life.augmented

UM3407

User manual

Teseo VI and Teseo APP2–NMEA specifications and commands

Introduction

The purpose of this document is to provide an overview of the various NMEA commands and messages for the STMicroelectronics´ GNSS systems. This document is relevant for the following baseband processors and related GNSS software products. Any other specific constraints related to version of products and software are specified inside the document.

1 Communication interface

Communication between a host processor and the ST GNSS system can be established in different ways, depending on the implementation of the baseband processor as a standalone unit or as an integrated subsystem on a “System on chip”.

For simplicity reasons this document refers to “standalone processors” only and the interface described in the examples is a UART.

All information contained in this document is related to the “NMEA port” of the baseband processor. STMicroelectronics GNSS systems may contain an additional “Debug port” but the data exchanged on the “Debug port” is not within the scope of this document.

1.1 Commands

A command is a defined data packet which is sent from a host processor to the GNSS-baseband controller to control the GNSS system behavior. The regular structure of a command is:

```
command-ID,<parameters>*<checksum><cr><lf>
```

In order to receive the commands, the GNSS receiver is connected to the host processor via the NMEA port. The user interaction can be achieved through the use of the Teseo-suite (the STMicroelectronics PC Windows program available on st.com) that is connected to the appropriate COM port with settings in the table below.

Table 1. Default UART configuration

Baud rate	Parity bits	Stop bit	Data bits	Hardware flow control
3000000	0	1	8	Enabled

The default NMEA baud rate value is automatically set at the system startup. It can be modified at system runtime using the appropriate command.

Once the command is executed the device replies with messages according to what specified in this document, after the message the command is sent back to the host as final confirmation of the execution. This functionality can be configured according to what specified in the firmware configuration document.

Note: That checksum can be used in a command. It is not mandatory as the GNSS receiver can manage a command with or without checksum.

Leading zero is mandatory if more than 1 digit is expected in a command field.

1.2 Messages

A message is a defined set of data sent from the GNSS 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 runtime. The basic structure of a message is:

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

1.2.1 Standard NMEA messages

Standard NMEA messages are defined in the “NMEA 0183” standard, issued from the “National marine electronics association”.

By default, standard NMEA messages are compliant with the “NMEA 0183” standard Rev. 4.11 dated November 2018.

To get an overview of the standard NMEA messages supported by ST´s GNSS systems refer to Section 3.1: Standard NMEA messages list in this document.

1.2.2 GNSS talker ID's

Standard NMEA messages start with the "message-ID" with:

```
$<TalkerID>
```

Supported talker IDs are returned in the table below according to the NMEA revision:

Table 2. GNSS talker IDs

Positioning system	"NMEA 0183" standard Rev. 4.11
Galileo	GA
BeiDou	GB
NavIC (IRNSS)	GI
GLONASS	GL
Global navigation satellite system (GNSS)	GN
GPS	GP
QZSS	GQ

1.2.3 Proprietary messages

The STMicroelectronics Teseo GNSS system can provide additional messages with more detailed data content. This is required to transmit GNSS and system information content which is not defined in the NMEA standard output.

Proprietary messages from STMicroelectronics start with:

```
$PSTM...
```

To get an overview of the main proprietary messages defined by STMicroelectronics refer to Section 3.2: ST NMEA messages list in this document.

Note: Other proprietary messages not defined in the list should not be considered being for ST internal usage only.

2 Commands

2.1 Software command list

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

Table 3. Commands

Syntax	Description
$PSTMINITGPS	Initialize GPS position and time using UTC format
$PSTMINITTIME	Initialize GPS time using the UTC format
$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
$PSTMSRR	Perform a reset of the GNSS receiver
$PSTMWARM	Perform WARM start
$PSTMHOT	Perform HOT start
$PSTMFORCESTANDBY	Force system in standby
$PSTMHWSTANDBY	Configure hardware standby
$PSTMGPSRESET	Reset the GPS engine
$PSTMGPSSUSPEND	Suspend GPS engine
$PSTMGPSRESTART	Restart GPS engine
$PSTMENABLEPOSITIONHOLD	Set status and position for the position hold feature.
$PSTMPPS	Command interface for pulse per second management.
$PSTMDUMPIONO	Get the iono packets for each constellation
$PSTMIONOPARAMS	Save iono packet in NVM
$PSTMFEDUMP	Get front-end register’s value
$PSTMDUMPGLONASSICB	Get the current values of the GLONASS ICB tables
$PSTMGLONASSICB	Save a specific GLONASS ICB table in NVM
$PSTMDUMPIFB	Get the current IFB values
$PSTMSETCONSTMASK	Set the GNSS constellation mask
$PSTMIFB	Save a set of custom IFB(s) in NVM
$PSTMDUMPCHIPINFO	Dump the CHIP ID, BCS version, DCF version and DCF value
$PSTMTRKJAMMER	Test jammer presence
$PSTMGETSWVER	Provide the GPS library version string
$PSTMNVMSWAP (1)	Execute a bank swap on the NVM GPS backup memory
$PSTMSETTHTRK	Configures the CN0 and Angle elevation mask thresholds for tracking
$PSTMSETTHPOS	Configures the CN0 and angle elevation mask thresholds for positioning
$PSTMGETRTCTIME	Get the current RTC time

Syntax	Description
$PSTMNMA	OSNMA parameters management
$PSTMNMAPK	Set PEM encoded OSNMA ECDSA public key at the specified index
$PSTMNMASETMTR	Set Merkle tree root
$PSTMNMAGETMTR	Get Merkle tree root
$PSTMNMACLRMTR	Clear Merkle tree root
$PSTMNMASETITN	Set intermediate Merkle tree node
$PSTMNMAGETITN	Get intermediate Merkle tree node
$PSTMNMACLRITN	Clear intermediate Merkle tree node
$PSTMNMACLRAM	Clear OSNMA alert message state
$PSTMNMACLRKROOT	Clear OSNMA root key

1. *This command is supported only by platforms or system configurations where the GNSS backup memory is based on flash NOR or SQI memories.*

2.2 NMEA commands

2.2.1 $PSTMINITGPS

Initialize GPS position and time using the UTC format. This command must be issued after a cold reset or it fails. The date issued with parameters day, month and year must be later than January 2020, this threshold can be changed using the configuration options (see UM3428). It is necessary to perform a system reset or suspend/restart to apply the position.

Synopsis:

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

Arguments:

Table 4. $PSTMINITGPS field description

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-miinute.minute decimals)
LonRef	‘E’ or ‘W’	Longitude direction (east or west)
Alt	dddddd – Decimal, 6 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 (2020 - …)
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
- The following message is output on NMEA communication channel:

```
$PSTMINITGPSOK*<checksum><cr><lf>  if success
$PSTMINITGPSERROR*<checksum><cr><lf>  if no success
```

Example:

```
$PSTMINITGPS,4811.365,N,01154.123,E,0530,23,02,2020,09,44,12
```

2.2.2 $PSTMINITTIME

Initialize GPS time using the UTC format. The date issued with parameters day, month and year must be later than January 2020, this threshold can be changed using the configuration options (see UM3428).

Synopsis:

```
$PSTMINITTIME,<Day>,<Month>,<Year>,<Hour>,<Minute>,<Second>*<checksum><cr><lf>
```

Arguments:

Table 5. $PSTMINITTIME field description

Parameter	Format	Description
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 (2020 - …)
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
- The following message is output on NMEA communication channel:

```
$PSTMINITTIMEOK*<checksum><cr><lf> if success
$PSTMINITTIMEERROR*<checksum><cr><lf> if no success
```

Example:

```
$PSTMINITTIME, 23, 02, 2020, 09, 44, 12
```

2.2.3 $PSTMINITFRQ

Sets the initial search NCO frequency. The frequency error is the error perceived on a satellite, either generated by doppler shift on the moving satellite, or the inherent oscillator error. The GNSS uses this error to determine where to start searching for the first satellite.

The frequency needs only to be set to the nearest 1 kHz since the GNSS makes wide searches (up to 9 kHz) when looking for the first satellite. An accurate measure of the centre frequency gives a short time to first fix.

If no frequency is set the GNSS uses the previous local oscillator offset stored in the backup memory. This is the normal case.

The centre frequency should only need to be set in exceptional circumstances for example, if backup memory is lost.

In case of successful operation, the new frequency is stored in the backup memory.

Synopsis:

```
$PSTMINITFRQ,<Frequency>*<checksum><cr><lf>
```

Arguments:

Table 6. $PSTMINITFRQ field description

Parameter	Format	Description
Frequency	Decimal, 6 digits	Day of month (-132000 to 132000)

Results:

- The NCO frequency is set.
- No message is sent as a reply.

Example:

```
$PSTMINITFRQ, 10*<checksum><cr><lf>
```

2.2.4 $PSTMCLREPHS

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

Synopsis:

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

Arguments:

None.

Results:

- All ephemeris, stored in the nonvolatile backup memory (either backup SRAM or flash), will be deleted.
- No message is sent as a reply.

Example:

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

2.2.5 $PSTMDUMPEPHEMS

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

Synopsis:

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

Arguments:

None.

Results:

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

Where:

Table 7. $PSTMDUMPEPHEMS field description

Parameter	Format	Description
sat_id	Decimal, 2 digits	Satellite number
N	Decimal, 1 digit	Number of the ephemeris data bytes
byte1	Hexadecimal, 2 digits	First byte of the ephemeris data
byteN	Hexadecimal, 2 digits	The last byte of the ephemeris data

The N bytes that are in the message are the dump of a structure that contain all the information of the ephemeris. Data structures per constellation are described in UM3397.

Example:

```
$PSTMDUMPEPHEMS<cr><lf>
$PSTMEPHEM,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400fbff33420000ee632f27698ef
001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ffc616fe03*<checksum>
$PSTMEPHEM,2,64,0f06bc34bc344f4f4f78110019a5ff00b004fa1d1e0e3f04c8ffcaff1937000033515726556ba
9048eae0da1b6c346bd8f985c93ade10c76db001d00f8c7c503*<checksum>
$PSTMEPHEM,4,64,0f06bb34bb344b4b4b98050038a4ff000005351e110eea041b00b8ffd037000020b84e26b5138
b0425580ca16b211030e68b1a949cac9615f30066ffea92f603*<checksum>
$PSTMEPHEM,9,64,0f06bc34bc341818189c0a0069aaff005f06eb249a09ca0477ff6c00f72e00005131d827592b9
50a91010da1c7af88538e7ca1122fb9be3df4001300c4a0c203*<checksum>
```

2.2.6 $PSTMEPHEM

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

To update the ephemeris, the following steps are recommended:

- Clear all ephemeris (for example $PSTMCLREPHS)
- Suspend GNSS app (for example $PSTMGPSSUSPEND)
- Update the ephemeris (for example $PSTMEPHEM)
- Restart GNSS app (for example $PSTMGPSRESTART)

Synopsis:

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

Arguments:

Table 8. $PSTMEPHEM field description

Parameter	Format	Description
sat_id	Decimal, 2 digits	Satellite number
N	Decimal, 1 digit	Number of the ephemeris data bytes
Byte1	Hexadecimal, 2 digits	First byte of the ephemeris data
ByteN	Hexadecimal, 2 digits	The last byte of the ephemeris data

The N bytes that are in the parameters are the dump of a structure that contain all the information of the ephemeris. Data structures per constellation are described in UM3397.

Results:

- The ephemeris is stored into backup RAM.
- No message is sent as a reply.

Example:

```
$PSTMEPHEM,12,64,0f06bc34bc3437373790f40045a7ff00fcf5d522480b4bf71b00fbff8931000096126f271f869101c3870ca107afce79a763e13e360a1ce8e7003100380ff903*36
```

2.2.7 $PSTMCLRALMS

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

Synopsis:

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

Arguments:

None.

Results:

- All almanacs, stored in the nonvolatile backup memory, will be deleted.
- No message is sent as a reply.

Example:

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

2.2.8 $PSTMDUMPALMANAC

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

Synopsis:

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

Arguments:

None.

Results:

```
$PSTMDUMPALMANAC,<sat_id>,<N>,<byte1>…<byteN>*<checksum><cr><lf>
```

Where:

Table 9. $PSTMDUMPALMANAC field description

Parameter	Format	Description
sat_id	Decimal, 2 digits	Satellite number
N	Decimal, 1 digit	Number of the almanac data bytes
byte1	Hexadecimal, 2 digits	First byte of the almanac data
byteN	Hexadecimal, 2 digits	Last byte of the almanac data

The N bytes that are in the message are the dump of a structure that contains all the information of the almanac. Data are stored in this structure according to the following tables for each constellations:

Table 10. GPS almanac field description

Bits	Structure member	Description
8	satid	The satellite number
16	week	The week number for the epoch
8	toa	Reference time almanac
16	eccentricity	Eccentricity
16	delta_i	Rate of inclination angle
16	omega_dot	Rate of right ascension
16	spare0	
24	root_A	Square root of semimajor axis
8	spare1	
24	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
8	spare2	
24	perigee	Argument of perigee
8	spare3	
24	mean_anomaly	Mean anomaly at reference time
8	spare4	
11	af0	Constant clock correction
11	af1	First order clock correction
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
8	spare5	
32	spare6	
32	spare7	

Table 11. GLONASS almanac field description

Bits	Structure member	Description
8	satid	The satellite number

Bits	Structure member	Description
16	week	The week number for the epoch
8	spare0	
20	toa	Reference time almanac
5	n_A	Slot number (1…24)
5	H_n_A	Carrier frequency channel number
2	M_n_A	Type of satellite: 00 = GLONASS 01 = GLONASS-M
10	tau_n_A	Satellite clock correction
15	epsilon_n_A	Eccentricity
7	spare1	
21	t_lambda_n_A	Time of the first ascending node passage
11	spare2	
21	lambda_n_A	Longitude of ascending node of orbit plane at almanac epoch
11	spare3	
18	delta_i_n_A	Inclination angle correction to nominal value
7	delta_T_n_dot_A	Draconian period rate of change
7	spare4	
22	delta_T_n_A	Draconian period correction
10	spare5	
16	omega_n_A	Argument of perigee
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
14	spare6	
32	tau_c	GLONASS to UTC(SU) time correction
11	NA	Calendar day number within the four-year period since the beginning of last leap year (almanac)
5	N4	Four-year interval number starting from 1996
16	spare7	

Table 12. Galileo almanac field description

Bits	Structure member	Description
16	satid	The satellite number
6	svid	Space vehicle identificator
10	spare0	
16	week	The week number for the epoch
16	spare1	
20	toa	Reference time almanac
12	spare2	
13	delta_a	Delta of semimajor axis
11	e	Eccentricity
8	spare3	

Bits	Structure member	Description
16	perigee	Argument of perigee
11	delta_i	Rate of inclination angle
5	spare4	
16	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
11	omega_dot	Rate of right ascension
5	spare5	
16	mean_anomaly	Mean anomaly at reference time
16	af0	Constant clock correction
13	af1	First order clock correction
2	E5b_HS	E5 signal health status
2	E1B_HS	E1-B signal health status
4	ioda_1	Issue of data almanac 1
4	ioda_2	Issue of data almanac 2
2	word_available	Contains 1 if ephemeris is available, 0 if not
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
3	spare6	Contains 1 if the satellite is unhealthy, 0 if healthy
32	spare7	
32	spare8	

Table 13. BeiDou almanac field description

Bits	Structure member	Description
8	prn	PRN number of the corresponding almanac data
16	week	Almanac reference week number
8	toa	Almanac reference time
17	eccentricity	Eccentricity
11	af0	Satellite clock time bias correction coefficient
1	is_geo	Satellite orbit type
1	WNa_valid	
2	spare0	
17	omega_dot	Rate of right ascension
11	af1	Satellite clock time drift correction coefficient
4	spare1	
24	root_a	Square root of semimajor axis
8	spare2	
24	omega_zero	Longitude of ascending node of orbital plane at weekly epoch
8	spare3	
24	perigee	Argument of perigee
8	spare4	
24	mean_anomaly	Mean anomaly at reference time
8	spare5	

Bits	Structure member	Description
16	delta_i	Correction of inclination angle at reference time
1	health	Satellite health information
1	available	Contains 1 if almanac is available 0 if not
8	last_received_toa	
6	spare6	

Table 14. IRNSS almanac field description

Bits	Structure member	Description
10	WNa	Week number for almanac
16	toa	Almanac reference time
6	prn_al	PRN ID for almanac
16	eccentricity	Eccentricity
16	omega_dot	Rate of right ascension
24	inclination	Inclination
8	ISC	Inter signal correction
24	root_a	Square root of the semi-major axis
8	spare0	
24	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
6	spare	
2	spare1	
24	perigee	Argument of perigee
6	prn	PRN ID
2	spare2	
24	mean_anomaly	Mean anomaly at reference time
8	spare3	
11	af0	Clock bias A0
11	af1	Clock drift A1
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
8	spare4	

Table 15. CNAV almanac field description

Bits	Structure member	Description
8	prn	The satellite number
16	week	The week number for the epoch
8	toa	Reference time almanac
11	eccentricity	Eccentricity
21	delta_i	Rate of inclination angle
11	omega_dot	Rate of right ascension
21	spare0	
17	root_a	Square root of semimajor axis

Bits	Structure member	Description
15	spare1	
16	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
16	spare2	
16	perigee	Argument of perigee
16	spare3	
16	Mean_anomaly	Mean anomaly at reference time
16	spare4	
11	af0	Constant clock correction
10	af1	First order clock correction
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
9	spare5	
32	spare6	
32	spare7	

Example:

```
$PSTMDUMPALMANAC<cr><lf> $PSTMALMANAC,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034
024200b4ffff00*<checksum>
$PSTMALMANAC,2,32,021a0690944b78fe37fd0800770da141ef0c5b0060487700989bd800d8088000*<checksum>
$PSTMALMANAC,3,32,031a06904f68a2f540fd0800f60ca141922a2c003cae27009496cf00020a8000*<checksum>

$PSTMALMANAC,4,32,041a0690a94aeffd36fd0800390ca141afc95b00de7a1700dfc74e004ddebf00*<checksum>
$PSTMALMANAC,5,32,051a0690940eee0b5efd0800900ca141582b8600d3000b0060641200e40f8000*<checksum>
```

2.2.9 $PSTMALMANAC

Save almanacs data. This command allows the user to save the almanacs data into backup memory.

Synopsis:

```
$PSTMALMANAC,<sat_id>,<N>,<byte1>…<byteN>*<checksum><cr><lf>
```

Arguments:

Table 16. $PSTMALMANAC field description

Parameter	Format	Description
sat_id	Decimal, 2 digits	Satellite number
N	Decimal, 1 digit	Number of the almanac data bytes
byte1	Hexadecimal, 2 digits	First byte of the almanac data
byteN	Hexadecimal, 2 digits	Last byte of the almanac data

The N bytes that are in the message are the dump of a structure that contains all the information of the almanac. Data are stored in this structure according to the following tables for each constellations:

Table 17. GPS Almanac field description

Bits	Structure member	Description
8	satid	The satellite number
16	week	The week number for the epoch
8	toa	Reference time almanac

Bits	Structure member	Description
16	eccentricity	Eccentricity
16	delta_i	Rate of inclination angle
16	omega_dot	Rate of right ascension
16	spare0	
24	root_A	Square root of semimajor axis
8	spare1	
24	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
8	spare2	
24	perigee	Argument of perigee
8	spare3	
24	mean_anomaly	Mean anomaly at reference time
8	spare4	
11	af0	Constant clock correction
11	af1	First order clock correction
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
8	spare5	
32	Spare6	
32	Spare7	

Table 18. GLONASS almanac field description

Bits	Structure member	Description
8	satid	The satellite number
16	week	The week number for the epoch
8	spare0	
20	toa	Reference time almanac
5	n_A	Slot number (1…24).
5	H_n_A	Carrier frequency channel number
2	M_n_A	Type of satellite 00=GLONASS 01=GLONASS-M
10	tau_n_A	Satellite clock correction
15	epsilon_n_A	Eccentricity
7	spare1	
21	t_lambda_n_A	Time of the first ascending node passage
11	Spare2	
21	lambda_n_A	Longitude of ascending node of orbit plane at almanac epoch
11	spare3	
18	delta_i_n_A	Inclination angle correction to nominal value
7	delta_T_n_dot_A	Draconian period rate of change
7	spare4	
22	delta_T_n_A	Draconian period correction
10	spare5	

Bits	Structure member	Description
16	omega_n_A	Argument of perigee
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
14	spare6	
32	tau_c	GLONASS to UTC(SU) time correction
11	NA	Calendar day number within the four-year period since the beginning of last leap year (almanac)
5	N4	Four-year interval number starting from 1996
16	Spare7	

Table 19. Galileo almanac field description

Bit	Structure member	Description
6	svid	Space Vehicle Identificator
10	Spare0	
16	satid	The satellite number
16	week	The week number for the epoch
16	Spare1	
20	toa	Reference time almanac
12	Spare2	
13	delta_a	Delta of semi-major axis
11	e	Eccentricity
8	Spare3	
16	perigee	Argument of perigee
11	delta_i	Rate of inclination angle
5	Spare4	
16	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
11	omega_dot	Rate of right ascension
5	Spare5	
16	mean_anomaly	Mean anomaly at reference time
16	af0	Constant clock correction
13	af1	First order clock correction
2	E5b_HS	E5 Signal health status
2	E1B_HS	E1-B Signal health status
4	ioda_1	Issue of data almanac 1
4	ioda_2	Issue of data almanac 2
2	word_available	Contains 1 if ephemeris is available, 0 if not
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
3	spare6	Contains 1 if the satellite is unhealthy, 0 if healthy
32	spare7	
32	spare8	

Table 20. **BeiDou almanac field description**

Bits	Structure member	Description
8	prn	PRN number of the corresponding almanac data
16	week	Almanac reference week number
8	toa	Almanac reference time
17	eccentricity	Eccentricity
11	af0	Satellite clock time bias correction coefficient
1	is_geo	Satellite orbit type
1	WNa_valid	
2	spare0	
17	omega_dot	Rate of right ascension
11	af1	Satellite clock time drift correction coefficient
4	spare1	
24	root_a	Square root of semimajor axis
8	spare2	
24	omega_zero	Longitude of ascending node of orbital plane at weekly epoch
8	spare3	
24	perigee	Argument of perigee
8	spare4	
24	mean_anomaly	Mean anomaly at reference time
8	spare5	
16	delta_i	Correction of inclination angle at reference time
1	health	Satellite health information
1	available	Contains 1 if almanac is available 0 if not
8	last_received_toa	
6	spare6	

Table 21. **IRNSS almanac field description**

Bits	Structure member	Description
10	WNa	Week number for almanac
16	toa	Almanac reference time
6	prn_al	PRN ID for almanac
16	eccentricity	Eccentricity
16	omega_dot	Rate of right ascension
24	inclination	Inclination
8	ISC	Inter signal correction
24	root_a	Square root of the semi-major axis
8	spare0	
24	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
6	spare	
2	spare1	
24	perigee	Argument of perigee

Bits	Structure member	Description
6	prn	PRN ID
2	spare2	
24	mean_anomaly	Mean anomaly at reference time
8	spare3	
11	af0	Clock bias A0
11	af1	Clock drift A1
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
8	spare4	

Table 22. CNAV almanac field description

Bits	Structure member	Description
8	prn	The satellite number
16	week	The week number for the epoch
8	toa	Reference time almanac
11	eccentricity	Eccentricity
21	delta_i	Rate of inclination angle
11	omega_dot	Rate of right ascension
21	spare0	
17	root_a	Square root of semimajor axis
15	spare1	
16	omega_zero	Longitude of ascending node of orbit plane at weekly epoch
16	spare2	
16	perigee	Argument of perigee
16	spare3	
16	Mean_anomaly	Mean anomaly at reference time
16	spare4	
11	af0	Constant clock correction
10	af1	First order clock correction
1	health	Contains 1 if the satellite is unhealthy 0 if healthy
1	available	Contains 1 if almanac is available 0 if not
9	spare5	
32	spare6	
32	spare7	

Results:

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

Example:

```
 $PSTMALMANAC,12,32,0c1a06907c1a971160fd0800fa0da141ae9f0600d912e90075669700490f8000*<checksum><cr><lf>
```

2.2.10 $PSTMCOLD

Perform a COLD start.

Synopsis:

```
$PSTMCOLD,<type>*<checksum><cr><lf>
```

Arguments:

Table 23. $PSTMCOLD field description

Parameter	Format	Description
type	Integer	Optional bitmask parameter to invalidate time, position, ephemeris and almanac, … 0x01 – Clear almanac 0x02 – Clear ephemeris 0x04 – Clear position 0x08 – Clear time 0x10 – Swap NVM 0x20 – Clear utc 0x40 – Clear iono 0x80 – Clear bias

Results:

- Coldstart initialization and system restart.

Note: The GPS engine will be reset. It is not a system reboot.

- If type parameter is used, only the selected GPS data is invalidated for this actual coldstart. Multiple selects are supported (for example, 13).
- If the type parameter is not used, default value is set from CDB.
- Type is a decimal value.

Example:

```
$PSTMCOLD,6*<checksum><cr><lf>
```

2.2.11 $PSTMWARM

Perform a WARM start.

Synopsis:

```
$PSTMWARM,<delta>*<checksum><cr><lf>
```

Arguments:

Table 24. $PSTMWARM field description

Parameter	Format	Description
delta	Integer	Optional parameter to add delta/1000 seconds to the GNSS time reference.

Results:

- Warm start initialization and system restart.

Note: The GPS engine is reset. It is not a system reboot.

Example:

```
$PSTMWARM, 1000*<checksum><cr><lf>
```

2.2.12 $PSTMHOT

Perform a HOT start.

Synopsis:

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

Arguments:

None.

Results:

- The system restarted start.

Note: The GPS engine is reset. It is not a system reboot.

Example:

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

2.2.13 $PSTMSRR

Executes a system reset. The GNSS firmware is rebooted.

Synopsis:

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

Arguments:

None.

Results:

- The GNSS firmware reboots.
- No message is sent as a reply.

Example:

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

2.2.14 $PSTMFORCESTANDBY

Forces system in standby after setting the wake-up parameters configuration.

Synopsis:

```
$PSTMFORCESTANDBY,<duration>,<wakeup_logic>,<pullUpDownDeactivated>,<sensor_interrupt>*<checksum><cr><lf>
```

Arguments:

Table 25. $PSTMFORCESTANDBY field description

Parameter	Format	Description
Duration	Decimal	RTC counter time out (second) to generate internal wake-up. If 0, wake-up by RTC is ignored.
wakeup_logic	Decimal, 1 digit	Logic of the STANDBY_IN pin. If 0, a high to low level transition wakes up the system. If 1, low to high level transition wakes up the system. Other values return an error.
pullUpDownDeactivated	Decimal, 1 digit	Deactivation of pull-up and pull-down. If 0, pull-up and pull-down are automatically set according to wakeup_logic parameter. If 1, pull-up and pull-down both deactivated.
sensor_interrupt	Decimal, 1 digit	If 0, deactivation of the sensor interrupt. If 1, activation of the sensor interrupts to act as a wake-up in case of move.

Parameter	Format	Description
		Other values return an error.

Results:

```
$PSTMFORCESTANDBYOK*<checksum><cr><lf>  if success
$PSTMFORCESTANDBYERROR*<checksum><cr><lf>  if no success
```

If success message and no wake-up condition are verified, the GNSS firmware is in standby mode.

Example:

```
$PSTMFORCESTANDBY,60,1,1,1*<checksum><cr><lf>
```

2.2.15 $PSTMHWSTANDBY

Set the configuration of hardware standby pin.

Synopsis:

```
$PSTMHWSTANDBY,<enable_hw_standby>,<enable_pull_down>,<enable_pull_up>*<checksum><cr><lf>
```

Arguments:

Table 26. $PSTMHWSTANDBY field description

Parameter	Format	Description
enable_hw_standby	Decimal, 1 digit	If 0, disable hardware standby. Standby pin is not used with no pull-up and no pull-down. If 1, enable hardware standby. The standby pin is set to standby configuration. Other values return an error.
enable_pull_down	Decimal, 1 digit	If 0, disable pull-down on standby pin. If 1, enable pull-down on standby pin. Other values return an error. Note the enabling pull-up and pull-down at the same time is not possible and returns an error.
enable_pull_up	Decimal, 1 digit	If 0, disable pull-up on standby pin. If 1, enable pull-up on standby pin. Other values return an error. Note the enabling pull-up and pull-down at the same time is not possible and returns an error.

Results:

```
$PSTMHWSTANDBYOK*<checksum><cr><lf> if success
$PSTMHWSTANDBYERROR*<checksum><cr><lf> if no success
```

The hardware standby pin is configured

Example:

```
$PSTMHWSTANDBY,1,1,0*<checksum><cr><lf>
```

2.2.16 $PSTMGPSRESET

Reset the GPS receiver engine.

Synopsis:

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

Arguments:

None.

Results:

- The GPS receiver engine is reset.
- No message is sent as a reply.

Note: Using this command the GPS module will not reboot.

Example:

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

2.2.17 $PSTMGPSSUSPEND

Suspend the GPS receiver engine.

Synopsis:

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

Arguments:

None.

Results:

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

The GPS receiver engine is suspended.

Example:

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

2.2.18 $PSTMGPSRESTART

Restart the GPS receiver engine.

Synopsis:

```
PSTMGPSRESTART*<checksum><cr><lf>
```

Arguments:

None.

Results:

- The GPS receiver engine is restarted.
- No message is sent as a reply.

Example:

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

2.2.19 $PSTMENABLEPOSITIONHOLD

Enable/disable and set position for the position hold feature.

Synopsis:

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

Arguments:

Table 27. $PSTMENABLEPOSITIONHOLD field description

Parameter	Format	Description
on_off	Decimal, 1 digit	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)

Parameter	Format	Description
Lon	DDDMM.MMMMM	Longitude (degree-minute.minute decimals)
LonRef	'E' or 'W'	Longitude direction (east or west)
Alt [1]	dddddd.dddd	Altitude in meters (-1500 to 100000)

1. *The altitude has to be referred to WSG84.*

Results:

If successfully 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:

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

Example:

```
$PSTMENABLEPOSITIONHOLD,1,4811.365,N,01164.123,E,0530.0*<checksum><cr><lf>
$PSTMPOSITIONHOLDENABLED*<checksum><cr><lf>
```

2.2.20 $PSTMPPS

Allow interfacing all parameters for pulse per second management. This is a parametric command.

Synopsis:

```
$PSTMPPS,<cmd_mode>,<cmd_type>,<par_1>,…,<par_N>*<checksum><cr><lf>
```

Arguments:

Table 28. $PSTMPPS field description

Parameter	Format	Description
cmd_mode	Decimal, 1 digit	Select the command operation mode: 1 = GET operation (to get data from PPS manager) 2 = SET operation (to set data into PPS manager)
cmd_type	Decimal, 1 digit	1 = PPS_IF_ON_OFF_CMD 2 = PPS_IF_OUT_MODE_CMD 3 = PPS_IF_REFERENCE_CONSTELLATION_CMD 4 = PPS_IF_PULSE_DELAY_CMD 5 = PPS_IF_PULSE_DURATION_CMD 6 = PPS_IF_PULSE_POLARITY_CMD 7 = PPS_IF_PULSE_DATA_CMD 8 = PPS_IF_FIX_CONDITION_CMD 9 = PPS_IF_SAT_TRHESHOLD_CMD 10 = PPS_IF_ELEVATION_MASK_CMD 11 = PPS_IF_COSTELLATION_MASK_CMD 12 = PPS_IF_TIMING_DATA_CMD 13 = PPS_IF_POSITION_HOLD_DATA_CMD 14 = PPS_IF_AUTO_HOLD_SAMPLES_CMD 15 = PPS_IF_TRAIM_CMD 16 = PPS_IF_TRAIM_USED_CMD

Parameter	Format	Description
		17 = PPS_IF_TRAIM_RES_CMD 18 = PPS_IF_TRAIM_REMOVED_CMD 19 = PPS_IF_REFERENCE_TIME_CMD 20 = PPS_IF_CONSTELLATION_RF_DELAY_CMD 21 = PPS_IF_WGS84_ALT_POSITION_HOLD
par_1 … par_N		Parameters list according to the command type specification (see below).

2.2.20.1 *Getting PPS data (cmd_mode = 1)*

PPS_IF_PULSE_DATA_CMD

```
$PSTMPPS,1,7*<checksum><cr><lf>
```

Result :

```
$PSTMPPS,1,7,<out_mode>,<reference_time>,<pulse_delay>,<pulse_duration>,<pulse_polarity>*<checksum><cr><lf>
```

Table 29. PPS_IF_PULSE_DATA_CMD field description

Parameter	Format	Description
out_mode	Decimal, 1 digit	0 = PPS always generated. 1 = PPS generated on even seconds 2 = PPS generated on odd seconds
reference_time	Decimal, 1 digit	0 = UTC 1 = GPS_UTC 2 = GLONASS_UTC 3 = UTC_SU 4 = GPS_UTC_FROM_GLONASS 5 =BEIDOU_UTC 6 = UTC_NTSC 7 = GST 8 = UTC_GST 9 = GPS_FROM_GST *Note: UTC(SU) is the Soviet Union UTC, it is derived from GLONASS time applying the UTC delta time downloaded from GLONASS satellites.* *GPS_UTC_FROM_GLONASS is the GPS time derived from GLONASS time applying the GPS delta time downloaded from GLONASS satellites.* *If the software is configured to work in GLONASS only mode, UTC(SU) is identical to UTC and GPS_UTC_FROM_GLONASS is identical to GPS_UTC.*
pulse_delay	Decimal	Pulse delay [ns]
pulse_duration	Double	Pulse duration [s]
pulse_polarity	Decimal, 1 digit	0 = Not inverted 1 = Inverted

PPS_IF_TIMING_DATA_CMD

```
$PSTMPPS,1,12*<checksum><cr><lf>
Result:
$PSTMPPS,1,12,<fix_condition>,<sat_th>,<elevation_mask>,<constellation_mask>,<gps_rf_delay>,<glonass_rf_delay>*<checksum><cr><lf>
```

Table 30. **PPS_IF_TIMING_DATA_CMD field description**

Parameter	Format	Description
fix_condition	Decimal, 1 digit	1 = NO_FIX 2 = 2D_FIX 3 = 3D_FIX 4 = RTK_FIXED_FIX 5 = RTK_FLOAT_FIX 6 = PPS_MODE_FIX
sat_th	Decimal	Minimum number of satellites for the PPS generation
elevation_mask	Decimal	Minimum satellite elevation for satellite usage in timing filtering
constellation_mask	Decimal (bit mask)	Satellite constellation selection for usage in timing filtering bit0 = GPS bit1 = GLONASS bit3 = GALILEO bit7 = BEIDOU
gps_rf_delay	Decimal	GPS path RF delay [ns]
glonass_rf_delay	Decimal	GLONASS path RF delay [ns]

PPS_IF_POSITION_HOLD_DATA_CMD and NMEA_PPSIF_CMDID_WGS84_ALT_POSITION_HOLD_DATA

```
$PSTMPPS,1,13*<checksum><cr><lf>
Result:
$PSTMPPS,1,13,<on_off>,<lat>,<lat_dir>,<lon>,<lon_dir>,<h_msl>*<checksum><cr><lf>
```

Table 31. **PPS_IF_POSITION_HOLD_DATA_CMD and NMEA_PPSIF_CMDID_WGS84_ALT_POSITION_HOLD_DATA field description**

Parameter	Format	Description
on_off	Decimal, 1 digit	0 = Position hold disabled 1 = Position hold enabled
lat	DDmm.mmmmm	Position hold position latitude
lat_dir	“N” or “S”	North or south direction
lon	DDDmm.mmmmm	Position hold position longitude
lon_dir	“E” or “W”	East or west direction
h_msl	Double	Position hold mean see level altitude (-1500m to 100000m)

PPS_IF_TRAIM_CMD

```
$PSTMPPS,1,15*<checksum><cr><lf>
```

Result:

```
$PSTMPPS,1,15,<traim_enabled>,<traim_solution>,<ave_error>,<used_sats>,<removed_sats>*<checksum><cr><lf>
```

Table 32. **PPS_IF_TRAIM_CMD field description**

Parameter	Format	Description
traim_enabled	Decimal, 1 digit	TRAIM ON/OFF status 0 = OFF 1 = ON
traim_solution	Decimal, 1 digit	TRAIM algorithm status: 0 = UNDER alarm 1 = OVER alarm 2 = UNKNOWN
ave_error	Decimal	Average time error [ns]
used_sats	Decimal	Number of satellites used for timing correction
removed_sats	Decimal	Number of satellites removed by the timing correction

PPS_IF_TRAIM_USED_CMD

```
$PSTMPPS,1,16*<checksum>*<checksum><cr><lf>
```

Result:

```
$PSTMPPS,1,16,<traim_enabled>,<used_sats>,<sat1>,..,<satN>*<checksum><cr><lf>
```

Table 33. **PPS_IF_TRAIM_USED_CMD field description**

Parameter	Format	Description
traim_enabled	Decimal, 1 digit	TRAIM ON/OFF status 0 = OFF 1 = ON
used_sats	Decimal	Number of satellites used for timing correction
sat1..satN	Decimal	List of satellites IDs

PPS_IF_TRAIM_RES_CMD

```
$PSTMPPS,1,17*<checksum><cr><lf>
```

Result:

```
$PSTMPPS,1,17,<traim_enabled>,<used_sats>,<res1>,..,<resN>*<checksum><cr><lf>
```

Table 34. **PPS_IF_TRAIM_RES_CMD field description**

Parameter	Format	Description
traim_enabled	Decimal, 1 digit	TRAIM ON/OFF status 0 = OFF 1 = ON
used_sats	Decimal	Number of satellites used for timing correction
res1..resN	Decimal	List of satellites residuals [ns]. Each residual corresponds to the satellite in the used sat list at the same message position.

PPS_IF_TRAIM_REMOVED_CMD

```
$PSTMPPS,1,18*<checksum><cr><lf>
```

Result:

```
$PSTMPPS,1,18,<traim_enabled>,<rem_sats>,<sat1>,..,<satN>*<checksum><cr><lf>
```

Table 35. PPS_IF_TRAIM_REMOVED_CMD field description

Parameter	Format	Description
traim_enabled	Decimal, 1 digit	TRAIM ON/OFF status 0 = OFF 1 = ON
rem_sats	Decimal	Number of satellites removed by timing correction
sat1..satN	Decimal	List of satellites IDs

2.2.20.2 *Setting PPS data (cmd_mode = 2)*

PPS_IF_ON_OFF_CMD

```
$PSTMPPS,2,1,<on_off>*<checksum><cr><lf>
```

Table 36. PPS_IF_ON_OFF_CMD field description

Parameter	Format	Description
on_off	Decimal, 1 digit	0 = PPS disabled 1 = PPS enabled

PPS_IF_OUT_MODE_CMD

```
$PSTMPPS,2,2,<out_mode>*<checksum><cr><lf>
```

Table 37. PPS_IF_OUT_MODE_CMD field description

Parameter	Format	Description
out_mode	Decimal, 1 digit	0 = PPS always generated 1 = PPS generated on even seconds 2 = PPS generated on odd seconds

PPS_IF_REFERENCE_TIME_CMD

```
$PSTMPPS,2,19,<reference_time>*<checksum><cr><lf>
```

Table 38. PPS_IF_REFERENCE_TIME_CMD field description

Parameter	Format	Description
reference_time	Decimal, 1 digit	0 = UTC 1 = GPS_UTC 2 = GLONASS_UTC 3 = UTC_SU 4 = GPS_UTC_FROM_GLONASS 5 = BEIDOU_UTC 6 = UTC_NTSC 7 = GST 8 = UTC_GST 9 = GPS_FROM_GST

Parameter	Format	Description
		Note: UTC(SU) is the Soviet Union UTC, it is derived from GLONASS time applying the UTC delta time downloaded from GLONASS satellites. *GPS_UTC_FROM_GLONASS is the GPS time derived from GLONASS time applying the GPS delta time downloaded from GLONASS satellites.* *If the software is configured to work in GLONASS only mode, UTC(SU) is identical to UTC and GPS_UTC_FROM_GLONASS is identical to GPS_UTC.*

PPS_IF_PULSE_DELAY_CMD

```
$PSTMPPS,2,4,<pulse_delay>*<checksum><cr><lf>
```

Table 39. PPS_IF_PULSE_DELAY_CMD field description

Parameter	Format	Description
pulse_delay	Decimal	Pulse delay [ns]

PPS_IF_CONSTELLATION_RF_DELAY_CMD

```
$PSTMPPS,2,20,<sat_type><time_delay>*<checksum><cr><lf>
```

Table 40. PPS_IF_CONSTELLATION_RF_DELAY_CMD field description

Parameter	Format	Description
sat_type	Decimal	Satellite constellation type : 0 = GPS 1 = GLONASS 3 = Galileo 7 = BEIDOU
time_delay	Decimal	Time delay [ns]

PPS_IF_PULSE_DURATION_CMD

```
$PSTMPPS,2,5,<pulse_duration>*<checksum><cr><lf>
```

Table 41. PPS_IF_PULSE_DURATION_CMD field description

Parameter	Format	Description
pulse_duration	Double	Pulse duration [s]

PPS_IF_PULSE_POLARITY_CMD

```
$PSTMPPS,2,6,<pulse_polarity>*<checksum><cr><lf>
```

Table 42. PPS_IF_PULSE_POLARITY_CMD field description

Parameter	Format	Description
pulse_polarity	Decimal, 1 digit	0 = Not inverted 1 = Inverted

PPS_IF_PULSE_DATA_CMD

```
$PSTMPPS,2,7,<out_mode>,<reference_time>,<pulse_delay>,<pulse_duration>,<pulse_polarity>*<checksum><cr><lf>
```

Table 43. **PPS_IF_PULSE_DATA_CMD field description**

Parameter	Format	Description
out_mode	Decimal, 1 digit	0 = PPS always generated 1 = PPS generated on even seconds 2 = PPS generated on odd seconds
reference_time	Decimal, 1 digit	0 = UTC 1 = GPS_UTC 2 = GLONASS_UTC 3 = UTC_SU 4 = GPS_UTC_FROM_GLONASS
pulse_delay	Decimal	Pulse delay [ns]
pulse_duration	Double	Pulse duration [s]
pulse_polarity	Decimal, 1 digit	0 = Not inverted 1 = Inverted

PPS_IF_FIX_CONDITION_CMD

```
$PSTMPPS,2,8,<fix_condition>*<checksum><cr><lf>
```

Table 44. **PPS_IF_FIX_CONDITION_CMD field description**

Parameter	Format	Description
fix_condition	Decimal, 1 digit	1 = NO_FIX 2 = 2D_FIX 3 = 3D_FIX 4 = RTK_FIXED_FIX 5 = RTK_FLOAT_FIX 6 = PPS_MODE_FIX

PPS_IF_SAT_TRHESHOLD_CMD

```
$PSTMPPS,2,9,<sat_th>*<checksum><cr><lf>
```

Table 45. **PPS_IF_SAT_TRHESHOLD_CMD field description**

Parameter	Format	Description
sat_th	Decimal	Minimum number of satellites for the PPS generation.

PPS_IF_ELEVATION_MASK_CMD

```
$PSTMPPS,2,10,<elevation_mask>*<checksum><cr><lf>
```

Table 46. **PPS_IF_ELEVATION_MASK_CMD field description**

Parameter	Format	Description
elevation_mask	Decimal	Minimum satellite elevation for satellite usage in timing filtering.

PPS_IF_CONSTELLATION_MASK_CMD

```
$PSTMPPS,2,11,<constellation_mask>*<checksum><cr><lf>
```

Table 47. **PPS_IF_CONSTELLATION_MASK_CMD field description**

Parameter	Format	Description
constellation_mask	Decimal (bit mask)	Satellite constellation selection for usage in timing filtering. bit0 = GPS bit1 = GLONASS *Note: This parameter enables the usage of mixed constellations satellites in the timing filtering. If bit0 is enabled GPS satellites are used to correct the GLONASS reference time together with GLONASS satellites. If bit1 is enabled, GLONASS satellites are used to correct the GPS reference time together with the GPS satellites. When constellation mask is zero (default) only GPS sats are used to correct the GPS reference time and only GLONASS sats are used to correct the GLONASS reference time.*

PPS_IF_TIMING_DATA_CMD

```
$PSTMPPS,2,12,<fix_condition>,<sat_th>,<elevation_mask>,<constellation_mask>*<checksum><cr><lf>
```

Table 48. **PPS_IF_TIMING_DATA_CMD field description**

Parameter	Format	Description
fix_condition	Decimal, 1 digit	1 = NO_FIX. 2 = 2D_FIX. 3 = 3D_FIX. 4 = RTK_FIXED_FIX 5 = RTK_FLOAT_FIX 6 = PPS_MODE_FIX
sat_th	Decimal	Minimum number of satellites for the PPS generation
elevation_mask	Decimal	Minimum satellite elevation for satellite usage in timing filtering
constellation_mask	Decimal (bit mask)	Satellite constellation selection for usage in timing filtering bit0 = GPS bit1 = GLONASS

PPS_IF_POSITION_HOLD_DATA_CMD

```
$PSTMPPS,2,13,<on_off>,<lat>,<lat_dir>,<lon>,<lon_dir>,<h_msl>*<checksum><cr><lf>
```

Table 49. **PPS_IF_POSITION_HOLD_DATA_CMD field description**

Parameter	Format	Description
on_off	Decimal, 1 digit	0 = Position hold disabled 1 = Position hold enabled
lat	DDmm.mmmmm	Position hold position latitude
lat_dir	“N” or “S”	North or south direction
lon	DDDmm.mmmmm	Position hold position longitude
lon_dir	“E” or “W”	East or west direction
h_msl	Double	Position hold mean see level altitude

PPS_IF_AUTO_HOLD_SAMPLES_CMD

```
$PSTMPPS,2,14,<auto_ph_samples>*<checksum><cr><lf>
```

Table 50. **PPS_IF_AUTO_HOLD_SAMPLES_CMD field description**

Parameter	Format	Description
auto_ph_samples	Decimal, 1 digit	Number of position samples for the auto position algorithm. If the number of samples is set to “0” the auto position hold feature is disabled. The position average evaluation is restarted every time the command is executed.

PPS_IF_TRAIM_CMD

```
$PSTMPPS,2,15,<on_off>,<alarm>*<checksum><cr><lf>
```

Table 51. **PPS_IF_TRAIM_CMD field description**

Parameter	Format	Description
on_off	Decimal, 1 digit	0 = TRAIM disabled 1 = TRAIM enabled
alarm	Double	TRAIM alarm [s] – scientific notation is allowed

Results:

According to the operation mode and to the command type, data is set into the PPS manager, or it is retrieved from the PPS manager.

2.2.21 $PSTMDUMPIONO

This command allows the user to dump the iono packet for each supported constellation.

Synopsis:

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

Arguments:

None.

Results:

- Returns the iono packet for each supported constellation.

```
 $PSTMIONOPARAMS,<sat_type>,<available>,<data0>,<data1>,<data2>,<data3>,<data4>,<data5>,<data6>,<data7>*<checksum><cr><lf>
```

Where:

Table 52. $PSTMDUMPIONO:sat_type description

Parameter	Format	Description
sat_type	Decimal	Satellite constellation type : • 0 = GPS • 3 = Galileo • 7 = BEIDOU • 10 = IRNSS

Table 53. $PSTMDUMPIONO data description for GPS, BEIDOU and IRNSS

Parameter	Format	Description
Available	Decimal	Data availability information flag
Data0	Decimal	Ionospheric cubic coefficient alpha0
Data1	Decimal	Ionospheric cubic coefficient alpha1
Data2	Decimal	Ionospheric cubic coefficient alpha2
Data3	Decimal	Ionospheric cubic coefficient alpha3
Data4	Decimal	Ionospheric cubic coefficient beta0
Data5	Decimal	Ionospheric cubic coefficient beta1
Data6	Decimal	Ionospheric cubic coefficient beta2
Data7	Decimal	Ionospheric cubic coefficient beta3

Table 54. $PSTMDUMPIONO data description for Galileo

Parameter	Format	Description
Available	Decimal	Data availability information flag
Data0	Decimal	Effective ionisation level 1st order parameter
Data1	Decimal	Effective ionisation level 2nd order parameter
Data2	Decimal	Effective ionisation level 3rd order parameter
Data3	Decimal	Raw ionospheric disturbance flag for region 1
Data4	Decimal	Raw ionospheric disturbance flag for region 2
Data5	Decimal	Raw ionospheric disturbance flag for region 3
Data6	Decimal	Raw ionospheric disturbance flag for region 4
Data7	Decimal	Raw ionospheric disturbance flag for region 5

2.2.22 $PSTMIONOPARAMS

Uploads a specific iono packet into the receiver NVM. The uploaded iono packet is retained until a new iono packet for the same constellation is successfully uploaded or downloaded from the navigation message. There is no answer from the receiver.

Synopsis in case of <sat_type=0>:

```
$PSTMIONOPARAMS,<sat_type=0>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>*<checksum><cr><lf>
```

Synopsis in case of <sat_type=7>:

```
$PSTMIONOPARAMS,<sat_type=7>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>*<checksum><cr><lf>
```

Synopsis in case of <sat_type=10>:

```
$PSTMIONOPARAMS ,<sat_type=10>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>*<checksum><cr><lf>
```

Synopsis in case of <sat_type=3>:

```
$PSTMIONOPARAMS,<sat_type=3>,1,<ai0>,<ai1>,<ai2>,<Region1>,<Region2>,<Region3>,<Region4>,<Reg
ion5>*<checksum><cr><lf>
```

Arguments:

Table 55. $PSTMIONOPARAMS field description

Parameter	Format	Description
sat_type	Decimal, 1 digit	0 = GPS 3 = Galileo 7 = BeiDou 10 = IRNSS
Availabe	Decimal, 1 digit	Data availability information flag
A0,A1,A2,A3	Decimal, 3 digits	These parameters are used only if sat_type=0,7 or 10 Iono parameters, raw integer values as from navigation messages
B0,B1,B2,B3	Decimal, 3 digits	These parameters are used only if sat_type=0,7 or 10 Iono parameters, raw integer values as from navigation messages
ai0,ai1,ai2	Decimal, 3 digits	These parameters are used only if sat_type=3 Iono parameters, raw integer values as from navigation messages
Region1,Region2,Region3,Region4,Region5	Binary	These parameters are used only if sat_type=3 Galileo iono regions (possible values are 0 or 1)

Results:

- No message is sent as a reply.

Example:

```
$PSTMIONOPARAMS,0,1,6,2,255,254,41,6,255,248*<checksum><cr><lf>
$PSTMIONOPARAMS,7,1,3,18,236,48,73,192,127,129*<checksum><cr><lf>
$PSTMIONOPARAMS,3,1,123,15,316,0,0,0,0,0*<checksum><cr><lf>
```

2.2.23 $PSTMFEDUMP

This message reports the current values of all RF front-end registers.

Synopsis:

```
$PSTMFEDUMP,<FE>,<reg_addr>*<checksum><cr><lf>
```

Arguments:

Table 56. **$PSTMFEDUMP field description**

Parameter	Format	Description
FE	1 Digit	0 = For internal front-end 1 = Reserved
reg_addr	Decimal	Optional. The relative address of the FE register to read. If omitted, then all registers are dump.

Results:

If the command is executed with success the following messages are sent for the internal front-end:

```
$PSTMFEDUMP,00,<R0>,…,37,<R55>*<checksum><cr><lf>
```

If the command is executed for one register only:

Example:

```
$PSTMFEDUMP,0,10
$PSTMFEDUMP, 0a,<R10>*<checksum><cr><lf>
```

2.2.24 $PSTMDUMPGLONASSICB

This message reports the current values of the GLONASS pseudorange Inter-Channel Biases (ICB) tables.

Synopsis:

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

Results:

If the command is successfully executed the following messages are sent:

```
$PSTMGLONASSICB,1,50,40,30,20,0,0,0,0,0,0,0,0,0*<checksum><cr><lf>
$PSTMGLONASSICB,2,0,0,0,0,0,0,0,0,0,0,0,0,0*<checksum><cr><lf>
```

If the command is not successful, the following messages are sent:

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

2.2.25 $PSTMGLONASSICB

Uploads a specific ICB table into the receiver NVM. There is no answer from the receiver.

Synopsis:

```
$PSTMGLONASSICB,<carrier>,<chan-7>,<chan-6>,<chan-5>,<chan-4>,<chan-3>,<chan-2>,<chan-1>,<cha
n0>,<chan+1>,<chan+2>,<chan+3>,<chan+4>,<chan+5>,<chan+6>*<checksum><cr><lf>
```

Arguments:

Table 57. **$PSTMGLONASSICB field description**

Parameter	Format	Description
carrier	Decimal, 1 digit	1 = GLONASS L1 2 = GLONASS L2
Chan n	Decimal, 3 digits	Pseudorange bias in [dm] for the GLONASS channel n = -7…+6

Example:

```
$PSTMGLONASSICB,1,-7,-6,-5,-4,-3,-2,-1,100,1,2,3,4,5,6*<checksum><cr><lf>
```

Note: Only one ICB per carrier (L1, L2) is stored on NVM. It is under the responsibility of the user to store ICB tables, which are matching the frequency plan in use.

Results:

If the command is successfully executed the following messages are sent:

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

If the command is not successful, the following messages are sent:

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

2.2.26 $PSTMDUMPIFB

Dump the current IFB values.

Synopsis:

```
$PSTMDUMPIFB,<data_set>*<checksum><cr><lf>
```

Arguments:

Table 58. $PSTMDUMPIFB field description

Parameter	Format	Description
Date_set	Decimal, 1 digit	Optional. Dataset: 0–Values of IFB 1–Mean value of PSR IFB 2–Value of PSR IFB-Rate EMA 3–Value of ADR IFB-Rate EMA

Note: Without data_set parameter 0 is default value.

Results:

If the command is successfully executed, the following message is sent:

```
$PSTMIFB<cr><lf>
$PSTMIFB,950,100,0,0,0,0,0,0,0,0,0,0*<checksum><cr><lf>
```

2.2.27 $PSTMIFB

Save a custom inter frequency pseudorange biases (IFB) array in NVM. There is no answer from the receiver.

Synopsis:

```
$PSTMIFB,<GPS_L2C_L1>,<GLONASS_L2_L1>,<BeiDou_B2I_B1I>,<GPS_L5_L1>,<GAL_E5a_E1>,<GAL_E5b_L1>,
<GAL_E6_E1>,<BeiDou_B2a_B1I>,<GPS_L1C_L1>,<BeiDou_B1c_B1I>,<BeiDou_B3I_B1I>,<BeiDou_B2b_B1I>*
<checksum><cr><lf>
```

Arguments:

Table 59. $PSTMIFB field description

Parameter	Format	Description
GPS_L2C_L1	Decimal, 3 digits	IFB [dm] between GPS L1 and L2 carriers
GLONASS_L2C_L1	Decimal, 3 digits	IFB [dm] between GLONASS L1 and L2 carriers
BeiDou_B2I_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B1I and B2I carriers
GPS_L5_L1	Decimal, 3 digits	IFB [dm] between GPS L1 and L5 carriers
GAL_E5a_E1	Decimal, 3 digits	IFB [dm] between GALILEO E5a and E1 carriers
GAL_E5b_L1	Decimal, 3 digits	IFB [dm] between GALILEO E5b and L1 carriers
GAL_E6_E1	Decimal, 3 digits	IFB [dm] between GALILEO E6 and E1 carriers
BeiDou_B2a_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B2a and B1I carriers
GPS_L1C_L1	Decimal, 3 digits	IFB [dm] between GPS L1C and L1 carriers

Parameter	Format	Description
Beidou_B1c_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B1c and B1I carriers
Beidou_B3I_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B3I and B1I carriers
Beidou_B2b_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B2b and B1I carriers

Example:

```
$PSTMIFB,1,2,3,0,0,0,0,0,0,0,0,0*<checksum><cr><lf>
```

Note: Only one set of custom IFB(s) is stored on NVM. It is the responsibility of the user to store an IFB array, which is matching the frequency plan in use.

2.2.28 $PSTMSETCONSTMASK

Set the GNSS constellation mask and usage (optional). It allows switching the multifrequency configuration at runtime. In case of reset, constellation mask is restored to default value.

Synopsis:

```
$PSTMSETCONSTMASK,<constellation mask>,<multi-freq. Mask>*<checksum><cr><lf>
```

or

```
$PSTMSETCONSTMASK,<constellation mask>,<multi-freq. Mask>,<usage mask>*<checksum><cr><lf>
```

Arguments:

Table 60. $PSTMSETCONSTMASK field description

Parameter	Format	Description
constellation_mask	Decimal, 1 – 15728639	It is a bit mask where each bit enables/disables a specific constellation independently of the others. See AN firmware configuration document CDB page 21 line 4 for details
multi-freq. mask	Decimal, 1 – 27795599	It is a bits mask where some bit configure multifrequency, pilot/data or tracking mode, see AN firmware configuration document page 21 line 5
usage_mask	Decimal, 1 – 122543	Specify which constellation is used for PVT (optional), see AN firmware configuration document page 21 line 6

Note: When only 2 arguments are specified then usage_mask == constellation_mask.

Note: All masks values must be converted in decimal.

Results:

If successfully, 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:

```
$PSTMSETCONSTMASK,527,0 à GPS L1CA+L2C, QZSS L1CA, GLO G1, GAL E1
$PSTMSETCONSTMASK,655,0,137 à GPS L1CA+L2C, QZSS L1CA, GLO G1, GAL E1, BDS B1i
```

2.2.29 $PSTMDUMPCHIPINFO

Dump the registers for current chip version, BCS version, DCF version and DCF value. Note that the description of the register's content is not described here, please refer to the appropriate datasheet document description.

Synopsis:

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

Results:

If the command is successfully executed, the following message is sent reporting the registers information:

```
 $PSTMDUMPCHIPINFO,<chip_id1>,<chip_id2>,<chip_id3>,<chip_id4>,<bcs_ver>,<reserved>,<dcf_ver>
,<dcf_388>,<dcf_389>,<dcf_38a>,<dcf_38b>,<dcf_38c>*<checksum><cr><lf>
```

Where:

Table 61. $PSTMDUMPCHIPINFO field description

Parameter	Format	Description
chip_id1	Hexadecimal, 8 digits	Chip ID 1 on 4
chip_id2	Hexadecimal, 8 digits	Chip ID 2 on 4
chip_id3	Hexadecimal, 8 digits	Chip ID 3 on 4
chip_id4	Hexadecimal, 8 digits	Chip ID 4 on 4
bcs_ver	Hexadecimal, 8 digits	BCS version
reserved	Hexadecimal, 8 digits	
dcf_ver	Hexadecimal, 8 digits	DCF version
dcf_388	Hexadecimal, 16 digits	DCF value offset 0x388
dcf_389	Hexadecimal, 16 digits	DCF value offset 0x389
dcf_38a	Hexadecimal, 16 digits	DCF value offset 0x38a
dcf_38b	Hexadecimal, 16 digits	DCF value offset 0x38b
dcf_38c	Hexadecimal, 16 digits	DCF value offset 0x38c

Otherwise, on DCF read error only this message is reported:

```
$PSTMDUMPCHIPINFO,<chip_id1>,<chip_id2>,<chip_id3>,<chip_id4>,<bcs_ver>,<reserved>,<dcf_ver>*
<checksum><cr><lf>
```

Examples:

```
$PSTMDUMPCHIPINFO,0x00513232,0x32313033,0x09076065,0x20230620,0x00030000,0x00000102,0x0002000
0,0x0200000200000015,0x0040004200000001,0x0200000a10180000,0x0200000c10000400,0x0c00002000000
000*3F
```

2.2.30 $PSTMTRKJAMMER

This command is intended to test the jammer presence in the hardware when no signal is applied at the input.

Synopsis:

To start the test

```
$PSTMTRKJAMMER,1*<checksum><cr><lf>
```

To stop the test

```
$PSTMTRKJAMMER,0*<checksum><cr><lf>
```

2.2.31 $PSTMGETSWVER

Get the version string of the libraries embedded in the software application.

Synopsis:

```
$PSTMGETSWVER,<id>*<checksum><cr><lf>
```

Arguments:

Table 62. **$PSTMGETSWVER field description**

Parameter	Format	Description
id	Integer	Depending on the value of the <lib_id> parameter, the following version numbering is delivered by the command: 0 = GNSS library version 1 = FRRTOS version 2 = SDK app version 6 = binary image version 8 = STAGPS version 10 = DR version 11 = Software configuration ID 255 = All versions strings (as reported at the NMEA startup).

Results:

Returns the requested library version in $PSTMVER message.

Example:

```
$PSTMGETSWVER,0
```

Note: If a wrong id is passed as a parameter to the command, the $PSTMVERERROR message is sent.

Note: When id is 255 consecutive messages are sent reporting the library version string on each line following the above message syntax.

2.2.32 $PSTMNVMSWAP

Execute a bank swap on the NVM GPS backup memory.

Synopsis:

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

Arguments:

None.

Results:

The nonvolatile backup memory banks are swapped.

Example:

```
$PSTMNVMSWAP
```

Note: This command is supported only by platforms or software configurations where the backup memory is based on flash NOR or SQI memories.

2.2.33 $PSTMSETANTSENSOPMODE

This command is used to change the antenna detection operating mode at runtime.

This can be used to switch in manual mode to take direct action on antenna power as well as RF path. This command can be even used to switch back to automatic mode after a manual service or an anomalous condition.

Synopsis:

```
$PSTMSETANTSENSOPMODE,<operating_mode><cr><lf>*<checksum><cr><lf>
```

Arguments:

Table 63. **$PSTMSETANTSENSOPMODE field description**

Parameter	Format	Description
operating_mode	Decimal, 1 digit	Current antenna detection operating mode: 0 = Automatic mode 1 = Manual mode

Results:

Antenna sensing operating mode is set according to the command parameter. In case of no errors, the following message is returned.

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

In case of errors, this error message is returned:

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

2.2.34 $PSTMSETANTSENSMANUAL

This command is used to act manually on the antenna detection at runtime.

This can be used to manually force antenna power on/off, to force RF, to request the antenna status as well as resuming the antenna sensing process.

Note: Before issuing this command, antenna detection operating mode shall be switched to manual (for more info see `$PSTMSETANTSENSOPMODE`).

Synopsis:

```
$PSTMSETANTSENSMANUAL,<pwr_switch>,<rf_path>,<get_update>,<start_stop>*<checksum><cr><lf>
```

Arguments:

Table 64. **$PSTMSETANTSENSMANUAL field description**

Parameter	Format	Description
pwr_switch	Decimal, 2 digits	Forces antenna power switch 0 = No effect 1 = Antenna power on 2 = Antenna power off
rf_path	Decimal, 2 digits	Forces antenna RF path *Warning: this setting shall not be used as only the external antenna is taken into account by the current implementation.* 0 = No effect 1 = Switches on external antenna 2 = Switches on internal antenna
get_update	Decimal, 1 digit	Forces the antenna status message once 0 = Do not send antenna status 1 = Send antenna status
start_stop	Decimal, 1 digit	Force start or stop of antenna detection process 0 = Antenna detection process set to active 1 = Antenna detection process set to stop

Results:

In case of no errors the message `$PSTMSETANTSENSOPMODEOK` is returned and if get_update is 1, the message `$PSTMANTENNASTATUS` is sent.

In case of errors, the error message `$PSTMSETANTSENSOPMODEERROR` is returned.

2.2.35 $PSTMSETTHTRK

Configures the CN0 and Angle elevation mask thresholds for tracking. This command changes these parameters at runtime and no reset is required. In case of reset tracking CN0 and Angle elevation mask are restored to default value.

Synopsis:

```
$PSTMSETTHTRK,<cn0>,<el>*<checksum><cr><lf>
```

Arguments:

Table 65. $PSTMSETTHTRK field description

Parameter	Format	Description
cn0	Decimal	Tracking CN0 threshold as dBHz
el	Double	Tracking elevation mask angle as degree

Results:

If the command syntax is correct and the tracking CN0 and elevation mask are correctly changed this message is returned:

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

In case of errors, this error message is returned:

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

2.2.36 $PSTMSETTHPOS

Configures the CN0 and Angle elevation mask thresholds for positioning. This command changes these parameters at runtime and no reset is required. In case of reset positioning CN0 and angle elevation mask are restored to default value.

Synopsis:

```
$PSTMSETTHPOS,<cn0>,<el>*<checksum><cr><lf>
```

Arguments:

Table 66. $PSTMSETTHPOS field description

Parameter	Format	Description
cn0	Decimal	Positioning CN0 threshold as dB.
el	Double	Positioning elevation mask angle as degree.

Results:

If the command syntax is correct and the positioning CN0 and elevation mask are correctly changed this message is returned

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

In case of errors, this error message is returned

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

2.2.37 $PSTMGETRTCTIME

Get the current RTC time.

Synopsis:

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

Arguments:

None.

Results:

System sends RTC data and status.

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

Where:

Table 67. $PSTMRTCTIME field description

Parameter	Format	Description
Time	hhmmss.mms	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 – TOW_TIME • 3 - USER_TIME • 4 - USER_RTC_TIME • 5 - RTC_TIME • 6 - RTC_TIME_ACCURATE • 7 - APPROX_TIME • 8 - ACCURATE_TIME • 9 - POSITION_TIME • 10 - EPHEMERIS_TIME

Example:

```
$PSTMGETRTCTIME
```

2.2.38 $PSTMSNWRITE

Save a serial number string into flash memory.

Synopsis:

```
$PSTMSNWRITE,<SerialNumber>*<checksum><cr><lf>
```

Arguments:

Table 68. $PSTMSNWRITE field description

Parameter	Format	Description
SerialNumber	String	Serial number is a string composed of characters [a-z], [A-Z], [0-9], or special characters % : ! ; . , ( ) # [ ] = {} / limit is 240 characters.

Results:

If the string is correctly saved in memory, this message is returned:

```
$PSTMSNWRITEOK*5e
```

In case of writing error, this error message is returned:

```
$PSTMSNWRITEERROR*02
```

2.2.39 $PSTMSNREAD

Read a serial number string from flash memory.

Synopsis:

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

Arguments:

None.

Results:

If the string is correctly read from memory, this message is returned:

```
$PSTMSNREAD,<SerialNumber>*<checksum><cr><lf>
```

If no serial number is available in memory, this message is returned:

```
$PSTMSNREADINVALID*44
```

In case of reading error, this error message is returned:

```
$PSTMSNREADERROR*4d
```

Where:

Table 69. $PSTMSNREAD field description

Parameter	Format	Description
SerialNumber	String	Serial number is a string composed of characters [a-z], [A-Z], [0-9], or special characters % : ! ; . , ( ) # [ ] = {} /.

2.2.40 RF test mode

Some NMEA commands are dedicated to perform multiband RF tests in production.

2.3 OSNMA NMEA commands

2.3.1 $PSTMNMA

Allow interfacing all parameters for OSNMA management. This is a parametric command.

Synopsis:

```
$PSTMNMA,<cmd_type>,<par_1>,…,<par_N>*<checksum><cr><lf>
```

Arguments:

Table 70. $PSTMNMA field description

Parameter	Format	Description
cmd_type	Decimal, 1 digit	0 = OSNMA NVM invalidate 1 = OSNMA enable / disable 2 = OSNMA NS 3 = Reserved 4 = OSNMA TSL 5 = OSNMA MAC Seq
par_1 … par_N		Parameters list according to the command type specification (see below).

Command to invalidate OSNMA NVM memory.

```
$PSTMNMA,0*<checksum><cr><lf>
```

OSNMA enable / disable

Command to enable or disable OSNMA.

```
$PSTMNMA,1,<on_off>*<checksum><cr><lf>
```

Table 71. $PSTMNMA enable/disable field description

Parameter	Format	Description
On_off	Decimal, 1 digit	OSNMA status 0 = OSNMA disable 1 = OSNMA enable

OSNMA NS

Command to set NS parameter

```
$PSTMNMA,2,<NS>*<checksum><cr><lf>
```

Table 72. $PSTMNMA NS field description

Parameter	Format	Description
NS	Decimal, 1 to 3 digits	Satellites (hash) per subframe (NS).

OSNMA TSL

```
$PSTMNMA,4,<TSL>*<checksum><cr><lf>
```

Table 73. $PSTMNMA TSL field description

Parameter	Format	Description
TSL	Decimal, 1 to 5 digits	Target security level

OSNMA MAC Seq

```
$PSTMNMA,5,<MAC_SEQ>*<checksum><cr><lf>
```

Table 74. $PSTMNMA MAC field description

Parameter	Format	Description
MAC_SEQ	Decimal (bit mask)	MACSEQ feature options flags Bit 0 = global feature enable 1=ON 0=OFF

Results:

If the command syntax is correct and the command is correctly executed this message returns.

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

In case of errors, this error message returns

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

2.3.2 $PSTMNMAPK

Set the OSNMA ECDSA public key at the specified index

Synopsis:

```
$PSTMNMAPK,<msg_id>,<key_index>,<pubk>*<checksum><cr><lf>
```

Arguments:

Table 75. $PSTMNMAPK field description

Parameter	Format	Description
msg_id	Decimal, 1 or 2 digits	Message ID (MID), from 0 to 15
key_index	Decimal, 1 or 2 digits	Key index from 0 to 15
Pubk	hexadecimal, 66 digits or more	Public key

Results:

If the command syntax is correct and the key is correctly changed this message returns

```
$PSTMNMAPKOK,<mtr_id>*<checksum><cr><lf>
```

The field mtr_id is present only if public key verification is enabled in firmware config ([P37,L10] bit 9).

In case of errors, this error message returns

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

2.3.3 $PSTMNMASETMTR

Set Merkle tree root value.

Synopsis:

```
$PSTMNMASETMTR,<mtr_id>,<sha_type>,<mtr_data>*<checksum><cr><lf>
```

Arguments:

Table 76. $PSTMNMASETMTR field description

Parameter	Format	Description
mtr_id	Decimal, 1 digit	Merkle tree root index: 0= Current 1= Future
sha_type	Decimal, 1 digit	Sha type 2 = sha 256 3 = sha 521
mtr_data	Hexadecimal, 64 digits	Merkle tree root value

Results:

If the command syntax is correct and the command is correctly executed this message returns

```
$PSTMNMAMTROK,<mtr_id>*<checksum><cr><lf>
```

The field mtr_id is valid only if public key verification is enabled in firmware config ([P37,L10] bit 9).

In case of errors, this error message returns:

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

2.3.4 $PSTMNMAGETMTR

Get Merkle tree root value

Synopsis:

```
$PSTMNMAGETMTR,<mtr_id>*<checksum><cr><lf>
```

Arguments:

Table 77. $PSTMNMAGETMTR field description

Parameter	Format	Description
mtr_id	Decimal, 1 digit	Merkle tree root index: 0= Current 1= Future

Results:

If the command syntax is correct and the command is correctly executed this message returns:

```
$PSTMNMAGETMTROK,<mtr_id>,<sha_type>,<mtr_data>*<checksum><cr><lf>
```

Where:

Table 78. $PSTMNMAGETMTROK field description

Parameter	Format	Description
mtr_id	Decimal, 1 digit	Merkle tree root index: 0= Current 1= Future
sha_type	Decimal, 1 digit	Sha type 2 = sha 256 3 = sha 521
mtr_data	Hexadecimal, 64 digits	Merkle tree root value

In case of errors, this error message returns:

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

2.3.5 $PSTMNMACLRMTR

Clear Merkle tree root

Synopsis:

```
$PSTMNMACLRMTR,<mtr_id>*<checksum><cr><lf>
```

Arguments:

Table 79. $PSTMNMACLRMTR field description

Parameter	Format	Description
mtr_id	Decimal, 1 digit	Merkle tree root index: 0= Current 1= Future

Results:

If the command syntax is correct and the command is correctly executed this message returns:

```
$PSTMNMACLRMTROK,<mtr_id>*<checksum><cr><lf>
```

Where:

Table 80. $PSTMNMACLRMTROK field description

Parameter	Format	Description
mtr_id	Decimal, 1 digit	Merkle tree root index: 0= Current 1= Future

In case of errors, this error message returns:

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

2.3.6 $PSTMNMASETITN

Set intermediate Merkle tree node data for level j

Synopsis:

```
$PSTMNMASETITN,<j>,<itn>*<checksum><cr><lf>
```

Arguments:

Table 81. $PSTMNMASETITN field description

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)
itn	Hexadecimal, 64 digits	Intermediate tree node data level j

Results:

If the command syntax is correct and the command is correctly executed this message returns:

```
$PSTMNMASETITNOK,<j>*<checksum><cr><lf>
```

Where:

Table 82. $PSTMNMASETITNOK field description

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)

In case of errors, this error message returns:

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

2.3.7 $PSTMNMAGETITN

Get intermediate Merkle tree node node data level j

Synopsis:

```
$PSTMNMAGETITN,<j>*<checksum><cr><lf>
```

Arguments:

Table 83. **$PSTMNMAGETITN field description**

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)

Results:

If the command syntax is correct and the command is correctly executed this message returns:

```
$PSTMNMAGETITNOK,<j>,<itn>*<checksum><cr><lf>
```

Where:

Table 84. **$PSTMNMAGETITNOK field description**

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)
itn	Hexadecimal, 64 digits	Intermediate tree node data level j

In case of errors, this error message returns:

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

2.3.8 $PSTMNMACLRITN

Clear intermediate Merkle tree node level j

Synopsis:

```
$PSTMNMACLRITN,<j>*<checksum><cr><lf>
```

Arguments:

Table 85. **$PSTMNMACLRITN field description**

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)

Results:

If the command syntax is correct and the command is correctly executed this message returns:

```
$PSTMNMACLRITNOK,<j>*<checksum><cr><lf>
```

Where:

Table 86. **$PSTMNMACLRITNOK field description**

Parameter	Format	Description
j	Decimal, 1 digit	Intermediate tree node level (0 to 3)

In case of errors, this error message returns:

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

2.3.9 $PSTMNMACLRAM

Clear OSNMA alert message state.

Synopsis:

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

Results:

If the command syntax is correct and the command is correctly executed this message returns.

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

In case of errors, this error message returns:

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

2.3.10 $PSTMNMACLRKROOT

Clear OSNMA root key stored in NVM. This command is used to perform OSNMA warm start. A reset is needed.

Synopsis:

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

Results:

If the command syntax is correct and the command is correctly executed this message returns:

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

In case of errors, this error message returns:

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

3 Messages

This section contains both the standard NMEA messages and the proprietary messages delivered by the ST-GNSS system.

3.1 Standard NMEA messages list

Table 87. Standard NMEA message list

Syntax	Default	Description
$--GNS	ON	NMEA: Global position system fix data
$--GGA	ON	NMEA: Global position system fix data
$--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
$--RMC	ON	NMEA: Recommended minimum specific GNSS data
$--VTG	ON	NMEA: Track made good and ground speed
$--GST	ON	NMEA: GNSS pseudorange noise statistics
$--GBS	ON	GNSS satellite fault detection
$--TXT	ON	Test transmission
$--AMC	OFF(1)	Authenticated minimum specific data
$--ASA	OFF(1)	Authenticated active satellites
$--DTM	OFF	Datum reference
$--ZDA	OFF	Time and date

1. *Enabled only on OSNMA firmware.*

3.2 ST NMEA messages list

Table 88. Proprietary NMEA message list

Syntax	Default	Description
$PSTMPRES	ON	Position residuals
$PSTMVRES	ON	Velocity residuals
$PSTMTG (1)	ON	Time and global information
$PSTMTS (1)	ON	Satellite observables data
$PSTMSAT (1)	ON	Reports satellite iono and differential data
$PSTMKFCOV	OFF	Standard deviation and covariance
$PSTMCPU	ON	Reports the CPU usage and CPU speed setting
$PSTMPPSDATA	OFF	Reports the pulse per second data
$PSTMTRAIMSTATUS	OFF	Reports the TRAIM status data
$PSTMTRAIMUSED	OFF	Reports the satellites used for timing correction
$PSTMTRAIMRES	OFF	Reports the residuals for used satellites
$PSTMTRAIMREMOVED	OFF	Reports the satellites removed by timing correction algorithm
$PSTMSBAS	ON	Transmits WAAS/SBAS information about the tracked SBAS satellites

Syntax	Default	Description
$PSTMSBASM	OFF	Output the SBAS satellite message frame
$PSTMNAVM	OFF	Output the navigation message frame
$PSTMIFBRES	ON	Output the IFB residuals
$PSTMUTC	OFF	Output the UTC time, date and time offset parameters
$PSTMCHMON	ON	Reports correlation for a given satellite
$PSTMIFBESTDATA	ON	
$PSTMTEMP	OFF	Output the temperature parameters
$PSTMBIASDATA	OFF	
$PSTMPV	OFF	Provides position, velocity and root square of covariance matrix values for position and velocity
$PSTMPVRAW	OFF	Provides not filtered position and velocity, and LMS fix related info
$PSTMPVQ	OFF	Provides position and velocity processing noise matrix values
$PSTMEPE	OFF	Provides estimated horizontal and vertical position error
$PSTMSPOOF	OFF	Provides information on the anti-spoofing algorithm status
$PSTMPA	OFF	Position algorithm
$PSTMRF	OFF	satellite signal data for each tracked satellite
$PSTMGSA	OFF	NMEA: GPS DOP and active satellites
$PSTMRFAGC	OFF	Provides RF AGC status
$PSTMNMASTATUS	OFF	OSNMA status
$PSTMNMAMACV	OFF	OSNMA event for MAC authentication status
$PSTMNMADSMV	OFF	OSNMA event for DSM KROOT validation status
$PSTMNMAPKRV	OFF	OSNMA event for DSM PKR validation status
$PSTMNMAAM	OFF	OSNMA alert message
$PSTMPD	ON	Reports constellation group delay from GPS
$PSTMRSS	OFF	Provides receiver system status.
$PSTM--GSV	On	NMEA $--GSV satellites in view extension
$PSTMALMANAC	Reply	ST: Dump almanac <data> when available
$PSTMEPHEM	Reply	ST: Dump ephemeris <data> when available

1. *More details on the usage of this command are in (2)*

3.3 Commands answers messages list

Table 89. Commands answers messages list

Syntax	Default	Description
$PSTMALMANAC	Reply	ST: Dump almanac <data>
$PSTMEPHEM	Reply	ST: Dump ephemeris <data>
$PSTMIONOPARAMS	Reply	ST: Dump iono <data>
$PSTMGLONASSICB	Reply	Dump GLONASS ICB
$PSTMIFB	Reply	Dump the IFB(s)
$PSTMIFBRES	Reply	Dump the IFB residuals
$PSTMFEDUMP	Reply	Dump the front-end registers
$PSTMVER	Reply	Return a version string of the libraries

Syntax	Default	Description
$PSTMRTCTIME	Reply	Displays RTC time

3.4 Sequence to change NVM parameter

Sequence example to change the value of configuration data block [P32,L00,F01] [b08-b11] to value “3”.

$PSTMSETPAR, 32,0,1,0,3

$PSTMSAVEPAR

$PSTMSRR

For more about CDB-ID parameters, refer to UM3428 document.

3.5 Preliminary notes about satellites’ PRN ranges

The satellite PRN is an ID used to identify satellites. Starting from Rev 4.11 more constraints about this info have been added. Thus, PRN ranges depend on NMEA revision in use.

PRN ranges for proprietary $PSTMxxx messages are:

- GPS from 1 to 32
- GPS L1C from 1301 to 1332
- GLONASS from 65 to 99
- SBAS from 120 to 140 and from 371 to 388
- BEIDOU from 141 to 177 and from 978 to 1003
  - Note that for input, it is also supported from 941 to 1003
- BEIDOU B1C from 1541 to 1603
- QZSS from 183 to 202
- QZSS L1C from 223 to 232
- GALILEO from 301 to 336
- IRNSS from 801 to 814
- ID’s for multifrequency are:
  - GPS L2C from 401 to 432
  - GLONASS G2 from 465 to 499
  - BEIDOU B2i from 541 to 599
  - BEIDOU B2a from 851 to 913
  - BEIDOU B3i from 1041 to 1103
  - BEIDOU B2b from 1141 to 1203
  - GPS L5 from 501 to 532
  - GALILEO E5a from 601 to 636
  - GALILEO E5b from 651 to 686
  - GALILEO E6 from 701 to 736
  - QZSS L2C 203 to 212
  - QZSS L5C 213 to 222
  - QZSS L1S 183 to 191

NMEA Rev 4.11, PRN ranges are:

- GPS from 1 to 32
- SBAS from 33 to 64
- GLONASS from 65 to 99
- BEIDOU from 1 to 64
- QZSS from 1 to 10
- GALILEO from 1 to 36
- IRNSS from 1 to 14

Signal channels are distinguished by the signal ID.

3.6 Standard NMEA messages specification

These messages are defined within the “NMEA 0183” specification.

3.6.1 $--GNS

Fix data for single or combined satellite navigation system (GNSS).

NMEA message list bitmask (64 bits): 0000 0000 0000 0001

Synopsis:

```
$<TalkerID>GNS,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<Mode>,
<Sats>,<HDOP>,<AltVal>,<GEOVal>,<DGPSAge>,<DGPSRef>,<Nav_Status>*<checksum><cr><lf>
```

Table 90. $--GNS field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) .sss: Decimal fraction of seconds (variable length) *Note: That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*
Lat	DDMM.MMMMM	Latitude as degrees: DD: Degree (fixed two digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
N/S	“N” or “S”	Latitude direction: north or south
Long	DDDMM.MMMMM	Longitude as degrees: DDD: Degree (fixed three digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
E/W	“E” or “W”	Longitude direction: east or west
Mode indicator	Char or string	In case of single constellation this is a character which can assume these values: N = NO fix A = Autonomous D = Differential GPS E = Estimated (dead reckoning mode) In multi-constellation mode this is a 5-letter string where each letter is the mode indicator of each constellation in this order: GPS, GLONASS, GALILEO, BEIDOU, QZSS, IRNSS
Sats	Decimal, 2 digits	Satellites in use: example: 8

Parameter	Format	Description
HDOP	Decimal, 3 digits	Horizontal dilution of precision, max: 99.0
Alt	Float	Height above mean sea level (geoid), max: 100000 m Number of digits after the decimal point configurable (firmware config page 32 line 0 field 0). Firmware config page 32 line 0 field 0: Is 0 then minimum number of characters is 1. Is not 0 then minimum number of characters is 2 + (number of digits after decimal point).
GEOSep	Decimal, 4 digits	Geoidal separation, meter
DGNSSAge	Empty field	Not supported
DGNSSRef	Empty field	Not supported
Nav_status	“S”, “C”, “U” or “V”	Navigational status indicator: “S” = Safe “C” = Caution “U” = Unsafe “V” = Not valid

Note: In case of single constellation set up the mode indicator consists in one character and the information about the constellation is given by talker id.

Synopsis:

```
$GNGNS,150733.000,4800.60526,N,00013.32783,E,AAAANN,38,0.5,096.15,47.5,,,C*33
```

3.6.2 $--GGA

Global positioning system fixed data.

NMEA message list bitmask (64 bits): 0000 0000 0000 0002

Synopsis:

```
$<TalkerID>GGA,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<GPSQual>,
 <Sats>,<HDOP>,<Alt>,<AltVal>,<GeoSep>,<GeoVal>,<DGPSAge>,<DGPSRef>*
<checksum><cr><lf>
```

Table 91. $--GGA field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) .sss: Decimal fraction of seconds (variable length)

Parameter	Format	Description
		Note: That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.
Lat	DDMM.MMMMM	Latitude as degrees: DD: Degree (fixed two digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
N/S	“N” or “S”	Latitude direction: north or south
Long	DDDMM.MMMMM	Longitude as degrees: DDD: Degree (fixed three digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
E/W	“E” or “W”	Longitude direction: east or west
GPSQual	Decimal, 1 digit	0 = Fix not available or invalid 1 = GPS, SPS mode, fix valid 2 = Differential GPS, SPS mode, fix valid 6 = Estimated (dead reckoning) mode
Sats	Decimal, 2 digits	Satellites in use: example: 8
HDOP	Decimal, 3 digits	Horizontal dilution of precision, max: 99.0
Alt	Float.	Height above mean sea level (geoid), max: 100000 m Number of digits after the decimal point configurable (firmware config page 32 line 0 field 0). Firmware config page 32 line 0 field 0: is 0 then minimum number of characters is 1. is not 0 then minimum number of characters is 2 + (number of digits after decimal point).
AltVal	“M”	Reference unit for altitude (“M” = meters)
GeoSep	Decimal, 4 digits	Geoidal separation measure in “M” = meters
GeoVal	“M”	Reference unit for GeoSep (“M” = meters)
DGPSAge	Empty	Not supported
DGPSRef	Empty	Not supported

Example:

```
$GPGGA,183417.000,4814.03970,N,01128.52205,E,0,00,99.0,495.53,M,47.6,M,,*53
```

3.6.3 $--GSA

GNSS DOP and active satellites. Satellites from different constellations are sent on separate messages.
NMEA message list bitmask (64 bits): 0000 0000 0000 0004

Synopsis:

```
$<TalkerID>GSA,<Mode>,<CurrentMode>,<SatPRN1>,...,<SatPRNn>,<PDOP>,<HDOP>,<VDOP>,<SystemID>*<
checksum><cr><lf>
```

Table 92. $--GSA field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode

Parameter	Format	Description
		GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
Mode	1 character	M = Manual, forced to operate in 2D or 3D mode (manual mode not supported) A = Automatic, allowed to automatically switch 2D/3D
CurrentMode	Decimal, 1 digit	Current mode: 1 = Fix not available or invalid 2 = GPS, SPS Mode, fix valid 3 = Differential GPS, SPS mode, fix valid
SatxPRN	Decimal, up to 3 digits	Satellites list used for positioning. See chapter 6.5 for more info about available values.
PDOP	x.x, variable length field	Position dilution of precision, max: 99.0
HDOP	x.x, variable length field	Horizontal dilution of precision, max: 99.0
VDOP	x.x, variable length field	Vertical dilution of precision, max: 99.0
SystemID	Hexadecimal, 1 digit	NMEA 4.11 only: The system ID of this message: 1 = GPS 2 = GLONASS 3 = GALILEO 4 = BEIDOU 5 = QZSS 6 = IRNSS

Example:

```
$GNGSA,A,3,22,03,14,01,28,08,27,10,32,17,,,0.7,0.4,0.6,1*32
$GNGSA,A,3,90,78,87,71,,,,,,,,0.7,0.4,0.6,2*38
$GNGSA,A,3,,,,,,,,,,,,,0.7,0.4,0.6,5*30
$GNGSA,A,3,03,15,27,30,05,09,,,,,,,0.7,0.4,0.6,3*3B
$GNGSA,A,3,30,27,33,41,16,32,14,06,39,,,,0.7,0.4,0.6,4*3D
$GNGSA,A,3,03,14,01,08,10,32,,,,,,,0.7,0.4,0.6,1*3B
$GNGSA,A,3,03,15,27,30,09,,,,,,,,0.7,0.4,0.6,3*3E
$GNGSA,A,3,30,27,33,41,32,39,,,,,,,0.7,0.4,0.6,4*39
```

3.6.4 $--GSV

GNSS satellites in view.

Usually, GSV messages are organised per constellation and each message carries information about up to 4 satellites in view. Thus, in certain cases, to describe all the satellites in view from a constellation more than a message is needed. This set of messages is printed once per each constellation with talker ID related to described constellation.

NMEA message list bitmask (64 bits): 0000 0000 0008 0000

Synopsis:

```
$--GSV,<GSVAmount>,<GSVNumber>,<TotSats>,<Sat1PRN>,<Sat1Elev>,
<Sat1Azim>,<Sat1CN0>,…,<Sat4PRN>,<Sat4Elev>,<Sat4Azim>,<Sat4CN0>,
<SignalID>*<checksum><cr><lf>
```

Table 93. **$--GSV field description**

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters). GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode
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, up to 3 digits	Satellite's list used for positioning. See Section 3.5: Preliminary notes about satellites' PRN ranges for more info about available values
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
SatxCN0	Decimal, 2 digits	Carrier to noise ratio for satellite x in dB, 00 … 99
SignalID	Hexadecimal, 1 digit	An identifier to indicate the signal in use.

Example:

```
$GPGSV,3,1,10,15,79,235,48,13,56,123,48,24,53,275,46,23,34,296,39,1*6B
$GPGSV,3,2,10,20,34,280,42,14,32,051,42,17,23,094,46,19,15,120,43,1*6B
$GPGSV,3,3,10,12,14,206,41,05,09,187,39,,,,,,,,,1*6A
$GLGSV,2,1,05,73,54,175,45,68,52,316,48,71,47,243,48,65,16,047,37,1*77
$GLGSV,2,2,05,92,12,158,38,,,,,,,,,,,,1*42
$GAGSV,1,1,04,01,72,076,48,04,69,167,47,09,49,303,44,31,46,213,45,7*73
$GBGSV,2,1,07,842,50,289,45,821,49,198,48,814,30,311,39,810,19,046,39,1*7F
$GBGSV,2,2,07,805,11,115,36,829,08,102,41,822,08,162,42,,,,,1*72
$GPGSV,1,1,03,24,,,48,23,,,44,14,,,48,,,,,8*6C
$GAGSV,1,1,04,31,,,47,09,,,46,04,,,49,01,,,50,1*76
$GBGSV,1,1,03,29,,,44,21,,,48,22,,,43,,,,,5*72
```

3.6.5 $--RMC

Recommended minimum specific GPS/transit data. Time, date, position and speed data provided by the GNSS receiver. This sentence is transmitted at intervals not exceeding 2 seconds and is always accompanied by RMB when destination waypoint is active.

NMEA message list bitmask (64 bits): 0000 0000 0000 0040

Synopsis:

```
$<TalkerID>RMC,<Timestamp>,<Status>,<Lat>,<N/S>,<Long>,<E/W>,<Speed>,
<Trackgood>,<Date>,<MagVar>,<MagVarDir>,<mode>,
<Nav_status>*<checksum><cr><lf>
```

Table 94. **$--RMC field description**

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters). GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode

Parameter	Format	Description
		GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode.
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) .sss: Decimal fraction of seconds (variable length) *Note: That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*
Status	“A” or “V”	Receiver warning: “A” = valid, “V” = warning *Note: “V” is reported in NO FIX conditions and “A” is reported in 2D and 3D fix conditions.*
Lat	DDMM.MMMMM	Latitude as degrees: DD: Degree (fixed two digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
N/S	“N” or “S”	Latitude direction: north or south
Long	DDDMM.MMMMM	Longitude as degrees: DDD: Degree (fixed three digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
E/W	“E” or “W”	Longitude direction: east or west
Speed	x.x, variable length field	Speed over ground in knots
Trackgood	x.x, variable length field	Course made good, max. 999.9
Date	Decimal, 6 digits	Date of fix: ddmmyy
MagVar	Decimal, 4 digits	Magnetic variation, max.: 090.0
MagVarDir	“E” or “W”	Magnetic variation direction
Mode	“D”, “A”, “N” or “E”	Positioning system mode indicator: “D” = Differential mode “A” = Autonomous mode “N” = Data not valid “E” = Estimated (dead reckoning) mode
Nav_status	“S”, “C”, “U” or “V”	Navigational status indicator: “S” = Safe “C” = Caution “U” = Unsafe “V” = Not valid

Example:

```
$GNRMC,081623.000,A,4759.63403,N,00011.13573,E,0.1,0.0,220321,,,A,C*1E
```

3.6.6 $--VTG

Course over ground and ground speed, this message provides the actual course and speed relative to ground.
NMEA message list bitmask (64 bits): 0000 0000 0000 0010

Synopsis:

```
$<TalkerID>VTG,<TMGT>,T,<TMGM>,M,<SoGN>,N,<SoGK>,K,<D>*<checksum><cr><lf>
```

Table 95. $--VTG field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters). GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
TMGT	ddd.d in degrees	Track in reference to “true” earth poles (empty if no fix)
T		Indicates “terrestrial”
TMGM	ddd.d in degrees	Track in reference to “magnetic” earth poles (always empty)
M		Indicates “magnetic”
SoGN	ddd.d in knots	Speed over ground in knots (empty if no fix)
N		Indicates “knots”
SoGK	ddd.d in km/h	Speed over ground in kilometers per hour (empty if no fix)
K		Indicates “kilometres”
D	char	Mode indicator: A= Autonomous mode D= Differential mode E= Estimated mode N = Not valid mode

Example:

```
$GPVTG,73.2,T,,M,0.2,N,0.4,K,D*50
```

3.6.7 $--GST

Global positioning system pseudorange noise statistics.
NMEA message list bitmask (64 bits): 0000 0000 0000 0008

Synopsis:

```
$<TalkerID>GST,<Timestamp>,<EHPE>,<Semi-major>,<Semi-minor>,<Angle>,<LatErr>,<LonErr>,<Alt Er
r Dev>*<checksum><cr><lf>
```

Table 96. $--GST field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode

Parameter	Format	Description
		GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) sss: Decimal fraction of seconds (variable length) *Note: That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*
EHPE	dd.d in m	Equivalent horizontal position error
Semi-major	dd.d in m	Standard deviation (meters) of semi-major axis of error ellipse
Semi-minor	dd.d in m	Standard deviation (meters) of semi-minor axis of error ellipse
Angle	dd.d in degree	Orientation of semi-major axis of error ellipse (true north degrees)
LatErr	dd.d in m	Standard deviation (meters) of latitude error
LonErr	dd.d in m	Standard deviation (meters) of longitude error
AltErr	dd.d in m	Standard deviation (meters) of altitude error

Example:

```
$GNGST,081623.000,1.0,3.1,2.8,-0.2,3.1,2.7,2.8*5C
```

3.6.8 $--GBS

GNSS satellite fault detection

NMEA message list bitmask (64 bits): 0000 2000 0000 0000

Synopsis:

```
$<TalkerID>GBS,<Timestamp>,<LatErr>,<LonErr>,<AltErr>,<SatPRN>,<Prob>, <Res>,<StdDev>,<System ID>,<SignalID>*<checksum><cr><lf>
```

Table 97. $--GBS field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits)

Parameter	Format	Description
		sss: Decimal fraction of seconds (variable length) (1)
LatErr	dd.d in m	Standard deviation (meters) of latitude error
LonErr	dd.d in m	Standard deviation (meters) of longitude error
AltErr	dd.d in m	Standard deviation (meters) of altitude error
SatPRN	Decimal, 2 digits	PRN number of most likely failed satellite This satellite is excluded by RAIM or FDE algorithm
Prob	Empty	Probability of missed detection for most likely failed satellite Not supported
Res	dd.d in m	Range residual of most likely failed satellite
StdDev	Empty	Standard deviation of bias estimate Not supported
SystemID	Hexadecimal, 1 digit	The system ID of this message: 1 = GPS 2 = GLONASS 3 = GALILEO 4 = BEIDOU 5 = QZSS 6 = IRNSS
SignalID	Hexadecimal, 1 digit	An identifier to indicate the signal in use.

1. *That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*

Example:

```
$GNGBS,081624.000,3.1,2.7,2.8,,,,,,*6B
```

3.6.9 $--GLL

Geographic positioning latitude / longitude.

NMEA message list bitmask (64 bits): 0000 0000 0010 0000

Synopsis:

```
$<TalkerID> GLL<Lat>,<N/S>,<Long>,<E/W>,<Timestamp>,<Status>,<mode indicator>*<checksum><cr><lf>
```

3.6.10 $--DTM

Datum reference. Local geodetic datum and datum offsets from a reference.

NMEA message list bitmask (64 bits): 0000 0080 0000 0000

Synopsis:

```
$--DTM,<LocDatCode>,<LocDatSubDiv>,<LatOffset>,<N/S>,<LonOffset>,<E/W>, <AltOffset>,<RefDatCode>*<checksum><cr><lf>
```

Table 98. $--DTM field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode

Parameter	Format	Description
		GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
LocDatCode	Char, 3 digits	Local datum code
LocDatSubDiv	Decimal, 3 digits	Local datum subdivision code
LatOffset	MM.MMMMM	Latitude offset MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
N/S	“N” or “S”	Latitude direction: north or south
LonOffset	MM.MMMMM	Longitude offset MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
E/W	“E” or “W”	Longitude direction: east or west
AltOffset	Float	Altitude offset, meters
RefDatCode	Char, 3 digits	Reference datum code

3.6.11 $--ZDA

Time and date. UTC, day, month, year and local time zone.

NMEA message list bitmask (64 bits): 000 000 0100 0000

Synopsis:

```
$--ZDA,<UTC>,<Day>,<Month>,<Year>,<LocalZoneHour>,<LocalZoneMin>*<checksum><cr><lf>
```

Table 99. $--ZDA field description

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (Fixed two characters) GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode GN: If system works in multi-constellation mode
UTC	hhmmss.sss	UTC time of GPS sample hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) .sss: Decimal fraction of seconds (variable length) (1)
Day	Decimal, 2 digits	Local datum subdivision code
Month	Decimal, 2 digits	Latitude offset MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
Year	Decimal, 4 digits	Latitude direction: north or south
LocalZoneHour	Empty	Local zone hour. Not supported.

Parameter	Format	Description
LocalZoneMin	Empty	Local zone minutes. Not supported.

1. *That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*

3.7 ST NMEA messages specification

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

There are two sorts of “proprietary messages” within an STMicroelectronics -GNSS system. They are either sent repeatedly with a defined or definable reporting rate or they are sent only once as a reaction to a command.

3.7.1 $PSTMRF

Satellite signal data for each tracked satellite.

For more information, refer to UM3397.

3.7.2 $PSTMTG

Time and global Information.

For more information, refer to UM3397.

3.7.3 $PSTMTS

This message reports the satellite observables data.

For more information, refer to UM3397.

3.7.4 $PSTMSAT

This message is repeated for each satellite tracked and contains satellite iono and differential data.

For more information, refer to UM3397.

3.7.5 $PSTMCHMON

This message reports the carrying correlation for a given satellite.

For more information, refer to UM3397.

3.7.6 $PSTMMG

This message reports the time week number, time of week, validity and constellation type of the current best time, in addition to signal and observation’s quality metrics masks

For more information, refer to UM3397.

3.7.7 $PSTMMS

This message reports the valid signal and observation quality metrics for each tracked satellite.

For more information, refer to UM3397.

3.7.8 $PSTMPRES

Position residual

Note: `$PSTMPRES` and `$PSTMVRES` are always enabled together.

NMEA message list bitmask (64 bits): 0000 0000 0000 1000

Synopsis:

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

N = number of tracked satellites

Table 100. **$PSTMPRES field description**

Parameter	Format	Description
RMSpos	dd.d	Position “rms” residual for the fix
resx	dd.d	Residual of tracked satellite x (corresponds to x satellite in $--GSA message)
*	Fixed character	Delimiter of datafield

Example:

```
$PSTMPRES,2.9,0.8,0.6,0.1,-0.3,9.1,-1.4,-0.2,-1.5,-0.8,3.5,1.6,1.2,0.2,0.2,-0.5,3.3,-1.9,0.4,
-2.1,0.4,0.3,-8.2,-1.1,-1.2,0.2,4.2,,,,,,,,,,,,,*27
```

3.7.9 $PSTMVRES

Velocity residual

Note: $PSTMPRES and $PSTMVRES are always enabled together.

NMEA message list bitmask (64 bits): 0000 0000 0000 1000

Synopsis:

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

N = number of tracked satellites

Table 101. **$PSTMVRES field description**

Parameter	Format	Description
RMSvel	dd.d	Velocity “rms” residual for the fix (m/s)
vresx	dd.d	Residual of tracked satellite x (m/s) (corresponds to x satellite in $--GSA message)
*	Fixed character	Delimiter of datafield

Example:

```
$PSTMVRES,0.0,-0.0,-0.0,0.0,-0.0,0.0,0.0,-0.0,-0.0,0.0,-0.0,0.0,-0.0,0.0,-0.0,0.0,-0.0,0.0,0.
1,0.0,0.0,0.0,0.0,0.0,-0.0,-0.0,0.0,,,,,,,,,,,,,*26
```

3.7.10 $PSTMCPU

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

NMEA message list bitmask (64 bits): 0000 0000 0080 0000

Synopsis:

```
$PSTMCPU,<CPU_Usage>,-1,<CPU_Speed>*<checksum><cr><lf>
```

Table 102. **$PSTMCPU field description**

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 -1: Not supported
CPU_Speed	Decimal, 1 digit	CPU clock frequency: example: 261 MHz

3.7.11 $PSTMPPSDATA

Reports the pulse per second data

NMEA message list bitmask (64 bits): 000 000 0020 0000

Synopsis:

```
$PSTMPPSDATA,<on_off>,<pps_valid>,<synch_valid>,<out_mode>,<ref_time>,<ref_constellation>,<pulse_duration>,<pulse_delay>,<gps_delay>,<glo_delay>,<bei_delay>,<gal_delay>,<inverted_polarity>,<fix_cond>,<sat_th>,<elev_mask>,<const_mask>,<ref_sec>,<fix_status>,<used_sats>,<gps_utc_delta_s>,<gps_utc_delta_ns>,<glonass_utc_delta_ns>,<galileo_utc_delta_ns>,<quantization_error>,<pps_clock_freq>,<tcxo_clock_freq>,<compass_utc_delta_time_s>*<checksum><cr><lf>
```

Table 103. $PSTMPPSDATA field description

Parameter	Format	Description
on_off	Decimal, 1 digit	PPS signal ON/OFF status 0: OFF 1: ON
pps_valid	Decimal, 1 digit	Global PPS validity flag 0: PPS not valid 1: PPS valid
synch_valid	Decimal, 1 digit	PPS synchronization validity 0: Not valid 1: Valid
out_mode	Decimal, 1 digit	0 = PPS_OUT_MODE_ALWAYS 1 = PPS_OUT_MODE_ON_EVEN_SECONDS 2 = PPS_OUT_MODE_ON_ODD_SECONDS
ref_time	Decimal, 1 digit	0 = UTC 1 = GPS_UTC (GPS Time) 2 = GLONASS_UTC (GLONASS Time) 3 = UTC_SU(1) 4 = GPS_UTC_FROM_GLONASS(2) *Note: If the software is configured to work in GLONASS only mode, UTC(SU) is identical to UTC and GPS_UTC_FROM_GLONASS is identical to GPS_UTC.*
ref_constellation	Decimal, 1 digit	0 = GPS 1 = GLONASS *Note: The reference constellation reports which reference time has been used for the PPS generation.*
pulse_duration	Double	Pulse duration [s]
pulse_delay	Decimal	Pulse delay [ns]
gps_delay	Decimal	GPS path RF delay [ns]
glo_delay	Decimal	GLONASS path RF delay [ns]
bei_delay	Decimal	BEIDOU path RF delay [ns] *Note: This parameter is always zero if Beidou constellation is not supported by the hardware platform.*
gal_delay	Decimal	GALILEO path RF delay [ns]
inverted_polarity	Decimal, 1 digit	Pulse polarity inversion:

Parameter	Format	Description
		0 = Not inverted 1 = Inverted
fix_cond	Decimal, 1 digit	Selected GNSS fix condition for PPS signal generation: 1 = NO_FIX. 2 = 2D_FIX. 3 = 3D_FIX. 4 = RTK_FIXED_FIX 5 = RTK_FLOAT_FIX 6 = PPS_MODE_FIX
sat_th	Decimal	Selected minimum number of satellites for PPS signal generation.
elev_mask	Decimal	Selected minimum satellite elevation for time correction.
const_mask	Decimal	Selected constellations for time correction.
ref_sec	Decimal, 2 digits	Second at which the reported PPS data is applied. According to the reference time configuration it could be a UTC or a GPS or a GLONASS time second.
fix_status	Decimal, 1 digit	GNSS position fix status when the time has been corrected.
used_sats	Decimal	Used satellites for time correction.
gps_utc_delta_s	Decimal	UTC leap seconds [s]
gps_utc_delta_ns	Decimal	UTC – GPS delta time [ns]
glonass_utc_delta_ns	Decimal	UTC – GLONASS delta time [ns]
galileo_utc_delta_ns	Decimal	UTC – GALILEO delta time [ns]
quantization_error	Double (scientific notation format)	Quantization error [s].
pps_clock_freq	Double, 2 fractional digits	PPS clock frequency [Hz]
tcxo_clock_freq	Double, 2 fractional digits	TCXO clock frequency [Hz]
compass_utc_delta_time_s	Decimal	UTC – BeiDou leap second [s]

1. *UTC(SU) is the Soviet Union UTC, it is derived from GLONASS time applying the UTC delta time downloaded from GLONASS satellites.*
2. *GPS_UTC_FROM_GLONASS is the GPS time derived from GLONASS time applying the GPS delta time downloaded from GLONASS satellites.*

3.7.12 $PSTMTRAIMSTATUS

Reports the TRAIM algorithm status.

Note: All TRAIM related messages are enabled/disabled all together by the same mask.

NMEA message list bitmask (64 bits): 0000 0000 0200 0000

Synopsis:

```
$PSTMTRAIMSTATUS,<on_off>,<traim_solution>,<alarm>,<ave_error> ,<used_sats>,<removed_sats>,<ref_second>*<checksum><cr><lf>
```

Table 104. $PSTMTRAIMSTATUS field description

Parameter	Format	Description
on_off	Decimal, 1 digit	TRAIM ON/OFF status 0: OFF 1: ON
traim_solution	Decimal, 1 digit	TRAIM algorithm status:

Parameter	Format	Description
		0 = UNDER alarm 1 = OVER alarm 2 = UNKNOWN
alarm	Decimal	Time error threshold [ns]
ave_error	Decimal	Average time error [ns]
used_sats	Decimal	Number of used satellites
removed_sats	Decimal	Number of removed satellites
ref_second	Decimal	Second at which the PPS signal is generated based on reported TRAIM status

3.7.13 $PSTMTRAIMUSED

Reports the satellite used for timing correction.

Note: All TRAIM related messages are enabled/disabled all together by the same mask.

NMEA message list bitmask (64 bits): 0000 0000 0200 0000

Synopsis:

```
$PSTMTRAIMUSED,<on_off>,<used_sats>,<sat1>,…,<satN>*<checksum><cr><lf>
```

Table 105. $PSTMTRAIMUSED field description

Parameter	Format	Description
on_off	Decimal, 1 digit	TRAIM ON/OFF status 0: OFF 1: ON
used_sats	Decimal	Number of used satellites
Sat1..satN	Decimal	Used satellites list

3.7.14 $PSTMTRAIMRES

Reports the time error residuals for satellites used for timing correction.

Note: All TRAIM related messages are enabled/disabled all together by the same mask.

NMEA message list bitmask (64 bits): 0000 0000 0200 0000

Synopsis:

```
$PSTMTRAIMRES,<on_off>,<used_sats>,<res1>,…,<resN>*<checksum><cr><lf>
```

Table 106. $PSTMTRAIMRES field description

Parameter	Format	Description
on_off	Decimal, 1 digit	TRAIM ON/OFF status: 0: OFF 1: ON
used_sats	Decimal	Number of used satellites.
res1..resN	Decimal	Time error residuals for satellites reported in the TRAIMUSED message. Each residual refers to the satellite in the same message position.

3.7.15 $PSTMDRSENMSG

Reports DR sensor message data, which is specific to the message id for each specific DR sensor configuration.

NMEA message list bitmask (64 bits): 1000 0000 0000 0000

Synopsis:

```
$PSTMDRSENMSG,3,<cpu timestamp>,<odometer>,<reverse status>*<checksum><cr><lf>
```

Table 107. $PSTMDRSENMSG in case of MsgID = 3 field description

Parameter	Format	Description
Message id	Decimal, 2 digits	Data type
Cpu timestamp	Decimal, 10 digits	Microseconds
Odometer	Decimal, 5 digits	Unsigned odometer count
Reverse status	Boolean	0 = Forward 1 = Reverse

Synopsis:

```
$PSTMDRSENMSG,24,<cpu timestamp>,<temperature>, <validity>*<checksum><cr><lf>
```

Table 108. $PSTMDRSENMSG in case of MsgID = 24 field description

Parameter	Format	Description
Message id	Decimal, 2 digits	Data type
cpu timestamp	Decimal, 10 digits	Microseconds
Temperature	Decimal	Gyro sensor temperature DR version < 4.7.1: 1°C/LSB DR version >= DR 4.7.1: fix point value format 8.8. The floating point value is recovered by dividing by 256
Validity	Boolean	0 = Temperature is not valid 1 = Temperature is valid

Synopsis:

```
$PSTMDRSENMSG,30,<cpu timestamp>,<raw_x>,<raw_y>,<raw_z>*<checksum><cr><lf>
```

Table 109. $PSTMDRSENMSG in case of MsgID = 30 field description

Parameter	Format	Description
Cpu timestamp	Decimal, 10 digits	Microseconds
raw_x	Decimal, 5 digits	Raw signed 16-bit X-axis acceleration
raw_y	Decimal, 5 digits	Raw signed 16-bit Y-axis acceleration
raw_z	Decimal, 5 digits	Raw signed 16-bit Z-axis acceleration

Synopsis:

```
$PSTMDRSENMSG,31,<cpu timestamp>,<raw_x>,<raw_y>,<raw_z>*<checksum><cr><lf>
```

Table 110. $PSTMDRSENMSG in case of MsgID = 31 field description

Parameter	Format	Description
Cpu timestamp	Decimal, 10 digits	Microseconds
raw_x	Decimal, 5 digits	Raw signed 16-bit X-axis angular rate
raw_y	Decimal, 5 digits	Raw signed 16-bit Y-axis angular rate

Parameter	Format	Description
raw_z	Decimal, 5 digits	Raw signed 16-bit Z-axis angular rate

3.7.16 $PSTMTRAIMREMOVED

Reports the satellite removed by the timing correction algorithm.

Note: All TRAIM related messages are enabled/disabled all together by the same mask.

NMEA message list bitmask (64 bits): 0000 0000 0200 0000

```
$PSTMTRAIMUSED,<on_off>,<removed_sats>,<sat1>,…,<satN>*<checksum><cr><lf>
```

Synopsis:

Table 111. $PSTMTRAIMREMOVED field description

Parameter	Format	Description
on_off	Decimal, 1 digit	TRAIM ON/OFF status 0: OFF 1: ON
removed_sats	Decimal	Number of removed satellites
Sat1..satN	Decimal	Removed satellites list

3.7.17 $PSTMKFCOV

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

NMEA message list bitmask (64 bits): 0000 0000 0800 0000

Synopsis:

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

Table 112. $PSTMKFCOV field description

Parameter	Format	Description
PosStd	ddd.d	Standard deviation of position in meters
PosNcov	ddd.d	Covariance (north/south) in m² (from kalman filter)
PosEcov	ddd.d	Covariance (east/west) in m² (from kalman filter)
PosVcov	ddd.d	Covariance (vertical) in m² (from kalman filter)
VelStd	ddd.d	Standard deviation of velocity in meter/second
VelNcov	ddd.d	Covariance (north/south) in m²/s (from kalman filter)
VelEcov	ddd.d	Covariance (east/west) in m²/s (from kalman filter)
VelVcov	ddd.d	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
```

3.7.18 $PSTMANTENNASTATUS

NMEA message list bitmask (64 bits): 0000 0010 0000 0000

This message reports most relevant information about the Antenna Detection module.

Synopsis:

```
$PSTMANTENNASTATUS,<ant_status>,<op_mode>,<rf_path>,<pwr_switch>*<checksum><cr><lf>
```

Table 113. $PSTMANTENNASTATUS field description

Parameter	Format	Description
ant_status	Decimal	Current antenna status 0 = Not initialized 1 = Normal condition 2 = Open condition 3 = Short ground condition 4 = Thermal shutdown condition 5 = Reverse current/short VCC condition 6 = Overcurrent condition
op_mode	Decimal	Current antenna detection operating mode 0 = Automatic mode 1 = Manual mode
rf_path	Decimal	Current RF path 0 = Not initialized 1 = External antenna 2 = Internal antenna
pwr_switch	Decimal	Current antenna power status 0 = Antenna power is off 1 = Antenna power is on

3.7.19 $PSTMSBAS

SBAS satellite data.

NMEA message list bitmask (64 bits): 0000 0000 0000 4000

Synopsis:

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

Arguments:

Table 114. $PSTMSBAS field description

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 2 = SBAS satellite tracked and decoded. Differential Mode ON
SatID	Decimal, 3 digits	SBAS satellite ID

Parameter	Format	Description
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 dBHz)

Example:

```
$PSTMSBAS,1,2,123,27,140,44*1A
```

3.7.20 $PSTMSBASM

Output the SBAS satellite message frame.

NMEA message list bitmask (64 bits): 0000 8000 0000 0000

Synopsis:

```
$PSTMSBASM,<SatID>,<Msg>*<checksum><cr><lf>
```

Arguments:

Table 115. $PSTMSBASM field description

Parameter	Format	Description
SatID	Decimal, 3 digits	SBAS Satellite ID
Msg	Hexadecimal, 64 digits	250-bit SBAS Message Frame padded with six 0 bits (32 Bytes encoded on 64 hexadecimal digits)
*	Fixed Character	Delimiter for data field

Example:

```
$PSTMSBASM,120,c660003fcbfa4000000003957bb9801e5bf3fe02007412fd82bfafee9c8a7040*22
```

This frame can be activated through the FW configuration with the following NMEA commands:

```
$PSTMSETPAR,32,6,0,1,0x8000
$PSTMSAVEPAR
$PSTMSRR
```

3.7.21 $PSTMNAVM

Navigation data frame.

NMEA message list bitmask (64 bits): 0000 0020 0000 0000

Synopsis:

```
$PSTMNAVM,<msg_id>,<prn>,<nav_frame>*<checksum><cr><lf>
```

Arguments:

Table 116. $PSTMNAVM field description

Parameter	Format	Description
msg_id	Decimal, 1 digit	Message ID: • GPS = 0 • GLONASS = 1 • GALILEO I/NAV PAYLOAD = 3 • BEIDOU = 7 • CNAV-L2-L5 = 9 • IRNSS = 10 • BEIDOU CNAV 2 = 11 • GALILEO FNAV = 12

Parameter	Format	Description
		• GALILEO E6B = 13 • BEIDOU B2B = 14 • GALILEO FULL I/NAV = 15 • GALILEO E5B = 16 • BEIDOU B3I = 17 • GPS L1C = 18 • QZSS L1C = 19
prn	Decimal, 3 digits	Satellite PRN (range: depending on the constellation)
nav_frame	Hexadecimal, up to 80 digits	Navigation data frame (length: depending on the constellation)

Example:

```
$PSTMNAVM,1,83,0DF30363100697D1557CB70CA9852A031A0C100654AE*15
$PSTMNAVM,1,71,0DF30363100697D1557CB70CA9852A031A0C100654AE*18
$PSTMNAVM,1,84,0DF30363100697D1557CB70CA9852A031A0C100654AE*12
$PSTMNAVM,1,90,0DF30363100697D1557CB70CA9852A031A0C100654AE*17
$PSTMNAVM,1,82,0DF30363100697D1557CB70CA9852A031A0C100654AE*14
$PSTMNAVM,1,72,0DF30363100697D1557CB70CA9852A031A0C100654AE*1B
$PSTMNAVM,1,87,0DF30363100697D1557CB70CA9852A031A0C100654AE*11
$PSTMNAVM,1,91,0DF30363100697D1557CB70CA9852A031A0C100654AE*16
```

This frame can be activated through the firmware configuration with the following NMEA commands:

```
$PSTMSETPAR,21,1,0,1,0x10
$PSTMSETPAR,32,6,0,1,0x20
$PSTMSAVEPAR
$PSTMSRR
```

3.7.22 $PSTMIFBRES

Residuals of the IFB(s) estimations done by the receiver for each satellite.

NMEA message list bitmask (64 bits): 8000 0000 0000 0000

Synopsis:

```
$PSTMIFBRES,<ifb_index>,<num_sats>,<sat_id_1>,<ifb_residual_1>, … *<checksum><cr><lf>
```

Arguments:

Table 117. $PSTMIFBRES field description

Parameter	Format	Description
ifb_index	Decimal, 1 digit	Carrier pair: 0 = GPS L2C L1 1 = GLONASS G2 G1 2 = BEIDOU B2I B1I 3 = GPS L5 L1 4 = GALILEO E5a E1 5 = GALILEO E5b E1 6 = GALILEO E6 E1 7 = BEIDOU B2a B1I 8 = GPS L1C L1 9 = BEIDOU B1C B1I 10 = BEIDOU B3I B1I 11 = BEIDOU B2B B1I
num_sats	Decimal, 2 digits	Number of satellites

Parameter	Format	Description
sat_id_x	Decimal, 2 digits	Satellite PRN (Range: depending on the constellation)
ifb_residual_x	Decimal, 3 digits	IFB residual [dm] for the corresponding satellite

Example:

```
$PSTMIFBRES,0,6,9,-0.3,6,56.6,3,22.8,30,-29.1,4,-32.8,7,-17.2*10
$PSTMIFBRES,1,5,69,-53.2,68,25.5,71,23.6,77,-10.1,72,14.2*0D
$PSTMIFBRES,2,2,145,-129.7,152,129.7*38
```

3.7.23 $PSTMIFBESTDATA

Inter-frequency bias (IFB) estimation.

NMEA message list bitmask (64 bits): 0010 0000 0000 0000

Synopsis:

```
$PSTMIFBESTDATA,<ifb_index>,<ifb_est_state>,<sat_type_alarm_mask>,<main_trk_mode_sat_type>,<dual_trk_mode_sat_type>,<ifb_value>,<traim_solution>,<size_of_table>,<num_residuals>,<ifb_count>,<psr_rate_count>,<adr_rate_count>,<ifb_mean>,<ifb_psr_rate>,<ifb_adr_rate>,<thr_mean>,<thr_prs_rate>,<thr_adr_rate>,*<checksum><cr><lf>
```

Arguments:

Table 118. $PSTMIFBESTDATA field description

Parameter	Format	Description
ifb_index	Decimal, 1 digit	Carrier pair: 0 = GPS L2C L1 1 = GLONASS G2 G1 2 = BEIDOU B2I B1I 3 = GPS L5 L1 4 = GALILEO E5a E1 5 = GALILEO E5b E1 6 = GALILEO E6 E1 7 = BEIDOU B2a B1I 8 = GPS L1C L1 9 = BEIDOU B1C B1I 10 = BEIDOU B3I B1I 11 = BEIDOU B2B B1I
ifb_est_state	Decimal, 1 digit	bit0 = global alarm bit1 = IFB estimation updated
sat_type_alarm_mask	Decimal, 1 digit	Constellation alarm mask, for each bit in the mask: 0=OK 1=ALARM Bit 0 = GPS Bit 1 = GLONASS Bit 3 = Galileo Bit 7 = BEIDOU
main_trk_mode_sat_type	Decimal, 1 digit	Main band tracking mode (sat_type only)
dual_trk_mode_sat_type	Decimal, 1 digit	Dual band tracking mode (sat_type only)
ifb_value	Decimal, up to 6 digits	PSR IFB estimate [dm]
traim_solution	Decimal, 1 digit	0 = OK 1 = Alarm 2 = Unknown
size_of_table	Decimal, up to 2 digits	Total number of PSR available
num_residuals	Decimal, up to 2 digits	Total number of PSR used (number of computed residuals)
ifb_count	Decimal, up to 2 digits	IFB number of samples accounted in the exponential moving average (EMA)
psr_rate_count	Decimal, up to 2 digits	PSR IFB-Rate number of samples accounted in the EMA
adr_rate_count	Decimal, up to 2 digits	ADR (carrier-phase) IFB-Rate number of samples accounted in the EMA
ifb_mean	Float	Current value of IFB EMA
ifb_psr_rate	Float	Current value of PSR IFB-Rate EMA

Parameter	Format	Description
ifb_adr_rate	Float	Current value of ADR IFB-Rate EMA
thr_mean	Float	Global alarm threshold on ifb_mean (if configured IFB will be updated automatically)
thr_prs_rate	Float	Global alarm threshold on ifb_psr_rate
thr_adr_rate	Float	Global alarm threshold on ifb_adr_rate

Example:

```
$PSTMIFBESTDATA,0,1,0,0,9,2428,0,5,24,4,60,60,-56.198,-0.46237,0.03814,200.0,3.000,1.000*3F
$PSTMIFBESTDATA,2,1,0,7,7,-411,1,2,24,0,0,0,12.568,-0.43576,0.13851,500.0,51.000,17.000*05
$PSTMIFBESTDATA,5,1,0,3,12,-2598,0,5,24,4,60,60,9.411,0.33952,0.00241,200.0,3.000,1.000*19
```

3.7.24 $PSTMRFI

MCNF, RF AGC, and narrow/wide band jamming status.

NMEA message list bitmask (64 bits): 0004 0000 0000 0000

Synopsis:

```
$PSTMRFI,<index1>,<GPS_TOW>,<detection_mask>,<insertion_mask>,<constel_mask>, <CW_val_mask>,<
CW_candidate_0>,<CW_candidate_1>,<CW_candidate_2>, <CW_candidate_3>,<CW_candidate_4>,<CW_cand
idate_5>,<CW_candidate_6>, <CW_candidate_7>,<CW_candidate_8>,<CW_candidate_9>,<CW_candidate_1
0>, <CW_candidate_11>,<AGC_val_mask>,<AGC_CH1_VGA>,<AGC_CH1_PRE_VGA>, <AGC_CH2_VGA>,<AGC_CH2_
PRE_VGA>,<AGC_CH3_VGA>,<AGC_CH3_PRE_VGA>, <jam_det_mask>,<jam_const_mask>,<NF_DDC1>,<NF_DDC2>
, <NF_DDC3>,<NF_DDC4>*<checksum><cr><lf>
$PSTMRFI,<index2>,<GPS_TOW>,<detection_mask>,<insertion_mask>,<constel_mask>, <CW_val_mask>,<
CW_candidate_12>,<CW_candidate_13>,<CW_candidate_14>, <CW_candidate_15>,<CW_candidate_16>,<CW
_candidate_17>,<CW_candidate_18>, <CW_candidate_19>,<CW_candidate_20>,<CW_candidate_21>,<CW_c
andidate_22>, <CW_candidate_23>,<AGC_val_mask>,<AGC_RFA_GAIN>,<AGC_ext_G2>, <AGC_ext_GC>,<res
erved>,<reserved>,<reserved>,<jam_det_mask>,<jam_const_mask>, <NF_DDC5>,<NF_DDC6>,<NF_DDC7>,<
NF_DDC8>*<checksum><cr><lf>
```

Arguments:

Table 119. $PSTMRFI field description

Parameter	Format	Description
index	Hexadecimal, 1 digit	Message index. From 1 to 2
GPS_TOW	Float	GPS time (time of week only, in seconds)
detection_mask	Hexadecimal, 1 digit	IFDATA CW detection mask For each bit: • 1=CW detected • 0= Unknown
insertion_mask	Hexadecimal, up to 3 digits	CW notch insertion status mask Index 1: CNF1…CNF12 Index 2: CNF13…CNF24

Bit	Index is 1	Index is 2
0	IF1 CW detected	IF5 CW detected
1	IF2 CW detected	IF6 CW detected
2	IF3 CW detected	IF7 CW detected
3	IF4 CW detected	IF8 CW detected

<table>
<tr><th>Parameter</th><th>Format</th><th>Description</th></tr>
<tr><td></td><td></td><td>For each bit:
• 1=ON (inserted)
• 0=OFF

<table><tr><th>Bit</th><th>Index is 1</th><th>Index is 2</th></tr><tr><td>0</td><td>CNF1 inserted</td><td>CNF13 inserted</td></tr><tr><td>1</td><td>CNF2 inserted</td><td>CNF14 inserted</td></tr><tr><td>2</td><td>CNF3 inserted</td><td>CNF15 inserted</td></tr><tr><td>3</td><td>CNF4 inserted</td><td>CNF16 inserted</td></tr><tr><td>4</td><td>CNF5 inserted</td><td>CNF17 inserted</td></tr><tr><td>5</td><td>CNF6 inserted</td><td>CNF18 inserted</td></tr><tr><td>6</td><td>CNF7 inserted</td><td>CNF19 inserted</td></tr><tr><td>7</td><td>CNF8 inserted</td><td>CNF20 inserted</td></tr><tr><td>8</td><td>CNF9 inserted</td><td>CNF21 inserted</td></tr><tr><td>9</td><td>CNF10 inserted</td><td>CNF22 inserted</td></tr><tr><td>10</td><td>CNF11 inserted</td><td>CNF23 inserted</td></tr><tr><td>11</td><td>CNF12 inserted</td><td>CNF24 inserted</td></tr></table></td></tr>
<tr><td>constel_mask</td><td>Hexadecimal, up to 8 digits</td><td>CW affected constellation mask</td></tr>
<tr><td>CW_val_mask</td><td>Hexadecimal, up to 3 digits</td><td>CW candidate validity mask
For each bit:
• 1=Candidate valid
• 0=Unknown or not reported

<table><tr><th>Bit</th><th>Index is 1</th><th>Index is 2</th></tr><tr><td>0</td><td>CW candidate 0</td><td>CW candidate 12</td></tr><tr><td>1</td><td>CW candidate 1</td><td>CW candidate 13</td></tr><tr><td>2</td><td>CW candidate 2</td><td>CW candidate 14</td></tr><tr><td>3</td><td>CW candidate 3</td><td>CW candidate 15</td></tr><tr><td>4</td><td>CW candidate 4</td><td>CW candidate 16</td></tr><tr><td>5</td><td>CW candidate 5</td><td>CW candidate 17</td></tr><tr><td>6</td><td>CW candidate 6</td><td>CW candidate 18</td></tr><tr><td>7</td><td>CW candidate 7</td><td>CW candidate 19</td></tr><tr><td>8</td><td>CW candidate 8</td><td>CW candidate 20</td></tr><tr><td>9</td><td>CW candidate 9</td><td>CW candidate 21</td></tr><tr><td>10</td><td>CW candidate 10</td><td>CW candidate 22</td></tr><tr><td>11</td><td>CW candidate 11</td><td>CW candidate 23</td></tr></table></td></tr>
<tr><td>CW_candidate_0
or CW_candidate_12</td><td>Decimal</td><td>CW candidate #0 or 12 power (unsigned bit 31-19) /Freq (signed bit 18-0).
CW candidate # depends on message index (index 1: #0, index 2: #12).</td></tr>
<tr><td>CW_candidate_1
or CW_candidate_13</td><td>Decimal</td><td>CW candidate #1 or 13 Power/Freq.</td></tr>
<tr><td>CW_candidate_2
or CW_candidate_14</td><td>Decimal</td><td>CW candidate #2 or 14 Power/Freq.</td></tr>
</table>

Parameter	Format	Description
CW_candidate_3 or CW_candidate_15	Decimal	CW candidate #3 or 15 Power/Freq.
CW_candidate_4 or CW_candidate_16	Decimal	CW candidate #4 or 16 Power/Freq.
CW_candidate_5 or CW_candidate_17	Decimal	CW candidate #5 or 17 Power/Freq.
CW_candidate_6 or CW_candidate_18	Decimal	CW candidate #6 or 18 Power/Freq.
CW_candidate_7 or CW_candidate_19	Decimal	CW candidate #7 or 19 Power/Freq.
CW_candidate_8 or CW_candidate_20	Decimal	CW candidate #8 or 20 Power/Freq.
CW_candidate_9 or CW_candidate_21	Decimal	CW candidate #9 or 21 Power/Freq.
CW_candidate_10 or CW_candidate_22	Decimal	CW candidate #10 or 22 Power/Freq.
CW_candidate_11 or CW_candidate_23	Decimal	CW candidate #11 or 23 Power/Freq.
AGC_val_mask	Hexadecimal, 1 digit	AGC validity mask
AGC_CH1_VGA or AGC_RFA_GAIN	Decimal	Channel 1 VGA or AGC RFA Gain
AGC_CH1_PRE_VGA or AGC_ext_G2	Decimal	Channel 1 PRE VGA or AGC external G2: I,Q (6-bit I + 6-bit Q)
AGC_CH2_VGA or AGC_ext_GC	Decimal	Channel 2 VGA or AGC external GC: I,Q (6-bit I + 6-bit Q)
AGC_CH2_PRE_VGA or reserved	Decimal	Channel 2 PRE VGA or reserved
AGC_CH3_VGA or reserved	Decimal	Channel 3 VGA or reserved
AGC_CH3_PRE_VGA or reserved	Decimal	Channel 3 PRE VGA
jam_det_mask	Hexadecimal, up to 2 digits	Wide-band (IFM) jamming detection mask DDC1…DDC8 For each bit: • 1=Jamming detected • 0=Unknown

Parameter	Format	Description
		<table><tr><th>Bit</th><th>Index is 1</th><th>Index is 2</th></tr><tr><td>0</td><td>DDC1</td><td>Jamming detected</td></tr><tr><td>1</td><td>DDC2</td><td>Jamming detected</td></tr><tr><td>2</td><td>DDC3</td><td>Jamming detected</td></tr><tr><td>3</td><td>DDC4</td><td>Jamming detected</td></tr><tr><td>4</td><td>DDC5</td><td>Jamming detected</td></tr><tr><td>5</td><td>DDC6</td><td>Jamming detected</td></tr><tr><td>6</td><td>DDC7</td><td>Jamming detected</td></tr><tr><td>7</td><td>DDC8</td><td>Jamming detected</td></tr></table>
jam_const_mask	Hexadecimal, up to 8 digits	Wide band jamming affected constellation mask
NF_DDC1 or NF_DDC5	Decimal	Noise floor DDC1 or 5 (0=unknown). Noise floor DDC number depends on message index (index 1: DDC1, index 2: DDC5).
NF_DDC2 or NF_DDC6	Decimal	Noise floor DDC2 or 6 (0=unknown)
NF_DDC3 or NF_DDC7	Decimal	Noise floor DDC3 or 7 (0=unknown)
NF_DDC4 or NF_DDC8	Decimal	Noise floor DDC4 or 8 (0=unknown)

CW candidate power bit 31 to 19 is the estimated DFT power divided by 4.

CW candidate Freq bit 18 to 0 is the cosine value (two's complement) and the real frequency value in the intermediate frequency domain can be calculated with this formula:

$$freq\_real = \frac{32 \times 1023000}{\pi} \times \text{acos}\left(\frac{freq\_cos}{262144}\right)$$

Example:

```
$PSTMRFI,1,42.00000000,f,000,000019,004,,,33044655,,,,,,,,,3f,0,0,47,0,37,0,1,0,16640,16640,
16640,16640*4D
$PSTMRFI,2,42.00000000,f,000,000019,000,,,,,,,,,,,,7,0,0,0,,,,1,0,16640,16640,16640,16640*19
```

3.7.25 $PSTMUTC

This message reports the UTC time, date and time offset parameters.

NMEA message list bitmask (64 bits): 0000 0004 0000 0000

Synopsis:

```
$PSTMUTC,<utc_time>,<utc_date>,<utc_timestamp>,<gps_utc_leap>,<gps_utc_validity>*<checksum><c
r><lf>
```

Table 120. $PSTMUTC field description

Parameter	Format	Description
utc_time	hhmmss.sss	UTC Time of Fix, example: 160836.000 ".sss" is the fraction of seconds; it assumes non zero values when the fix rate is bigger than 1 Hz.
utc_date	ddmmyyyy	Date of fix: ddmmyyyy
utc_timestamp	Decimal	UTC time expressed as number of seconds since January 6th 1980
gps_utc_leap	Decimal, 2 digits	UTC to GPS time offset [s]
gps_utc_validity	Decimal, 1 digit	UTC to GPS time offset validity 0 = NOT valid 1 = Read from NVM 2 = Valid (downloaded from sky)

Example:

```
$PSTMUTC,142417.000,19112019,1258208657,18,1*55
```

3.7.26 $PSTMTEMP

This message reports the temperature parameters.

Synopsis:

```
$PSTMTEMP,<Temperature>,<Calibration>*<checksum><cr><lf>
```

Table 121. $PSTMTEMP field description

Parameter	Format	Description
Temperature	Decimal	Temperature in degree Celsius. 1 LSB = 1 degree Celsius. Signed 16 bits. Temperature in range [-256°C; +256°C]
Calibration	Decimal, 1 digit	Thermal sensor calibration: 0: Not calibrated 1: Calibrated

Example:

```
$PSTMTEMP,46,0*39
```

3.7.27 $PSTMBIASDATA

NMEA message list bitmask (64 bits): 0000 1000 0000 0000

Synopsis:

```
$PSTMBIASDATA,<glonass_path_delay>,<ggto_validity_flags>,<ggto_brdc_bias>,<ggto_est_bias>*<checksum><cr><lf>
```

Arguments:

Table 122. $PSTMBIASDATA field description

Parameter	Format	Description
glonass_path_delay		Constellation group delay between GPS and glonass constellations (in meters)
ggto_validity_flags		Bit 0: validity of ggto brdc value (0 means not valid) Bit 1: validity of ggto_est value (0 means not valid)
ggto_brdc_bias		GPS-Galileo time offset broadcasted value (in meters)
ggto_est_bias		GPS-Galileo time offset estimated value (in meters)

Example:

```
$PSTMBIASDATA,0.00,3,0.03,0.00*3D
```

3.7.28 $PSTMSPFQ

This message reports the Anti-spoofing queue parameters.

Synopsis:

```
$PSTMSPFQ,<anti_spoofing1>,<prn1>,<anti_spoofing2>,<prn2>,…*<checksum><cr><lf>
```

Table 123. **$PSTMSPFQ field description**

Parameter	Format	Description
Anti-spoofing1	Decimal	Anti-spoofing 1
Prn1	Decimal	Satellite PRN ID 1
Anti-spoofing2	Decimal	Anti-spoofing 2
Prn2	Decimal	Satellite PRN ID 2

Example:

```
$PSTMSPFQ,7,32*38
```

3.7.29 $PSTMPV

Provides position (latitude, longitude, height), velocity (north, east, vertical) and root square of covariance matrix values for position and velocity.

NMEA message list bitmask (64 bits): 0000 0001 0000 0000

Synopsis:

```
$PSTMPV,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<Alt>,<AltVal>,<Vel_N>,<Vel_E>,<Vel_V>,<P_cov_N>
,<P_cov_NE>,<P_cov_NV>,<P_cov_E>,<P_cov_EV>,<P_cov_V>,<V_cov_N>,<V_cov_NE>,<V_cov_NV>,<V_cov_
E>,<V_cov_EV>,<V_cov_V>*<checksum><cr><lf>
```

Arguments:

Table 124. **$PSTMPV field description**

Parameter	Format	Description
Timestamp	hhmmss.sss	UTC Time of GPS sample, example: 160836.000 “.sss” is the fraction of seconds; it assumes non zero values when the fix rate is bigger than 1 Hz.
Lat	DDMM.MMMMM	Lat in degree: DD: Degree MM: Minutes .MMMMM: partsMinutes
N/S	“N” or “S”	Lat Direction: North or South
Long	DDDMM.MMMMM	Long in degree: DDD: Degree MM: Minutes .MMMMM: partsMinutes
E/W	“E” or “W”	Long direction: east or west
Alt	Decimal, 6 digits	Height above mean sea level, max: 100000 m
Alt-Val	“M”	Height measure in “M” = meters
Vel_N	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity north component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3.
Vel_E	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity east component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3.
Vel_V	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity vertical component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3.
P_cov_N	ddd.d	Position north covariance [m]
P_cov_NE	ddd.d	Position north-east covariance [m]
P_cov_NV	ddd.d	Position north-vertical covariance [m]

Parameter	Format	Description
P_cov_E	ddd.d	Position east covariance [m]
P_cov_EV	ddd.d	Position east-vertical covariance [m]
P_cov_V	ddd.d	Position vertical covariance [m]
V_cov_N	ddd.d	Velocity north covariance [m/s]
V_cov_NE	ddd.d	Velocity north-east covariance [m/s]
V_cov_NV	ddd.d	Velocity north-vertical covariance [m/s]
V_cov_E	ddd.d	Velocity east covariance [m/s]
V_cov_EV	ddd.d	Velocity east-vertical covariance [m/s]
V_cov_V	ddd.d	Velocity vertical covariance [m/s]

Example:

When slow pvt mode is enabled.

```
$PSTMPV,160635.000,4055.10928,N,01416.56027,E,026.96,M,0.245,0.012,0.132,22.6,12.8,5.8,17.2,10.9,18.8,5.5,4.1,1.7,4.6,0.0,2.7*70
```

When slow pvt speed mode is disabled.

```
$PSTMPV,160635.000,4055.10928,N,01416.56027,E,026.96,M,0.2,0.0,0.1,22.6,12.8,5.8,17.2,10.9,18.8,5.5,4.1,1.7,4.6,0.0,2.7*70
```

3.7.30 $PSTMPVRAW

Provides unfiltered position (latitude, longitude, height) and velocity (north, east, vertical) calculated through a Least Square fix, and related quality indicators info.

NMEA message list bitmask (64 bits): 0000 4000 0000 0000

Synopsis:

```
$PSTMPVRAW,<Timestamp>,<Lat>,<N/S>,<Long>,<E/W>,<GPSQual>,<Sats>,<HDOP>,<Alt>,<AltVal>,<GeoSep>,<GeoVal>,<Vel_N>,<Vel_E>,<Vel_V>*<checksum><cr><lf>
```

Arguments:

Table 125. $PSTMPVRAW field description

Parameter	Format	Description
Timestamp	hhmmss.sss	UTC time of GPS sample, example: 160836.000 ".sss" is the fraction of seconds; it assumes non zero values when the fix rate is bigger than 1 Hz.
Lat	DDMM.MMMMM	Lat in degree: DD: Degree MM: Minutes .MMMMM: partsMinutes
N/S	"N" or "S"	Lat direction: north or south
Long	DDDMM.MMMMM	Long in degree: DDD: Degree MM: Minutes .MMMMM: partsMinutes
E/W	"E" or "W"	Long direction: est or west
GPSQual	Decimal, 1digit	0 = Invalid 1 = GPS 2 = DGPS

Parameter	Format	Description
Sats	Decimal, 2 digits	Satellites in use: example: 8
HDOP	Decimal, 3 digits	Horizontal dilution of precision, max: 99.0
Alt	Decimal, 6 digits	Height above mean sea level, max: 100000 m
AltVal	“M”	Reference Unit for altitude (“M” = meters)
GeoSep	Decimal, 4 digits	Geoidal separation measure in “M” = meters
GeoVal	“M”	Reference unit for geoidal separation (“M” = meters)
Vel_N	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity north component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3
Vel_E	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity east component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3
Vel_V	ddd.d (slow mode off) ddd.ddd (slow mode on)	Velocity vertical component [m/s] Format change with firmware configuration slow pvt mode page 17 line 6 field 3

Example:

When slow pvt mode is enabled.

```
 $PSTMPVRAW,144056.000,5131.12414,N,00005.31484,W,2,09,1.2,043.31,M,47.0,M,-0.612,0.123,0.634
*58
```

When slow pvt mode is disabled.

```
 $PSTMPVRAW,144056.000,5131.12414,N,00005.31484,W,2,09,1.2,043.31,M,47.0,M,-0.6,0.1,0.6*58
```

3.7.31 $PSTMPVQ

Provides position and velocity process noise matrix values.

This message is for debug only.

NMEA message list bitmask (64 bits): 0000 0002 0000 0000

Synopsis:

```
$PSTMPVQ,<reserved>,...,<reserved>*<checksum><cr><lf>
```

Arguments:

Table 126. **$PSTMPVQ field description**

Parameter	Format	Description
Reserved	-	Reserved
…		
Reserved	-	Reserved

Example:

```
$PSTMPVQ,0.0,0.0,0.0,0.0,4.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,3.0,0.0,3.0,
0.0*4A
```

3.7.32 $PSTMEPE

Provides estimated horizontal and vertical position error.

NMEA message list bitmask (64 bits): 0000 0001 0000 0000

Synopsis:

```
$PSTMPVEPE,<EHPE>,<EVPE>*<checksum><cr><lf>
```

Arguments:

Table 127. **$PSTMEPE field description**

Parameter	Format	Description
EHPE	ddd.d	Estimated horizontal position error [m]
EVPE	ddd.d	Estimated vertical position error [m]

Example:

```
$PSTMEPE,1.4,2.4*49
```

3.7.33 $PSTMSPOOF

Provides the status of the anti-spoofing algorithm. Anti-spoofing algorithm is available for GPS, GALILEO, GLONASS and BEIDOU constellations.

NMEA message list bitmask (64 bits): 0020 0000 0000 0000

Synopsis:

```
$PSTMSPOOF,<GPS_SAT_N>,<GAL_SAT_N>,<GPS_STAT>,<GAL_STAT>*<checksum><cr><lf>
$PSTMSPOOF,<GPS_SAT_N>,<GAL_SAT_N>,<GPS_STAT>,<GAL_STAT>,<GLO_SAT_N>,<BEI_SAT_N>,<GLO_STAT>,<
BEI_STAT>*<*<checksum><cr><lf>
$PSTMSPOOF,<GPS_SAT_N>,<GAL_SAT_N>,<GPS_STAT>,<GAL_STAT>,<GLO_SAT_N>,<BEI_SAT_N>,<GLO_STAT>,<
BEI_STAT>,<use_auth_only>*<checksum><cr><lf>
```

Arguments:

Table 128. **$PSTMSPOOF field description**

Parameter	Format	Description
GPS_SAT_N	Decimal	Number of GPS SVs showing DSP spoofing alarm
GAL_SAT_N	Decimal	Number of GALILEO SVs showing DSP spoofing alarm
GPS_STAT	Decimal, 1 digit	AntiSpoofing algorithm status for GPS constellation: 0 = IDLE 1 = CONSTELLATION ALARM 2 = CONSTELLATION EXCLUDED FROM PVT
GAL_STAT	Decimal, 1 digit	AntiSpoofing algorithm status for GALILEO constellation 0 = IDLE 1 = CONSTELLATION ALARM 2 = CONSTELLATION EXCLUDED FROM PVT
GLO_SAT_N	Decimal	Number of GLONASS SVs showing DSP spoofing alarm
BEI_SAT_N	Decimal	Number of BEIDOU SVs showing DSP spoofing alarm
GLO_STAT	Decimal, 1 digit	AntiSpoofing algorithm status for GLONASS constellation 0 = IDLE 1 = CONSTELLATION ALARM 2 = CONSTELLATION EXCLUDED FROM PVT
BEI_STAT	Decimal, 1 digit	AntiSpoofing algorithm status for BEIDOU constellation 0 = IDLE 1 = CONSTELLATION ALARM 2 = CONSTELLATION EXCLUDED FROM PVT
use_auth_only	Decimal, 1 digit	For OSNMA: use authenticated satellites only.

Example:

```
$PSTMSPOOF,7,0,2,0*79
$PSTMSPOOF,7,0,2,0,7,0,2,0*79
$PSTMSPOOF,7,0,2,0,7,0,2,0,1*79
```

3.7.34 $PSTMPA

Position algorithm

NMEA message list bitmask (64 bits): 0000 0000 0000 0400

Synopsis:

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

Arguments:

Table 129. $PSTMPA field description

<table>
<tr><th>Parameter</th><th>Format</th><th>Description</th></tr>
<tr><td>PosA</td><td>Char, 2 digits</td><td>Position algorithm indicator
Empty = none
LS = LMS
KF = kalman filter</td></tr>
<tr><td>Dur</td><td>Decimal, 3 digits</td><td>Time period in which the position has been stationary (count in seconds)</td></tr>
<tr><td>pos_algo_flags</td><td>Hexadecimal, 8 digits</td><td>Position algorithm flags.
Byte #0: Kalman filter configuration
For each bit:
0 means feature disabled
1 means feature enabled
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Walking mode ON</td></tr>
<tr><td>1</td><td>Stop detection ON</td></tr>
<tr><td>2</td><td>Frequency ramp ON</td></tr>
<tr><td>3</td><td>Velocity estimator model:
• 1 means MULTIPLE MODEL
• 0 means SINGLE MODEL</td></tr>
<tr><td>4</td><td>Velocity estimator filter:
• 1 means SLOW
• 0 means FAST</td></tr>
<tr><td>5</td><td>FDE status ON</td></tr>
<tr><td>6</td><td>TCXO jump:
• 1 tcxo jump applied
0 tcxo jump not applied</td></tr>
<tr><td>7</td><td></td></tr>
</table>
Byte #1 – 3: reserved</td></tr>
</table>

Example:

```
$PSTMPA,KF,433,00000028*79
$PSTMPA, ,00,00000000*79
```

3.7.35 $PSTMGSA

GNSS DOP and active satellites. Satellites from different constellations are sent on separate messages.

NMEA message list bitmask (64 bits): 0000 0000 0001 0000

Synopsis:

```
$PSTMGSA,<Mode>,<CurrentMode>,<SatPRN1>,...,<SatPRNn>,<PDOP>,<HDOP>,<VDOP>,<SystemID>,<Signal ID>*<checksum><cr><lf>
```

Table 130. $PSTMGSA field description

Parameter	Format	Description
Mode	1 character	M = Manual, forced to operate in 2D or 3D mode A = Automatic, allowed to automatically switch 2D/3D
CurrentMode	Decimal, 1 digit	Current mode: 1 = Fix not available or invalid 2 = GPS, SPS mode, fix valid 3 = Differential GPS, SPS mode, fix valid
SatxPRN	Decimal, up to 3 digits	Satellites list used for positioning. See Section 3.5: Preliminary notes about satellites' PRN ranges for more info about available values.
PDOP	x.x, variable length field	Position dilution of precision, max: 99.0
HDOP	x.x, variable length field	Horizontal dilution of precision, max: 99.0
VDOP	x.x, variable length field	Vertical dilution of precision, max: 99.0
SystemID	Hexadecimal, 1 digit	NMEA 4.11 only: The system ID of this message: 1 = GPS 2 = GLONASS 3 = GALILEO 4 = BEIDOU 5 = QZSS 6 = IRNSS
SignalID	Hexadecimal, 1 digit	An identifier to indicate the signal in use.

Example:

```
$PSTMGSA,A,3,27,08,02,21,10,23,16,01,32,,,,0.7,0.4,0.6,1,1*32
$PSTMGSA,A,3,90,78,87,71,,,,,,,,0.7,0.4,0.6,2,1*38
$PSTMGSA,A,3,,,,,,,,,,,,0.7,0.4,0.6,5,1*30
$PSTMGSA,A,3,03,15,27,30,05,09,,,,,,0.7,0.4,0.6,3,7*3B
$PSTMGSA,A,3,30,27,33,41,16,32,14,06,39,,,,0.7,0.4,0.6,4,1*3D
$PSTMGSA,A,3,03,14,01,08,10,32,,,,,,0.7,0.4,0.6,1,5*3B
$PSTMGSA,A,3,03,15,27,30,09,,,,,,,0.7,0.4,0.6,3,1*3E
$PSTMGSA,A,3,30,27,33,41,32,39,,,,,,0.7,0.4,0.6,4,5*39
```

3.7.36 $PSTMSPOOF_DIFF

Provides the position errors of the anti-spoofing algorithm.

NMEA message list bitmask (64 bits): 0020 0000 0000 0000

Synopsis:

```
$PSTMSPOOF,<ACTIVE_MASK>,<ERR_GPS_GLO>,<ERR_GPS_GAL>,<ERR_GPS_BEI>,<ERR_GLO_GAL>,<ERR_GLO_BEI>,<ERR_GAL_BEI>*<checksum><cr><lf>
```

Arguments:

Table 131. **$PSTMSPOOF_DIFF field description**

Parameter	Format	Description
ACTIVE_MASK	Decimal	Active constellation mask, main values: 1111 1111 1111 1111 not valid 0000 0000 0000 0001 GPS 0000 0000 0000 0010 GLONASS 0000 0000 0000 1000 GALILEO 0000 0000 1000 0000 COMPASS/BEIDOU
ERR_GPS_GLO	ddd.d	Detected position error comparing GPS to GLONASS [m]
ERR_GPS_GAL	ddd.d	Detected position error comparing GPS to GALILEO [m]
ERR_GPS_BEI	ddd.d	Detected position error comparing GPS to BEIDOU [m]
ERR_GLO_GAL	ddd.d	Detected position error comparing GLONASS to GALILEO [m]
ERR_GLO_BEI	ddd.d	Detected position error comparing GLONASS to BEIDOU [m]
ERR_GAL_BEI	ddd.d	Detected position error comparing GALILEO to BEIDOU [m]

Example:

```
$PSTMSPOOF_DIFF,9,0.0,23.7,0.0,0.0,0.0,0.0*79
```

3.7.37 $PSTMRFAGC

RF AGC status.

NMEA message list bitmask (64 bits): 0000 0000 0002 0000

Synopsis:

```
$PSTMRFAGC,<RFA_GAIN>,
<CH1_VGA>,<CH1_PRE_VGA>,
<CH2_VGA>,<CH2_PRE_VGA>,
<CH3_VGA>,<CH3_PRE_VGA>,
<DBFS_REF_CH1>,<DBFS_ACTL_REF_CH1>,
<DBFS_REF_CH2>,<DBFS_ACTL_REF_CH2>,
<DBFS_REF_CH3>,<DBFS_ACTL_REF_CH3>,
<RFACTRL_FSM_STATE>,<PREVGACTRL_FSM_STATE>,
<ADC_OVL_IQ>,<ADC_OVL_ON_IQ>,
 <OVL_CNT_CH1>,<OVL_CNT_CH2>,<OVL_CNT_CH3>*<checksum><cr><lf>
```

Arguments:

Table 132. **$PSTMRFAGC field description**

Parameter	Format	Description
RFA_GAIN	Decimal	RFA Gain
CH1_VGA	Decimal	Channel 1 VGA
CH1_PRE_VGA	Decimal	Channel 1 PRE VGA
CH2_VGA	Decimal	Channel 2 VGA
CH2_PRE_VGA	Decimal	Channel 2 PRE VGA
CH3_VGA	Decimal	Channel 3 VGA
CH3_PRE_VGA	Decimal	Channel 3 PRE VGA
DBFS_REF_CH1	Decimal	Wanted DBFS channel 1
DBFS_ACTL_REF_CH1	Decimal	DBFS ACTL Ref channel 1

Parameter	Format	Description
DBFS_REF_CH2	Decimal	Wanted DBFS channel 2
DBFS_ACTL_REF_CH2	Decimal	DBFS ACTL Ref channel 2
DBFS_REF_CH3	Decimal	Wanted DBFS channel 3
DBFS_ACTL_REF_CH3	Decimal	DBFS ACTL Ref channel 3
RFACTRL_FSM_STATE	Hexadecimal, up to 2 digits	RFACTRL fsm state
PREVGACTRL_FSM_STATE	Hexadecimal, up to 2 digits	PRERFACTRL fsm state
ADC_OVL_IQ	Hexadecimal, up to 2 digits	ADC OVL IQ channel 1, 2, 3
ADC_OVL_ON_IQ	Hexadecimal, up to 2 digits	ADC OVL ON IQ channel 1, 2, 3
OVL_CNT_CH1	Hexadecimal, up to 8 digits	OVL CNT channel 1
OVL_CNT_CH2	Hexadecimal, up to 8 digits	OVL CNT channel 2
OVL_CNT_CH3	Hexadecimal, up to 8 digits	OVL CNT channel 3

Example:

```
$PSTMRFAGC,1,21,0,14,0,9,0,12,64,12,64,12,64,0,0,0,3f,0,0,0*21
```

3.7.38 $SPTMPD

Reports the constellation group delay from GPS.

NMEA message list bitmask (64 bits): 0000 1000 0000 0000

Synopsis:

```
$PSTMPD,<glonass_path_delay>,<galileo_path_delay>,<beidou_path_delay>,<gps_L5_path_delay>,<gal_E5_path_delay>,<bei_B2A_path_delay>,<irnss_path_delay>*<checksum><cr><lf>
```

Arguments:

Table 133. $SPTMPD field description

Parameter	Format	Description
glonass_path_delay		Constellation group delay between GPS and glonass constellations (in meters)
galileo_path_delay		Constellation group delay between GPS and glonass constellations (in meters)
beidou_path_delay		Constellation group delay between GPS and glonass constellations (in meters)
gps_L5_path_delay		Constellation group delay between GPS and GPS L5 constellations (in meters)
gal_E5_path_delay		Constellation group delay between GPS and Galileo L5 constellations (in meters)
bei_B2A_path_delay		Constellation group delay between GPS and Beidou L5 constellations (in meters)
irnss_path_delay		Constellation group delay between GPS and IRNSS L5 constellations (in meters)

Example:

```
$PSTMPD,-452.000,-3.016,-446.062,-0.484,-0.625,-282.938,0.000*0A
```

3.7.39 $PSTMRSS

Reports SIS layer error code, constellations alarm mask and monitor alarm mask.

NMEA message list bitmask (64 bits): 0400 0000 0000 0000

Synopsis:

```
$PSTMRSS,<error_code>,<sat_type_alarm_mask>,<monitor_alarm_mask>*<checksum><cr><lf>
```

Arguments:

Table 134. $PSTMRSS field description

Parameter	Format	Description
error_code	Decimal	SIS layer error code 0=HW configuration 1=SIS message timeout 2=Monitor timeout 3=SIS memory allocation 4=Unknown message type 5=Unknown message command 6=SIS initialization 7=Reserved 8=MTU check 9=Reserved 10…254=Reserved 255= No error
constellation_alarm_mask	Hexadecimal, 8 digits	Constellation alarm mask, 1 bit per constellation 0=OK 1=ALARM Bit #0: GPS L1CA Bit #1: GLONASS G1 Bit #2: QZSS L1 CA Bit #3: GALILEO E1 Bit #4: SBAS Bit #5: QZSS L1 SAIF Bit #6: QZSS L1C Bit #7: BEIDOU B1I Bit #8: BEIDOU B1C Bit #9: GPS L2C Bit #10: IRNSS L5 Bit #11: GPS L5 Bit #12: GALILEO E5A Bit #13: GALILEO E6 Bit #14: BEIDOU B2A Bit #15: QZSS L2C Bit #16: QZSS L5 Bit #17: GPS L1C Bit #18: BEIDOU B3I Bit #19: BEIDOU B2B Bit #20: RFU Bit #21: GLONASS G2 Bit #22: BEIDOU B2I Bit #23: GALILEO E5B
monitor_alarm_mask	Hexadecimal, 8 digits	Monitor alarm mask (see Table 135. Monitor alarm mask), 1 bit per monitor. 0=OK 1=Alarm

Table 135. **Monitor alarm mask**

ID	Name	Description (0 = ok, 1 = alarm)
0	SIS	Global SIS alarm
1	CIM-L1	Observable integrity
2	HWM	Hardware failure
3	IFM	Wideband interference
4	RFM	RF tuners
5	SYSTM	System time integrity
6	Reserved	
7	IFBM	Inter frequency biases
8	NVMM	NVM write-rate integrity
9	PPSOBSM	PPS vs. observable epoch
10	CWM	Narrow-band interference
11	PPSM	Timing and PPS integrity
12	CLKESTM	RX clock estimate integrity
13	ECM	E2E counter mismatch
14	ECRC	E2E CRC failure
15	EFM	E2E frame error
16	ASM	Antenna sensing
17	DCM	Data Corruption
18	PLM	Protection level monitor
19	SPFM	Spoofing monitor
20	MTM	Message timing monitor
21…31	n/a	Reserved

Example:

```
$PSTMRSS,255,00005C00,00000C00*53
```

3.7.40 $PSTM--GSV

GNSS satellites in view extension from $--GSV standard message.

Usually, GSV messages are organised per constellation and each message carries information about up to 4 satellites in view. Thus, in certain cases, to describe all the satellites in view from a constellation more than a message is needed. This set of messages is printed once per each constellation with talker ID related to described constellation.

NMEA message list bitmask (64 bits): 0000 0000 0008 0000

Synopsis:

```
$PSTM<TalkerID>GSV,<GSVAmount>,<GSVNumber>,<TotSats>,<Sat1PRN>,<Sat1Elev>,<Sat1Azim>,<Sat1CN0
>,…,<Sat4PRN>,<Sat4Elev>,<Sat4Azim>,<Sat4CN0>,<SignalID>*<checksum><cr><lf>
```

Table 136. **$PSTM--GS field description**

Parameter	Format	Description
TalkerID	String, 2 characters	The talker ID (fixed two characters). GP: If system works in GPS only mode GL: If system works in GLONASS only mode GA: If system works in GALILEO only mode

Parameter	Format	Description
		GB: If system works in BEIDOU only mode GQ: If system works in QZSS only mode GI: If system works in IRNSS only mode
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, up to 3 digits	Satellite's list used for positioning. See Section 3.5: Preliminary notes about satellites' PRN ranges for more info about available values
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
SatxCN0	Decimal, 2 digits	Carrier to noise ratio for satellite x in dB, 00 … 99
SignalID	Hexadecimal, 2 digits	An identifier to indicate the signal in use.

NMEA 0183 Rev 4.11 doesn't have signal ID L1C then the ST proprietary message $PSTM—GSV is added with signal ID 3E for L1C (P).

Table 137. GNSS identification proprietary extension

System	System ID	Satellite ID	Signal ID	Signal Channel
GPS	1 (GP)	1 – 99 1 – 32 served for GPS	3E	L1C (P)

Example:

```
$PSTMGPGSV,1,1,02,04,,,45,28,,,37,,,,,,,,3E*6B
```

3.7.41 $PSTMALMANAC

Almanac data dump when changed.

NMEA message list bitmask (64 bits): 0000 0200 0000 0000

Synopsis:

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

Table 138. $PSTMALMANAC field description

Parameter	Format	Description
SatID	Decimal, 3 digits	Satellite number (PRN)
DataSize	Decimal, 2 digits	Number of bytes contained in the "Hex-Data" field
HexData	Hexadecimal, n-times 2 digits	Almanac data in hex-format

Example:

```
$PSTMALMANAC,1,32,011a06903f1f9f0d58fd0800d90ca1418713060099ee260034024200b4ffff00*1a
```

3.7.42 $PSTMEPHEM

Ephemeris Data Dump when changed.

NMEA message list bitmask (64 bits): 0000 0100 0000 0000

Synopsis:

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

Table 139. **$PSTMEPHEM field description**

Parameter	Format	Description
SatID	Decimal, 3 digits	Satellite number (PRN)
DataSize	Decimal, 2 digits	Number of bytes contained in the “Hex-Data” field
HexData	Hexadecimal, n-times 2 digits	Ephemeris data in hex-format

Example:

```
$PSTMEPHEM,1,64,0f06bc34bc345f5f5f84f400dea4ff00f9f63c239f0a35f81400fbff33420000ee632f27698ef
001afa50da16cfcfa22e0b65a3e7a3cee27d700f7ffc616fe03*57
```

3.8 Commands answers messages specification

3.8.1 $PSTMVER

Answer of the $PSTMGETSWVER command. It returns the version string of the libraries embedded in the software application.

Synopsis:

```
$PSTMVER,<Lib>_<Ver>_<Type>*<checksum><cr><lf>
$PSTMVER,<Lib>_<Ver>_<Rel_status>_<Type>*<checksum><cr><lf>
```

Table 140. **$PSTMVER field description**

Parameter	Format	Description
Lib	Text, fixed	Text string identifying the library that the command is requiring the version: GNSSLIB if id = 0 FRRTOS if id = 1 GPSAPP if id = 2 BINIMG_STA8xxx if id = 6 SWCFG if id = 11
Ver	x.x.x.x, x.x.x, x_x_x_x or x_x_x	GNSS library version: example 7.1.1.15, 7.1.2, 6_1_2_1 or 6_1_2
Reserved		

Example:

```
$PSTMVER,GNSSLIB_10.3.0.0*7A
$PSTMVER,FreeRTOS_V10.4.4*51
$PSTMVER,GPSAPP_4.0.0*19
$PSTMVER,BINIMG_STA8600_6_5_3*09
$PSTMVER,SWCFG_92000003*66
```

3.8.2 $PSTMFEDUMP

This message reports the current values of RF front-end registers. This message is sent as a reply to a $PSTMFEDUMP command.

Synopsis for internal front end:

```
$PSTMFEDUMP,00,<R0>,01,<R1>,02,<R2>,03,<R3>,04,<R4>,05,<R5>,06,<R6>,07,<R7>*<checksum><cr><lf>
$PSTMFEDUMP,08,<R8>,09,<R9>,0a,<R10>,0b,<R11>,0c,<R12>,0d,<R13>,0e,<R14>,0f,<R15>*<checksum><cr><lf>
$PSTMFEDUMP,10,<R16>,11,<R17>,12,<R18>,13,<R19>,14,<R20>,15,<R21>,16,<R22>,17,<R23>*<checksum><cr><lf>
$PSTMFEDUMP,18,<R24>,19,<R25>,1a,<R26>,1b,<R27>,1c,<R28>,1d,<R29>,1e,<R30>,1f,<R31>*<checksum><cr><lf>
$PSTMFEDUMP,20,<R32>,21,<R33>,22,<R34>,23,<R35>,24,<R36>,25,<R37>,26,<R38>,27,<R39>*<checksum><cr><lf>
$PSTMFEDUMP,28,<R40>,29,<R41>,2a,<R42>,2b,<R43>,2c,<R44>,2d,<R45>,2e,<R46>,2f,<R47>*<checksum><cr><lf>
$PSTMFEDUMP,30,<R48>,31,<R49>,32,<R50>,33,<R51>,34,<R52>,35,<R53>,36,<R54>,37,<R55>*<checksum><cr><lf>
$PSTMFEDUMP*15
$PSTMFEDUMP,1*<checksum><cr><lf>
```

Table 141. **$PSTMFEDUMP field description**

Parameter	Format	Description
R1 – R60	Hexadecimal, 0x0 - 0xFF	The value of the specific register

Example:

```
$PSTMFEDUMP,00,fd,01,01,02,1f,03,ff,04,00,05,01,06,f0,07,36*13
$PSTMFEDUMP,08,36,09,01,0a,7f,0b,ff,0c,ff,0d,7f,0e,11,0f,12*14
$PSTMFEDUMP,10,19,11,f9,12,4e,13,03,14,6b,15,2b,16,05,17,2e*46
$PSTMFEDUMP,18,0f,19,36,1a,00,1b,00,1c,c7,1d,13,1e,10,1f,e0*42
$PSTMFEDUMP,20,00,21,01,22,01,23,ff,24,1a,25,28,26,e0,27,7f*4B
$PSTMFEDUMP,28,30,29,1a,2a,28,2b,e0,2c,7f,2d,ff,2e,00,2f,ff*4E
$PSTMFEDUMP,30,00,31,01,32,2c,33,2c,34,2e,35,2c,36,00,37,61*15
$PSTMFEDUMP*15
```

3.8.3 $PSTMIONOPARAMS

Iono packet dump. This message is sent as a reply to a $PSTMDUMPIONO command.

NMEA message list bitmask (64 bits): 0000 0400 0000 0000

Synopsis:

```
$PSTMIONOPARAMS,<sat_type=0>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>,<cr><lf>
$PSTMIONOPARAMS,<sat_type=7>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>,<cr><lf>
$PSTMIONOPARAMS,<sat_type=10>,1,<A0>,<A1>,<A2>,<A3>,<B0>,<B1>,<B2>,<B3>,<cr><lf>
$PSTMIONOPARAMS,<sat_type=3>,1,<ai0>,<ai1>,<ai2>,<Region1>,<Region2>,<Region3>,<Region4>,<Reg
ion5>*<checksum><cr><lf>
```

Arguments:

Table 142. **$PSTMIONOPARAMS field description**

Parameter	Format	Description
sat_type	Decimal, 1 digit	0 = GPS 3 = Galileo 7 = BeiDou 10 = IRNSS
A0,A1,A2,A3	Decimal, 3 digits	These parameters are used only if sat_type=0,7 or 10 Iono parameters, raw integer values as from navigation messages.
B0,B1,B2,B3	Decimal, 3 digits	These parameters are used only if sat_type=0,7 or 10 Iono parameters, raw integer values as from navigation messages.
ai0,ai1,ai2	Decimal, 3 digits	These parameters are used only if sat_type=3 Iono parameters, raw integer values as from navigation messages.
Region1, region2, region3,region4, region5	Binary	These parameters are used only if sat_type=3 Galileo iono regions

Example:

```
$PSTMDUMPIONO
$PSTMIONOPARAMS,0,1,6,2,255,254,41,6,255,248*3A
$PSTMIONOPARAMS,7,1,3,18,236,48,73,192,127,129*36
$PSTMIONOPARAMS,3,1,123,15,316,0,0,0,0,0*33
```

3.8.4 $PSTMGLONASSICB

GLONASS inter channel bias (ICB) tables dump. This message is sent as a reply to a $PSTMDUMPGLONASSICB command.

Synopsis:

```
$PSTMGLONASSICB,<carrier>,<chan-7>,<chan-6>,<chan-5>,<chan-4>,<chan-3>,<chan-2>,<chan-1>,<cha
n0>,<chan+1>,<chan+2>,<chan+3>,<chan+4>,<chan+5>,<chan+6>*<checksum><CR><LF>
```

Arguments:

Table 143. $PSTMGLONASSICB field description

Parameter	Format	Description
carrier	Decimal, 1 digit	1= GLONASS L1 2 = GLONASS L2
Chan n	Decimal, 3 digits	Pseudorange bias in [dm] for the GLONASS channel n = -7…+6

Example:

```
$PSTMGLONASSICB,1,50,40,30,20,0,0,0,0,0,0,0,0,0*04
$PSTMGLONASSICB,2,0,0,0,0,0,0,0,0,0,0,0,0,0*07
```

3.8.5 $PSTMIFB

Inter frequency pseudorange biases (IFB) array dump. This message is sent as a reply to the $PSTMDUMPIFB command.

NMEA message list bitmask (64 bits): 0008 0000 0000 0000

Synopsis:

```
$PSTMIFB,<GPS_L2C_L1>,<GLONASS_L2_L1>,<BeiDou_B2I_B1I>,<GPS_L5_L1>,<GAL_E5a_E1>,<GAL_E5b_L1>,
<GAL_E6_E1>,<BeiDou_B2a_B1I>,<GPS_L1C_L1>,<BeiDou_B1c_B1I>,<BeiDou_B3I_B1I>,<BeiDou_B2b_B1I>*
<checksum><CR><LF>
```

Arguments:

Table 144. $PSTMIFB field description

Parameter	Format	Description
GPS_L2C_L1	Decimal, 3 digits	IFB [dm] between GPS L1 and L2 carriers
GLONASS_L2C_L1	Decimal, 3 digits	IFB [dm] between GLONASS L1 and L2 carriers
BeiDou_B2I_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B1I and B2I carriers
GPS_L5_L1	Decimal, 3 digits	IFB [dm] between GPS L1 and L5 carriers
GAL_E5a_E1	Decimal, 3 digits	IFB [dm] between GALILEO E5a and E1 carriers
GAL_E5b_L1	Decimal, 3 digits	IFB [dm] between GALILEO E5b and L1 carriers
GAL_E6_E1	Decimal, 3 digits	IFB [dm] between GALILEO E6 and E1 carriers
BeiDou_B2a_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B2a and B1I carriers
GPS_L1C_L1	Decimal, 3 digits	IFB [dm] between GPS L1C and L1 carriers
Beidou_B1c_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B1c and B1I carriers
Beidou_B3I_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B3I and B1I carriers
Beidou_B2b_B1I	Decimal, 3 digits	IFB [dm] between BEIDOU B2b and B1I carriers

Example:

```
$PSTMIFB,950,100,0,0,0,0,0,0,0,0,0,0*5A
```

3.9 Standard OSNMA NMEA messages specification

3.9.1 $--AMC

Authenticated minimum specific (AMC) Data. Time, date, position and speed data provided by the GNSS receiver.
NMEA message list bitmask (64 bits): 0000 0000 0000 0040

Synopsis:

```
$GPAMC,<Timestamp>,<Status>,<Lat>,<N/S>,<Long>,<E/W>,<Speed>,
<Trackgood>,<Date>,<MagVar>,<MagVarDir>,<mode>,
<Nav_status>*<checksum><cr><lf>
```

Table 145. $--AMC field description

Parameter	Format	Description
Timestamp	hhmmss.sss	UTC time of GPS sample: hh: Hours (fixed two digits) mm: Minutes (fixed two digits) ss: Seconds (fixed two digits) .sss: Decimal fraction of seconds (variable length)(1)
Status	“A” or “V”	“A” = The reported position is authenticated and performed using only Galileo authenticated satellites. “V” = The reported position is not authenticated(2)
Lat	DDMM.MMMMM	Latitude as degrees: DD: Degree (fixed two digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
N/S	“N” or “S”	Latitude direction: north or south
Long	DDDMM.MMMMM	Longitude as degrees: DDD: Degree (fixed three digits) MM: Minutes (fixed two digits) MMMMM: Decimal fraction of minutes (variable)
E/W	“E” or “W”	Longitude direction: east or west
Speed	x.x, variable length field	Speed over ground in knots
Trackgood	x.x, variable length field	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” or “W”	Magnetic variation direction
Mode	“D”, “A”, “N” or “E”	Positioning system mode Indicator: “D” = Differential mode “A” = Autonomous mode “N” = Data not valid “E” = Estimated (dead reckoning) mode
Nav_status	“S”, “C”, “U” or “V”	NMEA 4.11 only: navigational status indicator: “S” = Safe “C” = Caution “U” = Unsafe “V” = Not valid

1. *That decimal fraction assumes non zero values when the fix rate is bigger than 1 Hz.*

2. *"V" is reported in NO FIX conditions and "A" is reported in 2D and 3D fix conditions.*

Example for NMEA 0183 Rev 4.11:

```
$GPAMC,081623.000,A,4759.63403,N,00011.13573,E,0.1,0.0,220321,,,A,C*1E
```

3.9.2 $--ASA

Authenticated active satellites (ASA). Satellites from different constellations are sent on separate messages.
NMEA message list bitmask (64 bits): 0000 0000 0000 0004

Synopsis:

```
$GNASA,<Mode>,<CurrentMode>,<SatPRN1>,...,<SatPRNN>,<PDOP>,<HDOP>,<VDOP>,<SystemID>*<checksum
><cr><lf>
```

Table 146. $--ASA field description

Parameter	Format	Description
Mode	1 character	M = Manual, forced to operate in 2D or 3D mode (not supported) A = Automatic, allowed to automatically switch 2D/3D
CurrentMode	Decimal, 1 digit	Current mode: 1 = Fix not available or invalid 2 = GPS, SPS mode, fix valid 3 = Differential GPS, SPS mode, fix valid
SatxPRN	Decimal, up to 3 digits	Authenticated Satellites list used for positioning. See Section 3.5: Preliminary notes about satellites' PRN ranges for more info about available values.
PDOP	x.x, variable length field	Position dilution of precision, max: 99.0
HDOP	x.x, variable length field	Horizontal dilution of precision, max: 99.0
VDOP	x.x, variable length field	Vertical dilution of precision, max: 99.0
SystemID	Hexadecimal, 1 digit	The system ID of this message: 3 = GALILEO

Example:

```
$GNASA,A,3,22,03,14,01,28,08,27,10,32,17,,,0.7,0.4,0.6,3*32
```

3.10 Standard OSNMA NMEA messages specification

3.10.1 $PSTMNMASTATUS

Provides the OSNMA status enable or disable.

Synopsis:

```
$PSTMNMASTATUS,<on_off>,<oam_flag>*<checksum><cr><lf>
```

Arguments:

Table 147. $PSTMNMASTATUS field description

Parameter	Format	Description
on_off	Decimal, 1 digit	OSNMA status 0 = OSNMA disable 1 = OSNMA enable
oam_flag	Decimal, 1 digit	OSNMA under alert message flag

3.10.2 $PSTMNMAMACV

OSNMA output event MACV that reports the MAC authentication status.

Synopsis:

```
$PSTMNMAMACV,<result>,<slc>,<gst_v_tow>,<gst_v_wn>,<cpu_time_v>,<cpu_time_r>,<cpu_time_d>,<svid_a>,<svid>,<adkd_macseq>,<cop>,<ctr>,<mac64>,<nma_status>,<cid>,<cpks>,<gst_tow>,<gst_wn>*<checksum><cr><lf>
```

Arguments:

Table 148. $PSTMNMAMACV field description

Parameter	Format	Description
result	Decimal, 1 to 3 digits	Result 0 = MAC authentic 1 = MAC not authentic (-1) = No input data (-2) = Processing error during MAC ver. (-3) = Removed because of max. age (-7) = Slow mac filtered (-10) = MACLT ADKD check failed (-14) = FLX with MAC0 not available (-15) = One or more FLX missing (-17) = MACSEQ not authentic (-19) = MACLT index not available (-20) = Processing error during MACSEQ
slc	Decimal	Security level counter (MAC accumulation)
gst_v_tow	Float	Time of week of verification
gst_v_wn	Decimal	Week number of verification
cpu_time_v	Decimal	Cpu time of verification
cpu_time_r	Decimal	Time of reception of the last bit of the MAC
cpu_time_d	Decimal	Time of reception of the last bit of the data to authenticate
svid_a	Decimal, 1 to 3 digits	Identifies the satellite transmitting the authentication information. Galileo satellite.
svid	Decimal, 1 to 3 digits	Authenticated PRN (for MAC0, PRN_N is filled with PRN_A), in the case this field is set to the value “255”, PRN=PRN A
adkd or_macseq	Decimal	If MAC0 (blk_idx=0, ctr=1), MAC sequence that allows the receiver to authenticate the tag-info field for the tags whose ADKD type is identified as flexible within the MAC look-up table. If not MAC0, authentication data and key delay describes the authenticated navigation data, used to generate the associated tag.
cop	Decimal, 1 to 3 digits	Cut-off point
ctr	Decimal, 1 to 3 digits	Position of the tag within the MACK message
mac64	Decimal	MacInfo or authentication tag.
nma_status	Decimal, 1 digit	NMA status

<table>
<tr><th>Parameter</th><th>Format</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>NMAS</th><th>Definition</th><th>Semantic</th></tr>
<tr><td>0</td><td>Reserved</td><td></td></tr>
<tr><td>1</td><td>Test</td><td>OSNMA is provided without any operational guarantees.</td></tr>
<tr><td>2</td><td>Operational</td><td>OSNMA is provided according to the specifications.</td></tr>
<tr><td>3</td><td>Don't use</td><td>Navigation data shall not be authenticated with the provided OSNMA information.</td></tr>
</table>
</td></tr>
<tr><td>cid</td><td>Decimal, 1 digit</td><td>Chain ID represents the ID of the key chain in force. Values from 0 to 3</td></tr>
<tr><td>cpks</td><td>Decimal, 1 digit</td><td>Chain and public key status, provides the status of the key chain and public key in force. Values 0 to 7</td></tr>
<tr><td>gst_tow</td><td>Decimal</td><td>TOW GST of the subframe containing the MAC</td></tr>
<tr><td>gst_wn</td><td>Decimal</td><td>Week number GST of the subframe containing the MAC</td></tr>
</table>

3.10.3 $PSTMNMADSMV

OSNMA output event DSMV that reports the DSM KROOT validation status.

Synopsis:

```
$PSTMNMADSMV,<result>,<gst_v_tow>,<gst_v_wn>,<cpu_time_v>,<dsm_id>,<nma_status>,<cid>,<cpks>,
<num_blk>,<pkid>,<cid_kr>,<res1>,<HF>,<MF>,<KS>,<TS>,<maclt>,<res2>,<kr_wn>,<kr_towh>*<checks
um><cr><lf>
```

Arguments:

Table 149. $PSTMNMADSMV field description

Parameter	Format	Description
Result	Decimal, 1 digit	Result 0= Authentic 1= Not authentic 2=vPK not available
gst_v_tow	Float	Gst tow of verification
gst_v_wn	Decimal	Gst week number of verification
cpu_time_v	Decimal	Receiver clock counter at validation epoch
dsm_id	Decimal, 1 to 2 digits	Digital signature message ID associated with current block. 0 to 11 = for DSM-KROOT
nma_status	Decimal, 1 digit	NMA status

<table>
<tr><th>Parameter</th><th>Format</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>NMAS</th><th>Definition</th><th>Semantic</th></tr>
<tr><td>0</td><td>Reserved</td><td></td></tr>
<tr><td>1</td><td>Test</td><td>OSNMA is provided without any operational guarantees.</td></tr>
<tr><td>2</td><td>Operational</td><td>OSNMA is provided according to the specifications.</td></tr>
<tr><td>3</td><td>Don’t use</td><td>Navigation data shall not be authenticated with the provided OSNMA information.</td></tr>
</table>
</td></tr>
<tr><td>cid</td><td>Decimal, 1 digit</td><td>Chain ID represents the ID of the key chain in force. Values from 0 to 3.</td></tr>
<tr><td>cpks</td><td>Decimal, 1 digit</td><td>Chain and public key status, provides the status of the key chain and public key in force. Values 0 to 7</td></tr>
<tr><td>num_blk</td><td>Decimal, 1 digit</td><td>Number of blocks of the DSM-KROOT in I/NAV subframe.</td></tr>
<tr><td>pkid</td><td>Decimal, 1 digit</td><td>ID of the public key (PK) used to verify the signature of the DSM-KROOT.</td></tr>
<tr><td>cid_kr</td><td>Decimal, 1 digit</td><td>KROOT Chain ID identifies the chain to which the signed KROOT belongs.</td></tr>
<tr><td>res1</td><td>Decimal, 1 digit</td><td>Reserved 1</td></tr>
<tr><td>HF</td><td>Decimal, 1 digit</td><td>Hash function used for the chain:
0 = SHA-256
1 = Reserved
2 = SHA3-256
3 = Reserved</td></tr>
<tr><td>MF</td><td>Decimal, 1 digit</td><td>MAC function used to authenticate the navigation data:
0 = HMAC-SHA-256
1 = CMAC-AES
2-3 = Reserved</td></tr>
<tr><td>KS</td><td>Decimal, 1 to 2 digits</td><td>Key size identifies the entry of a look-up table indicating the length l_k of the keys of the chain, expressed in bits.
<table>
<tr><th>KS</th><th>Key length</th><th>KS</th><th>Key length</th></tr>
<tr><td>0</td><td>96</td><td>5</td><td>160</td></tr>
<tr><td>1</td><td>104</td><td>6</td><td>192</td></tr>
<tr><td>2</td><td>112</td><td>7</td><td>224</td></tr>
<tr><td>3</td><td>120</td><td>8</td><td>256</td></tr>
<tr><td>4</td><td>128</td><td>9-15</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>TS</td><td>Decimal, 1 to 2 digits</td><td>Tag size identifies the entry of a look-up table indicating the tags length l_t, expressed in bits.
<table>
<tr><th>TS</th><th>Tag length</th><th>TS</th><th>Tag length</th></tr>
<tr><td>0-4</td><td>Reserved</td><td>8</td><td>32</td></tr>
<tr><td>5</td><td>20</td><td>9</td><td>40</td></tr>
<tr><td>6</td><td>124</td><td>10-15</td><td>Reserved</td></tr>
<tr><td>7</td><td>28</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>maclt</td><td>Decimal, 1 to 3 digits</td><td>Mac look-up table. Entry of look-up table specifying the authentication data and key (ADKD)</td></tr>
</table>

Parameter	Format	Description
res2	Decimal, 1 to 3 digits	Reserved 2.
kr_wn	Decimal	KROOT week number
kr_towh	Decimal, 1 to 3 digits	KROOT time of week

3.10.4 $PSTMNMAPKRV

OSNMA output event PKRV that reports the DSM-PKR validation status.

Synopsis:

```
$PSTMNMAPKRV,<result>,<mtr_id>,<gst_v_tow>,<gst_v_wn>,<cpu_time_v>,<dsm_id>,<nma_status>,<cid>,<cpks>,<num_blk>,<mid>,<npkt>,<npkid>*<checksum><cr><lf>
```

Arguments:

Table 150. $PSTMNMAPKRV field description

Parameter	Format	Description
Result	Decimal, 1 digit	Result: 0 = MAC authentic 1 = MAC not authentic
mtr_id	Decimal, 1 to 2 digits	Merkle tree root index 0 = Current 1 = Future -1 = If both failed (in that case result is not 0)
gst_v_tow	Float	TOW GST of the subframe containing the MAC
gst_v_wn	Decimal	Week number GST of the subframe containing the MAC
cpu_time_v	Decimal	Cpu time of verification
dsm_id	Decimal, 1 to 2 digits	Digital signature message ID associated with current block. 12 to 15 = for DSM-PKR
nma_status	Decimal, 1 digit	NMA status

NMAS	Definition	Semantic
0	Reserved	
1	Test	OSNMA is provided without any operational guarantees
2	Operational	OSNMA is provided according to the specifications
3	Don't use	Navigation data shall not be authenticated with the provided OSNMA information.

cid	Decimal, 1 digit	Chain ID represents the ID of the key chain in force. Values from 0 to 3.
cpks	Decimal, 1 digit	Chain and public key status, provides the status of the key chain and Public Key in force. Values 0 to 7
num_blk	Decimal, 1 digit	Number of blocks of the DSM-PKR in I/NAV subframe.
mid	Decimal, 1 digit	Message ID identifies which leaf of the merkle tree is provided
npkt	Decimal, 1 digit	New public key type represents the signature algorithm associated with the public key provided in the DSM-PKR : 1 = ECDSA P-256

Parameter	Format	Description
		3 = ECDSA P-521 Other = Reserved
npkid	Decimal, 1 digit	New public key ID

3.10.5 $PSTMNMAAM

Provides alert message broadcasted from satellites.

Synopsis:

```
$PSTMNMAAM,<cpu_time>,<dsm_id>,<authenticated>,<alert_msg>*<checksum><cr><lf>
```

Arguments:

Table 151. $PSTMNMAAM field description

Parameter	Format	Description
cpu_time	Decimal	Cpu time
dsm_id	Decimal, 1 to 2 digits	DSM ID from 0 to 15
authenticated	Decimal, 1 digit	0= Not authenticated 1= Authenticated
alert_msg	String, max 78 digits	Alert message

4 Firmware configuration management

The firmware config is defined as a block of memory divided in 64 pages of 16 lines of 32 bits which means a memory space of 4 kBytes defined in binary (default settings), in RAM (current configuration) and in backup memory (customized configuration saved in NVM).

4.1 $PSTMGETPAR

Read the parameters of the FW configuration either in RAM, in binary or in NVM (if available).

It is possible to read either each field of the FW config independently, dump a complete line, dump a complete page or dump the whole content of one of the FW configuration.

Synopsis:

```
$PSTMGETPAR,<ConfigBlock>,[<Page>],[<Line>],[<Field>]*<checksum><cr><lf>
```

Arguments:

Table 152. $PSTMGETPAR field description

Parameter	Format	Description
configBlock	Decimal	Indicates one of the following configuration blocks: • 1 = Current configuration (RAM) • 2 = Default configuration • 3 = NVM stored configuration
Page	Decimal	FW config page value in range [0-63]
Line	Decimal	FW config line value within page in range [0-15]
Field	Decimal	FW config field value within line in range [0-31]

If field is not specified, the complete line content is displayed.

If line/field is not specified, the complete page content is displayed

If page/line/field is not specified, the complete block content is displayed

If line/field corresponds to a floating-point type, the floating value is displayed. Note that a floating-point type is always aligned on an even line value. In this case, Field can be set to any value and will be forced to 0.

Results:

Read a field of the FW config:

```
$PSTMGETPAR,<ConfigBlock>,<Page>,<Line>,<Field>*<checksum><cr><lf>
```

Result:

```
$PSTMGETPAR,<ConfigBlock>,P<Page>,L<Line>,F<Field>,<Field value>*<checksum><cr><lf>
```

Dump a line of the FW config:

```
$PSTMGETPAR,<ConfigBlock>,<Page>,<Line>*<checksum><cr><lf>
```

Result:

```
$PSTMGETPAR,<ConfigBlock>,P<Page>,L<Line>,<Line dump>*<checksum><cr><lf>
```

Dump a page of the FW config:

```
$PSTMGETPAR,<ConfigBlock>,<Page>*<checksum><cr><lf>
```

Result:

```
$PSTMGETPAR,<ConfigBlock>,P<Page>,0,<dump line 0-7>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P<Page>,8,<dump line 8-15>*<checksum><cr><lf>
```

Dump a complete FW config:

```
$PSTMGETPAR,<ConfigBlock>*<checksum><cr><lf>
```

Result:

```
$PSTMGETPAR,<ConfigBlock>,P00,0,<dump line 0-7>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P00,8,<dump line 8-15>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P01,0,<dump line 0-7>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P01,8,<dump line 8-15>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P02,0,<dump line 0-7>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P02,8,<dump line 8-15>*<checksum><cr><lf>
……..
$PSTMGETPAR,<ConfigBlock>,P63,0,<dump line 0-7>*<checksum><cr><lf>
$PSTMGETPAR,<ConfigBlock>,P63,8,<dump line 8-15>*<checksum><cr><lf>
```

In case of error the following message will be sent:

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

Example 1: Read a field of the FW config.

Read the field 1 of line 14 in page 1 of the current configuration:

```
$PSTMGETPAR,1,1,14,1,0 or $PSTMGETPAR,1,1,14,1
```

Result:

```
$PSTMGETPAR,1,P01,L14,F01,00000002*07
```

Example2: Read a floating field of the FW config.

Read the floating type field at line 2-3 in page 19 of the current configuration:

```
$PSTMGETPAR,1,19,2,0
or
$PSTMGETPAR,1,19,2,1
or
$PSTMGETPAR,1,19,3,0
```

Result:

```
$PSTMGETPAR,1,P19,L02,F00,6.330000e-07*15
```

Example3: Dump a line of the FW config.

Dump the line 14 in page 1 of the current configuration:

```
$PSTMGETPAR,1,1,14
```

Result:

```
$PSTMGETPAR,1,P01,L14,0200000a*3D
```

Example4: Dump a page of the FW config.

Dump the page 2 of the current configuration:

```
$PSTMGETPAR,1,2
```

Result:

```
$PSTMGETPAR,1,P02,0,3fc00000,00000600,00000000,ffffffff,00000000,00000000,00000000,00000000*7
0
$PSTMGETPAR,1,P02,8,6019f861,0140000a,00000000,ffffffef,00000000,00000000,00000000,00000000*7
8
```

Example5: Dump a complete FW config.

Dump the default configuration (from binary)

```
$PSTMGETPAR,2
```

Result:

```
$PSTMGETPAR,2,P00,0,aef5321a,b0134ac0,3f855aa0,def0a534,90000001,00000000,00000000,00000000*70
$PSTMGETPAR,2,P00,8,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000*79
$PSTMGETPAR,2,P01,0,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000*70
$PSTMGETPAR,2,P01,8,00000000,00000000,00000000,00000000,0e801100,0100000c,0200000a,0100000c*76
$PSTMGETPAR,2,P02,0,3fc00000,00000600,00000000,ffffffff,00000000,00000000,00000000,00000000*73
$PSTMGETPAR,2,P02,8,6019f861,0140000a,00000000,ffffffef,00000000,00000000,00000000,00000000*7B
……
$PSTMGETPAR,2,P63,0,41464544,20544c55,43205746,00004746,00000000,00000000,00000000,00000000*22
$PSTMGETPAR,2,P63,8,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000*7C
```

Example 6: Error report.

Read the field 4 of line 1 in page 16 of the current configuration:

```
$PSTMGETPAR,1,16,1,4
```

Result:

```
$PSTMGETPARERROR*57
```

The specified field index does not exist in the current line.

4.2 $PSTMSETPAR

Update the parameters of the FW configuration in the current configuration in RAM.

See example in Section 3.4: Sequence to change NVM parameter for message sequence to store in NVM.

Synopsis:

```
$PSTMSETPAR,<Page>,<Line>,<Field>,<mode>,<param value>*<checksum><cr><lf>
```

Arguments:

Table 153. $PSTMSETPAR field description

Parameter	Format	Description
Page	Decimal	FW config page value in range [1-63] *Note: Page0 is reserved for internal usage and cannot be updated.*
Line	Decimal	FW config line value within page in range [0-15]
Field	Decimal	FW config field value within line in range [0-31]
Mode	Decimal	It allows to perform bit-to-bit “OR” or “AND” operations between the selected field and the value in input. It has the following meaning: 0: The field is overwritten by the *param_value*. This is the default action as in the case mode is omitted. 1: The field content is the result of bit-to-bit “OR” between old value and the given *param_value*. This is useful for bit mask setting. 2: The field content is the result of bit-to-bit “AND” between old value and NOT (*param_value*). This is useful for bit mask resetting. (*Not used when the field corresponds to a floating-point type*).
Param value	Hexadecimal, floating	Value to be written in the Field. If the field corresponds to a floating point type, value must be given as xxx.yyy or xxx.yyyE[±]z

Results:

```
$PSTMSETPAR,P<Page>,L<Line>,F<Field>,<new value>*<checksum><cr><lf>
```

In case of error the following message will be sent:

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

Example 1:

Update the field 1 of line 0 on page 35.

```
$PSTMSETPAR,35,0,1,0,0xAAAAAA or $PSTMSETPAR,35,0,1,0,AAAAAA
```

Result:

```
$PSTMSETPAROK,P35,L00,F01,00aaaaaa*42
```

Example 2:

Update the field 1 of line 0 on page 35.

```
$PSTMSETPAR,35,0,1,0,222.3333333
```

Result:

```
$PSTMSETPARERROR,EXPECTEDHEXAVALUE*2E
```

The specified line is not a floating-point type field, hence the value should be a hexadecimal value.

Example 3:

Update the floating type field of line 0 on page 19.

```
$PSTMSETPAR,19,0,0,0,222.3333333 or $PSTMSETPAR,19,1,0,0,222.3333333
```

Result:

```
$PSTMSETPAROK,P19,L00,F00,2.223333e+02*1D
```

Example 4:

Update the floating type field of line 0 on page 19.

```
$PSTMSETPAR,19,0,0,0,99E-2
```

Result:

```
$PSTMSETPAROK,P19,L00,F00,9.900000e-01*1A
```

Example 5:

Update the floating type field 0 of line 1 on page 19.

```
$PSTMSETPAR,19,0,0,0,1234 or $PSTMSETPAR,19,0,0,0,1234
```

Result:

```
$PSTMSETPARERROR,EXPECTEDFLOATINGVALUE*2A
```

The specified line belongs to a floating point type field, hence the value should have format xxx.yyy or xxx.yyyE[-/+]z.

Example 6:

Update the field 0 of line 0 on page 17.

```
$PSTMSETPAR,17,0,0,0,0x123
```

Result:

```
$PSTMSETPARERROR*43
```

The specified value is out of range for the defined field coded on 8 bits.

Example 7:

Update the field 4 of line 0 on page 16.

```
$PSTMSETPAR,16,0,4,0,0xAB
```

Result:

```
$PSTMSETPARERROR*43
```

The specified field index does not exist in the line.

4.3 $PSTMSETPARLINE

Update a complete line of the FW configuration in the current configuration in RAM.

Synopsis:

```
$PSTMSETPARLINE,<Page>,<Line>,<line value>*<checksum><cr><lf>
```

Arguments:

Table 154. $PSTMSETPARLINE field description

Parameter	Format	Description
Page	Decimal	FW config page value in range [1-63] *Note: Page0 is reserved for internal usage and cannot be updated.*
Line	Decimal	FW config line value within page in range [0-15]

Results:

```
$PSTMSETPARLINEOK,P<Page>,L<Line>,<new value>*<checksum><cr><lf>
```

In case of error the following message will be sent:

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

Example 1:

Update the line 3 on page 1.

```
$PSTMSETPARLINE,35,0,0xABC or $PSTMSETPARLINE,35,0,ABC
```

Result:

```
$PSTMSETPARLINEOK,P35,L00,00000abc*77
```

Example 2:

Update the line 0 on page 19.

```
$PSTMSETPARLINE,19,0,0xABC
```

Result:

```
$PSTMSETPARLINEERROR*4D
```

The specified line belongs to a floating point type field (coded on 2 lines), hence it cannot be modified as a single line.

4.4 $PSTMSAVEPAR

Save current configuration data block into the backup memory.

Synopsis:

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

Arguments:

None.

Results:

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

If there are no error the following message is returned:

```
$PSTMSAVEPAROK
```

In case of errors, the error message is returned:

```
$PSTMSAVEPARERROR
```

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

Example:

```
$PSTMSAVEPAR
```

4.5 $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*<checksum><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 configuration to get system working with default setting.

If there are no error the following message is returned:

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

In case of errors, the error message is returned:

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

Example:

```
$PSTMRESTOREPAR
```

5 Almanacs and ephemeris management

5.1 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 to obtain useful data it's better 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 is necessary to 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 to save commands in various text files that may be transmitted over the connection by the Teseo-Suite. This example makes use of the following files:

- SUSPEND.txt
- RESUME.txt
- DUMPEPHEMS.txt
- DUMPALMANAC.txt

Step 1

Ensure that the connection is working and that the user can see NMEA data displayed on their Teseo-Suite.

Step 2

Ensure that the Teseo-Suite is logging its input to a text file for example, 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 ensures 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 notices 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 needs of the user.

Step 5

Send the file DUMPEPHEMS.txt to the target. The user notices that the ephemeris data is displayed on the Teseo-Suite (as shown below). Note that if no data is displayed then there is no ephemeris data in the flash.

```
$PSTMEPHEM,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00ecff212c00000ced2b287d102
1031f5b0da1b0eabad3c9277301316763b9f90011009184c003*59
$PSTMEPHEM,2,64,4205706270629898984 1f60034a3ff0017014e23c90ad20095fffeff40360000e59fd126b3f39
f04ddda0ca160ecc10ed28dacaa512bc74edb000300e21eff03*09
$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5fff0ff5b36000089e92c26d3a67
00364ca0da109f24862068422525c188929f700f201032bc703*5b
$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00200053370000a4b113261c5b2
40333740da1b1d91e956051cf7e3f6ed4b3f60004006fa5db03*00
$PSTMEPHEM,14,64,420570627062c5c5c5e10e007ea9ff0064058520a30ea60416000200772c000024c01b28451e
1f01c49f0ca10aeb5ff83bcf570002bc35acec000400a632ff03*6b
$PSTMEPHEM,21,64,4205706270622121 2188f9009da5ff00e7004622cd0aba00d9ff9efffd3500001a618a2634ba
500506010ea1e9f9fa926c745cac2cc31f84e700200044a6c403*3c
$PSTMEPHEM,25,64,42056c626c62b2b2b20c04008ca5ff0007fc3b250b0820fd5b00290079370000ada6bd26d78f
350664e90ca176ebc4a6c5e0fd26c93f03c6f00007004d12c003*3d
$PSTMEPHEM,30,64,420570627062b0b0b091f800caa6ff00cff8e2179e1355f999ffc0ff553500003f077326f97e
6c04c8140da10c14be42db05f853b7a66b34ef005e009ff7cd03*3e
```

Step 6

Send the file DUMPALMANAC.txt to the target. As in the previous step the user notices that the almanac data is displayed on the Teseo-Suite (as shown below).

Note: 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,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f19000a3ef92030088020*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,25,32,1942056359637605 2ffd0140760ca1408bfd26603c01c600e9d9b42002008020*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
```

Step 7

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

Step 8

The almanac and ephemeris data should now be saved in 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.

5.2 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.

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 uses 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 that user wants 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 copies everything from one bank to the other before swapping banks. It will then continue writing to the new bank until the same condition arises. Then it copies everything to the other bank and swaps 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 from 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 Teseo-Suite.

This example uses the following files:

- SUSPEND.txt
- RESUME.txt
- LOADEPHEMS.txt
- LOADALMANAC.txt

Step 1

Ensure that the connection is working and that the user can see NMEA data displayed on their the Teseo-Suite.

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 makes a copy of the data already within its memory before loading the new data into memory. This results 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 notices 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 what user needs.

Step 4

Send the file LOADEPHEMS.txt to the target. This loads 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 (`$PSTMHOT`). Note that it is important that they resume the operation of the GPS library before issuing the hot start command otherwise the hot start command fails. This is possible via the `$PSTMRESUME` command.

Step 5

Send the file LOADALMANAC.txt to the target. This loads 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.

6 Summary of text files used in the examples

6.1 File: SUSPEND.txt

```
$PSTMSUSPEND
```

6.2 File: RESUME.txt

```
$PSTMRESUME
```

6.3 File: DUMPEPHEMS.txt

```
$PSTMDUMPEPHEMS
```

6.4 File: DUMPALMANAC.txt

```
$PSTMDUMPALMANAC
```

6.5 File: LOADALMANAC.txt

```
$PSTMALMANAC,1,32,0142056314325b1c5efd0140020da14009730160ad61b900caffe12011088020*1d
$PSTMALMANAC,2,32,02420563034ab50634fd01406c0ca1402eacaa6047c64e005b741c20e4078020*15
$PSTMALMANAC,3,32,03420563483df0f537fd0140bb0ca140807d7c60237f19000a3ef92030088020*1c
$PSTMALMANAC,4,32,04420563f93a700633fd0140450da140447bab606fd202008ec97f201e208020*1a
$PSTMALMANAC,5,32,054205630d3765fc3ffd0140500da14033225260f08929006cf96f20e6808020*1
$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,25,32,1942056359637605 2ffd0140760ca1408bfd26603c01c600e9d9b42002008020*28
$PSTMALMANAC,26,32,1a420563fd87eb1d61fd0140bc0ca140e5e2006013041e001389e320f7ffbf20*22
$PSTMALMANAC,27,32,1b4205630e9e660834fd0140720da140313f28606565ae002a2d772016008020*7b
$PSTMALMANAC,28,32,1c4205631756300b57fd0000dc0ca1402f06562082c6a12050f344002a008000*25
$PSTMALMANAC,29,32,1d4205638f49d21b60fd0140090da1407880ff60c018d50000953520952980 20*73
$PSTMALMANAC,30,32,1e420563ca46c70045fd0140a00ca140baf75360466c3400e26e5020bf198020*28
```

6.6 File: LOADEPHEMS.txt

```
$PSTMEPHEM,1,64,42056a626a6281818170100009a9ff00cb05e920580e65052f00ecff212c00000ced2b287d102
1031f5b0da1b0eabad3c9277301316763b9f90011009184c003*59
$PSTMEPHEM,2,64,42057062706298989841f60034a3ff0017014e23c90ad20095fffeff40360000e59fd126b3f39
f04ddda0ca160ecc10ed28dacaa512bc74edb000300e21eff03*09
$PSTMEPHEM,5,64,4205706270626f6f6fd1f600fea6ff0076f8491883120ff9c5fff0ff5b36000089e92c26d3a67
00364ca0da109f24862068422525c188929f700f201032bc703*5b
$PSTMEPHEM,6,64,4205706270627d7d7d800800a4a6ff007506cf18ee1178050a00200053370000a4b113261c5b2
40333740da1b1d91e956051cf7e3f6ed4b3f60004006fa5db03*00
$PSTMEPHEM,14,64,420570627062c5c5c5e10e007ea9ff0064058520a30ea60416000200772c000024c01b28451e
1f01c49f0ca10aeb5ff83bcf570002bc35acec000400a632ff03*6b
$PSTMEPHEM,21,64,42057062706221212188f9009da5ff00e7004622cd0aba00d9ff9efffd3500001a618a2634ba
```

```
500506010ea1e9f9fa926c745cac2cc31f84e700200044a6c403*3c
$PSTMEPHEM,25,64,42056c626c62b2b2b20c04008ca5ff0007fc3b250b0820fd5b00290079370000ada6bd26d78f
350664e90ca176ebc4a6c5e0fd26c93f03c6f00007004d12c003*3d
$PSTMEPHEM,30,64,420570627062b0b0b091f800caa6ff00cff8e2179e1355f999ffc0ff553500003f077326f97e
6c04c8140da10c14be42db05f853b7a66b34ef005e009ff7cd03*3e
```

Appendix A Local geodetic datum tables

Table 155. Local geodetic datum for Africa

Region	Code	Code value
ADINDAN		
MeanSolution(Ethiopia-Sudan)	ADI-M	0
BurkinaFaso	ADI-E	1
Cameroon	ADI-F	2
Ethiopia	ADI-A	3
Mali	ADI-C	4
Senegal	ADI-D	5
Sudan	ADI-B	6
AFGOOYE		
Somalia	AFG	7
ARC_1950		
Mean_Solution	ARF-M	8
Botswana	ARF-A	9
Burundi	ARF-H	10
Lesotho	ARF-B	11
Malawi	ARF-C	12
Swaziland	ARF-D	13
Zaire	ARF-E	14
Zambia	ARF-F	15
Zimbabwe	ARF-G	16
ARC_1960		
Mean_Solution	ARS-M	17
Kenya	ARS-A	18
Tanzania	ARS-B	19
AYABELLE_LIGHTHOUSE		
Djibouti	PHA	20
BISSAU		
Guinea-Bissau	BID	21
CAPE		
South_Africa	CAP	22
CARTHAGE		
Tunisia	CGE	23
DABOLA		
Guinea	DAL	24
EUROPEAN_1950		
Egypt	EUR-F	73
Tunisia	EUR-T	83

Region	Code	Code value
LEIGON		
Ghana	LEH	25
LIBERIA_1964		
Liberia	LIB	26
MASSAWA		
Eritrea(Ethiopia)	MAS	27
MERCHICH		
Morocco	MER	28
MINNA		
Cameroon	MIN-A	29
Nigeria	MIN-B	30
M'PORALOKO		
Gabon	MPO	31
NORTH_SAHARA_1959		
Algeria	NSD	32
OLD_EGYPTIAN_1907		
Egypt	OEG	33
POINT_58		
Mean_Solution (BurkinaFaso-Niger)	PTB	34
POINTE_NOIRE_1948		
Congo	PTN	35
SCHWARZECK		
Namibia	SCK	36
SIERRA_LEONE_1960		
SierraLeone	SRL	37
VOIROL_1960		
Algeria	VOR	38

Table 156. Local geodetic datum for Asia

Region	Code	Code value
AIN_EL_ABD_1970		
Bahrain_Island	AIN-A	39
Saudi_Arabia	AIN-B	40
DJAKARTA(BATAVIA)		
Sumatra(Indonesia)	BAT	41
EUROPEAN_1950		
Iran	EUR-H	77
HONG_KONG_1963		
Hong_Kong	HKD	42
HU-TZU-SHAN		
Taiwan	HTN	43

Region	Code	Code value
INDIAN		
Bangladesh	IND-B	44
India-Nepal	IND-I	45
INDIAN_1954		
Thailand	INF-A	46
INDIAN_1960		
Vietnam(near_16DegNorth)	ING-A	47
ConSonIsland(Vietnam)	ING-B	48
INDIAN_1975		
Thailand	INH-A	49
Thailand	INH-A1	50
INDONESIAN_1974		
Indonesia	IDN	51
KANDAWALA		
SriLanka	KAN	52
KERTAU_1948		
WestMalaysia-Singapore	KEA	53
KOREAN_1995		
SouthKorea	KGS	54
NAHRWAN		
MasirahIsland(Oman)	NAH-A	55
UnitedArabEmirates	NAH-B	56
SaudiArabia	NAH-C	57
OMAN		
Oman	FAH	58
QATAR_NATIONAL		
Qatar	QAT	59
SOUTH_ASIA		
Singapore	SOA	60
TIMBALAI_1948		
Brunei-East_Malaysia	TIL	61
TOKYO		
MeanSolution	TOY-M	62
Japan	TOY-A	63
Okinawa	TOY-C	64
South Korea	TOY-B	65
South Korea	TOY-B1	66

Table 157. **Local geodetic datum for Australia**

Region	Code	Code value
AUSTRALIAN_1966		
Australia-Tasmania	AUA	67
AUSTRALIAN_1984		
Australia-Tasmania	AUG	68

Table 158. **Local geodetic datum for Europe**

Region	Code	Code value
CO-ORDINATE SYSTEM 1937 OF ESTONIA		
Estonia	EST	69
EUROPEAN_1950		
MeanSolution	EUR-M	70
WesternEurope	EUR-A	71
Cyprus	EUR-E	72
Egypt	EUR-F	73
England,ChannelIslands,Scotland,ShetlandIslands	EUR-G	74
England,Ireland,Scotland,ShetlandIslands	EUR-K	75
Greece	EUR-B	76
Iran	EUR-H	77
ItalySardinia	EUR-I	78
ItalySicily	EUR-J	79
Malta	EUR-L	80
Norway,Finland	EUR-C	81
Portugal,Spain	EUR-D	82
Tunisia	EUR-T	83
EUROPEAN_1979		
MeanSolution	EUS	84
HJORSEY_1955		
Iceland	HJO	85
IRELAND_1965		
Ireland	IRL	86
ORDNANCE SURVEY OF GREAT BRITAIN 1936		
MeanSolution	OGB-M	87
England	OGB-A	88
England,IsleOfMan,Wales	OGB-B	89
Scotland,ShetlandIslands	OGB-C	90
Wales	OGB-D	91
ROME_1940		
Sardinia	MOD	92
S-42(PULKOVO_1942)		
Hungary	SPK-A	93

Region	Code	Code value
Poland	SPK-B	94
Czechoslovakia*	SPK-C	95
Latvia	SPK-D	96
Kazakhstan	SPK-E	97
Albania	SPK-F	98
Romania	SPK-G	99
S-JTSK		
Czechoslovakia	CCD	100

Table 159. Local geodetic datum for North America

Region	Code	Code value
CAPE_CANAVERAL		
MeanSolution(Florida,Bahamas)	CAC	101
NORTH AMERICAN 1927		
MeanSolution	NAS-C	102
WesternUnitedStates	NAS-B	103
EasternUnitedStates	NAS-A	104
Alaska(ExcludingAleutianIslands)	NAS-D	105
AleutianIslands(East180°W)	NAS-V	106
AleutianIslands(West180°W)	NAS-W	107
Bahamas(Excluding San Salvador Island)	NAS-Q	108
SanSalvadorIsland	NAS-R	109
CanadaMeanSolution(Including Newfoundland)	NAS-E	110
Alberta,BritishColumbia	NAS-F	111
EasternCanada	NAS-G	112
Manitoba,Ontario	NAS-H	113
NorthwestTerritories,Saskatchewan	NAS-I	114
Yukon	NAS-J	115
CanalZone	NAS-O	116
Caribbean	NAS-P	117
CentralMerica	NAS-N	118
Cuba	NAS-T	119
Greenland	NAS-U	120
Mexico	NAS-L	121
NORTH AMERICAN 1983		
Alaska(ExcludingAleutianIslands)	NAR-A	122
AleutianIslands	NAR-E	123
Canada	NAR-B	124
CONUS	NAR-C	125
Hawaii	NAR-H	126
Mexico,CentralAmerica	NAR-D	127

Table 160. **Local geodetic datum for South America**

Region	Code	Code value
BOGOTA OBSERVATORY		
Colombia	BOO	128
CAMPO NCHAUSPE 1969		
Argentina	CAI	129
CHUA ASTRO		
Paraguay	CHU	130
CORREGO ALEGRE		
Brazil	COA	131
PROVISIONAL SOUTH AMERICAN 1956		
MeanSolution	PRP-M	132
Bolivia	PRP-A	133
Northern Chile(near 19°S)	PRP-B	134
Southern Chile(near 43°S)	PRP-C	135
Colombia	PRP-D	136
Ecuador	PRP-E	137
Guyana	PRP-F	138
Peru	PRP-G	139
Venezuela	PRP-H	140
PROVISIONAL SOUTH CHILEAN		
Southern Chile(near 53°S)	HIT	141
SOUTH AMERICAN 1969		
MeanSolution	SAN-M	142
Argentina	SAN-A	143
Bolivia	SAN-B	144
Brazil	SAN-C	145
Chile	SAN-D	146
Colombia	SAN-E	147
Ecuador (Excluding Galapagos Islands)	SAN-F	148
Baltra,Galapagos Islands	SAN-J	149
Guyana	SAN-G	150
Paraguay	SAN-H	151
Peru	SAN-I	152
Trinidad and Tobago	SAN-K	153
Venezuela	SAN-L	154
SOUTH AMERICAN GEOCENTRIC REFERENCE SYSTEM(SIRGAS)		
South America	SIR	155
ZANDERIJ		
Suriname	ZAN	156

Table 161. **Local geodetic datum for Atlantic Ocean**

Region	Code	Code value
ANTIGUA ISLAND ASTRO 1943		
Antigua,Leeward Islands	AIA	157
ASCENSION ISLAND 1958		
Ascension Island	ASC	158
ASTRO DOS 71/4		
St.Helena Island	SHB	159
BERMUDA 1957		
Bermuda Islands	BER	160
CAPE CANAVERAL		
Mean Solution (Bahamas and Florida)	CAC	101
DECEPTION ISLAND		
Deception Islandand Antarctica	DID	161
FORT THOMAS 1955		
Nevis, St.Kitts and Leeward Islands	FOT	162
GRACIOSA BASE SW 1948		
Faial, Graciosa, Pico, SaoJorge and Terceira Islands (Azores)	GRA	163
HJORSEY 1955		
Iceland	HJO	85
ISTS 061 ASTRO 1968		
South Georgia Island	ISG	164
L.C. 5 ASTRO 1961		
Cayman Brac Island	LCF	165
MONTSERRAT ISLAND ASTRO 1958		
Montserrat and Leeward Islands	ASM	166
NAPARIMA,BWI		
Trinidad and Tobago	NAP	167
OBSERVATORIO METEOROLOGICO 1939		
Corvo and Flores Islands (Azores)	FLO	168
PICO DE LAS NIEVES		
Canary Islands	PLN	169
PORTO SANTO 1936		
Porto Santo and Madeira Islands	POS	170
PUERTO RICO		
Puerto Rico and Virgin Islands	PUR	171
QORNOQ		
South Greenland	QUO	172
SAO BRAZ		
Sao Miguel and Santa Maria Islands (Azores)	SAO	173
SAPPER HILL 1943		
East Falkland Island	SAP	174

Region	Code	Code value
SELVAGEM GRANDE 1938		
Salvage Islands	SGM	175
TRISTAN ASTRO 1968		
Tristan da Cunha	TDC	176

Table 162. Local geodetic datum for Indian Ocean

Region	Code	Code value
ANNA 1 ASTRO 1965		
Cocos Islands	ANO	177
GAN 1970		
Republic of Maldives	GAA	178
ISTS 073 ASTRO 1969		
Diego Garcia	IST	179
KERGUELEN ISLAND 1949		
Kerguelen Island	KEG	180
MAHE 1971		
Mahe Island	MIK	181
REUNION		
Mascarene Islands	REU	182

Table 163. Local geodetic datum for Pacific Ocean

Region	Code	Code value
AMERICAN SAMOA 1962		
American Samoa Islands	AMA	183
ASTRO BEACON “E” 1945		
Iwo Jima	ATF	184
ASTRO TERN ISLAND (FRIG) 1961		
Tern Island	TRN	185
ASTRONOMICAL STATION 1952		
Marcus Island	ASQ	186
BELLEVUE (IGN)		
Efate and Erromango Islands	IBE	187
CANTON ASTRO 1966		
Phoenix Islands	CAO	188
CHATHAM ISLAND ASTRO 1971		
Chatham Island (New Zealand)	CHI	189
DOS 1968		
Gizo Island (New Georgia Islands)	GIZ	190
EASTER ISLAND 1967		
Easter Island	EAS	191

Region	Code	Code value
GEODETIC DATUM 1949		
New Zealand	GEO	192
GUAM 1963		
Guam	GUA	193
GUX I ASTRO		
Guadalcanal Island	DOB	194
INDONESIAN 1974		
Indonesia	IDN	51
JOHNSTON ISLAND 1961		
Johnston Island	JOH	195
KUSAIE ASTRO 1951		
CarolineIslands, Fed.States of Micronesia	KUS	196
LUZON		
Philippines (Excluding Mindanao Island)	LUZ-A	197
Mindanao Island	LUZ-B	198
MIDWAY ASTRO 1961		
Midway Islands	MID_A	199
Midway Islands	MID_B	200
OLD_HAWAIIAN		
Mean Solution	OHA-M	201
Hawaii	OHA-A	202
Kauai	OHA-B	203
Maui	OHA-C	204
Oahu	OHA-D	205
OLD HAWAIIAN		
Mean Solution	OHI-M	206
Hawaii	OHI-A	207
Kauai	OHI-B	208
Maui	OHI-C	209
Oahu	OHI-D	210
PITCAIRN ASTRO 1967		
Pitcairn Island	PIT	211
SANTO (DOS) 1965		
Espirito Santo Island	SAE	212
VITI LEVU 1916		
VitiLevuIsland (Fiji Islands)	MVS	213
WAKE-ENIWETOK 1960		
Marshall Islands	ENW	214
WAKE ISLAND ASTRO 1952		
Wake Atoll	WAK	215

Table 164. **Non-satellite derived transformation parameter**

Region	Code	Code value
BUKIT RIMPAH		
Bangka and Belitung Islands (Indonesia)	BUR	216
CAMP AREA ASTRO		
Camp McMurdo Area, Antarctica	CAZ	217
EUROPEAN 1950		
Iraq, Israel, Jordan, Kuwait, Lebanon, Saudi Arabia, Syria	EUR-S	218
GUNUNG SEGARA		
Kalimantam (Indonesia)	GSE	219
HERAT NORTH		
Afghanistan	HEN	220
HERMANNSKOGEL		
Slovenia, Croatia, Bosnia and Herzegovina, Serbia	HER	221
INDIAN		
Pakistan	IND_P	222
PULKOVO 1942		
Russia	PUK	223
TANANARIVE OBSERVATORY 1925		
Madagascar	TAN	224
VOIROL 1874		
Tunisia, Algeria	VOI	225
YACARE		
Uruguay	YAC	226

Table 165. **Terrestrial reference systems**

	Code	Code value
GLONASS		
PZ90.2	PZ90_2	227
PZ90.11	PZ90_11	254

Appendix B Acronyms, definitions, and reference documents

Table 166. Acronyms and definitions

Keyword	Definition
Accuracy	Deviation of a GPS-based calculated position from the true position.
AGC	Automatic gain control
ADC	Analogue to digital converter.
Almanac	Contains the information about all available satellites, their orbit data and time of their clocks.
ANF	Adaptive notch filter.
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).
BEIDOU	China's regional navigation satellite system.
Checksum	Calculated from the transmitted characters of a message by "ex-OR"ing the 8 bit character values excluding delimiters $ and *.
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 are not available, too.
CWM	Narrow band carrier wave monitor
BeiDou	China's global navigation satellite system (also known as Beidou-2, BD2)
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.
FDE	False detection exclusion.
GALILEO	Europe's global navigation satellite system.
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, almanac and ephemeris already available. High time accuracy is required.
ICB	Inter channel bias.
IFB	Inter frequency biases.

Keyword	Definition
IMU	Inertial measurement unit.
IRNSS	Indian regional navigational satellite system.
Lat	Latitude - Angular difference of a given position to the equator. Values include 0°-90° either north or south.
Lat-Ref	Latitude 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.
MCNF	Moving normalized correlation function
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).
OSNMA	Galileo open service navigation message authentication.
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.
RAIM	Receiver autonomous integrity monitoring.
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.
SPS	Standard positioning service.
Static Position Filtering	Algorithm to detect that the GPS receiver doesn´t move and position output is kept stable.
Teseo-Suite	PC Windows program to use, configure and evaluate Teseo GNSS solution available on www.st.com
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 almanac, position and time availability. Ephemeris are not 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 167. Reference documents

Document name	Document title
UM3397	Teseo VI and Teseo APP2–NMEA observables description
UM3428	Teseo VI and Teseo APP2–Firmware configuration

Revision history

Table 168. **Document revision history**

Date	Revision	Changes
12-Nov-2024	1	Initial release.

Contents

List of tables