smart positioning

REV 1.1

NMEA Manual for Fastrax IT600 Series GNSS receivers

NMEA command manual for modules based on ST TESEO II chipset

December 27, 2011

Fastrax Ltd.

Fastrax

TRADEMARKS

Fastrax is a registered trademark of Fastrax Ltd.

All other trademarks are trademarks or registered trademarks of their respective holders.

COPYRIGHT

DISCLAIMER

This document is compiled and kept up-to-date as conscientiously as possible. Fastrax Ltd. cannot, however, guarantee that the data are free of errors, accurate or complete and, therefore, assumes no liability for loss or damage of any kind incurred directly or indirectly through the use of this document. The information in this document is subject to change without notice and describes only generally the product defined in the introduction of this documentation. Fastrax products are not authorized for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. Fastrax will not warrant the use of its devices in such applications.

CHANGE LOG

Rev.	Notes	Date
1.0	Initial revision	2011-11-18
1.1.	Added NMEA speed, message mask and Notch Filter configuration	2011-12-27

CONTENTS

COMPLEMENTARY READING

The following reference documents are complementary reading for this document:

Ref. #	File name	Document name
1		NMEA0183 specification

1 NMEA OUTPUT MESSAGES

1.1 Standard NMEA Messages

Standard messages that are defined within the “NMEA 0183” Specification.

Standard NMEA Messages are defined in the “NMEA 0183” Standard, issued from the “National Marine Electronics Association”. The messages follow Rev. 3.1 published January 2002. NMEA0183 refers to it as Sentences (single line message) and Messages (multiple line messages).
Default configuration output NMEA messages **RMC, GGA, VTG, and GSA**.

1.2 Proprietary Messages

The IT600 can provide additional messages with more detailed data content. This is required to transmit GPS and System information content which is not defined in the NMEA standard output.

Proprietary Messages from STMicroelectronics start with “$PSTM”
ST proprietary messages outputted by default are **$PSTMSBAS and $PSTMSBASMCH**.

Output can be configured to have user defined set of output messages.

Outputted messages are utilized in GPS Workbench 4 to visualize the GPS and GLONASS data. Default output messages can cover most of the applications.

1.3 GPGGA

Global Positioning System Fixed data

NMEA message list bitmask: 0x1

Format: $GPGGA,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<GPSQual>,<Sats>,<HDOP>, <Alt>,<AltVal>,<GEOSep>,<GEOVal>,<DGPSAge>,<DGPSRef>, <checksum><cr><lf>

Parameter	Format	Description
Timestamp	Hhmmss	UTC Time of GPS Sample, example: 160836
Lat	DDMM.MMM	Lat in Degree-Minutes.partsMinutes: 4208.536
N/S	“N” or “S”	Lat Direction: North or South
Long	DDMM.MMM	Long in Degree-Minutes.partsMinutes: 1105.345
E/W	“E” or “W”	Long Direction: East or West
GPSQual	Decimal, 1digit	0 = invalid 1 = GPS 2 = DGPS
Sats	Decimal, 2 digits	Satellites in view: example: 8
HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, max: 99.0
Alt	Decimal, 5 digits	Height above WGS84

		Elipsoid, max: 999.99
Alt-Val	"M"	Height measure in "M" = meters
GEOSep		
GEOVal		
DGPSAge		
DGPSRef		
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$GPGGA,183417.366,4814.03970,N,1128.52205,E,0,00,99.0,495.53,M,47.6,M,,*53

1.4 GPGGA5

Global Positioning System Fixed data (5 digits for latitude and longitude fractional parts)
NMEA message list bitmask: 0x2

Format:
$GPGGA,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<GPSQual>,<Sats>,<HDOP>,
<Alt>,<AltVal>,<GEOSep>,<GEOVal>,<DGPSAge>,<DGPSRef>, <checksum><cr><lf>

Parameter	Format	Description
Timestamp	Hhmmss	UTC Time of GPS Sample, example: 160836
Lat	DDMM.MMMMM	Lat in Degree-Minutes.partsMinutes: 4208.53683
N/S	"N" or "S"	Lat Direction: North or South
Long	DDMM.MMMMM	Long in Degree-Minutes.partsMinutes: 1105.34567
E/W	"E" or "W"	Long Direction: East or West
GPSQual	Decimal, 1digit	0 = invalid 1 = GPS 2 = DGPS
Sats	Decimal, 2 digits	Satellites in view: example: 8
HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, max: 99.0
Alt	Decimal, 5 digits	Height above WGS84 Elipsoid, max: 999.99
Alt-Val	"M"	Height measure in "M" = meters
GEOSep		
GEOVal		
DGPSAge		
DGPSRef		

checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$GPGGA5,183417.366,04814.03970,N,01128.52205,E,0,00,99.0,495.53,M,47.6,M,,*53

1.5 GPGLL

Geographic Positioning Latitude / Longitude
NMEA message list bitmask: 0x100000

Format:
$GPGLL,<Lat>,<N/S>,<Long>,<E/W>,<Timestamp>,<Status>,
<checksum><cr><lf>

Parameter	Format	Description
Lat	DDMM.MMMM	Latitude in Degree-Minutes.partsMinutes: 4208.5368
N/S	"N" or "S"	Latitude Direction: North or South
Long	DDMM.MMMM	Longitude in Degree-Minutes.partsMinutes: 1105.3456
E/W	"E" or "W"	Longitude Direction: East or West
Timestamp	hhmmss	UTC Time of GGL Sample, example: 160836
Status	"A"	Validity of Data: "A" = valid, "V" = invalid
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

-

1.6 GSA

GPS DOP and Active Satellites. The talker ID for this NMEA message depends on the enabled constellation as follows:
"GP" if only GPS constellation is enabled.

"GL" if only GLONASS constellation is enabled.

"GN" if both GPS and GLONASS constellation are enabled. This talker ID is used even if it is forced to be used in the configuration block (see Application ON/OFF parameter Bit 20).

NMEA message list bitmask: 0x4

Format:
$--GSA,<Mode>,<CurrentMode>,[<SatPRN1>],...,[<SatPRNN>],
<PDOP>,<HDOP>,<VDOP>, <checksum><cr><lf>

Parameter	Format	Description
Mode	"M" or "A"	Operating Mode: M = Manual, A = Auto (2D/3D)
CurrentMode	Decimal, 1 digit	Current Mode: 1 = no fix available 2 = 2D 3 = 3D
SatPRN1...N	Decimal, 2 digits	Satellites list used in position fix (max N 12)
PDOP	Decimal, 3 digits	Position Dilution of Precision, max: 99.0
HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, max: 99.0
VDOP	Decimal, 3 digits	Vertical Dilution of Precision, max: 99.0
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$GPGSA,A,3,05,21,07,24,30,16,12,,,,,,2.4,1.9,1.5*38

1.7 GSV

GPS Satellites in View. The talker ID for this NMEA message depends on the enabled constellation as follows:
"GP" is used only for GPS satellites. A set of $GPGSV messages is sent to report all GPS satellites.

"GL" is used only for GLONASS satellites. A set of $GLGSV messages is sent to report all GLONASS satellites.

"GN" if enabled in the configuration block (see Application ON/OFF parameter Bit 21) to report all satellites for all enabled constellation. A single set of $GNGSV messages is sent to report all satellites.

NMEA message list bitmask: 0x80000

Format:
$--GSV,<GSVAmount>,<GSVNumber>,<TotSats>,
[<Sat1PRN>,<Sat1Elev>,<Sat1Azim>,<Sat1C/N0>],
... [<SatNPRN>,<SatNElev>,<SatNAzim>,<SatNC/N0>],
<checksum><cr><lf>

Parameter	Format	Description
GSVAmount	Decimal, 1 digit	Total amount of GSV messages, max. 3
GSVNumber	Decimal, 1 digit	Continued GSV number of this message
TotSats	Decimal, 2 digits	Total Number of Satellites in view, max. 12
SatxPRN	Decimal, 2 digits	PRN Number of satellite x
SatxElev	Decimal, 2 digits	Elevation of satellite x in Degree, 0 … 90
SatxAzim	Decimal, 3 digits	Azimuth of satellite x in degree, ref. “North”, 000 … 359
SatxC/N0	Decimal, 2 digits	Carrier to Noise Ratio for satellite x in dB, 00 … 99
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$GPGSV,3,1,12,02,04,037,,05,27,125,44,06,78,051,23,07,83,021,30*7C

$GPGSV,3,2,12,10,16,067,30,12,11,119,36,16,24,301,41,21,44,175,50*73

$GPGSV,3,3,12,23,06,326,28,24,61,118,40,30,45,122,43,31,52,253,37*7C

Note: Due to the fact that up to 12 Satellites may be in view, this message can be repeated up to 3 times containing 4 different Satellites per message. GSVAmount reports the total number of GSV messages to be transmitted, while GSVNumber reports the actual number of the current message frame.

1.8 GPRMC

Recommended Minimum Specific GPS/Transit data

NMEA message list bitmask: 0x40

Format:

$GPRMC,<Timestamp>,<Status>,<Lat>,<N/S>,<Long>,<E/W>,<Speed>,
<Trackgood>,<Date>,<MagVar>,<MagVarDir> <checksum><cr><lf>

Parameter	Format	Description
Timestamp	hhmmss	UTC Time of RMC Sample, example: 160836
Status	“A” or “V”	Receiver warning: “A” = valid, “V” = Warning
Lat	DDMM.MMMM	Latitude in Degree-Minutes.partsMinutes: 4208.5368
N/S	“N” or “S”	Latitude Direction: North or South
Long	DDMM.MMMM	Longitude in Degree-Minutes.partsMinutes: 1105.3456
E/W	“E” or “W”	Longitude Direction: East or West
Speed	Decimal, 4 digits	Speed over ground in “km/h” : max. 999.9
Trackgood	Decimal, 4 digits	Course made good, max. 999.9
Date	Decimal, 6 digits	Date of Fix : ddmmyyyy
MagVar	Decimal, 4 digits	Magnetic Variation, max.: 090.0
MagVarDir	“E” , “W”	Magnetic Variation Direction
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$GPRMC,183417.366,V,4814.040,N,01128.522,E,0.0,0.0,170907,0.0,W*6C

1.9 GPVTG

Recommended Minimum Specific GPS/Transit data

NMEA message list bitmask: 0x10

Format:

$GPVTG,<TMGT>,T,<TMGM>,M,<SoGN>,N,<SoGK>,K*<checksum><cr><lf>

Parameter	Format	Description
TMGT	ddd.d in degrees	Track in reference to “true” earth poles
T	Indicates “terrestrian”	
TMGM	ddd.d in degrees	Track in reference to “magnetic” earth poles
M	Indicates “magnetic”	
SoGN	ddd.d in knots	Speed over Ground in knots
N	Indicates “knots”	

SoGK	ddd.d in km/h	Speed over Ground in kilometers per hour
K	Indicates “kilometers”	
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:

-

2 NMEA COMMANDS

Nmea commands are used to change or query settings of the module.

A Command is a defined Data Packet which is sent from a host processor to the GPS-Baseband Controller in order to control the GPS system behaviour. The regular structure of a command is:
command-ID, <parameters> <cr><lf>
In order to receive the commands the GPS receiver is connected to the PC via the NMEA port (make sure that the serial cable is the right one, sometimes it is necessary to use a cross-cable). The user interaction can be achieved through the use of a PC terminal emulator that is connected to the appropriate COM port with settings of:

115200 Baud

0 Parity Bits

1 Stop Bit

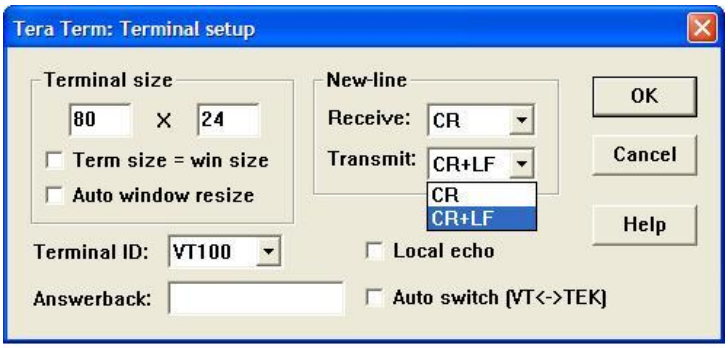
8 Data Bits

The NMEA baud rate at 115200 is the default value, automatically set at the system start-up. It can be modified at system runtime using the appropriate command.
The simplest way to send a command to the device is to write the command string in a text file and send it using the “send file” capability of the terminal emulator.

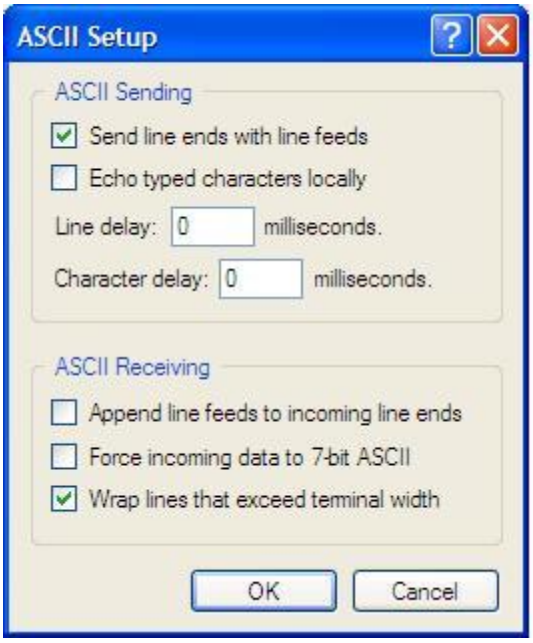
You can use for example TeraTerm as terminal program for giving the commands.

It can be downloaded from e.g. here: http://en.sourceforge.jp/projects/ttssh2/releases/

Remember to set the CR+LF for sending the command from terminal program.



Picture 1. TeraTerm: Terminal setup



Picture 2. HyperTerminal setup

Note: If not explicitly declared, all commands which modify the status of parameters, modifications are not saved in the backup memory. Any new setting will be replaced by default values after system reset or system power cycling.

2.1 ST Proprietary commands List

The table below summarizes all the commands supported by the ST NMEA layer:

$PSTMINITGPS	Initialize GPS position and time
$PSTMINITFRQ	Initialize centre frequency
$PSTMSETRANGE	Set the frequency range for satellite searching
$PSTMCLREPHS	Clear all ephemeris
$PSTMDUMPEPHEMS	Dump Ephemeris data
$PSTMEPHEM	Load Ephemeris data
$PSTMCLRALMS	Clear all almanacs
$PSTMDUMPALMANAC	Dump Almanacs data
$PSTMALMANAC	Load Almanacs data
$PSTMCOLD	Perform COLD start
$PSTMWARM	Perform WARM start
$PSTMHOT	Perform HOT start
$PSTMNMEAONOFF	Toggle ON/OFF the NMEA output
$PSTMDEBUGONOFF	Toggle ON/OFF the DEBUG output
$PSTMGPSRESET	Reset the GPS engine
$PSTMGPSSUSPEND	Suspend GPS engine
$PSTMGPSRESTART	Restart GPS engine

$PSTMTIMEINV	Invalidate the GPS time
$PSTMGETSWVER	Provide the GPS library version string.
$PSTMSBASONOFF	Enable/Disable the SBAS activity
$PSTMSBASSAT	Set the SBAS satellite‟s ID
$PSTM2DFIXONOFF	Enable/Disable the 2D fix algorithm
$PSTMGETRTCTIME	Get the current RTC time.
$PSTMSETPAR	Set System Parameter >>> see section TODO
$PSTMGETPAR	Get System Parameter >>> see section TODO
$PSTMSAVEPAR	Save System Parameters >>> see section TODO
$PSTMRESTOREPAR	Restore System Parameters (Factory Settings) >>> see section TODO

2.2 PSTMINITGPS

Initialize GPS position and time

Synopsis:

$PSTMINITGPS,<Lat>,<LatRef>,<Lon>,<LonRef>,<Alt>,<Day>,<Month>,
<Year>,<Hour>,<Minute>,<Second><cr><lf>

***Arguments:* Parameter**	**Format**	**Description**
Lat	DDMM.MMM	Latitude (Degree-Minute.Minute decimals)
LatRef	„N" or „S"	Latitude direction (North or South)
Lon	DDDMM.MMM	Longitude (Degree-Minute.Minute decimals)
LonRef	„E" or „W"	Longitude Direction (East or West)
Alt	dddd – Decimal,4 digits	Altitude in meters (0000 to 9999)
Day	dd – Decimal, 2 digits	Day of month (01 to 31)
Month	mm – Decimal, 2 digits	Month (01 to 12)
Year	YYYY – Decimal, 4 digits	Year (1994 - …)
Hour	HH – Decimal, 2 digits	Hour (00 to 23)
Minute	MM – Decimal, 2 digits	Minute (00 to 59)
Second	SS – Decimal, 2 digits	Second (00 to 59)

Results:

The position and time will be initialized

No message will be sent as reply.

Example:

$PSTMINITGPS,4811.365,N,01164.123,E,0530,23,02,2009,09,44,12

2.3 PSTMCLREPHS

Clear all ephemeris. This command erases all the ephemeris stored in the NVM backup memory.

Synopsis:
$PSTMCLREPHS<cr><lf>

Arguments:
None.

Results:
All ephemeris, stored in the non-volatile backup memory (either Backup-SRAM or Flash), will be deleted.

No message will be sent as reply.

Example:
$PSTMCLREPHS

2.4 PSTMDUMPEPHEMS

This command sends out all ephemeris stored in the backup memory.

Synopsis:
$PSTMDUMPEPHEMS<cr><lf>

Arguments:
None.

Results:
$PSTMEPHEM,<sat_id>,<N>,<byte1>,.....,<byteN>*<checksum><cr><lf>

Parameter	Format	Description
sat_id	nn – Decimal, 2 digits	Satellite number
N	N - Decimal, 1 Digit	Number of the ephemeris data bytes
byte1	bb - Hexadecimal, 2 digits	First byte of the ephemeris data
byteN	BB - Hexadecimal, 2 digits	Last byte of the ephemeris data
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$PSTMDUMPEPHEMS
$PSTMEPHEM,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400fbff33420000ee632f27698ef001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ffc616fe03*57
$PSTMEPHEM,2,64,0f06bc34bc344f4f4f78110019a5ff00b004fa1d1e0e3f04c8ffcaff1937000033515726556ba9048eae0da1b6c346bd8f985c93ade10c76db001d00f8c7c503*58
$PSTMEPHEM,4,64,0f06bb34bb344b4b4b98050038a4ff000005351e110eea041b00b8ffd037000020b84e26b5138b0425580ca16b211030e68b1a949cac9615f30066ffea92f603*06
$PSTMEPHEM,9,64,0f06bc34bc341818189c0a0069aaff005f06eb249a09ca0477ff6c00f72e00005131d827592b950a91010da1c7af88538e7ca1122fb9be3df4001300c4a0c203*52

2.5 PSTMEPHEM

This command allows the user to load the ephemeris data into backup memory.

Synopsis:

$PSTMEPHEM,<sat_id>,<N>,<byte1>,.....,<byteN>*<checksum><cr><lf>

Arguments: **Parameter**	Format	Description
sat_id	Ii - Decimal, 2 digits	Satellite number
N	N - Decimal, 1 digit	Number of the ephemeris data bytes
byte1	bb - Hexadecimal, 2 digits	First byte of the ephemeris data
byteN	BB - Hexadecimal, 2 digits	Last byte of the ephemeris data
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Results:
The ephemeris will be stored into backup RAM

No message will be sent as reply.

Example:
$PSTMEPHEM,12,64,0f06bc34bc3437373790f40045a7ff00fcf5d522480b4bf71b00fbff8931000096126f271f
869101c3870ca107afce79a763e13e360a1ce8e7003100380ff903*36

2.6 PSTMCLRALMS

This command erases all the almanacs stored in the NVM backup memory.

Synopsis:
$PSTMCLRALMS<cr><lf>

Arguments:
None.

Results:
All almanacs, stored in the non-volatile backup memory, will be deleted.

No message will be sent as reply.

Example:
$PSTMCLRALMS

2.7 PSTMDUMPALMANAC

Dump Almanac data. This command sends out all almanacs stored in the backup memory.

Synopsis:
$PSTMDUMPALMANAC <cr><lf>

Arguments:
None.

Results:
$PSTMDUMPALMANAC,<sat_id>,<N>,<byte1>,,<byteN>*<checksum><cr><lf>

Parameter	Format	Description
sat_id	ii - Decimal, 2 digits	Satellite number
N	N - Decimal, 1 digit	Number of the almanac data bytes
byte1	bb - Hexadecimal, 2 digits	First byte of the almanac data
byteN	BB- Hexadecimal, 2 digits	Last byte of the almanac data
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$PSTMDUMPALMANAC
$PSTMALMANAC,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034024200b4ffff00*1a
$PSTMALMANAC,2,32,021a0690944b78fe37fd0800770da141ef0c5b0060487700989bd800d8088000*1a
$PSTMALMANAC,3,32,031a06904f68a2f540fd0800f60ca141922a2c003cae27009496cf00020a8000*15

$PSTMALMANAC,4,32,041a0690a94aeffd36fd0800390ca141afc95b00de7a1700dfc74e004ddebf00*13
$PSTMALMANAC,5,32,051a0690940eee0b5efd0800900ca141582b8600d3000b0060641200e40f8000*14

2.8 PSTMALMANAC

Load Almanacs data. This command allows the user to load the almanacs data into backup memory.

Synopsis:
$PSTMALMANAC,<sat_id>,<N>,<byte1>,.....,<byteN>*<checksum><cr><lf>

Parameter	Format	Description
sat_id	ii - Decimal, 2 digits	Satellite number
N	N - Decimal, 1 digit	Number of the almanac data bytes
byte1	bb - Hexadecimal, 2 digits	First byte of the almanac data
byteN	BB - Hexadecimal, 2 digits	Last byte of the almanac data
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

2.9 PSTMCOLD

Perform a COLD start.

Synopsis:
$PSTMCOLD, < Mask ><cr><lf>

Parameter	Format	Description
Mask	Integer	Optional parameter to invalidate time, position, ephemeris and almanac : 0x1 – clear almanac 0x2 – clear ephemeris 0x4 – clear position 0x8 – clear time

Results:

Coldstart initialization and system restart3. If Mask parameter is used, only the selected GPS data is invalidated for this actual Coldstart. Multiple selects are supported (i.e. 0xD). If Mask parameter is not used, default is 0xE (clear ephemeris, time and position).

Example:

$PSTMCOLD,6

2.10 PSTMWARM

Perform a WARM start.

Synopsis: $PSTMWARM<cr><lf>

Arguments:

None.

Results:

Warm start initialization and system restart.

Example:

$PSTMWARM

2.11 PSTMHOT

Perform an HOT start.

Synopsis:
$PSTMHOT<cr><lf>

Arguments:
None.

Results:
The system restart5.

Example:

$PSTMHOT

2.12 PSTMNMEAONOFF

Toggle NMEA output. This command switches ON or OFF the output NMEA sentences.

Synopsis:

$PSTMNMEAONOFF,<on_off><cr><lf>

Arguments:

Parameter Format Description

on_off6

Integer

0 = NMEA output is turned OFF

1 = NMEA output is turned ON

Results:

If the NMEA output message is running, sending "$PSTMNMEAONOFF,0" the NMEA output is stopped. If the NMEA output message is OFF, sending "$PSTMNMEAONOFF,1" the NMEA output is started. Sending "$PSTMNMEAONOFF,1" while NMEA is running or sending "$PSTMNMEAONOFF,0" while NMEA is stopped the command is rejected with no effects.

Example:

$PSTMNMEAONOFF,0

2.13 PSTMDEBUGONOFF

Toggle DEBUG output. This command switches ON or OFF the output DEBUG sentences.

Synopsis:

$PSTMDEBUGONOFF,<on_off><cr><lf>

***Arguments:* Parameter**	**Format**	**Description**
on_off	Integer	0 = DEBUG output is turned OFF 1 = DEBUG output is turned ON

2.14 PSTMGPSRESET

Reset the GPS receiver engine.

Synopsis: $PSTMGPSRESET<cr><lf>

Arguments:

None.

Results: The GPS receiver engine will be reset No message will be sent as reply.

Note: using this command the GPS module won't reboot.

Example:

$PSTMGPSRESET

2.15 PSTMGPSSUSPEND

Suspend the GPS receiver engine.

Synopsis:
$PSTMGPSSUSPEND<cr><lf>

Arguments:
None.

Results:
The GPS receiver engine will be suspended

No message will be sent as reply.

Example:

$PSTMGPSSUSPEND

2.16 PSTMGPSRESTART

Restart the GPS receiver engine.

Synopsis:

PSTMGPSRESTART<cr><lf>

Arguments:

None.

Results:

The GPS receiver engine will be restarted No message will be sent as reply.

Example:

$PSTMGPSRESTART

2.17 PSTMTIMEINV

Invalidate the Real Time Clock (RTC).

Synopsis:

$PSTMTIMEINV<cr><lf>
Arguments:

None.
Results:

The RTC time will be invalidated.

Example:

$PSTMTIMEINV

2.18 PSTMGETSWVER

Get the version string of the GNSS library embedded in the software application.

Synopsis:

$PSTMGETSWVER <cr><lf>

Arguments:

None.

Results:

$PSTMVER,GNSSLIB_<Ver>,<Type>,<Date>,<Time> <cr><lf>

Where:

Parameter	Format	Description
GNSSLIB	Text, fixed	Text String
Ver	x.x.x.x	Library Version: example 7.1.1.15
Type	ARM, GNU	Compiler Type:ARM or GNU
Date	mm dd yyyy	Compile Date: example Sept 04 2008
Time	« hh :mm :ss	Compile Time: example 13:15:03

Example:

$PSTMGETSWVER

2.19 PSTMSBASONOFF

Suspend / resume the SBAS software execution.

Synopsis:

$PSTMSBASONOFF<cr><lf

Arguments:

None.

Results: If SBAS was running it will be suspended, if it was suspended it will start to run.

Example:

$PSTMSBASONOFF

2.20 PSTMSBASSAT

Change the SBAS satellite.

Synopsis:
$PSTMSBASSAT,<prn><cr><lf

Arguments: **Parameter**	**Format**	**Description**
prn	Decimal, 3 digit	Satellite PRN (Range: from 120 to 138 and 0)

2.21 PSTMRFTESTON

Enable the RF test mode for production line tests.

Synopsis:

$PSTMRFTESTON,<sat_id>,<cr><lf>

Arguments:

Parameter Format Description

sat_id

Decimal, 2 digits

Satellite number

Results: The GPS engine will restart in the RF test modality. This RF test forces the GPS acquiring process only on the provided satellite.s id. It could be useful to reduce the RF testing time in the production line where generally a single channel simulator is present

Example:

$PSTMRFTESTON,24

2.22 PSTMRFTESTOFF

Disable the RF test mode for production line tests.

Synopsis:

$PSTMRFTESTOFF <cr><lf>

Arguments:

None.

Results: The RF test modality will be disabled and the GPS engine will be restarted.

Note: the RF test mode can be disabled also resetting the GPS module.

Example:

$PSTMRFTESTOFF

2.23 PSTM2DFIXONOFF

Enable/Disable the GPS 2D fix algorithm.

Synopsis:

$PSTM2DFIXONOFF,<on_off><cr><lf>

Arguments:

Parameter Format Description

on_off

Decimal, 1 digit

0 = the 2D fix algorithm will be disabled.

1 = the 2D fix algorithm will be enabled.

Results:

If the input parameter is 0 the 2D fix algorithm will be disabled. The following message is send: $PSTM2DFIXDISABLED If the input parameter is 1 the 2D fix algorithm will be enabled. The following message will be send: $PSTM2DFIXENABLED In case of an error the system will reply: $PSTM2DFIXONOFFERROR.

Note:

The changes, made by the above command, will take effect only after a GPS engine reset. It is recommended to send the $PSTMGPSRESET command after the $PSTM2DFIXONOFF command

Example:

$PSTM2DFIXONOFF,1

2.24 PSTMGETRTCTIME

Get the current RTC time.

Synopsis:

$PSTMGETRTCTIME<cr><lf>

Arguments:

None.

Results:

System will send RTC Data and Status.

$PSTMGETRTCTIME,<time>,<date>,<rtc_status>,<time_validity>*<checksum><cr><lf>

Where:

Parameter	Format	Description
time	hhmmss.ms	Current time read on RTC.
date	ddmmyy	Current date read on RTC.
rtc_status	Decimal, 1 digit	Status: 0 - RTC_STATUS_INVALID 1 - RTC_STATUS_STORED 2 - RTC_STATUS_APPROXIMATE
time_validity	Decimal, 1 digit	Validity: 0 - NO_TIME 1 - FLASH_TIME 2 - USER_TIME 3 - USER_RTC_TIME 4 - RTC_TIME 5 - RTC_TIME_ACCURATE 6 - APPROX_TIME 7 - POSITION_TIME 8 - EPHEMERIS_TIME
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf>

		characters.

Example:

$PSTMGETRTCTIME

3 SYSTEM CONFIGURATION COMMANDS

The GNSS Software utilizes a "Configuration Data Block" that holds the working parameters for the system. The parameters can be set, read or store (in NVM) using the system configuration commands: $PSTMSETPAR, $PSTMGETPAR and $PSTMSAVEPAR. There is also a command to restore the factory setting parameters: $PSTMRESTOREPAR.

At run-time it could be possible to have up to three different configuration blocks: Current configuration: it is placed in RAM memory and it includes the current configuration of each parameter. This configuration block can be modified with the $PSTMSETPAR command. The $PSTMSAVEPAR command stores the current configuration data block into the NVM memory. At startup the current configuration block is loaded from NVM (if a stored data block is available) or it is loaded from default one embedded in the code (factory settings). Default configuration: it is generally placed in the flash/rom memory. It includes the factory setting for each parameter. This configuration is used at system startup if there is no configuration data into the NVM memory. NVM stored configuration: it is available in the NVM backup memory as soon as the $PSTMSAVEPAR command is executed. It includes all parameters modified and stored by the user. At system startup the SW configuration managements checks if a valid configuration block is available in the NVM backup memory. In case the stored configuration is available, it will be used for system configuration. If not available the default setting will be used.

Note: Other "Configuration Data Block" parameters not documented in this manual must be considered as reserved and must not be modified. Modifying any other parameter intentionally or unintentionally may stop the system from working and/or degrade the system performance.

3.1 PSTMSETPAR

This command sets the defined parameter (indicated by "ID") to the value provided as "param_value" in the commands parameter**.**

Synopsis: $PSTMSETPAR,<ConfigBlock><ID>,<param_value>[,<mode>]*<cr><lf>

Arguments:

Parameter	Format	Description
ConfigBlock	Decimal,1 digit	Indicates one of configuration blocks: 1=Current Configuration, 2 = Default Configuration, 3 = NVM Stored configuration.
ID	Decimal, 3 digits	ID - Identifier (see, Configuration Data Block)
param_value	1 up to 80 bytes	Parameter to be set, see "Allowed values"
mode	Decimal, 1 digit	This parameter is optional. It allows to perform bit-to-bit "OR" or "AND" operations between the selected parameter in the configuration block and the param_value in input. It has the following meaning: 0: the parameter in the

		configuration block is overwritten by the param_value. This is the default action as in the case mode is omitted. 1: the parameter in the configuration block is the result of bit-to-bit “OR” between old value and the param_value.This is useful for bit mask setting. 2: the parameter in the configuration block is the result of bit-to-bit “AND” between old value and NOT(param_value). This is useful for bit mask resettimg.

Results:

The parameter indicated by the ID value is set according to the parameters included in param_value. In case of no errors, the following message is returned $PSTMSETPAROK ,<ConfigBlock><ID>*<checksum><cr><lf> In case of errors, the error message is returned

$PSTMSETPARERROR*<checksum><cr><lf>

Parameter	Format	Description
ConfigBlock	Decima1,1 digit	Indicates one of configuration blocks: 1=Current Configuration, 2 = Default Configuration, 3 = NVM Stored configuration.
ID	Decimal, 3 digits	ID - Identifier (see, Configuration Data Block)
checksum	Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

Issuing the command: $PSTMSETPAR,1121,10*

You could have this answer: $PSTMSETPAROK,1121*

Note: The configuration block parameter is ignored by the “SET” command because only the current configuration, stored in the RAM memory, can be written. It is used only to keep same syntax as for the “GET” command. The configuration block stored in NVM will be overwritten by current configuration after the $PSTMSAVEPAR command.

Note: There is no comma and no space between ConfigBlock and ID parameters.

Note: The input param_value must be expressed in hexadecimal format without “0x” prefix for any integer value except DOP configuration. It must be decimal for any not integer value and DOP setting.

3.1.1 Set NMEA speed

Example of setting lower NMEA speed

Syntax for changing the baud rate:

*$PSTMSETPAR,1102,n*cs<CR><LF>*

where n can be set to configure speed as follows:

0 = 300 baud
1 = 600 baud
2 = 1200 baud
3 = 2400 baud
4 = 4800 baud
5 = 9600 baud
6 = 14400 baud
7 = 19200 baud
8 = 38400 baud
9 = 57600 baud
A = 115200 baud
B = 230400 baud
C = 460800 baud
D = 921600 baud

Command to set NMEA speed to 9600:

$PSTMSETPAR,1102,5*02<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

Example of setting default NMEA speed

Command to set NMEA speed to 115200:

$PSTMSETPAR,1102,A*58<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

3.1.2 Change NMEA mask

Example of setting lower NMEA speed

Syntax for changing the baud rate:

*$PSTMSETPAR,1201,nnnnn*cs<CR><LF>*

where nnnnn is the bitmask for the messages.

Bitmask can be configured as follows:

Bit16	Bitmask	Function
0	0x1	$GPGGA Message
1	0x2	$GPGGA5 Message
2	0x4	$GPGSA Message
3	0x8	Not used anymore
4	0x10	$GPVTG Message
5	0x20	Not used anymore
6	0x40	$GPRMC Message
7	0x80	$PSTMRF Message
8	0x100	$PSTMTG Message
9	0x200	$PSTMTS Message
10	0x400	$PSTMPA Message
11	0x800	$PSTMSAT Message
12	0x1000	$PSTMRES Message
13	0x2000	$PSTMTIM Message
14	0x4000	$PSTMWAAS Message
15	0x8000	$PSTMDIFF Message
16	0x10000	$PSTMCORR Message
17	0x20000	$PSTMSBAS Message
18	0x40000	$PSTMTESTRF Message
19	0x80000	$GPGSV Message
20	0x100000	$GPGLL Message

21	0x200000	Not used
22	0x400000	Not used
23	0x800000	Not used
24	0x1000000	Not used
25	0x2000000	Not used
26	0x4000000	Not used
27	0x8000000	$PSTMKFCOV Message
28	0x10000000	$PSTMAGPS Message
29	0x20000000	Not used
30	0x40000000	Not used
31	0x80000000	Not used

Command to enable only RMC and GSV messages:

$PSTMSETPAR,1201,8040*15<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

Command to enable only GGA5 and VTG messages:

$PSTMSETPAR,1201,0012*1A<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

3.1.3 Enable QZSS

Example of enabling QZSS satellite search. This will also enable GNGSV output message.

enable QZSS:

$PSTMSETPAR,1200,01EF9604*11<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

3.1.4 Select constellation

Example of setting used constellation.

Enable Glonass and GPS (hybrid navigation, default setting)

$PSTMSETPAR,1200,01639604*17<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Enable Glonass only

$PSTMSETPAR,1200,01229604*12<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Enable Glonass only

$PSTMSETPAR,1200,01419604*17<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

3.1.5 Enable Notch Filter

Example of enabling notch filter.

This parameter supported starting from 2.1.6.25 SW revision.

Syntax for configuring the Notch Filter:

*$PSTMSETPAR,1125,n*cs<CR><LF>*

where n can be set to configure Notch Filter as follows:

Bitmask	Description
0x0	Notch Filter is disabled on both GPS and GLONASS paths
0x1	Notch Filter is enabled only on GPS path
0x2	Notch Filter is enabled only on GLONASS path
0x3	Notch Filter is enabled on both GPS and GLONASS paths

Command to set Notch Filter active on GPS and Glonass paths:

$PSTMSETPAR,1125,3*2F<CR><LF>

Save parameter to flash:

$PSTMSAVEPAR*58<CR><LF>

Note: Remember to issue reset after saving parameter to flash.

3.2 PSTMGETPAR

This command reads the defined parameter (indicated by "ID") from the "Configuration Data Block" and returns it as a specific message.

Synopsis: $PSTMGETPAR,<ConfigBlock><ID>*<cr><lf>

Arguments:

Parameter	Format	Description
ConfigBlock	Decima1,1 digit	Indicates one of configuration blocks: 1=Current Configuration, 2 = Default Configuration, 3 = NVM Stored configuration.
ID	Decimal, 3 digits	ID - Identifier (see, Configuration Data Block)

Results: In case of no errors, the selected parameter ID value is returned in the following message $PSTMSETPAR,<ConfigBlock><ID>,<value>*<checksum><cr><lf> In case of errors, the error message is returned $PSTMGETPARERROR*<checksum><cr><lf>

Where:

Parameter	Format	Description
ConfigBlock	Decima1,1 digit	Indicates one of configuration blocks: 1=Current Configuration, 2 = Default Configuration, 3 = NVM Stored configuration.
ID	Decimal, 3 digits	ID - Identifier (see, Configuration Data Block)
value	Hexadecimal or Decimal	The value of returned parameter. According to the parameter type it could be expressed in hexadecimal format (in case parameter is integer) or decimal format (in case the parameter is floating).

checksum	Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

Issuing the command: $PSTMGETPAR,1403*

You could have this answer: $PSTMSET,1403,15,12,12,18*<checksum><cr><lf>

Note: there is no comma and no space between ConfigBlock and ID parameters.

Note: In case of no errors the answer is deliberately $PSTMSET and not $PSTMGET.

Note: if the parameter ID is "000" all the configuration block is printed out using one message for each parameter. The message syntax is the same as reported above.

3.3 PSTMSAVEPAR

Save current configuration data block into the backup memory.

Synopsis:

$PSTMSAVEPAR<cr><lf>

Arguments:

None.

Results:

The current configuration data block, including changed parameters, will be stored into the backup memory (NVM).

Note: the factory setting parameters can be restored using the $PSTMRESTOREPAR command.

Example:

$PSTMSAVEPAR

3.4 PSTMRESTOREPAR

Restore the factory setting parameters. The configuration data block stored in NVM, if present, will be invalidated. Any changed parameter will be lost.

Synopsis:
$PSTMRESTOREPAR<cr><lf>

Arguments:
None.

Results:
The factory setting parameters will be restored and the configuration block in the backup memory will be lost. A system reboot is needed to complete the factory reset restoring ad to get system working with default setting.

Example:

$PSTMRESTOREPAR

4 MESSAGES

This section contains both the standard NMEA messages and the proprietary messages delivered from any ST-GPS system. Additionally it contains messages which result from a specific command input.

7.1 Standard NMEA messages list

Syntax	Default	Description
$GPGGA	OFF	NMEA: Global Position System Fix Data
$GPGGA5	ON	NMEA: Global Position System Fix Data (as before) with 5 digits instead of 3 in the latitude and longitude fractional parts.
$GPGLL	OFF	NMEA: Geographic Position Latitude/Longitude
$--GSA	ON	NMEA: GPS DOP and Active Satellites. “GP”, “GL” and “GN” talker ID are supported according to the software configuration.
$--GSV	ON	NMEA: GPS Satellites in View. “GP”, “GL” and “GN” talker ID are supported according to the software configuration.
$GPRMC	ON	NMEA: Recom. Min. Spec. GPS/TRANSIT Data
$GPVTG	OFF	NMEA: Track made good and ground speed
$GPZDA	OFF	NMEA: Time and Date

4.1 ST NMEA messages list

Syntax	Default	Description
$PSTMDIFF	OFF	ST: Differential Correction Data
$PSTMPRES	OFF	ST: Position Residuals
$PSTMVRES	OFF	ST: Velocity Residuals
$PSTMPA	OFF	ST: Position Algorithm
$PSTMRF	OFF	ST: Radio Frequency
$PSTMSAT	OFF	ST: Satellite Information
$PSTMSBAS	ON	ST: Augmentation System
$PSTMSBASCORR	OFF	ST: Satellite Correction Data
PSTMTIM	OFF	ST: System Time
$PSTMTG	OFF	ST: Time and Number of used Satellites

$PSTMTS	OFF	ST: Tracked Satellite Data

4.2 Commands answers messages list

Syntax	Default	Description
$PSTMALMANAC	Reply	ST: Dump Almanac <Data>
$PSTMEPH	Reply	ST: Dump Ephemeris <Data>
$PSTMGETRTCTIME	Reply	ST: Get Real Time Clock Time
$PSTMSETRANGEERROR	Reply	ST: Error Message: Range set = failed
$PSTMSETRANGEOK	Reply	ST: Acknowledge Range set = OK
$PSTMVER	Reply	ST: Output Version String

5 ST NMEA MESSAGES SPECIFICATION

In order to provide further data and information from the GPS system, which are not provided by the standard NMEA messages, STMicroelectronics provides "proprietary messages". Any proprietary message on the NMEA port starts with "$Pxxxx…" and the following three letter indicate that it is a ST proprietary message ($PSTMxxx…)

There are two sorts of "proprietary messages" within a ST-GPS system. They are either send repeatedly with a defined or defineable reporting rate or they are send only once as a reaction to a command.

5.1 PSTMRF

Provides "satellite signal data" for each tracked satellite. Single message contains the relevant fields for 3 satellites. For all satellites the message is repeated with the data of the other satellites.

NMEA message list bitmask: 0x80

Format:

$PSTMRF,<MessgAmount>,<MessgIndex>,<used_sats>, [<Sat1ID>,<Sat1PhN>,<Sat1Freq>,<Sat1CN0>], ... [<SatNID>,<SatNPhN>,<SatNFreq>,<SatNCN0>], <checksum><cr><lf>

Parameter	Format	Description
MessgAmount	Decimal, 1 digit	Number of consecutive $PSTMRF messages
MessgIndex	Decimal, 1 digit	Current number in the sequence of messages
used_sats	Decimal, 2 digits	Number of satellites used in the fix
SatxID	Decimal, 2 digits	Satellite x Number (PRN)
SatxPhN	Decimal, 5 digits	Satellite x Phase Noise
SatxFreq	Decimal, 6 digits	Satellite x Frequency
SatxCN0	Decimal, 2 digits	Satellite x Carrier to Noise Ratio (in dB)
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example: $GPRMC,183417.366,V,4814.040,N,01128.522,E,0.0,0.0,170907,0.0,W*6C

5.2 PSTMTESTRF

Specific message containing information on just one satellite for RF testing purposes.
NMEA message list bitmask: 0x40000

Format:

$PSTMTESTRF,<Sat-ID>,<Sat-Freq>,<Sat-PhN><Sat-CN0>,<checksum><cr><lf>

Parameter	Format	Description
Sat-ID	Decimal, 2 digits	Satellite Number (PRN)
Sat-Freq	Decimal, 5 digits	Satellite Frequency
Sat-PhN	Decimal, 5 digits	Satellite Phase Noise
Sat-CN0	Decimal, 2 digits	Satellite Carrier to Noise Ratio (in dB)

checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

5.3 PSTMTG

Time and Satellites Information
NMEA message list bitmask: 0x100

Format:
$PSTMTG,<Week>,<TOW>,<Tot-Sat>,<CPU-Time><Timevalid><cr><lf>

Parameter	Format	Description
Week	Decimal, 4 digits	Week Number
TOW	Decimal, 10 digits	Time of Week
Tot-Sat	Decimal, 2 digits	Total Number of satellites used for fix
CPU-Time	Decimal, 10 digits	CPU Time
Timevalid	Decimal, 2 digits	0 = no time 1 = time read from flash 2 = time set by user 3 = time set user RTC 4 = RTC time 5 = RTC time, accurate 6 = time approximate 7 = “not used” 8 = time accurate 9 = position time 10 = Ephemeris time

5.4 PSTMTS

This message is repeated for each satellite tracked and used for the calculation of a fix

NMEA message list bitmask: 0x200

Format:

$PSTMTS,<dsp-dat>,<SatID>,<PsR>,<Freq>,<plf>,<CN0>,<ttim>,<Satdat>,
<Satx>,<Saty>,<Satz>,<Velx>,<Vely>,<Velz>,<src>,<ac>,
<difdat>,<drc>,<drrc><predavl>,<predage>,<predeph>,<predtd>, <cr><lf>

Parameter	Format	Description
dsp-dat	Decimal, 1 digit	DSP data available: 0 = satellite not tracked 1 = satellite tracked
Sat-ID	Decimal, 2 digits	Satellite Number (PRN)
PsR	Decimal, 10 digits	Pseudo range
Freq	Decimal, 8 digits	Satellite tracking Frequency (Offset ???)
Plf	Decimal, 1 digit	Preamble Lock Flag

		0 = Navigation data stream preamble not locked 1 = Navigation data stream preamble locked
CN0	Decimal, 3 digits	Satellite Carrier to Noise Ratio (in dB)
Ttim	Decimal, 6 digits	Track Time of Satellite (in seconds)
Satdat	Decimal, 1 digit	Satellite Data available Flag 0 = Sat. Ephemeris not available or unhealthy Sat. 1 = Sat. Ephemeris available and healthy Satellite
Satx	Decimal, 10 digits	Satellite Position , X-Coordinate
Saty	Decimal, 10 digits	Satellite Position , Y-Coordinate
Satz	Decimal, 10 digits	Satellite Position , Z-Coordinate
Velx	Decimal, 8 digits	Satellite Velocity , X-Coordinate
Vely	Decimal, 8 digits	Satellite Velocity , Y-Coordinate
Velz	Decimal, 8 digits	Satellite Velocity , Z-Coordinate
Src	Decimal, 6 Digits	Satellite Range Correction
Ac	Decimal, 3 Digits	Atmospheric Correction
Difdat	Decimal, 1 digit	Differential Data available Flag 0 = Differential Corrections not available 1 = Differential Corrections available
Drc	Decimal, 3 digits	Differential Range Correction (from DGPS Station)
Drrc	Decimal, 3 digits	Differential Range Rate Correction (from DGPS Stat.)
predavl	Decimal, 1 digit	Prediction available Flag 0 = Predicted Ephemeris not available 1 = Predicted Ephemeris available
predage	Decimal, 1 digit	Age of predicted Ephemeris (in hours)
predeph	Decimal, 1 digit	Number of satellites ued for prediction (1 or 2)
predtd	Decimal, 1 digit	Time distance of Ephemeris calculated from 2 Sats. Only valid if <pred-eph> = 2
Src	Decimal, 6 Digits	Satellite Range Correction

Note: <pred-xxx> fields are only included within the message if the AGPS software module has been included.

Example:

$PSTMTS,1,05,15748178.41,30992.22,1,44,306150,1,16278399.26,20504574.30,4653136.69,38.03,703.04,-3046.01,141169.29,11.45,1,-12.75,0.00, $PSTMTS,1,31,14242886.83,-28462.15,1,37,304775,1,20641723.13, -8713847.54,14517949.66,1788.86,311.39,-2382.23,1804.01,7.09,1, -5.74,0.00, $PSTMTS,1,21,14885540.17,-25018.74,1,50,301653,1,25482227.75, 6629457.30,5528104.33,-699.61,220.74,2983.68,23248.85,8.12,1, -2.84,0.00, $PSTMTS,1,07,13337296.04,-27966.11,1,31,296621,1,15777659.46, 4155044.35,21301094.71,-1287.52,2301.27,509.20,-15394.31,5.65,1, -3.83,0.00, $PSTMTS,1,06,1216319.39,-28367.75,0,23,40492,1,14595868.85, 6511991.60,21397698.91,-1394.03,2294.91,251.81,70766.81,5.72,1, -3.28,0.00, $PSTMTS,1,24,13629659.89,-27176.62,1,40,298187,1,17698708.17, 12886703.95,15024752.78,-1901.12,-1.00,2298.33,11530.25,6.39,1, -9.27,0.00, $PSTMTS,1,30,14421546.48,-30401.97,1,44,298264,1,17539544.73, 16864817.03,10440026.12,394.97,1346.12,-2741.16,14708.79,7.87,1, -9.96,0.00, $PSTMTS,1,16,16177492.44,-24593.30,1,40,298572,1,6202032.13, -17659074.51,18852818.90,1139.40,2098.88,1613.11,35896.88,12.03,1,-4.54,0.00, $PSTMTS,1,10,16728325.63,-26663.46,1,30,124750,1,-2057875.88, 21248945.17,15476302.66,-1018.51,-1731.48,2256.47, -32564.02,15.33,1,-12.86,0.00, $PSTMTS,1,12,17539958.05,-31018.23,1,35,10528,1,11788804.59, 23841922.01,245355.77,-236.27,137.48,-3173.58,-103404.01,20.66,1, -19.21,0.00, $PSTMTS,1,23,17770191.78,-27801.14,1,28,196026,1,-6131001.55, -15740405.01,20363733.86,1549.10,-2097.11,-1173.09,89981.45, 27.98,0,0.00,0.00,

5.5 PSTMPA

Position Algorithm

NMEA message list bitmask: 0x400

Format:

$PSTMPA,<PosA>,<Dur><cr><lf>

Parameter	Format	Description
PosA	ASCII, 2	Position Algorithm Indicator Empty = none LS = LMS KF = Kalman Filter
Dur	Decimal, 3 digits	Time period in which the position has been stationary (count in seconds)

Example:

$PSTMPA,KF,433 $PSTMPA, ,00

5.6 PSTMSAT

This message is repeated for each satellite tracked and used for the calculation of a fix. The information contained in this message is a subset of the $PSTMTS message.
NMEA message list bitmask: 0x800

Format:

$PSTMSAT,<SatID>,<PsR>,<Freq>,<Satx>,<Saty>,<Satz><cr><lf>

Parameter	Format	Description
SatID	Decimal, 2 digits	Satellite Number (PRN)
PsR	Decimal, 10 digits	Pseudo Range
Freq	Decimal, 8 digits	Tracking Frequency of Satellite
Satx	Decimal, 10 digits	Satellite Position, X-Coordinate
Saty	Decimal, 10 digits	Satellite Position, Y-Coordinate
Satz	Decimal, 10 digits	Satellite Position, Z-Coordinate

Example:

-

5.7 PSTMPRES

Position Residual
NMEA message list bitmask: 0x1000 ($PSTMPRES and $PSTMVRES are always enabled together)

Format:

$PSTMPRES,<RMSpos>,<res1>,...,<resN>*<checksum><cr><lf>

N = number of tracked satellites

Parameter	Format	Description
RMSpos	dd.d Decimal, 3 digits	position “rms” residual for the fix
resx	dd.d Decimal, 3 digits	Residual of tracked satellite x (Corresponds to x satellite in $GPGSA Message)
Fixed Character	Delimiter of datafield	
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:

$PSTMPRES,8.1,-0.2,-0.2,-0.1,-0.3,-0.3,-0.4,,,,,,*2D $PSTMPRES,0.0,,,,,,,,,,,,*20

5.8 PSTMVRES

Position Residual

NMEA message list bitmask: 0x1000 ($PSTMPRES and $PSTMVRES are always enabled together)

Format:

$PSTMPRES,<RMSvel>,<vres1>,...,<vresN>*<checksum><cr><lf>

N = number of tracked satellites

Parameter	Format	Description
RMSvel	dd.d Decimal, 3 digits	velocity “rms” residual for the fix
vresx	dd.d Decimal, 3 digits	Residual of tracked satellite x (Corresponds to x satellite in $GPGSA Message)
Fixed Character	Delimiter of datafield	
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:
$PSTMVRES,0.0,0.0,0.0,0.0,,,,,,,,,*26

PSTMKFCOV

This message contains the Standard Deviations for position and velocity and their split into north, east and vertical components.

NMEA message list bitmask: 0x8000000

Format:

$PSTMKFCOV,<PosStd>,<PosNcov>,<PosEcov>,<PosVcov>, <VelStd>,<VelNcov><VelEcov>,<VelVcov> <cr><lf>

Parameter	Format	Description
PosStd	ddd.d Decimal, 2 digit	Standard Deviation of Position in meters
PosNcov	ddd.d Decimal, 4 digit	Covariance (North/South) in m^2 (from Kalman Filter)
PosEcov	ddd.d Decimal, 4 digit	Covariance (East/West) in m^2 (from Kalman Filter)
PosVcov	ddd.d Decimal, 4 digit	Covariance (Vertical) in m^2 (from Kalman Filter)
VelStd	ddd.d Decimal, 2 digit	Standard Deviation of Velocity in meter/second
VelNcov	ddd.d Decimal, 4 digit	Covariance (North/South) in m^2/s (from Kalman Filter)
VelEcov	ddd.d Decimal, 4 digit	Covariance (East/West) in m^2/s (from Kalman Filter)
VelVcov	ddd.d Decimal, 4 digit	Covariance (Vertical) in m^2/s (from Kalman Filter)

Example:

$PSTMKFCOV,8.7,50.9,25.4,150.7,0.4,0.1,0.0,0.2*49

5.9 PSTMAGPS

This message has the same syntax of standard NMEA GSA message. It provides dynamically standard GSA data or STAGPS related information according to the status of predicted ephemeris for each satellite. To send out different types of information for each satellite, an integer number is sent in the message fields instead of the satellite PRN ID; it should be decoded to get all the message info. If a satellite is not using a predicted ephemeris its PRN id is reported as in the standard GSA message case (the integer number will be identical to the satellite PRN ID – see formula below when AGE is 0). If a satellite is using a predicted ephemeris a number which is related to sat PRN and predicted ephemeris age is reported instead of simple PRN id. It is generated using the formula: satID + 32 * STAGPS_AGE_DAYS where STAGPS_AGE_DAYS is the number of days from current time back to the most recent ephemeris used for STAGPS predictions. STAGPS_AGE_DAYS = 1: most recent ephemeris has been downloaded from 0 up to 24 hours in the past. STAGPS_AGE_DAYS = 2: most recent ephemeris has been downloaded from 24 up to 48 hours in the past. STAGPS_AGE_DAYS = 3: most recent ephemeris has been downloaded from 48 up to 72 hours in the past.

This message could be used to replace the standard GSA in all devices where STAGPS is enabled. It allows, decoding a single sentence, to show on the screen satellite bars coloured with different colours according to each ephemeris prediction age. Of course, if STAGPS is not enabled, it will behave in the same way of NMEA GSA sentence.

NMEA message list bitmask: 0x10000000

Format: $PSTMAGPS,<Mode>,<CurrentMode>,[<SatPRN1>],...,[<SatPRNN>], <PDOP>,<HDOP>,<VDOP>, <checksum><cr><lf>

Parameter	Format	Description
Mode	“M” or “A”	Operating Mode: M = Manual, A = Auto (2D/3D)
CurrentMode	Decimal, 1 digit	Current Mode: 1 = no fix available 2 = 2D 3 = 3D
SatPRN1...N	Decimal, 2 digits	Satellites list used in position fix (max N 12)
PDOP	Decimal, 3 digits	Position Dilution of Precision, max: 99.0
HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, max: 99.0
VDOP	Decimal, 3 digits	Vertical Dilution of Precision, max: 99.0
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

$PSTMAGPS,A,3,05,85,103,24,30,48,12,,,,,2.4,1.9,1.5*38

The example above should be read in the following way: Satellites 5, 24, 30, 12 don.t have predicted ephemeris (they are reported as in the case of standard GSA message . basically all satellites reported with

a number less or equal 32 have no predicted ephemeris). Satellite 21 has a predicted ephemeris 2 days old. Satellite 7 has predicted ephemeris 3 days old. Satellite 16 has predicted ephemeris 1 day old.

Here are two simple decoding functions to get satellite ID and ages: Age = (int)((<reported number> - 1) / 32) Satid = <reported number> - 32 * Age

5.10 PSTMTIM

Time Validity
NMEA message list bitmask: 0x2000

Format:
$PSTMTIM,<Tvalid><cr><lf>

Parameter	**Format**	**Description**
Tvalid	ASCII	"RTC" = time read from RTC "VALID" = time downloaded from satellite or corrected using position "INVALID" = time is not valid

Example:

-

5.11 PSTMDIFF

Time Validity
NMEA message list bitmask: 0x8000

Format:
$PSTMDIFF,<ListSize>,<NCS>, [<Sat1ID>,<Corr1Avl>,][<SatNID>,<CorrNAvl>,]
*<checksum><cr><lf>

N = number of tracked satellites **Parameter**	**Format**	**Description**
ListSize	Decimal, 2 digits	Amount of visible satellites in this message (n)
NCS	Decimal, 2 digits	Number of corrected satellites
SatxID	Decimal, 2 digits	Satellite x ID (PRN)
CorrxAvl	Decimal	Correction available for Satellite x
checksum	Hexadecimal,2 digits	Checksum of the message bytes without <checksum><cr><lf> characters

Example:

-

5.12 PSTMSBASCORR

SBAS differential corrections
NMEA message list bitmask: 0x10000

Format:
$PSTMCORR,<MessgAmount>,<MessgIndex>,<visiblesats>,
[<Sat1ID>, <Corr1Total>,<Corr1Sat>,<Corr1Iono>,<Corr1Tropo>,]
...
[<SatNID>, <CorrNTotal>,<CorrNSat>,<CorrNIono>,<CorrNTropo>,]
*<checksum><cr><lf>

N = number of tracked satellites

Parameter	Format	Description
MessgAmount	Decimal, 1 digit	Number of consecutive $PSTMSBASCORR messages
MessgIndex	Decimal, 1 digit	Current number in the sequence of messages
visiblesats	Decimal, 2 digits	Number of visible satellites within list
SatxID	Decimal, 2 digits	Satellite x PRN
CorrxTotal	Floating	Total correction applied to satellite x
CorrxSat	Floating	Satellite x contribution to correction
CorrxIono	Floating	Satellite x Ionosphere correction contribution
CorrxTropo	Floating	Satellite x Troposhere correction contribution
checksum	Hexadecimal,2 digits	Checksum of the message bytes

Example:

-

5.13 PSTMSBAS

SBAS Satellite Data
NMEA message list bitmask: 0x20000

Format:
$PSTMSBAS,<Status>,<SatTrk>,<SatID>,<Elev>,<Azim>,<Sig>,*
<checksum><cr><lf>

N = number of tracked satellites

Parameter	Format	Description
Status	Decimal, 1 digit	SBAS Status 0 = no SBAS used

		1 = SBAS used
SatTrk	Decimal, 1 digit	SBAS Satellite tracked 0 = SBAS Satellite not tracked 1 = SBAS Satellite tracked, decoding is ongoing
SatID	Decimal, 3 digits	SBAS Satellite ID
Elev	Decimal, 2 digits	SBAS Satellite Elevation (in degrees)
Azim	Decimal, 3 digits	SBAS Satellite Azimuth (in degrees)
Sig	Decimal, 2 digits	SBAS Satellite Signal Strength CN0 (in dB)
Fixed Character	Delimiter for data field	
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:

$PSTMSBAS,1,0,124,65,090,00*09

6 COMMANDS ANSWER MESSAGES SPECIFICATION

6.1 PSTMALMANAC

Almanac Data Dump. This message is sent as a reply to a $PSTMDUMPALMANAC command.

Format: $PSTMALMANAC,<SatID>,<DataSize>,<HexData>*<checksum><cr><lf>

Parameter	Format	Description
SatID	Decimal, 2 digits	Satellite Number (PRN)
DataSize	Decimal, 2 digits	Number of bytes contained in the “Hex-Data” field
HexData	Hex, n-times 2 digits	Almanac Data in Hex-Format
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:
$PSTMALMANAC,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034024200b4ffff00*1a

6.2 PSTMEPH

Ephemeris Data Dump. This message is sent as a reply to a $PSTMDUMPEPHEMS command.

Format:

$PSTMEPHEM,<SatID>,<DataSize>,<HexData>*<checksum><cr><lf>

Parameter	Format	Description
SatID	Decimal, 2 digits	Satellite Number (PRN)
DataSize	Decimal, 2 digits	Number of bytes contained in the “Hex-Data” field
HexData	Hex, n-times 2 digits	Ephemeris Data in Hex-Format
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters

Example:

$PSTMEPHEM,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400fbff33420000ee632f27698ef001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ffc616fe03*57

7 ALMANACS AND EPHEMERIS MANAGEMENT

Please note that in order for new almanacs and ephemeris data to be stored correctly it is essential that the baud rate is at a maximum of 115200 baud. A higher baud rate will cause the stored data to be corrupted so, it is recommended to use the command to change the port baud rate before start the following procedures (an example is available in the appendix A).

7.1 Using the Assist Commands to Obtain Almanac and Ephemeris Data from a Reference GPS Receiver

The following steps may be used to obtain Ephemeris and Almanac data from the GPS receiver. In order for useful data to be obtained it is best that the GPS receiver has been running long enough to receive a full set of Ephemeris and Almanac data from the satellites.
Note: the Ephemeris data must be less than one hour old, while Almanac can tolerate some days/weeks delay between collection and use.
To ensure the validity of the ephemeris and almanac data it is advisable to clear the Ephemeris and Almanac data stored in the flash of the receiver. This may be done by sending the commands $PSTMCLREPHS and $PSTMCLRALMS. Once this has been done it will be necessary wait for the reference receiver to receive up to date Ephemeris and Almanac data from the satellites, before issuing the dump commands.
It is also useful that the commands have been saved in various text files that may be transmitted over the connection by the terminal emulator. This example makes use of the following files:
SUSPEND.txt

RESUME.txt

DUMPEPHEMS.txt

DUMPALMANAC.txt

The content of these files has been reproduced in section TODO.
Step 1
Ensure that the connection is working and that the user can see NMEA data displayed on their terminal emulator.
Step 2
Ensure that the terminal emulator is logging its input to a text file e.g. log.txt.
Step 3
Before downloading the Almanac and Ephemeris data from the reference receiver, it is advisable to clear any existing Almanac and Ephemeris data from its memory and waiting until a full set of Ephemeris and Almanac data has been received from the satellites. This will ensure the validity of the data downloaded from the reference GPS receiver. This can be achieved by sending the $PSTMCLREPHS and the $PSTMCLRALMS commands.

Step 4
Send the file SUSPEND.txt to the target. The user will notice that the target appears to have stopped working. This is because the GPS library has been suspended.
Note: Steps 5 and 6 are separate operations and may be carried out individually or together depending on the wishes of the user.
Step 5
Send the file DUMPEPHEMS.txt to the target. The user will notice that the Ephemeris data is displayed on the terminal emulator (as shown below). Note that if no data is displayed then there is no Ephemeris data in the flash.
$PSTMEPHEM,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00ecff212c00000ced2b287d1021031f5b0da1b0eabad3c9277301316763b9f90011009184c003*59
$PSTMEPHEM,2,64,42057062706298989841f60034a3ff0017014e23c90ad20095fffeff40360000e59fd126b3f39f04ddda0ca160ecc10ed28dacaa512bc74edb000300e21eff03*09

$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5fff0ff5b36000089e92c26d3a670
0364ca0da109f24862068422525c188929f700f201032bc703*5b
$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00200053370000a4b113261
c5b240333740da1b1d91e956051cf7e3f6ed4b3f60004006fa5db03*00
$PSTMEPHEM,14,64,420570627062c5c5c5e10e007ea9ff0064058520a30ea60416000200772c000024c01b28
451e1f01c49f0ca10aeb5ff83bcf570002bc35acec000400a632ff03*6b
$PSTMEPHEM,21,64,4205706270622121212188f9009da5ff00e7004622cd0aba00d9ff9efffd3500001a618a2634
ba500506010ea1e9f9fa926c745cac2cc31f84e700200044a6c403*3c
$PSTMEPHEM,25,64,42056c626c62b2b2b20c04008ca5ff0007fc3b250b0820fd5b00290079370000ada6bd26d
78f350664e90ca176ebc4a6c5e0fd26c93f03c6f00007004d12c003*3d
$PSTMEPHEM,30,64,420570627062b0b0b091f800caa6ff00cff8e2179e1355f999ffc0ff553500003f077326f97e
6c04c8140da10c14be42db05f853b7a66b34ef005e009ff7cd03*3e

Step 6

Send the file DUMPALMANAC.txt to the target. As in the previous step the user will notice that the Almanac data is displayed on the terminal emulator (as shown below). Note that if no data is displayed then there is no Almanac data in the flash.

$PSTMALMANAC,1,32,0142056314325b1c5efd0140020da14009730160ad61b900caffe12011088020*1d
$PSTMALMANAC,2,32,02420563034ab50634fd01406c0ca1402eacaa6047c64e005b741c20e4078020*15
$PSTMALMANAC,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f1
9000a3ef92030088020*1c

$PSTMALMANAC,4,32,04420563f93a700633fd0140450da140447bab606fd202008ec97f201e208020*1a
$PSTMALMANAC,5,32,054205630d3765fc3ffd0140500da14033225260f08929006cf96f20e6808020*19
$PSTMALMANAC,6,32,064205634532d6fa3ffd0140fc0ca14018cf7e600cd4b30037d0a22075038020*49
$PSTMALMANAC,7,32,07420563f56cd9fb3ffd0140d20da1402eb77d6082d2b7003bdcfa2099218020*13
$PSTMALMANAC,8,32,08420563ee4e011242fd0140190da14072452c609b4a6900fbe2a620d0078020*1b
$PSTMALMANAC,9,32,09420563588ed00938fd0140cf0ca1406728296083eb3000c2729720f1078020*44
$PSTMALMANAC,10,32,0a420563ed35ee155ffd0140ac0da140f82cd6609c7a0e004eb22a204c008020*76
$PSTMALMANAC,11,32,0b420563fc2632e406fd0140fc0ca1403c39a56064700a00608bbe2023098020*7b
$PSTMALMANAC,13,32,0d4205632315171f64fd0140ca0ca140d1d4006012ed2d00d0a1242016088020*2c
$PSTMALMANAC,14,32,0e420563f711581b5efd0140480ca140b2570060bd35ac002a110620e6078020*20
$PSTMALMANAC,15,32,0f420563f14a070b3bfd0140780ba1400dc3ad60b14366000ce9a92017128020*2f
$PSTMALMANAC,16,32,10420563c917770c58fd0140550ca140199f55601c2bd800a2196b200d008020*24
$PSTMALMANAC,17,32,114205630c0d1a0c54fd0140430ca140aeef7f6043406d0008044920c427c020*79
$PSTMALMANAC,18,32,12420563c0367d0b50fd0140b30ca140c130d76094349100f755672031ffbf20*25
$PSTMALMANAC,19,32,13420563b01ad60a51fd01409a0da140d1628260fc19c500a7d23520e4078020*72
$PSTMALMANAC,20,32,14420563e0133f0b4efd0140830ca140db0ad560ed613a00a1365a20d3078020*7c
$PSTMALMANAC,21,32,154205630955410230fd0140880da1400d5cac60921f84007faca02095088020*29
$PSTMALMANAC,22,32,164205631029da094efd0140140da140808ad7608e4abf00dbfc212032088020*27
$PSTMALMANAC,23,32,174205630f23bf0f51fd0140a50ca140a0f0ff60905c6100172d0720aff8bf20*7d
$PSTMALMANAC,24,32,184205634b4a1f0d3ffd01404d0da1400ec6ac604db9d40006aac7203c088020*2c
$PSTMALMANAC,25,32,19420563596376052ffd0140760ca1408bfd26603c01c600e9d9b42002008020*28
$PSTMALMANAC,26,32,1a420563fd87eb1d61fd0140bc0ca140e5e2006013041e001389e320f7ffbf20*22
$PSTMALMANAC,27,32,1b4205630e9e660834fd0140720da140313f28606565ae002a2d772016008020*7b
$PSTMALMANAC,28,32,1c4205631756300b57fd0000dc0ca1402f06562082c6a12050f344002a008000*25

$PSTMALMANAC,29,32,1d4205638f49d21b60fd0140090da1407880ff60c018d500009535209529 8020*73
$PSTMALMANAC,30,32,1e420563ca46c70045fd0140a00ca140baf75360466c3400e26e5020bf198020*28

Step 7

To resume the GPS library operation send the file RESUME.txt.

Step 8

The Almanac and Ephemeris data should now be saved the log file. These can be extracted for loading to a new target GPS receiver by copying the $PSTMALMANAC and $PSTMEPHEM lines into a new file, ensuring that there is no wrapping of lines introduced by the editor.

7.2 Using the Assist Commands to Load Almanacs and Ephemeris Data into a Target Receiver

The following steps may be used to load Ephemeris and Almanac data to the GPS receiver. All the explanations in this chapter are related to a system that includes Flash Memory for data storage, it will however also work in a system with battery backup to retain data in an embedded SRAM. All data storage management is supported by ST´s GPS Library.

Note: Ephemeris data must be less than one hour old, while Almanac can tolerate some days/weeks delay between collection and use.

Data within the GPS receiver is stored in a double buffered arrangement controlled by NVM management software. The double buffering makes use of two banks of flash to store data. This means that if new data is being written to the flash and fails for whatever reason, the previous version of the data can be recovered to ensure that the receiver software can continue to function.

The mechanism that is employed to achieve this double buffering results in the following effect. Assuming that 4 almanac entries are already existing in the NVM flash, and we wish to download a complete almanac to the receiver. When the NVM management software detects that a version of the data it is trying to write already exists then it will copy everything from one bank to the other before swapping banks. It will then continue writing to the new bank until it the same condition arises. Then it will copy everything to the other bank and swap banks again.

In order to prevent the multiple copying and swapping of banks it is better to ensure that the NVM area of flash is clear of Almanac and Ephemeris data before loading new Ephemeris and Almanac data to the receiver. In a production environment it should be the case that there is no Ephemeris and Almanac data in the flash. However if the Almanac and Ephemeris data is being loaded in the field it is important to clear any existing data using the $PSTMCLREPHS and $PSTMCLRALMS commands.

It is useful that the commands have been saved in various text files that may be transmitted over the connection by the terminal emulator.

This example makes use of the following files:

SUSPEND.txt

RESUME.txt LOADEPHEMS.txt LOADALMANAC.txt

The content of these files has been reproduced in section TODO.

Step 1

Ensure that the connection is working and that the user can see NMEA data displayed on their terminal emulator.

Step 2

Before loading the receiver with new Almanac and Ephemeris data it is necessary to clear any existing Almanac and Ephemeris data from its memory. If this is not done the receiver will make a copy of the data already within its memory before loading the new data into memory. This will result in twice as many erase and write operations occurring on the flash memory of the receiver. This can be achieved by sending the $PSTMCLREPHS and the $PSTMCLRALMS commands.

Step 3

Send the file SUSPEND.txt to the target. The user will notice that the target appears to have stopped working. This is because the GPS library has been suspended.

Note: Steps 4 and 5 are separate operations and may be carried out individually or together depending on the wishes of the user.

Step 4

Send the file LOADEPHEMS.txt to the target. This will load the ephemeris data into the target flash. If the user wishes to verify that the ephemeris data has been downloaded they can do so by issuing a hot start command ($PSTMHOT). Note that it is important that they resume the operation of the GPS library before issuing the hot start command otherwise the hot start command will fail. This is possible via the $PSTMRESUME command.

Step 5

Send the file LOADALMANAC.txt to the target. This will load the almanac data into the target flash.

Step 6

To resume the GPS library operation send the file RESUME.txt.

In order to use these commands to truly assist a GPS receiver in a cold start scenario, it is also necessary to issue position and time information using the $PSTMINITGPS command before loading the Almanac and Ephemeris data. It is important that the time in this case corresponds to the Ephemeris and Almanac data otherwise the receiver will reject the data as being invalid.

7.3 Summary of text files used in the examples

7.3.1 File: SUSPEND.txt $PSTMSUSPEND

7.3.2 File: RESUME.txt $PSTMRESUME

7.3.3 File: DUMPEPHEMS.txt $PSTMNMEAONOFF $PSTMDUMPALMANAC

7.3.4 File: DUMPALMANAC.txt $PSTMNMEAONOFF $PSTMDUMPALMANAC

File: LOADALMANAC.txt $PSTMLOADALM

$PSTMALMANAC,1,32,0142056314325b1c5efd0140020da14009730160ad61b900caffe12011088020*1d
$PSTMALMANAC,2,32,02420563034ab50634fd01406c0ca1402eacaa6047c64e005b741c20e4078020*15
$PSTMALMANAC,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f19000a3ef92030088020*1c
$PSTMALMANAC,4,32,04420563f93a700633fd0140450da140447bab606fd202008ec97f201e208020*1a
$PSTMALMANAC,5,32,054205630d3765fc3ffd0140500da14033225260f08929006cf96f20e6808020*19
$PSTMALMANAC,6,32,064205634532d6fa3ffd0140fc0ca14018cf7e600cd4b30037d0a22075038020*49
$PSTMALMANAC,7,32,07420563f56cd9fb3ffd0140d20da1402eb77d6082d2b7003bdcfa2099218020*13
$PSTMALMANAC,8,32,08420563ee4e011242fd0140190da14072452c609b4a6900fbe2a620d0078020*1b
$PSTMALMANAC,9,32,09420563588ed00938fd0140cf0ca1406728296083eb3000c2729720f1078020*44
$PSTMALMANAC,10,32,0a420563ed35ee155ffd0140ac0da140f82cd6609c7a0e004eb22a204c008020*76
$PSTMALMANAC,11,32,0b420563fc2632e406fd0140fc0ca1403c39a56064700a00608bbe2023098020*7b
$PSTMALMANAC,13,32,0d4205632315171f64fd0140ca0ca140d1d4006012ed2d00d0a1242016088020*2c

$PSTMALMANAC,14,32,0e420563f711581b5efd0140480ca140b2570060bd35ac002a110620e6078020*20

$PSTMALMANAC,15,32,0f420563f14a070b3bfd0140780ba1400dc3ad60b14366000ce9a92017128020*2f

$PSTMALMANAC,16,32,10420563c917770c58fd0140550ca140199f55601c2bd800a2196b200d008020*24$PSTMALMANAC,17,32,114205630c0d1a0c54fd0140430ca140aeef7f6043406d0008044920c427c020*79$PSTMALMANAC,18,32,12420563c0367d0b50fd0140b30ca140c130d76094349100f755672031ffbf20*25$PSTMALMANAC,19,32,13420563b01ad60a51fd01409a0da140d1628260fc19c500a7d23520e4078020*72

$PSTMALMANAC,20,32,14420563e0133f0b4efd0140830ca140db0ad560ed613a00a1365a20d3078020*7c

$PSTMALMANAC,21,32,154205630955410230fd0140880da1400d5cac60921f84007faca02095088020*29

$PSTMALMANAC,22,32,164205631029da094efd0140140da140808ad7608e4abf00dbfc212032088020*27

$PSTMALMANAC,23,32,174205630f23bf0f51fd0140a50ca140a0f0ff60905c6100172d0720aff8bf20*7d

$PSTMALMANAC,24,32,184205634b4a1f0d3ffd01404d0da1400ec6ac604db9d40006aac7203c088020*2c

$PSTMALMANAC,25,32,19420563596376052ffd0140760ca1408bfd26603c01c600e9d9b42002008020*28

$PSTMALMANAC,26,32,1a420563fd87eb1d61fd0140bc0ca140e5e2006013041e001389e320f7ffbf20*22

$PSTMALMANAC,27,32,1b4205630e9e660834fd0140720da140313f28606565ae002a2d772016008020*7b

$PSTMALMANAC,28,32,1c4205631756300b57fd0000dc0ca1402f06562082c6a12050f344002a008000*25

$PSTMALMANAC,29,32,1d4205638f49d21b60fd0140090da1407880ff60c018d5000095352095298020*73

$PSTMALMANAC,30,32,1e420563ca46c70045fd0140a00ca140baf75360466c3400e26e5020bf198020*28

7.4 File: LOADEPHEMS.txt

$PSTMLOADEPHEMS

$PSTMEPHEM,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00ecff212c00000ced2b287d1021031f5b0da1b0eabad3c9277301316763b9f90011009184c003*59

$PSTMEPHEM,2,64,42057062706298989841f60034a3ff0017014e23c90ad20095fffeff40360000e59fd126b3f39f04ddda0ca160ecc10ed28dacaa512bc74edb000300e21eff03*09

$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5fff0ff5b36000089e92c26d3a6700364ca0da109f24862068422525c188929f700f201032bc703*5b

$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00200053370000a4b113261c5b240333740da1b1d91e956051cf7e3f6ed4b3f60004006fa5db03*00

Contact Information

Fastrax Ltd.

Street Address: Polaris Business Park, Itsehallintokuja 6,

02600 Espoo, FINLAND

Tel: +358 (0)424 733 1

Fax: +358 (0)9 8240 9691

http://www.fastraxgps.com

E-mail:

Sales: sales@fastraxgps.com

Support: support@fastraxgps.com