

## Trimble MB-Two OEM GNSS Board

Revision F  
December 2019



## Legal Notices

© 2016-2019, Trimble Inc. All rights reserved.

Trimble and the Globe & Triangle logo are trademarks of Trimble Inc., registered in the United States and in other countries. Ashtech, Z-Blade, CMR+, EVEREST, Maxwell, and Zephyr are trademarks of Trimble Inc.

Microsoft, Internet Explorer, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

Support for Galileo is developed under a license of the European Union and the European Space Agency.

Website: <https://www.trimble.com/Precision-GNSS/>

## Release Notice

This is the December 2019 release of the Trimble MB-Two Reference Manual.

## LIMITED WARRANTY TERMS AND CONDITIONS

### Product Limited Warranty

Subject to the following terms and conditions, Trimble Inc. ("Trimble") warrants that for a period of one (1) year from date of purchase unless otherwise specified, this Trimble product (the "Product") will substantially conform to Trimble's publicly available specifications for the Product and that the hardware and any storage media components of the Product will be substantially free from defects in materials and workmanship.

### Product Software

Product software, whether built into hardware circuitry as firmware, provided as a standalone computer software product, embedded in flash memory, or stored on magnetic or other media, is licensed solely for use with or as an integral part of the Product and is not sold. If accompanied by a separate end user license agreement ("EULA"), use of any such software will be subject to the terms of such end user license agreement (including any differing limited warranty terms, exclusions, and limitations), which shall control over the terms and conditions set forth in this limited warranty.

### Software Fixes

During the limited warranty period you will be entitled to receive such Fixes to the Product software that Trimble releases and makes commercially available and for which it does not charge separately, subject to the procedures for delivery to purchasers of Trimble products generally. If you have purchased the Product from an authorized Trimble dealer rather than from Trimble directly, Trimble may, at its option, forward the software Fix to the Trimble dealer for final distribution to you. Minor Updates, Major Upgrades, new products, or substantially new software releases, as identified by Trimble, are expressly excluded from this update process and limited warranty. Receipt of software Fixes or other enhancements shall not serve to extend the limited warranty period.

For purposes of this warranty the following definitions shall apply: (1) "Fix(es)" means an error correction or other update created to fix a previous software version that does not substantially conform to its Trimble specifications; (2) "Minor Update" occurs when enhancements are made to current features in a software program; and (3) "Major Upgrade" occurs when significant new features are added to software, or when a new product containing new features replaces the further development of a current product line.

Trimble reserves the right to determine, in its sole discretion, what constitutes a Fix, Minor Update, or Major Upgrade.

### Warranty Remedies

If the Trimble Product fails during the warranty period for reasons covered by this limited warranty and you notify Trimble of such failure during the warranty period, Trimble will repair OR replace the nonconforming Product with new, equivalent to new, or reconditioned parts or Product, OR refund the Product purchase price paid by you, at Trimble's option, upon your return of the Product in accordance with Trimble's product return procedures then in effect.

### How to Obtain Warranty Service

To obtain warranty service for the Product, please contact your local Trimble authorized dealer. Alternatively, you may contact Trimble to request warranty service by e-mailing your request to [GNSSOFMSupport@trimble.com](mailto:GNSSOFMSupport@trimble.com). Please be prepared to provide:

- Your name, address, and telephone numbers
- Proof of purchase
- A copy of this Trimble warranty
- A description of the nonconforming Product including the model number
- An explanation of the problem

The customer service representative may need additional information from you depending on the nature of the problem.

### Warranty Exclusions or Disclaimer

This Product limited warranty shall only apply in the event and to the extent that (a) the Product is properly and correctly installed, configured, interfaced, maintained, stored, and operated in accordance with Trimble's applicable operator's manual and specifications, and; (b) the Product is not modified or misused.

This Product limited warranty shall not apply to, and Trimble shall not be responsible for, defects or performance problems resulting from (i) the combination or utilization of the Product with hardware or software products, information, data, systems, interfaces, or devices not made, supplied, or specified by Trimble; (ii) the operation of the Product under any specification other than, or in addition to, Trimble's standard specifications for its products; (iii) the unauthorized installation, modification, or use of the Product; (iv) damage caused by: accident, lightning or other electrical discharge, fresh or salt water immersion or spray (outside of Product specifications); or exposure to environmental conditions for which the Product is not intended; (v) normal wear and tear on consumable parts (e.g., batteries); or (vi) cosmetic damage.

Trimble does not warrant or guarantee the results obtained through the use of the Product, or that software components will operate error free.

**NOTICE REGARDING PRODUCTS EQUIPPED WITH TECHNOLOGY CAPABLE OF TRACKING SATELLITE SIGNALS FROM SATELLITE BASED AUGMENTATION SYSTEMS (SBAS) (WAAS/EGNOS, AND MSAS), OMNISTAR, GPS, MODERNIZED GPS OR GLONASS SATELLITES, OR FROM IALA BEACON SOURCES: TRIMBLE IS NOT RESPONSIBLE FOR THE OPERATION OR FAILURE OF OPERATION OF ANY SATELLITE BASED POSITIONING SYSTEM OR THE AVAILABILITY OF ANY SATELLITE BASED POSITIONING SIGNALS.**

THE FOREGOING LIMITED WARRANTY TERMS STATE TRIMBLE'S ENTIRE LIABILITY, AND YOUR EXCLUSIVE REMEDIES, RELATING TO THE TRIMBLE PRODUCT. EXCEPT AS OTHERWISE EXPRESSLY PROVIDED HEREIN, THE PRODUCT, AND ACCOMPANYING DOCUMENTATION AND MATERIALS ARE PROVIDED "AS-IS" AND WITHOUT EXPRESS OR IMPLIED WARRANTY OF ANY KIND, BY EITHER TRIMBLE OR ANYONE WHO HAS BEEN INVOLVED IN ITS CREATION, PRODUCTION, INSTALLATION, OR DISTRIBUTION, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. THE STATED EXPRESS WARRANTIES ARE IN LIEU OF ALL OBLIGATIONS OR LIABILITIES ON THE PART OF TRIMBLE ARISING OUT OF, OR IN CONNECTION WITH, ANY PRODUCT. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW LIMITATIONS ON DURATION OR THE EXCLUSION OF AN IMPLIED WARRANTY, THE ABOVE LIMITATION MAY NOT APPLY OR FULLY APPLY TO YOU.

**Limitation of Liability**

TRIMBLE'S ENTIRE LIABILITY UNDER ANY PROVISION HEREIN SHALL BE LIMITED TO THE AMOUNT PAID BY YOU FOR THE PRODUCT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL TRIMBLE OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGE WHATSOEVER UNDER ANY CIRCUMSTANCE OR LEGAL THEORY RELATING IN ANYWAY TO THE PRODUCTS, SOFTWARE AND ACCOMPANYING DOCUMENTATION AND MATERIALS, (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF DATA, OR ANY OTHER PECUNIARY LOSS), REGARDLESS OF WHETHER TRIMBLE HAS BEEN ADVISED OF THE POSSIBILITY OF ANY SUCH LOSS AND REGARDLESS OF THE COURSE OF DEALING WHICH DEVELOPS OR HAS DEVELOPED BETWEEN YOU AND TRIMBLE. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY OR FULLY APPLY TO YOU.

**PLEASE NOTE: THE ABOVE TRIMBLE LIMITED WARRANTY PROVISIONS WILL NOT APPLY TO PRODUCTS PURCHASED IN THOSE JURISDICTIONS (E.G., MEMBER STATES OF THE EUROPEAN ECONOMIC AREA) IN WHICH PRODUCT WARRANTIES ARE THE RESPONSIBILITY OF THE LOCAL TRIMBLE AUTHORIZED DEALER FROM WHOM THE PRODUCTS ARE ACQUIRED. IN SUCH A CASE, PLEASE CONTACT YOUR LOCAL TRIMBLE AUTHORIZED DEALER FOR APPLICABLE WARRANTY INFORMATION.**

**USA**

Supplier's Declaration of Conformity  
We, Trimble,  
declare under sole responsibility that the product:  
MB-Two GNSS Board  
complies with Part 15 of FCC Rules.  
Operation is subject to the following two  
conditions:  
(1) this device may not cause harmful interference,  
(2) and this device must accept any interference received,  
including interference that may cause  
undesired operation.

Trimble Inc.  
10368 Westmoor Dr.  
Westminster, CO 80021

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment 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's by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into 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.

Any changes or modifications to this equipment not expressly approved by Trimble Inc. may cause, harmful interference and void the FCC authorization to operate this equipment.

**PLEASE NOTE: THE ABOVE TRIMBLE LIMITED WARRANTY PROVISIONS WILL NOT APPLY TO PRODUCTS PURCHASED IN THOSE JURISDICTIONS (E.G., MEMBER STATES OF THE EUROPEAN ECONOMIC AREA) IN WHICH PRODUCT WARRANTIES ARE THE RESPONSIBILITY OF THE LOCAL TRIMBLE AUTHORIZED DEALER FROM WHOM THE PRODUCTS ARE ACQUIRED. IN SUCH A CASE, PLEASE CONTACT YOUR LOCAL TRIMBLE AUTHORIZED DEALER FOR APPLICABLE WARRANTY INFORMATION.**

## Europe



Hereby, Trimble, declares that the MB-Two GNSS board is in compliance with the following

directives:

- RED 2014/53/EU
- RoHS Directive 2011/65/EU.

The products covered by this guide may be operated in all EU member countries (BE, BG, CZ, DK, DE, EE, IE, EL, ES, FR, HR, IT, CY, LV, LT, LU, HU, MT, NL, AT, PL, PT, RO, SI, SK, FI, SE, UK), Norway, and Switzerland.

### Official Language

THE OFFICIAL LANGUAGE OF THESE TERMS AND CONDITIONS IS ENGLISH. IN THE EVENT OF A CONFLICT BETWEEN ENGLISH AND OTHER LANGUAGE VERSIONS, THE ENGLISH LANGUAGE SHALL CONTROL.

### COCOM Limits

The U.S. Department of Commerce requires that all exportable GPS products contain performance limitations so that they cannot be used in a manner that could threaten the security of the United States. The following limitations are implemented on this product:

- Immediate access to satellite measurements and navigation results is disabled when the receiver velocity is computed to be greater than 1,000 knots, or its altitude is computed to be above 18,000 meters. The receiver GPS subsystem resets until the COCOM situation clears. As a result, all logging and stream configurations stop until the GPS subsystem is cleared.

### Restriction of Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS).

Trimble products in this guide comply in all material respects with DIRECTIVE 2002/95/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS Directive) and Amendment 2005/618/EC filed under C(2005) 3143, with exemptions for lead in solder pursuant to Paragraph 7 of the Annex to the RoHS Directive applied.



### Waste Electrical and Electronic Equipment (WEEE)

For product recycling instructions and more information, please go to:

<https://www.trimble.com/Corporate/RCC/Environment.aspx>

**Recycling in Europe:** To recycle Trimble WEEE (Waste Electrical and Electronic Equipment, products that run on electrical power.), Call +31 497 53 24 30, and ask for the "WEEE Associate". Or, mail a request for recycling instructions to:

Trimble Europe BV  
c/o Menlo Worldwide Logistics  
Meerheide 45  
5521 DZ Eersel, NL

## MB-Two Reference Manual Release Notes, December 2019

The content of this new MB-Two Reference Manual reflects the changes and enhancements made to the board compared to the July 2019 edition of this manual.

1. Chapter 5 about Web server and the corresponding on-line help files have been upgraded regarding access rights and their consequences.
2. Some \$PASHS commands updated: ATH,PWD - CST,ON - NTR,PAR - TCP,PAR - WEB,PAR.
3. Two \$PASHS commands removed: WEB,USR,ADD - WEB,USR,DEL.

NOTICE: This manual, and more specifically its appendix, provides general information on GNSS systems and frequencies, irrespective of the product to which the manual is dedicated. For example the L5 frequency is mentioned repeatedly in the appendix, but remember this frequency is not supported by the MB-Two. If you are in doubt about a particular feature and would like to know whether it is supported in the MB-Two or not, please refer to the Specifications section in Chapter 1.



# Table of Contents

<b>Chapter 1. Introduction</b> .....	<b>1</b>
Introduction .....	1
Operating Modes .....	3
MB-Two Top and Bottom Views .....	6
List of Items .....	6
Specifications .....	12
Board Layout and Dimensions .....	17
I/O Connector Pin Out .....	18
LED Indicator .....	19
Communication Ports .....	20
<b>Chapter 2. First Steps With the MB-Two</b> .....	<b>21</b>
Using the Evaluation Kit .....	21
Using AshCom to Communicate with MB-Two .....	23
Connecting MB-Two to your Application .....	27
Precautions Before Turning Off the Board .....	29
GNSS Antenna Setup for Heading/Attitude Measurements .....	29
<b>Chapter 3. Typical Applications Explained</b> .....	<b>35</b>
Preamble .....	35
Rover Computing RTK Position .....	37
Rover Computing Internal Heading + RTK Position .....	38
Rover Computing Hot Standby RTK Position .....	41
Rover Computing Internal Vector .....	43
Rover Computing Attitude .....	45
Rover Computing External Vector (Relative Positioning) .....	47
Rover Computing Internal Heading and External Vector .....	50
Rover Computing External Heading .....	52
Rover Computing Two Independent RTK Positions (Dual RTK) .....	54
Rover Delivering Two External Vectors (Dual Relative RTK) .....	56
Rover Computing RTX Position .....	58
Base Configurations .....	59
Raw Data Recording .....	62
Adding Startup Protection to the Board .....	63
<b>Chapter 4. Trimble RTX Correction Service</b> .....	<b>67</b>
Introduction .....	67
Computing RTX Position in MB-Two .....	67
Implementing Trimble RTX using the MB-Two Web Server .....	68
Trimble RTX Accuracy Figures .....	70
<b>Chapter 5. Setting the MB-Two from the Web Server</b> .....	<b>71</b>
Running the Embedded Web Server .....	71
Managing Access Rights .....	75
Home Page .....	76
Receiver Page .....	79

<b>Chapter 6. Configuring the MB-Two Using Serial Commands.....</b>	<b>125</b>
Introduction to Serial Commands.....	125
Overview of the MB-Two Proprietary Commands .....	128
<b>Chapter 7. Set Command Library.....</b>	<b>139</b>
3DF: Running Attitude Processing on Set Baselines.....	139
3DF,ANG: Maximum Baseline Elevation .....	141
3DF,CLB: Running/Resetting Baseline Auto-Calibration .....	141
3DF,MXB: Setting Baseline Length Tolerance .....	142
3DF,OFS: Defining Heading-Pitch-Roll Offset Values .....	143
3DF,RST: Resetting Attitude Computation and Calibration .....	145
3DF,Vx: Entering Vector Components .....	146
ALR,ACK: Acknowledging Alarms.....	147
ANH: Antenna Height .....	147
ANP,OUT: Defining a Virtual Antenna.....	148
ANP,OWN: Naming Local Antennas.....	150
ANP,REF: Naming the Antenna Used at the Base.....	151
ANR: Antenna Reduction Mode .....	152
ANT: Antenna Height.....	154
ATH,PWD: Defining the Startup Password.....	156
ATL: Debug Messages.....	156
ATM: Enabling/Disabling ATOM Messages .....	158
ATM,ALL: Disabling All ATOM Messages.....	161
BDS: BeiDou Tracking .....	161
BLN: Defining Baselines and their Base/Rover Data Sources.....	163
BRD: Enabling/Disabling the RTK Bridge Function .....	165
BRV: Relative Processing .....	166
BRV,RST: Resetting Relative Positioning Mode .....	167
CAN,OFF: Disabling the CAN Port.....	168
CAN,ON: Enabling the CAN Port .....	168
CAN,PAR: Setting the CAN Parameters.....	169
CMD,LOD: Running a List of \$PASH Commands .....	170
CMD,WTI: Inserting Wait Times .....	171
CMR: Enabling/Disabling CMR or CMR+ Messages .....	172
CMR,ALL: Disabling All CMR or CMR+ Messages .....	173
COO,REF: Entering a Reference Position Manually in a Rover .....	174
CST,ON: Starting the Embedded NTRIP Caster .....	175
CST,OFF: Stopping the Embedded NTRIP Caster.....	176
CST,MTP,ADD: Adding/Modifying Mount Points .....	176
CST,MTP,DEL: Deleting a Mount Point.....	178
CST,PAR: Embedded NTRIP Caster Parameters .....	178
CST,RST: Resetting the Embedded NTRIP Caster.....	180
CST,USR,ADD: Adding/Modifying NTRIP Caster Users .....	181
CST,USR,DEL: Deleting an NTRIP Caster User.....	182
DFC,DEL: Deleting Default Configuration File.....	182
DFC,GET: Duplicating the Default Configuration File .....	183
DFC,SET: Setting the Default Configuration File.....	184

DFC,TST: Testing the Default Configuration File.....	185
DDN,PAR: Setting the DynDNS Service .....	185
DDN,SET: Sending the IP Address Manually to DynDNS .....	187
DIF,BDS: Enabling or Disabling BeiDou Correcting Data.....	187
DIF,NET: Specifying the Processing Mode for a Network Rover ...	188
DIF,SBA: Specifying Which SBAS Corrections to Use.....	189
DIP,OFF: Terminating Direct IP Connection .....	190
DIP,ON: Establishing the Programmed Direct IP Connection.....	191
DIP,PAR: Setting Direct IP Parameters.....	191
DRD: Data Recording Duration .....	193
DRI: Raw Data Recording Rate .....	194
DSY: Daisy Chain .....	194
DYN: Receiver Dynamics.....	195
ELM: Setting the Elevation Mask for Raw Data Output .....	196
ENC: Setting Transport Mode for Differential Data.....	197
ETD,PAR: Setting Ethernet Driver Parameters .....	198
ETH,OFF: Powering Off the Ethernet Port.....	199
ETH,ON: Powering On the Ethernet Port.....	199
ETH,PAR: Ethernet Parameters.....	200
FIL,CPY: Copying Files from Internal to External Memory .....	201
FIL,DEL: Deleting Files and Directories .....	202
FIL,GET: Downloading a File Thru the Current Port.....	203
FIL,WRT: Writing a File.....	205
FMT: Formatting Internal Memory or SD Card .....	207
FTP,OFF: Ending Data Transfer with FTP.....	207
FTP,PAR: FTP Settings.....	208
FTP,PUT: Uploading Files to FTP.....	209
GAL: Galileo Tracking .....	211
GEM: Choosing a Geoid Model.....	212
GLB: Choice of Antenna Input for L-Band Reception .....	212
GLO: GLONASS Tracking .....	213
GLx: Tracking Signals in Different Bands.....	214
<GNS>,USE: Enabling or Disabling the Tracking of a GNSS Satellite .....	215
GPS: GPS Tracking.....	217
GSF: Enabling/Disabling GSOF Messages .....	217
GSF,ALL: Disabling all GSOF Messages on a Port.....	218
INI: Resetting Receiver According to Your Preference .....	219
LBN,BEM: Additional Settings for User-Added L-Band Satellite..	220
LBN,SAT: Adding a New L-Band Satellite.....	222
LBN,USE: Tracking L-Band Providers/Geo-satellites .....	223
LCK,MOD: Controlling Receiver Lock.....	224
LCK,OFF: Unlocking the Receiver .....	225
LCK,ON: Locking the Receiver.....	226
LCS: Enabling/Disabling Use of Local Coordinate System .....	227
LOG,DEL: Deleting Log Files .....	228
LOG,PAR: Log File Settings.....	228

MDP: Choosing Communication Mode on Port D.....	229
MEM: Selecting Memory Device Used .....	230
MET,OWN - MET,REF: Entering Meteorological Parameters.....	230
MSG: Defining a User Message.....	232
NME: Enabling/Disabling NMEA Messages.....	233
NME,ALL: Disabling All NMEA and NMEA-Like Messages .....	235
NME,ALR: Setting an Alarm Message.....	236
NME,TTT: Requesting Output of Event Marker .....	237
NPT: Defining How RTX and SBAS Positions are Tagged in POS Messages.....	238
NTP: Controlling the Network Time Protocol Server.....	239
NTR,LOD: Loading the NTRIP Caster Source Table.....	240
NTR,MTP: Connecting Receiver to NTRIP Caster Mount Point ....	241
NTR,PAR: NTRIP Settings.....	242
OBS: Setting Preference for GNSS Signals Tracking .....	243
OCC: Writing Occupation Data to Raw Data File .....	245
OPTION: Receiver Firmware Options .....	246
OUT: Suspending/Resuming Message Output.....	249
OUT,ALL: Disabling All Periodic Messages.....	250
OUT,DIF,OFF: Disabling All Differential Messages .....	251
PAR,LOD: Configuring the Receiver From a PAR File .....	252
PAR,SAV: Saving the Receiver Configuration To a PAR File .....	253
PEM: Setting the Position Elevation Mask.....	254
PGN: Enabling/Disabling CAN-NMEA 2000 Messages.....	254
PGN,ALL: Disabling all CAN-NMEA 2000 Messages .....	255
PGS: Defining the Primary GNSS System.....	256
PHE: Setting the Active Edge of the Event Marker Pulse .....	259
PIN: Assigning Function to Programmable Pin on I/O Connector ..	260
POP: Setting Internal Update Rates for Measurements and PVT..	264
POS: Setting the Antenna Position .....	265
POS,CUR: Making the Current Position the Reference Position (Static) .....	267
POS,MOV: Making the Current Position the Reference Position (Dynamic) .....	269
PPP: Selecting a PPP Service.....	270
PPP,RST: Resetting the Selected PPP Service .....	270
PPP,SRC: Defining Channel Delivering Corrections to the Selected PPP Service .....	271
PPS: Setting PPS Pulse Properties.....	271
PRT: Setting Baud Rates.....	272
PWR,OFF: Powering Off the Board .....	273
QZS: Enabling/Disabling QZSS Tracking .....	274
RCP,OWN: Naming Local Receiver.....	274
RCP,REF: Naming Reference Receiver .....	275
RDP,PAR: Setting the Radio.....	276
RDP,TYP: Defining the Type of Radio Used.....	281
REC: Enable/Disable, Start/Stop Raw Data Recording .....	282

RFM: Enabling/Disabling Ring File Memory .....	283
RST: Default Settings .....	284
RT2: Enabling/Disabling RTCM 2.3 Messages .....	285
RT2,ALL: Disabling All RTCM 2.3 Messages .....	286
RT3: Enabling/Disabling RTCM 3.1 Messages .....	287
RT3,ALL: Disabling All RTCM 3.2 Messages .....	289
RTK: Running RTK Processing on Set Baselines.....	289
RTK,RST: Resetting RTK or RTX Computation .....	290
RTK,STI: Defining Which Base Data Stream to Use .....	291
RTX,DTM: RTX Datum Transformation.....	292
RTX,KPI: Entering Known RTX Initialization Point .....	295
RTX,MOD: Specifying the RTX Corrections Service Used .....	297
RTX,RST: Resetting RTX Position Computation .....	298
RTX,SRC: Specifying RTX Corrections Source .....	299
RXC,PAR: Setting the Embedded RINEX Converter .....	301
RXC,RUN: Converting a G-File into RINEX Files .....	303
SBA: Enabling/Disabling SBAS Tracking.....	304
SIT: Defining a Site Name.....	305
SNS: Configuring the M-Sensor .....	305
STI: Defining a Station ID .....	308
TCP,PAR: TCP/IP Server Settings.....	309
TCP,UID: TCP/IP Authentication .....	310
TOP: Defining the Type of Output Position .....	311
UDP: User-Defined Dynamic Model Parameters.....	312
UPL,PAR: Setting the Firmware Upgrade Procedure .....	313
UPL,UPG: Upgrading the Receiver Firmware from FTP.....	314
VIP: Defining a Virtual Port .....	315
WEB,OWN: Setting Owner Information .....	316
WEB,PAR: Web Server Control & Administrator Profile .....	316
ZDA: Setting Date & Time .....	317
<b>Chapter 8. Query Command Library .....</b>	<b>319</b>
Response Format .....	319
ALR: Listing Current Alarms .....	320
ANH: Antenna Height .....	321
ANP: Antenna Parameters .....	322
ANP,OUT: Virtual Antenna .....	323
ANP,OWN: Local Antenna Used.....	323
ANP,RCV: Antenna Name and Offsets of Received Base.....	324
ANP,REF: Antenna Used at the Base .....	325
ANR: Antenna Reduction Mode .....	326
ANT: Antenna Height.....	326
ARA: Attitude Rate and Accuracy.....	327
ARR: Vector & Accuracy Data .....	329
AST: Status of Antennas .....	331
ATH: Listing Anti-Theft Parameters .....	332
ATL: Debug Data Recording .....	333
ATT: Heading, Roll and Pitch .....	335

AVR: Trimble Time-Yaw-Tilt-Range Message .....	336
BDS: BeiDou Tracking Status .....	338
BLN: Reading Baseline Definitions & Status .....	339
BRD: Reading RTK Bridge Status .....	341
BRV: Reading Relative Mode Status .....	341
CAN: CAN Bus Status & Parameters.....	342
COO,REF: Base Position .....	343
CST: NTRIP Caster Parameters .....	344
CST,USR: Listing NTRIP Caster Users.....	345
DDM: Differential Decoder Message .....	346
DDN: DynDNS Parameters .....	348
DDS: Differential Decoder Status .....	349
DFC: Reading the Content of the Default Configuration File.....	351
DIF,BDS: BeiDou Correcting Data .....	351
DIF,NET: Source of Correction Data Currently Used .....	352
DIF,SBA: Source of SBAS Corrections .....	353
DIP: Direct IP Parameters .....	354
DRD: Data Recording Duration.....	355
DRI: Raw Data Recording Rate .....	355
DSY: Daisy Chain Status .....	356
DTM: Datum Reference.....	357
DTH: Dithering Mode Status.....	358
DYN: Receiver Dynamics.....	359
ELM: Elevation Mask .....	360
ETD: Reading Ethernet Driver Parameters .....	360
ETH: Ethernet Status and Parameters .....	361
FIL,CUR: Information On G-File Being Recorded .....	362
FIL,LST: Listing Files in Receiver Memory or USB Key.....	363
FIL,STS: Memory Status .....	364
FTP: FTP Status and Settings.....	365
GAL: GALILEO Tracking Status.....	366
GBS: GNSS Satellite Fault Detection.....	367
GEM: Geoid Model .....	369
GGA: GNSS Position Message.....	370
GGKx: GNSS Position Message .....	372
GLB: Choice of Antenna Input for L-Band Reception.....	373
GLL: Geographic Position - Latitude/Longitude .....	374
GLO: GLONASS Tracking Status .....	375
GLx: Checking Which GNSS Signals are Tracked .....	376
GMP: GNSS Map Projection Fix Data .....	377
GNS: GNSS Fix Data .....	379
GPS: GPS Tracking Status.....	380
GRS: GNSS Range Residuals.....	381
GSA: GNSS DOP and Active Satellites.....	382
GST: GNSS Pseudo-Range Error Statistics .....	385
GSV: GNSS Satellites in View .....	386
HDT: True Heading.....	388

HPR: Extended Attitude Message, UTC Time-Tagged .....	389
LBN: L-Band Almanac and Status.....	391
LBN,BEM: User-Defined L-Band Satellites .....	393
LCK: Lock Mode Status & Receiver Lock Status .....	394
LCS: Local Coordinate System Status .....	394
LOG: Editing a Log File.....	395
LOG,LST: Listing Log Files.....	396
LOG,PAR: Log File Settings.....	397
LTN: Latency .....	398
MDP: Port Communication Mode .....	399
MEM: Selected Memory Device.....	400
NPT: How Position Type is Reported in POS Message .....	400
NTP: Network Time Protocol Server.....	401
NTR: NTRIP Settings.....	402
NTR,MTP: Connection to Mount Point.....	403
NTR,TBL: Source Table .....	404
OBS: Observations Usage.....	406
OCC: Occupation State and Parameters .....	406
OPTION: Installed Receiver Firmware Options.....	407
OPTION,EXP: Installed Firmware Options and Expiration Dates...	410
OUT: Current Status of Periodic Messages .....	411
OUT,GSF: Current Status of GSOF Messages .....	412
OUT,MSG: Output Messages .....	413
OUT,PGN: Current Status of CAN-NMEA 2000 Messages.....	415
PEM: Position Elevation Mask .....	416
PGS: Primary GNSS System .....	417
PHE: Active Edge of Event Marker Pulse.....	417
PIN: Programmable Pin .....	418
POP: Reading Internal Update Rate .....	419
POS: Computed Position Data .....	420
POS,REF: Base Position.....	422
PPP: Reading the Current PPP Settings.....	423
PPS: PPS Settings .....	424
PRT: Baud Rate Settings .....	425
PSP: Reading the Position of the Physical Reference Station Received .....	426
PTT: PPS Time Tag .....	427
QZS: QZSS Tracking Status.....	429
RCA: Reading the Received CMR Type 2 Attribute.....	430
RCP,OWN: Receiver Name .....	430
RCP,REF: Reference Receiver Name.....	431
RCS: Reading the Recording Status .....	432
RDP,CHT: Radio Channel Table .....	433
RDP,PAR: Radio Parameters .....	435
RDP,PWR: Reading Radio Type Used and Radiated Power .....	438
RDP,TYP: Radio Type Used .....	439
REC: Raw Data Recording Status.....	440

RFM: Ring File Memory .....	441
RID: Receiver Identification .....	442
RMC: Recommended Minimum Specific GNSS Data.....	443
ROT: Rate of Turn .....	445
RSP: Reading the Position of the Reference Station Received.....	446
RTK: Reading RTK Processing Status.....	447
RTX: Reading RTX Processing Status.....	448
SAT: Satellites Status (Obsolete Command) .....	450
SBA: SBAS Tracking Status.....	452
SBD: BeiDou Satellites Status .....	452
SGA: GALILEO Satellites Status.....	453
SGL: GLONASS Satellites Status .....	454
SGP: GPS Satellites Status .....	456
SIT: Site Name .....	457
SLB: L-Band Satellites Status .....	457
SNS: M-Sensor Setting .....	458
SQZ: QZSS Satellites Status.....	459
SSB: SBAS Satellites Status .....	461
STI: Station ID .....	462
TCP: TCP/IP Server Settings .....	463
TEM: Die Temperature.....	464
THS: True Heading and Status .....	465
TOP: Type of Position Delivered .....	466
TRS: Reading Information on TERIASat .....	466
TTT: Event Marker.....	467
UDP: User-Defined Dynamic Model .....	468
UPL: FTP Server Providing Firmware Upgrades.....	469
UPL,LOG: Editing the Firmware Upgrade Log File.....	470
UPL,LST: Listing the Firmware Upgrades Available on FTP.....	472
VCR: Vector & Accuracy Data.....	473
VCT: Vector & Accuracy Data .....	476
VEL: Velocity & Accuracy Message .....	478
VERSION: Firmware Version .....	479
VLP: Receiver Validity Period.....	480
VTG: Course Over Ground and Ground Speed .....	480
WEB: Web Server Control, Owner Data & Connection Profiles.....	481
ZDA: Time & Date .....	482
<b>Chapter 9. DCOL Commands &amp; GSOF Messages .....</b>	<b>485</b>
Introduction.....	485
06h: GETSERIAL .....	486
07h: RETSERIAL .....	486
4Ah: GETOPT .....	488
4Bh: RETOPT .....	489
54h: GETSVDATA.....	492
55h: RETSVDATA.....	494
64h: APPFILE.....	496



65h: GETAPPFIL	501
6Eh: BREAKRET	502
6Fh: BREAKREQ	504
GSOF Messages	505
<b>Chapter 10. Appendices</b>	<b>507</b>
MB Loader Software Utility for Upgrades	507
Base Antenna Issues	510
Rover Antenna Issues	512
“Virtual Antenna” Concept	513
Antenna Height Transformations	514
How a Rover Reacts Dynamically to a Change of Base ID or Antenna Name	515
Useful Definitions	516
RJ 45 Internet Connector Schematic Diagram	527
Transitioning from Ashtech DG14 or MB100 to Trimble MB-Two	527



# Chapter 1. Introduction

## Introduction

---



The Trimble MB-Two is the successor to the Trimble MB-One OEM receiver module.

The MB-Two maintains the identical form-factor as its predecessor and offers customers a drop-in replacement to utilize the latest GNSS design innovations.

Versatile, powerful, compact and smart; the Trimble MB-Two provides faster Dual Frequency based Heading acquisition and an improved positioning engine with additional GNSS signals.

In addition two MB-Two modules may be used for Precise Platform Positioning using data from three antennas for precise position and attitude.

- **Advanced Features with Z-Blade Technology:** The MB-Two allows a wide range of option-upgradable GNSS configurations, from single frequency (GPS, SBAS) to dual frequency GNSS (GPS, QZSS, GLONASS, BeiDou, Galileo).

Trimble's patented Z-Blade technology drives a powerful GNSS agnostic engine allowing the MB-Two to use any GNSS system for positioning, without any dependency on GPS.

The GNSS engine utilizes over-the-air satellite corrections via embedded L-Band hardware to achieve centimeter/decimeter level accuracy with Trimble RTX corrections. With worldwide availability of RTX corrections, the MB-Two delivers centimeter level positioning without a dedicated base station.

- **Dual-Antenna Input for Heading+Pitch or Roll:**
  - GNSS Heading + Pitch or Roll.
  - Fast Time to First Heading (TTFH).
  - Two antenna inputs (SOLO/DUO mode).
  - Single board for RTK/RTX and Heading.

- **Powerful RTK Engine:** The MB-Two has a powerful RTK engine that delivers centimeter-level accuracy for systems using corrections from a local base or RTK network. It also features RTK against a moving base for relative positioning.

The network RTK capabilities include third-party network corrections such as VRS, FKP, and MAC. The advanced RTK algorithms use all available satellites to compute Heading with baseline length auto calibration together with Pitch or Roll.

- **Next Generation Hardware Design:**
  - Low-power consumption in a compact size
  - Dual Core CPU for optimal performance
  - Web User Interface for ease of use
  - Effective GNSS RF design.

## Operating Modes

---

Referring to the terminology used in \$PASH proprietary commands, the MB-Two board can operate in one of the following two modes:

- SOL mode (with a single GNSS antenna; “SOL” for SOLO)
- DUO mode (with two GNSS antennas).

### SOL Mode

In SOL mode, the board consists of a single L1/(L2+L3) GNSS sensor, which serves either antenna input #1 or antenna input #2.

The SOL mode is the board’s default operating mode. It can also be selected after running the \$PASHS,SNS command as follows:

```
$PASHS,SNS,SOL[,d2]
```

Where:

- d2= 1 means the GNSS sensor always serves antenna input #1.
- d2= 2 means the GNSS sensor always serves antenna input #2.
- d2= 0 or is left empty means the GNSS sensor serves the detected antenna, or if two antennas are detected, it serves antenna input #1.



So when the board is operated in default mode, you don’t have to worry about which antenna input should be used. Just connect your GNSS antenna to any of the two inputs and your application will work in all cases.

In SOL mode, you are allowed to choose the GNSS signal tracking mode using the \$PASHS,OBS command. Possible choices are “ALL”, for all signals tracked, or “OPT” to let the board choose the best signals available.

In SOL mode, power consumption is less than in DUO mode.

In SOL mode, the board can deliver:

- Conventional, autonomous DGNSS/DSBAS positioning
- Conventional DGNSS/RTK base functionality
- Conventional RTK function against physical/virtual base or reference network
- Conventional PPP (Trimble RTX) function using L-band or IP reference data (only with antenna input #1)
- RTK against moving base
- Heading/attitude against external sensor data

- Different combinations of the above

## DUO Mode

The DUO mode can be used only if firmware option [D] or [E] has been installed in the board.

In DUO mode, the board consists of two GNSS sensors, one per antenna input. The GNSS sensor serving antenna input #1 is always L1/(L2+L3).

The DUO mode is NOT the board's default operating mode so it can only be implemented after running the \$PASHS,SNS command as follows:

```
$PASHS,SNS,DUO[,d2]
```

Where:

- d2= 1 means the GNSS sensor serving antenna input #2 is L1 only.
- d2= 2 means the GNSS sensor serving antenna input #2 is L1/(L2+L3).
- d2= 0 or is left empty is equivalent to setting d2 to "1".

Basically, more power is consumed in this mode than in SOL mode.

In DUO mode, only the GNSS sensor serving antenna input #1 is allowed to track L-band signals, provided d2= "0" or "1" or is empty. L-band tracking is not allowed for d2="2". Reminder: L-band signal tracking allows the board to provide Trimble RTX positioning using correcting data from a geo-satellite.

In DUO mode, the board will always operate using the best signals available, meaning you are NOT allowed to change the GNSS signal tracking mode using the \$PASHS,OBS command. In DUO mode, L-band signal tracking is unconditionally set to 'OPT'.

After the DUO mode has been selected, the Blade engine automatically starts computing the heading for the vector oriented from antenna #2 to antenna #1.

By default, when antenna input #2 is L1 only (d2= 1), first the heading process goes through a calibration stage to determine the baseline length. Then the process is constrained with the computed baseline length to improve the performance of the L1-only heading computation. This is called computing heading *in Fixed mode*.

By default, when antenna input #2 is L1/(L2+L3) (d2= 2), the heading is computed without the need to determine the baseline length through a prior calibration stage. This is called computing heading *in Flex mode*.

However, additional commands exist to:

- Switch from flex to fixed and vice versa (\$PASHS,3DF)
- Ask for baseline length calibration (\$PASHS,3DF, CLB)
- Enter the baseline length manually (\$PASHS,3DF,V12)

In DUO mode, for both antennas simultaneously, the board can deliver:

- Conventional, autonomous DGNSS/DSBAS positioning
- Conventional DGNSS/RTK base functionalities
- Conventional RTK function against physical/virtual base or reference network
- Conventional PPP (Trimble RTX) function using L-band or IP reference data (only with antenna input #1)
- RTK against moving base
- Heading for vector from antenna #2 to antenna #1
- Heading/attitude against external sensor data
- Different combinations of the above

## Selecting the Operating Mode From the Web Server

Go to **Receiver> Position> Sensors/Antennas Setup**. Make your choice of operating mode in the **Multiple-Sensor Mode** field. The available options reflect the explanations provided in the above two subsections, but using a slightly different terminology:

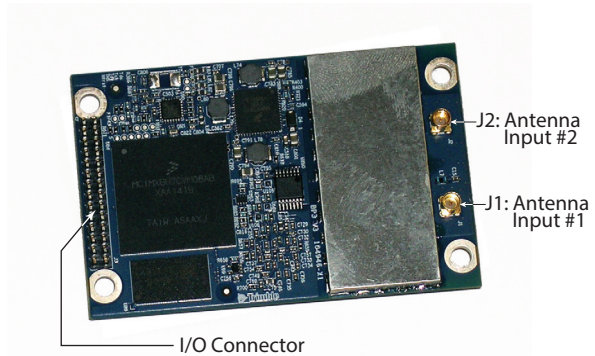
- **Single Antenna (automatic)**: SOL mode where the best antenna input is used (default).
- **Single Antenna (primary)**: SOL mode, antenna input # 1 (primary antenna) is used.
- **Single Antenna (secondary)**: SOL mode, antenna input # 2 (secondary antenna) is used.
- **Double Antenna (L1/L2/L-band+L1)**: DUO mode, GNSS sensor serving antenna input #2 is L1 only.
- **Double Antenna (L1/L2 + L1/L2)**: DUO mode, GNSS sensor serving antenna input #2 is L1/(L2+L3), just like GNSS sensor serving antenna input #1.

In DUO mode, to choose between “fixed” and “flex”, ask for baseline calibration or enter the baseline length, go to **Receiver> Position> Attitude/Heading Setup** and make your changes within the Vector Description pane.

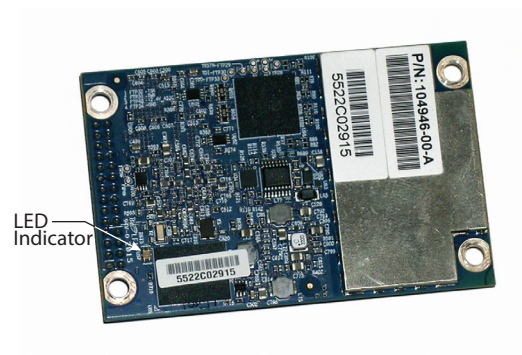
## MB-Two Top and Bottom Views

---

Top view:



Bottom View:



## List of Items

---

The tables below provide an overview of the different items that may be delivered with an MB-Two board. Depending on your purchase, only some of the listed items may have been shipped. Please refer to the packing list for an accurate description of the items that were ordered.

Trimble reserves the right to make changes to the list below without prior notice.





## Basic Supply

Item	Part No.	Pre-installed Firmware Options
MB-Two board, in one of the following configurations:		
• "Stock" product (hardware only)	106960-01	None
• Autonomous, SBAS & DGPS Positioning, GPS L1, 2 Hz, DGPS Rover*	106960-02	[2][N][X]**
• Raw Data Version with L1 GPS / GLONASS, 2 Hz, DGPS Rover*	106960-03	[2][N][G][X]**
• L1-only Heading Rover (GPS / GLONASS), DGPS Base/Rover*, 20 Hz	106960-04	[W][N][G][X][D]**
• RTK Rover L1/L2 GPS/GLONASS, 20 Hz	106960-05	[W][N][G][X][Y][J]**
• RTK Base Station Only (With Moving Base Capability), 2 Hz	106960-06	[2][N][G][X][Y][K]**
• Full Capability RTK Moving Base L1/L2 GPS / BeiDou, 2 Hz	106960-07	[2][N][B][X][Y][K]**
• L1 + L2 GPS / GLONASS Heading Rover, DGPS Base/Rover, 20 Hz	106960-08	[W][N][G][X][Y][D]**
• L1/L2 GPS + B1/B2 BeiDou Heading Rover, DGPS Base/Rover*, 20 Hz	106960-09	[W][N][B][X][Y][D]**
• RTK Rover L1/L2 GPS + B1/B2 BeiDou, 20 Hz	106960-10	[W][N][B][X][Y][J]**
• Full Capability RTK Base/Rover (Moving Base + Rover) with L1/L2 GPS/GLONASS + B1/B2 BeiDou, 20 Hz	106960-11	[W][N][G][B][X][Y][D][K][J]**
• Full Capability RTK Base/Rover with 3D Attitude Support Triple Frequency (GPS/GLONASS/BeiDou), 20Hz	106960-12	[W][N][G][B][X][Y][K][J][E]**
• RTK Rover L1/L2 GPS/GLONASS, 20Hz, 30/30 Dithering	106960-33	[W][N][G][X][Y][J][3]
• RTK Rover L1/L2 GPS/GLONASS, 20Hz, 10/10 Dithering	106960-77	[W][N][G][X][Y][J][10]
• RTK Rover L1/L2 GPS/GLONASS, 20Hz, 7/2 Dithering	106960-72	[W][N][G][X][Y][J][7]

\*: Limited DGNSS available.




\*\* : Also include the following options: SBAS, PPS, Event, Multipath Mitigation.

## Recommended GNSS Antennas


Item	Part No.	Picture
Trimble AV28 antenna, L1/L2/L5 GNSS + L Band aviation/marine/machine, Not TSO certified, 37 dB gain	112735	
Trimble AV33 antenna, L1 GNSS compact aviation/marine/machine, 43 dB gain	83553	

Item	Part No.	Picture
Trimble AV34 antenna, L1/L2 GNSS compact aviation/marine/machine, 43 dB gain	86362	
Trimble AV59 antenna, L1/L2/L5 GNSS + L-Band aviation/marine/machine, not TSO certified, 39 dB gain	C02992	
Trimble LV59 antenna, L1/L2/L5 GNSS + L-Band aviation/marine/machine, not TSO certified, 5/8" mount, 39 dB gain	C03167	
Zephyr Model 3, L1/L2/L5 Base	115000-50-INT	
Zephyr Model 3, L1/L2/L5 Rover	105000-50-INT	
L1/L2 Aviation Antenna (TSO certified)	C02817	
Trimble AG25 L1/L2/L5 L-Band Antenna	99038-00-INT	

## Antenna Accessories

Item Description	Part No.	Picture
5m TNC-TNC Antenna Cable	58957-05-INT	-
10m TNC-TNC Antenna Cable	A02500	-
30m TNC-TNC Antenna Cable	A02501	-
CBL ASSY TNC-MMCX	A02584	-
Zephyr mounting bracket	F00922	
AV33/34 antenna bracket	84902	
Mag mount with 5/8" x 11 bolt	86693	

## Evaluation Kit

Item	Part No.	Picture
Includes: <ul style="list-style-type: none"> <li>• Evaluation PCBA with two Antenna Cable Adaptors (MMCX-RA / TNC-F)</li> <li>• Power supply (19 V DC, 3.42 A)</li> <li>• Universal power supply cable</li> <li>• 5 spare jumpers</li> <li>• Utility Software (AshCom) downloadable from: <a href="ftp://ftp.ashtech.com/OEM_Sensor_ADU/">ftp://ftp.ashtech.com/OEM_Sensor_ADU/</a></li> </ul>	102926-00	

## Firmware Options

### Update Rates (computed data, raw data):

Option Code	Short Descr.	Part No.	Functionality
[5]	5 (5 Hz)	101530-13	Output rate up to 5 Hz (includes 1, 2 and 5 Hz)
[W]	W (20 Hz)	101530-10	Output rate up to 20 Hz (includes 1, 2, 5, 10, 20 Hz)
[8]	8 (50Hz)	101530-12	Output rate up to 50 Hz (includes 1, 2, 5, 10, 20 and 50 Hz)



**Installing one of these firmware options using \$PASHS,OPTION is not enough to make the new update rate effective for your messages.** You will then have to apply the new update rate to the board's internal processes using the \$PASHS,POP command (be aware this results in resetting the update rate of all your existing messages to their default values - see \$PASHS,POP command), and finally re-define the update rate of each of your existing messages.

### GNSS Tracking:

Option Code	Short Descr.	Part No.	Functionality
[N]	N (GPS)	101530-23	GPS+SBAS+QZSS constellations
[G]	G (GLN)	101530-20	GLONASS constellation
[B]	B (BDS)	101530-21	BeiDou constellation
[O]	O (GAL)	101530-22	Galileo constellation

### GNSS Frequency Bands:

Option Code	Short Descr.	Part No.	Functionality
[Y]	L2	101530-30	GPS/QZSS L2, GLONASS G2, GALILEO E5b, BEIDOU B2
[L]	L-Band	101530-70	L-band reception

**Board Configuration:**

Option Code	Short Descr.	Part No.	Functionality
[K]	K (RTK Base)	101530-40	Enables the output of all differential protocols (RTCM,CMR and ATOMI) for RTK operation. Being always enabled, standard DGPS differential messages (including those in standard or full ATOM protocol) are not tied to this option.
[F]	F (Flying RTK)	101530-41	Enables the board to deliver a "Flying RTK" position solution.
[J]	J (RTK Rover)	101530-42	Enables the board to deliver a "Fixed RTK" position solution (so this option includes the [F] option), an RTK position solution computed from a static/moving base.
[D]	D (Duo)	101530-51	Dual -sensor mode: Enables the board to deliver heading between the two antennas connected to the board. Also enables the board to deliver an independent RTK position solution for each antenna connected to the board (in which case the [J] option is also required). The board can also generate raw data from each of the two sensors. NOTE: With only the [D] option installed, the board cannot make heading measurements through an association with an external receiver.
[E]	A (Attitude)	101530-50	Enables the board to deliver attitude measurements provided the appropriate data are applied to the board. Option [E] includes option [D]. If you have purchased option [E], you don't need to purchase option [D]. Also allows the board to compute an external heading if another board is used as a moving base, providing the required corrections for computing heading.

**Other Options:**

Option Code	Short Descr.	Part No.	Functionality
[I]	I (RAIM)	101530-60	Enables output of RAIM-related messages (Same as MBxxx series)
[R]		101530-61	Data recording
		101530-25	MB-Two Field Configuration Upgrade
		97768-EW-1	1-year extended warranty on MB-Two
		97768-EW-2	2-year extended warranty on MB-Two

NOTE: An MB-Two board that returns “*This is a STOCK product, to enable GNSS tracking and data you must purchase activation code*” when queried about its firmware options currently installed (using \$PASHQ,RID) is a “stock” product.

That means the board was purchased just for its hardware and has no firmware options installed yet (the board is unable to track any GNSS signals and can only respond to \$PASHQ,RID).

The board is however ready for adding the new firmware options you will purchase, using the MB Loader software, the Web Server or command \$PASHS,OPTION.

## Specifications

---

### GNSS Engine

- 240 tracking channels.
- Two tightly coupled, all-in-view GNSS sensors delivering simultaneously:
  - GPS L1+L2
  - QZSS L1+L2
  - BeiDou B1+B2 (Phase III satellites supported)
  - GLONASS G1+G2 FDMA <sup>(1)</sup>
  - GALILEO E1+E5b
  - SBAS L1 <sup>(2)</sup>
- 2 x MSS L-Band Tracking Channels

### Features

- Ashtech patented Strobe Correlator™ to reduce GNSS multi path
- Ashtech patented Z-tracking to track encrypted GPS P(Y) signal

- Ashtech patented Z-Blade technology to process multi-GNSS data
- GPS-only, GLONASS-only, BeiDou-only or Galileo-only solutions possible (from Autonomous to full RTK)
- Fast Search Engine to improve TTFF
- Position in local datums and projections with RTCM-3 transformation data
- Trimble RTX™ PPP engine
- Ashtech Hot Standby RTK Algorithms
- Ashtech Flying RTK Algorithms
- Full attitude engine with two MB-Two boards connected
- RTK with Static & Moving Base corrections supported
- Heading engine with optional baseline length self-calibration
- Multi-dynamic mode (static/moving Base and Rover functions simultaneously)
- Adaptive velocity filter to meet specific dynamic applications
- Up to 250 MB of internal memory for data logging; on-board memory for various applications
- Up to 50 Hz position/velocity/heading/observables output <sup>(3)</sup>
- Reference Inputs/Outputs: RTCM 3.3 <sup>(4)</sup>, RTCM2.3, CMR <sup>(5)</sup>, CMRx <sup>(5)</sup>, ATOM <sup>(6)</sup>.
  - CMR (including CMR+) is an open format. The board supports CMR both in rover and base modes (GPS+GLONASS).
  - CMRx is an encrypted format. The board supports both the rover mode and base mode (all constellations). CMRx data part of Hot Standby RTK mode.
  - sCMR in rover mode only
- RTK Networks Supported: VRS, FKP, MAC
- Navigation Outputs: NMEA-0183, GSOF, NMEA 2000, ATOM format
- One-push Ashtech Trouble Log (ATL)
- Programmable startup protection
- Embedded RINEX converter (supports dual-antenna mode)

## GNSS Sensor Performance

- Position accuracy (HRMS), SBAS: 0.50 m <sup>(7)</sup>
- Update rate: Up to 50 Hz
- Latency: < 10 ms <sup>(8)</sup>
- Cold start: <60 seconds  
(No ephemeris or almanac data, no last computed position, reception channels reset and processor restarted.)
- Warm start: <45 seconds  
(Last computed position known, but no ephemeris data, reception channels reset and processor restarted.)
- Hot start: <11 seconds  
(Valid almanac information at board startup)
- Signal re-acquisition: <2 seconds  
(Following temporary but complete GNSS signal loss.)
- Maximum operating limits:
  - Velocity: 515 m/s
  - Altitude: 18,000 m

## Precise Positioning Performance

### RTK

<sup>(9)</sup> <sup>(10)</sup> <sup>(11)</sup>

L1 only (fixed ambiguity):

- Accuracy (HRMS): < 12 mm + 1.5 ppm
- Initialization time: < 10 min typical
- Operating range: < 10 km

L1/L2 (fixed ambiguity):

- Accuracy (HRMS): 8 mm + 1 ppm
- Initialization time: < 1 min typical
- Operating range: > 40 km

### RTX

<sup>(12)</sup> <sup>(13)</sup>

#### CenterPoint:

- Accuracy (RMS): < 4 cm
- Initialization time: < 30 min. typical
- Operating range (inland): Almost unlimited

#### RangePoint:

- Accuracy (RMS): < 50 cm
- Initialization time: < 5 min



- Operating range (inland): Almost unlimited

## HEADING

(10) (14) (15)

- Accuracy (RMS): Depends on the baseline length used. See table below.

Baseline length (m)	Accuracy (°)
1	0.2
2	0.1
3	0.067
5	0.04
10	0.02

- Initialization time: < 10 s typical
- Baseline length: <100 m

## Physical Characteristics

- Size (W x H x D): 71 x 46 x 11 mm
- Power Voltage: 3.2 to 4.5 V DC
- Power Consumption <sup>(16)</sup>: < 1.2 W
- Weight: 24 g (0.85 oz)
- Connectors:
  - I/O: 28-pin dual-row male header
  - Antennas: 2 x MMCX female connectors
- Antenna LNA Power Input:
  - Input voltage range: 4.0 to 12.0 V DC on I/O connector pin 5 <sup>(17)</sup>. This input voltage will be used if greater than the main power input voltage. If not, the main power input voltage will be used to power the antenna LNA as well.
  - Maximum current 150 mA
  - Minimum current 5 mA
  - LNA gain range (minus signal loss): 17 to 37 dB

## Environmental Characteristics

(18)

- Operating temperature: -40° to +85°C (-40° to +185°F)
- Storage temperature: -40° to +85°C (-40° to +185°F)
- Vibration: MIL-STD-810F, Fig 514.5C-17, Random 6.2g RMS operating, Random 8g RMS survival
- Mechanical Shock: MIL-STD 810F, Fig 516.5-10 (40g, 11 ms, sawtooth)
- Operating Humidity: 95%, non-condensing
- Maximum Acceleration: 11g

## I/O Interface

- SAMTEC 28-pin I/O connector (TMM-114-03-G-D) with backward compatibility for current industry standards
- 3 x LVTTTL serial ports (UART type) at up to 921,600bps
- USB 2.0 OTG port allowing up to 12 Mbps (USB/Serial Link, USB Memory Stick, On-board Memory Access)
- CAN bus interface
- 1 PPS Out / Event In (accuracy better than 20 ns)
- 1 LAN Ethernet port:
  - Supports links to 10BaseT/100BaseT networks
  - All functions are performed through a single IP address simultaneously – including web GUI access and raw data streaming
  - Network Protocols supported:
    - > HTTP (web GUI)
    - > NTripCaster, NTripServer, NTripClient
    - > Dynamic DNS
    - > NTP Server
    - > TCP/IP or UDP/IP

## Recommended Antennas

- Compact GNSS Machine/Marine/Aviation Antennas: Trimble AV33 & AV 34
- GNSS Machine/Marine/Aviation Antennas: Trimble AV59 & LV59

## Configuration Tool

AshCom (Ashtech Communicator) is a GNSS utility for boards and sensors evaluation and configuration:

- Preset command scripts
- Real-time data logging
- Real-time data visualization

(1) Hardware ready for G1 and G2 CDMA. This is based on the assumption that these new signals will be transmitted within natural GLONASS L1, L2 or within GPS L1/L2 frequency bands.

(2) In some modes, SBAS L1 is available only for single sensor.

(3) At 50 Hz, a limited set of messages can be generated simultaneously through a single port.

(4) RTCM 3.3 and lower versions (down to 3.0). In this manual, you will see a lot of mentions of RTCM v3.1 and 3.2 messages specifically, but remember these can all be regarded as RTCM3.3 messages. RTCM-3.2 and 3.3 Multiple Signal Messaging (MSM) guarantees compatibility with 3rd party for each GNSS data.

(5) A Trimble proprietary format.

(6) ATOM: Open Ashtech format

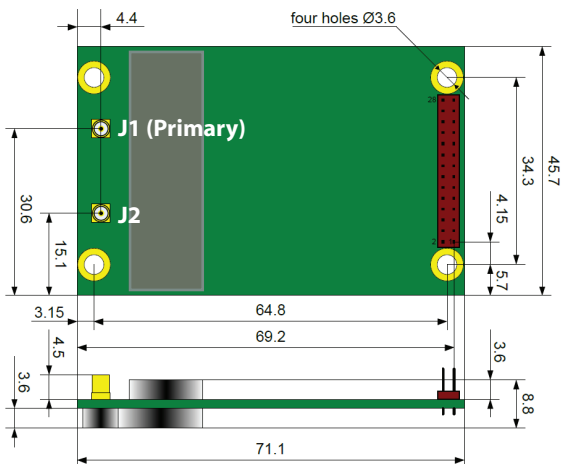
(7) VRMS for Autonomous/SBAS positions are usually twice as high as HRMS

- (8) Heading latency is usually twice as high.
- (9) VRMS = 2 x HRMS
- (10) Accuracy and TTFF specifications may be affected by atmospheric conditions, signal multipath, satellite geometry and corrections availability and quality.
- (11) Same for single base and network.
- (12) Requires L1/L2 GPS+GLONASS at a minimum.
- (13) Accuracy and TTFF specifications may be affected by atmospheric conditions, signal multipath, satellite geometry and L-band service availability. Trimble RTX correction services are only available on land.
- (14) L1/L2 data required.
- (15) Figures of pitch accuracy are twice as high.
- (16) Typical power consumption for single antenna L1 GPS/GLONASS.
- (17) This will be used if greater than the main power input voltage.
- (18) Dependent on appropriate mounting / enclosure design

NOTE: All performance values are given assuming a minimum of five satellites are used, and following the procedures recommended in the product manual. High multipath areas, high PDOP values and periods of severe atmospheric conditions may degrade performance.

## Board Layout and Dimensions

Board overall dimensions (W x H x D): 71.1 x 45.7 x 8.8 mm  
(2.8 x 1.8 x 0.34 inches)



The board can be secured onto a chassis of your choice through 4 holes dia. 3.6 mm (Use M3 threads preferably).

The above diagram (top and side views) gives all the dimensions (in mm) to allow you to perform this operation.

## I/O Connector Pin Out

Connector type: 28-pin SAMTEC TMM-114-03-G-D

Pin	Name	Type	Description	Levels
1	USB_ID	Input	Provides distinction between A-device and B-device	<ul style="list-style-type: none"> <li>A-device: To be connected to signal ground</li> <li>B-device: Keep open (not connected)</li> </ul>
2	USB_VBUS	Output	+5 V USB Power output	For A-device only (host)
3	BOOT	Input	Boot mode selection	Used for service purpose only. Normal use: Leave unconnected.
4	TPO-MID	Power	+3.3 V Ethernet	To be connected to center tap of Ethernet magnetics
5	LNA_PWR	Power	Antenna power input	Input voltage range: 4.0 to 12 V DC
6	3V3	Power	Supply voltage input	Input voltage range: 3.2 to 4.5 V DC
7	USB_D-	I/O	USB Data -	To be routed as a 90-ohm differential pair if USB is required (please use same length for the pair of wires)
8	USB_D+	I/O	USB Data +	
9	/RESETIN	Input	Reset input (see also NOTE below)	Active low
10	MFO1	Output	Multi-functional (programmable) output #1	LV-TTL, set to "1" by default (LED red) Functions: RSP, LED, PAV, BDL, OFF
11	MFO2	Output	Multi-functional (programmable) output #2	LV-TTL, set to "1" by default (LED green) Functions: RSP, LED, PAV, BDL, OFF
12	RXD3/ CANRX/ Bus I2C Data	Input	Multiplexed input	Default output= RXD3
13	EVENT	Input	Event input	LV-TTL (3.3 V) levels. Includes pull-down resistor. Pulse width equal to or greater than 500 ns.
14	GND	Power	Signal and Power Ground	-
15	TXD1	Output	COM1 Transmit Data	LV-TTL (3.3 V) levels (Port A)
16	RXD1	Input	COM1 Receive Data	LV-TTL (3.3 V) levels (Port A)
17	GND	Power	Signal and Power Ground	-
18	TXD2	Output	COM2 Transmit Data	LV-TTL (3.3 V) levels (Port B)
19	RXD2	Input	COM2 Receive Data	LV-TTL (3.3 V) levels (Port B)
20	GND	Power	Signal and Power Ground	-
21	PV	Output	"Position Valid" indicator	Active high (PV= PAV, GPIO, LV-TTL)
22	GND	Power	Signal and Power Ground	-
23	PPS	Output	PPS Output	LV-TTL (3.3 V) level
24	TXD3/ CANTX/ Bus I2C Clock	Output	Multiplexed output	Default output= TXD3
25	TPO+	Output	Ethernet twisted pair output +	To be connected to Ethernet Magnetics TD+
26	TPI+	Input	Ethernet twisted pair input +	To be connected to Ethernet Magnetics RD+

Pin	Name	Type	Description	Levels
27	TPO-	Output	Ethernet twisted pair output -	To be connected to Ethernet Magnetics TD-
28	TPI-	Input	Ethernet twisted pair input -	To be connected to Ethernet Magnetics RD-

NOTE: The reset signal on Pin 9 will not reset the configuration to the factory defaults.

Pin 9 controlling the hardware reset of the CPU, it is recommended to use it only when the board hangs up. Note that the reset button on the Evaluation Board connects pin 9 to ground.

The \$PASHS,RST reset command turns all the factory settings to defaults, including the baud rate of each port.

Sending a reset signal on pin 9 is similar to using the PASHS,INI,0 command. The CPU restarts, but all the settings are kept in the non-volatile memory and applied again right after restart. Therefore, the board will retain the baud rate changes as well as any messages that have been enabled.

## LED Indicator

---

A single LED indicator (see location on 6) allows you to monitor the board power status, the satellites received by each antenna, and the current computation status (for raw position and time) pertaining to each antenna, based on the use of different colors, as explained in the table below.

Color	Meaning
Red	Power
Green	GNSS reception
Orange	Delimiter between GNSS reception for antenna #1 and GNSS reception for antenna #2.

When no raw position and time are calculated yet for any of the two antennas, the sequence of red, green and orange flashes is the following:

1. **One red flash:** Means the board is powered properly.
2. **First sequence of green flashes:** Indicates the number of GNSS satellites received by antenna #1. The number of green flashes is however limited to 5. This means the real number of received satellites is equal to or greater than 5 when you see the green LED flashing fives times in a row.
3. **One orange flash:** Ends the first sequence of green flashes.



4. **Second sequence of green flashes:** Same as first sequence of green flashes, but this time for antenna #2.

**One red flash:** Indicates the beginning of a new sequence of flashes, as just described.

As soon as raw position and time are calculated for one antenna, or the two antennas, the sequence of red, green and orange flashes is the following:

1. **One red flash:** Means the board is powered properly.
2. **Long, solid green sequence:** Raw position and time now computed for antenna #1.
3. **One orange flash:** Ends LED indication for antenna #1.
4. **Long, solid green sequence:** Raw position and time now computed for antenna #2.

**One red flash:** Indicates the beginning of a new sequence of flashes, as just described.



Example:

**One red flash, long solid green sequence, orange flash, 3 green flashes**

Should be interpreted as:

- Raw position and time now computed for antenna #1
- Three satellites received by antenna #2



If the board is powered and there is not a single satellite received yet, the LED indicator will uniformly flash red then orange until the first satellite is received.



## Communication Ports

Port	Type	Bit rate	Notes
A	UART1	Up to 921 600 bits/s	LV-TTL levels
B	UART2	Up to 921 600 bits/s	LV-TTL levels
C	USB OTG	Up to 12 Mbits/s	Standard USB 2.0
D	UART3	Up to 921 600 bits/s	LV-TTL levels
M	Internal memory		
U	USB memory		
V	CAN	Up to 1 Mbits/s	LV-TTL levels
I	TCP/IP server		Default port ID: 8888
P	TCP/IP client		
Q	TCP/IP client		
F	TCP/IP server		Default port ID: 8889

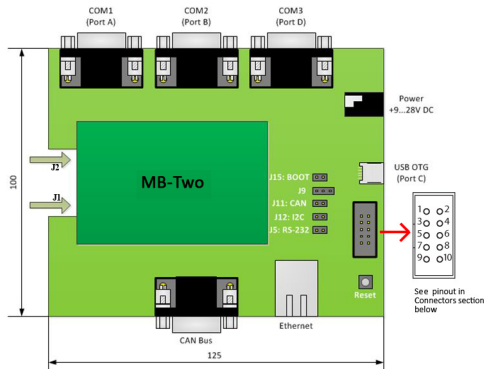
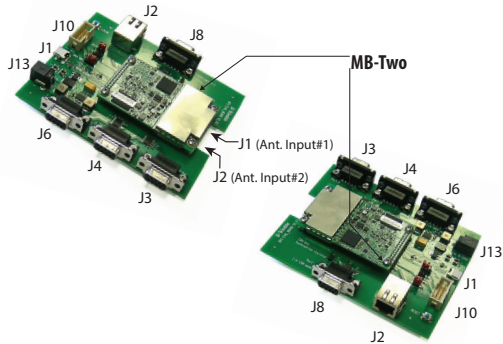
(I and F are server ports while P and Q are client ports.)

# Chapter 2. First Steps With the MB-Two

## Using the Evaluation Kit

The evaluation kit can be used at the design stage of a given application to test the performance of the board within that application.

### Board Layout



## List of Items Provided in the Kit

- 1 x Evaluation & Development I/O board
- 1 x DC power supply, 19 V DC, 3 A
- 2 x TNC-f/MMCX-m-right-angle coaxial adapters
- 5 x spare jumpers
- Utility Software (AshCom) downloadable from [ftp://ftp.ashtech.com/OEM\\_Sensor\\_ADU/](ftp://ftp.ashtech.com/OEM_Sensor_ADU/)

## Connectors

**J1: USB** port C: a standard mini male USB 2.0 connector.

**J2: Ethernet Port**, an RJ45 connector socket (PulseJack™ integrated magnetics connector model J0026D21NL).

**J3, J4, J6: RS232 ports A (J3), B (J4) and D (J6)**, standard RS232 9-pin Sub-D male-type connectors.

NOTE: Port D is available for use only if you insert jumper J5 on the evaluation kit board.

NOTE: When connecting one of the recommended external radio to an RS232 port on the evaluation kit, please use a crossover cable.

**J8: CAN Bus**, a Sub-D connector.

**J10: Terminal block**; 1PPS, Ext event input, PV, I<sup>2</sup>C, LNA power. See pinout below.

Pin	Signal	Description
1	NC	Not connected
2	/RESET IN	Reset input (active low)
3	LNA_PWR	Antenna power input (4.2-12 V DC)
4	PPS	PPS output (LV-TTL, 3.3 V)
5	SCL	Bus I2C clock (present only if jumper J12 inserted)
6	PV	"Position valid" indicator output (active high, LV-TTL, 3.3V)
7	SDA	Bus I2C data (present only if jumper J12 inserted)
8	EVENT	Event input (LV-TTL, 3.3 V)
9	GND	Signal & power ground
10	GND	Signal & power ground

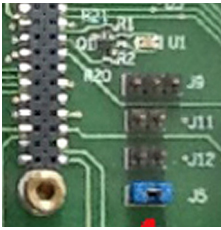
Connector type: 10-pin SAMTEC HTST-105-01-L-DV.

Use mate connector type: 10-pin SAMTEC IDSD or SMSD (not provided).

**J13: 9 to 28 V DC Power**, a standard jack connector.



## Jumpers



The board accommodates four locations for jumpers:

- J5, J11 and J12: Used to make some ports available on connectors. **Warning!** Only one jumper can be inserted at a time on one of these locations:
  - J5 inserted: Makes port D available on COM3.
  - J11 inserted: Makes CAN bus available on J8.
  - J12 inserted: Makes I2C bus available on J10.
- J9: Used to choose how to provide antenna LNAs with DC power. See section below.

### Providing Antenna LNA With Power

You have three options, depending on how you insert a jumper on J9:

- No jumper set on J9: 3 V DC is provided to the antenna(s).
- Jumper set between pins 1 and 2 on J9 (pin 1 is the closest to the USB connector): The power voltage available on pin 3 (J10) is provided to the antennas. This voltage may be as high as 12 V DC.
- Jumper set between pins 3 and 2 on J9 (pin 3 is the closest to the MB-Two board): The +5 V DC power voltage available on the evaluation board is provided to the antennas.

Choose the option that is suitable for the antennas you are using.

## Using AshCom to Communicate with MB-Two

---

AshCom is a software tool provided in the evaluation kit to communicate with the MB-Two through the Evaluation board. It is provided as a zip file.

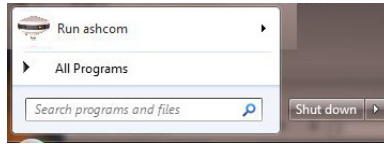
### Installing AshCom

Install AshCom on your computer as explained below:

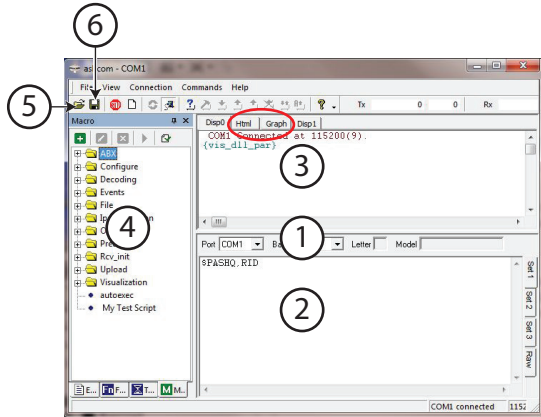
- Unzip the zip file on your computer (Windows XP or Windows 7 recommended). The result is a single msi file.
- Run the msi file by double-clicking on it. This starts the installation sequence.
- Press **Next>** and choose a folder where to install AshCom.
- Select an installation type: Choose “Typical” and click **Next>**.
- Agree to allow the program from an unknown publisher to make changes to the computer. Let the installation assistant complete the sequence.

## Introduction to AshCom User Interface

Find and run AshCom on your computer.



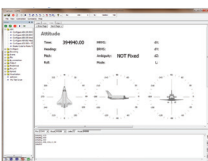
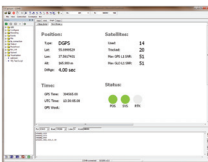
The AshCom main window is described below.



- ①: Used to set a connection between the computer and the MB-Two.
- ②: Area in which you can type \$PASH commands. Once a command is typed in, press the ENTER key to send it to the MB-Two.
- ③: Area where you can read the MB-Two response to your command.

The **html** and **Graph** tabs on top of this area give access to data visualization statuses in graphical form (see on the left).

- ④: Area in which you can select, edit and run a particular macro. Use the local toolbar (see below) to do this after you have selected the desired macro in the tree-like structure.



- **5**: Use this icon to open a command script file. Command scripts are text files with the “sc” extension. A command script is a series of \$PASH commands you want the MB-Two to execute. By running a command script, you will change the configuration of the board so that it meets your requirements.  
Command scripts may contain the following additional instructions:
  - *wait #*: Sets an idle/wait time (# expressed in seconds). Useful to let the board respond to the command that precedes in the script.
  - *log <filename>.log*: Used to start logging data to a file named “<filename>.log”.
  - *log*: Used to stop data logging.
- **6**: Use this icon to start/stop data logging (similar to log instruction in a command script file).

### Establishing a Serial Connection with MB-Two

- Install your MB-Two on the Evaluation board.
- Connect the computer to one of the serial ports on the Evaluation board using a serial DB9 cable (not provided).  
**Reminder:** A jumper on J5 is required to be able to use port D (J6). You may need a USB/Serial converter if your computer is not fitted with a serial DB9 connector.
- Connect the DC power source (provided) to J13 on the Evaluation kit and turn it on.
- Run AshCom on your computer.
- In **1**, click on the **Port** drop-down list and select the port ID used on computer side (see also *Keeping an Eye on the Computer's COM Ports Used on page 26*).
- The default baud rate of the COM ports on MB-Two is 115200 Bd. Make sure this is the selected value in the **Baud** field.

When the connection is established with the MB-Two, you will read “**COM# Connected at 115200 (9)**”. You can now start typing in and sending out commands to the MB-Two (refer to *Typical Applications Explained on page 35*).

### Establishing a TCP/IP Connection with MB-Two

- Install your MB-Two on the Evaluation board.
- Connect the Evaluation board's Ethernet port (J2) to a local network that uses a DHCP server.

- Connect the DC power source (provided) to J13 on the Evaluation kit and turn it on.
- Make a serial connection with the board as described above (*Establishing a Serial Connection with MB-Two on page 25*). This connection is required to read the IP address allocated to the MB-Two by the local network's DHCP server.

**Reminder:** A jumper on J5 is required to be able to use port D (J6). You may need a USB/Serial converter if your computer is not fitted with a serial DB9 connector.

- When this is done, run the following command in AshCom:

**\$PASHQ,TCP**

MB-Two reply example:

```
$PASHR,TCP,MOD=1,LGN=Asstech,PWD=u6huz8,ADD=192.34.76.1,  
PRT=8888*1B
```

(Where the IP address is 192.34.76.1 and port ID is 8888)

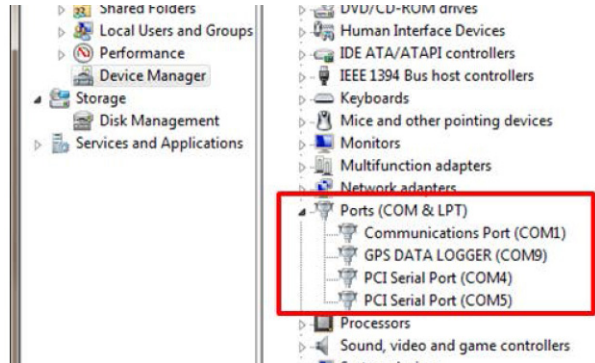
- Run a second instance of Ashcom and in **①**, click on the **Port** drop-down list and select **TCP/IP**.
- Enter the IP address and IP port (always "8888") provided by \$PASHQ,TCP. Wait until the connection is established.

## Keeping an Eye on the Computer's COM Ports Used

To know exactly which of the computer's serial ports are currently connected to the MB-Two, follow this procedure:

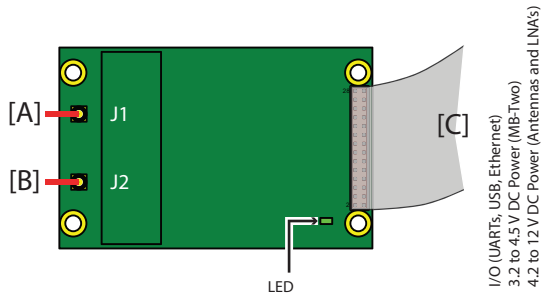
- Click on Windows 7 Start button, move the mouse pointer over **Computer**, right-click on it and select **Manage**.
- Click on **Device Manager** (on the left).

- Click on the right arrow before **Ports (COM & LPT)**. You can now see which ports on the computer are currently used. See example below.



## Connecting MB-Two to your Application

This section summarizes the different connections to be made to integrate the board directly into your application. This should be done after you have thoroughly tested the board with the evaluation kit and you have come up with a board configuration that suits your application.



Perform the connections required between the MB-Two and your application following the instructions below and referring to the above diagram.

- [A], [B]:** GNSS antenna inputs #1 (J1) and #2 (J2): Female MMCX straight connectors. For each input used, please use a coaxial cable fitted with a male MMCX connector to apply the antenna signal to the board.

In DUO mode, the good performance of the board is guaranteed provided the GNSS signals on inputs [A] and [B] are kept at similar levels. You may check these levels using command \$PASHQ,AST. The two levels should be both less than 4095 and not different from each other by more than 750. It is therefore your responsibility to use the suitable antennas and coaxial cables. You may have to use an additional preamplifier or attenuator if one antenna uses a very different length of coaxial cable compared to the other one.

**IMPORTANT!** Each antenna used should incorporate a built-in LNA (Low-Noise Amplifier). The gain of the LNA minus the signal loss in the coaxial cable connecting the LNA to the MB-Two board should be in the range 17 to 47 dB (for L1 band) and 23 to 50 dB (for L2 band).

A DC voltage is delivered on the center conductor of each GNSS antenna input, which may be either the external DC voltage provided on pin 5, I/O connector (4.0 to 12.0 V DC, see [C] below) if it's greater than the main power voltage, or the main power voltage otherwise. In this case, the actual DC power voltage applied to the antenna will be the DC voltage present on pin 6 (i.e. 3.2 to 4.5 V DC) minus 0.2 V.

This DC voltage is used to power the antenna LNA when connecting the antenna to the board's GNSS antenna input. The DC output current is monitored internally from which the antenna status is deduced:

Out DC current	Antenna status
5-150 mA	Normal
> 150 mA	Antenna shorted
< 5 mA	No antenna connected (or DC power used)

The DC output is also protected from high DC voltages (up to 12 V) that might unintentionally be applied to the GNSS antenna input via the antenna cable.

- **[C]:** I/O connector: A 28-contact connector (see page 18 for connector pinout and type as well as signal specifications). All non-RF connections are routed through this connector. Perform the connections you need for your application:
  - **Communication ports** (A, B, C). Choose the port, or ports, you need for applying serial commands and output the messages you need for your application.
  - **Main power supply.** Used to power the board. Input voltage range: + 3.2 - 4.5 V DC

Power consumption: 1.2 W typical

Power ON/OFF command input available.

- **Antenna power supply:** Used to power one or two of the antennas (and their associated LNAs) connected to the board (4.2 to 12 V DC)
- External event signal input (Event Marker)
- PPS output
- Reset input
- Miscellaneous signals (External LED outputs, etc.).

## Precautions Before Turning Off the Board

---



When integrating the MB-Two into your application, make sure that all power-off sequences will be executed this way:

- Apply the command below, followed by a 15-second idle time:

**\$PASHS,PWR,OFF**

Don't do anything during the next 15 seconds

- Then you can remove power from the board.

If you don't observe this rule or you don't wait long enough after running the command, the board may lose data definitively. See also *PWR,OFF: Powering Off the Board* on page 273.

## GNSS Antenna Setup for Heading/Attitude Measurements

---

### Choosing the Baseline Length

In theory, the baseline length (i.e. the horizontal distance between the phase centers of the two GNSS antennas used, also called antenna separation) can be set between 30 centimeters and 20 meters (or more).

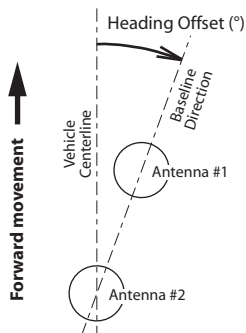
In practice, you will choose the baseline length taking into account the level of expected accuracy as well as the various installation constraints in the vehicle.

### Heading-Roll-Pitch Offsets

#### Heading Offset Definition

Ideally, the antennas should be installed to generate a baseline strictly parallel or perpendicular to the vehicle centerline. However, you may also be facing some installation constraints on your vehicle compelling you to install the

antennas differently. The heading offset describes the non-alignment of the baseline with the vehicle centerline. When the baseline is strictly parallel to the centerline and the baseline is oriented in the direction of forward movement, the heading offset is zero. In all other cases, the offset is not zero and should be measured as shown in the diagram below.



The non-alignment of the baseline with the vehicle centerline may be intentional (see explanations further below).

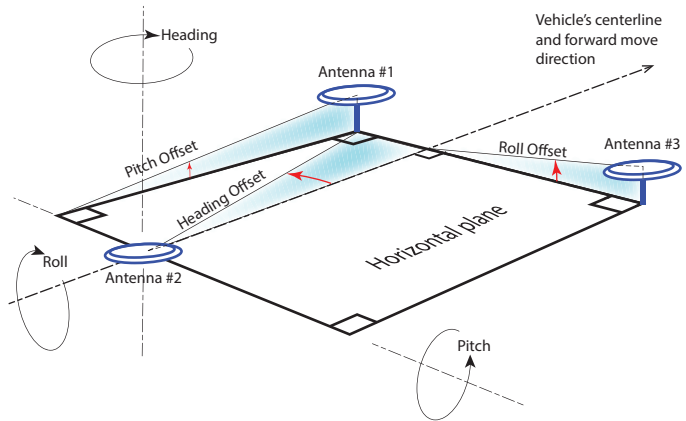
### Roll-Pitch Offset Definitions

Ideally, your antennas should be installed at the same elevation. You may however be facing some installation constraints on your vehicle compelling you to install the antennas at different elevations.

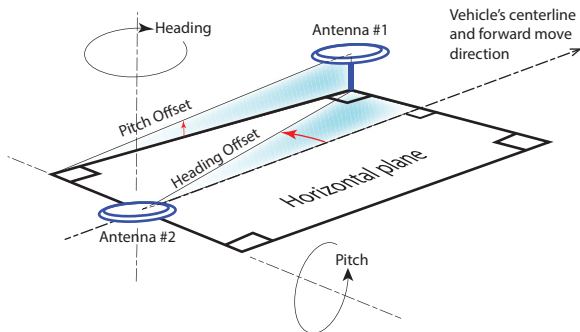
If that is the case, you should measure the pitch and roll offsets shown in the diagrams below (three antenna setup) using adequate instrumentation.



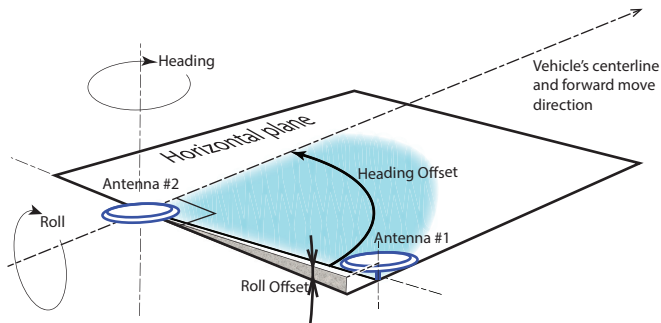
### Three-Antenna Setup (Attitude)



### Two Antenna-Setup (Heading+Pitch), Baseline quasi-parallel to vehicle's centerline:

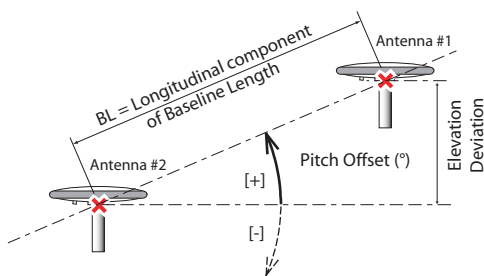


Two-Antenna Setup (Heading+Roll), Baseline quasi-perpendicular to vehicle's centerline:



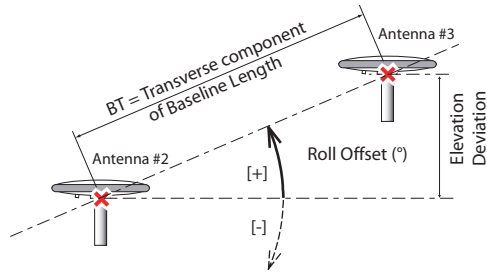
If you cannot measure angles directly, but only distances, you can still compute the angles from your distance measurements, as explained below (you however need to measure the heading offset):

- First measure the antenna elevation deviations and the baseline lengths.
- Assuming  $BL$  is the value of baseline length projected onto the vehicle's centerline, and  $BT$  the value of baseline length projected onto a plane perpendicular to the vehicle's centerline, calculate the pitch and roll offsets as shown in the two diagrams below (offset sign: positive if Antenna #1 is higher than Antenna #2, negative otherwise).



$$BL (m) = \text{Baseline Length (m)} \times \cos(\text{heading offset})$$

$$|\text{Pitch Offset (}^\circ\text{)}| = \arcsin \frac{\text{Elevation Deviation (m)}}{BL (m)}$$



$$BT (m) = \text{Baseline Length (m)} \times \sin(\text{heading offset})$$

$$|\text{Roll Offset (}^\circ\text{)}| = \arcsin \frac{\text{Elevation Deviation (m)}}{BT (m)}$$



**The pitch and roll offsets should not be greater than 45 degrees (or less than -45 degrees), or the receiver will consider the antenna setup to be invalid and so will not deliver any heading, roll or pitch measurements.**

### Entering Offsets

Use the \$PASHS,3DF,OFS command.

**Example:** If Heading Offset is +5°, Pitch Offset is +3° and Roll Offset is +6.5° enter these three offsets using the following command:

**\$PASHS,3DF,OFS,5,3,6.5**

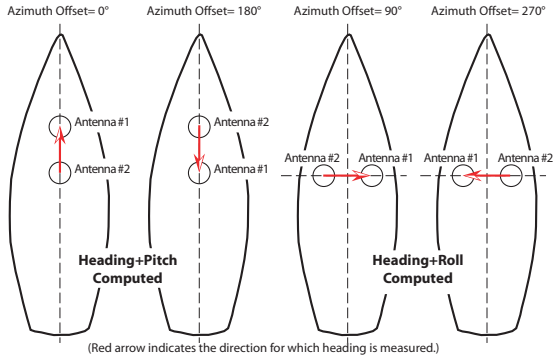
If you are using two antennas (i.e. one baseline), depending on how you orientate the baseline with respect to the vehicle's centerline, i.e. parallel or perpendicular to the vehicle's centerline (see above), the roll or pitch angle will respectively be zero. To determine the roll or pitch angle when it exists, follow the same procedure as the one described above for three antennas.

With a two-antenna setup, the two possible offset angles are renamed "azimuth offset" (instead of "heading offset") and "elevation offset" (instead of "roll -or pitch- offset").

## Correlation Between Heading Offset, Antenna Setup & Measurements Made

Consider the following four setups before installing your antennas. A ship is shown in the examples but this could be any other type of vehicle.

Depending on the type of measurements you wish the receiver to perform (heading + roll or heading + pitch) and the installation possibilities offered in the vehicle, you will choose the most appropriate setup and set the heading offset accordingly.





# Chapter 3. Typical Applications Explained



## Preamble

---

This chapter presents the MB-Two's typical applications, introducing the underlying proprietary \$PASH commands supporting these operating modes. The commands are explained so that you can get a good understanding of the operation principles.

*Note: Refer to "Using AshCom to Communicate with MB-Two on page 23" to learn how to send \$PASH commands to the MB-Two.*

But remember it's generally easier and faster to use the built-in Web Server rather than working in command mode.

*Note: To become familiar with the built-in Web Server, see "Setting the MB-Two from the Web Server on page 71".*

However, while prototyping your MB-Two application, while testing the board before effective implementation in your application, for fine-tuning the board configuration, for some applications going off the beaten track, or for test purposes in general, using the command mode will be proving an essential tool.

### Interfacing with the Board

In all the typical applications described in this chapter, follow the instructions in *Establishing a Serial Connection with MB-Two on page 25* to communicate with the board using AshCom and a serial connection (in short, AshCom running on computer connected to port D for example; don't forget to insert jumper J5 on the Evaluation board).

## Preliminary Steps

As a preliminary step to setting up an “unknown” board, it is always a good practice to reset the board, then read some useful information about the board (identification, version, time & date, port settings):

Resetting the board:

**\$PASHS,RST**

(Wait about 15 seconds to let the board complete the reset)

Reading board identification, installed firmware and installed options:

**\$PASHQ,RID**

Reading port settings:

**\$PASHQ,PRT**

Reading time & date:

**\$PASHQ,ZDA**

Reading firmware version:

**\$PASHQ,VER**

Reading installed firmware options:

**\$PASHQ,OPTION**

Still as a preliminary step, some other commands should be routinely executed as well, unless in your application you agree with all the default settings, in which case nothing else needs to be done.

Enabling/disabling the tracking of GNSS constellations; See:

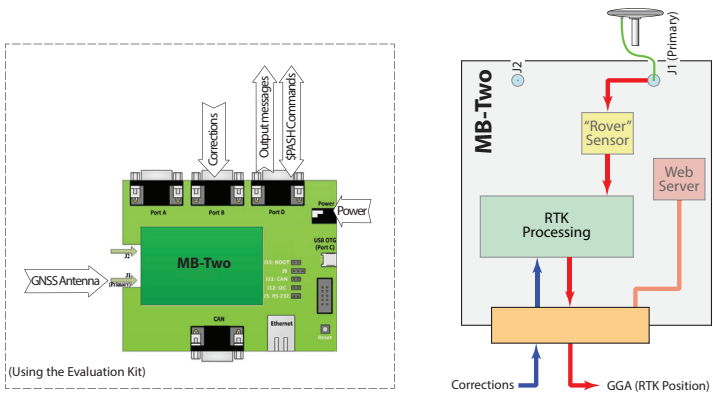
**\$PASHS,GPS \$PASHS,GLO \$PASHS,SBA \$PASHS,GAL  
\$PASHS,BDS \$PASHS,QZS**

Antenna settings; See:

**\$PASHS,ANH \$PASHS,ANT \$PASHS,ANP,OWN/REF  
\$PASHS,ANP,OUT \$PASHS,ANR**

*Note: Refer to chapters 6 and 7 for a detailed description of each command.*

## Rover Computing RTK Position



- MB-Two will operate with a single sensor (antenna #1 sensor) to provide an RTK solution. This is what we call operating the board in SOLO mode. Use this command to activate the SOLO mode (“SOL” for “SOLO”):

**\$PASHS,SNS,SOL**

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

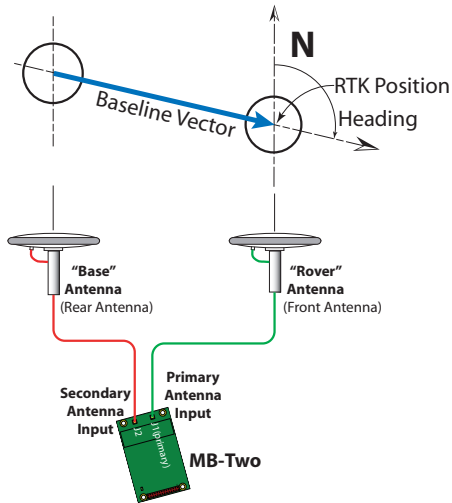
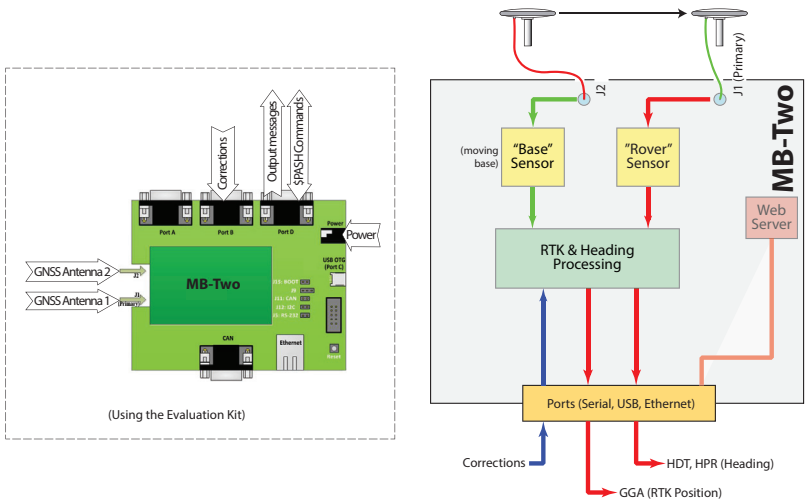
```
$PASHS,1,BLN,ON,?
$PASHS,2,BLN,SAM
$PASHS,3,BLN,SAM
$PASHS,3DF,OFF
$PASHS,RTK,ON,1,2,3
$PASHS,BRV,OFF
```

- The RTK position will be available by programming this output message (on port D for example, at 1 second):

**\$PASHS,NME,GG,A,D,ON,1**

# Rover Computing Internal Heading + RTK Position

**Important:** Heading is computed based on the reception of L1/L2 GNSS signals, just like RTK position, which is computed using preferably L1+L2 GPS/GLONASS/QZSS signals. You should therefore choose the antennas accordingly.



- MB-Two needs to operate with two internal sensors to compute heading. This is what we call operating the board



in DUO mode<sup>1</sup>. Run this command to activate the DUO mode:

**\$PASHS,SNS,DUO,2**

(The "2" means both antenna inputs are supposed to receive L1+L2 signals)

(For more information, see *SNS: Configuring the M-Sensor on page 305*.)

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

\$PASHS,2,BLN,ON,2,1

\$PASHS,1,BLN,ON,?

\$PASHS,3,BLN,SAM

\$PASHS,3DF,ON,2

\$PASHS,RTK,ON,1,3

\$PASHS,BRV,OFF

Auto-calibration will start automatically after the command is ACKed (unless you later enter the vector components of the baseline using \$PASHS,3DF,V12..).

- If you accurately know the components of the baseline vector (example: x12= 0.87 m) and you want to enter them now, use this command:

**\$PASHS,3DF,V12,0.87**

(This will stop the auto-calibration in progress.)

- Conversely, you may consider that the module of the baseline vector may slightly change over time, and want the board to keep on computing heading in this case, i.e. without the need to recalibrate. For this type of application, run for example this command if the baseline used is baseline #1:

**\$PASHS,3DF,ON,1,FLX**

- You may check the current M-sensor status and receiver status using the following command:

**\$PASHQ,SNS**

- Set the output messages you need on any port. These may be one of the following:

Heading at 10 Hz on port D:

**\$PASHS,NME,HDT,D,ON,0.1**

Vector at 10 Hz on port D:

**\$PASHS,NME,VCT,D,ON,0.1**

Heading + Pitch/Roll on port D

1. In this mode, the "base" sensor provides internally a flow of 10-Hz corrections to the "rover" sensor.

**\$PASHS,NME,HPR,D,ON,0,1**

Antenna #1 position (standalone) at 1 Hz on port D (NMEA):

**\$PASHS,NME,GGA,D,ON,1**

Antenna #1 position (standalone) at 1 Hz on port D (ATOM):

**\$PASHS,ATM,PVT,D,ON,1,&COO,ERR,LCY**

Same message types, but relevant to Antenna #2 location: Insert “,2” after the \$PASHS header:

**\$PASHS,2,NME,...**

**\$PASHS,2,ATM,...**

NOTE: When the same types of messages are output on the same port for two different locations (i.e. for antenna #1 and antenna #2), special markers are inserted in the flow of messages so the recipient device can recognize which messages describe which antennas.

Example with GGA messages:

**\$PASHD,#1,123456.00,ABCD,BEG\*cc<cr><lf>**

\$GPGGA,...

**\$PASHD,#1,123456.00,ABCD,END\*cc<cr><lf>**

**\$PASHD,#2,123456.00,ABCD,BEG\*cc<cr><lf>**

\$GPGGA,...

**\$PASHD,#2,123456.00,ABCD,END\*cc<cr><lf>**

Each GGA message is inserted between a beginning (BEG) and end (END) marker (shown in bold characters in the example above). The marker header indicates for which antenna (or sensor) the GGA message that immediately follows refers to.

For example, a GGA message inserted between two “\$PASHD,#1,..” lines means the message is about antenna #1.

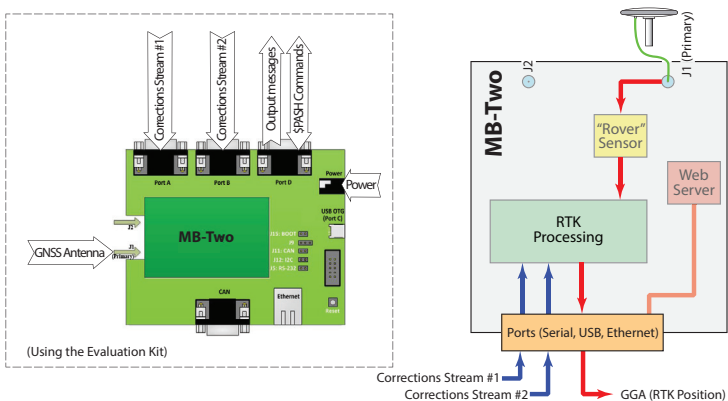
If you just want the MB-Two to compute internal heading (and not additionally an RTK position for antenna #1), **you don’t need to change anything to the configuration of the board, but simply disconnect the incoming correction stream from the board.**

## Rover Computing Hot Standby RTK Position

Hot Standby RTK is that process through which the MB-Two internally computes two RTK positions instead of one, but delivers only the best one through the programmed output messages. The important thing to mention here is that the two RTK positions should be computed from two independent correction streams.

If for some reason, one of the two RTK positions happened to be missing, then an internal switching mechanism would make sure the remaining valid RTK solution is always made available at the output.

In the description below, it is assumed that one correction stream enters the board through port A, and the other through port B. This is just an example. You may use different ports in your application.



- MB-Two will operate with a single sensor (antenna #1 sensor). This is what we call operating the board in SOLO mode. Use this command to activate the SOLO mode (“SOL” for “SOLO”):

```
$PASHS,SNS,SOL
```

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

```
$PASHS,1,BLN,ON,?  
$PASHS,2,BLN,SAM  
$PASHS,3,BLN,SAM
```

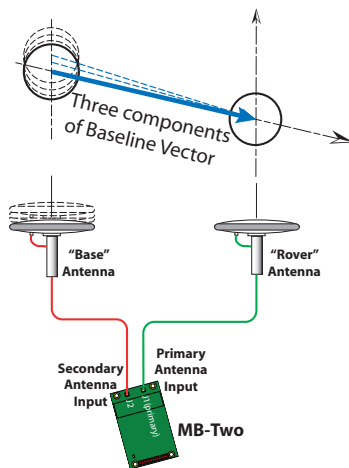
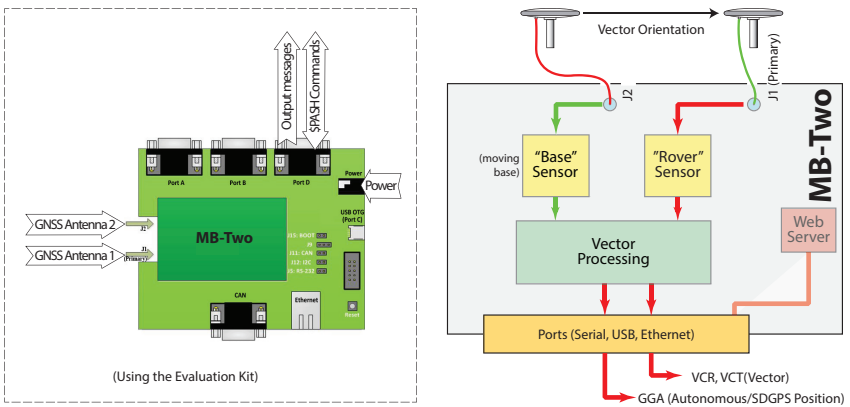
```
$PASHS,3DF,OFF  
$PASHS,RTK,ON,1,2,3  
$PASHS,BRV,OFF
```

- Set the first baseline to use corrections received on port A:  
**\$PASHS,1,BLN,ON,A,1**
- Set the second baseline to use corrections received on port B:  
**\$PASHS,2,BLN,ON,B,1**
- Ask to board to compute RTK positions from baselines 1 and 2:  
**\$PASHS,RTK,ON,1,2**
- In all cases, the RTK position will be available by programming this output message (on port D for example, at 1 second):  
**\$PASHS,NME,GGA,D,ON,1**

# Rover Computing Internal Vector

In this application, the MB-Two is connected to two antennas for which you want to continuously measure the three components of the vector connecting the two antennas. Because the two antennas are connected to the same board, the movements of one antenna with respect to the other can only be limited. In addition, the vertical slope of the baseline between two antennas should not exceed  $45^\circ$ .

**Important!** Vector is computed based on the reception of L1/L2 GNSS signals, just like RTK position, which is computed using preferably L1+L2 GPS/GLONASS/QZSS signals. You should therefore choose the antennas accordingly.



- MB-Two needs to operate in DUO mode, i.e. with its two internal sensors. Run this command to activate the DUO mode:  
**\$PASHS,SNS,DUO,2**  
 (The "2" means both antenna inputs are supposed to receive L1+L2 signals)

(For more information, see *SNS: Configuring the M-Sensor on page 305*.)

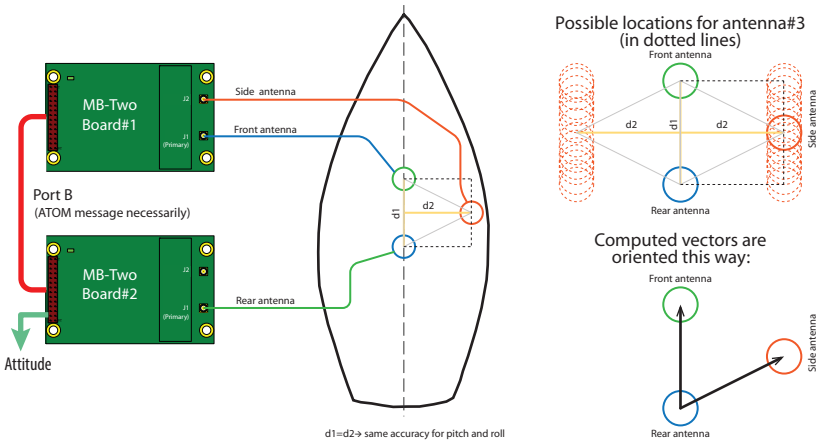
**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

```
$PASHS,2,BLN,ON,2,1
$PASHS,1,BLN,ON,?
$PASHS,3,BLN,SAM
$PASHS,3DF,ON,2
$PASHS,RTK,ON,1,3
$PASHS,BRV,OFF
```

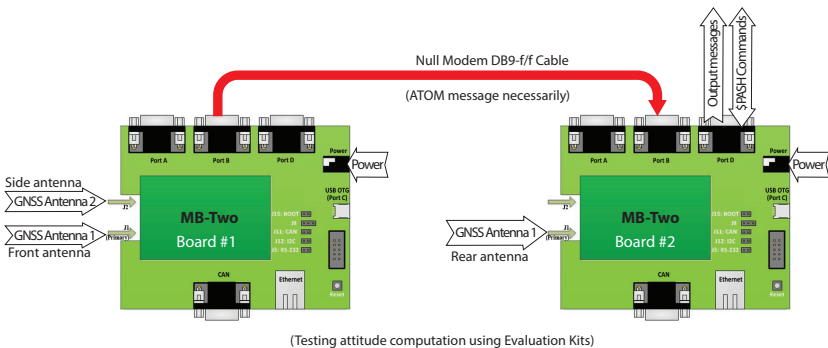
- Disable attitude and RTK computation:  
**\$PASHS,3DF,OFF**  
**\$PASHS, RTK,OFF**
- Complete the definition of the baseline by making sensor #2 the moving base (the secondary antenna is connected to sensor #2):  
**\$PASHS,1,BLN,ON,2**
- Disable baselines #2 and #3:  
**\$PASHS,2,BLN,OFF**  
**\$PASHS,3,BLN,OFF**
- Enable vector processing over baseline #1 (baseline connecting the secondary to the primary antenna):  
**\$PASHS,BRV,ON,1**
- Define the output message that will provide the computed vector, for example on port A at 0.5 second:  
**\$PASHS,NME,VCR,A,ON,0.5**

NOTE: You don't need to define the internal corrections message required from sensor #2. It is automatically generated by the board.

# Rover Computing Attitude



NOTE: Vectors are oriented in the opposite direction compared to the Heading setup (see page 38).



This requires firmware option [E] (Attitude) installed on the board computing attitude (board #2).

- Initiate bidirectional communication with board #1 through board #2 using the daisy chain command:  
`$PASHS,DSY,B,D`  
`$PASHS,DSY,D,B`
- Configure board#1:  
 Enabling use of second antenna:  
`$PASHS,SNS,DUO,2`  
 Disabling the three baseline engines:

**\$PASHS,1,BLN,OFF**

**\$PASHS,2,BLN,OFF**

**\$PASHS,3,BLN,OFF**

Disabling RTK:

**\$PASHS,RTK,OFF**

Disabling attitude and heading:

**\$PASHS,3DF,OFF**

Requesting generation of 10-Hz corrections stream (moving base) related to antenna #1 (board #2) on port B; Use ATOM format necessarily:

**\$PASHS,ATM,RNX,B,ON,0.1,&SCN,204**

Enabling output message delivering antenna #1 parameters:

**\$PASHS,ATM,ATR,B,ON**

Requesting generation of 10-Hz corrections stream (moving base) related to antenna #2 (board #2) on port B; Use ATOM format necessarily:

**\$PASHS,2,ATM,RNX,B,ON,0.1,&SCN,204**

Enabling output message delivering antenna #2 parameters:

**\$PASHS,2,ATM,ATR,B,ON**

- Configure board#2:

Set the board in SOLO mode:

**\$PASHS,SNS,SOL**

Disabling communication with board #2, restoring communication with board #1:

**\$PASHS,DSY,OFF**

Disabling first baseline engine:

**\$PASHS,1,BLN,OFF**

Setting second baseline engine to work with corrections stream from board#2, antenna #1 (rear antenna), via board#1's port B:

**\$PASHS,2,BLN,ON,B1**

Setting third baseline engine to work with corrections stream from board#2, antenna #2 (side antenna), via board#1's port B:

**\$PASHS,3,BLN,ON,B2**

Disabling RTK:

**\$PASHS,RTK,OFF**

Enabling attitude computation, specifying which baselines are processed:

**\$PASHS,3DF,ON,2,3**

Entering coordinates of first vector (will stop auto-calibration in progress):

**\$PASHS,3DF,V12,x12,y12,z12**

(V12 refers to baseline from front antenna to rear antenna)

Entering coordinates of second vector (will stop auto-calibration in progress):

**\$PASHS,3DF,V13,x13,y13,z13**

(V13 refers to baseline from front antenna to side antenna)

- Program output messages (heading, attitude) on board#2:

Enabling heading message at 10 Hz on port D:

**\$PASHS,NME,HDT,D,ON,0.1**

Enabling vector message at 10 Hz on port D:

**\$PASHS,NME,HPR,D,ON,0.1**



- You may define other messages on any port. See examples below:

Position at 1 Hz on port D:

**\$PASHS,NME,GGA,D,ON,1**

Satellite status messages on port D:

**\$PASHS,NME,GSV,D,ON**

Position accuracy messages at 1 Hz on port D:

**\$PASHS,NME,GST,D,ON,1**

Velocity at 1 Hz on port D:

**\$PASHS,NME,VTG,D,ON,1**

Sensor latency at 1 Hz on port D:

**\$PASHS,NME,LTN,D,ON,1**

Position data (standalone) at 1 Hz on port D in binary format (ATOM):

**\$PASHS,ATM,PVT,D,ON,1,&COO,ERR,LCY**

CAUTION: The pitch and roll computed in this configuration are relative values. A reported 0-degree pitch or roll means the angle has not changed since you calibrated the system. It does not mean that the three antennas are all in the same horizontal plane.

You may for example have a setup where two antennas have a pitch or roll offset of 15°. The calibration process assumes that this is the standby position of the antenna setup.

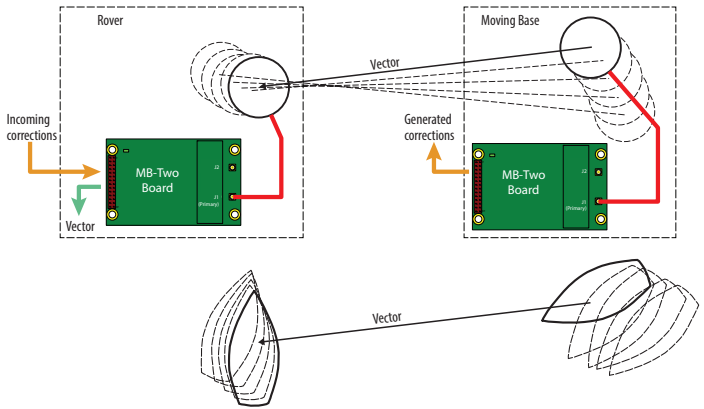
Following the calibration, the roll or pitch will stay at zero as long as no change in attitude is detected.

## Rover Computing External Vector (Relative Positioning)

---

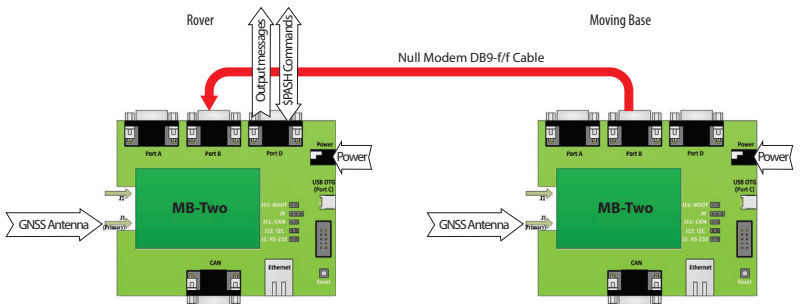
Relative positioning consists in computing the changing vector between two moving vehicles. The vehicle in which the

components of the vector need to be known is the rover, and the other one is a moving base.



NOTE: Use a dedicated transmission channel (radio, network, other) through the appropriate ports to transport base corrections to the rover.

To test this application with two evaluation kits, connect the two B ports together using a null modem cable.



(Testing vector computation using Evaluation Kits)

### Moving Base Configuration

See *Moving Base* on page 60.

### Rover Configuration

Whether the board has been set up in SOLO or DUO mode does not matter, provided you use the following commands to set the rover board (through its port D for example):

- Set the baseline dedicated to processing the vector:  
**\$PASHS,1,BLN,ON,B,1**
- Deactivate the other two baselines:  
**\$PASHS,2,BLN,OFF**  
**\$PASHS,3,BLN,OFF**
- Disable RTK position processing:  
**\$PASHS,RTK,OFF**
- Disable attitude processing:  
**\$PASHS,3DF,OFF**
- Enable vector processing over baseline #1:  
**\$PASHS,BRV,ON,1**
- Program the output message that will provide the computed vector, for example on port A at 0.2 second:  
**\$PASHS,NME,VCR,A,ON,0.2**

## Corrections Transport

In the rover configuration above, the corrections are routed via port B because the example described is based on the use of two Evaluation kits used side by side.

In real-life applications, the two boards will operate at different locations and so you should define another way of transporting the corrections, for example via modem or radio.

On the moving base, you should therefore generate the corrections message to the port connected to the transmitting device, for example port I:

Observations and kinematic position:  
**\$PASHS,ATM,RNX,I,ON,1,&SCN,204**

Receiver and antenna names, other attributes:  
**\$PASHS,ATM,ATR,I,ON**

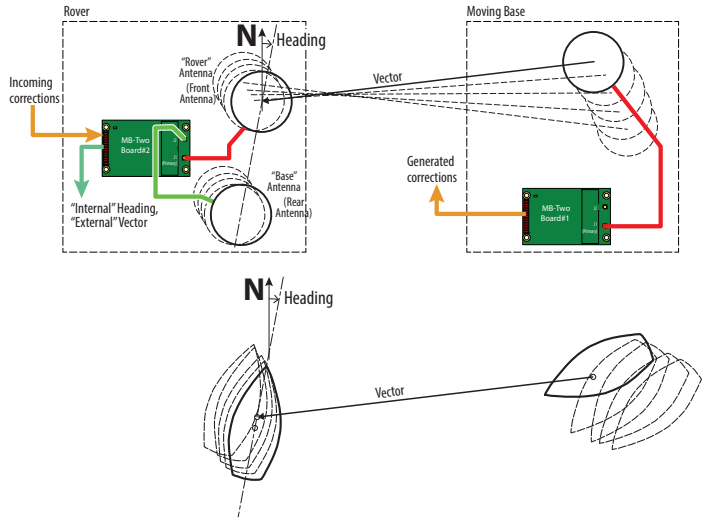
On the rover, the first baseline should be defined to use the corrections coming from the port connected to the receiving device, for example port C:

**\$PASHS,1,BLN,ON,C**

Knowing that you can always let the board find the port receiving the corrections using the “?” symbol instead of the port ID.

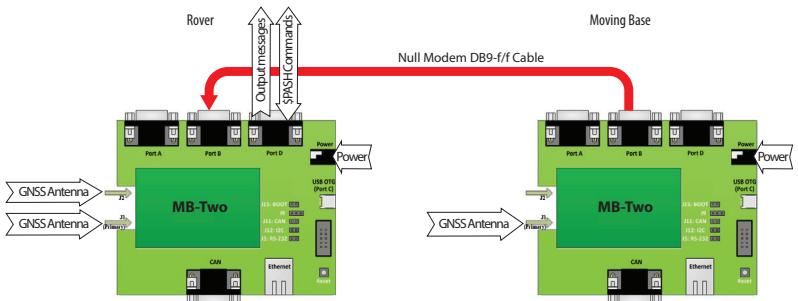
## Rover Computing Internal Heading and External Vector

You may combine the two applications described in *Rover Computing Internal Heading + RTK Position* on page 38 and *Rover Computing External Vector (Relative Positioning)* on page 47. In this case however, no RTK position can be delivered for the rover location.



NOTE: Use a dedicated transmission channel (radio, network, other) through the appropriate ports to transport base corrections to the rover.

To test this application with two evaluation kits, connect the two B ports together using a null modem cable.



(Testing vector computation using Evaluation Kits)

- Configure board#1 as a moving base:

```
$PASHS,SNS,SOL
$PASHS,STI,<Base ID>
$PASHS,ATM,RNX,B,ON,1,&SCN,204
$PASHS,ATM,ATR,B,ON,1
```

For more details, refer to *Moving Base on page 60*.

- Configure board#2 as follows:

```
$PASHS,SNS,DUO,2
$PASHS,3DF,V12,<baseline length>
$PASHS,NME,HDT,D,ON,<output rate>
$PASHS,NME,HPR,D,ON,<output rate>
```

For more details, refer to *Rover Computing Internal Heading + RTK Position on page 38*.

- Take these additional steps to complete the setup:

Re-define baseline#3 as follows (assuming corrections from the external moving base enter the board via port B):

```
$PASHS,3,BLN,ON,B
```

Enable the relative mode over baseline #3 (vector computed by baseline engine#3):

```
$PASHS,BRV,ON,3
```

Disable RTK position processing:

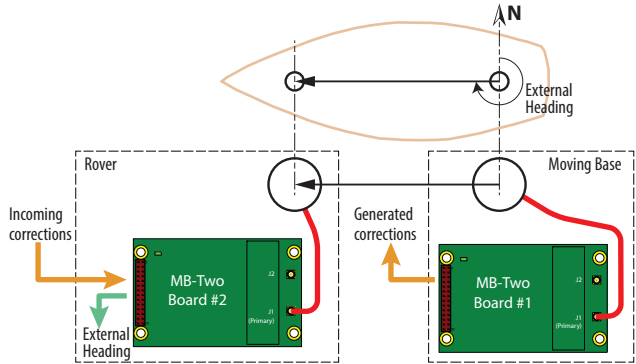
```
$PASHS,RTK,OFF
```

Program the output message that will provide the computed vector, for example on port A at 0.5 second:

```
$PASHS,NME,VCR,A,ON,0.5
```

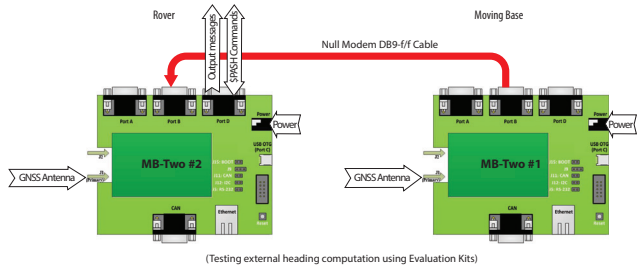
## Rover Computing External Heading

The board delivering external heading measurements should be fitted with firmware options [J] (or [F]) and [E]. Another MB-Two board configured as a moving base will provide the required connections.



NOTE: Use a dedicated transmission channel (radio, network, other) through the appropriate ports to transport base corrections to the rover. Corrections *must* be generated in ATM,RNX format.

To test this application with two evaluation kits, connect the two B ports together using a null modem cable.



- Configure board# 1 as a moving base:

```
$PASHS,SNS,SOL
$PASHS,STI,<Base ID>
$PASHS,ATM,RNX,B,ON,1,&SCN,204
$PASHS,ATM,ATR,B,ON,1
```

For more details, refer to *Moving Base* on page 60.

- Configure board#2 as a rover computing heading, using corrections from an external moving base:

**\$PASHS,SNS,SOL**

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

```
$PASHS,1,BLN,ON,?
$PASHS,2,BLN,SAM
$PASHS,3,BLN,SAM
$PASHS,3DF,OFF
$PASHS,RTK,ON,1,2,3
$PASHS,BRV,OFF
```

Disable RTK processing on all baselines:

**\$PASHS, RTK,OFF**

Then define the base and rover data processed in baseline#1 (assuming moving base data enter the board through port B):

**\$PASHS, 1,BLN,ON,B1,1**

Then enable the heading processing on baseline#1, specifying the Flex mode to let the board compute heading without having to go through a calibration step:

**\$PASHS,3DF,ON,1,FLX**

Then define the desired output messages:

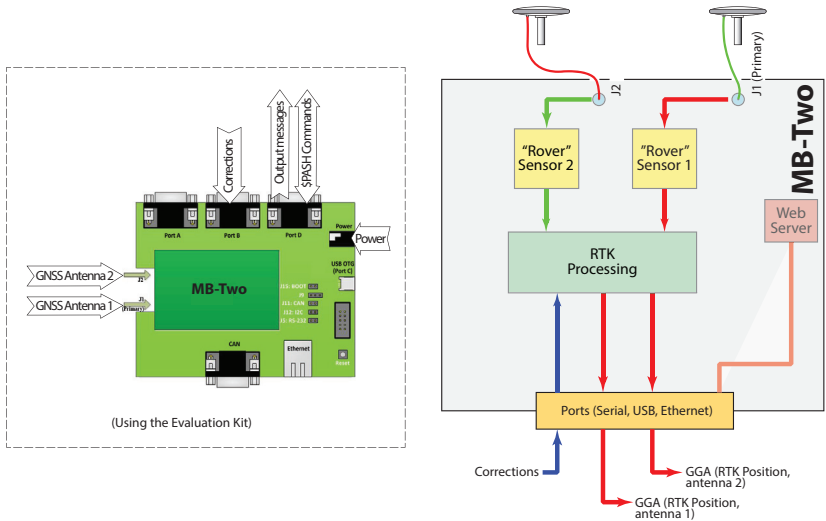
**\$PASHS,NME,HDT,D,ON,<output rate>**

or/and

**\$PASHS,NME,HPR,D,ON,<output rate>**

## Rover Computing Two Independent RTK Positions (Dual RTK)

Dual RTK allows the board to deliver two RTK positions, one for each of its antennas, using the same source of corrections.



- MB-Two has to operate with two internal sensors, one per antenna:

**\$PASHS,SNS,DUO,2**

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

**\$PASHS,2,BLN,ON,2,1**

**\$PASHS,1,BLN,ON,?**

**\$PASHS,3,BLN,SAM**

**\$PASHS,3DF,ON,2**

**\$PASHS,RTK,ON,1,3** (RTK position processing running on baselines 1 & 3)

**\$PASHS,BRV,OFF** (vector processing turned off)

- Assuming you will be using baselines #1 and #3, disable baseline #2:

**\$PASHS,2,BLN,OFF**

- Define baseline #3, for which sensor #2 (i.e. the secondary antenna) is used and any incoming corrections are used to compute an RTK position:

**\$PASHS,3,BLN,ON,?,2**



- Turn off attitude processing:  
**\$PASHS,3DF,OFF**
- Set the output messages (typically GGA) that will deliver RTK positions:  
Antenna #1 at 2 Hz on port D:  
**\$PASHS,1,NME,GGA,D,ON,0.5**  
Antenna #2 at 2 Hz on port D:  
**\$PASHS,2,NME,GGA,D,ON,0.5**

NOTE: When the same types of messages are output on the same port for two different locations (i.e. for antenna #1 and antenna #2), special markers are inserted in the flow of messages so the recipient device can recognize which messages describe which antennas.

Example with GGA messages:

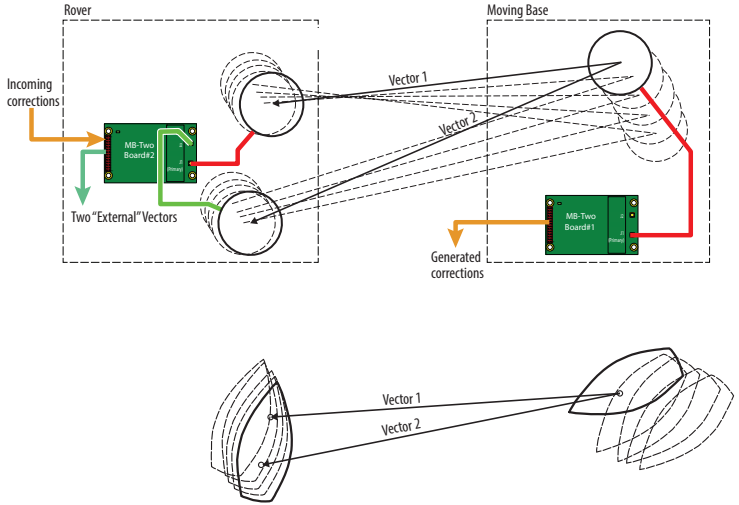
```
$PASHD,#1,123456.00,ABCD,BEG*cc<cr><lf>
$GPGGA,...
$PASHD,#1,123456.00,ABCD,END*cc<cr><lf>
$PASHD,#2,123456.00,ABCD,BEG*cc<cr><lf>
$GPGGA,...
$PASHD,#2,123456.00,ABCD,END*cc<cr><lf>
```

Each GGA message is inserted between a beginning (BEG) and end (END) marker (shown in bold characters in the example above). The marker header indicates for which antenna (or sensor) the GGA message that immediately follows refers to.

For example, a GGA message inserted between two "\$PASHD,#1,..." lines means the message is about antenna #1.

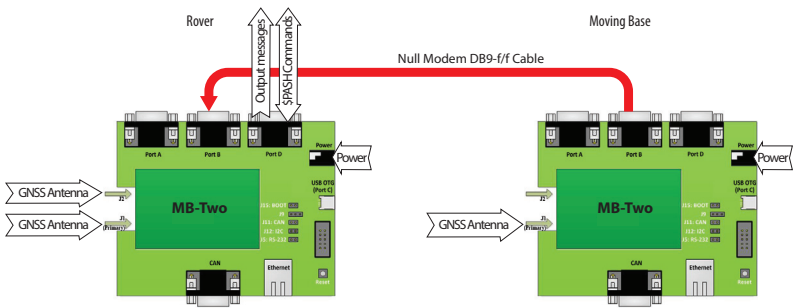
# Rover Delivering Two External Vectors (Dual Relative RTK)

Dual Relative RTK allows the board to deliver two vectors, one for each of its antennas, using the same source of corrections.



NOTE: Use a dedicated transmission channel (radio, network, other) through the appropriate ports to transport base corrections to the rover.

To test this application with two evaluation kits, connect the two B ports together using a null modem cable.



(Testing vector 1 & vector 2 computation using Evaluation Kits)

## Moving Base Configuration

See *Moving Base* on page 60.

## Rover Configuration

- Set the board to operate in DUO mode (two antennas used):

```
$PASHS,SNS,DUO,2
```

**Important!** After you have run this command, the board is in a new operating status, as if you had run the following commands:

```
$PASHS,2,BLN,ON,2,1
```

```
$PASHS,1,BLN,ON,?
```

```
$PASHS,3,BLN,SAM
```

```
$PASHS,3DF,ON,2
```

```
$PASHS,RTK,ON,1,3
```

```
$PASHS,BRV,OFF
```

- Set the baselines (#1 and #3) dedicated to computing vector 1 and vector 2:

```
{Baseline #1 already set thru previous command as: $PASHS,1,BLN,ON,?}
```

```
$PASHS,3,BLN,ON,?,2
```

- Deactivate the second baseline:

```
$PASHS,2,BLN,OFF
```

- Disable RTK position processing:

```
$PASHS,RTK,OFF
```

- Disable attitude processing:

```
$PASHS,3DF,OFF
```

- Enable vector processing over baseline #1 and baseline #3:

```
$PASHS,BRV,ON,1,3
```

- Program the output message that will provide the computed vectors, for example on port A at 0.2 second:

```
$PASHS,1,NME,VCR,A,ON,0.2
```

```
$PASHS,2,NME,VCR,A,ON,0.2
```

NOTE: When the same types of messages are output on the same port for two different sensors (i.e. for antenna #1 and antenna #2), special markers are inserted in the flow of messages so the recipient device can recognize which messages describe which antennas.

Example with VCR messages:

```

$PASHD,#1,123456.00,ABCD,BEG*cc<cr><lf>
$PASHR,VCR,...
$PASHD,#1,123456.00,ABCD,END*cc<cr><lf>
$PASHD,#2,123456.00,ABCD,BEG*cc<cr><lf>
$PASHR,VCR,...
$PASHD,#2,123456.00,ABCD,END*cc<cr><lf>

```

Each VCR message is inserted between a beginning (BEG) and end (END) marker (shown in bold characters in the example above). The marker header indicates for which antenna (or sensor) the VCR message that immediately follows refers to.

For example, a VCR message inserted between two “\$PASHD,#1,..” lines means the message is about antenna #1.

## Corrections Transport

In the rover configuration above, the corrections are routed via port B because the example described is based on the use of two Evaluation kits used side by side.

In real-life applications, the two boards will operate at different locations and so you should define another way of transporting the corrections, for example via modem or radio. On the moving base, you should therefore generate the corrections message to the port connected to the transmitting device, for example port I:

```

Observations and kinematic position:
$PASHS,ATM,RNX,I,ON,1,&SCN,204

```

```

Receiver and antenna names, other attributes:
$PASHS,ATM,ATR,I,ON

```

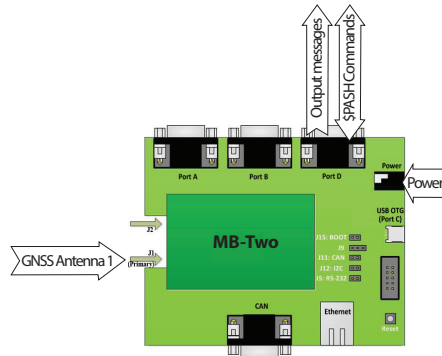
## Rover Computing RTX Position

---

Refer to Chapter *Trimble RTX Correction Service* on page 67.

# Base Configurations

## Static Base



Set the board through its port D as a static base:

- Unless already done, set the board to SOLO mode:  
**\$PASHS,SNS,SOL**
- Enter the base position. Example:  
**\$PASHS,POS,5542.00140500,N,03733.71603633,E,205.6450,PC1**
- Enter the antenna name. Example:  
**\$PASHS,ANP,OWN,TRM55971.00**
- Enter the base ID. Example:  
**\$PASHS,STI,0001**
- Ask the board to generate corrections. Among different possibilities, RTCM-3.2 (MSM) and ATOM are the most important ones.

If you want to serve third-party rovers, RTCM-3.2 (MSM) is the recommended format. As examples, set the following messages, assuming that the output port used is B and the output rate is 1 second (except for 1033):

GPS observations:  
**\$PASHS,RT3,1074,B,ON,1**

GLONASS observations:  
**\$PASHS,RT3,1084,B,ON,1**

Galileo observations:  
**\$PASHS,RT3,1094,B,ON,1**

QZSS observations:

**\$PASHS,RT3,1114,B,ON,1**

BDS observations:

**\$PASHS,RT3,1124,B,ON,1**

Static position:

**\$PASHS,RT3,1006,B,ON,1**

Antenna and receiver name:

**\$PASHS,RT3,1033,B,ON,31**

System parameters:

**\$PASHS,RT3,1013,B,ON,1**

ASCII message:

**\$PASHS,RT3,1029,B,ON,1**

GLONASS code phase bias:

**\$PASHS,RT3,1230,B,ON**

If you want to generate corrections for your own rovers, the standard ATOM format is recommended. As examples, set the following messages, still assuming that the output port used is B and the output rate is 1 second:

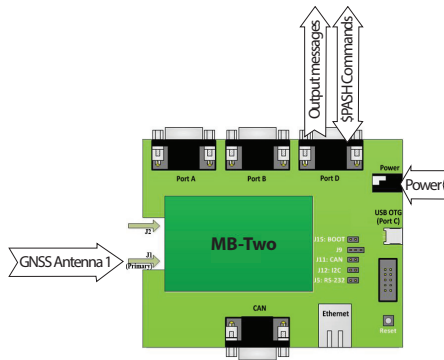
Observations and static position:

**\$PASHS,ATM,RNX,B,ON,1,&SCN,4**

Receiver and antenna names, other attributes:

**\$PASHS,ATM,ATR,B,ON,1**

## Moving Base



Set the board through its port D as a moving base:

- Unless already done, set the board to SOLO mode:  
**\$PASHS,SNS,SOL**
- Enter the base ID. Example:  
**\$PASHS,STI,0001**
- If you want to generate corrections for your own rovers, the standard ATOM format is recommended. As examples, set the following messages, still assuming that the output port used is B and the output rate is 1 second:

Observations and kinematic position:

**\$PASHS,ATM,RNX,B,ON,1,&SCN,204**

Receiver and antenna names, other attributes:

**\$PASHS,ATM,ATR,B,ON,1**

## Raw Data Recording

---

### In SOLO Mode

Raw data recording can take place using the standard RTCM3.2 or the proprietary ATOM format.

In all message examples given below, s1 designates the port ID and f2, the observation interval.

RTCM3.2:

```
$PASHS,RT3,1077,s1,ON,f2
$PASHS,RT3,1087,s1,ON,f2
$PASHS,RT3,1097,s1,ON,f2
$PASHS,RT3,1117,s1,ON,f2
$PASHS,RT3,1127,s1,ON,f2
$PASHS,RT3,1006,s1,ON
$PASHS,RT3,1033,s1,ON
$PASHS,RT3,1013,s1,ON
$PASHS,RT3,1029,s1,ON
$PASHS,RT3,1230,s1,ON
```

GPS ephemeris:

```
$PASHS,RT3,1019,s1,ON
```

GLONASS ephemeris:

```
$PASHS,RT3,1020,s1,ON
```

QZSS ephemeris:

```
$PASHS,RT3,1044,s1,ON
```

Galileo ephemeris:

```
$PASHS,RT3,1045,s1,ON
```

ATOM:

Full observations:

```
$PASHS,ATM,RNX,s1,ON,f2
```

Ephemeris, almanac, iono data:

```
$PASHS,ATM,NAV,s1,ON
```

Receiver, antenna names, etc.:

```
$PASHS,ATM,ATR,s1,ON
```



## In DUO Mode

You can record raw data separately for each sensor/antenna, using the RTCM3-2 or ATOM format, as explained in SOLO mode. Raw data may be routed on two distinct ports, one per sensor, or on the same port. In the latter case, it is more convenient to work with the ATOM format.

Full observations from primary sensor:

**\$PASHS,ATM,RNX,s1,ON,f2**

Full observations from secondary sensor:

**\$PASHS,2,ATM,RNX,s1,ON,f2**

Ephemeris, almanac, iono data:

**\$PASHS,ATM,NAV,s1,ON**

Receiver and antenna names and other attributes:

**\$PASHS,ATM,ATR,s1,ON**

## Adding Startup Protection to the Board

---

You can set up the board in such a way that it can only be run by an authorized user. This protection is called “startup protection”.

Through this protection, you can prevent anyone from using the board until the right password is provided. When the password is entered to remove the startup protection, the board will switch from the “Lock” to the “Unlock” state.

### Setting Up the Startup Protection

The implementation of the startup protection relies on the use of the following commands:

**\$PASHS,ATH,PWD**

**\$PASHS,LCK,MOD**

**\$PASHS,LCK,ON**

**\$PASHQ,LCK**

The startup protection is disabled by default. Follow the instructions below to activate the startup protection in the board:

- Define the password authorized users will need to remove the startup protection.

**\$PASHS,ATH,PWD,password**

(This password should be passed on to all authorized users. As the administrator, you can check which password was last defined using \$PASHQ,ATH.)

- Allow the board to be locked:

**\$PASHS,LCK,MOD,ON,password**

(This step requires re-typing the same password as the one you defined with \$PASHS,ATH,PWD, otherwise the command will be NAKed.)

The board will be effectively locked next time you power it up.

- While designing this feature, you may need to use the following command to read the current status of the startup protection:

**\$PASHQ,LCK**

The board response will be in the form:

\$PASHR,LCK,s1,d2\*cc

For more information, see *LCK: Lock Mode Status & Receiver Lock Status* on page 394.

## Allowing an Authorized User to Unlock the Board

When the board is only one part of the system you designed for your customer, you should make sure the system's user interface accommodates a mechanism allowing an authorized user to enter the password.

- When a password is entered from the system's user interface, make sure you pass on the following command to the board:

**\$PASHS,LCK,OFF,password**

- If the entered password is correct, the board will return:  
\$PASHR,ACK\*3D

This means the board has been unlocked successfully and the user can start using the system.

- If the entered password is incorrect, the board will return:  
\$PASHR,NAK\*30

This means the unlocking mechanism failed, keeping the board locked. From that point, you may create a new routine warning the user of unlock failure and you may also allow the user to submit a new password.

## Deactivating the Startup Protection

If you want to reverse to default operation (no startup protection), do the following:

- Run this command:  
**\$PASHS,LCK,MOD,OFF,password**

Note that the last password defined with \$PASHS,ATH,PWD must be entered in the command line, otherwise the command will be NAKed.





## Chapter 4. Trimble RTX Correction Service



### Introduction

---

Trimble RTX is a technology that leverages real-time data from a global tracking station network along with innovative positioning and compression algorithms to compute and relay satellite orbit, satellite clock, and other system adjustments to GNSS receivers.

This breakthrough technology provides satellite or IP delivered, high-accuracy GNSS positioning without the use of a traditional reference station-based infrastructure.

While standard autonomous GNSS positioning provides horizontal accuracies that are often more than 1 meter, Trimble RTX can provide horizontal accuracies of better than 4 cm (1.5 inches).

The MB-Two board supports four different Trimble RTX correction services, each of them offering a specific level of accuracy and a specific initialization time. Each of them is activated in the board through a specific firmware option:

- [C]: CenterPoint RTX (RMS accuracy < 4 cm)
- [9]: FieldPoint RTX (RMS accuracy < 20 cm)
- [P]: RangePoint RTX (RMS accuracy < 50 cm)
- [4]: ViewPoint RTX (RMS accuracy < 1 m)

### Computing RTX Position in MB-Two

---

The MB-Two can compute a centimeter-level position solution using Trimble RTX State Space Correcting Data. Data can enter the board either through an L-band (sky) data link or through NTRIP service via one of the board's communication ports (Internet Port).

Using the Trimble RTX correction service requires firmware option [C], [9], [P] or [4] to be installed on the board. In addition, to let the board use the L-band reception channel, firmware option [L] must be installed as well.

**Theory of operation:** Below are the commands required to enable MB-Two to make use of the Trimble RTX correction service:

- Define the source of RTX corrections (L-band or IP)

Selecting L-band:

**\$PASHS,RTX,SRC,LBN**

Or selecting IP:

**\$PASHS,RTX,SRC,IP**

Or automatic source selection:

**\$PASHS,RTX,SRC,AUT**

- If you want to use L-band reception:

Tracking all RTX satellites:

**\$PASHS,LBN,USE,RTX,ON**

Or tracking a particular satellite. Example:

**\$PASHS,LBN,USE,RTXAE,ON**

You may combine RTK solution (see *Rover Computing RTK Position on page 37*) and RTX solution computation. In that case, analyzing the GGA message will tell you whether the position solution is an RTX or RTK one:

- Position type: “4” for RTK or RTX fixed, “5” for RTK or RTX float
- Base station ID different in RTK and RTX

These commands are more conveniently handled using the Web Server (see section below).

## Implementing Trimble RTX using the MB-Two Web Server

---

- Run the Web Server (for more details, see *Running the Embedded Web Server on page 71*).
- Open the **Receiver** tab, then the **Position** tab.
- Select the **Sensors/Antennas Setup** section (click on the gray scroll-down list on the left to make that selection) and define the parameters relevant to the antenna you are using.

- Then select the **Rover Setup** section (still from the gray scroll-down list).

The screenshot shows the 'Rover Setup' configuration page. At the top, there are navigation tabs: 'Position' (highlighted in orange), 'I/Os', 'Radio', 'Network', and 'Satelli'. Below the tabs is a 'Rover Setup' header. The settings are organized into three sections:

- Rover Settings:** 'Processing Mode' is set to 'RTK'. 'Input Mode' has radio buttons for 'Automatic' (selected) and 'Manual'.
- Trimble RTX Settings:** 'Corrections Source' is set to 'L-Band'. 'Engine Mode' is set to 'ON'. 'Datum Transformation' is set to 'OFF'.
- Other Settings:** 'Primary GNSS system' is set to 'GPS'. 'Output Position Type' is set to 'RTK Position'. 'Rover Dynamics' is set to 'Adaptive'.

At the bottom of the form are two buttons: 'Configure' and 'Cancel'.

- In the **Rover Settings** sub-section, select “RTK” as the **Processing Mode**, and “Automatic” as the **Input Mode**.
- In the **RTX Settings** sub-section, select the source of RTX corrections (**Corrections Source= L-Band, NTRIP or Automatic**), the correction service used (**Engine Mode= ON or OFF**) and whether or not you need to convert the RTX computed position to another datum (**Datum Transformation= ON or OFF**). Without this transformation, the RTX position will be expressed in the ITRF2008 current epoch datum.
- In the **Other Settings** subsection, select “PPP Position” as the **Output Position Type**.
- Click on the **Configure** button to enter all the new parameters. MB-Two is now ready to operate with the Trimble RTX correction service.

For more information, refer to *Setting the MB-Two from the Web Server on page 71*.

## Trimble RTX Accuracy Figures

---

NOTE 1: Accuracy and TTFB specifications may be affected by atmospheric conditions, signal multipath, satellite geometry and corrections availability and quality. Vertical error is typically less than twice the horizontal error.

NOTE 2: Performance values assume a minimum of five satellites, following the procedures recommended in the product manual. High multipath areas, high PDOP values and periods of severe atmospheric conditions may degrade performance.

Service	Horizontal Accuracy (RMS)	Initialization Time	GNSS Frequencies
CenterPoint RTX	< 4 cm	< 30 min	L1+L2
FieldPoint RTX	< 20 cm	< 15 min	L1+L2
RangePoint RTX	< 50 cm	<5 min	L1+L2
ViewPoint RTX	< 1 m	<5 min	L1



# Chapter 5. Setting the MB-Two from the Web Server

The embedded Web Server offers the most convenient and fastest way of looking into both the configuration and operating status of the board.

## Running the Embedded Web Server

### MB-Two Connected to an Existing Network

Before you can run the MB-Two Web Server from your usual browser, you need to know the IP address of the board. The way you can read this address depends on whether the DHCP mode is active or not:

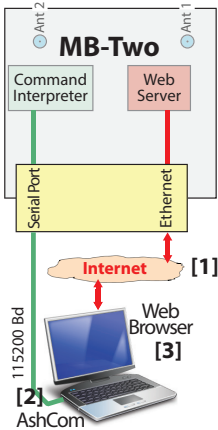
- **DHCP mode active:** By default, the MB-Two board is set up to operate in DHCP mode (dynamic IP address allocation), which means the IP address will be allocated by the local network, and not by you. In that case:
  - Connect the board's Evaluation kit to the local network through its Ethernet port, using an Internet cable between the RJ45 connector (J2 on the Evaluation kit) and the network (see [1]).
  - Make a serial connection between the board and your computer (default baud rate: 115200 Bd) (see [2]).
  - Power on the board and wait about 1 minute until the local network has assigned an IP address to the board.
  - On the computer, use AshCom (see *Using AshCom to Communicate with MB-Two on page 23*) to run the following query command:

**\$PASHQ,ETH**

You will get an answer of this type (if DHCP active):

```
$PASHR,ETH,<Ethernet Port>,<Ethernet Status>,<MAC Address>,<Current IP Address>,<DHP=1,ADD=<Static IP address when DHCP disabled>,MSK=<Subnetwork Mask>,GTW=<Gateway IP Address>,<DN1=<DNS1 IP Address>,DN2=<DNS 2 IP Address>,ACN=<(obsolete field)>*cc
```

**The board's IP address is provided in fourth position in the response line.** (For a complete description of the



\$PASHQ,ETH command, see *ETH: Ethernet Status and Parameters on page 361*.)

- Run a web browser on your computer (see [3]).
- Type the board's IP address in the web browser and press **Enter** on the keyboard. This opens the Web Server Home page (see next section).

NOTE 1: Being a dynamic IP address (DHCP mode on), you may need to read the board's IP address. Use command \$PASHQ,ETH for this purpose.

NOTE 2: The board queries the DHCP server every 60 seconds. If there is no DHCP server answering the board in the next 10 seconds, there is an attempt to allocate IP address "169.254.1.0" to the board. This address will be chosen if no address conflict is detected. Otherwise, IP address "169.254.1.1", "169.254.1.2", etc. will be tested successively until one is found that does not cause any IP address conflict (IP addresses tested up to "169.254.1.255").

- **Using a static IP address (DHCP mode off):** You will need to disable the DHCP mode by yourself and define each and every of the required board's network parameters, including the board IP address. In that case:

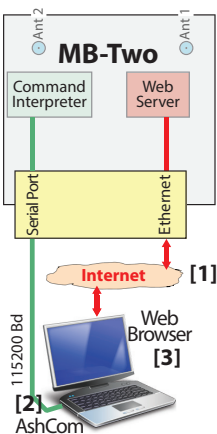
- Connect the board's Evaluation kit to the local network through its Ethernet port, using an Internet cable between the RJ45 connector (J2 on the Evaluation kit) and the network (see [1]).
- Make a serial connection between the board and your computer (default baud rate: 115200 Bd) (see [2]).
- Power on the board.
- On the computer, use AshCom to run the set command below. This command will disable the DHCP mode (i.e. "DHP,0" in the command line below) and define the different addresses required, including the board's IP address (in second position in the command line):

```
$PASHS,ETH,PAR,DHP,0,ADD,<IP address>,MSK,
<Subnetwork Mask>,GTW,<Gateway IP Address>,DN1,
<DNS 1 IP Address>,DN2,<DNS 2 IP Address>
```

- If the command is successful, you will get this response line:

```
$PASHR,ACK*3D
```

You may check your settings by running the command \$PASHQ,ETH (as described above when DHCP active).

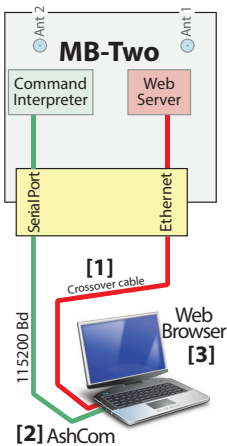


- Run a web browser on your computer ([3]).
- Type the board's IP address in the web browser and press **Enter** on the keyboard. This opens the Web Server Home page (see next section).

NOTE 1: If you are not familiar with network issues, you may seek advice from your local network manager.

NOTE 2: The Ethernet port is powered on by default. If for some reason, it was turned off, use \$PASHS,ETH,ON to turn it back on (see *ETH,ON: Powering On the Ethernet Port on page 199*).

## MB-Two Not Connected to any Network



When the board is not otherwise using its Ethernet port and there is no network connection available in the operating environment of the board, you should connect your computer directly to the board using a crossover Ethernet cable ([1]). Only through this connection will you be able to use the Web Server to control the board.

In a crossover cable, the pinout is inverted at one end of the cable. The crossover cable is not provided but is widely available from computer supply stores or on line, and is usually fitted with RJ45 connectors.

In this case, make the following settings BEFORE physically connecting the computer to the board through the crossover cable:

- On the board, disable the DHCP mode and define an arbitrary static IP address and a subnetwork mask for the receiver.
- On the computer, change the network configuration for an exclusive TCP/IP connection with the board.

Before changing the network configuration of the computer, it is advisable to write down all the current settings so that you can easily reverse to the previous network configuration when you are done with communicating with the board.

Below are more detailed instructions to complete the above two steps. Use AshCom ([2]) to send commands through a board's serial port:

1. Read the current settings of the Ethernet port. Write them all down so that later you can easily reverse to these settings.

**\$PASHQ,ETH**

Example of receiver response:

```
$PASHR,ETH,I,ON,00:09:66:00:10:a0,10.20.2.123,DHP=1,ADD=192.168.0.1,MSK=255.255.255.0,GTW=255.255.255.255,DN1=255.255.255.255,DN2=255.255.255.255,ACN=1*53
```

Should the Ethernet port be off (2nd parameter in the above response line is “OFF” instead of “ON”), use the following command to turn it back on:

```
$PASHS,ETH,ON
```

Receiver response should be the following if the set command is successful:  
\$PASHR,ACK\*3D

2. Disable the DHCP mode and define an arbitrary IP address for the board:

```
$PASHS,ETH,PAR,DHP,0,ADD,10.20.2.10,MSK,255.255.255.0,GTW,10.20.2.1  
$PASHR,ACK*3D
```

Where:

“10.20.2.10” is the arbitrary IP address assigned to the board.

“255.255.255.0” is the arbitrary, but also mandatory, subnetwork mask.

“10.20.2.1” is the arbitrary address for the gateway that will be assigned to the computer.

3. On the computer, change the network configuration in order to assign a static IP address to the computer. It’s a good practice to save or write down the parameters of the current network configuration before changing them. Unless already done, disable the DHCP mode and enter a static IP address for the computer (different from that assigned to the board, e.g. 10.20.2.2). Enter the same subnetwork mask and gateway as those you previously assigned to the board through the \$PASHS,ETH,PAR command. If you have problems completing this step, you may seek advice from your company’s network expert.
4. Connect the crossover cable (**[1]**) between the board and the computer.
5. Open a web browser (**[3]**) on the computer.
6. Type the board IP address in the address box. This launches the Web Server on the board.

## Managing Access Rights

---

The Web Server administrator may implement two types of access rights to protect the Web Server from unauthorized access and use:

- **Security enabled:** Only the owner or authorized user can access the Web Server. All changes are allowed. When starting the Web Server, a login and password is required first.

NOTE 1: The default credentials for a connection are the following:

- Login: admin
- Password: board serial number (from GNSS firmware v4.11)

(CAUTION: for boards manufactured before 3.2X, the default password is “changeme”).

NOTE 2: For boards received after January 1st, 2020, with GNSS firmware v4.11, you will have to change the default password the first time you connect to the board’s Web Server (see details in next section).

- **Anonymous access enabled:** This is an intermediate level of security in which anyone can access the Web Server, but the number of authorized actions are limited, in compliance with what the administrator has decided. There are four different levels of protection in this case:
  1. Read-only mode: Visitors are just allowed to view the different web pages. No changes are allowed, no files can be deleted or downloaded.
  2. Read only mode, but visitors are allowed to download files, where prompted.
  3. Read only mode, but visitors are allowed to delete files, where prompted.
  4. Read only mode, but visitors are allowed to download or delete files, where prompted.

With this protection active, the registered users and administrator will have to go to **Receiver > Security** to enter their login and password, thus recovering full access to the Web Server.


## Home Page


---

### First-Time Login

The first time you run the Web Server, you need to:

- Enter the login and default password (the default password is the board's serial number).

To read the serial number, click on  and then read that number in the table on the left, in the second row.

- Click on the **Log In** button. This opens a connection to the Web Server and you are right away requested to do the following:
  - Change the password you've just entered. Click on  to read all the instructions you need to know to define a valid password. As explained in that tip box, your new password should have at least medium strength to be valid. The password's current strength is reported underneath as you type in the new password.
  - Confirm your new password by entering it a second time in the field underneath.
- Click **Validate** to save the new password. This opens the Web Server's Home tab (described in the section below). End of first connection procedure.

### Second and Next Login Procedures

Access to the Web Server is restricted. If you are the owner of the board or an authorized user, enter the login and password to gain access to the Home page.



The board you are trying to connect to may also be startup-protected in which case you should also enter the required password to get through that protection (more on this in the **Security** on-line Help, **Startup Protection** section).

When protection is configured to allow anonymous access, then the Home page is opened directly when launching the Web Server, without the need for entering a login and password. The Web Server will just view the configuration of the board. You won't be allowed to make any changes in that case. You may however be allowed to download or delete certain files if the anonymous access mode was configured to let you do so.




Still in anonymous access mode, the administrator or registered users can recover full access to the board configuration (so they can make all the changes they want), by going to **Receiver** > **Security** and entering the login and password, thus escaping from the anonymous access mode.




In its upper-left corner, the **Home** page shows a blue frame recalling the name and brand of the board you can control through the Web Server. Elsewhere on this page, you can read various information about the board (see below).

At the top of the page is a banner (a gray area), **which will stay permanently visible whatever the data displayed underneath**. From left to right, the banner shows the following:

- If the board is startup-protected:
  -  : A close padlock icon is shown before the password relevant to this protection is entered.
  -  : An open padlock is shown after the correct password has been entered.
- The company logo. Clicking on this logo will take you directly to the company's website.
- Three menu buttons:
  - **Home** menu: Opens the Home page, the one you see when you start the Web Server.
  - **Receiver** menu: Provides access to all the pages allowing to set and monitor the board.
  - **Support** menu: Takes you directly to the Support page on our website (in a new window).
- The **Board Status banner**: This banner can be shown or hidden by clicking on the triangle button located on the right (see below). When shown, the Board Status banner provides the following information, from left to right, and for each antenna used if two are used:
  - 1st column: Rover mode or base mode status, position type, base station ID.
  - 2nd column: Number of satellites tracked, number of satellites used, PDOP.
  - 3rd column: Ethernet status.
  - 4th column: Recording status, site name, memory used and available space.
  - 5th column: Embedded NTRIP caster status.
  - Current UTC date and time.

When the Board Status banner is hidden, this area just shows the UTC date and time (in larger, bold characters).

-   : A button allowing you respectively to show or hide the Board Status banner.
-  : A button providing access to the on-line help.

-  : A button that opens a window listing information on the board and all installed firmware options. Click on the red-cross button to close that window.
-  : Click on this button to choose the Web Server interface language.
-  : This button appears only after the board has detected errors and set one or more alarm messages. Click on this button to read and acknowledge the alarm message(s).

As a general rule, the configuration functions are shown on the left of the screen (occupying a third of the page width) and the status/monitoring functions will be shown on the right, occupying the remaining two thirds of the page. An orange vertical line separates the two areas.

When you change some parameters in the configuration area, these will be effective only after you have clicked on the **Configure** button at the bottom of the page. Following this action, the status area will be updated accordingly, instantaneously or after some delay depending on the nature of the parameters you are changing.

The board identification information includes:

- Board type
- System name
- Serial number
- Ethernet MAC address
- Firmware version
- Firmware date
- **More**: Clicking on this link will open a table listing advanced information on the board, including those shown on the home page. The table is the same one you get by typing on the exclamation mark in the top banner.
- In the lower part of the Home page, you can read the list of installed firmware options.



## Receiver Page

The **Receiver** menu provides access to nine tabs:

1. Position
2. I/Os
3. Radio
4. Network
5. Satellites
6. Memory
7. Configuration
8. Security
9. Embedded NTRIP Caster

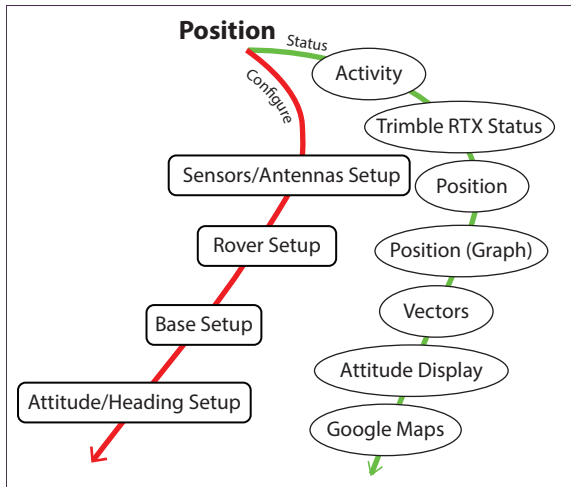
Each tab provides access to two categories of functions:

- The configure functions
- The status functions

The configure functions are accessible from the tab located on the left, and status functions from the tab located on the right. Each of these functions is detailed below.

### Position

You may either configure the board as a rover, as a base, or as a device computing attitude/heading measurements. You only need to complete the tab corresponding to the function you are interested in. You are however required to complete the **Antennas Setup** tab in all cases.



## Sensors/Antennas Setup

- **Multiple-Sensor Mode:** Specify the number of independent sensors used in the board (one or two):
  - **Single Antenna (automatic):** One sensor (one GNSS antenna input used). The detected antenna is used. If two antennas are detected, the primary antenna is used.
  - **Single Antenna (primary):** One sensor. The primary antenna (J1 input) is used.
  - **Single Antenna (secondary):** One sensor. The secondary antenna (J2 input) is used.
  - **Two Antennas (L1/L2/L-Band + L1):** Two sensors are used. The primary antenna may be up to L1/L2/L-Band and the secondary antenna may be just L1.
  - **Two Antennas (L1/L2 + L1/L2):** Two sensors are used. Both the primary and secondary antennas may be up to L1/L2.

- **Reference Position:** Tell the board for which point on the antenna the position will be calculated:

- **L1 Phase Center**
- **Antenna Reference Point (ARP)**
- **Ground Mark (GM)**

This selection will also apply to the second antenna, if any.

- For each antenna used (primary and secondary), specify the following parameters:
  - **Manufacturer:** Select the brand of GNSS antenna used. Choose “**UNKNOWN**” if the model used is unknown or not listed.
  - **Antenna Name:** Once you have selected the antenna manufacturer, use this drop-down list to select the model of antenna used.
  - **RINEX Name:** When you select an antenna model, this drop-down list field is automatically updated to select the equivalent RINEX name for this antenna model. For some antenna models, more than one RINEX name is possible so you should choose the one you want to use. If you chose **UNKNOWN** for the antenna, the field is made empty so you may freely type in a RINEX name for this antenna.
  - **Method:** Choose the point on the antenna used to make the height measurement. Depending on the selected antenna model, you may have to choose between

**Bottom of antenna mount, Bottom of edge, Line at edge of antenna** (vertical measurement), **Top of notch**, (slant measurement), etc.

- **Height:** Enter the height measurement, in meters, between the ground mark and the point on the antenna corresponding to the selected measurement method. This measurement is used to provide the correct vertical component of the computed position when **Reference Position= Ground Mark (GM)**.
- **Antenna Name** (of virtual antenna): Choose “**OFF**” if you don’t want to define a virtual antenna, otherwise choose the type of virtual antenna you wish to emulate (ADVNULLANTENNA or GPPNULLLANTENNA).  
Only the primary antenna can be defined as a virtual antenna. For more information on virtual antennas, see “*Virtual Antenna*” Concept on page 513.

When you are done, click on **Configure** to save all your changes and make their use effective in the board.

## Rover Setup

Use this menu to set up the board as a rover.

### RTK Settings:

- **Processing Mode:** Select the type of output the board must deliver:
  - **OFF:** Position computed by the receiver alone (no corrections used, metric accuracy) or using S-DGPS (submetric).
  - **RTK:** Position computed with centimeter-level accuracy using the corrections received.
  - **Hot Standby RTK:** Same as **RTK** above, but a second position, and possibly a third position, will be computed in the background, also with centimeter-level accuracy, each process using a different set of corrections. Only the best of the two or three computed positions will be made available at the output.
  - **RTK + Relative RTK:** The board will deliver both a centimeter-accurate position for the rover and a centimeter-accurate vector connecting an external moving base to the rover.
  - **Hot Standby RTK + Relative RTK:** The board will deliver the best centimeter-accurate position out of the two computed, and a centimeter-accurate vector connecting an external moving base to the rover.

- **Only Relative RTK:** The board will deliver a centimeter-accurate vector connecting the antenna of an external moving base to its single antenna, or connecting the board's secondary antenna to the board's primary antenna (two antennas are used here, i.e. "internal" relative RTK).
- **Dual RTK** (two antennas used): The board will deliver an RTK position for each of the antennas used.
- **Dual Relative RTK** (two antennas used): The board will deliver two independent, centimeter-accurate vectors connecting an external moving base to the two antennas used by the rover.
- **Input Mode:** Choose which ports will forward incoming differential corrections to the board's core.
  - **Automatic:** The board will recognize automatically what the corrections are, and which port or ports they come from.
  - **Manual:** Specify the port (in RTK and Only Relative RTK) or ports (all other processing modes) through which corrections enter the board. For each set of corrections, when applicable, you may choose a specific port, or you may let the board find by itself by selecting "**Auto**".

If the board is used with two antennas and is configured to operate in "internal" Relative RTK, then an additional option is available for use in the **Manual** field. This option is named "**Internal**" and when selected, allows you to route the internal corrections message from the board's "base" sensor to the board's "rover" sensor.

#### PPP Settings:

- Enable or disable (**OFF**) the use of PPP. The PPP service that you can possibly use is the following:
  - **Trimble RTX:** Corrections can be received via either L-Band or NTRIP and you can also choose which RTX satellite the receiver should work with specifically. Using Trimble RTX is possible only if one (or more) RTX option has been installed in the receiver.
- **Corrections Source:** Choose the transmission channel that will be delivering RTX corrections to the board:
  - **Automatic:** The board will use RTX corrections received through whatever transmission channel is currently active.

- **L-Band:** The board will use RTX corrections received from satellites.
- **NTRIP:** The board will use RTX corrections delivered by a service provider over the Internet.
- **OFF:** No RTX corrections will be used.
- **SV Name:** (Visible only if you have ticked the **Trimble RTX** box and **Corrections Source** is either **Automatic** or **L-Band**). Specify the name of the satellite (i.e. RTX...) from which to receive RTX corrections. By choosing **Automatic**, you will let the receiver decide which satellite to use.
- **Datum Transformation (ON or OFF):** If you choose “ON”, please indicate the names of the datum and tectonic plate in which to express the coordinates of the computed position. If you don’t know which datum and/or tectonic plate you are on, choose **Automatic**.  
If you choose “OFF” the computed position will be expressed in the ITRF2014 current epoch datum.

#### Other Settings:

- **Primary GNSS System:** Choose a primary GNSS system (GPS, GLONASS, BeiDou or Galileo). This parameter impacts the choice of datum in which the position computed by the board will be expressed.
- **Output Position Type:** This choice impacts the level of accuracy you allow the board to deliver:
  - **RTK Position:** Centimeter level
  - **PPP Position:** Centimeter level after convergence time (Trimble RTX)
  - **(RTCM) Differential Position:** Decimeter level
  - **SBAS Differential Position:** Decimeter level
  - **Standalone Position:** Meter level
- **Rover Dynamics:** Choose the dynamic model that best suits the board you are setting (walking, ship, aircraft, etc.), depending on the application in which it is used. Choosing “**Adaptive**” means you allow the board to adapt itself to its own dynamics.  
If you choose **User-defined**, please enter your values for the following parameters: **Max. horizontal velocity** (in m/s), **Max. horizontal acceleration** (in m/s/s), **Max. vertical velocity** (in m/s), **Max. vertical acceleration** (in m/s/s).

When you are done, click on **Configure** to save all your changes and make their use effective in the board.

## Base Setup

Use this menu to set up the board as a base and define the reference position.

### Virtual Antenna:

- **Antenna Name:** Choose “**OFF**” if you don’t want to define a virtual antenna for the base antenna, otherwise choose the type of virtual antenna you wish to emulate.

### (Primary + Secondary) Sensor Settings:

If two antennas are used, the Web Server may let you define two bases (one in **Primary Sensor Settings** and the other in **Secondary Sensor Settings**).

If one antenna is used, only one base may be defined.

To define a base, enter the following set of parameters:

- **Station ID:** Enter an identification number for the base.
- **Position Type:** Choose between **Static** and **Moving**.

**Latitude, Longitude, Height, Height Reference:** If the base you are configuring is a static one, enter the coordinates of the base location (thus defining the reference position). These should be expressed in the datum of the primary GNSS system currently used. Use the **Height Reference** field to specify which location on the antenna the entered coordinates refer to (L1 phase center, ARP or ground mark).

Clicking on the **Get Current Position** button allows you to save the last position computed by the receiver as the reference position. The three above fields are then completed with these last computed coordinates.

Clicking on the **Import Position** button allows you to import a file containing base coordinates. The file should be a text file with the “csv” extension in which the comma is used as field separator. In the csv file, the coordinates should have been formatted this way: “P,Lat,Lon,h,Code” where:

- P: Point name
- Lat: WGS84 latitude (signed value, in degrees, with decimal places)
- Lon: WGS84 longitude (signed value, in degrees, with decimal places)
- Code: When importing the file, the content of this field will be ignored but remember this field needs to be present in the csv file (just insert a comma after the longitude).

Example: CARQ,47.29897,-1.50905,88.093,

After importing the csv file, the imported coordinates will appear in a table. Select the corresponding row in the table to assign this position to the sensor. Be aware imported coordinates are always seen as those of a ground mark (**Height Reference** field automatically set to **Ground Mark (GM)**) but you can always correct this field so it accurately describes what the imported coordinates really are.

It does not make sense to enter coordinates if the base you are configuring is a moving one.

### Other Settings

- **Primary GNSS System:** Choose a primary GNSS System (GPS, GLONASS, BeiDou or Galileo). This parameter impacts the choice of datum in which the position computed by the receiver will be expressed.

When you are done, click on **Configure** to save all your changes and make their use effective in the board.

### Attitude/Heading Setup

**NOTICE:** For attitude computation, you not only need to configure the MB-Two delivering attitude measurements, but also the external sensors involved in the process. These sensors should be configured separately.

- **Mode:** Choose whether you want the board to perform heading computation, attitude computation, or none of the two (Off).

If you chose to use two antennas (see **Multiple-Sensor Mode** field), then this field is preset to **“Heading”**. If you chose to use only one, then it's set to **“OFF”**.


- **Input** (if **Mode** other than **OFF**): Designates the source of corrections used to determine heading or attitude.

If **Heading** is selected:

- Select **“Internal”** to let the board compute heading on its own, that is, moving base data are generated for the primary antenna and processed together with rover data coming from the secondary antenna.
- Select **“External”** if moving base data are provided by another board (MB-Two using one antenna) connected to this board through one of its communication ports. (Option [E] is required on the board delivering external heading measurements.)

Then indicate the port on which the external data are applied to the board.

And if these data are coming from another MB-Two board configured with two antennas, indicate the antenna for which the corrections are computed (**Primary Antenna** or **Secondary Antenna**).

If **Attitude** is selected: Same as heading, plus you are allowed to add a correction stream using the  button.

- **Vector Description:**

In heading mode, you have first to indicate the nature of the vector, using the **Length Type** field. If the vector length is liable to vary over time, even in small proportions, choose “**Changing (Flex)**” and so there is no possibility for you to ask for auto-calibration in this case (as this would make no sense). Conversely, if the vector results from a setup making it strictly constant in length over time, choose “**Fixed**” and then decide on whether auto-calibration should be run or not (see below).

In attitude, just make a choice for auto-calibration and if appropriate, enter the vector lengths (see below).

- **Auto Calibration:** This setting is always available in attitude and heading modes, only if **Length Type**= “**Fixed**”. When enabling this function, you ask the board to determine the length of the vector(s) by itself, through auto-calibration. If you disable this function, you have to provide this or these lengths. See below.
- **Vector Length:** This setting is available only if **Auto Calibration** has been cleared. In heading mode, enter the known length of the vector. In attitude mode, enter the known length of each of the two vectors (**First Vector Length, Second Vector Length**).

- **Settings:**

You need to enter two offsets in heading mode, three in attitude mode.

Heading:

- **Azimuth Offset:** Enter the azimuth offset angle, in degrees (possible range:  $\pm 180^\circ$ ; default:  $0^\circ$ ). See *Heading-Roll-Pitch Offsets on page 29* for more information.
- **Elevation Offset:** Enter the elevation offset angle, in degrees (possible range:  $\pm 90^\circ$ ; default:  $0^\circ$ ). See *Heading-Roll-Pitch Offsets on page 29* for more information.

Attitude:



- **Heading Offset:** Enter the heading offset angle, in degrees (possible range:  $\pm 180^\circ$ ; default:  $0^\circ$ ). See *Heading-Roll-Pitch Offsets on page 29* for more information.
- **Pitch Offset:** Enter the pitch offset angle, in degrees (possible range:  $\pm 90^\circ$ ; default:  $0^\circ$ ). See *Heading-Roll-Pitch Offsets on page 29* for more information.
- **Roll Offset:** Enter the roll offset angle, in degrees (possible range:  $\pm 90^\circ$ ; default:  $0^\circ$ ). See *Heading-Roll-Pitch Offsets on page 29* for more information.
- **Max. Baseline Elevation:** Enter the maximum foreseeable baseline elevation angle, due to vehicle swaying, between the end of calibration and the time when valid heading/attitude data are delivered. A value expressed in degrees (permitted range:  $0-45^\circ$ ).  
The greater the elevation, the longer the time required for the board to deliver valid heading/attitude data after the calibration phase is complete.  
So if you want to reduce this wait time, make sure the swaying of the vehicle will be kept at a minimum between the end of calibration and the effective delivery of valid heading/attitude data, and define this parameter accordingly, i.e. as low as possible (default value:  $15^\circ$ ). When testing this value, the board will take into account the entered value of elevation offset as well.
- **Baseline Tolerance:** Enter the tolerated variations of the baseline length around its nominal value, in meters (default: 0.02 m; Range 0 to 1.0 m).

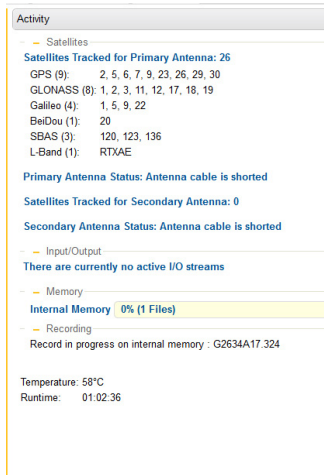
When you are done, click on **Configure** to save all your changes and make their use effective in the board.

### Activity

This tab shows the operation status of the board in real time:

- The list of satellites currently tracked by the primary antenna. Satellite numbers are sorted by GNSS constellation.
- The status of the secondary antenna and the list of satellites currently tracked by this antenna. Satellite numbers are sorted by GNSS constellation.
- The list of input and output messages (port, type, rate, etc.).
- The memory status and data recording status.
- Die temperature.

- The time elapsed since the board was last turned on (**Runtime**).



### Trimble RTX Status

- L-Band option (installed, or blank for not installed)
- Name of RTX option installed, if any
- Corrections Source (Automatic,..)
- Engine Mode (CenterPoint, FieldPoint, RangePoint or ViewPoint, depending on the RTX option installed)
- Datum Transformation (OFF, ON)
- Datum Name
- Plate Name
- SVs for tracking
- Antenna (primary, secondary or OFF)
- Antenna Status (Connected, not connected, shorted or unknown)
- Satellites Tracked
- Satellite Used
- Elevation [Deg.]
- Azimuth [Deg.]
- C/No [dBHz]
- Cumulative Tracking Time [Sec.]
- Trimble RTX position
- Trimble RTX Source ID
- Age of Corrections [Sec.]

See example below.

Trimble RTX Status	
L-Band Option	Installed
CenterPoint RTX Option	Installed
Corrections Source	Automatic
Engine Mode	CenterPoint RTX
Datum Transformation	OFF
Datum Name	ITRF2014 (current epoch)
Plate Name	
SVS for tracking	RTXEA,RTXAP,RTXWN,RTXEN,RTXCH,RTXSA,RTXAE
Antenna	Primary Antenna
Antenna Status	Antenna cable is shorted
Satellites Tracked	RTXAE
Satellite Used	RTXAE
Elevation [Deg.]	30.00
Azimuth [Deg.]	144.00
C/No [dBHz]	41
Cumulative Tracking Time [Sec.]	288
Trimble RTX Position	Computed
Trimble RTX Source ID	107
Age of Corrections [Sec.]	5

## Position

This tab displays the following information:

- Time information (GPS time, inter-constellation offsets)
- Clock information
- Current coordinates of primary antenna position, type of position solution, precise point on antenna for which position is computed, satellites used to compute the position.
- If two antennas are used, current coordinates of secondary antenna position, type of position solution, satellites used to compute the position.

Still with two antennas used, using the secondary tab (**All Antennas/Primary Antenna/Secondary Antenna**), you may view additional information about the position solution of each antenna (age and type of corrections, velocity, PDOP, error estimates, etc.).

This information is shown by default for the primary antenna when only this antenna is connected to the board.

Position ▼

**Time:**  
 Primary GNSS System: GPS  
 Week: 1862  
 Seconds: 141247

**Clock:**  
 Clock steering: Enable  
 External Clock: Disable  
 Offset: -65063.055 [m]  
 Drift: -59.468 [m/s]  
 TDOP: 0.6

All Antennas ▼

**Primary Antenna:**  
**Position:**  
 Lat: 47° 17' 56.79240" N  
 Lon: 1° 30' 33.15240" W  
 Hgt: 73.818 [m]  
 Type: Autonomous  
 Reference: L1 Phase Center  
 Datum Name: WGS-84

**Satellites Used: 22**  
 GPS (11): 1, 3, 4, 8, 11, 14, 19, 22, 27, 28, 32  
 GLONASS (7): 6, 7, 8, 14, 21, 22, 23  
 Galileo (3): 11, 12, 19  
 BeiDou (1): 14

**Secondary Antenna:**  
**Position:**  
 Lat: 47° 17' 56.81040" N  
 Lon: 1° 30' 33.17040" W  
 Hgt: 73.668 [m]  
 Type: Autonomous  
 Reference: L1 Phase Center  
 Datum Name: WGS-84

**Satellites Used: 21**  
 GPS (11): 1, 3, 4, 8, 11, 14, 19, 22, 27, 28, 32  
 GLONASS (7): 6, 7, 8, 14, 21, 22, 23  
 Galileo (3): 11, 12, 19

## Position (Graph)

This tab allows you to visualize any of the following parameters, all relevant to the position solution computed for the primary or secondary antenna:

- **Height:** Height coordinate vs. time
- **East:** East coordinate vs. time
- **North:** North coordinate vs. time
- **East/North:** East/North coordinates (“position cloud” diagram)
- **Horizontal:** Horizontal deviation of computed position compared to a reference (or first computed) position, vs. time
- **PDOP/#SVs:** PDOP value and number of satellites received vs. time
- **Age of corrections:** Age of corrections, in seconds, vs. time.

Each of these parameters is shown as a function of time (except for **East/North**; see above). The time axis is defined as follows:

- If you choose “**High-Rate Positions**”, the visualized parameter being a raw, non-smoothed one (with 10-Hz update rate), the X axis represents the last 15 minutes of operation elapsed.
- If you choose “**10-Second Positions**”, the visualized parameter being smoothed over 10 seconds, the X axis represents the last 24 hours of operation elapsed.

NOTE: For this particular viewing function, the Web Server may ask you to install/activate Adobe Flash Player if not installed yet on your computer, and/or to activate it if the web browser you are using currently blocks the use of this application.

## Vectors

This tab displays the following information:

- The results of the heading/attitude computation (solution status, heading angle, pitch angle, roll angle).
- The three components of each vector involved, its modulus (length), orientation and direction.
- The base ID attached to the origin of each vector.
- The three components of error estimates (in meters) for each vector.

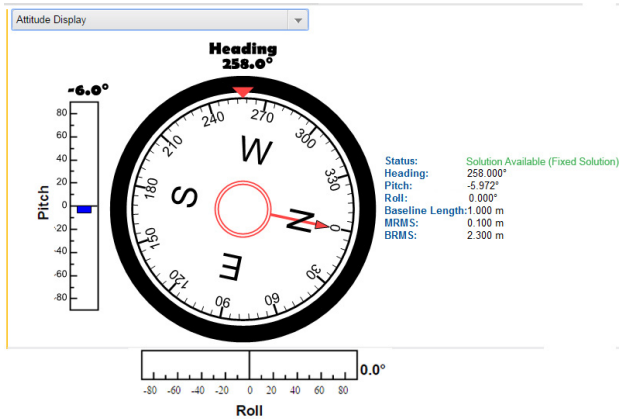
## Attitude Display

Depending on whether option [E] (Attitude) is missing or installed, this tab will be labeled respectively **Heading Display** or **Attitude Display**.

The tab provides a graphic representation of the heading, roll and pitch, as currently computed by the board:

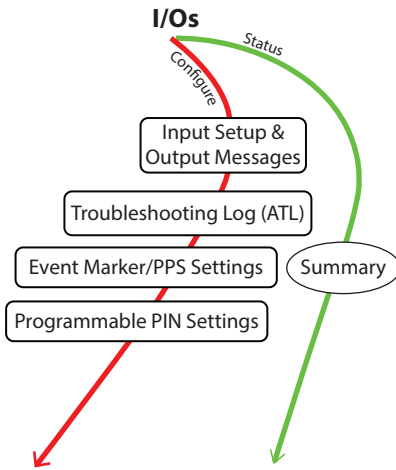
- The compass shows the heading angle, the horizontal bar graph located underneath shows the roll angle and the vertical bar graph on the left shows the pitch angle.

When heading, not attitude, is computed, the horizontal bar graph is not used. In this case, the vertical bar graph shows either the pitch or roll angle, depending on the orientation of the antennas on the vehicle.



- When no heading or attitude is computed, a large red cross is shown over the compass. The compass reading is static at 0°. The “**Heading not available**” status message appears in red.
- In Calibration mode, as long as measurement ambiguities are not solved, you can see a large red cross over a changing compass reading. The “**Under Calibration (Float Solution)**” status message appears in red.  
After measurement ambiguities are solved, you can see a large cross (this time orange in color), over a changing compass reading. The “**Under Calibration (Fixed Solution)**” status message appears in red.
- After calibration is complete and a valid value of heading is computed, the compass shows the heading angle and the heading status is either “**Solution Available (Float solution)**”, in orange characters, or “**Solution Available (Fixed solution)**”, in green characters.

## I/Os



### Input Setup & Output Messages

This function allows you to program all kinds of NMEA, NMEA 2000, ATOM, CMR, RTCM or GSOFF messages on the board's different communication ports. The complete list of programmed messages can be seen at all times on the I/O Status pane on the right.

When you are done, don't forget to click on the **Configure Setup** button to save your input setup. Likewise, don't forget to click on the **Configure Output** button to save all your output messages.

**List of Available Ports.** The following ports can be used to output data:

- **Serial** (three ports: A, B and D)
- **USB serial** (one port)
- **TCP/UDP xxxx** (two IP ports): Each of these ports is named after the protocol and IP port number you will have selected for it (e.g. "TCP 8886" will designate an IP port that uses the TCP protocol and has the number "8886").  
**Warning:** These two ports will accept \$PASH or DCOL commands only after the default password giving access to the Web Server has been changed. None of these commands will otherwise be accepted on these ports.
- **NTRIP/DirectIP x** (two IP ports): Each of these IP ports may be used in two different ways: for a Direct IP connection, or for an NTRIP connection.

The NTRIP connection type includes two special cases of use, which are IP connection to a Trimble RTX service and IP connection to the receiver's embedded NTRIP Caster.

- **Memory:** One port through which data can be routed to the selected memory, to be stored as a G-file recorded manually (i.e. out of any sessions).
- **CAN Bus** (one port).

**List of Available Data Messages.** Different message formats exist. For each of these formats, the lists of available message types are provided below:

- **ATOM:**
  - RNX (GNSS raw measurements)
  - PVT (Positioning results)
  - ATR (Receiver attributes)
  - NAV (GNSS navigation data)
  - DAT (Raw GNSS data)
  - ANG (Attitude)
  - LIN (Baseline)
  - VEC (Vectors)
  - OCC (Site occupation information)
  - PTT (PPS time tag)
  - TTT (External event time tag)

(The nature of the data in each of these messages - as recalled above between brackets - is shown in a tip box if you keep the mouse pointer on a message name after opening the drop-down list attached to the **Type** field).

NOTE 1: RNX messages may be formatted in different ways. You will choose the format that describes best the way your board is used, and in line with the specific requirements of your application:

- Static Base (RNX 4)
- Static Base, Compact Format (RNX 100)
- Static Base, Super Compact Format (RNX 101)
- Moving Base (RNX 204)
- Raw Data (PP) (RNX 0)

NOTE 2: When recording ATOM messages in memory (saved as a G-file), you may choose different "packages" of ATOM messages, as described below:

- **Standard:** ATR + NAV + RNX-0 (<Selected rate>) + OCC



- **Extended:** PVT (<Selected rate>) + ATR + NAV + DAT + RNX-0 (<Selected rate>) + STA (<Selected rate>) + ANG (<Selected rate>) + OCC
- **Customized:** You choose by yourself which of the messages (ATOM or other) you would like to record in memory.

NOTE 3: Some ATOM messages require an output rate, some others don't.

- **RTCM:** RTCM messages are available in three different versions: 2.3, 3.0/3.1 and 3.2.

After you have selected an RTCM version, you will be able to build directly the standard content of your RTCM messages by clicking on the **Apply in table below** button. Below are the preset lists of message types for the three RTCM versions (but remember you will always be allowed to customize these lists once loaded in the table):

- RTCM 2.3: Message types 1, 3, 31 and 32. The delivered data are assumed to be those from a DGNSS base.

- RTCM 3.0/3.1: Message types 1006, 1033, 1004 and 1012.

- RTCM 3.2:

Message types 1006, 1033, 1071, 1081, 1091, 1101, 1111, 1121 and 1230 (board operating as a DGNSS base).

Message types 1006 (every 7 sec), 1033, 1074, 1084, 1094, 1104, 1114, 1124 and 1230 (board operating as a Static RTK base).

Message types 1006 (at 1 Hz), 1033, 1074, 1084, 1094, 1104, 1114, 1124 and 1230 (board operating as a Moving RTK base).

Message types 1006, 1033, 1077, 1087, 1097, 1107, 1117, 1127, 1230, 1013, 1019, 1020, 1042, 1044 and 1046 (board operating as a raw data collector for post-processing).

The complete list of available RTCM message types is the following: 1, 3, 9, 16, 18-24, 31, 32, 34, 36, 1001-1013, 1019, 1020, 1029, 1033, 1042, 1044, 1045, 1046, 1071, 1081, 1091, 1101, 1111, 1121, 1074, 1084, 1094, 1104, 1114, 1124, 1077, 1087, 1097, 1107, 1117, 1127, 1230.

- **CMR/CMR+/CMRx**
- **NMEA:** The following messages are available:

- ALR, ARA, ARR, ATT, AVR
- BTS
- CAP, CPA, CPO
- DDM, DDS, DTM
- GBS, GGA, GGK, GGKX, GLL, GMP, GNS, GRS, GSA, GST, GSV
- HDT, HPR
- LTN
- MDM
- POS, PSP, PTT, PWR
- RCA, RCS, RMC, ROT, RSP
- SBD, SGA, SGO, SGL, SGP, SIR, SLB, SQZ, SSB
- TEM, THS, TTT
- VCR, VCT, VEL, VTG
- ZDA
- **NMEA 2000** (on CAN bus only):
  - System Time
  - Position, Rapid Update
  - COG and SOG, Rapid Update
  - Position Delta, High Precision Rapid Update
  - Altitude Delta, High Precision Rapid Update
  - GNSS Position Data
  - GNSS Satellites in View
  - GNSS Pseudorange Noise Statistics
- **GSOF** (except on CAN bus):
  - Position Time
  - Lat, Long, Ht
  - ECEF Position
  - Velocity
  - PDOP Info
  - Position VCV
  - Position Sigma
  - Attitude Info
  - Brief All SV Info
  - Position Type Info

**Port Configuration.** Before defining the message to be output on a port, do the following to configure the port:

- Serial ports **A**, **B** and **D**: Just choose an output baud rate.

- **USB serial** port: No additional setting required for this port.
- **TCP/UDP xxxx** (two ports): These ports allow the board to establish unidirectional or bidirectional communication with another device.

When one of these ports is used as an input (e.g. the board is a rover acquiring corrections through this port), you just have to configure the port (no programmed message is needed on the port).

When one of these ports is used as an output (e.g. the board is a base delivering corrections through this port), you need to configure the port and configure a message on the port.

Whether you are using the TCP or UDP protocol, you need to define an identification number (**IP Port** field; 100 to 65535) for each port.

If you choose **TCP**, you also need to choose one of these options:

- **Disable**: Use this option when you want to keep the port idle or you wish to end the connection that has been active so far on this port.
- **Input/output with authentication**: The port may be used as an input or output. In either case, the port will accept to communicate with another device if in the stream of exchanged data, the required login and password are provided and recognized. A **Show Login/Password** button appears in this case so you can access this information to read it (or change it). Note that this option cannot be activated if you leave the **Password** field empty. The port can be used to establish communication with only one device.
- **Input/output without authentication**: The port may be used as an input or output. No login and password are required to initiate the communication between the board and the other device. The port can be used to establish communication with only one device too.
- **Output only/Allow multiple connections**: The port is only an output. This time, several connections can be established at the same time through this port between the board and one or more devices.

If you choose **UDP**, you also need to choose one of these options:

- **Disable:** Use this option when you want to keep the port idle or you wish to end the connection that has been active so far on this port.
- **Broadcast transmit (output only):** The port is used as an output only. Messages will be broadcast to any devices on the same local network. Messages will be broadcast to IP address 255.255.255.255.
- **Broadcast receive (input only):** The port is used as an input only. Messages will be received from another device on the same local network, also using the UDP protocol.
- **Transmit to remote IP address (output only):** The port is used as an output only. Messages will be broadcast to a remote device on the same or on a different local network, whose IP address and IP port need to be specified in the field underneath (**Remote IP** field).
- **Receive from remote IP address (input only):** The port is used as an input only. Messages will be received from a remote device on the same or on a different local network, also using the UDP protocol. The IP address and IP port of this remote device need to be specified in the field underneath (**Remote IP** field).
- **NTRIP/DirectIPx** (two ports): Each of these two ports allows you to input or output data through an NTRIP or Direct IP connection. Special cases of NTRIP connections include Trimble RTX and Embedded NTRIP Caster.  
*NTRIP, general case:* Choose **Connection Type= NTRIP**. You then need to define the following parameters:
  - **Automatic Connection at Startup:** The NTRIP connection will be restored automatically at receiver power-up if this option is activated.
  - **Mode:** Choose whether the receiver will operate as a server or a client seen from this port.  
 If you choose **Server**, the receiver will be a base providing correction messages.  
 If you choose **Client**, the receiver will be a rover expected to receive correction messages through the port.  
 Additionally, you should indicate the version of the NTRIP protocol used (**V2** or other).
  - **IP Address** (or URL) and **IP port:** IP identification of the caster where to connect to.
  - **Login** and **Password** (characters hidden or in plain): For a server, only a password needs to be entered (as

provided by the caster owner). For a client, both a login and password are required (as provided by the NTRIP service), thus giving the client an authorization to use the service.

- **Mount Point:** For a server, enter the name of the mount point for which the server is allowed to deliver its corrections.

For a client, use the **Load** button nearby to interrogate the server so you can choose the mount point you wish the port to connect to.

- **Send NMEA:** For a client only, choose whether the receiver's current location should be sent to the server before the latter can deliver corrections. For this option, the choice is usually dictated by the requirements inherent in the chosen mount point.

*NTRIP, special case #1:* Choose **Connection Type= Trimble RTX**. Define the same parameters as in the NTRIP general case (above), except that the IP address to be used is firmware-set so the receiver can successfully communicate with Trimble RTX services. The drop-down list attached to the **Mount Point** field provides the list of mount points made available by the Trimble RTX services. Choose one.

*NTRIP, special case #2:* Choose **Connection Type= Embedded NTRIP Caster**. Define the same parameters as in the NTRIP general case (above), except that the IP address, IP port and password to be used are all firmware-set. The drop-down list attached to the **Mount Point** field provides the list of mount points you created for the Embedded NTRIP Caster (mount point name= <Name> (<Identifier>). Choose the one you want your corrections messages to be sent to.

*DirectIP.* Choose **Connection Type= DirectIP**. You then need to define the following parameters:

- **Automatic Connection at Startup:** The DirectIP connection will be restored automatically at receiver power-up if this option is activated.
- **IP Address** (or URL) and **IP port:** IP address and IP port of the remote device where the port should connect to in DirectIP mode.
- **Login** and **Password:** Login and password allowing the port to connect to this device in DirectIP mode (this information is provided by the device owner).

- **Memory:** Choose the memory in which to record the output messages (Internal Memory or USB device). As explained earlier, only ATOM messages may be sent to memory. Choose the pack of ATOM messages you want to record (see above, ATOM messages).
- **CAN Bus:** Enable the use of the bus and choose a baud rate.

**Message Configuration.** After you have set the port through which your message will flow, do the following to define the content of the message:

- Select a data format (ATOM, RTCM, CMR/CMR+, NMEA, GSOF, NMEA 2000).
- Select a message type (**Type** field) and define additional parameters, if any requested (see all message specifics in the above sections of this Help file).  
For almost all message types, when two antennas are used, you also need to specify the GNSS antenna for which the message type will be generated (most of them may be generated for both antennas).
- Click on the **Add/Modify** button. The selected message type is inserted as a new row into the table shown underneath.
- Repeat the previous two steps until all the message types making up the content of your message are listed in the table.
- Click **Configure** to save the definition of your message and of the port through which the message will be routed.

NOTE 1: For any of the message types in the table, you can click directly on the corresponding row to edit (Use **Add/Modify** button) or delete (Use **Delete** button) the message type. All message types listed in the table may be deleted in one operation using the **Clear All** button.

NOTE 2: As already explained, preset lists of message types exist for some of the available formats (ATOM and RTCM). Be aware however that you can always customize these lists by adding or remove message types.

NOTE 3: The **Suspend** button can be used to suspend the generation of the message selected in the I/O Status pane (on the right). The button is then changed to **Resume** so you can ask for the reverse operation.

### Troubleshooting Log (ATL)

This function allows you to generate debug ATL data. ATL data may either be recorded as a file, or sent out via a communication port.

- **State:** Use this field to start or stop ATL data generation. Select **OFF** to stop generating ATL data. Select **Recording** to start generating and writing ATL data to a file. The file is saved either on the internal or an external memory (USB key for example). When you select this option, two additional parameters need to be defined (**Maximum Duration** and **Record ATL automatically at startup**; see below) Select **Output** to start generating and delivering ATL data via the port you specify in the field that appears next to this one. Possible choices are serial port A, B or D, USB serial or one of the two TCP ports.
- **Rate:** Choose the output rate at which ATL data will be generated.
- **Maximum Duration** (only if **Recording** has been selected): Choose the duration of ATL recording, in minutes or hours.
- **Record ATL automatically at startup** (only if **Recording** has been selected): Enable or disable this option.

### Event Marker/PPS Settings

This tab first allows you to define the active transition (**Edge**) of the signal applied to the event marker input. The active edge may be one of the following: **Rising** or **Falling**.

You can also set up the One-PPS output by defining its period (**Period**= from 1 minute to 100 Hz), its offset (**Offset** is a positive or negative time value expressed in milliseconds) and its active edge (**Rising** or **Falling**). Set **Period**= **OFF** if you don't want the board to deliver its 1-PPS signal.

### Programmable PIN settings

This function allows you to program the two Multi-Function Outputs MFO1 (pin 10) and MFO2 (pin 11) on the board's 26-contact output connector.

Each of the two outputs can independently be configured to provide the following signals:

- **Disabled:** Permanently low level signal ("0")
- **Position available:** Signal is high ("1") when position is available (otherwise "0")
- **Radar simulated pulse**
- **Ashtech LED signal output:** The output is configured to drive a LED indicating the GNSS status the Ashtech way.
- **Trimble LED signal output:** The output is configured to drive a LED indicating the GNSS status the Trimble way.

- **General Purpose Output:** Choose the active level for this output. This can be a logic “1” or logic “0”.

Configure each of the two outputs to the desired function and then click **Configure** to save your changes.

### Summary

This tab is the I/O status pane (located on the right), which gives an overview of all the programmed messages. Each table row provides the definition of one message (port used, message content). The columns are labeled as follows:

- Type: Port type.
- Port: Serial port name, IP address and/or IP port.
- Name: Port ID.
- Input: For an input message, indicates the type of message, otherwise blank (the message is an output message).
- Output, primary antenna: For an output message, indicates the content of the message generated for the primary antenna, otherwise blank (the message is an input message).
- Output, secondary antenna: For an output message, indicates the content of the message generated for the secondary antenna, otherwise blank (the message is an input message).

Different row colors are used in this table:

- Green: Connection successful - message flowing normally.
- Red: Connection problem - no message flow.
- White/Blue-gray succession from one row to the next: No particular meaning (normal table formatting).

If you click on a row, and provided the **Input Setup and Output Messages** tab is displayed on the left, this will open the definition of the corresponding message in this tab so you can quickly check or modify the message definition.

## Radio External Radio

This tab allows you to choose the port and radio model used, download and modify the currently used radio settings.

1) For any radio model used other than XDL (vers.2.04 or higher) or an ADL (vers. 5.02 or higher), you need to define the following parameters:

- **Port** and **Baud Rate:** Serial port (A, B or D) to which the external radio is connected and baud rate used.



- **Type:** Model of radio used: PDL HPB/LPB, ADL Vantage/Pro/35, XDL Rover 2 or none.
- **Load Radio Settings** button: Click on this button to read the current radio settings. This will work only if the radio is connected to the receiver as specified above and the radio has been powered up.  
Radio settings are first as reported after clicking on the **Load Radio Settings** button. Then you can change these settings and upload them to the radio using the **Configure** button.
- **Channel:** Select the desired channel/frequency.
- **Channel Spacing:** Read-only field.
- **Protocol:** Select the protocol that suits the selected radio model.
- **Airlink Speed** (between 4800 and 19200 Bd) and **Modulation**.
- **Sensitivity:** Select the desired reception sensitivity (**Low, Medium, High**)
- **Transmit Power:** When applicable, choose the power you want the radio to radiate.
- **Forward Error Correction (FEC)** and **Scrambler:** Functions specific to Pacific Crest radio for safer radio transmission. Enable or disable these functions. See manufacturer documentation for more information.
- **Repeater Mode:** Enable this function if you want to board to forward the corrections it receives to the radio.

Remember most settings (i.e. channel, protocol, airlink speed, modulation, FEC, scrambler) must be consistent to allow two radios to work together.

2) If the external radio used is an XDL (vers.2.04 or higher) or an ADL (vers. 5.02 or higher), you need to define the following parameters:

- **Mode:** When this option is set to **Automatic**, the radio will be powered on automatically when you power on the receiver. If it's set to **Manual**, the radio will be powered on or off only by setting the previous field accordingly.
- **Channel:** Choose the carrier frequency to use from the list of available frequencies.
- **Channel Spacing:** A read-only field showing the channeling (12.5 kHz) for the type of radio used.
- **Radio Mode:** Choose the set of parameters defining the operating mode you want the internal radio to work in (up

to 19 different ones). Each set of parameters includes such data as protocol and airlink speed used as well as scrambling and FEC settings.

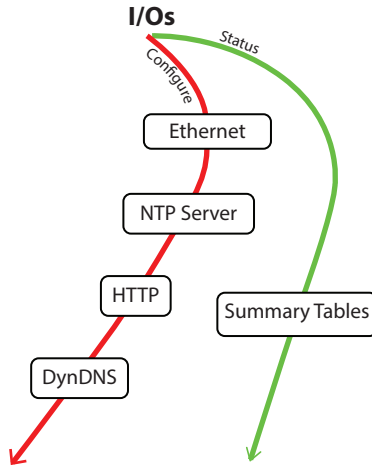
- **Sensitivity:** Select the desired reception sensitivity (**Low, Medium, High**).
- **Transmit Power:** Choose the radiated power (500 mW or 2 W) for when the radio is used as a transmitter.
- **Repeater Mode:** If you enable this function in the receiver, and you are using a Trimble protocol (TRIMTALK) as radio mode, you must indicate which specific function the receiver has to fulfill (complete the **Repeater Type** field): “**Base with 1 Repeater**”, “**Base with 2 Repeaters**”, “**Repeater One**” or “**Repeater Two**”.

If you are using another protocol (EOT, FST, etc.), enter the delay, in milliseconds, you want to introduce between the moment of reception and the moment of retransmission.

### Status Pane

The pane provides a summary of the current radio settings.

## Network



### Ethernet

- **MAC Address:** This field provides the board’s MAC address (read-only field).

- **Power State:** Select **ON** to power on the Ethernet device. Select **OFF** to switch it back off. The change will be effective after clicking **Configure**.

#### Ethernet Settings:

- **DHCP:** Use this field to define the IP address of the board. If you want the network to assign an IP address dynamically to the board, select **Client** or **Server**, depending on whether you want to board to operate as a client or a server. If you want to assign a static IP address, select **OFF**. In that case, enter the following parameters: **IP address**, **Netmask** (network mask), **Gateway**, **DNS Address** and **DNS address 2**.

When you are done, don't forget to click on the **Configure** button to save all your settings.

#### NTP Server

This tab allows you to enable the board's internal NTP server (NTP for Network Time Protocol).

When you are done, don't forget to click on the **Configure** button to save this setting.

When this option is active, the board can respond to any IP request for GNSS time (as soon as it can calculate GNSS time).

#### HTTP

This tab allows you to change the IP port through which the connection to the Web Server takes place. After you change the port ID and you click **Configure**, the IP connection is automatically restarted using the new IP port ID.

#### Dyn DNS

DynDNS is an update mechanism through which you can make sure the host name of your board will always match the dynamic IP address assigned to it by your Internet provider. This requires that you create an account on DynDNS and you choose the service you want to use (e.g. Standard DNS, Dynamic DNS or Managed DNS). Enter the following parameters:

- **Enable:** Check this button if you want to allow the board to connect to the DynDNS server after you have clicked on the **Configure** button.
- **System:** Type in "dyndns@dyndns.org"
- **Hostname:** Enter the host name you gave to the board when you created your DynDNS account.

- **User Name, Password:** Enter the user name and password you chose when creating your account on DynDNS. These will allow the board to establish an IP connection with the DynDNS server.
- **Forced Update Period:** Enter the interval of time (between 60 and 3600 s) at the end of which the board will be requested to automatically query the DynDNS server for updating the “host name vs. current IP address” pair.
- **Update Now** button: Click on this button if you want the board to query the DynDNS server for updating the “host name vs. current IP address” pair immediately after clicking on the **Configure** button.

When you are done, don't forget to click on the **Configure** button to save all your new settings.

To create an account on DynDNS:

- Go to <https://dyn.com/dns/>
- Click on SIGN IN.
- Choose the service you want to use and then follow the instructions to complete the creation of your account.

### Status Page

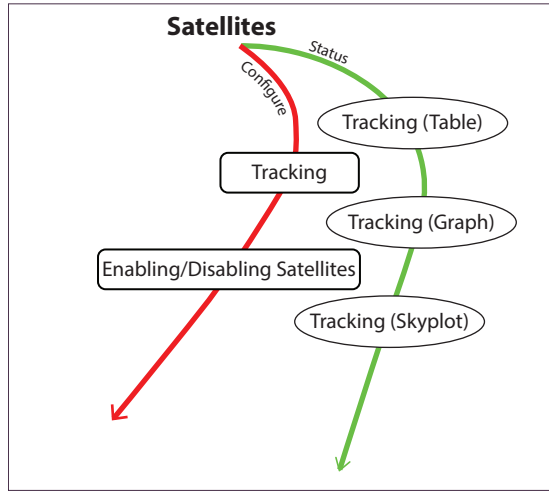
This page provides a summary of all the current settings pertaining to Ethernet and NTP Server.

See example below.

Ethernet	
Power State	ON
DHCP	ON
IP Address	10.20.4.38
Netmask	255.255.255.0
Gateway	10.20.4.1
DNS Address	10.20.8.30
DNS Address 2	10.2.1.70

NTP Server	
Mode	Disabled
Status	Stopped

## Satellites



### Tracking

This function is used to:

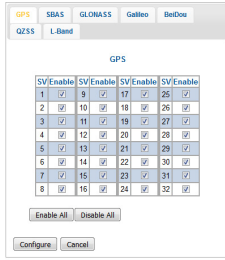
- Define the elevation masks, in degrees, used in position computation and for raw data collection (these angles are common to all GNSS constellations used).
- Select the GNSS constellations you want the board to work from (GPS, SBAS, GLONASS, Galileo, BeiDou, QZSS, L-Band). By default, all constellations are used.
- **Signals:** This option is used to place an input filter so that only some signals are actually tracked by the board. Select **All** for no filter (all signals tracked), **Optimal** for letting the board decide which signals should be tracked and used preferably, **Legacy** for asking the board to track only the signals that are compatible with previous generation software tools (GPS L2C discarded).
- Select the signal frequencies to use: L1/G1/E1/B1 (“L1” frequency) or L2/G2/E5b/B2 (“L2” frequency).

### Enabling/Disabling Satellites

For each visible constellation, this function lists all the satellite IDs of the constellation. Use the tabs on top of the page to select a constellation.

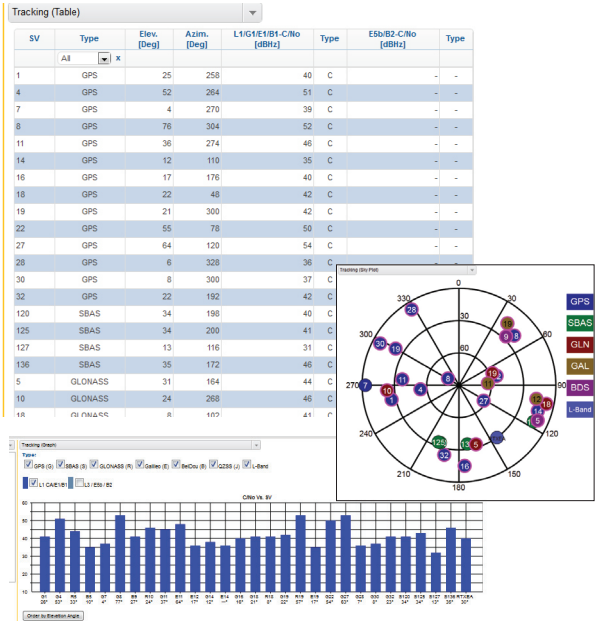
Use the check boxes in the table to enable or disable each of the satellites.

You can also use the **Enable All** or **Disable All** buttons to respectively use or reject all the satellites.



### Tracking (Table, Graph, SkyPlot)

For each antenna, or for the two antennas when applicable, this function gives a detailed report on the constellations used, in the form of a table, bar graph or sky plot.



Columns “L1/G1/E1/B1-C/No” list the codes used to describe the signals received on the L1 frequency:

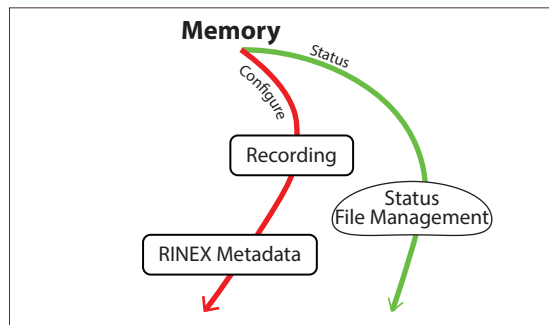
- C: L1 C/A (GPS, SBAS, QZSS), L1-SAIF (QZSS), G1 C/A (GLONASS), E1 (Galileo)
- I: B1 (BeiDou)
- L: L1C (GPS, QZSS), G1 P (GLONASS)
- W: L1 Z-tracking and similar (GPS)

Columns “L2/G2/E5b/B2-C/No” lists the codes used to describe the signals received on the L2 frequency:

- L: L2C (L) (GPS, QZSS)
- C: G2 C/A (GLONASS M)
- W: L2 Z-tracking or similar (GPS)
- P: G2 P (GLONASS)
- Q: E5b (Galileo)
- I: B2 (BeiDou)

NOTE: Signal codes may be combined. For instance, “C/W” reported in the tracking table means the two signals (C and W) are received simultaneously.

## Memory



## Recording

This tab deals with data recording (on the left), and memory statuses (on the right).

- **Data Recording** check box: Check on this box if you want to enable data recording. Keep it cleared otherwise.
- **Site Name**: Enter a 4-character name depicting the site where data recording will take place.
- **Memory Storage Location**: Tell the board where to store the recorded data. You can choose between **Internal Memory** and **USB Device**. On selecting one of these options, the line underneath (**Available space**) indicates the remaining free space on the selected memory.
- **Recording Interval**: Choose the recording rate (in minutes, seconds or Hertz). The lines underneath (**Data type...**) list the currently programmed messages and their respective output rates for each of the antennas used (if two are

used). Please select a recording interval that is equal to, or slower than the fastest output rate the board can use.

- **Ring File Memory:** With this option enabled, the board will be able to collect data for an unlimited period of time without user intervention. In practice, after this option has been enabled, the board will automatically delete the oldest file when the amount of available memory drops below 25 MBytes.
- **Record External Event:** Check this box if you wish to insert a TTT message into the record file whenever an external event is detected at the board's input.
- **Record PPS:** Check this button if you wish to insert PTT messages into the record file to time-tag the PPS output signal.
- **Split Data into Preset Duration Files** and File Duration: Enable the first of these two options if you wish to create several data files with fixed duration rather than one single (huge) data file. Then choose that duration, in minutes or hours. Files will end at round times. For example if you ask for data recording at 4:50 and you choose a file duration of 30 minutes, then recording will start at 4:50 and the first file will end at 5:00.

When you are done, don't forget to click on the **Configure** button to save all your new settings.

NOTE: If you want to log just NMEA messages and not the G-file and ATOM messages, you need to disable ATOM messages after enabling data recording. Go back to the **I/Os** tab to disable these messages.

### **RINEX Metadata**

You can define the following additional and optional parameters for insertion into the header of every single RINEX file the board will generate:

- Agency
- Observer
- Marker Name
- Marker Number
- Observation Comment
- Navigation Comment

When you are done, don't forget to click on the **Configure** button to save all your new settings.



## Right Side of the Web Page

Memory

Internal Memory : 50% (6 Files) 224.7 MB

Log Files Memory : 0% (11 Files) 26.3 MB

USB Device : 46% (7 Files) 3.7 GB

Files

Internal Memory  Log Files Memory  USB Device

G-File Only

/

<input type="checkbox"/>	Name	Size	Modification Date
<input type="checkbox"/>	uploading.log	28.0 KB	1970-01-01T00:00:53Z
<input type="checkbox"/>	G2639A17.205	132.0 KB	2017-07-24T15:14:31Z
<input type="checkbox"/>	SL	1.9 MB	2017-07-25T15:31:36Z
<input type="checkbox"/>	www		2017-07-25T15:31:49Z
<input type="checkbox"/>	G2639A17.206	25.3 MB	2017-07-25T15:31:54Z
<input checked="" type="checkbox"/>	G2639B17.206	125.4 MB	2017-07-26T07:19:35Z

Selected: < 1 KB

Delete All Files Delete Files Transfer files to FTP server Copy to USB Device Convert into RINEX

Transfer to External FTP Server

FTP Server  Username

FTP Port 21 Password

FTP Path

G-File Conversion

Rinex Conversion RINEX 3.02

File Compression: Hatanaka  Tar.Z

Select Data to Convert: GPS  GLONASS  SBAS  GALILEO  BEIDOU  QZSS

### Memory

In the right-hand part of the screen:

- **Internal Memory** bargraph: Indicates the current percentage of free/busy space, as well as the total number of files stored in the internal memory. The total memory size is displayed to the right of the bargraph.
- **Log Files Memory** bargraph: Indicates the current percentage of free/busy space, as well as the number of files stored in the log memory. The total memory size is displayed to the right of the bargraph.

The log memory is one of the two partitions created in the internal memory. It is dedicated to saving \*.log files. A log file is generated for every day of operation. The log file is like a log book in which the different events of the day are listed in order of appearance.

- **USB Device** bargraph: Indicates the current percentage of free/busy space, as well as the number of files stored on the USB device. The total memory size is displayed to the

right of the bargraph. If there is no USB device, the bargraph reads “No File” and no memory size is mentioned to the right of the bargraph.

### Files

- {List of files in selected memory}: Select a memory (**Internal Memory**, **Log Files Memory** or **USB Device**). The table located underneath lists the files currently stored in this memory. You can filter the memory content to list only the G files stored in memory (check on the **G-File Only** box in this case).

The list shows the name, size and last modification date of each file.

- **Delete All Files** button: Deletes all the files from the selected memory but keeps all the existing folders and subfolders.
- **Delete Files** button: Deletes the selected file(s) or directory (ies).
- **Transfer files to FTP server** button: Transfers the selected file or files to the external FTP server defined in the fields just underneath (**FTP Server**, **FTP Port**, **FTP Path**, **Username**, **Password**).
- **Copy to USB Device** button: Copies the selected file or files to the USB device connected to the receiver.
- **Convert into RINEX** button: Converts the selected file or files according to the settings made in the G-File Conversion pane below.
- **Download Selected Files** button: Copies the selected file(s) to the computer. Folders cannot be downloaded.

NOTE 1: When the number of files in a folder exceeds 1000 files, you may use the “1”, “2”, etc. buttons and “<” and “>” buttons showing up on top of the upper-right corner of the list of files to list only a portion of this list on the screen (a few hundreds files listed). In this case sorting files according to name, size or creation date will apply to that particular portion of files.

NOTE 2: The total size of all the selected files is shown just above the **Delete All Files** button.

### Transfer to External FTP Server

Enter the following parameters to define the external FTP server where the selected files will be copied after clicking on the **Transfer files to FTP server** button:

- **FTP Server**: Enter the FTP server name or IP address.

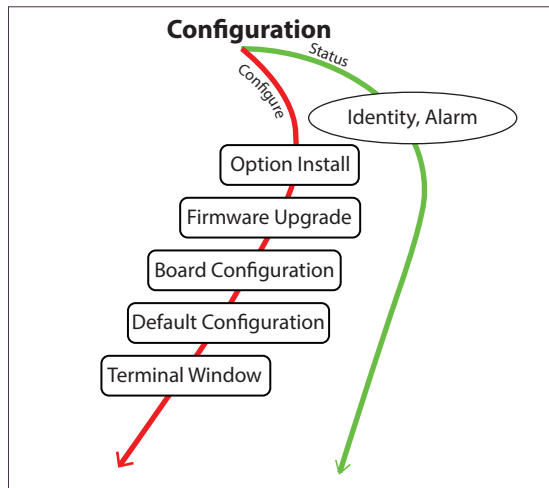
- **FTP Port:** Enter the IP port number of the external FTP server (default is 21).
- **FTP Path:** Enter the path of the directory on the external FTP server where the files will be stored.
- **Username, Password:** Enter the user profile allowing the receiver to access the external FTP server.

### G-File Conversion

Enter the following parameters to define how the selected files will be converted to RINEX when clicking on the **Convert into RINEX** button:

- **RINEX Conversion:** Choose in which RINEX version to convert the selected file or files.
- **Select Antenna:** Choose which data in the selected files should be converted to RINEX, based on whether these data originate from either the primary antenna, the secondary antenna, or both.
- **File Compression:** Choose whether the converted files should be compressed. The available two compression algorithms (Hatanaka, Tar.Z) may be combined.
- **Select Data to Convert:** Select the GNSS constellations for which you want the corresponding data present in the selected files to be converted to RINEX. More constellations are available for conversion when selecting the most recent RINEX versions.

## Configuration



This tab is used to:

- Install one or more firmware options, a geofencing zone or a warranty extension.
- List the versions of all the items used in the board (embedded firmware, firmware options, etc.).

### Option Install

To install a new firmware option, you should first make sure you have the option code available, following your purchase of this option. Then:

- Choose “**Automatic**” in the **Option** drop-down list.
- Enter the option code in the field underneath (**Code**).
- Click on **Install** to complete the installation.

To install a geofencing zone, through which you limit the use of the board to a particular geographical area, you should first make sure you have the code available, following your purchase of this feature. Then:

- Use the **Geofencing Zone** drop-down list to select the country or area where the board is supposed to be used.
- Enter the code in the field underneath (**Code**).
- Click on **Install** to complete the installation.


To install an extended warranty, you should first make sure you have the code available, following your purchase of this extended warranty. Then:


- Enter the code in the **Update Warranty Date** field.
- Click on **Install** to complete the installation.

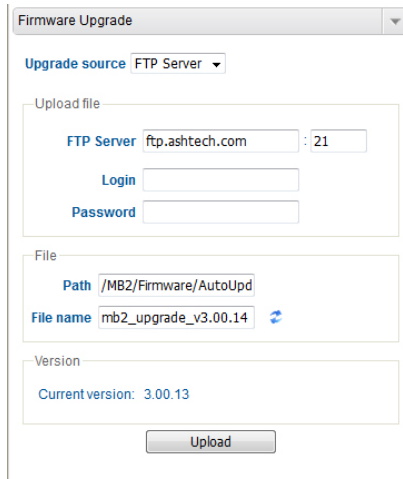
### Firmware Upgrade

The board firmware can be upgraded in two different ways, depending on where the upgrade file (a tar file) is located:

- If you downloaded it to your computer, select “**Local File**” in the **Upgrade source** field, then click on the **Choose File** button to find the file on your computer and select it. When the **Upload** button is usable, click on it and wait until the upgrade is complete.
- If it's posted on a remote server, select “**FTP Server**” in the **Upgrade source** field, then enter the address of the FTP server (IP address and port), as well as your login and password to access this server.

Click on  to browse the FTP for a tar file. If the tar file is in the FTP's root folder, its name will appear in the **File Name** field below.

If it's in a sub-folder on the FTP, first type in that folder's name in the **Path** field before clicking on . Click on the **Upload** button to start the upgrade. Wait until the upgrade is complete.



Firmware Upgrade

Upgrade source: FTP Server

Upload file


FTP Server: ftp.ashtech.com : 21

Login:

Password:

File

Path: /MB2/Firmware/AutoUpd

File name: mb2\_upgrade\_v3.00.14 

Version

Current version: 3.00.13

Upload

NOTE 1: The currently installed version of firmware is indicated just above the **Upload** button.

NOTE 2: Don't forget to click on the button shown at the end of the upgrade procedure. Through this action, all the information displayed on the different tabs of the Web Server will be properly refreshed following the installation of a new firmware version.

## Board Configuration

**Internal Data Update Rate Configuration (POP):** Choose the rate at which the board processes internally the GNSS data it receives (rate expressed in Hz, from 1 to 50 Hz). Then click on the **Configure** button just underneath to make your new setting effective.

**Save Configuration:** The current board configuration can be saved as a file (\*.par file) to the board's internal memory or to a USB device. Select the desired storage device and then click **Save**. The name of the saved file then appears under the button.

**Load Configuration:** Any board configuration file (a "\*.par" file) saved earlier to internal memory or USB device may be uploaded to another board thus providing this board with a new current configuration. Just select the desired file from

the selected memory. You may also use the **Choose File** button to browse for the desired configuration file if it is stored on your computer. Once the file has been selected, just click on **Load**.

You are allowed to load the configuration from another board provided this board is fitted with the same set of options as the board you are working on.

**IMPORTANT!** This action has no impact on the default configuration a board will be using following a reset operation on this board (see also next section).

**Reset Configuration:** You may change the configuration of your board by using one of the three buttons below:

- **Reset:** Will cause the board to restart with the default configuration (this will be either the factory settings or the user-defined default configuration if you created one (see below)).
- **Reboot:** Will cause the board to restart without changing anything to its current configuration.
- **Reset to Factory Settings:** Will force the board to restart with the factory-set configuration.

**Reset Embedded NTRIP Caster** (visible only if NTRIP Caster option enabled): This button deletes all the information pertaining to the embedded NTRIP caster (settings, users and mount points).

### User-Defined Default Configuration

Some users may be interested in giving their board a specific default configuration, different from the one they normally get after a reset operation. This is possible using a command file (a “\*.cmd” file) containing a list of executable proprietary \$PASH commands.

**NOTE:** Creating a cmd file requires some expertise in \$PASH commands. See *MB-Two Reference Manual*.

After the desired command file has been selected, the following takes place when you reset your board:

1. The factory defaults are first re-applied to the board.
2. The \$PASH commands listed in the specified command file are then executed, thus modifying or complementing the factory defaults and giving the board a user-defined default configuration instead of the standard default one.

**IMPORTANT:** This feature is fully independent of the one described in the previous **Board Configuration** tab and deals with completely different files.

The **User-Defined Default Configuration** tab is organized as follows:

- **Command File Header:** A read-only field showing the content of the first line (starting with a #) read from the currently selected command file (usually contains identification of the resulting user-defined default configuration).  
“**No File**” is displayed if there has been no command file selected so far.
- **Delete Command File** pane: If a command file is currently selected, you may want not to use it anymore. In this case, just click on the **Delete** button. A subsequent reset operation would then result in reloading the standard default configuration.
- **Save Command File** pane: Use this pane to save the currently used command file to the board’s selected storage device (**Internal Memory** or **USB Device**). The file may be renamed before being saved to the selected device.
- **Load New Command File** pane: Allows you to select and load a new command file:
  - Select the storage device (board’s **Internal Memory** or **USB Device**) where the desired file is currently stored, and then select this file in the list underneath.
  - Or browse your computer disk and select the desired command file.

As a result, the name of the currently selected file is now shown in the **Selected Command File** field.

Click the **Load** button to load the selected file. It is this file that will be run after a reset operation to change the standard default configuration into a user-defined configuration.

When the load operation is complete, the **Command File Header** field shows the content of the first line read from this file.

### Terminal Window

This tab allows you to apply \$PASH proprietary commands directly to the board. Type in the command in the text box and then press **Enter** on your keyboard or click on the **Send** button. Your command then appears in the text box underneath followed by the board response in a different color.

The syntax of all available commands is described in the *MB-Two Reference Manual*.

From time to time, use the **Clear View** button to empty the lower text box.

## Identity & Alarms

The Status pane of the **Configuration** page provides a detailed description of the board (hardware, firmware, OS, geofencing, firmware warranty date, etc.), including the firmware options installed permanently and those installed for a limited period of time (temporary options).

For each temporary option, the expiration date is indicated. That date is removed from the table as soon as the option has expired.

Identity	
Receiver Type	MB-Two
System Name	ASHTECH MB-TWO
Serial Number	5532C02639
MAC Address	00:09:66:01:7B:3A
Firmware Version	3.20.20
Firmware Date	2016.09.23
SL	RS72V25
API	0.10
PVT	RP72V13
DSP	RC72V13
OS2	3.0.101 #777 SMP Sep/Thu/MSK
BL2	0.21
HTML	RH72V12
WS	RW72V10
Antenna Database Version	8.13 beta

**Geofencing:** Worldwide, Brazil

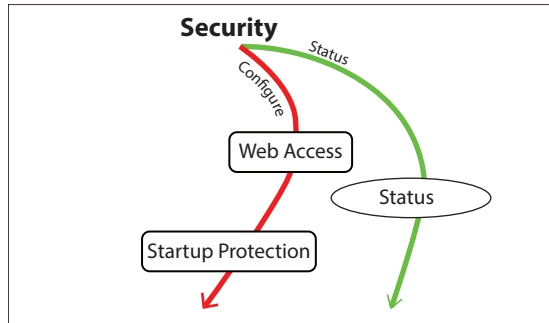
**Firmware Warranty Date:** 2018-02-23

Option	Installed
GPS	N ✓
GLONASS	G ✓
Galileo	O ✓
BeiDou	B ✓
L2 Frequency Tracking	Y ✓
L3 Frequency Tracking	S ✓
2Hz Output Rate	2 ✓
5Hz Output Rate	5
10Hz Output Rate	6
20Hz Output Rate	W ✓
50Hz Output Rate	8 ✓
RTK Rover	J ✓
RTK Base	K ✓
Flying RTK	F ✓

Option	Installed
RAIM	I ✓
Data Recording	R ✓
Altitude Limit Removed	A
Speed Limit Removed	V
DUO Mode	D
3D-attitude	E ✓
L-Band	L ✓
Dithered RTK 30/30	3
Dithered RTK 7/2	7
CenterPoint RTX	C
RangePoint RTX	P 2016-11-20 (E1)
ViewPoint RTX	4
FieldPoint RTX	9 2016-11-20 (E1)
RTX RAM	1



## Security



This tab deals with:

- Web Server access rights
- Startup protection

### Web Access

This tab deals with the level of security applied to the Web Server. Its content depends on the currently used level of security:

1. Security has been enabled (**Security= Enabled**):
 

A logged in user can:

  - Read or change the level of security applied to the Web Server.
  - Log out.
  - Change the password. The password must be entered twice and must have at least medium strength (strength indicated as you enter the password).
  - Change all the board settings using the different tabs in the Web Server.
2. Security has been enabled with anonymous access (**Security= Enabled with Anonymous Access**):
  - The tab shows that this mode is currently active and indicates whether you can download or/and delete files.
  - Read, but not change, the current board settings using the different tabs in the Web Server.
  - A user can escape from the anonymous mode after entering the login and password on this tab.

When you are done, don't forget to click on the **Configure** button to save all your new settings.

## Startup Protection

This tab allows you to control the startup protection.

When powering on a board with active startup protection, you are requested to enter the correct password to be able to use the board. As long as no correct password is typed in, the board is said to be “locked” and so operates with minimum functionality. Entering the correct password will unlock the board (the board will then switch to the “Unlocked” status). Only then will you be allowed to control the board from the Web Server.

**IMPORTANT!** By default the password does not exist. Because the protection can’t be activated until a valid password has been defined, you should first define and confirm the password. Once this is done and the protection has been activated using this password, no password change is allowed. You can however change the password if the startup protection is still active but you’ve already entered the password to unlock the board.

- If the startup protection is active when you open the tab, you will read **Startup Protection= Enabled**. In the status pane on the right, the board status will read “**Unlocked**” if the password has been typed in after powering on the board, or “**Locked**” if the board is still waiting for this password to be entered.

To disable the startup protection after you have logged in, enter the password and then click on the **Disable** button.

NOTE: When starting the Web Server of a currently startup-protected board, you will first be requested to log in, as you usually do as an authorized user, and then you will have to enter the password to unlock the board.

- If the startup protection is inactive when you open the tab, you will read **Startup Protection= Disabled**.

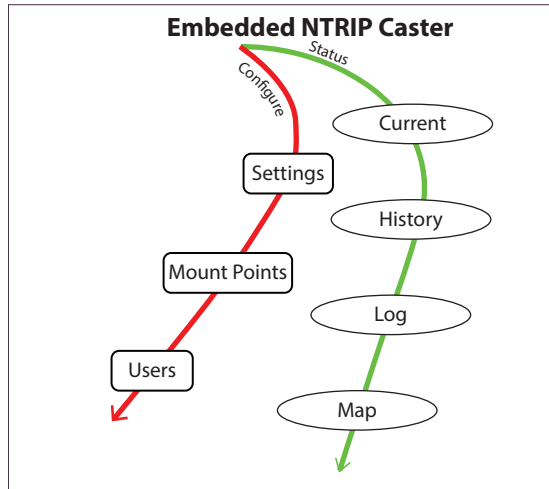
To enable the startup protection after you have logged in, enter the password and then click on the **Enable** button.

- If you are logged in, you can use the lower part of the tab to change the startup password. You need to enter the new password twice before you are allowed to save it by clicking on the **Configure** button.

## Status Pane

This pane provides a summary of the current settings pertaining to Web Server access and startup protection.

## Embedded NTRIP Caster



This tab is used to:

- Create a caster in the receiver
- Define miscellaneous information about this caster, to be forwarded to users, when this will be necessary.

By default, the NTRIP caster is disabled and there is no NTRIP caster password defined (the field is empty). Note that the NTRIP caster cannot be started until you define a password for the NTRIP caster.

### Settings

#### Caster Settings:

- **Enable/Disable:** Use this button to enable the use of the NTRIP caster defined below.
- **Hostname or IP Address**
- **Password:** Define the password a base owner should enter to be allowed to connect her/his base station to the caster as a data provider (i.e. as one of the corrections sources available through the caster).

The **Show Password** button allows you to view and possibly change the password. Clicking on this button will show the password with masked characters. Enable the check box next to it to show the password in plain. By default the **Password** field is empty.

If you change the password, don't forget to click on the **Configure** button to save it.

In anonymous mode, you are not allowed to view the password.

- **Maximum Simultaneous Connections per User:** Use this field to limit the number of simultaneous connections allowed per user.

#### **Caster Information:**

This optional information will be forwarded to users when starting a connection to the caster:

- **Caster Identifier**
- **Caster Operator**
- **Latitude:** Caster latitude (in degrees, and fraction of degree)
- **Longitude:** Caster longitude (in degrees, and fraction of degree)
- **Fall Back Caster:** Caster where to connect to in case this one breaks down.
- **Network Identifier**
- **Network Operator**
- **Country:** The three-letter international code identifying the country where the caster is operated.
- **Fee:** Indicate whether the caster can be accessed for free or not (clear the box if it's free).
- **Web Address for Network Information:** Enter the name of the website where users can get more information about the network.
- **Web Address for Stream Information:** Enter the name of the website where users can get more information about data streaming.
- **Web/Email Address for Registration:** Enter the name of the website or email address where users can register to be allowed to use the caster.

When you are done, don't forget to click on the **Configure** button to save all your new settings.

#### **Mount Points**

##### **Mount Point:**

Use this tab to define each of the mount points accessible via the caster. For each mount point, enter the following parameters:

- **Name**
- **Identifier**
- **Format**

- **Format Details**
- **Latitude:** Latitude of the mount point's geographical location.
- **Longitude:** Longitude of the mount point's geographical location.
- **Country:** The three-letter international code identifying the country where the mount point is operated.
- **Fee:** Indicate whether the mount point can be accessed for free or not (clear the box if it's free).
- **NMEA:** Indicate whether or not the mount point needs to receive the user's approximate position in NMEA format.

### Mount Points List:

This table lists all the mount points created so far.

Selecting a row in this table allows you to edit the corresponding mount point definition in the fields described above. After making the required changes, use the **Add/Modify** button to update the definition of the mount point in the table. If you define a mount point with a new name, clicking on this button will add this mount point to the table as a new row.

The **Clear** button is used to clear all the fields within the Mount Point frame.

The **Delete** button is used to delete the mount point you select in the table.

## Users

### User:

Use this tab to define, one by one, the users allowed to connect to the caster. For each user, enter the following parameters:

- **Username**
- **Password:** Define the password the user should enter to be allowed to use the caster. The password may be shown in plain (check the box nearby), or hidden, then displaying "\*" characters instead.
- **Mount Points List:** For each user, check the mount point or mount points this user will be allowed to use. Enabling the check box the closest to **Mount Points List** allows you to select in one operation all the existing mount points listed in the table.

### Users List:

This table lists all the users created so far.

Selecting a row in this table allows you to edit the corresponding user definition in the fields described above. After making the required changes, use the **Add/Modify** button to update the definition of the user in the table. If you define a user with a new name, clicking on this button will add this user to the table as a new row.

The **Clear** button is used to clear all the fields within the User frame.

The **Delete** button is used to delete the user you select in the table.

## **Current**

### **Source Activity:**

Lists the current status of each mount point, the time when a connection to this mount point started and the IP address used.

### **Client Activity:**

Lists the currently connected users, the mount point they are using, the start time and IP address of their connection.

## **History**

### **Source Activity:**

Lists the last connections made. Each line provides the mount point used, the times when the connection was started and stopped and the IP address used for the connection.

### **Client Activity:**

Lists the last users connected to the caster and now disconnected. Each line provides the user name, the mount point that was used and the times when the connection was started and stopped.

## **Log file**

This tab lists the last actions performed, relevant to the caster.

## **Map**

Geographic map showing the location of the caster (an orange spot). The map content is refreshed based on the value you give to **Refresh Interval**.



## Chapter 6. Configuring the MB-Two Using Serial Commands



The MB-Two can be configured using proprietary commands known as the “\$PASH” commands. Using these serial commands is not the fastest way of configuring a board. Usually, the web server – a smart interface also relying on the use of \$PASH commands – will usually be the preferred tool. However, in some cases, you may want to apply serial commands directly. This chapter explains how to use them.

### Introduction to Serial Commands

---

Serial commands allow you to communicate directly with the receiver in its proprietary command language. Serial commands can be used for various purposes such as:

- Changing default settings
- Monitoring different receiver statuses (internal operation, constellations, etc.)
- Outputting messages on request

Serial commands fall into two categories:

- *Set* commands (\$PASHS,...), used to set or modify the receiver’s internal parameters.
- *Query* commands (\$PASHQ,...), used to interrogate the receiver.

If you wish to change some internal settings in the receiver or modify the operating mode, then use the available set of \$PASHS commands.

If you want to read the instant status of the receiver, use the set of \$PASHQ commands. The receiver responds to \$PASHQ commands by returning one of the following, depending on which \$PASHQ command is sent:

- ASCII or binary formatted \$PASHR sentences, for your reading or/and automatic parsing.
- Non-formatted responses, like for example ASCII tables, only for your reading (not intended for automatic parsing).

If you want to program the receiver so that it delivers its results at regular intervals, then you should also use the set of available \$PASHS commands. The receiver will respond by delivering messages at the specified output rate. Messages fall into different groups:

- NME (standardized NMEA messages and NMEA-like Ashtech proprietary messages)
- ATM (Ashtech proprietary ATOM binary data)
- RT2 (Standardized RTCM-2 messages)
- RT3 (Standardized RTCM-3 messages)

CMR (Widely used TRIMBLE CMR messages)A three-letter identifier is part of the \$PASHS command header clearly identifying which group of data the command deals with. For example, "\$PASHS,NME,GGA,A,ON,1" will enable the GGA NMEA message on port A at an output rate of 1 second.

Some \$PASHS and \$PASHQ commands can initiate the same \$PASHR response. However, \$PASHS will return \$PASHR responses at regular intervals whereas \$PASHQ will only return a single \$PASHR response.

In general, all the messages of a given group are output inside a dedicated transport layer. For example, NMEA-like and RAW data are output using the Ashtech legacy \$PASHR frame, whereas ATM and RT3 data are output using the standardized RTCM-3 transport protocol. For more convenience, using the \$PASHS,ENC command, you can ask the receiver firmware to output all the groups via the same port and using the same frame (e.g. \$PASHR).

Standard NMEA messages will all be output with the standard ASCII NMEA preamble (e.g. \$GPGGA) and not with the "\$PASHR.." preamble.



The few conventions used to describe the serial commands in this manual are summarized in the table below.

String or symbol	Description
\$PASHS	Header for set commands (Whole line shown in bold characters)
\$PASHQ	Header for query commands (Whole line shown in bold characters)
\$PASHR	Receiver response line, in normal characters.
GP	Header in standard NMEA output messages for results provided by GPS.
GL	Header in standard NMEA output messages for results provided by GLONASS.
GA	Header in standard NMEA output messages for results provided by GALILEO.
GN	Header in standard NMEA output messages for results provided by GNSS (combination of several constellations).
GB	Header in standard NMEA output messages for results provided by BeiDou.
GQ	Header in standard NMEA output messages for results provided by QZSS.
\$--	Header prefix for all standard NMEA messages delivered by the receiver.
[ ]	Optional field or parameter
,	Field delimiter
.	Decimal point (used in f-type fields)
c..	One-character string
d..	Integer
f..	Real number, with decimal places
h..	Parameter in hexadecimal notation
m..	Denotes specific data format used, such as angles (e.g. ddm.mmm) or time (e.g. hhmmss.sss)
n	Used in the syntax of responses to query commands to indicate that a sequence of parameters will be repeated "n" times in the response. For example, n(f1,f2,f3) means the response will include the sequence "f1,f2,f3,f1,f2,f3,f1,f2,f3...". The value of n is specific to each command.
s..	Character string
*cc	Checksum

In response to a well recognized and properly executed set command, the receiver will return the message:

```
$PASHR,ACK*3D
```

A set command is said to be "NAKed" when it is not accepted or acknowledged. The following message is then returned:

```
$PASHR,NAK*30
```

If this happens, check that the command has been typed correctly and the number and format of parameters are correct. In some cases, the execution of a set command may be contingent upon the prior activation of the corresponding firmware option.

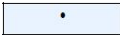
**Checksum Calculation:** The checksum is computed by “exclusive-ORing” all of the bytes in the message between, but not including, the “\$” and the “\*”. The result is “\*hh” where h is a hexadecimal character.

## Overview of the MB-Two Proprietary Commands

---

This section presents the complete list of proprietary commands applicable to the MB-Two board.

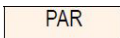
### How to Read the Tables Below



The tables that follow introduce all the available commands per topic, listed in alphabetical order.

- When a set command has a direct query command counterpart (e.g. “\$PASHQ,GPS” is the direct query command counterpart to “\$PASHS,GPS”), it is simply indicated by a “•” character in the right-hand column (cell with blue background).

In most cases the description of the query command is obtained simply by replacing the first word “Setting”, “Defining”, etc. in the set command description with the term “Reading”.



- When a query command has no direct set command counterpart, it is shown in a separate row. Its name appears in the right-hand column as well (orange background), together with its description in the middle column.



- When a set command has no direct query command counterpart, the corresponding cell in the right-hand column is empty.

For a detailed description of each command (syntax, fields, etc.), see *Chapter 7* for set commands, and *Chapter 8* for query commands.

## GNSS Sensor Tracking

Set Command	Description	Query Command
BDS	Enabling/disabling BeiDou tracking	•
GAL	Enabling/disabling Galileo tracking	•
GEM	Choosing geoid model	•
GLB	Choosing antenna input for L-band reception	•
GLx	Tracking signals in different bands	•
GLO	Enabling/disabling GLONASS tracking	•
<GNS>,USE	Enabling/disabling tracking of a GNSS Satellite	
GPS	Enabling/disabling GPS tracking	•
LBN,BEM	Other settings for user-added L-band satellites	•
LBN,USE	Tracking L-band providers/geo-satellites	
LBN,SAT	Adding new L-band satellite manually	
	Reading L-band setup	LBN
QZS	Enabling/disabling QZSS tracking	•
SBA	Enabling/disabling SBAS tracking	•
SNS	Configuring M-Sensor	•

## Application Setup

Set Command	Description	Query Command
3DF	Running attitude on baselines. Reading status.	•
3DF,ANG	Maximum baseline elevation	
3DF,CLB	Baseline auto-calibration	
3DF,MXB	Setting baseline length tolerance	
3DF,OFS	Heading-pitch-roll offset values	
3DF,RST	Resetting attitude and calibration	
3DF,Vx	Vector components	
BLN	Defining baselines and their data sources	•
BRV	Relative mode	•
BRV,RST	Resetting relative positioning mode	
RTK	Running RTK on set baselines	•
RTK,RST	RTK or RTX reset	
RTK,STI	Defining which base data stream to use	
RTX,DTM	RTX datum transformation	
RTX,KPI	Known RTX initialization point	
RTX,MOD	Specifying the RTX corrections service used	
RTX,RST	Resetting RTX position computation	
RTX,SRC	Specifying the RTX corrections source used	
	Reading RTX processing status	RTX

## Other

Set Command	Description	Query Command
MET,OWN	Entering local meteo parameters	
MET,REF	Entering meteo parameters of reference	
NME,TTT	Requesting the output of an event marker	
PHE	Setting the active edge of the event marker pulse	.
PIN	Assigning function to programmable pin on I/O connector	.
PPS	Setting PPS pulse properties	.

## GNSS PVT

## Raw/Differential Messages

Set Command	Description	Query Command
ATM	Enabling/disabling ATOM messages	
ATM,ALL	Disabling all ATOM messages	
CMR	Enabling/disabling CMR or CMR+ messages	
CMR,ALL	Disabling all CMR or CMR+ messages	
RT2	Enabling/disabling RTCM 2.3 messages	
RT2,ALL	Disabling all RTCM 2.3 messages	
RT3	Enabling/disabling RTCM 3.1/3.2 messages	
RT3,ALL	Disabling all RTCM 3.1/3.2 messages	

## NMEA and NMEA-Like Messages:

NMEA messages can be output manually using appropriate query commands (e.g. to output one GGA message, you can send the command "\$PASHQ,GGA" to the board and you will receive in return one GGA message tagged to the time of request.

NMEA messages can also be output periodically (which is usually done) by programming their output using the \$PASHS,NME command.

Query Command	Description
ARA	Attitude rate and accuracy
ARR	Vector & Accuracy Data
ATT	Legacy attitude message. HPR now preferred
AVR	Trimble "Time-Yaw-Tilt-Range" Message for Moving Baseline RTK
CAP	Received base antenna
CPA	Received antenna height
CPO	Received base position
DDM	Differential Decoder Message
DDS	Differential Decoder Status
DTM	Datum Reference
GBS	GNSS Satellite Fault Detection
GGA	GNSS position message
G GK	Position (Trimble proprietary message)
G GKx	Position & accuracy message (Trimble proprietary message)
GLL	Geographic Position - Latitude/Longitude
GMP	GNSS Map Projection Fix Data
GNS	GNSS Fix Data
GRS	GNSS Range Residuals
GSA	GNSS DOP and Active Satellites
GST	GNSS Pseudo-Range Error Statistics
GSV	GNSS Satellites in View
HDT	True Heading
HPR	Attitude, UTC time-tagged
LTN	Latency
POS	Position
PSP	Received physical reference station position
PTT	PPS time tag
RCA	Received CMR Type 2 Attribute
RCS	Recording status
RMC	Recommended Minimum Specific GNSS Data
ROT	Rate of turn
RSP	Received reference station position
SBD	BeiDou Satellites Status
SGA	GALILEO Satellites Status
SGL	GLONASS Satellites Status
SLB	L-Band Satellites Status
SQZ	QZSS Satellites Status
SSB	SBAS Satellites Status
TEM	Die temperature
THS	True Heading and Status
TTT	Event Marker

Query Command	Description
VCR	Vector & Accuracy Data
VCT	Vector & Accuracy Data
VEL	Velocity & Accuracy Data
VTG	Course Over Ground and Ground Speed
ZDA	Time & Date

## GSOFF Messages

Set Command	Description	Query Command
GSF	Enabling/Disabling GSOFF Messages	
GSF,ALL	Disabling all GSOFF Messages on a Port	
	Current Status of GSOFF Messages	OUT,GSF

Message #	Description
GSOFF 1	Position Time
GSOFF 2	Lat Lon Height
GSOFF 3	ECEF Position
GSOFF 9	PDOP Info
GSOFF 11	Position VCV Info
GSOFF 12	Position Sigma Info
GSOFF 16	Current UTC Time
GSOFF 38	Position Type Information
GSOFF 33	All SV Brief Info

## CAN-NMEA 2000 Messages

Set Command	Description	Query Command
	Current Status of NMEA 2000 Messages	OUT,"PGN"
PGN	Enabling/Disabling NMEA 2000 Messages	
PGN,ALL	Disabling all NMEA 2000 Messages	

NMEA 2000 Message Label
System Time (126992)
Position, Rapid Update (129025)
COG and SOG, Rapid Update (129026)
Position Delta, High Precision Rapid Update (129027)
Altitude Delta, High Precision Rapid Update (129028)
GNSS Position Data (129029)
GNSS Sats in View (129540)
GNSS Pseudorange Noise Statistics (129542)

## PPP Services

Set Command	Description	Query Command
PPP	Selecting PPP service	PPP
	Reading information on TERIASat	TRS
PPS,RST	Resetting PPS service	•
PPS,SRC	Defining corrections input for PPP service	•

## Antennas/Receiver Attributes

Set Command	Description	Query Command
ANH	Setting antenna height	•
	Antenna parameters	ANP
ANP,OUT	Defining a virtual antenna	•
ANP,OWN	Naming the local antenna/ 2nd local antenna	•
	Antenna name and offsets of received base	ANP,RCV
ANP,REF	Naming the antenna used at the base	•
ANR	Setting the antenna reduction mode	•
ANT	Setting the antenna height (alternative)	•
	Status of antennas	AST
RCP,OWN	Naming the local receiver	•
RCP,REF	Naming the reference receiver	•

## General Purpose

Set Command	Description	Query Command
	Listing anti-theft parameters	ATH
ATH,PWD	Defining the startup protection password	
DIF,BDS	BeiDou correcting data	•
DIF,SBA	Specifying which SBAS corrections to use	•
DYN	Setting receiver dynamics	•
ELM	Setting the elevation mask for raw data output	•
ENC	Setting data transport mode	
LCK,MOD	Controlling receiver lock	
LCK,OFF	Unlocking the receiver (startup protection)	
	Receiver lock (startup protection)	LCK
LCK,ON	Locking the receiver (startup protection)	
LCS	Enabling/disabling use of the local coordinate system	•
MSG	Defining a user message	
MET,REF	Entering meteorological parameters	
NPT	Defining how RTX and SBAS positions are tagged in POS messages	•
OCC	Writing occupation data to raw data file	•
	Output messages	OUT,MSG
PEM	Setting the position elevation mask	•
PGS	Defining the primary GNSS system	•
POP	Setting internal update rate for measurements and PVT	•
POS	Setting the antenna position	•
POS,CUR	Making the current position the reference position	
POS,MOV	Making the current position the reference position	
	Base position	POS,REF
	Site name	SIT
STI	Defining a station ID	•
TOP	Defining the type of output position	•
UDP	User-defined dynamic model parameters	•
VIP	Defining a virtual port	
	Receiver validity period	VLP
ZDA	Setting date & time	•



## Communication Ports

Set Command	Description	Query Command
	CAN Bus Status & Parameters	CAN
CAN,OFF	Disabling the CAN port	
CAN,ON	Enabling the CAN port	
CAN,PAR	Setting the CAN parameters	
DSY	Setting daisy chain	•
PRT	Setting baud rates for serial ports	•

## Differential Messages

Set Command	Description	Query Command
OUT,DIF,OFF	Disabling all differential messages	

## Ethernet

Set Command	Description	Query Command
DDN,PAR	Setting the DynDNS service	
DDN,SET	Sending the IP address manually to DynDNS	
ETD,PAR	Ethernet driver parameters	ETD
ETH,OFF	Powering off Ethernet port	
ETH,ON	Powering on Ethernet port	
ETH,PAR	Setting Ethernet parameters	
	Reading Ethernet status and parameters	ETH
FTP,OFF	Ending data transfer with FTP	
FTP,PAR	Setting an external FTP server	
FTP,PUT	Uploading files to FTP	
	Reading FTP status and settings	FTP
TCP,PAR	Setting the TCP/IP server	
TCP,UID	Entering login/password for TCP/IP connection	
	Reading TCP/IP server settings	TCP
WEB,OWN	Setting owner information	
WEB,PAR	Setting web server control & admin profile	
	Reading the web interface status	WEB

## GNSS Network

Set Command	Description	Query Command
CST,ON	Starting the embedded NTRIP caster	
CST,OFF	Stopping the embedded NTRIP caster	
CST,MTP,ADD	Adding/modifying mount points	
CST,MTP,DEL	Deleting a mount point	
CST,PAR	Setting embedded NTRIP caster parameters	
CST,RST	Resetting the embedded NTRIP caster	
CST,USR,ADD	Adding/modifying NTRIP caster users	
CST,USR,DEL	Deleting an NTRIP caster user	
	Reading the current NTRIP caster settings	CST
DIP,OFF	Terminating Direct IP connection	
DIP,ON	Establishing the Programmed Direct IP Connection	
DIP,PAR	Setting Direct IP parameters	
	Reading Direct IP settings	DIP
NTP	Network Time Controller server	.
NTR,LOD	Loading the NTRIP caster source table	
NTR,MTP	Connecting receiver to NTRIP caster mount point	.
NTR,PAR	Setting NTRIP parameters	
	Reading current NTRIP settings	NTR
	Reading source table stored in receiver	NTR,TBL

## Log & Alarms

Set Command	Description	Query Command
ALR,ACK	Acknowledging alarms	
	Listing current alarms	ALR
LOG,DEL	Deleting log files	
LOG,PAR	Setting the log file	.
	Editing a log file	LOG
	Listing log files	LOG,LST

## Memory & Data Recording

Set Command	Description	Query Command
DRD	Setting G-file duration	•
DRI	Setting raw data recording rate	•
FIL,CPY	Copying Files from Internal to External Memory	
FIL,DEL	Deleting Files and Directories	
FIL,GET	Downloading a File Thru the Current Port	
FIL,WRT	Writing a file	
	Reading information On G-file being recorded	FIL,CUR
	Listing files in receiver memory or USB key	FIL,LST
	Reading memory status	FIL,STS
FMT	Formatting internal memory	
MEM	Selecting memory device used	•
REC	Enable/disable, start/stop raw data recording	•
RFM	Enable/disable Ring File Memory	•

## RINEX Conversion

Set Command	Description	Query Command
RXC,PAR	Setting the Embedded RINEX Converter	
RXC,RUN	Converting a G-File into RINEX Files	

## Receiver Configuration

Set Command	Description	Query Command
CMD,LOD	Running a List of \$PASH commands	
CMD,WTI	Inserting Wait Times	
	Reading content of default configuration file	DFC
DFC,DEL	Deleting the default configuration file	
DFC,GET	Duplicating the default configuration file	
DFC,SET	Setting the default configuration file	
DFC,TST	Testing the default configuration file	
DIF,NET	Processing mode in network rover	•
OPTION	Installing a firmware option	•
	Installed firmware options & expiration dates	OPTION,EXP
PAR,LOD	Loading receiver configuration from PAR file	
PAR,SAV	Saving receiver configuration as PAR file	
	Reading the board's identification parameters	RID
UPL,PAR	Setting the FTP server providing firmware upgrades	
UPL,UPG	Upgrading the receiver firmware from FTP	
	FTP server providing firmware upgrades	UPL
	Editing the firmware upgrade log file	UPL,LOG
	Listing the firmware upgrades available on FTP	UPL,LST

Set Command	Description	Query Command
	Reading firmware version	VERSION

## UHF Radio

Set Command	Description	Query Command
RDP,PAR	Setting the radio	•
RDP,TYP	Defining the type of radio used	•
	Reading the radio channel settings	RDP,CHT
	Reading radio type used and radiated power	RDP,PWR

## Other Important Commands

Set Command	Description	Query Command
ATL	Controlling the recording of debug messages	•
INI	Resetting the board according to your preferences	
NME	Enabling/disabling NMEA messages	
NME,ALL	Disabling all NMEA and NMEA-like messages	
OUT	Suspending/resuming all periodic messages	•
OUT,ALL	Disabling all periodic messages	
PWR,OFF	Preparing the board before being turned off	
RST	Resetting the board parameters to their default values.	
	Reading the recording status	RCS



## Chapter 7. Set Command Library



This chapter provides a detailed description of all the \$PASHS commands applicable to the MB-Two board. The commands are listed in alphabetical order, irrespective of their use domain.

In this chapter, you will see the terms “board” and “receiver” mentioned repeatedly in the description of each command. Be aware the two terms are here strictly equivalent. They both designate the MB-Two.

### 3DF: Running Attitude Processing on Set Baselines

---

#### Function

This command is used to run the attitude process over all the baselines (up to three) you have defined using command \$PASHS,BLN. The antennas are in this case installed on a rigid platform, occupying fixed positions relatively to one another (“Steady” mode).

The command may also be used to compute heading from a single baseline of changing length (“Flex” mode). In Flex mode, no baseline length auto-calibration is required: Valid heading is delivered once baseline ambiguity has been resolved. This operating mode is quite similar to what you get with BRV.

The command can also be used to stop the attitude or heading process.

When a baseline engine is included in a 3DF process, it is assumed that the type of base it is working from is a moving one. Each of the baselines included in a 3DF process is a source of angle estimate and cannot be a source of position solution.

#### Command Format

#### Syntax

General:

```
$PASHS,3DF,ON,d1[,d2[,d3]][*cc]
```

“Flex” mode:

**\$PASHS,3DF,ON,d1,FLX[\*cc]**

To stop the attitude process:

**\$PASHS,3DF,OFF[\*cc]**

### Parameters

Parameter	Description	Range
d1	First baseline number	1, 2, 3
d2	Second baseline number	1, 2, 3
d3	Third baseline number	1, 2, 3
*cc	Optional checksum	*00-*FF

### Comments

The following combinations are possible today:

- Heading computed from baseline #2 (engine #2):  
**\$PASHS,3DF,ON,2**
- Full attitude computed from baselines #2 and #3 (engines #2 and #3):  
**\$PASHS,3DF,ON,2,3**
- Full attitude computed from baselines #2, #3 and #1 (engines #2, #3 and #1):  
**\$PASHS,3DF,ON,2,3,1**
- Flex attitude computed from baseline #1:  
**\$PASHS,3DF,ON,1,FLX**
- Flex attitude computed from baseline #2:  
**\$PASHS,3DF,ON,2,FLX**
- Flex attitude computed from baseline #3:  
**\$PASHS,3DF,ON,3,FLX**

Auto-calibration should be resumed or baseline vectors should be re-defined whenever you change the operating mode using this command.

For a smooth transition from steady to flex mode (and vice versa), it's a good practice to stop the attitude/heading mode temporarily using command **\$PASHS,3DF,OFF**.

## 3DF,ANG: Maximum Baseline Elevation

---

**Function** This command is used to define the maximum elevation possible for each baseline. In most cases, this amounts to setting the maximum pitch or/and roll angle possible.

### Command Format Syntax

`$PASHS,3DF,ANG,f1[*cc]`

### Parameters

Parameter	Description	Range	Default
f1	Maximum value of baseline elevation, in degrees.	0-45	15
*cc	Optional checksum	*00-FF	

### Example

Setting maximum baseline elevation to 30°:

`$PASHS,3DF,ANG,30*0F`

- Comments**
- If the elevation determined by the receiver for a baseline is greater than this preset limit, then the baseline is rejected from the attitude process.
  - The value you assign to the maximum permitted baseline elevation has no impact on the calibration process.

## 3DF,CLB: Running/Resetting Baseline Auto-Calibration

---

**Function** This command is used to auto-calibrate the attitude processing by calculating each of the vectors resulting from the antenna setup you implemented.

Depending on your antenna setup, you may have one (V12), two (V12+V13) or three (V12+V13+V14) vectors involved in the calibration.

Once determined, the calibration values will then be used for attitude determination.

### Command Format Syntax

`$PASHS,3DF,CLB[*cc]`

**Parameters**

None.

**Comments**

- It is your own responsibility to align or not the antenna setup with the vehicle's centerline and set offset values accordingly (see \$PASHS,3DF,OFFS).
- If some of the vectors have already known values (they were entered manually or calculated in a previous calibration), then running a new calibration will result in overwriting these known values with the results of the new calibration.
- If you run this command while a calibration is already in progress, this will reset/restart the calibration process.
- If this command is issued after the attitude mode has been turned off (\$PASHS,3DF,MOD,OFF), it is saved in memory and started automatically as soon as the attitude mode is turned back on.
- While auto-calibration is in progress, all attitude-related messages, such as HPR or HDT, will not deliver angles (the corresponding fields will all be empty), but some other fields may be populated.

## 3DF,MXB: Setting Baseline Length Tolerance

---

**Function**

This command is used to specify the accepted tolerance on the possible variations of baseline lengths.

When the attitude mode calculates a baseline length that exceeds the tolerance, that baseline is rejected from the attitude computation.

**Command Format****Syntax**

```
$PASHS,3DF,MXB,f1[*cc]
```

**Parameters**

Parameter	Description	Range	Default
f1	Baseline tolerance, in meters	0-1.0	0.02
*cc	Optional checksum	*00-*FF	

**Example**

Allowing the computed values of baseline lengths to vary by no more than 15 centimeters:



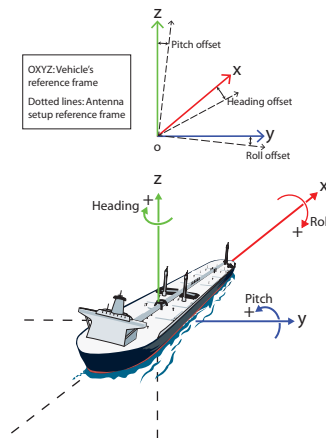
\$PASHS,3DF,MXB,0.15\*09

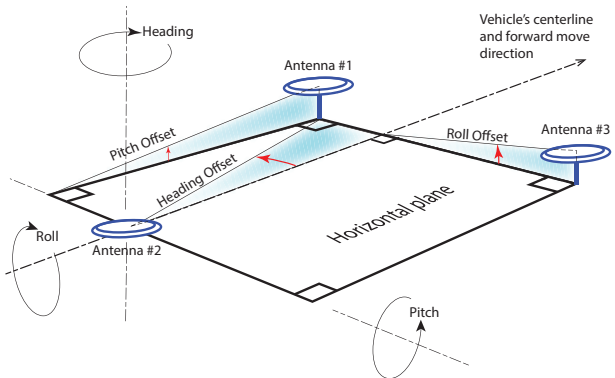
- Comments**
- Setting the baseline tolerance has no impact on the calibration process.
  - In most cases, the baseline length tolerance can be interpreted as the sum of a-priori uncertainty and admissible flexibility during operation.

## 3DF,OF5: Defining Heading-Pitch-Roll Offset Values

---

**Function** This command is used to give the orientation of the antenna setup with respect to the vehicle. By defining the offset values, you define the rotation matrix from the antenna setup to the vehicle's reference frame (OXYZ).





By default, the antenna setup is assumed to be aligned with the vehicle and all antennas are the same height (all offset values are 0 by default).

## Command Format Syntax

`$PASHS,3DF,OFS,f1,f2,f3[*cc]`

## Parameters

Parameter	Description	Range	Default
f1	Heading offset, in degrees. See also comment below.	$\pm 180^\circ$	0
f2	Pitch offset, in degrees. See also comment below.	$\pm 90^\circ$	0
f3	Roll offset, in degrees	$\pm 90^\circ$	0
*cc	Optional checksum	*00-*FF	

## Example

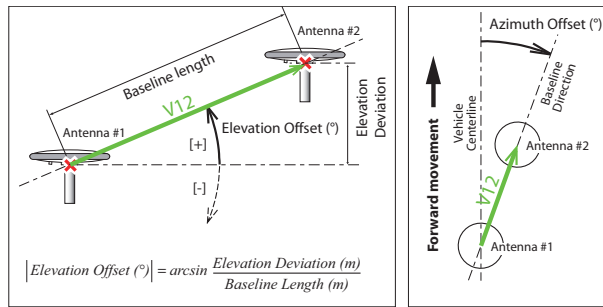
Entering a heading offset for the antenna setup:

`$PASHS,3DF,OFS,-12.5,0,0*2B`

## Comments

- Offset values apply to both auto-calibrated vectors and those entered manually via `$PASHS,3DF,Vx`
- With only one vector specified (`$PASHS,3DF,x`), fields f1 and f2 designate respectively the azimuth offset and

elevation offset of the vector (V12), but signs are opposite those .



## 3DF,RST: Resetting Attitude Computation and Calibration

**Function** This command is used to reset the carrier ambiguity for each baseline and forces each of them to re-initialize. You should therefore expect some short delay before attitude data are delivered, corresponding to the time elapsed before ambiguities are solved.

**Command Format** **Syntax**  
`$PASHS,3DF,RST[*cc]`

### Parameters

None.

- Comments**
- All settings made earlier using the other \$PASHS,3DF,... commands are not affected by this command.
  - This command has no effect on which GNSS signals are tracked.
  - This command is always valid ("ACKed") whether the attitude mode is enabled or not, but it has no effect if the attitude mode is currently off.
  - This command has no effect if it is issued while a calibration is in progress.

## 3DF,Vx: Entering Vector Components

---

**Function** This command is used to enter the components of each vector used in the attitude computation.

**Command Format** **Syntax**

```
$PASHS,3DF,V12,x12,y12,z12[*cc]
$PASHS,3DF,V13,x13,y13,z13[*cc]
$PASHS,3DF,V14,x14,y14,z14[*cc]
```

### Parameters

Parameter	Description	Range
V12	Designates first vector	-
x12,y12,z12	x,y,z components of first vector, in meters.	±999.999
V13	Designates second vector	-
x13,y13,z13	x,y,z components of second vector, in meters.	±999.999
V14	Designates third vector	-
x14, y14, z14	x,y,z components of second vector, in meters.	±999.999
*cc	Optional checksum	*00.*FF

### Example

Entering vector components for first vector:

```
$PASHS,3DF,V12,0.8,0.3,0*2A
```

### Comments

- Depending on how you configured the attitude mode (see \$PASHS,3DF,CFG), you will be required to enter the components of one vector (V12), two vectors (V12+V13) or three vectors (V12+V13+V14) before they can actually be used in attitude mode.
- With a single vector (V12), remember this vector should be roughly aligned with the vehicle's centerline to allow computation of heading and pitch, or roughly perpendicular to the vehicle's centerline to allow computation of heading and roll.  
If you don't follow these recommendations, no attitude solution will be delivered.
- Vector components should be expressed in the reference frame of the antenna setup. If you are using non-zero offset values, be aware these components will be different if expressed in the vehicle's reference frame.
- This command will be ignored if calibration is in progress.

## ALR,ACK: Acknowledging Alarms

---

**Function** This command is used to acknowledge the alarms (alerts) raised by the receiver. When this command is issued, all the reported alarms are removed. Query command \$PASHQ,ALR will then stop reporting any alarm until a new one is set.

**Command Format**    **Syntax**  
                           \$PASHS,ALR,ACK[\*cc]

**Parameters**

None.

**Example**

Acknowledging all alarms:

\$PASHS,ALR,ACK

**Query Command**    \$PASHQ,ALR

## ANH: Antenna Height

---

**Function** This command allows you to enter the antenna height (vertical measurement only). See \$PASHS,ANT for slant measurement).

**Command Format**    **Syntax**  
                           To define the height of the primary antenna (main antenna or antenna #1):  
                           \$PASHS,1,ANH,f1[,c2][\*cc]  
                           or you can simply skip the second term ("1"):  
                           \$PASHS,ANH,f1[,c2][\*cc]  
                           To define the height of antenna #2:  
                           \$PASHS,2,ANH,f1[,c2][\*cc]

## Parameters

Parameter	Description	Range
f1	Antenna height.	0-6.553 m 6.553-99.999 m
c2	Antenna height measurement type ("V" for Vertical) c2 may be omitted. The entered value will always be seen as a vertical measurement)	V
*cc	Optional checksum	*00-*FF

## Example

Entering the vertical measurement (2 m) of a rover antenna:

**\$PASHS,ANH,2.000**

## Comments

When you enter an antenna height greater than 6.553 m, be aware this will NOT be the value of antenna height broadcast through RTCM messages and the one saved to the G-file. Instead, a fixed value of 6.553 meters will be used.

## Related Commands

\$PASHQ,ANH (query command)

\$PASHS,ANR

\$PASHS,ANT

## ANP,OUT: Defining a Virtual Antenna

---

### Function

This command allows you to specify the name of an antenna that raw data will be adjusted to. By specifying the name of a virtual antenna, you ask the receiver to correct ("reduce") the raw and differential data it generates from the received GNSS signals to make them available as if they had been received through *that* antenna.

### Command Format

#### Syntax

**\$PASHS,ANP,OUT,s1[\*cc]**

## Parameters

Parameter	Description	Range
s1	Virtual antenna name (case-sensitive) Blank or "OFF" to specify that no virtual antenna is used.	31 characters max., blank or OFF
*cc	Optional checksum	*00-*FF

## Examples

Setting the ADVNULLANTENNA as the virtual antenna:

```
$PASHS,ANP,OUT,ADVNULLANTENNA*73
```

Disabling the use of a virtual antenna:

```
$PASHS,ANP,OUT,OFF*2B
```

## Comments

- By default, the receiver observables are not corrected for the type of GNSS antenna used. It's only by providing separately the name of the GNSS antenna used (declared as the OWN antenna) that the antenna corrections can be performed when processing the receiver observables. Now precisely, the ANP,OUT command allows you to directly generate the raw and differential observables for the type of antenna you specify in the command (e.g. ADVNULLANTENNA).
- Be aware that the raw data reduction process is possible only if the name of the antenna physically used by the receiver has been specified through the \$PASHS,ANP,OWN command and declared in the receiver's antenna database as one of the default or user-defined antennas. Otherwise, the command will be NAKed.
- Raw data reduction will not be performed on data from any satellite located below the elevation mask.
- When raw data reduction is effective, any antenna name messages generated by the receiver will include the name of the virtual antenna, and not the antenna serial number or the setup ID.
- Antenna reduction is performed in such a way that the ARP is unchanged. If the reference position is given with respect to the ARP, and not to the L1 phase center, then the receiver computes the position of the ARP using the physical parameters of the antenna, and then re-computes the position of the L1 phase center according to the ANP,OUT antenna parameters. This guarantees that the reported reference position, the antenna name and the observables are all consistent with one another.

- With a dual sensor (see \$PASHS,SNS,DUO), the command will affect the two antennas, provided each of them has a known physical name, otherwise the command is NAKed.

**Related Commands** \$PASHQ,ANP - \$PASHQ,ANP,OUT (query commands)  
\$PASHS,ANP,OWN

## ANP,OWN: Naming Local Antennas

---

**Function** This command is used to enter the names of the antennas connected to the receiver (local antennas).

### Command Format Syntax

Naming local antenna #1 (main antenna):

\$PASHS,1,ANP,OWN,s1[,s2[,d3]][\*cc]

or

\$PASHS,1,ANP,OWN,s1,,d3[\*cc]

or

\$PASHS,ANP,OWN,s1[,s2[,d3]][\*cc]

or

\$PASHS,ANP,OWN,s1,,d3[\*cc]

Naming local antenna #2:

\$PASHS,2,ANP,OWN,s1[,s2[,d3]][\*cc]

or

\$PASHS,2,ANP,OWN,s1,,d3[\*cc]

### Parameters

Parameter	Description	Range
s1	User-defined antenna name (case-sensitive). There is no default name (s1 empty). The command will be "NAKed" if s1 consists of more than 31 characters.	31 characters max.
s2	Antenna serial number	31 characters max.
d3	Antenna setup ID	0-255
*cc	Optional checksum	*00-*FF

### Comments

- Antenna names must be chosen to be consistent with the built-in antenna database, which is a hard-coded database. The firmware has the capability to extract



numerical values from the parameters entered under a given antenna name.

- The active antenna input (whether antenna connector 1 or antenna connector 2) can be specified externally, or detected automatically. In both cases, the firmware is informed of the antenna input providing the GNSS signal and, as a result, of the name of the antenna providing the signal
- Parameters s2 and d3 have little interest for a rover (that's why they are optional). If however they are specified, they should be inserted in such RTCM messages as type 1008 or 1033, in which room is reserved for these parameters.
- With the receiver used as an RTK base, the s1 parameter (and also the s2 and d3 parameters if available) are inserted into antenna name messages (e.g. RTCM 23 or RTCM 1007, 1008 or 1033). If needed, the receiver performs the transformation of the entered base position from ARP to APC, or vice versa.
- With the receiver used as an RTK rover, the numerical values corresponding to the s1 parameter are used to appropriately correct the local antenna data.

### Example

Entering "ASH111661" as the name of the receiver antenna:

```
$PASHS,ANP,OWN,ASH111661*26
```

## ANP,REF: Naming the Antenna Used at the Base

---

### Function

This command is used to enter the name of the antenna used by the base the receiver is working with.

Using this command only makes sense to name the antenna of the base defined as the base data source for the first baseline (i.e. as defined using \$PASHS,1,BLN,...).

### Command Format

#### Syntax

```
$PASHS,ANP,REF,s1[*cc]
```

## Parameters

Parameter	Description	Range	Default
s1	User-defined antenna name (case-sensitive).	31 characters max.	UNKNOWN
*cc	Optional checksum	*00-*FF	

## Comments

- The antenna name you enter through this command should be strictly the same as the one specified in the receiver built-in, hard-coded antenna database, otherwise the receiver won't be able to make the appropriate processing relative to this antenna.
- When used as a rover, the receiver will correct the received reference data, using the physical parameters saved for the antenna whose name was entered through this command.
- The s1 parameter will be ignored if the incoming reference data include the name of the base antenna used (even if this name is unknown or a blank field).

## Example

Entering "ASH802147" as the name of the base antenna:  
**\$PASHS,ANP,REF,ASH802147**

**Related Commands**    \$PASHQ,ANP,REF (query command)  
 \$PASHQ,ANP

## ANR: Antenna Reduction Mode

---

**Function**    This command is used to define the exact location for which the receiver (a base or rover) computes a position (your choice applies to both antennas, if your application uses two).

**Command Format**    **Syntax**  
**\$PASHS,ANR,s1[\*cc]**

## Parameters

Parameter	Description	Range	Default
s1	Antenna reduction mode: <ul style="list-style-type: none"> <li>• <b>OFF</b> or <b>PC1</b>: Position tagged to the L1 phase center.</li> <li>• <b>ON</b> or <b>SPT</b>: Position tagged to the survey point (ground mark).</li> <li>• <b>ARP</b>: Position tagged to the Antenna Reference Point.</li> </ul>	OFF, PC1, ON, SPT, ARP	OFF
*cc	Optional checksum	*00-*FF	

## Example

Setting the antenna reduction mode to ARP:

**\$PASHS,ANR,ARP\*47**

## Comments

- Internally, the receiver will always compute a position tagged to the antenna's L1 phase center (PC1). Using this command, you may ask the firmware to make the necessary transformation so that the position delivered be tagged to another point:
  - The transformation from PC1 to ARP or ARP to PC1 relies on the parameters entered through the \$PASHS,ANP,OWN command.
  - The transformation from ARP to SPT or SPT to ARP relies on the parameters entered through the \$PASHS,ANT/ANH command.

Changes made “on-line” using the \$PASHS,ANP,OWN or \$PASHS,ANT/ANH commands will cause the position delivered to leap.
- The messages that are affected by this command are the following:
  - All NMEA messages
  - All NMEA-like messages
  - ATOM,PVT message. The MIS block in the ATM,PVT message will tell you the exact location the position is tagged to.
- The messages that are NOT affected by this command are the following:
  - ATM,RNX
  - All RTCM-2 position messages
  - All RTCM-3 position messages
  - CMR and CMR+ messages

- Changing the antenna reduction mode setting (s1) “on-line” will NOT reset the PVT engine. You should therefore be aware that in this case, the position will jump noticeably, and most notably the altitude, which may jump by as much as a few meters.

**Related Commands** \$PASHQ,ANR (query command)  
 \$PASHS,ANH  
 \$PASHS,ANT

## ANT: Antenna Height

---

**Function** This command is used to define the antenna height, especially when it was determined using the slant measurement method. However, a vertical measurement can also be entered through this command.  
 Using the \$PASHS,ANT command overwrites all previous settings performed with the \$PASHS,ANH command.

### Command Format Syntax

To define the height of antenna #1 (main, or primary, antenna):

**\$PASHS,ANT,1,f1,f2,f3[\*cc]**

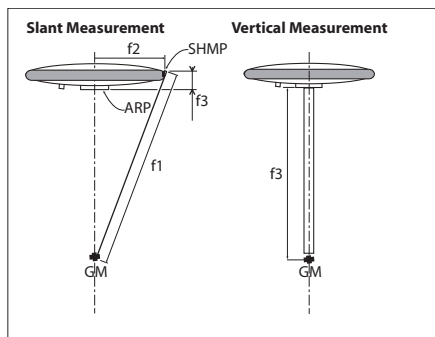
or

**\$PASHS,ANT,f1,f2,f3[\*cc]**

To define the height of antenna #2:

**\$PASHS,ANT,2,f1,f2,f3[\*cc]**

### Diagrams and Definitions



- ARP: Antenna Reference Point (usually bottom of the antenna).
- SHMP: Slant Height Measurement Point (usually at the hedge of the antenna, above the ARP).
- Ground Mark (GM): above the ARP (same horizontal coordinates).

### Parameters

Parameter	Description	Range
f1	Slant height measurement, from ground mark (GM) to antenna edge (SHMP).	0-6.553 m 6.553-99.999 m
f2	Antenna radius: horizontal distance from the geometrical center to the antenna edge.	0-6.553 m
f3	Vertical offset: <ul style="list-style-type: none"> <li>• From ARP to SHMP, if radius and slant height are not null.</li> <li>• From Ground Mark to ARP, if radius and slant height are null.</li> </ul>	0 to ±6.553 m 6.553-99.999 m
*cc	Optional checksum	*00-*FF

### Examples

Entering the vertical measurement (2 m) (antenna #1):

```
$PASHS,ANT,0,0,2.000*2E
```

Entering a slant measurement (1.543 m) (antenna #1):

```
$PASHS,ANT,1.543,0.0980,-0.0400*07
```

### Comments

- When you enter an antenna height greater than 6.553 m (see f1 and f2 above), be aware this will NOT be the value of antenna height broadcast through RTCM messages and the one saved to the G-file. Instead, a fixed value of 6.553 meters will be provided.
- The vertical height from ARP to ground mark can also be entered through the ANT command, which in this case should be used as follows:
  - Set **f1** and **f2** to “0.0”
  - Enter the antenna height from ARP to ground mark as **f3**. Only when **f1=f2=0.0** can you define **f3** this way.
  - **f3** is negative when the ARP is below the SHMP.

### Related Commands

```
$PASHQ,ANT (query command)
$PASHS,ANH
$PASHS,ANR
```

## ATH,PWD: Defining the Startup Password

---

**Function** This command is used to define the startup password.

**Command Format Syntax**  
**\$PASHS,ATH,PWD,s[\*cc]**

### Parameters

Parameter	Description	Default	Range
s	Password	"empty"	6 to 64 characters. The following characters are allowed: A-Z, a-z, 0-9
*cc	Optional checksum		*00-*FF

### Example

Entering new startup password "Theft125":

```
$PASHS,ATH,PWD,Theft125
```

**Related Commands** \$PASHQ,ATH (query command)

## ATL: Debug Messages

---

**Function** This command allows you to:

- Enable or disable the recording of ATL data. ATL data are written to a file named *ATL\_<yymdd\_hhmmss>.log*. The file is saved to the memory you last selected through the \$PASHS,MEM command.
- Output ATL data via the specified port.

**Command Format Syntax**  
 Enabling/disabling ATL data recording:  
**\$PASHS,ATL,s1[f3][,SCN,d4][,MEM,d5](MDR,d6)[\*cc]**

Sending ATL data through a port:

```
$PASHS,ATL,d2[f3][,SCN,d4][*cc]
```

## Parameters

Parameter	Description	Range	Default
s1	Enabling/disabling ATL data recording: <ul style="list-style-type: none"> <li>• ON: Enable (recording does not restart after power cycle)</li> <li>• AUT: Enable (record will restart automatically after power cycle)</li> <li>• OFF: Disable</li> </ul>	ON, AUT, OFF	OFF
d2	Output port: <ul style="list-style-type: none"> <li>• Serial port (A, B, D)</li> <li>• USB serial port (C)</li> <li>• TCP/IP port (I, F)</li> </ul>	A-D, I, F	
f3	Output interval, in seconds	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1	1
d4	Configuration index	0, 1	0
d5	Memory where ATL data are recorded: <ul style="list-style-type: none"> <li>• 0: Internal memory</li> <li>• 1: SD Card</li> <li>• 2: USB memory</li> </ul> <p>If this parameter is not specified, ATL data are recorded in the memory specified by \$PASHS, MEM.</p>	0-2	0
d6	Maximum duration, in minutes (0: unlimited)	0, 15, 20, 30, n x 60 with n between 1 and 24	0
*cc	Optional checksum	*00-*FF	

## Examples

Recording ATL data to memory is only possible if the [R] firmware option is installed.

First choose the memory [internal memory (0) or USB memory (2)] where to save ATL data (e.g. to USB memory):

```
$PASHS, MEM, 2
```

Then enable ATL data recording to the chosen memory:

```
$PASHS, ATL, ON
```

ATL data can be output independently via a port through the following command (e.g. ATL data output on port C):

```
$PASHS, ATL, C
```

## Comments

- The command (first syntax) will be ACKed even in the case where the selected memory is unavailable at the time of request. Please use \$PASHQ, ATL to read the ATL data status in that case.

- The command (first syntax) will be NAKed if you are trying to record ATL data while these are being sent to a port.
- Conversely, the command (second syntax) will be NAKed if you are trying to output ATL data through a port while ATL data are being written to memory.
- You may customize ATL data recording when setting it to ON or AUT. If a file named “atl.ini” is found on the SD card, then the receiver will run the commands found in this file rather than running the default recording command. Please note that the last command in the “atl.ini” file should be followed by the <cr><lf> characters to be seen as valid.
- You don’t normally have to use this command but Technical Support may ask you to do so if a problem occurs and they need to analyze the resulting log file to fix the problem.

The content of the file can only be analyzed by Technical Support as ATL data files use a proprietary, undisclosed data format, which in addition is subject to change without notice.

**Related Commands**    \$PASHQ,ATL (query command)  
                               \$PASHS,MEM

## ATM: Enabling/Disabling ATOM Messages

---

**Function**    This command allows you to enable or disable ATOM messages on the specified port. For more details about the ATOM format, please refer to the *ATOM Reference Manual*.

**Command Format**    **Syntax**  
                               \$PASHS[,d0],ATM,s1,c2,s3[,f4][\*cc]  
                               \$PASHS[,d0],ATM,RNX,c2,s3[,f4],&SCN,d5  
                               \$PASHS[,d0],ATM,s1,c2,s3[,f4],&s7[,s8]...



## Parameters

Parameter	Description	Range
d0	Sensor ID (antenna ID; 1: primary, 2: secondary); 1 if omitted.	1, 2
s1	ATOM message type	See table below
c2	Port ID: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> <li>• M: Internal memory</li> <li>• U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
s3	Enable (ON) or disable (OFF) this ATOM message type.	ON, OFF
f4	Output rate for PVT, RNX and SUP messages, in seconds.	0.02, 0.05, 0.1, 0.2, 0.5, 1-6, 10, 12, 15, 20, 30, 60, 120, etc. integer minute, up to 960 seconds
	Output rate for NAV, ATR and STA messages, in seconds.	1-999 s
d5	Scenario for ATOM RNX	See list of scenarios below
s7, s8, etc.	Sub-message	See ATOM Reference Manual
*cc	Optional checksum	*00-*FF

### ATOM Messages (preliminary list)

Data	ATOM Number	Description	Default Sub-messages	Default Output rate
ALR	0	Receiver alarms	USR	N/A
SUP	1	Supplementary data	CPI	1 second
PVT	3	Positioning results	COO, ERR, LCY, SVS	1 second
ATR	4	Receiver attributes	ANM, RNM, CPB, AOP	30 seconds
NAV	5	Navigation information	EPH, GIT, GFT	300 seconds
DAT	6	Binary data frames	EXT, FRM	N/A
RNX	7	Receiver observables	SCN,0	1 second
STA	13	Receiver status	BA,DDS,GFN	5 seconds
EVT	14	Receiver event	TTT, PTT	N/A

### Default Settings on Ports M and U for Raw Data Recording:

Data	ATOM Number	Description	Default Sub-messages	Default Output rate
PVT	3	Positioning results	COO, ERR, LCY	1 second
ATR	4	Receiver attributes	ANM, RNM, AOP, CPB	30 seconds

Data	ATOM Number	Description	Default Sub-messages	Default Output rate
NAV	5	Navigation information	ALM, EPH, GIT, GFT	300 seconds
RNX	7	Receiver observables	SCN,0	1 second

### ATM RNX Scenarios

Scenario	Description
0	All available raw data in full presentation, fully computed reference position follows each epoch ( <b>this scenario is not recommended as differential protocol</b> ).
1	L1 pseudo-range and carrier phase in full presentation, extended fixed position follows every 12 epochs.
2	L1 SNR, pseudo-range and carrier phase in full presentation, extended fixed position follows every 12 epochs.
3	L1 & L2 pseudo-range and carrier phase in full presentation, extended fixed position follows every 12 epochs.
4	L1 & L2 SNR, pseudo-range and carrier phase in full presentation, extended fixed position follows every 12 epochs.
100	L1 & L2 compact pseudo-range and full carrier phase, extended fixed position follows every 12 epochs, all data are decimated in 5 times compared to L1 carrier phase.
101	L1 & L2 compact pseudo-range and compact carrier phase, extended fixed position follows every 12 epochs, all data are decimated in 5 times compared to L1 carrier phase. <b>Cannot be used with moving receiver.</b>
201	Same as scenario 1, but extended computed reference position follows each epoch.
202	Same as scenario 2, but extended computed reference position follows each epoch.
203	Same as scenario 3, but extended computed reference position follows each epoch.
204	Same as scenario 4, but extended computed reference position follows each epoch.
300	Same as scenario 100, but extended computed reference position follows each epoch.

Scenarios 1 to 300 are used as differential messages sent to ports A-D, F, I, P, Q. These are output only with the receiver used in base mode (not in rover mode). However, they can be sent to port M or U for data recording regardless of whether the receiver is used as a base or a rover.

### Example

Enabling ATOM message type RNX on serial port A at a 1-second output rate:

```
$PASHS,ATM,RNX,A,ON,1.0,&SCN,0*4E
```

**Related Commands** \$PASHQ,PAR,ATM (query command)  
\$PASHS,ATM,ALL

## ATM,ALL: Disabling All ATOM Messages

---

**Function** This command disables all ATOM messages currently enabled on the specified port.

**Command Format** **Syntax**  
\$PASHS[,d0],ATM,ALL,c1,OFF[\*cc]

### Parameters

Parameter	Description	Range
d0	Sensor ID (antenna ID; 1: primary, 2: secondary); 1 if omitted.	1, 2
c1	Port routing the ATOM message(s) you want to disable. <ul style="list-style-type: none"> <li>A, B, D: Serial port</li> <li>C: USB serial port</li> <li>I, P, Q, F: TCP/IP stream</li> <li>M: Internal memory</li> <li>U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
*cc	Optional checksum	*00-*FF

### Example

Disabling all ATOM messages on port A:

```
$PASHS,1,ATM,ALL,A,OFF*53
```

**Related Commands** \$PASHS,ATM

## BDS: BeiDou Tracking

---

**Function** This command is used to enable or disable BeiDou tracking. The command is valid only if the [B] option has been activated in the receiver.

**Command Format Syntax****\$PASHS,BDS,s1[\*cc]****Parameters**

Parameter	Description	Range	Default
s1	Enables (ON) or disables (OFF) BeiDou tracking.	ON, OFF	ON
*cc	Optional checksum	*00-*FF	

**Example**

Enabling BeiDou:

**\$PASHS,BDS,ON****Related Commands**

\$PASHQ,BDS (Query command)

\$PASHS,SBA

\$PASHS,GPS

\$PASHS,GAL

\$PASHS,QZS

\$PASHS,GLO

## BLN: Defining Baselines and their Base/Rover Data Sources

---

**Function** This command is used to define all the baselines that needs to be processed in your application.

By default, a baseline that you define and enable is a *primary* baseline. If you create it that way, a baseline may also be a *supplementary* baseline, therefore coming as a “supplement”, or support, to a primary baseline.

Whatever their type (primary or supplementary), every baseline should be defined with a source of base data and a source of rover data.

A supplementary baseline uses the same sources of base/rover data as the primary baseline. Only its baseline number is different. It is created just for the sake of cross-checking its results with those of the primary baseline (which should be very similar in theory).

This command is also used to deactivate (disable) an existing baseline.

### Command Format **Syntax**

General Case:

```
$PASHS,d1,BLN,X,c2[,c3][*cc]
```

To define and enable baseline “d1”:

```
$PASHS,d1,BLN,ON,c2[,c3][*cc]
```

To enable the one baseline (will necessarily be baseline #1) you are using in your application:

```
$PASHS,BLN,ON,c2[,c3][*cc]
```

To disable a baseline (the corresponding baseline engine is stopped right away):

```
$PASHS,d1,BLN,OFF[*cc]
```

To disable all baselines:

```
$PASHS,,BLN,OFF[*cc]
```

## Parameters

Parameter	Description	Range
d1,BLN	Baseline number. Particular cases of use: <ul style="list-style-type: none"> <li>d1 skipped: The command refers to baseline #1</li> <li>d1 empty: The command applies to all existing baselines</li> </ul>	<ul style="list-style-type: none"> <li>1,BLN or 2,BLN or 3,BLN</li> <li>Skipped,</li> <li>Empty</li> </ul>
X	Baseline definition and processing control: <ul style="list-style-type: none"> <li>ON: Enables processing of considered baseline (as primary baseline).</li> <li>OFF: Disables processing of considered baseline.</li> <li>SAM: Defines considered baseline as supplementary and enables processing of this baseline.</li> </ul>	ON, OFF, SAM
c2	Base data source: <ul style="list-style-type: none"> <li>A, B, C, etc. : A board's physical or virtual port through which external data enter the board.</li> <li>"?" : Automatic selection of the board's physical port through which external data enter the board.</li> <li>1, 2, 3, etc: Sensor internal to the board. The figure indicates the number of the sensor used.</li> <li>A1, A2, B1, B2, etc.: Multi-board configuration. The letter indicates the local board's port to which an external multi-antenna board is connected, and the figure indicates the number of the sensor (on that external board) that is the source of data to be processed.</li> <li>?1, ?2, etc.: Multi-board configuration. The "?" symbol indicates that there is an automatic selection of the local board's physical port through which data enter the local board, and the figure indicates the number of the sensor (on the external board) that is the source of data to be processed.</li> </ul>	<ul style="list-style-type: none"> <li>A, B, C, etc.</li> <li>?</li> <li>1, 2, 3, etc.</li> <li>A1, A2, B1, B2, etc.</li> <li>?1, ?2, etc.</li> </ul>
c3	Same as c2 above, but for rover data source.	Same as c2
*cc	Optional checksum	*00-*FF

### Comments

- You can define a maximum of three baselines.
- Baseline #1 can only be a primary baseline

- The normal step before setting baselines is to configure your application using commands \$PASHS,RTK (for RTK position), \$PASHS,BRV (for baseline/vector) or \$PASHS,3DF (for attitude).
- This command does not support CMRx data sources (RTK correcting data, physical station data, VRS data). Conversely, any baseline engine set to process CMRx data cannot take part in an RTK, BRV or 3DF application.

**Examples**    Creating (primary) baseline #2 using port A as base data source and B as rover data source:

```
$PASHS,2,BLN,ON,A,B
```

Disabling all existing baselines:

```
$PASHS,,BLN,OFF
```

## BRD: Enabling/Disabling the RTK Bridge Function

---

**Function**    This command is used to control the RTK Bridge function. Its use is required only in a board in charge of forwarding its RTK corrections (received via modem or WiFi) to other nearby boards through a radio transmitter.

**Command Format**    **Syntax**

```
$PASHS,BRD,s1[,c2][,c3][*cc]
```

## Parameters

Parameter	Description	Range	Default
s1	Controls the availability of RTK corrections on the specified output port: <ul style="list-style-type: none"> <li>• OFF: No RTK corrections forwarded to the output port.</li> <li>• ON: RTK corrections forwarded to the output port.</li> </ul>	ON, OFF	OFF
c2	Input port ID (port on which RTK corrections are received): <ul style="list-style-type: none"> <li>• Serial port: A, B, D</li> <li>• USB serial port: C</li> <li>• TCP/IP stream: P, Q</li> </ul>	A, B, D, C, P, Q	P
c3	Output port ID (port on which RTK corrections are made available): <ul style="list-style-type: none"> <li>• Serial port: A, B, D</li> <li>• USB serial port: C</li> </ul>	A, B, D, C	D
*cc	Optional checksum	*00-FF	

## Examples

Enabling RTK Bridge with input port P and default output port D:

```
$PASHS,BRD,ON,P*70
```

Disabling RTK Bridge:

```
$PASHS,BRD,OFF*42
```

**Relevant Query Command**    \$PASHQ,BRD

**See also**    \$PASHS,NTR  
 \$PASHS,DIP  
 \$PASHS,RDP,TYP  
 \$PASHS,RDP,PAR  
 \$PASHS,CPD,REM

## BRV: Relative Processing

---

**Function**    This command is used to set up the relative mode in the board. Use this command typically when you want to determine the vector originating from a moving base. The command can also be used to stop this process. When a baseline engine is included in a BRV process (“BRV” for Base-Rover Vector), it is assumed that the type of base it



is working from is a moving one. Each of the baselines included in a BRV process cannot be a source of position solution.

## Command Format Syntax

```
$PASHS,BRV,ON,d1[,d2[,d3]][*cc]
$PASHS,BRV,OFF[*cc]
```

## Parameters

Parameter	Description	Range
d1	First baseline number	1, 2, 3
d2	Second baseline number	1, 2, 3
d3	Third baseline number	1, 2, 3
*cc	Optional checksum	*00-*FF

## Comments

To date, only the following combinations are supported:

Vector computed by baseline engine 1 (baseline 1):

```
$PASHS,BRV,ON,1
```

Vector computed by baseline engine 2 (baseline 2):

```
$PASHS,BRV,ON,2
```

Vector computed by baseline engine 3 (baseline 3):

```
$PASHS,BRV,ON,3
```

## BRV,RST: Resetting Relative Positioning Mode

---

**Function** This command is used to reset the process taking place in all baseline engines delivering a solution of the BRV type. This command is also applicable to all baseline engines serving an application of the BRV type.

## Command Format Syntax

```
$PASHS,BRV,RST[*cc]
```

## Parameters

None.

## Comments

- This command does not apply to engines delivering solutions of the 3DF or RTK type.

- Resetting baseline engines means resetting the current estimates of single-difference (SD) carrier ambiguities for all processed signals.
- Just after issuing this command, you should expect jumps on baseline estimation, increased baseline RMS and temporary float solution status.

## CAN,OFF: Disabling the CAN Port

---

**Function** This command is used to disable the CAN port. By default the CAN port is on.

**Command Format** **Syntax**  
`$PASHS,CAN,OFF[*cc]`

### Parameters

None.

### Example

Disabling the CAN port:

```
$PASHS,CAN,OFF
```

**See also** \$PASHQ,CAN (relevant query command)  
 \$PASHS,CAN,ON  
 \$PASHS,CAN,PAR

## CAN,ON: Enabling the CAN Port

---

**Function** This command is used to enable the CAN port. By default the CAN port is on.

**Command Format** **Syntax**  
`$PASHS,CAN,ON[*cc]`

### Parameters

None.

### Example

Enabling the CAN port:

**\$PASHS,CAN,ON**

**See also** \$PASHQ,CAN (relevant query command)  
 \$PASHS,CAN,OFF  
 \$PASHS,CAN,PAR

## CAN,PAR: Setting the CAN Parameters

---

**Function** This command is used to define the parameters of the CAN bus. These are saved in permanent memory and so are not affected by commands \$PASHS,RST and \$PASHS,INI.

### Command Format Syntax

**\$PASHS,CAN,PAR[,BRT,d1][,FMT,d2][\*cc]**

#### Parameters

Parameter	Description	Range	Default
BRT,d1	CAN bus bit rate (in kbits/second): <ul style="list-style-type: none"> <li>• 0: 62.5</li> <li>• 1: 125</li> <li>• 2: 250</li> <li>• 3: 500</li> <li>• 4: 1000</li> </ul>	0-4	2
FMT,d2	Data format: <ul style="list-style-type: none"> <li>• 0: Raw data</li> <li>• 1: NMEA 2000</li> </ul>	0-1	1
*cc	Optional checksum	*00-*FF	

#### Example

Setting the CAN bus to use a bit rate of 500 kbits/s and handle NMEA 2000 data:

**\$PASHS,CAN,PAR,BRT,3,FMT,1**

**See also** \$PASHQ,CAN (relevant query command)  
 \$PASHS,CAN,ON  
 \$PASHS,CAN,OFF

## CMD,LOD: Running a List of \$PASH Commands

---

**Function** This command is used to run the complete list of \$PASH commands stored in a file found on USB memory. This implies that the file (in text editable format) should have first been saved to that medium.

### Command Format Syntax

```
$PASHS,CMD,LOD[,s][*cc]
```

### Parameters

Parameter	Description	Range	Default
s	File name. If s is omitted, it is assumed that the file to be run is "autoconfig.cmd".	255 characters max.	autoconfig.cmd
*cc	Optional checksum	*00-*FF	

### Examples

Running the serial commands in autoconfig.cmd:

```
$PASHS,CMD,LOD*54
```

Running the serial commands in a file named "myconfig.cmd":

```
$PASHS,CMD,LOD,myconfig.cmd*02
```

Example of log file:

```
314 09:11:07 > $PASHQ,RID
314 09:11:07 < $PASHR,RID,PF,30,S525Gh23,FKSZP---,,200913024*37
314 09:11:07 > $PASHQ,POS
314 09:11:07 < $PASHR,POS,1,14,091107.80,4717.938167,N,
00130.543280,W,86.457,3,0,0,0,-0,0,1,2,0,8,0,9,0,6,Gh23*20
314 09:11:07 > $PASHS,NME,GGA,A,ON
314 09:11:07 < $PASHR,ACK*3D
```

### Comments

- The file may contain any \$PASHS or \$PASHQ commands.
- If the file contains the \$PASHS,REC or \$PASHS,INI command, this command will always be run last, whatever its position in the file.
- All data lines returned by the receiver in response to the executed commands are written to a log file named as follows:

```
<command_file_name>.log
```

- To insert an idle wait time of several seconds between any two \$PASH commands, you can insert a specific command named \$PASHS,CMD,WTI between these two commands. The \$PASHS,CMD,WTI command may be inserted as many times as necessary in the file.
- Naming the command file “autoconfig.cmd” or “uploadconfig.cmd” will allow the receiver to automatically start the execution of all the commands stored in the file once it is detected by the receiver. The difference between the two file names is in the need for a user confirmation before running the file: “autoconfig.cmd” will require one, not “uploadconfig.cmd”.

**Related Commands**    \$PASHS,CMD,WTI

## CMD,WTI: Inserting Wait Times

---

**Function**    This command can be inserted one or more times in the list of \$PASH commands run with the CMD,LOD command. When running this command, in fact the receiver inserts a wait time of the requested value in the execution of the \$PASH commands.

**Command Format**    **Syntax**

\$PASHS,CMD,WTI,d[\*cc]

### Parameters

Parameter	Description	Range
d	Wait time generated by the command, in seconds.	1-3600
*cc	Optional checksum	*00-*FF

### Example

The command line below inserted in a command file will generate a 10-s wait time when executed:

\$PASHS,CMD,WTI,10\*74

**Comments**    This command will be interpreted by the receiver only if found in a command file.

**Related Commands**    \$PASHS,CMD,LOD

## CMR: Enabling/Disabling CMR or CMR+ Messages

---

**Function**    This command is used to enable or disable the continuous output of CMR/CMR+ messages on the specified port.

**Command Format**    **Syntax**

`$PASHS,CMR,d1,c2,s3[,f4][*cc]`

### Parameters

Parameter	Description	Range	Default
d1	Data message	See table below	-
c2	Port ID: • A, B, D: Serial port • C: USB serial port • I, P, Q, F: TCP/IP stream	A-D, F, I, P, Q	-
s3	Enabling/disabling command	ON, OFF	OFF
f4	Output interval, in seconds	See table below.	1
*cc	Optional checksum	*00-*FF	-

The list of supported data messages is the following:

Data	Description	f4 Range
0	GPS observations	0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, etc. integer minutes, up to 960.
1	Reference WGS84 position (location) tagged to L1 phase center	0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, etc. integer seconds, up to 999.
2	Reference site description (as entered through \$PASHS,MSG)	0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, etc. integer seconds, up to 999.
3	GLONASS observations	0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, etc. integer minutes, up to 960.

Data	Description	f4 Range
10	GPS observation and position/description scrolling message. It can be requested in place of message 0, 1 or 2 thereby transforming the message into a "CMR+" formatted message.	Any period can be formally requested. However, the firmware has hard-coded the message with an output interval equal to 7 times the output interval of message 0.
X	Encrypted CMRx SBL message. Its content actually dominates the content of messages 0,1,2,3,10. So there is no need to request any other CMR message if CMR,X is requested	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 sec.

### Example

Setting default CMR messages at a base:

```
$PASHS,POS,<coordinates of position>
```

or

```
$PASHS,POS,CUR*51
```

```
$PASHS,CMR,0,A,ON,1*68
```

```
$PASHS,CMR,1,A,ON,13*5A
```

```
$PASHS,CMR,2,A,ON,31*59
```

```
$PASHS,CMR,3,A,ON,1*6B
```

Setting default CMR+ messages at a base:

```
$PASHS,POS,'position'*cc
```

or

```
$PASHS,POS,CUR*51
```

```
$PASHS,CMR,10,A,ON*68
```

```
$PASHS,CMR,3,A,ON,1*6B
```

## CMR,ALL: Disabling All CMR or CMR+ Messages

---

**Function** This command is used to disable all the CMR or CMR+ messages currently enabled on the specified port.

**Command Format** **Syntax**

```
$PASHS,CMR,ALL,c1,OFF[*cc]
```

## Parameters

Parameter	Description	Range
c1	Port ID: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A-D, F, I, P, Q
*cc	Optional checksum	*00-*FF

## Example

Disabling all CMR messages on port A:

```
$PASHS,CMR,ALL,A,OFF*4A
```

**Related Commands**    \$PASHS,CMR

## COO,REF: Entering a Reference Position Manually in a Rover

---

### Function

This command is useful when some providers deliver corrections streams without including the reference position whereas that position is usually known and can be found on provider's websites.

So this command is used at a rover, to enter the reference position of the base the rover is working with. It allows the rover to process reference data without having the reference position included in these data. If the reference data contains the required reference position, then the position you entered through COO,REF may be ignored or used, depending on the preference rule you set with this command.

This command only makes sense to enter the reference position of the base defined as the base data source for the first baseline (i.e. as defined using \$PASHS,1,BLN,...).

### Command Format    Syntax

```
$PASHS,COO,REF,f1,f2,f3[,d4][*cc]
```



## Parameters

Parameter	Description	Range	Default
f1	X cartesian coordinate of base, in meters		
f2	Y cartesian coordinate of base, in meters		
f3	Z cartesian coordinate of base, in meters		
d4	Preference rule: • 0: Incoming position is preferred • 1: Always ignore incoming position	0, 1	0
*cc	Optional checksum	*00-*FF	

## Example

Entering a reference position:

**\$PASHS,COO,REF,2352345.2800,2717465.7080,5251459.2240**

## Comments

- It is the user's responsibility to provide the coordinates in the appropriate datum and for the appropriate point: The entered position is assumed to be that of the PC1 point on the reference antenna and its coordinates are expressed in the datum last selected through \$PASHS,PGS.
- If a new message is received containing the reference position, this position overwrites the point you last entered through \$PASHS,COO,REF.

## CST,ON: Starting the Embedded NTRIP Caster

---

**Function** This command is used to launch the embedded NTRIP caster in the receiver. By default, the embedded NTRIP caster is off and cannot be started until a password is defined (by default the password field is empty).

## Command Format Syntax

**\$PASHS,CST,ON[\*cc]**

## Parameters

None.

## Example

Starting the embedded NTRIP caster:

**\$PASHS,CST,ON\*1C**

**Relevant Query Command**    \$PASHQ,CST

**See Also**    \$PASHS,CST,OFF  
\$PASHS,CST,PAR

## CST,OFF: Stopping the Embedded NTRIP Caster

---

**Function**    This command is used to ask the receiver to stop running the embedded NTRIP caster. By default, the embedded NTRIP caster is off.

**Command Format Syntax**  
\$PASHS,CST,OFF[\*cc]

### Parameters

None.

### Example

Stopping the embedded NTRIP caster:

**\$PASHS,CST,OFF\*52**

**Relevant Query Command**    \$PASHQ,CST

**See Also**    \$PASHS,CST,ON  
\$PASHS,CST,PAR

## CST,MTP,ADD: Adding/Modifying Mount Points

---

**Function**    This command is used to add or modify a mount point in the embedded NTRIP caster. All the information you enter with this command is made available to users through the source table.

**Warning!** Make sure the command does not exceed 349 characters in length before sending it to the receiver.

## Command Format Syntax

`$PASHS,CST,MTP,ADD,s1[,s2,s3,s4,s5,f6,f7,s8][*cc]`

### Parameters

Parameter	Description	Range	Default
s1	Mount point name. An abbreviated name is recommended (no space character allowed). The identifier field (s3) may be used to enter a more detailed definition of the mount point name.	100 characters max.	-
s2	Mount point identifier	100 characters max.	-
s3	Format of the data available through the mount point (ATOM, RTCM, etc.)	100 characters max.	-
s4	Details of the data format (message types, etc.). Comma symbols may be entered as delimiters provided quotation marks are used to encompass the whole string (see example below). The semicolon character is not allowed in the string.	100 characters max	-
s5	Country code.	3 characters	FRA
f6	Latitude, in degrees, with two decimal places.	±90.00	0.00
f7	Longitude, in degrees, with two decimal places.	±180.00	0.00
s8	Fee indicator: <ul style="list-style-type: none"> <li>• Y: Use of the mount point is subject to a fee.</li> <li>• N: Use of the mount point is free.</li> </ul>	Y, N	N
*cc	Optional checksum	*00-*FF	

### Example

Creating the “NAN2” mount point for an NTRIP server delivering RTCM3.0 data, messages 1014 and 1012:

`$PASHS,CST,MTP,ADD,NAN2,Nantes LF2,RTCMV3.0,“1004(1s),1012(1s),1006(13s)”,FRA,47.17,1.00,N*7A`

**Relevant Query Command** `$PASHQ,PAR,CST`

**See Also** `$PASHS,CST,PAR`  
`$PASHS,CST`  
`$PASHS,CST,MTP,DEL`

## CST,MTP,DEL: Deleting a Mount Point

---

**Function** This command is used to delete a mount point from the embedded NTRIP caster source table.

**Command Format Syntax**  
**\$PASHS,CST,MTP,DEL,s1[\*cc]**

### Parameters

Parameter	Description	Range	Default
s1	Name of the mount point you want to delete.	100 characters max.	-
*cc	Optional checksum	*00-*FF	

### Example

Deleting the “NAN2” mount point:

**\$PASHS,CST,MTP,DEL,NAN2\*6A**

**Relevant Query Command** \$PASHQ,PAR,CST

**See Also** \$PASHS,CST,PAR  
 \$PASHS,CST,MTP,ADD

## CST,PAR: Embedded NTRIP Caster Parameters

---

**Function** This command is used to define the parameters of the embedded NTRIP caster. All these parameters will appear in the NTRIP caster source table.

**Warning!** Make sure the command does not exceed 349 characters in length before sending it to the receiver.

**Command Format Syntax**  
**\$PASHS,CST,PAR,d1,s2,s3,d4,s5,s6,s7,f8,f9,s10,d11,s12,s13,c14[,s15,s16,s17][\*cc]**

## Parameters

Parameter	Description	Range	Default
d1	IP port number of the NTRIP caster	100-65535	2101
s2	Host domain name or IP address of the NTRIP caster. By default, the address of the NTRIP caster is the receiver's IP address. In this case, s2 does not need to be specified. If another IP address is used, please mention it as s2.	128 characters max.	x.x.x.x
s3	NTRIP caster password. This password is used by NTRIP servers (data sources) to connect to the NTRIP caster.	32 characters max.	
d4	Number of simultaneous connections per user.	1-100	1
s5	NTRIP caster identifier. Use this field to provide more information describing/identifying the NTRIP caster.	100 characters max.	MB2
s6	NTRIP caster operator: Name of the institution, agency or company running the caster.	100 characters max.	Ashtech
s7	Country code	3 characters	FRA
f8	Latitude, in degrees with two decimal places.	±90.00	0.00
f9	Longitude, in degrees with two decimal places.	0.00 to 359.99	0.00
s10	Fallback caster IP address. (Fallback caster: the caster where to connect to in case this one breaks down).	128 characters max	0.0.0.0
d11	Fallback caster IP port number	0, 100-65535	0
s12	Network identifier, e.g. name of a network of GNSS permanent stations.	100 characters max	-
s13	Network operator: Name of the institution, agency or company running the network.	100 characters max	-
c14	Fee indicator: • Y: Usage is charged • N: No user fee	Y, N	N
s15	Web address where network information can be found.	100 characters max	-
s16	Web address where data stream information can be found.	100 characters max	-
s17	Web or email address where registration information can be found.	100 characters max	-
*cc	Optional checksum	*00-*FF	

**Example**

Entering parameters defining the embedded NTRIP caster:

```
$PASHS,CST,PAR,2102,83.165.25.14,password,10,
NTRIP Caster MB-Two,Trimble,FRA,47.10,-1.00,123.12.12,2101,
My Network,Trimble,Y,www.trimble.com, www.trimble.com,
mb-two@trimble.com
```

**Related Commands**    \$PASHQ,CST (query command)  
                           \$PASHQ,PAR  
                           \$PASHS,CST,ON  
                           \$PASHS,CST,OFF  
                           \$PASHS,CST,USR,ADD  
                           \$PASHS,CST,USR,DEL  
                           \$PASHS,CST,MTP,ADD  
                           \$PASHS,CST,MTP,DEL

## CST,RST: Resetting the Embedded NTRIP Caster

---

**Function**    This command is used to reset the embedded NTRIP caster in the receiver.  
 Resetting the caster means deleting all existing mount points and users and setting the caster definition to its default values.

**Command Format**    **Syntax**  
                           \$PASHS,CST,RST[\*cc]

**Parameters**

None.

**Example**

Resetting the embedded NTRIP caster:

```
$PASHS,CST,RST*48
```

**Related Commands**    \$PASHS,CST,PAR

## CST,USR,ADD: Adding/Modifying NTRIP Caster Users

---

**Function** This command is used to add or modify a user allowed to connect the embedded NTRIP caster. Up to 100 users may be defined.

### Command Format Syntax

```
$PASHS,CST,USR,ADD,s1,s2,s3[,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13][*cc]
```

### Parameters

Parameter	Description	Range	Default
s1	Name of the new user (case sensitive).	32 characters max.	-
s2	User password	32 characters max.	-
s3	Indicator for user-authorized mount points: <ul style="list-style-type: none"> <li>• ALL: all existing mount points can be accessed by the user.</li> <li>• SEL: Only the listed mount points (see s4,...,s13 below) can be accessed by the user.</li> </ul>	ALL, SEL	ALL
s4,...,s13	List of existing mount points the user is allowed to connect to. Mount point name 1, up to mount point name 10	100 characters max. (each)	-
*cc	Optional checksum	*00-*FF	

### Examples

Entering a user named “Ashtech” allowed to connect to all the existing mount points managed by the embedded NTRIP caster:

```
$PASHS,CST,USR,ADD,Ashtech,password,ALL*16
```

Modifying the “Ashtech” user so it is only allowed to use only two of the existing mount points:

```
$PASHS,CST,USR,ADD,Ashtech,password,SEL,NAN1,NAN2*0E
```

### Comments

- If a user is created with no mount point associated to it, then this user is allowed to connect to all existing mount points.
- If a mount point is created with no user associated to it, then the mount point is accessible to all users (not a protected mount point).

**Related Commands**    \$PASHQ,PAR,CST (query command)  
                               \$PASHS,CST,PAR  
                               \$PASHS,CST,USR,DEL

## CST,USR,DEL: Deleting an NTRIP Caster User

---

**Function**            This command is used to delete a user declared as an NTRIP Caster user.

**Command Format**    **Syntax**  
                               \$PASHS,CST,USR,DEL,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Name of the user you want to delete.	32 characters max.	-
*cc	Optional checksum	*00-*FF	

### Example

Deleting the “Ashtech” user:

```
$PASHS,CST,USR,DEL,Ashtech*44
```

**Related Commands**    \$PASHQ,PAR,CST (query command)  
                               \$PASHS,CST,PAR  
                               \$PASHS,CST,USR,ADD

## DFC,DEL: Deleting Default Configuration File

---

**Function**            This command is used to delete the default configuration file (*default\_config.cmd* file).

**Command Format**    **Syntax**  
                               \$PASHS,DFC,DEL[\*cc]



**Parameters**

None.

**See also** \$PASHQ,DFC (relevant query command)  
 \$PASHS,RST  
 \$PASHS,INI  
 \$PASHS,DFC,SEL  
 \$PASHS,DFC,GET  
 \$PASHS,DFC,TST

**DFC,GET: Duplicating the Default Configuration File**

---

**Function** This command is used to duplicate the default configuration file (*default\_config.cmd*) currently used in the board. The resulting file is saved to the specified memory under the specified name (default name: *default\_config.cmd*).

**Command Format Syntax**

```
$PASHS,DFC,GET,d1[,s2][*cc]
```

**Parameters**

Parameter	Description	Range	Default
d1	Storage medium where to save the file: <ul style="list-style-type: none"> <li>• 0: Internal memory.</li> <li>• 2: USB memory</li> </ul>	0, 2	0
s2	File name (*.cmd)		default_config.cmd
*cc	Optional checksum	*00-*FF	

**Example**

Duplicating the default configuration file as file “myconfig.cmd” saved to the USB memory:

```
$PASHS,DFC,GET,2,myconfig.cmd*26
```

**Comments** If the *default\_config.cmd* file does not exist in the board, then the created file is empty.

**See also** \$PASHQ,DFC (relevant query command)  
 \$PASHS,RST  
 \$PASHS,INI

```
$PASHS,DFC,DEL
$PASHS,DFC,SET
$PASHS,DFC,TST
```

## DFC,SET: Setting the Default Configuration File

---

**Function** This command is used so that the specified configuration file found on the specified memory becomes the board's new default configuration file.

### Command Format Syntax

```
$PASHS,DFC,SET,d1,s2[*cc]
```

### Parameters

Parameter	Description	Range	Default
d1	Storage medium where to find the configuration file: • 0: Internal memory. • 2: USB memory	0, 2	0
s2	File name (*.cmd)		default_config.cmd
*cc	Optional checksum	*00-*FF	

### Example

Making the “myconfig.cmd” file found on the USB memory the board's new default configuration file:

```
$PASHS,DFC,SET,2,myconfig.cmd*32
```

### Comments

- The specified file is copied to a partition in the internal memory. The copy is renamed “*default\_config.cmd*”.
- The default configuration file should only contain one comment line (starting with #) and \$PASHS commands. All other lines will be ignored. The first line must be a comment with a title. Below is an example of default configuration file:

```
# Default configuration for my company – Created on June 27, 2016
$PASHS,ELM,10
$PASHS,NME,GGA,A,ON,1
$PASHS,NME,GSV,A,ON,5
```

**See also** \$PASHQ,DFC (relevant query command)  
\$PASHS,RST

\$PASHS,INI  
 \$PASHS,DFC,DEL  
 \$PASHS,DFC,GET  
 \$PASHS,DFC,TST

## DFC,TST: Testing the Default Configuration File

---

**Function** This command is used to test the default configuration file. All commands found in the file are run in succession. The *default\_config.log* file is created on the specified storage medium collecting all the responses to these commands.

### Command Format Syntax

\$PASHS,DFC,TST[d1][\*cc]

### Parameters

Parameter	Description	Range	Default
d1	Storage medium where to save the log file: <ul style="list-style-type: none"> <li>• 0: Internal memory.</li> <li>• 2: USB memory</li> </ul>	0, 2	0
*cc	Optional checksum	*00-*FF	

**See also** \$PASHQ,DFC (relevant query command)  
 \$PASHS,RST  
 \$PASHS,INI  
 \$PASHS,DFC,DEL  
 \$PASHS,DFC,GET  
 \$PASHS,DFC,SET

## DDN,PAR: Setting the DynDNS Service

---

**Function** This command is used to activate or deactivate a connection to a service ensuring that the receiver hostname will always be associated with the dynamic IP address your Internet Service Provider has last assigned to the receiver.

The successful use of the service requires that you first open an account on this service.

## Command Format Syntax

```
$PASHS,DDN,PAR[,DYN,d1][,SYS,s2][,USR,s3][,PWD,s4][,HNM,s5]
[,PER,d6][*cc]
```

## Parameters

Parameter	Description	Range	Default
DYN,d1	Enabling/disabling the use of the service: <ul style="list-style-type: none"> <li>• 0: Enable</li> <li>• 1: Disable</li> </ul>	0, 1	0
SYS,s2	Address of the service used.	100 characters max.	dyndns@dyndns.org
USR,s3	Username you chose when creating your personal account on the DynDNS web site.	32 characters max.	-
PWD,s4	Password you chose when creating your personal account on the DynDNS web site.	32 characters max.	-
HNM,s5	Hostname you declared on the DynDNS web site for the receiver.	100 characters max.	-
PER,d6	Update rate, in seconds	60-3600	600
*cc	Optional checksum	*00-*FF	

## Example

Enabling the use of the DynDNS service, for a receiver accessible through hostname "ashtech1":

```
$PASHS,DDN,PAR,DYN,1,SYS,dyndns@dyndns.org,USR,psmith,PWD,ashtech,HNM,ashtech1.dyndns.org,PER,600*0C
```

## Comment

- After running this command with d1=1 to enable the service, the receiver will try to connect to the service. If the connection is successful, the receiver will return \$PASHR,DDN,OK. If it fails, the receiver will return \$PASHR,DDN,FAIL, causing d1 to be reset to "0".
- Running commands \$PASHS,RST and \$PASHS,INI will reset d1 to 0 but will keep all other parameters unchanged.

## Related Commands

```
$PASHQ,PAR
$PASHQ,DDN
```

\$PASHS,ETH,PAR

\$PASHS,DDN,SET

## DDN,SET: Sending the IP Address Manually to DynDNS

---

**Function** This command is used to force the receiver to send at once its IP address to the DynDNS service. Typically this command may be used when you have noticed that the ISP has just changed the (public) IP address of the receiver. By default, the IP address is sent to the DynDNS server every 10 minutes.

**Command Format** **Syntax**  
`$PASHS,DDN,SET[*cc]`

### Parameters

None.

### Example

Sending immediately the IP address to the DynDNS service:

```
$PASHS,DDN,SET*55
```

**Relevant Query Commands** \$PASHQ,DDN (query command)  
 \$PASHQ,PAR  
 \$PASHS,DDN,PAR  
 \$PASHS,ETH,PAR

## DIF,BDS: Enabling or Disabling BeiDou Correcting Data

---

**Function** This command is used to enable or disable the use of Beidou correcting data in the position computation. It will be rejected (“NAKed”) if the BeiDou tracking firmware option has not been installed.

**Command Format** **Syntax**  
`$PASHS,DIF,BDS,s1[*cc]`

## Parameters

Parameter	Description	Range	Default
s1	BeiDou on/off control: <ul style="list-style-type: none"> <li>• OFF: BeiDou correcting data not used</li> <li>• AUT: BeiDou correcting data used (in China only)</li> </ul>	OFF, AUT	OFF
*cc	Optional checksum	*00-*FF	

**Example** Enabling BeiDou correcting data:  
**\$PASHS,DIF,BDS,AUT**

**Relevant Query Command** \$PASHQ,DIF,BDS

## DIF,NET: Specifying the Processing Mode for a Network Rover

---

**Function** This command is used to choose which incoming correction data to use when several data sources exist. The command will be NAKed if the RTK Rover ([J]) firmware option has not been activated.

**Command Format Syntax**  
**\$PASHS,DIF,NET,s1[\*cc]**

### Parameters

Parameter	Description	Range	Default
s1	Choice of processing mode: <ul style="list-style-type: none"> <li>• IP: IP network. The RTK engine will reset on detecting a change in the ID of the base providing correction data.</li> <li>• UHF: UHF networking. A new RTK engine will be run when a new base ID is detected.</li> </ul>	IP, UHF	IP
*cc	Optional checksum	*00-*FF	

**Example** Allowing UHF networking:  
**\$PASHS,DIF,NET,UHF\*3A**

**See also** \$PASHQ,DIF,NET (relevant query command)  
 \$PASHS,RTK,STI  
 \$PASHS,BLN

## DIF,SBA: Specifying Which SBAS Corrections to Use

---

**Function** This command is used to specify the provider(s) of SBAS corrections (SBAS: Space Based Augmentation System).

### Command Format Syntax

```
$PASHS,DIF,SBA,[n(s1)],s2[*cc]
```

### Parameters

Parameter	Description	Range
n(s1)	List of SBAS providers used (comma used as separator): <ul style="list-style-type: none"> <li>• WAA (WAAS)</li> <li>• EGNOS (EGN)</li> <li>• MSAS (MSA)</li> <li>• GAGAN (GAG)</li> <li>• SDCM (Luch) (SDC)</li> <li>• BEIDOU: BDS</li> <li>• s1 omitted: see s2 below.</li> </ul>	WAA, EGN, MSA, GAG, SDC, QZS, BDS
s2	Enables (ON) or disables (OFF) SBAS. Combined with s1 omitted, enables or disables all SBAS providers.	ON, OFF
*cc	Optional checksum	*00-FF

### Comments

- BeiDou geostationary satellites generate SBAS-like ionosphere grid data and for this reason are regarded as being part of the SBAS providers. If BeiDou tracking is enabled and BeiDou is declared as one of the SBAS providers through this command, then ionosphere grid data will be decoded and applied. The way the receiver selects/combines corrections from the different SBAS providers is an internal process not described here.
- When disabling an SBAS provider, the corresponding corrections stop being used immediately but the satellite can still be tracked.
- When enabling a new SBAS provider, it is immediately added to the current list of providers and its corrections will be used as soon as they are received.
- Disabling a provider for a very short time does not mean its corrections will be available immediately after re-enabling it.

- The settings made with this command are saved to receiver memory. These will be restored after a power cycle.

## Examples

Enabling MSAS:

```
$PASHS,DIF,SBA,MSA,ON
```

Enabling SBAS mode:

```
$PASHS,DIF,SBA,,ON
```

Disabling GAGAN:

```
$PASHS,DIF,SBA,GAG,OFF
```

Disabling SBAS mode:

```
$PASHS,DIF,SBA,,OFF
```

## DIP,OFF: Terminating Direct IP Connection

---

### Function

This command is used to terminate the current IP connection to a server.

### Command Format

#### Syntax

```
$PASHS,DIP,OFF[,c1][*cc]
```

#### Parameters

None.

Parameter	Description	Range
c1	IP port used for the connection to the server: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul> When c1 is omitted, the concerned port is the one specified in the last \$PASHS,DIP, PAR command run.	P, Q
*cc	Optional checksum	*00.*FF

#### Examples

Terminating the current connection:

```
$PASHS,DIP,OFF*4B
```

### Related Commands

\$PASHQ,DIP (query command)

\$PASHS,DIP,PAR

\$PASHS,DIP,ON



## DIP,ON: Establishing the Programmed Direct IP Connection

---

**Function** This command is used to establish the programmed IP connection.

**Command Format** **Syntax**  
`$PASHS,DIP,ON[,c1][*cc]`

### Parameters

None.

Parameter	Description	Range
c1	IP port used for the connection to the server: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul> When c1 is omitted, the concerned port is the one specified in the last \$PASHS,DIP, PAR command run.	P, Q
*cc	Optional checksum	*00-*FF

### Example

Establishing the programmed Direct IP connection:

`$PASHS,DIP,ON*05`

**Related Commands** \$PASHQ,DIP (query command)  
 \$PASHS,DIP,PAR  
 \$PASHS,DIP,OFF

## DIP,PAR: Setting Direct IP Parameters

---

**Function** This command is used to set the different parameters allowing the receiver to perform a Direct IP connection to an external server, typically a base.

## Command Format Syntax

```
$PASHS,DIP,PAR,ADD,s1,PRT,d2[,LGN,s3,PWD,s4][,IPP,c5]
[,ACN,d6]*cc
```

### Parameters

Parameter	Description	Range	Default
ADD,s1	IP address or host name	32 characters max.	
PRT,d2	IP port number	0-65535	
LGN,s3	User name (optional)	32 characters max.	
PWD,s4	Password (optional)	32 characters max.	
IPP,c5	Port used in the receiver to establish the IP connection: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul>	P, Q	P
ACN,d6	Auto-connection: <ul style="list-style-type: none"> <li>• 0: After a power cycle, there won't be any direct IP connection (default)</li> <li>• 1: After a power cycle, the direct IP connection will be restored</li> </ul>	0, 1	0
*cc	Optional checksum	*00-*FF	

### Comments

- When connecting to the specified server requires a user name and password, then the receiver will send the serial command \$GPAID,s3,s4 after the IP connection with the server has been established.
- P and Q are generic TCP/IP ports. By default, the receiver determines which physical port can be used (Modem, Wifi, Ethernet). You can however specify which physical port should be used through command \$PASHS, TCP, RTE.
- The last \$PASHS,DIP,PAR command issued determines which of the two ports (P or Q) are used by default in commands \$PASHS,DIP,ON and \$PASHS,DIF,OFF when these are run without specifying a port (c1 omitted).

### Examples

Entering the parameters of the server the receiver has to connect to (through an IP address):

```
$PASHS,DIP,PAR,ADD,192.65.54.1,PRT,80
```

Entering the parameters of the server the receiver has to connect to (through a host name):

```
$PASHS,DIP,PAR,ADD,www.MyRec.com,PRT,8080
```

**Related Commands**    \$PASHQ,DIP (query command)  
                               \$PASHS,DIP,ON  
                               \$PASHS,DIP,OFF

## DRD: Data Recording Duration

---

**Function**            This command sets a duration for all the G-files that the receiver will log. When a duration is set, the receiver automatically creates a new G-file right after the currently logged G-file has reached the specified duration.

### Command Format    Syntax

`$PASHS,DRD,d[*cc]`

### Parameters

Parameter	Description	Range	Default
d	Data recording duration: <ul style="list-style-type: none"> <li>• 0: Unlimited duration</li> <li>• Other than 0: Duration in minutes</li> </ul>	0, 15, 20, 30, (n x 60). Where n is an integer between 1 and 24	0
*cc	Optional checksum	*00-*FF	

### Comments

- The command will be NAKed if the ring file buffer is currently active (see \$PASHS,RFB).
- The recording of G-files are all started at round hour values of GPS time. This means the first file may be shorter in duration than all those that will follow.

### Example

Setting the duration to 15 minutes:

`$PASHS,DRD,15*0F`

**Relevant Query Command**    \$PASHQ,DRD

**See also**            \$PASHS,REC to start/stop data recording.

## DRI: Raw Data Recording Rate

---

**Function** This command sets the recording rate for all raw data logged in the internal or external memory. It also changes the output rate of messages ATOM RNX and PVT on memory ports.

**Command Format Syntax**

`$PASHS,DRI,f[*cc]`

**Parameters**

Parameter	Description	Range	Default
f	Raw data recording rate, in seconds. f=0 means no data output	0, 0.05, 0.1, 0.2, 0.5, 1-6, 10, 12, 15, 20, 30, 60, 120, etc. integer minute up to 960.	1 s
*cc	Optional checksum	*00-*FF	

**Example**

Setting the recording rate to 5 seconds:

`$PASHS,DRI,5*33`

**Related Commands** \$PASHQ,DRI (query command)  
\$PASHS,ATM  
\$PASHS,REC

## DSY: Daisy Chain

---

**Function** This command is used to redirect all the characters flowing through a given serial port (source port) to another (destination port), without interpreting the flow of redirected data.

Once the daisy chain mode is on, only the command used to discontinue this mode can be interpreted on the source port. Redirection can be in both directions, in which case two DSY commands, instead of one, are required to allow bidirectional data flow.

**Command Format Syntax**

Redirecting data from a source port to a destination port:

`$PASHS,DSY,c1,c2[*cc]`

Discontinuing the daisy chain mode from a specified source port:

```
$PASHS,DSY,c1,OFF[*cc]
```

Discontinuing the daisy chain mode for all source ports:

```
$PASHS,DSY,OFF[*cc]
```

### Parameters

Parameter	Description	Range
c1	Source port ID: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> <li>• V: CAN port</li> </ul>	A-D, F, I, P, Q, V
c2	Destination port ID: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> <li>• V: CAN port</li> </ul>	A-D, F, I, P, Q, V
*cc	Optional checksum	*00-*FF

### Examples

Redirecting port D to port A:

```
$PASHS,DSY,D,A*3E
```

Redirecting port D to port A and port A to port D:

```
$PASHS,DSY,D,A*3E
```

```
$PASHS,DSY,A,D*3E
```

Discontinuing the daisy chain mode from port A:

```
$PASHS,DSY,A,OFF*35
```

Discontinuing the daisy chain mode from all source ports:

```
$PASHS,DSY,OFF*58
```

## DYN: Receiver Dynamics

---

**Function** This command allows you to define the receiver dynamics. The chosen number best represents the receiver motion.

**Command Format** **Syntax**  

```
$PASHS,DYN,d1[*cc]
```

## Parameters

Parameter	Description	Range	Default
d1	Receiver dynamics: <ul style="list-style-type: none"> <li>• 1: Static</li> <li>• 2: Quasi-static</li> <li>• 3: Walking</li> <li>• 4: Ship</li> <li>• 5: Automobile</li> <li>• 6: Aircraft</li> <li>• 7: Unlimited</li> <li>• 8: Adaptive</li> <li>• 9: User-defined (see also \$PASHS,UDP)</li> </ul>	1-9	8
*cc	Optional checksum	*00-*FF	

### Example

Setting rover dynamics to “Walking”:

```
$PASHS,DYN,3*39
```

### Comments

- If the receiver is set as an RTK rover using a moving base mode, it is recommended to use d1=8 (Adaptive).
- In the adaptive mode (8), the receiver analyzes its own motion and automatically chooses one of the dynamic models that is the most suitable. The possible dynamic models are those corresponding to the other choices in the command (i.e. 2 to 7, but not 1 or 9). Using the adaptive mode rejects the possible use of the user-defined dynamic model.

**Related Commands**    \$PASHQ,DYN (query command)  
                               \$PASHQ,PAR  
                               \$PASHS,UDP

## ELM: Setting the Elevation Mask for Raw Data Output

---

**Function**    This command is used to set the minimum satellite elevation for the output of raw and differential measurement data. The following data are impacted: RT2, RT3, CMR, CMR+, ATM regardless of their use (i.e. as input protocol for differential

data or in raw data recording). The value entered through this command applies to all sensors on the board.

NOTE: If the mask value set with command PEM exceeds the one set with ELM, then it is the mask set with PEM that is used to output raw /differential data.

## Command Format Syntax

```
$PASHS,ELM,d1[*cc]
```

### Parameters

Parameter	Description	Range	Default
d1	Elevation mask, in degrees.	0-90°	5
*cc	Optional checksum	*00-*FF	

### Example

Setting the elevation mask to 10 degrees:

```
$PASHS,ELM,10*1C
```

## ENC: Setting Transport Mode for Differential Data

---

**Function** This command defines the transport mode used for every differential data message flowing through the specified physical port.

## Command Format Syntax

```
$PASHS,ENC,c1,s2[*cc]
```

### Parameters

Parameter	Description	Range	Default
c1	Port ID.	A, B, C, D, F, I, P, Q, M, U	-
s2	Transport identifier.	See table below	NTV
*cc	Optional checksum	*00-*FF	

The list of transport identifiers is the following:

Data	Description
ASH	\$PASHR transport used for encapsulation of all non-Ashtech messages
RT3	Standardized RTCM-3 transport for encapsulation of all non-RT3 messages.
NTV	NaTiVe transport, i.e. no encapsulation for all output messages.

**Example**

Setting the data encapsulation mode to “NTV” for port A:

```
$PASHS,ENC,A,NTV*30
```

**Comments**

This command is only applicable to physical ports.  
The RT3 encapsulation mode is no longer supported.

**ETD,PAR: Setting Ethernet Driver Parameters**

---

**Function**

This command is used to change the parameters of the Ethernet drivers. These parameters are stored in permanent memory. These settings will not be affected if you further run commands \$PASHS,RST or \$PASHS,INI.

**Command Format****Syntax**

```
$PASHS,ETD,PAR[,AUT,s1][[,DPL,s2][[,SPD,d3] ]]*cc]
```

**Parameters**

Parameter	Description	Range	Default
AUT,s1	Auto-negotiation: • ON: Enable • OFF: Disable	ON, OFF	ON
DPL,s2	Duplex mode: • : Set full-duplex mode • : Set half-duplex mode  (s2 required only if s1= OFF)	FULL, HALF	FULL
SPD,d3	Speed, in Mbits/second. (d3 required only if s1= OFF)	10, 100	100
*cc	Optional checksum	*00-*FF	

**Example**

Disabling auto-negotiation:

```
$PASHS,ETD,PAR,AUT,OFF,DPL,FULL,SPD,100
```

Enabling auto-negotiation:

```
$PASHS,ETD,PAR,AUT,ON
```

**Related Commands**

```
$PASHQ,ETD  
$PASHS,ETH,ON  
$PASHS,ETH,OFF
```



## ETH,OFF: Powering Off the Ethernet Port

---

**Function** This command is used to power off the Ethernet port. By default, the Ethernet port is on.

**Command Format** **Syntax**  
`$PASHS,ETH,OFF[*cc]`

**Parameters**

None.

**Example**

Turning off the Ethernet port:

`$PASHS,ETH,OFF*4F`

**Related Commands** \$PASHQ,ETH (query command)  
\$PASHS,ETH,ON  
\$PASHS,ETH,PAR

## ETH,ON: Powering On the Ethernet Port

---

**Function** This command is used to power on the Ethernet port. By default, the Ethernet port is on.

**Command Format** **Syntax**  
`$PASHS,ETH,ON[*cc]`

**Parameters**

None.

**Example**

Turning on the Ethernet port:

`$PASHS,ETH,ON*01`

**Related Commands** \$PASHQ,ETH (query command)  
\$PASHS,ETH,OFF  
\$PASHS,ETH,PAR

## ETH,PAR: Ethernet Parameters

**Function** This command is used to set the Ethernet parameters.

### Command Format Syntax

```
$PASHS,ETH,PAR[,DHP,s1][,ADD,s2][,MSK,s3][,GTW,s4][,DN1,s5]
[,DN2,s6][,MAC,s7][,ACN,d8]*cc
```

### Parameters

Parameter	Description	Range	Default
DHP,s1	DHCP mode: 0: Disabled (static IP address) 1: Enabled (dynamic IP address)	0, 1	1
ADD,s2	IP address when s1=0	0.0.0.0-255.255.255.255	192.168.0.1
MSK,s3	Sub-network mask when s1=0	0.0.0.0-255.255.255.255	255.255.255.0
GTW,s4	Gateway IP address when s1=0	0.0.0.0-255.255.255.255	255.255.255.255
DN1,s5	DNS 1 IP address when s1=0	0.0.0.0-255.255.255.255	255.255.255.255
DN2,s6	DNS 2 IP address when s1=0	0.0.0.0-255.255.255.255	255.255.255.255
MAC,s7	MAC address	00:00:00:00:00:00-FF/FF/FF/FF/FF/FF	
ACN,d8	(Obsolete)		
*cc	Optional checksum	*00-*FF	

### Example

Ethernet configuration with DHCP:

```
$PASHS,ETH,PAR,DHP,1*2E
```

Ethernet configuration without DHCP (static IP address):

```
$PASHS,ETH,PAR,DHP,0,ADD,10.20.2.28,MSK,255.255.255.0,GTW,
10.20.2.1,DN1,134.20.2.16,DN2,134.20.2.3,MAC,01:02:03:04:05:06,ACN,1*
70
```

### Relevant Query Command

```
$PASHQ,ETH (query command)
```

```
$PASHS,ETH,OFF
```

```
$PASHS,ETH,ON
```

## FIL,CPY: Copying Files from Internal to External Memory

---

**Function** Typically, this command is used to copy the specified file(s), or all the files, from the internal to external memory (USB).

### Command Format Syntax

`$PASHS,FIL,CPY,s1[*cc]`

### Parameters

Parameter	Description	Range
s1	Specifies file(s) to be copied: <ul style="list-style-type: none"> <li>• Complete file name</li> <li>• Or range of files using the "*" character combined with quotation marks. Examples:               <ul style="list-style-type: none"> <li>*.*</li> <li>"&lt;filename&gt;.*"</li> <li>*.&lt;file_extension&gt;'</li> </ul> </li> <li>• Or s1=ALL: All files present in internal memory are copied to external memory.</li> <li>• or s1=STP: Instantly stops copying files.</li> </ul>	<filename(s)>, ALL, STP
*cc	Optional checksum	*00-*FF

### Examples

Copying one G file:

`$PASHS,FIL,CPY,GabcdA09.241`

Copying all files:

`$PASHS,FIL,CPY,ALL`

Copying all files recorded on day 241:

`$PASHS,FIL,CPY,"*.241"`

### Comments

- After the file or files have been copied, the receiver returns:

`$PASHR,FIL,OK*1F`

- If an error occurs during the copy operation, the receiver returns:

`$PASHR,FIL,FAIL*19`

- If copying files is stopped using the command, the receiver returns:  
**\$PASHR,FIL,STP\*4C**

**Related Commands** \$PASHQ,FIL,LST  
No query command

## FIL,DEL: Deleting Files and Directories

---

**Function** This command allows you to delete files and directories from the selected internal or external memory.

**Command Format** **Syntax**  
**\$PASHS,FIL,DEL,[d1],[s2],s3[s4[...sn]][\*cc]**

### Parameters

Parameter	Description	Range
d1	Memory from which to delete files or directories: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 2: USB memory</li> <li>• 3: Internal memory (log file partition)</li> </ul> If d1 is omitted, files or directories are deleted from the memory specified by the last run \$PASHS,MEM command.	0, 2, 3
s2	Path	255 characters max.
s3	Name of the file or directory you want to delete.	255 characters max.
	...	
sn	Name of the file or directory you want to delete.	255 characters max.
*cc	Optional checksum	*00-*FF

### Comments

- To delete a file or directory located in a subdirectory, the full path to this file or directory should be specified in the s2 field. You cannot enter a path in the s3 field.
- The "\*" character can be used as a wild card to delete several files at the same time. In this case, the complete string should be placed between single or double quotation marks.

**Examples**

Deleting a G file:

```
$PASHS,FIL,DEL,,,GabcdA09.241*69
```

Deleting three G files:

```
$PASHS,FIL,DEL,,,GabcdA09.241,GabcdB09.242,GabcdC09.242*68
```

**Related Commands**    \$PASHQ,FIL,LST  
                           \$PASHS,MEM

## **FIL,GET: Downloading a File Thru the Current Port**

---

**Function**    This command is used to download a file from the board through the currently used port. There's a size limit for the file which is 2 Gbytes maximum.

**Command Format**    **Syntax**  
                           \$PASHS,FIL,GET,s1,[d2],[d3],[d4],[d5][\*cc]

Suspending current download:  
 \$PASHS,FIL,GET,OFF[\*cc]

## Parameters

Parameter	Description	Range	Default
s1	File name (including path if the file is stored in a sub-folder)	255 characters max.	
d2	Memory where the file is stored: • 0: Internal memory (user data partition) • 2: USB memory • 3: Internal memory (log file partition)	0, 2, 3	0
d3	Position in the file where to start downloading from (in bytes)	0-2,048,000,000	0
d4	Permitted size of downloaded file, in bytes d4=0: The entire file will be transferred	0-2,048,000,000	
d5	Packet size, in bytes. If omitted or "0", the packet size is "2048".	0-64,000	2048
*cc	Optional checksum	*00-*FF	

## Response Format Syntax

When issuing this command, first \$PASHR,ACK\*3D is returned, followed by a number of messages, formatted according to the syntax below.

```
$PASHR,FIL,GET,<length><flags><file position><data><checksum> <cr><lf>
```

## Parameters

Parameter	Size	Description
length	2 bytes	Length of data packet
flags	1 byte	Packet identification: • 0: This packet is the last one (end of file). • 1: This is a data packet, not the last one in the file.
file position	4 bytes	Indicates the position within the file of the first data byte in this packet measured from the beginning of the file.
data	Length	Data
checksum	1 byte	Checksum: <length>+<flags>+<file position>+<data> modulo 256

## Comments

If the piece of software in charge of collecting the \$PASHR,FILE,GET messages detects missing data, it must be capable of suspending the download and restart it from the right index in the file, i.e. from where missing data were detected.

## Examples

```
$PASHS,FIL,GET,GazerA09.123,0,0,0,2048
$PASHR,FIL,GET,<2048><1><0><data><checksum>
$PASHR,FIL,GET,<2048><1><4096><data><checksum>
$PASHR,FIL,GET,<2048><1><6144><data><checksum>
```

```
$PASHR,FIL,GET,<1502><0><8192><data><checksum>
```

### \$PASHS,FIL,GET,2013/123/GazerA09.123

```
$PASHR,FIL,GET,<2048><1><0><data><checksum>
```

```
$PASHR,FIL,GET,<2048><1><4096><data><checksum>
```

```
$PASHR,FIL,GET,<2048><1><6144><data><checksum>
```

```
$PASHR,FIL,GET,<1502><0><8192><data><checksum>
```

**Related Commands**    \$PASHQ,FIL,LST

## FIL,WRT: Writing a File

---

**Function**    This command is used to create a file in the board. If the file already exists, it is overwritten.

### Command Format    Syntax

Specifying where to create the file and naming it

```
$PASHS,FIL,WRT,OPN,d1,s2,d3*cc
```

Writing data to the file:

```
$PASHS,FIL,WRT,DAT,<length><flags><file  
position><data><checksum><cr><lf>
```

### Parameters

Parameter	Description	Range	Default
d1	Memory where the file is created: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 2: USB memory</li> </ul>	0, 2	0
s2	File name (including path if the file is created in a sub-folder)	255 char. max.	
d3	Open file mode: <ul style="list-style-type: none"> <li>• 0: Creates a new file (or overwrites the existing one)</li> <li>• 1: Opens an existing file</li> </ul>	0,1	0
*cc	Optional checksum	*00-*FF	

Parameter	Description	Size	Range
length	Length of data packet	2 bytes	0-2048

Parameter	Description	Size	Range
flags	Packet identification: <ul style="list-style-type: none"> <li>• 0: This packet is the last one (end of file).</li> <li>• 1: This is a data packet, not the last one in the file.</li> </ul>	1 byte	
file position	Indicates the position within the file of the first data byte in this packet measured from the beginning of the file.	4 bytes	0-4,096,000,000
data	Data	Length	
checksum	Checksum: <length>+<flags>+<file position>+<data> modulo 256	1 byte	0-255

### Comments

After receiving \$PASHS,FIL,WRT,OPN..., the board will expect \$PASHS,FIL,WRT command lines exclusively (all other commands will be ignored).

The board will return to normal operation after it has received the "end of file" flag, or no data was received for the last 30 seconds, or an error was detected.

## Response Format Syntax

In response to each \$PASHS,FIL,WRT,DAT/OPN,... command, the board returns a status report, using the syntax below.

```
$PASHR,FIL,WRT,d3*cc
```

### Parameters

Parameter	Description	Range
d3	Result code: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Not enough memory</li> <li>• ...</li> </ul>	255 char. max.
*cc	Checksum	1 byte

### Example

```
$PASHS,FIL,WTR,OPN,0,myfile
$PASHS,FIL,WRT,DAT,<2048><1><0><data><checksum>
$PASHR,FIL,WRT,0
$PASHS,FIL,WRT,DAT,<2048><1><2048><data><checksum>
$PASHR,FIL,WRT,0
$PASHS,FIL,WRT,DAT,<2048><1><4096><data><checksum>
$PASHR,FIL,WRT,0
$PASHS,FIL,WRT,DAT,<1502><0><6144><data><checksum>
$PASHR,FIL,WRT,0
etc.
```



## FMT: Formatting Internal Memory or SD Card

---

**Function** This command is used to format the board's internal memory or the SD card inserted in the board.

**Command Format** **Syntax**  
`$PASHS,FMT,d[*cc]`

### Parameters

Parameter	Description	Range
d	Formatting option: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 1: Internal memory (all partitions)</li> <li>• 3: USB memory</li> </ul>	0, 1, 3
*cc	Optional checksum	*00-*FF

### Example

Formatting the internal memory:

```
$PASHS,FMT,1
```

- Comments**
- The board will then be re-started if you use formatting option "0" or "1".
  - Formatting option "0" formats the internal memory where G-files are stored. After formatting, all G-files, if any left there, are lost.
  - Formatting option "1" formats the internal memory entirely. After formatting, G-files and log files, if any left there, and receiver settings are all lost.
  - Formatting option "3" formats the USB memory.

**Related Commands** None.

## FTP,OFF: Ending Data Transfer with FTP

---

**Function** This command is used to stop the data transfer currently in progress with an FTP server.

**Command Format Syntax**`$PASHS,FTP,OFF[*cc]`**Parameters**

None.

**Example**

Stop data transfer:

`$PASHS,FTP,OFF*54`

**Related Commands** `$PASHQ,FTP` (query command)  
`$PASHS,FTP,PAR`  
`$PASHS,FTP,PUT`

**FTP,PAR: FTP Settings**

---

**Function** This command is used to enter the settings of an external FTP server.

**Command Format Syntax**`$PASHS,FTP,PAR[ADD,s1][PRT,d2][LGN,s3][PWD,s4][PTH,s5][*cc]`**Parameters**

Parameter	Description	Range	Default
ADD,s1	IP address or host name of the FTP server	32 characters max.	-
PRT,d2	FTP server port number	0-65535	21
LGN,s3	FTP server login	32 characters max.	
PWD,s4	FTP server password	32 characters max.	
PTH,s5	Path used on the FTP server	255 characters max.	
*cc	Optional checksum	*00-*FF	

**Example**

```
$PASHS,FTP,PAR,ADD,ftp.ashtech.com,PRT,21,LGN,Ashtech,PWD,
u6huz8,PTH,/my folder*14
```

**Relevant Query Command**    \$PASHQ,FTP

**See Also**    \$PASHS,FTP,PUT

## FTP,PUT: Uploading Files to FTP

---

**Function**    This command is used to send files from the receiver's internal memory or USB key to the FTP server, as defined through the \$PASHS,FTP,PAR command.  
Up to 10 files may be transferred through a single command line.

**Command Format**    **Syntax**  
\$PASHS,FTP,PUT,[d1],[s2],[s3],s4,[s5,...[s13]][\*cc]

## Parameters

Parameter	Description	Range	Default
d1	Memory where the files to be transferred can be found: <ul style="list-style-type: none"> <li>• 0: Receiver's internal memory</li> <li>• 2: USB key</li> </ul> If d1 is missing, the memory selected through \$PASHS, MEM is the one where the files should be found.	0,2	-
s2	Subfolder created on the FTP server, in the folder specified in the Path parameter (PTH,s5) of the \$PASHS,FTP,PAR command. If s2 is not specified, files are saved directly in the <Path> folder.	255 characters max.	Empty
s3	Remote path on FTP server	255 characters max.	Empty
s4-s13	Names of the files to be uploaded to the FTP server. The "*" character can be used to select several files. In this case, the filename string should be placed between quotation marks (" or ')	255 characters max.	
*cc	Optional checksum	*00-*FF	

## Examples

Transferring a single file (G1234A09.134) to the FTP server:  
**\$PASHS,FTP,PUT,,,G1234A09.134\*59**

Transferring two files (GabcdA09.134 and GabcB09.134) to the FTP server:

**\$PASHS,FTP,PUT,0,,GabcdA09.134,GabcB09.134\*11**

Transferring all the files from the internal memory to the FTP server:

**\$PASHS,FTP,PUT,0,,,\*.\*\*\*54**

Transferring all the files from the USB key collected on day 65 to the FTP server:

**\$PASHS,FTP,PUT,2,,,\*.65\*ED**

## Comments

- Right after submitting a command line, the following response line will be returned if the command syntax is correct:

**\$PASHR,ACK\*3D**

- After a successful file transfer, the following response line is returned:  
\$PASHR,FTP,OK\*1E
- If the file transfer fails, the following response line is returned:  
\$PASHR,FTP,FAIL\*18
- If you submit a new command while a file transfer sequence is still in progress, your new command is rejected and the following response line is returned:  
\$PASHR,FTP,BUSY\*07

**Related Commands** \$PASHQ,FTP (query command)  
\$PASHS,FTP,PAR  
\$PASHS,MEM

## GAL: Galileo Tracking

---

**Function** This command is used to enable or disable Galileo tracking.

**Command Format** **Syntax**  
\$PASHS,GAL,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Enabling/disabling Galileo tracking: • On: Track and use Galileo satellites • Off: Do not track Galileo satellites	ON, OFF	
*cc	Optional checksum	*00-*FF	-

### Comments

The command is NAKed if the [O] option is not installed.

### Example

Enabling Galileo:  
\$PASHS,GAL,ON\*12

**Related Commands** \$PASHS,QZS  
\$PASHS,BDS

## GEM: Choosing a Geoid Model

---

**Function** This command is used to tell the receiver which of the two available geoid models it should be using. The geoid model primarily affects the height (above the selected geoid) reported in every NMEA GGA message.

### Command Format Syntax

\$PASHS,GEM,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Geoid model to be used: • STANAG: STANAG 4294 geoid model • EGM96: EGM 96 geoid model	STANAG, EGM96	STANAG
*cc	Optional checksum	*00-*FF	

### Example

Choosing EGM96:

\$PASHS,EGM,EGM96

**Related Commands** \$PASHQ,GEM (query)

## GLB: Choice of Antenna Input for L-Band Reception

---

**Function** This command is used to define the antenna input used for receiving the L-band signal. It does not cause the receiver to reset.

### Command Format Syntax

\$PASHS,GLB,c1[\*cc]

## Parameters

Parameter	Description	Range	Default
c1	Antenna used for L-band reception: <ul style="list-style-type: none"> <li>ON (or A1): Antenna input #1 is used for L-band signal reception.</li> <li>OFF: No L-band signal reception.</li> </ul>	ON, OFF, A1	ON
*cc	Optional checksum	*00-*FF	-

## Example

Receiving L-band signal from antenna input #1:

**\$PASHS,GLB,A1**

## Comments

This command is similar to \$PASHS,GPS or \$PASHS,GLO as it does not require any additional setting to start or stop tracking the L-band signal. It is applicable only for specific DUO combinations.

## See also

\$PASHQ,GLB (relevant query command)  
 \$PASHS,SNS  
 \$PASHS,LBN,USE

## GLO: GLONASS Tracking

---

## Function

This command is used to enable or disable GLONASS tracking. The command is valid only if the GLONASS option has been activated.

## Command Format

### Syntax

**\$PASHS,GLO,s1[\*cc]**

## Parameters

Parameter	Description	Range	Default
s1	Enables (ON) or disables (OFF) GLONASS tracking.	ON, OFF	ON
*cc	Optional checksum	*00-*FF	

## Example

Enabling GLONASS:

**\$PASHS,GLO,ON\*1C**

**Related Commands**    \$PASHQ,GLO (query command)  
                               \$PASHS,GPS  
                               \$PASHS,GAL  
                               \$PASHS,BDS  
                               \$PASHS,QZS  
                               \$PASHS,SBA

## GLx: Tracking Signals in Different Bands

---

**Function**            This command is used to enable or disable the tracking of each of the possible signals in the corresponding GNSS RF band (impacts receiver operation in the same way as the corresponding firmware options do).

### Command Format    Syntax

```
$PASHS,GL1,s1[*cc]
$PASHS,GL2,s2[*cc]
$PASHS,GL6,s6[*cc]
```

### Parameters

Parameter	Description	Range	Default
s1	Enabling/disabling L1 tracking: <ul style="list-style-type: none"> <li>• ON: Track and use L1 GNSS signals</li> <li>• OFF: Do not track L1 GNSS signals</li> </ul>	Hard coded to ON	ON
s2	Enabling/disabling L2 tracking: <ul style="list-style-type: none"> <li>• ON: Track and use L2 GNSS signals</li> <li>• OFF: Do not track L2 GNSS signals</li> </ul>	ON, OFF	ON
s6	Enabling/disabling L6 tracking: <ul style="list-style-type: none"> <li>• ON: Track and use L6 GNSS signals</li> <li>• OFF: Do not track L6 GNSS signals</li> </ul>	ON, OFF	
*cc	Optional checksum	*00-*FF	-

### Comments

The table below indicates which RF bands the “Lx” labels in the above table refer to.

“Lx” Label	RF ID
L1	GPS L1 GLONASS G1 BEIDOU B1 GALILEO E1 QZSS L1 SBAS L1



"Lx" Label	RF ID
L2	GPS L2 GLONASS G2 BEIDOU B2 GALILEO E5b
L6	GALILEO E6 BEIDOU B3

**Example**

Enabling L2 tracking:

`$PASHS,GL2,ON`

**Related Commands**    `$PASHQ,GLx` (query commands)

## <GNS>,USE: Enabling or Disabling the Tracking of a GNSS Satellite

---

**Function:** This function is used to enable or disable the tracking of one or more particular GNSS satellites.

**Command Format**    **Syntax**  
`$PASHS,s1,USE,[n(d2)],s3[*cc]`

## Parameters

Parameter	Description	Range	Default
s1	GNSS type: <ul style="list-style-type: none"> <li>• GPS: GPS</li> <li>• GLO: GLONASS</li> <li>• GAL: GALILEO</li> <li>• SBA: SBAS</li> <li>• QZS: QZSS</li> <li>• BDS: BeiDou</li> </ul>	GPS, GLO, GAL, SBA, QZS, BDS	-
d2	Satellite PRN: <ul style="list-style-type: none"> <li>• For GPS: 1-32</li> <li>• For GLONASS: 1-24</li> <li>• For GALILEO: 1-36</li> <li>• For SBAS: 1-39</li> <li>• For QZSS: 1-10</li> <li>• For BDS: 1-30</li> </ul> <p>d2 omitted in the command line combined with s3=ON: Re-enables all the satellites you previously disabled.</p>	1-39	-
s3	Enable (ON) or Disable (OFF) tracking	ON,OFF	ON
*cc	Optional checksum	*00-*FF	-

## Comments

- Use the command as many times as the number of satellites you want to disable from tracking. If a specified satellite number is out of range, the command is NAKed.
- The tracking of a given satellite is suspended immediately after disabling it. The satellite is also excluded from the list of searched/tracked satellites.
- Conversely, re-enabling a previously disabled satellite consists of re-inserting it into the list of searched/tracked satellites.
- Be aware that re-enabling the tracking of a satellite shortly after having disabled it does not mean that the receiver will be able to track back this satellite quickly.

## Examples

Disabling GLONASS satellite PRN 5:

```
$PASHS,GLO,USE,5,OFF
```

Disabling all GLONASS satellites:

```
$PASHS,GLO,USE,,OFF
```

Re-enabling all GPS satellites previously disabled:

```
$PASHS,GPS,USE,,ON
```

## GPS: GPS Tracking

---

**Function** This command is used to enable or disable GPS tracking. The command is valid only if the GPS option ([N]) has been activated.

**Command Format** **Syntax**  
**\$PASHS,GPS,s1[\*cc]**

### Parameters

Parameter	Description	Range	Default
s1	Enables (ON) or disables (OFF) GPS tracking.	ON, OFF	ON
*cc	Optional checksum	*00-*FF	

### Example

Enabling GPS:

**\$PASHS,GPS,ON**

**Related Commands** \$PASHQ,GPS (query command)  
 \$PASHS,SBA  
 \$PASHS,GLO  
 \$PASHS,GAL  
 \$PASHS,QZS  
 \$PASHS,BDS

## GSF: Enabling/Disabling GSOF Messages

---

**Function** This command is used to enable or disable GSOF messages.

**Command Format** **Syntax**  
**\$PASHS,GSF,s1,c2,s3[,f4][\*cc]**

## Parameters

Parameter	Description	Range	Default
s1	GSOFF message type (see table below)	1-3, 8, 9, 11, 12, 27, 33, 38	-
c2	Output port: • Serial port: A, B or D • USB serial port: C • TCP/IP client stream: P, Q • TCP/IP server stream: I, F • To G-file: M	A, B, C, D, F, I, P, Q, M	-
s3	Enabling/disabling command	ON, OFF	-
f4	Output rate, in seconds	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600	1 s
*cc	Optional checksum	*00-*FF	

Supported GSOFF messages:

Message Type	Description
1	POSITION TIME
2	LAT, LONG, HEIGHT
3	ECEF POSITION
8	VELOCITY DATA
9	PDOP INFO
11	POSITION VCV INFO
12	POSITION VCV INFO
16	CURRENT TIME
27	ATTITUDE INFO
33	ALL SV BRIEF INFO
38	POSITION TYPE INFORMATION

### Example

Enabling GSOFF message type "1" at 1 second:

```
$PASHS,GSF,1,C,ON,1
```

**Related Commands** \$PASHQ,OUT,GSF (query command)  
\$PASHS,GSF,ALL

## GSF,ALL: Disabling all GSOFF Messages on a Port

---

**Function** This command is used to disable all the GSOFF messages currently delivered on the specified output port.

**Command Format Syntax**

\$PASHS,GSF,ALL,c1,OFF[\*cc]

**Parameters**

Parameter	Description	Range	Default
c1	Output port delivering GSOF messages: <ul style="list-style-type: none"> <li>Serial port: A, B or D</li> <li>USB serial port: C</li> <li>TCP/IP client stream: P, Q</li> <li>TCP/IP server stream: I, F</li> <li>To G-file: M</li> </ul>	A, B, C, D, F, I, P, Q, M	-
*cc	Optional checksum	*00-*FF	

**Example**

Disabling all GSOF messages on port A:

\$PASHS,GSF,ALL,A,OFF

**Related Commands** \$PASHS,GSF**INI: Resetting Receiver According to Your Preference**

---

**Function** This command resets the receiver processor and memory according to the option you choose.**Command Format Syntax**

\$PASHS,INI,d1[\*cc]

**Parameters**

Parameter	Description	Range
d1	Memory reset code: <ul style="list-style-type: none"> <li>0: No memory reset</li> <li>1: Cold start</li> <li>2: Cold start + memory formatting</li> <li>3: Warm start + memory formatting</li> <li>4: No memory reset+ RTC time cleared</li> <li>5: Warm start</li> <li>9: Cold start (ignore default_config.cmd)</li> </ul>	0, 1, 2, 3, 4, 5, 9 See Memory Reset Codes table below
*cc	Optional checksum	*00-*FF

## Memory Reset Codes:

Reset Code:	0	1	2	3	4	5	9
Clear receiver parameters in backup memory		.	.				.
Clear ephemeris data		.	.	.		.	.
Clear almanac data		.	.				.
Clear latest position		.	.				.
Clear RTC time		.			.		
Reset channels	.	.	.	.	.	.	.
Restart processor	.	.	.	.	.	.	.
Clear receiver parameters in RAM	.	.	.	.	.	.	.
Format internal memory			.	.			
Run default_config.cmd		.	.				

**Comment**

If the board contains a *default\_config.cmd* file, all commands read from this file will be run after resetting the board through \$PASHS,INI,1 or \$PASHS,INI,2.

**Example**

Asking for a cold start followed by a reconfiguration of the board using *default\_config.cmd*:

```
$PASHS,INI,1
```

**Related Commands**

```
$PASHS,RST
$PASHS,DFC,SET
```

## LBN,BEM: Additional Settings for User-Added L-Band Satellite

---

**Function**

This command is used to enter additional parameters for a manually specified L-band satellite (and only for this type of satellite). These parameters come as a supplement to those defined with \$PASHS,LBN,SAT.

Applying this command makes sense when a new L-band satellite is introduced and there has been no firmware update yet that would have taken into account the introduction of this new satellite.

**Command Format****Syntax**

```
$PASHS,LBN,BEM,s1,n(B,f2,f3,f4,f5,f6)[*cc]
```

## Parameters

Parameter	Description	Range
s1	Name of L-band satellite	31 characters max.
n	Indicates that the expression between brackets that follows will be repeated n times. There may be up to 5 different ellipses possible for a given L-band satellite, hence "n" ranging from 1 to 5.	1-5
B	Separator between foot print beam ellipses	-
f2	Beam foot note orientation, in degrees	-89.9 to 90.0
f3	Latitude axis radius, in degrees	0.0 to 90.0
f4	Longitude axis radius, in degrees	0.0 to 180.0
f5	Latitude center	-90.0 to 90.0
f6	Longitude center	-179.9 to 180.0
*cc	Optional checksum	*00.*FF

## Example

Setting parameters for satellite RTXDD:

```
$PASHS,LBN,BEM,RTXDD,B,70.1,40,30,0,37,B,0.1,40,60,35,0*11
```

## Comments

- All the parameters of an ellipse must be specified explicitly, otherwise none of the submitted parameters for this ellipse will be accepted.
- If not a single foot print beam ellipse is specified for a given L-band satellite, then this satellite will have the lowest probability of being used for tracking, compared to other L-band satellites also in view.  
In other words, without one or more ellipses specified for a given L-band satellite, this satellite is likely not to be found as the best one by the automatic best L-band search procedure.  
On the other hand, if this L-band satellite is expected to be assigned for tracking manually, then no ellipse parameters are needed.

**See also** \$PASHQ,LBN,BEM (relevant query command)  
\$PASHS,LBN,USE  
\$PASHS,LBN,SAT

## LBN,SAT: Adding a New L-Band Satellite

---

**Function** This command is used to add a new L-band satellite manually, by entering its name and a few other primary parameters.

Applying this command makes sense when a new L-band satellite is introduced and there has been no firmware update yet that would have taken into account the introduction of this new satellite.

### Command Format Syntax

```
$PASHS,LBN,SAT,s1,d2,d3[,f4][,s5][*cc]
```

To delete a satellite:

```
$PASHS,LBN,SAT,s1,DELETE[*cc]
```

### Parameters

Parameter	Description	Range
s1	Name of L-band satellite	31 characters max.
d2	Carrier frequency, in Hertz	
f3	Baud rate, in bits/second	600, 1200, 2400, 4800
f4	Orbital satellite longitude, in degrees	-179.9 to 180.0
s5	Provider name: • RTX: Trimble RTX (default) • TRS: TERIASat	RTX, TRS
*cc	Optional checksum	*00-*FF

### Example

Entering manually newly introduced satellite RTXDD:

```
$PASHS,LBN,SAT,RTXDD,1525000000,600,22*34
```

### Comments

- Up to four new L-band satellites may be added using this command. To add more satellites, use \$PASHS,INI,1 to delete the already four added ones. Then you can add another new L-band satellite.
- The command will be NAKed if you try to add an L-band satellite that is originally present in the default L-band almanac.
- To modify the parameters of an L-band satellite you have added manually, resume the command in which you will specify the same satellite name, but this time with the correct accompanying parameters.



- Any L-band satellite added manually is stored in the default L-band almanac. That means it is an equal partner to any other hard coded L-band satellite provided its orbital satellite longitude (see f4 above) and its foot print ellipse(s) (see command \$PASHS,LBN,BEM) have been defined correctly. It will have the lowest probability of being used for tracking among all L-band satellites in view if its orbital satellite longitude and/or foot print ellipses have not been defined.
- The tracking of manually added L-band satellites relies on the same command as the other L-band satellites (that is command \$PASHS,LBN,USE). Through this command, you decide whether you want automatic or manual selection of the L-band satellite used.
- In NMEA SLB and ATOM OSS messages, the numbers reported for manually added L-band satellites are in the range (N+1) to (N+4) where N is the size of the hard-coded L-band satellite almanac (all providers included). In GGA, the reported base ID is correspondingly in the range 99 to 96.
- You can delete an L-band satellite from the L-band almanac using command \$PASHS,LBN,SAT,s1,DELETE where s1 is the name of the L-band satellite you want to delete.

**See also** \$PASHQ,LBN (relevant query command)  
 \$PASHS,LBN,USE  
 \$PASHS,LBN,BEM

## **LBN,USE: Tracking L-Band Providers/Geo-satellites**

---

**Function** This command is used to start or stop tracking the specified L-band provider/satellite.  
 Enabling a provider enables all satellites from this provider and disables all satellites from the other providers. Enabling a satellite disables all other satellites.

**Command Format** **Syntax**  
 \$PASHS,LBN,USE,s1,s2[\*cc]

## Parameters

Parameter	Description	Range
s1	L-band provider names or names of specific L-band satellites	See table below.
s2	Tracking request: • ON: Enable L-band tracking (ON by default) • OFF: Disable L-band tracking	ON, OFF
*cc	Optional checksum	*00-*FF

### List of Today's L-Band Providers/Satellites

L-Band Provider or Sat Name	"s1" Field	Provider	Frequency (MHz)	Baud Rate	Sat ID in SLB	Base ID in GGA
RTX	RTX (default)	Trimble RTX				
RTXAP	RTXAP		1539.8325	600	2	102
RTXSA	RTXSA		1539.8325	600	6	106
RTXEA	RTXEA		1545.4900	2400	7	107
RTXIO	RTXIO		1545.5300	600	8	108
RTXNA	RTXNA		1555.8080	2400	9	109
TRS	TRS	TERIA sat				
TERIA	TRS		1546.3100	2400	16	116
User defined					63	99
User defined					62	98
User defined					61	97
User defined					60	96

## Examples

Enabling tracking of all RTX satellites:

```
$PASHS,LBN,USE,RTX,ON
```

Enabling tracking of all RTX satellites:

```
$PASHS,LBN,USE,RTX,ON
```

Tracking only satellite RTXNA:

```
$PASHS,LBN,USE,RTXNA,ON
```

**Related Commands**    \$PASHQ,LBN  
                              \$PASHQ,PAR

## LCK,MOD: Controlling Receiver Lock

---

**Function**    This command is used to allow, or not allow the receiver to be locked.

It is the password defined using \$PASHS,ATH,PWD that must be entered when applying this command.

## Command Format

### Syntax

\$PASHS,LCK,MOD,s1,s2[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Receiver lock control: <ul style="list-style-type: none"> <li>• ON: To authorize receiver lock</li> <li>• OFF: To not authorize receiver lock</li> </ul>	ON, OFF	OFF
s2	Password; See \$PASHS,ATH,PWD	6 to 64 characters	trimble
*cc	Optional checksum	*00-*FF	

### Example

Allowing the receiver to accept a lock command:

\$PASHS,LCK,MOD,ON,trimble

### Comments

- After the receiver has been locked, only the following commands can be applied to the board:
  - \$PASHS,LCK,OFF
  - \$PASHQ,RID
  - \$PASHQ,RCP,OWN
  - \$PASHQ,VERSION
  - \$PASHQ,LCK
- The board firmware cannot be upgraded when receiver lock is authorized (even when the board is not currently locked).

### Related Commands

\$PASHS,LCK,ON  
 \$PASHS,LCK,OFF  
 \$PASHS,ATH,PWD

### Query Command

\$PASHQ,LCK

## LCK,OFF: Unlocking the Receiver

---

### Function

This command is used to unlock the receiver.

**Command Format Syntax**`$PASHS,LCK,OFF,s1[*cc]`**Parameters**

Parameter	Description	Range	Default
s1	Password; See \$PASHS,ATH,PWD	6 to 64 characters	trimble
*cc	Optional checksum	*00-*FF	

**Example**

Unlocking the receiver:

`$PASHS,LCK,MOD,OFF,trimble`

**Related Commands** \$PASHS,LCK,MOD  
 \$PASHS,LCK,ON  
 \$PASHS,ATH,PWD

**Query Command** \$PASHQ,LCK

**LCK,ON: Locking the Receiver**

---

**Function** This command is used to lock the receiver. The prior condition for this command to be effective is the receiver should have been set to accept a locking command (see command \$PASHS,LCK,MOD).

When the receiver is locked, it won't operate normally next time you power it on until you provide a password through command \$PASHS,LCK,OFF.

**Command Format Syntax**`$PASHS,LCK,ON,s1[*cc]`**Parameters**

Parameter	Description	Range	Default
s1	Password; See \$PASHS,ATH,PWD	6 to 64 characters	trimble
*cc	Optional checksum	*00-*FF	

**Example**

Locking the receiver:

`$PASHS,LCK,ON,trimble`

- Comments**
- When locked, the receiver can only respond to the following commands:
    - \$PASHS,LCK,OFF
    - \$PASHQ,RID
    - \$PASHQ,RCP,OWN
    - \$PASHQ,VERSION
    - \$PASHQ,LCK
  - When the receiver is locked, there is no way you can upgrade or downgrade the receiver firmware.

**Related Commands** \$PASHS,LCK,MOD  
 \$PASHS,LCK,OFF  
 \$PASHS,ATH,PWD

**Query Command** \$PASHQ,LCK

## LCS: Enabling/Disabling Use of Local Coordinate System

---

**Function** This command is used to enable or disable the use of the local coordinate system in the receiver. Having the receiver using a local coordinate system requires that it receives RTCM 3.1 message type 1021-1028 from the base. Use exclusively NMEA output messages to get the position expressed in local coordinates.

**Command Format** **Syntax**

\$PASHS,LCS,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	ON: Local coordinate system used if RTCM 3.1 messages received (1021, 1022, 1023, 1025). OFF: Coordinate system used is WGS84. HOR: Horizontal coordinates are local ones VER: Vertical coordinates are local ones	ON, OFF, HOR, VER	OFF
*cc	Optional checksum	*00-*FF	-

### Example

Enabling the use of the local coordinate system in the receiver:

**\$PASHS,LCS,ON\*04**

**Related Commands** \$PASHQ,LCS (query command)  
\$PASHQ,PAR

**LOG,DEL: Deleting Log Files**

---

**Function** This command is used to delete log files.

**Command Format** **Syntax**  
\$PASHS,LOG,DEL,d[\*cc]

**Parameters**

Parameter	Description	Range
d	Index of the log file you want to delete. Use the \$PASHQ,LOG, LST command to read the index associated with each existing log file. Use d=999 to delete all the log files, but the current one.	0 to no limit
*cc	Optional checksum	*00-*FF

**Example**

Deleting all log files:

\$PASHS,LOG,DEL,999\*45

**Related Commands** \$PASHQ,LOG,LST (query command)  
\$PASHQ,LOG

**LOG,PAR: Log File Settings**

---

**Function** This command is used to set the log file. A log file keeps track of the different connections performed in a day (one file created per day).

**Command Format** **Syntax**  
\$PASHS,LOG,PAR,s1,d2,d3[\*cc]

## Parameters

Parameter	Description	Range	Default
s1	Enabling/disabling the log file: <ul style="list-style-type: none"> <li>• ON: Enable</li> <li>• OFF: Disable</li> </ul>	ON, OFF	ON
d2	Maximum size, in Mbytes, allowed for a log file.	1-90	1
d3	Number of days during which log files are kept in memory. After this delay, they are automatically deleted.	1-100	10
*cc	Optional checksum	*00-*FF	

## Example

Enabling the log file with a maximum size of 2 Mbytes and 10 days of backup:

```
$PASHS,LOG,PAR,ON,2,10*40
```

**Related Commands**    \$PASHQ,LOG (query command)  
                           \$PASHS,LOG,DEL  
                           \$PASHS,LOG,LST

## MDP: Choosing Communication Mode on Port D

---

**Function**    This command is used to set port D in the desired communication mode.

**Command Format**    **Syntax**

```
$PASHS,MDP,c1,s2[*cc]
```

## Parameters

Parameter	Description	Range
c1	Port number (D)	D
s2	Select the desired communication mode: <ul style="list-style-type: none"> <li>• 232: RS232 serial port (default)</li> <li>• CAN: CAN bus</li> <li>• I2C: I2C port</li> </ul>	232, CAN, I2C
*cc	Optional checksum	*00-*FF

## Example

Setting port D as an I2C port:

**\$PASHS,MDP,I2C\*38**

**Related Commands** \$PASHQ,MDP (query command)  
\$PASHS,PRT

## **MEM: Selecting Memory Device Used**

---

**Function** This command is used to select the memory used by the receiver for data storage.

**Command Format** **Syntax**

**\$PASHS,MEM,d[\*cc]**

### **Parameters**

Parameter	Description	Range	Default
d	Memory used: • 0: Internal memory • 2: USB memory	0, 2	0
*cc	Optional checksum	*00-*FF	

### **Example**

Selecting internal memory as the memory used by the receiver:

**\$PASHS,MEM,0\*2C**

**Related Commands** \$PASHQ,MEM (query command)  
\$PASHS,FIL,DEL

## **MET,OWN - MET,REF: Entering Meteorological Parameters**

---

**Function** These two commands are used to enter the current values of three meteorological parameters respectively for the local



receiver, or for the reference receiver your board is working with.

## Command Format Syntax

`$PASHS,MET,OWN,f1,f2,f3[*cc]`

or

`$PASHS,MET,REF,f1,f2,f3[,d4][*cc]`

## Parameters

Parameter	Description	Range
f1	Local temperature, in degrees Celsius.	±100
f2	Local pressure, in mBar	0-1300
f3	Local relative humidity, in percent	0-100
d4	Preference	0, 1
*cc	Optional checksum	*00-*FF

## Comments

- As an RTK base, the board will insert its own f1 to f3 parameters into a meteo message such as ATM,ATR,MET.
- As an RTK rover, the board can use the meteo parameters corresponding to the local (OWN) and reference (REF) boards to adjust troposphere models.
- As an RTK rover, the board will use or ignore the meteo data entered locally for the reference receiver (REF), depending on the value given to d4 when running the `$PASHS,MET,REF` command:
  - d4=0: Meteo data entered locally for REF will be ignored if incoming reference data contain such data.
  - d4=1: Always used, regardless of what's decoded from the incoming reference data.
- The default values of meteo parameters for both the local and reference receivers are set to “unknown” so that the PVT engine can internally apply its own defaults for these parameters. When a binary meteo message is generated, then these unknown default values will be mapped accordingly (i.e. they will be delivered as invalid values).

## Example

Entering meteo values for the local receiver:

`$PASHS,MET,OWN,+21,+1200,+51*50`

## MSG: Defining a User Message

---

**Function** This command is used to define a user message to be generated “as is” or for insertion into one or more of the following standard messages:

- ATOM,ATR,UEM, if requested
- RTCM-2 message type 16 or 36
- RTCM-3 message type 1029
- CMR type 2 or 10

Any new MSG command you send to the receiver overwrites the user message previously defined through the same command.

### Command Format Syntax

`$PASHS,MSG,[c1],s2[*cc]`

### Parameters

Parameter	Description	Range	Default
c1	Message interpreter (provision for future uses): • 0: ASCII message generated periodically. If c1 is omitted, this is interpreted as c1=0	0	0
s2	User message	100 characters max.	(receiver name)
*cc	Optional checksum	*00-*FF	

### Example

Sending ASCII message “going to stop at 11pm” periodically:

`$PASHS,MSG,0,going to stop at 11pm*5F`

### Comments

The port and output rate used to deliver the user message depends on the output message chosen for conveying this user message.

**Warning!** Only the first 90 characters from the user message will actually be transferred through RTCM-2 message types 16 and 36.

## NME: Enabling/Disabling NMEA Messages

**Function** This command is used to enable or disable NMEA messages and NMEA-like messages.

### Command Format Syntax

```
$PASHS[d0],NME,s1,c2,s3[f4][*cc]
```

### Parameters

Parameter	Description	Range
d0	Sensor ID	1, 2
s1	Data message type	See tables below
c2	Port routing the message: <ul style="list-style-type: none"> <li>• <b>A, B, D</b>: Serial ports</li> <li>• <b>C</b>: USB serial port</li> <li>• <b>P, Q</b>: TCP/IP client streams</li> <li>• <b>I, F</b>: TCP/IP server streams</li> <li>• <b>M</b>: G-file (since v.3.50); Internal memory before v3.50</li> <li>• <b>U</b>: USB memory (before v.3.50); Since v3.50, does not exist anymore.</li> </ul>	A-D, F, I, M, P, Q (U)
s3	Enables (ON) or disables (OFF) the message. OFF by default.	ON, OFF
f4	Output rate, in seconds. Default is 1 second.	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600
*cc	Optional checksum	*00-*FF

### NMEA Messages

Data	Description	Out Rate
DTM	Datum Reference	1 s
GBS	GNSS satellite fault detection	See f4
GGA	Global Positioning System fix data	See f4
GLL	Geographic position - Latitude / Longitude	See f4
GMP	GNSS Map Projection Fix Data	See f4
GNS	GNSS Fix Data	See f4
GRS	GNSS range residual	See f4
GSA	GNSS DOP and active satellites	See f4
GST	GNSS pseudo-range error statistics	See f4
GSV	GNSS satellites in view	See f4
HDT	True heading	See f4

Data	Description	Out Rate
RMC	Recommended minimum specific GNSS data	See f4
ROT	Attitude rate and accuracy	See f4
THS	True heading and status	See f4
VTG	Course over ground and ground speed	See f4
ZDA	Time and date	See f4

## NMEA-like messages:

Data	Description	Out Rate
ALR	Alarms (see \$PASHS,NME,ALR)	No rate
ARR	Vector and accuracy (3DF mode)	See f4
ATT	True heading	See f4
AVR	Trimble "Time-Yaw-Tilt-Range" Message for Moving Baseline RTK	See f4
CAP	Received base antenna (see \$PASHQ,ANP,RCV)	No rate
CPA	Received antenna height (see \$PASHQ,CPD,ANT)	No rate
CPO	Received base position (see \$PASHQ,CPD,POS)	
DDM	Differential decoder message	No rate
DDS	Differential decoder status	1 s min.
GGK	Position (Trimble proprietary message). See Trimble documentation.	See f4
GGKX	Position & accuracy data (Trimble proprietary message). Message actually substitutes the combo GGA+GST. See Trimble documentation.	See f4
HPR	Attitude message	See f4
LTN	Latency	See f4
POS	Position	See f4
PSP	Received physical reference station position	No rate
PTT	PPS time tag	No rate
RCA	Received CMR type 2 attributes	No rate
RCS	Recording status	1 s min.
RSP	Received reference station position	No rate
SBD	Beidou satellite status	See f4
SGA	Galileo satellite status	See f4
SGL	Glonass satellite status	See f4
SLB	L-Band satellite status	See f4
SQZ	QZSS satellite status	See f4
SSB	SBAS satellite status	See f4
TEM	Board temperature	1 s min.
TTT	Event marker (see \$PASHS,NME,TTT)	No rate
VCR	Vector and accuracy	See f4
VCT	Vector (with static base)	See f4
VEL	3D velocity and velocity accuracy	-

**Example**

Setting GGA message on Bluetooth port at 1-second output rate:

```
$PASHS,NME,GGA,C,ON,1*01
```

**Relevant Query Command** \$PASHQ,PAR (query command)

\$PASHS,NME,ALL

## NME,ALL: Disabling All NMEA and NMEA-Like Messages

---

**Function** This command is used to disable all NMEA messages and NMEA-like messages currently enabled on the specified port.

**Command Format** **Syntax**

```
$PASHS[,d0],NME,ALL,c1,OFF[*cc]
```

**Parameters**

Parameter	Description	Range
d0	Sensor ID (antenna ID; 1: primary, 2: secondary); 1 if omitted.	1, 2
c1	Port routing the message: <ul style="list-style-type: none"> <li>A, B, D: Serial</li> <li>C: USB serial</li> <li>I, P, Q, F: TCP/IP stream</li> <li>M: Internal memory</li> <li>U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
*cc	Optional checksum	*00-*FF

**Example**

Disabling all NMEA and NMEA-like messages on port A:

```
$PASHS,NME,ALL,A,OFF*50
```

**Related Commands** \$PASHS,NME

## NME,ALR: Setting an Alarm Message

---

**Function** This command is used to output an alarm message on the specified port. The alarm level can be set to filter alarms.

### Command Format Syntax

\$PASHS,NME,ALR,c1,s2[,d3][\*cc]

### Parameters

Parameter	Description	Range
c1	Port routing the message: <ul style="list-style-type: none"> <li>• A, B, D: Serial</li> <li>• C: USB serial</li> <li>• I, P, Q, F: TCP/IP stream</li> <li>• M: Internal memory</li> <li>• U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
s2	ON/OFF control	ON, OFF
d3	Alarm level: <ul style="list-style-type: none"> <li>• 0: High, medium and low</li> <li>• 1: High and medium only</li> <li>• 2: High only</li> </ul>	0-2
*cc	Optional checksum	*00-*FF

**Response** The message returned by the receiver on applying the NME,ALR command is described below.

### Syntax

\$PASHR,ALR,d1,d2,c3,s4,d5,s6\*cc

## Parameters

Parameter	Description	Range
d1	Alarm code	0-255
d2	Alarm sub-code	0-255
c3	Port routing the message: <ul style="list-style-type: none"> <li>• A, B, D: Serial</li> <li>• C: USB serial</li> <li>• I, P, Q, F: TCP/IP stream</li> <li>• M: Internal memory</li> <li>• U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
s4	Source of alarm	BLUETOOTH INPUT MEMORY MODEM NETWORK OTHER POWER PVT RADIO WIFI
d5	Alarm level: <ul style="list-style-type: none"> <li>• 0: High, medium and low</li> <li>• 1: High and medium only</li> <li>• 2: High only</li> </ul>	0-2
s6	Description	
*cc	Optional checksum	*00-*FF

### Example

```
$PASHS,NME,ALR,C,ON
$PASHR,ALR,45,0,C,GSM,2,"PIN code invalid"*AC
```

**Related Commands**    \$PASHS,NME

## NME,TTT: Requesting Output of Event Marker

---

**Function**    This command is used to issue an event marker message. The GPS time is provided to within 1 $\mu$ s following the time of request.

This command should not be confused with \$PASHS,NME where s1 would be given the value "TTT" (see this command for which event marker messages are requested by the event signal applied to the dedicated pin on the I/O connector and messages are output on the port specified by the command).

**Command Format Syntax**

\$PASHS,NME,TTT,c1,s2[\*cc]

**Parameters**

Parameter	Description	Range
c1	Port routing the message: <ul style="list-style-type: none"> <li>A, B, D: Serial port</li> <li>C: USB serial port</li> <li>I, P, Q, F: TCP/IP stream</li> <li>M: Internal memory</li> <li>U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
s2	Enabling/disabling output	ON, OFF
*cc	Optional checksum	*00-*FF

**Response Format Syntax**

\$PASHR,TTT,d1,m2\*cc

**Parameters**

Parameter	Description	Range
d1	Day in week. 1: Sunday; 7: Saturday	1-7
m2	GPS time tag, in hours, minutes, seconds and fraction of a second (9 decimal places).	0-23:59:59.999999999
*cc	Checksum	*00-*FF

**Example**

Requesting an event marker on port A:

\$PASHS,NME,TTT,A,ON

...

\$PASHR,TTT,3,18:01:33.1200417\*04

**Related Commands** \$PASHS,NME**NPT: Defining How RTX and SBAS Positions are Tagged in POS Messages**

---

**Function** This command allows you to define a set of flags allowing you to tag the RTX and SBAS position solutions delivered in POS messages.



**Command Format Syntax**`$PASHS,NPT,d1,d2[*cc]`**Parameters**

Parameter	Description	Range	Default
d1	Tagging position solution in POS messages: <ul style="list-style-type: none"> <li>• If "0" (standard position type), possible tags are:               <ul style="list-style-type: none"> <li>– 1: SBAS or BeiDou Differential</li> <li>– 2: RTX Float or RTX Float dithered</li> <li>– 3: RTX Fixed or RTX Fixed dithered</li> </ul> </li> <li>• If "1" (specific position type), possible tags are:               <ul style="list-style-type: none"> <li>– 9: SBAS differential</li> <li>– 10: BeiDou Differential</li> <li>– 12: RTX Float</li> <li>– 13: RTX Fixed</li> <li>– 22: RTK Float dithered</li> <li>– 23: RTK Fixed dithered</li> </ul> </li> </ul>	0,1	1
d2	This field is ignored.		
*cc	Optional checksum	*00- *FF	

**Example**

Choosing to report standard position types in POS messages:

`$PASHS,NPT,0`

**Related Commands** `$PASHQ,NPT`  
`$PASHQ,PAR`

**NTP: Controlling the Network Time Protocol Server**

---

**Function** This command is used to enable or disable the NTP (Network Time Protocol) server in the receiver.

The NTP server will respond to IP requests for GNSS time only after the receiver has been computing GNSS time.

The NTP server is useful to synchronize devices on a network.

**Command Format Syntax**  
`$PASHS,NTP,s[*cc]`

## Parameters

Parameter	Description	Range
s	NTP server control: <ul style="list-style-type: none"> <li>• ON: Enables NTP server</li> <li>• OFF: Disables NTP server</li> </ul>	ON, OFF
*cc	Optional checksum	*00-*FF

## Example

Starting the NTP server:

```
$PASHS,NTP,ON*12
```

**Query Command**    \$PASHQ,NTP

## NTR,LOD: Loading the NTRIP Caster Source Table

---

**Function**    This command is used to load the source table from the NTRIP caster. The receiver stores only one source table. If there is already one source table stored in the receiver when running the NTR,LOD command, this table will be overwritten.

**Command Format**    **Syntax**

```
$PASHS,NTR,LOD[,c1][*cc]
```

## Parameters

Parameter	Description	Range
c1	Internet port used to connect to the caster: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul> <p>If c1 is omitted, the port used is the port defined through the last \$PASHS,NTR,PAR command run.</p>	P, Q
*cc	Optional checksum	*00-*FF

## Example

Loading the source table:

```
$PASHS,NTR,LOD
```

If the source table is downloaded successfully, the following response line will be returned:

```
$PASHR,NTR,OK*14
```

If the receiver fails to download the source table, the following response line will be returned:

```
$PASHR,NTR,FAIL*12
```

**Related Commands**    \$PASHQ,NTR,TBL  
                           \$PASHS,NTR,PAR

## **NTR,MTP: Connecting Receiver to NTRIP Caster Mount Point**

---

**Function**    This command allows you to connect the receiver to a NTRIP caster mount point.

**Command Format**    **Syntax**

```
$PASHS,NTR,MTP,s1[,c2][*cc]
```

### **Parameters**

Parameter	Description	Range
s1	Name of the NTRIP mount point, or OFF command (ending the connection to the current mount point).	100 characters max., or OFF
c2	Internet port used to connect to the caster: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul> If c2 is omitted, the port used is the port defined through the last \$PASHS,NTR,PAR command run.	P, Q
*cc	Optional checksum	*00-*FF

### **Example**

Connecting to mount point MUWFO:

```
$PASHS,NTR,MTP,MUWFO*4D
```

If the connection is successful, the following response line will be returned:

```
$PASHR,NTR,OK*cc
```

If the connection failed, the following response line will be returned:

```
$PASHR,NTR,FAIL*12
```

**Related Commands** \$PASHQ,NTR,TBL

## NTR,PAR: NTRIP Settings

---

**Function** This command allows you to set all the NTRIP parameters.

**Command Format Syntax**  
**\$PASHS,NTR,PAR**[,ADD,s1][,PRT,d2][,LGN,s3][,PWD,s4][,TYP,d5  
 ][,IPP,c6][,ACN,d7][\*cc]

### Parameters

Parameter	Description	Range
ADD,s1	Caster IP address or host name	32 characters max.
PRT,d2	Caster port number	0-65535
LGN,s3	Login	64 characters max.
PWD,s4	Password	32 characters max.
TYP,d5	Caster type: <ul style="list-style-type: none"> <li>• 0: Client</li> <li>• 1: Server</li> <li>• 2: Client NTRIP V2</li> <li>• 3: Server NTRIP V2</li> </ul>	0-1
IPP,c6	Internet port used to connect to the caster: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul>	P, Q
ACN,d7	Auto-connection: <ul style="list-style-type: none"> <li>• 0: After a power cycle, there won't be an automatic NTRIP connection (default)</li> <li>• 1: After a power cycle, the NTRIP connection will be restored</li> </ul>	0, 1
*cc	Optional checksum	*00-*FF

### Example

Entering NTRIP settings for a client caster by specifying its IP address, port number, login and password:

```
$PASHS,NTR,PAR,ADD,192.34.76.1,PRT,2100,LGN,trimble,PWD,
u6huz8,TYP,0,IPP,P,ACN,1
```

### Comments

- P and Q are generic TCP/IP ports. By default, the board determines which physical port can be used (Modem, Wifi, Ethernet). You can however specify which physical

port should be used through command \$PASHS,TCP, RTE.

- The caster settings for each port (P and Q) are saved independently of each other meaning that several NTRIP connections can be used at the same time.
- The last \$PASHS,NTR,PAR command issued determines which of the two ports (P or Q) are used by default in commands \$PASHS,NTR,MTP and \$PASHS,NTR,LOD.

**Related Commands**

\$PASHQ,NTR  
 \$PASHQ,PAR  
 \$PASHS,NTR,MTP  
 \$PASHS,NTR,LOD

## OBS: Setting Preference for GNSS Signals Tracking

---

**Function** This command is used to choose how to best use the available GNSS signals. Changing the preference by running this command may cause the board to restart.

**Command Format** **Syntax**

\$PASHS,OBS,c1[\*cc]

### Parameters

Parameter	Description	Range	Default
c1	Tracks all GNSS signals (ALL), or only the best (OPT or LEG).	ALL, OPT, LEG	OPT
*cc	Optional checksum	*00-*FF	

### Example

Enabling the tracking of all GNSS signals:

\$PASHS,OBS,ALL

**Comments**

- Choosing “OPT” does not mean disabling all the signals from a particular band, but simply tracking one band (the best) per GNSS band.  
 For example, with GLONASS, choosing “ALL” means tracking L1CA, L2CA, L1P and L2P, while choosing “OPT” means tracking L1CA and L2CA only.

- The “LEG” option is similar to the “OPT” option, except that only the L1P and L2P (not L2C) signals are output for all GPS satellites. This option is intended for users who are not comfortable with mixing L2P and L2C GPS observations.
- If you choose “ALL” Whatever the option you choose for OBS, the only way you can see all the tracked data is through one of these messages, internally defined as raw data messages raw and differential data messages delivering receiver observables to users are the following:
  - ATM,ADM
  - ATM,RNX,scn,0&999
  - MSM 5&7
  - ATM,PVT (tracking status and SNR/CTT only)

All other messages, defined internally as differential messages, such as those listed hereafter, will only generate the following reduced data subsets: OPT data subset if “OPT” or “ALL” option used, or LEG data subset if “LEG” option used.

- ATM,RNX (all scenarios except 0&999)
- RTCM-3.1
- RTCM-2.3
- RTCM-3.2 MSM 1&2&3&4&6
- CMR

Note 1: The legacy RTCM-2.3 and CMR differential messages do not make any distinction between data flagged GPS L2P(Y) and GPS L2C.

Note 2: SBAS observables are not provided in differential messages, but are however available in raw data messages.

Note 3: The baseline engine does not use SBAS ranging data. The only mode where SBAS ranging is used is internal heading in SNS,DUO mode.

- Conversely, the following messages will deliver sets of observables corresponding to the “OPT” choice, regardless of whether you chose “ALL” or “OPT”:
  - ATM,RNX (all scenarios except 0&999)
  - RTCM-3.1
  - RTCM-2.3
  - RTCM-3.2 MSM 1&2&3&4&6
  - CMR

- The tracking/usage status reported in ATM,PVT and ATM,RNX (except for scenario 999) should be the same. There may be a difference however because ATM,RNX scenarios may additionally mask some signals originally presented in ATM,PVT.  
Reminder: The tracking/usage status includes such information as tracked/used signals, SNR, Cumulative Tracking Time and warnings.
- **Important!** Switching from SOL (single) to Duo mode (see SNS command) will result in resetting the OBS command to option “OPT”. Conversely, an attempt to set the “ALL” option with the OBS command will always fail if the board is currently in DUO mode.

## OCC: Writing Occupation Data to Raw Data File

---

**Function** This command is used to write information about the current occupation to the raw data file being logged.  
Every time this command is issued, its content is duplicated in the ATM,ATR,OCC message (if requested on a port).

### Command Format Syntax

```
$PASHS,OCC,d1,d2,s3[s4][*cc]
```

### Parameters

Parameter	Description	Range
d1	Occupation type: • 0: Static • 1: Quasi-static • 2: Dynamic • 3: Event	0-3
d2	Occupation event: • 0: Begin • 1: End	0-1
s3	Occupation name	255 characters max.
s4	Occupation description	255 characters max.
*cc	Optional checksum	*00-*FF

### Examples

Starting a static occupation on point “SITE01”:

```
$PASHS,OCC,0,0,SITE01,Park_Entrance*63
```

Ending the static occupation on point “SITE01”:

```
$PASHS,OCC,0,1,SITE01,Park_Entrance*62
```

**Related Commands**    \$PASHQ, OCC  
                               \$PASHS, REC  
                               \$PASHS, ATM

## OPTION: Receiver Firmware Options

---

**Function**            This command is used to install the receiver firmware options that are purchased at a later date. Options purchased at the time of receiver purchase are factory pre-loaded.

### Command Format    Syntax

`$PASHS, OPTION, s1, h2[*cc]`

Disabling all options and resetting/rebooting the receiver:  
`$PASHS, OPTION, ALL, 0[*cc]`

### Parameters

Parameter	Description	Range
s1	Option ID	(See table below)
h2	Unlock code (or "0" to remove the option)	13 characters max.
*cc	Optional checksum	*00-*FF

Option ID	Label	Description
@1	GEOFENCING_WW	Worldwide enabled receiver
@2	GEOFENCING_CHINA	China-only enabled receiver
@3	GEOFENCING_JAPAN	Japan-only enabled receiver
@4	GEOFENCING_BRAZIL	Brazil-only enabled receiver
@5	GEOFENCING_N_AMERICA	North-America-only enabled receiver
@6	GEOFENCING_L_AMERICA	Latin-America-only enabled receiver
@7	GEOFENCING_RUSSIA	Russia-only enabled receiver
@8	GEOFENCING_INDIA	India-only enabled receiver
@9	GEOFENCING_TURKEY	Turkey-only-enabled receiver
2	2HZ	Enables output rate up to 2 Hz (for position/raw data)
4	ViewPoint RTX	RTX L1-only mode



Option ID	Label	Description
5	5HZ	Enables update rate up to 5 Hz for position and raw data. With this option enabled, when using \$PASHS,POP to set the internal update rate, remember only "POP,1" and "POP,5" are allowed, not "POP,2".
6	10HZ	Enables update rate up to 10 Hz for position and raw data
8	50HZ	Enables update rate up to 50 Hz for position and raw data
9	FieldPoint RTX	CPT degraded version. Dominates RangePoint RTX
B	BEIDOU	BEIDOU satellites tracking
C	CPRTX	Enables Trimble CenterPoint RTX
c	CASTER	Embedded NTRIP caster
D	DUO	Allows use of dual-sensor mode, raw data output from both sensors, internal heading between the two antennas, or two independent standalone/DGPS/SBAS position for each antenna. External heading as well.
E	ATTITUDE	Same as D option. Additionally, enables full-attitude engine with up to 4 sources of raw data (up to 4 antennas) regardless of their assignment across two or more boards. Computing external heading also possible.
F	FLYINGRTRK	Enables Flying RTK computation (position and vector). [J] predominates over [F]. See NOTE 2.
G	GLONASS	GLONASS satellites tracking
I	RAIM	Enables receiver to output RAIM-related messages
J	RTKROVER	Allows fixed RTK computations (position and vector). ([J] option predominates over [F] option. See NOTE 2.
K	RTKBASE	Enables the receiver to generate differential messages: <ul style="list-style-type: none"> <li>• RTCM 2.3</li> <li>• RTCM 3.0</li> <li>• CMR / CMR+</li> <li>• ATOM</li> </ul> See NOTE 3.
L	LBAND	Enables L-band reception
N	GPS	GPS+SBAS+QZSS satellites tracking
O	GALILEO	GALILEO satellites tracking

Option ID	Label	Description
P	RPRTX	Enables Trimble RangePoint RTX
R	RECORD	Enables data recording in memory. ATL.LOG data can be recorded without this option installed.
W	20HZ	Enables output rate up to 20 Hz (for position/raw data)
W1	WARRANTY 1 YEAR	Extends warranty period by one year. See NOTE 4.
W2	WARRANTY 2 YEAR	Extends warranty period by two years. See NOTE 4.
X	L1 TRACKING	Enables tracking of the following signals: <ul style="list-style-type: none"> <li>• GPS, QZSS, SBAS L1</li> <li>• GLONASS G1</li> <li>• GALILEO E1</li> <li>• BEIDOU B1 (phase 2)</li> </ul> This option is ALWAYS set.
Y	L2 TRACKING	Enables tracking of the following signals: <ul style="list-style-type: none"> <li>• GPS, QZSS L2</li> <li>• GLONASS G2</li> <li>• GALILEO E5b</li> <li>• BEIDOU B2</li> </ul>

NOTE 1: If option [J] is installed, then using the board as RTK rover is possible with naturally achievable accuracy.

NOTE 2: If there is no [F] or [J] option installed, the board can receive any set of corrections (ATOM, RTCM, CMR, etc.) but output only DGPS/DGNSS decimeter-level position by using/applying the L1 C/A code measurements portion from these correcting data.

Adding the [F] option gives the board the ability to run in Flying RTK (position and vector) using additionally carrier phase data from multiple signals.

Adding the [J] option gives the board the ability to run in full RTK (position and vector), up to fixing carrier ambiguity to integer. No dithering is applied with [J] installed.

Option [J] always dominates [F]. An RTK rover using the CMRx protocol will not be able to start with option [F]: option [J] will be required to make it work.

NOTE 3: If option [K] is missing, the board cannot generate most of the RTCM2, RTCM3, CMR and ATOM,RNX messages. The following exceptions are possible on each port (with possible limitations if [A] and [V] are not installed):

- ATOM,RNX scenario 0 and 204

- RTCM-2.3 type 1,3,9,31

- RTCM-3.2 type 1005,1006,1071,1081,1091,1101,1111,1211

Once option [K] has been installed, all differential and ATM,RNX messages can be generated on each port.

NOTE 4: Options [W1] and [W2] modify the expiration date in the board. The unlock code (or password) is computed as a one-month temporary option using the current expiration date as the start date.

### Comments

- Firmware options may be activated for limited periods of time, depending on the type of unlock code generated for each of them (**WARNING: in MB-Two, only firmware options tied to Trimble RTX may be made temporary**).
- Options [D] and [E] are totally independent of options [K] [F] and [J].

When option [D] is installed in a board, the second antenna sensor can be turned on and the board can compute heading between the two antennas. But the board cannot compute either heading implying the processing of data external to the board, or attitude using external data or even data from the other internal sensor. Option [D] is a subset of option [E], so you don't need to install option [D] if option [E] is already installed.

- Option [E]: When this option is installed, compared to option [D], you can additionally compute heading or attitude in which the board's second antenna sensor may be involved, as well as data external to the board, provided these data are in ATM,RNX,scn,0 (or 204) format.
- If none of the update rate options are installed ([2], [5], [6], [8], [W]), the board will output data at up to 1 Hz speed.

### Example

Enabling the [K] option:

```
$PASHS,OPTION,K,878A887*cc
```

**Related Commands**    \$PASHQ,OPTION (query command)  
\$PASHQ,RID

## OUT: Suspending/Resuming Message Output

---

**Function**    This command is used to suspend or resume the output of the requested periodic messages (those requested using \$PASHS commands) on the specified port.

**Command Format Syntax**`$PASHS,OUT,c1,s2[*cc]`**Parameters**

Parameter	Description	Range
c1	ID of port on which to suspend or resume the output of periodic messages: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A-D, F, I, P, Q
s2	Suspend (OFF) or resume (ON) message output control	ON, OFF
*cc	Optional checksum	*00-*FF

**Comments**

- Suspending all periodic messages on a given port implies that the receiver “keeps in mind” all the settings of these messages. When later you apply the “ON” command, the receiver will resume the output of these messages as if nothing had happened in the meantime.
- With all the periodic messages suspended on a given port, you are still allowed to modify the settings of these suspended messages, or even add new ones.
- With all the periodic messages suspended on a given port, you are still allowed to apply \$PASHQ commands on this port to get \$PASHR responses from the receiver through the same port.

**Example**

Suspending all messages on port A:

`$PASHS,OUT,A,OFF*35`

**Related Commands**    \$PASHQ,OUT (query command)  
                               \$PASHS,NME  
                               \$PASHS,ATM

**OUT,ALL: Disabling All Periodic Messages**

---

**Function**    This command is used to disable all the periodic messages programmed to be output on the specified port. Periodic

messages are all those programmed using \$PASHS commands.

## Command Format Syntax

\$PASHS,OUT,ALL,c1,OFF[\*cc]

### Parameters

Parameter	Description	Range
c1	ID of port on which to disable the output of periodic messages: <ul style="list-style-type: none"> <li>A, B, D: Serial port</li> <li>C: USB serial port</li> <li>I, P, Q, F: TCP/IP stream</li> <li>M: Internal memory</li> <li>U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
*cc	Optional checksum	*00-*FF

### Example

Disabling all periodic messages on port A:

\$PASHS,OUT,ALL,A,OFF\*58

**Related Commands** \$PASHS,NME  
\$PASHS,ATM

## OUT,DIF,OFF: Disabling All Differential Messages

---

**Function** This command is used to disable the output of all differential messages on all ports.

Using this command is equivalent to using this series of commands, applied to the receiver to disable all differential messages successively on ports A to D, F, I, P and Q:

```
$PASHS,RT3,ALL,c,OFF
$PASHS,RT2,ALL,c,OFF
$PASHS,CMR,ALL,c,OFF
$PASHS,ATM,ALL,c,OFF
```

## Command Format Syntax

\$PASHS,OUT,DIFF,OFF[\*cc]

## Parameters

None.

# PAR,LOD: Configuring the Receiver From a PAR File

---

**Function** This command configures the receiver in one step, using the data stored in the specified PAR file. The PAR file may have been saved previously to the receiver's internal memory or on a USB key.

## Command Format Syntax

```
$PASHS,PAR,LOD[,d1][,s2][*cc]
```

## Parameters

Parameter	Description	Range	Default
d1	Memory where the PAR file can be found: <ul style="list-style-type: none"> <li>• 0: Internal memory (NAND Flash)</li> <li>• 2: USB key</li> </ul> If d1 is omitted, the receiver will assume that the PAR file is on the USB key.	0, 2	2
s2	File name (MB2_SSSSS_dddhhmmss.par) where: <ul style="list-style-type: none"> <li>• SSSSS: Last 5 digits from serial number</li> <li>• ddd: Day number (1.. 366)</li> <li>• hhmmss: Time</li> </ul> If s2 is omitted, the receiver checks that only one PAR file is found in the specified memory. If that is the case, the receiver will be configured according to this file. If several PAR files are found, then \$PASHR,NAK will be returned and the receiver will keep its current configuration.	-	-
*cc	Optional checksum	*00-*FF	-

## Examples

Changing the receiver configuration by loading the PAR file saved on the USB memory:

```
$PASHS,PAR,LOD*5D
```

Changing the receiver configuration by loading the PAR file named "MB2\_95685\_145084518.par" located in the internal memory:

```
$PASHS,PAR,LOD,0,MB2_95685_145084518.par*1A
```

**Relevant Query Command** None.

**See also** \$PASHS,PAR,SAV

## PAR,SAV: Saving the Receiver Configuration To a PAR File

---

**Function** This command is used to save the current receiver configuration to a PAR file.

**Command Format Syntax**  
\$PASHS,PAR,SAV[,d1][\*cc]

### Parameters

Parameter	Description	Range	Default
d1	Memory where the PAR file will be written: <ul style="list-style-type: none"> <li>• 0: Internal memory (NAND Flash)</li> <li>• 2: USB key</li> </ul> If d1 is omitted, the receiver will assume that the PAR file should be saved to the USB key.	0, 2	2
*cc	Optional checksum	*00-*FF	-

### Comments

The command will create a PAR file named as follows:

**MB2\_SSSSS\_dddhhmss.par**

Where:

- SSSSS: Last 5 digits from receiver serial number
- ddd: Day number (1.. 366)
- hhmmss: Current time

The command will be rejected (\$PASHR,NAK) in the following cases:

- No USB key detected and d1=2 or is omitted
- Not enough space available on the specified memory
- The PAR file already exists.

### Example

Saving the receiver configuration to the USB key:

**\$PASHS,PAR,SAV\*5E**

**Relevant Query Command** None.

**See also** \$PASHS,PAR,LOD

## PEM: Setting the Position Elevation Mask

---

**Function** This command is used to set the elevation mask used in the position processing. This value of elevation mask applies to all PVT engines, to all sensors, in all positioning modes, and to all satellite observations.

**Command Format Syntax**

\$PASHS,PEM,d1[\*cc]

### Parameters

Parameter	Description	Range	Default
d1	Elevation mask angle, in degrees	0-30°	5
*cc	Optional checksum	*00-*FF	

### Example

Setting the elevation mask for position processing to 15 degrees:

\$PASHS,PEM,15\*05

**Related Commands** \$PASHQ,PEM  
\$PASHQ,PAR  
\$PASHS,ELM

## PGN: Enabling/Disabling CAN-NMEA 2000 Messages

---

**Function** This command is used to enable or disable NMEA 2000 messages on the CAN bus.

**Command Format Syntax**

\$PASHS[,d0],PGN,s1,c2,s3[,f4][\*cc]



## Parameters

Parameter	Description	Range	Default
d0	Sensor ID (sensor 1 if d0 missing)	1, 2	-
s1	GSOFF message type	See table below.	-
c2	Output port: • CAN port: V	V	-
s3	Enabling/disabling command	ON, OFF	-
f4	Output rate, in seconds	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600	See table below.
*cc	Optional checksum	*00-*FF	

Supported PGN messages:

Message Type	Description	Default Rate (s)	Minimum Rate (s)
126992	System Time	1	0.01
129025	Position, Rapid Update	0.1	0.01
129026	COG & SOG, Rapid Update	0.2	0.01
129027	Position Delta, High Precision Rapid Update	0.1	0.01
129028	Altitude Delta, High Precision Rapid Update	0.1	0.01
129029	GNSS Position Data	1	0.01
129540	GNSS Sats in View	1	0.01
129542	GNSS Pseudorange Noise Statistics	1	0.01

### Example

Enabling “System Time” message at 1 second:

```
$PASHS,PGN,126992,V,ON,1
```

**Related Commands**    \$PASHQ,OUT,PGN - \$PASHQ,PAR (query commands)  
\$PASHS,PGN,ALL

## PGN,ALL: Disabling all CAN-NMEA 2000 Messages

---

**Function**    This command is used to disable all the NMEA 2000 messages currently delivered on the CAN port.

**Command Format Syntax**`$PASHS[d0],PGN,ALL,c1,OFF[*cc]`**Parameters**

Parameter	Description	Range	Default
d0	Sensor ID ("1" if d0 missing)	1, 2	1
c1	Output port delivering NMEA 2000 messages: • CAN port: V	V	-
*cc	Optional checksum	*00-*FF	

**Example**

Disabling all NMEA 2000 messages:

`$PASHS,PGN,ALL,V,OFF`**Related Commands** `$PASHS,PGN`

## PGS: Defining the Primary GNSS System

---

**Function** This command is used to define the primary GNSS system used in the receiver.

The choice of a primary system impacts the following:

- Time tagging of some messages
- Use of a default position datum (e.g. WGS-84 for GPS, PZ-90.02 for GLONASS, CGCS2000 for BeiDou)
- Reference time scale for reported clock estimates. More information on this point in the *ATOM Reference Manual*).

**Command Format Syntax**`$PASHS,PGS,s1[*cc]`

## Parameters

Parameter	Description	Range	Default
s1	Desired primary GNSS system: <ul style="list-style-type: none"> <li>• "GPS": GPS system</li> <li>• "GLO": GLONASS system</li> <li>• "BDS": BEIDOU system</li> <li>• "GAL": Galileo system</li> </ul>	GPS, GLO, BDS, GAL	GPS
*cc	Optional checksum	*00-*FF	

## Comments

- The choice of a primary GNSS system has no impact whatsoever on the internal algorithms used by the receiver. For example, the way channels are assigned for satellite tracking or the way observables from different systems are weighted in the PVT solution are not impacted by this choice.
- The choice of a primary system is an "absolute" setting. That choice is indeed totally independent of the receiver configuration parameters. It remains valid even when the chosen primary system is NOT currently tracked. In this case however, you may expect some approximations due to the use of a priori information about time/datum differences between the different GNSS's.  
 For example, GPS can be designated as the primary system, but may be disabled for tracking using the \$PASHS,GPS,OFF command. In this case, the reported values of time tag and clock offset parameters will only use a priori information about GPS-GLONASS differences.
- The ATM,RNX and ATM,PVT messages are affected when you change the primary GNSS system, owing to a different time tag used. On the other hand, NMEA time tags are not affected as they always refer to UTC time.
- Additional details about the receiver-clock-estimate and clock-steering procedures when the primary GNSS system is GLONASS or BeiDou can be found in the *ATOM Reference Manual*.
- Changing the primary GNSS system causes the board to be restarted. The restart condition is similar to running the \$PASHS,INI,0 command.
- Not all differential protocols can be used when the primary GNSS system is GLONASS or BeiDou. In either case, better use the ATOM or RTCM3.2 data format.
- The messages affected by this command are ATOM RNX, ATOM PVT, all NMEA messages and all reference positions

in differential messages generated for a given primary GNSS system.

- When specifying the position of the receiver through external means (i.e. by using the \$PASHS,POS command), you should always keep in mind that it should be in agreement with the currently selected primary system (i.e. expressed on the same datum).
- GPS is the default primary GNSS system. That's why running \$PASHS,PGS,GLO is recommended when the GPS option is not installed, otherwise raw data cannot be generated (because of unknown GPS time).
- About GNSS time validity: Some GNSS observables may be delivered with invalid time. Internally, the time status for each GNSS may range from INVALID to VALID with many intermediate states (e.g. SET\_BY\_ANOTHER\_SYS; see table below, statuses listed from fully valid to fully invalid).

Time Status Code	Description	Comment on Time Tag
0	TIMEFACE_TIME_VALID	
1	TIMEFACE_TIME_FIRST_CORR	Invalid
2	TIMEFACE_TIME_PRE_FIRST_CORR	Invalid
3	TIMEFACE_TIME_RTC_HOT	Invalid
4	TIMEFACE_TIME_SET_BY_ANOTHER_SYS	Invalid
5	TIMEFACE_TIME_RTC	Invalid
6	TIMEFACE_TIME_NOT_CONFIRMED	Invalid
7	TIMEFACE_TIME_SET_BY_USER	Invalid
8	TIMEFACE_TIME_NOT_SET	Invalid
9-15	Reserved	Invalid, if any

Depending on their current time status, GNSS observables may be processed differently:

- ATOM RNX message scenario 0 (default raw data format) will be output only if the relevant time status is equal to or better than SET\_BY\_ANOTHER\_SYS.
- ATOM RNX messages in all other scenarios will be output only if time is VALID.
- Internally, observables will be used in computing position only if their time status is better than SET\_BY\_ANOTHER\_SYS.

NOTE: There are no messages available to check the time status of each GNSS.

**Example**

Choosing GLONASS as the primary GNSS system:

```
$PASHS,PGS,GLO*59
```

**Related Commands**    \$PASHQ,PGS (query command)  
                           \$PASHQ,PAR

## PHE: Setting the Active Edge of the Event Marker Pulse

---

**Function**            This command is used to set the active edge (rising or falling) of the event marker pulse used in photogrammetry time-tagging.

**Command Format**    **Syntax**  
                           \$PASHS,PHE,c1[\*cc]

**Parameters**

Parameter	Description	Range	Default
c1	Active edge code: <ul style="list-style-type: none"> <li>• "R" for rising edge</li> <li>• "F" for falling edge</li> </ul>	R, F	R
*cc	Optional checksum	*00-*FF	

**Example**

Making the falling edge active:

```
$PASHS,PHE,F*42
```

**Related Commands**    \$PASHQ,PHE

## PIN: Assigning Function to Programmable Pin on I/O Connector

**Function** This command is used to assign a specific function to the programmable pin on the board's I/O connector.

### Command Format Syntax

```
$PASHS,PIN,d1,s2,d3[*cc]
```

### Parameters

Parameter	Description	Range	Default
d1	Programmable pin ID: <ul style="list-style-type: none"> <li>0: MFO1</li> <li>1: MFO2</li> </ul>	0 or 1	-
s2	Function: <ul style="list-style-type: none"> <li>OFF: no function assigned</li> <li>PAV: Position available (see comment 1)</li> <li>RSP: Radar simulated pulse (see comment 2)</li> <li>LED: LED signal output</li> <li>BDL: Trimble BDxxx LED signal output (see <i>Trimble BDxxx LED Functionality and Operation on page 263</i>).</li> <li>GPO: General Purpose Output</li> </ul>	OFF, PAV, RSP, LED, BDL, GPO	LED (red LED for MFO1, green LED for MFO2)

Parameter	Description	Range	Default
d3	<p>Depends on s2 value:</p> <ul style="list-style-type: none"> <li>s2=OFF: d3 should be omitted</li> <li>s2= PAV: d3 is the number of consecutive seconds while position is not computed that causes the signal on the pin to switch to low level.</li> <li>s2= RSP: d3 should be set to "1" to make Radar Simulated Pulse output available on the pin.</li> <li>s2=LED: Defines the LED used. "1" for red LED; "2" for green LED.</li> <li>s2= BDL; d3=1 means sat tracking status is output; d3=2 means RTK link status is output; d3=3 means power status is output.</li> </ul> <p>If d3 is omitted or empty, then sat tracking status is output for MFO1 and RTK link status is output for MFO2.</p> <ul style="list-style-type: none"> <li>s2= GPO: d3=0 means active logic level is "0"; d3=1 means active logic level is "1".</li> </ul>	<p>Omitted1-3600</p> <p>1</p> <p>1 or 2</p> <p>1-3</p> <p>0 or 1</p>	-
*cc	Optional checksum	*00-*FF	

### Example

Assigning the PAV function to the programmable pin and asking for a low-level signal on that pin after 60 seconds of position unavailability:

```
$PASHS,PIN,0,PAV,60*7F
```

### Comment

- When the PAV function is assigned to the pin, a low level will occur on the pin in the following cases:
  - Immediately after the \$PASHS,PIN,d1,PAV command has been executed and the position has not been computed yet at that time.
  - If for any reason, the position has not been computed for more than the time specified as d3.
  - After running the \$PASHS,INI,x,y,0 or \$PASHS,INI,x,y,5 command.
  - After running the POP and <GNS>,USE commands following a re-start of the board.

High level is restored on the pin right after the board has re-started computing the position.

2. After the \$PASHS,PIN,d1,RSP,1 command has been accepted, and if the ground speed is within the range 0.8 to 322 km/hr, an LV-TTL pulse signal with a 50% duty cycle is made available on the pin. The frequency conversion is then 94 Hz/(m/s).

If the ground speed is out of the range 0.8 to 322 km/hr, the pin level is kept at a high level.



## Trimble BDxxx LED Functionality and Operation

The MFO1 and MFO2 pins of MB-Two board can be programmed to drive external LEDs similar to Trimble BDxxx boards LEDs (see \$PASHS,PIN command description).

The initial boot-up sequence for a receiver lights all the three LEDs for about three seconds followed by a brief duration where all three LEDs are off. Thereafter, use the following table to confirm tracking of satellite signals or for basic troubleshooting.

For single antenna configurations, the following LED patterns apply:

Power LED	RTK Corrections LED	SV Tracking LED	Status
On (continuous)	Off	Off	The receiver is turned on, but not tracking satellites.
On (continuous)	Off	Blinking at 1 Hz	The receiver is tracking satellites, but no incoming RTK corrections are being received.
On (continuous)	Blinking at 1 Hz	Blinking at 1 Hz	The receiver is tracking satellites and receiving incoming RTK corrections.
On (continuous)	Off or blinking (receiving corrections)	Blinking at 5 Hz for a short while	Occurs after a power boot sequence when the receiver is tracking less than 5 satellites and searching for more satellites.
On (continuous)	Blinking at 1 Hz	Off	The receiver is receiving incoming RTK corrections, but not tracking satellites.
On (continuous)	Blinking at 5 Hz	Blinking at 1 Hz	The receiver is receiving Moving Base RTK corrections at 5 Hz.
On (continuous)	On (continuous)	Blinking at 1 Hz	The receiver is receiving Moving Base RTK corrections at 10 or 20 Hz (the RTK LED turns off for 100 ms if a correction is lost).
On (continuous)	On, blinking off briefly at 1 Hz	Blinking at 1 Hz	The receiver is in a base station mode, tracking satellites and transmitting RTK corrections.

For two antenna configurations, the following LED patterns apply:

Power LED	RTK Corrections LED	SV Tracking LED	Status
On (continuous)	Off	Off	The receiver is turned on, but not tracking satellites.
On (continuous)	Off	Blinking at 1 Hz then a high-frequency blinking burst every 5 seconds	The receiver is tracking satellites on the position antenna and the vector antenna. However, no incoming RTK corrections are being received.
On (continuous)	Blinking at 1 Hz	Blinking at 1 Hz then a high-frequency blinking burst every 5 seconds	The receiver is tracking satellites on the position antenna and the vector antenna, and incoming RTK corrections are being received.
On (continuous)	Off or blinking (receiving corrections)	Blinking at 5 Hz for a short while	Occurs after a power boot sequence when the position antenna is searching for satellites.
On (continuous)	Off or blinking (receiving corrections)	Off, then a high-frequency blinking burst every 5 seconds	The receiver is tracking satellites on the vector antenna only. The position antenna is not tracking.
On (continuous)	Blinking at 1 Hz	Off	The receiver is receiving incoming RTK corrections, but not tracking satellites on either the position or vector antenna.
On (continuous)	Blinking at 5 Hz	Blinking at 1 Hz then a high-frequency blinking burst every 5 seconds	The position antenna is receiving Moving Base RTK corrections at 5 Hz.
On (continuous)	Continuously on	Blinking at 1 Hz then a high-frequency blinking burst every 5 seconds	The position antenna is receiving Moving Base RTK corrections at 10 or 20 Hz (the RTK LED turns off for 100 ms if a correction is lost).
On (continuous)	On, blinking off briefly at 1 Hz	Blinking at 1 Hz then a high-frequency blinking burst every 5 seconds	The position antenna is in a base station mode, tracking satellites and transmitting RTK corrections.

## POP: Setting Internal Update Rates for Measurements and PVT

**Function** This command allows you to set the update rate used internally in the processing of measurements and position.

**Command Format**    **Syntax**  
**\$PASHS,POP,d1[\*cc]**

## Parameters

Parameter	Description	Range	Default
d1	Internal update rate, in Hz, for measurements.	1, 2, 5, 10, 20, 50 with option [8] 1, 2, 5, 10, 20 with option [W] 1, 2, 5, 10 with option [6] 1, 2, 5 with option [5] 1, 2 with option [2] Else 1 Hz	Firmware option dependent (see below)
*cc	Optional checksum	*00-*FF	

### Example

Setting both update rates to 20 Hz:

```
$PASHS,POP,20*14
```

### Comments

- Changing the POP setting causes all the update rates of the output messages to be reset to their default values. It is therefore recommended to set these update rates only after having run the \$PASHS,POP command.
- The default POP setting depends on which firmware option has been installed:
  - Default is “10 Hz” if option [8], [W] or [6] is enabled
  - Default is “5 Hz” if option [5] is enabled
  - Default is “2 Hz” if option [2] is enabled
  - Default is “1 Hz” otherwise

### Related Commands

\$PASHQ,POP (query command)

\$PASHS,NME

\$PASHS,ATM

## POS: Setting the Antenna Position

---

### Function

This command is used to enter the position of the antenna or antennas connected to the board. The entered positions should be expressed on the same datum as the one corresponding to the primary system used, i.e. WGS-84 if GPS is primary, PZ-90.02 if GLONASS is primary, CGCS2000 if BeiDou is primary).

The command is either used in a rover to speed up the board start-up, in which case the entered position may be very approximate (to within a few kilometers), or on the contrary, it is used to enter the accurate coordinates of a base antenna.

## Command Format Syntax

Board used in SOL mode (a single antenna used):

**\$PASHS,POS,m1,c2,m3,c4,f5[,s6][\*cc]**

Board used in DUO mode (two antennas used)

**\$PASHS,d0,POS,m1,c2,m3,c4,f5[,s6][\*cc]**

## Parameters

Parameter	Description	Range
d0	ID of the antenna the command applies to: <ul style="list-style-type: none"> <li>• 1: Antenna #1 (primary antenna)</li> <li>• 2: Antenna #2 (secondary antenna)</li> <li>• Skipped: Both antenna #1 and antenna #2</li> </ul>	1, 2, skipped
m1	Latitude in degrees and minutes with 7 decimal places (ddmm.mmmmmmm)	0-90
c2	North (N) or South (S)	N, S
m3	Longitude in degrees, minutes with 7 decimal places (dddmm.mmmmmmm)	0-180
c4	West (W) or East (E)	W, E
f5	Height in meters	±0-99999.999
s6	Position attribute (see table below)	PC1, ARP, SPT
*cc	Optional checksum	*00-*FF

Position Attributes:

Parameter	Description
PC1	Position is referenced to L1 phase center (default).
ARP	Position is referenced to ARP.
SPT	Position is referenced to survey point (ground mark).

## Examples

Setting the base antenna position to 37°22.2912135'N, 121°59.7998217'W and 15.25 m:

**\$PASHS,POS,3722.2912135,N,12159.7998217,W,15.25\*1F**

## Comments

- You can enter a reference position tagged to either of the points (PC1, ARP or SPT) on the antenna. If the local antenna name (OWN) is known, the receiver will be able to re-compute internally the reference position entered, from SP to ARP or vice versa.

If the ANT/ANH parameters are known, the receiver will be able to re-compute internally the reference position entered, from SP to ARP or vice versa.

Before entering the coordinates of a reference position using \$PASHS,POS, you should be aware that depending on the protocol you will use to let the base generate its differential data, the reference position needs to be expressed exclusively on one of these points. For example, the reference position in RTCM-3 protocol needs to be tagged to ARP whereas in CMR, this position should be tagged to PC1.

- The position you enter through \$PASHS,POS will be NAKed if it differs from the autonomous one computed internally by more than a certain threshold (dependent on hardware, antenna type and firmware version).

If however, the internally computed position is not available at the time you run the \$PASHS,POS command, then the entered position will be accepted whatever it is.

- By default, the base position is the one you would obtain in response to the following request: \$PASHS,POS,MOV.

**Related Commands**    \$PASHQ,CPD,POS (query command)  
                               \$PASHS,ANH  
                               \$PASHS,ANR  
                               \$PASHS,POS,CUR  
                               \$PASHS,POS,MOV

## POS,CUR: Making the Current Position the Reference Position (Static)

---

**Function**    This command is used to define the reference position. This is done by saving the current position as the new permanent reference position. The “current” position is the one delivered by the receiver at the moment the command is issued.

When two antennas are used, the command assigns the “current” position to both antennas.

This command may be used in a base for quick performance evaluation when there is no opportunity to have an accurate position available.

**Command Format    Syntax**`$PASHS,POS,CUR[*cc]`**Parameters**

None

**Comments**

- The reference position inserted in RTCM, CMR or ATM messages is either a manually entered one, or the one resulting from running \$PASHS,POS,CUR or \$PASHS,POS,MOV, whichever was run last. The one that would result from running \$PASHS,POS,MOV will be inserted by default if there is no entered position or no \$PASHS,POS,CUR has been issued.
- The reference position inserted in ATM,RNX messages (scenario 1, 2, 3, 4, 100 or 101) should be either entered manually or through \$PASHS,POS,CUR, whichever was done last. In absence of this, the reference position will be a moving position (as per \$PASHS,POS,MOV).
- If you are using the ATOM super-compact format, remember this format is designed to be used with a static reference position only, so you should make sure this is the case.
- In ATM,RNX messages using scenario 0, 201, 202, 203, 204 and 300, the reference position is always a moving one (\$PASHS,POS,MOV)

**Related Commands**

\$PASHQ,CPD,POS (query command)

\$PASHS,POS

\$PASHS,POS,MOV

## POS,MOV: Making the Current Position the Reference Position (Dynamic)

---

**Function** This command is used to tell the board to save the last computed position as the reference position. This is done EVERY TIME a new current position is computed. That means the reference position will be an ever changing one if the base is moving.

The “current” position computed by the receiver may be a standalone or DGPS one, but never an RTK one.

By default the receiver uses this “POS,MOV” operating mode to define the reference position.

When two antennas are used, the command assigns a specific position to each of the two antennas. For each antenna, the last computed position results from the data received by this antenna.

### Command Format Syntax

`$PASHS,POS,MOV[*cc]`

### Parameters

None.

### Comments

- The reference position inserted in RTCM, CMR or ATM messages is either a manually entered one, or the one resulting from running `$PASHS,POS,CUR` or `$PASHS,POS,MOV`, whichever was run last. The one that would result from running `$PASHS,POS,MOV` will be inserted by default if there is no entered position or no `$PASHS,POS,CUR` has been issued.
- The reference position inserted in ATM,RNX messages (scenario 1, 2, 3, 4, 100 or 101) should be either entered manually or through `$PASHS,POS,CUR`, whichever was done last. In absence of this, the reference position will be a moving position (as per `$PASHS,POS,MOV`).
- If you are using the ATOM super-compact format, remember this format is designed to be used with a static reference position only, so you should make sure this is the case.
- In ATM,RNX messages using scenario 0, 201, 202, 203, 204 and 300, the reference position is always a moving one (`$PASHS,POS,MOV`)

**Related Commands**    \$PASHQ,CPD,POS (query command)  
                               \$PASHS,POS  
                               \$PASHS,POS,CUR

## PPP: Selecting a PPP Service

---

**Function**            This command is used to select a PPP service (Trimble RTX or TERIASat). TERIASat only covers Metropolitan France.

**Command Format**    **Syntax**  
                               \$PASHS,PPP,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Choice of PPP services: <ul style="list-style-type: none"> <li>• OFF: No PPP service</li> <li>• RTX: Trimble RTX</li> <li>• TRS: TERIASat</li> </ul>	OFF, RTX, TRS	RTX
*cc	Optional checksum	*00-*FF	

### Example

Activating the RTX PPP service:

**\$PASHS,PPP,RTX**

**Related Commands**    \$PASHQ,PPP  
                               \$PASHS,PPP,SRC  
                               \$PASHS,RTX,MOD

## PPP,RST: Resetting the Selected PPP Service

---

**Function**            This command is used to reset the PPP service.

**Command Format**    **Syntax**  
                               \$PASHS,PPP,RST[\*cc]

### Parameters

None.



**Related Commands**    \$PASHS,PPP  
                               \$PASHQ,PPP  
                               \$PASHS,PPP,SRC  
                               \$PASHS,RTX,MOD

## PPP,SRC: Defining Channel Delivering Corrections to the Selected PPP Service

---

**Function**            This command is used to define the transmission channel used to route corrections to the selected PPP service.

**Command Format**    **Syntax**  
                               \$PASHS,PPP,SRC,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Possible routing options are: <ul style="list-style-type: none"> <li>• AUT: Corrections are delivered by either L-band channel or receiver port</li> <li>• LBN: Corrections only come from the L-band channel.</li> <li>• IP: Corrections only come from a port (serial, USB, Bluetooth, IP).</li> <li>• OFF: No corrections are provided to the PPP service.</li> </ul>	AUT, LBN, IP, OFF	AUT
*cc	Optional checksum	*00-*FF	

### Example

Using corrections received via L-band:

**\$PASHS,PPP,SCR,LBN**

**Related Commands**    \$PASHQ,PPP  
                               \$PASHS,PPP

## PPS: Setting PPS Pulse Properties

---

**Function**            This command is used to set the period, offset and GPS synchronized edge (rising or falling) of the PPS pulse.

**Command Format Syntax**`$PASHS,PPS,f1,f2,c3[*cc]`**Parameters**

Parameter	Description	Range	Default
f1	PPS time period, a multiple or fraction of 1 second. • 0: 1 PPS disabled	0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 or 60	0
f2	Time offset in milliseconds.	± 999.9999	0
c3	GPS-synchronized edge code: • "R" for rising edge • "F" for falling edge	R, F	R
*cc	Optional checksum	*00-*FF	

**Example**

Setting the PPS signal to a period of 2 seconds, with an offset of 500 ms and a GPS-synchronized rising edge:

`$PASHS,PPS,2,+500,R*74`**PRT: Setting Baud Rates**

---

**Function** This command is used to set the baud rate of any of the serial ports used in the receiver.

**Command Format Syntax**`$PASHS,PRT,c1,d2[*cc]`**Parameters**

Parameter	Description	Range
c1	Port ID	A, B, D
d2	Baud rate	All ports: 2-12 (Default 9) (see table below)
*cc	Optional checksum	*00-*FF

Code	Baud Rate	Code	Baud Rate
		7	38400
		8	57600
2	1200	9	115200
3	2400	10	230400
4	4800	11	480600
5	9600	12	921600
6	19200		

**Example**

Setting port A to 19200 Bd:

```
$PASHS,PRT,A,6
```

**Related Commands**    \$PASHQ,PRT (query command)  
                           \$PASHS,CTS

## PWR,OFF: Powering Off the Board

---

**Function**    This command is used to prepare the board before it is turned off. Using this command allows all the settings and parameters to be saved in the non-volatile memory. This command DOES NOT switch off the on-board power supply.

**Command Format**    **Syntax**  
                           \$PASHS,PWR,OFF[\*cc]

**Parameters**

None.

**Comments**

Whenever you run a \$PASHS command (set command), you must be aware that the resulting change is not saved to backup memory instantly, but only after a certain delay, which is estimated to be not greater than 120 seconds. There is a requirement behind this operating mode, which is to extend the chip's life cycle as much as possible by reducing the number of write operations in the memory chip.

Because the \$PASHS commands causing the receiver to restart (i.e. INI, RTS, CFG, POP, PWR, etc.) are also part of the "delayed" commands (seen from the backup memory), it is therefore recommended that you run \$PASHS,PWR,OFF about 10 seconds before you initiate a power cycle or reset through one of these commands. After running the command, the on-board LED will turn solid red until the save operation is complete. When the LED goes off, the board can safely be powered off. **Example**

Preparing the board to be turned off:

```
$PASHS,PWR,OFF*43
```

## QZS: Enabling/Disabling QZSS Tracking

---

**Function** This command is used to enable or disable QZSS tracking. The QZSS constellation tracking function is on by default (if [N] option enabled). QZSS signal L1 SAIF is not considered as QZSS and so should be disabled or enabled using \$PASHS,SBA.

### Command Format Syntax

\$PASHS,QZS,s[\*cc]

### Parameters

Parameter	Description	Range	Default
s	Programmable pin ID: • ON: QZSS satellites tracked and used • OFF: QZSS satellites not tracked	ON or OFF	
*cc	Optional checksum	*00-*FF	

### Example

Enabling QZSS tracking:

\$PASHS,QZS,ON

## RCP,OWN: Naming Local Receiver

---

**Function** This commands are used to enter the receiver's own name as well as the reference receiver name.

### Command Format Syntax

\$PASHS,RCP,OWN,s1[,s2[,s3]][\*cc]

or

\$PASHS,RCP,OWN,s1,,s3[\*cc]

## Parameters

Parameter	Description	Range	Default
s1	Receiver name (case-sensitive).	31 characters max.	"ASTECH MB-TWO"
s2	Receiver firmware version	31 characters max.	
s3	Receiver serial number	31 characters max.	
*cc	Optional checksum	*00-*FF	

## Comments

- With the receiver used as an RTK base, the s1 parameter is inserted into receiver name messages (e.g. RTCM-1033).
- When the receiver is used as an RTK rover, it will use the bias values corresponding to the entered local and reference names to correct the local and reference data accordingly.
- Because the RINEX format reserves only 20 characters for receiver names, including the 8 characters used when the \$PASHS,AGB command is set to "ON", we recommend you specify receiver names in 12 characters max. so that they can be converted to RINEX format without being truncated.

## Example

Entering "Unknown" as the name of the local receiver:

```
$PASHS,RCP,OWN,UNKNOWN*2A
```

## RCP,REF: Naming Reference Receiver

---

**Function** This command is used at a rover to specify the name of the base receiver sending differential data. Using this command only makes sense to name the receiver of the base defined as the base data source for the first baseline (i.e. as defined using \$PASHS,1,BLN,...).

**Command Format**    **Syntax**

```
$PASHS,RCP,REF,s1[*cc]
```

## Parameters

Parameter	Description	Range	Default
s1	Receiver name (case-sensitive).	31 characters max.	Empty
*cc	Optional checksum	*00-*FF	

## Comment

- All receiver names are case-sensitive.
- When a rover is assigned the RTK rover function, the entered base receiver name (s1 above) will be ignored if the differential data received from the base include the name of the base receiver.

## Example

Entering “MyBase” as the name of the reference receiver:

```
$PASHS,RCP,REF,MyBase*cc
```

## Related Commands

\$PASHQ,RCP,REF (query command)

\$PASHQ,RCP

\$PASHQ,PAR

\$PASHS,ANP,REF

## RDP,PAR: Setting the Radio

---

### Function

This command is used to set the radio connected to the specified port.

### Command Format

#### Syntax

```
$PASHS,RDP,PAR,c1,s2,d3,[s4],[c5],[d6],[s7],[c8],[c9][d10][s11],[d12],[d13][*cc]
```

## Parameters

Parameter	Description	Range
c1	ID of the port connected to the radio you want to set.	A, B, D

Parameter	Description	Range
s2	Radio Model: <ul style="list-style-type: none"> <li>• PDL (Pacific Crest):               <ul style="list-style-type: none"> <li>– PDL HPB</li> <li>– PDL LPB</li> </ul> </li> <li>• ADL (Pacific Crest):               <ul style="list-style-type: none"> <li>– ADL Vantage</li> <li>– ADL Vantage Pro</li> </ul> </li> <li>• XDL:               <ul style="list-style-type: none"> <li>– Pacific Crest XDL Micro</li> <li>– Pacific Crest XDL Rover</li> </ul> </li> </ul>	PDL, ADL, XDL (ports A, B, D)
d3	Channel number: <ul style="list-style-type: none"> <li>• 0-15 (PDL)</li> <li>• 1-32 (ADL)</li> <li>• 1-32 (XDL)</li> </ul>	0-32
s4	Power management (if port D is used) <ul style="list-style-type: none"> <li>• AUT: Automatic</li> <li>• MAN: Manual</li> </ul>	AUT, MAN
c5	Protocol used: PDL: <ul style="list-style-type: none"> <li>• 0: Transparent</li> <li>• 1: TRIMTALK</li> <li>• 2: DSNP</li> </ul> ADL, XDL: <ul style="list-style-type: none"> <li>• 0: Transparent (w EOT time out)</li> <li>• 1: TRIMTALK 450S</li> <li>• 2: Not used</li> <li>• 3: SATEL</li> <li>• 4: TrimMarkII/IIe</li> <li>• 5: TT450S (HW)</li> <li>• 6: TRIMMARK3</li> <li>• 7: Transparent FST</li> <li>• 8: U-Link, available only with specific radio firmware</li> </ul>	0-8

Parameter	Description	Range
d6	Air link speed (in baud): PDL: <ul style="list-style-type: none"> <li>• 4800 (GMSK modulation)</li> <li>• 9600 (GMSK or 4FSK modulation)</li> <li>• 19200 (4FSK modulation)</li> </ul> ADL or XDL (12.5 kHz): <ul style="list-style-type: none"> <li>• 4800 (GMSK modulation)</li> <li>• 8000 (GMSK modulation)</li> <li>• 9600 (4FSK modulation)</li> </ul> ADL or XDL (25 kHz): <ul style="list-style-type: none"> <li>• 4800 (GMSK modulation)</li> <li>• 9600 (GMSK modulation)</li> <li>• 16000 (GMSK modulation)</li> <li>• 19200 (4FSK modulation)</li> </ul>	4800, 8000, 9600, 16000, 19200
s7	Radio sensitivity	LOW, MED, HIG, OFF
c8	Scrambler: <ul style="list-style-type: none"> <li>• 0: Off</li> <li>• 1: On</li> </ul>	0, 1
c9	Forward Error Correction (FEC): <ul style="list-style-type: none"> <li>• 0: Off</li> <li>• 1: On</li> </ul>	0,1
d10	Radiated power for ADL Vantage: <ul style="list-style-type: none"> <li>• 0: 100 mW</li> <li>• 1: 500 mW</li> <li>• 2: 1 W</li> <li>• 3: 2 W</li> <li>• 4: 4 W</li> </ul> Radiated power for ADL Vantage Pro: <ul style="list-style-type: none"> <li>• 0: Level 1 (2 W)</li> <li>• 1: Level 2</li> <li>• 2: Level 3</li> <li>• 3: Level 4</li> <li>• 4: Level 5</li> </ul> See \$PASHQ,RDP,PWR to set the power (in Watts)  Radiated power for XDL Micro: <ul style="list-style-type: none"> <li>• 0: 500 mW</li> <li>• 2: 2 W</li> </ul>	0-4
s11	Repeater mode (XDL and ADL only) (PCC command 0x58)	OFF, ON
d12	Repeater number (for c5= 1, 4, 5 or 6) <ul style="list-style-type: none"> <li>• 0: Base with one repeater</li> <li>• 1: Base with two repeaters</li> <li>• 2: Repeater 1</li> <li>• 3: Repeater 2</li> </ul> (PCC command 0x6D)	0-3



Parameter	Description	Range
d13	Repeater delay in milliseconds (for c5= 0, 7 or 8). Default delay is 5 ms. (PCC command 0x1A)	
*cc	Optional checksum	*00-*FF

### Comments

- The command will be NAKed if the receiver has not been told on which port the radio is connected. Use command \$PASHS,RDP,TYP to declare the port used.
- **If a PDL radio is used**, depending on its channel spacing, the air link speed you select may force the use of a particular type of modulation and protocol, as well as a particular FEC setting. The different possible combinations are summarized in the table below.

Channel Spacing is:	You set c6 to:	Then modulation can only be:	Protocol can only be:	FEC Setting
12.5 kHz	4800	GMSK	The 3 protocols are possible.	May be set to ON for Transparent protocol (FEC1). Forced to OFF for the other two protocols.
12.5 kHz	8000	Command NAKed		
12.5 kHz	9600	4FSK	Transparent	May be set to ON (FEC1)
25 kHz	4800	GMSK	TRIMTALK or DSNP	Forced to OFF
25 kHz	8000	Command NAKed		
25 kHz	9600	GMSK	TRIMTALK or Transparent	Forced to OFF for TRIM-TALK. May be set to ON for Transparent protocol (FEC1).
25 kHz	16000	Command NAKed		
25 kHz	19200	4FSK	Transparent	May be set to ON (FEC1)

Conversely, the choice of a protocol or modulation may force the use of a particular air link speed.

- **If an ADL radio is used**, depending on its channel spacing, the air link speed you select may force the use of a particular type of modulation and protocol, as well as a

particular FEC setting. The different possible combinations are summarized in the table below.

Channel Spacing is:	You set c6 to:	Then modulation can only be:	Protocol can only be:	FEC Setting
12.5 kHz	4800	GMSK	Transparent, TRIMALK 450S, TT450S (HW) or TrimMark II/IIe.	Maybe set to ON for Transparent protocol (FEC1). Forced to OFF for the other three.
12.5 kHz	8000	GMSK	TRIMALK 450S or TT450S (HW)	Forced to OFF
12.5 kHz	9600	GMSK	TrimMark3 or U-Link	Forced to OFF
		4FSK	Transparent, SATEL or Transparent FST	May be set to ON (FEC 1 for Transparent protocol, FEC2 for the other two)
25 kHz	4800	GMSK	TRIMTALK 450S, TT450S (HW) or TrimMark II/IIe or U-Link	Forced to OFF
25 kHz	8000	Command NAKed		
25 kHz	9600	GMSK	Transparent, TrimTalk 450S, TT450S (HW) or U-Link	May be set to ON for Transparent protocol (FEC1). Forced to OFF for the other three.
25 kHz	16000	GMSK	TrimTalk 450S or TT450S (HW)	Forced to OFF
25 kHz	19200	GMSK	TrimMark3	Forced to OFF
		4FSK	Transparent, SATEL or Transparent FST	May be set to ON (FEC 1 for Transparent protocol, FEC2 for the other two)

Conversely, the choice of a protocol or modulation may force the use of a particular air link speed.

### Example

Setting the internal XDL radio:

```
$PASHS,RDP,PAR,D,XDL,2,AUT,0,9600,LOW,0,0,OFF*1E
```

### Related Commands

```
$PASHQ,RDP,PAR (query command)
```

```
$PASHS,RDP,TYP
```

```
$PASHQ,RDP,CHT
```

## RDP,TYP: Defining the Type of Radio Used

---

**Function** This command is used to set manually the type of radio connected to the specified port. Normally, the type of internal radio (typically connected to port D) is detected automatically.

### Command Format Syntax

```
$PASHS,RDP,TYP,c1,s2[*cc]
```

### Parameters

Parameter	Description	Range
c1	ID of the port connected to the radio you want to set.	A, B, D
s2	Radio Model: <ul style="list-style-type: none"> <li>• NONE: No radio</li> <li>• AUTO: Auto-detection</li> <li>• PDL: Pacific Crest radio <ul style="list-style-type: none"> <li>– PDL HPB</li> <li>– PDL LPB</li> </ul> </li> <li>• ADL: Pacific Crest radio <ul style="list-style-type: none"> <li>– ADL Vantage</li> <li>– ADL Vantage Pro</li> </ul> </li> <li>• XDL: Pacific Crest XDL Rover</li> </ul>	NONE, AUTO, PDL, ADL, XDL (ports A, B, D)
*cc	Optional checksum	*00-*FF

### Example

Setting the external radio as ADL Vantage:

```
$PASHS,RDP,TYP,A,ADL
```

**Related Commands** \$PASHQ,RDP,TYP (query command)  
 \$PASHS,RDP,PAR

## REC: Enable/Disable, Start/Stop Raw Data Recording

**Function** This command allows you to enable, disable, start or stop raw data recording. Raw data is recorded in the memory you selected with the \$PASHS,MEM command.

### Command Format Syntax

```
$PASHS,REC,c[*cc]
```

### Parameters

Parameter	Description	Range
c	Control character: <ul style="list-style-type: none"> <li>• Y: Yes. The receiver will immediately start recording data. This option also enables data recording at receiver power-up, i.e. recording will start every time you turn the receiver on, even if you stopped recording before the end of the previous session.</li> <li>• N: No. The receiver will immediately stop recording data. This option also disables data recording at receiver power up, i.e. the receiver won't resume data recording every time you turn it on for a new operating session. This is the default mode.</li> <li>• S: Stop. The receiver will immediately stop recording raw data. This option does not affect the way the receiver operates at power-up.</li> <li>• R: Restart. The receiver will immediately start recording raw data. This option does not affect the way the receiver operates at power-up.</li> </ul>	Y, N, S, R
*cc	Optional checksum	*00-*FF

### Examples

Starting raw data recording:

```
$PASHS,REC,Y*54
```

Stopping raw data recording:

```
$PASHS,REC,N*43
```

**Comment** If you want to log just NMEA messages and not the G-file and ATOM messages, you need to disable ATOM messages after enabling data recording.

See example below enabling only GGA and GSV messages to be saved to the current memory:

- Select USB to be the current memory:

**\$PASHS,MEM,2**

Enable NMEA messages to USB:

**\$PASHS,NME,GGA,U,ON**

**\$PASHS,NME,GSV,U,ON**

Enable data recording. (This results in also enabling ATOM messages for G-file):

**\$PASHS,REC,Y**

Disable ATOM messages:

**\$PASHS,ATM,ALL,U,OFF**

**Related Commands**    \$PASHQ,REC (query command)  
                           \$PASHS,MEM  
                           \$PASHS,ATM  
                           \$PASHS,NME

## **RFM: Enabling/Disabling Ring File Memory**

---

**Function**    This command is used to enable or disable the use of the ring file memory.

Enabling the ring file memory allows you to manage the free memory space in the receiver, making sure you can log new raw data files for an unlimited period of time without running out of memory.

**Command Format**    **Syntax**  
                           **\$PASHS,RFM,s1[,d2][\*cc]**

## Parameters

Parameter	Description	Range	Default
s1	Enabling or disabling the ring file memory: <ul style="list-style-type: none"> <li>• Y: Enables the use of the ring file memory. The oldest files will be deleted automatically when the amount of free memory space left drops below d2 (see below).</li> <li>• N: Disables the use of the ring file memory. The logging of raw data files will stop when there is no free space left in the memory used.</li> </ul>	Y, N	N
d2	Threshold of free memory space left (in MBytes) for which the use of the ring file memory will come into play.	1-1024	15
*cc	Optional checksum	*00-*FF	

### Example

Enabling ring file memory:

```
$PASHS,RFM,Y*59
```

**Relevant Query Command**    \$PASHQ,RFM

**See Also**    \$PASHS,REC

## RST: Default Settings

---

**Function**    This command is used to reset the receiver parameters to their default values. All parameters are reset except:

- Ephemeris data (except SBAS ephemeris)
- Almanac data
- Position
- Time

If the *default\_config.cmd* file is found on the board (see \$PASHS,DFC,SET), then all the commands listed in this file are run by the board just after it has been reset.

**Command Format Syntax**

```
$PASHS,RST[*cc]
```

**Parameters**

None.

**Example**

Resetting the receiver:

`$PASHS,RST*20`

**CAUTION-** Sending this command will cause all receiver parameters, including communication port settings, to be reset to their factory settings. If your application is using settings for communication ports that are different from factory defaults, then it may no longer be able to communicate with the board.

**RT2: Enabling/Disabling RTCM 2.3 Messages**

**Function** This command is used to individually enable or disable RTCM 2.3 message types on the specified port.

**Command Format** **Syntax**  
`$PASHS,RT2,s1,c2,s3[,f4][*cc]`

**Parameters**

Parameter	Description	Range	Default
s1	RTCM 2.3 message type	See table below	-
c2	Port ID	A, B, C, D, F, I, P, Q	-
s3	Enables (ON) or disables (OFF) RTCM 2.3 message type	ON, OFF	OFF
f4	Output rate in seconds	See table below	1
*cc	Optional checksum	*00-*FF	-

Supported Data Messages:

Data	Description	f4 Range
1, 9	GPS corrections	0.05, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5 sec., etc.
3	Geographical coordinates (ITRF) tagged to L1 phase center for reference position	(integer seconds up to 999 sec.)

Data	Description	f4 Range
31, 34	GLO corrections	0.05, 0.02, 0.05, 0.1, 0.2,
32	Reference GLONASS PZ 90 position	0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 sec

### Example

Setting the default RTCM 2.3 configuration in a base:

```
$PASHS,POS,<position coordinates> or $PASHS,POS,CUR*51
$PASHS,RT2,1,A,ON,1*52
$PASHS,RT2,3,A,ON,1*03
$PASHS,RT2,9,A,ON,1*09
$PASHS,RT2,31,A,ON,1*6B
```

### Comments

- **About DGNSS corrections:** Our receivers generate the same content in messages 3 and 32, which is coordinates entered by the \$PASHS,POS command. To transform the original geodetic position (\$PASHS,POS) into a Cartesian position (types 3 and 32), the WGS-84, PZ-90.02 or CGCS2000 ellipsoid is used depending on which primary GNSS system is used (PGS).

Our receivers generate DGNSS corrections (type 1 for GPS and type 31 for GLO) against reference position “type 3” (=32), using all GNSS satellites coordinates expressed either in WGS-84, PZ-90.02 or CGCS2000, depending on the primary GNSS system used.

Rovers ignore the content of message type 32. They need the “type 3” reference position to apply “type 1” and/or “type 31” corrections in DGNSS positions.

## RT2,ALL: Disabling All RTCM 2.3 Messages

---

### Function

This command is used to disable all the currently active RTCM 2.3 data messages on the specified port.

### Command Format

#### Syntax

```
$PASHS,RT2,ALL,c1,OFF[*cc]
```



## Parameters

Parameter	Description	Range
c1	Port ID • A, B, D: Serial port • C: USB serial port • I, P, Q, F: TCP/IP stream	A, B, C, D, F, I, P, Q
*cc	Optional checksum	*00-*FF

## Example

Disabling all RTCM 2.3 messages on port A:

```
$PASHS,RT2,ALL,A,OFF*22
```

## RT3: Enabling/Disabling RTCM 3.1 Messages

---

**Function** This command is used to individually enable or disable RTCM 3.1 message types on the specified port.

### Command Format Syntax

```
$PASHS,RT3,s1,c2,s3[,f4][*cc]
```

## Parameters

Parameter	Description	Range	Default
s1	RTCM 3.1 message type	See table below	-
c2	Port ID • A, B, D: Serial port • C: USB serial port • I, P, Q, F: TCP/IP stream	A, B, C, D, F, I, P, Q	-
s3	Enables (ON) or disables (OFF) RTCM 3.1 message type	ON, OFF	OFF
f4	Output rate in seconds	See table below	1
*cc	Optional checksum	*00-*FF	-

Supported Data Messages:

Data	Description	f4 range
1001-1004	GPS+SBAS raw observations	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 sec
1005-1006	ITRF coordinates of reference position, tagged to ARP	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5 sec., etc. (integer seconds up to 999 sec.)
1007-1008	Antenna name	

Data	Description	f4 range
1009-1012	GLONASS raw observations	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 sec
1013	System information	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5 sec., etc. (integer seconds up to 999 sec.)
1019	GPS ephemeris data	1, 2, 3, 4, 5 sec., etc. (integer seconds up to 999 sec.)
1020	GLONASS ephemeris data	
1029	User unicode message (as entered through \$PASHS,MSG)	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5 sec., etc. (integer seconds up to 999 sec.)
1033	Antenna and receiver names	
1042	BeiDou ephemeris data	1, 2, 3, 4, 5 sec, etc. (integer seconds up to 999 sec)
1044	QZSS Ephemeris data	1, 2, 3, 4, 5 sec, etc. (integer seconds up to 999 sec)
1045	GALILEO ephemeris data (F:NAV)	
1046	GALILEO ephemeris (I/NAV)	1, 2, 3, 4, 5 sec, etc. (integer seconds up to 999 sec)
1071-1077	GPS MSM messages	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 sec
1081-1087	GLONASS MSM messages	
1091-1097	GALILEO MSM messages	
1101-1107	SBAS MSM messages	
1111-1117	QZSS MSM messages	
1121-1127	BEIDOU MSM messages	
1230	GLONASS code bias message	

## Examples

Setting the default RTCM 3.1 configuration to serve the base mode:

```
$PASHS,POS,<position coordinates> or $PASHS,POS,CUR*51
$PASHS,RT3,1004,A,ON,1*34
$PASHS,RT3,1012,A,ON,1*33
$PASHS,RT3,1006,A,ON,13*05
$PASHS,RT3,1033,A,ON,31*03
```

Setting the default RTCM 3.1 configuration to serve the raw data collection mode:

```
$PASHS,POS,MOV*41
$PASHS,RT3,1004,A,ON,1*3
$PASHS,RT3,1012,A,ON,1*33
$PASHS,RT3,1006,A,ON,1*36
$PASHS,RT3,1013,A,ON,61*04
```

```
$PASHS,RT3,1033,A,ON,300*32
```

```
$PASHS,RT3,1019,A,ON,600*3F
```

```
$PASHS,RT3,1020,A,ON,600*35
```

## RT3,ALL: Disabling All RTCM 3.2 Messages

---

**Function** This command is used to disable all the currently active RTCM 3.2 data messages on the specified port.

### Command Format Syntax

```
$PASHS,RT3,ALL,c1,OFF[*cc]
```

### Parameters

Parameter	Description	Range
c1	Port ID <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A, B, C, D, F, I, P, Q
*cc	Optional checksum	*00-*FF

### Example

Disabling all RTCM 3. messages on port A:

```
$PASHS,RT3,ALL,A,OFF*23
```

## RTK: Running RTK Processing on Set Baselines

---

**Function** This command is used to run the RTK process over all the baselines (up to three) you have defined using command \$PASHS,BLN.

The command can also be used to stop this process.

When a baseline engine is included in an RTK process, it is assumed that the type of base it is working from is a static one. Each of the baselines included in an RTK process is a source of position solution, and not a source of angle estimates.

See also command \$PASHS,TOP.

## Command Format Syntax

General:

```
$PASHS,RTK,ON,d1[,d2[,d3]][*cc]
```

To stop the RTK process:

```
$PASHS,RTK,OFF[*cc]
```

### Parameters

Parameter	Description	Range
d1	First baseline number	1, 2, 3
d2	First baseline number	1, 2, 3
d3	First baseline number	1, 2, 3
*cc	Optional checksum	*00-*FF

### Comments

The following combinations are possible today:

- Conventional RTK:  
`$PASHS,RTK,ON,1`
- Combining positions or selecting the best position among engines #1 and #2:  
`$PASHS,RTK,ON,1,2`  
or  
`$PASHS,RTK,ON,2,1`
- Combining positions or selecting the best position among engines #1 and #3:  
`$PASHS,RTK,ON,1,3`  
or  
`$PASHS,RTK,ON,3,1`
- Combining positions or selecting the best position among engines #1, #2 and #3:  
`$PASHS,RTK,ON,1,2,3`  
(or any combination "1,3,2", "2,1,3", "2,3,1", etc.)

## RTK,RST: Resetting RTK or RTX Computation

---

### Function

This command is used to reset RTK or RTX computation by resetting every baseline engine delivering an RTK solution. This command is also applicable to all baseline engines serving an application of the RTK type.

**Command Format**    **Syntax**  
                           \$PASHS,RTK,RST[\*cc]

**Parameters**

None.

**Example**

Resetting RTK computation:

\$PASHS,RTK,RST

**Comments**

- This command is not applicable to engines delivering 3DF-type solutions (in this case, use \$PASHS,3DF,RST to reset those engines).
- Resetting RTK/RTX computation means resetting the current estimates of single-difference (SD) carrier ambiguities for all processed signals.
- Just after issuing this command, you should expect jumps on position, increased RMS and temporary float solution status.

## RTK,STI: Defining Which Base Data Stream to Use

---

**Function**

This command is used to specify the ID of the base station your RTK rover should work with. Your choice may be to let the board select the data stream by itself (Automatic base selection) or ask for a particular base to be used.

The choice made through this command is only applicable to the processing modes a network rover can operate in (see command \$PASHS,DIF,NET). It is NOT applicable if a Trimble RTX service, or an SBL CMRx data stream is used.

**Command Format**    **Syntax**  
                           \$PASHS,RTK,STI,d1[\*cc]

## Parameters

Parameter	Description	Range	Default
d1	ID of preferred base station: <ul style="list-style-type: none"> <li>AUT: Base selected automatically</li> <li>Base ID (0-4095): ID of preferred base station</li> </ul>	AUT, 0-4095	AUT
*cc	Optional checksum	*00-*FF	

## Example

Choosing base station ID 3001:

```
$PASHS,RTK,STI,3001*74
```

## Comments

- The command equally impacts all GNSS sensors.
- If no corrections are received from the chosen base station, then the rover will output position in the mode the closest to RTK, i.e. DGNSS (see \$PASHS,TOP), still using the last epoch of available corrections. In this case, the reported base ID and differential age will accurately reflect the currently working operating mode.

**See also** \$PASHQ,RTK (relevant query command)

\$PASHS,STI

\$PASHS,DIF,NET

## RTX,DTM: RTX Datum Transformation

---

**Function** This command is used to enable/disable a datum transformation so the computed RTX position can be delivered in the required datum.

**Command Format** **Syntax**

```
$PASHS,RTX,DTM,s1[,s2,s3][*cc]
```

## Parameters

Parameter	Description	Range
s1	Enables/disables datum transformation: <ul style="list-style-type: none"> <li>• ON: Enables transformation</li> <li>• OFF: Disables transformation. RTX position is expressed in ITRF 2008 current epoch.</li> </ul>	ON, OFF
s2	If s1= ON, name of the datum you want the computed RTX position to be transformed to.	AUTO, or choose datum name from list below.
s3	If s1=ON, name of the tectonic plate corresponding to where the position is computed.	AUTO, or choose plate name from list below.
*cc	Optional checksum	*00-*FF

Datum Name	Description
AUTO	Automatic detection of datum according to current position
ITRF1988	ITRF1988 (epoch 1988)
ITRF1989	ITRF1989 (epoch 1988)
ITRF1990	ITRF1990 (epoch 1988)
ITRF1991	ITRF1991 (epoch 1988)
ITRF1992	ITRF1992 (epoch 1988)
ITRF1993	ITRF1993 (epoch 1993)
ITRF1994	ITRF1994 (epoch 1993)
ITRF1996	ITRF1996 (epoch 1997)
ITRF1997	ITRF1997 (epoch 1997)
ITRF2000	ITRF2000 (epoch 1997)
ITRF2005	ITRF2005 (epoch 2000)
ITRF2008	ITRF2008 (epoch 2005)
NAD83	NAD83 (epoch 1997)
NAD83-CRS	NAD83-CRS (epoch 2002)
NAD83-CORS96	NAD83-CORS96 (epoch 2002)
ETRS89	ETRS89 (epoch 1989)
ETRF2000-R05	ETRF2000-R05 (epoch 2000)
GDA94	GDA94 (epoch 1994)
SIRGAS2000	SIRGAS2000 (epoch 2000,4)
SIRGAS95	SIRGAS95 (epoch 1995,4)
SIRGAS-CON	SIRGAS-CON (epoch 2005)
NAD83-2011	NAD83-2011 (epoch 2010)
NAD83-MA11	NAD83-MA11 (epoch 2010)
NAD83-PA11	NAD83-PA11 (epoch 2010)
RGF93v2	RGF93v2 (epoch 2009)
CGCS2000	CGCS2000 (epoch 2000)

<b>Tectonic Plate</b>	<b>Description</b>
AUTO	Automatic detection of plate according to current position
AEGS	Aegean Sea
ALTP	Altiplano
AMUR	Amurian
ANTL	Anatolia
ANTA	Antarctica
ARAB	Arabia
AUST	Australia
BLRF	Balmoral Reef
BANS	Banda Sea
BIRH	Birds Head
BURM	Burma
CAPR	Capricorn
CARB	Caribbean
CARL	Caroline
COCO	Cocos
CONR	Conway Reef
EASR	Easter
EURA	Eurasia
FUTU	Futuna
GALP	Galapagos
INDI	India
JUFU	Juan de Fuca
JUFE	Juan Fernandez
KERM	Kermadec
LWAN	Lwandle
MAQR	Macquarie
MANU	Manus
MAOK	Maoke
MARI	Mariana
MOLU	Molucca Sea
NAZC	Nazca
NHBR	New Hebrides
NIUA	Niuafou
NOAM	North America
NAND	North Andes
NOBI	North Bismarck
NUBI	Nubia
OKHO	Okhotsk
OKIN	Okinawa
PCFC	Pacific
PANM	Panama
PHIL	Philippine Sea
RIVR	Rivera



Tectonic Plate	Description
SNDW	Sandwich
SCOT	Scotia
SHTL	Shetland
SLMS	Solomon Sea
SOMA	Somalia
SOAM	South America
SOBI	South Bismarck
SUND	Sunda
SUR0	Sur
TIMO	Timor
TONG	Tonga
WDLR	Woodlark
YANG	Yangtze

### Example

Output RTX positions at fixed epoch: i.e. in ITRF2008 (epoch 2005), using auto-detection of tectonic plate:

```
$PASHS,RTX,DTM,ON,ITRF2008,AUTO
```

**Related Commands**

```
$PASHQ,RTX
$PASHS,RTX,MOD
$PASHS,RTX,SRC
```

## RTX,KPI: Entering Known RTX Initialization Point

---

**Function** This command is used to force the RTX engine to initialize on a known point.

Entering a known point for RTX initialization causes the RTX engine to reset and reinitialize assuming the receiver is currently occupying this point.

**Command Format** **Syntax**

```
$PASHS,RTX,KPI,m1,c2,m3,c4,f5[,f6,f7,f8][,c9][,c10,c11[,f12]][*cc]
```

## Parameters

Parameter	Description	Range
m1	Latitude in degrees and minutes with 7 decimal places (ddmm.mmmmmmm)	0-90
c2	North (N) or South (S)	N, S
m3	Longitude in degrees, minutes with 7 decimal places (dddmm.mmmmmmm)	0-180
c4	West (W) or East (E)	W, E
f5	Height in meters	±0-99999.999
f6	Accuracy (RMS) in Lat direction (default: 0.01)	0-99.999 m
f7	Accuracy (RMS) in Lon direction (default: 0.01)	0-99.999 m
f8	Accuracy (RMS) in Alt direction (default: 0.01)	0-99.999 m
c9	Position attribute (see table below)	PC1, ARP, SPT
c10	Datum name (default is AUTO) or "OFF"	See RTX,DTM
c11	Plate name (default is AUTO)	See RTX,DTM
f12	Epoch (year) of position	YYYY.YY
*cc	Optional checksum	*00.*FF

### Position Attributes:

Parameter	Description
PC1	Position is referenced to L1 phase center (default).
ARP	Position is referenced to ARP.
SPT	Position is referenced to survey point (ground mark).

## Examples

For a quick start on a point with coordinates  
 37°22.2912135'N, 121°59.7998217'W and 15.25 m:

**\$PASHS,RTX,KPI,3722.2912135,N,12159.7998217,W,15.25**

## Comments

- Default values will be used whenever parameter fields are found missing or intentionally left empty.
- The command will be NAKed if no information is available to convert the entered position into an L1 phase center position or to transform the entered position to the right datum and plate (ITRF 2008 current epoch year).
- The RTX,KPI command is not applicable to any PVT engines other than RTX engines.
- When "AUTO" is selected for names (c10, c11), the required datum should be selected from the current receiver position (like \$PASHS,RTX,DTM).
- c10=OFF refers to ITRF 2008 current epoch. No KPI coordinate transformation is required (like \$PASHS,RTX,

DTM,OFF). In this case c11 is ignored, while optional field f12 can report the KPI position epoch.

**Query Command** \$PASHQ,RTX,STS

## RTX,MOD: Specifying the RTX Corrections Service Used

---

- Function** This command is used to specify which RTX corrections service the receiver will use, depending on the desired level of accuracy:
- CenterPoint RTX (centimeter-level, steady-state accuracy)
  - FieldPoint RTX (decimeter-level, steady-state accuracy)
  - RangePoint RTX (submeter-level, steady-state accuracy)
  - ViewPoint RTX (meter-level, steady-state accuracy)

It can also be run to disable the use of Trimble RTX.

### Command Format Syntax

\$PASHS,RTX,MOD,s1[\*cc]

#### Parameters

Parameter	Description	Range	Default
s1	Trimble RTX service to be used: <ul style="list-style-type: none"> <li>• OFF: RTX service disabled (default)</li> <li>• ON: Best RTX service available, in line with the currently installed options.</li> <li>• CPT: CenterPoint RTX</li> <li>• RPT: RangePoint RTX</li> <li>• VPT: ViewPoint RTX</li> <li>• FPT: FieldPoint RTX</li> </ul>	OFF, ON, RPT, CPT, RPT, VPT, FPT	Depends on installed options (see table below)
*cc	Optional checksum	*00-*FF	

Options installed	Default Trimble RTX Service
[C], [P], [4] and [9] disabled	OFF
[C] only	CPT
[P] only	RPT
[4] only	VPT
[9] only	FPT

When several options are enabled, the mode providing the best accuracy is the default one. The order is the following: CPT(best accuracy), FPT, RPT, VPT.

### Example

Requesting the use of Trimble RTX service “CenterPoint RTX”:

```
$PASHS,RTX,MOD,CPT*2A
```

### Comments

- All supported Trimble RTX services except for ViewPoint RTX, requires that the L2 tracking option be installed. Applying the command to switch between VPT and any of the other three modes will cause the RTX engine to be restarted.
- The command will be NAKed if there is an attempt to activate a Trimble RTX service with higher accuracy than what is possible considering the currently installed options.
- After you’ve run \$PASHS,RTX,MOD,OFF, the RTX engine will still be running, but only to decode the subscription details, which will be forwarded to the Service Layer (if detected/decoded) via command \$PASHS,BSM. As soon as \$PASHS,RTX,MOD,OFF is applied, the RTX position is right away made no longer available.
- The ability to output RTX correcting streams using ATM,DAT messages is not affected by this command.

**See also** \$PASHQ,RTX (relevant query command)  
\$PASHS,RTX,SRC

## RTX,RST: Resetting RTX Position Computation

---

**Function** This command is used to reset the process through which an RTX (PPP) position solution is computed.

**Command Format** **Syntax**  
\$PASHS,RTX,RST[\*cc]

**Parameters**  
None.

- Comments**
- Resetting RTX means fully stopping and re-initializing from scratch all RTX computations in progress. The command does NOT however reset GNSS tracking and reception of correction data.
  - This command is not applicable to engines delivering RTK position solutions. \$PASHS,RTK,RST should be used to reset the RTK computation process.

## **RTX,SRC: Specifying RTX Corrections Source**

---

**Function** This command is used to specify the path through which corrections are applied to the Trimble RTX/RTK engine. This engine is dedicated to computing a position solution by processing all available GNSS observables corrected by Trimble RTX State Space Correcting Data or Trimble Base/Network CMRx/sCMRx data.

RTX/RTK correcting data may be fed to the board either through an L-band (sky) data link or through GPRS/NTRIP service via one of the board's communication ports.

**Command Format**    **Syntax**  
                           \$PASHS,RTX,SRC,s1[\*cc]

## Parameters

Parameter	Description	Range
s1	<p>Choosing how to apply corrections to the RTX/RTK engine:</p> <ul style="list-style-type: none"> <li>AUT (default setting): The RTX/RTK engine will process corrections data received via either L-band channel or via one of the board's communication ports. When data are available on both channels, the board selects the best source of RTX corrections, based on internal, proprietary algorithms. In most cases, CMRx/sCMRx corrections entering the board via a port will predominate over the L-band data stream.</li> <li>LBN: Corrections data provided to the RTX engine are expected to be received via the L-band channel. The command will be NAKed if the [L] firmware option is missing.</li> <li>IP: CMRx or sCMRx corrections data provided to the RTX/RTK engine are expected to be received via one of the board ports.</li> <li>OFF: No corrections are provided to the RTX/RTK engine, so no position solution is computed. However navigation data (e.g. ephemeris data) continue to be delivered to the RTX/RTK engine so it can quickly restart when requested to do so. On making this choice, RTX/RTK positions will continue to be reported for about 300 seconds.</li> </ul>	AUT, LBN, IP, OFF
*cc	Optional checksum	*00-*FF

## Example

Enabling RTX engine using the possible two RTX corrections reception channels:

**\$PASHS,RTX,SRC,AUT**

## Comments

- This command does not affect the possibility to output each of the corrections streams through ATOM message ATM,DAT.
- If AUT is selected, and if more than one CMRx/sCMRx stream enter the board via different ports, the stream detected first is the one used (and all the others are ignored).

## RXC,PAR: Setting the Embedded RINEX Converter

---

**Function** This command is used to set all the parameters of the RINEX converter. While parameters d1 to s10 in the command define the type of conversion performed by \$PASHS,RXC,RUN, parameters s11 to s16 define the different parameters found in the RINEX header of a converted file, following the conversion of this file by \$PASHS,RXC,RUN.

### Command Format Syntax

```
PASHS,RXC,PAR[,VER,d1][,CMP,d2][,GPS,s3][,GLO,s4][,SBA,s5]
[,GAL,s6][,BDS,s7][,QZS,s8][,IRN,s9][,AGY,s10][,OBN,s11][,MNM,s12]
[,MNB,s13][,OBS,s14][,NAV,s15][,MET,s16][,ANT,d17][*cc]
```

## Parameters

Parameter	Description	Default	Range
VER,d1	RINEX version: <ul style="list-style-type: none"> <li>• 1: RINEX 2.10</li> <li>• 2: RINEX 2.11</li> <li>• 3: RINEX 2.12</li> <li>• 4: RINEX 3.00</li> <li>• 5: RINEX 3.01</li> <li>• 6: RINEX 3.02</li> <li>• 7: RINEX 3.03</li> </ul>	2	1-7
CMP,d2	File compression: <ul style="list-style-type: none"> <li>• 0: None</li> <li>• 1: Hatanaka</li> <li>• 2: tarZ</li> <li>• 3: Hatanakla and tarZ</li> </ul>	3	0-3
GPS,s3	GPS measurement conversion	ON	ON, OFF
GLO,s4	GLONASS measurement conversion	ON	ON, OFF
SBA,s5	SBAS measurement conversion	ON	ON, OFF
GAL,s6	GALILEO measurement conversion	ON	ON, OFF
BDS,s7	BEIDOU measurement conversion	ON	ON, OFF
QZS,s8	QZSS measurement conversion	ON	ON, OFF
IRN,s9	IRNSS measurement conversion	ON	ON, OFF
AGY,s10	Agency name		20 char. max.
OBN,s11	Observer name		20 char. max.
MNM,s12	Antenna marker name		60 char. max.
MNB,s13	Antenna marker number		20 char. max.
OBS,s14	Observation file comment line		60 char. max.
NAV,s15	Navigation file comment line		60 char. max.
MET,s16	Meteorological file comment line		60 char. max.
ANT,d17	Antenna: <ul style="list-style-type: none"> <li>• 0: Converts raw data for both antennas</li> <li>• 1: Converts raw data for antenna 1 only</li> <li>• 2: Converts raw data for antenna 2 only</li> </ul>	1	0-2
*cc	Optional checksum	*00-*FF	

### Example

Setting the RINEX converter to produce RINEX 2.11, TarZ-compressed files:

```
$PASHS,RXC,PAR,VER,2,CMP,2,AGY,Intec,OBN,Peter Smith,
MNM,CARQ,MNB,1005M001*4E
```

### Related Commands

```
$PASHS,RXC,RUN
```



## RXC,RUN: Converting a G-File into RINEX Files

---

**Function** This command is used to convert a G-file into RINEX files.

### Command Format Syntax

```
$PASHS,RXC,RUN,[d1],[s2],s3[*cc]
```

### Parameters

Parameter	Description	Range	Default
d1	Memory location: • 0: Internal memory • 2: USB device  If d1 is omitted, the receiver looks for the specified file on the memory last selected with \$PASHS,MEM.	0 or 2	0
s2	Path on the selected memory where to find the G-file.	255 characters max.	-
s3	G-file name. No path allowed in this field.	13 characters in the form "GxxxxSyy.ddd"	-
*cc	Optional checksum	*00-*FF	

### Comments

- The headers of the RINEX files are built using the information provided through \$PASHS,RXC,PAR.
- The resulting RINEX files are stored in the same folder as the one containing the G-file specified in the command.
- \$PASHR,NAK\*30 is returned if the specified file does not exist, is not a G-file, or a RINEX conversion is currently in progress.
- \$PASHR,ACK\*3D is returned when the command is accepted, then \$PASHR,RXC,OK\*15 or \$PASHR,RXC,FAILED\*12, depending on whether the conversion respectively succeeded or failed.

### Examples

Converting a G-file to Rinex (in the same folder):

```
$PASHS,RXC,RUN,,,GabcdA09.241*67
$PASHR,ACK*3D
$PASHR,RXC,OK*15
```

Converting a G-file to Rinex and saving the resulting file in a sub-folder:

```
$PASHS,RXC,RUN,2,2009/241,GabcdA09.241*46
```

\$PASHR,ACK\*3D  
\$PASHR,RXC,OK\*15

**Relevant Query Command** None.

**See Also** \$PASHS,RNX,PAR  
\$PASHS,MEM

## SBA: Enabling/Disabling SBAS Tracking

---

**Function** This command is used to enable or disable SBAS tracking. QZSS signal L1 SAIF is considered as SBA and so can be enabled or disabled using this command.

**Command Format Syntax**

\$PASHS,SBA,s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Enables (ON) or disables (OFF) SBAS tracking	ON, OFF	ON (if option [N] enabled)
*cc	Optional checksum	*00-*FF	

### Example

Enabling SBAS tracking:

\$PASHS,SBA,ON

### Comments

- The command will be NAKed if the [N] option has not been activated beforehand.
- Disabling SBAS does not mean SBAS corrections stop to be applied to the PVT engine. The last SBAS corrections received will indeed continue to be applied for some time. So disabling SBAS tracking does not mean SBAS differential positioning is stopped immediately.

**Related Commands** \$PASHQ,SBA (query command)  
\$PASHS,GLO  
\$PASHS,GPS  
\$PASHS,QZS  
\$PASHS,GAL

\$PASHS,BDS  
\$PASHS,DIF,SBA

## SIT: Defining a Site Name

---

**Function** This command is used to define a site name that will be used in the naming of the next logged raw data file. The default site name is the last four digits of the receiver serial number.

**Command Format** **Syntax**

\$PASHS,SIT,s[\*cc]

### Parameters

Parameter	Description	Range
s	Site name (or site ID), a 4-character string where "*", ".", "/" and "\" are not allowed.	
*cc	Optional checksum	*00-*FF

### Example

Defining site name "ECC1":

\$PASHS,SIT,ECC1\*63

**Related Commands** \$PASHQ,SIT  
\$PASHS,REC

## SNS: Configuring the M-Sensor

---

**Function** This command is used to set the number of independent sensors on the board (M-Sensor= Multiple Sensors).

**Command Format** **Syntax**

\$PASHS,SNS,s1[,d2][\*cc]

## Parameters

Parameter	Description	Range
s1	Defines the number of “independent” sensors running on the board (default: SOL). <ul style="list-style-type: none"> <li>• SOL: One sensor (= one antenna)</li> <li>• DUO: Two sensors (= two antennas)</li> </ul>	SOL, DUO
d2	Index specifying antenna/sensor configuration (default: 0). See table below.	0-2
*cc	Optional checksum	*00-*FF

d2	s1=SOL	s1=DUO
0 or blank	<ul style="list-style-type: none"> <li>• Active antenna input is antenna input #1 or #2, whichever provides the best signal (default). In either case, L1/L2 GNSS is supported.</li> <li>• L-band sensor can only be enabled for antenna input #1 using command \$PASHS,GLB. L-band sensor won't start if antenna input #2 is used.</li> </ul>	<ul style="list-style-type: none"> <li>• Antenna input #1 is L1/L2 GNSS capable.</li> <li>• Antenna input #2 is L1 GNSS only.</li> <li>• L-band sensor can only be enabled for antenna input #1 using command \$PASHS,GLB.</li> </ul>
1	<ul style="list-style-type: none"> <li>• Active antenna input is antenna input # 1 only. L1/L2 GNSS supported by this input.</li> <li>• L-band sensor can only be enabled for antenna input #1 using command \$PASHS,GLB.</li> </ul>	
2	<ul style="list-style-type: none"> <li>• Active antenna input is antenna input # 2 only. L1/L2 GNSS supported by this input.</li> <li>• No L-band sensor can be activated.</li> </ul>	

## Example

Setting dual-sensor mode:

**\$PASHS,SNS,DUO**

## Comments

- **Important!** This command restores some default values automatically to all the parameters usually controlled by the following commands:
  - \$PASHS,1,BLN
  - \$PASHS,2,BLN

- \$PASHS,3,BLN
- \$PASHS,RTK
- \$PASHS,3DF
- \$PASHS,BRV

The restored default values relevant to these six commands depend on the mode (SOL or DUO) you chose with the \$PASHS,SNS command.

- Using this command does not request the board to be restarted.
- The command is NAKed if you choose “DUO” and the [D] option has not been activated beforehand.
- Default operation is in SOL mode with d2=0, meaning whatever the antenna input used, you can be sure the board will deliver a position.
- In DUO mode, the board’s CPU speed is doubled, allowing higher data throughput, but power consumption is increased compared to that using SOL.
- In DUO mode, signal tracking is forced to optimal (i.e. the setting controlled by \$PASHS,OBS cannot be “ALL” in this case, and so will automatically be set to “OPT” whenever you choose “DUO”). If you choose SOL however, you have the choice between “OPT” and “ALL” for \$PASHS,OBS.
- In DUO mode, the second of the three Blade engines automatically starts computing the heading relevant to the vector oriented from antenna 2 to antenna 1.  
 With d2=1, heading is determined with L1 only. By default, first, the heading process goes through a calibration stage to determine the baseline length. Then the process is constrained with the computed baseline length to improve the performance of the L1-only heading computation. This is called computing heading in fixed mode.  
 With d2=2, heading is determined using only L1, L2 and L3 signals by default, i.e. without the need to determine the baseline length through a prior calibration stage. This is called computing heading in flex mode.  
 However, in DUO mode and for all values of d2, additional commands exist to request auto-calibration, or to enter the baseline length manually (see 3DF - 3DF,CLB - 3DF,Vxx set commands).
- In DUO mode, Trimble RTX is allowed only using antenna input #1, which means d2 should be “0” or “1”. Trimble RTX is not allowed with antenna input #2.

**Related Commands** \$PASHQ,SNS (query command)  
\$PASHQ,AST

## STI: Defining a Station ID

---

**Function** This command is used to define the station ID the base will broadcast in its differential messages to the rover.

### Command Format Syntax

\$PASHS,STI,d1[,s2,s3,s4][\*cc]

### Parameters

Parameter	Description	Range
d1	Station ID (default is 31)	0-1023 (RTCM 2.3) 0-4095 (RTCM 3.x and ATOM) 0-31 (CMR & CMR+)
s2	Short base name (used to generate CMR and CMR+ messages)	8 characters max.
s3	Long base name (used to generate CMR and CMR+ messages)	51 characters max.
s4	Base code (used to generate CMR and CMR+ messages)	16 characters max.
*cc	Optional checksum	*00-*FF

### Examples

Defining station ID “150” for use in RTCM messages:

\$PASHS,STI,150\*23

### Comment

- If the chosen station ID is beyond the upper limit in the applicable range, then the value “31” is chosen instead (i.e. “31” instead of “56” for example if CMR/CMR+ messages are broadcast, or “31” instead of “1041” for example if RTCM 2.3 messages are broadcast).
- When used in DUO mode, the entered station ID is the one attached to the main sensor (antenna #1). The data tagged to the other sensor will be identified with an incremented station ID (entered station ID for main sensor

+1). This rule applies in all cases, including when “31” is used in lieu of the incorrectly entered ID.

**Related Commands** \$PASHQ,STI (query command)

## TCP,PAR: TCP/IP Server Settings

---

**Function** This command is used to set the TCP/IP server.

**Command Format** **Syntax**

```
$PASHS,TCP,PAR[,MOD,s1][,LGN,s2][,PWD,s3][,PRT,d4][,IPP,c5]
[,s6,RIP][*cc]
```

### Parameters

Parameter	Description	Range
MOD,s1	<ul style="list-style-type: none"> <li>• 0: Disabled</li> </ul> TCP/IP connection mode: <ul style="list-style-type: none"> <li>• 1: Input/output with authentication</li> <li>• 2: Input/output without authentication (default)</li> <li>• 3: Output only. Allows multiple connections</li> </ul> UDP/IP connection mode: <ul style="list-style-type: none"> <li>• 4: Broadcast transmit (output only)</li> <li>• 5: Broadcast receive (input only)</li> <li>• 6: Transmit to remote IP address (output only)</li> <li>• 7: Received from remote IP address (input only)</li> </ul>	0-7
LGN,s2	Login, case sensitive (default: trimble)	32 char. max.
PWD,s3	Password, case sensitive (default: empty)	32 char. max.
PRT,d4	Port number. Default is "8888"	100-65535
IPP,c5	Port name (default: I; or of c5 not specified)	I, F
RIP,s6	Remote IP address (for UDP/IP modes 6 and 7). Default: 255.255.255.255	32 char. max.
*cc	Optional checksum	*00-*FF

### Example

Enabling TCP/IP connection with authentication (login: BX312, password: xwsead):

```
$PASHS,TCP,PAR,MOD,1,LGN,BX312,PWD,xwsead,PRT,88889,IPP,F
```

### Comments

- When the TCP/IP server is enabled (s1=1 or 2) and the receiver is connected to a network via Ethernet, WiFi or Modem, an external device can open the port specified as

**d4** and communicate with the board. In this case, the current port is port “I” or “F” on the board.

- When `s1=1`, the board does not accept any incoming data or commands until the Web Server password has been changed (see `$PASHS,TCP,UID`). It will however output those messages that are programmed on port “I” or “F” even if it has not received authentication yet.

**Related Commands**    `$PASHQ,TCP` (query command)  
                               `$PASHS,TCP,UID`

## TCP,UID: TCP/IP Authentication

---

**Function**    This command is used to enter the login and a password allowing a TCP/IP connection (requiring authentication) to be established.

**Command Format**    **Syntax**

`$PASHS,TCP,UID,s1,s2[*cc]`

### Parameters

Parameter	Description	Range
<code>s1</code>	Login	32 characters max.
<code>s2</code>	Password	32 characters max.
<code>*cc</code>	Optional checksum	*00-*FF

### Example

Entering authentication parameters (login: `BX312`, password: `xwsead`):

`$PASHS,TCP,UID,BX312,xwsead*70`

### Comments

- The `$PASHS,TCP,UID` command should always be sent first every time a user tries to connect to a remote receiver through a secure TCP/IP connection (see `$PASHS,TCP`). Only after providing authentication parameters will the user be allowed to send commands or data to that receiver.
- When the login and password are correct, or no authentication is required, the receiver will return the following reply:

`$PASHR,TCP,OK*1B`



- If authentication is required and the login or password is wrong, the receiver will return the following reply:  
\$PASHR,TCP,FAIL\*1D

**Related Commands** \$PASHS,TCP,PAR

## TOP: Defining the Type of Output Position

---

**Function** This command is used to define the best position solution the receiver is allowed to output through NMEA and ATM,PVT messages.

**Command Format** **Syntax**

\$PASHS, TOP, s1[\*cc]

### Parameters

Parameter	Description	Range	Default
s1	Position type: <ul style="list-style-type: none"> <li>• “RTK”: Up to RTK</li> <li>• “PPP”: Up to PPP position</li> <li>• “DIF”: Up to RTCM differential</li> <li>• “SBA”: Up to SBAS differential</li> <li>• “ALO”: Only standalone</li> </ul>	RTK, PPP, DIF, SBA, ALO	RTK
*cc	Optional checksum	*00-*FF	

### Comments

- The receiver can compute several types of position solutions simultaneously. By default, the receiver will provide the best position solution computed (up to RTK). In some cases however, you may prefer to get a position solution of lesser quality because you think it is more robust. That’s exactly what this command allows you to do.
- Choosing a type of output position does not mean this type of position will always be output. If you choose RTK and the conditions to get this type of position status are not met (no corrections available, or computed position does not meet the quality criteria), then only a position of lesser quality will be provided.

- Whatever your choice of the output position type, the content of the VEC (baseline) and HPR (attitude) messages will not be affected by this choice.
- PPP position refers to Processing Precise State Space (SS) corrections like RTCM-3 SSR messages or Trimble Centerpoint/RTX data. PPP can often deliver the same level of accuracy as RTK but its convergence time is longer.

### Example

Choosing RTCM differential as the best position solution to output:

```
$PASHS, TOP, DIF*59
```

**Related Commands**    \$PASHQ, TOP  
                               \$PASHQ, PAR

## UDP: User-Defined Dynamic Model Parameters

---

**Function**            This command is used to set the upper limits of the dynamic model (velocity, acceleration).

**Command Format**    **Syntax**

```
$PASHS, UDP, f1, f2, f3, f4[*cc]
```

### Parameters

Parameter	Description	Range	Default
f1	Maximum expected horizontal velocity in m/s.	0-100 000	100 000
f2	Maximum expected horizontal acceleration in m/s/s.	0-100	100
f3	Maximum expected vertical velocity in m/s.	0-100 000	100 000
f4	Maximum expected vertical acceleration in m/s/s.	0-100	100
*cc	Optional checksum	*00-*FF	-

### Example

Setting the dynamic model:

```
$PASHS, UDP, 10, 1, 2, 0.5*1D
```

## Comments

The user-defined dynamic model is activated by the \$PASHS,DYN,9 command. Note that when the adaptive dynamic mode (DYN,8) is selected, the user-defined model is automatically excluded from the possible models that could best describe the current receiver dynamics.

**Related Commands** \$PASHQ,UDP (query command)  
\$PASHS,DYN

## UPL,PAR: Setting the Firmware Upgrade Procedure

---

**Function** This command allows you to decide on whether firmware upgrades will take place using an internal upgrade file copied earlier to the board's internal memory, or using a connection to an FTP server to download the upgrade file. If an FTP is used, the command also allows you to fully define access to this server.

**Command Format** **Syntax**  
\$PASHS,UPL,PAR[,MOD,d1][,ADD,s2][PRT,d3][,LGN,s4][,PWD,s5][,PTH,s6][\*cc]

### Parameters

Parameter	Description	Range	Default
Mod,d1	Upgrade mode: • 0: Internal upgrade file • 1: FTP	0, 1	0
ADD,s2	IP address or host name	32 characters max.	
PRT,d3	Port number	0-65535	21
LGN,s4	Login	32 characters max.	
PWD,s5	Password	32 characters max.	
PTH,s6	Path used on the FTP server	255 characters max.	
*cc	Optional checksum	*00-*FF	

### Example

```
$PASHS,UPL,PAR,MOD,1,ADD,ftp.intec.com,PRT,21,LGN,spectra,PWD,u6huz8,PTH,/my folder*7F
```

**Related Commands** \$PASHQ,UPL  
\$PASHS,UPL,UPG

\$PASHQ,UPL,LST

## UPL,UPG: Upgrading the Receiver Firmware from FTP

---

**Function** This command is used to download a firmware upgrade from the FTP server declared with \$PASHS,UPL,PAR, and then perform the upgrade.

### Command Format Syntax

```
$PASHS,UPL,UPG[s1]*cc
```

### Parameters

Parameter	Description	Range
s1	Name of the upgrade file that will be first downloaded to the receiver and then used to perform the firmware upgrade. <ul style="list-style-type: none"> <li>The file name can contain a relative path to the path defined BY \$PASHS,UPL,PAR.</li> <li>If s1 is missing or only consists of a path, then "MB2_upgrade_v1.2.tar" is downloaded, provided there is only one of these files available on the FTP server, otherwise the command will be NAKed.</li> </ul>	255 characters max.
*cc	Optional checksum	*00-*FF

### Example

Upgrading from file "MB2\_upgrade\_v1.2.tar" found on the FTP server:

```
$PASHS,UPL,UPG,MB2_upgrade_v1.2.tar*50
```

After successful completion of the file transfer to the receiver, the following response line is returned:

```
$PASHR,UPL,REBOOT,MB2_upgrade_v1.2.tar*63
```

Then, communication with the receiver is suspended until upgrade installation is complete.

Should the file transfer fail, the following response line will appear:

```
$PASHR,UPL,FAIL,MB2_upgrade_v1.2.tar*60
```

### Related Commands

```
$PASHQ,UPL
```

```
$PASHS,UPL,PAR
```

```
$PASHQ,UPL,LST
```

## VIP: Defining a Virtual Port

---

**Function** This command is used to define a virtual port. Using a virtual port is possible only after it has been associated with a physical port. It is indeed through the physical port that the virtual port can be accessed.

### Command Format Syntax

```
$PASHS,VIP,c1,c2[*cc]
$PASHS,VIP,c1,OFF[*cc]
```

### Parameters

Parameter	Description	Range
c1	Virtual port	Z
c2	Physical port	A, B, C, D, F, I, P, Q, M, U
*cc	Optional checksum	*00-*FF

### Comments

- To date, only virtual port “Z” can be defined. Port Z may be associated with any of the existing physical ports. By default, the virtual port is OFF.
- The transport layer used to create a virtual port is the ATM,DAT,EXT message (see *ATOM Reference Manual*). The frame of this message always includes a single, complete, original receiver message (any of those described in this manual).

### Example

Delivering simultaneously both corrections and raw data on port A:

- Delivering corrections on physical port A:  
**\$PASHS,ATM,RNX,A,ON,0.1,&SCN,204\*48**
- Delivering raw data on virtual port Z associated with physical port A:  
**\$PASHS,VIP,Z,A\*21**  
**\$PASHS,ATM,RNX,Z,ON,0.1,&SCN,204\*53**

Turning off virtual port Z:

```
$PASHS,VIP,Z,OFF*2F
```

## WEB,OWN: Setting Owner Information

---

**Function** This command is used to define the owner information displayed on the home page of the Web Server.

**Command Format Syntax**

`$PASHS,WEB,OWN,s1,s2,s3,s4[*cc]`

**Parameters**

Parameter	Description	Range
s1	Company name	255 characters max.
s2	Administrator name	255 characters max.
s3	Administrator email	255 characters max.
s4	Administrator phone number	255 characters max.
*cc	Optional checksum	*00-*FF

**Example**

`$PASHS,WEB,OWN,Ashtech,Peter Smith,psmith@ashtech.com,  
0228093800*5C`

**Relevant Query Command** `$PASHQ,WEB`

**See Also** `$PASHS,WEB,PAR`

## WEB,PAR: Web Server Control & Administrator Profile

---

**Function** This command is used to enable or disable the use of the Web Server and define the profile of the receiver administrator. There is necessarily one –and just one– administrator profile per receiver.

**Command Format Syntax**

`$PASHS,WEB,PAR,s1[,s2,s3[,d4]][*cc]`

## Parameters

Parameter	Description	Range	Default
s1	Enables (ON) or disables (OFF) the Web Server	ON, OFF	ON
s2	Administrator login	32 char. max.	admin
s3	Administrator password	32 char. max.	password
d4	httpd port	0-65535	80
d5	Security level: • 0: Enabled • 1: Enabled with anonymous access	0-1	0
d6	Privileges for anonymous access: • 0: No privileges • 1: Only downloading files is allowed. • 2: Only deleting files is allowed. • 3: Downloading or deleting files is allowed.	0-3	0
*cc	Optional checksum	*00-*FF	

## Example

Enabling the use of the Web Server with specific login and password on httpd port 2500:

```
$PASHS,WEB,PAR,ON,Smith,u7lmyt,2500*69
```

## Comments

- The login and password are NOT set to their default values when running the \$PASHS,RST or \$PASHS,INI command.
- The httpd port is used to access the Web Server through the network.

If for example the IP address of the receiver is 10.20.2.18 and d4=2500, you should enter the following in the address bar of your web browser to open the Web Server:

```
10.20.2.18:2500
```

## See Also

\$PASHQ,WEB (relevant query command)

\$PASHS,WEB,PAR

## ZDA: Setting Date & Time

---

### Function

This command is used to set the date and time in the receiver.

**Command Format Syntax**`$PASHS,ZDA,m1,d2,d3,d4[*cc]`**Parameters**

Parameter	Description	Range
m1	UTC time (hhmmss.ss)	000000.00-235959.99
d2	Current day	01-31
d3	Current month	01-12
d4	Current year	1980-9999
*cc	Optional checksum	*00-*FF

**Example**`$PASHS,ZDA,151145.00,13,03,2008*0A`**Relevant Query Command**`$PASHQ,ZDA`



# Chapter 8. Query Command Library

This chapter provides a detailed description of all the \$PASHQ commands applicable to the MB-Two board. The commands are listed in alphabetical order, irrespective of their use domain.

## Command Format Syntax

\$PASHQ,3DF[\*cc]

## Response Format

---

### Syntax

\$PASHR,3DF,s1[,d2[,d3[,d4]]],V12,f5,f6,f7,f8,V13,f9,f10,f11,f12,V14,f13,f14,f15,f16,CLB,d17,OFS,f18,f19,f20,ANG,d21,MXB,f22,OUT,s23,s24\*cc

### Parameters

Parameter	Description	Range
s1	Attitude processing status: <ul style="list-style-type: none"><li>• OFF: Processing disabled</li><li>• HED: Heading processing active</li><li>• ADU: Attitude processing active</li></ul>	OFF, HED, ADU
d2	Number of 1st baseline (i.e. referring to V12) used in attitude processing.	1-3
d3	Number of 2nd baseline (i.e. referring to V13) used in attitude processing.	1-3
d4	Number of 3rd baseline (i.e. referring to V14) used in attitude processing.	1-3
V12,f5-f8	Calibrated or entered V12 vector components and corresponding RMS value, in meters.	
V13,f9-f12	Calibrated or entered V13 vector components and corresponding RMS value, in meters.	
V14,f13-f16	Calibrated or entered V14 vector components and corresponding RMS value, in meters.	
CLB,d17	Calibration indicator: <ul style="list-style-type: none"><li>• "0": Calibration complete</li><li>• "1": Calibration in progress</li><li>• Empty: Attitude mode off</li></ul>	
OFS,f18-f20	Heading offset, pitch offset, roll offset respectively, all in degrees (from \$PASHS,3DF,OFS)	

Parameter	Description	Range
ANG,d21	Maximum value of baseline elevation, in degrees (from \$PASHS,3DF,ANG)	
MXB,f22	Vector tolerance, in meters (from \$PASHS,3DF,MXM)	
OUT,s23	Type of attitude output: <ul style="list-style-type: none"> <li>• TT: Time-Tagged</li> <li>• FST: Fast</li> </ul>	TT, FST
s24	In heading mode only, "FLX" means heading computed in flex mode. If this field is empty, heading is computed with fixed baseline length.	FLX or empty
*cc	Checksum	*00-*FF

**Example**

\$PASHQ,3DF

```
$PASHR,3DF,HED,2,V12,45.870,,,0.003,V13,,,,,V14,,,,,CLB,1,OFS,0.000,0.000,0.000,ANG,15,MXB,0.010*,OUT,FST,FLX*1A
```

**ALR: Listing Current Alarms****Function**

This command queries the receiver for all the currently raised alarms. You can select which alarm levels you want to list.

**Command Format****Syntax**

```
$PASHQ,ALR[,d][*cc]
```

**Parameters**

Parameter	Description	Range
d	Alarm level: <ul style="list-style-type: none"> <li>• 1: High and medium</li> </ul>	1
*cc	Optional checksum	*00-*FF

**Response Format****Syntax**

```
$PASHR,ALR,d1,d2,c3,s4,d5,s6*cc
```

## Parameters

Parameter	Description	Range
d1	Alarm code	0-255
d2	Alarm sub-code	0-255
c3	Stream ID reporting the alarm (if relevant, otherwise blank field): <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• M: Internal memory</li> <li>• F, I, P, Q: TCP/IP stream</li> <li>• U: USB memory</li> </ul>	A-D, F, I, P, Q, M, U
s4	Alarm category	INPUT MEMORY NETWORK OTHER PVT RADIO
d5	Alarm level: <ul style="list-style-type: none"> <li>• 1: Medium</li> <li>• 2: High</li> </ul>	1, 2
s6	Description	
*cc	Checksum	*00-*FF

**Example**      `$PASHQ,ALR`  
`$PASHR,ALR,2,0,B,INPUT,"Bad parameter"*1F`

## ANH: Antenna Height

---

**Function**      This command allows you to read the entered antenna height as well as the measurement type used for each antenna.

**Command Format**      **Syntax**  
`$PASHQ,ANH[*cc]`

**Response Format**      **Syntax**  
`$PASHR,d0,ANH,f1,c2*cc`

## Parameters

Parameter	Description	Range
d0	Antenna identification: • 1: Primary antenna • 2: Secondary antenna	1, 2
f1	Antenna height.	0-6.553 m 6.553-100 m
c2	Antenna height measurement type: • V: Vertical measurement • S: Slant measurement	V, S
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,ANH
$PASHR,1,ANH,1.604,V*55
$PASHR,2,ANH,1.600,V*52
```

### Related Commands

```
$PASHS,ANH (set command)
$PASHQ,ANR
```

## ANP: Antenna Parameters

---

### Function

This command allows you to read the antenna parameters of the specified antenna name.

### Command Format

#### Syntax

```
$PASHQ,ANP,s1[*cc]
```

#### Parameters

Parameter	Description	Range
s1	Antenna name (case sensitive). The command will be NAKed if this field is missing.	31 characters max.
*cc	Optional checksum	*00-*FF

### Response Format

#### Example

```
$PASHQ,ANP,ASH111660
BEGIN ANTENNA
ASH111660
L1 N: +000.30 E: -000.70 U: +075.40
L1 PAE: +000.0 +000.7 +001.7 +002.8 +004.1 +005.3 +006.3 +007.2 +007.7
+007.8
```

```

+007.6 +006.9 +005.6 +003.7 +001.2 -001.9 -005.7 +000.0 +000.0
L2 N: +000.00 E: +000.00 U: +000.00
L2 PAE: +000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0
+000.0
+000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0 +000.0
END ANTENNA

```

## ANP,OUT: Virtual Antenna

---

**Function** This command returns the name of the virtual antenna currently selected in the receiver. The choice of virtual antenna reported by the command is effective for the two antennas.

**Command Format Syntax**  
**\$PASHQ,ANP,OUT[\*cc]**

**Parameters**

None.

**Response Format Syntax**  
**\$PASHR,ANP,OUT,s1\*cc**

**Parameters**

Parameter	Description	Range
s1	Name of the virtual antenna used. If s1 is blank, that means no virtual antenna is selected.	31 characters max.
*cc	Checksum	*00-*FF

**Example**  
**\$PASHQ,ANP,OUT**  
**\$PASHR,ANP,OUT,ADVNULLANTENNA\*72**

**Relevant Set Command**  
**\$PASHS,ANP,OUT**

## ANP,OWN: Local Antenna Used

---

**Function** This command returns the name of the GNSS antenna currently used by the receiver. The command returns a line per antenna.

**Command Format Syntax**

\$PASHQ,ANP,OWN[\*cc]

**Parameters**

None.

**Response Format Syntax**

\$PASHR,d0,ANP,OWN,s1[,s2][,s3]\*cc

**Parameters**

Parameter	Description	Range
d0	Antenna identification: <ul style="list-style-type: none"> <li>• 1: Primary antenna</li> <li>• 2: Secondary antenna</li> </ul>	1, 2
s1	Name of the local antenna	31 characters max.
s2	Antenna serial number	31 characters max.
s3	Antenna setup ID	0-255
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,ANP,OWN
$PASHR,1,ANP,OWN,UNKNOWN,,0*18
$PASHR,2,ANP,OWN,ASH111661,,0*55
```

**Related Commands** \$PASHS,ANP,OWN (set command)**ANP,RCV: Antenna Name and Offsets of Received Base**

---

**Function** This command queries the receiver for the antenna name and offsets of the received base.**Command Format Syntax**

\$PASHQ,ANP,RCV[\*cc]

**Response Format Syntax**

\$PASHR,ANP,RCV,s1,f2,f3,f4,f5,f6,f7\*cc

**Parameters**

Parameter	Description
s1	Antenna name, "NONE" if non name received for the base antenna.
f2	L1 North offset, in mm

Parameter	Description
f3	L1 East offset, in mm
f4	L1 Up offset, in mm
f5	L2 North offset, in mm
f6	L2 East offset, in mm
f7	L2 Up offset, in mm
*cc	Checksum

**Example**      \$PASHQ,ANP,RCV  
 \$PASHR,ANP,RCV,ASH802147,-2.00,0.70,103.00,-3.4,-2.2,103.80\*09

## ANP,REF: Antenna Used at the Base

---

**Function**      This command returns the name of the GNSS antenna assumed to be used by the base currently sending data to the interrogated receiver (a rover).

**Command Format**      **Syntax**  
 \$PASHQ,ANP,REF[\*cc]

**Parameters**

None.

**Response Format**      **Syntax**  
 \$PASHR,ANP,REF,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Name of the antenna used at the base	31 characters max.
*cc	Checksum	*00-*FF

**Example**      \$PASHQ,ANP,REF  
 \$PASHR,ANP,REF,ASH111661\*20

**Related Commands**      \$PASHS,ANP,REF (set command)

## ANR: Antenna Reduction Mode

---

**Function** This command is used to read the current setting for the antenna reduction mode. This setting defines the physical location on the system for which the position is computed. This setting applies to both antennas

**Command Format Syntax**  
\$PASHQ,ANR[\*cc]

**Response Format Syntax**  
\$PASHR,ANR,s1\*cc

### Parameters

Parameter	Description	Range
s1	Antenna reduction mode: <ul style="list-style-type: none"> <li>• OFF or PC1: The computed position is assumed to be the location of the antenna's L1 phase center.</li> <li>• ON or SPT: The computed position is assumed to be the location of the ground mark.</li> <li>• ARP: The computed position is assumed to be the location of the Antenna Reference Plane (ARP).</li> </ul>	OFF, ON, ARP, PC1, SPT
*cc	Checksum	*00-*FF

**Example** \$PASHQ,ANR  
\$PASHR,ANR,ON\*04

**Related Commands** \$PASHS,ANR  
\$PASHS,ANH

## ANT: Antenna Height

---

**Function** This command is used to read the current setting for the antenna height (one response line per antenna).

**Command Format Syntax**  
\$PASHQ,ANT[\*cc]



**Response Format Syntax**

\$PASHR,d0,ANT,f1,f2,f3\*cc

**Parameters**

Parameter	Description	Range
d0	Antenna identifier: • 1: Primary antenna • 2: Secondary antenna	1, 2
f1	Slant height measurement, from ground mark to antenna edge (SHMP)	0-6.553 m 6.553-99.999 m
f2	Antenna radius: horizontal distance from the geometrical center to the antenna edge.	0-6.553 m
f3	Antenna vertical offset: • Offset between SHMP and ARP if both slant height measurement and antenna radius are different from zero. • Offset between ground mark and ARP if either slant height measurement or radius is zero.	± 0-6.553 m 6.553-99.999 m
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,ANT
$PASHR,1,ANT,0,0,2.000*32
$PASHR,2,ANT,0,0,1.985*36
```

**Related Commands**

```
$PASHS,ANT
$PASHQ,ANR
$PASHQ,ANH
```

**ARA: Attitude Rate and Accuracy**

---

**Function**

This command is used to output the ARA message providing values and variation speeds of the different components of attitude. This messages comes as a supplement to the HPR message.

**Command Format Syntax**

\$PASHQ,ARA[\*cc]

**Response Format Syntax**

\$PASHR,ARA,f1,m2,f3,f4,f5,f6,f7,f8,f9\*cc

## Parameters

Parameter	Description	Range
f1	Fixed parameter set to "0". If it weren't "0", then the next f3-f8 fields in the response line would provide invalid values.	0
m2	Current UTC time of attitude fix (hhmmss.ss)	000000.00-235959.99, same as in \$PASHQ,POS
f3	Heading speed in degrees/second	Full range of real variables
f4	Pitch speed in degrees/second	Full range of real variables
f5	Roll speed in degrees/second	Full range of real variables
f6	RMS heading angle, in degrees	Full range of real variables
f7	RMS pitch angle, in degrees	Full range of real variables
f8	RMS roll angle, in degrees	Full range of real variables
f9	Attitude extrapolation interval, in milliseconds	Empty if not defined
*cc	Checksum	*00-*FF

## Comments

- Variation speeds of angles may be provided even though the values of these angles are missing. Reciprocally, speed values may be missing while the corresponding angles are determined and output.
- Speed values are signed as follows:
  - A minus sign for pitch speed means going downwards and a positive sign means going upwards.
  - A minus sign for roll speed means going to the left (to port) and a positive sign means going to the right (to starboard).
  - A minus sign for heading speed means the bow turning to the left and a positive sign means the bow turning to the right.

## Example

### \$PASHQ,ARA

```
$PASHR,ARA,0,103024.00,-0.016,-0.036,,0.154,0.271,,0*26
$PASHR,ARA,0,103025.00,0.011,0.004,,0.154,0.271,,0*21
$PASHR,ARA,0,103026.00,-0.001,-0.002,,0.154,0.271,,0*25
$PASHR,ARA,0,103027.00,-0.005,-0.036,,0.155,0.271,,0*26
$PASHR,ARA,0,103028.00,-0.016,0.004,,0.155,0.271,,0*07
$PASHR,ARA,0,103029.00,0.034,-0.002,,0.155,0.271,,0*00
$PASHR,ARA,0,103030.00,-0.026,0.004,,0.155,0.271,,0*0D
```

**Related Commands**    \$PASHS,NME

## Automatic Output of ARR Messages

This is a reminder on how to output ARA messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,ARA,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output ARA messages on port A at a rate of 0.2 second:

```
$PASHS,NME,ARA,A,ON,0.2
```

## ARR: Vector & Accuracy Data

---

**Function** This command is used to output the ARR message, which is an extension of the legacy VEC message. VCT can be requested only in a mode set up by \$PASHS,RTK. The baseline vector is oriented from the base to the rover.

**Command Format** **Syntax**  
\$PASHQ,ARR[\*cc]

**Response Format** **Syntax**  
\$PASHR,ARR,d0,d1,d2,m3,f4,f5,f6,f7,f8,f9,f10,f11,f12,d13,d14,d15,d16\*cc

## Parameters

Parameter	Description	Range
d0	Vector number	1, 2, 3
d1	Vector mode: <ul style="list-style-type: none"> <li>• 0: Invalid baseline</li> <li>• 1: Differential</li> <li>• 2: RTK float</li> <li>• 3: RTK fixed</li> <li>• 5: Other (dead reckoning, bad accuracy, difference between standalone positions). Messages with d1=5 may further be masked if users only want proven vector estimates.</li> </ul>	0-3, 5
d2	Number of SVs used in baseline computation (L1 portion)	0-99
m3	UTC time (hhmmss.ss)	000000.00-235959.99
f4	Delta antenna position, ECEF 1st coordinate (in meters)	±99999.999
f5	Delta antenna position, ECEF 2nd coordinate (in meters)	±99999.999
f6	Delta antenna position, ECEF 3rd coordinate (in meters)	±9999.999
f7	1st coordinate of standard deviation	99.999
f8	2nd coordinate of standard deviation	99.999
f9	3rd coordinate of standard deviation	99.999
f10	1st/2nd coord. correlation	±99.999999
f11	1st/3rd coord. correlation	±99.999999
f12	2nd/3rd coord. correlation	±99.999999
c13	Reference data ID	1, 2, port letter
d14	Vector coordinate frame ID: <ul style="list-style-type: none"> <li>• 0: XYZ</li> <li>• 1: ENU centered on rover</li> <li>• 2: ENU centered on base</li> </ul>	0-2
d15	Vector operation: <ul style="list-style-type: none"> <li>• 0: Fixed mode (vector length is constrained)</li> <li>• 1: Calibration (vector length is being calibrated)</li> <li>• 2: Flex mode</li> </ul>	0-2
d16	Clock assumption: <ul style="list-style-type: none"> <li>• 0: Clock is assumed to be different for the “head” and “tail” of the vector (see Comments below)</li> <li>• 1: Clock is assumed to be the same for the “head” and “tail” of the vector (see Comments below)</li> </ul>	
*cc	Checksum	*00-*FF

## Comments

- “Head” refers to the location the vector arrow points to (that is, the rover location) while “Tail” refers to the other

end of the vector (the vector origin), which represents the moving base location.

- So depending on the operating mode (internal heading, attitude, etc.) and the board configuration (one-sensor, dual-sensor, multi-board), field c13 will be an internal sensor number (case of internal heading or attitude) and/or and external sensor ID identified by the ID of the port through which its corrections enter the board.

It is assumed that the “head” of a vector may be identified by reading \$PASHD messages (see *page 40*, *page 40* or *page 40*).

### Example

#### \$PASHQ,ARR

```
$PASHR,ARR,2,3,26,100613.00,-1.445,0.298,1.320,0.005,0.003,0.006,-
0.021143,0.573289,0.184693,D,0,1,0*5D
```

```
$PASHR,ARR,3,3,26,100613.00,-1.179,2.232,1.151,0.005,0.003,0.006,-
0.021143,0.573289,0.184693,D,0,1,0*50
```

**Related Commands**    \$PASHS,NME

### Automatic Output of ARR Messages

This is a reminder on how to output ARR messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,ARR,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output ARR messages on port A at a rate of 0.2 second:

```
$PASHS,NME,ARR,A,ON,0.2
```

## AST: Status of Antennas

---

**Function**    This command queries the board for the status of the main antenna (if the board is in SOLO mode) or of the two antennas (if in DUO mode).

**Command Format**    **Syntax**  
                           \$PASHQ,AST[\*cc]

**Response Format Syntax**

\$PASHR,d1,AST,s2,d3,s4\*cc

**Parameters**

Parameter	Description	Range
d1	Antenna input index: <ul style="list-style-type: none"> <li>• 1: Main antenna (antenna #1)</li> <li>• 2: Antenna #2</li> </ul>	1, 2
s2	Antenna status: <ul style="list-style-type: none"> <li>• CON: Antenna is connected (DC output current is in the range 5 mA to 150 mA)</li> <li>• NC: No antenna connected (DC output current is less than 5 mA)</li> <li>• SHR: Antenna cable shorted (DC output current is greater than 150 mA)</li> <li>• UNK: Status unknown</li> </ul>	CON, NC, SHR, UNK
d3	RF signal level: <ul style="list-style-type: none"> <li>• Empty field if s2= NC, SHR or UNK</li> </ul>	0... 4095 or empty
s4	Always OFF.	OFF
*cc	Checksum	*00.*FF

**Example** With two antennas used (DUO mode active):

```
$PASHQ,AST
$PASHR,1,AST,CON,3516,OFF*0F
$PASHR,2,AST,3842,OFF*00
```

**ATH: Listing Anti-Theft Parameters**

---

**Function** This command queries the receiver for the current anti-theft parameters.**Command Format Syntax**

\$PASHQ,ATH[\*cc]

**Parameters**

None.

**Response Format Syntax**

\$PASHR,ATH,s1[,s2,SMS,d3,...,d7,EML,d8,...,d12,MSG,s13,s14,s15]\*cc

## Parameters

Parameter	Description	Range
s1	Anti-theft protection current status: <ul style="list-style-type: none"> <li>• OFF: Disabled</li> <li>• ON: Enabled</li> <li>• ALR: Enabled + Alarm raised</li> </ul>	ON, OFF,ALR
s2	Password (only if anti-theft protection is OFF)	64 char. max.
d3... s15	Not applicable	
*cc	Checksum	*00-*FF

## Examples

```
$PASHQ,ATH
$PASHR,ATH,OFF,password,SMS,,,,,EML,,,,,MSG,,, *29
```

```
$PASHQ,ATH
$PASHR,ATH,ON*cc
```

## Related Commands

```
$PASHS,ATH,ON
$PASHS,ATH,OFF
$PASHS,ATH,PWD
$PASHS,ATH,EML
$PASHS,ATL,SMS
$PASHS,ATH,MSG
```

## ATL: Debug Data Recording

---

**Function** This command queries the receiver for the current status of the data recording function used for debugging (ATL data).

**Command Format** **Syntax**  
 \$PASHQ,ATL[\*cc]

**Response Format** **Syntax**  
 \$PASHR,ATL,s1,d2,c3,f4,d5,d6,d7\*cc

## Parameters

Parameter	Description	Range
s1	ON/OFF/AUT status: <ul style="list-style-type: none"> <li>• ON: ATL data recording is enabled but will not re-start after a power cycle.</li> <li>• OFF: ATL data recording is disabled.</li> <li>• AUT: ATL data recording is enabled and will re-start after a power cycle.</li> </ul>	ON, OFF, AUT
d2	Identification of output port used: <ul style="list-style-type: none"> <li>• Serial Port: A, B, D</li> <li>• USB serial port: C</li> <li>• TCP/IP port: I</li> <li>• (empty if no output)</li> </ul>	A, B, C, D, I, blank
c3	Recording status: <ul style="list-style-type: none"> <li>• R: The receiver is currently recording data for debugging.</li> <li>• S: No debug data currently recorded.</li> </ul>	R, S
f4	Output rate, in seconds (default: 1 sec.)	0.05, 0.1, 0.2, 0.5, 1
d5	Configuration index	0, 1
d6	Memory location: <ul style="list-style-type: none"> <li>• 0: Internal memory</li> <li>• 2: USB memory</li> </ul>	0, 2
d7	Maximum duration, in minutes (0: unlimited duration)	0, 15, 20, 30, n x 60 with n between 1 and 24
*cc	Checksum	*00-*FF

**Examples** ATL data recording disabled:

```
$PASHQ,ATL*2E
$PASHR,ATL,OFF,,S,1,0*cc
```

ATL data recording enabled and in progress:

```
$PASHQ,ATL*2E
$PASHR,ATL,ON,,R,1,0*cc
```

ATL data recording is enabled but for some reason, no data is being recorded:

```
$PASHQ,ATL*2E
$PASHR,ATL,ON,,S,1,0*cc
```

ATL data output set on port A:

```
$PASHQ,ATL*2E
$PASHR,ATL,OFF,A,S,1,0*cc
```



## ATT: Heading, Roll and Pitch

---

**Function** This command allows you to output the heading, roll and pitch message.

**Command Format Syntax**  
`$PASHQ,ATT[*cc]`

**Response Format Syntax**  
`$PASHR,ATT,f1,f2,f3,f4,f5,f6,d7*cc`

### Parameters

Parameter	Description	Range
f1	Week time in seconds.	000000.00-604799.99
f2	True heading angle in degrees.	000.00-359.999999
f3	Pitch angle in degrees.	±90.00000
f4	Roll angle in degrees.	±90.00000
f5	Carrier measurement RMS error, in meters.	Full range of real variables
f6	Baseline RMS error, in meters.	Full range of real variables
d7	Integer ambiguity is "Fixed" or "Float": <ul style="list-style-type: none"> <li>• 0: Fixed</li> <li>• &gt;0: Float</li> </ul>	0, >0
*cc	Checksum	*00-*FF

### Comments

- All parameters are necessarily output at every epoch, except for f5 and f6 which may not be output for fractions of a second.
- This message can still potentially be output when attitude or heading is disabled (see 3DF), provided a differential mode is still running, in which case the reported heading stands for the baseline azimuth (base to rover) and the reported pitch stands for the baseline elevation.
- When the heading mode is enabled and as long as calibration is in progress, field f6 (baseline RMS error) is forced to zero. This field is therefore indicative of the calibration status:
  - f6 = 0, calibration in progress
  - f6 ≠ 0, calibration complete, heading computed and output

- Field  $d7=0$  does not necessarily mean that the position message (e.g. POS) will report a “fixed” position.
- Field  $d7\neq 0$  does not necessarily mean that the attitude solution is bad. With long baselines, a sub-degree attitude accuracy may be achieved even it’s a float-ambiguity solution.
- During calibration, fields f2 to f4 are reported to be invalid while some other fields may still have sense.

### Example

Querying the heading and roll/pitch message on the current port:

```
$PASHQ,ATT
$PASHR,ATT,310080.0,248.57,+04.22,,0.0027,0.0000,0*2B
```

### Automatic Output of ATT Messages

This is a reminder on how to output ATT messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,ATT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output ATT messages on port A at a rate of 0.5 second:

```
$PASHS,NME,ATT,A,ON,0.5
```

## AVR: Trimble Time-Yaw-Tilt-Range Message

---

### Function

This command is used to output an AVR message. AVR is similar to the existing Trimble Yaw-Tilt-Range message. It is available only for baselines involved in 3DF processing. When the AVR message is set to be output at regular intervals of time, and at a given moment the fields in the message happen to be all empty, be aware this won’t prevent the board from continuing to normally output the message at the required output rate (as opposed to some other messages which stop being output when they are empty).

NOTE: AVR data are affected by 3DF,OFS parameters and 3DF,OUT settings.

### Command Format

#### Syntax

```
$PASHQ,AVR[*cc]
```

## Response Format Syntax

```
$PTNL,AVR,m1,f2,Yaw,f3,Tilt,f4,s5,f6,d7,f8,d9*cc
```

### Parameters

Parameter	Description	Range
m1	Current UTC time of vector fix (hhmmss.ss), same as time field in HPR message	000000.00-235959.99
f2,Yaw	Yaw angle, in degrees. Message provides same value as "Heading" in HPR message. "+0.0000" if not available.	
f3,Tilt	Tilt angle, in degrees. Message provides same value as "Pitch" in HPR message. "+0.0000" if not available.	
f4,s5	Two empty fields	
f6	Range, in meters, representing the baseline length computed by the corresponding RTK engine. "0.000" if not available. In fixed mode, this field should report a value close to the actual baseline length. Variations can be of the centimeter level in flex mode, and of the millimeter level in fixed mode.	
d7	GNSS quality indicator: <ul style="list-style-type: none"> <li>• 0: Fix not available or invalid</li> <li>• 1: Autonomous position; yaw and tilt are invalid.</li> <li>• 2: Differential carrier phase solution RTK (float)</li> <li>• 3: Differential carrier phase solution RTK (fixed)</li> <li>• 4: Differential code-based solution; yaw and tilt are invalid.</li> </ul>	0-4
f8	PDOP (same value as the one reported in the HPR message. "0.0" if not available.	
d9	Number of satellites being used in the solution (i.e. for which carrier phases were used in RTK engine). "0.0" if not available. Not necessarily the same number as in HPR message.	
*cc	Checksum	*00-*FF

### Example

```
$PTNL,AVR,212405.20,+52.1531,Yaw,-0.0806,Tilt,,12.575,3,1.4,16*39
```

### Related Command \$PASHS,NME

### Automatic Output of AVR Messages

This is a reminder on how to output AVR messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,AVR,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output AVR messages on port A at a rate of 0.5 second:

```
$PASHS,NME,AVR,A,ON,0.5
```

## BDS: BeiDou Tracking Status

---

**Function** This command returns the current BeiDou tracking status.

**Command Format Syntax**  
\$PASHQ,BDS[\*cc]

**Parameters**  
None.

**Response Format Syntax**  
\$PASHR,BDS,s1,USE,n(c2)\*cc

### Parameters

Parameter	Description	Range
s1	BeiDou tracking status: <ul style="list-style-type: none"> <li>• ON: Currently tracking and using BeiDou satellites</li> <li>• OFF: Not tracking BeiDou satellites</li> </ul>	ON, OFF
n(c2)	Provides tracking status of each BeiDou satellite: <ul style="list-style-type: none"> <li>• Y: Tracking enabled for this satellite</li> <li>• N: Tracking disabled for this satellite</li> </ul> This status is provided n times, where n is the number of satellites in the BeiDou constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Checksum	*00-*FF

**Example** \$PASHQ,BDS  
\$PASHR,BDS,ON,USE,YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\*  
16

**Related Commands**    \$PASHS,BDS (Set command)

## **BLN: Reading Baseline Definitions & Status**

---

**Function**    This command is used to check the definition and status of each baseline you set with \$PASHS,BLN.

**Command Format**    **Syntax**  
                          \$PASHQ,BLN[\*cc]

**Response Format**    **Syntax**  
For each baseline:  
                          \$PASHR,d1,BLN,s2,BAS,s3,S4,ROV,s5,ARR,s6,f7,f8,f9,f10,d11\*cc

## Parameters

Parameter	Description	Range
d1	Client ID (baseline number).	1-3
s2	Baseline status or type: <ul style="list-style-type: none"> <li>• PRI: Primary baseline, processed</li> <li>• ON: Baseline processed</li> <li>• SAM: Supplementary baseline.</li> <li>• OFF: Baseline disabled (not processed)</li> </ul>	PRI, ON, SAM, OFF
s3	Base data source: <ul style="list-style-type: none"> <li>• A, B, C, etc.: A board's physical or virtual port through which external data enter the board.</li> <li>• "?" : Automatic selection of the board's physical port through which external data enter the board.</li> <li>• 1, 2, 3, etc: Sensor internal to the board. The figure indicates the number of the sensor used.</li> <li>• A1, A2, B1, B2, etc.: Multi-board configuration. The letter indicates the local board's port to which an external multi-antenna board is connected, and the figure indicates the number of the sensor (on that external board) that is the source of data to be processed.</li> <li>• ?1, ?2, etc.: Multi-board configuration. The "?" symbol indicates that there is an automatic selection of the local board's physical port through which data enter the local board, and the figure indicates the number of the sensor (on the external board) that is the source of data to be processed.</li> </ul>	<ul style="list-style-type: none"> <li>• A, B, C, ..</li> <li>• ?</li> <li>• 1, 2, 3, ..</li> <li>• A1, A2, B1, B2, ..</li> <li>• ?1, ?2, etc.</li> </ul>
s4	Base status: <ul style="list-style-type: none"> <li>• S: Static base</li> <li>• M: Moving base</li> </ul>	S, M
s5	Same as s3 above, but for rover data source.	Same as s3
s6	Arrow (heading) mode status (ON or OFF)	ON, OFF
f7, f8, f9	XYZ vector components. When only the length parameter is set or calibrated, f8 and f9 should be both "0" or empty.	
f10	RMS value of calibrated or entered vector, in meters	
d11	Arrow vector status: <ul style="list-style-type: none"> <li>• 0: Being calibrated</li> <li>• 1: Calibrated</li> <li>• 2: Entered</li> </ul>	0, 1, 2
*cc	Checksum	*00-*FF

### Example

**\$PASHQ,BLN**

\$PASHR,1,BLN,PRI,BAS,?1,S,ROV,1,ARR,OFF,,,\*7D

```
$PASHR,2,BLN,SAM,BAS,2,M,ROV,1,ARR,ON,26.004,0.000,0.000,0.001,1*
4A
$PASHR,3,BLN,SAM,BAS,?1,S,ROV,1,ARR,OFF,,,*21
```

## BRD: Reading RTK Bridge Status

---

**Function** This command allows you to list the current settings of the RTK Bridge function.

**Command Format Syntax**  
**\$PASHQ,BRD[\*cc]**

**Response format Syntax**  
**\$PASHR,BRD,s1,c2[,c3]\*cc**

### Parameters

Parameter	Description	Range
s1	Availability of RTK corrections on the specified output port: <ul style="list-style-type: none"> <li>• OFF: No RTK corrections forwarded to the output port.</li> <li>• ON: RTK corrections forwarded to the output port.</li> </ul>	ON, OFF
c2	Input port used	A, B, D, C, P, Q
c3	Output port used	A, B, D, C
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,BRD
$PASHR,BRD,ON,P,A*1C
```

**Relevant Set Command** \$PASHS,BRD

## BRV: Reading Relative Mode Status

---

**Function** This command is used to read the current status of the Relative mode.

**Command Format Syntax**  
**\$PASHQ,BRV[\*cc]**

**Response Format Syntax**

```
$PASHR,BRV,s1,d2[,d3[,d4]]*cc
```

**Parameters**

Parameter	Description	Range
s1	Relative mode status: • ON: Relative mode currently on. • OFF: Relative mode currently off	ON, OFF
d2	Index of first baseline used	
d3	Index of second baseline used	
d4	Index of third baseline used	
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,BRV
$PASHR,BRV,OFF*51
```

**CAN: CAN Bus Status & Parameters**

---

**Function** This command allows you to read the current status of the CAN bus as well as its parameters.

**Command Format Syntax**

```
$PASHQ,CAN[*cc]
```

**Response Format Syntax**

```
$PASHR,CAN,c1,s2,BRT=d3,FMT=d4*cc
```



## Parameters

Parameter	Description	Range
c1	CAN port identification	L
s2	CAN status: ON or OFF	ON, OFF
BRT=d3	Bit rate (in kbits/second): <ul style="list-style-type: none"> <li>• 0: 62.5</li> <li>• 1: 125</li> <li>• 2: 250 (default)</li> <li>• 3: 500</li> <li>• 4: 1000</li> </ul>	0-4
FMT=d4	Data format: <ul style="list-style-type: none"> <li>• 0: Raw data</li> <li>• 1: NMEA 2000 data</li> </ul>	0, 1
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,CAN
$PASHR,CAN,L,ON,BRT=4*47
```

**Relevant Set Command**    \$PASHS,CAN

## COO,REF: Base Position

---

**Function**    This command is used to read the base position entered in a rover using the \$PASHS,COO,REF command.

**Command Format Syntax**  
**\$PASHQ,COO,REF[\*cc]**

**Response Format Syntax**  
**\$PASHR,COO,REF,f1,f2,f3,d4\*cc**

## Parameters

Parameter	Description	Range
f1	X component of Cartesian base position, in meters	
f2	Y component of Cartesian base position, in meters	
f3	Z component of Cartesian base position, in meters	
d4	Preference rule currently used: <ul style="list-style-type: none"> <li>• 0: Incoming position is preferred (default)</li> <li>• 1: Always ignore incoming position</li> </ul>	0, 1
*cc	Checksum	*00-*FF

**Example****\$PASHQ,COO,REF**

\$PASHR,POS,REF,2352345.2800,2717465.7080,5251459.2240,0\*61

**Related Commands** \$PASHS,COO,REF (set command)**CST: NTRIP Caster Parameters**

---

**Function** This command is used to query the receiver for the current NTRIP caster settings.**Command Format Syntax**  
\$PASHQ,CST[\*cc]**Response Format Syntax**  
\$PASHR,CST,s1,s2,d3,s4,d5,s6,s7,s8,f9,f10,s11,d12,s13,s14,c15,s16,s17,s18[\*cc]**Parameters**

Parameter	Description	Range
s1	NTRIP caster status	ON, OFF
s2	IP address of the NTRIP caster.	100 characters max.
d3	IP port number of the NTRIP caster	100-65535
s4	NTRIP caster password. This password is used by NTRIP servers (data sources) to connect to the NTRIP caster.	32 characters max.
s6	NTRIP caster identifier. Use this field to provide more information describing/identifying the NTRIP caster.	100 characters max.
s7	NTRIP caster operator: Name of the institution, agency or company running the caster.	100 characters max.
s8	Country code	3 characters
f9	Latitude in degrees.	±90.00
f10	Longitude in degrees.	0.00 to 359.99
s11	Fallback caster IP address. (Fallback caster: the caster where to connect to in case this one breaks down).	128 characters max
d12	Fallback caster IP port number	100-65535
s13	Network identifier, e.g. name of a network of GNSS permanent stations.	100 characters max
s14	Network operator: Name of the institution, agency or company running the network.	100 characters max

Parameter	Description	Range
c15	Fee indicator: • Y: Usage is charged • N: No user fee	Y, N
s16	Web address where network information can be found.	100 characters max
s17	Web address where data stream information can be found.	100 characters max
s18	Web or email address where registration information can be found.	100 characters max
*cc	Optional checksum	*00-*FF

**Example****\$PASHQ,CST**

\$PASHS,CST,ON,124.61.55.10,2102,MB2\_NTRIP Caster,  
Trimble,FRA,47.10,-1.00,123.12.132.12,2101,My Network,Trimble,  
Y,www.trimble.com, www.trimble.com, MB2@trimble.com\*20

**See also** \$PASHS,CST  
\$PASHS,CST,USR,ADD  
\$PASHS,CST,USR,DEL

**CST,USR: Listing NTRIP Caster Users**

---

**Function** This command is used to query the receiver for the complete list of users currently using the embedded NTRIP caster, or for the parameters of the specified user.

**Command Format Syntax**

\$PASHQ,CST,USR[,s][\*cc]

**Parameters**

Parameter	Description	Range
s	Name of the user for which you want the receiver to return its parameters.	32 characters max
*cc	Optional checksum	*00-*FF

**Response Format Syntax**

\$PASHR,CST,USR,d1,d2,s3,s4,s5,,s7]...[s16]\*cc

**Parameters**

The receiver returns one response line per user name.

Parameter	Description	Range
d1	Number of user names part of the list	0-100
d2	Index of this user name within the list	0-100
s3	User name	32 char. max.
s4	Password	32 char. max.
s5	Available mount points for this user: <ul style="list-style-type: none"> <li>• ALL: all mount points are available for this user</li> <li>• SEL: Only a selection of mount points is available for this user.</li> </ul>	ALL, SEL
d6	Number of mount points.	0-10
s7.. s16	Mount point name	100 char. max.
*cc	Optional checksum	*00-*FF

### Example

Querying the list of users

#### **\$PASHQ,CST,USR**

```
$PASHR,CST,USR,5,0,user1,password1,SEL,2,NAN1,NAN2
```

```
$PASHR,CST,USR,5,1,user2,password2,ALL
```

```
$PASHR,CST,USR,5,2,user3,password3,SEL,1,NAN1
```

```
$PASHR,CST,USR,5,3,user4,password4,SEL,3,NAN1,NAN2,NAN3
```

```
$PASHR,CST,USR,5,4,user5,password5,SEL,1,NAN3
```

**See also** \$PASHS,CST  
\$PASHS,CST,USR,ADD  
\$PASHS,CST,USR,DEL

## DDM: Differential Decoder Message

---

**Function** This command returns a report about the decoded differential messages.

The report may refer to several types of differential messages which the built-in Differential Decoder gets from different receiver ports.

**Command Format** **Syntax**  
\$PASHQ,DDM[\*cc]

### Parameters

None.

## Response Format Syntax

\$PASHR,DDM,c1,s2,s3,d4,s5,f6,f7,s8\*cc

### Parameters

Parameter	Description	Range
c1	Port receiving corrections (may be empty, see below)	A-D, F, I, P, Q
s2	Message transport	RT2, RT3, CMR, CMX or ATM
s3	Message number/identifier	e.g. 1004 for RT3, RNX for ATM, etc.
d4	Counter of decoded messages	0-9999
s5	Base ID (empty if Base ID not available)	
f6	Time tag, in seconds, as read from the decoded message	
f7	Age of corrections, in seconds	
s8	Attribute	60 characters max.
*cc	Checksum	*00-*FF

### Comments

- Parameter d4 counts the number of decoded messages modulo 10000. It is incremented by one each time a new message is decoded.

Each receiver port has its own counter. All counters are initialized to 0 after the \$PASHS,RST or \$PASHS,INI command has been issued. Counters are very useful to check if some DDM messages are lost at the output.

- The f5 field may be empty if the base ID is not available for a given message.
- Parameter f6 is the time tag, in seconds, as read from the decoded message (independently of what \$PASHS,PGS is requesting). Regardless of its original presentation, the time tag provided in the DDM message is always expressed in GPS time, within the GPS week.

For example the time tag from message RTCM-3 MT 1012 (referring to GLONASS time) will be transformed to a GPS time tag in the DDM message.

Another example is the time tag from message RTCM-2 MT 18. It is originally presented modulo 3600 seconds, but appears as a complete time tag in the DDM message.

If the decoded message does not contain any time tag (e.g. RT3 1005), the f6 field is empty.

- The age of corrections is defined as the difference, in seconds, between the receiver time at the end of the decoding process and the time tag read from the decoded message. If no tag time is provided in the message (e.g. RT3 1005), the f7 field is empty.
- Parameter s8 holds some vital attributes from the decoded message. It contains a number of parameters delimited by “slash” characters (/). It may also be empty.
- Although controlled by the \$PASHS,NME,DDM command, the DDM message is independent of the NMEA period. It is output every time a new differential message is decoded.
- CMR stands for both CMR and CMR+. CMX stands for both CMRx and sCMRx.

**Example**

```
$PASHQ,DDM
$PASHR,DDM,D,RT3,1004,5,4095,1,*49
```

**Related Commands**

```
$PASHS,NME
```

**DDN: DynDNS Parameters**

---

**Function**

This command is used to query the receiver for the current DynDNS settings.

**Command Format****Syntax**

```
$PASHQ,DDN[*cc]
```

**Response Format****Syntax**

```
$PASHR,DDN,DYN=d1,SYS=s2,USR=s3,PWD=s4,HNM=s5,PER=d6*cc
```

**Parameters**

Parameter	Description	Range
DYN=d1	Current DynDNS service status: <ul style="list-style-type: none"> <li>• d1=0: Enabled</li> <li>• d1=1: Disabled</li> </ul>	0, 1
SYS=s2	Address of the free service used.	100 characters max.
USR=s3	Username chosen when creating an account on the DynDNS web site.	32 characters max.
PWD=s4	Password chosen when creating an account on the DynDNS web site.	32 characters max.

Parameter	Description	Range
HNM=s5	Hostname declared on the DynDNS web site for the receiver.	100 characters max.
PER=d6	Update rate, in seconds.	60-3600
*cc	Optional checksum	*00-*FF

### Example

**\$PASHQ,DDN**

\$PASHR,DDN,DYN=1,SYS=dyn dns@dyn dns.org,USR=psmith,PWD=ashtech  
h,HNM=ashtech1.dyn dns.org,PER=600\*62

**See also** \$PASHS,DDN,PAR

## DDS: Differential Decoder Status

---

**Function** This command allows you to output a message providing status data on the corrections received.

**Command Format** **Syntax**  
\$PASHQ,DDS[\*cc]

**Response Format** **Syntax**  
\$PASHR,DDS,d1,m2,d3,c4,s5,c6,d7,d8,d9,d10,d11,f12,f13,d14,n(d15,  
f16,f17)\*cc

### Parameters

Parameter	Description	Range
d1	Differential decoder number. "1" corresponds to first decoder, etc. An empty field means the decoder used is not known.	1-3
m2	GNSS (output) time tag	000000.00-235959.99
d3	Number of decoded messages since last stream change (see table below)	0-127
c4	ID of port from which corrections are received (may be empty, see below)	A-D, F, I, P, Q
s5	Protocol detected (empty means "no data")	RT2, RT3, CMR, ATM, CMX
d6	Time window, in seconds: • "0" if not defined or just initialized • "200" means equal to or greater than 200	0-200

Parameter	Description	Range
d7	Percentage of estimated overall data link quality/availability. Empty if not defined.	0-100
d8	Percentage of deselected information. Empty if not defined.	0-100
d9	CRC percentage. Empty if not defined.	0-100
d10	Standard of latency, in milliseconds	0-16383
d11	Mean latency, in milliseconds	0-16383
f12	Mean epoch interval, in seconds	0.00-3600
f13	Min epoch interval, in seconds	0.00-3600
d14	Number (n) of different messages detected since last stream change	0-63
d15	Message type	RT2: 1-63 RT3: 1001-4094 CMR: 0(obs), 1(loc), 2(desc), 3(glo), 12(cmr+), 20 (glo encrypted) ATM: 0-15 CMX: no message reported
f16	Interval of last message, in seconds	0.000-1023.000
f17	Age of last message, in seconds	0.000-1023.000
*cc	Checksum	

CMR stands for both CMR and CMR+. CMX stands for both CMRx and sCMRx.

### Ranges of Counters

Reported indicator (d3 field)	Counter Range
0-23	0-23
24-47	24-71
48-71	72-167
72-95	168-359
96-119	360-743
120-126	744-935
127	936 or greater

### Example

**\$PASHQ,DDS**

\$PASHR,DDS,1,140235.33,A,RT3,200,100,0,100,5,50,1.05,1.00,3,1004,1.00,0,0.500,1005,30.000,18.000,1006,30.000,18.000\*49

### Related Commands

**\$PASHS,NME**



## DFC: Reading the Content of the Default Configuration File

---

**Function** This command allows you to read the content of the default configuration file (*default\_config.cmd*).

**Command Format**    **Syntax**  
                           \$PASHQ,DFC[\*cc]

**Response Format**   **Syntax**  
 See the two examples below.  
 If a default configuration file exists:

```
$PASHQ,DFC
$PASHR,DFC,BEGIN*5E
# Default configuration for my company – Created on June 27, 2016
$PASHS,ELM,10
$PASHS,NME,GGA,A,ON,1
$PASHS,NME,GSV,A,ON,5
$PASHR,DFC,END*56
```

If there is no default configuration file:

```
$PASHQ,DFC
$PASHR,DFC,BEGIN*5E
$PASHR,DFC,END*56
```

**See Also**    \$PASHS,RST  
                   \$PASHS,INI  
                   \$PASHS,DFC,DEL  
                   \$PASHQ,DFC  
                   \$PASHS,DFC,GET  
                   \$PASHS,DFC,TST

## DIF,BDS: BeiDou Correcting Data

---

**Function** This command allows you to know whether BeiDou correcting data are used in the receiver or not.

**Command Format**   **Syntax**  
                           \$PASHQ,DIF,BDS[\*cc]

**Response Format Syntax**

\$PASHR,DIF,BDS,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Status: • OFF: Not used in position computation • AUT: Used in position computation, but only in China.	OFF, AUT
*cc	Checksum	

**Example**

```
$PASHQ,DIF,BDS
$PASHR,DIF,BDS,AUT*2A
```

**DIF,NET: Source of Correction Data Currently Used**

---

**Function** This command tells which incoming correction data are currently used in an RTK rover when several data sources exist.

**Command Format Syntax**

\$PASHQ,DIF,NET[\*cc]

**Response Format Syntax**

\$PASHR,DIF,NET,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Corrections currently used: • IP: Are those delivered through an IP connection • UHF: Are those delivered through a UHF radio link	IP, UHF
*cc	Checksum	

**Example**

```
$PASHQ,DIF,NET
$PASHR,DIF,NET,IP*79
```

## DIF,SBA: Source of SBAS Corrections

---

**Function** This command allows you to know which source of SBAS corrections is currently used to calculate the position.

**Command Format Syntax**  
**\$PASHQ,DIF,SBA[\*cc]**

**Response Format Syntax**  
**\$PASHR,DIF,SBA,d1,d2,s3,s4,n(d5,c6,c7)\*cc**

### Parameters

Parameter	Description	Range
d1	Source of SBAS corrections: <ul style="list-style-type: none"> <li>• ALL: from all possible sources</li> <li>• OFF: All SBAS corrections rejected from position processing</li> </ul>	1-99
d2	Message number	1-d1
s3	SBAS provider name: <ul style="list-style-type: none"> <li>• WAA: WAAS</li> <li>• EGN: EGNOS</li> <li>• MSA: MSAS</li> <li>• GAG: GAGAN</li> <li>• SDC: SDCM (Luch)</li> <li>• UKN: Unknown. The unknown provider message is always the last one, with the corresponding s4 field always set to "ON". The provider can be reported unknown in transient cases when a PRN (in the range 120-193) is detected but cannot be assigned to a known provider yet.</li> </ul>	WAA,EGN,MSA,GAG,SDC,QZS,UKN
s4	SBAS provider status: <ul style="list-style-type: none"> <li>• ON: Enabled to be used as a source of differential data</li> <li>• OFF: Disabled</li> </ul>	ON, OFF
d5	SBAS satellite ID. Satellite IDs of a given provider are all reported regardless of its status value (see s4).	120-193
c6	Tracking status: <ul style="list-style-type: none"> <li>• T: Tracked</li> <li>• -: Not tracked</li> </ul>	-, T
c7	Usage status: <ul style="list-style-type: none"> <li>• S: Used</li> <li>• -: Not used</li> </ul>	-, S
*cc	Checksum	

**Example** **\$PASHQ,DIF,SBA**  
**\$PASHR,DIF,SBA,7,1,WAA,ON,133,-,135,-,138,-,\*03**

```

$PASHR,DIF,SBA,7,2,EGN,ON,120,T,-,123,T,-,124,-,-,126,-,-,136,T,S*13
$PASHR,DIF,SBA,7,3,MSA,ON,129,-,-,137,-,-*16
$PASHR,DIF,SBA,7,4,GAG,ON,127,-,-,128,-,-*0F
$PASHR,DIF,SBA,7,7,SDC,ON,125,T,-,140,-,-,141,-,-*76
$PASHR,DIF,SBA,7,7,QZS,ON,183,-,-*0D
$PASHR,DIF,SBA,7,7,UKN,ON*12

```

## DIP: Direct IP Parameters

---

**Function** This command is used to query the parameters used for a Direct IP connection. When c6 is omitted in the query command, the returned Direct IP settings are those for the port defined through the \$PASHS,DIP,PAR or \$PASHS,DIP command last run.

**Command Format Syntax**  
**\$PASHQ,DIP[,c6][\*cc]**

**Response Format Syntax**  
**\$PASHR,DIP,RIP,s1,PRT,d2[,LGN,s3,PWD,s4],IPP,c6,STS,d7,ACN,d8\*cc**

### Parameters

Parameter	Description	Range
RIP,s1	IP address (xxx.xxx.xxx.xxx) or host name	32 char. max.
PRT,d2	Port number	0-65535
LGN,s3	User name (optional)	32 char. max.
PWD,s4	Password (optional)	32 chars max.
IPP,c6	Internet port used on the receiver to establish the connection with the base (server): <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul>	P, Q
STS,d7	Status: <ul style="list-style-type: none"> <li>• 0: Not connected</li> <li>• 1: Connected</li> </ul>	0, 1
ACN,d8	Auto-connection: <ul style="list-style-type: none"> <li>• 0: After a power cycle, there won't be any direct IP connection (default)</li> <li>• 1: After a power cycle, the direct IP connection will be restored</li> </ul>	0, 1
*cc	Checksum	*00-*FF

### Examples

```

$PASHQ,DIP
$PASHR,DIP,RIP,www.trimble.com,PRT,8080,IPP,Q,STS,0,ACN,0*4B

```

**See also** \$PASHS,DIP,PAR (set command)

## DRD: Data Recording Duration

---

**Function** This command returns the duration that was last set for all the G-files that the receiver will be recording.

**Command Format Syntax**  
\$PASHQ,DRD[\*cc]

**Parameters**

None.

**Response Format Syntax**  
\$PASHR,DRD,d1\*cc

**Parameters**

Parameter	Description	Range
d1	Duration of data recording held in one G-file, in seconds	15-1440
*cc	Checksum	*00-*FF

**Example** \$PASHQ,DRD  
\$PASHR,DRD,60\*0C

**Relevant Set Command** \$PASHS,DRD

## DRI: Raw Data Recording Rate

---

**Function** This command returns the current value of recording rate for raw data.

The output rate of ATOM RNX and PVT messages are returned:

- On port M if the internal memory is selected
- On port S or U if the USB memory is selected

**Command Format Syntax**  
\$PASHQ,DRI[\*cc]

**Response Format Syntax**

\$PASHR,DRI,f1\*cc

**Parameters**

Parameter	Description	Range
f1	Current raw data recording rate	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1-6, 10, 12, 15, 20, 30, 60, 120, 180, 240, 300, 360, 600, 720, 900, 1200, 1800, 3600 seconds
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,DRI
$PASHR,DRI,1.00*18
```

**Related Commands** \$PASHS,DRI (set command)  
 \$PASHQ,REC  
 \$PASHS,MEM

**DSY: Daisy Chain Status**

---

**Function** This command queries the receiver for the status of the daisy chain function.

**Command Format Syntax**

\$PASHQ,DSY[\*cc]

**Parameters**

None.

**Response Format Syntax**

```
$PASHR,DSY,OFF*59
or
$PASHR,DSY,c1,c2*cc
```

## Parameters

Parameter	Description	Range
c1	Source port: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A-D, F, I, P, Q
c2	Destination port: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A-D, F, I, P, Q
*cc	Checksum	*00.*FF

## Example

Command reporting data on port A forwarded to port C:

```
$PASHQ,DSY
$PASHR,DSY,A,C*38
```

**Relevant Set Command**    \$PASHS,DSY

## DTM: Datum Reference

---

**Function**    This command asks the receiver to output the content of the NMEA DTM message.

**Command Format Syntax**  
 \$PASHQ,DTM[\*cc]

### Parameters

None.

**Response Format Syntax**  
 \$GPDTM,s1,,f2,c3,f4,c5,f6,s7\*cc

## Parameters

Parameter	Description	Range
s1	Local datum code: <ul style="list-style-type: none"> <li>W84: WGS84 used as local datum</li> <li>999: Local datum computed using the parameters provided by the RTCM3.1 data stream.</li> </ul>	W84, 999
f2	Latitude offset, in meters	0-59.999999
c3	Direction of latitude	N, S
f4	Longitude offset, in meters	0-59.999999
c5	Direction of longitude	E, W
f6	Altitude offset, in meters	±0-99.999
s7	Reference datum code	W84
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,DTM
```

```
$GPDTM,999,2.324525,N,1.499476,W,1.365,W84*37
```

**See Also** \$PASHS,NME

## Automatic Output of DTM Messages

This is a reminder on how to output DTM messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,DTM,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output DTM messages on port A at a rate of 2 seconds:

```
$PASHS,NME,DTM,A,ON,2
```

## DTH: Dithering Mode Status

---

**Function** This command is used to read the dithering mode status.

**Command Format** **Syntax**

```
$PASHQ,DTH[*cc]
```



**Response Format Syntax**

\$PASHR,DTH,d1\*cc

**Parameters**

Parameter	Description	Range
d1	Dithering mode currently enabled: <ul style="list-style-type: none"> <li>• 0: Up to RTK fixed (the most accurate)</li> <li>• 1: 7 cm horizontal and 7 cm vertical</li> <li>• 2: 7 cm horizontal and 2 cm vertical</li> <li>• 3: 30 cm horizontal and 30 cm vertical</li> </ul>	0-3
*cc	Checksum	

**Example**

```
$PASHQ,DTH
$PASHR,DTH,2*18
```

**DYN: Receiver Dynamics**

---

**Function** This command allows you to query the current setting for the receiver dynamics.

**Command Format Syntax**

\$PASHQ,DYN[\*cc]

**Response Format Syntax**

\$PASHR,DYN,d\*cc

**Parameters**

Parameter	Description	Range
d	Receiver dynamics: <ul style="list-style-type: none"> <li>• 1: Static</li> <li>• 2: Quasi-static</li> <li>• 3: Walking</li> <li>• 4: Ship</li> <li>• 5: Automobile</li> <li>• 6: Aircraft</li> <li>• 7: Unlimited</li> <li>• 8: Adaptive</li> <li>• 9: User-defined</li> </ul>	1-9
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,DYN
$PASHR,DYN,8*33
```

**Related Commands**    \$PASHS,DYN (set command)  
                               \$PASHS,UDP

## ELM: Elevation Mask

---

**Function**            This command is used to read the current value of the elevation mask. The elevation mask impacts data recording, data output and satellite reception at the base. The same value of elevation mask is used for all sensors on the board.

**Command Format**    **Syntax**  
                               \$PASHQ,ELM[\*cc]

**Response Format**    **Syntax**  
                               \$PASHR,ELM,d1\*cc

### Parameters

Parameter	Description	Range
d1	Current value of elevation mask, in degrees	0-90
*cc	Checksum	*00-*FF

**Example**             \$PASHQ,ELM  
                               \$PASHR,ELM,5\*29

**Related Commands**    \$PASHS,ELM  
                               \$PASHQ,PEM

## ETD: Reading Ethernet Driver Parameters

---

**Function**            This command queries the board for its current values of Ethernet driver parameters.

**Command Format**    **Syntax**  
                               \$PASHQ,ETD[\*cc]

**Response Format Syntax**

```
$PASHR,ETD,AUT=s1,DPL=s2,SPD=d3*cc
```

**Parameters**

Parameter	Description	Range
AUT=s1	Auto-negotiation enabled (ON) or disabled (OFF)	ON, OFF
DPL=s2	Duplex mode (full or half) (if s1=OFF)	FULL, HALF
SPD=d3	Speed in Mbits/s (if s1= OFF)	10, 100
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,ETD
```

```
$PASHR,ETD,AUT=OFF,DPL=FULL,SPD=100*02
```

**Related Commands**

```
$PASHS,ETD,PAR
```

```
$PASHS,ETH,ON
```

```
$PASHS,ETH,OFF
```

**ETH: Ethernet Status and Parameters**

---

**Function**

This command is used to read the current status of the Ethernet port as well as all the parameters relevant to this port.

**Command Format Syntax**

```
$PASHQ,ETH[*cc]
```

**Response Format Syntax**

```
$PASHR,ETH,c1,s2,s3,s4,DHP=d5,ADD=s6,MSK=s7,GTW=s8,DN1=s9,DN2=s10,ACN=d11*cc
```

## Parameters

Parameter	Description	Range
c1	Ethernet port (TCP/IP server)	1
s2	Ethernet status	OFF, ON
s3	MAC address (xx:xx:xx:xx:xx:xx)	17 characters
s4	Current IP address (=s6 when DHCP disabled)	0.0.0.0-255.255.255.255
DHP=s5	DHCP mode (0: disabled; 1: enabled)	0, 1
ADD=s6	Static IP address assigned to the receiver when DHCP is disabled	0.0.0.0-255.255.255.255
MSK=s7	Sub-network mask	0.0.0.0-255.255.255.255
GTW=s8	Gateway IP address	0.0.0.0-255.255.255.255
DN1=s9	DNS 1 IP address	0.0.0.0-255.255.255.255
DN2=s10	DNS 2 IP address	0.0.0.0-255.255.255.255
ACN=d11	(Obsolete)	
*cc	Checksum	*00-*FF

Parameters s6, s7, s8, s9, s10 are the Ethernet parameters used when the DHCP mode is disabled. In that case, s4=s6.

### Example

**\$PASHQ,ETH**

\$PASHR,ETH,1,ON,02:03:04:85:06:07,10.20.2.74,DHP=1,ADD=10.20.2.28,  
MSK=255.255.255.0,GTW=10.20.2.1,DN1=134.20.2.16,DN2=134.20.2.3,AC  
N=1\*1B

### See also

\$PASHS,ETH,PAR

\$PASHS,ETH

## FIL,CUR: Information On G-File Being Recorded

---

### Function

This command allows you to read information about the G-file currently being recorded.

### Command Format

#### Syntax

\$PASHQ,FIL,CUR[\*cc]

### Response Format

#### Syntax

General form:

\$PASHR,FIL,CUR,s1,d2,s3,s4,d5\*cc

If no G-file recording is in progress:

\$PASHR,FIL,CUR,NONE\*79

## Parameters

Parameter	Description	Range
s1	Filename (including path)	255 characters max.
d2	Size in bytes	0-134217728
s3	Date (ddmmyyyy)	
s4	Time (hhmmss)	000000-235959
d5	Memory location: <ul style="list-style-type: none"> <li>• 0: Internal memory.</li> <li>• 2: USB memory</li> </ul>	0, 2
*cc	Checksum	*00-*FF

## Example

\$PASHQ,FIL,CUR

\$PASHR,FIL,CUR,GazerA09.123,1769897,14032009,130850,0\*63

## Related Commands

\$PASHS,REC

\$PASHS,MEM

## FIL,LST: Listing Files in Receiver Memory or USB Key

---

### Function

This command allows you to list the names of the files stored in the receiver's internal memory or on the USB key connected to the receiver.

### Command Format

#### Syntax

\$PASHQ,FIL,LST[,d1][,s2][\*cc]

### Parameters

Parameter	Description	Range
d1	Memory type: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 2: USB memory</li> <li>• 3: Internal memory (log file partition)</li> <li>• d1 omitted: Memory is as defined with \$PASHS,MEM</li> </ul>	0, 2, 3
s2	Path name	
*cc	Optional checksum	*00-*FF

### Response format

#### Syntax

\$PASHR,FIL,LST,d1,d2,s3,d4,s5,s6[,c7]\*cc

## Parameters

Parameter	Description	Range
d1	Number of files	
d2	File index	
s3	File name or directory name	255 characters max.
d4	Size in bytes	0-134217728
s5	Date (ddmmyyyy)	
s6	Time (hhmmss)	000000-235959
c7	=D when s3 is a directory name	D
*cc	Optional checksum	*00-*FF

## Example

### \$PASHQ,FIL,LST\*53

\$PASHR,FIL,LST,4,0,GazerA14.123,1769897,14032014,130850\*74

\$PASHR,FIL,LST,4,1,GazerB14.123,1769876,10032014,110952\*7C

\$PASHR,FIL,LST,4,2,GazerC14.123,1769787,01032014,181856\*72

\$PASHR,FIL,LST,4,3,GazerD14.123,1769787,01032014,181856\*74

**Related Commands**    \$PASHS,REC  
                               \$PASHS,MEM  
                               \$PASHQ,FIL,STS

## FIL,STS: Memory Status

---

**Function**    This command is used to read the current status of the receiver memory (SD card or internal memory).

**Command Format**    **Syntax**  
                               \$PASHQ,FIL,STS[d1][\*cc]

## Parameters

Parameter	Description	Range
d1	Queried memory: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 2: USB memory</li> <li>• 3: Internal memory (log file partition)</li> </ul>	0, 2, 3
*cc	Optional checksum	*00-*FF

**Response Format**    **Syntax**  
                               \$PASHR,FIL,STS,d1,d2,d3\*cc

## Parameters

Parameter	Description	Range
d1	Queried memory: <ul style="list-style-type: none"> <li>• 0: Internal memory (user data partition)</li> <li>• 2: USB memory</li> <li>• 3: Internal memory (log file partition)</li> </ul>	0, 2, 3
d2	Total memory size, in bytes. 0 if memory not available	
d3	Free memory size, in bytes. 0 if memory not available	
*cc	Checksum	*00-*FF

## Examples

**Related Commands**    \$PASHS,REC  
                               \$PASHS,MEM  
                               \$PASHQ,FIL,LST

## FTP: FTP Status and Settings

---

**Function**            This command is used to query the status and settings of the FTP server used to upload files from the receiver.

**Command Format**    **Syntax**  
                               \$PASHQ,FTP[\*cc]

### Parameters

None.

**Response format**    **Syntax**  
                               \$PASHR,FTP,s1,d2,d3,s4,d5,d6,ADD=s7,PRT=d8,LGN=s9,PWD=s10,  
                               PTH=s11\*cc

### Parameters

Parameter	Description	Range
s1	File transfer status: <ul style="list-style-type: none"> <li>• NONE: no transfer to FTP</li> <li>• PUT: File being uploaded to FTP</li> </ul>	NONE, PUT
d2	Number of files to be transferred	0-255
d3	Number of files already transferred	0-255
s4	Name of the file being transferred	255 characters max.

Parameter	Description	Range
d5	Size, in bytes, of the file being transferred	0-134217728
d6	Percentage of data already transferred for the file transfer currently in progress.	0-100
ADD=s7	FTP server IP address or host name	
PRT=d8	FTP server port number	0-65535
LGN=s9	FTP server login	32 characters max.
PWD=s10	FTP server password	32 characters max.
PTH=s11	Path used on the FTP server	255 characters max.
*cc	Checksum	*00-*FF

### Example

**\$PASHQ,FTP\*35**

\$PASHR,FTP,PUT,10,3,GabcdA9.145,1769897,56,ADD=ftp.ashtech.com,  
PRT=21,LGN=Ashtech,PWD=u6huz8,PTH=/my folder\*11

**See Also** \$PASHS,FTP,PAR  
\$PASHS,FTP,PUT

## GAL: GALILEO Tracking Status

---

**Function** This command queries the receiver for the current GALILEO tracking status.

**Command Format** **Syntax**  
\$PASHQ,GAL[\*cc]

**Response Format** **Syntax**  
\$PASHR,GAL,s1,USE,n(c2)\*cc

### Parameters

Parameter	Description	Range
s1	Galileo tracking status: • ON: Currently tracking and using Galileo satellites • OFF: Not tracking Galileo satellites	ON, OFF
n(c2)	Provides tracking status of each Galileo satellite: • Y: Tracking enabled for this satellite • N: Tracking disabled for this satellite  This status is provided n times, where n is the number of satellites in the Galileo constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Checksum	



**Example**      \$PASHQ,GAL  
 \$PASHR,GAL,ON,USE,YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY\*50

**Related Commands**    \$PASHS,GAL (set command)

## GBS: GNSS Satellite Fault Detection

---

**Function**      This command returns the GBS message supporting RAIM (Receiver Autonomous Integrity Monitoring) on the port routing the query command if no output port is specified.

**Command Format**    **Syntax**  
 \$PASHQ,GBS[\*cc]

**Parameters**

None.

**Response Format**    The message header is in the form:

\$--GBS

If only GPS satellites are used in the position solution, the message header is:

\$GPGBS

If only GLONASS satellites are used in the position solution, the message header is:

\$GLGBS

If only BeiDou satellites are used in the position solution, the message header is:

\$GBGBS

If several types of satellites are used in the position solution, the message header is:

\$GNGBS

**Syntax**

\$--GBS,m1,f2,f3,f4,d5,f6,f7,f8,h9,h10\*cc

## Parameters

Parameter	Description	Range
m1	UTC time of the GGA or GNS fix associated with this message (hhmmss.ss)	000000.00-235959.99
f2	Expected error in latitude, in meters, due to bias, with noise= 0	0.0-99.9
f3	Expected error in longitude, in meters, due to bias, with noise= 0	0.0-99.9
f4	Expected error in altitude, in meters, due to bias, with noise= 0	0.0-99.9
d5	ID number of most likely failed satellite	See table below
f6	Probability of missed detection for most likely failed satellite	0.00-1.00
f7	Estimate of bias, in meters, on most likely failed satellite	0.0-99.9
f8	Standard deviation of bias estimate	0.0-99.9
h9	GNSS system ID	0-F; See table below
h10	GNSS signal ID	0-F; See table below
*cc	Checksum	*00-*FF

System	System ID	Satellite ID	Signal ID	Signal Channel
GPS	1 (GP)	1-99 1-32 reserved for GPS 33-64 reserved for SBAS (65-99 undefined)	0	All signals L1
			1	C/A L1 P(Y)
			2	L1 M
			3	L2 P(Y) L2C-
			4	M L2C-L L5-I L5-I
			5	L5-Q
			6	Reserved
			7	
			8	
			9-F	
GLONASS	2 (GL)	33-99 33-64 reserved for SBAS 65-99 reserved for GLONASS	0	All signals G1
			1	C/A G1 P G2
			2	C/A
			3	GLONASS (M) G2 P
			4	Reserved
			5-F	
GALILEO	3 (GA)	1-99 1-36 reserved for Galileo 37-64 reserved for Galileo SBAS (65-99 undefined)	0	All signals E5a
			1	E5b
			2	E5 a+b E6-A
			3	E6-BC L1-A
			4	L1-BC
			5	Reserved
			6	
			7	
			8-F	

System	System ID	Satellite ID	Signal ID	Signal Channel
IRNSS	4 (GI)	1-99 1-15 reserved for IRNSS (15-32 undefined) 33-64 reserved for SBAS (65-99 undefined)	0	All signals
			1	L5-SPS
			2	S-SPS
			3	L5-RS
			4	S-RS
			5	L1-SPS
			6-F	Reserved
BEIDOU	5 (GB)	1-99 1-40 reserved for SBS (41-99 unidentified)	0	All signals
			1	B1I
			2	B2a
			3	B2I
			4	Reserved
			5-F	Reserved
QZSS	6 (GQ)	1-99 1-9 reserved for QZSS* 55-63 is reserved for QZSS SBAS* (64-99 undefined)	0	All signals L1
			1	L1 C/A
			2	L1C (D) L1C (P)
			3	L2C-L L5-I
			4	L5-Q
			5	L6D L6E
			6-F	Reserved
Reserved	7 to F			

\* Satellite ID is obtained by taking the 6 Low Significant Digits of the 8-bit PRN Number (e.g. Satellite ID of PRN 193 is 1).

## Automatic Output of GBS Messages

This is a reminder on how to output GBS messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GBS,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GBS messages on port A at a rate of 120 seconds:

```
$PASHS,NME,GBS,A,ON,120
```

## GEM: Geoid Model

**Function** This command is used to read the name of the geoid model currently used.

**Command Format** **Syntax**  
\$PASHQ,GEM[\*cc]

**Response Format Syntax**

\$PASHR,GEM,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Geoid model currently used: • STANAG: STANAG 4294 geoid model • EGM96: EGM 96 geoid model	STANAG, EGM96
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,GEM
$PASHR,GEM,STANAG*1A
```

**Related Commands** \$PASHS,GEM (set command)

**GGA: GNSS Position Message**

---

**Function** This command is used to output a GGA message containing the last computed position. If no position is computed, the message will be output anyway, but with some blank fields.

**Command Format Syntax**

\$PASHQ,GGA[\*cc]

**Response Format Syntax**

\$GPGGA,m1,m2,c3,m4,c5,d6,d7,f8,f9,M,f10,M,f11,d12\*cc

**Parameters**

Parameter	Description	Range
m1	Current UTC time of position (hhmmss.ss)	000000.00-235959.99
m2	Latitude of position (ddmm.mmmmmm)	0-90 0-59.999999
c3	Direction of latitude	N, S
m4	Longitude of position (dddmm.mmmmmm)	0-180 0-59.999999
c5	Direction of longitude	E,W

Parameter	Description	Range
d6	Position type: <ul style="list-style-type: none"> <li>• 0: Position not available or invalid</li> <li>• 1: Autonomous position</li> <li>• 2: RTCM Differential (or SBAS Differential)</li> <li>• 3: Not used</li> <li>• 4: RTK fixed</li> <li>• 5: RTK float</li> <li>• 9: SBAS Differential.</li> </ul>	0-5, 9
d7	Number of GNSS Satellites being used in the position computation	3-26
f8	HDOP	0-99.9
f9,M	Altitude, in meters, above mean seal level. "M" for meters	± 99999.999,M
f10,M	Geoidal separation in meters. "M" for meters. Based on the official NATO's standard mean-sea-level algorithm (5-degree grid of height).	± 999.999,M
f11	Age of differential corrections, in seconds	0-999
d12	Base station ID (RTCM only)	0-4095
*cc	Checksum	*00.*FF

## Example

**\$PASHQ,GGA**

```
$GPGGA,131745.00,4717.960847,N,00130.499476,W,4,10,0.8,35.655,M,
47.290,M,3.0,1000*61
```

## Related Commands

**\$PASHS,NME**

### \$PASHS,NPT Automatic Output of GGA Messages

This is a reminder on how to output GGA messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GGA,<port_ID>,<ON>,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GGA messages on port A at a rate of 0.5 second:

```
$PASHS,NME,GGA,A,ON,0.5
```

## GGKx: GNSS Position Message

---

**Function** This command is used to output a GGKx message containing the last computed position. If no position is computed, the message will be output anyway, but with some blank fields.

**Command Format Syntax**  
`$PASHQ,GGKX[*cc]`

**Response Format Syntax**  
`$PTNL,GGKx,m1,m2,m3,c4,m5,c6,d7,d8,f9,f10,M,d11,f12,f13,f14,f15*cc`

### Parameters

Parameter	Description	Range
m1	Current UTC time of position (hhmmss.ss)	000000.00-235959.99
m2	UTC date of position (mmdyy)	010101-123199
m3	Latitude of position (ddmm.mmmmm)	0-90 0-59.999999
c4	Direction of latitude	N, S
m5	Longitude of position (dddmm.mmmmm)	0-180 0-59.999999
c6	Direction of longitude	E,W
d7	Position type: <ul style="list-style-type: none"> <li>• 0: Position not available or invalid</li> <li>• 1: Autonomous GPS fix</li> <li>• 2: RTK float solution or RTK location status</li> <li>• 3: RTK fix solution</li> <li>• 4: Differential, code phase only solution</li> <li>• 5: SBAS solution</li> <li>• 6: 3D network solution for RTK float or RTK location</li> <li>• 7: RTK fixed 3D network solution</li> <li>• 8: 2D network solution for RTK float or RTK location</li> <li>• 9: RTK fixed 2D network solution</li> <li>• 12: RTK location</li> <li>• 13: Beacon DGPS</li> <li>• 14: RTK Global</li> </ul> (Reminder: The board DOES NOT SUPPORT OMNISTAR.)	0-14
d8	Number of GNSS Satellites being used in the position computation	3-26
f9	PDOP	0-99.9

Parameter	Description	Range
f10,M	Ellipsoid height of fix (antenna height above ellipsoid. "M" for meters.	± 99999.999,M
d11	Number of extension fields to follow.	
f12	Sigma East	0.000-999.999
f13	Sigma North	0.000-999.999
f14	Sigma Up	0.000-999.999
f15	Propagation age	
*cc	Checksum	*00-*FF

**Related Command** \$PASHS,NME

### Automatic Output of GGKX Messages

This is a reminder on how to output GGKX messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GGKX,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GGKX messages on port A at a rate of 0.5 second:

```
$PASHS,NME,GGKX,A,ON,0.5
```

## GLB: Choice of Antenna Input for L-Band Reception

---

**Function** This command reads the antenna input currently selected to receive L-band signals.

**Command Format** **Syntax**  
\$PASHQ,GLB[\*cc]

#### Parameters

None.

**Response Format** **Syntax**  
\$PASHR,GLB,c1\*cc

## Parameters

Parameter	Description	Range
c1	Antenna chosen for L-band reception: <ul style="list-style-type: none"> <li>• ON or A1: Antenna input #1 is used for L-band signal reception.</li> <li>• OFF: No L-band signal reception.</li> </ul>	ON, OFF, A1
*cc	Optional checksum	*00-*FF

## Example

```
$PASHQ,GLB
$PASHR,GLB,OFF*5E
```

## GLL: Geographic Position - Latitude/Longitude

---

**Function** This command is used to output a GLL message containing the last computed position. The message is output on the port on which the query is made. If no position is computed, the message will be output anyway, but all position-related fields will be blank.

### Command Format Syntax

```
$PASHQ,GLL[*cc]
```

### Response Format Syntax

```
$GPGLL,m1,c2,m3,c4,m5,c6,c7*cc
```

## Parameters

Parameter	Description	Range
m1	Latitude of position (ddmm.mmmmm)	0-90 0-59.999999
c2	Direction of latitude	N, S
m3	Longitude of position (dddmm.mmmmm)	0-180 0-59.999999
c4	Direction of longitude	E, W
m5	Current UTC time of position (hhmmss.ss)	000000.00- 235959.99
c6	Status <ul style="list-style-type: none"> <li>• A: Data valid</li> <li>• V: Data not valid</li> </ul>	A, V



Parameter	Description	Range
c7	Mode indicator: <ul style="list-style-type: none"> <li>• A: Autonomous mode</li> <li>• D: Differential mode</li> <li>• N: Data not valid</li> </ul>	A, D, N
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ, GLL
$GPGLL,4717.960853,N,00130.499473,W,132331.00,A,D*7D
```

**Related Commands**

```
$PASHS, NME
```

**Automatic Output of GLL Messages**

This is a reminder on how to output GLL messages at regular intervals of time: Use the \$PASHS, NME command with the syntax below:

```
$PASHS, NME, GLL, <port_ID>, ON, <Rate>
```

For more details on the \$PASHS, NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GLL messages on port A at a rate of 0.5 second:

```
$PASHS, NME, GLL, A, ON, 0.5
```

**GLO: GLONASS Tracking Status**

---

**Function**

This command is used to query the GLONASS tracking status.

**Command Format****Syntax**

```
$PASHQ, GLO[*cc]
```

**Response Format****Syntax**

```
$PASHR, GLO, s1, USE, n(c2)*cc
```

## Parameters

Parameter	Description	Range
s1	GLONASS tracking status: <ul style="list-style-type: none"> <li>• ON: Currently tracking and using GLONASS satellites</li> <li>• OFF: Not tracking GLONASS satellites</li> </ul>	ON, OFF
n(c2)	Provides tracking status of each GLONASS satellite: <ul style="list-style-type: none"> <li>• Y: Tracking enabled for this satellite</li> <li>• N: Tracking disabled for this satellite</li> </ul> This status is provided n times, where n is the number of satellites in the BeiDou constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,GLO
$PASHR,GLO,ON,USE,YYYYYYYYYYYYYYYYYYYY*5E
```

**Related Commands**    \$PASHS,GLO (set command)

## GLx: Checking Which GNSS Signals are Tracked

---

**Function**    This command is used to see which GNSS signals are currently tracked or not tracked on the different frequency bands.

**Command Format**    **Syntax**

```
$PASHQ,GL1[*cc]
$PASHQ,GL2[*cc]
```

### Parameters

None

**Response Format**    **Syntax**

```
$PASHR,GL1,s1*cc
$PASHR,GL2,s2*cc
```

## Parameters

Parameter	Description	Range
s1	L1 GNSS signals: • ON: Currently tracked • OFF: Currently not tracked	(Forced to ON)
s2	L2 GNSS signals: • ON: Currently tracked • OFF: Currently not tracked	ON, OFF
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,GL2
$PASHR,GL2,ON*60
```

**Related Commands**    \$PASHS,GLx

## GMP: GNSS Map Projection Fix Data

---

**Function**    This command is used to output a GMP message containing the last computed position. If no position is computed, the message will be output anyway, but with some blank fields.

**Command Format**    **Syntax**  
                           \$PASHQ,GMP[\*cc]

## Parameters

None.

**Response Format**    **Syntax**  
                           \$--GMP,m1,s2,s3,f4,f5,s6,d7,f8,f9,f10,f11,d12\*cc

## Parameters

Parameter	Description	Range
"\$--GMP" Header	\$GPGMP: Only GPS satellites are used. \$GLGMP: Only GLONASS satellites are used. \$GNGMP: Several constellations (GPS, SBAS, GLONASS) are used.	\$GPGMP, \$GLGMP, \$GNGMP
m1	Current UTC time of position (hhmmss.ss)	000000.00- 235959.99

Parameter	Description	Range
s2	Map projection identification RTCM3.1 - message 1024: • LOC: Local coordinate system  RTCM3.1 - message 1025, 1026 or 1027: • TM: Transverse Mercator • TMS: Transverse Mercator (West orientated) • LCC1SP: Lambert Conic Conformal (1SP) • LCC2SP: Lambert Conic Conformal (2SP) • LCCW: Lambert Conic Conformal (West orientated) • CS: Cassini-Soldner • OM: Oblique Mercator • OS: Oblique Stereographic • MC: Mercator • PS: Polar Stereographic • DS: Double Stereographic	LOC, TM, TMS, LCC1SP, LCC2SP, LCCW, CS, OM, OS, MC, PS, DS
s3	Map zone (RTCM3.1: empty)	
f4	X (Northern) component of grid (or local) coordinate, in meters	±999999999.999
f5	Y (Eastern) component of grid (or local) coordinate, in meters	±999999999.999
s6	Mode indicator: • N: No fix • A: Autonomous • D: Differential • R: Fixed RTK • F: Float RTK	N, A, D, R, F
d7	Number of GNSS Satellites being used in the position computation	3-26
f8	HDOP	0-99.9
f9	Altitude above mean seal level, or local altitude, in meters.	± 99999.999,M
f10	Geoidal separation in meters.	± 999.999,M
f11	Age of differential corrections, in seconds	0-999.9
d12	Base station ID	0-4095
*cc	Checksum	*00-*FF

**Example****\$PASHQ,GMP**

```
$GPGMP,131745.00,LOC,,45215.125,14587.298,R,11,1.5,125.221,5.214,1.5,454*xx
```

**Related Commands**

\$PASHS,NME

## Automatic Output of GMP Messages

This is a reminder on how to output GMP messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GMP,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GMP messages on port A at a rate of 0.5 second:

```
$PASHS,NME,GMP,A,ON,0.5
```

## GNS: GNSS Fix Data

---

**Function** This command allows you to output the standard NMEA GNS message.

If no computed position is available when you request the message, the message will nonetheless be output, but with all the position-related fields left blank.

**Command Format** **Syntax**

```
$PASHQ,GNS[*cc]
```

**Response Format** **Syntax**

```
$--GNS,m1,m2,c3,m4,c5,s6,d7,f8,f9,f10,f11,d12*cc
```

If the receiver is configured in GPS mode only, the message header is \$GPGNS. If it's configured in GLONASS only, the message header is \$GLGNS. If it's configured in BEIDOU only, it's \$GBGNS. With several GNSS constellations used, the message header is \$GNGNS.

### Parameters

Parameter	Description	Range
m1	Current UTC time of position (hhmmss.ss)	000000.00-235959.99
m2	Latitude of position (ddmm.mmmmmm)	0-90 0-59.999999
c3	Direction of latitude	N, S
m4	Longitude of position (dddmm.mmmmmm)	0-180 0-59.999999

Parameter	Description	Range
c5	Direction of longitude	E, W
s6	Mode indicator (1 character by constellation): <ul style="list-style-type: none"> <li>• N: No fix</li> <li>• A: Autonomous position</li> <li>• D: Differential</li> <li>• R: RTK Fixed</li> <li>• F: RTK Float</li> </ul>	N, A, D, R, F
d7	Number of GNSS satellites being used in the position computation.	3-26
f8	HDOP	0-99.9
f9	Altitude above mean sea level.	±99999.999
f10	Geoidal separation, in meters	±999.999
f11	Age of differential corrections, in s	0-999
d12	Base station ID (RTCM only)	0-4095
*cc	Checksum	

**Example****\$PASHQ,GNS**

```
$GNGNS,131745.00,4717.960847,N,00130.499476,W,RR,10,0.8,35.655,47.290,3.0,1000*61
```

**Related Commands**

\$PASHS,NME

**Automatic Output of GNS Messages**

This is a reminder on how to output GNS messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GNS,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GNS messages on port A at a rate of 10 seconds:

```
$PASHS,NME,GNS,A,ON,10
```

**GPS: GPS Tracking Status**

---

**Function**

This command is used to query the receiver for the GPS tracking status.

**Command Format Syntax**  
 \$PASHQ,GPS[\*cc]

**Response Format Syntax**  
 \$PASHR,GPS,s1,USE,n(c2)\*cc

### Parameters

Parameter	Description	Range
s1	GPS tracking status: <ul style="list-style-type: none"> <li>• ON: Currently tracking and using GPS satellites</li> <li>• OFF: Not tracking GPS satellites</li> </ul>	ON, OFF
n(c2)	Provides tracking status of each GPS satellite: <ul style="list-style-type: none"> <li>• Y: Tracking enabled for this satellite</li> <li>• N: Tracking disabled for this satellite</li> </ul> This status is provided n times, where n is the number of satellites in the GPS constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,GPS
$PASHR,GPS,ON,USE,NYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY*5E
```

**Related Commands** \$PASHS,GPS (set command)

## GRS: GNSS Range Residuals

---

**Function** This command is used to output a GRS message containing the satellite range residuals. The message is output on the port on which the query is made. No message will be output until a position is computed.

**Command Format Syntax**  
 \$PASHQ,GRS[\*cc]

**Response Format Syntax**  
 \$--GRS,m1,d2,n(f3)\*cc

## Parameters

Parameter	Description	Range
"\$-GRS" Header	\$GPGRS: Only GPS satellites are used. \$GLGRS: Only GLONASS satellites are used. \$GBGRS: Only BeiDou satellites are used. \$GNGRS: Several constellations are used (GPS, SBAS, GLONASS, QZSS, BeiDou)	\$GPGRS \$GLGRS \$GBGRS \$GNGRS
m1	Current UTC time of GGA position (hhmmss.ss)	000000.00- 235959.99
d2	Mode used to compute range residuals	Always "1"
f3	Range residual for satellite used in position computation (repeated "n" times, where n is the number of satellites used in position computation). Residuals are listed in the same order as the satellites in the GSA message so that each residual provided can easily be associated with the right satellite.	±999.999
*cc	Checksum	*00-*FF

### Example

**\$PASHQ,GRS**

\$GNGRS,141003.50,1,1.14,-0.48,0.26,0.20,-0.94,-0.28,-1.18\*61

\$GNGRS,141003.50,1,-0.20\*4F

### See also

\$PASHS,NME

## Automatic Output of GRS Messages

This is a reminder on how to output GRS messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GRS,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GRS messages on port A at a rate of 0.5 second:

```
$PASHS,NME,GRS,A,ON,0.5
```

## GSA: GNSS DOP and Active Satellites

---

### Function

This command is used to output a GSA message containing data related to DOP values and satellites used in the position solution.



**Command Format Syntax**

\$PASHQ,GSA[\*cc]

**Response Format Syntax**

\$--GSA,c1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13,d14,f15,f16,f17,d18\*cc

**Parameters**

Parameter	Description	Range
"\$--GSA" Header	\$GPGSA: Only GPS satellites are used. \$GLGSA: Only GLONASS sats are used. \$GBGSA: OnlyBeiDou satellites are used. \$GNGSA: Several constellations (GPS, SBAS, GLONASS, BeiDou) are used.	\$GPGSA, \$GLGSA, \$GBGSA, \$GNGSA
c1	Output mode: • M: Manual • A: Automatic	M, A
d2	Position indicator: • 1: No position available • 2: 2D position • 3: 3D position	1-3
d3-d14	Satellites used in the position solution (blank fields for unused channels)	See table below.
f15	PDOP	0-9.9
f16	HDOP	0-9.9
f17	VDOP	0-9.9
d18	GNSS System ID	1-6; See table below.
*cc	Checksum	*00-*FF

System	System ID	Satellite ID	Signal ID	Signal Channel
GPS	1 (GP)	1-99 1-32 reserved for GPS 33-64 reserved for SBAS (65-99 undefined)	0 1 2 3 4 5 6 7 8 9-F	All signals L1 C/A L1 P(Y) L1 M L2 P(Y) L2C- M L2C-L L5-I L5-I L5-Q Reserved
GLONASS	2 (GL)	33-99 33-64 reserved for SBAS 65-99 reserved for GLONASS	0 1 2 3 4 5-F	All signals G1 C/A G1 P G2 C/A GLONASS (M) G2 P Reserved

System	System ID	Satellite ID	Signal ID	Signal Channel
GALILEO	3 (GA)	1-99 1-36 reserved for Galileo 37-64 reserved for Galileo SBAS (65-99 undefined)	0	All signals E5a
			1	E5b
			2	E5 a+b E6-A
			3	E6-BC L1-A
			4	L1-BC
			5	Reserved
			6	
			7 8-F	
IRNSS	4 (GI)	1-99 1-15 reserved for IRNSS (15-32 undefined) 33-64 reserved for SBAS (65-99 undefined)	0	ALI signals
			1	L5-SPS
			2	S-SPS
			3	L5-RS
			4	S-RS
			5	L1-SPS
			6-F	Reserved
BEIDOU	5 (GB)	1-99 1-40 reserved for SBS (41-99 unidentified)	0	All signals
			1	B1I
			2	B2a
			3	B2I
			4	Reserved
			5-F	
QZSS	6 (GQ)	1-99 1-9 reserved for QZSS* 55-63 is reserved for QZSS SBAS* (64-99 undefined)	0	All signals L1
			1	L1 C/A
			2	L1C (D) L1C (P)
			3	L2C-L L5-I
			4	L5-Q
			5	L6D L6E
			6-F	Reserved
Reserved	7 to F			

\* Satellite ID is obtained by taking the 6 Low Significant Digits of the 8-bit PRN Number (e.g. Satellite ID of PRN 193 is 1).

## Example

### \$PASHQ,GSA

```
$GNGSA,A,3,20,11,13,23,17,04,31,,,,,1.6,0.9,1.3,1*3C
```

```
$GNGSA,A,3,81,83,68,,,,,,,,,1.6,0.9,1.3,2*32
```

## Automatic Output of GSA Messages

This is a reminder on how to output GSA messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GSA,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GSA messages on port A at a rate of 0.5 second:

\$PASHS,NME,GSA,A,ON,0.5

## GST: GNSS Pseudo-Range Error Statistics

---

**Function** This command is used to output a GST message containing standard deviations relevant to the position solution.

**Command Format Syntax**  
\$PASHQ,GST[\*cc]

**Response Format Syntax**  
\$--GST,m1,f2,f3,f4,f5,f6,f7,f8\*cc

### Parameters

Parameter	Description	Range
"\$--GST" Header	\$GPGST: Only GPS satellites are used. \$GLGST: Only GLONASS satellites are used. \$NGGST: Several constellations (GPS, SBAS, GLONASS, BEIDOU) are used.	\$GPGST, \$GLGST, \$NGGST
m1	Current UTC time of position (hhmmss.ss)	000000.00- 235959.99
f2	RMS value of standard deviation of range inputs (DGNS corrections included), in meters	0.000-999.999
f3	Standard deviation of semi-major axis of error ellipse, in meters	0.000-999.999
f4	Standard deviation of semi-minor axis of error ellipse, in meters	0.000-999.999
f5	Orientation of semi-major axis of error ellipse, in degrees from true North	0 to 180
f6	Standard deviation of latitude error, in meters	0.000-999.999
f7	Standard deviation of longitude error, in meters	0.000-999.999
f8	Standard deviation of altitude error, in meters	0.000-999.999
*cc	Checksum	*00-*FF

**Example** \$PASHQ,GST  
\$NGGST,154013.80,0.642,1.746,1.303,27.197,1.663,1.407,2.456\*79

**See also** \$PASHS,NME

## Automatic Output of GST Messages

This is a reminder on how to output GST messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GST,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GST messages on port A at a rate of 0.5 second:

```
$PASHS,NME,GST,A,ON,0.5
```

## GSV: GNSS Satellites in View

---

**Function** This command is used to output a GSV message containing information on the satellites in view.

**Command Format** **Syntax**  
 \$PASHQ,GSV[\*cc]

**Response Format** **Syntax**  
 \$--GSV,d1,d2,d3,n(d4,d5,d6,f7),d8\*cc

The set of parameters (d4,d5,d6,f7) can be repeated up to 4 times in a single response line, corresponding to the description of 4 different satellites. The number of response lines is therefore dependent on the number of satellites in view (e.g. three response lines if between 9 and 12 satellites are visible).

## Parameters

Parameter	Description	Range
"\$-GSV" Header	\$GPGSV: GPS satellites. \$GLGSV: GLONASS satellites \$GAGSV: GALILEO satellites \$GSGSV: SBAS satellites (including QZSS L1 SAIF) \$GQGSV: QZSS satellites \$GBGSV: BeiDou satellites	\$GPGSV, \$GLGSV \$GAGSV \$GSGSV \$GQGSV \$GBGSV
d1	Total number of messages	1-4
d2	Message number	1-4
d3	Total number of satellites in view	0-16
d4	Satellite PRN	GPS: 1-32 GLONASS: 65-96 SBAS: 1-44 GALILEO: 1-36 QZSS: 1-5 BEIDOU: 1-32 IRNSS: 1-7
d5	Elevation in degrees	0-90
d6	Azimuth in degrees	0-359
f7	SNR in dB.Hz	30.0-60.0
d8	Galileo satellite indicator: • 1: not a Galileo satellite • 7: a Galileo satellite	1, 7
*cc	Checksum	*00-*FF

### Example

#### \$PASHQ,GSV

```
$GPGSV,3,1,12,04,21,308,44,11,16,311,42,14,62,185,50,15,03,047,,1*67
```

```
...
```

```
$GSGSV,1,1,04,09,09,110,,14,31,153,42,16,35,200,46,19,36,165,48,1*67
```

```
$GLGSV,3,2,09,76,19,038,40,77,20,093,44,82,35,192,48,83,58,285,52,1*7A
```

```
...
```

```
$GQGSV,1,1,00,1*65
```

```
$GAGSV,1,1,03,12,30,311,42,19,37,256,43,20,,,43,7*7A
```

```
$GBGSV,1,1,04,08,10,333,36,11,44,273,48,12,31,205,45,14,09,263,38,1*70
```

### See also

\$PASHS,NME

## Automatic Output of GSV Messages

This is a reminder on how to output GSV messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,GSV,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output GSV messages on port A at a rate of 10 seconds:

```
$PASHS,NME,GSV,A,ON,10
```

## HDT: True Heading

---

**Function** This command is used to output an HDT message (last computed true heading in degrees).

**Command Format** **Syntax**  

```
$PASHQ,HDT[*cc]
```

**Response Format** **Syntax**  

```
$GPHDT,f1,T*cc
```

### Parameters

Parameter	Description	Range
f1,T	Last computed heading value, in degrees "T" for "True".	0-359.99
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,HDT
$GPHDT,121.2,T*35
```

### See Also

#### \$PASHS,NME **Automatic Output of HDT Messages**

This is a reminder on how to output HDT messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,HDT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output HDT messages on port A at a rate of 1 second:

```
$PASHS,NME,HDT,A,ON,1
```

## HPR: Extended Attitude Message, UTC Time-Tagged

---

**Function** This command is used to output an attitude message tagged to UTC time. In terms of content, the HPR message is similar to the AT2 message.

**Command Format Syntax**  
\$PASHQ,HPR[\*cc]

**Response Format Syntax**  
\$PASHR,HPR,m1,f2,f3,f4,f5,f6,d7,d8,d9,f10\*cc

## Parameters

Parameter	Description	Range
m1	UTC time of attitude data (hhmmss.ss).	000000.00-235959.99
f2	True heading angle in degrees.	000.00-359.99999
f3	Pitch angle in degrees.	±90.00000
f4	Roll angle in degrees.	±90.00000
f5	Carrier measurement RMS error, in meters.	Full range of real variables
f6	Baseline RMS error, in meters. (=0 if baseline is not constrained)	Full range of real variables
d7	Integer ambiguity: • 0: Fixed • >0: Float	0, >0
d8	Attitude/heading mode status: • 0: Operation with fixed baseline length • 1: Calibration in progress • 2: Flex (flexible) baseline mode ON	0, 1, 2
d9	Character string of the type "y.xxx" defined as follows: • "y" refers to the antenna setup: y=0: no length constraint is applied y=1: heading mode (one vector) y=2: attitude mode (2 vectors) y=3: attitude mode with 3 or more vectors • Each "x" (0 to 9) represents the number of Double Differences (DD) used in the corresponding baseline. If this number is greater than 9, then "9" is reported. If there are only 2 vectors, the last x is "0"  Double differences refer to the very last integer second time-tagged epoch.	y.xxx
f10	PDOP corresponding to vector V12, as computed for the very last integer second (time-tagged epoch). Empty if PDOP unknown.	
*cc	Checksum	*00.*FF

## Related Commands

\$PASHS,NME

### Automatic Output of HPR Messages

This is a reminder on how to output HPR messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,HPR,<port_ID>,ON,<Rate>
```



For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output HPR messages on port A at a rate of 0.2 second:

```
$PASHS,NME,HPR,A,ON,0.2
```

## LBN: L-Band Almanac and Status

---

**Function** This command queries the receiver for the L-band configuration and status.

**Command Format** **Syntax**  
\$PASHQ,LBN[\*cc]

**Parameters**

None.

**Response Format** **Syntax**  
\$PASHR,LBN,d1,d2,s3,s4,d5,d6,d7,d8,s9,s11,s12\*cc

## Parameters

Parameter	Description	Range
d1	Total number of messages	1-99
d2	Message number	1-d1
s3	L-band provider name	31 chars max.
s4	L-band satellite name	31 chars max.
d5	ID of internally assigned L-band satellite	1-99
d6	Frequency (Hz)	
d7	Baud rate (bits/s)	600, 1200, 2400, 2800
d8	Longitude, in degrees	-179.9-180.0
s9	Type (hard coded or user-defined). See \$PASHS,LBN,SAT)	ALM, USR
s10	Satellite usage status: • ON: Used • OFF: Not used	ON, OFF
s11	Satellite tracking status: • 1 or 2: Channel number • OFF: Not tracked	1, 2, OFF
s12	Corrections from this satellite are: • Used (ON) • Not used (OFF)	ON, OFF
*cc	Checksum	*00-*FF

## Examples

### \$PASHQ,LBN

```
$PASHR,LBN,RTX,USE,ONRTXAP,RTXWN,RTXEN,RTXCEN,RTXSA*46
```

### \$PASHQ,LBN

```
$PASHR,LBN,RTX,USE,OFF*4A
```

### \$PASHQ,LBN

```
$PASHR,LBN,8,1,RTX,RTXAP,2,1539832500,600,109.0,ALM,OFF,OFF,OFF*03
$PASHR,LBN,8,2,RTX,RTXWN,3,1557861500,600,-101.0,ALM,OFF,OFF,OFF*22
$PASHR,LBN,8,3,RTX,RTXEN,4,1557859000,600,-101.0,ALM,OFF,OFF,OFF*38
$PASHR,LBN,8,4,RTX,RTXCEN,5,1557815000,2400,-101.0,ALM,OFF,OFF,OFF*00
$PASHR,LBN,8,5,RTX,RTXSA,6,1539832500,600,-98.0,ALM,OFF,OFF,OFF*14
$PASHR,LBN,8,6,RTX,RTXEA,7,1545490000,2400,25.0,ALM,OFF,OFF,OFF*11
$PASHR,LBN,8,7,RTX,RTXIO,8,1545530000,600,82.0,ALM,ON,1,ON*55
$PASHR,LBN,8,8,TRS,TERIA,16,1546310000,2400,25.0,ALM,OFF,OFF,OFF*39
```

## Comments

- \$PASHQ,LBN will not return any information about an L-band provider that has been made unavailable. The command will be NAKed if none of the L-band providers are available.
- If a parameter of an L-band provider is unavailable, inapplicable or unknown, then the corresponding field is reported empty.

**See Also** \$PASHS,LBN,USE  
\$PASHS,LBN,SAT

## LBN,BEM: User-Defined L-Band Satellites

---

**Function** This command provides the list of user-defined L-band satellites.

**Command Format** **Syntax**  
\$PASHQ,LBN,BEM[\*cc]

### Parameters

None.

**Response Format** **Syntax**  
\$PASHR,LBN,BEM,s1,n(B,f2,f3,f4,f5,f6)\*cc

### Parameters

Parameter	Description	Range
s1	Name of L-band satellite	31 characters max.
n	Indicates that the expression between brackets that follows will be repeated n times. There may be up to 5 different ellipses possible for a given L-band satellite, hence "n" ranging from 1 to 5.	1-5
B	Separator between foot print beam ellipses	-
f2	Beam foot note orientation, in degrees	-89.9 to 90.0
f3	Latitude axis radius, in degrees	0.0 to 90.0
f4	Longitude axis radius, in degrees	0.0 to 180.0
f5	Latitude center	-90.0 to 90.0
f6	Longitude center	-179.9 to 180.0
*cc	Optional checksum	*00-*FF

### Examples

\$PASHQ,LBN,BEM  
\$PASHR,LBN,BEM,RTXDD,B,70.1,40,30,0,37,B,0.1,40,60,35,0\*1C  
\$PASHR,LBN,BEM,RTXEE\*1C

**See Also** \$PASHS,LBN,BEM

## LCK: Lock Mode Status & Receiver Lock Status

---

**Function** This command is used to read the current status of the lock mode and whether the receiver is currently locked or not.

**Command Format Syntax**  
**\$PASHQ,LCK[\*cc]**

**Response Format Syntax**  
**\$PASHR,LCK,s1,d2\*cc**

### Parameters

Parameter	Description	Range
s1	Lock mode status: <ul style="list-style-type: none"> <li>• ON: Lock mode enabled</li> <li>• OFF: Lock mode disabled</li> </ul>	ON, OFF
d2	Receiver lock status: <ul style="list-style-type: none"> <li>• 0: Receiver unlocked</li> <li>• 1: Receiver locked</li> </ul>	0, 1
*cc	Optional checksum	*00-*FF

### Example

**\$PASHQ,LCK**  
**\$PASHR,LCK,ON,1\*00**  
 (Lock mode is ON and receiver currently locked)

**Related Commands** \$PASHS,LCK,MOD (set command)  
 \$PASHS,LCK,ON  
 \$PASHS,LCK,OFF  
 \$PASHS,ATH,PWD  
 \$PASHQ,ATH

## LCS: Local Coordinate System Status

---

**Function** This command asks the receiver to indicate the coordinate system it currently uses to deliver its position solution. A local coordinate system may be used provided its characteristics are received through the appropriate RTCM 3.1 message (1021-1028) from the base used.

**Command Format Syntax**  
 \$PASHQ,LCS[\*cc]

**Parameters**

None.

**Response Format Syntax**  
 \$PASHR,LCS,s\*cc

**Parameters**

Parameter	Description	Range
s	Status: <ul style="list-style-type: none"> <li>• ON: Local coordinate system used when available</li> <li>• OFF: Coordinate system used is WGS84 necessarily.</li> <li>• HOR: Horizontal coordinates are local</li> <li>• VER: Vertical coordinate is local</li> </ul>	ON, OFF, HOR, VER
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,LCS
$PASHR,LCS,ON*05
```

**Relevant Set Command** \$PASHS,LCS

## LOG: Editing a Log File

---

**Function** This command is used to edit the specified or current log file. A log file lists all events related to IP connections with the receiver.

**Command Format Syntax**  
 \$PASHQ,LOG[,d][\*cc]

**Parameters**

Parameter	Description	Range
d	index number of the log file you want to edit. If d is omitted, the current log file is edited.	0-900
*cc	Optional checksum	*00-*FF

**Response format Syntax**

The response is formatted as follows:

```
Date: <Year>-<Month>-<Day>
Maximum size: x Mb Duration: xx days
hh:mm:ss: <message 1>
hh:mm:ss: <message 2>
...
hh:mm:ss: <message n>
```

**Parameters**

- The first line contains the date when the log file was created.
- The second line indicates the maximum size (in Mb) permitted for the file as well as the time, in days, during which it is kept in memory.
- Each of the lines that follow contains a message that describes a connection event (time of event, beginning or end of connection, type of connection, identification of the connected device).

**Example**

```
$PASHQ,LOG*33
Date: 2014-04-08
Maximum size: 1 Mb Duration: 20 days
14:12:34: connect server,stream=l1,port=1001,IP=12.34.87.22
14:15:33: connect client,stream=l2,IP=23.33.43.12,port=7721
15:36:12: disconnect server,stream=l1,port=1001,IP=12.34.87.22
```

**Related Commands**    \$PASHS,LOG,PAR  
                           \$PASHS,LOG,DEL  
                           \$PASHQ,LOG,LST

**LOG,LST: Listing Log Files**

---

**Function**    This command is used to read the list of log files present in the receiver.

**Command Format Syntax**

```
$PASHQ,LOG,LST[*cc]
```

**Parameters**

None.

**Response format Syntax**

\$PASHR,LOG,LST,d1,d2,s3,d4\*cc

**Parameters**

Parameter	Description	Range
d1	Current number of log files in the receiver	0-900
d2	File index	0-900
s3	Filename	255 characters max.
d4	Size, in bytes	0-134217728
*cc	Optional checksum	*00-*FF

**Example****\$PASHQ,LOG,LST\*54**

\$PASHR,LOG,LST,4,0,20090408.log,1769897\*01

\$PASHR,LOG,LST,4,1,20090407.log,1769876\*00

\$PASHR,LOG,LST,4,2,20090406.log,1769787\*03

\$PASHR,LOG,LST,4,3,20090405.log,1769787\*01

**Related Commands** \$PASHS,LOG,PAR

\$PASHS,LOG,DEL

\$PASHQ,LOG

**LOG,PAR: Log File Settings**

---

**Function** This command is used to read the settings of any new log file created in the receiver.**Command Format Syntax**

\$PASHQ,LOG,PAR[\*cc]

**Parameters**

None.

**Response format Syntax**

\$PASHR,LOG,PAR,s1,d2,d3\*cc

## Parameters

Parameter	Description	Range
s1	Log file control parameter: • ON: Generation of log files enabled • OFF: Generation of log files disabled	ON, OFF
d2	Maximum size, in Mbytes	1-90
d3	Number of days during which a log file is kept in memory.	1-100
*cc	Optional checksum	*00-*FF

## Example

```
$PASHQ,LOG,PAR*5C
$PASHR,LOG,PAR,OFF,1,20*0F
```

**Related Commands** \$PASHS,LOG,PAR (set command)

## LTN: Latency

---

**Function** This command returns the current value of latency.

**Command Format Syntax**  
\$PASHQ,LTN[\*cc]

## Parameters

None.

**Response Format Syntax**  
\$PASHR,LTN,d1\*cc

## Parameters

Parameter	Description	Range
d1	Latency in milliseconds.	0-10000
*cc	Optional checksum	*00-*FF

**Example** Querying the value of latency:

```
$PASHQ,LTN
$PASHR,LTN,60*08
```

**Comments** Latency refers to the time it takes for the receiver to compute a position from the measurement time tag and prepare data



to be transmitted through the serial port. The value of latency depends on the number of locked satellites.

**Related Commands**    \$PASHS,NME

### Automatic Output of LTN Messages

This is a reminder on how to output LTN messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,LTN,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output LTN messages on port A at a rate of 10 seconds:

```
$PASHS,NME,LTN,A,ON,10
```

## MDP: Port Communication Mode

---

**Function**    This command is used to read the communication mode currently used on port D.

**Command Format**    **Syntax**  
                           \$PASHQ,MDP,c1[\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,MDP,c1,s2\*cc

### Parameters

Parameter	Description	Range
c1	Communication port (D)	D
	Communication mode currently used: <ul style="list-style-type: none"> <li>• 232, RS232 serial port</li> <li>• CAN: CAN bus</li> <li>• I2C: I<sup>2</sup>C serial bus</li> </ul>	232, CAN, I2C
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,MDP,D
$PASHR,MDP,D,232*5A
```

**See Also** \$PASHS,MDP

## MEM: Selected Memory Device

---

**Function** This command is used to read which memory the receiver is currently using.

**Command Format Syntax**  
\$PASHQ,MEM[\*cc]

**Response Format Syntax**  
\$PASHR,MEM,d[\*cc]

### Parameters

Parameter	Description	Range
d	Memory used: • 0: Internal memory • 2: USB memory	0, 2
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,MEM
$PASHR,MEM,0*2D
```

**Related Commands** \$PASHS,MEM (set command)

## NPT: How Position Type is Reported in POS Message

---

**Function** This command is used to query the receiver for the set of flags currently used to describe the position solution type provided in POS messages.

**Command Format Syntax**  
\$PASHQ,NPT[\*cc]

**Response Format Syntax**  
\$PASHR,NPT,d1,d2\*cc

## Parameters

Parameter	Description	Range
d1	Flag set number: <ul style="list-style-type: none"> <li>• If "0" (standard position type), possible tags are:               <ul style="list-style-type: none"> <li>– 1: SBAS or BeiDou Differential</li> <li>– 2: RTX Float or RTX Float dithered</li> <li>– 3: RTX Fixed or RTX Fixed dithered</li> </ul> </li> <li>• If "1" (specific position type), possible tags are:               <ul style="list-style-type: none"> <li>– 9: SBAS differential</li> <li>– 10: BeiDou Differential</li> <li>– 12: RTX Float</li> <li>– 13: RTX Fixed</li> <li>– 22: RTK Float dithered</li> <li>– 23: RTK Fixed dithered</li> </ul> </li> </ul>	0,1
d2	(Not significant)	
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,NPT
$PASHR,NPT,0,0*3E
(standard type is used)
```

**Related Commands**    \$PASHS,NPT (set command)

## NTP: Network Time Protocol Server

---

**Function**    This command is used to query the receiver for the NTP (Network Time Protocol) server status.

**Command Format**    **Syntax**  
                           \$PASHQ,NTP[\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,NTP,s1,d2\*cc

## Parameters

Parameter	Description	Range
s	NTP server ON/OFF setting: • ON: Enabled • OFF: Disabled	ON, OFF
d2	NTP server current state: • 0: Stopped • 1: Running	0, 1
*cc	Optional checksum	*00-*FF

## Examples

NTP server disabled:

```
$PASHQ,NTP
$PASHR,NTP,OFF,0*41
```

NTP server enabled but not active (because no GPS time available):

```
$PASHQ,NTP
$PASHR,NTP,ON,0*0F
```

NTP server enabled and active:

```
$PASHQ,NTP
$PASHR,NTP,ON,1*0E
```

**Query Command**    \$PASHQ,NTP

## NTR: NTRIP Settings

---

**Function**    This command is used to read the current NTRIP settings. When c6 is omitted in the query command, the returned NTRIP settings are those for the port defined through the \$PASHS,NTR,PAR command last run.

**Command Format**    **Syntax**  
                           \$PASHQ,NTR[,c6][\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,NTR,ADD=s1,PRT=d2,LGN=s3,PWD=s4,TYP=d5,IPP=c6,  
                           ACN=d7\*cc

## Parameters

Parameter	Description	Range
ADD=1	Caster IP address or host name	32 characters max.
PRT=d2	Caster port number	0-65535
LGN=s3	Login	32 characters max.
PWD=s4	Password	32 characters max.
TYP=d5	Caster type: <ul style="list-style-type: none"> <li>• 0: Client</li> <li>• 1: Server</li> <li>• Client V2</li> <li>• Server V2</li> </ul>	0-3
IPP=c6	Internet port used to connect to the caster: <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul>	P, Q
ACN=d7	Auto-connection: <ul style="list-style-type: none"> <li>• 0: After a power cycle, there won't be an automatic NTRIP connection (default)</li> <li>• 1: After a power cycle, the NTRIP connection will be restored</li> </ul>	0, 1
*cc	Checksum	*00.*FF

### Example

```
$PASHQ,NTR
$PASHR,NTR,ADD=192.34.76.1,PRT=2100,LGN=trimble,PWD=u6huz8,
TYP=0,IPP=P,ACN=1*40
```

**See also** \$PASHS,NTR,PAR  
\$PASHQ,NTR,TBL

## NTR,MTP: Connection to Mount Point

---

**Function** This command is used to read the current NTRIP mount point to which the specified Internet port is connected.

**Command Format** **Syntax**  
\$PASHQ,NTR,MTP[,c1]\*cc]

**Parameters**

None.

Parameter	Description	Range
c1	Internet port used for the connection to the embedded NTRIP caster. <ul style="list-style-type: none"> <li>• P: TCP/IP stream 1</li> <li>• Q: TCP/IP stream 2</li> </ul> If c1 is omitted, the receiver will return the mount point name corresponding to the port last defined through the \$PASHS,NTR,PAR command.	P, Q
*cc	Checksum	*00-*FF

**Response Format****Syntax**

\$PASHR,NTR,MTP,s1\*cc

**Parameters**

Parameter	Description	Range
s1	NTRIP mount point name If "OFF", the port is not connected to any NTRIP caster mount point.	100 characters max. or "OFF"
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,NTR,MTP,P
$PASHQ,NTR,MTP
$PASHR,NTR,MTP,NAN2*06
```

**Related Commands**

\$PASHS,NTR,MTP (set command)

**NTR,TBL: Source Table**

---

**Function**

This command is used to read the source table stored in the receiver.

**Command Format****Syntax**

\$PASHQ,NTR,TBL[\*cc]

**Response Format****Syntax**

```
$PASHR,NTR,TBL
SOURCETABLE 200 OK
<source table as specified in the RTCM standard>
```

## ENDSOURCETABLE

**Parameters**

Source table as defined in the NTRIP standard.

**Example**

```

$PASHQ,NTR,TBL
$PASHR,NTR,TBL
SOURCETABLE 200 OK
Content-Type: text/plain
Content-Length: 7864
CAS;129.217.182.51;80;ICD;BKG;0;GER;51.5;7.5;Trial Broadcaster
NET;GREF;BKG;B;N;http://igs.ifag.deGREF.htm;none;
denise.dettmering@bkg.bund.de;none
NET;IGSIGLOS;BKG;B;N;http://igsch.jpl.nasa.gov/projects/rtwg
;none;denise.dettmering@bkg.bund.de;none
STR;FFMJ2;Frankfurt;RTCM2.0;1(1),3(19),16(59);0;GPS;GREF;GER;50.12;8
.68;0;1;GPSNetV1.9;none;N;N;560;DemoSTR;FFMJ1;Frankfurt;RTCM
2.1;3(19),16(59),18(1),19(1);2;GPS;GREF;GER;50.09;8.66;0;0;GPSNet
V1.9;none;N;N;2800;Demo
STR;FFMJ0;Frankfurt;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;50.09;8.66;0;0;Javad Legacy E;none;N;N;3600;Demo
STR;LEIJ0;Leipzig;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;51.33;12.37;0;0;Javad Legacy E;none;B;N;3600;none
STR;WTZJ0;Wetzell;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;49.13;12.88;0;0;Javad Legacy E;none;B;N;3600;none
STR;HELJ0;Helgoland;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;54.18;7.88;0;0;Javad Legacy E;none;B;N;3600;none
STR;TITZ0;Titz;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;51.00;6.42;0;0;Javad Legacy E;none;B;N;3600;none
STR;HUEG0;Huegelheim;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;47.82;7.62;0;0;Javad Legacy E;none;B;N;3600;none
STR;DREJ0;Dresden;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;51.05;13.73;0;0;Javad Legacy E;none;B;N;3600;none
STR;SASS0;Sassnitz;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;54.51;13.64;0;0;Javad Legacy E;none;B;N;3600;none
STR;KARJ0;Karlsruhe;RAW;Compact(1);2;GPS+GLO;IGSIGLOS;
GER;49.01;8.41;0;0;Javad Legacy E;none;B;N;3600;none
STR;WILH0;Wilhelmshaven;RTCM
2.0;1(1),3(19),16(59);0;GPS;GREF;GER;53.52;8.10;0;1;GPSNet
V1.9;none;B;N;560;VRS
ENDSOURCETABLE

```

**See also** \$PASHS, NTR, LOD  
 \$PASHS, NTR, PAR  
 \$PASHS, NTR, MTP

## OBS: Observations Usage

---

**Function** This command is used to ask the receiver in which way the available GNSS signals are used.

**Command Format Syntax**  
\$PASHQ,OBS[\*cc]

**Response Format Syntax**  
\$PASHR,OBS,s[\*cc]

### Parameters

Parameter	Description	Range
s	Way the available GNSS signals are used: <ul style="list-style-type: none"> <li>• ALL: All signals are used</li> <li>• OPT: Only the best signal per GNSS band is used</li> <li>• LEG: Same as OPT but only legacy L1P and L2P (not L2C) signals are output.</li> </ul>	ALL, OPT, LEG
*cc	Optional checksum	*00-*FF

### Example

```
$PASHQ,OBS
$PASHR,OBS,OPT*4D
```

**Relevant Set Command** \$PASHS,OBS

## OCC: Occupation State and Parameters

---

**Function** This command is used to read the current occupation settings.

**Command Format Syntax**  
\$PASHQ,OCC[\*cc]

**Response Format Syntax**  
\$PASHR,OCC,d1,d2[,s3,s4]\*cc



## Parameters

Parameter	Description	Range
d1	Occupation type: • 0: Static • 1: Quasi-static • 2: Dynamic	0-2
d2	Occupation state: • 0: Occupation in progress • 1: No occupation in progress	0-1
s3	Occupation name	255 characters max.
s4	Occupation description	255 characters max.
*cc	Checksum	*00-*FF

## Examples

\$PASHQ,OCC

\$PASHR,OCC,2,1\*38

**Relevant Set Command**    \$PASHS,OCC

## OPTION: Installed Receiver Firmware Options

---

**Function**    This command is used to list the firmware options currently installed in the receiver. The returned message includes one response line per installed option. The first line provides the receiver serial number.

**Command Format Syntax**  
\$PASHQ,OPTION[\*cc]

**Response Format Syntax**  
\$PASHR,OPTION,c1,s2,h3\*cc

## Parameters

Parameter	Description	Range
c1	Option ID	(See table below)
s2	Option label	(See table below)
h3	Hexadecimal unlock code	13 characters max.
*cc	Checksum	*00-*FF

Option ID	Label	Description
@1	GEOFENCING_WW	Worldwide enabled receiver
@2	GEOFENCING_CHINA	China-only enabled receiver
@3	GEOFENCING_JAPAN	Japan-only enabled receiver
@4	GEOFENCING_BRAZIL	Brazil-only enabled receiver
@5	GEOFENCING_N_AMERICA	North-America-only enabled receiver
@6	GEOFENCING_L_AMERICA	Latin-America-only enabled receiver
@7	GEOFENCING_RUSSIA	Russia-only enabled receiver
@8	GEOFENCING_INDIA	India-only enabled receiver
@9	GEOFENCING_TURKEY	Turkey-only-enabled receiver
2	2HZ	Enables output rate up to 2 Hz (for position/raw data)
3	Dithered RTK 30/30	Dithered RTK 30/30: 30 cm horizontal, 30 cm vertical (RTK only). [J] and [7] predominate over [3].
4	ViewPoint RTX	RTX L1-only mode
5	5HZ	Enables update rate up to 5 Hz for position and raw data. With this option enabled, when using \$PASHS,POP to set the internal update rate, remember only "POP,1" and "POP,5" are allowed, not "POP,2".
6	10HZ	Enables update rate up to 10 Hz for position and raw data
7	Dithered RTK 7/2	Dithered RTK 7/2: 7 cm horizontal, 2 cm vertical (RTK only). [J] predominates over [7].
8	50HZ	Enables update rate up to 50 Hz for position and raw data
9	FieldPoint RTX	CPT degraded version. Dominates RangePoint RTX
10	10-10CM	Dithered RTK 10/10: 10cm Horizontal Precision, 10cm Vertical Precision. (RTK only). [J] and [F] override [10]
B	BEIDOU	BEIDOU satellites tracking
C	CPRTX	Enables Trimble CenterPoint RTX
c	CASTER	Embedded NTRIP caster
D	DUO	Allows use of dual-sensor mode, raw data output from both sensors, internal heading between the two antennas, or two independent position for each antenna.

Option ID	Label	Description
E	ATTITUDE	Same as D option. Additionally, enables full-attitude engine with up to 4 sources of raw data (up to 4 antennas) regardless of their assignment across two or more boards. Computing external heading is also possible.
F	FLYINGRTRK	Enables Flying RTK computation (position and vector). [J], [7] and [3] predominate over [F].
G	GLONASS	GLONASS satellites tracking
I	RAIM	Enables receiver to output RAIM-related messages
J	RTKROVER	Allows fixed RTK computations (position and vector). ([J] option predominates over all other RTK options ([F], [3], [7]).
K	RTKBASE	Enables the receiver to generate differential messages: <ul style="list-style-type: none"> <li>• RTCM 2.3</li> <li>• RTCM 3.0</li> <li>• CMR / CMR+</li> <li>• ATOM</li> </ul>
L	LBAND	Enables L-band reception
N	GPS	GPS+SBAS+QZSS satellites tracking
O	GALILEO	GALILEO satellites tracking
P	RPRTX	Enables Trimble RangePoint RTX
R	RECORD	Enables data recording in memory. ATL.LOG data can be recorded without this option installed.
W	20HZ	Enables output rate up to 20 Hz (for position/raw data)
W1	WARRANTY 1 YEAR	Extends warranty period by one year.
W2	WARRANTY 2 YEAR	Extends warranty period by two years.
X	L1 TRACKING	Enables tracking of the following signals: <ul style="list-style-type: none"> <li>• GPS, QZSS, SBAS L1</li> <li>• GLONASS G1</li> <li>• GALILEO E1</li> <li>• BEIDOU B1 (phase 2)</li> </ul> This option is ALWAYS set.
Y	L2 TRACKING	Enables tracking of the following signals: <ul style="list-style-type: none"> <li>• GPS, QZSS L2</li> <li>• GLONASS G2</li> <li>• GALILEO E5b</li> <li>• BEIDOU B2</li> </ul>

**Example**

```
$PASHQ,OPTION
$PASHR,OPTION,0,SERIAL NUMBER,200751223*7A
$PASHR,OPTION,K,RTKBASE,6756975c71766*36
$PASHR,OPTION,G,GLONASS,6756945714671*7B
```

**Related Commands**    \$PASHS,OPTION (set command)

## OPTION,EXP: Installed Firmware Options and Expiration Dates

---

**Function**    This command is used to read the currently installed firmware options, and for those that are installed only temporarily, their expiration dates. Each response line describes an option. The first line contains the receiver's serial number.

**Command Format**    **Syntax**  
                           \$PASHQ,OPTION,EXP[\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,OPTION,s1,s2,h3[,s4]\*cc

**Parameters**

Parameter	Description	Range
s1	Option ID	2 characters max.
s2	Option label	
h3	Unlock code (hexadecimal)	13 characters max.
s4	Expiration date (ddmmyyyy) Blank if the option is permanent.	8 characters
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,OPTION,EXP
$PASHR,OPTION,0,SERIAL NUMBER,5426C01549*37
$PASHR,OPTION,@1,GEOFENCING_WW,45CE159F386CE*31
$PASHR,OPTION,N,GPS,45CE1567A006E*61
$PASHR,OPTION,G,GLONASS,45CE15D2B731B*17
$PASHR,OPTION,O,GALILEO,45CE14B8994AA*1D
$PASHR,OPTION,B,BEIDOU,45CE166CC5179*35
$PASHR,OPTION,X,L1TRACKING,45CE13C47D59B*3E
$PASHR,OPTION,Y,L2TRACKING,45CE10952A424*38
$PASHR,OPTION,W,20HZ,45CE14ED8C006*52
```

```

$PASHR,OPTION,Z,100HZ,45CE16C6EF32F*13
$PASHR,OPTION,J,RTKROVER,45CE1290294B0*40
$PASHR,OPTION,K,RTKBASE,45CE123D822BA*00
$PASHR,OPTION,F,FLYINGRTK,45CE1366D41BF*03
$PASHR,OPTION,D,DUO,45CE11D078636*0E
$PASHR,OPTION,I,RAIM,45CE14BD203F9*4F
$PASHR,OPTION,E,ATTITUDE,45CE1667B74DB*50
$PASHR,OPTION,L,LBAND,45CE13D29974A*63
$PASHR,OPTION,R,RECORD,45CE13C5DF089*44
$PASHR,OPTION,C,CPRTX,45CE10966E6AD*1C

```

**Related Commands**    \$PASHS,OPTION

## OUT: Current Status of Periodic Messages

---

**Function**    This command is used to read the status of all the periodic messages programmed on the requested port (or the current port if no port is specified).

**Command Format**    **Syntax**  
                           \$PASHQ,OUT[,c1][\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,OUT,c1,s2\*cc

### Parameters

Parameter	Description	Range
c1	ID of interrogated port: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• I, P, Q, F: TCP/IP stream</li> </ul>	A-D, F, I, P, Q
s2	Message status: <ul style="list-style-type: none"> <li>• ON: All periodic messages are enabled</li> <li>• OFF: All periodic messages are suspended</li> </ul>	ON, OFF
*cc	Checksum	*00-*FF

### Example

```

$PASHQ,OUT
$PASHR,OUT,A,ON*55

```

**Related Commands**    \$PASHS,OUT (set command)

## OUT,GSF: Current Status of GSOF Messages

---

**Function** This command is used to list the active GSOF messages, if any, on a given port. If no port is specified, the command returns the list of active GSOF messages on all ports.

**Command Format Syntax**  
**\$PASHQ,OUT,GSF[,c1][\*cc]**

### Parameters

None.

**Response Format Syntax**  
**\$PASHR,OUT,GSF,c1,s2,[n(s3,s4,f5)][\*cc]**

### Parameters

Parameter	Description	Range
c1	Output port for which you are listing the currently active GSOF messages: <ul style="list-style-type: none"> <li>Serial port: A, B, D</li> <li>USB serial port: C</li> <li>TCP/IP client stream: P, Q</li> <li>TCP/IP server stream: I, F</li> <li>Routed to be saved as G-file: M</li> </ul>	A,B,C,D,F,I,P,Q,M
s2	Output control: <ul style="list-style-type: none"> <li>ON: GSOF messages enabled</li> <li>OFF: GSOF messages disabled</li> </ul>	ON, OFF
s3	Message group	GSF
s4	Message number	See table below.
f5	Message output rate (in seconds)	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, etc., integer minute up to 960.
*cc	Checksum	*00-*FF

Message Type	Description
1	POSITION TIME
2	LAT, LONG, HEIGHT
3	ECEF POSITION
8	VELOCITY DATA
9	PDOP INFO

Message Type	Description
11	POSITION VCV INFO
12	POSITION VCV INFO
27	ATTITUDE INFO
33	ALL SV BRIEF INFO
38	Position Type Information

**Examples**

```
$$PASHQ,OUT,GSF
```

```
$PASHR,OUT,GSF,A,ON,GSF,1,1.000,GSF,33,1.000*62
```

```
$PASHR,OUT,GSF,C,ON,GSF,1,1.000,GSF,2,1.000,GSF,8,1.000*74
```

```
$PASHQ,OUT,GSF,A
```

```
$PASHR,OUT,GSF,A,ON,GSF,1,1.000,GSF,2,1.000,GSF,8,1.000*74
```

**Related Commands**

```
$PASHS,OUT
```

```
$PASHS,NME
```

```
$PASHS,ATM
```

```
$PASHS,RT2
```

```
$PASHS,RT3
```

```
$PASHS,CMR
```

**OUT,MSG: Output Messages**

---

**Function**

This command is used to list all the output messages currently active on the specified port, or on all the ports if no port is specified in the command.

**Command Format****Syntax**

```
$PASHQ,OUT,MSG[,c1][*cc]
```

## Parameters

Parameter	Description	Range
c1	Port selection: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• M: Internal memory</li> <li>• F, I, P, Q: TCP/IP stream</li> <li>• c1 missing: Lists all currently active output messages on all ports.</li> </ul>	A-D, F, I, M, P, Q
*cc	Checksum	*00-*FF

## Response Format Syntax

```
$PASHR,[d0,]OUT,MSG,c1,s2,[n(s3,s4,f5)][*cc]
```

## Parameters

Parameter	Description	Range
d0	Sensor number	1, 2
c1	Selected output port: <ul style="list-style-type: none"> <li>• A, B, D: Serial port</li> <li>• C: USB serial port</li> <li>• M: Internal memory</li> <li>• F, I, P, Q: TCP/IP stream</li> </ul>	A-D, F, I, M, P, Q
s2	Periodic output message status: <ul style="list-style-type: none"> <li>• ON: Enabled</li> <li>• OFF: Suspended</li> </ul>	ON, OFF
s3	Message group	ATM, CMR, NME, RT2, RT3
s4	Message name	See \$PASHS commands relevant to ATM, CMR, NME, RT2 and RT3.
f5	Message output rate, in seconds	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, etc., integer minutes up to 960.
*cc	Checksum	*00-*FF

## Examples

### \$PASHQ,OUT,MSG

```
$PASHR,1,OUT,MSG,A,ON,ATM,BLN,1.000,ATM,TT1,1.000,ATM,TT2,1.000,ATM,TT3,1.000*62
```

```
$PASHR,1,OUT,MSG,B,ON*11
```

```
$PASHR,1,OUT,MSG,D,ON,ATM,PVT,1.000,ATM,ANG,1.000,ATM,BLN,1.000,ATM,VEC,1.000,ATM,NAV,1.000,ATM,DAT,1.000,ATM,RNX004,1.000,ATM,TT1,1.000,ATM,TT2,1.000,ATM,TT3,1.000,ATM,ATR,1.000,ATM,EVT,1.000*55
```

```
$PASHR,1,OUT,MSG,G,ON,NME,ZDA,1.000,ATM,PVT,0.100,ATM,ANG,1.000,ATM,BLN,1.000,ATM,VEC,1.000,ATM,NAV,1.000,ATM,STA,1.000*1B
```



```
$PASHR,1,OUT,MSG,M,OFF,ATM,PVT,1.000,ATM,NAV,1.000,ATM,RNX000,
1.000,ATM,ATR,1.000,ATM,EVT,1.000*74
$PASHR,2,OUT,MSG,A,ON*11
$PASHR,2,OUT,MSG,B,ON*12
$PASHR,2,OUT,MSG,D,ON*14
$PASHR,2,OUT,MSG,G,ON,ATM,PVT,0.100*1E
$PASHR,2,OUT,MSG,M,OFF*53
```

**\$PASHQ,OUT,MSG,M**

```
$PASHR,1,OUT,MSG,M,OFF,ATM,PVT,1.000,ATM,NAV,1.000,ATM,RNX000,
1.000,ATM,ATR,1.000,ATM,EVT,1.000*74
$PASHR,2,OUT,MSG,M,OFF*53$PASHR,2,OUT,MSG,M,OFF*53
```

**Related Commands**

```
$PASHS,OUT
$PASHS,NME
$PASHS,ATM
$PASHS,RT2
$PASHS,RT3
$PASHS,CMR
```

## OUT,PGN: Current Status of CAN-NMEA 2000 Messages

---

**Function** This command is used to list the active NMEA 2000 messages, if any, on the CAN port.

**Command Format** **Syntax**

```
$PASHQ,OUT,PGN[,c1][*cc]
```

**Parameters**

None.

**Response Format** **Syntax**

```
$PASHR,OUT,PGN,c1,s2,[n(s3,s4,f5)][*cc]
```

**Parameters**

Parameter	Description	Range
c1	CAN port (V necessarily):	V
s2	Output control: <ul style="list-style-type: none"> <li>• ON: GSOF messages enabled</li> <li>• OFF: GSOF messages disabled</li> </ul>	ON, OFF
s3	Message group	PGN
s4	Message type	See table below.

Parameter	Description	Range
f5	Message output rate (in seconds)	0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, etc., integer minute up to 960.
*cc	Checksum	*00-*FF

Message Type	Description
126992	System Time
129025	Position, Rapid Update
129026	COG & SOG, Rapid Update
129027	Position Delta, High Precision Rapid Update
129029	GNSS Position Data
129551	GNSS Differential Correction Receiver Signal

**Example**

```
$$PASHQ,OUT,PGN
$PASHR,OUT,PGN,V,ON,PGN,129026,1.000,PGN,129029,0.100*17
```

**Related Commands**    \$PASHS,PGN

## PEM: Position Elevation Mask

---

**Function**

This command is used to read the current value of the elevation mask used in the position processing. The read value is the one used in all engines and sensors.

**Command Format**    **Syntax**

```
$PASHQ,PEM[*cc]
```

**Response Format**    **Syntax**

```
$PASHR,PEM,d1*cc
```

**Parameters**

Parameter	Description	Range
d1	Elevation mask angle	0-30°
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,PEM
$PASHR,PEM,9*39
```

**Related Commands**    \$PASHS,PEM (set command)  
                               \$PASHQ,ELM

## PGS: Primary GNSS System

---

**Function**            This command is used to know which GNSS is currently used as the primary GNSS system.

**Command Format**    **Syntax**  
                               \$PASHQ,PGS[\*cc]

**Response Format**    **Syntax**  
                               \$PASHR,PGS,s1\*cc

### Parameters

Parameter	Description	Range
s1	GNSS system used as primary: <ul style="list-style-type: none"> <li>• GPS: GPS</li> <li>• GLO: GLONASS</li> <li>• BDS: BeiDou</li> <li>• GAL: Galileo</li> </ul>	GPS, GLO, BDS, GAL
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,PGS
$PASHR,PGS,GPS*4D
```

**Related Commands**    \$PASHS,PGS (set command)

## PHE: Active Edge of Event Marker Pulse

---

**Function**            This command is used to read the current choice of active edge for the event marker pulse (used in photogrammetry).

**Command Format**    **Syntax**  
                               \$PASHQ,PHE[\*cc]

**Response Format Syntax**

\$PASHR,PHE,c\*cc

**Parameters**

Parameter	Description	Range
c	Active edge: • R: Rising • F: Falling	R, F
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,PHE
$PASHR,PHE,R*57
```

**Related Commands** \$PASHS,PHE  
\$PASHS,NME,TTT

**PIN: Programmable Pin**

---

**Function** This command returns the current settings of the specified programmable pin on the I/O connector. The response is returned on the port routing the query command.

**Command Format Syntax**

\$PASHQ,PIN[,d1]\*cc]

**Parameters**

Parameter	Description	Range
d1	Identification of the programmable pin ID. If d1 is not specified, the response will include information about all the available programmable pins.	0
*cc	Optional checksum	*00-*FF

**Response Format Syntax**

\$PASHR,PIN,d1,s2[,d3]\*cc

## Parameters

Parameter	Description	Range
d1	Programmable pin ID recalled in this field <ul style="list-style-type: none"> <li>• 0: MFO1</li> <li>• 1: MFO2</li> </ul>	0, 1
s2	Pin function status: <ul style="list-style-type: none"> <li>• OFF: no function assigned</li> <li>• PAV: Position available</li> <li>• RSP: Radar simulated pulse</li> <li>• LED: LED signal output</li> <li>• BDL: Trimble (BDxxx) LED signal output</li> <li>• GPO: General purpose output</li> </ul>	OFF, PAV, RSP, LED, BDL or GPO
d3	Depends on s2 value: <ul style="list-style-type: none"> <li>• s2=OFF: d3 has been omitted</li> <li>• s2= PAV: d3 is the number of consecutive seconds while position is not computed that causes the signal on the pin to switch to low level.</li> <li>• s2= RSP: d3 has been set to "1" to make Radar Simulated Pulse output available on the pin.</li> <li>• s2=LED: "1" means red LED is used, and "2", green LED is used.</li> <li>• s2= BDL; d3=1 means sat tracking status is output; d3=2 means RTK link status is output; d3=3 means power status is output.</li> </ul> <p>If d3 is omitted or empty, then sat tracking status is output for MFO1 and RTK link status is output for MFO2.</p> <ul style="list-style-type: none"> <li>• s2= GPO: d3=0 means active logic level is "0"; d3=1 means active logic level is "1".</li> </ul>	Omitted1-3600  1  1 or 2  1-3  0 or 1
*cc	Optional checksum	*00-*FF

### Example

```
$PASHQ,PIN
$PASHR,PIN,0,LED,2*40
(TIOA1 is green LED output)
or
$PASHR,PIN,0,LED,1*42
(TIOB2 is red LED output)
```

## POP: Reading Internal Update Rate

---

**Function** This command is used to read the internal update rate currently used for measurements and PVT process.

**Command Format Syntax**

\$PASHQ,POP[\*cc]

**Parameters**

None.

**Response format Syntax**

\$PASHR,POP,d\*cc

**Parameters**

Parameter	Description	Range
d	Current update rate, in Hz.	1, 2, 5, 10, 20, 50
*cc	Optional checksum	*00-*FF

**Example**

```
$PASHQ,POP*38
$PASHR,POP,10*16
```

**Relevant Set Command** \$PASHS,POP

## POS: Computed Position Data

---

**Function** This command allows you to query the receiver for the last computed position.

**Command Format Syntax**

\$PASHQ,POS[\*cc]

**Response Format Syntax**

\$PASHR,POS,d1,d2,m3,m4,c5,m6,c7,f8,f9,f10,f11,f12,f13,f14,f15,f16,d17\*cc

## Parameters

Parameter	Description	Range
d1	Flag describing position solution type: <ul style="list-style-type: none"> <li>• 0: Autonomous position</li> <li>• 1: RTCM code differential (or SBAS/BDS differential) (see NOTE 1)</li> <li>• 2: RTK float (or RTX) (see NOTE 2)</li> <li>• 3: RTK fixed (or RTX) (see NOTE 2)</li> <li>• 5: Estimated (dead-reckoning) mode</li> <li>• 9: SBAS differential (see NOTE 1)</li> <li>• 10: BeiDou Differential (see NOTE 3)</li> <li>• 12: RTK float (see NOTE 2)</li> <li>• 13: RTK fixed (see NOTE 2)</li> <li>• 22: RTK Float Dithered (see NOTE 4)</li> <li>• 23: RTK Fixed, Dithered (see NOTE 4)</li> </ul>	0-3, 5, 9-13, 22, 23
d2	Count of satellites used in position computation	0-26
m3	Current UTC time of position (hhmmss.ss)	000000.00-235959.99
m4	Latitude of position (ddmm.mmmmmm)	0-90° 00-59.999999 minutes
c5	North (N) or South (S)	N, S
m6	Longitude of position (dddmm.mmmmmm)	0-180° 00--59.999999 minutes
c7	East (E) or West (W)	E, W
f8	Altitude above the WGS84 ellipsoid	±9999.000
f9		
f10	True Track/Course Over Ground, in degrees	0.0-359.9
f11	Speed Over Ground, in knots	0.0-999.999
f12	Vertical velocity in m/s	±999.999
f13	PDOP	0-99.9
f14	HDOP	0-99.9
f15	VDOP	0-99.9
f16	TDOP	0-99.9
d17	Base station ID (see NOTE 5)	0-4095
*cc	Checksum	*00-*FF

### Example

#### \$PASHQ,POS

```
$PASHR,POS,3,10,151858.00,4717.960848,N,00130.499487,W,82.972,,0.0,
0.0,-0.0,2.0,1.1,1.7,1.3,2500*38
```

### NOTES

1. The reported mode for a position computed in SBAS Differential is “1” or “9”, depending on the last \$PASHS,NPT command run.

2. The reported mode for a position computed in Trimble RTX is “1” or “10”, depending on the last \$PASHS,NPT command run.
3. The reported mode for a position computed in BeiDou Differential is “2/3” or “12/13”, depending on the last \$PASHS,NPT command run.
4. The reported mode for a position computed in Dithered RTK is “2/3” or “22/23”, depending on the last \$PASHS,NPT command run.
5. When the board delivers a State Space Differential position, the base station ID field (d17) may take one of the following values:
  - 0120-0158 for conventional SBAS
  - 0183-0187 for QZSS L1 SAIF
  - 0100-0107 for RTX
  - 0119 for BeiDou Differential

### Automatic Output of POS Messages

This is a reminder on how to output POS messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,POS,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output POS messages on port A at a rate of 0.2 second:

```
$PASHS,NME,POS,A,ON,0.2
```

## POS,REF: Base Position

---

### Function

This command is used to read the base position entered manually. The coordinate system in which the position is expressed depends on the last \$PASHS,PGS command issued. No position is returned if the base has not been set up yet when you make the request or the base is a moving one.

### Command Format

#### Syntax

```
$PASHQ[,d0],POS,REF[*cc]
```



**Response Format Syntax**

```
$PASHR[,d0],POS,REF,m1,c2,m3,c4,f5*cc
```

**Parameters**

Parameter	Description	Range
d0	Antenna ID	1-2
s1	Latitude, in degrees and minutes with 6 decimal places (ddmm.mmmmmm)	0-90
c2	North (N) or South (S)	N, S
m3	Longitude, in degrees and minutes with 6 decimal places (dddmm.mmmmmm)	0-180
c4	East (E) or West (W)	E, W
f5	Height, in meters	±0-99999.999
*cc	Checksum	*00-*FF

**Examples**

```
$PASHQ,POS,REF
```

```
$PASHR,POS,REF,4717.959483,N,00130.500968,W,70.229*59
```

```
$PASHQ,POS,REF
```

```
$PASHR,POS,REF,,,,,*69
```

**Related Commands**    \$PASHS,POS (set command)

## PPP: Reading the Current PPP Settings

---

**Function**    This command is used to read the current settings of the PPP service.

**Command Format Syntax**  

```
$PASHQ,PPP[*cc]
```

**Response Format Syntax**  

```
$PASHR,PPP,s1,SRC,s2*cc
```

## Parameters

Parameter	Description	Range
s1	Selected PPP service: <ul style="list-style-type: none"> <li>• OFF: No PPP service</li> <li>• RTX: Trimble RTX</li> <li>• TRS: TERIASat</li> </ul>	OFF, RTX, TRS
s2	Selected channel to deliver corrections to PPP service: <ul style="list-style-type: none"> <li>• AUT: Corrections are delivered by either L-band channel or receiver port</li> <li>• LBN: Corrections only come from the L-band channel.</li> <li>• IP: Corrections only come from a port (serial, USB, Bluetooth, IP).</li> <li>• OFF: No corrections are provided to the PPP service.</li> </ul>	AUT, LBN, IP, OFF
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,PPP
$PASHR,PPP,RTX,SRC,AUT*1C
```

**Relevant Set Command**    \$PASHS,PPP  
**Command**                \$PASHS,PPP,SRC

## PPS: PPS Settings

---

**Function**            This command is used to read the current settings (signal period, offset and valid edge) of the PPS signal.

**Command Format Syntax**  
 \$PASHQ,PPS[\*cc]

**Response Format Syntax**  
 \$PASHR,PPS,f1,f2,c3\*cc

**Parameters**

Parameter	Description	Default	Range
f1	Period, in seconds (0= disabled)	0	0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 or 60
f2	Offset in milliseconds	0	±999.9999
c3	Active edge: • R: Rising • F: Falling	R	R, F
*cc	Checksum		*00.*FF

**Example**

```
$PASHQ,PPS
$PASHR,PPS,1,500,R*5D
```

**Relevant Set Command**    \$PASHS,PPS

**PRT: Baud Rate Settings**

---

**Function**    This command is used to read the current baud rate setting for any of the serial ports used in the receiver.

**Command Format Syntax**  
\$PASHQ,PRT[,c1][\*cc]

**Parameters**

Parameter	Description	Range
c1	Port ID of queried port. If c1 is omitted, the returned baud rate setting is that of the port routing the command.	A, B, D
*cc	Optional checksum	*00.*FF

**Response Format Syntax**  
\$PASHR,PRT,c1,d2\*cc

## Parameters

Parameter	Description	Range
c1	ID of port for which baud rate setting is returned.	A, B, D
d2	Baud rate code	0-12 (see table below)
*cc	Checksum	*00-*FF

Code	Baud Rate	Code	Baud Rate	Code	Baud Rate
2	1 200	6	19200	10	230400
3	2 400	7	38400	11	460800
4	4 800	8	57600	12	921600
5	9 600	9	115200		

## Example

```
$PASHQ,PRT,A
$PASHR,PRT,A,6*55
```

**Related Commands** \$PASHS,PRT (set command)  
\$PASHQ,CTS

## PSP: Reading the Position of the Physical Reference Station Received

---

**Function** This command queries the receiver for the position of the physical reference station received when this station is a VRS station.

**Command Format** **Syntax**  
\$PASHQ,PSP[\*cc]

### Parameters

None

**Response Format** **Syntax**  
\$PASHR,PSP,d1,d2,d3,d4,d5,m6,c7,m8,c9,f10,f11\*cc

## Parameters

Parameter	Description	Range
d1	Differential processor number	0, 3
d2	Counter reporting changes to the PSP parameters. The counter is incremented by one every time the PSP message content changes. Counter rolls back to 0 after reaching value "15".	0, 15
d3	Reference station ID	0-4095
d4	Reference station type: • 0: Physical base station • 1: Virtual base station	
d5	Position tagging: • 0: ARP • 1: L1 phase center	0, 1
m6	Latitude in degrees, decimal minutes (ddmm.mmmmm).	0, 90
c7	North (N) or South (S)	N, S
m8	Longitude in degrees, decimal minutes (dddmm.mmmmm).	0-180
c9	East (E) or West (W)	E, W
f10	Height, in meters	±0-9999.9999
f11	Antenna height, in meters	0-6.5535
*cc	Checksum	*00-*FF

## Examples

\$PASHQ,PSP

\$PASHR,,PSP,0,2,311,0,0,4717.959483,N,00130.500968,W,70.229,0.000\*61

**Related Commands** \$PASHS,NME  
\$PASHQ,RSP

## PTT: PPS Time Tag

---

**Function** This command asks for the PPS time tag message to be output.

**Command Format Syntax**  
\$PASHQ,PTT[\*cc]

**Response Format Syntax**  
\$PASHR,PTT,d1,m2\*cc

## Parameters

Parameter	Description	Range
d1	Day of week: • 1: Sunday • 7: Saturday	1-7
m2	GPS time tag in hours, minutes, seconds and fraction of a second (9 decimal places)	0-23:59:59.999999999
*cc	Checksum	*00-*FF

## Example

Enabling the receiver to output the PTT message on port A:

```
$PASHS,NME,PTT,A,ON
```

Generating the PPS time tag message on port A:

```
$PASHQ,PTT,A
```

```
$PASHR,PTT,6,20:41:02.0000000*2D
```

## Comments

- The response to this command will be sent out once, right after the next PPS pulse is generated.
- The response contains the GPS time at which the PPS pulse was sent, including the offset if an offset was set when the PPS pulse was enabled.
- Being set to a periodical output by the \$PASHS,NME,PTT command, this message is independent of the NMEA period. It is only linked to the PPS period.

## Automatic Output of PTT Messages

This is a reminder on how to output PTT messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,PTT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output PTT messages on port A at a rate of 2 seconds:

```
$PASHS,NME,GSA,A,ON,2
```

## QZS: QZSS Tracking Status

---

**Function** This command is used to read the current status of QZSS tracking.

**Command Format Syntax**  
\$PASHQ,QZS[\*cc]

**Parameters**

None.

**Response Format Syntax**  
\$PASHR,QZS,s1,USE,n(c2)\*cc

**Parameters**

Parameter	Description	Range
s	QZSS tracking status: • ON: QZSS satellites tracked and used • OFF: QZSS satellites not tracked	ON or OFF
s1	QZSS tracking status: • ON: Currently tracking and using QZSS satellites • OFF: Not tracking QZSS satellites	ON, OFF
n(c2)	Provides tracking status of each QZSS satellite: • Y: Tracking enabled for this satellite • N: Tracking disabled for this satellite  This status is provided n times, where n is the number of satellites in the QZSS constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Optional checksum	*00-*FF

**Example**

Reading QZSS tracking:

```
$PASHQ,QZS
$PASHR,QZS,ON,USE,YYYYY*1B
```

**Relevant Set Command** \$PASHS,QZS

## RCA: Reading the Received CMR Type 2 Attribute

---

**Function** This command queries the receiver for the received CMR Type 2 attribute.

**Command Format Syntax**  
 \$PASHQ,RCA[\*cc]

**Parameters**  
 None

**Response Format Syntax**  
 \$PASHR,RCA,d1,d2,d3,s4,s5,s6,s7\*cc

### Parameters

Parameter	Description	Range
d1	Differential processor number	0, 3
d2	Counter reporting changes to the RCA parameters. The counter is incremented by one every time the RCA message content changes. Counter rolls back to 0 after reaching value "15".	0, 15
d3	Base ID	0-4095
s4	Physical station name	
s5	Short station ID	
s6	COGO code	
s7	Long station ID/Number	
*cc	Checksum	*00-*FF

### Examples

\$PASHQ,RCA  
 \$PASHR,RCS,Y,0,G1235A13.25,1,1,0,P1\*2A

**Related Commands** \$PASHS,NME

## RCP,OWN: Receiver Name

---

**Function** This command is used to read the name assigned to the receiver.



**Command Format**    **Syntax**  
                           \$PASHQ,RCP,OWN[\*cc]

**Parameters**

None.

**Response format**    **Syntax**  
                           \$PASHR,RCP,OWN,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Receiver name	TRIMBLE MB-TWO
*cc	Optional checksum	*00-*FF

**Example**

\$PASHQ,RCP,OWN\*4C  
 \$PASHR,RCP,OWN,TRIMBLE MB-TWO\*64

## RCP,REF: Reference Receiver Name

---

**Function**    This command is used to query the receiver for the name assigned locally to the base receiver from which the differential stream is received.

**Command Format**    **Syntax**  
                           \$PASHQ,RCP,REF[\*cc]

**Parameters**

None.

**Response format**    **Syntax**  
                           \$PASHR,RCP,REF,s1\*cc

**Parameters**

Parameter	Description	Range
s1	Reference receiver name	
*cc	Optional checksum	*00-*FF

**Example**

\$PASHQ,RCP,REF\*4B

\$PASHR,RCP,REF,SPECTRA,0\*26

**Related Commands**    \$PASHS,RCP,REF (set command)

## RCS: Reading the Recording Status

---

**Function**    This command queries the receiver for its current data recording status.

**Command Format**    **Syntax**  
                          \$PASHQ,RCS[\*cc]

**Parameters**  
None

**Response Format**    **Syntax**  
                          \$PASHR,RCS,c1,d2,s3,f4,d5,d6,s7\*cc

## Parameters

Parameter	Description	Range
c1	Recording status: <ul style="list-style-type: none"> <li>• Y: Data recording in progress; receiver will keep on recording data after a power cycle.</li> <li>• N: No data recording in progress; after a power cycle, no recording will start either.</li> <li>• S: No data recording in progress, but receiver will start recording data after a power cycle.</li> <li>• R: Data recording in progress, but receiver will stop recording data after a power cycle.</li> </ul>	Y, N, S, R
d2	Memory where data file is recorded: <ul style="list-style-type: none"> <li>• 0: Internal memory</li> <li>• 2: USB memory</li> </ul>	0, 2
s3	Data filename	255 char. max.
f4	Recording rate, in seconds:	0.05-960
d5	Occupation type: <ul style="list-style-type: none"> <li>• 0: Static</li> <li>• 1: Quasi-static</li> <li>• 2: Dynamic</li> </ul>	0-2
d6	Occupation state: <ul style="list-style-type: none"> <li>• 0: In progress</li> <li>• 1: No occupation</li> </ul>	0-1
s7	Occupation name	255 char. max.
*cc	Checksum	*00-*FF

## Examples

Data recording in progress, file G1235A13.25 being written to internal memory, recording rate: 1 second, "P1" occupation in progress:

```
$PASHQ,RCS
$PASHR,RCS,Y,0,G1235A13.25,1,1,0,P1*2A
```

No data recording in progress:

```
$PASHQ,RCS
$PASHR,RCS,N,0,,0.1,2,1,*5F
```

**Related Commands**    \$PASHS,NME

## RDP,CHT: Radio Channel Table

---

**Function**    This command is used to read the radio channel settings.

**Command Format Syntax**

```
$PASHQ,RDP,CHT,c1[*cc]
```

**Parameters**

Parameter	Description	Range
c1	Serial port used to communicate with the radio	A, B, D
*cc	Optional checksum	*00-*FF

**Response Format Syntax**

```
$PASHR,RDP,CHT,s1,d2,n(d3,f4,f5)*cc
```

(Where n=d2)

Or, if the channel table does not exist: \$PASHR,RDP,CHT,s1,0

**Parameters**

Parameter	Description	Range
s1	Radio Model: <ul style="list-style-type: none"> <li>• NONE: No radio</li> <li>• PDL: Pacific Crest <ul style="list-style-type: none"> <li>– PDL HPB</li> <li>– PDL LPB</li> </ul> </li> <li>• ADL: Pacific Crest <ul style="list-style-type: none"> <li>– ADL Vantage</li> <li>– ADL Vantage Pro</li> </ul> </li> <li>• XDL: Pacific Crest <ul style="list-style-type: none"> <li>– XDL Rover</li> </ul> </li> </ul>	PDL, ADL, XDL, NONE
d2	Total number of available channels	0-16 (PDL) 0-32 (ADL or XDL)
d3	Channel index	0-15 (PDL) 1-32 (ADL or XDL)
f4	Receive frequency	410-470 MHz
f5	Transmit frequency	410-470 MHz
*cc	Checksum	*00-*FF

**Comments**

The number of (d3,f4,f5) data sets (n) in the response line is equal to the number of channels (d2).

**Examples**

```
$PASHQ,RDP,CHT,D
```

```
$PASHR,RDP,CHT,PDL,7,0,446.7750,446.7750,1,444.1000,444.1000,2,445.1000,445.1000,3,446.1000,446.1000,4,447.1000,447.1000,5,448.1000,448.1000,6,449.1000,449.1000*35
```

```
$PASHQ,RDP,CHT,A
```

```
$PASHR,RDP,CHT,NONE,0*7B
```

**Related Commands**    \$PASHS,RDP,TYP  
                               \$PASHQ,RDP, PAR

## RDP,PAR: Radio Parameters

---

**Function**            This command allows you to query the radio settings relevant to the port used to communicate with the radio.

**Command Format**    **Syntax**  
                               \$PASHQ,RDP,PAR,c1[\*cc]

### Parameters

Parameter	Description	Range
c1	Serial port used to communicate with the radio	A, B, D
*cc	Optional checksum	*00-*FF

**Response Format**    **Syntax**  
                               \$PASHR,RDP,PAR,c1,s2,s3,c4,s5,c6,c7,s8,f9,f10,c11,s12,s13,,c15,c16,s17,  
                               s18,s19,d20,d21,s22,d23,d24\*cc

### Parameters

Parameter	Description	Range
c1	The port ID you specified in the command is replicated in this field	A, B, D
s2	Radio type: <ul style="list-style-type: none"> <li>• UNKNOWN: Auto-detection</li> <li>• NONE: No radio</li> <li>• PDL: Pacific Crest               <ul style="list-style-type: none"> <li>– PDL HPB</li> <li>– PDL LPB</li> </ul> </li> <li>• ADL: Pacific Crest               <ul style="list-style-type: none"> <li>• ADL Vantage</li> <li>• ADL Vantage Pro</li> </ul> </li> <li>• XDL: Pacific Crest               <ul style="list-style-type: none"> <li>• XDL Rover</li> </ul> </li> </ul>	UNKNOWN, PDL, ADL, XDL, NONE
s3	Radio state	ON, OFF
c4	Channel number	0-15 (PDL) 1-32 (ADL, XDL))
s5	Power management <ul style="list-style-type: none"> <li>• AUT: Automatic</li> <li>• MAN: Manual</li> </ul>	AUT, MAN

Parameter	Description	Range
c6	Protocol used: PDL: <ul style="list-style-type: none"> <li>• 0: Transparent</li> <li>• 1: TRIMTALK</li> <li>• 2: DSNP</li> </ul> ADL, XDL: <ul style="list-style-type: none"> <li>• 0: Transparent (w EOT time out)</li> <li>• 1: TRIMTALK 450S</li> <li>• 2: Not used</li> <li>• 3: SATEL</li> <li>• 4: TrimMarkII/IIe</li> <li>• 5: TT450S (HW)</li> <li>• 6: TRIMMARK3</li> <li>• 7: Transparent FST</li> <li>• 8: U-Link</li> </ul>	0-7
c7	Air link speed	4800, 8000, 9600, 16000, 19200
s8	Radio sensitivity	LOW, MED, HIGH, OFF
f9	Receive frequency, in MHz	410-470
f10	Transmit frequency, in MHz	410-470
c11	Channel spacing, in kHz	12.5, 25
s12	RF band, in MHz	410-430, 430-450, 450-470, 430-473
s13	Firmware version	
f14	Not used	
c15	Scrambler status <ul style="list-style-type: none"> <li>• 0: Off</li> <li>• 1: On</li> </ul>	0, 1
c16	Forward Error Correction status: <ul style="list-style-type: none"> <li>• 0: FEC Off</li> <li>• 1: Hamming FEC On</li> </ul>	0, 1
s17	RF output power: <ul style="list-style-type: none"> <li>• ADL Vantage: 100 or 500 mW, 1, 2 or 4 W</li> <li>• ADL Vantage Pro: 2 to 35 W</li> <li>• XDL: 500 mW or 2 W</li> </ul>	100 or 500 mW 1, 2, 4 or 35 W
s18	Maximum output power	100 or 500 mW 1, 2, 4 or 35 W
s19	Modulation format	4FSK, GMSK
d20	Model ID for ADL radios: <ul style="list-style-type: none"> <li>• 0: ADL RXO</li> <li>• 2: ADL Vantage</li> <li>• 3: ADL Vantage Pro</li> <li>• 4: XDL (XDL Micro)</li> </ul>	0, 2, 3, 4

Parameter	Description	Range
d21	Current output power (index) Output power for ADL Vantage: <ul style="list-style-type: none"> <li>• 0: 100 mW</li> <li>• 1: 500 mW</li> <li>• 2: 1 W</li> <li>• 3: 2 W</li> <li>• 4: 4 W</li> </ul> Output power for ADL Vantage Pro: <ul style="list-style-type: none"> <li>• 0: Level 1 (2 W)</li> <li>• 1: Level 2</li> <li>• 2: Level 3</li> <li>• 3: Level 4</li> <li>• 4: Level 5</li> </ul> See command \$PASHQ,RDP,PWR to read the current value of power in Watts. Output power for XDL Micro: <ul style="list-style-type: none"> <li>• 0: 500 mW</li> <li>• 2: 2 W</li> </ul>	0-4
s22	Repeater mode (XDL and ADL only)	OFF, ON
d23	Repeater number (for protocol 1, 4, 5 or 6 (see d5): <ul style="list-style-type: none"> <li>• 0: Base with one repeater</li> <li>• 1: Base with two repeaters</li> <li>• 2: Repeater 1</li> <li>• 3: Repeater 2</li> </ul>	0-3
d24	Repeater delay, in milliseconds (for protocol 0, 7 or 8 (see d5))	
*cc	Checksum	*00-*FF

### Example

If an internal PDL radio receiver is used:

**\$PASHQ,RDP,PAR,D**

\$PASHR,RDP,PAR,D,XDL,ON,4,AUT,2,4800,MED,447.1000,447.1000,25,0,4  
30-450,V02.53,,0,0,500mW,GMSK,4,0,OFF,0,0\*7A

### Comments

The command will be NAKed if the receiver has not been told on which port the radio is connected. Defining the port may be done using command \$PASHS,RDP,TYP.

### Related Commands

\$PASHS,RDP,PAR

\$PASHS,RDP,TYP

\$PASHS,RDP,PWR

## RDP,PWR: Reading Radio Type Used and Radiated Power

---

**Function** This command queries the receiver for the radio connected to the specified port.

The set of returned data is called “power table” describing the type of radio used, the number of channels and the radiated power.

If there’s no power table existing for the specified port, the response will be:

```
$PASHR,RDP,PWR,s1,0
```

### Command Format Syntax

```
$PASHQ,RDP,PWR,c1[*cc]
```

#### Parameters

Parameter	Description	Range
c1	Port for which you want the radio power table.	A, B, D
*cc	Optional checksum	*00-*FF

### Response Format Syntax

```
$PASHR,RDP,PWR,s1,d2,n(d3,f4)*cc
```

#### Parameters

Parameter	Description	Range
s1	Radio type: <ul style="list-style-type: none"> <li>NONE: No radio</li> <li>ADL: ADL Vantage or ADL Vantage Pro (Port A, B or D)</li> <li>XDL: XDL Rover</li> </ul>	ADL, XDL, NONE
d2	Total number of available channels	1-5 for ADL 1-3 for XDL
n	n designates the number of (d3, f4) pairs returned in the response; n= d2	n=d2
d3	Channel index	0-4 for ADL 0-2 for XDL
f4	Power, in watts	0.1-35
*cc	Checksum	*00-*FF

### Examples

Reading radio power table for port A:

```
$PASHQ,RDP,PWR,A
```

```
$PASHR,RDP,PWR,ADL,5,0,2,1,8,2,16,3,25,4,35*35
```

```
$PASHQ,RDP,PWR,A
```



\$PASHR,RDP,PWR,NONE,0\*7B

**Related Commands**    \$PASHS,RDP,TYP  
                               \$PASHS,RDP,PAR

## RDP,TYP: Radio Type Used

---

**Function**            This command is used to query the type of radio used on the specified port.

**Command Format**    **Syntax**  
                               \$PASHQ,RDP,TYP,c1[\*cc]

### Parameters

Parameter	Description	Range
c1	Serial port used to communicate with the radio	A, B, D
*cc	Optional checksum	*00-*FF

**Response Format**    **Syntax**  
                               \$PASHR,RDP,TYP,c1,s2\*cc

## Parameters

Parameter	Description	Range
c1	The port ID you specified in the command is replicated in this field	A, B, D
s2	Radio type: <ul style="list-style-type: none"> <li>• NONE: No radio</li> <li>• AUTO: Auto-detection</li> <li>• PDL: Pacific Crest <ul style="list-style-type: none"> <li>– PDL HPB</li> <li>– PDL LPB</li> </ul> </li> <li>• ADL: Pacific Crest <ul style="list-style-type: none"> <li>– ADL Vantage</li> <li>– ADL Vantage Pro</li> </ul> </li> <li>• XDL: Pacific Crest <ul style="list-style-type: none"> <li>– XDL Rover</li> </ul> </li> </ul>	PDL, ADL, XDL, NONE, AUTO
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,RDP,TYP,A
$PASHR,RDP,TYP,A,ADL*44

$PASHQ,RDP,TYP,D
$PASHR,RDP,TYP,D,XDL*5F
```

**Related Commands**    \$PASHS,RDP,TYP (set command)

## REC: Raw Data Recording Status

---

**Function**    This command allows you to read the current raw data recording status.

**Command Format**    **Syntax**  
   \$PASHQ,REC[\*cc]

**Response Format**    **Syntax**  
   \$PASHR,REC,c\*cc

## Parameters

Parameter	Description	Range
c	Control character: <ul style="list-style-type: none"> <li>• Y: Yes. Data recording in progress. Receiver will start recording data automatically when you next turn it on.</li> <li>• N: No. No data recording in progress. Receiver will not start recording data automatically when you next turn it on.</li> <li>• S: Stop. No data recording in progress but the receiver will start recording data automatically when you next turn it on.</li> <li>• R: Record. Data recording in progress but the receiver will not start recording data automatically when you next turn it on.</li> </ul>	Y, N, S, R
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,REC
$PASHR,REC,N*42
```

**Relevant Set Command**    \$PASHS,REC

## RFM: Ring File Memory

---

**Function**    This command returns the status of the ring file memory.

**Command Format Syntax**  
  \$PASHQ,RFM[\*cc]

### Parameters

None.

**Response Format Syntax**  
  \$PASHR,RFM,s1,d2,d3\*cc

## Parameters

Parameter	Description	Range
s1	Status of the ring file memory: <ul style="list-style-type: none"> <li>Y: Use of ring file memory enabled: The oldest files will be deleted automatically when the threshold of free memory space left is reached.</li> <li>N: Use of ring file memory disabled: The logging of raw data files will stop when there is no free space left in the memory used.</li> </ul>	Y, N
d2	Current value of free memory space left (in MBytes) for which the use of the ring file memory will come into play.	1-1024
d3	Not significant	
*cc	Checksum	*00-*FF

**Example**      \$PASHQ,RFM  
                   \$PASHR,RFM,Y,15,160\*6B

**Relevant Set Command**    \$PASHS,RFM

## RID: Receiver Identification

---

**Function**      This command allows you to read the receiver's identification parameters.

**Command Format Syntax**  
                   \$PASHQ,RID[\*cc]

### Parameters

None.

**Response Format Syntax**  
                   \$PASHR,RID,s1,s2,s3,s4,s5,s6,s7\*cc

### Parameters

Parameter	Description	Range
s1	Receiver name	MB2
s2	Firmware version: <ul style="list-style-type: none"> <li>Official version: x.y</li> <li>Non-official version: x.y.z</li> </ul>	"d.d" or "d.d.d"
s3	Version date	dd/mm/yyyy
s4	Expiration date (end of warranty period)	dd/mm/yyyy

Parameter	Description	Range
s5	Receiver options. When an option is valid, a letter is displayed, else a dash is displayed. See \$PASHS,OPTION for the list of possible options.	
s6	Part number	
s7	Serial number	10-char. string
*cc	Checksum	*00-*FF

### Example

\$PASHQ,RID\*28

\$PASHR,RID,MB2,1,2,15/10/2013,15/02/2014,

@123456789NGOBXYQT2WZJKFILMUR,91564,5331M00010\*37

### Comments

With a non-official version, the receiver works normally during the 90 days following the reported version date. After this time has elapsed, only the following commands are accepted by the receiver:

- \$PASHQ,RID
- \$PASHQ,VERSION
- \$PASHQ,ALR
- \$PASHQ,LOG

**Related Commands**    \$PASHQ,VERSION

## RMC: Recommended Minimum Specific GNSS Data

---

**Function**    This command is used to output an RMC message containing the last computed position as well as navigation-related data.

**Command Format**    **Syntax**  
                           \$PASHQ,RMC[\*cc]

**Response Format**    **Syntax**  
                           \$GPRMC,m1,c2,m3,c4,m5,c6,f7,f8,d9,f10,c11,c12\*cc

### Parameters

Parameter	Description	Range
m1	Current UTC time of position (hhmmss.ss)	000000.00-235959.99

Parameter	Description	Range
c2	Status <ul style="list-style-type: none"> <li>• A: Data valid</li> <li>• V: Navigation receiver warning</li> </ul>	A, V
m3	Latitude of position (ddmm.mmmmm)	0-90 0-59.999999
c4	Direction of latitude	N, S
m5	Longitude of position (dddmm.mmmmm)	0-180 0-59.999999
c6	Direction of longitude	E,W
f7	Speed Over Ground, in knots	000.0-999.9
f8	Course Over Ground, in degrees (true)	000.0-359.9
d9	Date (ddmmyy)	010100-311299
f10	Magnetic variation, in degrees	0.00-99.9
c11	Direction of variation	E, W
c12	Mode indicator: <ul style="list-style-type: none"> <li>• A: Autonomous mode</li> <li>• D: Differential mode</li> <li>• N: Data not valid</li> </ul>	A, D, N
*cc	Checksum	*00.*FF

**Example****\$PASHQ,RMC**

```
$GPRMC,160324.50,A,4717.959275,N,00130.500805,W,0.0,0.0,250208,1.9,
W,A*3D
```

**Related Commands****\$PASHS,NME****Automatic Output of RMC Messages**

This is a reminder on how to output RMC messages at regular intervals of time: Use the **\$PASHS,NME** command with the syntax below:

```
$PASHS,NME,RMC,<port_ID>,ON,<Rate>
```

For more details on the **\$PASHS,NME** command, refer to the *Set Command Library* Chapter.

As an example, the command below will output RMC messages on port A at a rate of 0.5 second:

```
$PASHS,NME,RMC,A,ON,0.5
```

## ROT: Rate of Turn

---

**Function** This command allows you to output the current value of rate of turn.

**Command Format Syntax**  
**\$PASHQ,ROT[\*cc]**

**Response Format Syntax**  
**\$--ROT,f,c\*cc**

### Parameters

Parameter	Description	Range
f	Rate of turn, in degrees/minute.	
C	Rate of turn status: <ul style="list-style-type: none"> <li>• A: Data valid</li> <li>• V: Data invalid</li> </ul>	A, V
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,ROT
$GNROT,123.56,A*61
```

### Related Command

\$PASHS,NME

### Automatic Output of ROT Messages

This is a reminder on how to output ROT messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,ROT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output ROT messages on port A at a rate of 0.5 second:

```
$PASHS,NME,ROT,A,ON,0.5
```

## RSP: Reading the Position of the Reference Station Received

---

**Function** This command queries the receiver for the position of the reference station received.

**Command Format Syntax**  
**\$PASHQ,RSP[\*cc]**

**Parameters**  
 None

**Response Format Syntax**  
**\$PASHR,RSP,d1,d2,d3,d4,d5,m6,c7,m8,c9,f10,f11\*cc**

### Parameters

Parameter	Description	Range
d1	Differential processor number	0, 3
d2	Counter reporting changes to the RSP parameters. The counter is incremented by one every time the PSP message content changes. Counter rolls back to 0 after reaching value "15".	0, 15
d3	Reference station ID	0-4095
d4	Reference station type: <ul style="list-style-type: none"> <li>• 0: Physical base station</li> <li>• 1: Virtual base station</li> </ul>	
d5	Position tagging: <ul style="list-style-type: none"> <li>• 0: ARP</li> <li>• 1: L1 phase center</li> </ul>	0, 1
m6	Latitude in degrees, decimal minutes (ddmm.mmmmm).	0, 90
c7	North (N) or South (S)	N, S
m8	Longitude in degrees, decimal minutes (dddmm.mmmmm).	0-180
c9	East (E) or West (W)	E, W
f10	Height, in meters	±0-9999.9999
f11	Antenna height, in meters	0-6.5535
*cc	Checksum	*00-*FF

### Examples

**\$PASHQ,RSP**  
**\$PASHR,RSP,0,2,811,1,0,4717.959483,N,00130.500968,W,70.229,0.000\*61**

**Related Commands** \$PASHS,NME  
 \$PASHQ,PSP



## RTK: Reading RTK Processing Status

---

**Function** This command is used to read the status of the currently defined RTK processing.

**Command Format Syntax**  
**\$PASHQ,RTK[\*cc]**

**Response Format Syntax**  
**\$PASHR[,d0],RTK,s1,d2[,d3[,d4]],STI,s5\*cc**

### Parameters

Parameter	Description	Range
d0	Sensor ID	1, 2
s1	RTK processing status (ON/OFF)	ON, OFF
d2	Number of first baseline used in the RTK processing.	1-3
d3	Number of second baseline used in the RTK processing.	1-3
d4	Number of third baseline used in the RTK processing.	1-3
s5	Desired station ID (as entered using command \$PASHS,RTK,STI)	AUT, 0-4095
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,RTK
$PASHR,1,RTK,ON,1,3,2,AUT*79
```

### Comment

In SOLO mode, the response is in the form:

```
$PASHR,1,RTK...
```

In DUO mode, the response consists of two lines, and is in the form:

```
$PASHR,1,RTK...
$PASHR,2,RTK...
```

## RTX: Reading RTX Processing Status

---

**Function** This command is used to read the status of the currently defined RTX processing.

**Command Format**    **Syntax**  
                          \$PASHQ,RTX[\*cc]

**Response Format**    **Syntax**  
                          \$PASHR,RTX,SRC,s1,MOD,s2,DTM,s3,s4,s5,s6,s7,RAM,s8\*cc

## Parameters

Parameter	Description	Range
s1	RTX corrections source: <ul style="list-style-type: none"> <li>• AUT: Corrections data may be received via either L-band or one of the board's communication ports.</li> <li>• LBN: RTX engine forced to use only the corrections data received via the L-band channel only.</li> <li>• IP: RTX engine forced to use only the corrections data received via its ports only.</li> <li>• OFF: RTX processing inactive</li> </ul>	AUT, LBN, IP, OFF
s2	RTX corrections service used: <ul style="list-style-type: none"> <li>• OFF: No RTX position computation, only subscribing data (BSM) are decoded.</li> <li>• CPT: CenterPoint RTX.</li> <li>• RPT: RangePoint RTX.</li> <li>• VPT: ViewPoint RTX</li> <li>• FPT: FieldPoint RTX</li> </ul>	OFF, CPT, RPT
s3	Indicates whether a datum transformation is used (ON) or not (OFF).	ON, OFF
s4	Name of datum used; See \$PASHS,RTX,DTM for datum name list.	
s5	Name of tectonic plate used; See \$PASHS,RTX,DTM for plate name list.	
s6	If s4=AUTO, firmware-set datum name, depending on position computed, otherwise see s6=s4. s6 empty: RTX not used.	
s7	If s5=AUTO, firmware-set tectonic plate name, depending on position computed, otherwise see s7=s5. s7 empty: RTX not used.	
s8	RAM (OFF)	OFF
*cc	Checksum	*00.*FF

### Example

\$PASHQ,RTX

\$PASHR,RTX,SRC,AUT,MOD,CPT,DTM,ON,ETRS89,EURO,ETRS89,EURO,OFF\*36

### Related Commands

\$PASHS,RTX,SRC

\$PASHS,RTX,MOD

## SAT: Satellites Status (Obsolete Command)

---

With the increasing number of available constellations, it has become more and more questionable to maintain a single message that would show, in a compact and clear form, all the satellites tracked by the board.

For this reason, and although the board can still respond to the \$PASHQ,SAT command, it was decided not to maintain this command and to suggest users to prefer the use of the following commands, each of them being specific to a constellation:

- \$PASHQ,SGP: GPS satellites
- \$PASHQ,SGL: GLONASS satellites
- \$PASHQ,SGA: Galileo satellites
- \$PASHQ,SSB: SBAS satellites
- \$PASHQ,SQZ: QZSS satellites
- \$PASHQ,SBD: BeiDou satellites

Note that because the SAT message has been made obsolete for quite some time, it does no longer give an accurate report of all the satellites tracked by the board. Its description is however given below, again with no guarantee that the returned information is accurate.

**Command Format**    **Syntax**  
                           \$PASHQ,SAT[\*cc]

**Response Format**    **Syntax**  
                           \$PASHR,SAT,d1,n(d2,d3,d4,f5,c6)\*cc

## Parameters

Parameter	Description	Range
d1	Number of satellites locked	1-57
d2	SV PRN number	1-32: GPS 33-51: SBAS 65-88: GLONASS 97-126: GALILEO 193-197: QZSS
d3	SV azimuth, in degrees	0-359
d4	SV elevation angle, in degrees	0-90
f5	SV signal-noise ratio, in dB.Hz	30.0-60.0
c6	SV used in computation or not • U: SV used • -: SV not used	U, -
*cc	Checksum	*00-*FF

The GPS PRN number is d2.

The EGNOS PRN number is d2 plus 87.

The GLONASS slot number is d2 minus 64.

The GALILEO PRN number is d2 minus 96.

The QZSS PRN number is d2 minus 192.

### Example

#### \$PASHQ,SAT

```
$PASHR,SAT,13,20,092,32,44.0,U,13,206,78,50.0,U,23,056,55,48.0,U,33,19
8,34,44.0,-,17,218,13,42.0,U,25,152,34,38.0,U,04,276,65,50.0,U,02,308,31,
48.0,U,77,052,37,48.0,U,84,294,33,48.0,U,83,234,23,48.0,U,78,124,42,46.0,
U,68,034,65,48.0,U*35
```

**See also** \$PASHS,NME

## Automatic Output of SAT Messages

Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,SAT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SAT messages on port A at a rate of 60 seconds:

```
$PASHS,NME,SAT,A,ON,60
```

## SBA: SBAS Tracking Status

---

**Function** This command is used to query the SBAS tracking status.

**Command Format Syntax**  
 \$PASHQ,SBA[\*cc]

**Response Format Syntax**  
 \$PASHR,SBA,s1,USE,n(c2)\*cc

### Parameters

Parameter	Description	Range
s1	SBAS tracking status: <ul style="list-style-type: none"> <li>ON: Currently tracking and using SBAS satellites</li> <li>OFF: Not tracking SBAS satellites</li> </ul>	ON, OFF
n(c2)	Provides tracking status of each SBAS satellite: <ul style="list-style-type: none"> <li>Y: Tracking enabled for this satellite</li> <li>N: Tracking disabled for this satellite</li> </ul> This status is provided n times, where n is the number of satellites in the SBAS constellation. See also the \$PASHS,<GNS>,USE command.	Y, N
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,SBA
$PASHR,SBA,ON,USE,YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
YYYYYYYYYY*4A
```

**Related Commands** \$PASHS,SBA (set command)

## SBD: BeiDou Satellites Status

---

**Function** This command is used to see which BeiDou satellites the receiver is currently tracking.

**Command Format Syntax**  
 \$PASHQ,SBD[\*cc]

**Parameters**

None

**Response Format Syntax**

\$PASHR,SBD,d1,n(d2,d3,d4,f5,f6,f7,c8,c9)\*cc

**Parameters**

Parameter	Description	Range
d1	Number of visible satellites	0-36
s2	Satellite PRN number	1-36
d3	Satellite azimuth, in degrees	0-359
d4	Satellite elevation, in degrees	0-90
f5	Satellite B1 signal/noise in dB.Hz	0.0-60.0
f6	Satellite B2 signal/noise in dB.Hz:	0.0-60.0
f7	(empty)	
c8	Satellite usage status	See NOTE below
c9	Satellite correcting status	See NOTE below
*cc	Checksum	*00-*FF

f5-f7 are empty if the corresponding signal is not tracked.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

**Example**

```
$PASHQ,SBD
$PASHR,SBD,13,20,092,32,44.0,35.0,,2,4,...
```

**Related Commands** \$PASHS,NME

**SGA: GALILEO Satellites Status**

**Function** This command is used to read the status of each GALILEO satellite received.

**Command Format Syntax**  
\$PASHQ,SGA[\*cc]

**Response Format Syntax**  
\$PASHR,SGA,d1,n(d2,d3,d4,f5,,f7,c8,c9)\*cc

## Parameters

Parameter	Description	Range
d1	Number of satellites locked	0-36
d2	SV PRN number	1-36
d3	SV azimuth in degrees	0-359
d4	SV elevation angle in degrees	0-90
f5	SV E1 signal/noise in dB.Hz	0.0-60.0
f6	SV E5b signal/noise in dB.Hz	0.0-60.0
f7	SV E5a signal/noise in dB.Hz (not tracked)	0.0-60.0
c8	Satellite usage status (see NOTE below)	
c9	Satellite correcting status (see NOTE below)	
*cc	Checksum	*00-*FF

Fields f5 to f7 are empty if the corresponding signal is not tracked.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

### Example

**\$PASHQ,SGA**

\$PASHR,SGA,5,12,303,22,40.4,38.7,,?,N,18,,,47.7,44.1,,?,N,19,199,21,39.8,37.8,,?,N,20,,,42.1,,,?,N,26,122,49,48.4,45.8,,?,N\*6C

### Related Commands

**\$PASHS,NME**

### Automatic Output of SGA Messages

This is a reminder on how to output SGA messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

\$PASHS,NME,SGA,<port\_ID>,ON,<Rate>

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SGA messages on port A at a rate of 10 seconds:

**\$PASHS,NME,SGA,A,ON,10**

## SGL: GLONASS Satellites Status

---

### Function

This command is used to read the status of each GLONASS satellite received.



**Command Format Syntax**  
**\$PASHQ,SGL[\*cc]**

**Response Format Syntax**  
**\$PASHR,SGL,d1,n(d2,d3,d4,f5,f6,,c8,c9)\*cc**

### Parameters

Parameter	Description	Range
d1	Number of visible satellites	0-24
d2	SV PRN number	1-24
d3	SV azimuth in degrees	0-359
d4	SV elevation angle in degrees	0-90
f5	SV L1 signal/noise in dB.Hz	0.0-60.0
f6	SV L2 signal/noise in dB.Hz	0.0-60.0
f7	Empty field	
d8	Satellite usage status (see NOTE below)	
d9	Satellite correcting status (see NOTE below)	
*cc	Checksum	*00-*FF

Fields f5 to f7 are empty if the corresponding signal is not tracked.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

**Example**  
**\$PASHQ,SGL**  
**\$PASHR,SGL,13,20,092,32,44.0,35.0,,2,4,...**

**Related Commands** **\$PASHS,NME**

### Automatic Output of SGL Messages

This is a reminder on how to output SGL messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

**\$PASHS,NME,SGL,<port\_ID>,ON,<Rate>**

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SGL messages on port A at a rate of 10 seconds:

**\$PASHS,NME,SGL,A,ON,10**

## SGP: GPS Satellites Status

---

**Function** This command is used to read the status of each GPS satellite received.

**Command Format Syntax**  
**\$PASHQ,SGP[\*cc]**

**Response Format Syntax**  
**\$PASHR,SGP,d1,n(d2,d3,d4,f5,f6,f7,c8,c9)\*cc**

### Parameters

Parameter	Description	Range
d1	Number of visible satellites	0-63
d2	SV PRN number	1-63
d3	SV azimuth in degrees	0-359
d4	SV elevation angle in degrees	0-90
f5	SV L1 signal/noise in dB.Hz	0.0-60.0
f6	SV L2 signal/noise in dB.Hz	0.0-60.0
f7	Reserved	
c8	Satellite usage status (see NOTE below)	
c9	Satellite correcting status (see NOTE below)	
*cc	Checksum	*00-*FF

Fields f5-f7 are empty if the corresponding signal is not tracked.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

**Example** **\$PASHQ,SGP**  
**\$PASHR,SGP,13,20,092,32,44.0,35.0,,2,4,...**

**Related Commands** **\$PASHS,NME**

**Automatic Output of SGP Messages** This is a reminder on how to output SGP messages at regular intervals of time: Use the **\$PASHS,NME** command with the syntax below:

**\$PASHS,NME,SGP,<port\_ID>,ON,<Rate>**

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SGP messages on port A at a rate of 10 seconds:

```
$PASHS,NME,SGP,A,ON,10
```

## SIT: Site Name

---

**Function** This command is used to read the name of the site on which data is currently being logged.

**Command Format Syntax**  
\$PASHQ,SIT[\*cc]

**Response Format Syntax**  
\$PASHR,SIT,s\*cc

### Parameters

Parameter	Description	Range
s	Site name	4 characters max.
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,SIT
$PASHR,SIT,SITE*1D
```

**Related Commands** \$PASHS,SIT

## SLB: L-Band Satellites Status

---

**Function** This command is used to read the status of each L-Band satellite received.

**Command Format Syntax**  
\$PASHQ,SLB[\*cc]

**Response Format Syntax**

\$PASHR,SLB,d1,n(d2,d3,d4,d5,f6)\*cc

**Parameters**

Parameter	Description	Range
d1	Number of visible satellites	0-7
d2	L-Band satellite number See \$PASHS,LBN,USE for satellite names	1-7
d3	Continuous tracking interval, in seconds	
d4	SV azimuth angle, in degrees	0-359
d5	SV elevation angle, in degrees	0-90
f6	SV signal/noise in dB.Hz	0.0-60.0
*cc	Checksum	*00-*FF

**Example**

```
$PASHQ,SLB
$PASHR,SLB,1,1,2356,092,32,44.0*7D
```

**Related Commands** \$PASHS,NME**Automatic Output of SLB Messages**

This is a reminder on how to output SLB messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,SLB,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SLB messages on port A at a rate of 10 seconds:

```
$PASHS,NME,SLB,A,ON,10
```

**SNS: M-Sensor Setting**

---

**Function** This command is used to query the receiver for the M-Sensor setting.

**Command Format Syntax**  
\$PASHQ,SNS[\*cc]

**Response Format Syntax**  
\$PASHR,SNS,s1,d2\*cc

## Parameters

Parameter	Description	Range
s1	M-Sensor current setting: <ul style="list-style-type: none"> <li>SOL: SOLO mode (one sensor)</li> <li>DUO: DUO mode (two sensors)</li> </ul>	SOL, DUO
d2	Index specifying antenna/sensor configuration (default: 0). See table below.	0-2
*cc	Checksum	*00-*FF

d2	s1=SOL	s1=DUO
0 or blank	<ul style="list-style-type: none"> <li>L1/L2 GNSS sensor selects antenna connector automatically.</li> <li>L-band sensor can be enabled only for 1st antenna input by command \$PASHS,GLB.</li> <li>L-band sensor cannot start, if GNSS sensor connected to 2nd antenna.</li> </ul>	<ul style="list-style-type: none"> <li>One L1/L2 GNSS sensor serves antenna connector #1, other L1-only GNSS sensor serves antenna connector #2.</li> <li>L-band sensor can be enabled only for 1st antenna input by command \$PASHS,GLB.</li> </ul>
1	<ul style="list-style-type: none"> <li>L1/L2 GNSS sensor always serves antenna connector #1.</li> <li>L-band sensor can be enabled only for 1st antenna input by command \$PASHS,GLB.</li> </ul>	<ul style="list-style-type: none"> <li>One L1/L2 GNSS sensor serves antenna connector #1, other L1-only GNSS sensor serves antenna connector #2.</li> <li>L-band sensor can be enabled only for 1st antenna input by command \$PASHS,GLB.</li> </ul>
2	<ul style="list-style-type: none"> <li>L1/L2 GNSS sensor always serves antenna connector #2.</li> <li>L-band sensor cannot start.</li> </ul>	<ul style="list-style-type: none"> <li>One L1/L2 GNSS sensor serves antenna connector #1, other similar L1/L2 GNSS sensor serves antenna connector #2.</li> <li>L-band sensor cannot start.</li> </ul>

### Example

```
$PASHQ,SNS
$PASHR,SNS,SOL,0*46
```

## SQZ: QZSS Satellites Status

---

**Function** This command is used to read the status of each QZSS satellite received.

**Command Format Syntax**  
 \$PASHQ,SQZ[\*cc]

## Response Format Syntax

```
$PASHR,SQZ,d1,n(d2,d3,d4,f5,f6,f7,c8,c9)*cc
```

### Parameters

Parameter	Description	Range
d1	Number of visible satellites	0-5
d2	SV PRN number	1-5
d3	SV azimuth in degrees	0-359
d4	SV elevation angle in degrees	0-90
f5	SV L1 signal/noise in dB.Hz	0.0-60.0
f6	SV L2 signal/noise in dB.Hz	0.0-60.0
f7	Reserved	
c8	Satellite usage status (see NOTE below)	
c9	Satellite correcting status (see NOTE below)	
*cc	Checksum	*00-*FF

Fields f5-f7 are empty if the corresponding signal is not tracked. QZSS L1 SAIF tracking status is not reported here.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

### Example

```
$PASHQ,SQZ
$PASHR,SQZ,13,20,092,32,44.0,35.0,,2,4,...
```

### Related Commands

```
$PASHS,NME
$PASHQ,SSB (for QZSS L1 SAIF tracking status)
```

### Automatic Output of SQZ Messages

This is a reminder on how to output SQZ messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,SQZ,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SQZ messages on port A at a rate of 10 seconds:

```
$PASHS,NME,SQZ,A,ON,10
```

## SSB: SBAS Satellites Status

---

**Function** This command is used to read the status of each SBAS satellite received.

**Command Format Syntax**  
**\$PASHQ,SSB[\*cc]**

**Response Format Syntax**  
**\$PASHR,SSB,d1,n(d2,d3,d4,f5,f6,f7,c8,c9)\*cc**

### Parameters

Parameter	Description	Range
d1	Number of visible satellites	1-39
d2	SV PRN number	1-39
d3	SV azimuth in degrees	0-359
d4	SV elevation angle in degrees	0-90
f5	SV L1 signal/noise in dB.Hz	0.0-60.0
f6	Empty field	
f7	Reserved	
c8	Satellite usage status (see NOTE below)	
c9	Satellite correcting status (see NOTE below)	
*cc	Checksum	*00-*FF

Fields f5-f7 are empty if the corresponding signal is not tracked. QZSS L1 SAIF tracking status is reported here and corresponds to PRN numbers 40-44.

NOTE: For more information on satellite usage status and satellite correcting status, refer to *Satellite Status on page 516*.

**Example** **\$PASHQ,SSB**  
**\$PASHR,SSB,13,20,092,32,44.0,35.0,,2,4,...**

**See also** **\$PASHS,NME**

**Automatic Output of SSB Messages** This is a reminder on how to output SSB messages at regular intervals of time: Use the **\$PASHS,NME** command with the syntax below:

**\$PASHS,NME,SSB,<port\_ID>,ON,<Rate>**

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output SSB messages on port A at a rate of 10 seconds:

```
$PASHS,NME,SSB,A,ON,10
```

## STI: Station ID

---

**Function** This command is used to query the receiver for the station ID it transmits to the rover through the corrections message.

**Command Format Syntax**  
\$PASHQ,[d0,]STI[\*cc]

**Response Format Syntax**  
\$PASHR,d0,STI,d1\*cc

### Parameters

Parameter	Description	Range
d0	Interrogated sensor number: <ul style="list-style-type: none"> <li>• 1: Primary antenna sensor</li> <li>• 2: Secondary antenna sensor</li> <li>• Blank: Both</li> </ul>	1, 2, blank
d1	Station ID	0-1023 (RTCM 2.3) 0-4095 (RTCM 3.x)/ATOM 0-31 (CMR & CMR+)
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,STI
$PASHR,1,STI,817*35
$PASHR,2,STI,880*38
```

**Relevant Set Command** \$PASHS,STI



## TCP: TCP/IP Server Settings

---

**Function** This command is used to query the settings of the TCP/IP server.

**Command Format Syntax**  
**\$PASHQ,TCP[,c6][\*cc]**

**Response Format Syntax**  
 \$\$PASHR,TCP,MOD=s1,LGN=s2,PWD=s3,ADD=s4  
 ,PRT=d5,IPP=c6[,RIP=s7]\*cc

### Parameters

Parameter	Description	Range
MOD=s1	<ul style="list-style-type: none"> <li>• 0: Disabled</li> </ul> TCP/IP connection mode: <ul style="list-style-type: none"> <li>• 1: Input/output with authentication</li> <li>• 2: Input/output without authentication (default)</li> <li>• 3: Output only. Allows multiple connections</li> </ul> UDP/IP connection mode: <ul style="list-style-type: none"> <li>• 4: Broadcast transmit (output only)</li> <li>• 5: Broadcast receive (input only)</li> <li>• 6: Transmit to remote IP address (output only)</li> <li>• 7: Received from remote IP address (input only)</li> </ul>	0-7
LGN=s2	Login	32 char. max.
PWD=s3	Password	32 char. max.
ADD=s4	IP address	0.0.0.0-255.255.255.255
PRT=d5	Port number	0-65535
IPP=c6	Port name	I, F
RIP=s7	Remote IP address (returned when UPD/IP mode is used)	32 char. max.
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,TCP
$PASHR,TCP,MOD=1,LGN=Trimble,PWD=u6huz8,ADD=192.34.76.1,
PRT=8888,IPP=F*2C
```

**Related Commands** \$PASHS,TCP,PAR

## TEM: Die Temperature

---

**Function** This command is used to query the receiver for the die (or junction) temperature. The reading does not reflect the case temperature or ambient temperature. It is not exactly the temperature of the entire die either, but rather the temperature measured by the sensor located in the vicinity of the die (the temperature being not uniformly distributed).

**Command Format Syntax**  
**\$PASHQ,TEM[\*cc]**

**Response Format Syntax**  
**\$PASHR,TEM,d1\*cc**

### Parameters

Parameter	Description	Range
d1	Die temperature, in degrees C	
*cc	Checksum	*00-*FF

### Example

```
$PASHQ,TEM
$PASHR,TEM,54*05
```

### Automatic Output of TEM Messages

This is a reminder on how to output TEM messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,TEM,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output TEM messages on port A at a rate of 10 seconds:

```
$PASHS,NME,TEM,A,ON,10
```

## THS: True Heading and Status

---

**Function** This command is used to output the NMEA “THS” message providing the last computed value of true heading.

**Command Format Syntax**  
**\$PASHQ,THS[\*cc]**

**Response Format Syntax**  
**\$PASHR,THS,f1,c2\*cc**

### Parameters

Parameter	Description	Range
f1	Last computed heading value, in degrees (true).	000.00-359.99
c2	Solution status: <ul style="list-style-type: none"> <li>• A: Autonomous</li> <li>• E: Estimated (dead reckoning)</li> <li>• M: Manual input</li> <li>• S: Simulator</li> <li>• V: Data not valid (including standby)</li> </ul>	A, E, M, S, V
*cc	Checksum	*00-*FF

**Example**  
**\$PASHQ,THS**  
**\$GPTH,123.56,A\*61**

**Related Commands** \$PASHS,NME

**Automatic Output of THS Messages** This is a reminder on how to output THS messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

**\$PASHS,NME,THS,<port\_ID>,ON,<Rate>**

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output THS messages on port A at a rate of 10 seconds:

**\$PASHS,NME,THS,A,ON,10**

## TOP: Type of Position Delivered

---

**Function** This command is used to read the type of position the receiver is allowed to deliver in all NMEA and ATOM PVT output messages.

**Command Format Syntax**  
 \$PASHQ, TOP[\*cc]

**Response Format Syntax**  
 \$PASHR, TOP, s1\*cc

### Parameters

Parameter	Description	Range
s1	Allowed type of position delivered by the receiver: <ul style="list-style-type: none"> <li>• RTK: Up to RTK</li> <li>• PPP: Up to PPP</li> <li>• DIF: Up to differential (RTCM)</li> <li>• SBA: Up to differential (SBAS)</li> <li>• ALO: Standalone only (autonomous)</li> </ul>	RTK, PPP, DIF, SBA, ALO
*cc	Checksum	*00-*FF

**Example**  
 \$PASHQ, TOP  
 \$PASHR, TOP, RTK\*4D

**Related Commands** \$PASHS, TOP (set command)

## TRS: Reading Information on TERIAsat

---

**Function** This message provides information on the TERIAsat service.

**Command Format Syntax**  
 \$PASHQ, TRS[\*cc]

**Parameters**  
 None.

**Response Format Syntax**

\$PASHR,TRS,d1,s2,s3,d4,s5\*cc

**Parameters**

Parameter	Description	Range
d1	TERIASat installation status: • 0: Not installed • 1: Installed	0, 1
s2	Version number	
s3	Version date (dd/mm/yy)	
d4	License validity status: • 0: No valid license • 1: Valid license	0, 1
s5	License expiration date (dd/mm/yy)	
*cc	Optional checksum	*00-*FF

**Response Examples**

TERIASat not installed:

\$PASHR,TRS,0,,,0,

TERIASat installed but not licensed:

\$PASHR,TRS,1,1.05,08/10/2018,0,

TERIASat installed and license has expired:

\$PASHR,TRS,1,1.05,08/10/2018,0,01/08/2018

TERIASat installed and license is valid:

\$PASHR,TRS,1,1.05,08/10/2018,1,31/12/2018

**Related Commands** \$PASHS,PPP**TTT: Event Marker**

---

**Function** This message provides the GPS time of the external event. The time is provided with an accuracy of 1 microsecond

**Response Format Syntax**

\$PASHR,TTT,d1,m2\*cc

## Parameters

Parameter	Description	Range
d1	Day of week: • 1: Sunday • 7: Saturday	1-7
m2	GPS time tag in hours, minutes, seconds	0-23:59:59.9999999
*cc	Optional checksum	*00-*FF

## Example

```
$PASHR,TTT,3,18:01:33.1200417*AC
```

## Comments

- Issuing this output message is tied to the prior execution of the appropriate \$PASHS,NME,TTT command (see *NME: Enabling/Disabling NMEA Messages on page 233*), and the detection of a signal at the board's external event input. There is no query command associated with the output of the TTT message.
- This message is independent of the NMEA period. It can be output faster or slower than the NMEA period, depending on the period of the event.

## UDP: User-Defined Dynamic Model

---

**Function** This command is used to read the parameters of the user-defined dynamic model.

**Command Format Syntax**  
\$PASHQ,UDP[\*cc]

**Response Format Syntax**  
\$PASHR,UDP,f1,f2,f3,f4\*cc

## Parameters

Parameter	Description	Range	Default
f1	Maximum expected horizontal velocity, in m/s	0-100 000	100 000
f2	Maximum expected horizontal acceleration, in m/s <sup>2</sup>	0-100	100
f3	Maximum expected vertical velocity, in m/s	0-100 000	100 000
f4	Maximum expected vertical acceleration, in m/s <sup>2</sup>	0-100	100
*cc	Checksum	*00-*FF	

## Example

```
$PASHQ,UDP
$PASHR,UDP,10,1,2,0.5*1D
```

**Related Commands** \$PASHS,UDP (set command)  
\$PASHS,DYN

## UPL: FTP Server Providing Firmware Upgrades

---

**Function** This command is used to read the status and settings of the FTP server used to perform firmware upgrades.

**Command Format** **Syntax**  
\$PASHQ,UPL[\*cc]

## Parameters

None.

**Response format** **Syntax**  
\$PASHR,UPL,s1,s2,d3,d4,ADD=s5,PRT=d6,LGN=s7,PWD=s8,PTH=s9\*cc

## Parameters

Parameter	Description	Range
s1	FTP data transfer status: <ul style="list-style-type: none"> <li>NONE: No data transfer in progress</li> <li>GET: Firmware upgrade being downloaded from FTP</li> </ul>	NONE, GET
s2	Name of the file being transferred	255 char max.
d3	Size, in bytes, of the file being transferred	0-134217728

Parameter	Description	Range
d4	Percentage of the file transferred so far	0-100
ADD=s5	FTP server IP address or host name	
PRT=d6	FTP server port number	0-65535
LGN=s7	FTP server log in	32 char max.
PWD=s8	FTP server password	32 char max.
PTH=s9	Path used on FTP server to access the upgrade file	255 char max.
*cc	Optional checksum	*00-*FF

### Example

```
$PASHQ,UPL*3E
$PASHR,UPL,GET,pf800_upgrade_V227Ga21.tar.bz2,1769897,56,
ADD=ftp.ashtech.com,PRT=21,LGN=Ashtech,PWD=u6huz8,
PTH=/my folder*7D
```

**See Also** \$PASHS,UPL,PAR  
 \$PASHS,UPL,UPG  
 \$PASHQ,UPL,LST

## UPL,LOG: Editing the Firmware Upgrade Log File

---

**Function** This command is used to edit the firmware upgrade log file. This file logs all the actions performed during a firmware upgrade routine.

**Command Format** **Syntax**  
 \$PASHQ,UPL,LOG[,d][\*cc]

### Parameters

None.

**Response format** **Syntax**  
 The response is formatted as follows:

```
$PASHR,UPL,LOG
Starting script at <Day> <Month> <Time> UTC <Year>
Programming tool is /usr/local/bin/dataflash_tool
...
```

### Example

```
$PASHR,UPL,LOG
Starting upgrade script 1.0.6 at Wed Feb 19 16:07:59 UTC 2014 smod=1
Linux am37x-evm 2.6.37 #646 PREEMPT Tue Dec 3 18:26:43 RST 2013
armv7l GNU/Linux
```



```

rootfs on / type rootfs (rw)
/dev/root on / type romfs (ro,relatime)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw,relatime)
tmpfs on /var/volatile type tmpfs (rw,relatime,size=16384k)
tmpfs on /dev/shm type tmpfs (rw,relatime,mode=777)
tmpfs on /media/ram type tmpfs (rw,relatime,size=16384k)
ubi0:bin on /mnt/bin type ubifs (rw,relatime)
ubi1:conf on /mnt/config type ubifs (rw,relatime)
ubi2:log on /mnt/log type ubifs (rw,relatime)
ubi3:pvt on /mnt/pvt type ubifs (rw,relatime)
ubi4:card on /media/card type ubifs (rw,relatime)
/dev/mmcbk0p1 on /media/cf type vfat
(rw,relatime,fmask=0022,dmask=0022,codepage=cp437,iocharset=iso8859
1,shortname=mixed,errors=remount-ro)
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        43451      43451      0 100% /
tmpfs            16384         16   16368  0% /var/volatile
tmpfs            58092          0   58092  0% /dev/shm
tmpfs            16384         16   16368  0% /media/ram
ubi0:bin         23552      2436   19864  11% /mnt/bin
ubi1:conf        15856       76   14912  1% /mnt/config
ubi2:log         15856        32   14956  0% /mnt/log
ubi3:pvt         15856        68   14920  0% /mnt/pvt
ubi4:card        1501776      656  1496284  0% /media/card
/dev/mmcbk0p1    495168     21648  473520  4% /media/cf
Previous product version 1.0.42
-rwxr-xr-x 1 500 500 15952 Feb 6 16:18 Upgrade.sh
-rwxr--r-- 1 500 500 3 Feb 18 08:36 productVersion.txt
Checking Warranty at Wed Feb 19 16:08:00 UTC 2014
flist:Upgrade.sh
productVersion.txt
nwrite
upgrade.md5
MB1_upgrade_gnss.bz2
MB1_upgrade_appl.bz2
MB1_upgrade_pmu.bz2
MB1_upgrade_boot.bz2
MB1_upgrade_kern.bz2
MB1_upgrade_rfs.bz2
-rwxr-xr-x 1 500 500 15952 Feb 6 16:18 Upgrade.sh
-rwxr-xr-x 1 500 500 17465 Nov 25 16:01 nwrite
-rwxr--r-- 1 500 500 3 Feb 18 08:36 productVersion.txt
-rwxr--r-- 1 500 500 314876 Jan 30 15:24 u-boot.bin
-rwxr--r-- 1 500 500 45 Feb 18 08:36 u-boot.bin.md5
Upgrading Step 1/9
unpack sp4_rootfs to /media/cf
Unpack ret 0
-rwxr-xr-x 1 root root 44688384 Feb 18 08:36 sp4_rootfs

```

```

-rwxr-xr-x 1 root root 45 Feb 18 08:36 sp4_rootfs.md5
Orig hash 743136798392617e7cf21187ae70369a sp4_rootfs
File hash 743136798392617e7cf21187ae70369a sp4_rootfs
Upgrading Step 2/9
wr sp4_rootfs to /dev/mtdblock6
MTD total size: 8000000 bytes, MTD erase size: 40000 bytes
writing 44688384 bytes from /media/cf/sp4_rootfs to /dev/mtdblock6 (/dev
mtd6), offset 0x1000000, block size 0x40000
complete 44688384
md5sum 743136798392617e7cf21187ae70369a
Upgrading Step 3/9
unpack ulmage-sp4 to /media/cf
Unpack ret 0
-rwxr-xr-x 1 root root 2834628 Feb 18 08:36 ulmage-sp4
-rwxr-xr-x 1 root root 45 Feb 18 08:36 ulmage-sp4.md5
203aa4a2d531d3ecbbcbdea809e9664d ulmage-sp4
Orig hash 203aa4a2d531d3ecbbcbdea809e9664d ulmage-sp4
File hash 203aa4a2d531d3ecbbcbdea809e9664d ulmage-sp4
Upgrading Step 4/9
wr ulmage-sp4 to /dev/mtdblock6
MTD total size: 8000000 bytes, MTD erase size: 40000 bytes
writing 2834628 bytes from /media/cf/ulmage-sp4 to /dev/mtdblock6 (/dev
mtd6), offset 0x800000, block size 0x40000
block 0 offs 0x800000 0
block 1 offs 0x840000 0
block 2 offs 0x880000 0
block 3 offs 0x8c0000 0
block 4 offs 0x900000 0
block 5 offs 0x940000 0
etc.

```

## UPL,LST: Listing the Firmware Upgrades Available on FTP

---

**Function** This command is used to list the upgrade files and/or upgrade directories found on the FTP server.

**Command Format** **Syntax**  
**\$PASHQ,UPL,LST[,s][\*cc]**

## Parameters

Parameter	Description	Range
s	Path that extends the one defined with \$PASHS,UPL,PAR. If s is omitted, the command lists the content of the default directory (i.e. as defined with \$PASHS,UPL,PAR).	255 characters max.
*cc	Optional checksum	*00-*FF

## Response format Syntax

```
$PASHR,UPL,LST,d1,d2,s3,s4,d5,s6,s7*cc
```

## Parameters

Parameter	Description	Range
d1	Number of listed files or subdirectories	
d2	Index of file or subdirectory	
s3	Indicates whether the listed item is a file or a directory: • DIR: Directory • FIL: File	DIR,FIL
s4	Name of the file or subdirectory	255 characters max.
d5	Size, in bytes	0-134217728
s6	Date of creation (ddmmyyyy)	
s7	Time of creation (hhmmss)	000000-235959
*cc	Optional checksum	*00.*FF

## Example

```
$PASHQ,UPL,LST,  
$PASHR,UPL,LST,4,0,FIL,mb1_upgrade_v1.2.tar,1769897,14032009,  
130850*50  
$PASHR,UPL,LST,4,1,FIL,mb1_upgrade_v1.3.tar,1769876,10032009,  
110952*5A  
$PASHR,UPL,LST,4,2,FIL,mb1_upgrade_v1.4.tar,1769787,01032009,  
181856*52  
$PASHR,UPL,LST,4,3,DIR,my directory,1769787,01032009,181856*48
```

**See Also** \$PASHS,UPL,PAR  
\$PASHS,UPL,UPG

## VCR: Vector & Accuracy Data

---

**Function** This command is used to output a VCR message, whose content is similar to that of the VCT message. It can be

requested only when the receiver operated in relative mode (see \$PASHS,BRV).

Unlike VCT, VCR is a time-tagged message because it describes a vector connecting two moving receivers. Up to three different VCR messages can be generated from the same time tag and antenna ID (e.g. all baselines assigned to relative mode).

**Command Format**    **Syntax**  
                          \$PASHQ,VCR[\*cc]

**Response Format**    **Syntax**  
                          \$PASHR,VCR,d0,c1,d2,m3,f4,f5,f6,f7,f8,f9,f10,f11,f12,d13,c14\*cc

## Parameters

Parameter	Description	Range
d0	Baseline number (see \$PASHS,BRV)	1, 2, 3
c1	Baseline mode: <ul style="list-style-type: none"> <li>• 0: Invalid baseline</li> <li>• 1: Differential</li> <li>• 2: RTK float</li> <li>• 3: RTK fixed</li> <li>• 5: Other (dead reckoning, bad accuracy, or baseline computed just as the difference between standalone base position and standalone rover position)</li> </ul>	0-3
d2	Number of SVs used in baseline computation (L1 portion)	0-99
m3	UTC time (hhmmss.ss)	000000.00-235959.99
f4	First coordinate of delta antenna position, ECEF, in meters	±99999.999
f5	Second coordinate of delta antenna position, ECEF, in meters	±99999.999
f6	Third coordinate of delta antenna position, ECEF, in meters	±9999.999
f7	Standard deviation, first coordinate	99.999
f8	Standard deviation, second coordinate	99.999
f9	Standard deviation, third coordinate	99.999
f10	Correlation (half)	±9.999999
f11	Correlation (one third)	±9.999999
f12	Correlation (two third)	±9.999999
d13	Base station ID (same as GGA)	0-4095
c14	Baseline coordinate frame ID: <ul style="list-style-type: none"> <li>• 0: XYZ</li> <li>• 1: ENU centered on rover</li> <li>• 2: ENU centered on base</li> </ul>	0-2
*cc	Checksum	*00-*FF

NOTE: This message does not in any case complement position messages as its content cannot be related to any reported position.

### Example

```
$PASHQ,VCR
$PASHR,VCR,1,3,09,130924.00,-37.683,55.081,-17.925,
0.016,0.012,0.026,0.234765,0.098765,0.098763,0001*3B
```

**Related Commands**    \$PASHS,NME

## Automatic Output of VCR Messages

This is a reminder on how to output VCR messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,VCR,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output VCR messages on port A at a rate of 0.2 second:

```
$PASHS,NME,VCR,A,ON,0.2
```

## VCT: Vector & Accuracy Data

---

**Function** This command is used to output the VCT message, which is an extension of the legacy VEC message. VCT can be requested only in a mode set up by \$PASHS,RTK. The baseline vector is oriented from the base to the rover.

**Command Format** **Syntax**  
\$PASHQ,VCT[\*cc]

**Response Format** **Syntax**  
\$PASHR,VCT,c1,d2,m3,f4,f5,f6,f7,f8,f9,f10,f11,f12,d13,d14,d15\*cc

## Parameters

Parameter	Description	Range
c1	Baseline mode: <ul style="list-style-type: none"> <li>• 0: Invalid baseline</li> <li>• 1: Differential</li> <li>• 2: RTK float</li> <li>• 3: RTK fixed</li> <li>• 5: Other (dead reckoning, bad accuracy, or baseline is computed just as difference between standalone base position and standalone rover position)</li> </ul>	0-3, 5
d2	Number of SVs used in baseline computation	0-99
m3	UTC time (hhmmss.ss)	000000.00-235959.99
f4	Delta antenna position, ECEF 1st coordinate (in meters)	±99999.999
f5	Delta antenna position, ECEF 2nd coordinate (in meters)	±99999.999
f6	Delta antenna position, ECEF 3rd coordinate (in meters)	±9999.999
f7	1st coordinate standard deviation	99.999
f8	2nd coordinate standard deviation	99.999
f9	3rd coordinate standard deviation	99.999
f10	1st/2nd coord. correlation	±9.999999
f11	1st/3rd coord. correlation	±9.999999
f12	2nd/3rd coord. correlation	±9.999999
d13	Base station ID (same as in GGA)	0-4095
d14	Baseline coordinate frame ID: <ul style="list-style-type: none"> <li>• 0: XYZ</li> <li>• 1: ENU centered on rover</li> <li>• 2: ENU centered on base</li> </ul>	0-2
d15	Baseline number (see \$PASHS,BLN)	1-3
*cc	Checksum	*00-*FF

### Example

**\$PASHQ,VCT**

```
$PASHR,VCT,3,09,130924.00,-37.683,55.081,-17.925,0.016,0.012,
0.026,0.234765,0.098765,0.098763,0001,0,1*01
```

### Related Commands

\$PASHS,NME

### Automatic Output of VCT Messages

This is a reminder on how to output VCT messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,VCT,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output VCT messages on port A at a rate of 0.2 second:

```
$PASHS,NME,VCT,A,ON,0.2
```

## VEL: Velocity & Accuracy Message

---

**Function** This command allows you to output the velocity & accuracy message.

**Command Format Syntax**  
\$PASHQ,VEL[\*cc]

**Response Format Syntax**  
\$PASHR,VEL,f1,m2,f3,f4,f5,f6,f7,f8,d9\*cc

### Parameters

Parameter	Description	Range
f1	Reserved, set to 1. Means that velocity and accuracy correspond to local ENU (East, North, Up) system.	1
m2	Current UTC time of velocity fix (hhmmss.ss)	
f3	Easting velocity, in m/s	
f4	Northing velocity, in m/s	
f5	Vertical velocity, in m/s	
f6	Easting velocity RMS error, in mm/s	
f7	Northing velocity RMS error, in mm/s	
f8	Vertical velocity RMS error, in mm/s	
d9	Applied effective velocity smoothing interval, in ms (empty if unknown)	
*cc	Checksum	*00.*FF

**Comments** Velocity is not delivered in the message if it cannot be computed reliably. RMS errors can however provided in this case. Frozen RMS values of velocity for Easting, Northing and Vertical (respectively 500, 500, 100) indicate that velocities are equal to or greater than these values.



## Automatic Output of VEL Messages

This is a reminder on how to output VEL messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,VEL,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output VEL messages on port A at a rate of 0.5 second:

```
$PASHS,NME,VEL,A,ON,0.5
```

## VERSION: Firmware Version

---

**Function** This command is used to list the firmware versions installed in the receiver, including those of the modem and internal radio.

**Command Format** **Syntax**  
**\$PASHQ,VERSION[\*cc]**

**Response Format** **Syntax**  
 (Through an example)  
**\$PASHQ,VERSION**  
 VERSION: 1.0.40  
 VERSION DATE: 10/12/2013  
 EXPIRATION DATE: 10/12/2015  
 SL: 0.37 (API: 0.23)  
 PVT: LP60V27  
 DSP: LC60V27  
 XL1: 1.0  
 BL1: 1.21  
 OS2: 2.6.37 #646 PREEMPT 3/12/2013  
 XL2: 1.0  
 BL2: 1.21  
 HWID: 9dca23207931

**Related Commands** \$PASHQ,RID

## VLP: Receiver Validity Period

---

**Function** This command is used to query the receiver for its validity period, if any.

**Command Format Syntax**  
**\$PASHQ,VLP[\*cc]**

**Response Format Syntax**  
**\$PASHR,VLP,d1,s2[\*cc]**

### Parameters

Parameter	Description	Range
d1	Remaining number of days. Empty if no validity period defined.	0, 1-365
s2	Day when validity period expires (last day of validity). Empty if no validity period defined.	dd/mm/yyyy
*cc	Optional checksum	*00-*FF

### Examples

Validity period not defined:

```
$PASHQ,VLP  

$PASHR,VLP,,*3E
```

Validity period defined and still running:

```
$PASHQ,VLP  

$PASHR,VLP,10,15/12/2014*3F
```

Validity period defined and expired:

```
$PASHQ,VLP  

$PASHR,VLP,0,01/12/2014*0B
```

**Relevant Set Command** **\$PASHS,VLP**

## VTG: Course Over Ground and Ground Speed

---

**Function** This command is used to output a VTG message.

**Command Format Syntax**  
**\$PASHQ,VTG[\*cc]**

**Response Format Syntax**

\$GPVTG,f1,T,f2,M,f3,N,f4,K,c5\*cc

**Parameters**

Parameter	Description	Range
f1,T	COG (with respect to True North) T for "True" North: COG orientation	000.00-359.99
f2,M	COG (with respect to Magnetic North) M for "Magnetic" North: COG orientation	000.00-359.99
f3,N	SOG (Speed Over Ground) N for "knots": SOG unit	000.00-999.999
f4,K	SOG (Speed Over Ground) K for "km/hr": SOG unit	000.00-999-999
c5	Mode indicator: • A: Autonomous mode • D: Differential mode • N: Data not valid	A, D, N
*cc	Checksum	*00-*FF

**Example**

\$PASHQ,VTG

\$GPVTG,128.00,T,129.92,M,0.17,N,0.31,K,A\*2D

**Related Commands**

\$PASHS,NME

**Automatic Output of VTG Messages**

This is a reminder on how to output VTG messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

\$PASHS,NME,VTG,&lt;port\_ID&gt;,ON,&lt;Rate&gt;

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output VTG messages on port A at a rate of 0.5 second:

\$PASHS,NME,VTG,A,ON,0.5

**WEB: Web Server Control, Owner Data & Connection Profiles****Function**

This command returns the status of the web interface in a free-form format.

**Command Format    Syntax**

\$PASHQ,WEB[\*cc]

**Parameters**

None.

**Response format    Syntax**

Through an example:

```
$PASHQ,WEB*27
WEB INTERFACE: ON
HTTPD PORT: 80
COMPANY: Ashtech
ADMINISTRATOR NAME: Peter Smith
ADMINISTRATOR EMAIL: psmisth@ashtech.com
ADMINISTRATOR PHONE: 0228093838
ADMINISTRATOR LOGIN: smith
ADMINISTRATOR PASSWORD: 255kj631
SECURITY:
PRIVILEGES FOR ANONYMOUS ACCESS:
USER LOGIN: Andrew
USER PASSWORD: 25ml55
USER LOGIN: Yves
USER PASSWORD: 25ml55
```

**See Also**    \$PASHS,WEB,OWN  
\$PASHS,WEB,PAR  
\$PASHS,WEB,USR,ADD

## ZDA: Time & Date

---

**Function**    This command returns the receiver date & time.

**Command Format    Syntax**

\$PASHQ,ZDA[\*cc]

**Response Format    Syntax**

\$GPZDA,ZDA,m1,d2,d3,d4,d5,d6\*cc

## Parameters

Parameter	Description	Range
m1	UTC time (hhmmss.ss)	000000.00-235959.99
d2	Current day	01-31
d3	Current month	01-12
d4	Current year	0000-9999
d5	Local zone offset from UTC time (hour)	-13 to +13
d6	Local zone offset from UTC time (minutes)	00-59
*cc	Checksum	*00-*FF

## Example

```
$PASHQ,ZDA
```

```
$GPZDA,162256.27,25,02,2008,+00,00*43
```

## Related Commands

```
$PASHS,ZDA
```

```
$PASHS,NME
```

## Automatic Output of ZDA Messages

This is a reminder on how to output ZDA messages at regular intervals of time: Use the \$PASHS,NME command with the syntax below:

```
$PASHS,NME,ZDA,<port_ID>,ON,<Rate>
```

For more details on the \$PASHS,NME command, refer to the *Set Command Library* Chapter.

As an example, the command below will output ZDA messages on port A at a rate of 60 seconds:

```
$PASHS,NME,ZDA,A,ON,60
```



# Chapter 9. DCOL Commands & GSOF Messages

## Introduction

The MB-Two now supports the Trimble proprietary interface control protocol designed for all BD-9xx and BX-9xx products.

Refer to Chapters 1- to 4 of the Trimble OEM BD9xx GNSS Receiver Family ICD document for more information on the protocol. The same information can also be found in the on-line Web Help at [http://www.trimble.com/OEM\\_ReceiverHelp/v5.11/en/default.html](http://www.trimble.com/OEM_ReceiverHelp/v5.11/en/default.html).

Below is the list of commands supported by the MB-Two:

Command	Function
06h GETSERIAL	Requests Receiver and Antenna Information.
07h RSERIAL	Returns the receiver and antenna serial numbers, types, processor versions and number of channels.
4Ah GETOPT	Requests a list of receiver options that are installed on the receiver.
4Bh RETOPT	Returns the receiver options installed at the factory
54h GETSVDATA	Requests satellite data. Request may be for an array of flags indicating what satellite data is available, a particular satellite ephemeris or almanac. In addition, satellites may be enabled/disabled with this command.
55h RETSVDATA	Response to the GETSVDATA command. Returns the Ephemeris/Almanac for the SV PRN requested or an array of flags indicating which satellites are being tracked, have Ephemeris/Almanac available and Enable/Disable state.
64h APPFILE	Transfers an appfile between an application and a receiver.
65h REQAPPFILE	Requests an appfile (by system number).
6Eh BREAKRET	Returns receivers current serial port baud rate and protocols. This is the response to a BREAK SEQUENCE (as defined below), or a BREAKREQ.
6Fh BREAKREQ	Request for current receiver product, firmware and software versions, serial port baud rate and protocols.

Details of each of these commands are given below. Where differences exist between this description and the Trimble OEM BD9xx GNSS Receiver Family ICD document, those differences are mentioned in bold characters. When used, the term “receiver” always refers to the MB-Two board.

NOTE: Below are useful links to the Web Help about the following fields:

- Receiver Status Code
- Packet Structure

## 06h: GETSERIAL

---

Command Packet 06h requests board and antenna information. The board responds by sending the data in the Report Packet 07h.

All data in the packet flows from the data collector to the board.

Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start transmission
1	STATUS	1 (Char)	??h	Receiver status code
2	PACKET TYPE	1 (Char)	06h	Command packet 06h
3	LENGTH	1 (Char)	00h	Data byte count
4	CHECKSUM	1 (Char)	??h	Checksum value
5	ETX	1 (Char)	03h	End transmission

This command is usually used during the initial connection sequence when an app is learning about whatever instrument is attached.

## 07h: RETSERIAL

---

Report Packet 07h is sent in response to command packet 06h. The report returns the board and antenna serial number, antenna type, software processor versions, and the number of reception channels.

All data in the packet flows from the board to the data collector. (See table on next page.)



Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start transmission
1	STATUS	1 (Char)	??h	Receiver status code
2	PACKET TYPE	1 (Char)	07h	Command packet 07h
3	LENGTH	1 (Char)	??h	Bytes of data after this byte (excluding checksum and ETX)
4-11	RECEIVER SERIAL #	8 (Chars)	ASCII text	Receiver serial number. Note – On newer receivers such as the MB-Two, this field gives the lowest (least significant) 8 characters of the serial number and so the LONG SERIAL NUMBER field should be used instead.
12-19	RECEIVER TYPE	8 (Chars)	<b>“MB2”</b>	<b>Board model</b>
20-24	NAV PROCESS VERSION	5 (Chars)	ASCII text	<b>Reference version (e.g. v0.62 reported as “00062”)</b>
25-29	SIG PROCESS VERSION	5 (Chars)		<b>Reference version</b>
30-34	BOOT ROM VERSION	5 (Chars)		<b>Boot loader version, e.g. v1.32 reported as “00132”</b>
35-42	ANTENNA SERIAL #	8 (Chars)		
43-44	ANTENNA TYPE	2 (Chars)		
45-46	# CHANNELS	2 (Chars)	<b>“00”</b>	Not Applicable
47-48	# CHANNELS L1	2 (Chars)	<b>“00”</b>	Not Applicable
49-58	LONG SERIAL NUMBER	10 (Chars)		This is the serial number that should be used instead of RECEIVER SERIAL #.
58-89	LOCAL LONG ANT SERIAL	31 (Chars)		Not Applicable
90-120	BASE LONG ANT SERIAL	31 (Chars)		Not Applicable
121-151	BASE NGS ANT DESCRIPTION	31 (Chars)		<b>Antenna name</b>
152-153	USABLE CHANNELS	2 (Bytes)		Maximum number of usable channels with current configuration
154-155	# PHYSICAL CHANNELS	2 (Bytes)		Total number of hardware channels present.
156	# SIMULTANEOUS CHANNELS	1 (Byte)		Number of satellites that the receiver can track at one time.
157-161	Antenna INI version	5 (Chars)	ASCII text	<b>Antenna.ini file version (e.g. v7.13 reported as “00713”)</b>
162	CHECKSUM	1 (Char)	??h	Checksum value
163	ETX	1 (Char)	03h	End transmission

## 4Ah: GETOPT

This command requests a list of receiver options that are installed in the board. All data in DCOL packets should be stored in Big-Endian format. All data in the packet flows from the data collector to the board. The board responds by sending the data in response 4Bh, RETOPT.

Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start transmission
1	STATUS	1 (Char)	00h	Receiver status code
2	PACKET TYPE	1 (Char)	4Ah	Command packet 4Ah
3	LENGTH	1 (Char)	<b>00h or 01h</b>	Bytes of data after this byte (excluding CHECKSUM and ETX).
4	OPTIONS PAGE	1 (byte)	<b>00h, 01h or 02h</b>	Selects the required options information page: <ul style="list-style-type: none"> <li>• <b>0: (or not present) Send original (legacy) options packet</b></li> <li>• <b>1...n: Send subsequent options pages</b></li> </ul>
5	CHECKSUM	1 (Char)	??h	Checksum value
6	ETX	1 (Char)	03h	End transmission

## 4Bh: RETOPT

This packet response returns all the options installed in the board. All data in the packet flows from the board to the data collector. Report Packet 4Bh is sent in response to command 4Ah, GETOPT.

Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start transmission.
1	STATUS	1 (Char)	??h	Receiver status code.
2	PACKET TYPE	1 (Char)	4Bh	Command packet 4Bh
3	LENGTH	1 (Char)	<b>31h or ??h</b>	Bytes of data after this byte (excluding checksum and ETX).
4	<b>ELEVATION MASK / PAGE</b>	1 (byte)		<b>Elevation Mask, or 5Bh, or 5Ch</b>
<b>----- If ELEVATION MASK/PAGE=91 (90, 91 and 92 available)</b>				
5	Number of Pages Supported	1 (byte)	<b>02h</b>	
6-7	RESERVED	2 (short)		
8-9	RESERVED	2 (short)		
10-11	RESERVED	2 (short)		
12-15	OPTION BLOCK BITS 0-31 (LSB is bit 0)	4 (int)		OPTION BLOCK BITS 0-95 best processed as 3 unsigned long integers. For example, to determine if the GLONASS option is installed in receiver, check if bit #32 is enabled (1).
16-19	OPTION BLOCK BITS 32-63 (LSB is bit 32)	4 (int)		GLONASS enable bit (data byte 16 counting from 1 page number) and least significant bit (0) of that byte. The currently defined bits are given below this table.
20-23	OPTION BLOCK BITS 64-95 (LSB is bit 64)	4 (int)		
24-25	RESERVED	2 (short)		
26-28	RESERVED	3 (bytes)		
29-31	RESERVED	3 (bytes)		
32	RESERVED	1 (byte)		
33	RESERVED	1 (byte)		
34	RESERVED	1 (byte)		
35	RESERVED	1 (byte)		
36-37	RESERVED	2 (short)		
38	CHECKSUM	1 (Char)	??h	Checksum value. See Packet structure, page 10
39	ETX	1 (Char)	03h	End transmission.

Where:

Option block bits are defined as follows: bit = 1 (enabled), bit = 0 (disabled)

bit 0: CMR inputs

bit 1: CMR outputs **(if option [K] is active)**

bit 2: RTCM inputs

bit 3: RTCM outputs **(if option [K] is active)**

bit 4: N/A

bit 5: N/A

bit 6: N/A

bit 7: **N/A**

bit 8: Moving base

bit 9: 10Hz measurements **(if option [6] active)**

bit 10: 20Hz measurements **(if option [W] active)**

bit 11: **Data logging (internal Rx memory), if option [R] active**

bit 12: N/A

bit 13: N/A

bit 14: Event markers

bit 15: N/A

bit 16: **N/A**

bit 17: N/A

bit 18: N/A

bit 19: N/A

bit 20: N/A

bit 21: L2CS support **(if option [Y] active)**

bits 22–31: N/A

bit 32: GLONASS enabled **(if option [G] active)**

bit 33: Enable Web UI support

bit 34: Enable USB on the go and hard drive logging support

bit 35-43: N/A

bit 44: Disable SBAS **(if option [N] active)**

bits 45-46: N/A

bit 47: Disable CMRx output

bits 48-51: N/A

bit 52: **Enable compass (if option [B] active)**

bits 53-54: N/A

bit 55: Galileo enabled (if option [O] active)

bits 56-63: N/A

bit 64: N/A

bit 65: N/A  
bit 66: QZSS enabled (if option [N] active)  
bit 67: N/A  
bit 68: L1 RTK support enabled  
bit 69: N/A  
bit 70: **N/A**  
bits 71-76: N/A  
bit 77: **N/A**  
bits 78-80: N/A  
bit 81: Disable GPS (if option [N] active)  
bits 82-95: N/A

## 54h: GETSVDATA

Command 54h requests satellite information. The request may be for an array of flags showing the availability of satellite information such as an ephemeris or almanac. In addition, satellites may be enabled or disabled with this command.

All data in the packet flows from the data collector to the board.

Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start transmission
1	STATUS	1 (Char)	00h	Receiver status code
2	PACKET TYPE	1 (Char)	54h	Command type
3	LENGTH	1 (Char)	??h	Data byte count
4	SUBTYPE	1 (Byte)		<b>Supported subtypes: 20 (SV Enable/Disable)</b>
SUBTYPE= 20				
5	SV PRN NUMBER	1 (Byte)		
6	SAT TYPE	1 (Byte)		<b>0: GPS</b> <b>1: SBAS (WAAS, EGNOS, MSAS etc)</b> <b>2: GLONASS</b> <b>3: Galileo</b> <b>4: QZSS</b> <b>5: Beidou (pre-ICD SV numbering)</b> <b>6: OmniStar</b> <b>7: Beidou (ICD SV numbering)</b> <b>9: IRNSS</b> <b>10: Beidou (with 1/2 phase correction on B1 signal from Geo SVs)</b>
7	MODE	1 (Byte)		<b>0=Return SV flags for SAT TYPE;</b> <b>1=Disable SV</b> <b>2=Enable SV</b>
	CHECKSUM	1 (Char)	??h	Checksum value
	ETX	1 (Char)	03h	End transmission

Where:

**SUBTYPE:** indicates what is requested:

- 20: SV Enable/Disable/Ignore Health Controls.

**SV PRN Number:** Satellite number for which ephemeris/almanac is required or to be enabled/ disabled; ignored if SV flags or ION / UTC data is requested:

- 1–32: GPS satellites.
- 1-24: (GLONASS SV 1–24).
- Galileo SV Range: 1–36.
- QZSS SV Range: 193 - 198.
- BeiDou SV Range: 1–30.

**FLAGS:** Bitmapped field having the following values:

- Bit 0 set: Return GLONASS Flags appended to the GPS FLAGS replies (subtypes 0, 4, 5).
- Bit 1 set: Return Galileo and GLONASS Flags appended to the GPS FLAGS replies (subtypes 0, 4, 5). SV being controlled is Galileo, not GPS or GLONASS (subtypes 4, 5).
- **Bits 2 and 3 are ignored**

**SAT TYPE:** Subtype 20 only. Specifies the Satellite System for the PRN which is being configured.

- 0: GPS: 1–32
- 1: SBAS (WAAS, EGNOS, MSAS, etc.): 1–39 (PRN 120–158)
- 2: GLONASS: 1–24
- 3: Galileo: 1–36
- 4: QZSS: 1–5 (PRN 193-197)
- 7: BeiDou: 1–30

**MODE:** Subtype 20 only.

- 0: Return SV flags for SAT TYPE
- 1: Disable SV
- 2: Enable SV
- 3: **Not supported**
- Only Mode = 0 is valid for SBAS.

The reply for this command will be a RETSVDATA report packet, or a NAK if the request failed.

Enable/Disable satellite (subtype 20) always returns RETSVDATA (subtype 20) as if SV Flags were requested.

## 55h: RETSVDATA

Report Packet 55h is sent in response to Command Packet 54h. The report includes either the ephemeris or almanac information for a specific satellite, or ION/UTC data, the Enabled/Disabled state and Heed/Ignore Health state of all satellites, or the condition of satellite status flags for one satellite or all satellites.

All data in the packet flows from the board to the data collector.

Byte	Item	Type	Value	Meaning
0	STX	1 (Char)	02h	Start Transmission
1	STATUS	1 (Char)	??h	Receiver Status Code
2	PACKET TYPE	1 (Char)	55h	Response Packet Type
3	LENGTH	1 (Char)	??h	Bytes of data after this byte (excluding checksum and ETX)
4	SUBTYPE	1 byte	Refer subtype values below	The supported subtypes are: <ul style="list-style-type: none"> <li>• 0: SV Flags</li> <li>• 20: Generic SV Flags</li> </ul>

----- DATA RECORDS 0—22 -----

Data from one of the records that are encoded in type 55h records, as indicated by byte #4 SUBTYPE.

**The supported subtypes are:**

**20: Generic SV Flags**

Last subtype byte + 1	CHECKSUM	Char	??h	Checksum Value
-----------------------	----------	------	-----	----------------

Last subtype byte + 2	ETX	Char	03h	End Transmission
-----------------------	-----	------	-----	------------------

Only the satellite information, requested by Command Packet 54h, is sent in the report packet. As a result, several forms of the Report Packet 55h can be requested.

Returns a NAK if the GETSVDATA request meets one of the following criteria:

- SV PRN is out of range (except for SV flags)
- Data Switch is out of range
- Data is not available for the requested SV.

### SV Flags report

The SV FLAGS report is sent when Command Packet 54h is used to request the status of the SV Flags for one satellite or all satellites. The Command Packet 54h SUBTYPE byte (byte 4) is set to twenty (20) when requesting the report.



The following table shows the additional records provided in Report Packet 55h when SV flags data is included:

Subtype 20: SV Flags Report				
Byte	Item	Type	Value	Meaning
6	SAT TYPE	1 byte		<b>0: GPS</b> <b>1: SBAS (WAAS, EGNOS, MSAS, etc.)</b> <b>2: GLONASS</b> <b>3: Galileo</b> <b>4: QZSS</b> <b>5: BeiDou (pre-ICS SV numbering)</b> <b>6: Omnistar</b> <b>7: BeiDou (ICD SV numbering)</b> <b>9: IRNSS</b> <b>10: BeiDou (with 1/2 phase correction on B1 signal from Geo SVs)</b>
7	MODE	1 byte	See below	<b>0= Return SV flags for SAT TYPE</b> <b>1= Reply to disable SV</b> <b>2= Reply to enable SV</b> <b>(3: Not supported)</b>
8	Number of bytes (n) for each of the following fields.	1 byte	1-?	Number of bytes in each of the following flags.
	EPHEMERIS FLAGS	n bytes	? flag bits	Flags show availability of ephemeris, LSB = first PRN. Set to 1 when ephemeris is available.
	ALMANAC FLAGS	n bytes	? flag bits	Flags show availability of almanac, LSB = first PRN. Set to 1 when almanac is available.
	SV DISABLED FLAGS	n bytes	? flag bits	Flags show disabled SVs, LSB = first PRN. Set to 1 when satellite is disabled.
	SV UNHEALTHY FLAGS	n bytes	? flag bits	Flags show the health of satellites, LSB = first PRN. Set to one when satellite is unhealthy.
	SV "IGNORE HEALTH" FLAGS	n bytes	? flag bits	<b>Not supported.</b>
----- End of Subtype 20 -----				

## 64h: APPFILE

The following table shows the structure of the report packet containing the application file. All data in the packet flows from the data collector to the board.

Byte	Item	Type	Value	Notes
0	STX	1 (Char)	02h	Start transmission.
1	STATUS	1 (Char)	00h	Receiver status code.
2	PACKET TYPE	1 (Char)	64h	Command Packet 64h.
3	LENGTH	1 (Char)	??h	Data byte count.
4	TRANSMISSION NUMBER	<b>1 (Byte)</b>		A Transmission Number is a unique transmission serial number ranging from 0-255, typically incrementing by 1 each time an application file is sent.
5	PAGE INDEX	<b>1 (Byte)</b>		Index number (0-255) assigned to the current page.
6	MAXIMUM PAGE INDEX	<b>1 (Byte)</b>		Index number (0-255) assigned to the last page of the packet.
<b>File Control Information Block:</b>				
7	APPLICATION FILE SPECIFICATION VERSION	1 (Char)	03h	<b>Ignored</b>
8	DEVICE TYPE	1 (Char)	00h	<b>Ignored</b>
9	START APPLICATION FILE FLAG	1 (Char)	01h	<b>Ignored</b>
10	FACTORY SETTINGS FLAG	1 (Char)	00h	<b>Ignored</b>
<b>Application File Records:</b>				
				<b>Supported records:</b> <ul style="list-style-type: none"> <li>• General Controls (01h)</li> <li>• Output message (07h)</li> <li>• Antenna (08h)</li> <li>• Static kinematic (0Ah)</li> </ul>
Length+4	CHECKSUM	CHAR	??h	Checksum value
Length+5		CHAR	03h	End transmission

## Output Messages Record

The output message record selects the outputs for a specified serial port, the frequency of message transmissions, the integer second offset from the scheduled output rate, and output-specific flags.

The following table shows the additional records provided when the output message record is included with Command Packet 64h.

Byte	Item	Type	Value	Notes
0	RECORD TYPE	1 (Char)	07h	Output Message Record.
1	RECORD LENGTH	1 (Char)	04h, 05h or 06h	Number of bytes in the record, excluding bytes 0 and 1. The number of bytes is dependent on the number of output specific flags.
2	OUTPUT MESSAGE TYPE	1 (Char)	See Output message type byte values	<b>Type of message or packet:</b> <ul style="list-style-type: none"> <li>• 00h: Turns off all outputs on all ports</li> <li>• FFh: Turns off all outputs on a given port</li> <li>• 0Ah: GSOF output</li> <li>• NMEA: Supported messages are: GGA (byte value: 6), GGK (7), ZDA (8), GSOF (10), VTG (12), GST (13), GSV (18), HDT (31), ROT (32), VRSGGA (37), GSA (38), RMC (40), GLL (44), GRS (45), GBS (49), GNS (50), DTM (51), MSS (53).</li> </ul>
3	PORT INDEX	1 (Char)	See port number values	<b>Port number values:</b> <ul style="list-style-type: none"> <li>• Port 1: External serial port A</li> <li>• Port 2: Serial port B</li> <li>• Port 3: Serial port D</li> <li>• Port 16: USB port C</li> <li>• Port 21: TCP/IP port P (client)</li> <li>• Port 22: TCP/IP port Q (client)</li> <li>• Port 23: TCP/IP port I (server)</li> <li>• Port 24: TCP/IP port F (server)</li> </ul>
4	FREQUENCY	1 (Char)	See Frequency byte values	Frequency byte values (see table below).
5	OFFSET	1 (Char)	00h–FFh	N/A
6-7	OUTPUT SPECIFIC FLAGS			These flags are dependent upon the output message type, byte 2. One or two flag bytes can be sent, but 2 are always stored in the board. For a GSOF output, the first flag is the GSOF subtype
8				N/A
10				N/A

## Frequency byte values:

Byte Value	Meaning
0	Off
1	10 Hz= 100 milliseconds
2	5 Hz= 200 milliseconds
3	1 Hz= 1 second
4	2 seconds
5	5 seconds
6	10 seconds
7	30 seconds
8	60 seconds
9	5 minutes
10	10 minutes
11	2 Hz= 500 milliseconds
12	15 seconds
13	20 Hz= 50 milliseconds
14	Triggered
15	50 Hz= 20 milliseconds
16	100 Hz= 10 milliseconds
255= 0xff	Once

## Antenna Record

The antenna record identifies the antenna type and the true vertical height of the antenna above the ground mark.

The following table shows the additional records provided when the antenna record is included with Command Packet 64h.

Byte	Item	Type	Value	Notes
0	RECORD TYPE	1 (Char)	08h	Reference Node record.
1	RECORD LENGTH	1 (Char)	2Ch	Number of bytes in record, excluding bytes 0 and 1.
2-9	ANTENNA HEIGHT	8 (Double)	Meters	Vertical height of antenna, in meters.
10-11	ANTENNA TYPE	2 (Short)	See Antenna type byte values below.	Defines the type of antenna connected to the receiver.
12	RESERVED	1 (Char)	00h	Reserved (set to zero).
13	MEASUREMENT METHOD	1 (Char)		00h = Bottom of antenna mount FFh = Antenna phase center
14-45	ANTENNA SERIAL NUMBER	32 (Char)		Space for 31 characters + null terminator
<b>46-49</b>	<b>RESERVED</b>	<b>4 x 1 (byte)</b>		

The following table provides byte values for the information in the antenna record.

### Antenna type byte values:

Byte value		Notes
Dec	Hex	
85	55h	Zephyr (KZ)
86	56h	Zephyr Geodetic™ (GZ)
184	B8h	Zephyr – Model 2 (KS)
185	B9h	Zephyr Geodetic 2 (GS)
266	01h 0Ah	AG25 (KT)
309	01h 35h	AV59 (FA)
349	01h 5Dh	AV33 (M0)
403	01h 93h	AV34 (M1)
404	01h 94h	AV37 (M2)
406	01h 96h	LV59 (M3)

For a complete and up-to-date list of antennas, go to [www.trimble.com/trimbleconfiguration\\_ts.asp](http://www.trimble.com/trimbleconfiguration_ts.asp) and install the Trimble Configuration Utility. Then open the Antenna.INI file (the default location is C:\Program Files\Common Files\Trimble\Config\antenna.ini) with an ASCII viewer to find antennas by name and their antenna type value.

### Static/Kinematic Record

The static/kinematic record identifies the antenna type and the true vertical height of the antenna above the ground mark.

The following table shows the additional records provided when the static/kinematic record is included with Command Packet 64h.

Byte	Item	Type	Value	Notes
0	RECORD TYPE	1 (Char)	0Ah	Static/Kinematic record.
1	RECORD LENGTH	1 (Char)	01h	Number of bytes in record, excluding bytes 0 and 1.
2	STATIC/KINEMATIC MODE	1 (Char)	See Static/kinematic byte values below.	Configures receiver for static or kinematic operation.

Static/kinematic byte values:

Byte value		Notes
Dec	Hex	
0	00h	Kinematic
1	01h	Static
2–255	02h–FFh	Reserved

## 65h: GETAPPPFILE

A specific application file can be downloaded from the board by sending the Command Packet 65h. If the request is valid, a copy of the application file is downloaded to the remote device in Report Packet 64h.

The board can store multiple application files (including a default application file, containing the factory default parameter settings) in the application file directory. Each application file is assigned a number to give the file a unique identity within the directory. The application file containing the factory default values is assigned a system file index code of zero (0).

The following table shows the packet structure. All data in the packet flows from the data collector to the board. For more information, see Command 64h.

Byte	Item	Type	Value	Notes
0	STX	1 (Char)	02h	Start transmission
1	STATUS	1 (Char)	Receiver status byte	Receiver status indicator.
2	PACKET TYPE	1 (Char)	65h	Command Packet Type
3	LENGTH	1 (Char)	Packet structure	Data byte count.
4-5	SYSTEM FILE INDEX	2 (Short)	0-n	Unique number (ID code) assigned to each of the application files stored in the application file directory. <b>Only 1 is accepted.</b>
6	CHECKSUM	1 (Char)	See Packet structure	Checksum value.
7	ETX	1 (Char)	03h	End transmission

## 6Eh: BREAKRET

---

Response Packet 6Eh is sent in response to the Command Packet 6Fh. Response Packet 6Eh returns the board's current serial port communication parameters, board version numbers and dates, and communication protocol settings when the remote device sends a 250 millisecond (minimum duration) break sequence.

### **Sending a Break Sequence:**

To initiate a break sequence return, the following events need to occur:

1. The remote device sends a break sequence with a minimum duration of 250 milliseconds to the board. For example, pressing [Ctrl] + [Break] from an office computer is equivalent to sending a break sequence.
2. The board detects the break signal and responds by setting the communication parameters for the serial port to 9600 baud, 8 data bits, no parity, and 1 stop bit.
3. The board outputs an Identity Message through the serial port to the remote device.



The following table describes the structure of Report Packet 6EH

Byte	Item	Type	Value	Notes
0	STX	1 (Char)		Start transmission.
1	STATUS	1 (Char)	??h	<b>Board status indicator</b>
2	PACKET TYPE	1 (Char)	6Eh	Report Packet Type.
3	LENGTH	1 (Char)	??h	Data byte count.
	PRODUCT	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the receiver product family name.
	PORT SETTING	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the serial port settings and the break sequence acknowledgment code.
	PORT STATUS	(Chars)	FIX" / "ADJ"	FIX: Port settings cannot be changed. ADJ: Port settings can be changed.
	VERSION	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the software version number and version release date.
	COMM PROTOCOL	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the communication protocols supported on serial port.
	SERIAL NUMBER	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the board serial number.
	NAME (optional)	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the board name and serial number.
	IP ADDRESS	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the IP address of the board. If no IP address is assigned or set, 0.0.0.0 is returned.
	WLANIP (optional)	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the receiver Wireless LAN IP address. If no IP address is assigned or set, 0.0.0.0 is returned.
	CORE VER	(Chars)	Comma- delimited ASCII string	Comma-delimited ASCII string indicating the board firmware version number.
	CHECKSUM	1 (Char)	See Packet structure	Checksum value.
	ETX	1(Char)	03h	End transmission.

## 6Fh: BREAKREQ

---

Command Packet 6Fh requests the board's current serial port communication parameters, receiver version numbers and dates, and communication protocol settings. The board responds by sending the data in the Report Packet 6Eh.

All data in the packet flows from the data collector to the board.

Byte	Item	Type	Value	Notes
0	STX	1 (Char)	02h	Start transmission.
1	STATUS	1 (Char)	See Receiver status byte	Receiver status indicator.
2	PACKET TYPE	1 (Char)	6Fh	Command packet 6Fh.
3	LENGTH	1 (Char)	00h	Data byte count.
4	CHECKSUM	1 (Char)	See Packet structure	Checksum value.
5	ETX	1 (Char)	03h	End transmission.

## GSOF Messages

---

The description of the GSOF messages supported by the MB-Two is available on line by clicking on the links below:

- [GSOF messages overview](#)
- [1 Position Time](#)
- [2 Lat Lon Height](#)
- [3 ECEF Position](#)
- [9 PDOP Info](#)
- [11 Position VCV Info](#)
- [12 Position Sigma Info](#)
- [16 Current UTC Time](#)
- [38 Position Type Information](#)
- [33 All SV Brief Info](#)

NOTE: GSOF is the acronym for “General Serial Output Format”.



# Chapter 10. Appendices

## MB Loader Software Utility for Upgrades

---

*MB Loader* is a software tool provided in the evaluation kit. Use *MB Loader* software to:

1. Upgrade the board firmware
2. Install new firmware options
3. Validate RTX subscription.
4. Read the warranty expiration date of the board.

### Installing MB Loader

*MB Loader* can also be downloaded from:

[ftp://ftp.ashtech.com/OEM\\_Sensor\\_ADU/MBTwo/Firmware%20Serial%20Loader](ftp://ftp.ashtech.com/OEM_Sensor_ADU/MBTwo/Firmware%20Serial%20Loader)

The install file is an exe file. Simply double-click on this file to start installation. Follow the instructions on the screen to complete the installation.

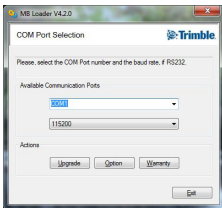
### Getting Started With MB Loader

MB Loader will use either a serial (RS232), Bluetooth or USB connection to communicate with the board. USB is recommended.

1. Connect your computer to the board using a USB connection.
2. Run MB Loader on your computer.
3. Select the computer's port ID used to communicate with the board. This port ID should correspond to the computer's USB port.

NOTE: An easy way to identify which port ID on your computer is the USB port is to run MB Loader first without the USB connection and read the list of available ports in MB Loader. After restoring the USB connection with the board, check that list again. An extra port ID will then be listed, being the one assigned to the USB port. Select that port. (You don't need to define a baud rate for a USB port.)

4. To upgrade board firmware, install a new firmware option or validate a Trimble RTX subscription, see sub-sections below.



## Upgrading Board Firmware

*You are not allowed to upgrade a board if the startup protection is active or if the board is operated with an in-progress or expired validity period.*

Firmware upgrades will be downloaded from:

[ftp://ftp.ashtech.com/OEM\\_Sensor\\_ADU/MBTwo/Firmware](ftp://ftp.ashtech.com/OEM_Sensor_ADU/MBTwo/Firmware)

Upgrades are delivered in the form of compressed “.tar” files. The name of the “.tar” file, as well as the step-by step upgrade procedure will be given in the accompanying *Release Note*.

Unless otherwise specified in the *Release Note* attached to the upgrade package, follow the instructions below to complete the upgrade of your board:

1. Follow the first three steps described in *Getting Started With MB Loader on page 507*.
2. Click **Upgrade**. Wait until MB Loader has detected the board.
3. Browse your computer in search of the upgrade file.
4. Select the file and click **Open**. MB Loader then provides information on the currently installed firmware.



5. When you are ready, click on the **Update** button.
6. Let the board proceed with the upgrade (a status window is displayed showing a progress bar). **Take care not to turn off the board while the upgrade is in progress.**
7. After successful completion of the upgrade, click **Close** to close the status window. Check that the new firmware is now installed (version and date displayed in the MB Loader main window).
8. Click **Close** again, then **Exit** to quit MB Loader.

### Upgrading Board Firmware Using a USB Memory Stick

This is another nice way to upgrade the firmware. The board should be installed on the evaluation kit.

You need to use a computer running AshCom, a USB mass storage key (memory stick), a USB OTB cable and the

\$PASHS,UPL,UPG proprietary command (see also *UPL,UPG: Upgrading the Receiver Firmware from FTP on page 314*).

The upgrade procedure is as follows:

- Copy the upgrade file (*MB2\_upgrade\_vx.x.tar*) to the memory stick.
- Connect the memory stick to the USB OTG cable.
- Connect the other end of the USB OTG cable to the evaluation kit.
- On your computer, from AshCom's command box, send this command to the board:

**\$PASHS,UPL,UPG**

The board will first return an acknowledge message, then a reboot message:

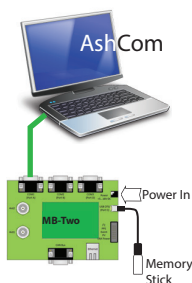
\$PASHR,ACK\*3D

\$PASHR,UPL,REBOOT,MB2\_upgrade\_vx.X.tar\*<checksum>

This starts the upgrade. Note that the second message may appear more than once during the procedure.

When the upgrade is complete and successful, the board returns the following line:

\$PASHR,UPL,OK,MB2\_upgrade\_vx.X.tar\*<checksum>



## Reading Board Warranty Expiration Date

*MB Loader* can be used to query the Trimble database for the warranty expiration date of your GNSS receiver. (After a receiver warranty has expired, remember receiver firmware upgrades are no longer free of charge.)

You don't need to have your receiver connected to *MB Loader* to read its warranty expiration date. Just enter its type and serial number and *MB Loader* will return this information to you, provided there is an active Internet connection on your computer, and your receiver is known to the database.

- Run *MB Loader* on your computer.
- Click on **Warranty**
- Select the type of your receiver and enter its serial number
- Click on **Compute**. *MB Loader* returns the warranty expiration date in a field underneath the **Compute** button. Additionally, *MB Loader* generates a proprietary command that you can run in your receiver if you want to be sure your receiver has the correct warranty expiration date in memory. Carefully write down this command  
Use AshCom to apply this command to the board.

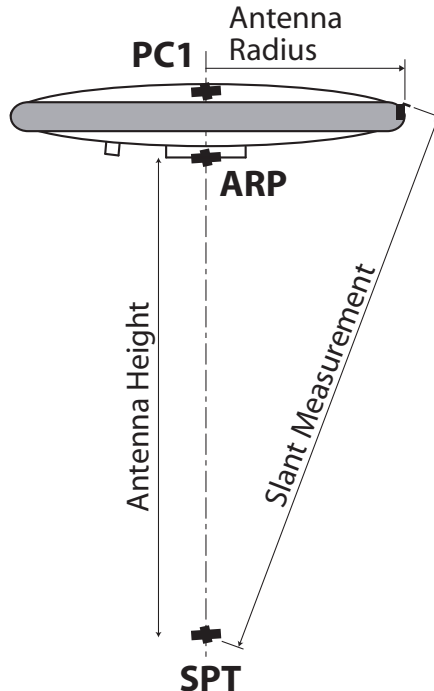
NOTE: When upgrading the board firmware using a computer with an Internet connection, be aware MB Loader will at the same time automatically check the warranty expiration date of your receiver. MB Loader will ask you if it can update this date if it is found wrong.

## Base Antenna Issues

The firmware has to deal internally with positions tagged to different points on the antenna:

- The L1 Phase Center (PC1)
- The Antenna Reference Point (ARP)
- The ground mark, or Survey Point (SPT)

By default, the software assumes that PC1, ARP and SPT are the same point.



Through the \$PASHS,POS command, you can enter the exact position of the base and tell the firmware if it is tagged to PC1, ARP or SPT. The position you enter through this



command is the one that will be transmitted to rovers, along with RTK differential messages.

To allow the firmware to determine the position of the other two points, starting from the one you enter, antenna phase offset parameters are required. These parameters include antenna height, slant measurement and radius for SPT.

ARP and SPT are related to each other through commands \$PASHS,ANH and \$PASHS,ANP.

PC1 and ARP are related to each other via the Phase Center Offset table (PCO table), referring to the specified antenna name.

Whenever a position is entered using \$PASHS,POS, the firmware re-calculates the positions of the other two points, using the current antenna name, as entered using the \$PASHS,ANP,OWN command. If the antenna name is unknown (the antenna name field has been left blank), then the firmware assumes that the three points (PC1, ARP, SPT) are physically the same point in space.

Whenever you change the antenna name, using \$PASHS,ANP,OWN, or antenna parameters, using \$PASHS,ANP,<name>,<parameters>, the firmware re-calculates the other two points, keeping the user-entered position unchanged. For example, if the position entered through \$PASHS,POS is that of PC1, then ARP will be re-calculated. Conversely, PC1 will be re-calculated if the entered position is that of the ARP.

Depending on the protocol and message set used, the transmitted reference position will be tagged to a specific point. See table below.

Message Type	PC1	ARP	SPT
RTCM2.3 messages 3+22	•		
RTCM2.3 message 24		•	
RTCM3.1 messages 1005 or 1006		•	
CMR	•		
ATOM,RNX		•	

All the raw and differential data the firmware generates are always fully consistent with the reference position and the antenna name.

## Rover Antenna Issues

---

Antenna information is vital for RTK operation, because not only does it contain the information for PC1-ARP transformation, but also the PC2 offset (PC2= L2 Phase Center) and the PCV (Elevation Dependent Phase Center Variations). Not making this information available to the rover may lead to a noticeable degradation of the position determination or the inability to reliably fix the ambiguities.

Since both rover and base data are involved in RTK operation, in which single-differencing is performed (i.e. subtracting base correction from rover data), it is essential that the parameters of both the reference antenna (REF) used at the base and the rover antenna (OWN) be known to the rover.

As the internal RTK engine always works on the PC1 position, any position received by the differential processor will be transformed to PC1, using base antenna parameters. Base antenna parameters are retrieved from hard-coded or user-defined antenna lists, using the antenna name decoded from the differential stream.

Unfortunately, only a few protocols provide the antenna name. These are listed below:

- RTCM2.3 message type 23
- RTCM3.1 message types 1007, 1008 and 1033
- Set of ATOM ATR messages

If the name of the antenna used at the base (reference antenna) is not available in the differential stream, you should use the \$PASHS,ANP,REF command in the rover to specify that name. Since this command does not carry a default reference antenna name (by default, the corresponding field is empty), the rover will assume that PC1, PC2 and ARP at the base are all the same point in space, and the elevation-dependent biases are all zero. Besides, the rover will use either the entered or received reference antenna name, whichever is available last.

Even if the differential stream conveys a PC1-tagged position, which is needed for RTK, the antenna name remains a fundamental parameter, especially if the reference antenna has a noticeable L1-L2 phase center offset and/or a noticeable PCV.

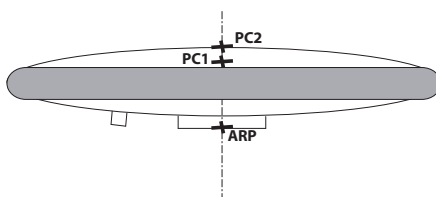
## “Virtual Antenna” Concept

Experience has shown that a lot of rovers from other manufacturers do not know the PCO parameters of the antennas used by our base receivers (PCO= Phase Center Offset). As a result, these rovers cannot fully benefit from the streams of reference data delivered by our bases.

However, as there are a few antennas the names of which are well known to most vendors (e.g. ADVNULLANTENNA), we implemented a special firmware feature allowing the owner of a base to modify the complete flow of raw and differential data, as if they had been collected by a well known antenna, rather than the one actually used. This summarizes the concept of “virtual antenna”.

To enable this feature in a base, you should specify an antenna name for the truly used reference antenna (OWN), as well as a virtual antenna name (OUT) using the \$PASHS, ANP,OUT command. Looking up the antenna name in the hard-coded or user-defined list of antennas, the base will then be able to make the following corrections:

- Virtual ARP coordinates= True ARP coordinates.
- Virtual PC1 coordinates: Deduced from the true ARP coordinates and the antenna parameters retrieved from the virtual (OUT) antenna name.
- Raw data: Adjusted to match the virtual antenna, i.e. L1 data will be centered on virtual PC1 and L2 data on virtual PC2. The adjusted raw data include both code and carrier L1 and L2, GPS and GLONASS.



Every time you modify any vital parameter through \$PASHS,POS or \$PASHS,ANP, the firmware will do the following automatically:

- Re-calculating the true ARP and/or the true PC1, using the data from the POS and OWN antenna names

- Calculating virtual PC1, using the true ARP and the data from the antenna name
- Adjusting raw data using data from the OWN and OUT antenna names

This mechanism guarantees the consistency of the position, observables/corrections and antenna name transmitted, regardless of the differential protocol and message set used. This leads to a rover being able to calculate a correct RTK position, provided it has been able to decode the antenna name provided in the differential stream (or use the antenna name entered through the \$PASHS,ANP,REF command) and retrieve the parameters of this antenna from its list of antenna names.

## Antenna Height Transformations

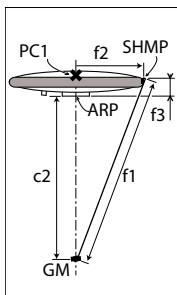
---

The entered reference position may be defined as:

- The position of the surveyed point (SP),
- The position of the Antenna Reference Point (ARP).
- The position of the antenna Phase Center.

The ground mark is defined as the vertical projection of the ARP to the ground.

The board firmware must be able to convert the originally entered position, first into ARP position, and then into L1 phase center position. To determine the ARP position, the firmware uses the data entered through either the ANT or ANH command, whichever was run last, to perform the required conversions.



- If the ANH command was the last run, the firmware will use the entered antenna height (**c2**) to determine the ARP position, expressed in local ENU coordinates:

$$ARP(North) = GM(North)$$

$$ARP(East) = GM(East)$$

$$ARP(Up) = (GM(Up) + c2)$$

- If the ANT command was the last run, the firmware will use the entered parameters (**f1**, **f2** and **f3**) to determine the ARP position, expressed in local ENU coordinates:

$$ARP(North) = GM(North)$$

$$ARP(East) = GM(East)$$

$$ARP(Up) = GM(Up) + [\sqrt{f1^2 - f2^2}] + f3$$

Remember the vertical offset (**f3**) is entered as a negative value if the SHMP is above the ARP. It is otherwise positive.

**Warning!** The value of **f1** is physically limited to 99.9999 meters but may be transmitted as a lesser value (for example, in RTCM3 format, the maximum allowed value is 6.5535 meters).

With a known antenna name, the APC (Antenna Phase Center) positions for both L1 and L2 may be deduced from the ARP, using L1 and L2 3D offsets available from the hard-coded IGS antenna table:

$$APC(North)_{L1} = ARP(North) + L1Offset(North)$$

$$APC(East)_{L1} = ARP(East) + L1Offset(East)$$

$$APC(Up)_{L1} = ARP(Up) + L1Offset(Up)$$

$$APC(North)_{L2} = ARP(North) + L2Offset(North)$$

$$APC(East)_{L2} = ARP(East) + L2Offset(East)$$

$$APC(Up)_{L2} = ARP(Up) + L2Offset(Up)$$

## How a Rover Reacts Dynamically to a Change of Base ID or Antenna Name

- Whenever the rover decodes a new base ID in the differential stream, all base-dependent parameters are reset to their default state in the rover. Through this reset,

the default antenna name is made unknown (the corresponding field is made empty). This way, no previously received antenna name or reference position can be used.

- While the base ID is kept unchanged, a change of protocol will not cause any reset. The new position received through the new protocol will be processed normally, just as a new message including the reference position.
- Whenever a rover decodes a new antenna name, or a new antenna name is entered through the appropriate \$PASH command, PC1 is re-calculated (if position received is tagged to ARP). The antenna parameters are also retrieved from the list of antennas for use in the processing run by the RTK engine.
- With a base declared as “static”, any change in the coordinates of the reference position provided by the differential stream will be interpreted by the rover as a true change in the base installation, or a change in the base setup.

If however the difference between the new and old coordinates are greater than 10 km, the RTK engine is reset (similar to a base ID change). If the difference is between 1 km and 10 km, only the baseline estimate is reset. The baseline is otherwise simply adjusted, without any RTK reset.

- With the RTCM2.3 protocol, the reference position can be provided either in message type 3+22 or message type 24. This means the reference position received is either tagged to PC1 or ARP. Usually the positions provided by message types 3+22 and 24 refer to the same antenna and their relationship is as described in the antenna PCO table.

If both messages are received, message type 3+22 (PC1-tagged position) has the priority. If message type 24 is received after message type 3+22, it will therefore be rejected, unless the difference between the two positions is greater than 25 meters.

## Useful Definitions

---

### Satellite Status

A GNSS satellite is considered to be **visible** if the corresponding healthy almanac is available, a receiver position is available, topo data for the satellite can be

computed (at least from the almanac) and the satellite is above the horizon.

In some situations, the receiver can track a satellite below local horizon. These satellites are also considered as visible. All visible satellites are reported in NMEA GSV messages and ATM,PVT,SVS blocks. Each visible satellite must report valid elevation and azimuth.

The receiver saves almanac data for all GNSS's. All saved almanac data may be deleted using the appropriate INI command. However there also exists default almanac data for geostationary satellites (SBAS, GEO BDS) that cannot be cleaned up. These are seen as data staying valid over time, which cannot be changed unless you perform a firmware update.

So if GLONASS is disabled for tracking (e.g. by setting \$PASHS,GLO,OFF), this does not mean that GLONASS satellites are no longer visible. On the contrary, these satellites may stay visible (although not tracked) because command \$PASHS,GLO,OFF does not clean up the GLONASS almanac.

A visible GNSS satellite is reported to be **tracked** if the corresponding DLL (Delay Locked Loop) is locked for at least one satellite signal. Generally, the fact that a satellite is tracked does not necessarily mean that it can provide the corresponding raw data (pseudo-range, Doppler and carrier phase observables).

A tracked satellite is always associated with a corresponding SNR reported in NMEA GSV messages and ATM,PVT,SVS blocks.

A visible satellite that is not tracked has a "zero" SNR value. NMEA messages always report an SNR value, usually for the highest-frequency signal of a given satellite (L1 for GPS, SBAS and GLONASS, E1 for Galileo, B1 for BeiDou). The ATM,PVT,SVS block reports SNR values for all the tracked signals from a given satellite.

The raw/differential data from a tracked satellite can be available internally but not output from the receiver if these data do not meet some quality requirements.

Raw data can be available via different groups of messages, e.g. standardized RTCM-3 data and proprietary ATM,RNX data.

A tracked GNSS satellite is reported to be **used** in the internal PVT process if at least one observable (pseudo-range, Doppler or Carrier phase) from at least one signal from a given satellite took part in the position epoch update. A satellite may be

tracked and its raw data output while not being used in the internal PVT process. Conversely, the raw data from a satellite can be disabled for output, while being used in the internal PVT process. The information about satellite usage status is available via the ATM,PVT message as well as its six ASCII counterparts (i.e. SGP, SGL, SGA, SBD, SSB, SQZ).

If not said explicitly, **used** refers to the position reported for the primary antenna. It must be noted that with platform positioning (M-sensor, P-cube concept), the PVT process includes computing different positions, baselines, vectors and angles. Generally, the same satellite can be used in one type of P-cube solution, and NOT in another type of P-cube solution. When and where required, the concerned PVT estimate is clearly specified when reporting a usage status.

About SBAS ranging data (code and carrier) used in position computation, you should know these mostly appear to be controlled by quite an instable clock. For this reason, they are not used in position computation, except in 1-Hz time-tagged RTK where instable SBAS clock is compensated for by using Single Difference (SD). On the contrary, SBAS code can be used in position computation, but only at integer seconds. As a result, SBAS data are used at integer seconds only, not at fractional seconds. That's why the SBAS usage flag changes regularly.

When used in differential or RTK mode, the receiver can mix corrected and uncorrected data in the internal PVT process. For example a satellite may be marked as used while there are no corrections received for this satellite. Conversely, a satellite for which corrections are available may not be used in the internal PVT process. In addition, and in RTK only, a satellite will not be reported as used if no corrections for this satellite are received from the base. And because of some specificity in RTK, a satellite may temporarily be set as used while being reported as being not tracked.

The table below shows what the RF bands actually are when the reported names are L1, L2, L3 and L5 (QZSS and SBAS are assumed to be part of GPS). Note that generally, L5 covers both L5 and L3 sub-bands.

#	RF ID	Conditional Name
1	GPS L1/GAL E1	L1
2	GPS L2	L2
3	GPS L5/GAL E5a	L5
4	GLO G1	L1
5	GLO G2	L2



#	RF ID	Conditional Name
6	GLO G3/GAL E5b/BDS B2	L3 (L5/L2)
7	GAL E6/BDS B3	L6
8	BDS B1 (phase 2)	L1

Of the different signals from different constellations pertaining to a specific sub-band, the board selects the one with the best SNR and reports it in the following NMEA messages as the sole SNR for the sub-band: SGP, SGL, SGA, SBD, SSB, SQZ. Except when tracking GPS L1P(Y) and GPS L2P(Y), all signals from the same GNSS band present very similar SNRs, so there is no special need indicating which signal the reported SNR corresponds to.

### Satellite Usage as Reported in NMEA and ATOM Messages

Usage Status	Reported as a letter in NMEA messages	Reported as a number in ATOM messages
Satellite is not tracked	-	0
Code and carrier/Doppler data used	U	1
Code only data used	U	2
Carrier/Doppler only data used	U	3
Reserved	{blank}	4
Differential corrections problem	D	7
Smoothing counter is less than mask	S	10
No integer millisecond computed	P	12
Suspicious range problems detected	Q	13
Channels data detected as invalid	W	14
Unknown usage status	{blank}	15
No navigation data for this satellite	E	16
Satellite is below elevation mask	M	17
Satellite is marked as unhealthy in ephemeris	H	18
Satellite was intentionally disabled using \$PASHS,xxx,USE command	I	20
URA in ephemeris is not acceptable	B	21
Satellite is unhealthy according to almanac	Z	22
Too low SNR	L	23
Satellite suspected to be a ghost satellite	G	24

Correct system time for satellite is not available	T	26
Other cause	X	31

Some other letters or figures may be reported, all indicating that satellites are not used in these cases.

### Satellite Correcting Status as Reported in NMEA and ATOM Messages

Correcting Status	Reported as a letter in NMEA messages	Reported as a number in ATOM messages
Satellite is not tracked	-	0
Satellite is not corrected	N	1
SBAS corrected	W	2
DGNSS corrected	C	3
RTK/RTX corrected (single-band)	S	4
RTK/RTX corrected (multi-band)	D	5
Reserved	T	6-14
Unknown correcting status	{blank}	15

### Position Status

In most cases, the messages generated by a GNSS board appear at the output in a fixed order.

The time preference list within each group tries to follow this concept: Position first, then observables, attributes last.

There are however a few exceptions:

- Some messages are generated according to the “change” (e.g. ephemeris messages) or “event” (e.g. TTT) principle. These may not be tagged exactly to an epoch. That is why in some cases they can appear at unexpected locations in the output stream.
- LTN/ZDA goes earlier than GLL/GGA in the NME group. This is due to the highest importance given to latency and time-tag information in some applications.
- TT1/TT2/TT3 goes after RNX. This is due to the fact that TT1/TT2/TT3 messages contain time-tagged RTK results not synchronized with the receiver's current time tag. Also, when the receiver is configured in time-tagged RTK mode, some messages can also show unusual output behavior depending on local delays caused by the data link.

All the positions the receiver generates in primary position messages refer to the “default” datum. This requires some clarifications.

The Cartesian position the receiver generates (e.g. in ATM,PVT message) is defined by:

- The primary GNSS selected
- The current datum realization used for broadcast ephemeris (now IGS05 on current epoch if GPS is primary)
- The datum of the reference position (applicable to DGNSS and RTK modes only).

Generally the receiver does not know a priori what the datum of the reference position is (this depends on the data provider in the local area). As a result, the standalone position a receiver generates is tagged to IGS05 (if GPS is primary). At the same time, the differential position is computed as the reference position (in whichever datum) plus the baseline estimate (IGS05 datum).

To date, the GNSS firmware can support GPS, GLONASS or BeiDou as the primary GNSS system (see \$PASHS,PGS description). Once specified, the primary GNSS system defines a “default” receiver datum associated with the given GNSS system:

- WGS-84 (IGS05) if GPS is primary,
- PZ90.02 if GLONASS is primary
- CGCS2000 if BeiDou is primary

The three datums are quite close to each other, and yet with the following differences:

- Ellipsoid centers are about 0.5 meters apart
- Ellipsoid parameters are different

Also, the default BeiDou datum is even closer to the GPS default datum.

Different ellipsoid parameters means that a given cartesian position will be transformed into different geodetic coordinates, depending on which GNSS is used as primary.

It is therefore the end user's responsibility to be sure that the expected ellipsoid center and parameters are used so that the receiver can provide correct position estimates. For the same purpose, the end user should also make sure the same GNSS primary system is used at the base and rover.

The receiver can accept some user positions and use them in internal algorithms. These are set using the POS command.

When entering these positions, the user must realize that these will be processed differently depending on which GNSS is used as primary. These positions are always entered as geodetic coordinates (lat, lon, alt) and it is the user's responsibility to make sure the entered coordinates are expressed in the selected primary GNSS.

In most cases, the entered positions are transformed into cartesian coordinates (so they can for example be inserted into the generated reference station message). Transformation parameters are different for GPS, GLO and BDS, so cartesian positions will be "primary GNSS" dependent.

Working in RTK rover mode, it is the end user's responsibility to inquire exactly what datum is used to express the received reference position and set the primary GNSS system accordingly (using PGS).

Working in RTK base mode, it is also the end user's responsibility to inquire what datum is used to express the generated (entered) reference position and set the primary GNSS system accordingly.

Working in some NTRIP Networks (e.g. SAPOS) transmitting the so-called coordinate transformation messages, the receiver can additionally compute the position in the local system/datum. This position is available via special blocks from the ATM,PVT message. So the user can get both "default" and "local" positions simultaneously via the ATM,PVT message.

The GNSS firmware applies the following default geoid model: *NATO STANAG 4294 Navstar Global Positioning System (GPS) System Characteristics-ED 2*.

This model is used to generate the geoid-ellipsoid separation value in GGA (and similar) messages, regardless of the primary GNSS system selected (PGS command).

The GNSS firmware applies the Magnetic table corresponding to model *WMM-2005* (published 12/2004). This model is used for some position/velocity transformations applied in some NMEA messages.

The GNSS firmware applies the default ionosphere model (when no other models, e.g. SBAS, are available), extracted from the GPS navigation stream (called *Klobuchar* model).

The GNSS firmware applies proprietary troposphere models. By default, user positions reported in all messages are tagged to the antenna L1 phase center. Command ANR (for ANtenna Reduction) allows the receiver to tag all user positions to either the Antenna Reference Point or the Ground Mark. This

command does not affect reference positions generated in various correcting data streams. These positions may be tagged to either the L1 phase center or the ARP, as required by the standard used.

By default, the reported user positions are always the “best” positions, depending on the availability of correcting data at that time. The formal preference is the following (from best to worst):

- RTK
- DGNSS
- SBAS
- Standalone (autonomous)

PPP stands for “Precise Point Positioning”, which today is delivered by the RTX Trimble engine.

Using the TOP command, users can choose which level of position accuracy they wish to output.

Being of centimeter level accuracy, the reported RTK positions are very sensitive to the availability of supplementary data. They may leap by several centimeters following the change of the reference or local (own) antenna name (resulting in new PCO parameters).

The clock steering mechanism not only corrects receiver raw data, but also the reported position, for boards with high dynamics, when extrapolating it for up to 1 millisecond forward or backward.

## About the Reported DOP Values

DOP figures always correspond to the satellites actually used in the position computation, i.e. they are consistent with the sat usage status. The reported DOPs are consistent with all the generated position messages referring to the same antenna/engine/correction.

DOP values will be reported even though the position may not be computed.

Any valid DOP greater than 99.9 is output as being equal to 99.9.

DOP values will not be output if they cannot be computed.

Not delivering DOP values does not mean the position cannot be computed and vice versa.

## Differential Positions

Differential positioning is the process of correcting local receiver data (or internal PVT engine states) using some data available from external sources.

Correcting data are divided into two groups:

- *Measurement Space (MS) corrections*
- *State Space (SS) corrections*

**MS corrections** are numbers that are simply added to the respective local measurement to cancel (or at least reduce) some systematic errors. A typical example of MS corrections is either DGNSS corrections (e.g. RTCM MT 1,31) or RTK corrections (e.g. RTCM-3 MT 1004,1012).

The primary attribute of MS corrections is the position (reference) they are tagged to. MS corrections usually serve a local area around the reference position. MS corrections can correspond to either a physical or virtual reference station. RTK network corrections (MAC, FKP) can be considered as augmentations of MS corrections to extend the area of applicability.

**SS corrections** are the estimates of particular errors affecting GNSS observations. SS corrections include satellite orbit and clock corrections, ionosphere corrections, satellite signal bias estimates and some others.

SS corrections are usually not tagged to any reference position, serve wider areas compared to MS corrections, but are not always global.

For example the ionosphere correction grid can be available only for some continents, like WAAS ionosphere correction, which is available over the USA only.

The typical example of SS corrections is SBAS and L-band. Unlike precise MS corrections, the current status of SS corrections does not allow receivers to determine centimeter level positions.

But the latest progress with the so-called PPP (Precise Point Positioning) solutions using precise, almost real-time IGS products will lead to centimeter level accuracies in the future.

Depending on the organization of the PVT engine, SS corrections can be applied differently. Our receivers transform SS corrections into MS corrections (and associated accuracy figures) referring to the receiver's current position. The receiver then applies these MS corrections to receiver observations.

One of the primary attributes of any differential position is differential age. Our GNSS firmware reports age (e.g. in GGA messages) as follows:

- For time-tagged RTK positions, it is always reported as zero.

- For Fast RTK and any MS DGNSS positions, age is the difference between position time tag and the time tag of the last decoded differential corrections. It must be emphasized that the last decoded differential corrections are not necessarily applied to the current position.
- For SS differential positions (e.g. SBAS), age is computed by proprietary exclusive formulas because a number of particular correcting data that take part in position determination are tagged to different times.

## Time-Tagged RTK NMEA-Like Messages

When working with RTK MS corrections, our RTK receivers can output either time-tagged (TT) or Fast (FST) position/vector/angles estimates, using different messages.

**TT output** refers to a particular event that occurs when incoming RTK correcting data are decoded and processed inside the TT RTK engine. TT output can be irregular (data link blockage) and feature large latency (data link delays). At the same time, TT output provides the best performance in terms of accuracy.

**FST output** refers to receiver epochs. It is therefore independent of the arrival of the correcting data in the receiver. FST output is regular and features minimum possible latency. FST estimates can be degraded if some delay is introduced by the data link, increasing the age of corrections and adding error to FST estimates.

Baseline-related estimates can be divided into three groups:

- Position
- Vector
- Angles

In conventional RTK, these groups are related to each other by this simple formula:

**Position = Base + Vector**

**Angles = Vector azimuth and elevation**

Where **Base** is the reference position, provided that both **Base** and **Position** refer to the L1 antenna phase center.

In some applications, Position, Vector and Angles can have no tight relationship between each other, because generally different local and reference data are used to compute each of them.

To output position, vector and angles, we populate the existing POS, VCT and HPR messages. These messages always refer to the FST estimates. Also, all standard NMEA user messages (e.g. GGA, HDT) correspond to FST output.

In most cases, users do not need TT estimates.

## **Datum for SBAS Differential Positions**

For ITRF, strictly speaking, there is no predefined ellipsoid at all. However, it is common practice to use the GRS 1980 ellipsoid for this purpose (there is even official approval for ETRF to do so). This ellipsoid is defined as follows:

- $A=6378137$  m
- $1/f=298.257222101$

This ellipsoid (claimed as ITRF2000) has to be utilized for SBAS differential position.

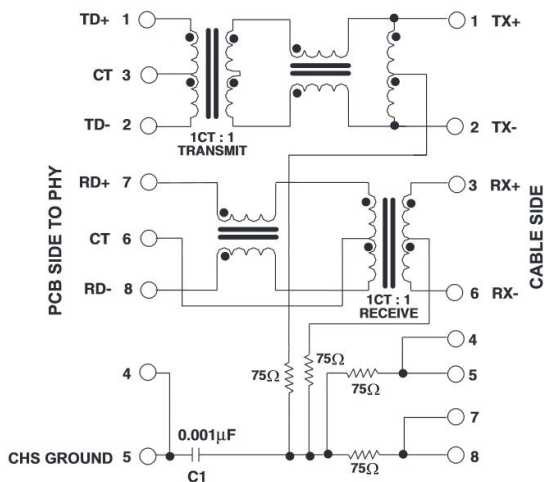
Also, this ellipsoid has to be utilized if position is claimed to be provided in the ITRS05 (or ITRS2000) reference frame, and not in the WGS84.

Applying high accuracy real-time corrections requires knowledge and proper application of datum transformations. It is crucial that you understand the reference datum of correction sources so you can maximize the accuracy of your corrected GPS data.

The Trimble TerraSync™ 3.20 software can now log measurement data in terms of a single datum (WGS-84) thus minimizing the chances of any double datum transformations when exporting data—an implied transformation at time of data collection, and a further transformation at time of export in the GPS Pathfinder® Office software.



## RJ 45 Internet Connector Schematic Diagram



NOTE: Connect CHS GND to PCB ground.

## Transitioning from Ashtech DG14 or MB100 to Trimble MB-Two

This section provides a detailed comparison between Ashtech DG14, MB100 and MB-Two OEM boards in terms of operating parameters and output messages so that you can better control the transitioning from DG14 or MB100 to MB-Two.

General:

General Parameters	DG14	MB100	MB-Two	Comments
Input Voltage Range	5 V DC $\pm 5\%$	3 V DC	3.2 to 4.5 V DC	
Antenna	1 GPS L1	2 GNSS	2 GNSS	Dual antenna for internal heading only
GPS L1	Yes	Yes	Yes	
SBAS L1	Yes	Yes	Yes	
GLONASS	No	Yes	Yes	
Galileo	No	No	Yes	
Beidou	No	No	Yes	
Coast Guard Beacon	Yes	No	No	
DGPS Base & Rover	Yes	Yes	Yes	

RTK Base & Rover	Yes	Yes	Yes	DG14 RTK L1 only
L-Band	No	No	Yes	
L2	No	GPS only	Yes	
RS232 Ports	3	2	3	
USB Port	No	1	1	
Ethernet Port	No	No	Yes	
CAN Bus	No	No	Future	
Max. Update Rate	20 Hz	20 Hz	50 Hz	MB-Two at 100 Hz in future
Fully Compatible with Evaluate	Yes	No	No	
Fully Compatible with AshCom	No	Yes	Yes	
Ethernet Web GUI to Support Web Setup and Monitoring	No	No	Yes	
PPS Out/Event In	Yes	Yes	Yes	

Binary Messages:

Binary Messages	DG14	MB100	MB-Two	Comments
ATOM Messages	No	Yes	Yes	
RTCM RT3	Rover only	Yes	Yes	
CMR & CMR+	No	Yes	Yes	
RTCM V2.3	Yes	Yes	Yes	
RTK DBEN Rover	Yes	Yes	No	
RTK DBEN Base	No	Yes	No	
ATL	No	Yes	Yes	
VTB	Yes	No	No	
MCA	Yes	Yes	No	Replaced with ATOM RNX
MBN	Yes	Yes	No	Replaced with ATOM RNX
PBN	Yes	Yes	No	Replaced with ATOM PVT or RNX
MIS	Yes	No	No	
XYZ	Yes	No	No	
DIF	Yes	No	No	
MSB	Yes	No	No	
GGB	Yes	No	No	
MCB	Yes	No	No	
CMB	Yes	No	No	
CT1	Yes	No	No	
CT2	Yes	No	No	
CT3	Yes	No	No	
SNV	Yes	Yes	No	Replaced with ATOM NAV
SAL	Yes	Yes	No	Replaced with ATOM NAV
SNW	Yes	Yes	No	Replaced with ATOM NAV
SAW	Yes	Yes	No	Replaced with ATOM NAV

## NMEA or NMEA-like, ASCII Messages:

ASCII Messages	DG14	MB100	MB-Two	Comments
AIM	Yes	No	No	
ALM	No	Yes	Yes	
AT2	No	No	No	
ATT	No	Yes	No	
CRT	Yes	No	No	
DDM	No	Yes	Yes	Returns decoded differential messages
DTM	Yes	No	No	
GBS	No	Yes	Yes	Use GBS for RAIM
GDC	Yes	No	No	
GLL	Yes	Yes	Yes	
GMP	No	Yes	Yes	Format described in MB-Two manual only
GNS	Yes	No	Yes	MB100 GNS configuration command and query
GRS	Yes	Yes	Yes	
GSA	Yes	Yes	Yes	
GSN	Yes	No	No	
GSV	Yes	Yes	Yes	
GXP	Yes	No	No	
HDT	No	Yes	Yes	
HPL	Yes	No	No	
LTN	Yes	Yes	Yes	
MSG	Yes	Yes	Yes	
POS	Yes	Yes	Yes	
PTT	Yes	Yes	Yes	
RMC	Yes	Yes	Yes	
RRE	Yes	Yes	Yes	
SAT	Yes	Yes	Yes	
SUD	Yes	No	No	
TCM	Yes	No	No	
THS	No	Yes	Yes	
TTT	Yes	Yes	Yes	
UKO	Yes	No	No	
UTM	Yes	No	No	
VCT	No	Yes	Yes	
VEC	Yes	Yes	No	
VTG	Yes	Yes	Yes	
XMG	Yes	No	No	
ZDA	Yes	Yes	Yes	
GLL	Yes	Yes	Yes	

For more information on ATOM messages, please refer to the *ATOM Reference Manual*.



**Symbols**

\$PASHQ,ALR 320  
 \$PASHQ,ANH 321  
 \$PASHQ,ANP 322  
 \$PASHQ,ANP,OUT 323  
 \$PASHQ,ANP,OWN 323  
 \$PASHQ,ANP,RCV 324  
 \$PASHQ,ANP,REF 325  
 \$PASHQ,ANR 326  
 \$PASHQ,ANT 326  
 \$PASHQ,ARA 327  
 \$PASHQ,ARR 329  
 \$PASHQ,AST 331  
 \$PASHQ,ATH 332  
 \$PASHQ,ATL 333  
 \$PASHQ,ATT 335  
 \$PASHQ,AVR 336  
 \$PASHQ,BDS 338  
 \$PASHQ,BRD 341  
 \$PASHQ,BRV 341  
 \$PASHQ,CAN 342  
 \$PASHQ,COO,REF 343  
 \$PASHQ,CST 344  
 \$PASHQ,CST,USR 345  
 \$PASHQ,d1,BLN 339  
 \$PASHQ,DDM 346  
 \$PASHQ,DDN 348  
 \$PASHQ,DDS 349  
 \$PASHQ,DFC 351  
 \$PASHQ,DIF,BDS 351  
 \$PASHQ,DIF,NET 352  
 \$PASHQ,DIF,SBA 353  
 \$PASHQ,DIP 354  
 \$PASHQ,DRD 355  
 \$PASHQ,DRI 355  
 \$PASHQ,DSY 356  
 \$PASHQ,DTH 358  
 \$PASHQ,DTM 357  
 \$PASHQ,DYN 359  
 \$PASHQ,ELM 360  
 \$PASHQ,ETH 361  
 \$PASHQ,FIL,CUR 362  
 \$PASHQ,FIL,LST 363  
 \$PASHQ,FIL,STS 364  
 \$PASHQ,FTP 365  
 \$PASHQ,GAL 366  
 \$PASHQ,GBS 367  
 \$PASHQ,GEM 369  
 \$PASHQ,GGA 370  
 \$PASHQ,GGKX 372  
 \$PASHQ,GLB 373  
 \$PASHQ,GLL 374  
 \$PASHQ,GLO 375  
 \$PASHQ,GLx 376  
 \$PASHQ,GMP 377  
 \$PASHQ,GNS 379  
 \$PASHQ,GPS 380  
 \$PASHQ,GRS 381  
 \$PASHQ,GSA 382  
 \$PASHQ,GST 385  
 \$PASHQ,GSV 386  
 \$PASHQ,HDT 388  
 \$PASHQ,HPR 389  
 \$PASHQ,LBN 391  
 \$PASHQ,LBN,BEM 393  
 \$PASHQ,LCK 394  
 \$PASHQ,LCS 394  
 \$PASHQ,LOG 395  
 \$PASHQ,LOG,LST 396  
 \$PASHQ,LOG,PAR 397  
 \$PASHQ,LTN 398  
 \$PASHQ,MDP 399  
 \$PASHQ,MEM 400  
 \$PASHQ,NPT 400  
 \$PASHQ,NTP 401  
 \$PASHQ,NTR 402  
 \$PASHQ,NTR,MTP 403  
 \$PASHQ,NTR,TBL 404  
 \$PASHQ,OBS 406  
 \$PASHQ,OCC 406  
 \$PASHQ,OPTION 407, 411  
 \$PASHQ,OPTION,EXP 410  
 \$PASHQ,OUT,GSF 412  
 \$PASHQ,OUT,PGN 415  
 \$PASHQ,PGS 417  
 \$PASHQ,PHE 417  
 \$PASHQ,PIN 418  
 \$PASHQ,POP 419  
 \$PASHQ,POS 420, 422  
 \$PASHQ,PPP 423  
 \$PASHQ,PPS 424  
 \$PASHQ,PRT 425  
 \$PASHQ,PSP 426, 446  
 \$PASHQ,PTT 427  
 \$PASHQ,QZS 429  
 \$PASHQ,RCA 430  
 \$PASHQ,RCP,OWN 430  
 \$PASHQ,RCP,REF 431  
 \$PASHQ,RCS 432  
 \$PASHQ,RDP,CHT 433  
 \$PASHQ,RDP,LVL 472  
 \$PASHQ,RDP,PAR 435  
 \$PASHQ,RDP,PWR 438

\$PASHQ,RDP,TYP 439  
\$PASHQ,REC 440  
\$PASHQ,RFM 441  
\$PASHQ,RID 442  
\$PASHQ,ROT 445  
\$PASHQ,RTK 447  
\$PASHQ,RTX 448  
\$PASHQ,SAT 450  
\$PASHQ,SBA 452  
\$PASHQ,SBD 452  
\$PASHQ,SGA 453  
\$PASHQ,SGL 454  
\$PASHQ,SGP 456  
\$PASHQ,SIT 457  
\$PASHQ,SLB 457  
\$PASHQ,SNS 458  
\$PASHQ,SQZ 459  
\$PASHQ,SSB 461  
\$PASHQ,STI 462  
\$PASHQ,TCP 463  
\$PASHQ,TEM 464  
\$PASHQ,THS 465  
\$PASHQ,TOP 466  
\$PASHQ,TRS 466  
\$PASHQ,UDP 468  
\$PASHQ,UPL 469  
\$PASHQ,UPL,LOG 470  
\$PASHQ,VCR 473  
\$PASHQ,VCT 476  
\$PASHQ,VEL 478  
\$PASHQ,VERSION 479  
\$PASHQ,VLP 480  
\$PASHQ,VTG 480  
\$PASHQ,WEB 481  
\$PASHQ,XDR 413  
\$PASHQ,ZDA 482  
\$PASHR,TTT 467  
\$PASHS,,USE 215  
\$PASHS,3DF 139  
\$PASHS,3DF,ANG 141  
\$PASHS,3DF,CLB 141  
\$PASHS,3DF,MXB 142  
\$PASHS,3DF,OFS 143  
\$PASHS,3DF,RST 145  
\$PASHS,3DF,Vx 146  
\$PASHS,ALR,ACK 147  
\$PASHS,ANH 147  
\$PASHS,ANP,OUT 148  
\$PASHS,ANP,OWN 150  
\$PASHS,ANP,REF 151  
\$PASHS,ANR 152  
\$PASHS,ANT 154, 515

\$PASHS,ATH,PWD 156  
\$PASHS,ATL 156  
\$PASHS,ATM 158  
\$PASHS,ATM,ALL 161  
\$PASHS,BDS 161  
\$PASHS,BLN 163  
\$PASHS,BRD 165  
\$PASHS,BRV 166  
\$PASHS,BRV,RST 167  
\$PASHS,CAN,OFF 168  
\$PASHS,CAN,ON 168  
\$PASHS,CAN,PAR 169  
\$PASHS,CMD,LOD 170  
\$PASHS,CMD,WTI 171  
\$PASHS,CMR 172  
\$PASHS,CMR,ALL 173  
\$PASHS,CST,MTP,ADD 176  
\$PASHS,CST,MTP,DEL 178  
\$PASHS,CST,OFF 176  
\$PASHS,CST,ON 175  
\$PASHS,CST,PAR 178  
\$PASHS,CST,RST 180  
\$PASHS,CST,USR,ADD 181  
\$PASHS,CST,USR,DEL 182  
\$PASHS,DDN,PAR 185  
\$PASHS,DDN,SET 187  
\$PASHS,DFC,DEL 182  
\$PASHS,DFC,GET 183  
\$PASHS,DFC,SET 184  
\$PASHS,DFC,TST 185  
\$PASHS,DIF,BDS 187  
\$PASHS,DIF,NET 188  
\$PASHS,DIF,SBA 189  
\$PASHS,DIFF,OFF 251  
\$PASHS,DIP,OFF 190  
\$PASHS,DIP,ON 191  
\$PASHS,DIP,PAR 191  
\$PASHS,DRD 193  
\$PASHS,DRI 194  
\$PASHS,DSY 194  
\$PASHS,DYN 195  
\$PASHS,ELM 196  
\$PASHS,ENC 197  
\$PASHS,ETD,PAR 198  
\$PASHS,ETH,OFF 199  
\$PASHS,ETH,ON 199  
\$PASHS,ETH,PAR 200  
\$PASHS,FIL,CPY 201  
\$PASHS,FIL,DEL 202  
\$PASHS,FIL,GET 203  
\$PASHS,FIL,WRT 205  
\$PASHS,FMT 207

\$PASHS,FTP,OFF 207  
\$PASHS,FTP,PAR 208  
\$PASHS,FTP,PUT 209  
\$PASHS,GAL 211  
\$PASHS,GEM 212  
\$PASHS,GL... 214  
\$PASHS,GLB 212  
\$PASHS,GLO 213  
\$PASHS,GPS 217  
\$PASHS,GSF 217  
\$PASHS,GSF,ALL 218  
\$PASHS,INI 219  
\$PASHS,LBN,BEM 220  
\$PASHS,LBN,SAT 222  
\$PASHS,LBN,USE 223  
\$PASHS,LCK,MOD 224  
\$PASHS,LCK,OFF 225  
\$PASHS,LCK,ON 226  
\$PASHS,LCS 227  
\$PASHS,LOG,DEL 228  
\$PASHS,LOG,PAR 228  
\$PASHS,MDP 229  
\$PASHS,MEM 230  
\$PASHS,MET,OWN 230  
\$PASHS,MET,REF 230  
\$PASHS,MSG 232  
\$PASHS,NME 233  
\$PASHS,NME,ALL 235  
\$PASHS,NME,ALR 236, 237  
\$PASHS,NPT 238  
\$PASHS,NTP 239  
\$PASHS,NTR,LOD 240  
\$PASHS,NTR,MTP 241  
\$PASHS,NTR,PAR 242  
\$PASHS,OBS 243  
\$PASHS,OCC 245  
\$PASHS,OPTION 246  
\$PASHS,OUT,ALL 250  
\$PASHS,OUT,x,ON/OFF 249  
\$PASHS,PAR,LOD 252  
\$PASHS,PAR,SAV 253  
\$PASHS,PEM 254  
\$PASHS,PGN 254  
\$PASHS,PGN,ALL 255  
\$PASHS,PGS 256  
\$PASHS,PHE 259  
\$PASHS,PIN 260  
\$PASHS,POP 264  
\$PASHS,POS 265  
\$PASHS,POS,CUR 267  
\$PASHS,POS,MOV 269  
\$PASHS,POS,REF 174

\$PASHS,PPP 270  
\$PASHS,PPP,RST 270  
\$PASHS,PPP,SRC 271  
\$PASHS,PPS 271  
\$PASHS,PRT 272  
\$PASHS,PWR,OFF 273  
\$PASHS,QZS 274  
\$PASHS,RCP,OWN/REF 274  
\$PASHS,RCP,REF 275  
\$PASHS,RDP,PAR 276  
\$PASHS,RDP,TYP 281  
\$PASHS,REC 282  
\$PASHS,RFM 283  
\$PASHS,RST 284  
\$PASHS,RT2 285  
\$PASHS,RT2,ALL 286  
\$PASHS,RT3 287  
\$PASHS,RT3,ALL 289  
\$PASHS,RTK 289  
\$PASHS,RTK,RST 290  
\$PASHS,RTK,STI 291  
\$PASHS,RTX,DTM 292  
\$PASHS,RTX,KPI 295  
\$PASHS,RTX,MOD 297  
\$PASHS,RTX,RST 298  
\$PASHS,RTX,SRC 299  
\$PASHS,RXC,PAR 301  
\$PASHS,RXC,RUN 303  
\$PASHS,SBA 304  
\$PASHS,SIT 305  
\$PASHS,SNS 305  
\$PASHS,STI 308  
\$PASHS,TCP,PAR 309  
\$PASHS,TCP,UIDM 310  
\$PASHS,TOP 311  
\$PASHS,UDP 312  
\$PASHS,UPL,PAR 313  
\$PASHS,UPL,UPG 314  
\$PASHS,VIP 315  
\$PASHS,WEB,OWN 316  
\$PASHS,WEB,PAR 316  
\$PASHS,ZDA 317

## **Numerics**

3DF 139  
3DF,ANG 141  
3DF,CLB 141  
3DF,MXB 142  
3DF,OFS 143  
3DF,RST 145  
3DF,Vx 146

## **A**

Accessories 9  
ACK 127  
Active edge (event marker) 259, 417  
Adaptive 195, 359  
Adding L-band satellite manually 222  
Alarm message 236  
Alarms 147  
Alarms (listing) 320  
Alarms message (NMEA-like) 237  
Alerts 147  
ALR 320  
ANH 147, 321  
ANP 322  
ANP,OUT 148, 323  
ANP,OWN 150, 323  
ANP,RCV 324  
ANP,REF 151, 325  
ANR 152, 326  
ANT 154, 326  
Antenna height 147, 321, 326  
Antenna inputs 27  
Antenna name and status of received base 324  
Antenna parameters 322  
Antenna reduction 326  
Antenna reduction mode 152  
Antenna status 331  
Antenna used at the base 325  
Antenna used to receive L-band 212  
Antenna/receiver attributes 133  
Antennas 16  
Anti-theft parameters (listing) 332  
Application setup commands 129  
ARA 327  
ARP 153, 154, 266, 296, 514  
ARR 329  
ASH 197  
AshCom 16, 23  
Assigning function to programmable pin on I/O connector 260  
AST 28, 331  
ATH 332  
ATL 333  
ATL command 156  
ATL messages 156  
ATM 158, 196  
ATM,ALL 161  
ATOM messages 158  
ATT 335  
Attitude 389

Attitude (auto-calibration) 141

Attitude (reset) 145

Attitude components 327

Auto-calibration 141

AVR 131, 234, 336

## **B**

Balanced antenna signal levels in DUO mode 28

Base antenna (naming) 151

Base data stream (preferred) 291

Base position entered manually 422

Base position used in rover 343

Baseline definitions & status 339

Baseline elevation (max) 141

Baseline length (choice of) 29

Baselines and respective base/rover data sources 163

Basic supply 7

Baud rate 425

BDS 338

BeiDou 12, 187

BeiDou correcting data 351

BeiDou satellites status 452

BeiDou tracking 161

BeiDou tracking status 338

BLN 163, 339

Board configurations 7

Bottom view 6

BRD 165, 341

BRV 166, 341

BRV,RST 167

## **C**

Calibration 141

CAN 342

CAN bus 18, 20

CAN bus parameters 169

CAN port 168

CAN,OFF 168

CAN,ON 168

CAN,PAR 169

Caster IP address 242

Changing the administrator profile 316

Channels 12

CMD,LOD 170

CMD,WTI 171

CMR 172, 196

CMR messages 172

CMR,ALL 173

CMR/CMR+ messages 173

CMR+ 172, 196

Cold start 219



Communication ports *135*  
Communicator *16*  
Computed position *420*  
Configuring the receiver from a PAR file *252*  
Connection profiles *481*  
COO,REF *174, 343*  
Copying files from internal to external memory *201*  
Course over ground *480*  
Create file from the default configuration *183*  
Crossover cable *22*  
Crossover cable (Ethernet) *73*  
CST *344*  
CST,MTP,ADD *176*  
CST,MTP,DEL *178*  
CST,OFF *176*  
CST,ON *175*  
CST,PAR *178*  
CST,RST *180*  
CST,USR *345*  
CST,USR,ADD *181*  
CST,USR,DEL *182*

## **D**

D (port D) *229*  
Daisy chain *194, 356*  
Data recording status *432*  
Date & time *317*  
Datum Reference *357*  
DDM *346*  
DDN *348*  
DDN,PAR *185*  
DDN,SET *187*  
DDS *349*  
Debug data recording *333*  
Debug messages *156*  
Default configuration file (read content) *351*  
default\_config.cmd *184*  
Delete default configuration file *182*  
Deleting files and directories *202*  
Deleting log files *228*  
DFC *351*  
DFC,DEL *182*  
DFC,GET *183*  
DFC,SET *184*  
DFC,TST *185*  
DHCP *71, 73*  
Die temperature *464*  
DIF,BDS *351*  
DIF,NET *352*  
DIF,SBA *353*

Source of SBAS corrections *353*  
Differential Decoder Message *346*  
Differential decoder status *349*  
Differential messages *135*  
Differential messages (disable all) *251*  
Dimensions *17*  
DIP *354*  
DIP,OFF *190*  
DIP,ON *191*  
DIP,PAR *191*  
Direct IP *354*  
Direct IP connection *190, 191*  
Direct IP parameters *191*  
Disabling all ATOM messages *161*  
Disabling all GSOF messages on a port *218*  
Disabling all NMEA 2000 messages *255*  
Disabling satellite tracking *215*  
Dithering mode *358*  
Downloading a file from the receiver *203*  
DRD *193, 355*  
DRI *194, 355*  
DSY *194, 356*  
DTH *358*  
DTM *357*  
Duration (of data recording in G-file) *355*  
DYN *195, 359*  
Dynamic IP address *185, 187*  
Dynamics *195, 312*  
DynDNS *185, 187*  
DynDNS parameters *348*

## **E**

Editing a log file *395*  
Editing the firmware upgrade log file *470*  
EGM96 *212, 369*  
Elevation mask *196, 254*  
Elevation mask (position) *416*  
ELM *196, 360*  
Embedded NTRIP caster *175, 176, 178, 180, 181, 182*  
Enable/disable Galileo *211*  
Enabling the Web Server *316*  
ENC *197*  
Ending data transfer with FTP *207*  
Enter owner information *316*  
ETD,PAR *198*  
ETH *361*  
ETH,OFF *199*  
ETH,ON *199*  
ETH,PAR *200*  
Ethernet *18, 361*  
Ethernet driver parameters *198*

Ethernet parameters *200*  
Ethernet port *199*  
Ethernet protection *22*  
Ethernet-related commands *135*  
Evaluation kit *9, 21*  
Event marker *467*  
Event marker (active edge) *259, 417*  
Expiration date *509*  
Expiration dates *410*  
Extended Attitude message *445*  
External event signal *18*

## **F**

Fallback caster *344*  
Field delimiter *127*  
FIL,CPY *201*  
FIL,CUR *362*  
FIL,DEL *202*  
FIL,GET *203*  
FIL,LST *363*  
FIL,STS *364*  
FIL,WRT *205*  
Firmware (upgrade) *508*  
Firmware options *7, 246, 407*  
Firmware upgrade *508*  
Firmware version *479*  
Foot print beam ellipses *220*  
Formatting *207*  
FST (Fast) *525*  
FTP *365*  
FTP settings *208*  
FTP,OFF *207*  
FTP,PAR *208*  
FTP,PUT *209*

## **G**

GAL *211, 366*  
Galileo *12*  
Galileo (enable/disable) *211*  
GALILEO satellites status *453*  
GALILEO tracking status *366*  
Gateway *74*  
GBS *367*  
GEM *369*  
General purpose commands *134*  
Geoid model *212, 369, 371, 378*  
G-file being recorded *362*  
GGA *233, 370*  
GGKX *372*  
GLB *212, 373*  
GLL *233, 374*  
GLO *213, 375*  
GLONASS *12*

GLONASS satellites status *454*  
GLONASS tracking *213*  
GLx *214, 376*  
GM *514*  
GMP *377*  
GNS *379*  
GNS,USE (,USE to be accurate) *215*  
GNSS Fix Data *379*  
GNSS network-related commands *136*  
GPS *12, 380*  
GPS & SBAS satellites status *456*  
GPS tracking *217*  
Ground speed *480*  
GRS *233, 381*  
GSA *233*  
GSOF messages (controlling output of) *217*  
GSOF messages output *412*  
GST *233, 385*  
GSV *233, 386*

## **H**

Hardware options *7*  
HDT *388*  
Header *127*  
Heading *335*  
Hot Standby RTK *13*  
HPR *389*

## **I**

I/O connector *18, 28, 260*  
I/O data formats *13*  
I/O Interface *16*  
IGS antenna source table *150*  
INI *219*  
Insert wait times while a command file is executed *171*  
Installation *17*  
Installed firmware options *410*  
IP address (board's) *71*  
ITRF *265, 295*  
ITRF2008 *292*

## **J**

Jumpers *23*

## **L**

L1, L2 *214*  
Latency *398*  
L-Band *12*  
L-band reception antenna input *373*  
L-band satellite (adding manually) *220, 222*  
L-band satellites (user defined) *393*  
L-Band satellites status *457*  
L-band setup *391*

- L-band tracking 223
- LBN 391
- LBN,BEM 220, 393
- LBN,SAT 222
- LBN,USE 223
- LCK 394
- LCK,MOD 224
- LCK,OFF 225
- LCK,ON 226
- LCS 227, 394
- LED indicator 19
- Listing files stored in receiver memory or USB key 363
- Listing log files 396
- Listing Web Server settings 481
- LNA 15, 23, 28
- Local antenna 323
- Local antennas (naming) 150
- Local coordinate system 227, 394
- Lock 224, 226, 394
- LOG 395
- Log & alarm-related commands 136
- Log file settings 397
- LOG,DEL 228
- LOG,LST 396
- LOG,PAR 228, 397
- LTN 398
- LV-TTL 20
- Lx signals (enabling/disabling) 214

## **M**

- Magnetics 22
- Mask angle 254
- Max elevation (for baselines) 141
- MCA 196
- MDP 229, 399
- MEM 230, 400
- Memory & data recording commands 137
- Memory device 400
- Memory device used 230
- Memory reset codes 220
- Memory status 364
- MET,OWN 230
- MET,REF 230
- Meteorological parameters 230
- Mount point (connection to) 241
- Mount points (allowed to users) 181
- Mount points (create/modify) 176
- Mount points (delete) 178
- MPC 196
- M-Sensor setting 458
- MSG 232

## **N**

- NAK 127
- NATO standard mean seal level 371, 378
- Network rover 188
- NME 233
- NME messages 130
- NME,ALL 235
- NME,ALR 236, 237
- NMEA 2000 messages (controlling output of) 254
- NMEA 2000 messages output 415
- NMEA, NMEA-like messages 233, 235
- NPT 238, 400
- NTP 239, 401
- NTP server 239, 401
- NTR 402
- NTR,LOD 240
- NTR,MTP 241, 403
- NTR,PAR 242
- NTR,TBL 404
- NTRIP caster 403
- NTRIP caster parameters 344
- NTRIP caster source table 240
- NTRIP caster user (delete) 182
- NTRIP caster users 345
- NTRIP mount point 403
- NTRIP settings 242, 402
- NTV 197

## **O**

- OBS 406
- Observations (preference) 243
- Observations usage 406
- OCC 245, 406
- Occupation 245, 406
- Offset (heading) 29
- Offset (pitch) 30
- Offset (roll) 30
- Offsets (attitude) 143
- Offsets (entering) 33
- OPTION 246, 407
- OPTION,EXP 410
- OUT 411
- OUT,ALL 250
- OUT,DIFF,OFF 251
- OUT,GSF 412
- OUT,MSG 413
- OUT,PGN 415
- OUT,x,ON/OFF 249
- Output messages 413
- Owner information 481

## **P**

PAR file *252, 253*  
PAR,LOD *252*  
PAR,SAV *253*  
Password (anti-theft) *156*  
PC1 *153, 266, 296*  
P-Cube *1*  
PEM *254, 416*  
Periodic message output (suspend/resume) *249*  
Periodic message status *411*  
Periodic messages (disabling all) *250*  
PGS *256, 417*  
PHE *259, 417*  
PIN *260, 418*  
Pitch *335*  
POP *264, 419*  
Port D communication mode *399*  
POS *265, 422*  
POS message  
    how position solution type is tagged  
        *238, 400*  
POS,CUR *267*  
POS,MOV *269*  
Position of Physical Reference Position  
Received *426, 446*  
PPP *270, 423*  
PPP service (reset) *270*  
PPP services *270, 271*  
PPP,RST *270*  
PPP,SRC *271*  
PPS *271, 424*  
PPS (1 PPS) *16*  
PPS signal *18*  
PPS time tag *427*  
Primary GNSS *256*  
Primary GNSS system *417*  
Profiles *481*  
Programmable pin on I/O connector *418*  
PRT *272*  
Pseudo-range error statistics *385*  
PSP *426, 446*  
PTT *427*  
PWR,OFF *273*

## **Q**

Query commands *125*  
QZS *274, 429*  
QZSS *12*  
QZSS satellites status *459*  
QZSS tracking *274, 429*

## **R**

Radio channel settings *433*  
Radio parameters *435*  
Radio power table *438*  
Radio settings *276*  
Radio type *281, 439*  
Radio-related commands *138*  
RAIM *367*  
Raw data recording *282*  
Raw data recording duration *193*  
Raw data recording rate *194, 355*  
Raw data recording status *440*  
Raw/Differential Message commands *130*  
RCA *131, 430*  
RCP,OWN *430*  
RCP,OWN/REF *274*  
RCP,REF *275, 431*  
RCS *432*  
RDP,CHT *433*  
RDP,LVL *472*  
RDP,PAR *276, 435*  
RDP,PWR *438*  
RDP,TYP *281, 439*  
Reading FTP status and settings *365*  
Reading RTK Bridge settings *341*  
Reading the internal update rate *419*  
Reading the radio reception level *472*  
Reading the receiver name *430*  
Reading the reference receiver name *431*  
Reading the status and settings of the FTP server providing firmware upgrades *469*  
REC *282, 440*  
Received CMR Type 2 Attribute *430*  
Receiver Autonomous Integrity Monitoring *367*  
Receiver configuration commands *137*  
Receiver dynamics *359*  
Receiver identification *442*  
Receiver name *274*  
Reference position ("one-shot" current position) *267*  
Reference position (current position, potentially ever changing) *269*  
Reference position (entering at rover) *174*  
Reference receiver name (base receiver name) *275*  
Relative mode status *341*  
Relative processing *166*  
Reset (attitude computation) *145*  
Reset (relative positioning) *167*  
Reset processor and memory *219*

Reset receiver parameters *284*  
Reset RTK/RTX computation *290*  
Reset signal *18*  
Resetting embedded NTRIP caster *180*  
Resetting RTX computation *298*  
RFM *283, 441*  
RINEX conversion parameters *301*  
RINEX header *303*  
RINEX header definition *301*  
Ring file memory *283, 441*  
RMC *234*  
Roll *335*  
ROT *445*  
RST *284*  
RT2 *196*  
RT3 *196, 287*  
RT3,ALL *289*  
RTCM 2.3 messages *285, 286*  
RTCM 3.1 messages *287, 289*  
RTK *447*  
RTK (command) *289*  
RTK Bridge *165*  
RTK processing status *447*  
RTK,RST *290*  
RTK,STI *291*  
RTX *448*  
RTX - Computing RTX position *67*  
RTX processing status *448*  
RTX,DTM *292*  
RTX,KPI *295*  
RTX,MOD *297*  
RTX,RST *298*  
RTX,SRC *299*  
Run a series of commands *170*  
Running attitude processing on set baselines *139*  
Running RTK processing on set baselines *289*  
RXC,PAR *301*  
RXC,RUN *303*

**S**

SAT *450*  
Saving the receiver configuration to a PAR file *253*  
SBA *304, 452*  
SBAS *12*  
SBAS messages (enable/disable) *304*  
SBAS satellites status *461*  
SBAS tracking status *452*  
SBD *452*  
Semi-major axis *385*

Semi-minor axis *385*  
Serial commands *125*  
Serial connection *25*  
Set commands *125*  
Set date & time *317*  
Set default configuration file *184*  
Set FTP providing firmware upgrades *313*  
Setting primary GNSS system *256*  
Setting the log file *228*  
SGA *453*  
SGL *454*  
SGP *456*  
SHMP *154*  
SIT *305, 457*  
Site name *305, 457*  
Size *15*  
SLB *457*  
SNS *458*  
Source of correction data *352*  
Source table *176, 178, 404*  
SP Loader *507*  
Space required *17*  
SPT *153, 266, 296*  
SQZ *459*  
SSB *461*  
STANAG *212, 369*  
Starting embedded NTRIP caster *175*  
Startup protection *13, 63*  
Station ID *308, 462*  
STI *308, 462*  
Stock product *12*  
Stopping embedded NTRIP caster *176*  
Subnetwork mask *74*

**T**

TCP *463*  
TCP,PAR *309*  
TCP,UID *310*  
TCP/IP *20*  
TCP/IP authentication *310*  
TCP/IP connection *25*  
TCP/IP server settings *309, 463*  
Tectonic plates *292*  
TEM *464*  
TERIASat *270, 466*  
Test default configuration file *185*  
THS *465*  
Time & date *482*  
Tolerance (on baseline lengths) *142*  
TOP *311, 466*  
Top view *6*  
Tracked signals on different bands *376*

Tracking commands *129*  
Transport mode *197*  
Trimble RTX *270, 292, 297, 299*  
True heading and status *465*  
TT (TimeTagged) *525*  
Turning receiver on *273*  
Type of output position *311*  
Type of position delivered *466*

## **U**

UDP *312, 468*  
ULP,UPG *314*  
Unlock *225*  
Update rates *264*  
Upgrade (firmware) *508*  
Upgrade procedure (firmware) *508*  
Upgrade receiver firmware *508*  
Upgrade receiver firmware from FTP *314*  
UPL *469*  
UPL,LOG *470*  
UPL,PAR *313*  
Uploading raw data files to FTP *209*  
USB 2.0 *20*  
User messages *232*  
User-defined dynamic model *312, 468*

## **V**

Validity period *480*  
VCR *473*  
VCT *476*  
Vector components *146, 329, 473, 476*  
Vector, relative processing *166*  
VEL *478*  
Velocity & accuracy *478*  
VERSION *479*  
Vertical measurement *147*  
VIP *315*  
Virtual antenna *323*  
Virtual antenna (specify name) *148*  
Virtual port *315*  
VLP *480*  
VTG *234, 480*

## **W**

Warm start *219*  
Warranty (end of) *509*  
WEB *481*  
WEB,OWN *316*  
WEB,PAR *316*  
Weight *15*  
Writing a file *205*

## **Z**

Z-Blade *1*  
ZDA *234, 317, 482*