



A Precise Positioning Technology Company



SPAN-SE™

## USER MANUAL

OM-20000124

Rev 8

# SPAN-SE User Manual

**Publication Number:** OM-20000124  
**Revision Level:** 8  
**Revision Date:** May/2012

This manual reflects SPAN-SE™ firmware version SCD000302RN0000, SPAN-MPPC™ firmware version SKS000200RN0000 and OEMV firmware version 3.803 with a Q model.

## Proprietary Notice

Information in this document is subject to change without notice and does not represent a commitment on the part of NovAtel Inc. The software described in this document is furnished under a licence agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of a duly authorized representative of NovAtel Inc.

The information contained within this manual is believed to be true and correct at the time of publication.

ALIGN, NovAtel, OEMV, ProPak, and RT-20 are registered trademarks of NovAtel Inc.

OEMV-2, OEMV-3, SPAN, SPAN-SE, SPAN-MPPC, ProPak-V3, and RT-2 are trademarks of NovAtel Inc.

All other product or brand names are trademarks of their respective holders.

Manufactured and protected under the following U.S. patents:

#5,101,416	#6,243,409 B1
#5,390,207	#6,608,998 B1
#5,414,729	#6,664,923 B1
#5,495,499	#6,721,657 B2
#5,734,674	#6,728,637 B2
#5,736,961	#6,750,816 B1
#5,809,064	#7,193,559 B2
#6,184,822 B1	#7,346,452
#6,211,821 B1	



© Copyright 2012 NovAtel Inc. All rights reserved. Unpublished rights reserved under International copyright laws. Printed in Canada on recycled paper. Recyclable.

# Table of Contents

<b>Proprietary Notice</b>	<b>2</b>
<b>Software License</b>	<b>14</b>
<b>Terms and Conditions</b>	<b>16</b>
<b>Warranty Policy</b>	<b>18</b>
<b>Customer Support</b>	<b>20</b>
<b>Firmware Updates and Model Upgrades</b>	<b>22</b>
<b>Notices</b>	<b>23</b>
<b>Foreword</b>	<b>27</b>
<b>1 Introduction</b>	<b>29</b>
1.1 Fundamentals of GNSS/INS .....	30
1.2 Models and Features .....	31
<b>2 SPAN-SE Installation</b>	<b>33</b>
2.1 SPAN-SE Hardware Description .....	33
2.2 SPAN-SE Hardware Installation .....	37
2.2.1 Mount Antenna .....	37
2.2.2 Mount IMU .....	38
2.2.3 Connect Interface Cables .....	38
2.2.4 Connect Power .....	40
2.2.5 Power Button .....	40
2.2.6 Connect Additional Equipment .....	43
<b>3 SPAN-SE Operation</b>	<b>44</b>
3.1 Definition of Reference Frames within SPAN .....	44
3.1.1 Local-Level Frame (ENU) .....	44
3.1.2 SPAN Computation Frame .....	45
3.1.3 Enclosure Frame .....	45
3.1.4 Vehicle Frame .....	46
3.2 Communicating with the SPAN System .....	46
3.2.1 INS Window in Connect .....	49
3.3 Software Configuration .....	50
3.3.1 GNSS Configuration .....	50
3.3.2 SPAN IMU Configuration .....	52
3.4 Real-Time Operation .....	54
3.4.1 Configuration for Alignment .....	55
3.4.2 INS Configuration Command Summary .....	55
3.4.3 System Start-Up and Alignment Techniques .....	55
3.4.4 Navigation Mode .....	57
3.4.5 Data Collection .....	58
3.4.6 Lever Arm Calibration Routine .....	59
3.4.7 Vehicle to Body Rotations .....	60
3.4.8 Vehicle to SPAN Frame Angular Offsets Calibration Routine .....	63
3.5 SPAN Wheel Sensor Configuration .....	64
3.5.1 Wheel Sensor Updates Using the Event Input Lines .....	64
3.5.2 Wheel Sensor Updates using the iIMU-FSAS IMU .....	65

3.5.3 Wheel Sensor Updates using the WHEELVELOCITY Command .....	65
3.5.4 Logging Wheel Sensor Data from SPAN-SE .....	65
3.6 Data Collection for Post Processing .....	66
3.7 Status Indicators .....	67
3.7.1 SPAN-SE Status LEDs .....	67
3.7.2 SPAN-MPPC Status LED .....	68
3.8 SD Card .....	69
3.9 Logging Data to the SD Card .....	70
3.9.1 Insert the SD Card .....	70
3.9.2 Prepare the Card .....	70
3.9.3 Select Logs to Send to the SD Card .....	71
3.9.4 Start and Stop Logging .....	71
3.9.5 Log a Pre-Defined List of Logs .....	72
3.9.6 Auto-Logging on Start-Up .....	72
3.9.7 Reading data from the card .....	73
3.10 Adding Timed Camera Triggers .....	73
3.10.1 Configuring the Hardware .....	73
3.10.2 Configuring the Software .....	74
3.10.3 Using Timed Event Pulses .....	74
3.11 Synchronizing External Equipment .....	74
3.11.1 Configuring a Synchronous Output Pulse .....	74
3.11.2 Configuring an Input Strobe .....	75
3.12 SPAN-SE Ethernet Connection .....	76
3.12.1 Configuring for TCP or UDP Operation .....	76
3.12.2 Configuring the Ethernet Connection Settings .....	76
3.12.3 Configuring Log Requests Destined for the Ethernet Port .....	77
3.12.4 Connecting to the Ethernet Port .....	77

## **4 Dual-Antenna Functionality 78**

4.1 Overview .....	78
4.2 Installation .....	78
4.3 Configuring ALIGN with SPAN-SE-D .....	79
4.4 Configuring ALIGN with SPAN-MPPC-D .....	79
4.5 Configuring SPAN with ALIGN on SE-D and MPPC-D Models .....	79
4.5.1 Alignment on a Moving Vessel - Aided Transfer Alignment (Default) .....	80
4.5.2 Alignment on a Stationary Vehicle - Aided Static Alignment .....	80
4.5.3 Unaided Alignment .....	80
4.6 SPAN ALIGN Attitude Updates .....	81

## **5 API Functionality 82**

5.1 Overview .....	82
5.2 Loading User Applications .....	82
5.2.1 Verifying Loaded User Applications .....	83
5.3 Starting, Stopping and Removing a User Application .....	83
5.3.1 Starting a User Application .....	83
5.3.2 Stopping a User Application .....	83
5.3.3 Removing a User Application .....	84

## **6 Variable Lever Arm 85**

6.1 Overview .....	85
6.2 Technical Description .....	85
6.3 Usage .....	87

---

## **Appendix A Technical Specifications** **88**

A.1 SPAN-SE.....	88
A.1.1 SPAN-SE Receiver .....	88
A.2 Inertial Measurement Units (IMUs).....	97
A.2.1 Universal IMU Enclosure.....	97
A.2.2 HG1700 IMU (Single-Connector Enclosure) .....	104
A.2.3 LN-200 IMU (Single-Connector Enclosure).....	107
A.2.4 iIMU-FSAS .....	110
A.2.5 IMU-CPT .....	120

## **Appendix B Commands** **125**

B.1 Command Formats.....	125
B.2 Using a Command as a Log.....	125
B.3 DOS Commands .....	126
B.3.1 DIR - Show Directory .....	127
B.3.2 CD - Change Directory .....	127
B.3.3 FORMAT - Format storage medium.....	127
B.3.4 MKDIR - Make Directory .....	127
B.3.5 RMDIR - Remove Directory .....	127
B.3.6 PWD - Present Working Directory.....	128
B.3.7 FTP .....	128
B.4 SPAN-SE Command Reference .....	129
B.4.1 ALIGNMENTMODE Set the alignment mode .....	129
B.4.2 APPLICATION Start and remove applications .....	130
B.4.3 APPLYVEHICLEBODYROTATION Enable vehicle to body rotation.....	131
B.4.4 ASSIGNLBAND Set L-band satellite communication parameters.....	132
B.4.5 COM Port configuration control .....	135
B.4.6 COMCONTROL Control the RS232 hardware control lines.....	138
B.4.7 COMVOUT Turn power to the ports on or off.....	140
B.4.8 DUALANTENNAPORTCONFIG SPAN-MPPC Dual Antenna Port Setup.....	141
B.4.9 EVENTINCONTROL Control mark input properties .....	142
B.4.10 EVENTOUTCONTROL Control event-out properties.....	143
B.4.11 EXTHDGOFFSET Set the Angular Offset.....	144
B.4.12 FORMAT Format the SD Card .....	145
B.4.13 FRESET Factory reset .....	146
B.4.14 GIMBALSPANROTATION Gimbal frame to SPAN frame rotation .....	147
B.4.15 GNSSCARDCONFIG GNSS port configuration .....	148
B.4.16 ICOMCONFIG Set ICOMx Protocol .....	151
B.4.17 INPUTGIMBALANGLE Input gimbal angles into the SPAN-SE .....	152
B.4.18 IPCONFIG Set IP information.....	153
B.4.19 INSCOMMAND INS control command .....	154
B.4.20 INSZUPT Request Zero Velocity Update .....	155
B.4.21 INSZUPTCONTROL INS Zero Velocity Update Control .....	156
B.4.22 LEVERARMCALIBRATE INS Calibration Command .....	157
B.4.23 LOG Request logs from the receiver .....	159
B.4.24 LOGFILE Log Data to a File on the SD Card .....	163
B.4.25 NMEATALKER Set the NMEA Talker ID.....	164
B.4.26 PSRDIFFSOURCE Set the pseudorange correction source .....	166
B.4.27 RESET Perform a hardware reset.....	169
B.4.28 RTKCOMMAND Reset or set the RTK filter to its defaults.....	170
B.4.29 RTKDYNAMICS Set the RTK dynamics mode.....	171
B.4.30 RTKSOURCE Set the RTK correction source .....	172

B.4.31 RVBCALIBRATE Vehicle to Body Rotation Control .....	174
B.4.32 SAVECONFIG Save current configuration in NVM .....	175
B.4.33 SBASCONTROL Set SBAS test mode and PRN .....	176
B.4.34 SETAUTOLOGGING Start SD Card Logging at Boot-Up.....	178
B.4.35 SETGIMBALORIENTATION .....	179
B.4.36 SETHEAVEWINDOW Set Heave Filter Length .....	180
B.4.37 SETIMUORIENTATION Set IMU Orientation .....	181
B.4.38 SETIMUTOANTOFFSET Set IMU to antenna offset .....	184
B.4.39 SETIMUTOANTOFFSET2 Set IMU to GPS2 antenna offset .....	185
B.4.40 SETIMUTYPE Set IMU type .....	186
B.4.41 SETINITATTITUDE Set initial attitude of SPAN in degrees .....	188
B.4.42 SETINITAZIMUTH Set initial azimuth and standard deviation .....	190
B.4.43 SETINSOFFSET Set INS offset .....	192
B.4.44 SETMARK1OFFSET, SETMARK2OFFSET, SETMARK3OFFSET, SETMARK4OFFSET Set Mark offset.....	193
B.4.45 SETUPSENSOR Add a new camera sensor .....	194
B.4.46 SETWHEELPARAMETERS Set wheel parameters .....	195
B.4.47 SOFTLOADFILE Loads a file to the SPAN-SE.....	197
B.4.48 SOFTPOWER Power down the SPAN-SE .....	198
B.4.49 SPANAUTH Add an authorization code for a new model .....	199
B.4.50 SPANMODEL Switch to a previously authorized model .....	201
B.4.51 TAGNEXTMARK Add a new camera event.....	202
B.4.52 TIMEDEVENTPULSE Add a new camera event .....	203
B.4.53 UNLOG Remove a log from logging control .....	204
B.4.54 UNLOGALL Remove all logs from logging control .....	206
B.4.55 VEHICLEBODYROTATION Vehicle to SPAN frame rotation.....	207
B.4.56 WHEELVELOCITY Wheel velocity for INS augmentation .....	209

## **Appendix C Command Prompt Interface 210**

C.1 DOS.....	211
C.2 Windows .....	212

## **Appendix D Data Logs 213**

D.1 Log Types.....	214
D.1.1 ASCII .....	215
D.1.2 Abbreviated ASCII.....	217
D.1.3 Binary .....	218
D.1.4 ASCII and Binary Logs with Short Headers .....	221
D.1.5 GPS Time Status .....	221
D.1.6 Message Time Stamps .....	223
D.1.7 Log Type Examples .....	224
D.2 NMEA Standard Logs.....	225
D.3 SPAN-SE Logs.....	227
D.3.1 APPLICATIONSTATUS Display installed and running applications.....	228
D.3.2 BESTPOS Best Position and BESTGPSPOS Best GPS Position .....	229
D.3.3 BESTVEL Best Available Velocity Data and BESTGPSVEL Best Available GPS Velocity Data .....	235
D.3.4 BESTLEVERARM/BESTLEVERARM2 IMU to Antenna Lever Arm.....	237
D.3.5 COMCONFIG Current COM Port Configuration .....	238
D.3.6 COMPROTOCOL COM Port Protocol .....	240
D.3.7 CORRIMUDATA and CORRIMUDATAS Corrected IMU measurements .....	241
D.3.8 DIRENT SD Card File List .....	243
D.3.9 EXTHDGOFFSET Log the Angular Offset .....	245

D.3.10 GIMBALLEDPVA	Display gimballed position .....	246
D.3.11 GLOCLOCK	GLONASS Clock Information .....	247
D.3.12 GLOEPHEMERIS	GLONASS Ephemeris Data .....	249
D.3.13 GLORAWEPHEM	Raw GLONASS Ephemeris Data .....	252
D.3.14 GPALM	Almanac Data .....	253
D.3.15 GPGGA	GPS Fix Data and Undulation .....	255
D.3.16 GPGLL	Geographic Position .....	257
D.3.17 GPGRS	GPS Range Residuals for Each Satellite .....	259
D.3.18 GPGSA	GPS DOP and Active Satellites .....	260
D.3.19 GPGST	Pseudorange Measurement Noise Statistics .....	262
D.3.20 GPGSV	GPS Satellites in View .....	264
D.3.21 GPHDT	NMEA Heading Log .....	266
D.3.22 GPVTG	Track Made Good And Ground Speed .....	267
D.3.23 GPZDA	UTC Time and Date .....	268
D.3.24 HEADING	Heading Information .....	269
D.3.25 HEAVE	Heave Filter Log .....	271
D.3.26 ICOMSTATUS	Show communication port status .....	272
D.3.27 IMUTOANTOFFSETS	IMU to Antenna(s) Lever Arm .....	273
D.3.28 INSATT	INS Attitude .....	276
D.3.29 INSATTS	Short INS Attitude .....	277
D.3.30 INSCOV	INS Covariance Matrices .....	278
D.3.31 INSCOVs	Short INS Covariance Log .....	280
D.3.32 INSPOS	INS Position .....	281
D.3.33 INSPOSS	Short INS Position .....	282
D.3.34 INSPOSSYNC	Time Synchronized INS Position .....	283
D.3.35 INSPVA	INS Position, Velocity and Attitude .....	284
D.3.36 INSPVAS	Short INS Position, Velocity and Attitude .....	285
D.3.37 INSSPD	INS Speed .....	286
D.3.38 INSSPDS	Short INS Speed .....	287
D.3.39 INSUPDATE	INS Update .....	288
D.3.40 INSVEL	INS Velocity .....	290
D.3.41 INSVELS	Short INS Velocity .....	291
D.3.42 IPSTATUS	Displays ethernet port settings .....	292
D.3.43 LBANDINFO	L-band Configuration Information .....	293
D.3.44 LBANDSTAT	L-band Status Information .....	295
D.3.45 LOGFILESTATUS	Displays information about system logging .....	300
D.3.46 LOGLIST	List of System Logs .....	301
D.3.47 MAC	MAC Address .....	304
D.3.48 MARK1COUNT, MARK2COUNT, MARK3COUNT, MARK4COUNT	Mark Count..	305
D.3.49 MARK1PVA, MARK2PVA, MARK3PVA, MARK4PVA	Position, Velocity and Attitude at Mark .....	306
D.3.50 MARK1TIME, MARK2TIME, MARK3TIME, MARK4TIME	Time of Mark Input Event 307	
D.3.51 PASHR	NMEA Inertial Attitude Data .....	309
D.3.52 PORTSTATS	Port Statistic .....	310
D.3.53 RANGE	Satellite Range Information .....	312
D.3.54 RANGECMP	Compressed Version of the RANGE Log .....	316
D.3.55 RAWEPHEM	Raw Ephemeris .....	318
D.3.56 RAWIMU	Raw IMU Data .....	319
D.3.57 RAWIMUS	Short Raw IMU Data .....	327
D.3.58 RXCONFIG	Receiver Configuration .....	329
D.3.59 RXSTATUS	Receiver Status .....	331

D.3.60 RXSTATUSEVENT	Status Event Indicator .....	337
D.3.61 SPANVALIDMODELS	Valid Model Information .....	339
D.3.62 TAGGEDMARKxPVA	Output information from a MARK input .....	340
D.3.63 TIME	Time Data .....	341
D.3.64 TIMEDWHEELDATA	Timed Wheel Data .....	343
D.3.65 VARIABLELEVERARM	Display variable lever arm details .....	344
D.3.66 VEHICLEBODYROTATION	Vehicle to SPAN Frame Rotation .....	345
D.3.67 VERSION	Version Information .....	346
D.3.68 WHEELSIZE	Wheel Size .....	348
<b>Appendix E</b>	<b>HG1700 IMU in Universal Enclosure</b>	<b>349</b>
E.1	Disassemble the Universal Enclosure .....	350
E.2	Install the HG1700 Sensor Unit .....	353
<b>Appendix F</b>	<b>HG1700 IMU in SPAN HG Enclosure</b>	<b>360</b>
F.1	Disassemble the SPAN IMU Enclosure .....	361
F.2	Install the HG1700 Sensor Unit .....	363
F.3	Make the Electrical Connections .....	364
F.4	Re-Assemble the SPAN IMU Enclosure .....	365
<b>Appendix G</b>	<b>LN-200 IMU in Universal Enclosure</b>	<b>366</b>
G.1	Disassemble the Universal Enclosure .....	367
G.2	Install the LN-200 Sensor Unit .....	370
<b>Appendix H</b>	<b>LN-200 IMU in SPAN IMU Enclosure</b>	<b>377</b>
H.1	Disassemble the SPAN IMU Enclosure .....	378
H.2	Install the LN-200 Sensor Unit .....	380
H.3	Make the Electrical Connections .....	381
H.4	Re-Assemble the SPAN IMU Enclosure .....	383
<b>Appendix I</b>	<b>SPAN-SE Interface Card</b>	<b>384</b>
<b>Appendix J</b>	<b>SPAN-MPPC Interface Card</b>	<b>390</b>
J.1	SPAN-MPPC Mechanical Drawings .....	391
J.2	SPAN-MPPC Connector Pin-Outs .....	394
J.3	SPAN MPPC Breakout Board .....	398
J.4	SPAN-MPPC to IMU Connection .....	405
<b>Appendix K</b>	<b>Replacement Parts</b>	<b>406</b>
K.1	SPAN System .....	406
K.2	Accessories and Options .....	406
K.3	Manufacturer's Part Numbers .....	407
<b>Appendix L</b>	<b>Frequently Asked Questions</b>	<b>408</b>
	<b>Index</b>	<b>410</b>



# Figures

1	Primary and Secondary Lightning Protection .....	26
1	SPAN-SE Receiver .....	29
2	SPAN System IMUs .....	29
3	SPAN-SE Example Installation .....	34
4	Typical Static Base Set-Up .....	35
5	SD Memory Card .....	36
6	Receiver Enclosure Back Panel .....	36
7	SPAN-SE cable connections .....	39
8	SPAN-SE Power Button .....	41
9	SPAN-SE Power Connection .....	41
10	SPAN-SE LED Indicators .....	42
11	Connect a Device With a MARKIN Pulse .....	43
12	Local-Level Frame (ENU) .....	44
13	Enclosure Frame .....	45
14	Vehicle Frame .....	46
15	SPAN-SE LED Indicators .....	67
16	Status LED Flash Sequence Example .....	68
17	SD Card Access Door .....	69
18	Event Out .....	75
19	Simple Configuration .....	86
20	Rotation Results .....	86
21	SPAN-SE Power Cable .....	92
22	SPAN-SE I/O 1 Green Cable .....	93
23	SPAN-SE I/O 2 Yellow Cable .....	95
24	Universal IMU Enclosure Top/Bottom Dimensions .....	97
25	Universal IMU Enclosure Side Dimensions .....	98
26	IMU Center of Navigation .....	99
27	Universal IMU Enclosure Interface Cable .....	100
28	Universal IMU Enclosure Interface Cable .....	100
29	HG1700 Top/Bottom Dimensions .....	104
30	HG1700 Enclosure Side Dimensions .....	105
31	LN-200 IMU Enclosure Top/Bottom Dimensions and Centre of Navigation .....	107
32	LN-200 Enclosure Side Dimensions .....	108
33	iIMU-FSAS Top/Bottom Dimensions .....	110
34	iIMU-FSAS Enclosure Side Dimensions .....	111
35	iIMU-FSAS Center of Navigation .....	112
36	Universal IMU Interface Cable .....	113
37	iIMU Interface Cable Connections with a SPAN-SE .....	114
38	Corrsys Datron WPT .....	117
39	iMAR iMWS Pre-Installed .....	117
40	IMU-CPT - Side and Perspective View .....	120
41	IMU-CPT Top, Front and Bottom View .....	121
42	IMU-CPT Development Terminated Cable .....	122
43	IMU Center of Navigation (example) .....	181
44	Remove Base .....	350
45	Disconnect Wiring Harness from Enclosure Body .....	351
46	Remove IMU Mounting Plate and Bracket .....	352
47	Remove IMU Mounting Screws .....	353

---

48	Connect IMU to IMU Mounting Plate .....	353
49	Installing IMU to Mounting Plate .....	354
50	Assemble Into Enclosure Body .....	355
51	Fasten Internal Cable Harness .....	356
52	Install O-rings .....	356
53	Install Enclosure Body on the Base .....	357
54	Screw Enclosure Base to Body .....	358
55	Final Assembly .....	359
56	Required Parts .....	360
57	Bolts and Hex Key .....	361
58	Lift Top Cover, Tube Body and 3 Ring Spacer Screws .....	362
59	SPAN IMU Re-Assembly .....	363
60	Attach Flex Cable .....	364
61	Incorrect (Bowed) Flex Cable Installation .....	365
62	Correct (Flat) Flex Cable Installation .....	365
63	HG1700 SPAN IMU .....	365
64	Remove Base .....	367
65	Disconnect Wiring Harness from SDLC Card .....	368
66	IMU Bracket .....	368
67	Remove IMU Bracket/SDLC .....	369
68	Install LN-200 IMU to Base .....	370
69	Install Bracket to Base .....	371
70	Making Connections .....	372
71	Connect Internal Cable Harness .....	373
72	Installing the Enclosure Body to the Base .....	374
73	Screw Enclosure Base to Body .....	375
74	Final Assembly .....	376
75	Required Parts .....	377
76	Bolts and Hex Key .....	378
77	Lift Top Cover and Tube Body .....	379
78	SPAN IMU Re-Assembly .....	380
79	Attach Wiring Harness .....	381
80	Attach Samtec Connector .....	382
81	LN-200 SPAN IMU .....	383
82	SPAN-SE Interface Card .....	385
83	SPAN-MPPC Interface Card .....	391
84	SPAN-MPPC Board Stack (perspective view) .....	392
85	SPAN-MPPC Board Stack (dimensions) .....	393
86	SPAN MPPC Breakout Board .....	398
87	SPAN MPPC Breakout Board - Connectors .....	398

# Tables

1	Supported IMU Types.....	32
2	Receiver Enclosure Back Panel Labels.....	37
3	IMU Power Supply.....	40
4	Power Button States.....	40
5	Inertial Solution Status.....	54
6	Solution Parameters.....	58
7	Positioning Mode LEDs.....	67
8	I/O 1 Green Cable Connector Pin-Outs.....	94
9	I/O 2 Yellow Cable Connector Pin-Outs.....	96
10	Universal IMU Enclosure Specifications.....	97
11	Universal IMU Enclosure Interface Cable Pinouts.....	101
12	Universal IMU Enclosure Electrical and Environmental.....	103
13	HG1700 IMU Specifications.....	104
14	LN-200 IMU Specifications.....	107
15	iIMU-FSAS Specifications.....	110
16	IMU Interface Cable Pin-Out.....	115
17	Legacy iIMU-FSAS Cable Y Adapter Pin-Out.....	116
18	Cable Modification for Corrsys Datron WPT.....	118
19	Technical HW Specs for IMU-CPT.....	120
20	IMU-CPT Connector Pin-Out Descriptions.....	122
21	Mass Storage Device.....	126
22	L-band Mode.....	133
23	COM Serial Port Identifiers.....	136
24	Parity.....	136
25	Handshaking.....	136
26	Tx, DTR and RTS Availability.....	138
27	SPAN-SE COM Port Values.....	139
28	FRESET Target.....	146
29	Serial Port Interface Modes.....	150
30	NMEA Messages.....	164
31	DGPS Type.....	167
32	Dynamics Mode.....	171
33	System Types.....	176
34	Full Mapping Definitions.....	183
35	IMU Type.....	187
36	SETWHEELPARAMETERS Input.....	196
37	Log Type Triggers.....	214
38	ASCII Message Header Structure.....	215
39	Binary Message Header Structure.....	218
40	Detailed Serial Port Identifiers.....	220
41	Short ASCII Message Header Structure.....	221
42	Short Binary Message Header Structure.....	221
43	GPS Time Status.....	221
44	Position or Velocity Type.....	229
45	Solution Status.....	231
46	Signal-Used Mask.....	233
47	Extended Solution Status.....	233
48	Port Protocol.....	240
49	GLONASS Ephemeris Flags Coding.....	250

50	Bits 0 - 1: P1 Flag Range Values.....	250
51	NMEA Positioning System Mode Indicator .....	257
52	Position Precision of NMEA Logs .....	257
53	Lever Arm Type .....	274
54	Lever Arm Source .....	275
55	Heading Update Enums.....	289
56	Wheel Status .....	289
57	L-band Subscription Type .....	293
58	L-band Signal Tracking Status.....	295
59	OmniSTAR VBS Status Word.....	296
60	OmniSTAR HP/XP Additional Status Word .....	297
61	OmniSTAR HP/XP Status Word .....	298
62	Clock Model Status .....	307
63	Tracking State.....	313
64	Correlator Type .....	313
65	Channel Tracking Example.....	313
66	Channel Tracking Status.....	314
67	Range Record Format (RANGECMP only).....	316
68	iIMU-FSAS Status.....	321
69	Litef LCI-1 IMU Status.....	322
70	Mode Indication .....	323
71	HG1700 and LN200 Status.....	323
72	IMU-CPT Status.....	325
73	HG1900 and HG1930 Status .....	326
74	Raw IMU Scale Factors .....	328
75	SPAN Receiver Error .....	332
76	SPAN Receiver Status.....	334
77	Auxiliary 1 Status .....	335
78	OEMV-3 Status.....	335
79	OEMV-2 Status.....	335
80	Status Word .....	338
81	Event Type.....	338
82	OEMV in SPAN-SE Model Designators.....	346
83	SPAN-SE Model Designators .....	346
84	Component Types .....	347
85	VERSION Log: Field Formats .....	347
86	SPAN-SE Interface Card Header Description.....	386
87	J401 (LED Header).....	386
88	J500 (Power Button Header) .....	387
89	J702 (Input Power Header).....	387
90	J601 (Multi Communication Header A).....	388
91	J606 (Multi Communication Header B).....	389
92	SPAN-MPPC Interface Card Header Description .....	394
93	J600 (Multi Communication Header A).....	394
94	J601 (Multi Communication Header B).....	396
95	SPAN MPPC Breakout Board - J1-J23 Test Points.....	398
96	SPAN MPPC Breakout Board - J24 Main Board Connector .....	399
97	SPAN MPPC Breakout Board - J25 USB Host Connector .....	400
98	SPAN MPPC Breakout Board - J26 Main Board Connector.....	400
99	SPAN MPPC Breakout Board - J27 USB Device Connector.....	402
100	SPAN MPPC Breakout Board - J28 Ethernet Connector.....	402
101	SPAN MPPC Breakout Board - J29 Power Connector.....	402

---

102	SPAN MPPC Breakout Board - J30 Serial Port Connection.....	402
103	SPAN MPPC Breakout Board - J31 Serial Port Connection.....	403
104	SPAN MPPC Breakout Board - J32 Serial Port Connection.....	403
105	SPAN MPPC Breakout Board - J33 Serial Port Connection.....	403
106	SPAN MPPC Breakout Board - J34 Serial Port Connection.....	404
107	SPAN MPPC Breakout Board - J35 Serial Port Connection.....	404
108	SPAN-MPPC IMU Connections.....	405

# 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 website) (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 upgrade from the NovAtel website); 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. Term and Termination:** 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. Warranty:** 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. Indemnification:** 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. Disclaimer and Limitation of Liability:**

- (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. Governing Law:** 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. Customer Support:** 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, website: <http://www.novatel.com> or write to:  
NovAtel Inc.

Customer Service Dept.  
1120 - 68 Avenue NE,  
Calgary, Alberta, Canada T2E 8S5

# Terms and Conditions

## 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™ and SPAN-MPPC™	One (1) Year
IMU Units (return to manufacturer) <sup>1</sup>	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 19*).



**Only return an IMU to its manufacturer and not to NovAtel.  
When returning LN200 and HG1700 to the manufacturer, the IMU sensor must  
be removed from the NovAtel housing prior to shipment.**

## **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, "EXPEDITORS", or for Litton, "FOR CUSTOMS CLEARANCE BY: FedEx Trade Networks, 19601 Hamilton Ave. Torrance, CA 90502-1309, U.S.A.".

### **1. Return IMU to its manufacturer, as follows:**

Honeywell:	Honeywell International Inc. 2600 Ridgway Parkway ( <i>Ridgway is really not spelled with an 'e'</i> ) Minneapolis, MN 55413
iMar:	iMAR GmbH Im Reihersbruch 3, D-66386 St. Ingbert, Germany
KVH:	KVH IMUs should be returned to NovAtel for repair. Contact Customer Support for detailed information.
Litef:	IMU-LCI IMUs should be returned to NovAtel for repair. Contact Customer Support for detailed information.
Litton:	Northrop Grumman/Litton Systems, Inc. Navigation Systems Division (NSD) 21240 Burbank Blvd., Woodland Hills, CA 91367

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 Support

## NovAtel Knowledge Base

If you have a technical issue, browse to the NovAtel website at [www.novatel.com](http://www.novatel.com) | *Support | Helpdesk and Solutions | Search Knowledge and Forums*. Through this page, you can search for general information about GNSS and other technologies, information about NovAtel hardware and software, and installation and operation issues.

## Before Contacting Customer Support

Before contacting NovAtel Customer Support about a software problem perform the following step

1. Log the following data to a file on your PC for 30 minutes:

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

2. Send the file containing the log to NovAtel Customer Support, using either the NovAtel FTP site at *Support | Firmware/Software and Manuals | Access FTP Site* on the NovAtel website at [www.novatel.com](http://www.novatel.com) or through the [support@novatel.com](mailto:support@novatel.com) e-mail address.
3. You can also issue a FRESET command to the receiver to clear any unknown settings.



The FRESET command will erase all user settings. You should know your configuration and be able to reconfigure the receiver before you send the FRESET command.

If you are having a hardware problem, send a list of the troubleshooting steps taken and results.

## Contact Information

Use one of the following methods to contact NovAtel Customer Support:

Call the NovAtel Hotline at 1-800-NOVATEL (U.S. & Canada) [1-800-668-2835] or +1-403-295-4900 (international)	
Fax: +1-403-295-4901 E-mail: <a href="mailto:support@novatel.ca">support@novatel.ca</a> Website: <a href="http://www.novatel.com">http://www.novatel.com</a>	Write: NovAtel Inc. Customer Support Department 1120 - 68 Avenue NE Calgary, AB Canada, T2E 8S5

# Firmware Updates and Model Upgrades

Firmware *updates* are firmware releases, which include fixes and enhancements to the receiver functionality. Firmware updates are released on the website as they become available. Model *upgrades* enable features on the receiver and may be purchased through NovAtel authorized dealers.

Contact your local NovAtel dealer first for more information. To locate a dealer in your area visit [www.novatel.com](http://www.novatel.com) | *Where to Buy* | *Dealer Network* or contact NovAtel Customer Support directly.



You can only perform a complete firmware reload over a serial COM1, but you can load application firmware with the SOFTLOADFILE command.

# Notices

The following notices apply to the SPAN-SE and SPAN-SE-D.



Changes or modifications to this equipment not expressly approved by NovAtel Inc. could result in violation of FCC, Industry Canada and CE Marking rules and void the user's authority to operate this equipment.

## FCC NOTICES

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

SPAN-SE and SPAN-SE-D comply with the radiated and conducted emission limits for a Class B digital device. The Class B limits are designed to provide reasonable protection against harmful interference in a residential installation.

The equipment listed generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Re-orient or relocate the receiving antenna
- Increase the separation between the equipment and the receiver
- Connect the equipment to an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help



To maintain compliance with the limits of a Class B digital device, you must use properly shielded interface cables (such as Belden #9539 or equivalent) when using the serial data ports, and double-shielded cables (such as Belden #9945 or equivalent) when using the I/O strobe port.



A shielded Ethernet cable must be used to maintain emission and immunity compliance!

## Industry Canada

SPAN-SE Class B digital apparatus comply with Canadian ICES-003.


SPAN-SE appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

## CE NOTICE

The enclosures carry the CE mark.

"Hereby, NovAtel Inc. declares that this SPAN-SE and SPAN-SE-D are in compliance with the essential requirements and other relevant provisions of Directive 1999/5/EC."

## WEEE NOTICE

If you purchased your SPAN-SE or SPAN-SE-D product in Europe, please return it to your dealer or supplier at the end of its life. The objectives of the European Community's environment policy are, in particular, to preserve, protect and improve the quality of the environment, protect human health and utilise natural resources prudently and rationally. Sustainable development advocates the reduction of wasteful consumption of natural resources and the prevention of pollution. Waste electrical and electronic equipment (WEEE) is a regulated area. Where the generation of waste cannot be avoided, it should be reused or recovered for its material or energy. WEEE products may be recognized by their wheeled bin label ().<sup>1</sup>

## RoHS NOTICE

SPAN-SE and SPAN-SE Dual are compliant with the European Union (EU) Restriction of Hazardous Substances (RoHS) Directive 2002/95/EC.

## Lightning Protection Notice

### What is the hazard?

A lightning strike into the ground causes an increase in the earth's potential which results in a high voltage potential between the centre conductor and shield of the coax cable. This high voltage develops because the voltage surge induced onto the center conductor lags in time behind the voltage surge induced onto the shield.

### Hazard Impact

A lightning strike causes the ground potential in the area to rise to dangerous levels resulting in harm to personnel or destruction of electronic equipment in an unprotected environment. It also conducts a portion of the strike energy down the inner conductor of the coax cable to the connected equipment.



Only qualified personnel, (electricians as mandated by the governing body in the country of installation) may install lightning protection devices.

---

1. Visit [www.novatel.com](http://www.novatel.com) | *Products* | *WEEE and RoHS* for more information.



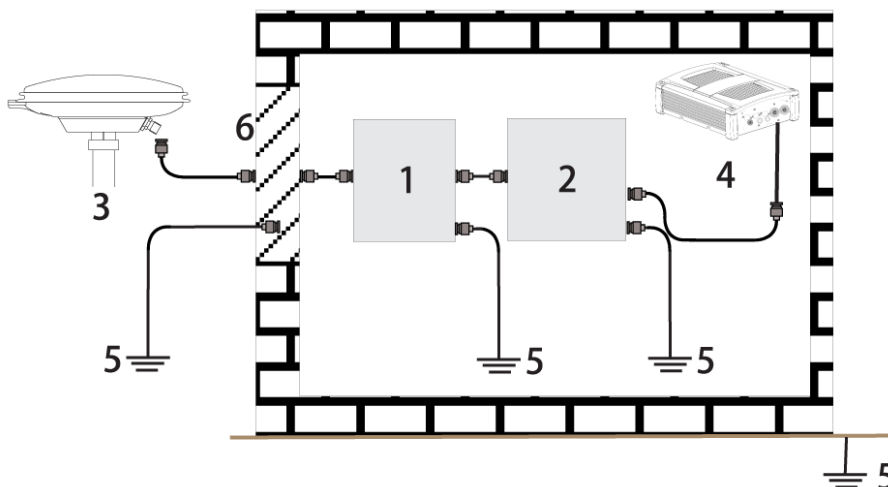
### **Actions to Mitigate Lightning Hazards**

1. Do not install antennas or antenna coaxial cables outside the building during a lightning storm.
2. It is not possible to avoid over-voltages caused by lightning, but a lightning protection device may be used to shunt a large portion of the transient energy to the building ground reducing the over-voltage condition as quickly as possible.
3. Primary lightning protection must be provided by the operator/customer according to local building codes as part of the extra-building installation.
4. To ensure compliance with clause 7 "Connection to Cable Distribution Systems" of EN 60950-1, Safety for Information Technology Equipment, a secondary lightning protection device must be used for in-building equipment installations with external antennas. The following device has been approved by NovAtel Inc.:

Polyphaser - Surge Arrestor DGXZ+24NFNF-A

If this device is not chosen as the primary lightning protection device, the device chosen must meet the following requirements:

- UL listed, or equivalent, in country of installation (for example, TUV, VDE and so on) for lightning surge protection
  - The primary device must be capable of limiting an incoming surge to 10kV
5. The shield of the coaxial cable entering the building should be connected at a grounding plate at the building's entrance. The lightning protection devices should have their chassis grounded to the same ground near to the building's entrance.
  6. The primary and secondary lightning protections should be as close to the building's entrance as possible. Where feasible they should be mounted onto the grounding plate itself. See also *Figure 1, Primary and Secondary Lightning Protection* below.



**Figure 1: Primary and Secondary Lightning Protection**

Reference	Description
1	Primary Lightning Protection Device
2	Secondary Lightning Protection Device
3	External Antenna
4	GNSS Receiver
5	To Ground
6	Grounding plate or grounding point at the building's entrance



Acceptable choices for Earth Grounds, for central buildings, are the following:

- Grounded interior metal cold water pipe within five feet (1.5 m) of the point where it enters the building
- Grounded metallic service raceway
- Grounded electrical service equipment enclosure
- Eight-foot grounding rod driven into the ground (only if bonded to the central building ground by #6, or heavier, bonding wire)

These installation instructions are the minimum requirements for receiver and antenna installations. Where applicable, follow the electrical codes for the country of installation. Examples of country codes include:

- USA            National Electrical Code (NFPA 70)
- Canada       Canadian Electrical Code (CSA C22)
- UK            British Standards Institute (BSI 7671)

# 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 21.

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](http://www.novatel.com) through *Support | Firmware/Software and Manuals* 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 website at [www.novatel.com](http://www.novatel.com) for details.




## What's new in this version of the User Guide?

This version of the SPAN-SE user manual provides users with additional information about the following:

- The SPAN-SE operational procedures have been changed to show NovAtel Connect.
- Added the new log and command for Heave functionality.

## Conventions

The following conventions have been used in this manual:

	Note that provides information to supplement or clarify the accompanying text.
	Caution that a certain action, operation or configuration may result in incorrect or improper use of the product.
	Warning that a certain action, operation or configuration may result in regulatory noncompliance, safety issues or equipment damage.

## Prerequisites

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

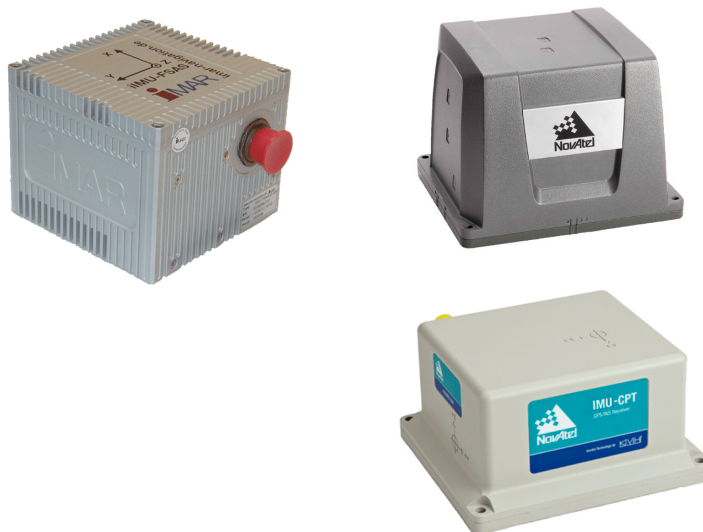
Although previous experience with Windows is not necessary to use the SPAN system software, familiarity with certain actions that are customary in Windows will assist in the usage of the program. This manual has been written with the expectation that you already have a basic familiarity with Windows.

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 offers 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 (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 three 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, Supported IMU Types on page 32* and *Table 35, IMU Type on page 187*.
- PC Software - Real-time data collection, status monitoring and receiver configuration is possible through NovAtel's Connect software utility, see *Section 3.2 on page 46*.
- A dual-frequency GNSS antenna.

The GNSS receiver is connected to the IMU enclosure with an RS-232 or RS-422 serial link, which is the dedicated IMU COM port in SPAN-SE (and SPAN-MPPC). 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, 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.

Each model is capable of multiple positioning modes of operation. For a discussion on GNSS positioning, refer to *APN-051 Positioning Modes of Operation* available from our website at [www.novatel.com](http://www.novatel.com) | *Support* | *Knowledge and Learning* | *Application Notes*.

Each model has the following standard features:

- NovAtel's advanced OEMV L1/L2 GNSS/GLONASS and PAC technology
- Four bidirectional COM ports, three of which support data transfer rates of up to 921,600 bits/s <sup>1</sup>
- A USB port for PC communication
- A serial port capable of communication with an IMU. See also *Table 1 on page 32*.
- An Ethernet port for TCP (or UDP) communication with the receiver
- Field-upgradeable firmware (program software). What makes most models different from others 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 199* for further details on this topic.

---

1. Rates higher than 115,200 are not standard on most PCs and may require extra PC hardware



NovAtel's SPAN-SE-S receiver is configured at the factory for single antenna operation and features one GNSS OEMV-3 receiver inside, while the SPAN-SE-D is manufactured with dual GNSS receivers and can be configured for single or dual antenna operation. There are no hardware or software upgrade paths available from the SE-S to the SE-D, so NovAtel recommends the dual-antenna SPAN-SE-D if you are unsure if your application required a single or dual antenna.

SPAN currently supports specific Honeywell, iMAR, Litton, KVH and Litef 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 88* for details.

**Table 1: Supported IMU Types**

Model Name	Max. Output Rate	Compatible IMUs
IMU-H62 UIMU-H62	100 Hz	HG1700-AG62
IMU-H58 UIMU-H58	100 Hz	HG1700-AG58
IMU-LN200 UIMU-LN200	200 Hz	LN-200 200 and 400 Hz models
IMU-SE-FSAS-EI	200 Hz	iIMU-FSAS
UIMU-LCI	200 Hz	LITEF LCI-1
IMU-CPT	100 Hz	IMU-CPT
IMU-HG1900	100 Hz	HG1900-CA50
IMU-HG1930	100 Hz	HG1930-CA50
No IMU	-	-

Each SPAN-SE has the following extra features:

- Rugged shock, water, and dust-resistant enclosure
- A removable SD Card slot for on-board data collection

Each SPAN-MPPC has the following extra features:

- Small volume for size-restricted applications. See *Appendix J, SPAN-MPPC Interface Card* starting on *page 390* for specifications.
- OEMV3 form factor for easy integration into current systems.



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, since the raw data (i.e. RAWIMUS and INSPVAS) and solution (i.e. INSPVA) have a low output latency.

### 2.1 SPAN-SE Hardware Description

The basic hardware setup consists of a SPAN-SE receiver (see *Figure 1* on *page 29*) connected to an IMU (see *Figure 2* on *page 29*), a GNSS antenna and a power supply.

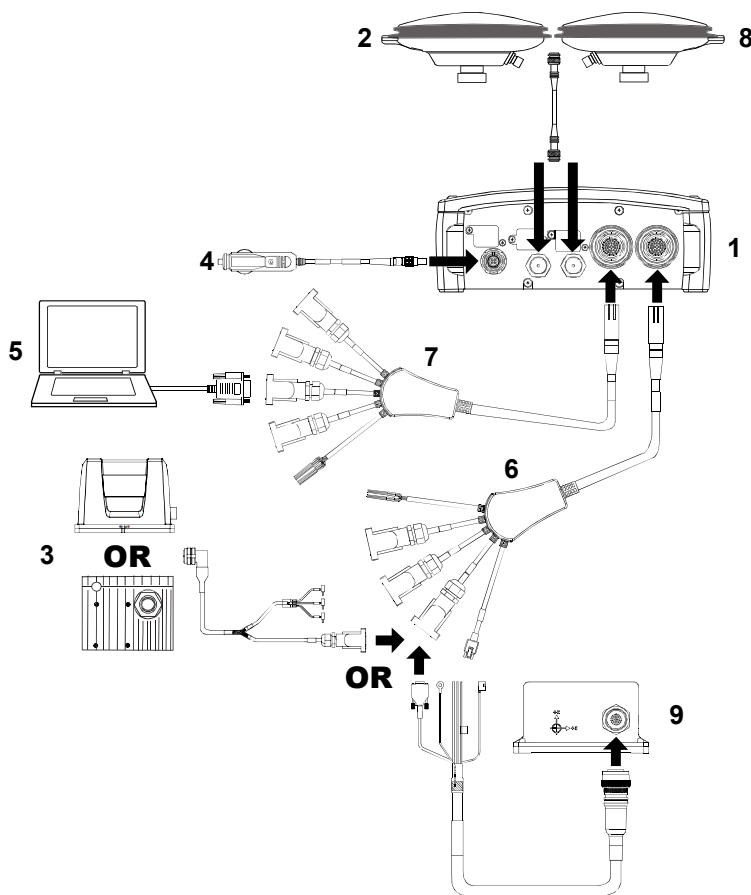
For real time differential operation, a communication link between the base and rover(s) is necessary. This can be a null-modem cable or a radio link.

*Figure 4* on *page 35* shows a basic setup, and also shows the setup with an optional second antenna. For more details on the connections between the SPAN-SE receiver and the iIMU, see *Figure 37*, *iIMU Interface Cable Connections with a SPAN-SE* on *page 114*.

If your IMU enclosure and IMU have come separately, additional installation instructions for installing the IMU can be found in *Appendix E, IMU Installation* starting on *page 341*.

For more information on SPAN-SE cables, please see *Appendix A, Technical Specifications* on *page 88*.

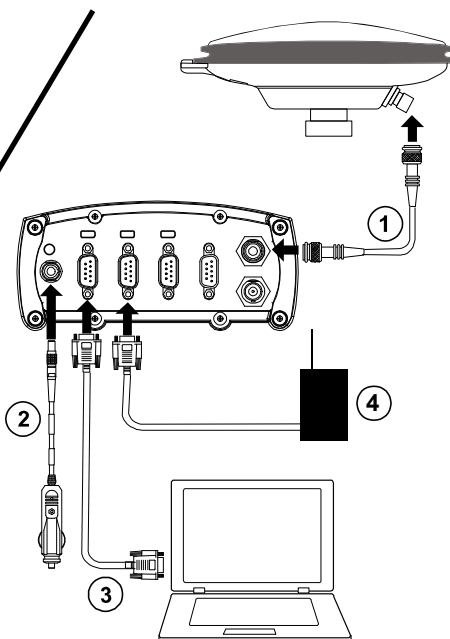
### Figure 3: SPAN-SE Example Installation



1. SPAN-SE receiver with an on-board SD Card for data storage.
2. User-supplied NovAtel GNSS antenna.
3. Connect an LN-200, HG-1700, iIMU FSAS or a Litef-LCI with an 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.2.4.1, iIMU-FSAS Interface Cable* starting on *page 113*.  
Connections for the HG-900 and HG-930 IMUs are not shown on this diagram. For information about integrating the HG1900 and HG1930 IMUs with a SPAN-SE, contact NovAtel Customer Support.
4. User-supplied power supply:  
SPAN-SE (1): +9 to +28 V DC  
Separate supply for IMU (3): see *Table 3* on *page 40*
5. User-supplied computer for setting up and monitoring, to one of the four available COM ports, the USB device port or the Ethernet port.
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 95*.
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 93*.
8. Optional second user-supplied NovAtel GNSS antenna. Refer to Chapter 4 Dual-Antenna Functionality on *page 78*.
9. IMU-CPT interface cable connected to the connector labelled IMU and the event line labelled EVENT-IN 4 both on the SPAN-SE I/O 2 yellow cable.

**Figure 4: Typical Static Base Set-Up**

Rover set up (refer to  
*Figure 3 on page 34*)



1. Connect antenna to the receiver.
2. Connect user supplied power (refer to *Table 3 on page 40*).
3. Connect user supplied PC for set up and monitoring to COM1.
4. Connect user supplied radio device to COM2.



Ensure a user-supplied radio device is connected to the connector labelled OEMV3 on the SPAN-SE I/O green cable.

The sections that follow outline how to set up the system's parts and cables.



Data can be collected through any of the peripheral devices: USB, Ethernet, or serial COM ports. If you have chosen to connect the receiver using a COM port, the port must be 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, SD Card* starting on *page 69*.



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



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.

### 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 a way 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 34, Full Mapping Definitions on page 183*, but this can make interpreting the raw IMU and attitude output more difficult.

Also, it is important to measure the distance from the IMU center of navigation to the antenna phase center (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 59*. See also *Appendix A, Technical Specifications starting on page 88*, 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, especially in RTK mode.**
2. The offset from the IMU center of navigation to the antenna phase center, 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.2.2.3, Electrical and Environmental on page 106* for more information on signals, wiring and pin-out information of the SPAN-SE port and its cables.

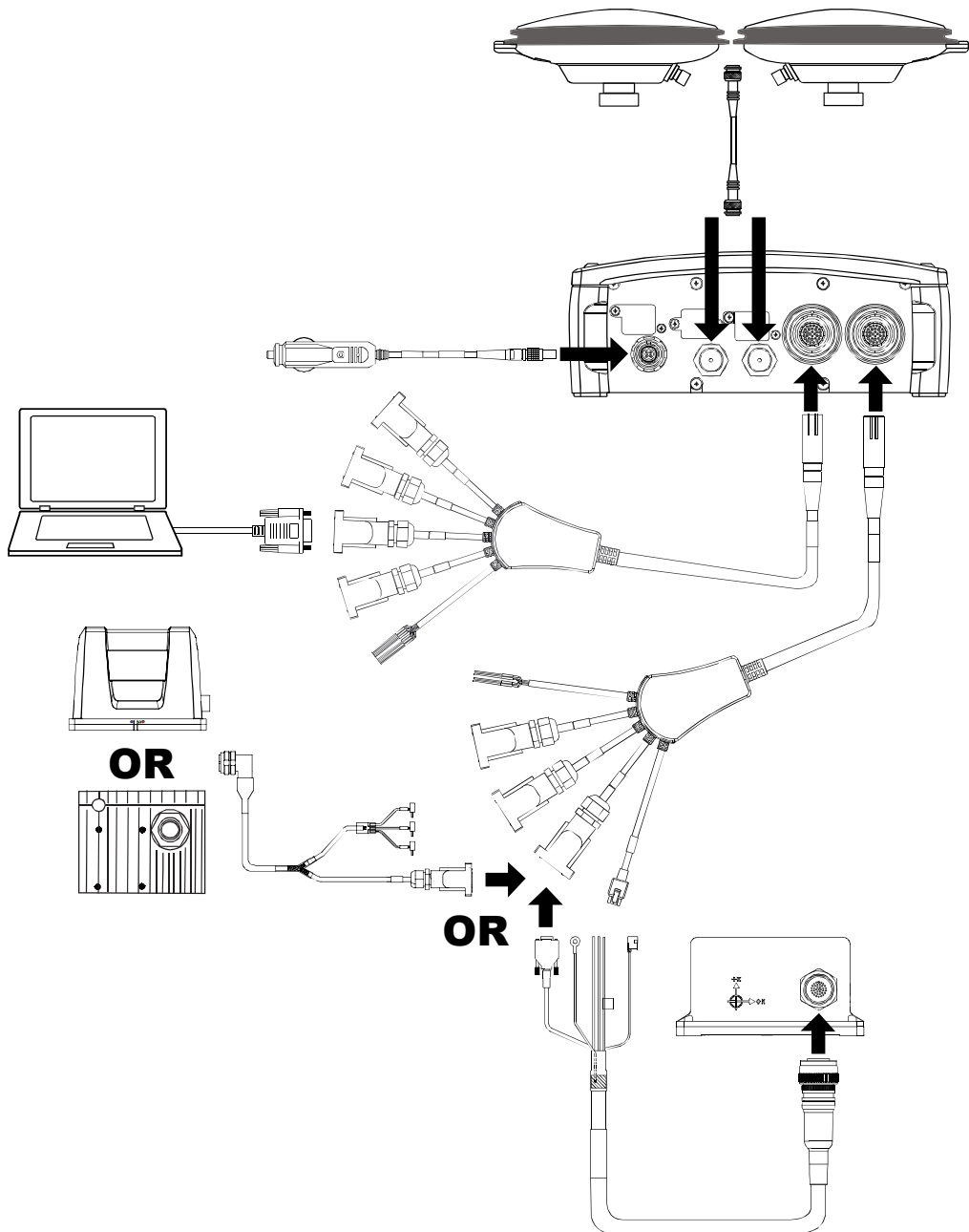
1. Connect the I/O 1 green cable's 30-pin connector to the I/O 1 green port on the SPAN-SE.
2. Connect the I/O 2 yellow cable's 30-pin connector to the I/O 2 yellow port on the SPAN-SE.
3. Connect a communications cable.

If you want to connect via a serial connection, the I/O 1 green cable has 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.

If an Ethernet connection is required, connect a shielded 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 interface cable. The figure below shows the Universal Enclosure connections. IMU-SE-FSAS-EI connections with a SPAN-SE, refer to *iIMU-FSAS Interface Cable on page 113*. For the IMU-CPT connections with a SPAN-SE, refer to *IMU-CPT Cable on page 122*.  
Connections for the HG1900 and HG1930 IMUs are not show on this diagram. For information about integrating the HG1900 and HG1930 IMUs with a SPAN-SE, contact NovAtel Customer Support.



**Figure 7: SPAN-SE cable connections**

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 40*.

**Table 3: IMU Power Supply**

IMU	Power Requirement
LN-200, HG1900, HG1930, HG1700 and Litef LCI-1	+12 to +28 V DC
iIMU-FSAS	+10 to +34 V DC
IMU-CPT	+9 to +18 V DC

For pin-out information on the power connector on the SPAN-SE, see *Section A.1.1.1, Power Adapter Cable (NovAtel part number 01018135S) on page 92*. Details on each IMU's power ports and cables are in the IMU appendices starting on *page 97*.

## 2.2.5 Power Button

The power button on the front of the SPAN-SE, see *Figure 8*, 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**

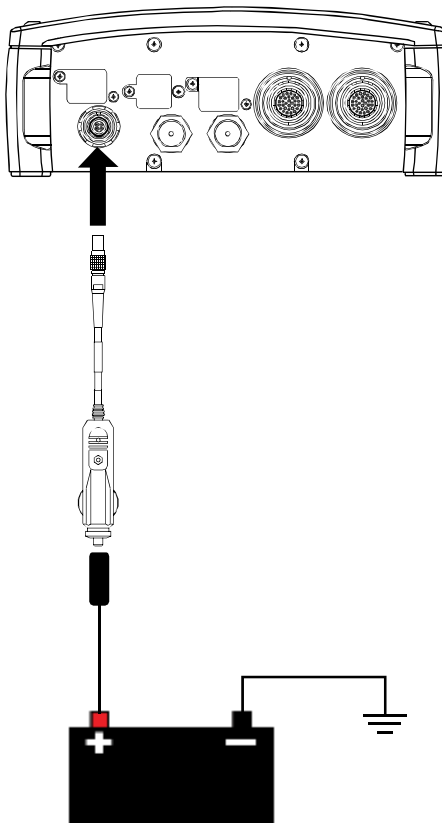
Seconds Button is Held	Button Action
3-10 seconds	SPAN-SE is powered off
> 10 seconds	Factory reset (see the FRESET command on <i>page 146</i> )





**Figure 8: SPAN-SE Power Button**

When installing the SPAN-SE in a vehicle, we recommend that you put a back-up battery between the receiver and its voltage supply to act as a power buffer. 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 reset.



**Figure 9: SPAN-SE Power Connection**

2.2.5.1 Power Button Behavior

After you press the power button, the front panel LED indicators (shown in *Figure 10*) will operate as follows, with the changes in LED status occurring in 0.5 second intervals:

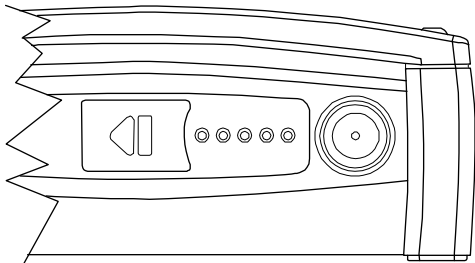


Figure 10: SPAN-SE LED Indicators

*Press and hold down power button.*

SD Card	GPS 1	GPS 2	INS	IMU
RED	OFF	OFF	OFF	OFF
RED	RED	OFF	OFF	OFF
RED	RED	RED	OFF	OFF
RED	RED	RED	RED	OFF
RED	RED	RED	RED	RED
RED	RED	RED	RED	RED
OFF	OFF	OFF	OFF	OFF

*Release power button to RESET.*

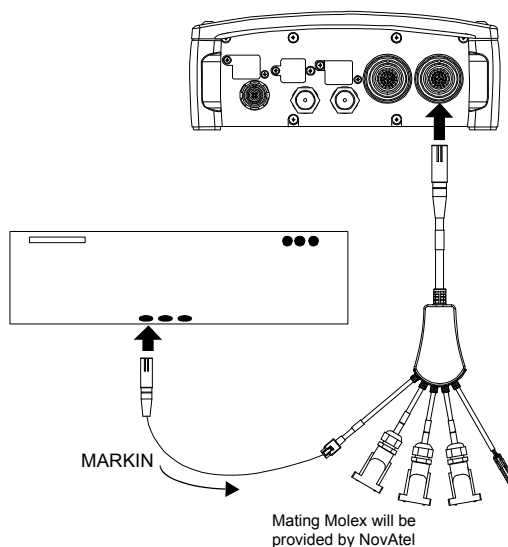
*If power button is held for another 7 seconds, all LEDs flash RED again indicating a FRESET.*

SD Card	GPS 1	GPS 2	INS	IMU
OFF	OFF	OFF	OFF	OFF
RED	RED	RED	RED	RED
OFF	OFF	OFF	OFF	OFF

*Release button for FRESET.*

## 2.2.6 Connect Additional Equipment

If you have additional equipment to connect to your system requiring an output timing pulse, or an input pulse into SPAN-SE, see *Section 3.11, Synchronizing External Equipment* starting on *page 74*. See *Section A.1.1.3, I/O 2 Yellow Cable (NovAtel part number 01018133)* on *page 95* 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 left to the SPAN-SE I/O port.



**Figure 11: Connect a Device With a MARKIN Pulse**

Before operating your SPAN system, ensure that you have followed the installation and setup instructions in *Chapter 2, SPAN-SE Installation* starting on page 33.

You can use NovAtel's Connect software to configure receiver settings and monitor data in real-time.

SPAN system output is compatible with post-processing software from NovAtel's Waypoint Products Group. Visit our website at [www.novatel.com](http://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 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

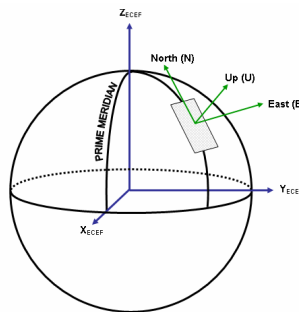


Figure 12: Local-Level Frame (ENU)

### 3.1.2 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 34* on *page 183*. This frame is also known as the body frame and, in this manual, the SPAN frame. It is the default frame that attitude is output in, and its origin is at the Center of Navigation.

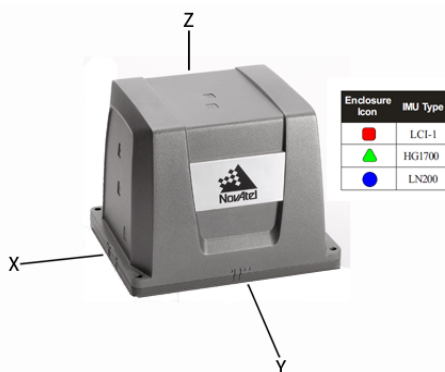
### 3.1.3 Enclosure Frame

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.

When the IMU enclosure Z is pointing up, the mapping (see SETIMUORIENTATION command on *page 181*) is 5. You can force the SPAN computation frame to be the same as the IMU enclosure frame by sending the following command: SETIMUORIENTATION 5.

The origin of this frame is not the enclosure center but the Center of Navigation (sensor center). You can determine the Center of Navigation of the enclosure using the enclosure markings. In the example of the Litef LCI-1 IMU, shown in *Figure 13* in the Universal IMU Enclosure, the Center of Navigation is indicated by the  icon. The HG1700 and LN200 IMUs may be housed in their own enclosures. See *Appendix A.2.3, LN-200 IMU (Single-Connector Enclosure)* starting on *page 107* and *Appendix A.2.2, HG1700 IMU (Single-Connector Enclosure)* starting on *page 104* for the LN200 and HG1700 enclosure markings respectively.

Contact NovAtel for center of navigation drawings for the HG1900 and HG1930 IMUs.



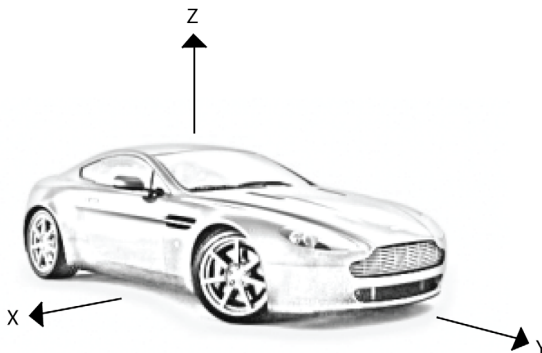
**Figure 13: Enclosure Frame**

### 3.1.4 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 207* for information on entering the rotation into the system and see the RVBCALIBRATE command on *page 174* for information on calibrating this rotation.



**Figure 14: Vehicle Frame**

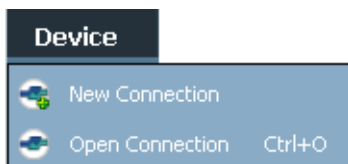
## 3.2 Communicating with the SPAN System

Once the receiver is connected to the PC, antenna, and power supply, install NovAtel's OEMV PC Utilities (Connect and Convert4), available from the NovAtel website at [www.novatel.com](http://www.novatel.com) through *Support | Firmware/Software and Manuals*. 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 Connect Help file for more details on Connect. The Help file is accessed by choosing *Help* from the main menu in Connect.

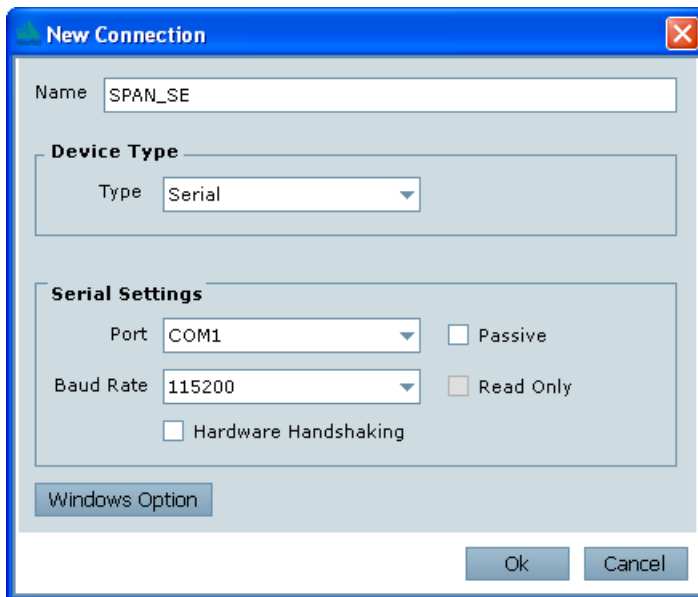
Start Connect on your PC to enable communication:

1. Launch Connect from the *Start* menu folder specified during the installation process. The default location is *Start | All Programs | NovAtel PC Software | NovAtel Connect*.
2. To define a new connection, select *New Connection* from the *Device* menu. If a connection is already defined or if connections were imported from NovAtel Connect, choose *Open*

Connection and skip to step 9.

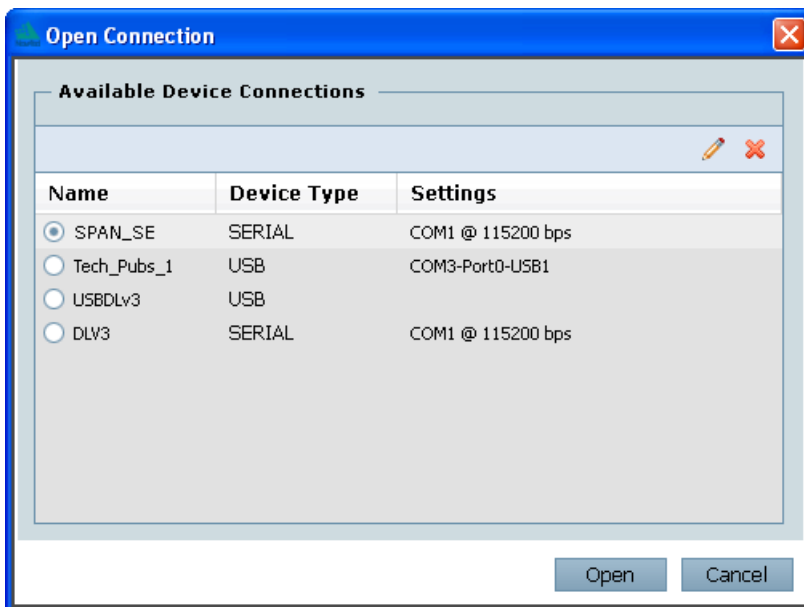


3. Use the New connection dialog to add a new configuration.



4. Select *Serial* or *Network* from the *Type* list.  
If selecting a network connection, you must have configured an IP address into the receiver prior to connecting, see *SPAN-SE Ethernet Connection* on page 76.
5. Select the computer port that the SPAN receiver is connected to from the *Port* list.
6. Select *115200* from the *Baud Rate* list.
7. Ensure the *Hardware Handshaking* check box is cleared.
8. Select *OK* to save the new device settings.

9. Select the new configuration from the *Available Device Connections* area of the *Open Connection* window.



10. Select the *Open* button to open SPAN receiver communications.  
As Connect establishes the communication session with the receiver, a progress box is displayed.
11. Select *Tools | Logging Control Window* from the Connect main menu to control the receiver's logging to files and serial ports. Refer to the Connect on-line Help for more information.
12. Use the *Console* window to enter commands. See also *Section 3.6, Data Collection for Post Processing* on page 66.




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

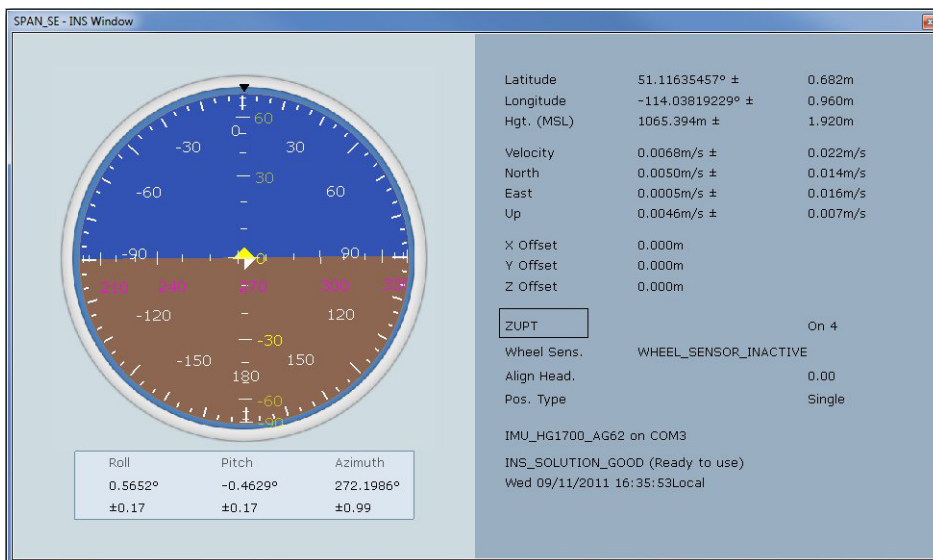


### 3.2.1 INS Window in Connect

Connect 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 Connect is described below. Please refer to the *OEMV Family Installation and Operation User Manual* for more details on Connect 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 the Connect SPAN wizard. The *Wheel Sens.* field shows the status of the wheel sensor, as specified by the INSUPDATE log. The *ALIGN® Head.* field shows the heading from the dual antenna ALIGN solution as specified by the HEADING log. The Position Type field shows the GPS-only position type from the BESTGPSPOS log. 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 or OmniSTAR VBS.

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

After the base and SPAN rover are set up, you can configure them as shown in the configuration examples below, which show how to configure 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 Connect software utility. See the *Connect Help* and its wizard screens for more information.

See *GNSSCARDCONFIG* GNSS port configuration on page 148 for details.

#### RTK BASE CONFIGURATION



The Connect position mode wizard currently cannot configure firmware version SCD000300 or SKS000200 to receive RTK corrections. Instead, configuration should be done manually through the Connect console using the *GNSSCARDCONFIG* command.

For information about RTK configuration of an OEMV-equipped base station, refer to *OEMV Family Installation and Operation User Manual*.

**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]
[GPGBA Output]
```

For example, through the SPAN-SE COM1-COM4, USB, or Ethernet port, enter:

<b>RTCA</b>	<code>gnsscardconfig rtca none 9600</code>	
<b>RTCM</b>	<code>gnsscardconfig rtcm none 9600</code>	
<b>RTCMV3</b>	<code>gnsscardconfig rtcmv3 none 9600</code>	
<b>CMR+</b>	<code>gnsscardconfig cmr none 9600</code>	
<b>CMR</b>	<code>gnsscardconfig cmr none 9600</code>	<i>(same as CMR+)</i>



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

If you are using SPAN-SE or SPAN-MPPC with an RTK correction methodology that requires knowledge of the rover location, you can configure SPAN to output GPGBA messages through the OEMV3 COM1 port.

For example, to configure SPAN to receive RTCA corrections and output a 2 Hz GPGBA message containing the rover location while otherwise using default settings, send the following command:

```
gnsscardconfig card1 com1 novatel novatel on 9600 n 8 1 n off on 0.5
```



Valid values for high-rate GPGBA output are 20, 10, 5, 4 and 2 Hz.  
For logging at 1 Hz or slower, any integer value (in seconds) is accepted.

### 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 Connect* on page 52 to see the preferred method using NovAtel's *Connect* 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 receiver clock 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 center of navigation to the antenna phase center, see page 184.

The offset between the antenna phase center 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 center of navigation to antenna phase center precisely, the offset can be estimated by carrying out the Lever Arm Calibration Routine. See Section 3.4.8, *Vehicle to SPAN Frame Angular Offsets Calibration Routine* on page 63.

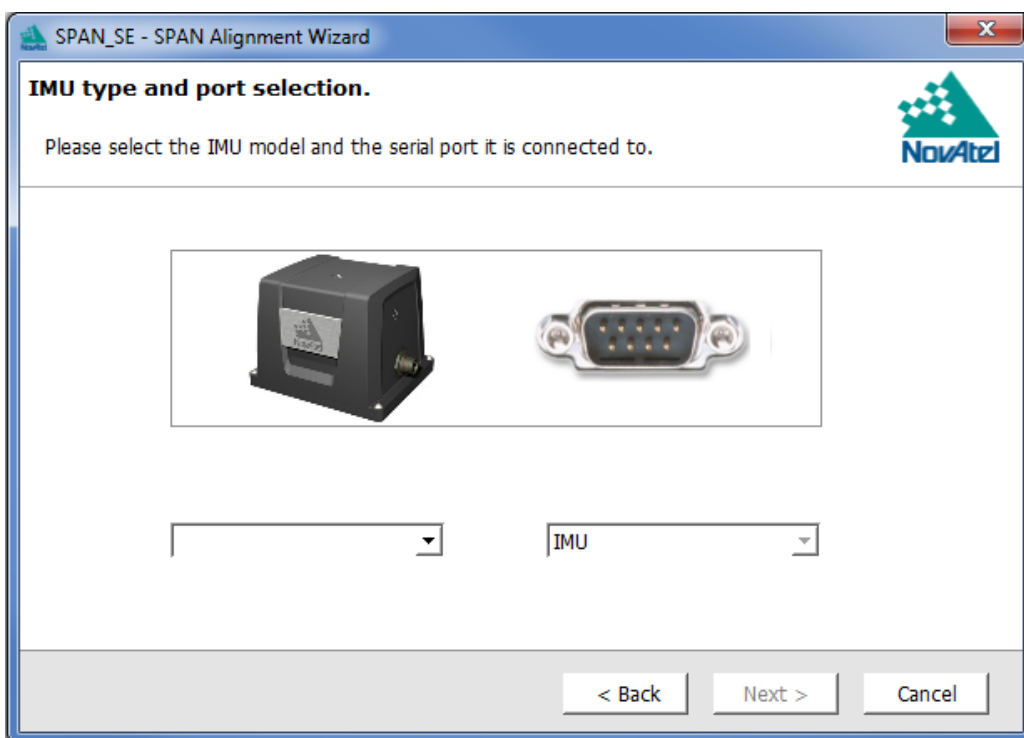
#### 3.3.2.2 SPAN Configuration with Connect

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



The NovAtel Connect screen shots in this manual may differ from your version on NovAtel Connect.

1. **SPAN basic configuration:** Select *Wizards | SPAN Alignment* from the Connect toolbar. This wizard takes you through the steps to complete a coarse, fast or aided alignment, set up a wheel sensor (if applicable), select the type of IMU and configure the receiver port connected to the IMU to accept IMU data:



## 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 *D* of this manual respectively.

Real-time operation notes:

- Inertial data does not start until FINESTEERING status is reached 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 D* 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. <sup>1</sup>
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 #7 on page 409

### 3.4.1 Configuration for Alignment



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.



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 3.4.3* on *page 55* 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 SETIMUTYPE command on *page 186*.

```
setimutype imu_ln200
```

2. **Issue the SETIMUTOANTOFFSET command** to enter the distance from the IMU to the GNSS antenna, see *page 184*.

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

### 3.4.3 System Start-Up and Alignment Techniques

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 319* and *page 327*. 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.



Depending on the alignment method, the INS Status field may go directly to INS\_SOLUTION\_GOOD.

7. The solution is refined using updates from GNSS. Once the system is operating within specifications 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\_NOT\_GOOD.

### 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 57*).

If the system is stationary for less than 1 minute, the coarse alignment finishes early, provided at least 30 stationary seconds were detected. The quality of the coarse alignment is poorer with stationary durations of less than 1 minute.

The IMU-CPT and HG1930 IMUs cannot perform coarse alignments, as these IMUs cannot accurately sense Earth's rotation. For these IMUs the default alignment routine is the Kinematic Alignment.

If a stationary alignment is required, see *Section 3.4.3.3, Manual Alignment on page 57* and *Section 3.4.3.4, Dual Antenna Alignment on page 57*.

### 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. The kinematic alignment is the default alignment routine for IMU-CPT.

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 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 199*. 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 3.4.8, Vehicle to SPAN Frame Angular Offsets Calibration Routine* starting on *page 63*.

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 than 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

If you know the attitude of your vehicle (roll, pitch, azimuth), you can manually enter the attitude information using the SETINITATTITUDE command. See *Section B.4.41, SETINITATTITUDE* Set initial attitude of SPAN in degrees on *Page 188*.

Alternatively, if you know only the azimuth of your vehicle, you can manually enter the azimuth information using the SETINITAZIMUTH command. See *Section B.4.42, SETINITAZIMUTH* Set initial azimuth and standard deviation on *Page 190*.

### 3.4.3.4 Dual Antenna Alignment

SPAN can also use information available from a NovAtel ALIGN solution to perform an alignment. See *Chapter 4, Dual-Antenna Functionality* starting on *page 78* for details.

## 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\_NOT\_GOOD. 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 <i>or</i> INSPOSS INSPVA <i>or</i> INSPVAS
Velocity	INSVEL <i>or</i> INSVELS INSPD <i>or</i> INSPDS INSPVA <i>or</i> INSPVAS
Attitude	INSATT <i>or</i> INSATTS INSPVA <i>or</i> INSPVAS
Solution Uncertainty	INSCOV <i>or</i> INSCOVs

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



The BESTPOS (GNSS only) position log can be logged at rates up to 20 Hz directly from the OEMV port. A BESTPOS (GNSS/INS) position log is available at <1Hz, 1Hz, 2Hz, 4Hz and 5Hz 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 <1Hz, 1Hz, 2Hz, 4Hz and 5Hz.



Ensure that all windows, other than the Console, are closed in Connect 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 66.

To store data directly to the internal SD Card, see *Section 3.8, SD Card* starting on page 69.

### 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 can only be used when the receiver is operating in RTK mode.



The Lever Arm Calibration Routine is not available for the IMU-CPT or the HG1930

Initial estimates and uncertainties for the lever arm are entered using the SETIMUTOANTOFFSET command, see *page 184*. 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 88*.
2. Configure the RTK corrections and make sure that the BESTGPSPOS log, see *page 229*, reports a good RTK solution.
3. Configure the IMU, see *Section 3.3.2, SPAN IMU Configuration* starting on *page 52*.
4. Enter the initial estimate for the lever arm using the SETIMUTOANTOFFSET command, see *page 184*.
5. Specify the limits of the calibration through the LEVERARMCALIBRATE command, see *page 157*. 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 54*. Another indication that the alignment is complete is the availability of INSCOV log on *page 278*.
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 237*, 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 Body Rotations

This section provides general guidance on how to extract the vehicle's attitude with respect to the local level frame.

#### 3.4.7.1 Frame Definitions

The vehicle frame is defined as x (perpendicular to the direction of travel in the horizontal plane), y (direction of travel), and z (up).

The body frame is nominally the frame as marked on the IMU enclosure. If you do not mount the IMU with the z-axis approximately up, you must check the new IMU axis orientation that SPAN automatically uses. SPAN forces z to be up in the internal computation frame. Output attitude (in INSPVA, INSATT, and so on) is with respect to that computation frame. Refer to the SETIMUORIENTATION command description to see what mapping definition applies, depending on which IMU axis most closely aligns to gravity. Essentially, this means that if you do not mount the IMU with the z-axis approximately up (as marked on the enclosure); you have a new IMU frame that defines what mapping applies. This new IMU frame will not match what is marked on the IMU enclosure and will need to be determined by checking the Full Mapping Definition table documented with the SETIMUORIENTATION command. Also, in this case, begin with the new IMU frame aligned with the vehicle frame and record your vehicle to body rotations with respect to the frame SPAN will be using as the computation frame.

The output roll is the angle of rotation about the y-axis, the output pitch is about the x-axis, and the output azimuth is about the z-axis and is measured to the y-axis. Note that azimuth is positive in the clockwise direction when looking towards the origin. However, the input vehicle to body rotation about the z-axis follows the right hand rule convention.

X (east), Y (north), and Z (up) define the local level frame.

#### 3.4.7.2 Relating the Vehicle Frame to the Body Frame:

Form the rotation matrix from the vehicle frame to the body frame using the vehicle to body frame angles measured according to the procedure described in the VEHICLETOBODYROTATION log as:

$\gamma$  is the rotation about the z-axis

$\alpha$  is the rotation about the x-axis

$\beta$  is the rotation about the y-axis

This direction cosine matrix (DCM) expresses mathematically the sequence of rotations as:

$$R_v^b = R_2(\beta)R_1(\alpha)R_3(\gamma)$$

where:

$R_3$  is a rotation around the z-axis (that is, the third axis of the x, y, and z set), as in:

$$R_3(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$R_1$  is a rotation around the x-axis, as in:

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

$R_2$  is a rotation around the y-axis, as in:

$$R_2(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}$$

These three rotations define the transformation from the vehicle frame to the body frame.

To go the other way between the frames (from the body frame to the vehicle frame), we take the transpose of this, which is:

$$R_b^v = (R_v^b)^T = R_3(\gamma)^T R_1(\alpha)^T R_2(\beta)^T = R_3(-\gamma) R_1(-\alpha) R_2(-\beta)$$

$$R_b^v = R_3(-\gamma) R_1(-\alpha) R_2(-\beta)$$

Explicitly, the elements of this matrix are:

$$R_b^v = \begin{bmatrix} \cos \gamma \cos \beta - \sin \gamma \sin \alpha \sin \beta & -\sin \gamma \cos \alpha & \cos \gamma \sin \beta + \sin \gamma \sin \alpha \cos \beta \\ \sin \gamma \cos \beta + \cos \gamma \sin \alpha \sin \beta & \cos \gamma \cos \alpha & \sin \gamma \sin \beta - \cos \gamma \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \sin \alpha & \cos \alpha \cos \beta \end{bmatrix}$$

### 3.4.7.3 Relating the Body Frame to the Local Level Frame

The body to local level matrix can be formed from the roll, pitch and azimuth (convert to yaw to form the matrix) angles found in the INSPVA (or INSATT) log, where roll is about the y-axis, pitch is about the x-axis, and azimuth/yaw is about the z-axis. The body frame (IMU axes) are nominally assumed to have Y pointing in the direction of travel, X to the right (perpendicular to the direction of travel), and

Z up. That means that azimuth references from north to the IMU y-axis, and roll is about y, and pitch is about x. Refer to *Section 3.4.7.1* on *page 60* for instructions on what to do if you are not able to mount your IMU with Z approximately up.

Using the angles in the INSPVA log, you can form the body frame to local-level frame-rotation matrix.

$\varphi$  is roll (rotation around the Y-axis)

$\theta$  is pitch (rotation around the X-axis)

$\psi$  is yaw (rotation around the Z-axis)

$$R_b^l = \begin{bmatrix} \cos \psi \cos \varphi - \sin \psi \sin \theta \sin \varphi & -\sin \psi \cos \theta & \cos \psi \sin \varphi + \sin \psi \sin \theta \cos \varphi \\ \sin \psi \cos \varphi + \cos \psi \sin \theta \sin \varphi & \cos \psi \cos \theta & \sin \psi \sin \varphi - \cos \psi \sin \theta \cos \varphi \\ -\cos \theta \sin \varphi & \sin \theta & \cos \theta \cos \varphi \end{bmatrix}$$

#### 3.4.7.4 Relating the Vehicle Frame to the Local Level Frame

Knowing the attitude of the body frame with respect to the local level frame, from the INSPVA (or INSATT) log, and the rotational relationship between the body frame and the vehicle frame, you can solve for the vehicle attitude with respect to local level. Given the two previous matrices, a new DCM can form relating the vehicle frame to the local level frame.

$$R_l^v = R_b^v R_l^b$$

$$\text{where } R_l^b = (R_b^l)^T$$

We can solve the individual angles, from the local level to vehicle frame, using the numerical values of  $R_v^l$  since we know how to form the DCM. The angles from the vehicle frame to the local level frame are denoted as  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$ . We give the matrix indices first as rows 1-3 and then columns 1-3.

$$\theta_x = \sin^{-1} [R_v^l(3,2)]$$

$$\theta_y = \tan^{-1} \left[ \frac{-R_v^l(3,1)}{R_v^l(3,3)} \right] \quad (\text{in code, use atan2 for proper quadrant resolution.})$$

$$\theta_z = \tan^{-1} \left[ \frac{-R_v^l(1,2)}{R_v^l(2,2)} \right]$$

### 3.4.8 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 199*. 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 88*.
2. Configure the RTK corrections and make sure that the BESTGPSPOS log, see *page 229*, reports a good RTK solution.
3. Configure the IMU, see *Section 3.3.2, SPAN IMU Configuration* starting on *page 52*.
4. Ensure that an accurate lever arm has been entered into the system either manually or through a lever arm calibration, see *page 59*.
5. Allow the system to complete a coarse alignment, see *page 56*. 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 54*. Another indication that the alignment is complete is the availability of INSCOV log on *page 278*.
6. Enable the vehicle to body calibration using the RVBCALIBRATE ENABLE command, see *page 174*.
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 345*, 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 56* for details on coarse and kinematic alignment.



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 55). The angular offset values are not applied to the attitude output, unless the APPLYVEHICLEBODYROTATION command is enabled, see page 131.

## 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 estimates the 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 288.

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 209, 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 95).

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 iIMU-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> 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 116.

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 209). 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 143.

### 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 using GPS only:

- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW

From a base if using GPS+GLONASS:

- GLOEPHEMERISB ONNEW
- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW

From a rover if using GPS only:

- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW
- RAWIMUSB ONNEW
- IMUTOANTOFFSETSB ONCHANGED
- HEADINGB ONNEW (Dual antenna only)

From a rover if using GPS+GLONASS:

- GLOEPHEMERISB ONNEW
- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW
- RAWIMUSB ONNEW
- IMUTOANTOFFSETSB ONCHANGED
- HEADINGB ONNEW (Dual Antenna Only)

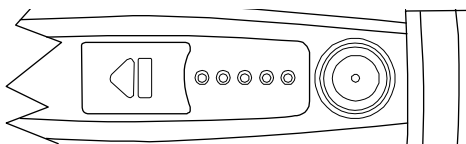
Post processing is performed through the Waypoint Inertial Explorer software package available from NovAtel's Waypoint Products Group. Visit our website at [www.novatel.com](http://www.novatel.com) for details.

## 3.7 Status Indicators

### 3.7.1 SPAN-SE Status LEDs

LED indicators on the front of the SPAN-SE, see *Figure 15* 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).

In the event of a receiver error on the SPAN-SE, the LEDs flash the binary number of the receiver error. Red is “1” and green is “0”. If you have an invalid auth code, the LEDs will be green-green-red-green-red, which is 5 in binary indicating bit 5 of the receiver error word is set. The 5-bit binary error code corresponds to the 32 possible errors defined in *Appendix D.3.59, RXSTATUS Receiver Status* starting on *page 331*.



**Figure 15: 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. Logging stops automatically (Flashing: SD card error that can occur at any time regardless of remaining space)
GPS 1 (OEMV-3)	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 (OEMV-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_NOT_GOOD status)	INS_BAD_GPS_AGREEMENT status	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 <sup>1</sup>	No power to unit	Unit powered but off and not operational (Flashing: unit powered, on, and operational)	N/A	N/A	N/A

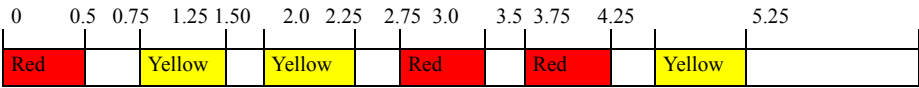
1. The power LED is on the power button, see *SPAN-SE Power Button* on *page 41*.

3.7.2 SPAN-MPPC Status LED

The diagnostic LED provided on the SPAN-MPPC is solid yellow to indicate normal operation.

Error bits and status bits that have been priority masked, as errors, will cause the LED to flash a code in a binary sequence. The binary sequence will be a 6 flash (0.5 second on and 0.25 second off per flash) sequence followed by a 1 second delay. The sequence will repeat indefinitely. If there is more than one error or status present, the lowest number will be output. The codes are ordered to have the highest priority condition output first.

The first flash in the 6 flash sequence indicates if the code that follows is an error bit or a status bit. Error bits will flash red and status bits will flash yellow. The next 5 flashes will be the binary number of the code (most significant bit first). A red flash indicates a one and a yellow flash indicates a zero. For example, for an error bit 6, the binary number is 00110 so the output sequence would be:



followed by a 1 second delay. The sequence repeats indefinitely until the receiver is reset.

In the example below, the first flash in the sequence is red, which means that a bit is set in the receiver error word. The next five flashes give a binary value of 00111. Converting this value to decimal results in a value of 7. Therefore, bit 7 of the receiver error word is set, indicating there is a problem with the supply voltage of the receiver’s power circuitry.

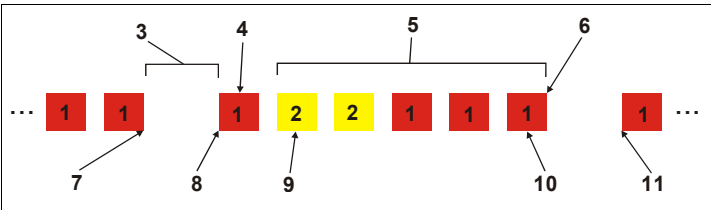


Figure 16: Status LED Flash Sequence Example

Ref	Description	Ref	Description
1	Red	7	End of previous sequence
2	Yellow	8	Beginning of sequence
3	1 second pause	9	Most significant bit of binary value
4	Word identifier flash	10	Lease significant bit of binary value
5	Bit identifier flashes	11	Start of next sequence
6	End of sequence		

## 3.8 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.



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 36.

The SD Card access door is shown closed in *Figure 17*. 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 17: 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.

The SPAN-SE does not support SD Cards formatted with partition tables. If you insert an SD Card with partition tables, the SD Card LED lights solid red indicating that the SD Card is full. If you check the contents of the SD Card using the *DIR* command or *DIRENT* log, the listing shows no files and 0 bytes free. To use the SD Card, you need to format the card using the *FORMAT* command. Note that formatting the SD Card erases all of the contents of the card.

## 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.



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 72*. 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 159*, and its *FILE* designator, to specify which logs to send to the SD Card. For example, a standard logging configuration for GPS-INS-only post-processing applications would be:

```
LOG FILE RANGECPMB ONTIME 1
LOG FILE RAWEPHEMB ONNEW
LOG FILE RAWIMUSB ONNEW
LOG FILE IMUTOANTOFFSETSB 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 163*.

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
- BESTGPSOSB ONTIME 1
- HEADINGB ONNEW
- IMUTOANTOFSSETSB ONNEW
- RANGECMPB ONTIME 1
- RAWEPHEMB ONNEW
- GLOEPHEMERISB ONNEW
- Press the SD Logging button to stop logging, or use the LOGFILE CLOSE command, see *page 163*, 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 186*, 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
```



Log requests using the ONCE trigger are not saveconfigable.

Every time the receiver powers-up, the SD Card logging configuration you specified starts. See also the SAVECONFIG command on *page 175* and the SETAUTOLOGGING command on *page 178*.



### 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.



#### SD 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.

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 IPCONFIG command.
  - Only use FTP on a secure connection as this port has no security settings at this time.



For information about setting up the Ethernet port, refer to *Section 3.12, SPAN-SE Ethernet Connection* starting on page 76. For information on connecting the Ethernet port, refer to *Section 3.12.4, Connecting to the Ethernet Port* starting on page 77. For information about the FTP functionality and Ethernet port commands, refer to *Appendix B.3.7, FTP* starting on page 128.



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

## 3.10 Adding Timed Camera Triggers

Use the SPAN EVENT IN and SPAN EVENT OUT connections to attach up to three cameras to a SPAN-SE and then trigger the cameras based upon GPS information.

You can send multiple trigger events marked with a user-specified ID and a GPS time to trigger the camera. At the requested trigger time, the SPAN outputs a user-specified LVTTTL level pulse to the sensors and waits for a response pulse to indicate a camera exposure. When this response is received, the SPAN system sends out a position log with the ID of the event.

### 3.10.1 Configuring the Hardware

A camera's trigger input connected to a valid MARK OUT and the camera exposure output connected to a valid MARK IN.

Valid MARK IN: MARK1, MARK2, MARK3, and MARK4

Valid MARK OUT: MARK1, MARK2, and MARK3

The camera can be connected to any combination of valid MARK IN and MARK OUT connections. The number of valid MARKs on the SPAN system limits the number of independently controlled cameras to three.

### 3.10.2 Configuring the Software

Use the SETUPSENSOR and TAGGEDMARKxPVA commands to setup the camera sensors on the SPAN system.

See SETUPSENSOR on page 194, TAGGEDMARKxPVA on page 340 for details.

### 3.10.3 Using Timed Event Pulses

When sensors have been connected and configured, use the TIMEEVENTPULSE command to queue events on the system. TIMEEVENTPULSE specifies the sensors that are affected, the GPS time for the event (in weeks and seconds), and an event ID. You can queue 10 unprocessed events at a time.

See TIMEEVENTPULSE on page 203 for details.

## 3.11 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.

### 3.11.1 Configuring a Synchronous Output Pulse

The EVENTOUTCONTROL command, see *page 143*, 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:

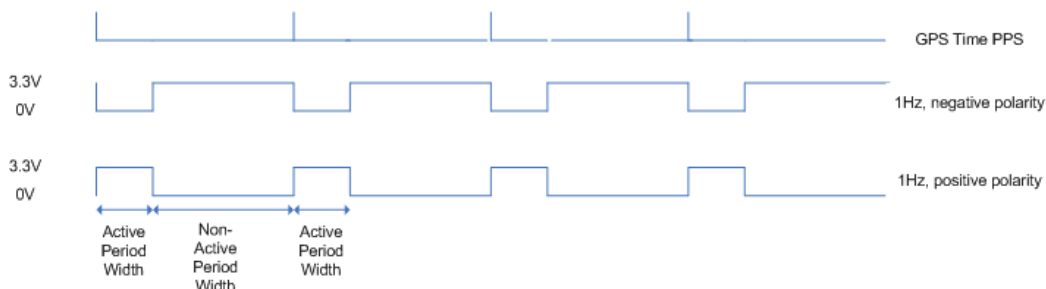
Polarity:	The polarity defines the signal state of the active portion of the signal. A positive polarity dictates that the active portion of the signal is in a high state (3.3 V).
Active Period Width:	The active period starts at the GPS time synchronized edge (rising for negative polarity and falling 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 25 ns. The maximum period is 999 999 975 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:  

$$K \text{ (active + not-active)} = 1\,000\,000\,000, \text{ where } K = 1, 2, 3, \dots, 500\,000$$



**Figure 18: Event Out**

### 3.11.2 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 142*, 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 `MARKxTIME` logs, see *page 307*. The solution synchronous with the event pulses is available from the `MARKxPVA` logs, see *page 306*. The logs required for input strobes are:

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

The above example is for the `MARK1` event input. The input signal levels are 3.3 V to 0 V. Signal voltages outside these bounds damage the receiver. The minimum detectable pulse duration must be greater than or equal to 1 microsecond.

### 3.11.2.1 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 MARKxCOUNT logs, see *page 305*. For example, the following gives the total pulses on event strobe 1 every second:

```
LOG MARK1COUNTA ONNEW
```

## 3.12 SPAN-SE Ethernet Connection

The SPAN receiver has four Ethernet ports. The device has a unique Media Access Control (MAC) address, hard coded into flash, and user-configurable IP information. There are four ports available for Ethernet: Ports 3000, 3001, 3002 and 3003 are used to access ICOM1, ICOM2, ICOM3 and ICOM4 respectively. Each port 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

The MAC address is available to the user through the MAC log.

### 3.12.1 Configuring for TCP or UDP Operation

Each SPAN-SE Ethernet port can be configured for either TCP or UDP. The default configuration of the Ethernet ports is for TCP operation. To configure the Ethernet ports, see the ICOMCONFIG command on *page 151*.

### 3.12.2 Configuring the Ethernet Connection Settings

Use the IPCONFIG command, see *page 153*, to set the static IP Address, the subnet mask and the gateway. An example of the IPCONFIG command is:

```
IPCONFIG 10.1.100.25 255.255.255.0 10.1.100.1
```

### 3.12.3 Configuring Log Requests Destined for the Ethernet Port.

The COM port identifier for the Ethernet port is ICOM1 in ASCII or 23 in binary. A sample log request for the Ethernet port is:

```
LOG ICOM1 RANGECPMB ONTIME 1
```

### 3.12.4 Connecting to the Ethernet Port

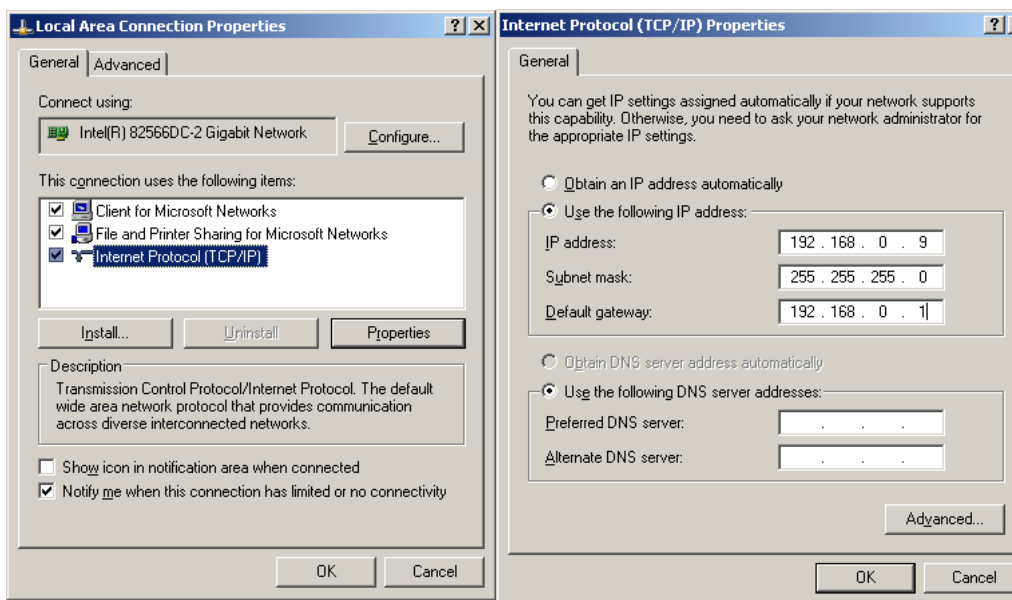
If the port is configured in TCP mode, only one connection to the receiver is allowed at a time. Data automatically streams to the IP address that connects to the port.

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 128*.

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

1. Connect the computer's Ethernet port to the SPAN-SE Ethernet port using a shielded Ethernet cross-over cable.
2. Set the static IP address on the computer to the following settings in the Local Area Connection Properties dialog box:



The last octet of the IP address can be any number from 1 to 255 inclusive except for 10, which is the last value in the SPAN-SE default IP address. In the above example, we have used 9 for the last octet.

This chapter describes the SPAN-SE dual-antenna functionality, with references to the user commands and logs that have been added for this feature. In this chapter, “SPAN-SE-D” refers to SPAN-SE with dual-antenna functionality.



Not all SPAN-SE enclosures have the hardware necessary to perform dual antenna operations. The dual antenna functionality requires both extra HW and SW from the base model SPAN-SE.

## 4.1 Overview

NovAtel ALIGN heading technology generates distance and bearing information between a master and one or more rover receivers. This feature is ideal for customers who want relative directional heading, separation heading between two objects or heading information with moving base and pointing applications. Heading applications can be applied over various markets, including machine control, unmanned vehicles, marine and agricultural markets.

SPAN-SE-D and SPAN-MPPC-D provide the hardware necessary to run an ALIGN baseline inside one enclosure. However, SPAN-MPPC-D must be paired with the OEMV3 receiver and an OEMV-based remote to run ALIGN.

Two antennas can be connected to the SPAN-SE-D to compute a GNSS heading solution. From any of the SPAN-SE-D or SPAN-MPPC-D COM ports, the ALIGN baseline solution can be logged along with the standard OEMV logs.

If the SPAN-SE-D or SPAN-MPPC-D is configured as a SPAN-capable model, the ALIGN GNSS baseline can also be used to assist the initial alignment of the SPAN solution. In addition, the ALIGN baseline solution will aide the heading solution from SPAN if the heading drifts due to slow or constant dynamics.

For GPS+GLONASS+SBAS models, ALIGN is capable of a 10 Hz heading output rate. A GPS-only model is also available for dual frequency receivers.

## 4.2 Installation

The hardware for the dual-antenna functionality is installed in a manner similar to that for other SPAN-SE-D and SPAN-MPPC-D systems. Consider the following during the installation:

1. Install the IMU and the two antennas to the vehicle so that the relative distance between them is fixed.
2. The antennas should be mounted where the view of the satellites will not be obstructed by any part of the vehicle.
3. If you are running SPAN with an IMU and ALIGN, the lever arms or distance from the IMU's Center of Navigation to the antenna phase center needs to be accurately measured using the coordinate axes defined on the outside of the IMU (IMU enclosure frame). The ALIGN baseline does NOT need to be aligned with the vehicle axes or with the axes of the IMU.

## 4.3 Configuring ALIGN with SPAN-SE-D

The SPAN-SE-D hardware is built into a single enclosure, so there is no hardware setup required other than having two GNSS antennas connected to the GPS1 and GPS2 connectors on the SPAN-SE.

SPAN-SE-D are pre-configured to compute the GNSS baseline ALIGN solution. No specific configuration commands need to be sent to the receiver to start the ALIGN processing. When the antennas are connected properly to the GPS1 and GPS2 connectors on the SPAN-SE the dual antenna ALIGN solution will start automatically on boot-up.

As with all ALIGN capable products, the baseline solution is available from the GPHDT and HEADING logs.

## 4.4 Configuring ALIGN with SPAN-MPPC-D

ALIGN on SPAN-MPPC-D requires the use of an external NovAtel receiver, typically an OEMV2. The external receiver must be an ALIGN-capable model.

Connect the SPAN-MPPC-D to the external receiver with a null serial cable between COM2 on the external receiver and any SPAN COM port (COM2 by default). To select a different SPAN COM port, use the DUALANTENNAPORTCONFIG command. See *Section B.4.8, DUALANTENNAPORTCONFIG SPAN-MPPC Dual Antenna Port Setup on page 141*.

Once the two receivers are connected, SPAN will automatically configure ALIGN communication. When the antennas are properly connected to each receiver, the dual antenna ALIGN solution will start automatically.

As with all ALIGN capable products, the baseline solution is available from the GPHDT and HEADING logs.

## 4.5 Configuring SPAN with ALIGN on SE-D and MPPC-D Models

To enable the dual-antenna ALIGN solution to aid the INS alignment and provide heading updates, the offset between the antennas and the IMU must be known. This is achieved by entering lever arms to both antennas. This is done via the SETIMUTOANTOFFSET and SETIMUTOANTOFFSET2 commands.

To configure SPAN with ALIGN Aiding:

1. Enter the lever arm from the IMU to the primary antenna (primary antenna is connected to the connector labeled *GPS 1*) using the SETIMUTOANTOFFSET command.

### Abbreviated ASCII Example:

```
SETIMUTOANTOFFSET 0.54 0.32 1.20 0.03 0.03 0.05
```

2. Enter the lever arm from the IMU to the secondary antenna (secondary antenna is connected to the connector labeled *GPS 2* on SPAN-SE, or the external receiver for SPAN-MPPC.) using the SETIMUTOANTOFFSET2 command.

#### Abbreviated ASCII Example:

```
SETIMUTOANTOFFSET2 0.54 0.32 1.20 0.03 0.03 0.05
```

SPAN-SE-D and SPAN-MPPC can be configured for different alignment routines depending on the motion conditions experienced during the alignment period. For example, in marine applications, the vehicle cannot be guaranteed to be stationary during the alignment process, so the standard coarse alignment routine will not work.

The different alignment routines are described in the following sections.

### 4.5.1 Alignment on a Moving Vessel - Aided Transfer Alignment (Default)

If your vehicle is not stationary during the alignment, such as on a ship, the Aided Transfer Alignment is the alignment routine to use. This is the default alignment for SPAN-SE-D so the configuration is relatively simple. This alignment method uses the ALIGN baseline solution to perform an instantaneous alignment of the vehicle attitude.

No additional configuration is required for this alignment routine. The alignment will happen instantaneously once the receiver establishes communication with the IMU and computes a fixed integer, verified ALIGN solution. The INS status will remain in INS\_ALIGNING mode until the aided transfer is complete.

### 4.5.2 Alignment on a Stationary Vehicle - Aided Static Alignment

An alternative to the instant transfer alignment, the ALIGN heading can be used as a seed for a coarse static alignment. In this mode, the standard coarse alignment routine will run given the initial azimuth value. As with the transfer alignment, the first verified fixed RTK solution will be used to provide the alignment seed after which the 60s coarse alignment will begin.



NovAtel recommends the transfer alignment for the dual-antenna solution.

To use this alignment mode, the configuration command ALIGNMENTMODE must be sent to the receiver.

```
ALIGNMENTMODE AIDED_STATIC
```

### 4.5.3 Unaided Alignment

This returns the SPAN system to its default alignment options: a coarse static alignment, kinematic alignment, or manual alignment. See *Section 3.4.1, Configuration for Alignment on page 55* for details on standard alignment modes.



This alignment mode will be default for any single antenna, or ‘S’ model. When using a dual-antenna ‘D’ model, the ALIGNMENTMODE command must be sent to the receiver.

ALIGNMENTMODE UNAIDED



A static coarse alignment is not available for the IMU-CPT or the HG1930

## 4.6 SPAN ALIGN Attitude Updates

The INS heading updates are used as a way to help constrain the azimuth drift of the INS solution whenever possible. This is of greatest value with lower quality IMUs and in environments with poor ability to observe Earth rotation. Slow-moving marine, or train applications are good examples of the intended use. By providing an external heading source, the solution drift can be constrained in these environments.

This chapter describes basic API functionality in SPAN-SE. It does not provide detailed usage information or information required for developers.

## 5.1 Overview

To enable SPAN API and enable SPAN to execute user applications, the receiver must have a correctly configured model for the SPAN firmware. Use the LOG VERSION command determine if the correct SPAN API model is configured. SPANCARD displays the model string immediately after the SPANCARD string. In addition to other model options, the API model must include an A character to enable SPAN API. If the API model is not correct, contact NovAtel Customer Service to obtain an authorization code that includes the SPAN API model option.

When installing new SPAN firmware, WinLoad prompts you for an authorization code. After you configure the SPAN firmware authorization code, run the LOG VERSION command to verify that the model options include the required A character.

## 5.2 Loading User Applications

NovAtel uses a proprietary tool called WinLoad to load SPAN user applications. You can get WinLoad from NovAtel through customer support, or through the NovAtel FTP site at: <ftp://ftp.novatel.ca/outgoing/support/Software/Winload>.

Read the following instructions before using WinLoad to add software to the SPAN-SE: <http://www.novatel.com/assets/Documents/Bulletins/apn052.pdf>.

In cases where WinLoad is not practical, you can also use SoftLoad to add user applications. SoftLoad reprograms the SPAN with a hex file that is stored on an SD card, USB stick or RAM Drive.

### Syntax:

```
SOFTLOADFILE <MassStorageEnum> <FileName> <Destination>
```

### Example:

```
SOFTLOADFILE SD 3800.hex OEMV3
```

In the example, softload reprograms the OEMV3 card with 3800.hex from an SD card.



The SOFTLOADFILE command does not display a success or failure notice until the programming is complete. This process can take several minutes.

## 5.2.1 Verifying Loaded User Applications

To verify that a user application has been loaded, use the APPLICATIONSTATUS log. The log's output contains relevant information about user applications in the three available slots. Below is an abbreviated ASCII example:

```
[COM3]log applicationstatus
<OK
[COM3]<APPLICATIONSTATUS COM3 0 99.0 UNKNOWN 0 0.000 404c0040 0000 326
< 3
< 9 FALSE 00000000 00000000 "UserApp1" "1.000A3" "2010/Dec/20" "11:26:51"
< 9 FALSE 00000000 00000000 "UserApp2" "1.000A1" "2010/Dec/24" "09:09:09"
< 9 FALSE 00000000 00000000 "UserApp2" "1.000A1" "2010/Dec/24" "09:09:09"
```

In the example above, UserApp1 appears in slot1, and UserApp2 appears in slots 2 and 3.

## 5.3 Starting, Stopping and Removing a User Application

### 5.3.1 Starting a User Application

To start a user application, use the APPLICATION command. An abbreviated ASCII example of starting the application loaded in the second slot is:

```
application userapp2 start
```

The application named userapp2 opens. If queried, the APPLICATIONSTATUS log will reflect the fact that userapp2 is running. For example:

```
[COM3]application userapp2 start

<OK
[COM3]log applicationstatus
[COM3]<APPLICATIONSTATUS COM3 0 99.0 FINESTEERING 1615 239395.238 40000040
0000 326
< 3
< 9 FALSE 00000000 00000000 "NovApp1" "1.000A3" "2010/Dec/20" "11:26:51"
< 9 TRUE 00000000 00000000 "NovApp2" "1.000A1" "2010/Dec/24" "09:09:09"
< 9 FALSE 00000000 00000000 "NovApp2" "1.000A1" "2010/Dec/24" "09:09:09"
[COM3]
```

### 5.3.2 Stopping a User Application

When a user application is opened, it continues to execute until the SPAN receiver is powered down. The SPAN-SE is a specialized system and does not allow for de-allocation of resources like a typical desktop computer.

---

### 5.3.3 **Removing a User Application**

To delete a user application, use APPLICATION <userapp> REMOVE command. For example:

```
APPLICATION USERAPP2 REMOVE
```

Alternatively, you can use the application loading procedure on page 82 to overwrite a user application in a slot with a newer user application.

## 6.1 Overview

The variable lever arm concept arose to support applications in which the IMU is no longer rigidly fixed to the vehicle, but rather on a gimballed mount. This creates an issue where the input lever arm offsets to the GNSS antenna are no longer correct, because the IMU can rotate on its mount, while the antenna remains fixed.

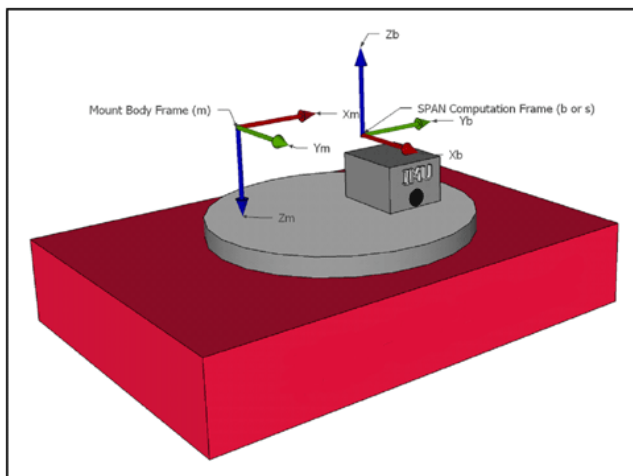
The use of the variable lever arm functionality requires that the device to which the IMU is attached be able to send its gimbal rotation angles back to SPAN. These angles are used to re-calculate the lever arm at the rate that they are received. SPAN will also be able to output a gimballed solution at the rate the gimbal angles are received.

## 6.2 Technical Description

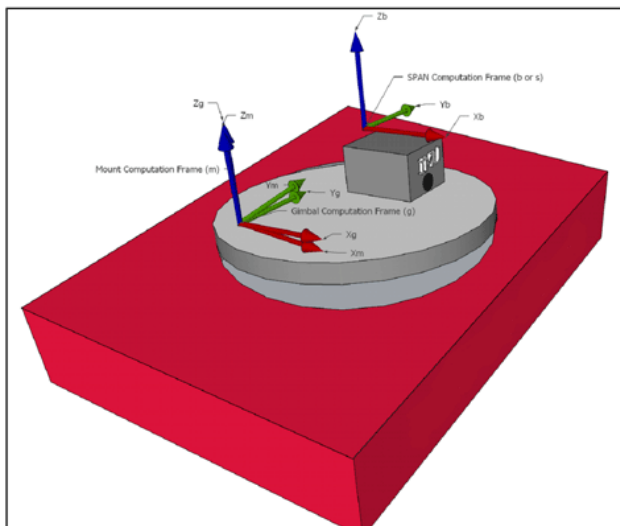
There are several frames of reference involved when dealing with a gimballed mount. The frames are all very similar, but can be quite confusing due to small differences. Below are all frames applicable to the implementation of the variable lever arm:

Frame	Description	Notation
SPAN body frame	The physical IMU axes	(b)
SPAN computational frame	Standard SPAN computational frame where Z is up, Y is forward and X is right	(b)
Vehicle frame	SPAN computational frame may be rotated to match the vehicle frame using the VEHICLEBODYROTATION command	(v)
Mount body frame	The physical axes of the gimballed mount	(mb)
Gimbal body frame	The physical axes of the gimbal plane that matches the mount body frame when gimbal angles are zero	(gb)
Mount computation frame	Mapping 5 applied to the mount body frame	(mc)
Gimbal computation frame	Mapping 5 applied to the gimbal body frame	(gc)
Mount SPAN frame	Mapping five and GIMBALSPANROTATION applied to the mount frame so that the mount and SPAN frames are coincident when gimbal angles are zero	(m)
Gimbal SPAN frame	Mapping 5 and GIMBALSPANROTATION applied to the gimbal frame so that the gimbal and SPAN frames are coincident	(g)

The next two figures illustrate a basic scenario for the information in the table above. The first figure shows a possible configuration for the mount body frame and SPAN frame. The second figure shows the desired result of all rotations: all three frames have the same axis definitions.



**Figure 19: Simple Configuration**



**Figure 20: Rotation Results**

## 6.3 Usage

The following are basic commands that you can use to affect the variable lever arm:

- SETGIMBALORIENTATION rotates the standard gimbal axis definition to the standard SPAN definition (see SETIMUORIENTATION on page 181 for more information)
- GIMBALSPANROTATION defines the rotation from the gimbal mount computation frame (mc) to the span computation frame (b). This applies if the axes of the IMU and gimbal mount are not the same after SETIMUORIENTATION and SETGIMBALORIENTATION commands are applied
- VEHICLEBODYROTATION defines the rotation from the vehicle frame (v) to the SPAN computation frame (b). See 3.1.4, Vehicle Frame on Page 46 for more information.

This appendix details the technical specifications of the IMUs and the SPAN-SE receiver.

## 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 $\Omega$ 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 System Power Consumption Single Antenna: 10 W (typical) Dual Antenna: 12 W (typical)
USB Host	USB-A
USB Device	USB-B
Ethernet	RJ-45 Ethernet
I/O 1 (Green)	ODU Mini Snap, Series K, 30-pin connector, see <i>Table 8 on page 94</i>
I/O 2 (Yellow)	ODU Mini Snap, Series K, 30-pin connector, see <i>Table 9 on page 96</i>
NOVATEL PART NUMBER	
SPAN-SE	01018071
PHYSICAL	
Size	247 x 199 x 76 mm
Weight	3.4 kg maximum
ENVIRONMENTAL	
Operating Temperature	-40°C to +65°C
Storage Temperature	-50°C to +95°C
Humidity	Not to exceed 95% non-condensing

*Continued on the following page*

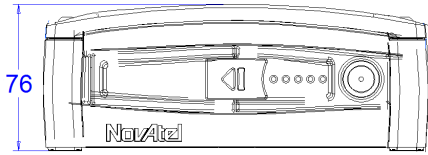


---

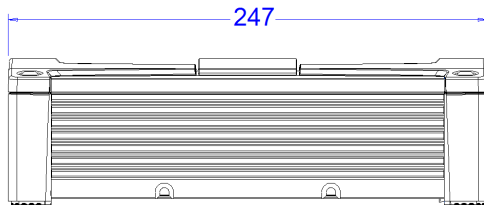
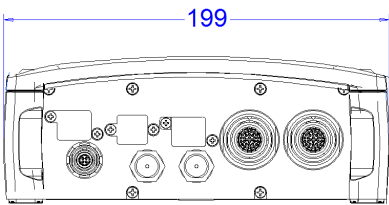
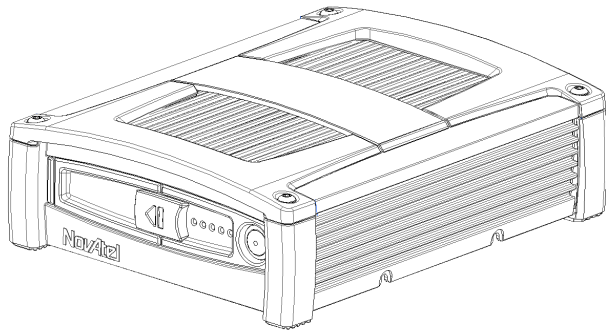
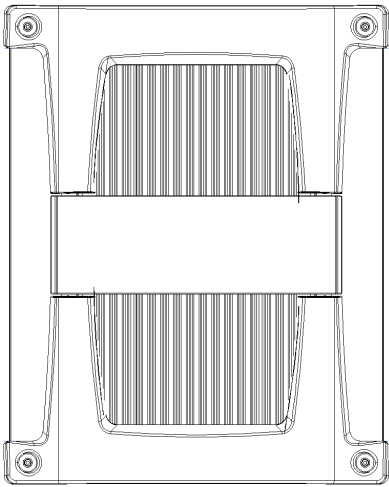
## ***ENVIRONMENTAL***

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

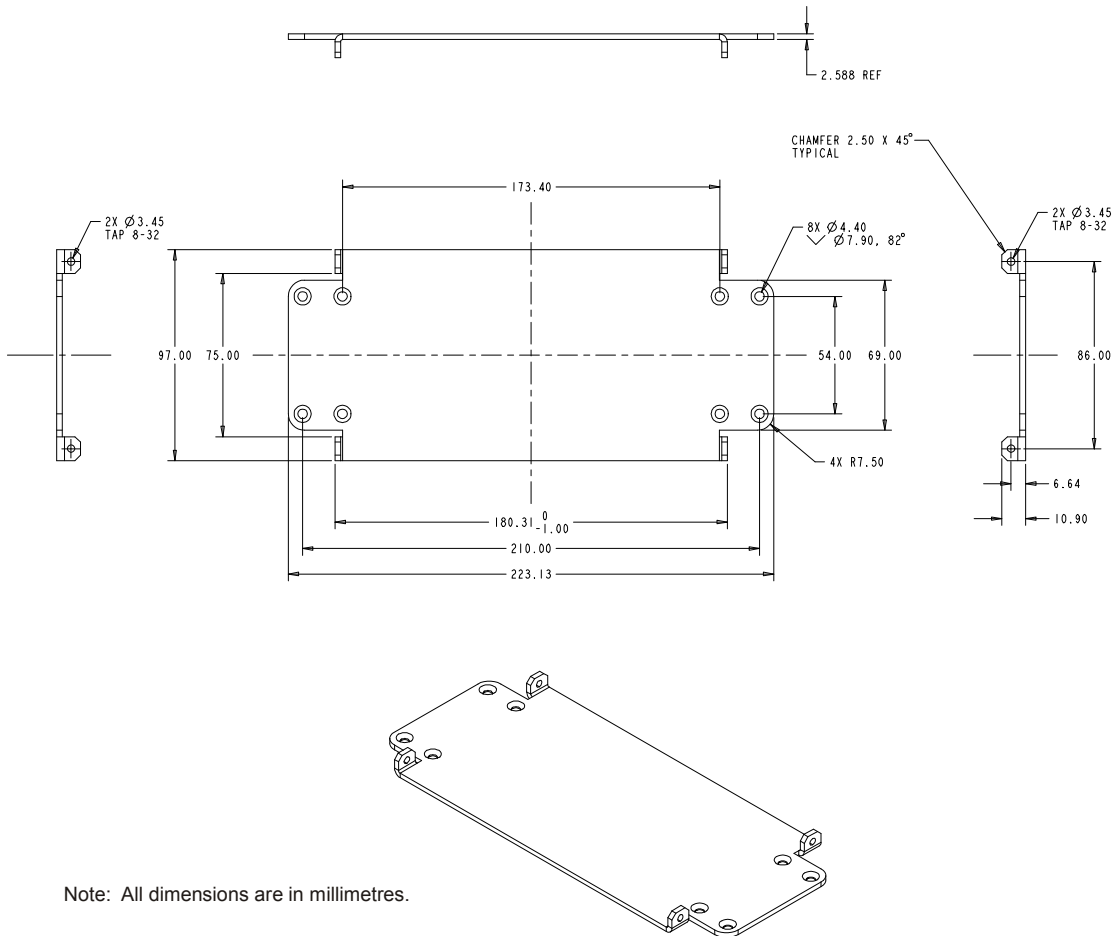
## DIMENSIONS



Note: All dimensions are in millimetres.



## SPAN-SE Mounting Plate Dimensions:



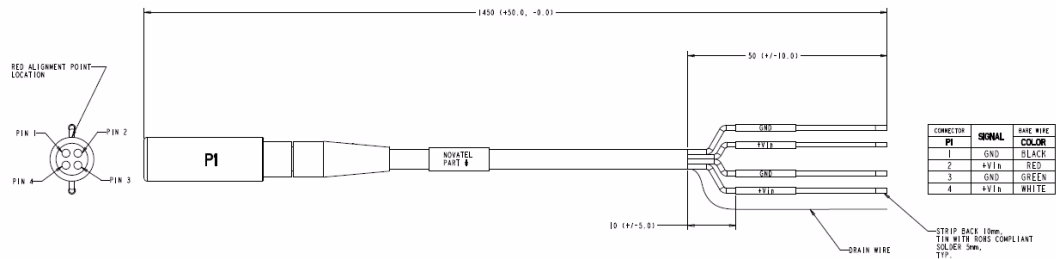
**A.1.1.1 Power Adapter Cable (NovAtel part number 01018135S)**

The power adapter cable supplied with the SPAN-SE, see *Figure 21*, 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 40*.



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

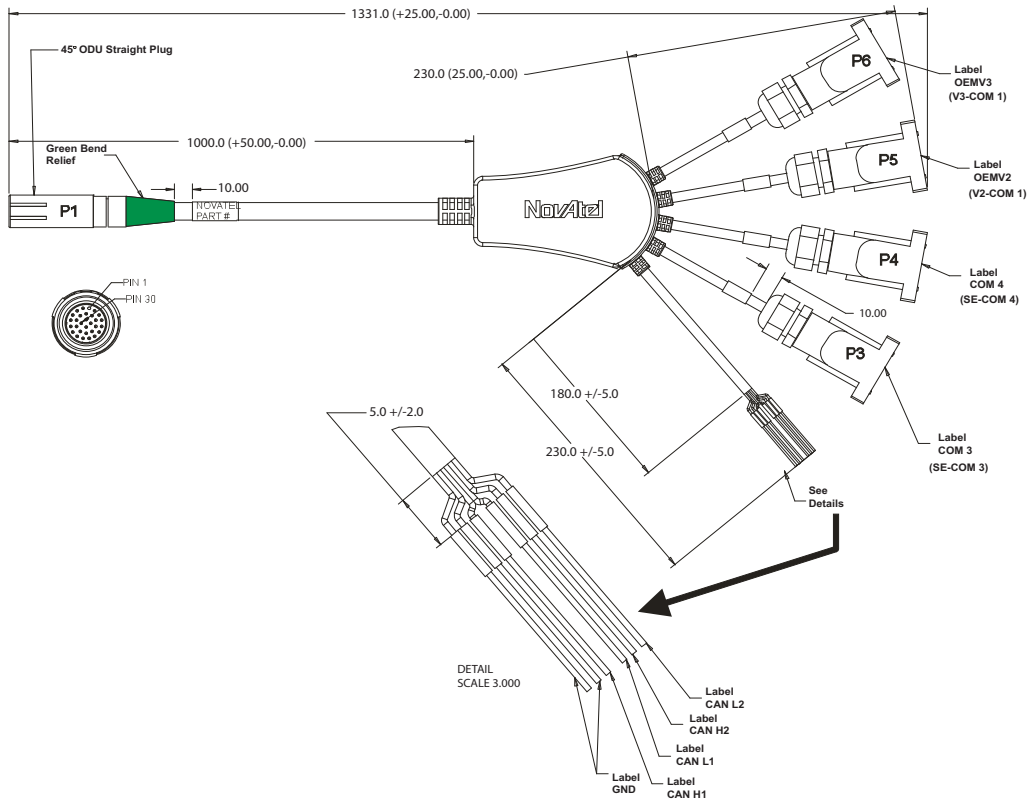


**Figure 21: SPAN-SE Power Cable**

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

This cable, supplied with the SPAN-SE, see *Figure 22*, 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 94.

This cable is RoHS compliant.



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



**Figure 22: SPAN-SE I/O 1 Green Cable**

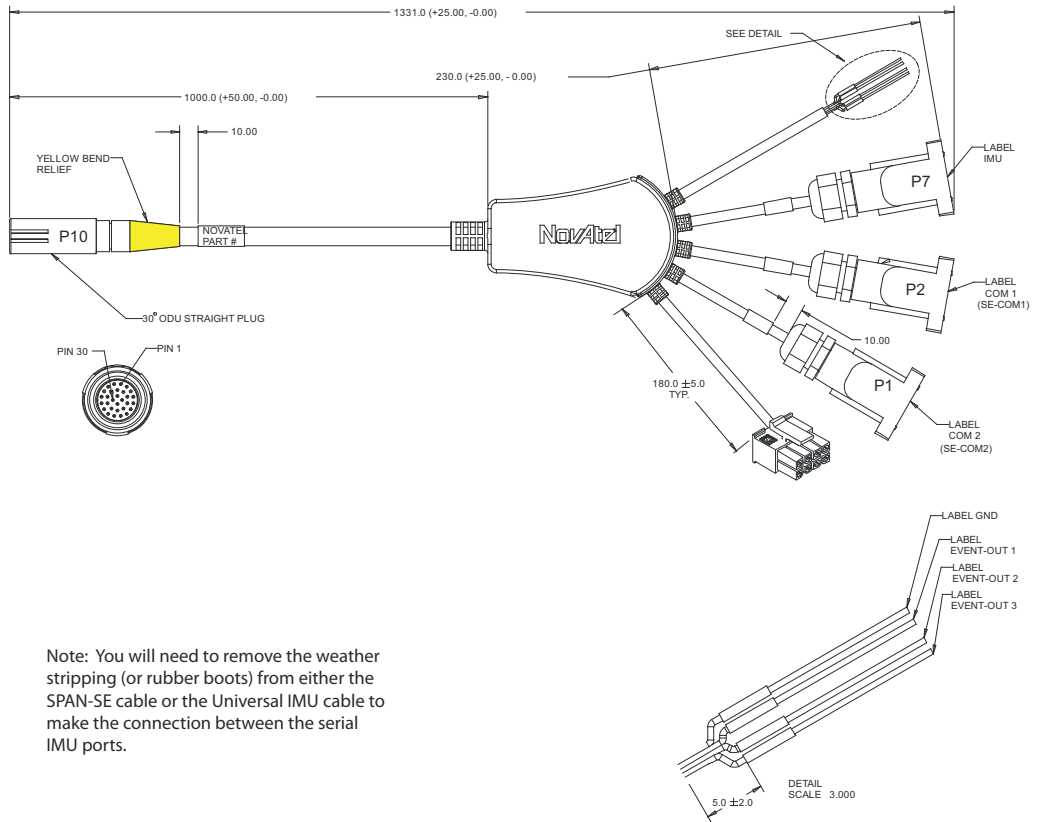
**Table 8: I/O 1 Green Cable Connector Pin-Outs**

<b>P1 <sup>a</sup></b>		<b>Remote Connectors</b>	
<b>Pin #</b>	<b>Function</b>	<b>Connector</b>	<b>Pin # <sup>a</sup></b>
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 22 on page 93*

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

This cable, supplied with the SPAN-SE, see *Figure 23*, 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 96. This cable is RoHS compliant.



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



**Figure 23: SPAN-SE I/O 2 Yellow Cable**

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

<b>P1 <sup>a</sup></b>		<b>Remote Connectors</b>	
<b>Pin #</b>	<b>Function</b>	<b>Connector</b>	<b>Pin # <sup>a</sup></b>
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	MOLEX	1
5	EVENT-IN 2	MOLEX	2
20	EVENT-IN 3	MOLEX	3
19	EVENT-IN 4	MOLEX	4
28	GND	MOLEX	5 (GND1), 6 (GND2)
29	GND	MOLEX	7 (GND3), 8 (GND4)
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 23* on *page 95*



## A.2 Inertial Measurement Units (IMUs)

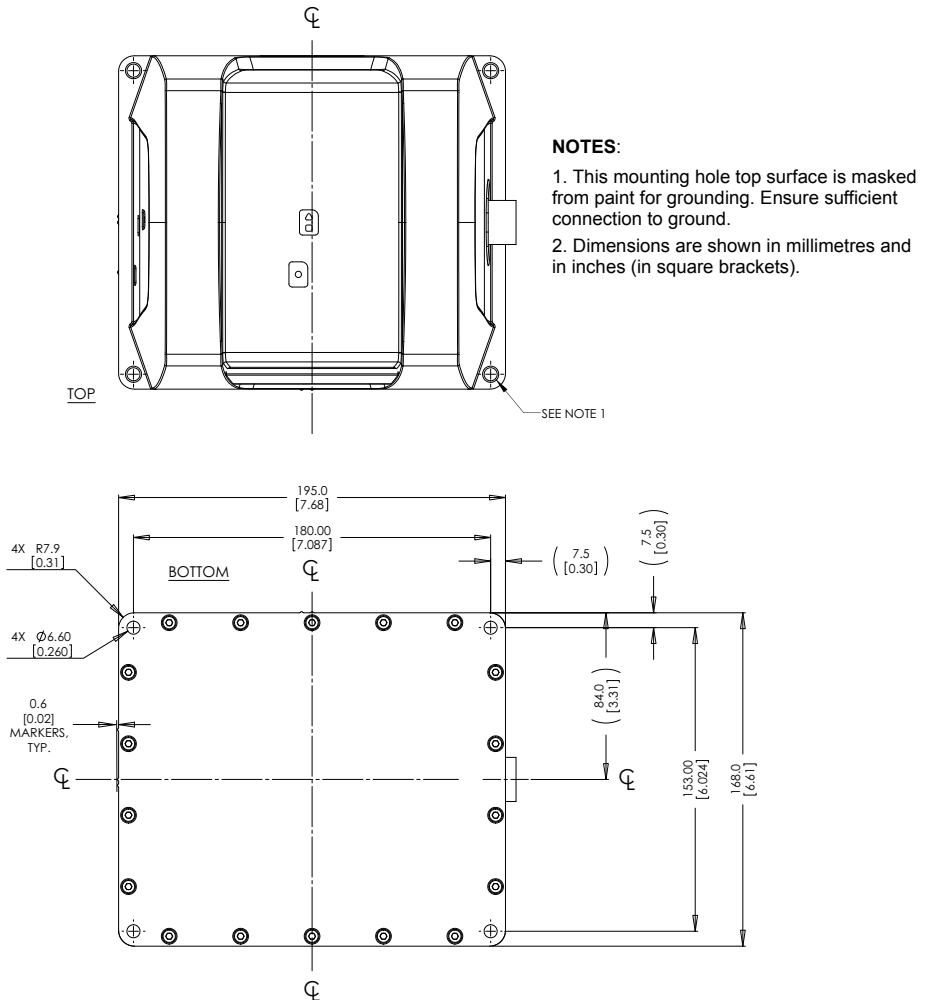
### A.2.1 Universal IMU Enclosure

The Universal IMU Enclosure is available with the LCI-1, HG1700 and LN200 IMU's.

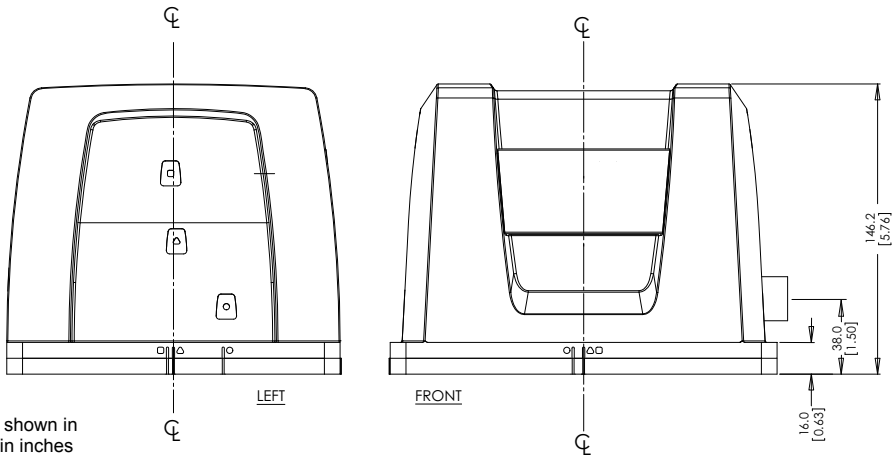
**Table 10: Universal IMU Enclosure Specifications**

PHYSICAL	
IMU Size	168 mm x 195 mm x 146 mm
IMU Weight	4.25 kg

#### MECHANICAL DRAWINGS






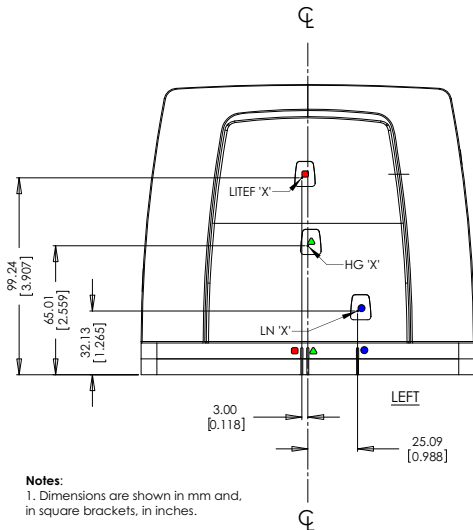
**Figure 24: Universal IMU Enclosure Top/Bottom Dimensions**



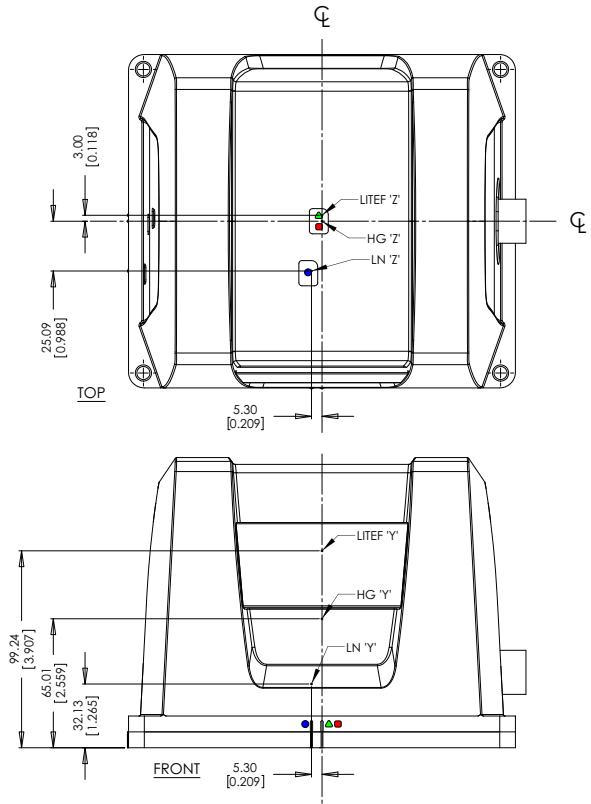
Dimensions are shown in millimetres and in inches (in square brackets).

**Figure 25: Universal IMU Enclosure Side Dimensions**

Enclosure Icon	IMU Type
	LCI-1
	HG1700
	LN200



**Notes:**  
1. Dimensions are shown in mm and, in square brackets, in inches.



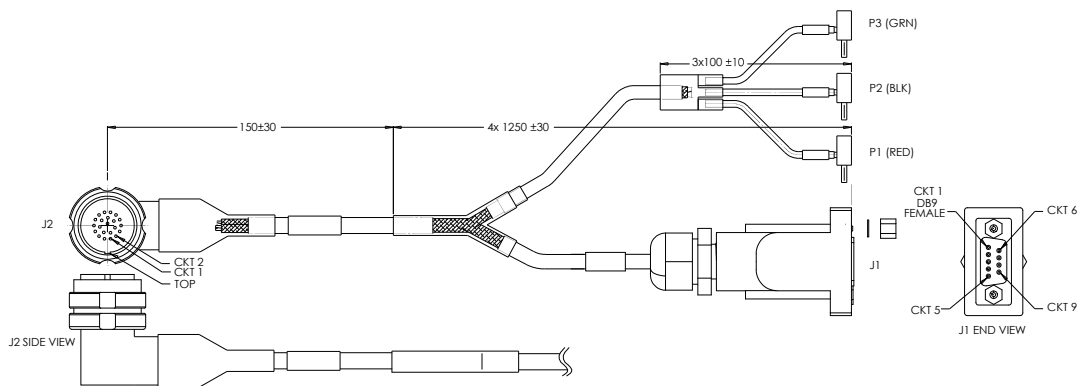
**Figure 26: IMU Center of Navigation**

### A.2.1.1 Universal IMU Enclosure Interface Cable

NovAtel's part number for the Universal IMU Enclosure interface cable is 01018299 (see *Figure 27*). This cable provides power to the IMU from an external power source, and enables input and output between the receiver and the IMU.



**Figure 27: Universal IMU Enclosure Interface Cable**



**Figure 28: Universal IMU Enclosure Interface Cable**

**Table 11: Universal IMU Enclosure Interface Cable Pinouts**

J2		REMOTE CONNECTION	
PINOUT	FUNCTION	CONNECTOR	PIN (COLOR)
1	V <sub>IN</sub> (-) PIN 1	P2	(BLK)
22	V <sub>IN</sub> (-) PIN 22		(GRN)
2	Not used		
3	V <sub>IN</sub> (+) PIN 3	P1	(RED)
21	V <sub>IN</sub> (+) PIN 21		(WHT)
4	Not used		
5	Not used		
6	Not used		
7	DAS (+)	J1	1
8	Not used		
9	DAS GND (-)	J1	5
10	Not used		
11	OEM_CTS <sup>a</sup> /Rx-	J1	8
12	OEM_Rx/Rx+	J1	2
13	Not used		
14	DGND	J1	5
15	DGND	J1	5
16	Not used		
17	Not used		
18	Not used		
19	OEM_Tx/Tx+	J1	3
20	OEM_RTS/Tx-	J1	7
	Shield	P3	(GRN)

a. OEM stands for Original Equipment Manufacturer.

### A.2.1.2 IMU Performance

<b>PERFORMANCE (IMU)</b>		
HG1700-AG58	Gyro Input Range	$\pm 1000$ deg/sec
	Gyro Rate Bias	1.0 deg/hr
	Gyro Rate Scale Factor	150 ppm
	Angular Random Walk	0.125 deg/rt-hr
	Accelerometer Range	$\pm 50$ g
	Accelerometer Linearity	500 ppm
	Accelerometer Scale Factor	300 ppm
	Accelerometer Bias	1.0 mg
HG1700-AG62	Gyro Input Range	$\pm 1000$ deg/sec
	Gyro Rate Bias	5.0 deg/hr
	Gyro Rate Scale Factor	150 ppm
	Angular Random Walk	0.5 deg/rt-hr
	Accelerometer Range	$\pm 50$ g
	Accelerometer Linearity	500 ppm
	Accelerometer Scale Factor	300 ppm
	Accelerometer Bias	3.0 mg
LN-200	Gyro Input Range	$\pm 1000$ deg/sec
	Gyro Rate Bias	1.0 deg/hr
	Gyro Rate Scale Factor	100 ppm
	Angular Random Walk	0.07 deg/rt-hr
	Accelerometer Range	$\pm 40$ g
	Accelerometer Linearity	-
	Accelerometer Scale Factor	300 ppm
	Accelerometer Bias	0.3 mg
LCI-1	Gyro Input Range	$\pm 800$ deg/sec
	Gyro Rate Bias	$< 1.0$ deg/hr
	Gyro Rate Scale Factor	$< 500$ ppm
	Angular Random Walk	$< 0.15$ deg/rt-hr
	Accelerometer Range	$\pm 40$ g
	Accelerometer Linearity	-
	Accelerometer Scale Factor	$< 1000$ ppm
	Accelerometer Bias	$< 1.0$ mg

**Table 12: Universal IMU Enclosure Electrical and Environmental**

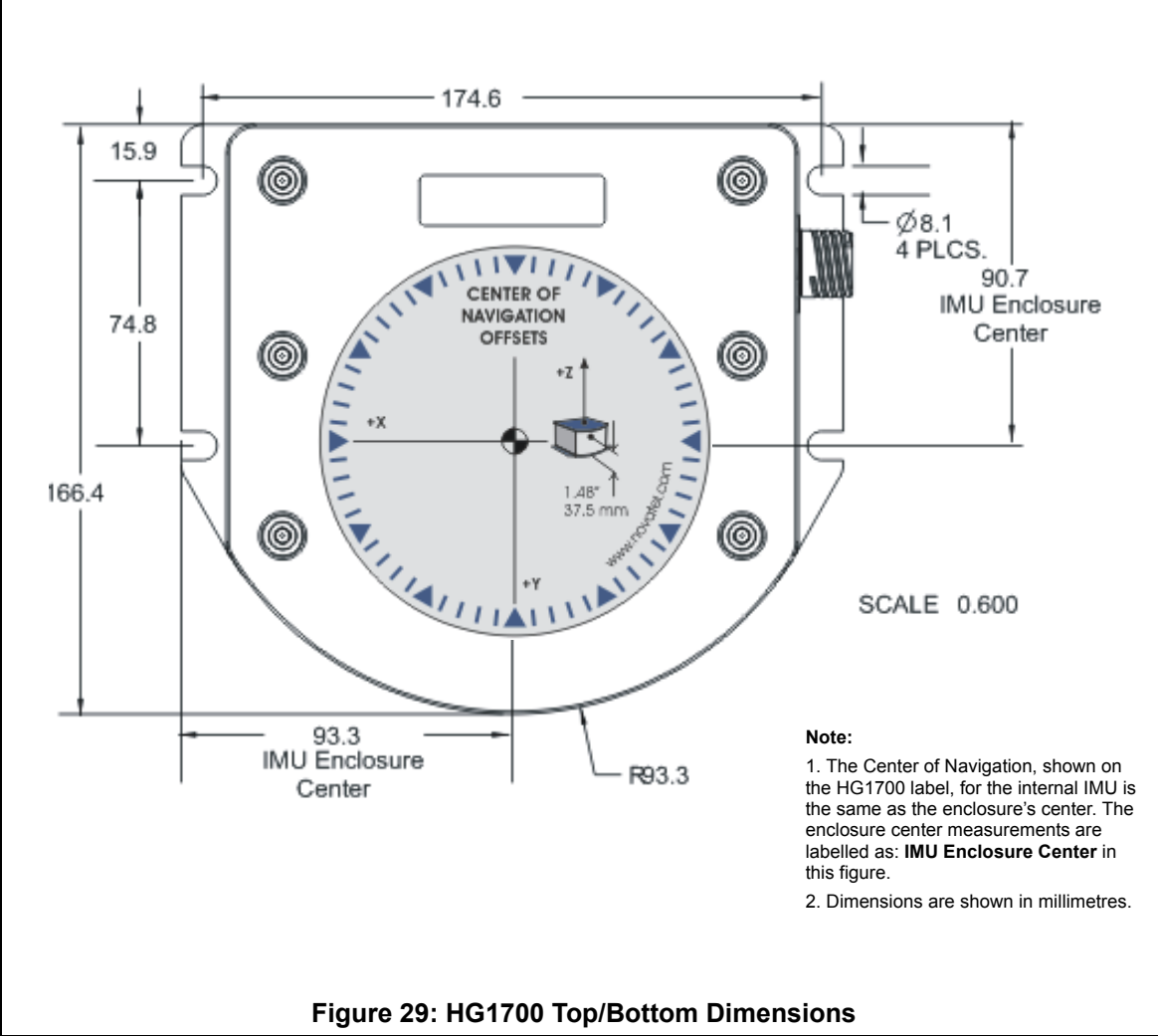
<b><i>ELECTRICAL</i></b>	
IMU Power Consumption	HG1700-AG58: 9 W (max) HG1700-AG62: 8 W (max) LN-200: 16 W (max) LCI-1: 16 W (typical)
IMU Input Voltage	+12 to +28 V DC (all IMU's)
System Power Consumption	14.8 W (typical, for all IMU's)
Input/Output Connectors	MIL-C-38999-III, 22 pin (all IMU's)
IMU Interface	RS-232 or RS-422
<b><i>ENVIRONMENTAL</i></b>	
Temperature	HG1700-AG58, HG1700-AG62
	Operating        -30°C to +60°C
	Storage         -45°C to +71°C
	LN200
	Operating        -30°C to +60°C
	Storage         -45°C to +80°C
	LCI-1
	Operating        -40°C to +60°C
	Storage         -40°C to +71°C
Humidity	Operates at 95% RH, non-condensing (all IMU's)

A.2.2 HG1700 IMU (Single-Connector Enclosure)

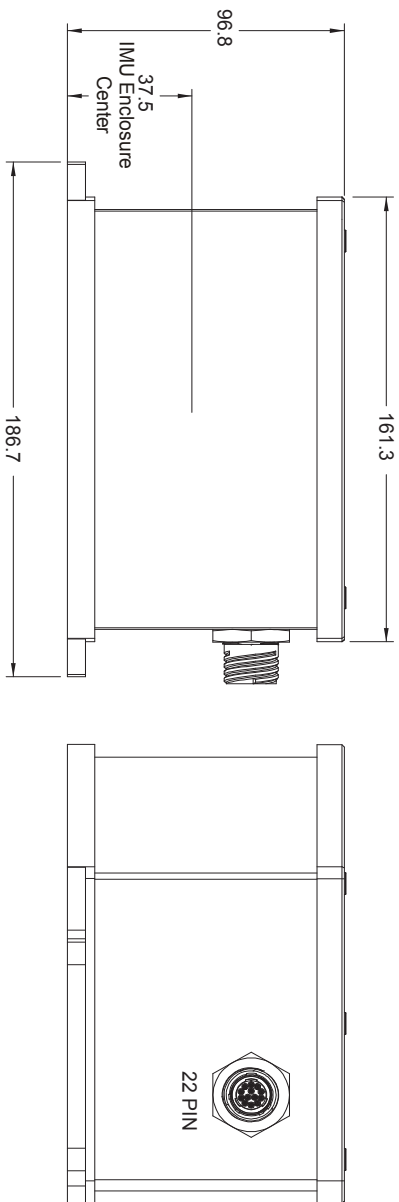
Table 13: HG1700 IMU Specifications

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:**

1. 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.
2. Dimensions are shown in millimetres.

**Figure 30: HG1700 Enclosure Side Dimensions**

### A.2.2.1 HG1700 IMU Interface Cable

The IMU interface cable provides power to the IMU from an external power source and enables input and output between the receiver and IMU. It is the same as the cable supplied with the Universal Enclosure, shown in *Figure 27* on *page 100*.

### A.2.2.2 IMU Performance

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

### A.2.2.3 Electrical and Environmental

<b>ELECTRICAL</b>	
IMU Power Consumption	IMU-H58: 9 W (max) IMU-H62: 8 W (max)
IMU Input Voltage	+12 to +28 V DC
System Power Consumption	14.8 W (typical)
Input/Output Connectors	MIL-C-38999-III, 22 pin (all IMU's) <sup>a</sup>
IMU Interface	RS-232 or RS-422
<b>ENVIRONMENTAL (IMU)</b>	
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 and power cables, see *Section K.3, Manufacturer's Part Numbers* on *page 407*.

### A.2.3 LN-200 IMU (Single-Connector Enclosure)

Table 14: 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

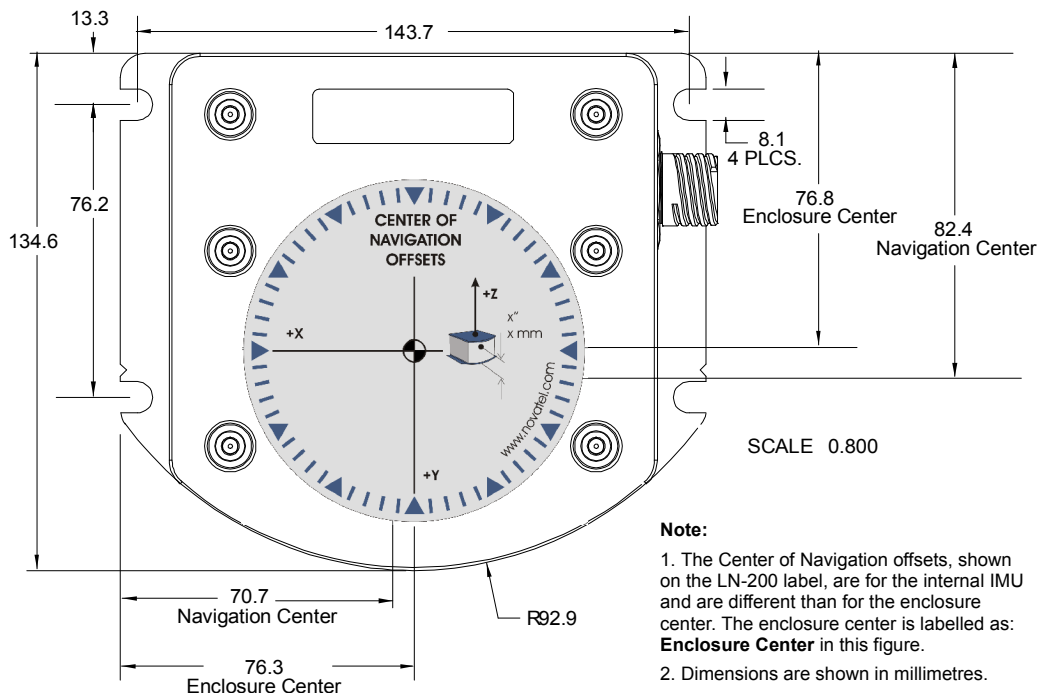
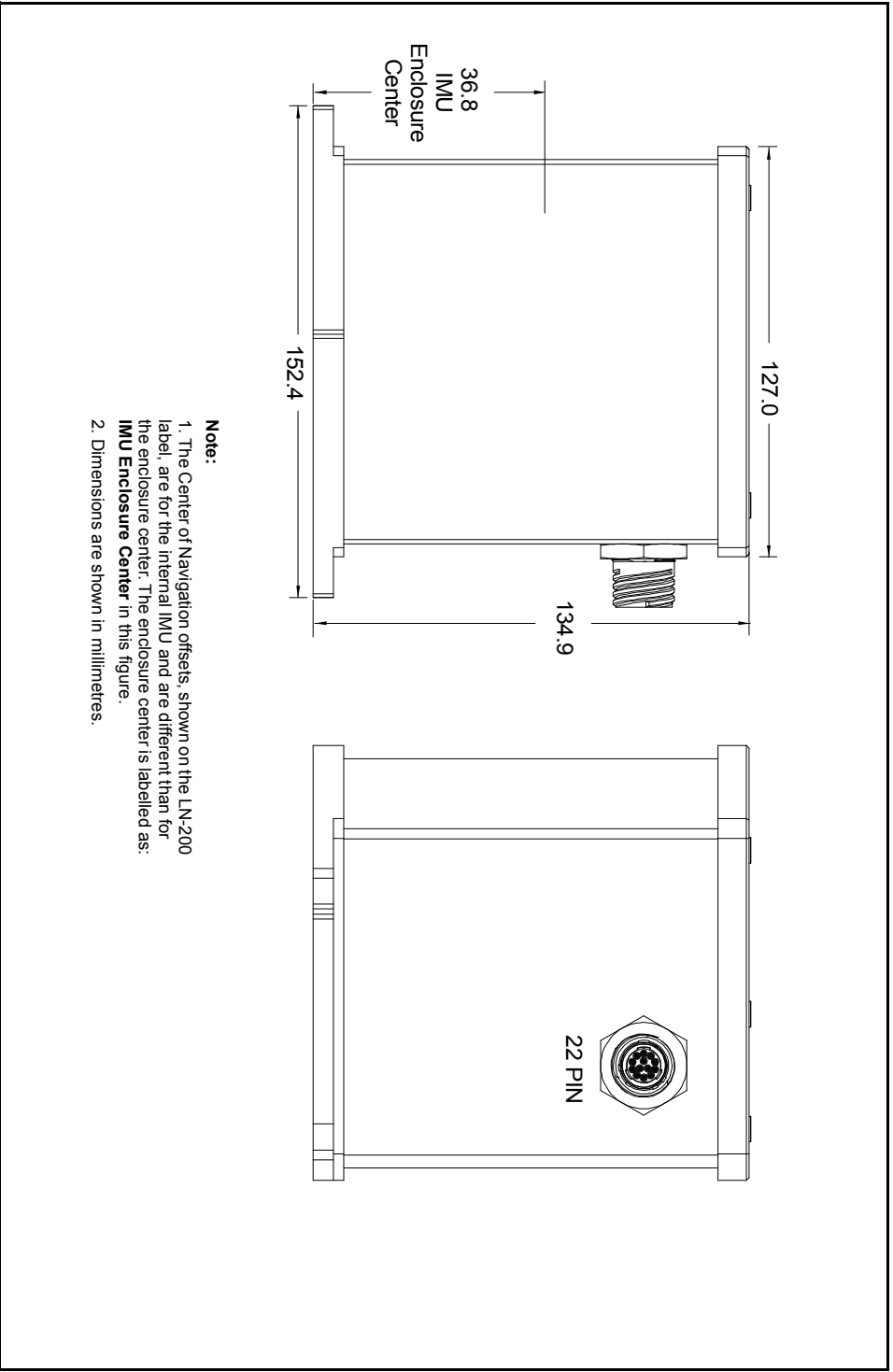


Figure 31: LN-200 IMU Enclosure Top/Bottom Dimensions and Centre of Navigation



**Note:**

1. The Center of Navigation offsets, shown on the LN-200 label, are for the internal IMU and are different than for the enclosure center. The enclosure center is labelled as: **IMU Enclosure Center** in this figure.
2. Dimensions are shown in millimetres.

**Figure 32: LN-200 Enclosure Side Dimensions**

### A.2.3.1 LN-200 IMU Interface Cable

The IMU interface cable provides power to the IMU from an external power source and enables input and output between the receiver and IMU. It is the same as the cable supplied with the Universal Enclosure, shown in *Figure 27* on *page 100*.

### A.2.3.2 IMU Performance

<b>PERFORMANCE (IMU)</b>		
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.2.3.3 Electrical and Environmental

ELECTRICAL		
IMU Power Consumption	16 W (max)	
IMU Input Voltage	+12 to +28 V DC	
System Power Consumption	ProPak-V3	14.8 W (typical)
Input/Output Connectors	MIL-C-38999-III, 22 pin (all IMU's) <sup>a</sup>	
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 and power cables, see *Section K.3, Manufacturer's Part Numbers* on *page 407*.

A.2.4 iIMU-FSAS

Table 15: 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 <sup>a</sup>	

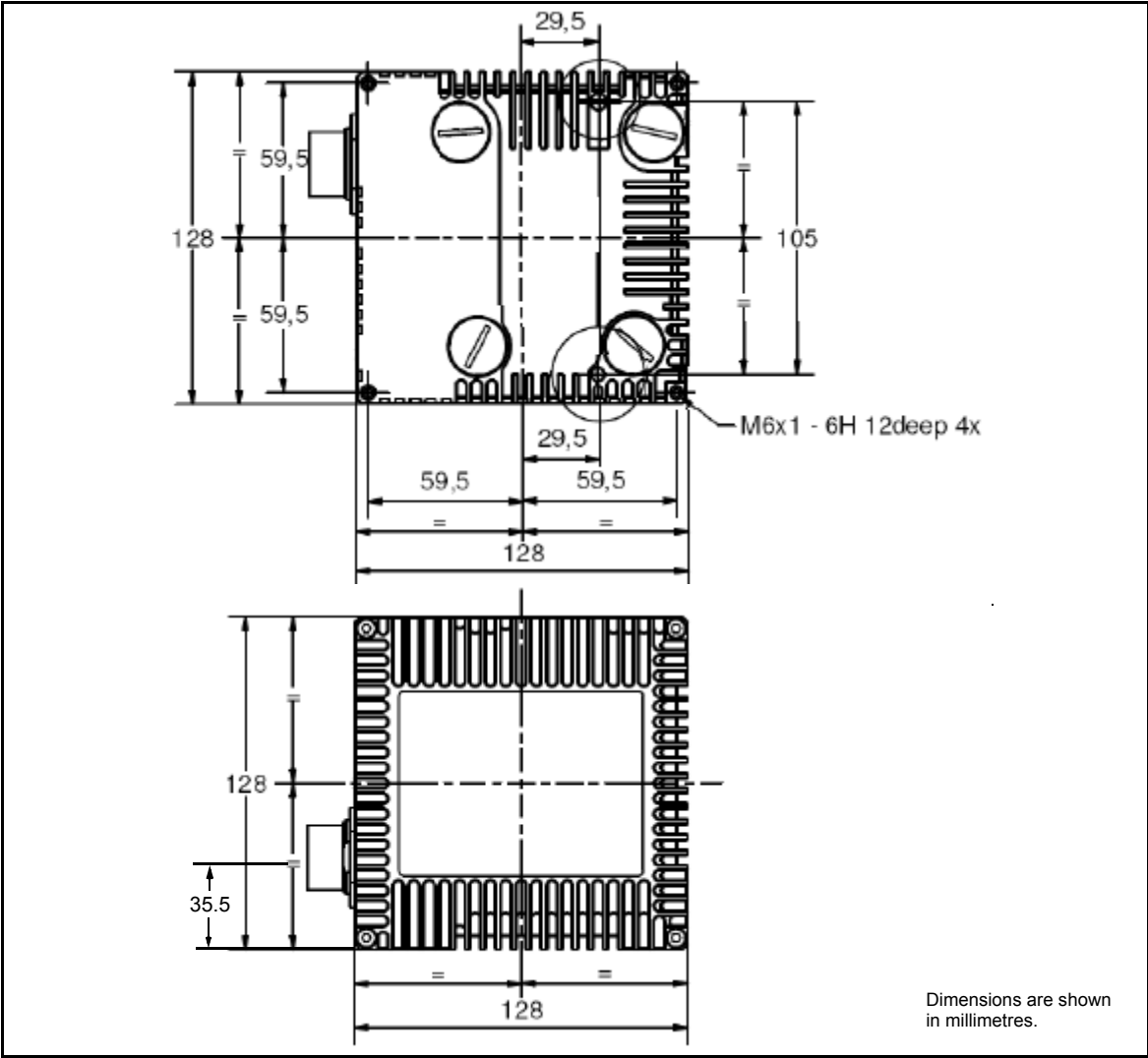
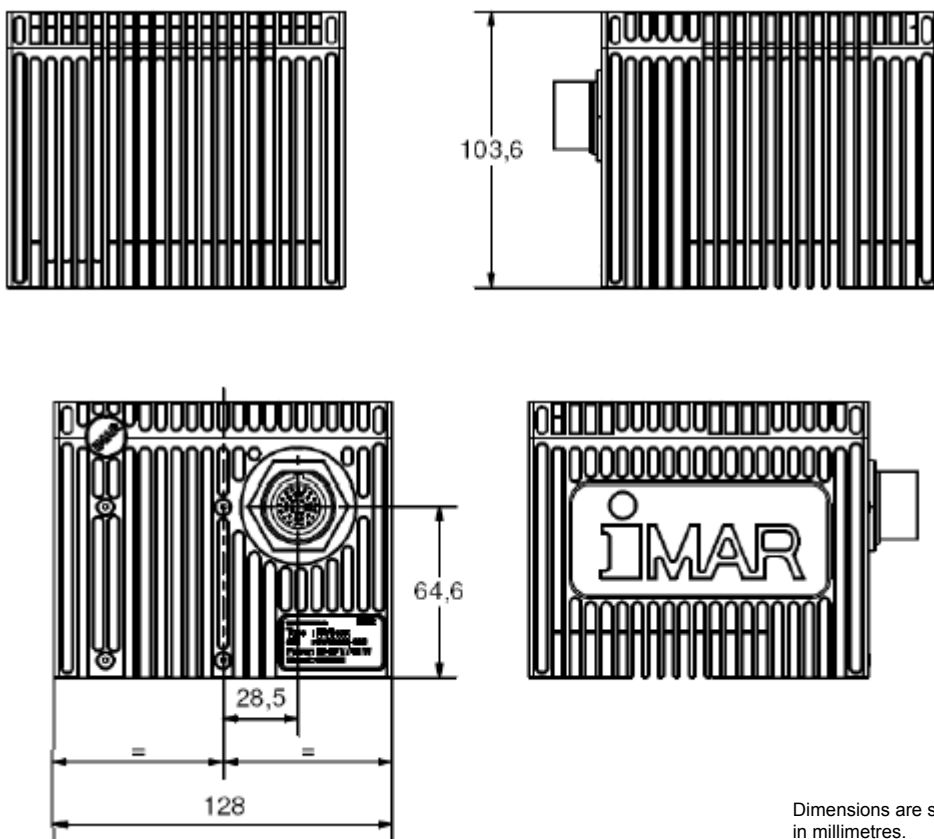


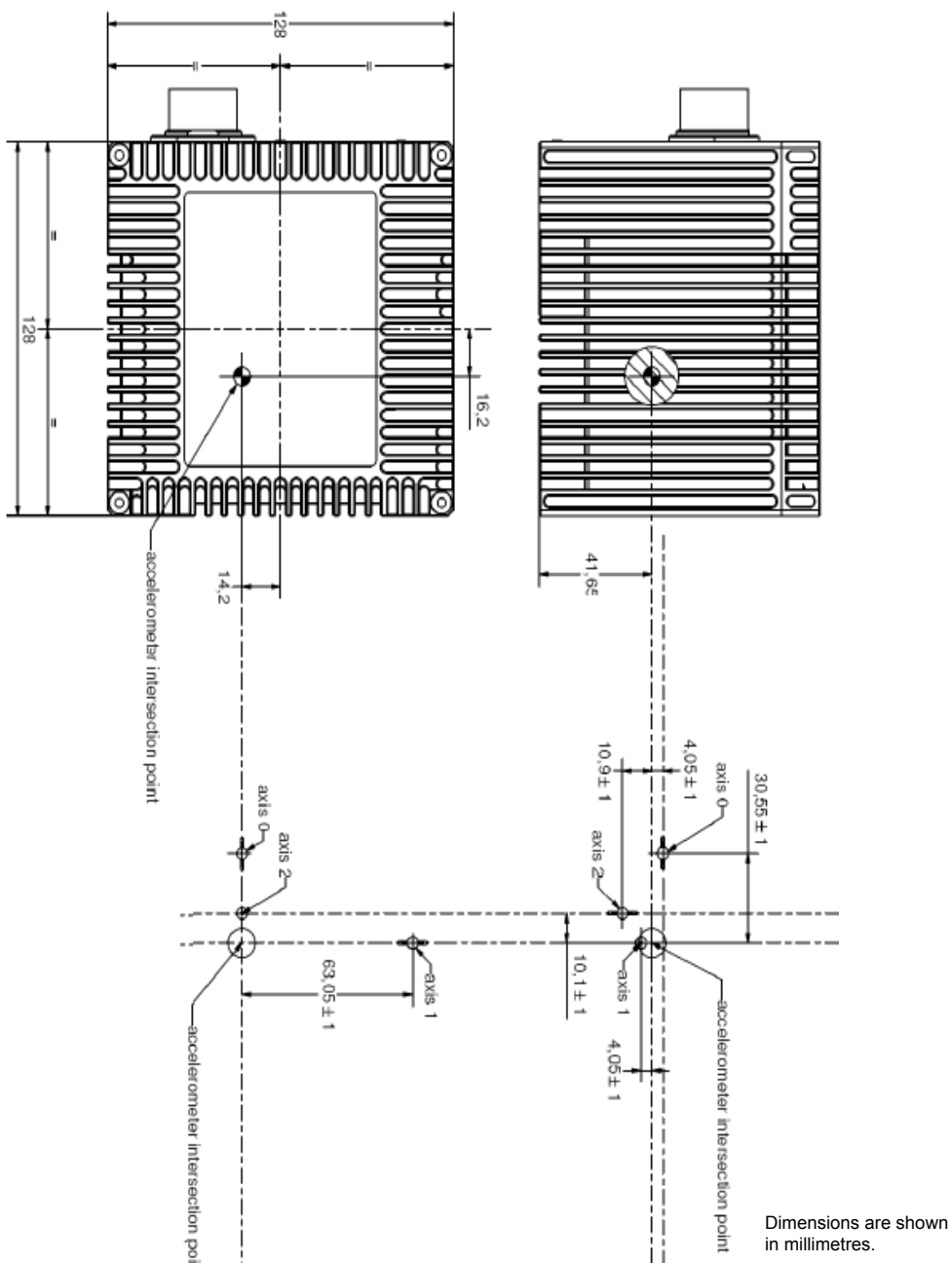
Figure 33: iIMU-FSAS Top/Bottom Dimensions

a. See Figure 35 on page 112 for the center of navigation dimensions



Dimensions are shown  
in millimetres.

**Figure 34: iLMU-FSAS Enclosure Side Dimensions**



**Figure 35: iIMU-FSAS Center of Navigation**



---

#### A.2.4.1 iIMU-FSAS Interface Cable

The Universal IMU Interface Cable, shown in *Figure 36*, is provided with new iIMU-FSAS IMUs. The NovAtel part number is 01018299 (see *Table 11, Universal IMU Enclosure Interface Cable Pinouts* on *page 101* for pinout descriptions). This cable has an optional odometer connector (NovAtel part number 01018388). See also *Section A.2.4.2, iIMU-FSAS Odometer Cabling* on *page 116* if applicable.

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 37* on *page 114* shows the iIMU interface cable connections when used with a SPAN-SE receiver.



To connect the SPAN-SE with a legacy iIMU-FSAS interface cable (NovAtel part number 01018221), a FSAS SPAN-SE Y Adapter cable is needed. Please see *Table 17* on *page 116* for cable pin-out information.



**Figure 36: Universal IMU Interface Cable**

114

**Table 16: 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 <sup>a</sup>
3	V <sub>IN</sub>	Color: red Label: 10-34 VDC				+10 to +34 VDC
4	ODO_A				6	Odometer input A(+), opto-coupler: +2 to +6 V <sup>a</sup>
5-6	Reserved					
7	DAS			1 and 6		Shielded data acquisition signal (LVTTTL 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 <sup>a</sup>
17	ODO_BN				1	Odometer input B(-), opto-coupler: +2 to +6 V <sup>a</sup>
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 17: Legacy iIMU-FSAS Cable Y Adapter Pin-Out**

Function	DB-9 Male to FSAS COM 3 Cable	DB-9 Female to FSAS I/O Cable	Description
DAS		1	Data acquisition and control signals
DO	2		Data output signal / RS-422(+)
DI	3		Data input signal / RS-422(+)
DGND	5	9	Digital ground
DIN	7		Data input signal / RS-422(-)
DON	8		Data output signal / RS-422(-)

#### A.2.4.2 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 39* 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 38* on *page 117*.



**Figure 38: 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 39: 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.2.4.1* 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 64*. The cable modification is shown in *Table 18* below.



Connect the female DB9 connector to the male ODO end of the Universal IMU interface cable.

---

**Table 18: Cable Modification for Corrsys Datron WPT**

8-pin M12 Connector on the Corrsys Datron Cable <sup>a, b</sup>			Female DB9 Connector
Pin #	Description	Color	
1	GND	White	No change
2	+U <sub>B</sub> (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.

#### A.2.4.3 IMU Performance

<b>PERFORMANCE (IMU)</b>		
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

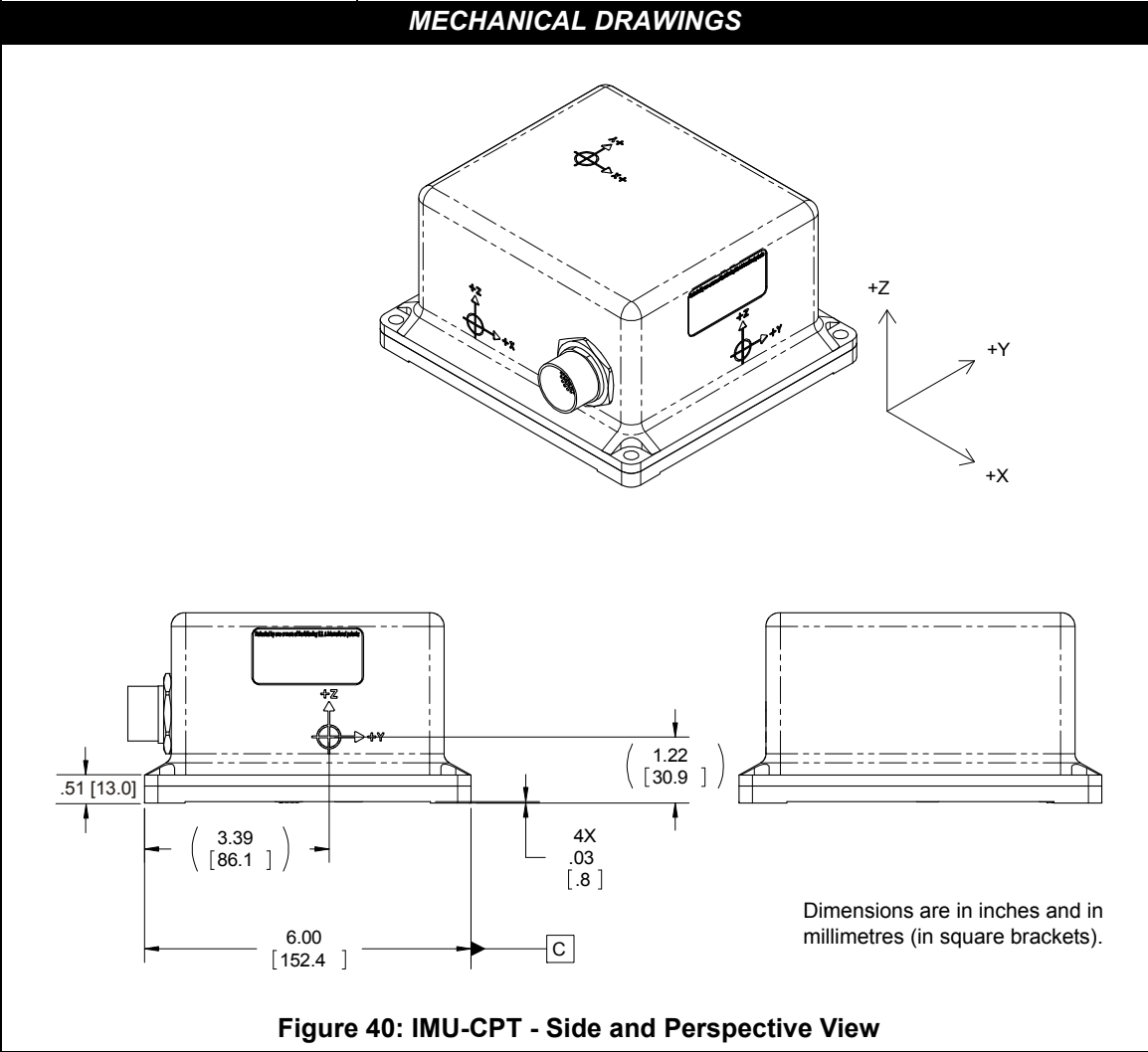
#### A.2.4.4 Electrical and Environmental

ELECTRICAL		
IMU Power Consumption	16 W (max)	
IMU Input Voltage	+10 to +34 V DC	
System Power Consumption	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	-40°C to +71°C (-40°F to 160°F)
	Storage	-40°C to +85°C (-40°F to 185°F)
Humidity	95% non-condensing	

A.2.5 IMU-CPT

Table 19: Technical HW Specs for IMU-CPT

PHYSICAL	
IMU-CPT Enclosure Size	152 mm L X 168 mm W X 89 mm H
IMU-CPT Weight	2.29 kg

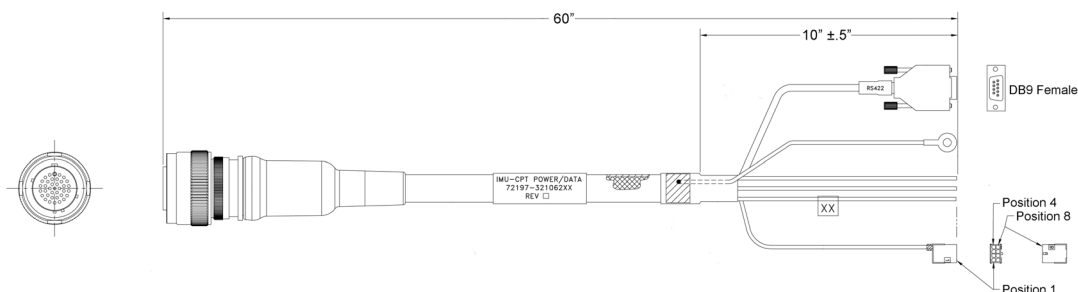






### A.2.5.1 IMU-CPT Cable

The NovAtel part number for the IMU-CPT cable is 60723114. This cable provides power to the IMU from an external power source, and enables input and output between the receiver and the IMU.



**Figure 42: IMU-CPT Development Terminated Cable**



The IMU-CPT cable has a green ground line, terminated in a ring lug as shown in *Figure 42*, that is grounded to the IMU-CPT connector body and enclosure.

**Table 20: IMU-CPT Connector Pin-Out Descriptions**

Pin No.	Function	Bare Connectors	Female DB9	8-POS MOLEX	Comments
1	Power Return	BLK			
2	9-16 VDC Power Input	WHT			
3-20	Reserved				
21	IMU RS422 TX+		2		
22	IMU RS422 TX-		8		
23-24	Reserved				
25	IMU RS422 Signal Ground		5		
26-34	Reserved				
35	TOV Output			4 <sup>a</sup>	
36	Reserved				
37	Chassis GND	WHT			
	Chassis GND	GRN			Ring lug

a. This pin connects to EVENT-IN 4 of the yellow SPAN-SE cable. Refer to *Table 9* on *page 96* for pin-out information.

---

#### A.2.5.2 IMU-CPT Sensor Specs

<b>PERFORMANCE - FIBER OPTIC GYROS</b>		
Bias Offset	± 20	°/hr
Turn On To Turn On Bias Repeatability (Compensated)	± 3	°/hr
In Run Bias Variation, At Constant Temperature	1	°/hr @ 1σ
Scale Factor Error (Total)	1500	ppm, 1σ
Scale Factor Linearity	1000	ppm, 1σ
Temperature Dependent SF Variation	500	ppm, 1σ
Angular Random Walk	0.0667	°/√hr @ 1σ
Max Input	± 375	°/sec
<b>PERFORMANCE - ACCELEROMETERS</b>		
Bias Offset	± 50	mg
Turn On To Turn On Bias Repeatability	± 0.75	mg
In Run Bias Variation, At Constant Temperature	0.25	mg @ 1σ
Temperature Dependent Bias Variation	0.5	mg/°C @ 1σ
Scale Factor Error (Total)	4000	ppm, 1σ
Temperature Dependent SF Variation	1000	ppm, 1σ
Accel Noise	55	μg/√Hz @ 1σ
Bandwidth	50	Hz
Max Input	±10	g

---

### A.2.5.3 IMU-CPT Electrical and Environmental

<b>CONNECTORS</b>	
Power and I/O	MIL-DTL-38999 Series 3
RF Antenna Connector	TNC Female
<b>ELECTRICAL</b>	
Input Power	9 - 18 VDC
Power consumption	15 W (Max)
Start-Up Time (Valid Data)	< 5 secs
<b>ENVIRONMENTAL</b>	
Temperature, operational	-40°C to +65°C
Temperature, non-operational	-50°C to +80°C
Vibration, operational	6 g rms, 20 Hz -2 KHz
Vibration, non-operational	8 g rms, 20 Hz -2 KHz
Shock, operational	7g 6-10 msec, 1/2 sine
Shock, non-operational	60 g 6-10 msec, 1/2 sine
Altitude	-1000 to 50,000 ft.
Humidity	95% at 35°C, 48 hrs
MTBF	≥ 10,500 hours

# Appendix B Commands

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

For information on OEMV-specific 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.

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 (GNSS only) position log can be logged at rates up to 20 Hz directly from the OEMV port. A BESTPOS (GNSS/INS) position log is available at < 1 Hz, 1 Hz, 2 Hz, 4 Hz and 5Hz from any SPAN-SE port. All BEST filter-related logs, such as BESTPOS, BESTVEL, BESTXYZ and BESTGPSPOS, can only be output from SPAN-SE ports at < 1 Hz, 1 Hz, 2 Hz, 4 Hz and 5Hz. Other GNSS logs (RANGE, PSRPOS, and so on) can be logged at rates up to 20 Hz from SPAN-SE ports.



Ensure that all windows, other than the Console, are closed in Connect 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 21* below.



You cannot use this method to view acknowledgement messages that the receiver sends when it receives a command. If the receiver rejects a command, you will not see a notification.

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 21: Mass Storage Device**

ASCII	Binary	Description
SD	0	Internal SD Card (default)
USBSTICK	1	External USB thumb drive

---

### B.3.1 DIR - Show Directory

Command: DIR

(Message ID = 1055)

Parameter	Values
COM Port Enum, see <i>Table 23 on page 136</i>	(THISPORT = default)
Mass Storage Device Enum, see <i>Table 21 on page 126</i>	(SD = default)

### B.3.2 CD - Change Directory

Command: CD

(Message ID = 1054)

Parameter	Values
Mass Storage Device Enum, see <i>Table 21 on page 126</i>	(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 <i>Table 21 on page 126</i>	(SD = default)
Volume Name	Optional string

### B.3.4 MKDIR - Make Directory

Command: MKDIR

(Message ID = 1060)

Parameter	Values
Mass Storage Device Enum, see <i>Table 21 on page 126</i>	(SD = default)
Path	Null terminated string

### B.3.5 RMDIR - Remove Directory

Command: RMDIR

(Message ID = 1058)

Parameter	Values
Mass Storage Device Enum, see <i>Table 21 on page 126</i>	(SD = default)
Path	Null terminated string

---

## B.3.6 PWD - Present Working Directory

Command: PWD

(Message ID = 1061)

Parameter	Values
COM Port Enum, see <i>Table 23</i> on <i>page 136</i>	(COM1 = default)
Mass Storage Device Enum, see <i>Table 21</i> on <i>page 126</i>	(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 IPCONFIG 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 computer
DIR	Directory listing of SD Card
DEL <filename>	Delete file from SD Card
CWD	Change Working Directory
QUOTE DEV <device>	Change the currently selected media device. The valid options for <device> are: <b>a:</b> , <b>b:</b> or <b>z:</b> where: a: = SD card b: = USB stick z: = internal 32 MB RAM drive
PUT <filename>	Upload a file to the currently selected media device. To change the selected media device, use the QUOTE DEV command.



## 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 ALIGNMENTMODE Set the alignment mode

**Abbreviated ASCII syntax:**

```
ALIGNMENTMODE mode
```

**Abbreviated ASCII Example:**

```
ALIGNMENTMODE AIDED_TRANSFER
```

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	-	H	0
2	mode	UNAIDED	0	Regular SPAN static coarse or kinematic alignment mode.	Enum	4	H
		AIDED_STATIC	1	Seed the static coarse alignment with an initial azimuth,			
		AIDED_TRANSFER	2	Seed the full attitude from an ALIGN solution. Pitch and Heading taken from ALIGN. Roll will be assumed 0.			

Depending on the model of the receiver, the default value will change. The default for any single antenna ‘S’ model will be UNAIDED, while the default for a dual antenna ‘D’ model will be AIDED\_TRANSFER.

**B.4.2 APPLICATION Start and remove applications**

Use this command to start applications or remove applications from the system.

Abbreviated ASCII Syntax:

Message ID: 413

APPLICATION AppSlot Function

Abbreviated ASCII Example:

```
APPLICATION USERAPP1 START
```

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	-	H	0
2	AppSlot			Application slot. Can be USERAPP1, USERAPP2 or USERAPP3	Enum	4	H
3	Function			Application function. Can be START, REMOVE, AUTOSTART_ON or AUTOSTART_OFF	Enum	4	H+4
4	xxxx			32-bit CRC	Hex	4	H+16
5	CRLF			Sentence terminator (ASCII only)	-	-	-

---

### B.4.3 APPLYVEHICLEBODYROTATION *Enable vehicle to body rotation*

Use this command to apply the vehicle to body rotation to the output attitude (which was entered with the VEHICLEBODYROTATION command, see *page 207*). 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]

#### Abbreviated ASCII Example:

APPLYVEHICLEBODYROTATION ENABLE

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	-	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				

---

#### **B.4.4 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 service is required. Contact NovAtel for details, see *page 20*.
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 website is 1557.8550 MHz. Input into the receiver: assignlb主band omnistar 1557855 1200.

##### **Abbreviated ASCII Syntax:**

**Message ID: 729**

ASSIGNLBAND mode freq baud

##### **Factory Default:**

ASSIGNLBAND IDLE

##### **Abbreviated ASCII Example 1:**

ASSIGNLBAND OMNISTAR 1557855 1200

**Table 22: L-band Mode**

Binary	ASCII	Description
0	Reserved	
1	OMNISTAR	When you select OmniSTAR, enter a dedicated frequency and baud rate.
2	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.
3	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 <i>page 295</i> . <sup>1</sup>
4	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 <i>page 146</i> . <sup>2</sup>

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 application note *APN-043 L-band Tracking and Data Output without GPS* available on our website at [www.novatel.com](http://www.novatel.com) through *Support | Knowledge and Learning*.

#### B.4.4.1 Beam Frequencies

You can switch to Omnistar VBS by using the following commands:

##### Use OmniStar VBS

```
ASSIGNLBAND OMNISTAR <freq> 1200
PSRDIFFSOURCE OMNISTAR
```

The OmniStar beam frequency chart for <freq> 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	-	H	0
2	mode	See <i>Table 22</i>		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 <i>Beam Frequencies</i> on <i>page 133</i> . (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.4.5 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 206*)
- 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 148*)



1. The COMCONTROL command, see *page 138*, 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 115200 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 23: COM Serial Port Identifiers**

Binary	ASCII	Description
1	COM1	COM Port 1
2	COM2	COM Port 2
3	COM3	COM Port 3
6	THISPORT	Currently connected port
7	FILE	SD Card
8	ALL	All ports
13	USB1	USB Device
19	COM4	COM Port 4
23	ICOM1	10/100 Ethernet 1
24	ICOM2	10/100 Ethernet 2
25	ICOM3	10/100 Ethernet 3
29	ICOM4	10/100 Ethernet 4

**Table 24: Parity**

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

**Table 25: 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	-	H	0
2	port	See <i>Table 23</i> on <i>page 136</i>		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). Baud rates of 460800 and 921600 are also available on COM2, COM3 and COM4 ports.	ULong	4	H+4
4	parity	See <i>Table 24</i> on <i>page 136</i>		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 <i>Table 25</i> on <i>page 136</i>		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.4.6 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  $\mu$ s. 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 26: 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 27: SPAN-SE COM Port Values**

Binary	ASCII
1	COM1
2	COM2
3	COM3
6	THISPORT
7	FILE
8	ALL
13	USB1
19	COM4
23	ICOM1
24	ICOM2
25	ICOM3
29	ICOM4

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	-	H	0
2	port	See <i>Table 27</i> on <i>page 139</i>		RS232/RS422 port to control. Valid ports are COM1, COM2, COM3 and COM4	Enum	4	H
3	signal	Unused on the SPAN-SE			Enum	4	H+4
4	control	Unused on the SPAN-SE			Enum	4	H+8
5	mode	RS232	0	RS-232 mode	Enum	4	H+12
		RS422	1	RS-422 mode			
		N/A	2	Used only for ICOM1 and USB1 information			

**B.4.7 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 and through Pin 4 of the OEMV3 COM1. 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	-	H	0
2	switch	0	OFF	The state of the output power lines.	Enum	4	H
		1	ON				

---

## **B.4.8 DUALANTENNAPORTCONFIG SPAN-MPPC Dual Antenna Port Setup**

Use this command to select the SPAN-MPPC COM port that is used to communicate with an external ALIGN-capable receiver. Although the SPAN-MPPC COM port is selectable, the port you choose must be connected to COM2 of the external ALIGN receiver.

### **Abbreviated ASCII Syntax:**

**Message ID: 1356**

DUALANTENNAPORTCONFIG port

### **Abbreviated ASCII Example:**

DUALANTENNAPORTCONFIG COM4

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	-	H	0
2 mark		NOPORT	0	The port that is used to communicate with an external ALIGN receiver. Selecting NOPORT disables dual antenna functionality	Enum	4	H
		COM1	1				
		COM2 (default)	2				
		COM3	3				
		COM4	19				

## B.4.9 EVENTINCONTROL Control mark input properties

This command controls up to four Event-In input triggers. See also *Section 3.11, Synchronizing External Equipment* starting on page 74.

**Abbreviated ASCII Syntax:**

**Message ID: 614**

EVENTINCONTROL mark [polarity] [t\_bias] [t\_guard]

**Abbreviated ASCII Example:**

EVENTINCONTROL MARK1 COUNT

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	-	H	0
2	mark	MARK1	0	Choose the Event-In 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 = 2 maximum = 3 599 999 If Field #3 is COUNT: This field is not used	Ulong	4	H+16

## B.4.10 EVENTOUTCONTROL Control event-out properties

This command controls up to three Event-Out output triggers. See also *Section 3.11, Synchronizing External Equipment* starting on page 74.



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]

### Abbreviated ASCII Example:

```
EVENTOUTCONTROL MARK3 ENABLE
```

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	-	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	default: 500 000 000 minimum: 25 maximum: 999 999 975		Active period of the Event Out signal in nanoseconds.	Ulong	4	H+12
6	not-active period	default: 500 000 000 minimum: 25 maximum: 999 999 975		Not-active period of the Event Out signal in nanoseconds.	Ulong	4	H+16

## B.4.11 EXTHDGOFFSET Set the Angular Offset

The EXTHDGOFFSET command can be used to specify the angular offset from the dual antenna baseline to the SPAN computation frame. It is highly recommended that these offsets be entered by entering a lever arm to both antennas as the measurement errors will be lower (see *Section 4.5, Configuring SPAN with ALIGN on SE-D and MPPC-D Models on page 79*). However this command can be used to enter the offsets directly if necessary.



EXTHDGOFFSET is also available as a log, when both lever arms are entered. Refer to *Section D.3.9, EXTHDGOFFSET Log the Angular Offset on page 245*.

### Abbreviated ASCII Syntax:

Message ID: 1204

EXTHDGOFFSET heading headingSTD [pitch] [pitchSTD]]

### Abbreviated ASCII Example:

```
EXTHDGOFFSET 0.5 1.0 -0.23 1.0
```

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	Heading	-180.0 – 180.0		Angular offset for the azimuth, or heading between the external aiding source and the IMU forward axis. Input in degrees.	DOUBLE	8	H
3	HeadingSTD	0 to 10.0		Input heading offset standard deviation. Input in degrees.	DOUBLE	8	H+8
4	Pitch	-90.0 to 90.0		Angular offset for the pitch between the external aiding source and the IMU forward axis. Input in degrees. Default = 0.0	DOUBLE	8	H+16
5	PitchSTD	0 to 10.0		Input pitch offset standard deviation. Input in degrees. Default = 0.0	DOUBLE	8	H+24



---

### B.4.12 **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]

#### **Abbreviated ASCII Example:**

FORMAT SD

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	-	H	0
2	device			Choose a mass storage device, see <i>Table 21</i> on <i>page 126</i>	Enum	4	H
3	volume			DOS volume label	String[11]	11	H+4

### B.4.13 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]

#### Abbreviated ASCII Example:

```
FRESET USER_CFG
```

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	-	H	0
2	target	See Table 28		What data is to be reset by the receiver.	Enum	4	H

Table 28: 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
18	INS_LEVER_ARM2	Resets the secondary GNSS antenna to IMU lever arm.

#### B.4.14 GIMBALSPANROTATION Gimbal frame to SPAN frame rotation

This command is used to specify a rotational offset between a gimbal mount and the SPAN computation frame. This command must be used if the mount frame and SPAN computation frame do not match after the mapping from SETGIMBALORIENTATION is applied to the mount. See Chapter 6, **Variable Lever Arm** for details on frame definitions.

The message's format and definitions are identical to those in the VEHICLEBODYROTATION command. The angles must be entered in the SPAN computation frame and the direction of the angles is from the mount to the SPAN computation frame.

##### Abbreviated ASCII Syntax:

**Message ID: 1319**

GIMBALSPANROTATION XAngle YAngle ZAngle [XUncert ][YUncert ][ZUncert ]

##### Abbreviated ASCII Examples:

```
GIMBALSPANROTATION 0 0 90
GIMBALSPANROTATION 0 0 90 0.1 0.1 1.0
```

Field	Field Type	ASCII / 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	-	H	0
2	XAngle	-180 – 180 degrees	Right hand rotation around the SPAN frame X axis in degrees	Double	8	H
3	YAngle	-180 – 180 degrees	Right hand rotation around the SPAN frame Y axis in degrees	Double	8	H+8
4	ZAngle	-180 – 180 degrees	Right hand rotation around the SPAN frame Z axis in degrees	Double	8	H+16
5	XUncert	0 – 180 degrees	Uncertainty of X rotation in degrees. Default is 0	Double	8	H+24
6	YUncert	0 – 180 degrees	Uncertainty of Y rotation in degrees. Default is 0	Double	8	H+32
7	ZUncert	0 – 180 degrees	Uncertainty of Z rotation in degrees. Default is 0	Double	8	H+40

---

### **B.4.15 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. The GNSSCARDCONFIG command is especially useful for configuring RTK because the OEMV-3 COM1 port is used for RTK correction input data.

You cannot use this command with the OEMV-2 and OEMV-3 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.

The GNSSCARDCONFIG command can also be used to request that the OEMV-3 output a GPGLG log at the specified rate. This is required for use with some Virtual Reference Station (VRS) networks. Note that the logging rate is restricted to the usual OEMV-3 logging rates, and the Transmit Interface Mode must be NOVATEL.

#### **Abbreviated ASCII Syntax:**

**Message ID: 1092**

GNSSCARDCONFIG [card] [port] rx\_inter tx\_inter [response] bps [parity] [data bits] [stop bits] [handshaking] [echo] [break] [GPGLG period]

#### **Abbreviated ASCII Example:**

```
GNSSCARDCONFIG CARD1 COM1 RTCA NOVATEL ON 57600 N 8 1 N OFF ON 0
```

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	-	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 <i>Table 29</i> on <i>page 150</i>		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)			
14	GPGGA output period	0 to disable output (default) Valid values for 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.		Logging period for GPGGA output	Float	4	H+48

**Table 29: 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 <sup>1</sup>
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.

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.

## B.4.16 ICOMCONFIG Set ICOMx 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. Each port can be used for both TCP and UDP traffic but not simultaneously. You must configure the system for either UDP or TCP communication. The default is TCP.

You can also use this command to program a destination or peer IP. When using UDP, the destination or peer IP allows SPAN to start sending data to the remote peer without waiting to receive data first (by sending something first, SPAN then gains knowledge of the remote IP). This command is saved using the SAVECONFIG command.

To configure only the ICOM transport protocol, use the ICOMCONFIG command with a port name and its one non-optional parameter.



The SPAN-SE does not currently use the ULPort parameter. A dummy value (e.g., 0) must be entered but is ignored by the firmware. Default TCP port numbers persist for each ICOM port.

### Abbreviated ASCII Syntax

Message ID = 1248

ICOMCONFIG Port IPProtocol [ULPort][IPAddress][Interface]

### Abbreviated ASCII Example:

```
ICOMCONFIG ICOM2 TCP 3001 192.168.1.25
```

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	-	H	0
2	Port	See Table 23, COM Serial Port Identifiers, on page 136		Port to configure	Enum	4	H
3	IPProtocol	DISABLE	0	Not used	Enum	4	H+4
		TCP	1	Transport Control Protocol (default)			
		UDP	2	User Datagram Protocol	Enum	4	H+8
4	ULPort			Not used	Enum	4	H+12
5	IPAddress			Peer IP Address	Enum	4	H+16
6	Interface	ALL	1	The Ethernet adaptor to use. SPAN-SE can currently only use ETHA	Enum	4	H+20
		ETHA (default)	2				
		ETHB	3				

---

### ***B.4.17 INPUTGIMBALANGLE    Input gimbal angles into the SPAN-SE***

This command is used by the external gimble source to input gimble angles to the SPAN-SE body mount frame. Angles are input in the body mount frame.

**Abbreviated ASCII Syntax:**

**Message ID: 1317**

INPUTGIMBALANGLE XAngle YAngle ZAngle [XUncert][YUncert][ZUncert]

**Abbreviated ASCII Examples:**

INPUTGIMBALANGLE 0.003 -0.1234 12.837

INPUTGIMBALANGLE 0.003 -0.1234 12.837 0.001 0.001 0.005

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	-	H	0
2	XAngle	-180 – 180		Right hand rotation from mount to gimbal plane about mount body frame X axis in degrees	Double	8	H
3	YAngle	-180 – 180		Right hand rotation from mount to gimbal plane about mount body frame Y axis in degrees	Double	8	H+8
4	ZAngle	-180 – 180		Right hand rotation from mount to gimbal plane about mount body frame Z axis in degrees	Double	8	H+16
4	XUncert	0 – 180		Uncertainty of X rotation in degrees. Default is 0	Double	8	H+24
5	YUncert	0 – 180		Uncertainty of Y rotation in degrees. Default is 0	Double	8	H+32
6	ZUncert	0 – 180		Uncertainty of Z rotation in degrees. Default is 0	Double	8	H+40



---

## B.4.18 IPCONFIG Set IP information

Use this command to configure Internet Protocol (IP) information. See also *Section 3.12, SPAN-SE Ethernet Connection* on page 76.

### Abbreviated ASCII Syntax:

Message ID: 1243

IPCONFIG Interface Mode IP Mask Gateway

### Abbreviated ASCII Example:

```
IPCONFIG ETHA STATIC 10.1.100.25 255.255.255.0 10.1.100.1
```

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	-	H	0
2	Interface			Physical interface. SPAN-SE can currently only use ETHA	Ulong	4	H
3	Mode	DHCP	1	Automatic IP address assignment	Ulong	4	H+4
		Static	2	Manual IP address assignment			
4	IP			IP Address	Enum	4	H+8
5	Mask			Subnet mask	Enum	4	H+12
6	Gateway			IP gateway	Enum	4	H+16

**B.4.19 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 55*

Abbreviated ASCII Syntax:

Message ID: 379

INSCOMMAND action

Abbreviated ASCII Example:

```
INSCOMMAND ENABLE
```

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	-	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)			

---

### ***B.4.20 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.



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

#### **Abbreviated ASCII Syntax:**

**Message ID: 382**

INSZUPT

**B.4.21 INSZUPTCONTROL    INS Zero Velocity Update Control**

This command allows you to control whether ZUPTs are performed by the system.

When enabled, ZUPTs allow the INS to reduce its accumulated errors. Typically, the system will automatically detect when it is stationary, and apply a ZUPT. For certain applications where it is known that the system will never be stationary, such as marine or airborne applications, ZUPTs can be disabled altogether.

Abbreviated ASCII Syntax:

Message ID: 1293

INSZUPTCONTROL switch

Abbreviated ASCII Example:

```
INSZUPTCONTROL DISABLE
```

Field	Field Type	ASCII Value	Binary Value	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	-	-	This field contains the command name or message header depending on whether the command is abbreviated ASCII, ASCII or binary	-	H	0
2	Switch	Disable	0	Disable INS zero velocity updates.	Enum	4	H
		Enable	1	Enable INS zero velocity updates (default)			

---

## B.4.22 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 center of navigation to the phase center of the antenna. See also the SETIMUTOANTOFFSET command starting on *page 184* and *Section 3.4.6, Lever Arm Calibration Routine* starting on *page 59*.

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 237*.



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 SETIMUANTOFFSET (or NVM). Ensure the initial standard deviations are representative of the initial lever arm values.



LEVERARMCALIBRATE is not available for the IMU-CPT or HG1930 IMUs.

### Abbreviated ASCII Syntax:

Message ID: 675

LEVERARMCALIBRATE [switch] maxtime [maxstd]

### 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  $\leq 0.05$  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.

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	-	H	0
2	switch	OFF	0	Enable or disable the lever arm calibration routine.	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

### B.4.23 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 D.1, Log Types* starting on *page 214* 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 204*. To remove all logs that have the [hold] parameter, use the UNLOGALL command with the *held* field set to 1, see *page 206*.

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 MARKxTIME or MARKxPVA logs, see *page 306*
5. Only the MARKxPVA logs, or MARKxTIME 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 B.1, Command Formats* on page 125 for additional examples.

Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
1	LOG (binary) header	(See <i>Table 39, Binary Message Header Structure</i> , on page 218)	This field contains the message header.	-	H	0
2	port	See <i>Table 23, COM Serial Port Identifiers</i> , on page 136	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

*Continued on the following page*



Field	Field Name	Binary Value	Description	Field Type	Binary Bytes	Binary Offset
6	trigger	0: ONNEW	Does not output current message but outputs when the message is updated (not necessarily changed) <sup>1</sup>	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 <sup>2</sup>	Double	8	H+12
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 MARKxPVA and MARKxTIME logs starting on *page 306*.
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 <i>Table 23, COM Serial Port Identifiers</i> , on <i>page 136</i>	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 <i>Footnote 1</i> on <i>page 161</i> )	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 is 0) (see <i>Footnote 2</i> on <i>page 161</i> )	Double
6	offset	Any positive double value smaller than the period	Offset for period (ONTIME trigger) in seconds. To log data at 1 second after every minute, set the period to 60 and the offset to 1 (default is 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.4.24 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]

#### Abbreviated ASCII Example:

```
LOGFILE OPEN SD SITE1.GPS
```

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	-	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 <i>Table 21</i> on <i>page 126</i> . Default is SD	Enum	4	H+4
4	filename			Filename where filenames have a maximum 12 character limit. Default is SPAN_#.log where # is the next number in the list starting at 0	Char[12]	12	H+8

**B.4.25 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 or GN ID. *Table 30* shows the NMEA logs and whether they use GP or GP + IN IDs with `nmeatalker auto`.

**Table 30: NMEA Messages**

NMEA Message	NMEA Talker Options
GLMLA	GL
GPALM	GP
GPGGA	GP
GPGLL	GP or IN
GPGRS	GP or GN
GPGSA	GP or GN
GPGST	GP or GN
GPGSV	GP and GL
GPRMB	GP or GN
GPRMC	GP or GN
GPVTG	GP or IN
GPZDA	GP
GPHDT	GP
PASHR	N/A (always PASHR)

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	-	H	0
2	ID	GP	0	GPS (GP) only	Enum	4	H
		AUTO	1	GNSS, GP, GL, GN and/or Inertial (IN)			

---

## **B.4.26 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 172*.



1. To use L-band differential corrections, an L-band receiver and a subscription to the OmniSTAR service is required. Contact NovAtel for details, see *page 20*.
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 31: DGPS Type**

Binary	ASCII	Description
0	RTCM <sup>3</sup>	RTCM ID: $0 \leq \text{RTCM ID} \leq 1023$ <b>or</b> ANY
1	RTCA <sup>3</sup>	RTCA ID: A four character string containing only alpha (a-z) or numeric characters (0-9) <b>or</b> ANY
2	CMR <sup>3</sup>	CMR ID: $0 \leq \text{CMR ID} \leq 31$ <b>or</b> ANY Cannot be used in PSRDIFFSOURCE command.
3	OMNISTAR <sup>1 3</sup>	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.
5	SBAS <sup>1 3</sup>	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.
6	RTK <sup>2</sup>	In the PSRDIFFSOURCE command, RTK enables using RTK correction types for PSRDIFF positioning. When using multiple correction types such as RTCM, RTCA, RTCMV3, or CMR, the positioning filter selects the first received message.
10	AUTO <sup>1 2</sup>	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 <sup>1 3</sup>	Disables all differential correction types
12	Reserved	
13	RTCMV3	RTCM Version 3.0 ID: $0 \leq \text{RTCMV3 ID} \leq 4095$ <b>or</b> ANY Cannot be used in PSRDIFFSOURCE command/

1. Base Station ID parameter in the command is ignored
2. Available only with the PSRDIFFSOURCE command (refer to *page 166*)
3. All PSRDIFFSOURCE entries fall back to SBAS (except NONE).

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	-	H	0
2	type	See <i>Table 31</i> on <i>page 167</i>		ID Type. All types may revert to SBAS (if enabled) or SINGLE position types. See also <i>Table 44, Position or Velocity Type</i> , on <i>page 229</i> . <sup>1</sup>	Enum	4	H
3	Base Station ID	Char [5] or ANY		ID string	Char[5]	8 <sup>2</sup>	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.4.27 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 146*.

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	-	H	0
2	delay			Seconds to wait before resetting. Default is 0	Ulong	4	H

**B.4.28 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 AdvVance 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	-	H	0
2	type	USE_DEFAULTS	0	Reset to defaults	Enum	4	H
		RESET	1	Reset RTK algorithm			

### B.4.29 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 32: 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	-	H	0
2	mode	See Table 32		Set the dynamics mode	Enum	4	H

---

### B.4.30 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 166*. To set up RTK differential corrections, see the GNSSCARDCONFIG command on *page 148*.



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
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	-	H	0
2	type	See <i>Table 31, DGPS Type</i> , on <i>page 167</i>		ID Type <sup>1</sup>	Enum	4	H
3	Base Station ID	Char [5] or ANY		ID string	Char[5]	8 <sup>2</sup>	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.4.31 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.



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 345*. 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 55*). The angular offset values are not applied to the attitude output, unless the APPLYVEHICLEBODYROTATION command is disabled.

Abbreviated ASCII Syntax:

RVBCALIBRATE reset

Message ID: 641

**Abbreviated ASCII Example:**

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	-	H	0
2	Switch	RESET	0	Control the vehicle/body rotation computation	ENUM	4	H
		DISABLE	1				
		ENABLE	2				

---

### **B.4.32 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 329*. See also the FRESET command, *page 146*.



If you are using this command in Connect, 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.4.33 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 33: 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	-	H	0
2	keyword	DISABLE	0	Receiver does not use SBAS corrections	Enum	4	H
		ENABLE	1	Receiver uses SBAS corrections			
3	system	See <i>Table 33</i> on <i>page 176</i>		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.4.34 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 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

#### Abbreviated ASCII Example:

```
setautologging 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	-	H	0
2	switch	OFF	0	Enable or disable auto-logging on boot-up	Enum	4	H
		ON	1				

B.4.35 SETGIMBALORIENTATION

This command is used to convert Mount Body frame to Mount Computation frame in the exact same manner as SETIMUORIENTATION does for SPAN. The mapping definitions here are the same as they are there. However, unlike an IMU, SPAN will not be able to auto-detect the orientation of the mount used, so this command must be sent to SPAN. If the command is not sent, SPAN will assume a default mapping of 5, which if incorrect would result in bad results.

Abbreviated ASCII Syntax:

Message ID: 1318

SETGIMBALORIENTATION mapping

Abbreviated ASCII Example:

SETGIMBALORIENTATION 6

Field	Field Type	Value Range	Description	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	–	H	0
2	Mapping	1	Mount <b>X</b> axis is pointing <b>UP</b>	Double <sup>1</sup>	8	H
		2	Mount <b>X</b> axis is pointing <b>DOWN</b>			
		3	Mount <b>Y</b> axis is pointing <b>UP</b>			
		4	Mount <b>Y</b> axis is pointing <b>DOWN</b>			
		5	Mount <b>Z</b> axis is pointing <b>UP</b> (default)			
		6	Mount <b>Z</b> axis is pointing <b>DOWN</b>			

1. See Table 35, IMU Type, on page 187 for details

**B.4.36   SETHEAWEWINDOW   Set Heave Filter Length**

This command allows user control over the length of the heave filter. This filter determines the heave (vertical displacement) of the IMU, relative to a long-term level surface.



Heave functionality is available on SPAN-SE only.

**Abbreviated ASCII Syntax:**

**Message ID: 1383**

SETHEAWEWINDOW filterlength

**Abbreviated ASCII Example**

SETHEAWEWINDOW 35

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	Filter Length	Integer (1 – 300s) (default = 20 s)		This filter length will be used in the heave filter. Typically, set the filter length to 5 x Wave Period.	Int	4	H

## B.4.37 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 57*.



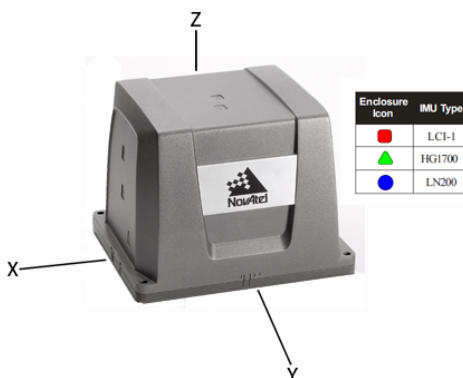
Ensure that all windows, other than the Console, are closed in Connect and then use the SAVECONFIG command to save settings in NVM. Otherwise, unnecessary data logging occurs and may overload your system.

This orientation command serves to transform the incoming IMU signals in such a way that a 5 mapping is achieved, see *Table 34 on page 183*. 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:

$X \Rightarrow Z, Y \Rightarrow X, Z \Rightarrow Y$  (where the default is  $X \Rightarrow X, Y \Rightarrow Y, Z \Rightarrow Z$ )

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 88*. The example of the Litef is shown in Figure 43.



**Figure 43: IMU Center of Navigation (example)**



1. Azimuth is positive in a clock wise 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

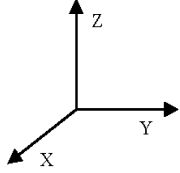
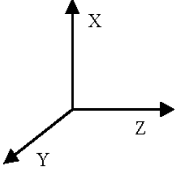
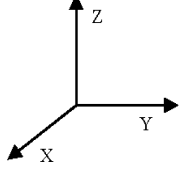
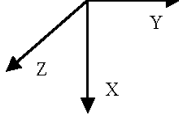
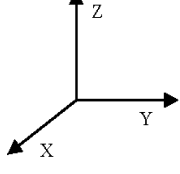
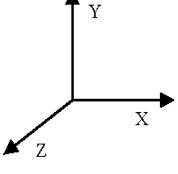
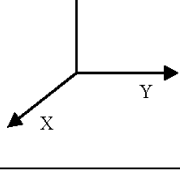
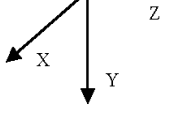
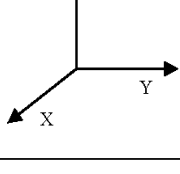
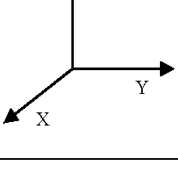
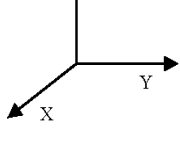
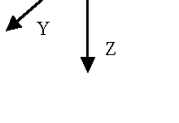
SETIMUORIENTATION switch

### Abbreviated ASCII Example:

```
SETIMUORIENTATION 1
```

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	-	H	0
2	Switch	0	0	IMU determines axis orientation automatically during coarse alignment. (default)	ENUM	4	H
		1	1	IMU <b>X</b> axis is pointing <b>UP</b>			
		2	2	IMU <b>X</b> axis is pointing <b>DOWN</b>			
		3	3	IMU <b>Y</b> axis is pointing <b>UP</b>			
		4	4	IMU <b>Y</b> axis is pointing <b>DOWN</b>			
		5	5	IMU <b>Z</b> axis is pointing <b>UP</b>			
		6	6	IMU <b>Z</b> axis is pointing <b>DOWN</b>			

**Table 34: 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.4.38 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 millimetres 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 centimetre 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) or the INS will reset and require alignment again.

#### Abbreviated ASCII Syntax:

Message ID: 383

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

#### Abbreviated ASCII Example:

```
SETIMUTOANTOFFSET 0.54 0.32 1.20 0.03 0.03 0.05
```

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	-	H	0
2	x	± 100		x offset (m)	Double	8	H
3	y	± 100		y offset (m)	Double	8	H+8
4	z	± 100		z offset (m)	Double	8	H+16
5	a	0 to +10		Uncertainty in x (m). Default is 10% of the x offset to a minimum of 0.01 m	Double	8	H+24
6	b	0 to +10		Uncertainty in y (m). Default is 10% of the y offset to a minimum of 0.01 m	Double	8	H+32
7	c	0 to +10		Uncertainty in z (m). Default is 10% of the z offset to a minimum of 0.01 m	Double	8	H+40



### B.4.39 SETIMUTOANTOFFSET2 Set IMU to GPS2 antenna offset

Set the lever arm for the GPS2 antenna. Preferably, the GPS1 antenna will be set up behind the IMU forward axis and the GPS2 antenna will be set up ahead of the IMU forward axis. Entering both lever arms will automatically compute the angular offset between the ALIGN antennas and the IMU axes.

The format of this command is identical to the SETIMUTOANTOFFSET command.

**Abbreviated ASCII Syntax:**

**Message ID: 1205**

SETIMUTOANTOFFSET2 x y z [a] [b] [c].

**Abbreviated ASCII Example:**

SETIMUTOANTOFFSET2 0.54 0.32 1.20 0.03 0.03 0.05

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	-	H	0
2	x	± 100		x offset (m)	Double	8	H
3	y	± 100		y offset (m)	Double	8	H+8
4	z	± 100		z offset (m)	Double	8	H+16
5	a	0 to +10		Uncertainty in x (m). Default is 10% of the x offset to a minimum of 0.01 m	Double	8	H+24
6	b	0 to +10		Uncertainty in y (m). Default is 10% of the y offset to a minimum of 0.01 m	Double	8	H+32
7	c	0 to +10		Uncertainty in z (m). Default is 10% of the z offset to a minimum of 0.01 m	Double	8	H+40

**B.4.40 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.



Ensure that all windows, other than the Console, are closed in Connect 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

**Abbreviated ASCII Example:**

```
SETIMUTYPE IMU_IMAR_FSAS
```

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	-	H	0
2	Switch	See Table 35, IMU Type, on page 187		IMU Type	ENUM	4	H

**Table 35: 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	IMU_HG1900_CA29	Honeywell HG1900 CA29
6, 7	Reserved	
8	IMU_LN200	Litton LN-200 (200 Hz model)
9	IMU_LN200_400HZ	Litton LN-200 (400 Hz model)
10	Reserved	
11	IMU_HG1700_AG58	Honeywell HG1700 AG58
12	IMU_HG1700_AG62	Honeywell HG1700 AG62
13	IMU_IMAR_FSAS	iMAR iIMU-FSAS
14-15	Reserved	
16	IMU_KVH_COTS	IMU-CPT
17, 18	Reserved	
19	IMU_LITEF_LCI1	NG Litef LCI 1 IMU
20	IMU_HG1930_AA99	Honeywell HG1930 AA99

---

### **B.4.41 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 on page 181 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 175) 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 179, unless you have your IMU mounted with the Z-axis not pointing up. Then use the tables in the SETIMUORIENTATION command, on pages 182-183, to determine the azimuth axis that SPAN is using.

#### **Abbreviated ASCII Syntax:**

**Message ID: 862**

SETINITATTITUDE pitch roll azimuth pitchSTD rollSTD azSTD

#### **Abbreviated ASCII Example:**

```
SETINITATTITUDE 0 0 90 5 5 5
```

In this example, the initial roll and pitch have 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 190), also with a 5 degrees standard deviation.

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	-	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° <sup>1</sup> 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

---

## B.4.42 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 179*, 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 175*) 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 179*, unless you have your IMU mounted with the Z-axis not pointing up. Then, use the tables in the SETIMUORIENTATION command, on *pages 182-183*, to determine the azimuth axis that SPAN is using.

### Abbreviated ASCII Syntax:

Message ID: 863

SETINITAZIMUTH azimuth azSTD

### 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.

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	-	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

---

### **B.4.43 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

#### **Abbreviated ASCII Example:**

SETINSOFFSET 0.15 0.15 0.25

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	-	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



#### **B.4.44 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 306*. 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  $\alpha$ offset  $\beta$ offset  $\gamma$ offset

##### **Abbreviated ASCII Syntax:**

**Message ID: 1070**

SETMARK2OFFSET xoffset yoffset zoffset  $\alpha$ offset  $\beta$ offset  $\gamma$ offset

##### **Abbreviated ASCII Syntax:**

**Message ID: 1116**

SETMARK3OFFSET xoffset yoffset zoffset  $\alpha$ offset  $\beta$ offset  $\gamma$ offset

##### **Abbreviated ASCII Syntax:**

**Message ID: 1117**

SETMARK4OFFSET xoffset yoffset zoffset  $\alpha$ offset  $\beta$ offset  $\gamma$ offset

##### **Abbreviated ASCII Example:**

SETMARK1OFFSET -0.324 0.106 1.325 0 0 0

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	-	H	0
2	x offset	$\pm 360$		Offset along the IMU enclosure frame X-axis (m) for Mark1, 2, 3 or 4	Double	8	H
3	y offset	$\pm 360$		Offset along the IMU enclosure frame Y-axis (m) for Mark1, 2, 3 or 4	Double	8	H+8
4	z offset	$\pm 360$		Offset along the IMU enclosure frame Z-axis (m) for Mark1, 2, 3 or 4	Double	8	H+16
5	$\alpha$ offset	$\pm 360$		Roll offset for Mark in degrees	Double	8	H+24
6	$\beta$ offset	$\pm 360$		Pitch offset for Mark in degrees	Double	8	H+32
7	$\gamma$ offset	$\pm 360$		Azimuth offset for Mark in degrees	Double	8	H+40

## B.4.45 SETUPSENSOR Add a new camera sensor

Use the SETUPSENSOR command to add a new camera sensor to the system.

### Abbreviated ASCII Syntax:

Message ID: 1333

SETUPSENSOR SensorID EventOut OPP OAP EventIn EIC MIP MITB MITG

### Abbreviated ASCII Example:

SETUPSENSOR SENSOR3 MARK1 POSITIVE 2 MARK4 EVENT POSITIVE 0 2

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	-	H	0
2	SensorID	SENSOR1	0	The sensor being configured	Enum	4	H
		SENSOR2	1				
		SENSOR3	2				
3	EventOut	MARK1	0	Connected to a Sensor Input Trigger	Enum	4	H+4
		MARK2	1				
		MARK3	2				
		MARK4	3				
4	OPP	NEGATIVE	0	Output pulse polarity	Enum	4	H+8
		POSITIVE	1				
5	OAP	2 – 500ms Value must be divisible by 2		Output active period in milliseconds. Default is 2 ms	Ulong	4	H+12
6	EventIn	MARK1	0	Connected to a sensor output. Default is MARK2	Enum	4	H+16
		MARK2	1				
		MARK3	2				
		MARK4	3				
7	EIC	DISABLE	0	Event in control. Default is DISABLE	Enum	4	H+20
		EVENT	1				
8	MIP	NEGATIVE	0	Mark input polarity. Default is POSITIVE	Enum	4	H+24
		POSITIVE	1				
9	MITB	-99999999 – 99999999 ms		Mark input time bias in nanoseconds. Default is 0	Long	4	H+28
10	MITG	2 – 3599999	??	Mark input time guard in milliseconds. Default is 4	Ulong	4	H+32

---

## B.4.46 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 3.5, SPAN Wheel Sensor Configuration on page 64*)

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

### 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.

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	-	H	0
2	input	See <i>Table 36</i> on <i>page 196</i>		Optional field to specify to which input the command should be applied. Default is 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 is POSITIVE	Enum	4	H+4
		POSITIVE	1				
4	ticks	1-10 000		Number of ticks per revolution	Ushort	4 <sup>1</sup>	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 36: SETWHEELPARAMETERS Input**

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

---

### B.4.47 SOFTLOADFILE Loads a file to the SPAN-SE

Use this command to load a hexadecimal file from an external source to reprogram the SPAN-SE.

The SOFTLOADFILE command reprograms the system firmware using a NovAtel formatted input message. This allows anybody to reprogram the system in an embedded environment where using WinLoad isn't practical or possible.

The SoftLoad feature allows reprogramming of the SPAN-SE firmware, any of the three user applications and the firmware on the OEMV2 and OEMV3.

#### Abbreviated ASCII Syntax:

Message ID: 1302

SOFTLOADFILE Storage File [Dest]

#### Abbreviated ASCII Example:

```
SOFTLOAD SD SCD000300RN0000.hex SPAN
```

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	-	H	0
2	Storage	SD	0x00	The type of storage that contains the file to load. Filenames have a 64 character limit	Enum	4	H
		USBSTICK	0x01				
		RAMDRIVE	0x02				
3	File			The hex file to load	Char[64]	64	H+4
4	Dest	SPAN	0	SPAN destination device	Enum	4	H+68
		OEMV3	1	OEMV3 destination device			
		OEMV2	2	OEMV2 destination device			

**B.4.48 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

Abbreviated ASCII Example:

SOFTPOWER NOW

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	-	H	0
2	priority	NOW	1	Power down the SPAN-SE immediately	Enum	4	H

---

## B.4.49 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 339*, 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*. The OEMV-3 card and SPAN-SE will reset after receiving the AUTH or MODEL command. We recommend that you contact *NovAtel Customer Support* for assistance in doing this, see *page 20* 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-upgradable 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.



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	SPANAUTH header	-	-	This field contains the command name or the message header depending on whether the command is abbreviated ASCII, ASCII or binary	-	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]	Variable <sup>1</sup>	Variable
9	date	Numeric	Null terminated	Expiry date entered as yymmdd in decimal	String [max. 7]	Variable <sup>1</sup>	Variable

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



### B.4.50 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 199*. 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 339*. Use the VERSION log to output the active model, see *page 346*.



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	-	H	0
2	model	Max 16 character null-terminated string (including the null)		SPAN model name	String [max. 16]	Variable <sup>1</sup>	Variable
3	Reserved				Ulong	4	Variable

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

---

### **B.4.51 TAGNEXTMARK Add a new camera event**

Use this command with the TAGGEDMARKxPVA log. TAGNEXTMARK inserts a unique identifier into the markpva to associate with a particular event.

#### **Abbreviated ASCII Syntax**

**Message ID: 1257**

TAGNEXTMARK Mark TagNum

#### **Abbreviated ASCII Example:**

```
TAGNEXTMARK MARK1 1000
TAGNEXTMARK MARK2 3465
```

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	-	H	0
2	Mark	Mark1	1	The input event mark	Enum	4	H
		Mark2	2				
		Mark3	3				
		Mark4	4				
3	TagNum			The ID tag that you want to associate with the next input event on the selected mark	Ulong	4	H+4
4	xxxx			32-bit CRC	Hex	4	H+8
5	[CR][LF]			Sentence Terminator (ASCII only)	-	-	-

## B.4.52 TIMEDEVENTPULSE Add a new camera event

Use this command to add a new camera event to the system. TIMEDEVENTPULSE will send a pulse on the sensor MARK output at the selected GPS time and set the next trigger on the sensor MARK input to be tagged with an event ID.

### Abbreviated ASCII Syntax

Message ID: 1337

TIMEDEVENTPULSE Sensors GPSWeek GPSSeconds EventID

### Abbreviated ASCII Example:

```
TIMEDEVENTPULSE -1 1617 418838 100
```

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	-	H	0
2	SensorID	ALL SENSORS	-1	The sensor(s) affected by the trigger command. The decimal representation of the combination of bits 0-2 can be used to select a combination of active sensors. (eg. <b>5</b> [101] will select sensors 1 and 3)	Long	4	H
		SENSOR1	1				
		SENSOR2	2				
		SENSOR3	4				
3	GPSWeek	0 – 4294967295		The GPS week that triggers the event.	Ulong	4	H+4
4	GPSSeconds	0 – 604800		The GPS seconds that trigger the event.	Double	8	H+8
5	EventID	0 – 4294967295		The event's identifier, used to tag the TAGGEDMARKxPVA logs if a sensor input is enabled. Default is 0	Ulong	4	H+16

### B.4.53 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 <i>Table 39, Binary Message Header Structure</i> , on page 218)	This field contains the message header.	-	H	0
2	port	See <i>Table 23, COM Serial Port Identifiers</i> , on page 136	Port to which log is being sent. Default is 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	-	H	0
2	port	See <i>Table 23, COM Serial Port Identifiers</i> , on page 136		Port to which log is being sent. Default is THISPORT	Enum	4	H
3	message	Message Name	N/A	Message name of log to disable	ULong	4	H+4

**B.4.54 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	-	H	0
2	port	See Table 23, COM Serial Port Identifiers, on page 136		Port to clear. Default is 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.4.55 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 3.4.8, Vehicle to SPAN Frame Angular Offsets Calibration Routine* starting on *page 63* for more details. For more information on reference frames, see *Section 3.1, Definition of Reference Frames within SPAN* starting on *page 44*. RVBCALIBRATE command information is on *page 174*.

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 131*, 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.

To estimate your VEHICLEBODYROTATION angles, follow these steps:

1. Imagine a frame coincident with the vehicle frame.
2. Rotate your imagined frame about the vehicle Z-axis so that the Y-axis of your frame is approximately aligned (coplanar) with the assumed Y-axis of the IMU enclosure, as defined by your IMU orientation in *Table 34* on *page 183*. This angle is the gamma-angle in the command and follows the right-hand rule for sign correction.
3. Rotate about the vehicle X-axis. This angle is the alpha-angle in the command.
4. Finally, rotate about the vehicle Y-axis. This angle is the beta-angle in the command. Your imagined frame should now be coincident with the assumed IMU enclosure frame.



Enter rotation angles in degrees.



For further information, refer to *Section 3.4.7, Vehicle to Body Rotations* on *page 60*.

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.4.3, APPLYVEHICLEBODYROTATION Enable vehicle to body rotation* starting on *page 131*.

VEHICLEBODYROTATION  $\alpha$   $\beta$   $\gamma$  [ $\delta\alpha$ ] [ $\delta\beta$ ] [ $\delta\gamma$ ]

## Abbreviated ASCII Example:

VEHICLEBODYROTATION 0 0 90 0 0 5

Field	Field Type	Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	X Angle	Right hand rotation about SPAN computation frame X-axis in degrees	Double	8	H
3	Y Angle	Right hand rotation about SPAN computation frame Y-axis in degrees	Double	8	H+8
4	Z Angle	Right hand rotation about SPAN computation frame Z-axis in degrees	Double	8	H+16
5	X Uncertainty	Uncertainty of X rotation in degrees. Default is 0	Double	8	H+24
6	Y Uncertainty	Uncertainty of Y rotation in degrees. Default is 0	Double	8	H+32
7	Z Uncertainty	Uncertainty of Z rotation in degrees. Default is 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 *APN-037 Vehicle to Body Rotations* available on our website at [www.novatel.com](http://www.novatel.com) through *Support | Knowledge and Learning*.



## B.4.56 WHEELVELOCITY Wheel velocity for INS augmentation

The WHEELVELOCITY command is used to input wheel sensor data into the SPAN-SE receiver.

### Abbreviated ASCII Syntax:

Message ID: 504

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

### 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

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	-	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 per second	Ushort	2	H+4
5	Reserved				Ushort	2	H+6
6	fwheel vel			Float wheel velocity in ticks per second	Float	4	H+8
7	Reserved				Ulong	4	H+12
8					Ulong	4	H+16
9	ticks/s			Cumulative number of ticks per second	Ulong	4	H+20

Refer also to our application note *APN-036 Using a Wheel Sensor with SPAN*, available on our website at [www.novatel.com](http://www.novatel.com) through *Support | Knowledge and Learning*.

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 want to use wheel velocities in post-processing, complete fields 4 and 6.

## Appendix C 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, ICOM1, ICOM2, ICOM3 or ICOM4. 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.

---

## C.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.

---

## C.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 inspvash 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 center to the GNSS antenna phase center, 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 D 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 the full rate of the 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 D.1.4, ASCII and Binary Logs with Short Headers* on page 221.
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 54 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.

## D.1 Log Types

Refer to the LOG command, see *page 159*, 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 are collected on a regular schedule, they do not output the most current data. The data is delayed until the scheduled output time. The data in polled logs is generated on demand. An example of a polled log is RXCONFIG. 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 37: Log Type Triggers**

Type	Recommended Trigger	Illegal Trigger
Synch	ONTIME	ONNEW, ONCHANGED
Asynch	ONCHANGED	-
Polled	ONCE or ONTIME <sup>1</sup>	ONNEW, ONCHANGED

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

See *Section D.1.6, Message Time Stamps* on *page 223* 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 MARKxTIME or MARKxPVA 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.

## D.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 first character for each record is #.
2. Each log or command is of variable length depending on amount of data and format.
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 an \* 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* available on our website at [www.novatel.com](http://www.novatel.com) through *Support | Firmware/Software and Manuals | Manuals and Documentation Updates | OEMV™ Family* for 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 38: 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. See <i>Table 40</i> on <i>page 220</i> .	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

*Continued on the following page*

Field #	Field Name	Field Type	Description	Ignored on Input
6	GPS Time Status	Enum	This value indicates the quality of the GPS time (see <i>Table 43, GPS Time Status</i> on <i>page 221</i> )	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 <i>Table 76, SPAN Receiver Status</i> on <i>page 334</i> )	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,8b055
0a189aa6ff925386228f97eabf9c8047e34a70ec5a10e486e794a7a,8b0550a18a2effc2f80061c2f
ffc267cd09f1d5034d3537affa28b6ff0eb*7a22f279
```



---

### **D.1.2 Abbreviated ASCII**

This message format is designed to make the entering and viewing of commands and logs by the user as simple as possible. The data is represented as simple ASCII characters separated by spaces or commas and arranged in an easy to understand fashion. There is also no 32-bit CRC for error detection because it is meant for viewing by the user.

#### **Example Command:**

```
log com1 loglist
```

#### **Resultant Log:**

```
[COM1]LOGLIST COM1 0 99.0 FINESTEERING 1635 487565.230 00000000 0000 406
<      5
<      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 LOGLIST ONCE 0.000000 0.000000 NOHOLD
```

As you can see the array of five logs are offset from the left hand side and start with '<'.

### D.1.3 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, 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	Three 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 39*.

**Table 39: 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 Length	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 that you can find as part of each log description in this chapter	2	4	N

*Continued on the following page*

Field #	Field Name	Field Type	Description	Binary Bytes	Binary Offset	Ignored on Input
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 <i>Table 40 on page 220</i> (decimal values greater than 16 may be used) [lower 8 bits only] <sup>1</sup>	1	7	N <sup>2</sup>
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
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	GPS Time Status	Enum	Indicates the quality of the GPS time (see <i>Table 40 on page 220</i> )	1 <sup>3</sup>	13	N <sup>4</sup>
12	Week	Ushort	GPS week number	2	14	N <sup>4</sup>
13	ms	GPSec	Milliseconds from the beginning of the GPS week	4	16	N <sup>4</sup>
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 between 0 – 65535 that represents the receiver software build number	2	26	Y

1. The 8 bit size means that you will only see 0xA0 to 0xBF when the top bits are dropped from a port value greater than 8 bits; for example ASCII port USB1 will be seen as 0xA0 in the binary output.

**Table 40: Detailed Serial Port Identifiers**

ASCII Port Name	Hex Port Value	Decimal Port Value	Description
COM1	20	32	COM port 1
COM2	40	64	COM port 2
COM3	60	96	COM port 3
SPECIAL	A0	160	Unknown port
THISPORT	C0	192	Current COM port
FILE	E	224	User-specified file destination, SD card
USB1	5A0	1440	USB port 1
ICOM1	0CA0	3232	ICOM port 1
ICOM2	10A0	4256	ICOM port 2
ICOM3	11A0	4512	ICOM port 3
ICOM4	15A0	5536	ICOM port 4
COM4	0EA0	3744	COM port 4
XCOM1	1A0	416	Internal virtual COM channels for use with SPAN API.
XCOM2	2A0	672	Internal virtual COM channels for use with SPAN API.
XCOM3	9A0	2464	Internal virtual COM channels for use with SPAN API.

### D.1.4 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 41 and 42*). For the message header structure of OEMV-3 regular Binary and ASCII logs, please refer to the *OEMV Family Firmware Reference Manual*.

**Table 41: 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	Seconds	Ulong	Seconds from the beginning of the GPS week

**Table 42: 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

### D.1.5 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 43*:

**Table 43: GPS Time Status**

GPS Time Status (Decimal)	GPS Time Status <sup>1</sup> (ASCII)	Description
20	UNKNOWN	Time validity is unknown.
60	APPROXIMATE	Time is set approximately.
100	COARSE	This time is valid to coarse precision.

*Continued on the following page*

GPS Time Status (Decimal)	GPS Time Status <sup>1</sup> (ASCII)	Description
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 D.1.6, Message Time Stamps* on *page 223*

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  $\pm 10$  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 adjust again, this time to an accuracy of  $\pm 1$  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.

---

### **D.1.6 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, GPSEPHM) 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.

---

## D.1.7 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 ONTIME 0.05      [20 Hz]
LOG PSRPOS ONTIME 0.1       [10 Hz]
LOG PSRPOS ONTIME 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]
```



---

## D.2 NMEA Standard Logs

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

<b>GPALM</b>	<b>ALMANAC DATA</b>
<b>GPGGA</b>	<b>GLOBAL POSITION SYSTEM FIX DATA AND UNDULATION</b>
<b>GPGLL</b>	<b>GEOGRAPHIC POSITION</b>
<b>GPGRS</b>	<b>GPS RANGE RESIDUALS FOR EACH SATELLITE</b>
<b>GPGSA</b>	<b>GPS DOP AND ACTIVE SATELLITES</b>
<b>GPGST</b>	<b>PSEUDORANGE MEASUREMENT NOISE STATISTICS</b>
<b>GPGSV</b>	<b>GPS SATELLITES IN VIEW</b>
<b>GPZDA</b>	<b>UTC TIME AND DATE</b>
<b>PASHR</b>	<b>PITCH, ROLL AND HEADING INFORMATION</b>

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 *Standards and References*, available on our website at [www.novatel.com](http://www.novatel.com) through *Support | Knowledge and Learning*. 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	III.II	Fixed/Variable length field: degrees minutes.decimal - 2 fixed digits of degrees, 2 fixed digits of mins and a <u>variable</u> 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	yyyy.yy	Fixed/Variable length field: degrees minutes.decimal - 3 fixed digits of degrees, 2 fixed digits of mins and a <u>variable</u> 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 <u>variable</u> 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", "III.II", "x", "yyyy.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

## D.3 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.

INS-specific commands and logs provide attitude data such as roll, pitch and azimuth.



### Logging Restriction Important Notice

Note the following when configuring your SPAN system:

1. Recommend logging rates:

RANGECMPB      maximum 20 Hz

RANGE    B      maximum 10 Hz

TRACKSTATB    maximum 10 Hz

No more than one of the above logs should be output at one time. We do not recommend ASCII for these logs. These recommendations apply to all large OEMV logs.

2. When logging to the SD memory card, if you log at more than 40 kb/sec, you will get a warning in RXSTATUS.
3. BESTPOS, BESTGPSPOS, GPGGA, GPVTG and GPGLL logs are available at <1, 1 and 5 Hz only on SPAN-SE with an inertial model.
4. When requesting high-rate data over COM1, COM2, COM3 or COM4, be careful not to overrun the baud rate. Refer to *Appendix B.4.5, COM Port configuration control* starting on *page 135* for further information.
5. 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 319*.
6. To collect wheel sensor information, useful in post-processing, the TIMEDWHEELDATA log should only be used with the ONNEW trigger. See also *page 343* 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.

### D.3.1 APPLICATIONSTATUS Display installed and running applications

Use this log to query the card to display a list of installed or running applications.

Message ID: 520

Log Type: ASYNC

#### Recommended Input:

```
log applicationstatusa once
```

#### ASCII Example:

```
#APPLICATIONSTATUSA,COM1,0,98.0,UNKNOWN,0,0.000,00000000,0000,407;4,14,TRUE,0
0000000,00000000,"SPAN API","14","May 3
2011","15:09:46",11,FALSE,00000000,00000000,"UserApp1","1.000","2011/May/
05","15:00:56",0,FALSE,00000000,00000000,"","","11,FALSE,00000000,000000
00,"UserApp3","1.000","2011/Mar/28","13:10:27"*39144e87
```

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	-	H	0
2	APIVer	0 – MAXLONG		The APU1 version with installed firmware	Ulong	4	H+4
3	RunStat	TRUE or FALSE		Is the application currently running?	Boolean	4	H+8
4	BaseAddr	0 – 0xFFFFFFFF		The running application's base memory address	Hex	4	H+12
5	Size	0 – 0xFFFFFFFF		The application's size in memory	Hex	4	H+16
6	Name			The application's name	Char[16]	16	H+20
7	Version			Application version string	Char[16]	16	H+36
8	CompDate			The date on which the application was compiled	Char[12]	12	H+52
9	CompTime			The time at which the application was compiled	Char[12]	12	H+64
10				Next component offset			**
Variable	xxxx			32-bit CRC	Hex	4	**
Variable	CRLF			Sentence terminator (ASCII only)			

\*\* – offset = H+(component number x 76)

### D.3.2 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.

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*.

When in INS mode, the position is calculated at the antenna phase center.



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

**BESTGPSPOS Message ID: 423**

**BESTPOS Message ID: 42**

**Log Type: Synch**

#### 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
```

**Table 44: 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

*Continued on the following page*

Position Type (binary)	Position Type (ASCII)	Description
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
20	OMNISTAR	OmniSTAR VBS position (L1 sub-meter) <sup>1</sup>
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. <sup>2</sup>
52	INS	INS calculated position corrected for the antenna <sup>2</sup>
53	INS_PSRSP	INS pseudorange single point solution - no DGPS corrections <sup>2</sup>
54	INS_PSRDIFF	INS pseudorange differential solution <sup>2</sup>
55	INS_RTKFLOAT	INS RTK floating point ambiguities solution <sup>2</sup>
56	INS_RTKFIXED	INS RTK fixed ambiguities solution <sup>2</sup>
57	INS_OMNISTAR	INS OmniSTAR VBS position (L1 sub-meter) <sup>1</sup>
58	INS_OMNISTAR_HP	INS OmniSTAR high precision solution <sup>1</sup>
59	INS_OMNISTAR_XP	INS OmniSTAR extra precision solution <sup>1</sup>
64	OMNISTAR_HP	OmniSTAR high precision <sup>1</sup>
65	OMNISTAR_XP	OmniSTAR extra precision <sup>1</sup>
66	CDGPS	Position solution using CDGPS corrections <sup>1</sup>

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 45: 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 <sup>1</sup>
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 <i>Table 45 on page 231</i>	Enum	4	H
3	Pos Type	Position type, see <i>Table 44 on page 229</i>	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 <i>OEMV Family Firmware Reference Manual</i> )	Enum	4	H+36
9	Lat $\sigma$	Latitude standard deviation	Float	4	H+40
10	Lon $\sigma$	Longitude standard deviation	Float	4	H+44
11	Hgt $\sigma$	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	#solnSVs	Number of satellite solutions used in solution	Uchar	1	H+65
17	#L1	Number of GPS and GLONASS L1 ranges above the RTK mask angle	Uchar	1	H+66
18	#L2	Number of GPS and GLONASS 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)	-	-	-



**Table 46: Signal-Used Mask**

Bit	Mask	Description
0	0x01	GPS L1 used in Solution
1	0x02	GPS L2 used in Solution
2	0x04	GPS L5 used in Solution
3	0x08	Reserved
4	0x10	GLONASS L1 used in Solution
5	0x20	GLONASS L2 used in Solution
6-7	0x40-0x80	Reserved

**Table 47: Extended Solution Status**

Bit	Mask	Description
0	0x01	AdVance RTK Verified 0: Not Verified 1: Verified
1-3	0x0E	Pseudorange Iono Correction 0: Unknown <sup>1</sup> 1: Klobuchar Broadcast 2: SBAS Broadcast 3: Multi-frequency Computed 4: PSRDiff Correction 5: NovAtel Blended Iono value
4-7	0xF0	Reserved

1. Unknown can indicate that the Iono Correction type is None or that the default Klobuchar parameters are being used.

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	BESTPOS header	Log header		H	0
2	sol stat	Solution status, see <i>Table 45 on page 231</i>	Enum	4	H
3	pos type	Position type, see <i>Table 44 on page 229</i>	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 - the relationship between the geoid and the ellipsoid (m) of the chosen datum <sup>1</sup>	Float	4	H+32
8	datum id#	Datum ID number	Enum	4	H+36
9	lat $\sigma$	Latitude standard deviation	Float	4	H+40
10	lon $\sigma$	Longitude standard deviation	Float	4	H+44
11	hgt $\sigma$	Height standard deviation	Float	4	H+48
12	stn id	Base station ID	Char[4]	4	H+52
13	diff_age	Differential age in seconds	Float	4	H+56
14	sol_age	Solution age in seconds	Float	4	H+60
15	#SVs	Number of satellite vehicles tracked	Uchar	1	H+64
16	#solnSVs	Number of satellite vehicles used in solution	Uchar	1	H+65
17	#ggL1	Number of GPS and GLONASS L1 used in RTK solution	Uchar	1	H+66
18	#ggL1L2	Number of GPS and GLONASS L1 and L2 used in RTK solution	Uchar	1	H+67
19	Reserved		Uchar	1	H+68
20	ext sol stat	Extended solution status (see <i>Table 47, Extended Solution Status on page 233</i> )	Hex	1	H+69
21	Reserved		Hex	1	H+70
22	sig mask	Signals used mask - if 0, signals used in solution are unknown (see <i>Table 46 on page 233</i> )	Hex	1	H+71
23	xxxx	32-bit CRC (ASCII and Binary only)	Hex	1	H+72
24	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

1. When using a datum other than WGS84, the undulation value also includes the vertical shift due to differences between the datum in use and WGS84

---

### **D.3.3 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 less than 1 Hz, 1 Hz or 5 Hz. Higher rate velocity information is available in the INSVEL, INSPVA or INSSPD logs.

**BESTGPSVEL Message ID: 506**

**BESTVEL Message ID: 99**

**Log Type: Synch**

#### **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
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	Log header	Log header	-	H	0
2	Sol Status	Solution status, see <i>Table 45</i> on <i>page 231</i>	Enum	4	H
3	Vel Type	Velocity type, see <i>Table 44</i> on <i>page 229</i>	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	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)	-	-	-

### D.3.4 BESTLEVERARM/BESTLEVERARM2 IMU to Antenna Lever Arm

The BESTLEVERARM log contains the distance between the IMU and the primary GNSS antenna in the IMU enclosure frame and its associated uncertainties. If the you enter the lever arm through the SETIMUTOANTOOFFSET command, see *page 184*, these values are reflected in this log. When the lever arm calibration is complete, see the LEVERARMCALIBRATE command on *page 157*, the solved values are also output in this log.

The BESTLEVERARM2 log contains the distance between the IMU and the secondary GNSS antenna in the IMU enclosure frame. Currently the secondary lever arm cannot be calibrated so must be entered using the SETIMUTOANTOOFFSET2 command.

The values in the BESTLEVERARM and BESTLEVERARM2 logs are also available in the IMUTOANTOOFFSETS log, see *page 273*.

The default X (pitch), Y (roll) and Z (azimuth) directions of the inertial frame are clearly marked on the IMU, see *Figure 43* on *page 181*.

**BESTLEVERARM Message ID: 674**

**BESTLEVERARM2 Message ID: 1256**

**Log Type (s): Asynch**

#### Recommended Input:

```
log bestleverarma unchanged
```

#### 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
```

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 <i>Table 34, Full Mapping Definitions</i> on <i>page 183</i>	Integer	4	H+48
9	xxxx	32-bit CRC	Hex	4	H+52
10	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

---

### D.3.5 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 examples:

```
#COMCONFIGA,COM1,0,98.5,FINESTEERING,1635,320272.363,00000000,0000,407;10,COM
1,9600,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,COM2,9600,N,8,1,N,OFF,ON,NOVATEL,NOV
ATEL,ON,COM3,9600,N,8,1,N,OFF,ON,NOVATEL,NOVATEL,ON,COM4,9600,N,8,1,N,OFF,ON,
NOVATEL,NOVATEL,ON,IMU,9600,N,8,1,N,OFF,OFF,IMU,IMU,OFF,USB1,12000000,N,0,0,N
,OFF,OFF,NOVATEL,NOVATEL,ON,ICOM1,10000000,N,0,0,N,OFF,OFF,NOVATEL,NOVATEL,ON
,ICOM2,10000000,N,0,0,N,OFF,OFF,NOVATEL,NOVATEL,ON,ICOM3,10000000,N,0,0,N,OFF
,OFF,NOVATEL,NOVATEL,ON,ICOM4,10000000,N,0,0,N,OFF,OFF,NOVATEL,NOVATEL,ON*a8e
10684
```

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 <i>Table 23</i> on <i>page 136</i>	Enum	4	H+4
4	baud	Communication baud rate	Ulong	4	H+8
5	parity	See <i>Table 24</i> on <i>page 136</i>	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 <i>Table 25</i> on <i>page 136</i>	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 <i>Table 29, Serial Port Interface Modes</i> on <i>page 150</i>	Enum	4	H+36
12	tx type	The status of the transmit interface mode, see <i>Table 29</i> on <i>page 150</i>	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 number x 44)				
15	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+4+(#port x 44)
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

D.3.6    **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 138* 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 <i>Table 23, COM Serial Port Identifiers</i> on <i>page 136</i>	Enum	4	H+4
4	protocol	Port protocol, see <i>Table 48</i> below	Enum	4	H+8
5	next record offset = H + 4 + (record number x 8)				
6	xxxx	32-bit CRC	Hex	4	H+4+ (#recs x 8)
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

**Table 48: Port Protocol**

ASCII	Binary	Description
RS232	0	RS-232 mode
RS422	1	RS-422 mode



### D.3.7 **CORRIMUDATA and CORRIMUDATAS** *Corrected IMU measurements*

The CORRIMUDATA(S) log contains the RAWIMU data corrected for gravity, earth's rotation, and accelerometer and gyroscope biases. The values in this log are instantaneous, incremental values, in units of radians for the attitude rate and m/s for the accelerations. To get the full attitude rate and acceleration values, you must multiply the values in the CORRIMUDATA(S) log by the data rate of your IMU in Hz.



The short header format, CORRIMUDATAS, is recommended, as it is for all high data rate logs.

CORRIMUDATA(S) can be logged with the ONTIME trigger, up to the full data rate of the IMU.



Since the CORRIMUDATA(S) values are instantaneous, if you log at a rate less than full data rate of the IMU, you will receive the corrected IMU data at the epoch closest to the requested time interval.

If your IMU is mounted with the z axis, as marked on the enclosure, pointed up, the SPAN computation frame is the same as the IMU enclosure frame. The x, y, and z axes referenced in this log are of the SPAN computational frame by default. For more information on how the SPAN computational frame relates to the IMU enclosure frame, see *Section 3.1, page 44*, and the SETIMUORIENTATION command on *page 179*. If the APPLYVEHICLEBODYROTATION command has been enabled (see *page 131*), the values in CORRIMUDATA(S) logs will be in the vehicle frame, not the SPAN computation frame.

**Message ID: 812 and 813**

**Log Type: Synch**

#### **Recommended Input:**

```
log corrimudatasb ontime 0.01
```

#### **Example log:**

```
%CORRIMUDATASA,1581,341553.000;1581,341552.997500000,-0.000000690,-  
0.000001549,0.000001654,0.000061579,-0.000012645,-0.000029988*770c6232
```

#### **Recommended Input:**

```
log direnta ontime 30
```

## ASCII Example:

```
#DIRENTA,USB1,5,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SDLOG.GP
S",2019951634,1972610,20110323,220136*66b5b93c

#DIRENTA,USB1,4,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_0.L
OG",52700,52,20110324,220806*03fda3c9

#DIRENTA,USB1,3,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_1.L
OG",28570,28,20110324,220954*fb3e911e

#DIRENTA,USB1,2,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_2.L
OG",200454,196,20110510,213028*a9795169

#DIRENTA,USB1,1,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_3.L
OG",134452,132,20110510,213902*89e77dd0

#DIRENTA,USB1,0,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_4.L
OG",15008,15,20110510,214156*49df871c
```

Field #	Field Type	Description	Format	Binary Bytes	Binary Offset
1	header	Log header	-	H	0
2	Week	GPS week	ULONG	4	H+
3	Seconds	GPS seconds from week start	DOUBLE	8	H+4
4	PitchRate	About X-axis rotation	DOUBLE	8	H+12
5	RollRate	About Y-axis rotation	DOUBLE	8	H+20
6	YawRate	About Z-axis rotation (right handed)	DOUBLE	8	H+28
7	LateralAcc	INS Lateral Acceleration (along X-axis)	DOUBLE	8	H+36
8	LongitudinalAcc	INS Longitudinal Acceleration (along Y-axis)	DOUBLE	8	H+44
9	VerticalAcc	INS Vertical Acceleration (along Z-axis)	DOUBLE	8	H+52
10	xxxx	32-bit CRC	Hex	4	H+60T
11	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

---

### D.3.8 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. Note that the file size information in this log will only update once per second.

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:

$$\begin{aligned}\text{Ulong Year} &= (\text{Ulong})(\text{DateOfLastChange} / 10000) \\ \text{Ulong Month} &= (\text{Ulong})((\text{DateOfLastChange} - (\text{Year} * 10000)) / 100) \\ \text{Ulong Day} &= (\text{Ulong})(\text{DateOfLastChange} - (\text{Year} * 10000) - (\text{Month} * 100))\end{aligned}$$

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:

$$\begin{aligned}\text{Ulong Hour} &= (\text{Ulong})(\text{TimeOfLastChange} / 10000) \\ \text{Ulong Minutes} &= (\text{Ulong})((\text{TimeOfLastChange} - (\text{Hour} * 10000)) / 100) \\ \text{Ulong Seconds} &= (\text{Ulong})(\text{TimeOfLastChange} - (\text{Hour} * 10000) - (\text{Minutes} * 100))\end{aligned}$$

**Message ID = 159**

**Log Type: Polled**

#### Recommended Input:

```
log direnta ontime 30
```

#### ASCII Example:

```
#DIRENTA,USB1,5,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SDLOG.GP
S",2019951634,1972610,20110323,220136*66b5b93c
#DIRENTA,USB1,4,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_0.L
OG",52700,52,20110324,220806*03fda3c9
#DIRENTA,USB1,3,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_1.L
OG",28570,28,20110324,220954*fb3e911e
#DIRENTA,USB1,2,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_2.L
OG",200454,196,20110510,213028*a9795169
#DIRENTA,USB1,1,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_3.L
OG",134452,132,20110510,213902*89e77dd0
#DIRENTA,USB1,0,98.5,FINESTEERING,1635,320403.182,00000001,0000,394;"SPAN_4.L
OG",15008,15,20110510,214156*49df871c
```

---

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

---

### **D.3.9    *EXTHDGOFFSET Log the Angular Offset***

The EXTHDGOFFSET message will be available on the system after you enter both lever arms (refer to *Section 4.5 on page 79*). The angular offsets between the dual antenna baseline and the SPAN computation frame will be computed internally and be available for output via the EXTHDGOFFSET log. For message structure, refer to *Section B.4.11 on page 144*.

### D.3.10 GIMBALLEDPVA Display gimballed position

Use this log to view re-calculated gimballed position, velocity and attitude whenever a new INPUTGIMBALANGLE command is received.

**NOTE:** The log is not output until the INS alignment is complete.

**Message ID: 1321**

**Log Type: Asynch**

#### Recommended Input:

```
log gimballedpvaa onnew
```

#### ASCII Example:

```
#GIMBALLEDPVAA,COM1,0,93.5,FINESTEERING,1635,320568.514,00000000,0000,407;163  
5,320568.514000000,51.116376614,-114.038259915,1046.112025828,-0.000291756,-  
0.000578067,0.030324466,-0.243093917,-  
0.127718304,19.495023227,INS_ALIGNMENT_COMPLETE*32fbb61b
```

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	WGS84 latitude in degrees	Double	8	H+12
5	Longitude	WGS84 longitude in degrees	Double	8	H+20
6	Height	WGS84 ellipsoidal height	Double	8	H+28
7	NorthVelocity	Velocity in a northerly direction	Double	8	H+36
8	EastVelocity	Velocity in an easterly direction	Double	8	H+44
9	UpVelocity	Velocity in an upward direction	Double	8	H+52
10	Roll	Right handed rotation from local level around the 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	Right handed rotation from local level around Z-axis in degrees	Double	8	H+76
13	Status	INS status	Enum	4	H+84
6	xxxx	32-bit CRC	Hex	4	H+88
7	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

---

### **D.3.11 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 15 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 10785 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 <sup>4</sup>	Four-year interval number starting from 1996	Uchar	1 <sup>1</sup>	H+21 <sup>a</sup>
7	$\tau_{GPS}$	GPS time scale correction to UTC(SU) given at beginning of day N <sup>4</sup> , in seconds	Double	8	H+24
8	N <sup>A</sup>	GLONASS calendar day number within a four year period beginning since the leap year, in days	Ushort	2 <sup>a</sup>	H+32 <sup>a</sup>
9	$\tau_C$	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



### D.3.12 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-only 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](http://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, 0.0000000000000000,  
-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*a02d5ef15  
#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, 0.0000000000000000, -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 49: GLONASS Ephemeris Flags Coding**

N 0		Nibble Number							
4	3	2	1	0	Bit	Description	Range Values	Hex Value	
15	14	13	12	11	10	1b P1 FLAG- TIME INTERVAL BETWEEN ADJACENT ISSUE VALUES	See Table below (Table 50)	00000001	
						= 0			
						1			00000002
						2	P2 FLAG- ODDNESS OR EVENNESS OF ISSUE 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			

**Table 50: 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	GLOEPHEMERIS header	Log header		H	0
2	sloto	Slot information offset - PRN identification (Slot + 37). This is also called SLOTO in Connect	Ushort	2	H
3	frequ	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 <i>page 249</i>	Ushort	2	H+16
10	Reserved			1	H+18

*Continued on the following page*

Field#	Field type	Data Description	Format	Binary Bytes	Binary Offset
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
19	vel z	Z coordinate for satellite velocity at reference time (PZ-90.02), in meters per second	Double	8	H+68
20	LS acc x	X coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters per second per second	Double	8	H+76
21	LS acc y	Y coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters per second per second	Double	8	H+84
22	LS acc z	Z coordinate for lunisolar acceleration at reference time (PZ-90.02), in meters per second per second	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 per 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 <i>Table 49, GLONASS Ephemeris Flags Coding</i> on page 250	Ulong	4	H+140
31	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+144
32	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### D.3.13 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,0,0404f226600000000000065,0*17f3dd17  
...  
#GLORAWEPHEMA,COM1,0,47.0,SATTIME,1340,398653.000,00000000,332d,2020;  
41,13,0,1340,398653.078,4,  
0108d812532805bfa1cd2c,0,0208e0a36e8e0952b111da,0,  
03c02023b68c9a32410958,0,0401fda440000000000002a,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 Connect	Ushort	2	H
3	freqo	Frequency channel offset in the range 0 – 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.

---

### D.3.14 GPALM Almanac Data

This National Marine Electronics Association (NMEA) log, see also *Section D.2, NMEA Standard Logs* on page 225, 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 website. 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 <sup>a</sup> .	x.x	653
6	SV h1th	SV health, bits 17-24 of each almanac page <sup>b</sup>	hh	00
7	ecc	e, eccentricity <sup>c d</sup>	hhhh	3EAF
8	alm ref time	toa, almanac reference time <sup>c</sup>	hh	87
9	incl angle	(sigma) <sub>i</sub> , inclination angle <sup>c</sup>	hhhh	OD68
10	omegadot	OMEGADOT, rate of right ascension <sup>c</sup>	hhhh	FD30
11	rt axis	(A) <sup>1/2</sup> , root of semi-major axis <sup>c</sup>	hhhhhh	A10CAB
12	omega	omega, argument of perigee <sup>c e</sup>	hhhhhh	6EE732
13	long asc node	(OMEGA) <sub>o</sub> , longitude of ascension node <sup>c</sup>	hhhhhh	525880
14	M <sub>o</sub>	Mo, mean anomaly <sup>c</sup>	hhhhhh	6DC5A8
15	af0	af0, clock parameter <sup>c</sup>	hhh	009
16	af1	af1, clock parameter <sup>c</sup>	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 Sub-frame 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

---

### **D.3.15 GPGGA GPS Fix Data and Undulation**

This NMEA log provides time, position and fix-related data of the GNSS receiver. See also *Section D.2, NMEA Standard Logs* on page 225. For more on precision of NMEA logs, see *Table 52, Position Precision of NMEA Logs* on page 257.

GPGGA, like BESTPOS, contains the best available position, from either GNSS only, or GNSS/INS.

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
```

Field	Structure	Field Description	Symbol	Example
1	\$GPGGA	Log header		\$GPGGA
2	utc	UTC time status 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	GPS qual	GPS Quality indicator 0 = fix not available or invalid 1 = GPS fix 2 = C/A differential GPS, OmniSTAR HP, OmniSTAR XP or OmniSTAR VBS 4 = RTK fixed ambiguity solution (RT2) 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 <sup>1</sup>	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 = metres)	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 = metres)	M	M
14	age	Age of Differential GPS data (in seconds) <sup>2</sup>	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). This indicator can be customized using the GGAQUALITY command.
2. The maximum age reported here is limited to 99 seconds.



### D.3.16 GPGLL Geographic Position

This NMEA log provides altitude and longitude of the present vessel position, time of position fix, and status. See also *Section D.2, NMEA Standard Logs* on page 225.

GPGLL, like BESTPOS, contains the best available position, from either GNSS only, or GNSS/INS.

*Table 52* on page 257 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 164*, 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 gppll 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 51: NMEA Positioning System Mode Indicator**

Mode	Indicator
A	Autonomous
D	Differential
E	Estimated (dead reckoning) mode
M	Manual input
N	Data not valid

**Table 52: Position Precision of NMEA Logs**

NMEA Log	Latitude (# of decimal places)	Longitude (# of decimal places)	Altitude (# of decimal places)
GPGBA	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)	lll.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 <i>Table 51</i> on <i>page 257</i>	a	A
9	*xx	Checksum	*hh	*1B
10	[CR][LF]	Sentence terminator		[CR][LF]

### D.3.17 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 D.2, NMEA Standard Logs* on page 225.

Under mode 0, residuals output in this log are used to update the position solution output in the GPGLL message. Under mode 1, the residuals are re-computed after the position solution in the GPGLL message is computed. The receiver computes range residuals in mode 1. An integrity process using GPGRS would also require GPGLL (for position fix data), GPGLS (for DOP figures), and GPGLV (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  $\pm 99.9$ , then the decimal part is dropped. Maximum value for this field is  $\pm 999$ . The sign of the range residual is determined by the order of parameters used in the calculation as follows:

$$\text{range residual} = \text{calculated range} - \text{measured range}$$

2. There is no residual information available from the OmniSTAR HP/XP service, so the GPGRS contains the pseudorange position values when using it. Residual information is available for the OmniSTAR VBS.

**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 (a priori) (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 GPGLS	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]

---

### D.3.18 GPGSA GPS DOP and Active Satellites

This NMEA log provides GPS receiver operating mode, satellites used for navigation and DOP values. See also *Section D.2, NMEA Standard Logs* on page 225.

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 <sup>1</sup>	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.

### D.3.19 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 D.2, NMEA Standard Logs* on page 225.

This log reflects the accuracy of the solution type used in BESTGPSPOS, see page 229, and GPGBA, see page 255, 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]

---

### D.3.20 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 D.2, NMEA Standard Logs* on page 225.

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	# msgsv	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 <i>page 260</i> ).	xx	09
5	prn	Satellite PRN number GPS: 1 to 32 SBAS: 33 to 64 (add 87 for PRN#s) GLO: 65 to 96 <sup>1</sup>	xx	03
6	elev	Elevation, degrees, 90 maximum	xx	51
7	azimuth	Azimuth, degrees True, 000 – 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.

### D.3.21 GPHDT NMEA Heading Log



This log is only available on the SPAN-SE-D.

This log provides actual vessel heading in degrees True (from True North). Refer also to information in the HEADING command section of the Firmware Reference Manual. You can also set a standard deviation threshold for this log, as outlined in HDTOUTTHRESHOLD command section of the *OEMV Family Firmware Reference Manual*.

You must have a SPAN-SE dual antenna model to use this log. For further information, refer to Table 83 on page 346.



If the NMEATALKER command (see the NMEATALKER section of the Firmware Reference Manual) is set to AUTO, the talker (the first two characters after the \$ sign in the log header) is set to GP (GPS satellites only), GL (GLONASS satellites only), or GN (satellites from both systems). NovAtel does not support a GLONASS-only solution.

**Message ID: 1045**

**Log Type: Asynch**

#### Recommended Input:

```
log gphdt onchanged
```

#### Example 1 (GPS only):

```
$GPHDT,75.5664,T*36
```

#### Example 2 (Combined GPS and GLONASS):

```
$GNHDT,75.5554,T*45
```

Field	Structure	Field Description	Symbol	Example
1	\$GPHDT	Log header		\$GPHDT
2	heading	Heading in degrees	x.x	75.5554
3	True	Degrees True	T	T
4	*xx	Checksum	*hh	*36
5	[CR][LF]	Sentence terminator		[CR][LF]

### D.3.22 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.

GPVTG, like BESTPOS, contains the best available position, from either GNSS only, or GNSS/INS.

**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 164*, 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	a	A
11	*xx	Checksum	*hh	*7A
12	[CR][LF]	Sentence terminator		[CR][LF]

Refer also to *Table 51, NMEA Positioning System Mode Indicator* on *page 257*.

### D.3.23 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 D.2, NMEA Standard Logs* on page 225.

Message ID: 227

Log Type: Synchronizing

**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 – 31	xx	15
4	month	Month, 01 – 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 <sup>1</sup>	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.

---

### D.3.24 HEADING *Heading Information*

The heading is the angle from True North of the GPS1 to GPS2 vector in a clockwise direction.



Asynchronous logs, such as HEADING, should only be logged ONCHANGED, or the most current data is not available or included in the output. An example of this occurrence is in the ONTIME trigger. If this trigger is not logged ONCHANGED, it may cause inaccurate time tags.



This log is only available on the SPAN-SE-D.

**Message ID: 971**

**Log Type: Asynch**

#### **Recommended Input:**

```
log headinga onchanged
```

#### **ASCII Example:**

```
#HEADINGA, COM1, 0, 77.0, FINESTEERING, 1481, 418557.000, 00000000, 3663, 36137;  
SOL_COMPUTED, L1_INT, 5.913998127, 75.566444397, -0.152066842, 0.0, 0.104981117,  
0.222061798, "AAAA", 13, 10, 10, 0, 0, 00, 0, 11*481a5bab
```

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	HEADING header	Log header		H	0
2	sol stat	Solution status, see <i>Table 45 on page 231</i>	Enum	4	H
3	pos type	Position type, see <i>Table 44 on page 229</i>	Enum	4	H+4
4	length	Baseline length (0 – 3000 m)	Float	4	H+8
5	heading	Heading in degrees (0 – 360.0 degrees)	Float	4	H+12
6	pitch	Pitch ( $\pm 90$ degrees)	Float	4	H+16
7	Reserved		Float	4	H+20
8	hdg std dev	Heading standard deviation in degrees	Float	4	H+24
9	ptch std dev	Pitch standard deviation in degrees	Float	4	H+28
10	stn ID	Station ID string	Char[4]	4	H+32
11	#SVs	Number of observations tracked	Uchar	1	H+36
12	#solnSVs	Number of satellites in solution	Uchar	1	H+37
13	#obs	Number of satellites above the elevation mask	Uchar	1	H+38
14	#multi	Number of satellites above the mask angle with	Uchar	1	H+39
15	Reserved		Uchar	1	H+40
16	ext sol stat	Extended solution status (see <i>Table 47, Extended Solution Status on page 233</i> )	Uchar	1	H+41
17	Reserved		Uchar	1	H+42
18	sig mask	Signals used mask. If 0, signals used in solution are unknown (see <i>Table 46 on page 233</i> )	Uchar	1	H+43
19	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H+44
20	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### D.3.25 HEAVE Heave Filter Log

The log provides vessel heave computed by the integrated heave filter. Refer also to information in the SETHEAWEWINDOW command section. This log is asynchronous, but is available at approximately 10Hz.



Heave functionality is available on SPAN-SE only.

You must have an inertial solution to use this log.

**Message ID: 1382**

**Log Type: Asynch**

#### Abbreviated ASCII Syntax:

```
log heavea onnew
```

#### Example log:

```
#HEAVEA,USB1,0,38.5,FINESTEERING,1630,232064.599,00000000,a759,6696;1630,2320  
64.589885392,0.086825199*93392cb4
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	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	Heave	Instantaneous heave in metres	Double	8	H+12
3	XXXX	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+20
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### D.3.26 ICOMSTATUS Show communication port status

Use this command to view a list of all ICOM port ICOMCONFIG settings.



ETHA is the only Ethernet adapter available to SPAN-SE.

**Message ID: 1292**

#### Abbreviated ASCII Syntax

```
log ICOMSTATUS ONCE
```

#### Example output

```
<ICOMSTATUS ICOM1 0 98.5 UNKNOWN 0 87.125 404c0020 0000 411
< 4
< ICOM1 TCP 3000 0.0.0.0 ETHA
< ICOM2 TCP 3001 0.0.0.0 ETHA
< ICOM3 TCP 3002 0.0.0.0 ETHA
< ICOM4 TCP 3003 0.0.0.0 ETHA
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	Log Header	-	H	0
2	CommPort	Communications port. See <i>Table 23</i> on <i>page 136</i>	Enum	4	H
3	Protocol	Ethernet protocol TCP: 2 UDP: 3	Enum	4	H+4
	EthPort	Ethernet port number	Ulong	4	H+8
	IP	IP address	Ulong	4	H+12
3	XXXX	32-bit CRC (ASCII and Binary only)	Hex	4	H+16
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



---

### **D.3.27 IMUTOANTOFFSETS IMU to Antenna(s) Lever Arm**

This log contains the distance between the IMU and the GNSS antenna(s) in the IMU enclosure frame and its associated uncertainties. The number of lever arms supported will equal the number of antennas supported in the model. For example, one for single antenna, two for dual antenna. This log contains the same information as the BESTLEVERARM or BESTLEVERARM2 logs for each lever arm, but is intended as a single source for all antenna lever arm information available on the system.

**Message ID: 1270**

**Log Type: Asynch**

#### **Abbreviated ASCII Syntax:**

```
log imutoantoffsets
```

#### **Example log:**

```
<OK
[COM1]<IMUTOANTOFFSETS COM1 0 98.5 FINESTEERING 1581 339209.733 60000041 0000
265
<      0 2
<          LEVER_ARM_PRIMARY -0.326000000 0.126000000 1.285000000 0.032600000
0.012600000 0.128500000 LEVER_ARM_FROM_COMMAND
<          LEVER_ARM_SECONDARY -0.325000000 -1.155000000 1.287000000
0.032500000 0.115500000 0.128700000 LEVER_ARM_FROM_COMMAND
[COM1]
```

#### **ASCII Example:**

```
#IMUTOANTOFFSETSA,COM1,0,98.5,FINESTEERING,1581,339209.733,60000041,0000,265;
0,2,LEVER_ARM_PRIMARY,-0.326000000,0.126000000,1.285000000,0.032600000,0
.012600000,0.128500000,LEVER_ARM_FROM_COMMAND,LEVER_ARM_SECONDARY,-
0.325000000,-1.155000000,1.287000000,0.032500000,0.115500000,0.128700000,
LEVER_ARM_FROM_COMMAND*8f0f90b5
```

Field	Field Type	Description	Binary Format	Binary Bytes	Binary Offset
1	Header	Log Header	-	H	0
2	IMU Orientation	See <i>Table 34, Full Mapping Definitions</i> on <i>page 183</i>	ULong	4	H
3	Number of Entries	Number of stored lever arms	ULong	4	H+4
4	Lever Arm Type	Type of lever arm. See <i>Table 53, Lever Arm Type</i> on <i>page 274</i> .	Enum	4	H+8
5	X Offset	IMU Enclosure Frame (m)	Double	8	H+12
6	Y Offset	IMU Enclosure Frame (m)	Double	8	H+20
7	Z Offset	IMU Enclosure Frame (m)	Double	8	H+28
8	X Uncertainty	IMU Enclosure Frame (m)	Double	8	H+36
9	Y Uncertainty	IMU Enclosure Frame (m)	Double	8	H+44
10	Z Uncertainty	IMU Enclosure Frame (m)	Double	8	H+52
11	Lever Arm Source	Source of the lever arm. See <i>Table 54, Lever Arm Source</i> on <i>page 275</i> for the different values	Enum	4	H+60
12...	Next component offset = H + 8 + (#comp * 56)				
variable	XXXX	32-bit CRC (ASCII and Binary only)	Hex	4	H+8+ (#comp * 56)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

**Table 53: Lever Arm Type**

Value (binary)	Lever Arm Source (ASCII)	Description
0	LEVER_ARM_INVALID	An invalid lever arm
1	LEVER_ARM_PRIMARY	Primary lever arm entered for all SPAN systems
2	LEVER_ARM_SECONDARY	Secondary lever arm entered for dual antenna SPAN systems.

---

**Table 54: Lever Arm Source**

<b>Value (binary)</b>	<b>Lever Arm Source (ASCII)</b>	<b>Description</b>
0	LEVER_ARM_NONE	No lever arm exists
1	LEVER_ARM_FROM_NVM	Lever arm restored from NVM
2	LEVER_ARM_CALIBRATING	Lever arm currently calibrating via LEVERARMCALIBRATE command
3	LEVER_ARM_CALIBRATED	Lever arm computed from calibration routine via LEVERARMCALIBRATE command
4	LEVER_ARM_FROM_COMMAND	Lever arm entered via command
5	LEVER_ARM_RESET	If the current IMU orientation does not match the value restored from NVM then the lever arm will be reset to zero with this status.

### D.3.28 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 44* 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 of the SPAN computational frame with respect to local level, and not the frame marked on the enclosure. See the SETIMUORIENTATION command, on *page 179*, 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 131* for information on the APPLYVEHICLEBODYROTATION command.

**Message ID: 263**

**Log Type: Synch**

#### Recommended Input:

```
log insatta ontime 1
```

#### ASCII Example:

```
#INSATTA,COM1,0,94.5,FINESTEERING,1635,489927.000,00000000,0000,406;1635,4899  
26.997500000,-0.000837120,-0.000951510, 19.997813587,  
INS_SOLUTION_GOOD*ef487d49
```

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 <sup>1</sup> 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 <i>Table 5</i> on <i>page 54</i>	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 44* for frame definitions.

### D.3.29 INSATTS Short INS Attitude

This is a short header version of the *INSATT* log on page 276.

Message ID: 319

Log Type: Synch

#### Recommended Input:

```
log insattsa ontime 1
```

#### ASCII Example:

```
%INSATTSA,1541,487975.000;1541,487975.000549050,2.755452422,-  
4.127365126,323.289778434,INS_SOLUTION_GOOD*ba08754f
```

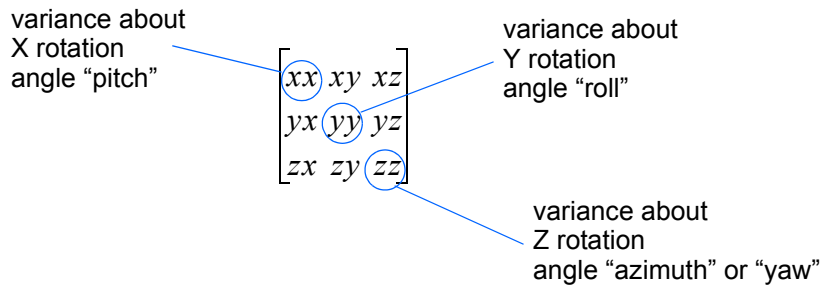
Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Week	GPS Week	Ulong	4	H
3	Seconds into Week	Seconds from week start	Double	8	H+4
4	Roll	Right handed rotation from local level around Y-axis <sup>1</sup> in degrees	Double	8	H+12
5	Pitch	Right handed 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 <i>Table 5</i> on <i>page 54</i>	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 44* for frame definitions.

---

### D.3.30 INSCOV INS Covariance Matrices

The position and velocity matrices in this log each contain 9 covariance values, with respect to the local level frame. The attitude angles are given in the SPAN Computation 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 55 and *Section 3.1, Definition of Reference Frames within SPAN* on page 44.

**Message ID: 264**

**Log Type: Asynch**

#### 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
```

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 per 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)	-	-	-

### D.3.31 INSCOVs Short INS Covariance Log

This is a short header version of the *INCOV* log on page 278. These values are also computed once per second.

**Message ID: 320**  
**Log Type: Asynch**

#### 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
```

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 per 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)	-	-	-



---

### D.3.32 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 center, unless you issue the SETINSOFFSET command, see *page 192*.

**Message ID: 265**

**Log Type: Synch**

#### Recommended Input:

```
log insposa ontime 1
```

#### ASCII Example:

```
#INSPOSA,COM1,0,94.5,FINESTEERING,1635,490452.000,00000000,0000,406;1635,4904  
51.997500000,51.116352320,-  
114.038192411,1047.501906280,INS_SOLUTION_GOOD*2ce15199
```

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 <i>Table 5</i> on <i>page 54</i>	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)	-	-	-

---

### D.3.33 INSPOSS Short INS Position

This is a short header version of the *INSPOS* log on *page 281*.

**Message ID: 321**

**Log Type: Synch**

#### Recommended Input:

```
log inspossa ontime 1
```

#### ASCII Example:

```
%INSPOSSA,1541,487916.000;1541,487916.000549050,51.115797277,-114.037811065,  
1039.030700122,INS_SOLUTION_GOOD*5ca30894
```

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 <i>Table 5</i> on <i>page 54</i>	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)	-	-	-

### D.3.34 INSPOSSYNC Time Synchronized INS Position

This log contains the time synchronized INS position. It is synchronized with GPS each second.

**Message ID: 322**

**Log Type: Asynch**

#### 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
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Sec	Age of synchronized 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)	-	-	-

### D.3.35 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 276*, 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 131*.

**Message ID: 507**

**Log Type: Synch**

#### 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
```

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 negative value implies a southerly direction)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative 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 <i>Table 5</i> on <i>page 54</i>	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

### D.3.36 INSPVAS Short INS Position, Velocity and Attitude

This is a short header version of the INSPVA log on page 284.

Message ID: 508

Log Type: Synch

#### 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
```

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 negative value implies a southerly direction)	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (a negative 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 <i>Table 5</i> on <i>page 54</i>	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

### D.3.37 INSSPD INS Speed

This log contains the most recent speed measurements in the horizontal and vertical directions, and includes an INS status indicator.

**Message ID: 266**

**Log Type: Synch**

#### Recommended Input:

```
log insspda ontime 1
```

#### ASCII Example:

```
#INSSPDA,COM1,0,94.0,FINESTEERING,1635,490702.000,00000000,0000,406;1635,4907  
01.997500000,341.850458219,0.001344556,-  
0.000962845,INS_SOLUTION_GOOD*95aa6f17
```

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 meters per second	Double	8	H+20
6	Vertical Speed	Magnitude of vertical speed in meters per second where a positive value indicates speed upward and a negative value indicates speed downward	Double	8	H+28
7	Status	INS status, see <i>Table 5 on page 54</i>	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)	-	-	-

---

### D.3.38 INSSPDS Short INS Speed

This is a short header version of the *INSPD* log on *page 286*.

**Message ID: 323**

**Log Type: Synch**

#### Recommended Input:

```
log insspdsa ontime 1
```

#### ASCII Example:

```
%INSSPDSA,1541,487975.000;1541,487975.000549050,323.101450813,9.787233999,-  
0.038980077,INS_SOLUTION_GOOD*105ba028
```

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 metres per second	Double	8	H+20
6	Vertical Speed	Vertical speed in metres per second	Double	8	H+28
7	Status	INS status, see <i>Table 5</i> on <i>page 54</i>	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)	-	-	-

### D.3.39 INSUPDATE INS Update

This log contains the most recent INS update information and provides information about the updates that were performed in the INS filter at the previous update epoch and a wheel sensor status indicator.

**Message ID: 757**

**Log Type: Asynch**

#### Recommended Input:

```
log insupdatea onnew
```

#### ASCII Example:

```
#INSUPDATEA,FILE,0,0.0,FINESTEERING,1549,165116.006,00000000,4289,0;SINGLE,0,0,0,FALSE,WHEEL_SENSOR_INACTIVE,HEADING_UPDATE_USED*5a16ecba
```

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	-	H	0
2	Solution Type	-		Type of GPS solution used for the last position update. See <i>Table 44</i> on page 229.	Enum	4	H
3	Reserved	-		-	Long	4	H+4
4	# Phase	0 to maximum GPS L1 channels - 1		Number of phase observations used in the last INS filter update.	Long	4	H+8
5	Reserved	-		-	Long	4	H+12
6	ZUPT	TRUE/FALSE		A zero velocity update was performed during the last INS filter update	Bool	4	H+16
7	Wheel Status	-		Status of a wheel sensor during the last INS update See <i>Table 56</i> on page 289.	Enum	4	H+20
8	Heading Update	Refer to <i>Table 55</i> on page 289		Status of the heading update during the last INS filter update	Enum	4	H+24



**Table 55: Heading Update Enums**

Binary	ASCII	Description
0	HEADING_UPDATE_INACTIVE	A heading updates was not available.
1	HEADING_UPDATE_ACTIVE	Heading updates are running, but the epoch is not used as an update. When all other rejection criteria pass, a heading update will still only be applied once every 5 seconds (20 seconds when stationary).
2	HEADING_UPDATE_USED	The update for that epoch was taken.
3	HEADING_UPDATE_HIGH_STD_DEV	The standard deviation of the update failed a 3 sigma check against the inertial standard deviation (azimuth checked only).
4	HEADING_UPDATE_HIGH_ROTATION	The last 1 second recorded a turn of over 5 degrees/second.
5	HEADING_UPDATE_BAD_MISC	The difference between the ALIGN heading and the INS heading failed a 3 sigma check with the inertial standard deviation.

The Heading Update enums are available on the SPAN-SE-D.

**Table 56: Wheel Status**

Binary	ASCII	Description
0	WHEEL_SENSOR_INACTIVE	The Wheel Sensor is not configured properly. The sensor is not operational.
1	WHEEL_SENSOR_ACTIVE	The Wheel Sensor is properly configured, but an update is not being applied.
2	WHEEL_SENSOR_USED	An update was taken from the Wheel Sensor for that epoch.
3	WHEEL_SENSOR_UNSYNCED	The time between the INS and the Wheel Sensor is >0.2 seconds. An update is not applied.
4	WHEEL_SENSOR_BAD_MISC	There is a gross misclosure between the INS and Wheel Sensor data (wheel spinning).
5	WHEEL_SENSOR_HIGH_ROTATION	The change in heading is >10 degrees. An update is not applied.

---

### D.3.40 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.

**Message ID: 267**

**Log Type: Synch**

#### Recommended Input:

```
log insvela ontime 1
```

#### ASCII Example:

```
#INSVELA,USB1,0,19.0,FINESTEERING,1543,236173.000,00000000,9c95,37343;1543,23  
6173.002500000,14.139471871,-0.070354464,-0.044204369,INS_SOLUTION_GOOD  
*3c37c0fc
```

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 metres per second	Double	8	H+12
5	East Velocity	Velocity East in metres per second	Double	8	H+20
6	Up Velocity	Velocity Up in metres per second	Double	8	H+28
7	Status	INS status, see <i>Table 5</i> on <i>page 54</i>	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)	-	-	-

---

### D.3.41 INSVELS Short INS Velocity

This is a short header version of the *INSVEL* log on *page 290*.

**Message ID: 324**

**Log Type: Synch**

#### Recommended Input:

```
log insvelsa ontime 1
```

#### ASCII Example:

```
%INSVELSA,USB2,0,18.5,FINESTEERING,1541,487942.000,00040000,9c95,37343;1541,4  
87942.000549050,12.656120921,-3.796947104,-0.100024422,INS_SOLUTION_GOOD  
*407d82ba
```

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 metres per second	Double	8	H+12
5	East Velocity	Velocity East metres per second	Double	8	H+20
6	Up Velocity	Velocity Up metres per second	Double	8	H+28
7	Status	INS status, see <i>Table 5</i> on <i>page 54</i>	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)	-	-	-

---

### D.3.42 IPSTATUS Displays ethernet port settings

Use this log to display settings for all Ethernet ports. If DHCP is used to configure IP addresses, this command displays automatically obtained dynamic addresses.

**Message ID: 1289**

**Log Type: Asynch**

**Recommended Input:**

```
log ipstatus
```

**ASCII Example:**

```
#IPSTATUSA,USB1,0,98.0,FINESTEERING,1635,320781.983,00000001,0000,394;1,ETHA,  
192.168.0.10,255.255.255.0,192.168.0.1*2854b03a
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Interface	Always ETHA	Enum	4	H
3	IPAddress	Ethernet IP address	Ulong	4	H+4
4	NetMask	Network mask	Ulong	4	H+8
5	Gateway	Network gateway	Ulong	4	H+12
6	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+16
7	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### D.3.43 LBANDINFO L-band Configuration Information

This log outputs configuration information for an L-band service.



In addition to a NovAtel receiver with L-band capability, a subscription to the OmniSTAR 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
```

**Table 57: 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 <i>Table 57</i> on <i>page 293</i> )	Enum	4	H+16
8	vbs exp week	GPS week number of L-band VBS expiration date <sup>1</sup>	Ulong	4	H+20
9	vbs exp secs	Number of seconds into the GPS week of L-band VBS expiration date <sup>a</sup>	Ulong	4	H+24
10	hp sub	OmniSTAR HP or XP subscription type (see <i>Table 57</i> on <i>page 293</i> )	Enum	4	H+28
11	hp exp week	GPS week number of OmniSTAR HP or XP expiration date <sup>a</sup>	Ulong	4	H+32
12	hp exp secs	Number of seconds into the GPS week of OmniSTAR HP or XP expiration date <sup>a</sup>	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.

### D.3.44 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 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 58: 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 59: OmniSTAR VBS Status Word**

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x0001	Subscription Expired <sup>1</sup>	False	True
	1	0x0002	Out of Region <sup>1</sup>	False	True
	2	0x0004	Wet Error <sup>1</sup>	False	True
	3	0x0008	Link Error <sup>1</sup>	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 60: 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 <sup>1</sup>	Authorized	Unauthorized
	5	0x0020	XP not authorized <sup>1</sup>	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 293*.

**Table 61: OmniSTAR HP/XP Status Word**

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Subscription expired <sup>1</sup>	False	True
	1	0x00000002	Out of region <sup>1</sup>	False	True
	2	0x00000004	Wet error <sup>1</sup>	False	True
	3	0x00000008	Link error <sup>1</sup>	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	Log 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 <i>Table 58</i> on <i>page 295</i> )	Hex	2	H+16
7	VBS status	Status word for OmniSTAR VBS (see <i>Table 59</i> on <i>page 296</i> )	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 <i>hp status 1</i> 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 <i>Table 60</i> on <i>page 297</i> )	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 <i>Table 61</i> on <i>page 298</i> )	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)	-	-	-

**D.3.45 LOGFILESTATUS    Displays information about system logging**

This log displays the current status of media logging. The output file is updated to reflect each new log entry. If no log file is open, FileState is CLOSED, FileName is NULL and Filesize is 0.

**Message ID: 1146**  
**Log Type: Polled**

**Recommended Input:**

```
log logfilestatusa once
```

**ASCII Example:**

```
#LOGFILESTATUSA,ICOM1,0,98.5,UNKNOWN,0,644.581,404c0020,0000,411;OPEN,"Datafile.dat",0,SD*8ef3dc64
```

Field #	Field Type	ASCII Value	Binary Value	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header			Log header	-	H	0
2	FileState	OPEN	0	The file status	Enum	4	H
		CLOSE	1				
3	FileName			Output filename in 8.3 format	Char	12	H+4
4	FileSize			Output filesize in KB	ULONG	4	H+16
5	Media	SD	0 (default)	Output media.	Enum	4	H+20
		USBSTICK	1				

---

### D.3.46 **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 329* 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
```



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: 60	Long	4	H
3	port	Output port, see <i>Table 23, COM Serial Port Identifiers</i> on page 136	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	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 + (number of logs x 34)				
variable	xxxx	32-bit CRC	Hex	4	H+4+(#logs x 34)

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 <i>Table 23, COM Serial Port Identifiers</i> on page 136	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	-

---

### D.3.47 MAC MAC Address

This log displays the SPAN-SE's Media Access Control (MAC) address. See also *Section 3.12, SPAN-SE Ethernet Connection* on page 76.

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.

**Message ID: 1100**

**Log Type: Asynch**

#### Recommended Input:

```
log maca once
```

#### ASCII Example:

```
#MACA,COM1,0,98.5,FINESTEERING,1637,317484.819,00000000,0000,416;"00:21:66:00  
:07:7C"*6abd783e
```

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	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+4
4	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



### D.3.48 **MARK1COUNT, MARK2COUNT, MARK3COUNT, MARK4COUNT** **Mark Count**

When the input mode is set to COUNT using the EVENTINCONTROL command, see *page 142*, the MARKxCOUNT logs become available.



1. Use the ONNEW trigger with this, the MARKxTIME, or the MARKxPVA logs.
2. Only the MARKxCOUNT, MARKxPVA logs, the MARKxTIME 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
```

#### **ASCII 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	MARKxCOUNT header	Log header		H	0
2	Period	Delta time	Ulong	4	H
3	Count	Tick count	Ushort	2	H+4

### D.3.49 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 MARKxPVA logs contain the solution at the IMU center in the SPAN computation frame. If the SETMARKxOFFSET command has been entered, the MARKxPVA log will contain the solution translated, and then rotated, by the values provided in the command. See also the SETMARKxOFFSET commands, valid at the time, on *page 193*.

**MARK1PVA Message ID: 1067**

**MARK2PVA Message ID: 1068**

**MARK3PVA Message ID: 1118**

**MARK4PVA Message ID: 1119**

**Log Type: Synch**

#### Recommended Input:

```
log markpvaa onnew 1
```

#### ASCII Example:

```
#MARK4PVAA,COM1,0,93.0,EXACT,1635,491268.341,00000000,0000,406;1635,491268.34  
1052475,51.116354114,-114.038189654,1047.542785217,-0.003821896,-  
0.000978162,0.001633065,-  
0.419833850,0.633450994,20.646068306,INS_SOLUTION_GOOD*620d9261  
1481 251850.001000000 51.116573435 -114.037237211 1040.805671970 0.000257666  
-0.003030102 -0.000089758 3.082229474 -1.019023628 89.253955744  
INS_SOLUTION_GOOD
```

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 (negative implies a southerly direction) at Mark1, 2, 3 or 4	Double	8	H+36
8	East Velocity	Velocity in an easterly direction (negative 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 <i>Table 5</i> on <i>page 54</i> at Mark	Enum	4	H+84
14	xxxx	32-bit CRC	Hex	4	H+88
15	[CR][LF]	Sentence Terminator (ASCII only)	-	-	-

### D.3.50 **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, MARK2TIME 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 SETMARKxOFFSET commands starting on *page 193*.



1. Use the ONNEW trigger with this or the MARKxPVA logs.
2. Only the MARKxPVA logs, the MARKxTIME 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 62: Clock Model Status**

<b>Clock Status (Binary)</b>	<b>Clock Status (ASCII)</b>	<b>Description</b>
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	MARKxTIME 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 <sup>1</sup>	Double	8	H+28
7	status	Clock model status, see <i>Table 62, Clock Model Status</i> on <i>page 307</i>	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.

### D.3.51 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 website. The PASHR log contains only INS derived attitude information and is only filled when an inertial solution is available.

**Message ID: 1177**

**Log Type: Synch**

#### 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
```

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 is always displayed	RRR.RR	+0.05
6	Pitch	Pitch in decimal degrees. The +/- sign is always 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]

---

### **D.3.52 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:**

Replace ASCII example with the following:

```
#PORTSTATSA,USB1,0,98.0,FINESTEERING,1635,321000.351,00000001,0000,394;10,COM
1,0,132,0,0,21,0,0,0,0,COM2,0,132,0,0,7,0,0,0,0,COM3,0,132,0,0,8,0,0,0,0,COM4
,0,132,0,0,7,0,0,0,0,IMU,0,0,0,0,0,0,0,0,USB1,208,5513,208,0,152,0,0,0,0,IC
OM1,0,0,0,0,0,0,0,0,0,ICOM2,0,0,0,0,0,0,0,0,ICOM3,0,0,0,0,0,0,0,0,ICOM4,0
,0,0,0,0,0,0,0,0*a9518ce6
```

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 <i>Table 23, COM Serial Port Identifiers</i> on page 136	Enum	4	H+4
4	rx chars	Total number of characters received through the port	Ulong	4	H+8
5	tx chars	Total number of characters transmitted through the port	Ulong	4	H+12
6	acc rx chars	Total number of accepted characters received through the 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 number 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)	-	-	-

---

### D.3.53 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 66, Channel Tracking Status* on *page 314*, 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 63: 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 64: 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 65: Channel Tracking Example**

	N7				N6				N5				N4				N3				N2				N1				N0			
0x	0				8				1				0				9				C				0				4			
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary <sup>1</sup>	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0
Data	Chan. Assignment	Reserved (R)			Primary L1	R	Signal Type				Grouping	R	Satellite System		Correlator Spacing		Code locked flag		Parity flag		Phase lock flag	Channel Number				Tracking State						
Value	Automatic				Primary		L1 C/A				Grouped		GPS		PAC		Locked		Known		Locked	Channel 0				L1 Phase Lock Loop						

1. For a complete list of hexadecimal and binary equivalents please refer to *Unit Conversions* on our website at [www.novatel.com](http://www.novatel.com), through *Support | Knowledge and Learning*.

**Table 66: Channel Tracking Status**

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Tracking state	0-11, see <i>Table 63, Tracking State</i> on <i>page 313</i>
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	SV channel number	0-n (0: first, n: last) n depends on the receiver
	5	0x00000020		
	6	0x00000040		
	7	0x00000080		
N2	8	0x00000100	Phase lock flag	0: Not locked, 1: Locked
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000	Code locked flag	0: Not locked, 1: Locked
	13	0x00002000	Correlator type	0-7, see <i>Table 64, Correlator Type</i> on <i>page 313</i>
	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 <sup>1</sup>	0: Not grouped, 1:Grouped
	21	0x00200000	Signal type	Dependant on satellite system above: GPS:                    GLONASS: 0: L1 C/A                0: L1 C/A 5: L2 P                    5: L2 P 9: L2 P codeless 17: L2C SBAS:                    Other: 0: L1 C/A                19: OmniSTAR
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Forward Error Correction (FEC)	0: Not FEC, 1: FEC
	25	0x02000000		
	26	0x04000000		
	27	0x08000000	Primary L1 channel	0: Not primary, 1: Primary

*Continued on the following page*

Nibble #	Bit #	Mask	Description	Range Value
N7	28	0x10000000	Carrier phase measurement <sup>2</sup>	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 <sup>1</sup>	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 $C/N_0 = 10[\log_{10}(S/N_0)]$ (dB-Hz)	Float	4	H+36
11	locktime	number of seconds of continuous tracking (no cycle slipping)	Float	4	H+40
12	ch-tr-status	Tracking status (see <i>Table 66, Channel Tracking Status</i> on <i>page 314</i> and the example in <i>Table D.3.57</i> )	ULong	4	H+44
13...	Next PRN offset = $H + 4 + (\#obs \times 44)$				
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.

### D.3.54 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,
2b9d301165c4f8fffb228820a500a089f31185fe0a8020000,
449d1018be18f41f2aacad0a1a934efc40074ecf88030000,
4b9d301182b9f69f38acad0a3e3ac28841079fcb88020000,
849d101817a1f95f16d7af0a69fbel1fa401d3fd064030000,
8b9d30112909fb2f20d7af0a9f24a687521ddece64020000,
249e1118af4e0470f66d4309a0a631cd642cf5b821320000,
2b9eb110a55903502f6e4309ee28d1ad032c7cb7e1320000,
849e1118b878f54f4ed2aa098c35558a532bde1765220000,
8b9eb110abcf7f1f5ed2aa09cb6ad0f9032b9d16c5220000*0eeeead18
```

**Table 67: Range Record Format (RANGECMP only)**

Data	Bit(s) first to last	Length (bits)	Scale Factor	Units
Channel Tracking Status	0 – 31	32	see Table 66, Channel Tracking Status on page 314	-
Doppler Frequency	32 – 59	28	1/256	Hz
Pseudorange (PSR)	60 – 95	36	1/128	m
ADR <sup>1</sup>	96 – 127	32	1/256	cycles
StdDev-PSR	128 – 131	4	see <sup>2</sup>	m
StdDev-ADR	132 – 135	4	(n + 1)/512	cycles
PRN/Slot <sup>3</sup>	136 – 143	8	1	-
Lock Time <sup>4</sup>	144 – 164	21	1/32	s
C/No <sup>5</sup>	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 ( $ADR\_ROLLS \leq 0$ )

$ADR\_ROLLS = ADR\_ROLLS - 0.5$

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 **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.*

$WAVELENGTH = 0.2442102134246$  for GPS L2

$MAX\_VALUE = 8388608$

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 30*.

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 ( $2097151 \div 32$ ).

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 <i>Table 67 on page 316</i>	Hex	24	H+4
4	Next rangecmp offset = $H + 4 + (\#obs \times 24)$				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	$H + 4 + (\#obs \times 24)$
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

### D.3.55 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)	-	-	-

---

### D.3.56 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 327*.

**Message ID: 268**

**Log Type: Asynch**

#### 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
```



If any of the non-reserved fields indicate a failure, contact NovAtel customer service for further information.

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, see <i>Table 71</i> . Also refer to the Interface Control Documentation as provided by Honeywell and Northrop Grumman, respectively. For the raw IMU status of the iIMU-FSAS, see <i>Table 68</i> . For the raw IMU status of the Litef LCI, refer to <i>Table 69</i> .	Long	4	H+12
5	Z Accel Output	Change in velocity count along Z-axis <sup>1</sup>	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along Z-axis) <sup>1, 2</sup>	Long	4	H+20
7	X Accel Output	Change in velocity count along Z-axis <sup>1</sup>	Long	4	H+24
8	Z Gyro Output	Change in angle count around Z-axis <sup>3</sup> right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around Y-axis) <sup>2, 3</sup> right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around X-axis <sup>3</sup> 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 74* on *page 328*. Multiply the scale factor in *Table 74*, by the count in this field, for the velocity increments. See *Table 1* on *page 32* for a list of IMU enclosures. To obtain acceleration in  $\text{m/s}^2$ , multiply the velocity increments by the output rate of the IMU (eg. 100Hz for HG1700, IMU-CPT, 200Hz iMAR-FSAS, LN200 and LCI).
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 74* on *page 328*. Multiply the appropriate scale factor in *Table 74*, by the count in this field, for the angle increments in radians.



**Table 68: 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	Test mode	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

**Table 69: Litef LCI-1 IMU Status**

Nibble #	Bit #	Mask	Description	Range Value	
N0	0	0x00000001	IBIT Error Flag	Normal: 0, IBIT Error: 1	
	1	0x00000002	CBIT Error Flag	Normal: 0, CBIT Error: 1	
	2	0x00000004	Calibration Status Flag	IMU Uncalibrated: 0, IMU Calibrated: 1	
	3	0x00000008	Not used		
N1	4	0x00000010	Mode Read Flag	Mode in Trans.: 0, Mode	Refer to <i>Table 70</i> on <i>page 323</i>
	5	0x00000020	IMU Mode Indication 1	Not Set: 0, Set: 1	
	6	0x00000040	IMU Mode Indication 2	Not Set: 0, Set: 1	
	7	0x00000080	IMU Mode Indication 3	Not Set: 0, Set: 1	
N2	8	0x00000100	Master NoGo	Normal: 0, NoGo: 1	
	9	0x00000200	IMU NoGo	Normal: 0, NoGo: 1	
	10	0x00000400	B-290 Z NoGo	Normal: 0, NoGo: 1	
	11	0x00000800	B-290 Y NoGo	Normal: 0, NoGo: 1	
N3	12	0x00001000	B-290 X NoGo	Normal: 0, NoGo: 1	
	13	0x00002000	FORS Z NoGo	Normal: 0, NoGo: 1	
	14	0x00004000	FORS Y NoGo	Normal: 0, NoGo: 1	
	15	0x00008000	FORS X NoGo	Normal: 0, NoGo: 1	
N4	16	0x00010000	Master Warning	Normal: 0, Warning: 1	
	17	0x00020000	IMU Warning	Normal: 0, Warning: 1	
	18	0x00040000	B-290 Z Warning	Normal: 0, Warning: 1	
	19	0x00080000	B-290 Y Warning	Normal: 0, Warning: 1	
N5	20	0x00100000	B-290 X Warning	Normal: 0, Warning: 1	
	21	0x00200000	FORS Z Warning	Normal: 0, Warning: 1	
	22	0x00400000	FORS Y Warning	Normal: 0, Warning: 1	
	23	0x00800000	FORS X Warning	Normal: 0, Warning: 1	
N6	24	0x01000000	Not Used		
	25	0x02000000			
	26	0x04000000			
	27	0x08000000			
N7	28	0x10000000			
	29	0x20000000			
	30	0x40000000			
	31	0x80000000			

**Table 70: Mode Indication**

MDI3	MDI2	MDI1	MRF	Current IMU Mode
0	0	0	0	Power On BIT (PBIT)
0	0	0	1	Standby Mode
0	1	1	0	Initiated BIT (IBIT)
0	1	1	1	IBIT Ready
1	1	0	1	Operational Mode

**Table 71: HG1700 and LN200 Status**

Nibble Number	Bit #	Mask	HG1700 Description		LN200 Description	
N0	0	0x00000001	Reserved		IMU Status	Passed: 0 Failed: 1
	1	0x00000002	Reserved		IMU Status	Passed: 0 Failed: 1
	2	0x00000004	Reserved		IMU Status	Passed: 0 Failed: 1
	3	0x00000008	Reserved		IMU Status	Passed: 0 Failed: 1
N1	4	0x00000010	IMU Status	Passed: 0 Failed: 1	IMU Status	Passed: 0 Failed: 1
	5	0x00000020	IMU Status	Passed: 0 Failed: 2	IMU Status	Passed: 0 Failed: 1
	6	0x00000040	IMU Status	Passed: 0 Failed: 3	IMU Status	Passed: 0 Failed: 1
	7	0x00000080	IMU Status	Passed: 0 Failed: 4	IMU Status	Passed: 0 Failed: 1
N2	8	0x00000100	Reserved		IMU Status	Passed: 0 Failed: 1
	9	0x00000200	Reserved		IMU Status	Passed: 0 Failed: 1
	10	0x00000400	Reserved		IMU Status	Passed: 0 Failed: 1
	11	0x00000800	Reserved		IMU Status	Passed: 0 Failed: 1

*Continued on the following page*

Nibble Number	Bit #	Mask	HG1700 Description		LN200 Description	
N3	12	0x00001000	Reserved		IMU Status	Passed: 0 Failed: 1
	13	0x00002000	Reserved		IMU Status	Passed: 0 Failed: 1
	14	0x00004000	Reserved		IMU Status	Passed: 0 Failed: 1
	15	0x00008000	Reserved		Reserved	
N4	16	0x00010000	Reserved		Reserved	
	17	0x00020000	Reserved		Reserved	
	18	0x00040000	Reserved		Reserved	
	19	0x00080000	Reserved		Reserved	
N5	20	0x00100000	Reserved		Reserved	
	21	0x00200000	Reserved		Reserved	
	22	0x00400000	Reserved		Reserved	
	23	0x00800000	Reserved		Reserved	
N6	24	0x01000000	Reserved		IMU Status	Passed: 0 Failed: 1
	25	0x02000000	Reserved		IMU Status	Passed: 0 Failed: 1
	26	0x04000000	Reserved		IMU Status	Passed: 0 Failed: 1
	27	0x08000000	IMU Status	Passed: 0 Failed: 1	IMU Status	Passed: 0 Failed: 1
N7	28	0x10000000	IMU Status	Passed: 0 Failed: 1	IMU Status	Passed: 0 Failed: 1
	29	0x20000000	IMU Status	Passed: 0 Failed: 1	Reserved	
	30	0x40000000	IMU Status	Passed: 0 Failed: 1	IMU Status	Passed: 0 Failed: 1
	31	0x80000000	IMU Status	Passed: 0 Failed: 1	Reserved	

**Table 72: IMU-CPT Status**

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Gyro X Status	1: Valid, 0: Invalid
	1	0x00000002	Gyro Y Status	1: Valid, 0: Invalid
	2	0x00000004	Gyro Z Status	1: Valid, 0: Invalid
	3	0x00000008	Unused	Set to 0
N1	4	0x00000010	Accelerometer X Status	1: Valid, 0: Invalid
	5	0x00000020	Accelerometer Y Status	1: Valid, 0: Invalid
	6	0x00000040	Accelerometer Z Status	1: Valid, 0: Invalid
	7	0x00000080	Unused	Set to 0
N2	8	0x00000100	Unused	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000		
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000		
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000		
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000		
	25	0x02000000		
	26	0x04000000		
	27	0x08000000		
N7	28	0x10000000		
	29	0x20000000		
	30	0x40000000		
	31	0x80000000		

**Table 73: HG1900 and HG1930 Status**

Nibble #	Bit #	Mask	Description	Range Value
N0	0	0x00000001	Reserved	
	1	0x00000002		
	2	0x00000004		
	3	0x00000008		
N1	4	0x00000010	IMU Failed	Passed: 0, Failed: 1
	5	0x00000020	Gyro Failed	Passed: 0, Failed: 1
	6	0x00000040	Accel Failed	Passed: 0, Failed: 1
	7	0x00000080	AGC Voltage	Passed: 0, Failed: 1
N2	8	0x00000100	Reserved	
	9	0x00000200		
	10	0x00000400		
	11	0x00000800		
N3	12	0x00001000	Reserved	
	13	0x00002000		
	14	0x00004000		
	15	0x00008000		
N4	16	0x00010000	Reserved	
	17	0x00020000		
	18	0x00040000		
	19	0x00080000		
N5	20	0x00100000	Reserved	
	21	0x00200000		
	22	0x00400000		
	23	0x00800000		
N6	24	0x01000000	Motor Bias Voltage	Passed: 0, Failed: 1
	25	0x02000000	Reserved	
	26	0x04000000	Processor	Passed: 0, Failed: 1
	27	0x08000000	Memory	Passed: 0, Failed: 1
N7	28	0x10000000	ASIC	Passed: 0, Failed: 1
	29	0x20000000	Gyro Frequency	Passed: 0, Failed: 1
	30	0x40000000	Accel Frequency	Passed: 0, Failed: 1
	31	0x80000000	Reserved	

### D.3.57 RAWIMUS Short Raw IMU Data

This is a short header version of the *RAWIMU* log on *page 319*.

Message ID: 325  
Log Type: Asynch

#### Recommended Input:

```
log rawimusa onnew
```

#### ASCII Example:

```
%RAWIMUSA,1105,425384.180;  
1105,425384.156166800,111607,43088060,430312,-3033352,  
-132863,186983,823*5aa97065
```

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, see <i>Table 71</i> on <i>page 323</i> . Refer also to the Interface Control Documentation as provided by Honeywell and Northrop Grumman, respectively. For the raw IMU status of the iIMU-FSAS, see <i>Table 68</i> . For the raw IMU status of the Litef LCI, refer to <i>Table 69</i> .	Long	4	H+12
5	Z Accel Output	Change in velocity count along Z-axis <sup>1</sup>	Long	4	H+16
6	- (Y Accel Output)	- (Change in velocity count along Y-axis) <sup>1, 2</sup>	Long	4	H+20
7	X Accel Output	Change in velocity count along X-axis <sup>1</sup>	Long	4	H+24
8	Z Gyro Output	Change in angle count around Z-axis <sup>3</sup> right-handed	Long	4	H+28
9	- (Y Gyro Output)	- (Change in angle count around Y-axis) <sup>2, 3</sup> right-handed	Long	4	H+32
10	X Gyro Output	Change in angle count around X-axis <sup>3</sup> 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 74* on *page 328*. Multiply the scale factor in *Table 74*, by the count in this field, for the velocity increments in m/s. See *Table 1* on *page 32* for a list of IMU enclosures. To obtain acceleration in m/s<sup>2</sup>, multiply the velocity increments by the output rate of the IMU (eg. 100Hz for HG1700, IMU-CPT, 200Hz iMAR-FSAS, LN200 and LCI).
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 74* on *page 328*. Multiply the appropriate scale factor in *Table 74*, by the count in this field, for the angle increments in radians.

**Table 74: Raw IMU Scale Factors**

Scale IMU	Gyroscope Scale Factor	Acceleration Scale Factor
HG1700-AG11 HG1700-AG58 HG1900-CA29 HG1930-AA99	$2.0^{-33}$ rad/LSB	$2.0^{-27}$ ft/s/LSB
HG1700-AG17 HG1700-AG62	$2.0^{-33}$ rad/LSB	$2.0^{-26}$ ft/s/LSB
IMU-CPT	$0.1 / (3600.0 \times 256.0)$ rad/LSB	$0.05 / 2^{15}$ m/s/LSB
iIMU-FSAS	$0.1 \times 2^{-8}$ arcsec/LSB	$0.05 \times 2^{-15}$ m/s/LSB
Litef LCI-1	$4 \times 2^{-31}$ deg/LSB	$2 \times 2^{-31}$ m/s/LSB
LN-200	$2^{-19}$ rad/LSB	$2^{-14}$ m/s/LSB



---

### D.3.58 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 301* for a list of currently active logs.

**Message ID: 128**

**Log Type: Polled**

#### Recommended Input:

```
log rxconfiga once
```

#### ASCII Example<sup>1</sup>:

```
#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*d9b9f449*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;ICOM1,NOVATEL,NOVATEL,ON*421e3cb1*e457f77e
#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
```

- 
1. The embedded CRCs are flipped to make the embedded messages recognizable to the receiver.  
For example, if the embedded message is:

```
91f89b07: 10010001111110001001101100000111
11100000110110010001111110001001: e0d91f89
```

its CRC is really e0d91f89.

```
#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
```



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 want 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)	-	-	-

### D.3.59 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. Receiver errors are handled differently based on where the error occurs. If an error occurs on the SPAN system (shown in the Receiver Error word) the receiver will continue to operate as best as possible. If an error occurs on either the internal OEMV-3 or OEMV-2 card (shown by the Error Flag field in the corresponding status word), that card will idle all channels, turn off its antenna, and disable the RF hardware, as these conditions are considered to be fatal errors. Shut down of either the OEMV-3 or OEMV-2 in this way will result in some loss of functionality for the SPAN-SE.

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 204*.



1. Field #4, the receiver status word as represented in *Table 76*, is also in Field #8 of the header. See the *ASCII Example* below and *Table 76* on *page 334* 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.

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.

In the event of a SPAN-SE receiver error, the LEDs flash the binary number of the receiver error. Red is “1” and green is “0”. If you have an invalid auth code, the LEDs will be green-green-red-green-red, which is 5 in binary and that means bit 5 of the receiver error word is set.

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;  
00000000,4,40000020,00000000,00000000,00000000,00000000,00000000,000  
0000,00000020,00000000,00000000,00000000,00000000,00000000,00000000  
*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;  
80000000,4,40000021,00000000,00000000,00000000,00000000,00000000,000  
0000,00000020,00000000,00000000,00000000,00000000,00000000,00000000  
*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 75: 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

*Continued on the following page*

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
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
N4	16	0x00010000	Software resource limit	OK	Error
	17	0x00020000	Model status	Valid	Invalid
	18	0x00040000	COM port power status	Not Over Current	Over Current
	19	0x00080000	Reserved		
N5	20	0x00100000	Remote loading in progress	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 76: SPAN Receiver Status**

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Error Flag, see <i>Table 75, SPAN Receiver Error</i> on page 332	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	INS reset detected	No INS Reset.	INS Reset
	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 is OK	buffer is greater than 80% full
N7	28	0x10000000	Forced UNLOGALL to OEMV3	Did not happen	Happened
	29	0x20000000	OEMV-2 status event	No event	Event
	30	0x40000000	OEMV-3 status event	No event	Event
	31	0x80000000	Reserved		

**Table 77: 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 78: OEMV-3 Status**

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Reserved		

**Table 79: OEMV-2 Status**

Nibble #	Bit #	Mask	Description	Bit = 0	Bit = 1
N0	0	0x00000001	Reserved		

Field #	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	RXSTATUS header	Log header		H	0
2	error	Receiver error (see <i>Table 75</i> on page 332). 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 <i>Table 76</i> on page 334)	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 <i>Table 77</i> on page 335)	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 <i>Table 78</i> on page 335)	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 <i>Table 79</i> on page 335)	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 + (number of 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)	-	-	-



---

### D.3.60 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 204*.

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 331*, 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 LEDs flash 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.

In the event of a SPAN-SE receiver error, the LEDs flash the binary number of the receiver error. Red is “1” and green is “0”. If you have an invalid auth code, the LEDs will be green-green-red-green-red, which is 5 in binary and that means bit 5 of the receiver error word is set. The 5-bit binary error code corresponds to the 32 possible errors defined in *Table 75* starting on *page 332*.



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 80: Status Word**

Word (binary)	Word (ASCII)	Description
0	ERROR	Receiver error word, see <i>Table 75</i> on <i>page 332</i>
1	STATUS	Receiver status word, see <i>Table 76</i> on <i>page 334</i>
2	AUX1	Auxiliary 1 status word, see <i>Table 77</i> on <i>page 335</i>
3	AUX2	Auxiliary 2 status word see <i>Table 78</i> on <i>page 335</i>
4	AUX3	Auxiliary 3 status word see <i>Table 79</i> on <i>page 335</i>

**Table 81: 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 <i>Table 80</i> above)	Enum	4	H
3	bit position	Location of the bit in the status word (see <i>Table 76</i> starting on <i>page 334</i> for the receiver status table or the auxiliary status tables on <i>page 335</i> )	Ulong	4	H+4
4	event	Event type (see <i>Table 81</i> 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)	-	-	-

### D.3.61 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 199* to add a model. See the VERSION log on *page 346* 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 spanvalidmodelsa 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	SPANVALIDMODELS 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 <sup>1</sup>	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 + (number of 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

### D.3.62 TAGGEDMARKxPVA Output information from a MARK input

Use this command to output Event ID tagged, position, velocity and attitude information received on a MARK input.

#### Abbreviated ASCII Syntax

TAGGEDMARKxPVA ONNEW

#### Message IDs:

MARK1: 1258

MARK2: 1259

MARK3: 1327

MARK4: 1328

#### Recommended Input:

```
log TAGGEDMARK1PVA ONNEW
```

#### Abbreviated ASCII Example:

```
log taggedmark1pva onnew
```

Field	Field Name	Description	Field Type	Binary Bytes	Binary Offset
1	Header	This field contains the message header	-	H	0
2	Week	The GPS week at the MARK1through MARK4 request	Ulong	4	H
3	Seconds	Seconds from the GPS week at the MARK1 through MARK4 request	Double	8	H+4
4	EventID	Tagged Event ID. This value corresponds to the ID set by a TAGNEXTMARK command associated with the specified mark input. If no TAGNEXTMARK is set before an event is received on the specified mark, this value will default to 0.	Ulong	4	H+12
5	Latitude	The latitude (WGS84) at MARK1 through MARK4	Double	8	H+16
6	Longitude	The longitude (WGS84) at MARK1through MARK4	Double	8	H+24
7	Height	The height (WGS84) at MARK1 through MARK4	Double	8	H+32
8	NVelocity	The velocity in a northern direction at MARK1, through MARK4. A negative value implies southern movement	Double	8	H+40
9	EVelocity	The velocity an eastern direction at MARK1, through MARK4. A negative value implies western movement	Double	8	H+48
10	UpVelocity	Velocity at an upward direction at MARK1 through MARK4	Double	8	H+56
11	Roll	Right-handed rotation from a local level around the Y-axis in degrees at MARK1 through MARK4	Double	8	H+64
12	Pitch	Right-handed rotation from a local level around the X-axis in degrees at MARK1 through MARK4	Double	8	H+70
13	Azimuth	Left-handed rotation around the Z-axis in a clockwise direction from north at MARK1 through MARK4	Double	8	H+78
14	Status	INS Status	Enum	4	H+86
15	xxxx	32-bit CRC	Hex	4	H+90
16	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

---

### **D.3.63 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 <i>Table 62</i> on <i>page 307</i>	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) <sup>1</sup>	Uchar	1	H+32
8	utc day	UTC day (0 – 31) <sup>1</sup>	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) <sup>2</sup>	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.

### D.3.64 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 D.1.4, ASCII and Binary Logs with Short Headers on page 221*.

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 64*.



If you are using an iMAR iMWS (Magnetic Wheel Speed Sensor and Convertor) connected directly to the iMU FSAS, Field #4, the float wheel velocity, is filled instead of Field #3, the unsigned short wheel velocity.

When you send a WHEELVELOCITY command, described *Section B.4.56, WHEELVELOCITY Wheel velocity for INS augmentation on page 209*, 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.

**Message ID: 622**

**Log Type: Asynch**

#### 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
```

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 per second	Ushort	2	H+2
4	fWheel Vel	Float wheel velocity in counts per second	Float	4	H+4
5	Reserved		Ulong	4	H+8
6			Ulong	4	H+12
7	Cumulative Ticks	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)	-	-	-

---

### D.3.65 VARIABLELEVERARM *Display variable lever arm details*

Use this log to redisplay the re-calculated variable lever arm whenever a new INPUTGIMBALANGLE command is received.

This log is not output until the INS alignment is complete.

**Message ID: 1320**

**Log Type: Asynch**

#### **Recommended Input:**

```
log variableleverarma onnew
```

#### **ASCII Example:**

```
#VARIABLELEVERARMA,SPECIAL,0,81.5,FINESTEERING,1614,495820.512,40040000,0000,320;-0.0959421909646755,0.1226971902356540,1.1319295452903300,0.0100057787272846,0.0122604827412661,0.1131929545290330*9611d3c6
```

Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	XOffset	SPAN body frame X-axis offset	Double	8	H
3	YOffset	SPAN body frame Y-axis offset	Double	8	H+8
4	ZOffset	SPAN body frame Z-axis offset	Double	8	H+16
5	XUncert	X-axis uncertainty in degrees	Double	8+24	H
6	YUncert	Y-axis uncertainty in degrees	Double	8+32	H
7	ZUncert	Z-axis uncertainty in degrees	Double	8+40	H
8	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+24
9	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



---

### **D.3.66 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 179*. 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 207*, or whatever was solved for after invoking the RVBCALIBRATE command, see *page 174*.

#### **Recommended Input:**

```
log vehiclebodyrotationa onchanged
```

#### **ASCII Example:**

```
#VEHICLEBODYROTATIONA,COM1,0,36.5,FINESTEERING,1264,144170.094,00000000,bcf2,1541;1.5869999997474209,2.6639999995760122,77.6649999876392343,2.0000000000000000,2.0000000000000000,5.000000000000000*25f886cc
```

### D.3.67 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 85, VERSION Log: Field Formats* on *page 347* 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","SCS000200RN0000","SCB000100RB0000","Mar  3 2009","16:35:00",
SPANFPGA,"","","","SCF000100RN0014","","","","
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","","","","2.010","","","Feb 09 2007","10:39:41"*ffb53221
```

**Table 82: OEMV in SPAN-SE Model Designators**

Designator	Description
G	12 L1 or 12 L1/L2 GLONASS channels, frequencies to match GPS configuration
L	1 L-band channel OmniSTAR HP/XP capability
Q	Allows OEMV3 to talk to SPAN Receiver (for OEMV FW Version 3.800 and later)
R	Receive RT2 and/or RT20 corrections
Z	Enable align header

**Table 83: SPAN-SE Model Designators**

Designator	Description
C	Low-cost INS. Alignment restricted to kinematic only.
D	SPAN-SE with dual-antenna functionality
I	SPAN supporting IMUs with data rates $\leq 100$ Hz
J	SPAN supporting IMUs with data rates $> 100$ Hz
S	A single GNSS card system where only the OEMV-3 is included

**Table 84: 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 85: 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) <sup>1</sup> CCC: COM port configuration (for example, 22T) <sup>2</sup>
sw version, boot version	PPPVVRRrrXCxxxx	PPP: product designation (for example, SCD) VV: version number RR: major revision number rr: minor revision number X: Special (S), Beta (B), Internal Development (D, A) C: control, either N (none) or M (military) xxxx: four-digit number indicating the build release
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.

D.3.68 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.

Message ID: 646  
Log Type: Asynch

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
```

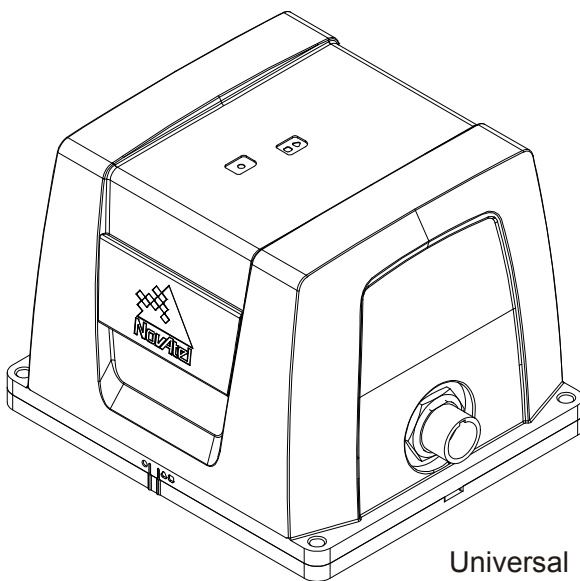
Field #	Field Type	Data Description	Format	Binary Bytes	Binary Offset
1	Log Header	Log header	-	H	0
2	Scale	Wheel sensor scale factor	Double	8	H
3	Circum	Wheel circumference (m)	Double	8	H+8
4	Var	Variance of circumference (m <sup>2</sup> )	Double	8	H+16
5	xxxx	32-bit CRC (ASCII, Binary and Short Binary only)	Hex	4	H+24
6	[CR][LF]	Sentence terminator (ASCII only)	-	-	-



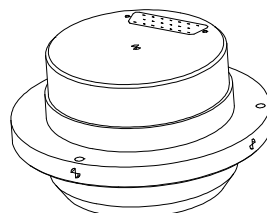
**Important!** Assemble in accordance with applicable industry standards. Ensure all ESD measures are in place, in particular, use a ground strap before exposing or handling any electronic items, including the IMU. Take care to prevent damaging or marring painted surfaces, O-rings, sealing surfaces, and the IMU.

The following procedure provides the necessary information to install the HG1700 sensor into the Universal Enclosure (NovAtel part number 01018589), both illustrated below. The steps required for this procedure are:

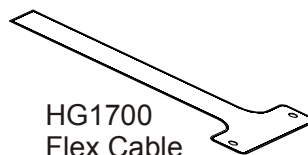
- Disassemble the Universal Enclosure
- Install the HG1700 Sensor Unit
- Reassemble the Universal Enclosure



Universal Enclosure



HG1700 Sensor Unit



HG1700 Flex Cable



1. Use thread-locking fluid on all fasteners except for the flex cable connectors.
2. Torque values for all fasteners, including those for the flex cable, are as follows:
 

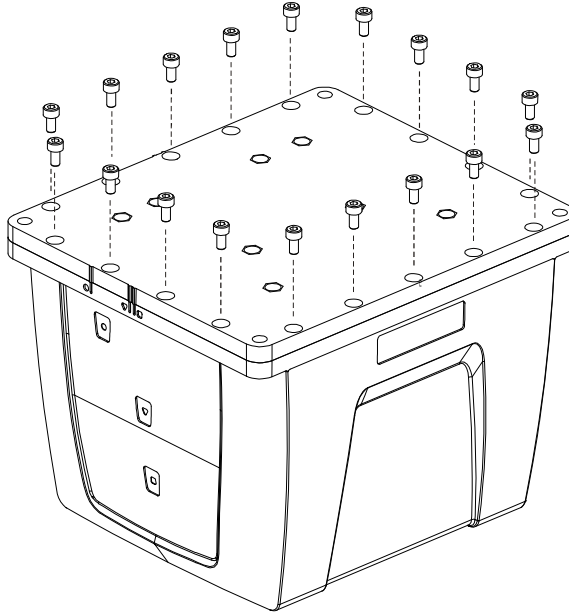
Size 2-56:	0.20-0.25 N-m (1.8-2.2 lb-in) [28-35 oz/in]
Size M4:	1.36-1.58 N-m (12.0-14.0 lb-in)
Size 8-32:	1.55-1.70 N-m (13.7-15.0 lb-in)

---

## E.1 Disassemble the Universal Enclosure

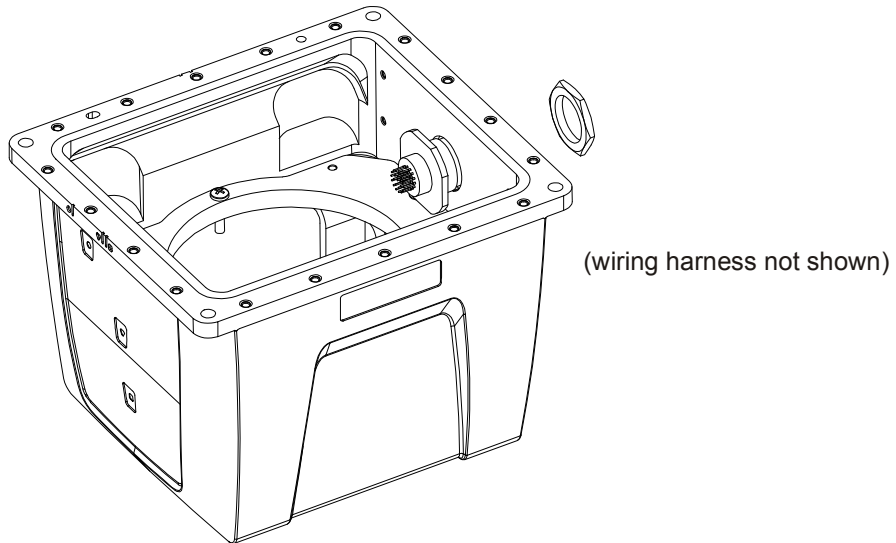
Disassemble the Universal Enclosure as follows:

1. Using a 3 mm hex bit, remove the M4 screws (they will be reused) and the base, as shown in *Figure 44*. Ensure the O-rings come with the base when it is removed, and that they are not damaged.



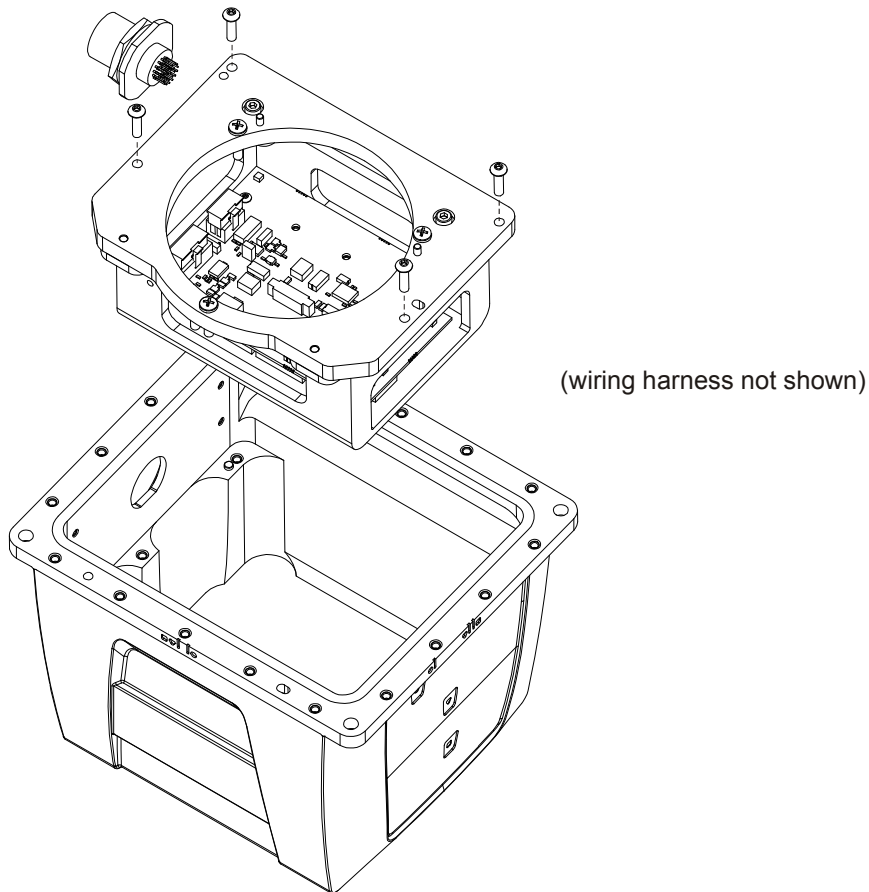
**Figure 44: Remove Base**

- 
2. Using a 30 mm socket, remove the jam nut and free the wiring harness connector from the body, as shown in *Figure 2*. Retain the O-ring and the jam nut for reassembly.



**Figure 45: Disconnect Wiring Harness from Enclosure Body**

- 
- Using a 2.5 mm hex bit, unscrew the M4 screws and remove the IMU mounting plate, bracket and cable harness, as shown in *Figure 46*:



**Figure 46: Remove IMU Mounting Plate and Bracket**

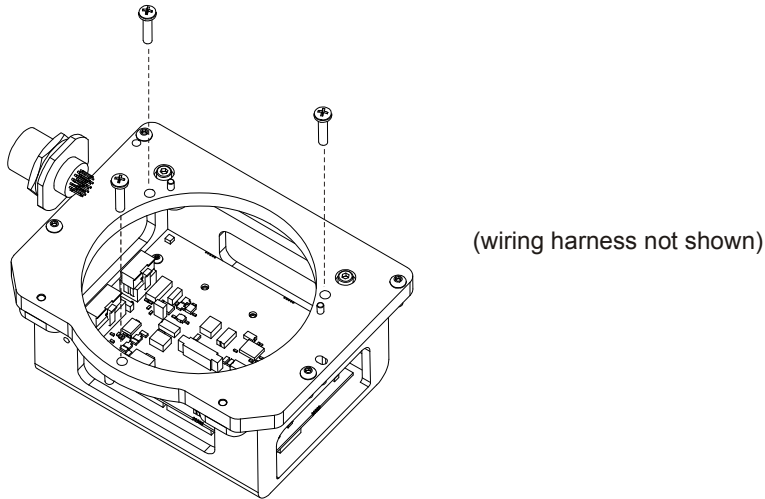


---

## E.2 Install the HG1700 Sensor Unit

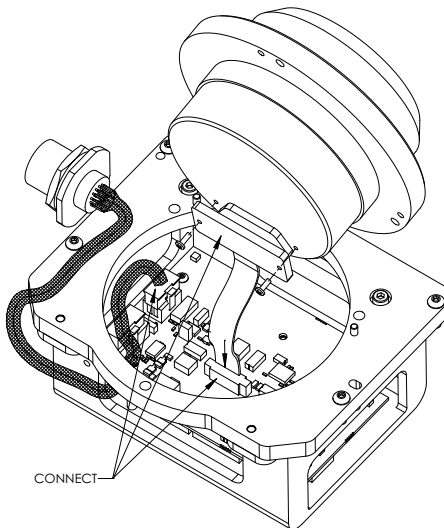
To install the HG1700 sensor unit in the Universal Enclosure:

1. Using a Phillips screwdriver, remove the 8-32 IMU mounting screws from the IMU mounting plate, as shown in *Figure 47*:



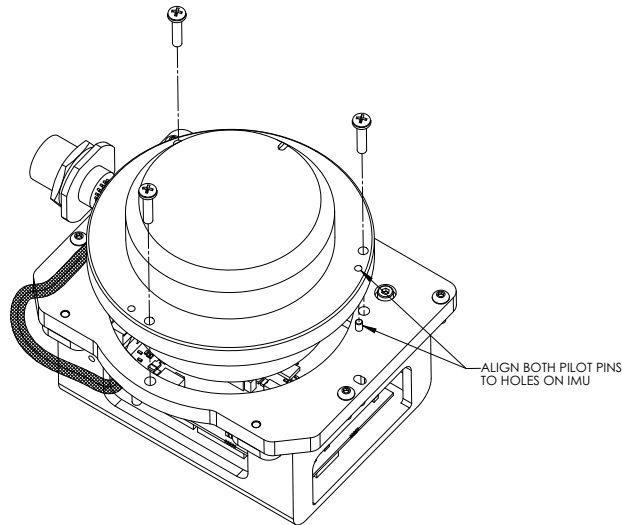
**Figure 47: Remove IMU Mounting Screws**

2. Check the connection of the internal cable harness to the board assembly and route as shown in *Figure 48*. Before you connect the IMU cable harness, make sure the connector on the board assembly is clicked open. Connect the IMU cable harness to the IMU (fasten the 2-56 screws but do not use thread-locking fluid), then connect to the board assembly. Ensure the cable housing latches.



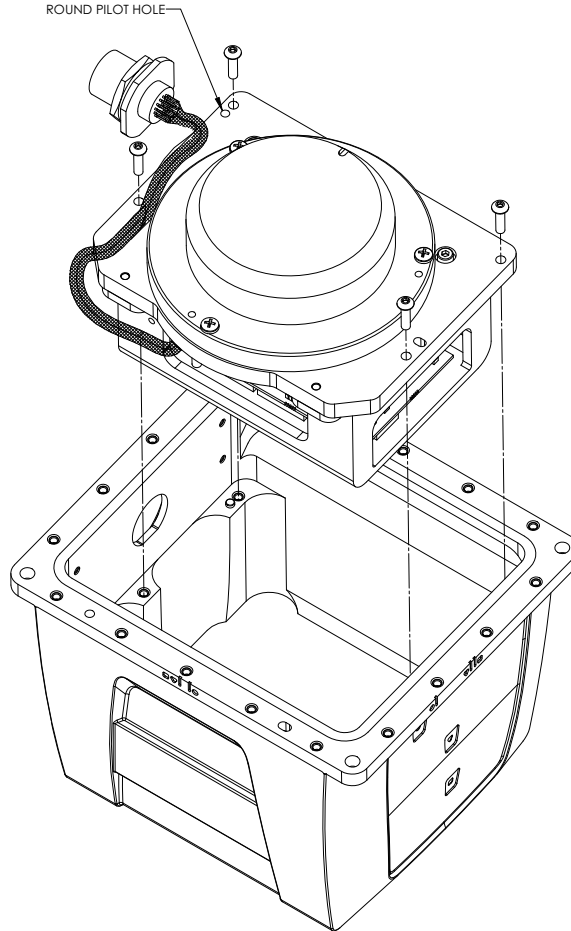
**Figure 48: Connect IMU to IMU Mounting Plate**

- 
3. Being careful of the connectors and the orientation, align the pilot holes of the IMU with the pilot pins of the mounting plate. Gently place the IMU and mounting plate together, being careful not to pinch the cable harness. Screw the IMU and mounting plate together, using thread-locking fluid on the 8-32 screws, as shown in *Figure 49*.



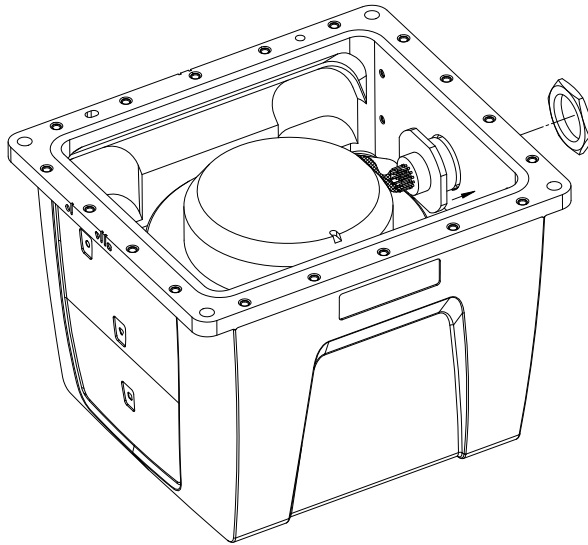
**Figure 49: Installing IMU to Mounting Plate**

- 
4. Starting with the round pilot hole, shown in *Figure 50*, align the pilot holes of the assembled plate (noting the orientation) with the pilot pins of the enclosure body. Lower the assembly into place, then fasten using thread-locking fluid on the M4 screws.



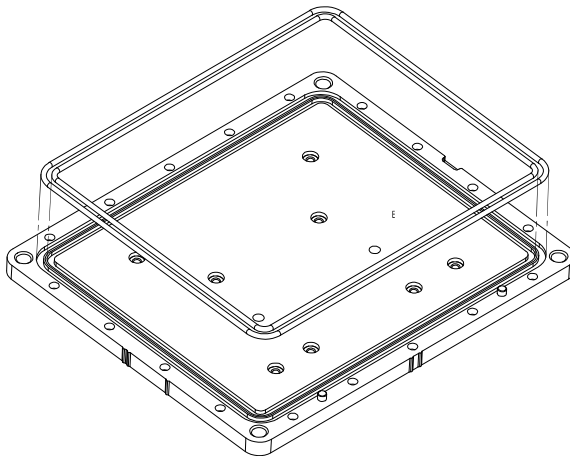
**Figure 50: Assemble Into Enclosure Body**

- 
5. Connect the internal cable harness to the enclosure body, as shown in *Figure 51*. During this step, ensure the connector O-ring (supplied with the connector of the internal cable harness) remains flat within the connector's groove, and make sure the groove is clean and free of debris. Fasten the connector to the enclosure body wall using the jam nut supplied with the connector. Apply thread-locking fluid then, with a 30 mm socket, tighten the jam nut to 6.9 N-m (61 lb-in/5.1 lb-ft).



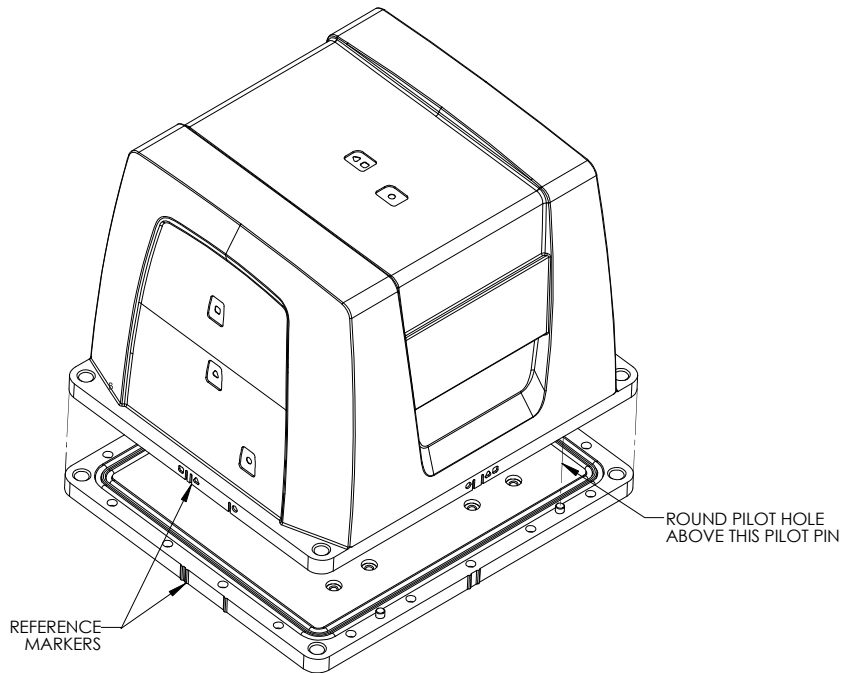
**Figure 51: Fasten Internal Cable Harness**

6. Ensure the O-rings are in place. If they are not, as necessary, make sure the grooves of the enclosure base are clean and free of debris, using isopropyl alcohol. As shown in *Figure 52*, install the outer environmental and inner EMI O-rings in the enclosure base, being careful not to stretch or twist them. O-rings must remain flat within the grooves during the remainder of the assembly procedure.



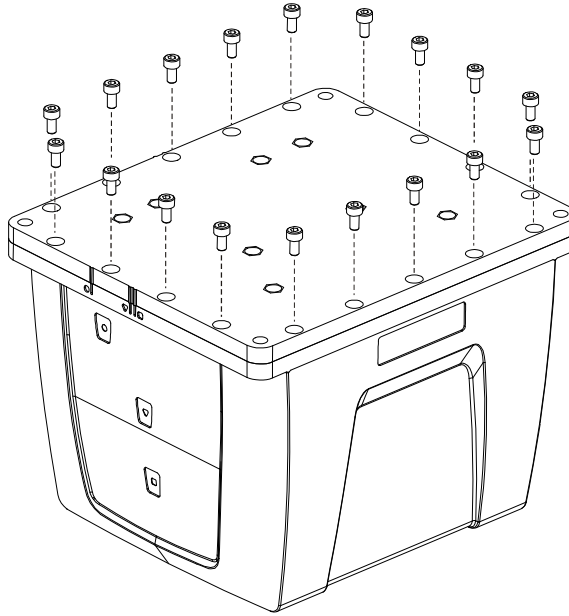
**Figure 52: Install O-rings**

7. Clean the surface of the enclosure body, where it mates with the O-rings, using isopropyl alcohol. As shown in *Figure 53*, align the reference markers and pilot holes/pins of the enclosure body and base. Carefully lower the body onto the base, observing the O-rings and alignment of corners. Press the enclosure body into place, starting with the round pilot hole indicated in *Figure 53*.



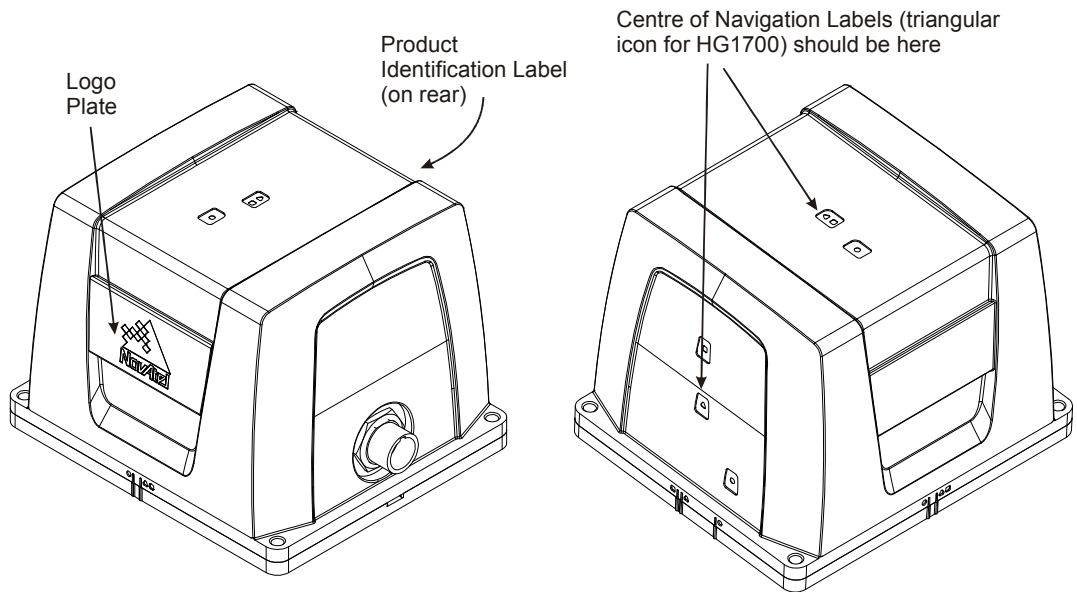
**Figure 53: Install Enclosure Body on the Base**

- 
8. While squeezing and holding the enclosure body and base together to maintain tight contact, carefully turn the assembly over and place it on its top, as shown in *Figure 54*. Using a 3 mm hex bit, lightly fasten four equally spaced M4 screws to hold the parts together. Apply thread-locking fluid to each screw before inserting. Install the remaining screws in similar fashion. Tighten all screws then check all of them again for tightness. Tighten these screws to 1.36-1.58 N-m (12-14 lb-in). Do not over-tighten.



**Figure 54: Screw Enclosure Base to Body**

9. Ensure the product identification label, the logo plate and the centre of navigation labels are properly affixed and contain the correct information. The final assembled unit will be similar to that shown in *Figure 55*:



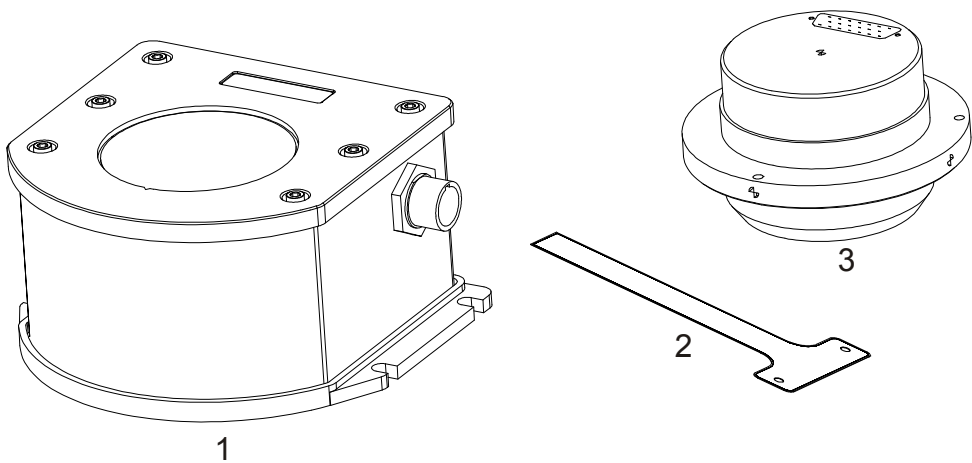
**Figure 55: Final Assembly**

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). 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 56: Required Parts**

Reference	Description
1	SPAN IMU Enclosure
2	HG1700 Flex Cable
3	HG1700 Sensor Unit

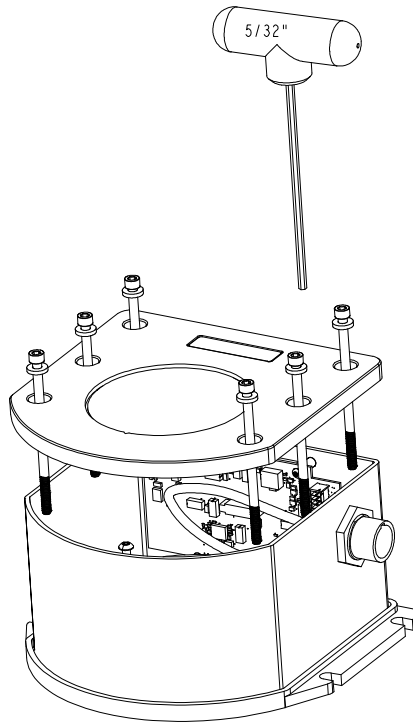


---

## F.1 Disassemble the SPAN IMU Enclosure

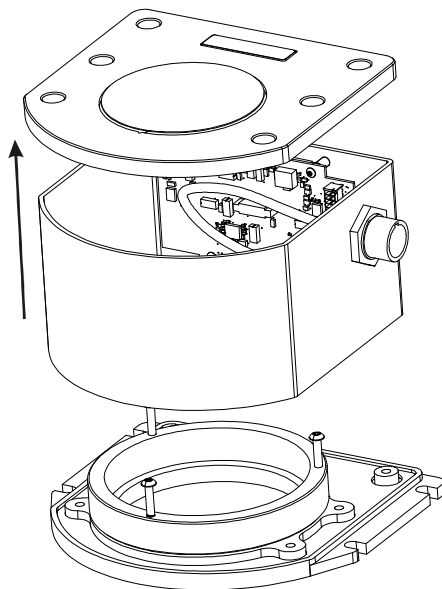
The SPAN IMU disassembly steps are as follows:

1. Remove the top cover's six bolts using a hex key, as shown in *Figure 57*:



**Figure 57: Bolts and Hex Key**

2. Set aside the bolts with their sealing washers.
3. Lift the top cover off the tube body and set it aside, as shown in *Figure 58* on *page 362*.
4. Lift the tube body away from its base plate and set it aside.
5. Remove the 3 ring spacer screws and set them aside.



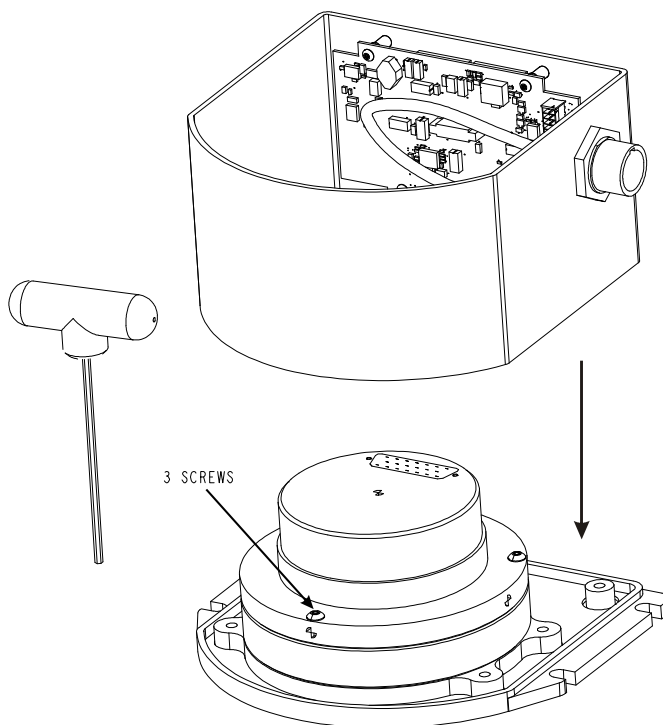
**Figure 58: Lift Top Cover, Tube Body and 3 Ring Spacer Screws**

---

## F.2 Install the HG1700 Sensor Unit

To re-assemble the SPAN IMU with the HG1700 sensor, see *Figure 59* and follow these steps:

1. Mount the HG1700 sensor with the attached #8 screws. Apply threadlock to the screw threads. Use a hex key to torque each screw to 10 in-lbs.
2. Fit the tube body over the HG1700 sensor and onto the base plate.



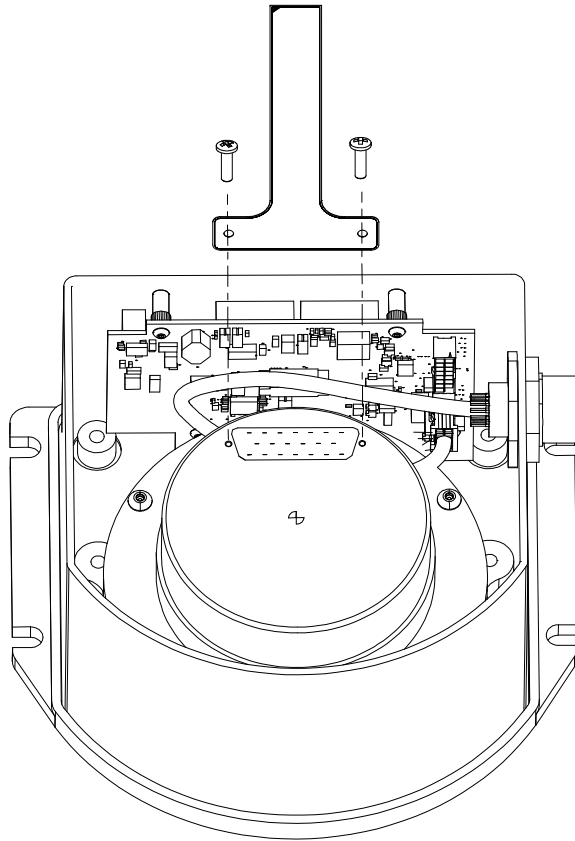
**Figure 59: SPAN IMU Re-Assembly**

---

## F.3 Make the Electrical Connections

To make the electrical connections you will need a 3/32" hex key, the flex cable and the partially assembled SPAN IMU from *Section F.2, Install the HG1700 Sensor Unit on page 363*. 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 60*.



**Figure 60: Attach Flex Cable**

2. Tighten the screws to 4 in-lbs.
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, as shown in *Figure 60*.

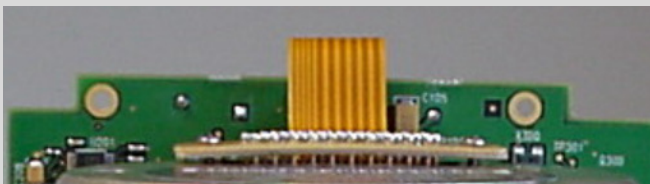
4. Check that the flex cable is locked in place.



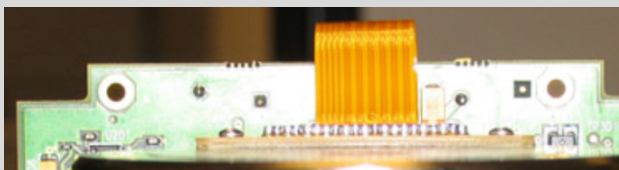
**Important!:**

*Figure 61* shows an incorrect installation of the flex cable where it is bowed in the middle. It will not operate properly in this position.

*Figure 62* shows the proper installation of the flex cable. Notice how the flex cable sits flush against the IMU surface.



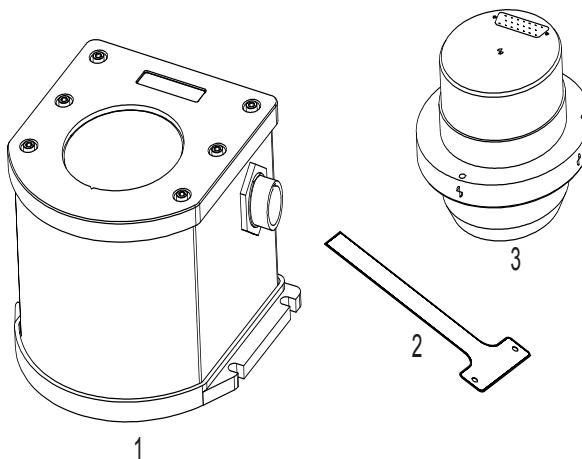
**Figure 61: Incorrect (Bowed) Flex Cable Installation**



**Figure 62: Correct (Flat) Flex Cable Installation**

## F.4 Re-Assemble the SPAN IMU Enclosure

Use a hex key to align the long bolts with the threaded holes in the base, as shown in *Figure 57* on *page 361*. Apply threadlock to threads. Finger tighten all bolts and torque them in a cross pattern to 12 in-lbs. The fully assembled IMU enclosure is shown in *Figure 63*.



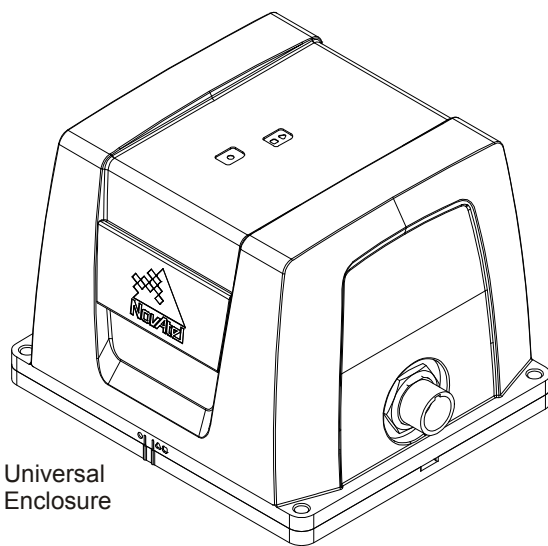
**Figure 63: HG1700 SPAN IMU**



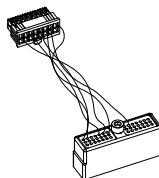
**Important!** Assemble in accordance with applicable industry standards. Ensure all ESD measures are in place, in particular, use a ground strap before exposing or handling any electronic items, including the IMU. Take care to prevent damaging or marring painted surfaces, O-rings, sealing surfaces, and the IMU.

The following procedure provides the necessary information to install the LN-200 sensor into the Universal Enclosure (NovAtel part number 01018590), both illustrated below. The steps required for this procedure are:

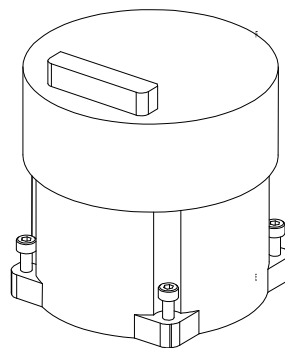
- Disassemble the Universal Enclosure
- Install the LN-200 Sensor Unit
- Reassemble the Universal Enclosure



Universal Enclosure



LN-200  
Wiring Harness



LN-200  
Sensor Unit



1. Use thread-locking fluid on all fasteners except for the cable harness connectors.
2. Torque values for all fasteners, including those for the cable harness screws, are as follows:
 

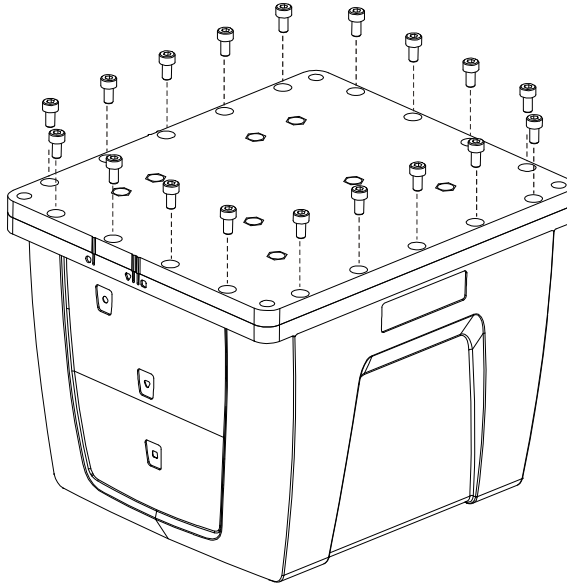
Size 6-32:	0.79-0.90 N-m (7.0-8.0 lb-in)
Size M4:	1.36-1.58 N-m (12.0-14.0 lb-in)

---

## G.1 Disassemble the Universal Enclosure

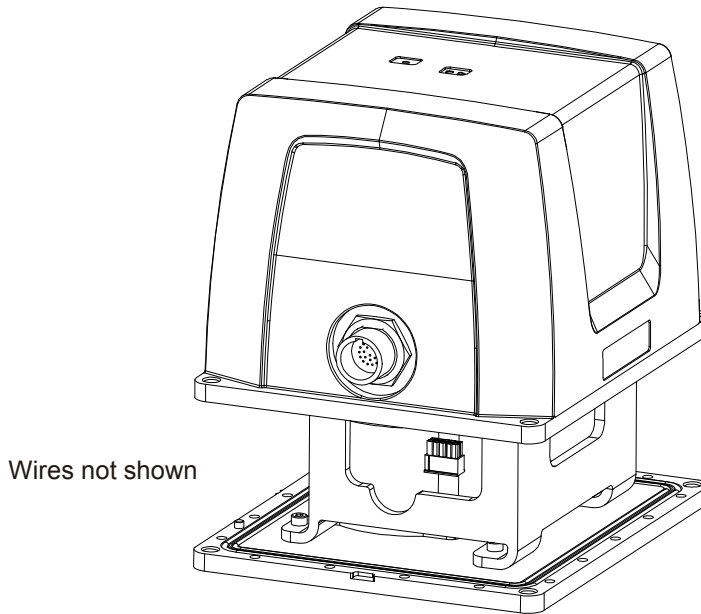
Disassemble the Universal Enclosure as follows:

1. Using a 3 mm hex bit, remove the M4 screws (they will be reused) and the base, as shown in *Figure 64*.



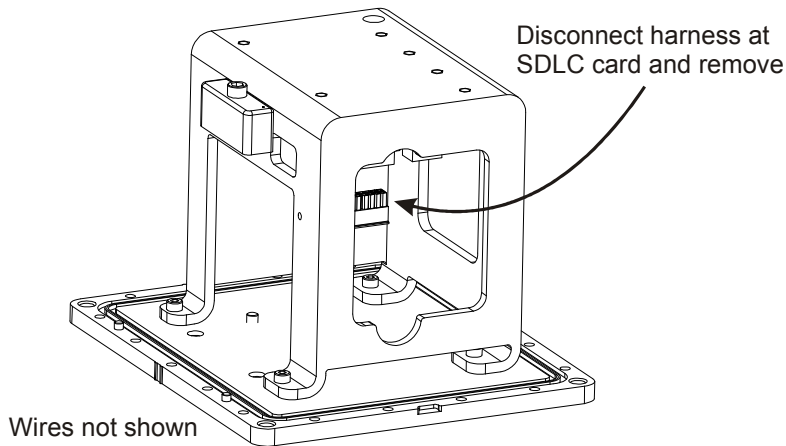
**Figure 64: Remove Base**

2. While squeezing and holding the assembly tightly together, carefully turn the assembly over and set it down as shown in *Figure 65*. Raise the enclosure body, and disconnect the internal cable harness at the SDLC board, as shown. Ensure the O-rings remain with the base when it is removed, and that they are not damaged.



**Figure 65: Disconnect Wiring Harness from SDLC Card**

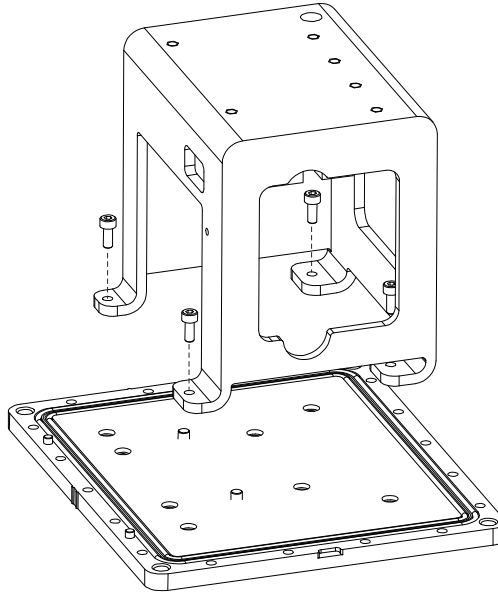
3. Lift the enclosure lid off the assembly to expose the IMU bracket, shown in *Figure 66*. Disconnect the harness at the SDLC card and remove.



**Figure 66: IMU Bracket**



- 
4. Using a 3 mm hex bit, unscrew 4 mm screws and remove the IMU bracket with SDLC, as shown in *Figure 67*.



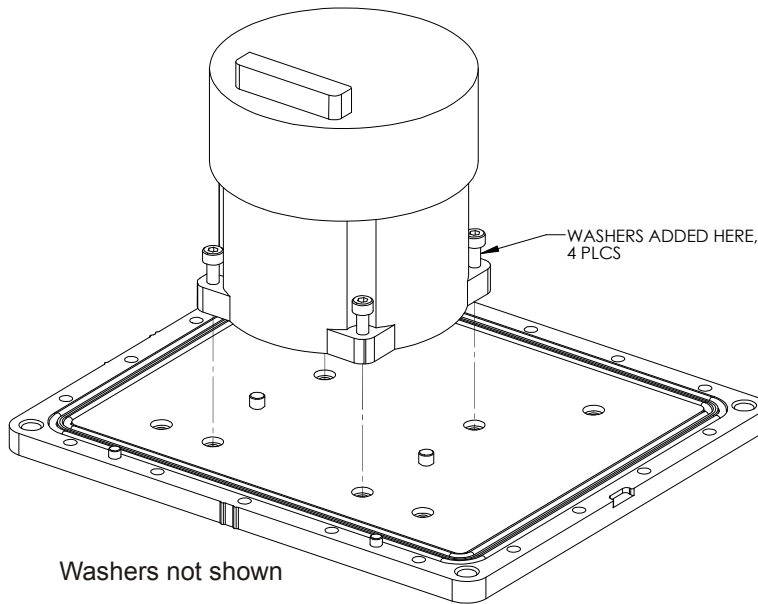
**Figure 67: Remove IMU Bracket/SDLC**

---

## G.2 Install the LN-200 Sensor Unit

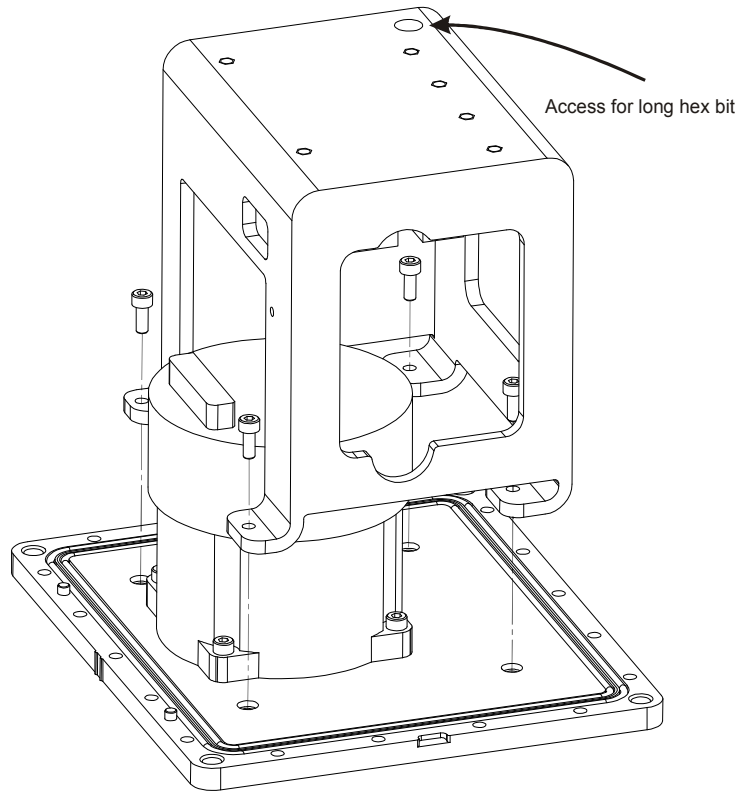
To install the LN-200 sensor unit in the Universal Enclosure:

1. Using a 3 mm hex bit, remove original captive 6-32 screws and washers (4 each) from the LN-200 IMU. Add three washers under each of the original washers and fasten the IMU to the enclosure base, as shown in *Figure 68*. Use thread-locking fluid on each screw.



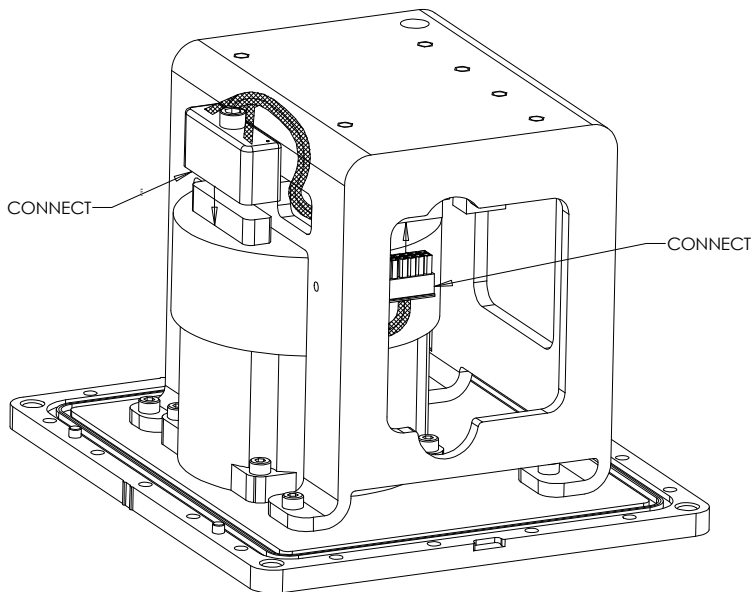
**Figure 68: Install LN-200 IMU to Base**

- 
2. Using a long 3 mm hex bit, install the IMU bracket/SDLC to the base, as shown in *Figure 69*. Use thread-locking fluid on each M4 screw.



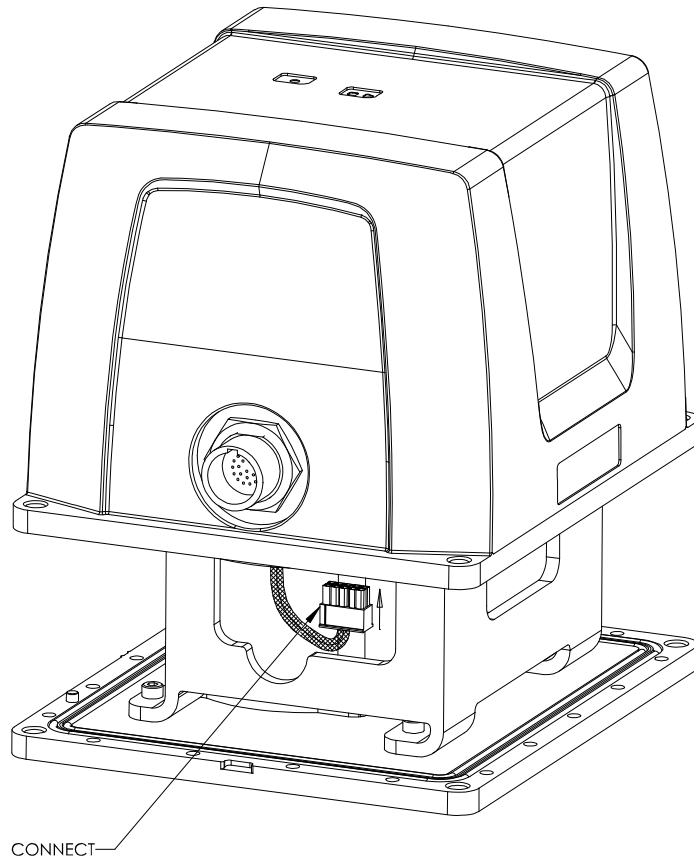
**Figure 69: Install Bracket to Base**

- 
3. Connect the cable harness to the board assembly and IMU, routing it as shown in *Figure 70*. Ensure latching of the cable connector housings and fasten the 6-32 screw at the IMU end using a 5/32" hex bit. Do not use thread-locking fluid and do not overtighten.



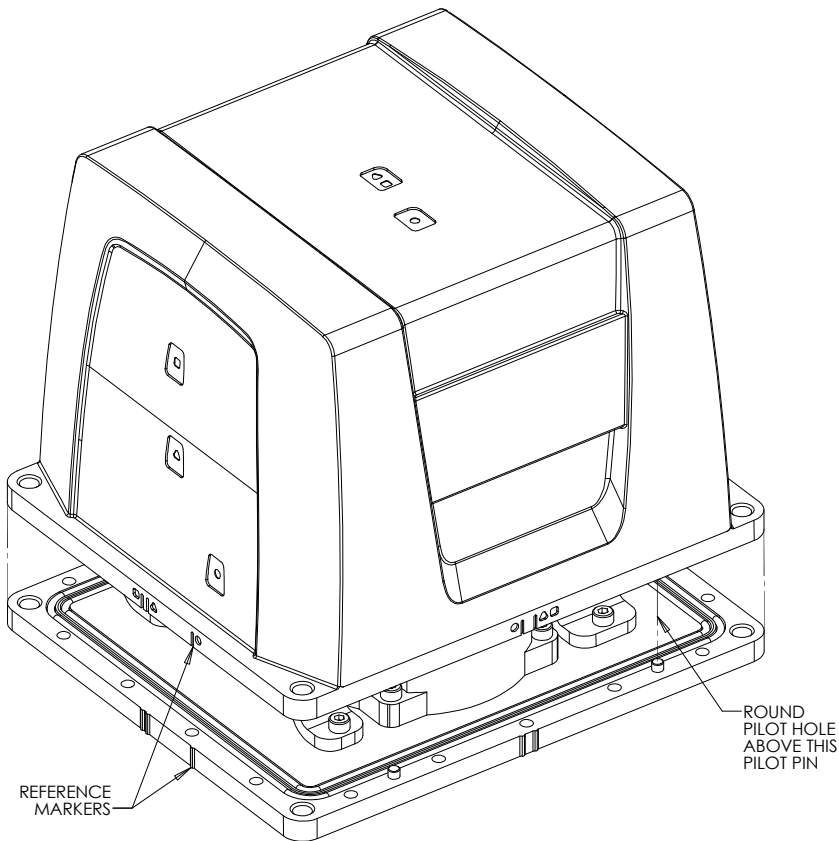
**Figure 70: Making Connections**

- 
4. While carefully holding the body over the bracket, connect the internal cable harness to the board assembly, as shown in *Figure 71*.



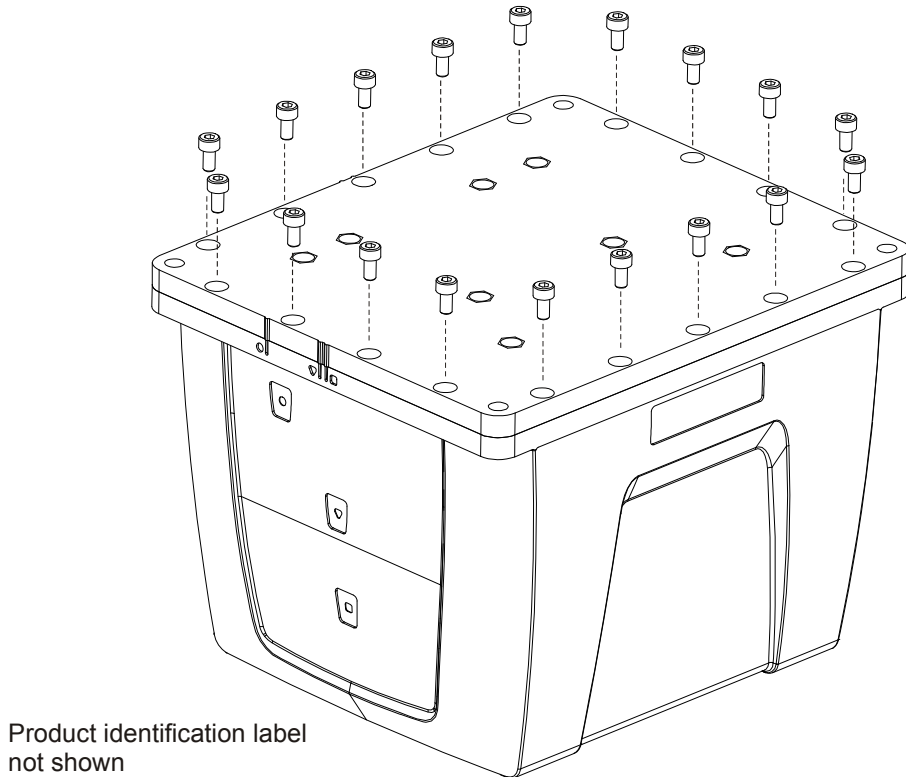
**Figure 71: Connect Internal Cable Harness**

5. Clean the surface of the enclosure body, where it will mate with the O-rings, using isopropyl alcohol. While ensuring all wires will fit inside the bracket without being pinched, align the reference markers and pilot holes/screws of the enclosure body and base, and carefully lower the body onto the base, observing the O-rings and the alignment of corners. Start with the round pilot hole indicated in *Figure 72*, then press the assembly into place.



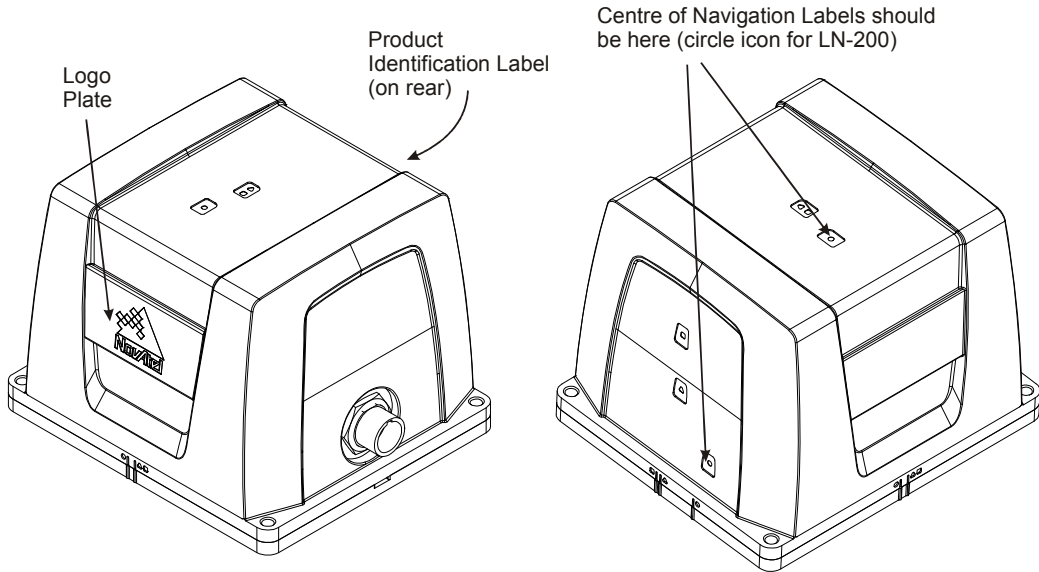
**Figure 72: Installing the Enclosure Body to the Base**

6. While squeezing and holding the enclosure body and base together to maintain tight contact, carefully turn the assembly over and place it on its top, as shown in *Figure 73*. Using a 3 mm hex bit, lightly fasten four equally spaced M4 screws to hold the parts together. Use thread-locking fluid on all screws. Install the remaining screws in similar fashion. Tighten all screws to 1.36-1.58 N-m (12-14 lb-in). Do not over-tighten.



**Figure 73: Screw Enclosure Base to Body**

7. Ensure the product identification label, the logo plate and the centre of navigation labels are properly affixed and contain the correct information. The final assembled unit is shown in *Figure 74*.



**Figure 74: Final Assembly**

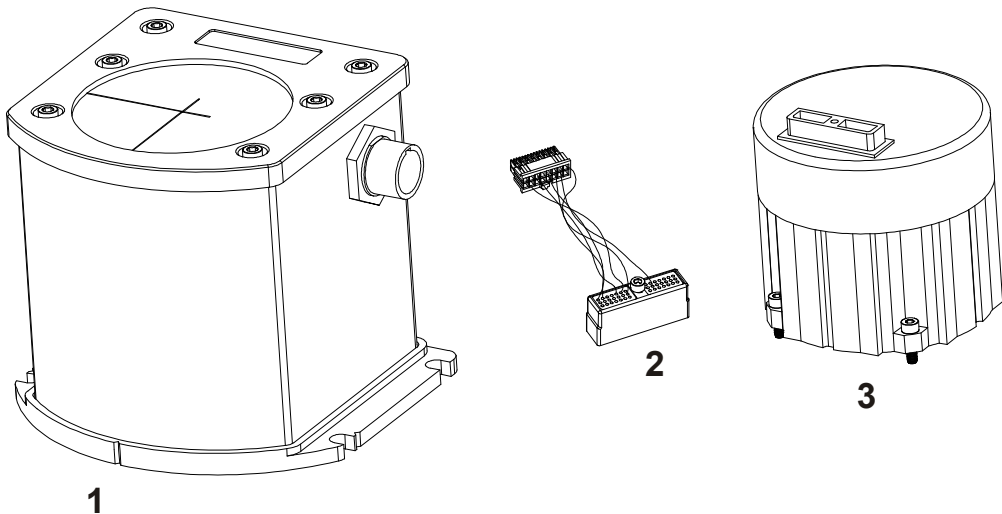


The following procedure, detailed in this appendix, provides the necessary information to install the LN-200 sensor (NovAtel part number 80023515) into the SPAN IMU enclosure (NovAtel part number 01017656) using the LN-200 wiring harness (NovAtel part number 01017655). The steps required for this procedure 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 75: Required Parts**

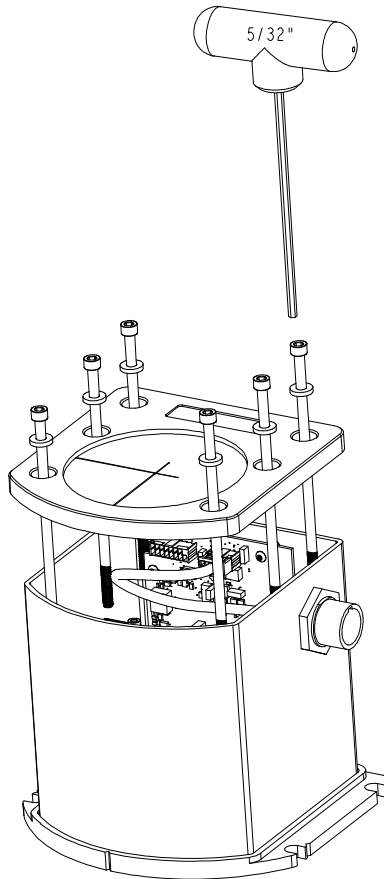
Reference	Description
1	SPAN IMU Enclosure
2	LN-200 Wiring Harness
3	LN-200 Sensor Unit

---

## H.1 Disassemble the SPAN IMU Enclosure

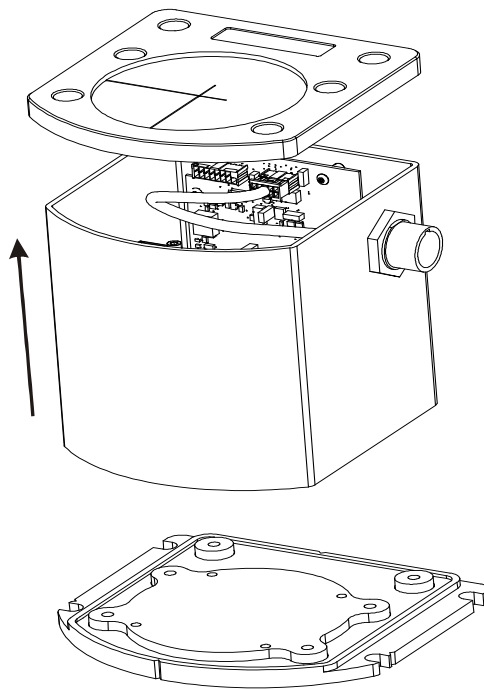
The SPAN IMU disassembly steps are as follows:

1. Remove the top cover's six bolts using a hex key, as shown in *Figure 76*:



**Figure 76: Bolts and Hex 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, as shown in *Figure 77* on *page 379*.



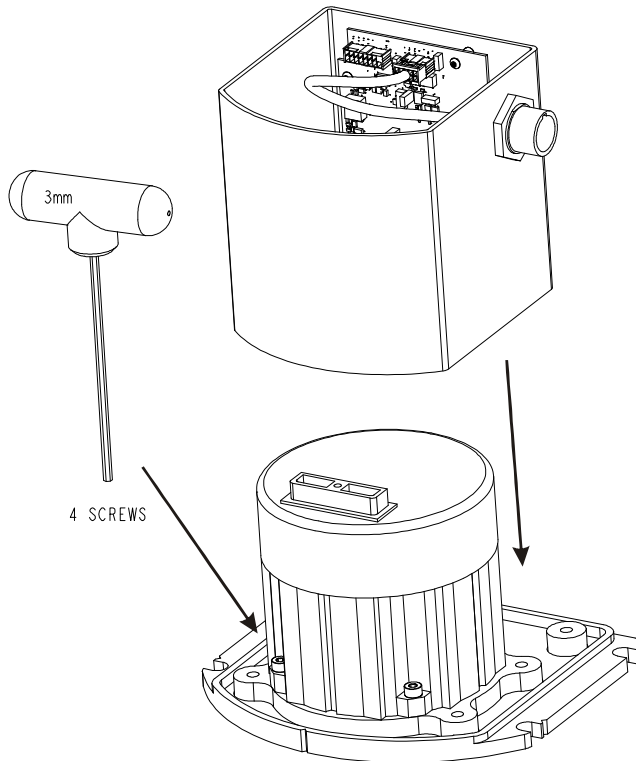
**Figure 77: Lift Top Cover and Tube Body**

---

## H.2 Install the LN-200 Sensor Unit

To re-assemble the SPAN IMU with the LN-200 sensor, follow these steps:

1. Mount the LN-200 sensor with the attached M4 screws. Apply threadlock to the screw threads. Use a hex key to torque each screw to 10 in-lbs.
2. Fit the tube body over the LN-200 sensor and onto the base plate.

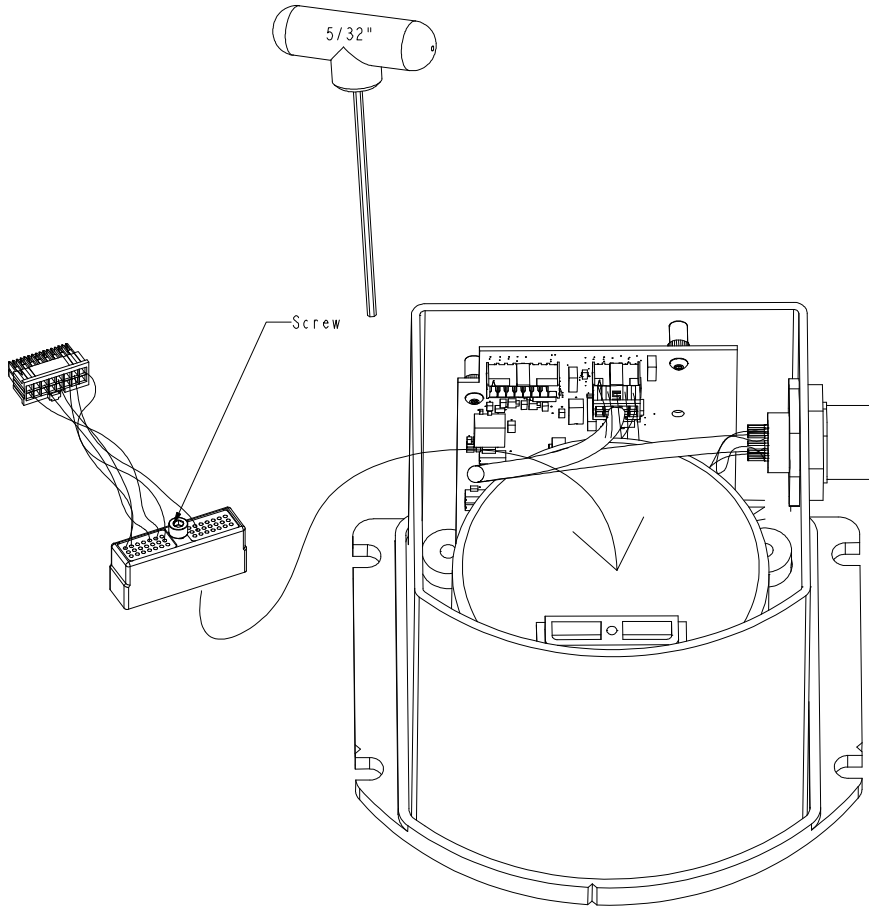


**Figure 78: SPAN IMU Re-Assembly**

## H.3 Make the Electrical Connections

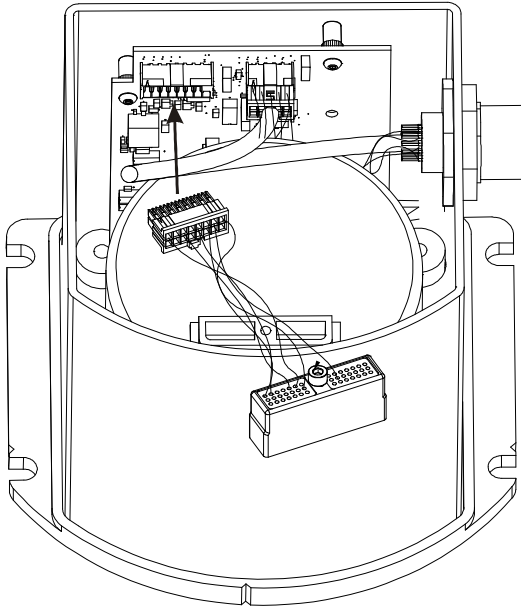
To make the electrical connections you will need a 3/32" hex key, the wiring harness and the partially assembled SPAN IMU from *Section H.2, Install the LN-200 Sensor Unit on page 380*. Now follow these steps:

1. Attach the LN-200 wire harness to the mating connector on the LN-200. Check that the connector is fully seated, as shown in *Figure 79 on page 381*.



**Figure 79: 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, as shown in *Figure 80*. Ensure that the connector is locked in place.

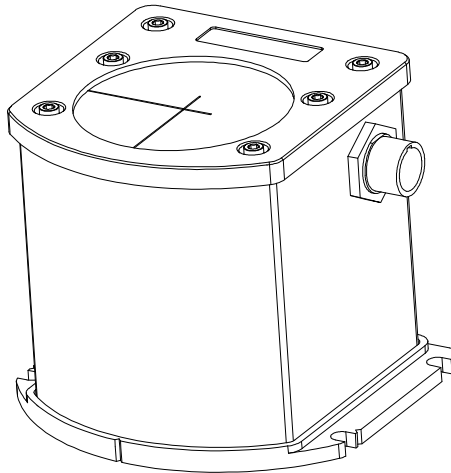


**Figure 80: Attach Samtec Connector**

---

## H.4 Re-Assemble the SPAN IMU Enclosure

Use a hex key to align the long bolts with the threaded holes in the base, as shown in *Figure 76* on *page 378*. Apply threadlock to threads. Finger tighten the 6 bolts then torque them in a cross pattern to 12 in-lbs. The fully assembled IMU enclosure is shown in *Figure 81*.

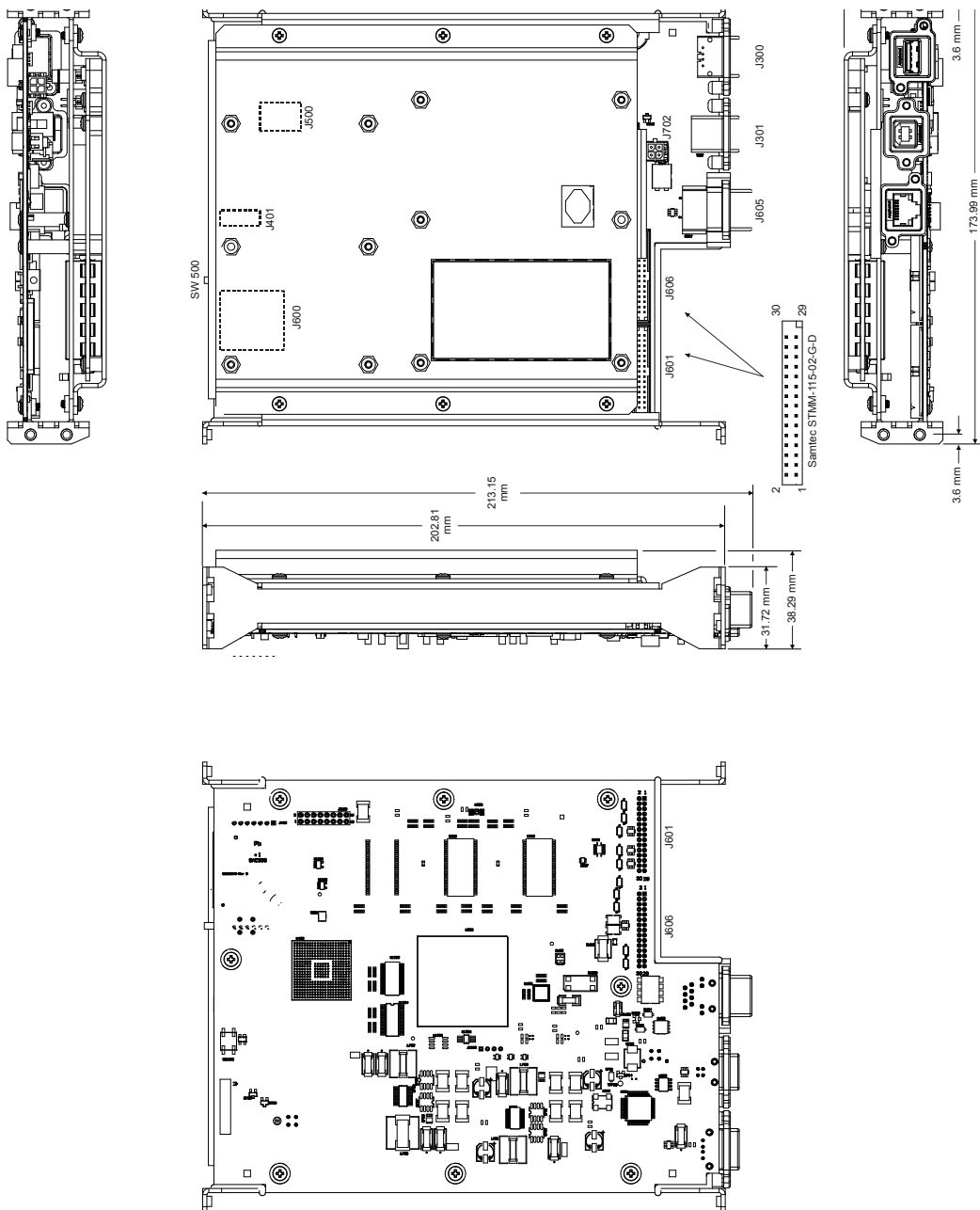


**Figure 81: LN-200 SPAN IMU**

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 82* 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*.





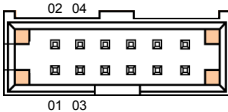
**Figure 82: SPAN-SE Interface Card**

Table 86 provides a description of the interface card headers, and the remaining tables provide pinouts for headers that external users may need to access.

**Table 86: 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 87: J401 (LED Header)**

Pin	Description <sup>a</sup>	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 88: J500 (Power Button Header)**

Pin	Description	Signal Levels
1	LED Positive	5.0 V (120 $\Omega$ series resistor)
2	LED Negative	Return
3	Power Switch	Pull up to 5V
4	GND	

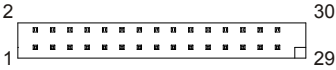
**Table 89: J702 (Input Power Header)**

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 90: 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	
8	IMU CTS	
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 91: 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

This appendix provides technical specifications, mechanical dimensioned drawings, connector pin outs, and cable information for the SPAN-MPPC Interface card (NovAtel part number 01018318). *Figure 83* shows the SPAN-MPPC interface card. *Figure 84* and *Figure 85* show the SPAN-MPPC board stack (NovAtel part number 01018411), which includes a SPAN-MPPC interface card and an OEMV receiver.

For further information on the OEMV receivers, refer to NovAtel technical publication *OM-20000093 OEMV Family Installation and Operation User Manual*.

## J.1 SPAN-MPPC Mechanical Drawings

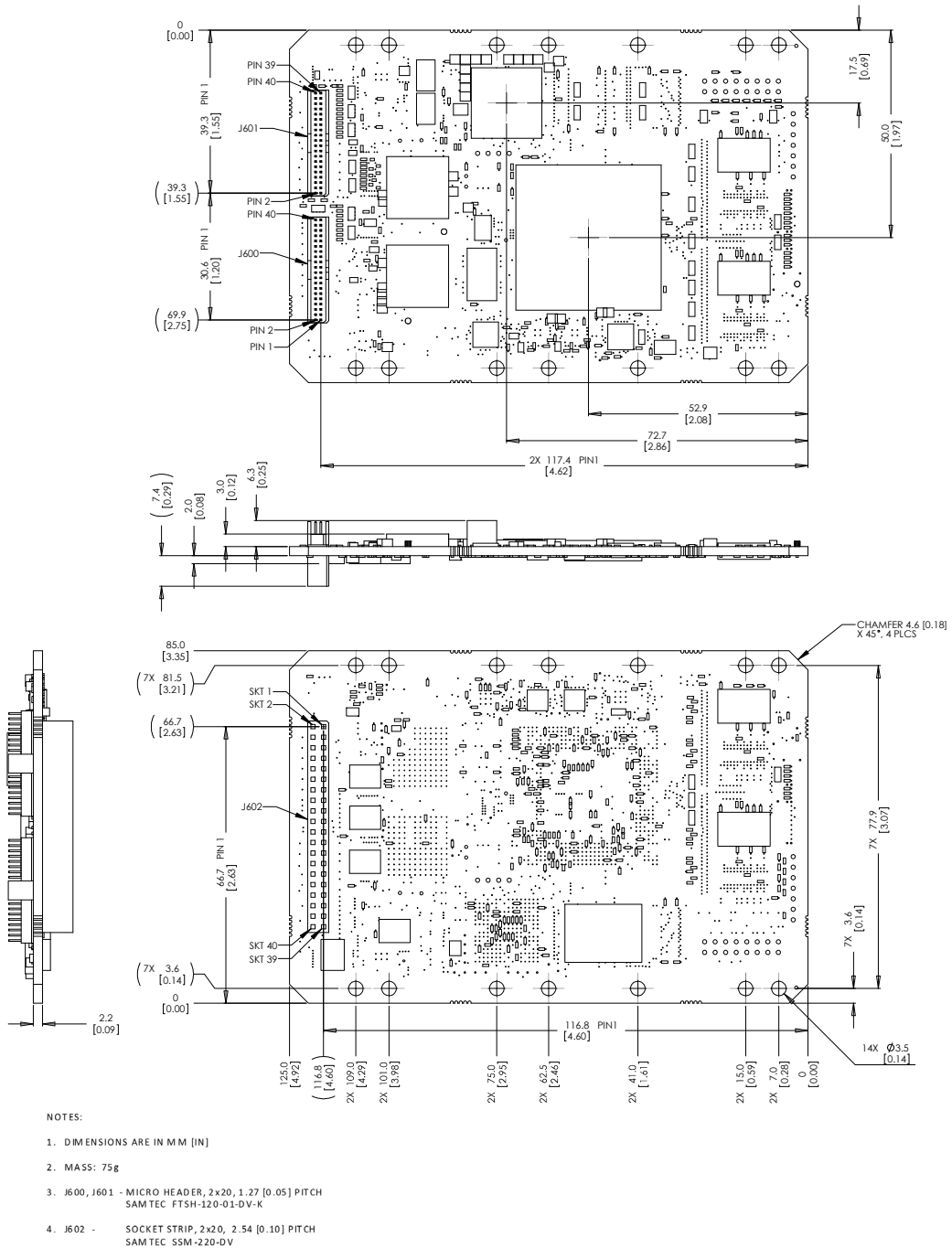
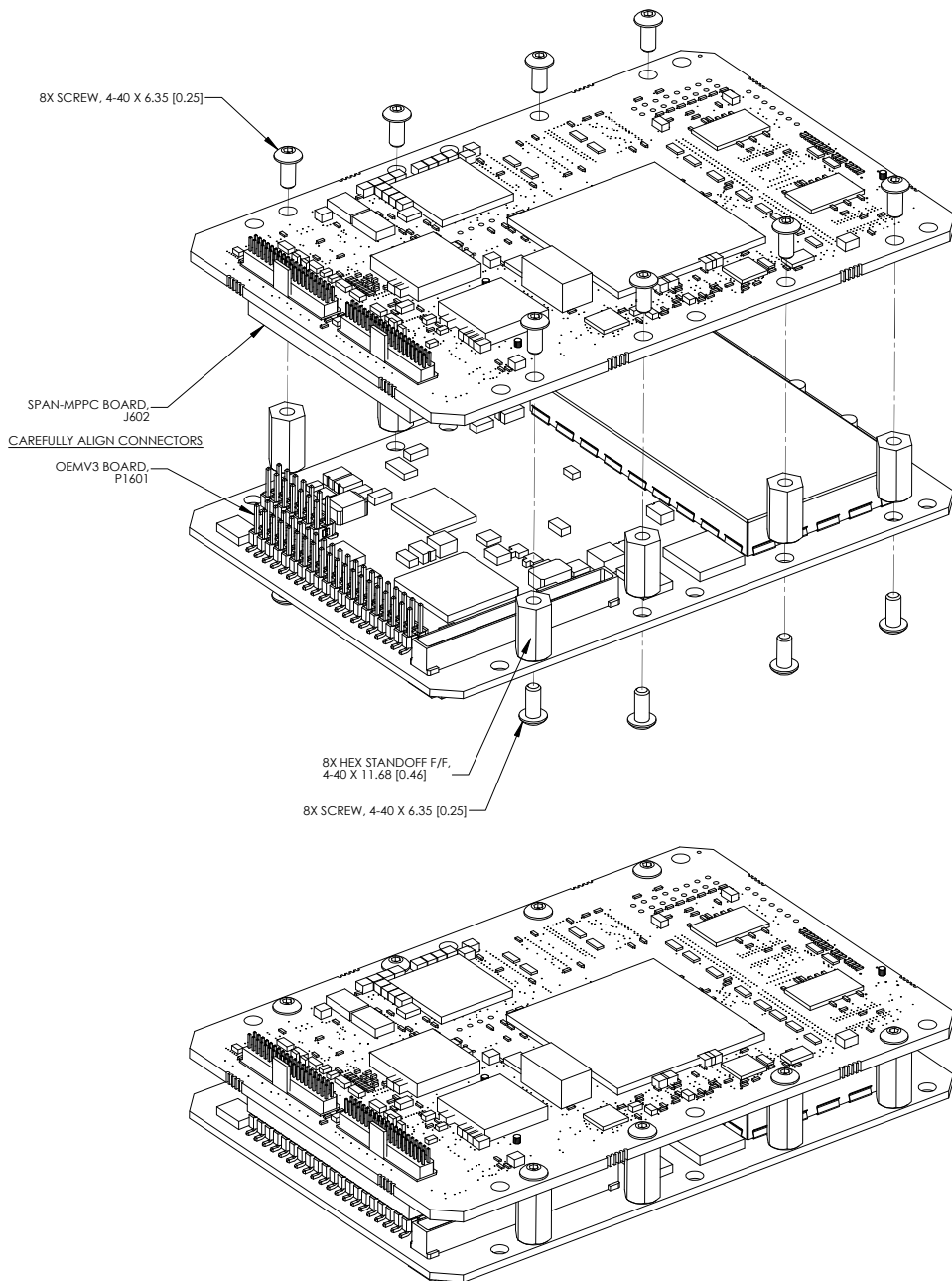
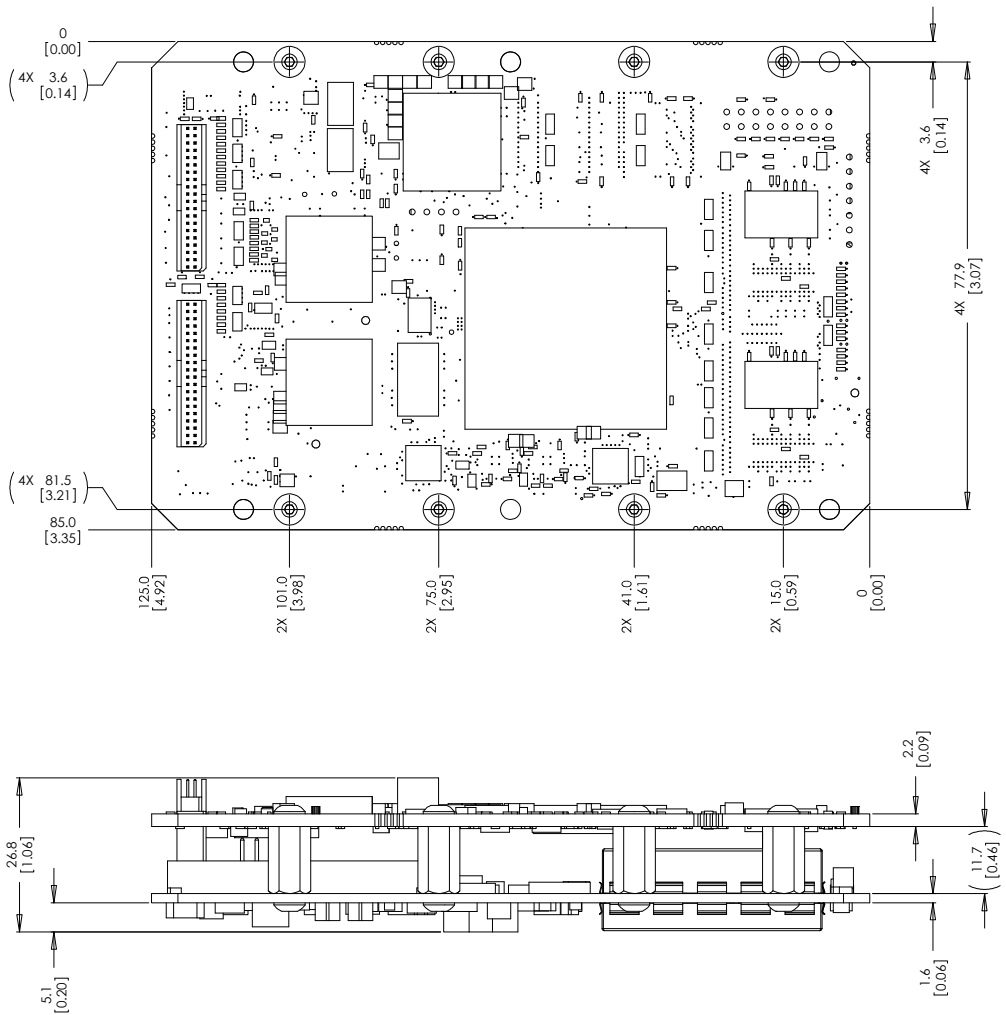


Figure 83: SPAN-MPPC Interface Card



**Figure 84: SPAN-MPPC Board Stack (perspective view)**





**Figure 85: SPAN-MPPC Board Stack (dimensions)**

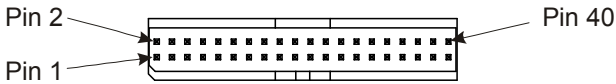
# J.2 SPAN-MPPC Connector Pin-Outs

Table 92 provides a description of the SPAN-MPPC interface card headers.

Table 92: SPAN-MPPC Interface Card Header Description

Header ID	Function	Connector Format
J600	Multi Communication Port A	2x20 pin header, male, 2mm pitch
J601	Multi Communication Port B	2x20 pin header, male, 2mm pitch

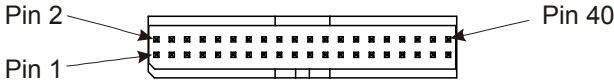
Table 93: J600 (Multi Communication Header A)

Pin	Description	Signal Levels
		
1	+Vin	+9 to +30V DC
2	Ground	
3	+Vin	+9 to +30V DC
4	Ground	
5	+Vin	+9 to +30V DC
6	Ground	
7	+Vin	+9 to +30V DC
8	Ground	
9	+Vin	+9 to +30V DC
10	Ground	
11	+Vin	+9 to +30V DC
12	Ground	
13	Ground	
14	Ground	
15	USB Host (-)	
16	USB Host (+)	
17	USB Host VBUS	
18	Ground	
19	USB Device (-)	

Continued on the following page

Pin	Description	Signal Levels
20	USB Device (+)	
21	USB Device VBUS	
22	Ground	
23	Ethernet TX (-)	
24	Ethernet TX (+)	
25	Ground	
26	Ground	
27	Ethernet RX (-)	
28	Ethernet RX (+)	
29	Ground	
30	Ground	
31	COM1 RX (-) [RS422] COM1 CTS [RS232]	Serial port is software configurable for RS232 or RS422 operation
32	COM1 RX (+) [RS422] COM1 RxD [RS232]	
33	COM1 TX (+) [RS422] COM1 TxD [RS232]	
34	COM1 TX (-) [RS422] COM1 RTS [RS232]	
35	Ground	
36	Ground	
37	IMU Port RX (-) [RS422] IMU Port CTS [RS232]	
38	IMU Port RX (+) [RS422] IMU Port RxD [RS232]	
39	IMU Port TX (-) [RS422] IMU Port RTS [RS232]	
40	IMU Port TX (+) [RS422] IMU Port TxD [RS232]	

**Table 94: J601 (Multi Communication Header B)**

Pin	Description	Signal Levels
		
1	OEMV Port CTS [RS232]	Connection to OEMV-3 receiver UART COM1, RS-232 only
2	OEMV Port RxD [RS232]	
3	OEMV Port RTS [RS232]	
4	OEMV Port TxD [RS232]	
5	Ground	
6	Event-out 1 Strobe	Software-configurable GPS synchronous output strobe [LVTTTL level]
7	Event-out 2 Strobe	Software-configurable GPS synchronous output strobe [LVTTTL level]
8	Event-out 3 Strobe	Software-configurable GPS synchronous output strobe [LVTTTL level]
9	Event-out 4 Strobe	Software-configurable GPS synchronous output strobe [LVTTTL level]
10	Ground	
11	Event-in 1 Strobe	Software-configurable LVTTTL level input event trigger
12	Event-in 2 Strobe	Software-configurable LVTTTL level input event trigger
13	Event-in 3 Strobe	Software-configurable LVTTTL level input event trigger
14	Event-in 4 Strobe	Software-configurable LVTTTL level input event trigger
15	Ground	
16	Ground	
17	CAN 1 (-)	
18	CAN 1 (+)	
19	CAN 2 (-)	
20	CAN 2 (+)	
21	Ground	
22	Ground	

*Continued on the following page*

Pin	Description	Signal Levels
23	COM2 RX (+) [RS422] COM2 RxD [RS232]	
24	COM2 RX (-) [RS422] COM2 CTS [RS232]	
25	COM2 TX (+) [RS422] COM2 TxD [RS232]	
26	COM2 TX (-) [RS422] COM2 RTS [RS232]	
27	Ground	
28	Ground	
29	COM3 RX (+) [RS422] COM3 RxD [RS232]	
30	COM3 RX (-) [RS422] COM3 CTS [RS232]	
31	COM3 TX (+) [RS422] COM3 TxD [RS232]	
32	COM3 TX (-) [RS422] COM3 RTS [RS232]	
33	COM4 RX (+) [RS422] COM4 RxD [RS232]	
34	COM4 RX (-) [RS422] COM4 CTS [RS232]	
35	COM4 TX (+) [RS422] COM4 TxD [RS232]	
36	COM4 TX (-) [RS422] COM4 RTS [RS232]	
37	Ground	
38	Ground	
39	ARINC (+)	
40	ARINC (-)	

### J.3 SPAN MPPC Breakout Board

The SPAN MPPC Breakout Board (NovAtel part number 01018504), connected to the MPPC stack, is shown in Figure 86.

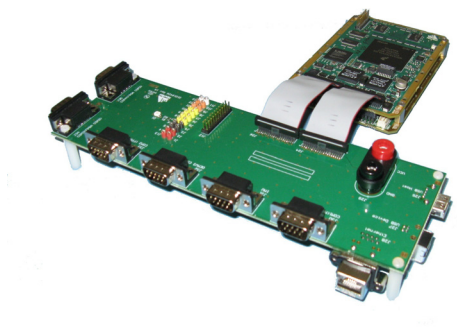


Figure 86: SPAN MPPC Breakout Board

The physical location of the connectors on the Breakout Board is shown in Figure 87:

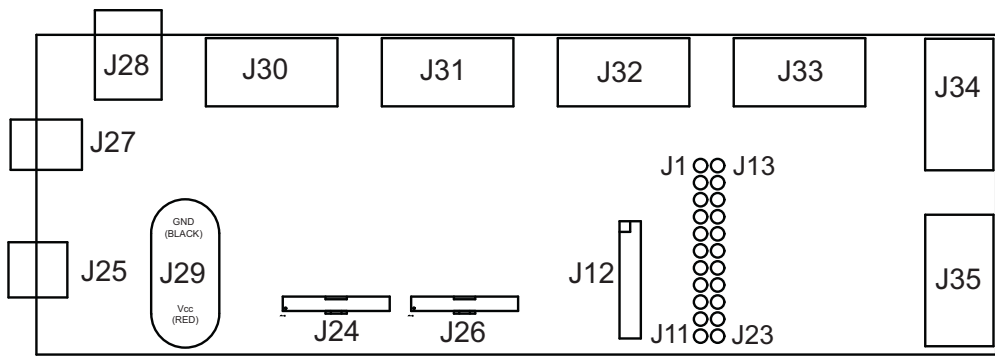


Figure 87: SPAN MPPC Breakout Board - Connectors

The SPAN MPPC Breakout Board provides access to SPAN MPPC signals, as outlined in the following tables.

Table 95: SPAN MPPC Breakout Board - J1-J23 Test Points

Pin	Description	Pin	Description	Color
J1	Vin	J13	Vin	Red
J2	Vin	J14	Vin	Red
J3	GND	J15	GND	Black
J4	GND	J16	GND	Black
J5	Event out1o	J17	Event out2o	Yellow
J6	Event out3o	J18	Event out4o	Yellow

Pin	Description	Pin	Description	Color
J7	Event in1o	J19	Event in2o	Yellow
J8	Event in3o	J20	Event in4o	Yellow
J9	CAN1o (-)	J21	CAN1o (+)	Orange
J10	CAN2o (+)	J22	CAN2o (+)	Orange
J11	ARINCo (+)	J23	ARINCo (-)	White

**Table 96: SPAN MPPC Breakout Board - J24 Main Board Connector**

Pin	Description
1	Vin
2	GND
3	Vin
4	GND
5	Vin
6	GND
7	Vin
8	GND
9	Vin
10	GND
11	Vin
12	GND
13	GND
14	GND
15	USBHo (-)
16	USBHo (+)
17	USBH Vbus
18	GND
19	USBD0 (-)
20	USBD0 (+)
21	USBD Vbus
22	GND
23	ETH TXo (-)
24	ETH TXo (+)
25	GND
26	GND

Pin	Description
27	ETH RXo (-)
28	ETH RXo (+)
29	GND
30	GND
31	COM1 RXo (-)
32	COM1 RXo (+)
33	COM1 TXo (+)
34	COM1 TXo (-)
35	GND
36	GND
37	IMU RXo (-)
38	IMU RXo (+)
39	IMU TXo (-)
40	IMU TXo (+)

**Table 97: SPAN MPPC Breakout Board - J25 USB Host Connector**

Pin	Description
1	USBH Vbus
2	USBHo (-)
3	USBHo (+)
4	GND
5	SHELL
6	SHELL

**Table 98: SPAN MPPC Breakout Board - J26 Main Board Connector**

Pin	Description
1	OEMV3 RXo (-)
2	OEMV3 RXo (+)
3	OEMV3 TXo (-)
4	OEMV3 TXo (+)
5	GND
6	Event out1o
7	Event out2o



Pin	Description
8	Event out3o
9	Event out4o
10	GND
11	Event in1o
12	Event in2o
13	Event in3o
14	Event in4o
15	GND
16	GND
17	CAN1o (-)
18	CAN1o (+)
19	CAN2o (-)
20	CAN2o (+)
21	GND
22	GND
23	COM2 RXo (+)
24	COM2 RXo (-)
25	COM2 TXo (+)
26	COM2 TXo (-)
27	GND
28	GND
29	COM3 RXo (+)
30	COM3 RXo (-)
31	COM3 TXo (+)
32	COM3 TXo (-)
33	COM4 RXo (+)
34	COM4 RXo (-)
35	COM4 TXo (+)
36	COM4 TXo (-)
37	GND
38	GND
39	ARINCo (+)
40	ARINCo (-)

---

**Table 99: SPAN MPPC Breakout Board - J27 USB Device Connector**

Pin	Description
1	USBD Vbus
2	USBD <sub>o</sub> (-)
3	USBD <sub>o</sub> (+)
4	GND
5	SHELL
6	SHELL

**Table 100: SPAN MPPC Breakout Board - J28 Ethernet Connector**

Pin	Description
1	ETH TX <sub>o</sub> (+)
2	ETH TX <sub>o</sub> (-)
3	ETH RX <sub>o</sub> (+)
4	Not used
5	Not used
6	ETH RX <sub>o</sub> (-)
7	Not used
8	Not used

**Table 101: SPAN MPPC Breakout Board - J29 Power Connector**

Pin	Description
1	PWR
2	GND

**Table 102: SPAN MPPC Breakout Board - J30 Serial Port Connection**

Pin	Description
1	Not used
2	COM1 RX <sub>o</sub> (+)
3	COM1 TX <sub>o</sub> (+)
4	Not used
5	GND
6	Not used
7	COM1 TX <sub>o</sub> (-)

Pin	Description
8	COM1 RXo (-)
9	Not used

**Table 103: SPAN MPPC Breakout Board - J31 Serial Port Connection**

Pin	Description
1	Event_out4o
2	IMU RXo (+)
3	IMU TXo (+)
4	Not used
5	GND
6	Not used
7	IMU TXo (-)
8	IMU RXo (-)
9	Not used

**Table 104: SPAN MPPC Breakout Board - J32 Serial Port Connection**

Pin	Description
1	Not used
2	OEMV3 RXo (+)
3	OEMV3 TXo (+)
4	Not used
5	GND
6	Not used
7	OEMV3 TXo (-)
8	OEMV3 RXo (-)
9	Not used

**Table 105: SPAN MPPC Breakout Board - J33 Serial Port Connection**

Pin	Description
1	Not used
2	COM2 RXo (+)
3	COM2 TXo (+)

Pin	Description
4	Not used
5	GND
6	Not used
7	COM2 TXo (-)
8	COM2 RXo (-)
9	Not used

**Table 106: SPAN MPPC Breakout Board - J34 Serial Port Connection**

Pin	Description
1	Not used
2	COM3 RXo (+)
3	COM3 TXo (+)
4	Not used
5	GND
6	Not used
7	COM3 TXo (-)
8	COM3 RXo (-)
9	Not used

**Table 107: SPAN MPPC Breakout Board - J35 Serial Port Connection**

Pin	Description
1	Not used
2	COM4 RXo (+)
3	COM4 TXo (+)
4	Not used
5	GND
6	Not used
7	COM4 TXo (-)
8	COM4 RXo (-)
9	Not used

## J.4 SPAN-MPPC to IMU Connection

This section outlines the connections that are required between the SPAN-MPPC and either an IMU Interface Cable, a Deutsch connector on an LN200 or HG1700 enclosure, or a NovAtel Power Communication Board (used with an LN200 or HG1700 IMU).



Pin 36 on the J600 connector is one of many available ground pins. See *Table 93, J600 (Multi Communication Header A)* on page 394 if an alternate ground connection is required.

**Table 108: SPAN-MPPC IMU Connections**

SPAN-MPPC Multi Communication Header A <sup>a</sup>	IMU DB-9 Interface Cable <sup>b</sup>	8 Position Molex	Power Communication Board P100 Header <sup>c</sup>
J601 - Pin 9	1 <sup>d</sup>	4 <sup>e</sup>	-
J600 - Pin 36	5	-	10
J600 - Pin 37	8	-	7
J600 - Pin 38	2	-	2
J600 - Pin 39	7	-	1
J600 - Pin 40	3	-	6

- a. J600 (Multi Communication Header A) on the SPAN-MPPC Interface card. See *Figure 83, SPAN-MPPC Interface Card* on page 391.
- b. Female DB-9 connector on the interface cable that runs from the receiver to the LN200 or HG1700 IMU enclosure.
- c. P100 header on the Power Communication Board (if there is no IMU Interface Cable or IMU enclosure).
- d. Only needed for use with iIMU-0FSAS
- e. Only needed for use with iIMU-CPT

# Appendix K 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.

## K.1 SPAN System

Part Description	NovAtel Part
IMUs (see <i>Table 1, Supported IMU Types</i> on <i>page 32</i> for details)	IMU-H58 IMU-H62 UIMU-H58 UIMU-H62 IMU-LN200 UIMU-LN200 IMU-SE-FSAS-EI IMU-LCI UIMU-CPT
Receivers (see <i>Table 1, Supported IMU Types</i> on <i>page 32</i> for details)	SPAN-SE SPAN-MPPC
ProPak-V3 to LN-200 IMU interface cable, see <i>Figure 33</i> on <i>page 110</i>	01017375
LN-200 power adapter cable, see <i>Figure 35</i> on <i>page 112</i>	01017821
ProPak-V3 to iIMU-FAS IMU interface cable, see <i>Table 16</i> on <i>page 115</i>	60723086
ProPak-V3 to HG1700 IMU interface cable (identical to LN-200 cable), see <i>Figure 33</i> on <i>page 110</i>	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
External Cable Harness FSAS IMU	01018388
External Cable Harness LiteF LCI-1 IMU (Universal IMU Enclosure Interface Cable)	01018299
KVH IMU-CPT terminated cable	60723114
OEMV, Connect and Convert4 disk (refer to <i>page 44</i> of this manual and to the <i>OEMV Family Installation and Operation User Manual</i> )	01017827
SPAN MPPC Breakout Board (with cables and packaging)	01018504
SPAN-SE User Guide	OM-20000124
SPAN Technology for OEMV User manual	OM-20000104
OEMV Family Installation and Operation User Manual	OM-20000093
OEMV Family Firmware Reference Manual	OM-20000094

## K.2 Accessories and Options

Part Description	NovAtel Part
Optional NovAtel GPSAntennas:	For aerodynamic applications ANT-A72GA-TW-N
	For high-accuracy applications GPS-702
	For L-band applications GPS-702L
	For high-performance base-station applications ANT-C2GA-TW-N
Optional RF Antenna Cable:	5 meters GPS-C006
	15 meters GPS-C016

---

## K.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	MIL Part
10-pin LEMO plug connector on the HG1700 interface cables	FGG.1K.310.CLAC60Z	-
Amphenol (or MIL equivalent) 22-pin connector on the universal IMU interface cable	TV-06-RW-13-35-S-A	MIL-DTL-38999
MIL equivalent 13-pin connector on the LN-200 interface cable	-	D38999/26B35SF
MIL equivalent 3-pin connector on the LN-200 power cable	-	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	S20K0C-P04MFG0-50EP [ROHS]	

## Appendix L Frequently Asked Questions

1. *I don't hear any sound from my IMU. Why?*
  - a. The LN-200, iIMU-FSAS, IMU-CPT, HG1900, HG1930 and Litef IMU 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.
2. *What types of IMUs are supported?*
  - a. SPAN currently supports the HG1700, HG1900 and HG1930 IMU family from Honeywell, the LN-200 from Litton, the iIMU-FSAS from iMAR, the IMU-CPT from KVH, and the Litef from Northrop Grumman. Use the SETIMUTYUPE command to specify the type of IMU used (see [page 186](#)).
3. *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.
4. *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 MARKxPVA 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 D, Data Logs](#) starting on [page 213](#).
5. *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 Hz 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.



---

6. *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 278*.

7. *What does it mean if my IMUCARD version string looks like this?:*

< IMUCARD "Test mode 20Hz" "" "" "r2.1.0.0" "" "Sep 13 2010" "09:34:20"

This means that the SPAN enabled receiver has detected the SDLC card and is communicating with card, but the SDLC card is not communicating with the IMU. Check that the SDLC to IMU connections to ensure both power and communication lines are connected to the IMU.

## Numerics

2-D 261

3-D 261

## A

abbreviated ascii 125, 217

accelerometers 319

accumulated doppler range (ADR) 312

accuracy 184, 222, 229

acquisition 313

ADR, *see* accumulated doppler range

AG11/AG58, AG17/AG62 187

age, solution 229, 232, 234

ALIGN

    configuring for SPAN-SE-D 79

alignment 55–57, 154, 184

ALIGNMENTMODE command 129

almanac 146

    data 253

    time status 222

ambiguity, half cycle 312

angular offset 63

antenna 184, 406

    altitude 256

    motion 171

antenna motion 171

API functionality 82

APPLICATION command 130

APPLICATIONSTATUS log 228

APPLYVEHICLEBODYROTATION  
command 131

ascii 125, 215

    abbreviated 217

ASSIGNLBAND command 132

asterisk 215

asynchronous log 214

atmospheric delay 312

attitude 154, 276, 284, 306

authorization 199–200

authorization code 199

automated set ups 198

axes

    enclosure frame 182, 184, 192, 237,  
    319

    local level frame 279–280, 290

    SPAN frame 181–182, 184, 188–190,  
    276–278, 280, 345

azimuth 181, 184, 237, 264, 276

## B

baseline, heading 270

baud rate, *see* bps

beam frequencies 133

BESTGPSPOS log 229

BESTGPSVEL log 235

BESTLEVERARM log 237

BESTPOS log 229

BESTVEL log 235

binary 68, 125, 150, 218, 318

bit rate, *see* bps

bps 137

break 135, 137, 311

broadcast almanac 253

buffer 159

Built-In Status Test (BIT) 331

byte 219

## C

C Model 346

C/No, *see* carrier to noise density ratio

cables

    antenna 406

    IMU interface 106, 109, 113

    power 40

    warranty 18

calibration 59, 63, 174, 237

card configuration 148

carrier to noise density ratio (C/No) 265,  
299, 315

CD command 127

channel

    range measurement 312

    tracking status 314, 316

checksum 215, 218

clock

- adjust 312
- internal 222
- offset 308
- receiver 342
- set 312
- status 308
- validity 307
- CMR 150
  - dgps type 167
- coarse alignment 56
- COM command 135
- COM port 160, 238, 310
- COMCONFIG command 238
- COMCONTROL command 138
- command
  - as a log 125
  - formats 125
  - prompt interface 210–212
- communication 46
- COMPROTOCOL log 240
- computation frame 45
- COMVOUT command 140
- configuration 346
  - alignment 55
  - non-volatile memory 146
  - port 135, 238
  - receiver 169, 329, 331
  - reset 169
  - save 175
- connector
  - antenna 88
  - green 94
  - input/output 88
  - yellow 96
- coordinated universal time (UTC)
  - log 268
  - offset 308
  - position 256, 263
  - status 342
- copyright 2
- correction
  - RTCA 150
  - RTK 346
- CORRIMUDATA log 241
- CPU 159
- cyclic redundancy check (CRC) 217–219

## D

- D model 78, 346
- data collection 58, 65, 69
- datum 232
  - best position 234
- DC, antenna 88
- default, factory 169
- DGPS command 263
- differential 229
- differential age 229, 232, 234
- differential correction
  - station 166, 172
- dilution of precision (DOP) 256
  - NMEA 260
- DIR command 127
- DIRENT log 243
- distance exceeded 231
- Doppler 315
  - accumulated 312, 317
  - instantaneous 315
  - range record 316
- DOS commands 126
- driving 63
- dual-antenna
  - functionality 78
  - installation 78
  - overview 78
- DUALANTENNAPORTCONFIG command 141
- dynamic 171
- Dynamic Host Configuration Protocol (DHCP) 76

## E

- east 290
- echo 137
- elevation, satellite visibility 264
- e-mail 21
- enclosure 32
- enclosure frame 45
- ephemeris
  - GLONASS 249
  - raw data 318
  - time status 222
- error
  - flag 331
  - multipath 312

- parity 311
- text description 338
- tracking 312
- ethernet 76–77, 153
- event 74, 142–143
  - message 338
  - text description 338
  - type 338
- EVENTINCONTROL command 142
- EVENTOUTCONTROL command 143
- expiry date 339
- EXTHDGOFFSET command 144
- EXTHDGOFFSET log 245

## F

- factory default
  - modify 175
  - reset 169
  - setting 135
- fast alignment 63
- features 31
- filter, RTK 170
- fine time 222
- firmware updates 22, 31
- fix
  - data 255
  - position 257
  - save setting 175
  - solution 170
- flag
  - error 331
  - parity 312
  - status 331
- float solution 170
- format 215, 217, 225
- FORMAT command 127, 145
- formats, command 125
- frame
  - see* axes
  - vehicle 57, 174, 208
- frequently asked questions 408
- FRESET command 146

## G

- G model 346
- GIMBALLEDPVA command 246
- GIMBALSPANROTATION command 147

- GLOCLOCK log 247
- GLOEPHEMERIS log 249
- GLORAWEPHEM log 252
- GNSSCARDCONFIG command 148
- GPALM log 253
- GPGGA log 255
- GPGLL log 257
- GPGRS log 259
- GPGSA log 260
- GPGST log 262
- GPGSV log 264
- GPHDT NMEA Heading log 266
- GPVTG log 267
- GPZDA log 268
- graphical user interface 49

## H

- handshaking 136–138
- hardware
  - reset 169
  - setup 33
- HDOP, *see* dilution of precision
- header 213, 221
  - ascii 215
  - log 312
- HEADING log 269
- health, almanac 254
- Heave 180, 271
- HEAVE log 271
- height 231–232
  - position 234
- help 46
- hexadecimal 215, 218
- HG1700
  - installation
    - Universal IMU Enclosure 349
- HG1700 installation
  - sensor 353, 363
  - SPAN HG enclosure 360
- hold 159, 161–162, 302–303
- HP/XP, OmniSTAR
  - expiration date 294
  - status 297, 299
  - tracking state 295

## I

- I model 346
- ICOMCONFIG command 151

ICOMSTATUS log 272  
 identifier  
     serial port 220, 239, 311  
 iIMU-FSAS  
     cables 113  
     commands 187  
     dimensions 110  
     FAQ 408  
     models 32  
     performance 119  
     replacement parts 407  
     scale factor 328  
     specifications 110  
 impedance 88  
 IMU-CPT  
     cable 122  
     dimensions 120  
     electrical 124  
     environmental 124  
     FAQ 408  
     models 32  
     sensor specifications 123  
     specifications 120  
 inertial measurement unit (IMU)  
     connection 106, 109, 113  
     status 321  
     type 186  
 inertial navigation system (INS) 154  
 input strobe 75  
 input/output (I/O) 88  
 INPUTGIMBALANCE command 152  
 INSATT log 276  
 INSATTS log 277  
 INSCOMMAND command 154  
 INSCOV log 278  
 INSCOVs log 280  
 INSPOS log 281  
 INSPOSS log 282  
 INSPOSSYNC log 283  
 INSPVA log 284  
 INSPVAS log 285  
 INSSPD log 286  
 INSSPDS log 287  
 INSUPDATE log 288  
 INSVEL log 290  
 INSVELS log 291  
 INSZUPT command 155  
 INSZUPTCONTROL 156

interface  
     cable 34, 36–38, 113, 115–116  
     command 148, 210  
     FSAS status 321  
     graphical 49  
     gyro time-out 321  
     log and command 54  
     port specific 150  
     replacement parts 406  
 Interface card  
     SPAN-MPPC 390  
     SPAN-SE 384, 390  
 Internet Protocol (IP) 73, 76–77, 153  
 interrupt 311  
 introduction 29  
 IPCONFIG command 153

## J

J model 346

## K

kinematic alignment 56, 63

## L

L model 346  
 latency 229  
 latitude/longitude 232  
     fix data 256  
     position 234  
 L-band 132, 166, 293, 295  
 LBANDINFO log 293  
 LBANDSTAT log 295  
 LED 67–68  
 lever arm 157, 237  
 LEVERARMCALIBRATE command 157  
 link 310  
 link, loss of 229  
 Litef LCI-1  
     connector 407  
     enclosure frame 45, 181  
     equipment returns 18  
     external cable harness 406  
     IMU status 322  
     IMU type 187  
     models 32  
     power requirements 40  
 LN-200

- cable 109
- commands 187
- dimensions 108
- FAQ 408
- models 32
- performance 109
- replacement parts 406
- scale factors 328
- sensor installation 364–366, 370, 377, 380–381, 383
- SPAN IMU enclosure 377
- universal enclosure 366
- local level coordinate frame 44
- locktime
  - current 315
  - L-band 299
- log
  - list 301
  - trigger 214
  - type 214
- LOG command 159
- LOGFILE command 163
- logging 69, 125, 213
- LOGLIST log 301

## M

- MAC log 304
- manual alignment 57
- mark input pulse 307
- Mark1 trigger 193
- MARKxCOUNT log 305
- MARKxPVA log 306
- MARKxTIME log 307
- mask, priority 336
- mean sea level 232
  - fix 256
  - position 234
- memory
  - buffer space 159
  - non-volatile 146
    - save configuration 175
- message
  - ascii 215
  - format 217
  - ID 302–303
  - time stamp 223
  - trigger 161–162
- MKDIR command 127

- mode
  - 2-D 261
  - 3-D 261
  - dynamic 171
  - operating 260
  - RTK 229
  - static 171
- model 31
  - authorization 199–201
  - card 201
  - clock 308
  - expiry date 339
  - switch 201
  - upgrades 22
  - version 339
- month 342, 347
- motion detector 171
- moving alignment 56
- MPPC. See SPAN-MPPC

## N

- navigation
  - data 260
  - mode 57
- NMEA 164
  - log list 225
  - position 258
  - pseudorange measurement noise
    - statistics 262
  - satellite range residuals 259
- NMEATALKER command 164
- node 254
- noise
  - statistic 262
  - thermal 312
- non-volatile memory (NVM) 146
  - save configuration 175
- north 290

## O

- offset 184, 192
  - clock 308
  - receiver clock 342
- OmniSTAR 150, 293
- operating mode 260
- orientation 181
- output 74, 88
- overload 159

**P**

parity 136–137  
     flag 312  
     port 311  
     removed 318  
 PASHR log 309  
 PC power settings 55  
 PDOP, *see* dilution of precision  
 period 159, 161–162, 302–303  
 phase lock loop (PLL) 313, 332  
 pitch 181, 184, 237, 276  
 PLL, *see* phase lock loop  
 polarity 74–75  
 polled log 214  
 port 40  
     ascii header 215  
     communication 218  
     configuration 135, 137, 148, 175, 238  
     identifier 136, 150  
     interface 150  
     interrupt 311  
     log request 160  
     output 160, 162, 302–303  
     power 140  
     RS232 138  
     serial 150, 310  
     status 310, 337  
     unlog 206  
 PORTSTATS log 310  
 position 192, 284  
     INS 154  
     mark 306  
     measurements 278, 281  
     synchronized 283  
 power 40, 140, 198  
 prerequisites 28  
 processing 312  
 processor 215, 219  
 pseudorange  
     error estimate 312  
     jump 312  
     measurement 262, 315–316  
     noise statistic 262  
     solutions 229  
 PSRDIFFSOURCE command 166  
 pulse 307  
 PWD command 128

**Q**

quality, NMEA 262  
 quotation mark 215

**R**

R model 346  
 range  
     bias 222  
     compressed 317  
     measurement 312  
     residual 259  
 RANGE log 312  
 range residual 259  
 RANGECMP log 316–317  
 raw almanac 253  
 RAWEPHEM log 318  
 RAWIMU log 319  
 RAWIMUS log 327  
 reacquisition 313  
 real-time kinematic (RTK) 229  
 receiver  
     character 311  
     status 159, 331, 334  
 reference frames 44  
 replacement parts 406–407  
 reset 68  
     complete 170  
     hardware 146, 169  
 RESET command 169  
 residual 259  
 resolution 170  
 RMDIR command 127  
 roll 181, 184, 237, 276  
 root mean square (RMS) 263  
 rotation 131, 174  
 RTCA 150  
     DGPS type 167  
 RTCM 150  
     DGPS type 167  
 RTCMV3 150  
     DGPS type 167  
 RTK filter 170  
 RTKCOMMAND command 170  
 RTKDYNAMICS command 171  
 RTKSOURCE command 172  
 RVB, *see* vehicle to body rotation  
 RVBCALIBRATE command 174

RXCONFIG log 329  
RXSTATUS log 331  
RXSTATUSEVENT log 337

## S

S model 346  
satellite  
    active 260  
    in view 264  
    number of 317  
    range 259  
    raw 318  
    tracking 312  
SAVECONFIG command 58, 126, 175, 181  
SBAS  
    control 176  
    differential 166  
    system type 176  
SBASCONTROL command 176  
scale factor 316  
scope 27  
SD Card data 69  
secure digital (SD Card) 69  
serial port 150, 239, 311  
set up hardware 33  
SETAUTOLOGGING command 178  
SETGIMBALORIENTATION command 179  
SETHEAVEWINDOW command 180  
SETIMUORIENTATION command 179, 181  
SETIMUTOANTOFFSET command 184  
SETIMUTOANTOFFSET2 command 185  
SETIMUTYPE command 186  
SETINITATTITUDE command 188  
SETINITAZIMUTH command 190  
SETINSOFFSET command 192  
SETMARKxOFFSET command 193  
SETUPSENSOR command 194  
SETWHEELPARAMETERS command 195  
short binary header 213, 221  
signal timing 138  
SOFTLOADFILE command 197  
SOFTPOWER command 198  
SPAN frame 181  
SPAN HG enclosure

    HG1700 IMU installation 360  
    SPAN IMU enclosure  
        LN-200 IMU 377  
SPANAUTH command 199  
SPANMODEL command 201  
SPAN-MPPC 390  
    connector pin-outs 394  
    firmware version 2  
    mechanical drawing 391  
SPAN-SE-D 78  
    configuring ALIGN 79  
SPANVALIDMODELS log 339  
speed 236, 286  
    over ground 267  
static mode 171  
status 229, 235  
    channel tracking 314–316  
    clock model 308  
    COM port 310  
    data 257  
    event 337  
    flag 331  
    indicators 67  
    receiver 159, 216, 331, 334  
    time 216  
    word 338  
steer, time 222  
strobe 74  
subframe 318  
support 21  
synchronize 74  
synchronize INS with GPS 283  
synchronous log 214

## T

talker, NMEA 164  
technical specifications 88, 102–103, 106, 109, 119, 124  
tests (built-in) 68  
time  
    coarse/fine 222  
    CPU 159  
    ephemeris 318  
    fine 222  
    GPS 342  
    interval 161–162  
    of mark in event 308  
    of position fix 257



- precision 222
- stamp 223
- status 216, 221–222
- steering 222
- synchronized 283
- UTC 256, 268
- validity 221
- TIME log 341
- TIMEDWHEELDATA log 343
- track over ground 236
- tracking
  - channel 312
  - continuous 299, 315
  - loop 312
- Transmission Control Protocol (TCP) 76–77
- transmit 137, 311
- trigger 142–143
  - error 331
  - log 159, 214, 302–303
  - option 160
- troubleshooting 408
- true north 236
  - pseudorange error orientation 263

## U

- undulation 232
  - best position 234
- universal enclosure
  - HG1700 349
- Universal IMU Enclosure
  - LN-200 IMU 366
  - specifications 97
  - technical specifications 120
- UNLOG command 204
- UNLOGALL command 206
- up 290
- User Datagram Protocol (UDP) 76–77
- using a command as a log 125

## V

- validity
  - clock model 307
  - receiver model 339
- variable lever arm 85
- VARIABLELEVERARM log 344
- VBS, OmniSTAR
  - HP/XP 299

- initiate 133
- subscription 294
- VDOP, *see* dilution of precision
- vector values 290
- vehicle frame 46
- vehicle to body rotation (RVB) 131, 174
- vehicle to SPAN frame angular offset calibration 63
- VEHICLEBODYROTATION log 345
- velocity 192, 284
  - computation 290
  - INS 154, 278
  - limit 231
  - mark 306
- version 199, 346
  - receiver model 346
- VERSION log 346

## W

- warranty 18
- Waypoint Products Group 249
- Web site 21
- week reference 318
- wheel sensor
  - data 209, 343
  - odometer 116
- WHEELSIZE log 348
- WHEELVELOCITY command 209
- word
  - raw ephemeris 318
  - status 338
  - week number 254

## Y

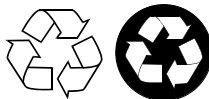
- year 342, 347

## Z

- Zero Velocity Update (ZUPT) 155



Recyclable



Printed in Canada on recycled paper