Automotive Product Group
Automotive Infotainment Division

Navigation & Multimedia System & Architecture
STAGPS™ Interface and Testing Procedures

1 Introduction

The STAGPS™ library is able to provide predicted ephemerides to the GNSS engine in a time frame less than the usual time (about 30 seconds) needed to download real ephemeris from the sky. This reduces considerably the time to get fix especially in critical environments when the ephemeris download time could be very long.

STAGPS™ library includes two operational modes (autonomous and server based) which work in parallel to provide each time the best predicted ephemeris for each satellite.

STAGPS™ autonomous solution works using the past real ephemeris (downloaded from the sky and stored in its internal database) to extrapolate the parameter of future ephemeris (up to 5 days of prediction). For these reason the STAGPS™ autonomous performances (in terms of position accuracy using predicted ephemeris) are strictly dependent on the real ephemeris database content. In normal usage of STAGPS™ autonomous, the system automatically uploads the real ephemeris into its database as soon as new ephemerides are downloaded from the sky. This means that the global content of the real ephemeris input database is determined by the history of device running periods in the past.

STAGPS™ served based solution uses a network connection to download the data package (seed) for predicted ephemeris extrapolation. In this case predicted ephemeris availability is not dependant on the usage of GPS device because data downloaded from server always includes information for all satellites. In addition the server based solution allows valid predictions up to 7 days in the future.

According to the run-time data availability, the STAGPS™ engine has the capability to decide, satellite by satellite, which is the best prediction solution in order to ensure the highest level of accuracy. If server based data is not available at system start-up, the STAGPS™ engine starts in autonomous mode; as soon as a new server based seed is available (the seed downloading process has been successfully completed and it has been passed to the STAGPS™ library) it starts to inject also the server based predicted ephemeris replacing, if necessary, the autonomous predictions. A mixed condition (some satellites have autonomous predictions and other satellites have server based predictions) is also possible.

The GNSS software with STAGPS™ capability implements a command interface at the NMEA level to allow interaction with the STAGPS™ library. It supports commands to enable/disable the STAGPS™ functionality or to upload ephemeris and seed into the STAGPS™ working memory. Most of implemented commands could be used for testing the STAGPS™ performance in different working scenarios.

This document describes all NMEA commands supported by STAPGS™ version 5, including some testing procedures to simulate different working scenarios. The NMEA commands and the testing procedures described in this document work for both GPS and GLONASS, with experimental support for GALILEO and BEIDOU[1].

[1] The device used may not support all types of constellation. See your device specifications document for the list of supported constellations.

2 Contents

2.1 Index

2.2 List of Tables

None.

2.3 List of Figures

None.

3 Document Management

3.1 Revision History

Rev	Date	Author	Notes
1.0	2011-07-27	A. Di Girolamo	Initial draft release.
1.1	2012-01-27	A. Di Girolamo	Document Review.
1.2	2013-03-26	G. Gogliettino	GLONASS has been added. $PSTMEPHBRCONOFF command added. Ephemeris_tool usage has been added.
1.3	2013-04-11	G. Gogliettino	$PSTMAGLO command added.
1.4	2013-04-17	G. Gogliettino	Added Ephemeris_tool alternative usage.
1.5	2014-01-13	G. Gogliettino	Section 7 has been improved.
1.6	2014-12-17	A. Furno	General review.
1.7	2015-03-25	G. Gogliettino	In section 4.6 the encoding of *ephem_data* field has been added.
1.8	2015-06-08	G. Gogliettino	Section 7 has been improved.
1.9	2015-07-01	G. Gogliettino	Sections 8.1 and 8.3 have been updated
1.10	2015-07-13	G. Gogliettino	Section 4.8 has been added
1.11	2015-10-08	G. Gogliettino	The encoding of ephemerides in C structures has been added to section 4.6. The table in section 4.7 has been expanded.
1.12	2015-10-09	G. Gogliettino	C sample code has been added to section 4.6.
1.13	2015-10-09	G. Gogliettino	BEIDOU ephemeris encoding has been added to section 4.6. Command $PSTMDUMPINDB description has been added to section 4.9
1.14	2015-11-19	G. Gogliettino	Section 1 has been improved.

3.2 Acronyms

None

3.3 Reference Documents

None

3.4 Contact info

Name	Mail
A. Di Girolamo	andrea.di-girolamo@st.com
F. Boggia	fulvio.boggia@st.com

4 NMEA commands for STAGPS™ testing

$PSTMSTAGPSONOFF	Turns ON/OFF the STAGPS™ engine
$PSTMSTAGPSINVALIDATE	Clears data stored in the STAGPS™ internal database
$PSTMREPHUPONOFF	Turns ON/OFF the real ephemeris update in the STAGPS™ database.
$PSTMPEPHUPONOFF	Turns ON/OFF the usage for positioning of predicted ephemeris.
$PSTMGETAGPSSTATUS	Returns the status of the STAGPS™ internal processing.
$PSTMINDBEPHEM	Uploads real ephemeris into the STAGPS™ database.
$PSTMEPHBRCONOFF	Enables or disables the use of broadcast ephemeris
$PSTMSTAGPSSETCONSTMASK	Switches among the ST-AGNSS constellation.
$PSTMDUMPINDB	Dumps real ephemeris out from the STAGPS™ database.

4.1 $PSTMSTAGPSONOFF

Turn ON/OFF the STAGPS™ engine; it affects both autonomous and server based solutions.

Command:

$PSTMSTAGPSONOFF,<*param*><cr><lf>

Parameter	Format	Description
param	Decimal, 1 digit	ON/OFF status : 0 = the STAGPS™ engine is suspended. 1 = the STAGPS™ engine is started

Results:

According to the command parameter, the STAGPS™ engine is started or suspended. The messages “$PSTMPOLSTARTED” or “$PSTMPOLSUSPENDED” is returned if the engine has been started or suspended. In case of error “$PSTMPOLONOFFERROR” is returned.

4.2 $PSTMSTAGPSINVALIDATE

Clear data stored in the STAGPS™ internal database. The input parameter allows selection of data to be cleared.

Command:

$PSTMSTAGPSINVALIDATE,<*param*><cr><lf>

Parameter	Format	Description
param	Decimal, 1 digitChar	Selects which database should be erased: 1 = Clear the real ephemeris database (only autonomous). 2 = Clear the satellite seeds database (autonomous and server based) 4 = Clear the satellite polys database (autonomous and server based) 7 = Clear all databases

Results:

According to the command parameter, the internal STAGPS™ databases will be erased. The input parameter should be considered as a mask where the first three bits select the database to be cleared (e.g. using 3 as input parameter the real ephemeris and seed databases will be cleared). At the end of a successful invalidation process the message "$PSTMSTAGPSINVALIDATEOK" is returned. In case of errors, $PSTMSTAGPSINVALIDATEERROR is returned.

4.3 $PSTMREPHUPONOFF

Turn ON/OFF the real ephemeris update in the STAGPS™ database. This command affects only the STAGPS™ autonomous solution. The ON/OFF status, configured by this command, is stored into the backup memory and so it is preserved on any system reset/reboot. It should be changed using again the same command with proper input parameter.

Command:

$PSTMREPHUPONOFF,<*param*><cr><lf>

Parameter	Format	Description
param	Decimal, 1 digit	Real ephemeris update ON/OFF status : 0 = real ephemeris update is disabled 1 = real ephemeris update is enabled

Results:

According to the command parameter, the real ephemeris update can be enabled or disabled. When the real ephemeris update is disabled the STAGPS™ internal real ephemeris database doesn't change even if new ephemerides are downloaded from the sky. In this case the predicted ephemerides are based always on the same ephemerides that are present in the database. This feature is useful to test the positioning performances when the age of predicted ephemeris increases. When the real ephemeris updated is enabled the STAGPS™ internal database is updated every time a new ephemeris is downloaded from the sky. In this way also the predicted ephemeris will be updated according to latest real ephemeris availability in the database.

4.4 $PSTMPEPHUPONOFF

Turn ON/OFF the usage for positioning of predicted ephemeris. This command affects both STAGPS™ autonomous and server based solutions. The ON/OFF status configured by this command is stored into the backup memory and so it is preserved at any system reset/reboot. It can be changed using again the same command with proper input parameter.

Command:

$PSTMPEPHUPONOFF,<*param*><cr><lf>

Parameter	Format	Description
param	Decimal, 1 digit	Predicted ephemeris usage ON/OFF status : 0 = predicted ephemeris usage is disabled 1 = predicted ephemeris usage is enabled

Results:

According to the command parameter, the usage of predicted ephemeris can be enabled or disabled. When predicted ephemeris usage is disabled the predicted ephemeris are never used for positioning even if the STAGPS™ is running updating continuously its internal databases and predictions. When predicted ephemeris usage is enabled, predicted ephemerides are provided to the GNSS engine and they are used in the position evaluation. They are also kept updated up to the real ephemeris of each visible satellite has been downloaded from the sky.

4.5 $PSTMGETAGPSSTATUS

Return the status of the STAGPS™ internal processing.

Command:

$PSTMGETAGPSSTATUS<cr><lf>

Results

The system sends back the STAGPS™ status in the following message:

$PSTMAGPSSTATUS,<*status*>*<*checksum*><cr><lf>

Parameter	Format	Description
status	Decimal, 1 digit	0 = the STAGPS™ processing is completed. Any number different from zero means that the STAGPS™ processing is ongoing and so the ephemeris prediction data has been not completely generated.

4.6 $PSTMINDBEPHEM

Upload the real ephemeris into the STAGPS database for a specific satellite.

Command:

$PSTMINDBEPHEM,<*sat_id*>,<*ephem_data_size*>,<*ephem_data*>*<cr><lf>

Parameter	Format	Description
sat_id	Decimal, up to 3 digits	Satellite PRN
ephem_data_size	Decimal, 2 digits	Size expressed in byte of the ephemeris data field.
ephem_data	Hexadecimal	Ephemeris data string (hex format)

The encoding of the *ephem_data* field is described in the following two tables.

The encoding for GPS ephemeris is:

Bits	*Structure Member*	*Description*
16	week	Week number of the Issue of Data
16	toe	Time of week for ephemeris epoch
16	toc	Time of week for clock epoch
8	iode1	Issue of data 1
8	iode2	Issue of data 2
10	iodc	Issue of data clock
14	i_dot	Rate of inclination angle.
8	reserved	
24	omega_dot	Rate of right ascension.
8	reserved	Must be 0.
16	crs	Amplitude of the sine harmonic correction to the orbit radius.
16	crc	Amplitude of the cosine harmonic correction to the orbit radius.
16	cus	Amplitude of the sine harmonic correction to the argument of latitude.
16	cuc	Amplitude of the cosine harmonic correction to the argument of latitude.
16	cis	Amplitude of the sine harmonic correction to the angle of inclination.
16	cic	Amplitude of the cosine harmonic correction to the angle of inclination.
16	motion_difference	Mean motion difference from computed value
16	reserved	Must be 0.
32	inclination	Inclination angle at reference time
32	e	Eccentricity.
32	root_A	Square root of major axis.
32	mean_anomaly	Mean anomaly at reference time.
32	omega_zero	Longitude of ascending node of orbit plane at weekly epoch.

32	perigee	Argument of perigee.
8	time_group_delay	Estimated group delay differential.
8	af2	Second order clock correction.
16	af1	First order clock correction.
22	af0	Constant clock correction.
1	reserved	Reserved for use by GNSS library – must be 1
1	reserved	Reserved for use by GNSS library – must be 1
1	reserved	Reserved for use by GNSS library – must be 1
1	available	Contains 1 if ephemeris is available, 0 if not
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy
1	reserved	Must be 0.
4	accuracy	Accuracy

The previous GPS encoding can be represented in C language with the following structure:

```
typedef struct gps_ephemeris_s
{
  unsigned int week                 : 16;
  unsigned int toe                  : 16;

  unsigned int toc                  : 16;
  unsigned int iode1                : 8;
  unsigned int iode2                : 8;

  unsigned int iodc                 : 10;
  unsigned int i_dot                : 14;
  unsigned int spare1               : 8;

  unsigned int omega_dot            : 24;
  unsigned int reserved1            : 2; // must be 0
  unsigned int reserved2            : 6; // must be 0

  unsigned int crs                  : 16;
  unsigned int crc                  : 16;

  unsigned int cus                  : 16;
  unsigned int cuc                  : 16;

  unsigned int cis                  : 16;
  unsigned int cic                  : 16;

  unsigned int motion_difference    : 16;
  unsigned int reserved3            : 10; // must be 0
  unsigned int spare3               : 6;

  unsigned int inclination          : 32;
  unsigned int eccentricity         : 32;
  unsigned int root_a               : 32;
  unsigned int mean_anomaly         : 32;
  unsigned int omega_zero           : 32;
  unsigned int perigee              : 32;

  unsigned int time_group_delay     : 8;
  unsigned int af2                  : 8;
  unsigned int af1                  : 16;

  unsigned int af0                  : 22;
  unsigned int reserved4            : 1; // must be 1
  unsigned int reserved5            : 1; // must be 1
  unsigned int reserved6            : 1; // must be 1
```

```
  unsigned int available            : 1;
  unsigned int health               : 1;
  unsigned int reserved7            : 1; // must be 0
  unsigned int accuracy             : 4;
} gps_ephemeris_t;
```

The encoding for GLONASS ephemeris is:

Bits	*Structure Member*	*Description*
16	week	Week number of the Issue of Data.
16	toe	Time of week for ephemeris epoch.
4	toe_lsb	Time of week for ephemeris epoch (LBS).
11	NA	Calendar day number within the four-year period since the beginning of last leap year (almanac).
7	tb	Time of ephemeris index.
2	M	Type of satellite 00=GLONASS 01=GLONASS-M .
2	P1	Time interval between two adjacent tb parameters.
1	P3	Number of satellites for which almanac is transmitted within this frame 0=4 1=5.
1	P2	Flag of oddness ("1") or evenness ("0") of the value of tb
1	P4	Flag to show that ephemeris parameters are present.
2	KP	Notification on forthcoming leap second correction of UTC
1	Reserved	
27	xn	Satellite PZ-90 x coordinate at epoch tb.
5	xn_dot_dot	Satellite PZ-90 x velocity at epoch tb.
24	xn_dot	Satellite PZ-90 x acceleration component at epoch tb.
5	n	Slot number (1…24).
3	Bn	Healthy flags.
27	yn	Satellite PZ-90 y coordinate at epoch tb.
5	yn_dot_dot	Satellite PZ-90 y acceleration component at epoch tb.
24	yn_dot	Satellite PZ-90 y velocity at epoch tb.
8	age_h	Age of predicted ephemeris (hours)
27	zn	Satellite PZ-90 z coordinate at epoch tb.
5	zn_dot_dot	Satellite PZ-90 z acceleration component at epoch tb.
24	zn_dot	Satellite PZ-90 z velocity at epoch tb.
8	reserved	Must be 0.

11	gamma_n	Satellite clock frequency drift at epoch tb.
5	E_n	Age of the ephemeris information.
4	freq_id	Frequency ID
12	reserved	
22	tau_n	Satellite clock correction at epoch tb.
10	reserved	Must be 0.
32	tau_c	GLONASS to UTC(SU) time correction.
22	tau_GPS	GLONASS to GPS system time correction.
10	reserved	
11	NT	Calendar day number of ephemeris within the four-year period since the beginning of last leap year.
5	N4	Four-year interval number starting from 1996.
12	tk	Satellite time referenced to the beginning of the frame.
4	FT	Predicted satellite user range accuracy at time tb
32	reserved	
5	m_available	Must be 0x1F
1	nvm_reliable	Must be 1.
26	spare	
25	reserved	
1	available	Contains 1 if ephemeris is available, 0 if not.
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy.
1	reserved	Must be 0.
4	reserved	

The previous GLONASS encoding can be represented in C language with the following structure:

```
typedef struct glonass_ephemris_s
{
  unsigned int week                : 16;
  unsigned int toe                 : 16;

  unsigned int toe_lsb             : 4;
  unsigned int NA                  : 11;
  unsigned int tb                  : 7;
  unsigned int M                   : 2;
  unsigned int P1                  : 2;
  unsigned int P3                  : 1;
  unsigned int P2                  : 1;
  unsigned int P4                  : 1;
  unsigned int KP                  : 2;
  unsigned int spare0              : 1;

  unsigned int xn                  : 27;
  unsigned int xn_dot_dot          : 5;
  unsigned int xn_dot              : 24;
  unsigned int n                   : 5;
  unsigned int Bn                  : 3;

  unsigned int yn                  : 27;
  unsigned int yn_dot_dot          : 5;
  unsigned int yn_dot              : 24;
  unsigned int spare4              : 8;

  unsigned int zn                  : 27;
  unsigned int zn_dot_dot          : 5;
  unsigned int zn_dot              : 24;
  unsigned int reserved1           : 2; // must be 0
  unsigned int reserved2           : 6; // must be 0
  unsigned int gamma_n             : 11;
  unsigned int E_n                 : 5;
  unsigned int freq_id             : 4;
  unsigned int spare3              : 12;

  unsigned int tau_n               : 22;
  unsigned int reserved3           : 10; // must be 0

  unsigned int tau_c               : 32;

  unsigned int tau_GPS             : 22;
  unsigned int spare5              : 10;
```

```
  unsigned int NT                   : 11;
  unsigned int N4                   : 5;
  unsigned int tk                   : 12;
  unsigned int FT                   : 4;

  unsigned int spare7               : 32;

  unsigned int m_available          : 5;
  unsigned int nvm_reliable         : 1;
  unsigned int spare8               : 26;

  unsigned int spare9               : 25;
  unsigned int available            : 1;
  unsigned int health               : 1;
  unsigned int reserved4            : 1; // must be 0
  unsigned int spare10              : 4;
} glonass_ephemeris_t;
```

The encoding for GALILEO ephemeris is:

Bits	*Structure Member*	*Description*
16	week	Week number of the Issue of Data
14	toe	Time of week for ephemeris epoch
2	reserved	
16	toc	Time of week for clock epoch
10	iod_nav	Issue of data
8	SISA	Signal In Space Accuracy
10	reserved	Must be 0.
10	BGD_E1_E5a	E1-E5a Broadcast Group Delay
10	BGD_E1_E5b	E1-E5b Broadcast Group Delay
2	E1BHS	E1-B Signal Health Status
32	inclination	Inclination angle at reference time
32	eccentricity	Eccentricity.
32	root_a	Square root of major axis.
32	mean_anomaly	Mean anomaly at reference time.
32	omega_zero	Longitude of ascending node of orbit plane at weekly epoch.
32	perigee	Argument of perigee.
14	i_dot	Rate of inclination angle.
1	available	Contains 1 if ephemeris is available, 0 if not
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy
16	motion_difference	Mean motion difference from computed value
16	crs	Amplitude of the sine harmonic correction to the orbit radius.
16	crc	Amplitude of the cosine harmonic correction to the orbit radius.
16	cus	Amplitude of the sine harmonic correction to the argument of latitude.
16	cuc	Amplitude of the cosine harmonic correction to the argument of latitude.

16	cis	Amplitude of the sine harmonic correction to the angle of inclination.
16	cic	Amplitude of the cosine harmonic correction to the angle of inclination.
24	omega_dot	Rate of right ascension.
6	SVID	Satellite Identification.
1	E1BDVS	E1-B Data Validity Status
1	reserved	Must be 0.
8	reserved	Must be 0.
16	reserved	Must be 0.
6	af2	Second order clock correction.
21	af1	First order clock correction.
5	word_available	Must be 0x1F.
31	af0	Constant clock correction.
1	reserved	
6	reserved	Must be 0
26	reserved	Reserved for use by GNSS library – must be 1
1	reserved	Must be 0.

The previous GALILEO encoding can be represented in C language with the following structure:

```
typedef struct galileo_ephemeris_s
{
  unsigned int week                 : 16;
  unsigned int toe                  : 14;
  unsigned int reserved1            : 2; // must be 0

  unsigned int toc                  : 14;
  unsigned int iod_nav              : 10;
  unsigned int SISA                 : 8;

  unsigned int reserved2            : 10;  // must be 0
  unsigned int BGD_E1_E5a           : 10;
  unsigned int BGD_E1_E5b           : 10;
  unsigned int E1BHS                : 2;

  unsigned int inclination          : 32;
  unsigned int eccentricity         : 32;
  unsigned int root_a               : 32;
  unsigned int mean_anomaly         : 32;
  unsigned int omega_zero           : 32;
  unsigned int perigee              : 32;

  unsigned int i_dot                : 14;
  unsigned int available            : 1;
  unsigned int health               : 1;
  unsigned int motion_difference    : 16;

  unsigned int crs                  : 16;
  unsigned int crc                  : 16;
  unsigned int cus                  : 16;
  unsigned int cuc                  : 16;
  unsigned int cis                  : 16;
  unsigned int cic                  : 16;

  unsigned int omega_dot            : 24;
  unsigned int SVID                 : 6;
  unsigned int E1BDVS               : 1;
  unsigned int reserved3            : 1; // must be 0

  unsigned int af2                  : 6;
  unsigned int af1                  : 21;
  unsigned int word_available       : 5;

  unsigned int af0                  : 31;
  unsigned int spare0               : 1;
```

```
  unsigned int reserved4            : 6; // must be 0
  unsigned int spare1               : 26;

} galileo_ephemeris_t;
```

The encoding for BEIDOU ephemeris is:

Bits	*Structure Member*	*Description*
32	inclination	Inclination angle at reference time
32	eccentricity	Eccentricity.
32	root_a	Square root of major axis.
32	mean_anomaly	Mean anomaly at reference time.
32	omega_zero	Longitude of ascending node of orbit plane at weekly epoch.
32	perigee	Argument of perigee.
17	toe	Time of week for ephemeris epoch
10	time_group_delay	Estimated group delay differential.
5	aode	Issue of data, ephemeris
24	omega_dot	Rate of right ascension.
8	A0	Ionospheric Delay Model Parameter α_0
24	af0	Constant clock correction.
8	A1	Ionospheric Delay Model Parameter α_1
20	sow	Seconds of week
11	af2	Second order clock correction.
1	is_geo	1 for Geostationary satellites, otherwise 0
22	af1	First order clock correction.
10	subframe_avail	Must be 0x3FF.
16	motion_difference	Mean motion difference from computed value
8	A2	Ionospheric Delay Model Parameter α_2
8	A3	Ionospheric Delay Model Parameter α_3
18	crs	Amplitude of the sine harmonic correction to the orbit radius.
8	B2	Ionospheric Delay Model Parameter β_2
4	urai	User range accuracy index
2	reserved	Must be 0.

18	crc	Amplitude of the cosine harmonic correction to the orbit radius.
8	B3	Ionospheric Delay Model Parameter β_3
5	aodc	Issue of data, clock
1	spare	
18	cus	Amplitude of the sine harmonic correction to the argument of latitude.
14	i_dot	Rate of inclination angle.
18	cuc	Amplitude of the cosine harmonic correction to the argument of latitude.
8	B0	Ionospheric Delay Model Parameter β_0
6	spare	
18	cis	Amplitude of the sine harmonic correction to the angle of inclination.
8	B1	Ionospheric Delay Model Parameter β_1
6	reserved	Must be 0.
18	cic	Amplitude of the cosine harmonic correction to the angle of inclination.
1	nvm_reliable	Must be 1.
11	reserved	Must be 0.
2	spare	
17	toc	Time of week for clock epoch
13	week	Week number of the Issue of Data
1	available	Contains 1 if ephemeris is available, 0 if not
1	health	Contains 1 if the satellite is unhealthy, 0 if healthy

The previous BEIDOU encoding can be represented in C language with the following structure:

```
typedef struct beidou_ephemeris_s
{
  unsigned int inclination       : 32;
  unsigned int eccentricity      : 32;
  unsigned int root_a            : 32;
  unsigned int mean_anomaly      : 32;
  unsigned int omega_zero        : 32;
  unsigned int perigee           : 32;

  unsigned int toe               : 17;
  unsigned int time_group_delay  : 10;
  unsigned int aode              : 5;

  unsigned int omega_dot         : 24;
  unsigned int A0                : 8;

  unsigned int af0               : 24;
  unsigned int A1                : 8;

  unsigned int sow               : 20;
  unsigned int af2               : 11;
  unsigned int is_geo            : 1;

  unsigned int af1               : 22;
  unsigned int subframe_avail    : 10;

  unsigned int motion_difference : 16;
  unsigned int A2                : 8;
  unsigned int A3                : 8;

  unsigned int crs               : 18;
  unsigned int B2                : 8;
  unsigned int urai              : 4;
  unsigned int reserved1         : 2;

  unsigned int crc               : 18;
  unsigned int B3                : 8;
  unsigned int aodc              : 5;
  unsigned int spare0            : 1;

  unsigned int cus               : 18;
  unsigned int i_dot             : 14;

  unsigned int cuc               : 18;
  unsigned int B0                : 8;
```

```
  unsigned int spare1            : 6;

  unsigned int cis               : 18;
  unsigned int B1                : 8;
  unsigned int reserved2         : 6;

  unsigned int cic               : 18;
  unsigned int nvm_reliable      : 1;
  unsigned int reserved3         : 1;
  unsigned int reserved4         : 10;
  unsigned int spare4            : 2;

  unsigned int toc               : 17;
  unsigned int week              : 13;
  unsigned int available         : 1;
  unsigned int health            : 1;

} beidou_ephemeris_t;
```

Note:

- A *little endian* architecture has been assumed. For example, in the case of GPS PRN #1 ephemeris where week number is 1831 (= 0x**0727**) and toe is 15750 (= 0x**3D86**), then the NMEA command string will begin with "*$PSTMINDBEPHEM,1,64,**2707863d***…".
- For the scale of data in the table for each constellation, see the respective ICDs.
- The $PSTMDUMPINDB command can be used to verify if an ephemeris, which has been injected using the $PSTMINDBEPHEM command, has been correctly sent – see section 4.9 for details.

Results:

The ephemeris data string is decoded and stored into the first free slot of the STAGPS input database. If more than one ephemeris is uploaded for each satellite, the oldest ephemeris must be provided first then the others following the temporal sequence up to the most recent ephemeris.

This command has the same syntax of the $PSTMEPHEM messages used for GPS, GLONASS, GALILEO and BEIDOU ephemeris upload or retrieved using the $PSTMDUMPEPHEMS command. The STAGPS ephemeris upload command strings can be generated starting from the real ephemeris dump ($PSTMDUMPEPHEMS) and replacing the "$PSTMEPHEM" with "$PSTMINDBEPHEM" and the checksum field with the correct checksum. Checksum is required for this command.

If more than one $PSTMINDBEPHEM commands need to be issued, between two consecutive commands there must be at least a 5 second delay.

C sample code:

The following example shows how to use the above structures. The example is for GPS ephemeris, but it also applies to GLOANSS, GALILEO and BEIDOU with very few changes.

```
void convert_gps_eph(int sat_id, gps_ephemeris_t *ephem_ptr, char
nmea_output[512])
{
  unsigned char* eph_ptr = (unsigned char*)(ephem_ptr);
  int  index, index2;

  index = sprintf((char *)nmea_output,"$PSTMINDBEPHEM,%i,%i,", sat_id,
sizeof(gps_ephemeris_t));

  for (index2 = 0; index2 < sizeof(gps_ephemeris_t); index2++)
  {
    index += sprintf((char *)(nmea_output +
index),"%02x",eph_ptr[index2]);
  }
}
```

4.7 $PSTMEPHBRCONOFF

Turn ON/OFF the real ephemeris usage in the GNSS engine. The ON/OFF status, configured by this command, is stored into the backup memory and so it is preserved at any system reset/reboot. The status can be changed using again the same command with proper input parameter.

Command:

$PSTMEPHBRCONOFF,<*sat_id*>,<*param*><cr><lf>

Parameter	Format	Description
sat_id	Decimal, 1 digit	Satellite PRN or: 0 = all satellites (GPS + GLONASS) 252 = all BEIDOU satellites only 253 = all GALILEO satellites only 254 = all GLONASS satellites only 255 = all GPS satellites only
param	Decimal, 1 digit	Real ephemeris usage ON/OFF status : 0 = real ephemeris usage is disabled 1 = real ephemeris usage is enabled

Results:

According to the command parameters, the real ephemeris usage can be enabled or disabled. When real ephemeris usage is disabled the new ephemerides are not downloaded from the sky anymore. In this case only the predicted ephemerides can be used if they are enabled. This feature is useful to test the positioning performances with predicted ephemerides only.

This command does not affect real ephemerides which have been already downloaded from the sky, so they are used until they expire. The $PSTMWARM command can be used to clear current real ephemerides just after that the $PSTMEPHBRCONOFF command has been issued.

4.8 $PSTMSTAGPSSETCONSTMASK

Set the ST-AGNSS constellation mask. It allows switching the ST-AGNSS constellation at run-time. All previous ST-AGNSS data will be erased.

Command:

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

Parameter	Format	Description
constellation_mask	*Decimal, 1 digit*	*It is a bit mask where each bit enable/disable a specific constellation independently by the others:* *bit 0: GPS constellation enabling/disabling* *bit 1: GLONASS constellation enabling/disabling* *bit 3: GALILEO constellation enabling/disabling* *bit 7: BEIDOU constellation enabling/disabling*

Results:

- If success the following message is sent:

$PSTMSTAGPSSETCONSTMASKOK,<constellation_mask>*<checksum><cr><lf>

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

$PSTMSTAGPSSETCONSTMASKERROR*<checksum><cr><lf>

Note:

GALILEO and BEIDOU support is still experimental and should not be used in production environment.

Examples:

Enabling GPS only:

$PSTMSETCONSTMASK,1

Enabling GLONASS only:

$PSTMSETCONSTMASK,2

Enabling GPS and GLONASS:

$PSTMSETCONSTMASK,3

4.9 $PSTMDUMPINDB

Return all the ephemerides stored into the STAGPS™ internal ephemeris database.

Command:

$PSTMDUMPINDB<cr><lf>

Results

The system, for each entry of the STAGPS™ internal ephemeris database, sends back a couple of messages containing one ephemeris from the STAGPS™ ephemeris database each:

$PSTMDUMPINDB,<*sat_id*>,<*ephem_data_size*>,<*ephem_data*>*<cr><lf>

$PSTMDUMPINDBINFO,<*sat_id*>,<*week_number*>,<*time_of_week*>,<*iodc*>*<cr><lf>

Parameter	Format	Description
sat_id	Decimal, up to 3 digits	Satellite PRN
ephem_data_size	Decimal, 2 digits	Size expressed in byte of the ephemeris data field.
ephem_data	Hexadecimal	Ephemeris data string (hex format)
week_number	Decimal, up to 4 digits	Week number of the time of the ephemeris
time_of_week	Decimal, up to 5 digits	Time of week of the time of the ephemeris
iodc	Decimal, up to 3 digits	Issue of data, clock

The encoding of the *ephem_data* field is described in the same of $PSTMINDBEPHEM command described in section 4.6.

Note:

This command can be used to verify if an ephemeris, which has been injected using the $PSTMINDBEPHEM command, has been correctly received by checking it is present in the STAGPS™ ephemeris database.

5 NMEA output messages for STAGPS™ testing

5.1 $PSTMAGPS

This message has the same syntax of standard NMEA GSA message. Each parameter in the satellites PRN fields is an integer number that reports the satellite PRN and, in case a satellite is using a predicted ephemeris, it reports also the age of predicted ephemeris available for that satellite. They are generated using the formula: satID + 32 * STAGPS_AGE_DAYS where STAGPS_AGE_DAYS is the number of days from current time back to the most recent ephemeris used for STAGPS predictions. If a satellite has no predicted ephemeris (STAGPS_AGE_DAYS = 0) the satellite parameter, reported in the sentence, is exactly the satellite PRN.

- STAGPS_AGE_DAYS = 1: most recent ephemeris has been downloaded from 0 up to 24 hours in the past.
- STAGPS_AGE_DAYS = 2: most recent ephemeris has been downloaded from 24 up to 48 hours in the past.
- STAGPS_AGE_DAYS = 3: most recent ephemeris has been downloaded from 48 up to 72 hours in the past.
- STAGPS_AGE_DAYS = 4: most recent ephemeris has been downloaded from 72 up to 96 hours in the past.
- STAGPS_AGE_DAYS = 5: most recent ephemeris has been downloaded from 96 up to 120 hours in the past.

This message could be used to replace the standard GSA in all devices where STAGPS is enabled. If STAGPS is not enabled, it behaves in the same way of NMEA GSA message.

NMEA message list bitmask: 0x10000000 – This message is not enabled by default, it should be enabled adding it in the NMEA message list.

Format:

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

Parameter	Format	Description
Mode	“M” or “A”	Operating Mode: M = Manual, A = Auto (2D/3D)
CurrentMode	Decimal, 1 digit	Current Mode: 1 = no fix available 2 = 2D 3 = 3D

SatPRN1...N	Decimal, 2 digits	Satellites list used in position fix (max N 12)
PDOP	Decimal, 3 digits	Position Dilution of Precision, from 0.0 to 99.0
HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, from 0.0 to 99.0
VDOP	Decimal, 3 digits	Vertical Dilution of Precision, from 0.0 to 99.0
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

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

The example above should be read in the following way:

- Satellites 5, 24, 30, 12 don't have predicted ephemeris (they are reported as in the case of standard GSA message – basically all satellites reported with a number less or equal 32 have no predicted ephemeris).
- Satellite 21 has a predicted ephemeris 2 days old (85 is reported in the sentence above).
- Satellite 7 has predicted ephemeris 3 days old (103 is reported in the sentence above).
- Satellite 16 has predicted ephemeris 1 day old (48 is reported in the sentence above).

Here are two simple decoding functions to get satellite ID and ages:

```
Age = (int)((<reported number> - 1) / 32)
Satid = <reported number> - 32 * Age
```

Note:

This message works for GPS satellites only.

If no GPS satellites are available this message will be not displayed.

5.2 $PSTMAGLO

This message has the same syntax of standard NMEA GSA message. Each parameter in the satellites PRN fields is an integer number that reports the satellite PRN and, in case a satellite is using a predicted ephemeris, it reports also the age of predicted ephemeris available for that satellite. They are generated using the formula: satID + 32 * STAGPS_AGE_DAYS where STAGPS_AGE_DAYS is the number of days from current time back to the most recent ephemeris used for STAGPS predictions. If a satellite has no predicted ephemeris (STAGPS_AGE_DAYS = 0) the satellite parameter, reported in the sentence, is exactly the satellite PRN.

- STAGPS_AGE_DAYS = 1: most recent ephemeris has been downloaded from 0 up to 24 hours in the past.
- STAGPS_AGE_DAYS = 2: most recent ephemeris has been downloaded from 24 up to 48 hours in the past.
- STAGPS_AGE_DAYS = 3: most recent ephemeris has been downloaded from 48 up to 72 hours in the past.
- STAGPS_AGE_DAYS = 4: most recent ephemeris has been downloaded from 72 up to 96 hours in the past.
- STAGPS_AGE_DAYS = 5: most recent ephemeris has been downloaded from 96 up to 120 hours in the past.

This message could be used to replace the standard GSA in all devices where STAGPS is enabled. If STAGPS is not enabled, it behaves in the same way of NMEA GSA sentence.

NMEA message list bitmask: 0x10000000 – This message is not enabled by default, it should be enabled adding it in the NMEA message list.

Format:

```
$PSTMAGLO,<Mode>,<CurrentMode>,[<SatPRN1>],...,[<SatPRNN>],
   <PDOP>,<HDOP>,<VDOP>,
   <checksum><cr><lf>
```

Parameter	Format	Description
Mode	“M” or “A”	Operating Mode: M = Manual, A = Auto (2D/3D)
CurrentMode	Decimal, 1 digit	Current Mode: 1 = no fix available 2 = 2D 3 = 3D
SatPRN1...N	Decimal, 2 digits	Satellites list used in position fix (max N 12)
PDOP	Decimal, 3 digits	Position Dilution of Precision, from 0.0 to 99.0

HDOP	Decimal, 3 digits	Horizontal Dilution of Precision, from 0.0 to 99.0
VDOP	Decimal, 3 digits	Vertical Dilution of Precision, from 0.0 to 99.0
checksum	Hexadecimal,2 digits	Checksum of the message bytes without *<checksum><cr><lf> characters.

Example:

```
$PSTMAGLO,A,3,84,109,196,78,71,,,,,,,,2.4,1.9,1.5*66
```

The example above should be read in the following way:

- Satellites 84, 78, 71 don't have predicted ephemeris (they are reported as in the case of standard GSA message – basically all satellites reported with a number between 65 and 92 have no predicted ephemeris).
- Satellite 77 has a predicted ephemeris 1 days old (109 is reported in the sentence above).
- Satellite 68 has predicted ephemeris 4 days old (196 is reported in the sentence above).

Here are two simple decoding functions to get satellite ID and ages:

```
Age = (int)((<reported number> - 65) / 32)
Satid = <reported number> - 32 * Age
```

Note:

This message works for GLONASS satellites only.

If no GLONASS satellites are available this message will be not displayed.

6 Real ephemeris uploading procedure

The STAGPS™ performances depend on the real ephemeris availability into its input database. If the STAGPS™ enabled device is used regularly (e.g. every day) its database is kept updated and predicted ephemeris availability is guaranteed for all satellites visible in the area of usage. The STAGPS engine selects the best sets of real ephemeris to make the most accurate predictions. Good predictions are achieved using 2 ephemerides per satellite (24 hours apart). The time to generate predicted ephemeris depend on the number of real ephemeris used (1 or 2) and on the time distance between the ephemeris (only in the 2 ephemeris case).

To allow testing different scenario, a procedure for real ephemeris uploading is supported. All the steps are listed below:

- $PSTMSTAGPSINVALIDATE,7
- $PSTMSTAGPSONOFF,0
- $PSTMINDBEPHEM,<sat_id_1>,<size>,<ephem_data_1>*<checksum>
- $PSTMINDBEPHEM,<sat_id_2>,<size>,<ephem_data_2>*<checksum>
- ……
- ……
- $PSTMINDBEPHEM,<sat_id_n>,<size>,<ephem_data_n>*<checksum>
- $PSTMSTAGPSONOFF,1

As soon as the ephemeris have been loaded into the database, the STAGPS™ engine starts to calculate ephemeris predictions. The time needed to complete the prediction's generation depends on the real ephemeris uploaded (especially on the time distance between ephemeris of the same satellite). The predictions phase is completed when all the satellites positions are generates. The generation process is fully completed when the STAGPS status, reported by $PSTMGETAGPSSTATUS command, is zero.

Ephemeris prediction generation process is made sequentially satellite by satellite. If device is turned OFF (or rebooted) when the process is still running, only the current generation will be broken and it restarts from the beginning the next time device is ON. All completed generations are preserved in the STAGPS™ internal database.

When predictions have been generated, the test session can start.

7 STAGPS™ Autonomous Testing Procedure

The main feature introduced by a GPS device with STAGPS™ capability is the ephemeris availability as soon as device is turned ON (and also GPS fix availability) even if device is used in critical environments where the ephemeris downloading from the sky is difficult(urban canyon, low signal strengths etc.). The STAGPS™ engine uses the old real ephemeris, stored in its internal database, to predict the future ephemeris. Having device ON with sky visibility, the internal database is continuously updated with new ephemeris, predictions are updated as well and the expiration time of prediction is moved forward. If the ephemeris database is not updated (e.g. the update is disabled) the predictions can be used for some days in the future and then they expire. The expiration time and the accuracy of GPS position based on predicted ephemeris relyon the real ephemeris (number of ephemeris per satellite and time distance between the ephemeris) used for prediction. The most important and useful test to measure the STAGPS™ performance is the WARM start (device is restarted clearing the real ephemeris stored in the GPS backup memory). In these conditions, a device without STAGPS™ capability usually gets the fix in a time longer than 18 seconds. A device with STAGPS™ capability will get the fix in the typical time of HOT start conditions.

The procedures, reported in this sections must be performed after real ephemeris have been uploaded in the STAGPS™ database (see previous sections) and device has been left running for a while in order to allow the ephemeris prediction generation.

- **Simple WARM start test session:**

  Using the standard NMEA command $PSTMWARM, device is restarted in WARM start conditions and the time to first fix and the position accuracy is evaluated. This test can be performed several times in order to have a good statistical result. The time to first fix (TTFF) should be in the range of common HOT start. The first fix is made using only the predicted ephemeris and so the position accuracy analysis is useful to evaluate the STAGPS™ performances.

- **WARM start test session with fixed STAGPS™ real ephemeris database.**

  Before starting the WARM start test session the update of STAGPS™ real ephemeris database should be disabled. This can be done using the NMEA command: $PSTMREPHUPONOFF,0 as described in the supported NMEA commands section (NOTE: the database update disabled/enabled status is preserved even if device is turned OFF/ON, it should be changed using again the NMEA command).

  A longer (e.g. several days) WARM start test session should be performed and the TTFF and position accuracy should be analyzed. Because the real ephemeris update has been disabled, the predictions based on the initial database layout (preloaded with desired scenario – see section 6) will be used up to they will expire. This test is useful to measure STAGPS™ performances time and to checkthe accuracy degradation approaching the ephemeris expiration time.

  For this test the following NMEA commands can be used:

  1) **$PSTMREPHUPONOFF,0** (to disable future real ephemeris updates).
  2) **$PSTMSTAGPSINVALIDATE,7** (to invalidate current STAGPS databases).
  3) **$PSTMSTAGPSONOFF,0** (to suspend STAGPS).
  4) For each satellite a **$PSTMINDBEPHEM** command which takes the ephemeris to inject as parameter should be sent (see section 6).

5) **$PSTMSTAGPSONOFF,1** (to restart STAGPS - after this the generation and the propagation for new seeds start).
6) **$PSTMPEPHUPONOFF,1** (to enable predicted ephemerides).
7) **$PSTMEPHBRCONOFF,0,0** (to disable all real ephemeris usage).
8) **$PSTMGETAGPSSTATUS** (to check the status of STAGPS seed generation and propagation, 0 = done) .
9) Once **$PSTMAGPSSTATUS** = 0 (done), the test can be started, for instance issuing a **$PSTMWARM** command every 1 minute.

Please note that the steps 2 to 5 are the same described in section 6.

8 Ephemeris_tool command line utility

"Ephemeris_tool.exe" is a command line tool which converts RINEX ephemeris file into a NMEA commands file which can be sent to the NMEA port of the device.

This tool is able to extract ephemeris data from a GPS or GLONASS RINEX file and write them into a text file. This file contains a sequence of "$PSTMINDBEPHEM" NMEA commands ready to be sent to the device.

8.1 Usage

The tool can be invoked from the command line in the following way:

```
Ephemeris_tool <mode> <year> <day of the year> <hour> <min> [<RINEX
file directory>]
```

Where:

<mode> is:

"-n" for GPS with NMEA output or

"-g" for GLONASS with NMEA output or

"-nt" for GPS with Testing Tool script output or

"-gt" for GLONASS with Testing Tool script output

<year> is 4-digit year of the ephemerides to convert,

<day of the year> is the day of the year of the ephemerides to convert,

<hour> is the hour from 0 to 23 of the ephemerides to convert (for GPS must be multiple of two)

<min> is 2-digit minutes of the ephemerides to convert (00 for GPS, 15 or 45 for GLONASS)

<RINEX file directory> is the directory containing the RINEX files

Internet FTP site for RINEX v.3 ephemeris files is:

```
ftp://cddis.gsfc.nasa.gov/gnss/data/campaign/mgex/daily/rinex3/2015/brdm
```

Notes:

- The ephemeris ftp folder contains all the daily ephemeris files;
- The name of ephemeris files are in the "*brdm<DDD>0.<YY>p*" format, where *<DDD>* is the day of the year and *<YY>* are the year (last two digits). For example, ephemeris file name for 11th of February 2015 is "*brdm0420.15p*".

- The ephemeris files downloaded from the server are is a compressed (*.Z*) format (for example, the compressed file for "*brdm0420.15p*" will be "*brdm0420.15p.Z*"). They need to be decompressed before using them (a compression utility is needed – for example "7-zip").
- The URL provided here is for the year 2015. To select a different year, "2015" should be changed to the desired year in the URL string.

The compressed ephemeris file should be downloaded and its content should be extracted; after that "Ephemeris_tool" should be called with desired options.

Important: the name of RINEX file must **not** be changed.

For example, to convert GLONASS ephemeris of 11 February 2015 at 22:15:

```
Ephemeris_tool.exe -g 2015 42 22 15 \.
```

Last parameter is optional; if the file is in the same folder of the tool then it can be omitted.

8.2 Output

The output file is a text file containing the NMEA sentences to be sent to the device.

Here is an example of the output file content (some lines have been omitted):

```
$PSTMINDBEPHEM,73,64,b30623420400400070101a0713a00401c91e0e009a1b020019
374088d2d2380000000000c5e00200000000000000000000009d01000000003f0000000
0000002
$PSTMINDBEPHEM,68,64,b306234204004000c2694406f943a1025c5d0a09f999030066
90dc0d5f8a2a0002040000fea705000000000000000000000009d01000000003f0000000
0000002
…
$PSTMINDBEPHEM,88,64,b30623420400400038b4e30cba0d9b18dac70a09be5aaa001d
f2ca8ada2b070001040000175c01000000000000000000000009d01000000003f0000000
0000002
```

In the case the GPSTestingTool should be used, it is possible to generate a script file which can be sent to the device by the GPSTestingTool itself.

In that case the "-nt" or "-gt" mode parameters should be used in place of "-n" or "-g" respectively.

8.3 Alternative usage

Ephemeris_tool can be also used to send NMEA script files to a COM port.

For example this feature allows the user to send the NMEA script generated from the tool directly to the device.

To send an NMEA script file to the COM port the tool can be invoked from the command line in the following way:

```
Ephemeris_tool <mode> <COM port number> <NMEA file name> <delay>
[<baudrate>]
```

Where:

<mode> is "-s" to send an NMEA script directly to the COM port

<COM port number> is the number of the COM port to send the NMEA script

<NMEA file name> is the name of the NMEA script file to be sent

<delay> is the delay in seconds between two consecutive NEMA commands

<baudrate> (optional) is the COM port speed in bit/sec (the default is 115200).

When injecting ephemerides a delay at least of 3 seconds must be inserted between two consecutive $PSTMINDBEPHEM sentences, so <delay> should be 3 or more.

For example, to send the “EphmGlo-13-105-22-15.txt” file to the COM port #2 with a 3 seconds delay between two consecutive sentences:

```
Ephemeris_tool.exe -s 2 EphmGlo-13-105-22-15.txt 3
```

9 Disclaimer

Please Read Carefully:

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

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

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

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

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

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

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

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

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

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

STMicroelectronics group of companies

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

http://www.st.com