c4f8ffb228820a500a089f31185fe0a8020000,
449d1018be18f41f2aacad0a1a934efc40074ecf88030000,
4b9d301182b9f69f38acad0a3e3ac28841079fcb88020000,
849d101817a1f95f16d7af0a69fbe1fa401d3fd064030000,
8b9d30112909fb2f20d7af0a9f24a687521ddece64020000,
249e1118af4e0470f66d4309a0a631cd642cf5b821320000,
2b9eb110a55903502f6e4309ee28d1ad032c7cb7e1320000,
849e1118b878f54f4ed2aa098c35558a532bde1765220000,
8b9eb110abcff71f5ed2aa09cb6ad0f9032b9d16c5220000*0eeead18
```

Table 59: Range Record Format (RANGECMP only)

Data	Bit(s) first to last	Length (bits)	Scale Factor	Units
Channel Tracking Status	0-31	32	see *Table 58, Channel Tracking Status* on *page 244*	-
Doppler Frequency	32-59	28	1/256	Hz
Pseudorange (PSR)	60-95	36	1/128	m
ADR [1]	96-127	32	1/256	cycles
StdDev-PSR	128-131	4	see [2]	m
StdDev-ADR	132-135	4	(n + 1)/512	cycles
PRN/Slot [3]	136-143	8	1	-
Lock Time [4]	144-164	21	1/32	s
C/No [5]	165-169	5	(20 + n)	dB-Hz
Reserved	170-191	22		

1. ADR (Accumulated Doppler Range) is calculated as follows:

```
ADR_ROLLS = (RANGECMP_PSR / WAVELENGTH + RANGECMP_ADR) / MAX_VALUE
Round to the closest integer
IF (ADR_ROLLS ≤ 0)
        ADR_ROLLS = ADR_ROLLS - 0.5
ELSE
        ADR_ROLLS = ADR_ROLLS + 0.5
At this point integerise ADR_ROLLS
CORRECTED_ADR = RANGECMP_ADR - (MAX_VALUE*ADR_ROLLS)
where
ADR has units of cycles
WAVELENGTH = 0.1902936727984 for GPS L1
WAVELENGTH = 0.2442102134246 for GPS L2
MAX_VALUE = 8388608
```

Note: *GLONASS satellites emit L1 and L2 carrier waves at a satellite-specific frequency, refer to the GNSS Reference Book for more on GLONASS frequencies.*

2.

Code	StdDev-PSR (m)
0	0.050
1	0.075
2	0.113
3	0.169
4	0.253
5	0.380
6	0.570
7	0.854
8	1.281
9	2.375
10	4.750
11	9.500
12	19.000
13	38.000
14	76.000
15	152.000

3. GPS: 1 to 32, SBAS: 120 to 138, and GLONASS: 38 to 61, see *Section 1.1* on *page 23*.
4. The *Lock Time* field of the RANGECMP log is constrained to a maximum value of 2,097,151 which represents a lock time of 65535.96875 s (2097151 ÷ 32).
5. C/No is constrained to a value between 20-51 dB-Hz. Thus, if it is reported that C/No = 20 dB-Hz, the actual value could be less. Likewise, if it is reported that C/No = 51, the true value could be greater.

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	RANGECMP header	Log header		H	0
2	#obs	Number of satellite observations with information to follow.	Long	4	H
3	1st range record	Compressed range log in format of *Table 59* on *page 246*	Hex	24	H+4
4	Next rangecmp offset = H + 4 + (#obs x 24)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H + 4 + (#obs x 24)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.44 RAWEPHEM Raw Ephemeris

This log contains the raw binary information for subframes one, two and three from the satellite with the parity information removed. Each subframe is 240 bits long (10 words - 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 subframes). This information is preceded by the PRN number of the satellite from which it originated. This message is not generated unless all 10 words from all 3 frames have passed parity.

Ephemeris data whose TOE (Time Of Ephemeris) is older than six hours is not shown.

Message ID: 41

Log Type: Asynch

Recommended Input:

log rawephema onnew

ASCII Example:

```
#RAWEPHEMA,COM1,15,60.5,FINESTEERING,1337,405297.175,00000000,97b7,1984;
3,1337,403184,8b04e4818da44e50007b0d9c05ee664ffbfe695df763626f00001b03c6b3,
8b04e4818e2b63060536608fd8cdaa051803a41261157ea10d2610626f3d,
8b04e4818ead0006aa7f7ef8ffda25c1a69a14881879b9c6ffa79863f9f2*0bb16ac3
.
.
.
#RAWEPHEMA,COM1,0,60.5,SATTIME,1337,405390.000,00000000,97b7,1984;
1,1337,410400,8b04e483f7244e50011d7a6105ee664ffbfe695df9e1643200001200aa92,
8b04e483f7a9e1faab2b16a27c7d41fb5c0304794811f7a10d40b564327e,
8b04e483f82c00252f57a782001b282027a31c0fba0fc525ffac84e10a06*c5834a5b
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RAWEPHEM header	Log header		H	0
2	prn	Satellite PRN number	Ulong	4	H
3	ref week	Ephemeris reference week number	Ulong	4	H+4
4	ref secs	Ephemeris reference time (s)	Ulong	4	H+8
5	subframe1	Subframe 1 data	Hex	30	H+12
6	subframe2	Subframe 2 data	Hex	30	H+42
7	subframe3	Subframe 3 data	Hex	30	H+72
8	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+102
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.45 RAWIMU Raw IMU Data

This log contains an IMU status indicator and the measurements from the accelerometers and gyros with respect to the IMU enclosure frame. If logging this data, consider the RAWIMUS log to reduce the amount of data, see *page 252*.

Structure: **Message ID: 268**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status of the HG1700 and the LN-200 IMUs, refer to the Interface Control Documentation as provided by Honeywell and Northorp Grumman, respectively. For the raw IMU status of the iIMU-FSAS, see *Table 60*.	Long	4	H+12
5	Z Accel Output	Change in velocity count along z axis [1]	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along y axis) [1, 2]	Long	4	H+20
7	X Accel Output	Change in velocity count along x axis [1]	Long	4	H+24
8	Z Gyro Output	Change in angle count around z axis [3]. Right-handed.	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis) [2, 3]. Right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis [3]. Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 61* on *page 253*. Multiply the scale factor in *Table 61,* by the count in this field, for the velocity increments. See also *Table 1* on *page 24* for a list of IMU enclosures.
2. A negative value implies that the output is along the positive Y-axis marked on the IMU. A positive value implies that the change is in the direction opposite to that of the Y-axis marked on the IMU.
3. The change in angle (gyro) scale factor can be found in *Table 61* on *page 253*. Multiply the appropriate scale factor in *Table 61,* by the count in this field, for the angle increments in radians.

Table 60: iIMU-FSAS Status

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	Gyro warm-up	Passed = 0, Failed = 1
	5	0x00000020	Gyro self-test active	Passed = 0, Failed = 1
	6	0x00000040	Gyro status bit set	Passed = 0, Failed = 1
	7	0x00000080	Gyro time-out command interface	Passed = 0, Failed = 1
N2	8	0x00000100	Power-up built-in test (PBIT)	Passed = 0, Failed = 1
	9	0x00000200	Reserved	
	10	0x00000400	Interrupt	Passed = 0, Failed = 1
	11	0x00000800	Reserved	
N3	12	0x00001000	Warm-up	Passed = 0, Failed = 1
	13	0x00002000	Reserved	
	14	0x00004000		
	15	0x00008000	Initiated built-in test (IBIT)	Passed = 0, Failed = 1
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000	Accelerometer	Passed = 0, Failed = 1
	19	0x00080000	Accelerometer time-out	Passed = 0, Failed = 1
N5	20	0x00100000	Reserved	
	21	0x00200000	Gyro initiated BIT	Passed = 0, Failed = 1
	22	0x00400000	Gyro self-test	Passed = 0, Failed = 1
	23	0x00800000	Gyro time-out	Passed = 0, Failed = 1
N6	24	0x01000000	Analog-to-Digital (AD)	Passed = 0, Failed = 1
	25	0x02000000	Testmode	Passed = 0, Failed = 1
	26	0x04000000	Software	Passed = 0, Failed = 1
	27	0x08000000	RAM/ROM	Passed = 0, Failed = 1
N7	28	0x10000000	Reserved	
	29	0x20000000	Operational	Passed = 0, Failed = 1
	30	0x40000000	Interface	Passed = 0, Failed = 1
	31	0x80000000	Interface time-out	Passed = 0, Failed = 1

Recommended Input:

log rawimua onnew

ASCII Example:

```
#RAWIMUA,COM3,0,0.0,EXACT,1105,425384.180,00040000,b8ed,0;
1105,425384.156166800,111607,43088060,430312,-3033352,
-132863,186983,823*5aa97065
```

C.4.46 RAWIMUS Short Raw IMU Data

This is a short header version of the *RAWIMU* log on *page 249*.

Structure: **Message ID: 325**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	IMU Status	The status of the IMU. This field is given in a fixed length (n) array of bytes in binary but in ASCII or Abbreviated ASCII is converted into 2 character hexadecimal pairs. For the raw IMU status of the HG1700 and the LN-200 IMUs, refer to the Interface Control Documentation as provided by Honeywell and Northorp Grumman, respectively. For the raw IMU status of the iIMU-FSAS, see *Table 60*	Long	4	H+12
5	Z Accel Output	Change in velocity count along z axis [1]	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along y axis) [1, 2]	Long	4	H+20
7	X Accel Output	Change in velocity count along x axis [1]	Long	4	H+24
8	Z Gyro Output	Change in angle count around z axis [3] Right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around y axis) [2, 3] Right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around x axis [3] Right-handed	Long	4	H+36
11	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+40
12	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. The change in velocity (acceleration) scale factor for each IMU type can be found in *Table 61* on *page 253*. Multiply the scale factor in *Table 61,* by the count in this field, for the velocity increments in m/s. See also *Table 1* on *page 24* for a list of IMU enclosures.
2. A negative value implies that the output is along the positive Y-axis marked on the IMU. A positive value implies that the change is in the direction opposite to that of the Y-axis marked on the IMU.
3. The change in angle (gyro) scale factor can be found in *Table 61* on *page 253*. Multiply the appropriate scale factor in *Table 61,* by the count in this field, for the angle increments in radians.

Recommended Input:

log rawimusa onnew

ASCII Example:

```
%RAWIMUSA,1105,425384.180;
1105,425384.156166800,111607,43088060,430312,-3033352,
-132863,186983,823*5aa97065
```

Table 61: Raw IMU Scale Factors

Scale / IMU	HG1700-AG11 HG1700-AG58	HG1700-AG17 HG1700-AG62	LN-200	iIMU-FSAS
Gyroscope Scale Factor	2.0^{-33} rad/LSB	2.0^{-33} rad/LSB	2^{-19} rad/LSB	0.1x 2^{-8} arcsec/LSB
Acceleration Scale Factor	2.0^{-27} ft/s/LSB	2.0^{-26} ft/s/LSB	2^{-14} m/s/LSB	0.05 x 2^{-15} m/s/LSB

C.4.47 RXCONFIG Receiver Configuration

This log is used to output a list of all current command settings. When requested, an RXCONFIG log is output for each setting. See also the LOGLIST log on *page 231* for a list of currently active logs.

Message ID: 128

Log Type: Polled

Recommended Input:

log rxconfiga once

ASCII Example[1]:

```
#RXCONFIGA,COM1,21,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMA,COM1,21,96.5,
UNKNOWN,0,0.000,40000020,0000,143;COM1,9600,N,8,1,N,OFF,ON*e4f2d9b6*3e13c235
#RXCONFIGA,COM1,20,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMA,COM1,20,96.5,
UNKNOWN,0,0.000,40000020,0000,143;COM2,9600,N,8,1,N,OFF,ON*1f0609b3*1f61f4e9
#RXCONFIGA,COM1,19,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMA,COM1,19,96.5,
UNKNOWN,0,0.000,40000020,0000,143;COM3,9600,N,8,1,N,OFF,ON*0678ad5c*aa03e067
#RXCONFIGA,COM1,18,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMA,COM1,18,96.5,
UNKNOWN,0,0.000,40000020,0000,143;COM4,9600,N,8,1,N,OFF,ON*ef7579e2*eed07f66
#RXCONFIGA,COM1,17,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMCONTROLA,COM1,
17,96.5,UNKNOWN,0,0.000,40000020,0000,143;COM1,RTS,DEFAULT,RS232*2c5c183c*2559fe22
#RXCONFIGA,COM1,16,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMCONTROLA,COM1,16,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM2,RTS,DEFAULT,RS232*dfb9f449*cd8f0a10
#RXCONFIGA,COM1,15,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMCONTROLA,COM1,15,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM3,RTS,DEFAULT,RS232*f98ecb75*d8c3a160
#RXCONFIGA,COM1,14,96.5,UNKNOWN,0,0.000,40000020,0000,143;#COMCONTROLA,COM1,14,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM4,RTS,DEFAULT,RS232*e3032ae2*1945e7f7
#RXCONFIGA,COM1,13,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,13,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM1,NOVATEL,NOVATEL,ON*bc4fff14*e7d5cb24
#RXCONFIGA,COM1,12,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,12,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM2,NOVATEL,NOVATEL,ON*9cd39f4b*12706c90
#RXCONFIGA,COM1,11,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,11,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM3,NOVATEL,NOVATEL,ON*b39ad4f3*e875ddd9
#RXCONFIGA,COM1,10,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,10,
96.5,UNKNOWN,0,0.000,40000020,0000,143;COM4,NOVATEL,NOVATEL,ON*ddeb5ff5*3b85fbde
#RXCONFIGA,COM1,9,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,9,
96.5,UNKNOWN,0,0.000,40000020,0000,143;USB1,NOVATEL,NOVATEL,ON*68b6a123*db99b6e7
#RXCONFIGA,COM1,8,96.5,UNKNOWN,0,0.000,40000020,0000,143;#INTERFACEMODEA,COM1,8,
96.5,UNKNOWN,0,0.000,40000020,0000,143;ETH1,NOVATEL,NOVATEL,ON*421e3cb1*e457f77e
```

---

1. The embedded CRCs are flipped to make the embedded messages recognizable to the receiver. For example, consider the first embedded message above.

   91f89b07: 10010001111110001001101100000111

   11100000110110010001111110001001:e0d91f89

   Its CRC is really e0d91f89.

```
#RXCONFIGA,COM1,7,96.5,UNKNOWN,0,0.000,40000020,0000,143;#NMEATALKERA,COM1,7,96.5,
UNKNOWN,0,0.000,40000020,0000,143;GP*1283d3e3*14a45bcc
#RXCONFIGA,COM1,6,96.5,UNKNOWN,0,0.000,40000020,0000,143;#MAGVARA,COM1,6,96.5,
UNKNOWN,0,0.000,40000020,0000,143;CORRECTION,0.000000000,0.000000000
*de7a1f83*b83f15d9
#RXCONFIGA,COM1,5,96.5,UNKNOWN,0,0.000,40000020,0000,143;#LOGA,COM1,5,96.5,UNKNOWN
,0,0.000,40000020,0000,143;COM1,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD
*4ae673c3*292b473e
#RXCONFIGA,COM1,4,96.5,UNKNOWN,0,0.000,40000020,0000,143;#LOGA,COM1,4,96.5,UNKNOWN
,0,0.000,40000020,0000,143;COM2,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD
*111160de*b9c857a8
#RXCONFIGA,COM1,3,96.5,UNKNOWN,0,0.000,40000020,0000,143;#LOGA,COM1,3,96.5,UNKNOWN
,0,0.000,40000020,0000,143;COM3,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD
*55434e6b*d01c75af
#RXCONFIGA,COM1,2,96.5,UNKNOWN,0,0.000,40000020,0000,143;#LOGA,COM1,2,96.5,UNKNOWN
,0,0.000,40000020,0000,143;COM4,RXSTATUSEVENTA,ONNEW,0.000000,0.000000,HOLD
*ed7ff685*bd419430
#RXCONFIGA,COM1,1,96.5,FINESTEERING,1521,320402.983,40000020,0000,143;
#SETIMUTYPEA,COM1,1,96.5,FINESTEERING,1521,320402.983,40000020,0000,143;IMU_LN200
*58dfc9b8*80e7837c
#RXCONFIGA,COM1,0,96.5,FINESTEERING,1521,320402.984,40000020,0000,143;#COMA,COM1,
0,96.5,FINESTEERING,1521,320402.984,40000020,0000,143;IMU,115200,N,8,1,N,OFF,OFF
*4a567775*82ce86cf
```

WARNING!: **Do not use undocumented commands or logs! Doing so may produce errors and void your warranty.**

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXCONFIG header	Log header	-	H	0
2	e header	Embedded header	-	h	H
3	e msg	Embedded message	Varied	a	H + h
4	e xxxx	Embedded (inverted) 32-bit CRC (ASCII and Binary only). The embedded CRC is inverted so that the receiver does not recognize the embedded messages as messages to be output but continues with the RXCONFIG message. If you wish to use the messages output from the RXCONFIG log, simply flip the embedded CRC around for individual messages.	Long	4	H+ h + a
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+ h + a + 4
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.48 RXSTATUS Receiver Status

This log conveys various status parameters of the SPAN-SE receiver system. These include the SPAN-SE Receiver Status and Error words which contain several flags specifying status and error conditions. If an error occurs (shown in the Receiver Error word) the receiver idles all channels, turns off the antenna, and disables the RF hardware as these conditions are considered to be fatal errors. The log contains a variable number of status words to allow for maximum flexibility and future expansion.

The receiver gives the user the ability to determine the importance of the status bits. In the case of the Receiver Status, setting a bit in the priority mask causes the condition to trigger an error. This causes the receiver to idle all channels, turn off the antenna, and disable the RF hardware, the same as if a bit in the Receiver Error word is set.

Receiver errors automatically generate event messages. These event messages are output in RXSTATUSEVENT logs. It is also possible to have status conditions trigger event messages to be generated by the receiver. This is done by setting/clearing the appropriate bits in the event set/clear masks. The set mask tells the receiver to generate an event message when the bit becomes set. Likewise, the clear mask causes messages to be generated when a bit is cleared.

If you wish to disable all these messages without changing the bits, simply UNLOG the RXSTATUSEVENT logs on the appropriate ports. See also the UNLOG command on *page 152*.

---

1. Field #4, the receiver status word as represented in *Table 63*, is also in Field #8 of the header. See the *ASCII Example* below and *Table 63* on *page 259* for clarification.
2. Many OEMV status bits have been redefined to match SPAN receiver hardware. Some bits (such as model, temperature, position solution) are mapped directly from the OEMV-3. In *Table 62, SPAN Receiver Error* on *page 257* and *Table 63, SPAN Receiver Status* on *page 259*, OEMV-3 values are indicated in blue, SPAN values are indicated in black and OEMV-3 values, that cause the OEMV-3 LED to turn red, are indicated in red.

   When logging RXSTATUS, the SPAN-SE receiver data is displayed first (error bits then status bits) then the OEMV-3 status bits then the OEMV-2 status bits.
3. Refer also to the chapter on *Built-In Status Tests* in the *OEMV Family Installation and Operation User Manual*.

Message ID: **93**

Log Type: **Asynch**

Recommended Input:

log rxstatusa onchanged

ASCII Examples:

An RXSTATUS log with a simple error:

```
#RXSTATUSA,COM1,0,98.5,FINESTEERING,1521,319258.697,40000020,0000,143;
00000000,4,40000020,00000000,00000000,00000000,00000000,00000000,00000000,000
00000,00000020,00000000,00000000,00000000,00000000,00000000,00000000,00000000
*cf7aa03a
```

The status bit 00000020 indicates antenna open.

An RXSTATUS log with a component hardware error:

```
#RXSTATUSA,COM1,0,99.5,FINESTEERING,1521,319470.627,40000021,0000,143;
80000000,4,40000021,00000000,00000000,00000000,00000000,00000000,00000000,000
00000,00000020,00000000,00000000,00000000,00000000,00000000,00000000,00000000
*dd24b521
```

The error bit 80000000 indicates a component hardware error. This means the OEMV-3 is not communicating. This is a non-recoverable error for SPAN-SE. It indicates that the OEMV-3 has experienced a USB overrun, or that the OEMV-3 is no longer powered. Since SPAN-SE controls the power to its internal OEMV-3, it is unlikely the OEMV-3 has lost power. It is more likely that the user has overloaded the OEMV-3 USB with excessive log requests. In this case, the RXSTATUSEVENT log would show:

```
#RXSTATUSEVENTA,COM1,0,0.0,FINESTEERING,1521,319470.627,404c0028,0000,143;
ERROR,31,SET,"Component Hardware Failure"*79a2006b
```

Table 62: SPAN Receiver Error

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	SDRAM Status	OK	Error
	1	0x00000002	Firmware Status	OK	Error
	2	0x00000004	ROM Status	OK	Error
	3	0x00000008	FPGA Status	OK	Error
N1	4	0x00000010	Electronic Serial Number (ESN) access status	OK	Error
	5	0x00000020	Authorization Code Status	OK	Error
	6	0x00000040	Slow ADC Status	OK	Error
	7	0x00000080	Supply Voltage Status	OK	Error
N2	8	0x00000100	Thermometer Status	OK	Error
	9	0x00000200	Temperature Status	OK	Error
	10	0x00000400	MINOS5 Status	OK	Error
	11	0x00000800	PLL RF1 Hardware Status - L1	OK	Error
N3	12	0x00001000	PLL RF2 Hardware Status - L2	OK	Error
	13	0x00002000	RF1 Hardware Status - L1	OK	Error
	14	0x00004000	RF2 Hardware Status - L2	OK	Error
	15	0x00008000	NVM status	OK	Error

Continued on page 258

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N4	16	0x00010000	Software resource limit	OK	Error
	17	0x00020000	Model Status	OK	Error
	18	0x00040000	COM Port Power Status	Not Over Current	Over Current
	19	0x00080000	Reserved		
N5	20	0x00100000	Remote Loading Has Begun	No	Yes
	21	0x00200000	Export Restriction	OK	Error
	22	0x00400000	Reserved		
	23	0x00800000			
N6	24	0x01000000			
	25	0x02000000			
	26	0x04000000			
	27	0x08000000			
N7	28	0x10000000			
	29	0x20000000			
	30	0x40000000			
	31	0x80000000	Component hardware failure	OK	Error

Table 63: SPAN Receiver Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Error Flag, see *Table 62, SPAN Receiver Error* on *page 257*	No error	Error
	1	0x00000002	Temperature Status	OK	Warning
	2	0x00000004	Power Supply	OK	Warning
	3	0x00000008	Antenna Power	Powered	Not Powered
N1	4	0x00000010	Reserved		
	5	0x00000020	Antenna Open	OK	Open
	6	0x00000040	Antenna Shorted	OK	Shorted
	7	0x00000080	SPAN CPU Overload	No Overload	Overload
N2	8	0x00000100	COM1 Buffer Overrun	No overrun	Overrun
	9	0x00000200	COM2 Buffer Overrun	No overrun	Overrun
	10	0x00000400	COM3 Buffer Overrun	No overrun	Overrun
	11	0x00000800	COM4 Buffer Overrun	No overrun	Overrun
N3	12	0x00001000	USB Buffer Overrun	No overrun	Overrun
	13	0x00002000	Ethernet Buffer Overrun	No overrun	Overrun
	14	0x00004000	IMU Buffer Overrun	No overrun	Overrun
	15	0x00008000	RF1 AGC Status	OK	Bad
N4	16	0x00010000	Reserved		
	17	0x00020000	RF2 AGC Status	OK	Bad
	18	0x00040000	Almanac /UTC Known	Valid	Invalid
	19	0x00080000	Position Solution	Valid	Invalid
N5	20	0x00100000	Position Fixed	Not Fixed	Fixed
	21	0x00200000	Clock Steering	Enabled	Disabled
	22	0x00400000	Clock Model	Valid	Invalid
	23	0x00800000	Reserved		
N6	24	0x01000000	Software Resource	OK	Warning
	25	0x02000000	OEMV-2 CPU Overload	No Overload	Warning
	26	0x04000000	OEMV-3 CPU Overload	No Overload	Warning
	27	0x08000000	SD Logging Warning	Buffer Fine	Buffer > 80% full

Continued on page 260

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N7	28	0x10000000	Reserved		
	29	0x20000000	OEMV-2 Status Event	No event	Event
	30	0x40000000	OEMV-3 Status Event	No event	Event
	31	0x80000000	Reserved		

Table 64: Auxiliary 1 Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Reserved		
	1	0x00000002			
	2	0x00000004			
	3	0x00000008	Position averaging	Off	On
N1	4	0x00000010	Reserved		
	5	0x00000020			
	6	0x00000040			
	7	0x00000080	OEMV-3 USB connection status	Connected	Not connected
N2	8	0x00000100	OEMV-3 USB1 buffer overrun flag	No overrun	Overrun
	9	0x00000200	OEMV-3 USB2 buffer overrun flag	No overrun	Overrun
	10	0x00000400	OEMV-3 USB3 buffer overrun flag	No overrun	Overrun
	11	0x00000800	Reserved		

Table 65: OEMV-3 Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0000001	Reserved		

Table 66: OEMV-2 Status

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0000001	Reserved		

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXSTATUS header	Log header		H	0
2	error	Receiver error (see *Table 62 on page 257*). A value of zero indicates no errors.	ULong	4	H
3	# stats	Number of status codes (including Receiver Status)	ULong	4	H+4
4	rxstat	Receiver status word (see *Table 63 on page 259*)	ULong	4	H+8
5	rxstat pri	Receiver status priority mask	ULong	4	H+12
6	rxstat set	Receiver status event set mask	ULong	4	H+16
7	rxstat clear	Receiver status event clear mask	ULong	4	H+20
8	aux1stat	Auxiliary 1 status word (see *Table 64 on page 261*)	ULong	4	H+24
9	aux1stat pri	Auxiliary 1 status priority mask	ULong	4	H+28
10	aux1stat set	Auxiliary 1 status event set mask	ULong	4	H+32
11	aux1stat clear	Auxiliary 1 status event clear mask	ULong	4	H+36
12	V3stat	OEMV-3 status word (see *Table 65 on page 261*)	ULong	4	H+40
13	V3stat pri	OEMV-3 status priority mask	ULong	4	H+44
14	V3stat set	OEMV-3 status event set mask	ULong	4	H+48
15	V3stat clear	OEMV-3 status event clear mask	ULong	4	H+52
16	V2stat	OEMV-2 status word (see *Table 66 on page 261*)	ULong	4	H+56
17	V2stat pri	OEMV-2 status priority mask	ULong	4	H+60
18	V2stat set	OEMV-2 status event set mask	ULong	4	H+64
19	V2stat clear	OEMV-2 status event clear mask	ULong	4	H+68
20...	Next status code offset = H + 8 + (# stats x 16)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+(#stats x 64)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.49 RXSTATUSEVENT Status Event Indicator

This log is used to output event messages as indicated in the RXSTATUS log. An event message is automatically generated for all receiver errors, which are indicated in the receiver error word. In addition, event messages can be generated when other conditions, which are indicated in the receiver status and auxiliary status words, are met.

On start-up, the receiver is set to log the RXSTATUSEVENTA log ONNEW on all ports. You can remove this message by using the UNLOG command, see *page 152*.

When a fatal event occurs (for example, in the event of a receiver hardware failure), a bit is set in the receiver error word, part of the RXSTATUS log on *page 256*, to indicate the cause of the problem. Bit 0 is set in the receiver status word to show that an error occurred, the error strobe is driven high, and the LED flashes red and yellow showing an error code. An RXSTATUSEVENT log is generated on all ports to show the cause of the error. Receiver tracking is disabled at this point but command and log processing continues to allow you to diagnose the error. Even if the source of the error is corrected at this point, the receiver must be reset to resume normal operation.

☒ See also the chapter on *Built-In Status Tests* in the *OEMV Family Installation and Operation User Manual*.

Message ID: **94**

Log Type: **Asynch**

Recommended Input:

log rxstatuseventa onchanged

ASCII Example 1:

```
#RXSTATUSEVENTA,COM1,0,17.0,FREEWHEELING,1337,408334.510,00480000,b967,1984;
STATUS,19,SET,"No Valid Position Calculated"*6de945ad
```

ASCII Example 2:

```
#RXSTATUSEVENTA,COM1,0,41.0,FINESTEERING,1337,408832.031,01000400,b967,1984;
STATUS,10,SET,"COM3 Transmit Buffer Overrun"*5b5682a9
```

Table 67: Status Word

Word (binary)	Word (ASCII)	Description
0	ERROR	Receiver Error word, see *Table 62* on *page 257*
1	STATUS	Receiver Status word, see *Table 63* on *page 259*
2	AUX1	Auxiliary 1 Status word, see *Table 64* on *page 261*
3	AUX2	Auxiliary 2 Status word see *Table 65* on *page 261*
4	AUX3	Auxiliary 3 Status word see *Table 66* on *page 261*

Table 68: Event Type

Event (binary)	Event (ASCII)	Description
0	CLEAR	Bit was cleared
1	SET	Bit was set

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXSTATUSEVENT header	Log header		H	0
2	word	The status word that generated the event message (see *Table 67* above)	Enum	4	H
3	bit position	Location of the bit in the status word (see *Table 63* starting on *page 259* for the receiver status table or the auxiliary status tables on *page 261*)	Ulong	4	H+4
4	event	Event type (see *Table 68* above)	Enum	4	H+8
3	description	This is a text description of the event or error	Char[32]	32	H+12
5	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.50 SPANVALIDMODELS Valid Model Information

This log gives a list of valid authorized models available and expiry date information.

Use the SPANVALIDMODELS log to output a list of available models for the receiver. You can use the SPANAUTH command, see *page 149* to add a model. See the VERSION log on *page 270* for the currently active model.

If a model has no expiry date it reports the year, month and day fields as 0, 0 and 0 respectively.

Message ID: **1089**

Log Type: **Polled**

Recommended Input:

log validmodelsa once

ASCII Example:

```
#SPANVALIDMODELSA, COM1,0,99.0,UNKNOWN,0,74.876,404c0020,0000,155;
1,"SJ",0,0,0*558ae6ab
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	SPANVALID-MODELS header	Log header		H	0
2	#mod	Number of models with information to follow	Ulong	4	H
3	model	Model name	String [max. 16]	Variable[1]	Variable
4	expyear	Expiry year	Ulong	4	Variable Max:H+20
5	expmonth	Expiry month	Ulong	4	Variable Max: H+24
6	expday	Expiry day	Ulong	4	Variable: Max: H+28
7...	Next model offset = H + 4 + (#mods x variable [max:28])				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	Variable
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. In the binary log case, additional bytes of padding are added to maintain 4-byte alignment

C.4.51 TIME Time Data

This log provides several time related pieces of information including receiver clock offset and UTC time and offset. It can also be used to determine any offset in the PPS signal relative to GPS time.

To find any offset in the PPS signal, log the TIME log 'ontime' at the same rate as the PPS output. For example, if the PPS output is configured to output at a rate of 0.5 seconds, log the TIME log 'ontime 0.5' as follows:

```
log time ontime 0.5
```

The TIME log offset field can then be used to determine any offset in PPS output relative to GPS time.

Message ID: **101**

Log Type: **Synch**

Recommended Input:

log timea ontime 1

ASCII Example:

```
#TIMEA,COM1,0,50.5,FINESTEERING,1337,410010.000,00000000,9924,1984;
VALID,1.953377165e-09,7.481712815e-08,-12.99999999492,2005,8,25,17,
53,17000,VALID*e2fc088c
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	TIME header	Log header		H	0
2	clock status	Clock model status (not including current measurement data), see *Table 54* on *page 237*	Enum	4	H
3	offset	Receiver clock offset, in seconds from GPS time. A positive offset implies that the receiver clock is ahead of GPS time. To derive GPS time, use the following formula: GPS time = receiver time - offset	Double	8	H+4
4	offset std	Receiver clock offset standard deviation.	Double	8	H+12
5	utc offset	The offset of GPS time from UTC time, computed using almanac parameters. UTC time is GPS time plus the current UTC offset plus the receiver clock offset: UTC time = GPS time + offset + UTC offset	Double	8	H+20
6	utc year	UTC year	Ulong	4	H+28
7	utc month	UTC month (0-12) [1]	Uchar	1	H+32
8	utc day	UTC day (0-31) [1]	Uchar	1	H+33
9	utc hour	UTC hour (0-23)	Uchar	1	H+34
10	utc min	UTC minute (0-59)	Uchar	1	H+35
11	utc ms	UTC millisecond (0-60999) [2]	Ulong	4	H+36
12	utc status	UTC status 0 = Invalid 1 = Valid	Enum	4	H+40
13	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
14	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. If UTC time is unknown, the values for month and day are 0.
2. Maximum of 60999 when leap second is applied.

C.4.52 TIMEDWHEELDATA Timed Wheel Data

This log contains time stamped wheel sensor data. The time stamp in the header is the time of validity for the wheel data, not the time the TIMEDWHEELDATA log was output. This is a short header log, see also *Section C.2, Description of ASCII and Binary Logs with Short Headers* on *page 168*.

This log contains the wheel sensor information received from any of the three sources SPAN-SE supports. See also *Section 3.5, SPAN Wheel Sensor Configuration* on *page 50*.

> If you are using an iMAR iMWS (Magnetic Wheel Speed Sensor and Convertor) connected directly to the iIMU FSAS, Field #4, the float wheel velocity, is filled instead of Field #3, the unsigned short wheel velocity.
>
> When you send a WHEELVELOCITY command, see *Page 69*, from an external wheel sensor, the TIMEDWHEELDATA log contains the same wheel velocity values, float or ushort, as those you entered.
>
> Note that neither velocity value is used by the SPAN filter. Rather, the SPAN filter uses cumulative ticks per second. If post-processing, the velocities may be used with the NovAtel Waypoint Group's Inertial Explorer software.

Structure: **Message ID: 622**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Ticks Per Rev	Number of ticks per revolution	Ushort	2	H
3	Wheel Vel	Wheel velocity in counts/s	Ushort	2	H+2
4	fWheel Vel	Float wheel velocity in counts/s	Ulong	4	H+4
5	Reserved		Ulong	4	H+8
6			Ulong	4	H+12
7	Ticks Per Second	Cumulative number of ticks	Ulong	4	H+16
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+20
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log timedwheeldataa onnew

ASCII Example:

This example is from the iMAR iMWS wheel sensor:

```
%TIMEDWHEELDATAA,1393,411345.001;58,0,215.814910889,0,0,1942255*3b5fa236
```

C.4.53 VEHICLEBODYROTATION Vehicle to SPAN Frame Rotation

The VEHICLEBODYROTATION log reports the angular offset from the vehicle frame to the SPAN frame. The SPAN computation frame is defined by the transformed IMU enclosure axis with Z pointing up, see the SETIMUORIENTATION command on *page 134*. If your IMU is mounted with the Z axis (as marked on the IMU enclosure) pointing up, the IMU enclosure frame is the same as the SPAN computation frame. This log reports whatever was entered using the VEHICLEBODYROTATION command, *page 155*, or whatever was solved for after invoking the RVBCALIBRATE command, see *page 128*.

Recommended Input:

log vehiclebodyrotationa onchanged

ASCII Example:

```
#VEHICLEBODYROTATIONA,COM1,0,36.5,FINESTEERING,1264,144170.094,00000000,bcf2,
1541;1.5869999997474209,2.6639999995760122,77.6649999876392343,2.000000000000
0000,2.0000000000000000,5.0000000000000000*25f886cc
```

C.4.54 VERSION Version Information

This log contains the version information for all components of a system.

A component may be hardware (for example, a receiver or data collector) or firmware in the form of applications or data (for example, data blocks for height models or user applications). See *Table 72, VERSION Log: Field Formats* on *page 271* for details on the format of key fields.

Message ID: **37**

Log Type: **Polled**

Recommended Input:

log versiona once

ASCII Example:

```
#VERSIONA,COM1,0,97.5,FINESTEERING,1521,318658.225,00000000,0000,149;
5,
SPANCARD,"SJ","DDV08490044","SPANPPC-3.00-A","SPPC1.000","1.100","Mar  3
2009","16:35:00",
SPANFPGA,"","","","00028","","","",
GPSCARD,"L12GRV","DAB08190083","OEMV3G-4.00-X2T","3.621","3.000","2009/Feb/
18","12:31:14",
GPSCARD," L12GRV ","BZZ08190377","OEMV2G-3.01-2T","3.200S3","3.000","2006/
Jul/14","12:28:52",
IMUCARD,"HG1700 100Hz","DAB08190083","OEMV3G-4.00-X2T","2.010","3.000","Feb
09 2007","10:39:41"*6f10750f
```

Table 69: OEMV in SPAN-SE Model Designators

Designator	Description
G	12 L1 or 12 L1/L2 GLONASS channels, frequencies to match GPS configuration
R	Receive RT2 and/or RT20 corrections
L	1 L-band channel with CDGPS and OmniSTAR HP/XP capability

Table 70: SPAN-SE Model Designators

Designator	Description
I	SPAN supporting IMUs with data rates ≤ 100 Hz
J	SPAN supporting IMUs with data rates > 100 Hz
S	A single GNSS card system where only the OEMV-3 is included

Table 71: Component Types

Binary	ASCII	Description
0	UNKNOWN	Unknown Component
1	GPSCARD	OEMV Family Component
7	IMUCARD	IMU Card
8192	SPANCARD	SPAN-SE Card
8193	SPANFPGA	SPAN-SE Field Programmable Gate Array (FPGA)

Table 72: VERSION Log: Field Formats

Field Type	Field Format (ASCII)	Description
hw version	P-RS-CCC	P = hardware platform (for example, OEMV) R = hardware revision (for example, 3.00) S = processor revision (for example, A) [1] CCC = COM port configuration (for example, 22T) [2]
sw version, boot version	VV.RRR[Xxxx]	VV = major revision number RRR = minor revision number X = Special (S), Beta (B), Internal Development (D, A) xxx = number
comp date	YYYY/MM/DD	YYYY = year MM = month DD = day (1 - 31)
comp time	HH:MM:SS	HH = hour MM = minutes SS = seconds

1. This field may be empty if the revision is not stamped onto the processor.
2. One character for each of the COM ports 1, 2, and 3. Characters are: 2 for RS-232, 4 for RS-422, T for LV-TTL, and X for user-selectable (valid for COM1 of the OEMV-2 only). Therefore, the example is for a receiver that uses RS-232 for COM 1 and COM 2 and LV-TTL for COM 3.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	VERSION header	Log header		H	0
2	# comp	Number of SPAN-SE components (cards, and so on)	Long	4	H
3	type	Component type (see *Table 71 on page 271*)	Enum	4	H+4
4	model	For the OEMV-3 inside the SPAN-SE, the base model name plus the model designators, see *Table 69* on *page 271* For the SPAN-SE, the SPAN-SE model designators only, see *Table 70* on *page 271*	Char[16]	16	H+8
5	psn	Product serial number	Char[16]	16	H+24
6	hw version	Hardware version, see *Table 72, VERSION Log: Field Formats* on *page 271*	Char[16]	16	H+40
7	sw version	Firmware software version, see *Table 72*	Char[16]	16	H+56
8	boot version	Boot code version, see *Table 72*	Char[16]	16	H+72
9	comp date	Firmware compile date, see *Table 72*	Char[12]	12	H+88
10	comp time	Firmware compile time, see *Table 72*	Char[12]	12	H+100
11...	Next component offset = H + 4 + (#comp x 108)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+ (#comp x 108)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

C.4.55 WHEELSIZE Wheel Size

The SPAN filter models the size of the wheel to compensate for changes in wheel circumference due to hardware or environmental changes. The default wheel size is 1.96 m. A scale factor to this default size is modeled in the filter and this log contains the current estimate of the wheel size.

Structure: **Message ID: 646**

Log Type: Asynch

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Scale	Wheel sensor scale factor	Double	8	H
3	Circum	Wheel circumference (m)	Double	8	H+8
4	Var	Variance of circumference (m^2)	Double	8	H+16
5	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+24
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

Recommended Input:

log wheelsizea onnew

ASCII Example:

```
#WHEELSIZEA,COM3,0,44.0,EXACT,0,0.000,00000000,85f8,33738;
1.025108123,2.009211922,0.000453791*157fd50b
```

Appendix D Command Prompt Interface

When the SPAN system turns on, no activity information is transmitted from the serial ports except for the port prompt. A terminal connected to the receiver display a messages on its monitor. For example:

[COM1] *if connected to COM1 port*

The COM port can be COM1, COM2, COM3, COM4, USB1, or ETH1. Commands are typed at the interfacing terminal's keyboard, and sent after pressing the terminal's <↵> or <Enter> key.

⌧ Most valid commands do produce a visible response on the screen. The indication that they have been accepted is a return of the port prompt from the receiver.

Example:

An example of no echo response to an input command is the SETIMUTOANTOFFSET command. It can be entered as follows:

```
[COM2]setimutoantoffset 0.1 0.1 0.1[Return]
<OK
[COM2]
```

The above example illustrates command input to the receiver COM2 serial port, which sets the antenna to IMU offset. However, your only confirmation that the command was actually accepted is the return of the [COM2] prompt.

If a command is incorrectly entered, the receiver responds with "Invalid Command Name" (or a more detailed error message) followed by the port prompt.

D.1 DOS

One way to initiate multiple commands and logging from the receiver is to create DOS command files relating to specific functions. This minimizes the time required to set up duplicate test situations. Any convenient text editor can be used to create command text files.

Example:

For this example, consider a situation where a laptop computer's appropriately configured COM1 serial port is connected to the receiver's COM1 serial port, and where a rover terminal is connected to the receiver's COM2 serial port. If you wish to monitor the SPAN system activity, the following command file could be used to do this.

1. Open a text editor on the PC and type in the following command sequences:

```
log com2 satvisa ontime 15
log com2 trackstata ontime 15
log com2 rxstatusa ontime 60 5
log com2 bestposa ontime 1
log com2 psrdopa ontime 15
```

2. Save this with a convenient file name (e.g. C:\GPS\BOOT1.TXT) and exit the text editor.
3. Use the DOS *copy* command to direct the contents of the BOOT1.TXT file to the PC's COM1 serial port:

```
C:\GPS>copy boot1.txt com1
1 files(s) copied
C:\GPS>
```

4. The SPAN system is now initialized with the contents of the BOOT1.TXT command file, and logging is directed from the receiver's COM2 serial port to the rover terminal.

D.2 WINDOWS

As any text editor or communications program can be used for these purposes, the use of Windows 98 is described only as an illustration. The following example shows how Windows 98 accessory programs *Notepad* and *HyperTerminal* can be used to create a hypothetical waypoint navigation file on a laptop computer, and send it to the receiver. It is assumed that the laptop computer's COM1 serial port is connected to the receiver's COM1 serial port, and that a rover terminal is connected to the receiver's COM2 serial port.

Example:

1. Open *Notepad* and type in the following command text:

```
setimutype imu_hg1700_ag58
setimutoantoffset 1.25 0.35 1.65 0.02 0.02 0.02
log com1 rawimusb onnew
log com1 rangecmpb ontime 1
log com1 inspvasb ontime 0.1
log com1 inscovsb onnew
```

2. Save this with a convenient file name (e.g. C:\GPS\BOOTNAV1.TXT) and exit *Notepad*.
3. Ensure that the *HyperTerminal* settings are correctly set up to agree with the receiver communications protocol; these settings can be saved (e.g. C:\GPS\OEMSETUP.HT) for use in future sessions. You may wish to use XON / XOFF handshaking to prevent loss of data.
4. Select Transfer | Send Text File to locate the file that is to be sent to the receiver. Once you double-click on the file or select Open, *HyperTerminal* sends the file to the receiver.

The above example sets the IMU type to be the HG1700 AG58. It also sets the leverarm, from the IMU centre to the GNSS antenna phase centre, with the SETIMUTOANTOFFSET command. Log requests on COM1 of SPAN-SE are also made. In this case, RAWIMUSB logs are logged asynchronously at 100 Hz, RANGECMPB logs synchronously at 1 Hz, INSPVASB at 10 Hz, and the INSCOVSB log would be logged when updated which is at 1 Hz also.

Appendix E HG1700 IMU Installation

The following procedure, detailed in this appendix, provides the necessary information to install the HG1700 sensor into the SPAN HG Enclosure (NovAtel part number 01017898), see also *Figure 34* below. The steps required for this procedure are:

- Disassemble the SPAN HG Enclosure
- Install the HG1700 Sensor Unit
- Make Electrical Connections
- Reassemble the SPAN HG Enclosure

Important!: Ensure you use a ground strap before installing the internal circuit boards. Do NOT scratch any surfaces of the unit.

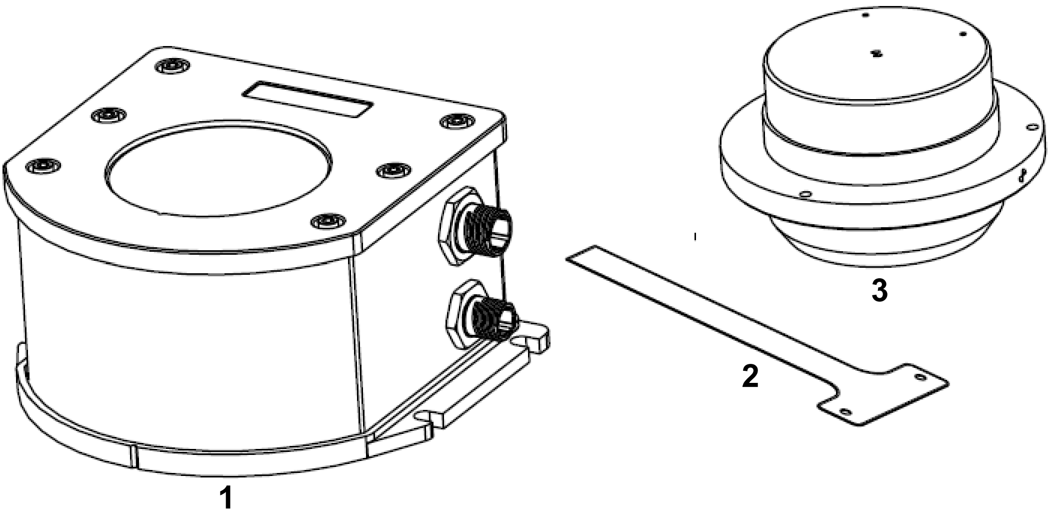


Figure 34: Required Parts

Reference	Description
1	SPAN IMU Enclosure
2	HG1700 Flex Cable
3	HG1700 Sensor Unit

E.1 Disassemble the SPAN IMU Enclosure

The SPAN IMU disassembly steps are as follows:

1. Remove the top cover's six bolts using an allan key, see *Figure 35*:

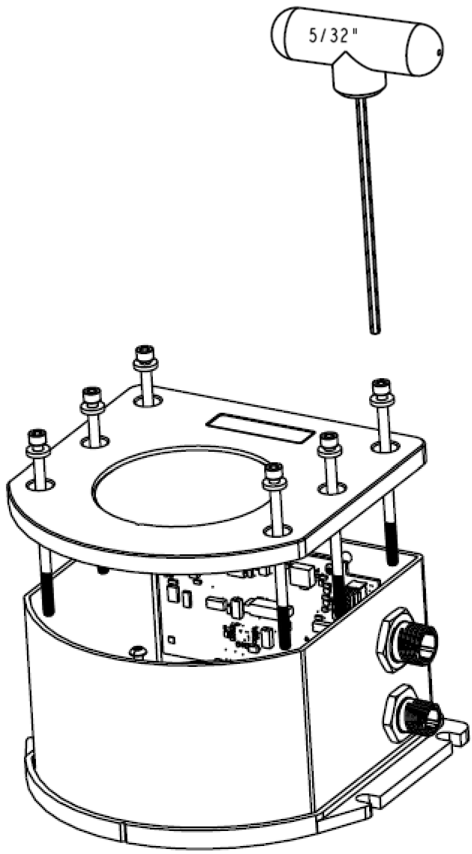


Figure 35: Bolts and Allan Key

2. Set aside the bolts with their sealing washers.

3. Lift the top cover off the tube body and set it aside, see *Figure 36* on *page 279*.

4. Lift the tube body away from its base plate and set it aside, see *Figure 36*.

5. Remove the 3 ring spacer screws and set aside, see *Figure 36*.

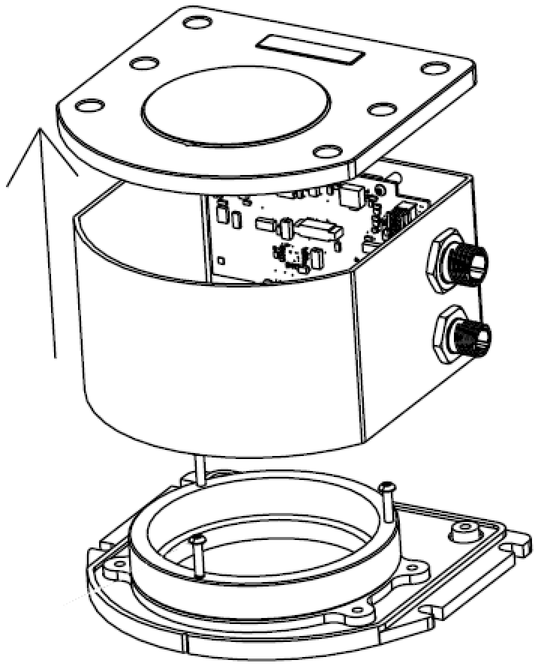


Figure 36: Lift Top Cover, Tube Body and 3 Ring Spacer Screws

E.2 Install the HG1700 Sensor Unit

To re-assemble the SPAN IMU with the HG1700 sensor, see *Figure 37* and follow these steps:

1. Mount the HG1700 sensor with the attached #8 screws. Apply threadlock to the screw threads. Use an allan key to torque each screw to 10 in-lbs.

2. Fit the tube body over the HG1700 sensor and onto the base plate.

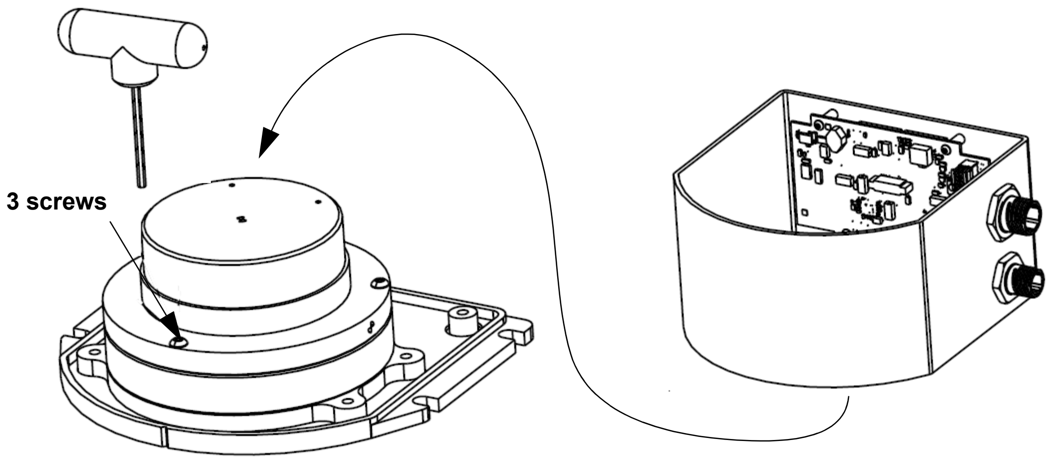


Figure 37: SPAN IMU Re-Assembly

E.3 Make the Electrical Connections

To make the electrical connections you will need a 3/32” allan key, the flex cable and the partially assembled SPAN IMU from *Section E.2, Install the HG1700 Sensor Unit* on *page 279*. Now follow these steps:

1. Attach the flex cable to the HG1700 sensor ensuring that all the pins are fully connected. Check also that the pins are fully seated and that the flex cable stiffener around the pins is not bent upward, see *Figure 38*.

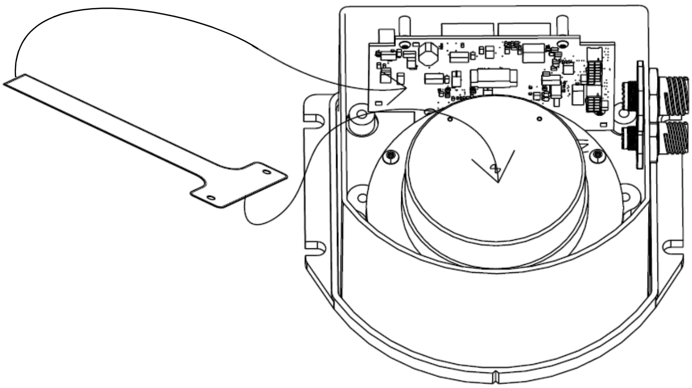


Figure 38: Attach Flex Cable

2. Tighten the screws to 4-in pounds.
3. Connect the opposite end of the flex cable to the corresponding connector on the IMU card ensuring that the contacts on the flex cable mate with the contacts on the connector, *Figure 38*.
4. Check that the flex cable is locked in place.

Important!: *Figure 39* shows an incorrect installation of the flex cable where it is bowed in the middle. It will not operate properly in this position. *Figure 40* shows the proper installation of the flex cable. Notice how the flex cable sits flush against the IMU surface.

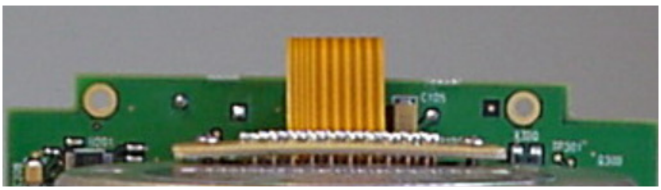


Figure 39: Incorrect (Bowed) Flex Cable Installation

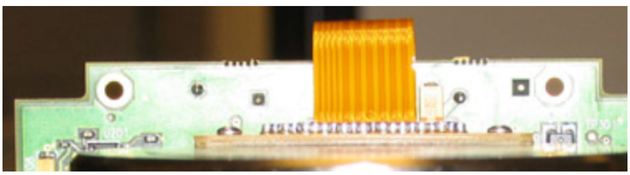


Figure 40: Correct (Flat) Flex Cable Installation

E.4 Re-Assemble the SPAN IMU Enclosure

Use an allan key to align the long bolts with the threaded holes in the base, see *Figure 35* on *page 278*. Apply threadlock to threads. Finger tighten all bolts and torque them in a cross pattern to 12 in-lops. The fully assembled IMU enclosure is shown in *Figure 41* below.

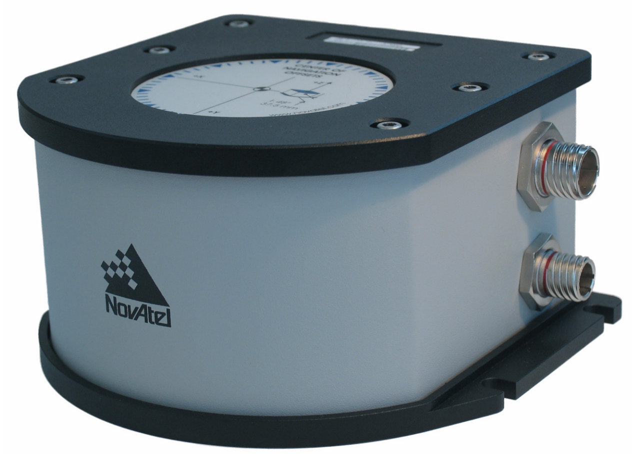


Figure 41: HG1700 SPAN IMU

Appendix F LN-200 IMU Installation

The following procedure, detailed in this appendix, provides the necessary information to install the LN-200 sensor (NovAtel part number 80023515) into the SPAN IMU enclosure (NovAtel part number 01017656) using the LN-200 wiring harness (NovAtel part number 01017655), see also *Figure 42* below. The steps required for this procedure are:

- Disassemble the SPAN IMU Enclosure
- Install the LN-200 Sensor Unit
- Make Electrical Connections
- Reassemble the SPAN IMU Enclosure

Important!: Ensure you use a ground strap before installing the internal circuit boards. Do NOT scratch any surfaces of the unit.

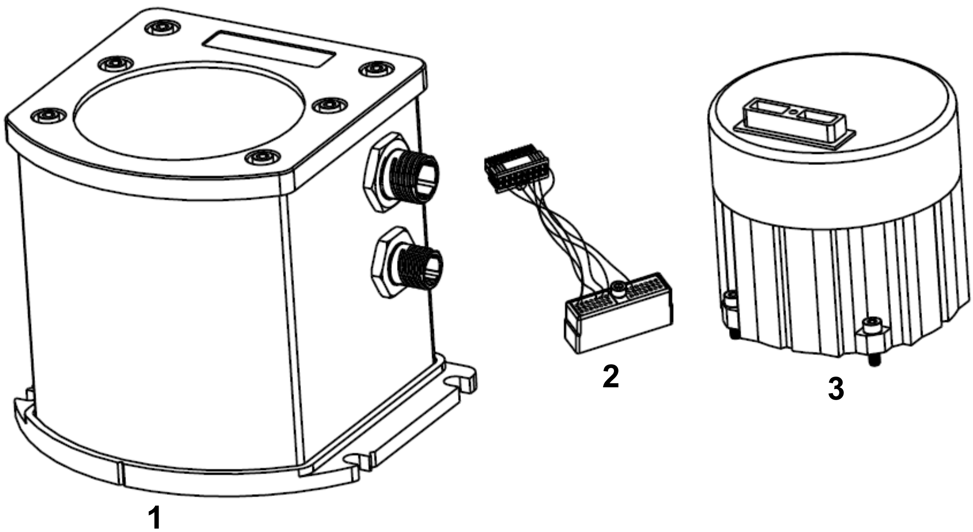


Figure 42: Required Parts

Reference	Description
1	SPAN IMU Enclosure
2	LN-200 Wiring Harness
3	LN-200 Sensor Unit

F.1 Disassemble the SPAN IMU Enclosure

The SPAN IMU disassembly steps are as follows:

1. Remove the top cover's six bolts using an allan key, see *Figure 43*:

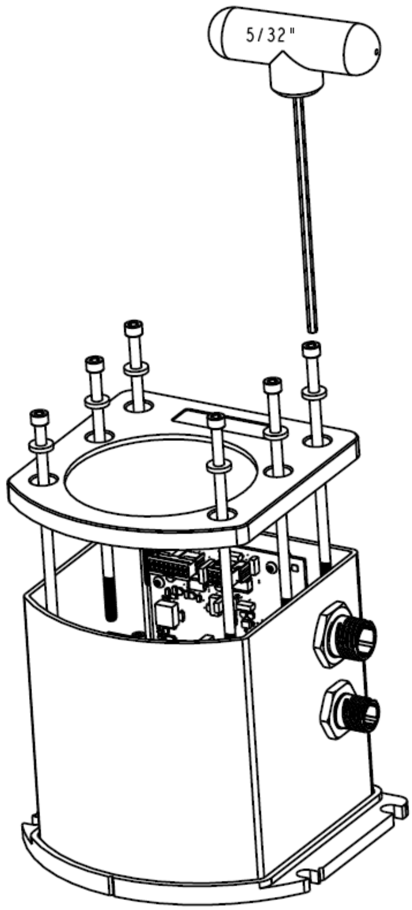


Figure 43: Bolts and Allan Key

2. Set aside the bolts with their sealing washers.

3. Lift the top cover off the tube body and set it aside.

4. Lift the tube body away from its base plate and set it aside, see *Figure 44* on *page 284*.

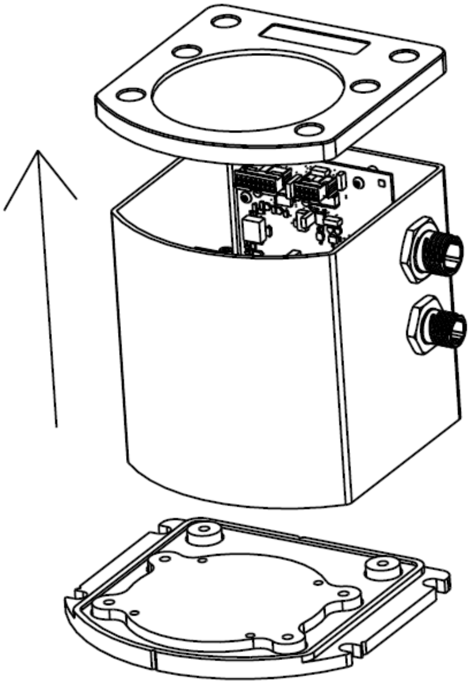


Figure 44: Lift Top Cover and Tube Body

F.2 Install the LN-200 Sensor Unit

To re-assemble the SPAN IMU with the LN-200 sensor, see *Figure 45* and follow these steps:

1. Mount the LN-200 sensor with the attached M4 screws. Apply threadlock to the screw threads. Use an allan key to torque each screw to 10 in-lbs.

2. Fit the tube body over the LN-200 sensor and onto the base plate.

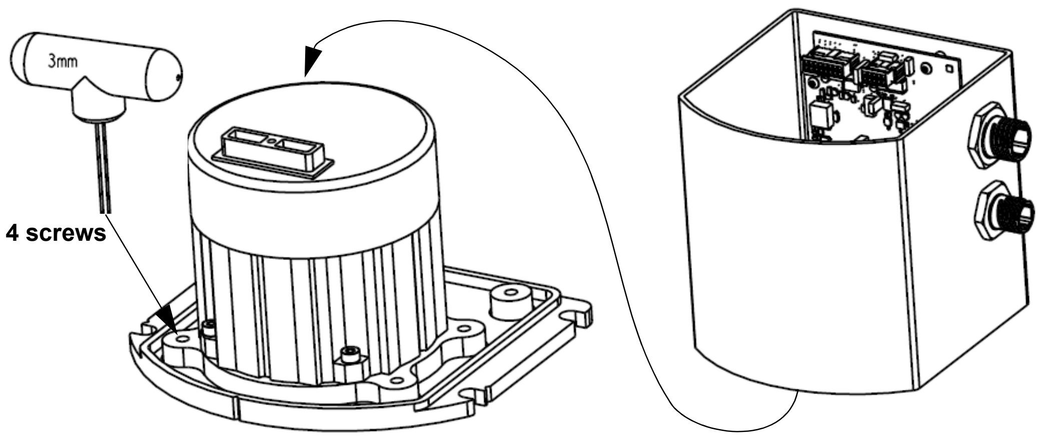


Figure 45: SPAN IMU Re-Assembly

F.3 Make the Electrical Connections

To make the electrical connections you will need a 3/32” allan key, the wiring harness and the partially assembled SPAN IMU from *Section F.2, Install the LN-200 Sensor Unit* on *page 284*. Now follow these steps:

1. Attach the LN-200 wire harness to the mating connector on the LN-200. Check that the connector is fully seated, see *Figure 46* on *page 285*.

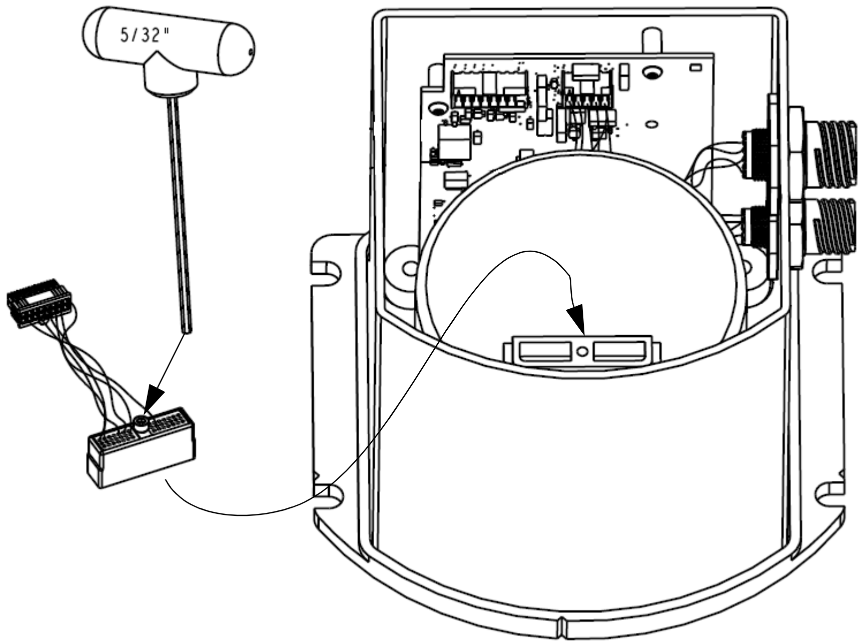


Figure 46: Attach Wiring Harness

2. Connect the Samtec connector at the other end of the wiring harness to the corresponding connector on the internal IMU card, see *Figure 47*. Ensure that the connector is locked in place.

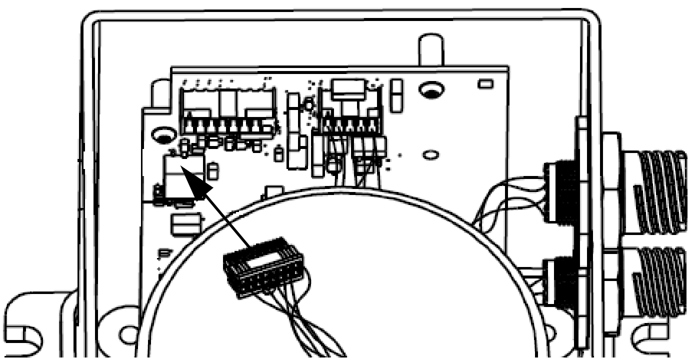


Figure 47: Attach Samtec Connector

F.4 Re-Assemble the SPAN IMU Enclosure

Use an allan key to align the long bolts with the threaded holes in the base, see *Figure 43* on *page 283*. Apply threadlock to threads. Finger tighten the 6 bolts and torque them in a cross pattern to 12 in-lbs. The fully assembled IMU enclosure is shown in *Figure 48* below.

Figure 48: LN-200 SPAN IMU

Appendix G SPAN-SE Interface Card

This appendix provides header descriptions for the SPAN-SE Interface card (NovAtel part number 01018070). The SPAN-SE Interface card is the main interface card within the SPAN-SE enclosure product. The interface card runs the SPAN application while interfacing with the OEMV3 and OEMV2 GNSS receivers. *Figure 49* shows the location of the interface card headers within the SPAN board stack. The board stack contains the interface card and OEMV receivers along with mounting brackets.

For further information on the OEMV receivers, refer to NovAtel technical publication *OM-20000093 OEMV Family Installation and Operation User Manual Rev 9*.

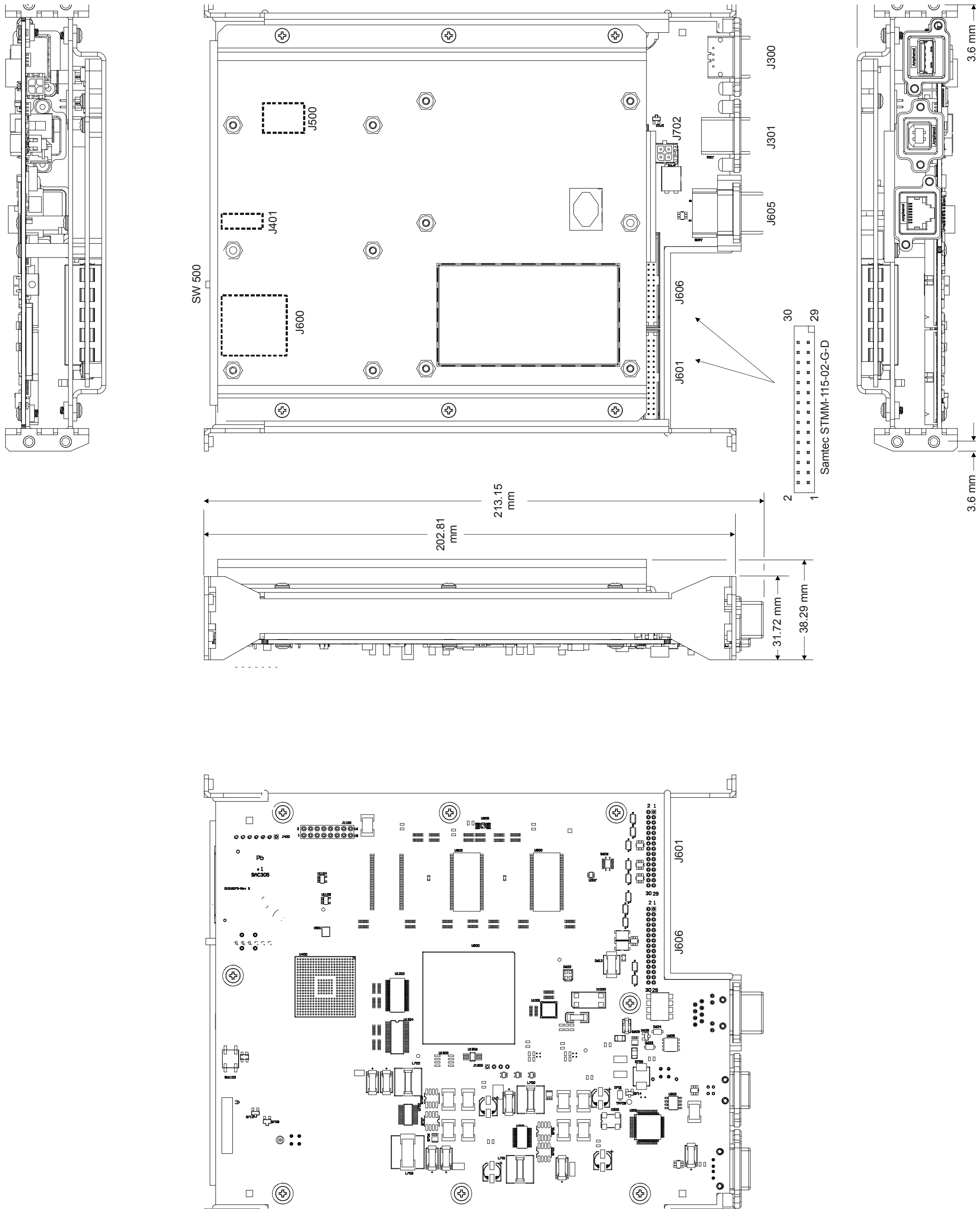


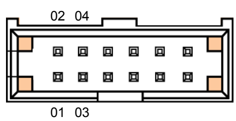
Figure 49: SPAN-SE Interface Card

Table 73 provides a description of the interface card headers, and the remaining tables provide pinouts for headers that external users may need to access.

Table 73: SPAN-SE Interface Card Header Description

Header ID	Function	Connector Format
J300	USB Host	USB Type A jack
J301	USB Device	USB Type B jack
J401	LED Control	2x6 pin header, male, 2mm pitch
J500	Power Button Control	4 position, 3mm pitch, right angle, female
SW500	SD Logging Button	Pushbutton
J600	SD Card Slot	SD memory card slot
J601	Multi Communication Port A	2x15 pin header, male, 2mm pitch
J605	Ethernet	RJ45 jack
J606	Multi Communication Port B	2x15 pin header, male, 2mm pitch
J702	Input Power	4 position, 3mm pitch, vertical, female

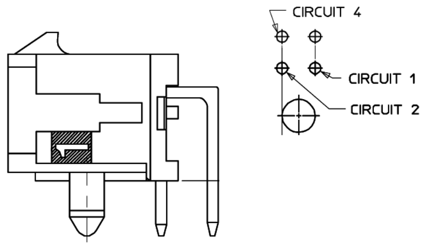
Table 74: J401 (LED Header)



Pin	Description[a]	Signal Levels
1	3v3	
2	3v3	
3	LED1a	3.3 V
4	LED1b	3.3 V
5	LED2a	3.3 V
6	LED2b	3.3 V
7	LED3a	3.3 V
8	LED3b	3.3 V
9	LED4a	3.3 V
10	LED4b	3.3 V
11	LED5a	3.3 V
12	LED5b	3.3 V

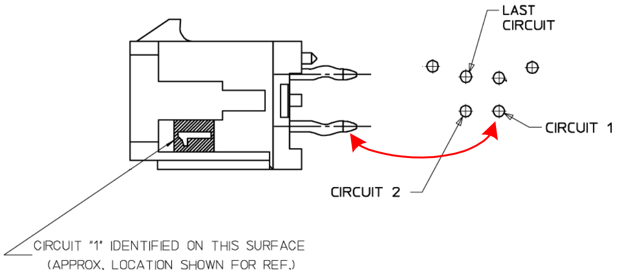
a. "a" lines have 470Ω series resistance and "b" lines have 330Ω.

Table 75: J500 (Power Button Header)



Pin	Description	Signal Levels
1	LED Positive	5.0 V (120Ω series resistor)
2	LED Negative	Return
3	Power Switch	Pull up to 5V
4	GND	

Table 76: J702 (Input Power Header)



Pin	Description	Signal Levels
1	GND	
2	+Vin	Voltage Range +9 to +28 VDC
3	GND	
4	+Vin	Voltage Range +9 to +28 VDC

Table 77: J601 (Multi Communication Header A)

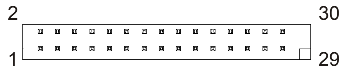
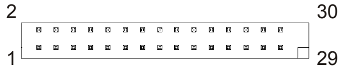
Pin	Description	Signal Levels
		
1	COM1 CTS	
2	GND	
3	COM1 Tx	RS232/RS422 configurable
4	COM1 Rx	RS232/RS422 configurable
5	GND	
6	COM1 RTS	RS232/RS422 configurable
7	IMU Rx	RS232/RS422 configurable
8	IMU CTS	RS232/RS422 configurable
9	IMU RTS	
10	IMU Tx	
11	Event Out 3	0V to 3.3 V
12	GND	
13	Event Out 2	0V to 3.3 V
14	Event Out 1	0V to 3.3 V
15	GND	
16	Event Out 4	0V to 3.3 V
17	GND	
18	Vcc	
19	Spare GPIO 0	
20	Spare GPIO 2	
21	Spare GPIO 3	
22	Spare GPIO 1	
23	Event In 3	-0.3 to 3.75 V
24	GND	
25	Event In 2	-0.3 to 3.75 V
26	Event In 1	-0.3 to 3.75 V
27	GND	
28	Event In 4	-0.3 to 3.75 V
29	COM2 RTS	
30	COM2 Tx	

Table 78: J606 (Multi Communication Header B)

Pin	Description
	
1	COM2 Rx
2	COM2 CTS
3	COM3 CTS
4	GND
5	COM3 Tx
6	COM3 Rx
7	GND
8	COM3 RTS
9	COM3 Rx
10	COM4 CTS
11	COM4 RTS
12	COM4 Tx
13	CAN1 H
14	GND
15	CAN2 L
16	CAN1 L
17	GND
18	CAN2 H
19	GND
20	Vcc
21	OEMV2 RTS
22	OEMV2 Tx
23	OEMV2 CTS
24	OEMV2 Rx
25	OEMV2 Tx
26	GND
27	OEMV3 Rx
28	OEMV3 RTS
29	GND
30	OEMV3 CTS

Appendix H Replacement Parts

The following are a list of the replacement parts available. Should you require assistance, or need to order additional components, please contact your local NovAtel dealer or Customer Service.

H.1 SPAN System

Part Description	NovAtel Part
IMUs (see *Table 1, SPAN-SE Compatible Receiver and IMU Models* on *page 24* for details)	IMU-H58 IMU-H62 IMU-LN200 IMU-FSAS-EI
Receivers (see *Table 1, SPAN-SE Compatible Receiver and IMU Models* on *page 24* for details)	ProPak-V3 SPAN-SE
ProPak-V3 to LN-200 IMU interface cable, see *Figure 19* on *page 72*	01017375
LN-200 power adapter cable, see *Figure 21* on *page 73*	01017821
ProPak-V3 to iIMU-FAS IMU interface cable, see *Table 12* on *page 79*	60723086
ProPak-V3 to HG1700 IMU interface cable (identical to LN-200 cable), see *Figure 19* on *page 72*	01017384
SPAN-SE I/O 1 green multi-connector cable	01018134
SPAN-SE I/O 2 yellow multi-connector cable	01018133
SPAN-SE power cable	01018135
OEMV, **CDU** and *Convert* disk (refer to *page 35* of this manual and to the *OEMV Family Installation and Operation User Manual*)	01017827
SPAN-SE User Guide	OM-20000124
SPAN Technology for OEMVUser manual	OM-20000104
OEMV Family Installation and Operation User Manual	OM-20000093
OEMV Family Firmware Reference Manual	OM-20000094

H.2 Accessories and Options

	Part Description	NovAtel Part
Optional NovAtel GPSAntennas:	Model 532 (for aerodynamic applications)	GPS-532
	Model 702 (for high-accuracy applications)	GPS-702
	Model 702L (for L-band applications)	GPS-702L
	Model 533 (for high-performance base station applications)	GPS-533
Optional RF Antenna Cable:	5 meters	C006
	15 meters	C016

H.3 Manufacturer's Part Numbers

The following original manufacturer's part numbers (and equivalents), for the IMU interface cables, are provided for information only and are not available from NovAtel as separate parts:

Part Description	Part	Deutsch Part	MIL Part
10-pin LEMO plug connector on the HG1700 interface cables	FGG.1K.310.CLAC60Z	-	-
Deutsch (or MIL equivalent) 13-pin connector on the LN-200 interface cable Deutsch (or MIL equivalent) 3-pin connector on the LN-200 power cable	- -	59064-11-35SF 59064-09-98SN	D38999/26B35SF D38999/26A98SN
MIL 22-pin connector on the iIMU-FSAS interface cable	-	-	D38999/26WC35SA
ODU-USA 30-pin connector on the SPAN-SE IMU cables	S23KAC-T30MFG0-01CP [ROHS]	-	-
ODU 4-pin connector on the SPAN-SE power cable	520K0C-P04MFG0-50EP [ROHS]		

Appendix I Frequently Asked Questions

1. *How do I know if my hardware is connected properly?*
   When powered, the HG1700 IMU will make a noticeable humming sound.

2. *I don't hear any sound from my IMU. Why?*
   a. The LN-200 and iIMU-FSAS do not make noise. Check that the IMU interface cable is connected to the IMU DB9 on the yellow SPAN-SE cable port on the SPAN-SE.
   b. When powered, the HG-1700 IMUs makes a noticeable humming sound. If no sound is heard, check that the cable between the receiver and IMU is connected properly. The cable should be connected to the port on the SPAN-SE.
   c. If the cable is connected properly and you still hear no sound from the IMU, check the flex cable mounted on top of the IMU. Refer to the instructions in this manual on proper IMU installation to ensure that the cable is seated properly on the IMU pins.
   d. Check the input power supply. A minimum of 12V should be supplied to the system for stable IMU performance. The supply should also be able to output at least 12W over the entire operating temperature range.

4. *What types of IMUs are supported?*
   a. SPAN currently supports the HG1700 IMU family from Honeywell, the LN-200 from Litton and the iIMU-FSAS from iMAR. Use the SETIMUTYUPE command to specify the type of IMU used (see *page 138*).

5. *Why don't I have any INS logs?*
   a. On start-up, the INS logs are not available until the system has solved for time. This requires that an antenna is attached, and satellites are visible, to the system. You can verify that time is solved by checking the time status in the header of any standard header SPAN log such as BESTPOS. When the time status reaches FINESTEERING, the inertial filter starts and INS messages are available.
   b. Check that the system has been configured properly. See question *3* above.

6. *How can I access the inertial solution?*
   The INS/GNSS solution is available from a number of specific logs dedicated to the inertial filter. The INSPOS, INSPVA, INSVEL, INSSPD, and INSATT logs are the most commonly used logs for extracting the INS solution. These logs can be logged at any rate up to the rate of the IMU data (100 or 200 Hz depending on your IMU model). The MARK*x*PVA logs provide the INS/GNSS solution at the time an input was received on EVENT IN *x*. Further details on these logs are available in *Appendix C, Data Logs* starting on *page 158*.

7. *Can I still access the GNSS-only solution while running SPAN?*
   The GNSS-only solution used when running the OEMV receiver without the IMU is still available when running SPAN. BESTGPSPOS solutions are available at 1 or 5 Hz from any port of SPAN-SE. Other GNSS logs (RANGE, PSRPOS, and so on) can be logged up to 20 Hz from the SPAN-SE ports.

8. *What will happen to the INS solution when I lose GNSS satellite visibility?*

   When GNSS tracking is interrupted, the INS/GNSS solution bridges through the gaps with what is referred to as free-inertial navigation. The IMU measurements are used to propagate the solution. Errors in the IMU measurements accumulate over time to degrade the solution accuracy. For example, after one minute of GNSS outage, the horizontal position accuracy is approximately 2.5 m when using an HG1700 AG58. The SPAN solution continues to be computed for as long as the GNSS outage lasts, but the solution uncertainty increases with time. This uncertainty can be monitored using the INSCOV log, see *page 209*.

Index

D

E

F

Y

Z

Recyclable

Printed in Canada on recycled paper