1 Introduction

The purpose of this document is to provide an overview of the various NMEA commands and messages for the STMicroelectronics´ GPS Systems. This document is relevant for the following Baseband Processors and related GPS Software Revisions.

Device Type	Software Release	Comment
STA 2062	7.1.1.15 or later	
STA 2064	7.1.1.15 or later	
STA 8088	7.1.1.15 or later	

2 Contents

2.1 Index

2.2 List of Tables

2.3 List of Figures

None

3 Document Management

3.1 Revision History

Rev	Date	Author	Notes
1.3	2006-10-25	A. Di Girolamo	Doc.Name: "GPS NMEA Commands"
2.3	2009-02-23	F. Henkel	Extended with NMEA Messages and reworked
2.4	2009-07-06	F. Henkel	Added "$PSTMKFCOV"
2.5	2009-07-07	F. Henkel	Added $GPGGA5 Message, Minor Changes in Descriptions and formatting
2.6	2009-07-15	F. Henkel	Correction of $GPGSA Fix Status
2.7	2009-07-21	F. Henkel	Correction of $PSTMSBAS Example, Removed "Draft"
2.8	2009-10-12	A. Di Girolamo	Added $PSTMAGPS. Added Footnotes. Removed $PSTMFDAONOFF, replaced by $PSTMGETALGO and $PSTMSETALGO. Changed bitmask for $PSTMKFCOV. Document review.
2.9	2010-01-27	A. Di Girolamo	STA205x Configuration Review Added Configuration for STA206x families
2.10	2010-04-02	A. Di Girolamo	Document Layout Review. Added Default Settings Configuration.
2.11	2010-09-10	F. Henkel	Added $PSTMCOLD Parameters
3.0	2011-04-07	F. Boggia	New layout
3.1	2011-04-11	A. Di Girolamo	Document review Initial draft release Updated to GNSS library 7.1.1.15
3.2	2011-06-17	A. Di Girolamo	Added new software configuration features introduced on rel. 7.1.6.25
3.3	2011-08-08	A. Cascella	Added new command to select local geodetic datum
3.4	2011-09-06	A. Di Girolamo	Added additional PPS configuration parameters. Added DEBUG ON/OFF command. Fixed $PSTMAGPS age and satid evaluation.

3.5	2011-12-23	A. Di Girolamo	Added additional configuration parameters introduced in SW 7.1.11.34
3.7	2012-02-27	A. Di Girolamo	Removed the firmware configuration section. It has been moved on separate document. Added commands and messages for the Position Hold management. Added command for GNSS constellation setting. Added CPU usage NMEA output message. Added command to perform a system reset. Added NMEA message for Notch Filter status. Removed $PSTMSBASCOR message.
3.7.1	2012-04-05	A. Occhipinti	Added commands to read/write/erase user SQI memory area.

Table 1: Revision history

3.2 Acronyms

Keyword	Definition
Accuracy	Deviation of a GPS-based calculated position from the true position
Almanac	Contains the information about all available satellites , their orbit data and time of their clocks.
Azim	Azimuth - Angular distance from a reference
Bank Swap	Exchanging two memory banks for storage of data
BAUD rate	Transmission Rate Measure for the effective transmission of data content. (may differ from Bits/sec).
Checksum	Calculated from the transmitted characters of a message by “ex-OR”ing the 8 bit character values including delimeters (without checksum).
CN0	Carrier to Noise Ratio - Identifies the quality of a received signal
Cold Start	Start Condition for a GPS system having no position nor time. Almanac and Ephemeris is not available, too.
Dead Reckoning	Sensor based process to determine the movement of a mobile unit, utilizing Gyro,Odometer and Wheel Pulses.
Delimiter (within NMEA 0183)	ASCII “$” to indicate Address Field ASCII “,” to indicate Data Field ASCII “*” to indicate Checksum Field
DGPS	Differential GPS - GPS Augmentation System providing the accurate location of a Reference Station to reduce system errors.
EGNOS	European Geostationary Navigation Overlay System
Elev	Elevation - Angle between a high level or non-earth bound point and the horizontal plane of the viewer.
Ephemeris	Ephemeris Data is transmitted by each satellite and contains current and predicted satellite position.
FDA	Failure Detection Algorithm - Specific Algorithm to detect failures in position calculation
GDOP	Geometric Dilution Of Position - Quality value representing all geometry based error factors in a system.
GNSS	Global Navigation Satellite System - Satellite based system to calculate the position of the receiver on the earth surface.
GPS	Global Positioning System - United States Satellite Navigation System

GPS Library	STMicroelectronics C-Library containing all GPS relevant Functions
Gyro	Gyroscope - Sensor to determine rotational movements
HDOP	Horizontal Dilution Of Precision - Quality value representing all 2D plane geometry based error factors in a system .
Hot Start	Start Condition for a GPS System having position, time, Alamanc and Ephemeris already available
Lat	Lattitude - Angular difference of a given position to the Equator. Values include 0°-90° either North or South
Lat-Ref	Lattitude Reference - Reference if a Latitude value is North or South
Long	Longitude - Angular difference to a “reference” Longitude indicated as “000”. Values include 0°… 180° either West or East.
Long-Ref	Longitude Reference - Reference if a Longitude value is East or West of the “000” Meridian.
NMEA	National Marine Electronics Association - United States Standards Organisation For Marine Equipment
NMEA 0183	National Marine Electronics Association - Standard for Interfacing Marine Electronics Devices
NVM	Non Volatile Memory - Any type of memory that conserves data in the absence of regular supply voltage (includes battery buffered memories)
Proprietary Message	Messages within the scope of NMEA0183 which are not standardized. They start with $P and a 3 character identifier.
PRN	Pseudo Random Number - Satellite Specific 1023 Bit Number used for Spread Spectrum Modulation
RF	Radio Frequency - High Frequency for Reception with a RF-Receiver
RS232	IEEE Standard - Physical Layer Standard for Data Transmission
Sat-ID	Satellite Identifier - Satellite specific Number used to generate the corresponding PRN code
SBAS	Satellite Based Augmentation System - GPS enhancement system based on geostationary satellites.
Static Position Filtering	Algorithm to detect that the GPS receiver doesn´t move and position output is kept stable.
UTC	Universal Time Coordinated
WAAS	Wide Area Augmentation System - American GPS Augmentation System delivering accurate Ionosphere Data

Warm Start	Start Condition for a GPS system having current Alamanc and Ephemeris available. Time needs to be available with reasonable accuracy (some seconds).
2D Fix	Fix based on the use of 3 satellites
3D Fix	Fix based on the use of 4 satellites

Table 2. Acronyms

3.3 Reference Documents

None

3.4 Contact info

Keyword	Definition
A. Di Girolamo	andrea.di-girolamo@st.com
A. Cascella	antonio.cascella@st.com
F. Boggia	fulvio.boggia@st.com

4 Communication Interface

Communication between a host processor and the ST GPS System can be established in different ways, depending on the implementation of the Baseband Processor as a stand-alone unit or as an integrated subsystem on a “System on Chip”.

For simplicity reasons this document will refer to “Stand-alone Processors” only and the interface described in the examples is a UART. There are other implementations to communicate like USB or SPI. The hardware interface used will not influence the data content send or received.

All information contained in this document is related to the “NMEA port” of the Baseband Processor. STMicroelectronics GPS Systems may contain an additional “Debug port” but the data exchanged on the “Debug Port” is not within the scope of this document.

4.1 Commands:

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 (please refer to the appendix A, at the end of the document, for some examples) and send it using the “send file” capability of the terminal emulator. For this reason, it is required that the terminal emulator (or production test program) running on the PC is capable of sending text files down the RS232 link to the GPS receiver.

4.2 Messages:

A Message is a defined set of data sent from the GPS System to a host processor using the same interface which is used to transfer commands to the system. Messages may not be enabled by default but can be switched on and off using a command at run-time. The basic structure of a message is:

```
message-ID, <parameters> <cr><lf>
```

There are two basic sets of messages implemented.

4.2.1 Standard NMEA Messages

Standard NMEA Messages are defined in the “NMEA 0183” Standard, issued from the “National Marine Electronics Association”. The latest issue is Rev. 3.1 dated January 2002. NMEA0183 refers to it as Sentences (single line message) and Messages (multiple line messages).

To get an overview on the standard NMEA messages supported by ST´s GPS Systems please refer to 7.1 “Standard NMEA Messages” in this document.

Standard NMEA messages start the “message-ID” with:

```
$<Talker ID>
```

Supported talker IDs are: “GP”, “GL” and “GN” for standard NMEA sentences.

4.2.2 Proprietary Messages

The STMicroelectronics GPS System 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…
```

To get an overview on the proprietary messages defined by STMicroelectronics please refer to chapter 4.2 in this document.

5 Commands[1]

5.1 Software Command List:

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

Syntax	Description
$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
$PSTMSRR	System Reset
$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.
$PSTMNVMSWAP[2]	Execute a bank swap on the NVM GPS backup memory

[1] If not explicitly declared, all commands which modify the status of parameters, modifications

$PSTMSBASONOFF	Enable/Disable the SBAS activity
$PSTMSBASSAT	Set the SBAS satellite's ID
$PSTMRFTESTON	Enable the RF test mode
$PSTMRFTESTOFF	Disable the RF test mode
$PSTMGETALGO	Get FDE algorithm ON/OFF status
$PSTMSETALGO	Set FDE algorithm ON/OFF status
$PSTM2DFIXONOFF	Enable/Disable the 2D fix algorithm
$PSTMGETRTCTIME	Get the current RTC time.
$PSTMSELECTDATUM	Set a geodetic local datum different from WGS84
$PSTMDATUMSETPARAM	Set parameters to local geodetic to WGS84 datum transformations
$PSTMENABLEPOSITIONHOLD	Set status and position for the Position Hold feature.
$PSTMSETCONSTMASK	Set GNSS constellation mask.
$PSTMNOTCH	Set the ANF operation mode.
$PSTMSQISET	Set bytes at the specified address.
$PSTMSQIGET	Get bytes at the specified address.
$PSTMSQIERASE	Erase the SQI data storage area.
$PSTMSETPAR	Set System Parameter in the configuration data block.
$PSTMGETPAR	Get System Parameter from configuration data block.
$PSTMSAVEPAR	Save System Parameters in the GNSS backup memory.
$PSTMRESTOREPAR	Restore System Parameters (Factory Settings).

Warning: The $PSTMSETPAR command allows the direct modification of the system parameters. Wrong Settings may degrade the GNSS system performance or even stop the system from working

[2] This command is supported only by platforms or SW configurations where the GNSS backup memory is based on Flash NOR or SQI memories.

5.2 NMEA commands

5.2.1 $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 (-1500 to 100000)
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
```

5.2.2 $PSTMINITFRQ

Initialize the centre frequency. This command can be used to set the local oscillator frequency offset.

Synopsis:

```
$PSTMINITFRQ,<offset><cr><lf>
```

Arguments:

Parameter	Format	Description
offset	dddddd - Decimal, 6 digits	Frequency offset in Hz

Results:

- The centre frequency will be initialized
- No message will be sent as reply.

Example:

```
$PSTMINITFRQ,-47000
```

5.2.3 $PSTMSETRANGE

Set the frequency range for satellite searching. The “min.” and “max.” values are used as offsets versus the centre frequency.

Synopsis:

```
$PSTMSETRANGE,<min>,<max><cr><lf>
```

Arguments:

Parameter	Format	Description
min	dddddd - Decimal, 6 digits	Lower limit range in Hz
max	dddddd - Decimal, 6 digits	Upper limit range in Hz

Results:

- The following message will be output on NMEA communication channel:

```
$PSTMSETRANGEOK<cr><lf>            if success
$PSTMSETRANGEERROR<cr><lf>         if no success
```

Example:

```
   $PSTMSETRANGE,-57000,-37000
```

5.2.4 $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
```

5.2.5 $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>
```

Where:

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,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400
fbff33420000ee632f27698ef001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ff
c616fe03*57
$PSTMEPHEM,2,64,0f06bc34bc344f4f4f78110019a5ff00b004fa1d1e0e3f04c8ff
caff1937000033515726556ba9048eae0da1b6c346bd8f985c93ade10c76db001d00
f8c7c503*58
$PSTMEPHEM,4,64,0f06bb34bb344b4b4b98050038a4ff000005351e110eea041b00
b8ffd037000020b84e26b5138b0425580ca16b211030e68b1a949cac9615f30066ff
ea92f603*06
$PSTMEPHEM,9,64,0f06bc34bc341818189c0a0069aaff005f06eb249a09ca0477ff
6c00f72e00005131d827592b950a91010da1c7af88538e7ca1122fb9be3df4001300
c4a0c203*52
```

5.2.6 $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,12,64,0f06bc34bc3437373790f40045a7ff00fcf5d522480b4bf71b0
0fbff8931000096126f271f869101c3870ca107afce79a763e13e360a1ce8e700310
0380ff903*36
```

5.2.7 $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
```

5.2.8 $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>
```

Where:

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,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034
024200b4ffff00*1a
$PSTMALMANAC,2,32,021a0690944b78fe37fd0800770da141ef0c5b006048770098
9bd800d8088000*1a
$PSTMALMANAC,3,32,031a06904f68a2f540fd0800f60ca141922a2c003cae270094
96cf00020a8000*15
$PSTMALMANAC,4,32,041a0690a94aeffd36fd0800390ca141afc95b00de7a1700df
c74e004ddebf00*13
$PSTMALMANAC,5,32,051a0690940eee0b5efd0800900ca141582b8600d3000b0060
641200e40f8000*14
```

5.2.9 $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>
```

Arguments:

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.

Results:

- The almanac will stored into backup memory
- No message will be sent as reply.

Example:

```
$PSTMALMANAC,12,32,0c1a06907c1a971160fd0800fa0da141ae9f0600d912e9007
5669700490f8000*75
```

Note: for further details about the almanacs management please refer to the dedicated chapter below.

5.2.10 $PSTMCOLD

Perform a COLD start.

Synopsis:

```
$PSTMCOLD, < Mask ><cr><lf>
```

Arguments:

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 restart[3].
- 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
```

[3] The GPS engine will be reset. It is not a system reboot.

5.2.11 $PSTMWARM

Perform a WARM start.

Synopsis:

```
$PSTMWARM<cr><lf>
```

Arguments:

None.

Results:

- Warm start initialization and system restart[4].

Example:

```
$PSTMWARM
```

[4] The GPS engine will be reset. It is not a system reboot.

5.2.12 $PSTMHOT

Perform an HOT start.

Synopsis:

```
$PSTMHOT<cr><lf>
```

Arguments:

None.

Results:

- The system restart[5].

Example:

```
$PSTMHOT
```

[5] The GPS engine will be reset. It is not a system reboot.

5.2.13 $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_off[6]	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
```

[6] The “on_off” input parameter has been added starting from SW re. 7.1.9.29. For backward compatibility the old command syntax is still supported: sending $PSTMNMEAONOFF with no input parameter the NMEA ON/OFF status is toggled.

5.2.14 $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

Results:

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

Example:

```
$PSTMDEBUGONOFF,0
```

5.2.15 $PSTMSRR

Executes a system reset. The GNSS firmware is rebooted.

Synopsis:

```
$PSTMSRR<cr><lf>
```

Arguments:

None.

Results:

- The GNSS firmware reboots.
- No message will be sent as reply.

Example:

```
$PSTMSRR
```

5.2.16 $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
```

5.2.17 $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
```

5.2.18 $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
```

5.2.19 $PSTMTIMEINV

Invalidate the Real Time Clock (RTC).

Synopsis:

```
$PSTMTIMEINV<cr><lf>
```

Arguments:

None.

Results:

- The RTC time will be invalidated.

Example:

```
$PSTMTIMEINV
```

5.2.20 $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
```

5.2.21 $PSTMNVMSWAP[7]

Execute a bank swap on the NVM GPS backup memory.

Synopsis:

```
$PSTMNVMSWAP<cr><lf>
```

Arguments:

None.

Results:

- The non-volatile backup memory banks will be swapped
- No message will be sent as reply

Example:

```
$PSTMNVMSWAP
```

[7] This command is supported only by platforms or software configurations where the backup memory is based on Flash NOR or SQI memories.

5.2.22 $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
```

5.2.23 $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)

Results:

- If the SBAS satellite is available in the above range, the software starts tracking. If the parameter is zero, the system automatically searches for the SBAS satellite available in the user region.

Example:

```
$PSTMSBASSAT,128
```

5.2.24 $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
```

5.2.25 $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
```

5.2.26 $PSTMGETALGO

Get False Detection and Exclusion (FDE) algorithm ON/OFF status.

Synopsis:

```
$PSTMGETALGO,<algo_type><cr><lf>
```

Arguments:

Parameter	Format	Description
algo_type	Decimal, 1 digit	1 = FDE algorithm on/off status is returned.

Results:

- If success the following message is sent:

```
$PSTMGETALGOOK,<algo_type>,<algo_status>*<checksum><cr><lf>
```

Parameter	Format	Description
algo_type	Decimal, 1 digit	1 = FDE algorithm on/off status is returned.
algo_status	Decimal, 1 digit	0 = the algorithm is disabled. 1 = the algorithm is enabled.

- In case of error the following message will be sent:

```
$PSTMGETALGOERROR*<checksum><cr><lf>
```

Example:

```
$PSTMGETALGO,1
```

5.2.27 $PSTMSETALGO

Set False Detection and Exclusion (FDE) algorithm ON/OFF status.

Synopsis:

```
$PSTMSETALGO,<algo_type>,<algo_status><cr><lf>
```

Arguments:

Parameter	Format	Description
algo_type	Decimal, 1 digit	1 = FDE algorithm on/off status is returned.
algo_status	Decimal, 1 digit	0 = the algorithm is disabled. 1 = the algorithm is enabled.

Results:

- If success the following message is sent:

```
$PSTMSETALGOOK,<algo_type>,<algo_status>*<checksum><cr><lf>
```

Parameter	Format	Description
algo_type	Decimal, 1 digit	1 = FDE algorithm on/off status is returned.
algo_status	Decimal, 1 digit	0 = the algorithm is disabled. 1 = the algorithm is enabled.

- In case of error the following message will be sent:

```
$PSTMSETALGOERROR*<checksum><cr><lf>
```

Example:

```
$PSTMSETALGO,1,0
```

5.2.28 $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
```

5.2.29 $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
```

5.2.30 $PSTMDATUMSELECT

Set a local geodetic datum different from WGS84 (defult).

Synopsis:

```
$PSTMDATUMSELECT,<datum_type><cr><lf>
```

Arguments:

Parameter	Format	Description
datum_type	Integer	The following datum are selectable: 0 : WGS84 1: TOKYO MEAN 2: OSGB

Results:

- If success the following message is sent:

```
$PSTMDATUMSELECTOK,<datum_type>*<checksum><cr><lf>
```

Parameter	Format	Description
datum_type	Integer	0 : WGS84 1: TOKYO MEAN 2: OSGB

- In case of error the following message will be sent:

```
$PSTMSELECTDATUMERROR*<checksum><cr><lf>
```

Example:

```
$PSTMSELETDATUM,1
```

5.2.31 $PSTMDATUMSETPARAM

Set parameters to local geodetic to WGS84 datum transformations.

Synopsis:

```
$PSTMDATUMSETPARAM,<d_x>,<d_y>,<d_z>,<d_a>,<d_f><cr><lf>
```

Arguments:

Parameter	Format	Description
d_x d_y d_z	Decimal	shifts between centers of the local geodetic datum andWGS84 Ellipsoid
d_a	Decimanl	differences between the semi-major axis of the local geodetic datum ellipsoid and the WGS 84 ellipsoid, respectively (WGS 84 minus Local)
d_f	Decimal	differences between flattening of the local geodetic datum ellipsoid and the WGS 84 ellipsoid, respectively (WGS 84 minus Local)

Results:

- If success the following message is sent:

```
$PSTMDATUMSETPARAMOK*<checksum><cr><lf>
```

- In case of error the following message will be sent:

```
$PSTMDATUMSETPARAMERROR*<checksum><cr><lf>
```

Example:

```
$PSTMDATUMSETPARAM,-375,111,-431,-573.60, -0.000011960023
```

5.2.32 $PSTMENABLEPOSITIONHOLD

Enable/disable and set position for the Position Hold feature.

Synopsis:

```
$PSTMENABLEPOSITIONHOLD,<on_off>,<Lat>,<LatRef>,<Lon>,<LonRef>,<Alt>
<cr><lf>
```

Arguments:

Parameter	Format	Description
on_off	%d	Set the position hold enable/disable status: 0: disabled. 1: enabled.
Lat	DDMM.MMMMM	Latitude (Degree-Minute.Minute decimals)
LatRef	'N' or 'S'	Latitude direction (North or South)
Lon	DDDMM.MMMMM	Longitude (Degree-Minute.Minute decimals)
LonRef	'E' or 'W'	Longitude Direction (East or West)
Alt[8]	dddddd.dddd	Altitude in meters (-1500 to 100000)

Results:

- If success the following message is sent:

  If on_off = 1

```
$PSTMPOSITIONHOLDENABLED*<checksum><cr><lf>
```

  If on_off = 0

```
$PSTMPOSITIONHOLDDISABLED*<checksum><cr><lf>
```

- In case of error the following message will be sent:

[8] The altitude value must be reported without any geoid correction. It means that if the altitude value is retrieved by the $GPGGA message it must be added to the geoid correction before using it in the $PSTMENABLEPOSITIONHOLD command. This limitation may be removed in the future releases.

```
$PSTMENABLEPOSITIONHOLDERROR*<checksum><cr><lf>
```

Example:

```
$PSTMENABLEPOSITIONHOLD,1,4811.365,N,01164.123,E,0530.0
```

5.2.33 $PSTMSETCONSTMASK

Set the GNSS constellation mask. It allows switching the GNSS constellation at run-time.

Synopsis:

```
$PSTMSETCONSTMASK,<constellation_mask><cr><lf>
```

Arguments:

Parameter	Format	Description
constellation_mask	%d	It is a bit mask where each bit enable/disable a specific constellation independently by the others: bit 0: GPS constellation enabling/disabling bit 1: GLONASS constellation enabling/disabling bit 2: QZSS constellation enabling/disabling

Results:

- If success the following message is sent:

```
$PSTMSETCONSTMASKOK,<constellation_mask>*<checksum><cr><lf>
```

- In case of error the following message will be sent:

```
$PSTMSETCONSTMASKERROR*<checksum><cr><lf>
```

Examples:

Enabling GPS only:

```
$PSTMSETCONSTMASK,1
```

Enabling GLONASS only:

```
$PSTMSETCONSTMASK,2
```

Enabling GPS and GLONASS:

```
$PSTMSETCONSTMASK,3
```

5.2.34 $PSTMNOTCH

Set the NOTCH filter operating mode.

Synopsis:

```
$PSTMNOTCH,<Sat_type>,<Mode><cr><lf>
```

Arguments:

Parameter	Format	Description
Sat_type	Decimal, 1 digits	Sat type ANF path [0 -> GPS; 1->GLONASS]
Mode	Decimal, 1 digits	ANF operation mode

Results:

- This command set the NOTCH filter operating mode in three different ways.

Example:

```
$PSTMNOTCH,0,1          [GPS path, always ON mode]
$PSTMNOTCH,0,2          [GPS path, auto insertion mode]
$PSTMNOTCH,0,0          [GPS path, ANF disabled]
$PSTMNOTCH,1,1          [GLONASS path, always ON mode]
$PSTMNOTCH,1,2          [GLONASS path, auto insertion mode]
$PSTMNOTCH,1,0          [GLONASS path, ANF disabled]
```

5.2.35 $PSTMSQISET

Sets 8 consecutive words into the SQI Data Storage Area starting from the specified address.

Synopsis:

```
$PSTMSQISET,<offset>,<word1>,…,<word8><cr><lf>
```

Arguments:

Parameter	Format	Description
offset	HexDecimal, 4 digits	Offset from the base address of the chosen sector
word1	HHHHHHHH - Hexadecimal, 8 digits	32 bits-wide word
word8	HHHHHHHH - Hexadecimal, 8 digits	32 bits-wide word
dest_addr	HexDecimal, 4 digits	Destination Address in which the data bytes are stored; it is composed by: *sector base address + offset*
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Results:

- if success the following message is sent

```
$PSTMSQISETOK,<dest_addr>*<checksum><cr><lf>
```

- in case of error the following message will be sent

```
$PSTMSQISETERROR*<checksum><cr><lf>.
```

Example:

```
$PSTMSQISET,0xa0,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88
the following 8 bytes (0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88) are consecutively written in the SQI Data Storage Area, strating from offset 0xa0 (i.e. at address 0x300F00a0)
```

5.2.36 $PSTMSQIGET

Starting from the specified address, it gets 8 consecutive words from the SQI Data Storage Area.

Synopsis:

```
$PSTMSQIGET,<offset><cr><lf>
```

Arguments:

Parameter	Format	Description
offset	HexDecimal, 4 digits	Offset from the base address of the chosen sector
word1	HHHHHHHH - Hexadecimal, 8 digits	32 bits-wide word
word8	HHHHHHHH - Hexadecimal, 8 digits	32 bits-wide word
dest_addr	HexDecimal, 4 digits	Destination Address in which the data bytes are stored; it is composed by: *sector base address* + *offset*
checksum	cc - Hexadecimal, 2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Results:

- if success the following message is sent

```
$PSTMSQIGETOK,<dest_addr>,<word1>,….,<word8>*<checksum><cr><lf>
```

- in case of error the following message will be sent

```
$PSTMSQIGETERROR*<checksum><cr><lf>
```

Example:

```
$PSTMSQIGET,0xa0
The following NMEA command gets the 8 consecutive words contained in the SQI Data Storage starting from offset 0xa0 (i.e. starting from destination address 0x300F00a0)
```

5.2.37 $PSTMSQIERASE

This NMEA command erases the specified sector (64kbytes wide) of the SQI Data Storage Area.

Synopsis:

```
$PSTMSQIERASE<cr><lf>
```

Arguments:

None.

Results:

- if success the following message is sent

```
$PSTMSQIERASEOK<cr><lf>
```

- in case of error the following message will be sent

```
$PSTMSQIERASEERROR<cr><lf>
```

Example:

```
$PSTMSQIERASE
fhe following NMEA command erases all the information inside the SQI Data
Storage Area (from 0x300F0000 to 0x300FFFFF)
```

5.3 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.

5.3.1 $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>
```

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)
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.

5.3.2 $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.

5.3.3 $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
```

5.3.4 $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
```

6 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.

6.1 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

6.2 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
$PSTMKFCOV	OFF	ST: Standard Deviation and Covariance
$PSTMAGPS[9]	OFF	ST: STAGPS predicted ephemeris information
$PSTMNOTCHSTATUS[10]	ON	ST: Reports the Notch filter status.
$PSTMCPU	ON	ST: Reports the CPU usage and CPU speed setting.
$PSTMPOSNHOLD	ON	ST: Reports the status and position of Position Hold.

6.3 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

[9] This message is available only if the STAGPS is supported.

[10] This message is automatically ON if the notch filter is enabled.

6.4 Standard NMEA messages specification

This messages are defined within the “NMEA 0183” Specification.

6.4.1 $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
```

6.4.2 $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, 6 digits	Height above WGS84 Elipsoid, max: 100000m
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
```

6.4.3 $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:

6.4.4 $--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
```

6.4.5 $--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>
```

N max 4

Parameter	Format	Description
GSVAmount	Decimal, 1 digit	Total amount of GSV messages, max. 8
GSVNumber	Decimal, 1 digit	Continued GSV number of this message
TotSats	Decimal, 2 digits	Total Number of Satellites in view, max. 32
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.

6.4.6 $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
```

6.4.7 $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:

6.5 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.

6.5.1 $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>
```

N max 3

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
```

6.5.2 $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

6.5.3 $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

6.5.4 $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

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,
```

6.5.5 $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
```

6.5.6 $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:

6.5.7 $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
```

6.5.8 $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
```

6.5.9 $PSTMCPU

This message contains the real time CPU usage and the CPU speed setting.

NMEA message list bitmask: 0x800000

Format:

```
$PSTMCPU,<CPU_Usage>,<PLL_ON_OFF>,<CPU_Speed>*<checksum><cr><lf>
```

Parameter	Format	Description
CPU_Usage	ddd.dd	CPU usage %
PLL_ON_OFF	Decimal, 1 digit	PLL enabling/disabling status: 0: PLL disabled 1: PLL enabled
CPU_Speed	Decimal, 1 digit	CPU clock frequency: 52, 104, 156, 208 MHz.

6.5.10 $PSTMPOSHOLD

Reports the Position Hold status and position.

NMEA message list bitmask: 0x4000000

Format:

```
$PSTMPOSHOLD,<on_off>,<Lat>,<N/S>,<Long>,<E/W>,<Alt>*<checksum><cr><lf>
```

Parameter	Format	Description
On_off	Decimal, 1 digit	Position Hold enabling/disabling status 0: disabled 1: enabled
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
Alt	Decimal, 8 digits	Height above WGS84 Elipsoid, max: 100000

6.5.11 $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 m² (from Kalman Filter)
PosEcov	ddd.d Decimal, 4 digit	Covariance (East/West) in m² (from Kalman Filter)
PosVcov	ddd.d Decimal, 4 digit	Covariance (Vertical) in m² (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 m²/s (from Kalman Filter)
VelEcov	ddd.d Decimal, 4 digit	Covariance (East/West) in m²/s (from Kalman Filter)
VelVcov	ddd.d Decimal, 4 digit	Covariance (Vertical) in m²/s (from Kalman Filter)

Example:

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

6.5.12 $PSTMAGPS[11]

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)

[11] This message is supported only if the STAGPS is supported by the used platform.

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
```

6.5.13 $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:

6.5.14 $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:

6.5.15 $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
```

6.5.16 $PSTMNOTCHSTATUS

Reports the Notch filter status.

NMEA message list bitmask: 0x40000000

Format:

```
$PSTMNOTCHSTATUS,<kfreq_now_Hz_gps>,<lock_en_gps>,<pwr_gps>,
<ovfs_gps>,<mode_gps>,<kfreq_now_Hz_gln>,<lock_en_gln>,<pwr_gln>,
<ovfs_gln>,<mode_gln><cr><lf>
```

Parameter	Format	Description
kfreq_now_Hz_gps	Decimal, 7 digits	Notch frequency estimation actual value [Hz] (GPS path)
lock_en_gps	Decimal, 1 digits	Frequency lock flag (GPS path)
pwr_gps	Decimal, 4 digits	Band Pass Filter power estimation (GPS path)
ovfs_gps	Decimal, 1 digits	Notch overflows flag (GPS path)
mode_gps	Decimal, 1 digits	ANF mode operation (GPS path) [1 → Always ON; 2 → Auto insertion mode; 0 → ANF disabled;]
kfreq_now_Hz_gln	Decimal, 7 digits	Notch frequency estimation actual value [Hz] (GLONASS path)
lock_en_gln	Decimal, 1 digits	Frequency lock flag (GLONASS path)
pwr_gln	Decimal, 4 digits	Band Pass Filter power estimation (GLONASS path)
ovfs_gln	Decimal, 1 digits	Notch overflows flag (GLONASS path)
mode_gln	Decimal, 1 digits	ANF mode operation (GLONASS path) [1 → Always ON; 2 → Auto insertion mode; 0 → ANF disabled;]

Example:

```
$PSTMNOTCHSTATUS,3672980,0,1463,0,2,6474453,0,2469,0,2*5B
```

6.6 Commands answers messages specification

6.6.1 $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,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034
024200b4ffff00*1a
```

6.6.2 $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,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400
fbff33420000ee632f27698ef001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ff
c616fe03*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,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00
ecff212c00000ced2b287d1021031f5b0da1b0eabad3c9277301316763b9f9001100
9184c003*59
$PSTMEPHEM,2,64,4205706270629898984 1f60034a3ff0017014e23c90ad20095ff
feff40360000e59fd126b3f39f04ddda0ca160ecc10ed28dacaa512bc74edb000300
e21eff03*09
$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5ff
f0ff5b36000089e92c26d3a6700364ca0da109f24862068422525c188929f700f201
032bc703*5b
$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00
200053370000a4b113261c5b240333740da1b1d91e956051cf7e3f6ed4b3f6000400
6fa5db03*00
$PSTMEPHEM,14,64,420570627062c5c5c5e10e007ea9ff0064058520a30ea604160
00200772c000024c01b28451e1f01c49f0ca10aeb5ff83bcf570002bc35acec00040
0a632ff03*6b
$PSTMEPHEM,21,64,4205706270622121 2188f9009da5ff00e7004622cd0aba00d9f
f9efffd3500001a618a2634ba500506010ea1e9f9fa926c745cac2cc31f84e700200
044a6c403*3c
$PSTMEPHEM,25,64,42056c626c62b2b2b20c04008ca5ff0007fc3b250b0820fd5b0
0290079370000ada6bd26d78f350664e90ca176ebc4a6c5e0fd26c93f03c6f000070
04d12c003*3d
$PSTMEPHEM,30,64,420570627062b0b0b091f800caa6ff00cff8e2179e1355f999f
fc0ff553500003f077326f97e6c04c8140da10c14be42db05f853b7a66b34ef005e0
09ff7cd03*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,1,32,0142056314325b1c5efd0140020da14009730160ad61b900ca
ffe12011088020*1d
$PSTMALMANAC,2,32,02420563034ab50634fd01406c0ca1402eacaa6047c64e005b
741c20e4078020*15
$PSTMALMANAC,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f19000a
3ef92030088020*1c
```

```
$PSTMALMANAC,4,32,04420563f93a700633fd0140450da140447bab606fd202008e
c97f201e208020*1a
$PSTMALMANAC,5,32,054205630d3765fc3ffd0140500da14033225260f08929006c
f96f20e6808020*19
$PSTMALMANAC,6,32,064205634532d6fa3ffd0140fc0ca14018cf7e600cd4b30037
d0a22075038020*49
$PSTMALMANAC,7,32,07420563f56cd9fb3ffd0140d20da1402eb77d6082d2b7003b
dcfa2099218020*13
$PSTMALMANAC,8,32,08420563ee4e011242fd0140190da14072452c609b4a6900fb
e2a620d0078020*1b
$PSTMALMANAC,9,32,09420563588ed00938fd0140cf0ca1406728296083eb3000c2
729720f1078020*44
$PSTMALMANAC,10,32,0a420563ed35ee155ffd0140ac0da140f82cd6609c7a0e004
eb22a204c008020*76
$PSTMALMANAC,11,32,0b420563fc2632e406fd0140fc0ca1403c39a56064700a006
08bbe2023098020*7b
$PSTMALMANAC,13,32,0d4205632315171f64fd0140ca0ca140d1d4006012ed2d00d
0a1242016088020*2c
$PSTMALMANAC,14,32,0e420563f711581b5efd0140480ca140b2570060bd35ac002
a110620e6078020*20
$PSTMALMANAC,15,32,0f420563f14a070b3bfd0140780ba1400dc3ad60b14366000
ce9a92017128020*2f
$PSTMALMANAC,16,32,10420563c917770c58fd0140550ca140199f55601c2bd800a
2196b200d008020*24
$PSTMALMANAC,17,32,114205630c0d1a0c54fd0140430ca140aeef7f6043406d000
8044920c427c020*79
$PSTMALMANAC,18,32,12420563c0367d0b50fd0140b30ca140c130d76094349100f
755672031ffbf20*25
$PSTMALMANAC,19,32,13420563b01ad60a51fd01409a0da140d1628260fc19c500a
7d23520e4078020*72
$PSTMALMANAC,20,32,14420563e0133f0b4efd0140830ca140db0ad560ed613a00a
1365a20d3078020*7c
$PSTMALMANAC,21,32,154205630955410230fd0140880da1400d5cac60921f84007
faca02095088020*29
$PSTMALMANAC,22,32,164205631029da094efd0140140da140808ad7608e4abf00d
bfc212032088020*27
$PSTMALMANAC,23,32,174205630f23bf0f51fd0140a50ca140a0f0ff60905c61001
72d0720aff8bf20*7d
$PSTMALMANAC,24,32,184205634b4a1f0d3ffd01404d0da1400ec6ac604db9d4000
6aac7203c088020*2c
$PSTMALMANAC,25,32,19420563596376052ffd0140760ca1408bfd26603c01c600e
9d9b42002008020*28
$PSTMALMANAC,26,32,1a420563fd87eb1d61fd0140bc0ca140e5e2006013041e001
389e320f7ffbf20*22
$PSTMALMANAC,27,32,1b4205630e9e660834fd0140720da140313f28606565ae002
a2d772016008020*7b
$PSTMALMANAC,28,32,1c4205631756300b57fd0000dc0ca1402f06562082c6a1205
0f344002a008000*25
```

```
$PSTMALMANAC,29,32,1d4205638f49d21b60fd0140090da1407880ff60c018d5000
095352095298020*73
$PSTMALMANAC,30,32,1e420563ca46c70045fd0140a00ca140baf75360466c3400e
26e5020bf198020*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.

8 Summary of text files used in the examples

8.1 File: SUSPEND.txt

```
$PSTMSUSPEND
```

8.2 File: RESUME.txt

```
$PSTMRESUME
```

8.3 File: DUMPEPHEMS.txt

```
$PSTMNMEAONOFF
$PSTMDUMPALMANAC
```

8.4 File: DUMPALMANAC.txt

```
$PSTMNMEAONOFF
$PSTMDUMPALMANAC
```

8.5 File: LOADALMANAC.txt

```
$PSTMLOADALM
$PSTMALMANAC,1,32,0142056314325b1c5efd0140020da14009730160ad61b900ca
ffe12011088020*1d
$PSTMALMANAC,2,32,02420563034ab50634fd01406c0ca1402eacaa6047c64e005b
741c20e4078020*15
$PSTMALMANAC,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f19000a
3ef92030088020*1c
$PSTMALMANAC,4,32,04420563f93a700633fd0140450da140447bab606fd202008e
c97f201e208020*1a
$PSTMALMANAC,5,32,054205630d3765fc3ffd0140500da14033225260f08929006c
f96f20e6808020*19
$PSTMALMANAC,6,32,064205634532d6fa3ffd0140fc0ca14018cf7e600cd4b30037
d0a22075038020*49
$PSTMALMANAC,7,32,07420563f56cd9fb3ffd0140d20da1402eb77d6082d2b7003b
dcfa2099218020*13
$PSTMALMANAC,8,32,08420563ee4e011242fd0140190da14072452c609b4a6900fb
e2a620d0078020*1b
$PSTMALMANAC,9,32,09420563588ed00938fd0140cf0ca1406728296083eb3000c2
729720f1078020*44
$PSTMALMANAC,10,32,0a420563ed35ee155ffd0140ac0da140f82cd6609c7a0e004
eb22a204c008020*76
$PSTMALMANAC,11,32,0b420563fc2632e406fd0140fc0ca1403c39a56064700a006
08bbe2023098020*7b
$PSTMALMANAC,13,32,0d4205632315171f64fd0140ca0ca140d1d4006012ed2d00d
0a1242016088020*2c
```

```
$PSTMALMANAC,14,32,0e420563f711581b5efd0140480ca140b2570060bd35ac002
a110620e6078020*20
$PSTMALMANAC,15,32,0f420563f14a070b3bfd0140780ba1400dc3ad60b14366000
ce9a92017128020*2f
$PSTMALMANAC,16,32,10420563c917770c58fd0140550ca140199f55601c2bd800a
2196b200d008020*24$PSTMALMANAC,17,32,114205630c0d1a0c54fd0140430ca14
0aeef7f6043406d0008044920c427c020*79$PSTMALMANAC,18,32,12420563c0367
d0b50fd0140b30ca140c130d76094349100f755672031ffbf20*25$PSTMALMANAC,1
9,32,13420563b01ad60a51fd01409a0da140d1628260fc19c500a7d23520e407802
0*72
$PSTMALMANAC,20,32,14420563e0133f0b4efd0140830ca140db0ad560ed613a00a
1365a20d3078020*7c
$PSTMALMANAC,21,32,154205630955410230fd0140880da1400d5cac60921f84007
faca02095088020*29
$PSTMALMANAC,22,32,164205631029da094efd0140140da140808ad7608e4abf00d
bfc212032088020*27
$PSTMALMANAC,23,32,174205630f23bf0f51fd0140a50ca140a0f0ff60905c61001
72d0720aff8bf20*7d
$PSTMALMANAC,24,32,184205634b4a1f0d3ffd01404d0da1400ec6ac604db9d4000
6aac7203c088020*2c
$PSTMALMANAC,25,32,19420563596376052ffd0140760ca1408bfd26603c01c600e
9d9b42002008020*28
$PSTMALMANAC,26,32,1a420563fd87eb1d61fd0140bc0ca140e5e2006013041e001
389e320f7ffbf20*22
$PSTMALMANAC,27,32,1b4205630e9e660834fd0140720da140313f28606565ae002
a2d772016008020*7b
$PSTMALMANAC,28,32,1c4205631756300b57fd0000dc0ca1402f06562082c6a1205
0f344002a008000*25
$PSTMALMANAC,29,32,1d4205638f49d21b60fd0140090da1407880ff60c018d5000
095352095298020*73
$PSTMALMANAC,30,32,1e420563ca46c70045fd0140a00ca140baf75360466c3400e
26e5020bf198020*28
```

8.6 File: LOADEPHEMS.txt

```
$PSTMLOADEPHEMS
$PSTMEPHEM,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00
ecff212c00000ced2b287d1021031f5b0da1b0eabad3c9277301316763b9f9001100
9184c003*59
$PSTMEPHEM,2,64,42057062706298989841f60034a3ff0017014e23c90ad20095ff
feff40360000e59fd126b3f39f04ddda0ca160ecc10ed28dacaa512bc74edb000300
e21eff03*09
$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5ff
f0ff5b36000089e92c26d3a6700364ca0da109f24862068422525c188929f700f201
032bc703*5b
$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00
200053370000a4b113261c5b240333740da1b1d91e956051cf7e3f6ed4b3f6000400
6fa5db03*00
```

```
$PSTMEPHEM,14,64,420570627062c5c5c5e10e007ea9ff0064058520a30ea604160
00200772c000024c01b28451e1f01c49f0ca10aeb5ff83bcf570002bc35acec00040
0a632ff03*6b
$PSTMEPHEM,21,64,4205706270622121212188f9009da5ff00e7004622cd0aba00d9f
f9efffd3500001a618a2634ba500506010ea1e9f9fa926c745cac2cc31f84e700200
044a6c403*3c
$PSTMEPHEM,25,64,42056c626c62b2b2b20c04008ca5ff0007fc3b250b0820fd5b0
0290079370000ada6bd26d78f350664e90ca176ebc4a6c5e0fd26c93f03c6f000070
04d12c003*3d
$PSTMEPHEM,30,64,420570627062b0b0b091f800caa6ff00cff8e2179e1355f999f
fc0ff553500003f077326f97e6c04c8140da10c14be42db05f853b7a66b34ef005e0
09ff7cd03*3e
```

9 Disclaimer

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

http://www.st.com