SF-6908A
GPS/GLONASS 接收机
使 用 说 明 书

版本号：1.1.005

http://www.unistarchips.com

北京联星科通微电子技术有限公司

目　录

SF-6908A GPS/GLONASS 接收机
使用说明书

欢迎选用SF-6908A GPS/GLONASS 接收机！用户使用SF-6908A GPS/GLONASS 接收机前请仔细阅读产品的使用说明书。本公司竭诚为用户提供完善的售前售后技术服务，为用户解决使用中的各种技术上和应用上的问题。

1. 产品概述

SF-6908A GPS/GLONASS 接收机是24通道GPS/GLONASS双系统L1频段C/A码卫星定位导航单片接收机，可同时接收来自GPS(美国)和GLONASS(俄罗斯)卫星导航定位系统的信号，解算输出高精度的位置和速度参数，具有4维导航功能和强大的数字接口；适用于航海、航空和航天使用环境中的各类载体导航定位。SF-6908A GPS/GLONASS 接收机的信号处理能力强，能够在高噪声和不规律电磁干扰情况下提供高精度的、位置、速度、高度和时间参数，保持导航参数的可靠连续地输出。SF-6908A GPS/GLONASS 接收机实物如下图所示(图1)。

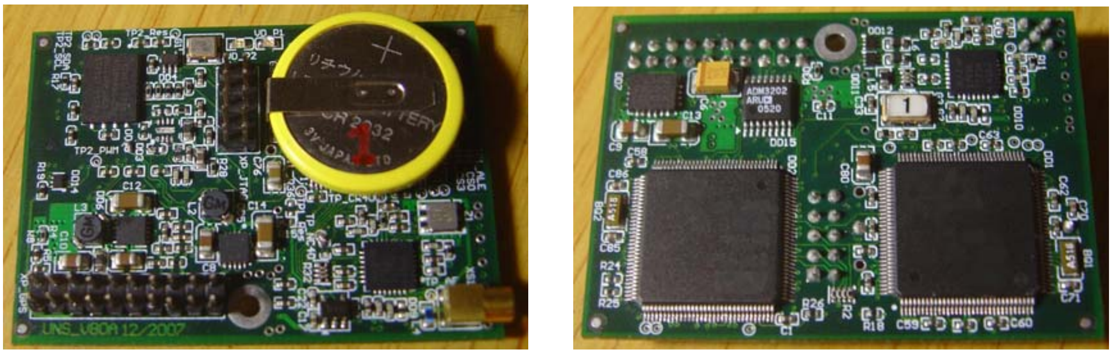

图 1.　SF-6908A GPS/GLONASS 接收机实物照片

2. 主要技术性能指标

2.1 冷启动

冷启动定位时间小于150秒。

注：SF-6908A GPS/GLONASS 接收机超过三个月没有加电工作，或者从5000公里以外运到本地第一次加电启动，或冷启动工作未超过15分钟关机时，属于冷启动。

2.2 温启动

温启动定位时间小于90秒。

注：SF-6908A GPS/GLONASS 接收机冷启动工作超过15分钟关机，距上次定位时间大于2小时再次加电时工作时，属于温启动。

2.3 热启动

热启动定位时间小于30秒。

注： SF-6908A GPS/GLONASS 接收机冷启动工作超过15分钟关机，距上次定位时间小于2小时，再次加电工作时，属于热启动。

2.4 重捕获

重捕获时间小于1秒。

注： SF-6908A GPS/GLONASS 接收机工作中，因天线受到短时间遮挡收不到卫星信号中断定位，天线不被遮挡再次定位的时间，属于重捕获时间。

2.5 导航定位精确度

位置：≤ 10 米，RMS

高度：≤ 15 米， RMS

速度：≤ 0.05 米/秒， RMS

时间：≤1微秒（UTC）

秒脉冲：≤ 40 纳秒

2.6 动态环境

速度：≤ 550米/秒, 可扩展到 8000米/秒

高度：≤ 18,000米，可扩展到 40,000米

加速度：≤ 100g

加加速度：≤ 20 g/s

2.7 电源功耗

直流：2.4V～5.5V

功率：＜0.6 瓦 (max)

后备电池：可保持数据10年

2.8 物理参数

外形尺寸： 50(长)×35(宽)×13(高) 毫米（见图2）

重量：＜ 20克

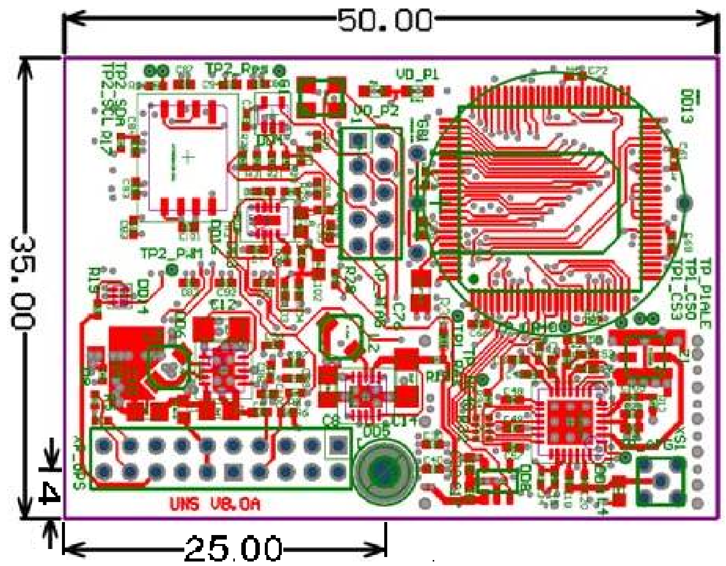

图2. 外形尺寸图

2.9 输入接口

卫星信号输入：MMCX天线接口

电源输入：从20脚数字信号接口中11、12针输入

2.10 输出接口

SF-6908A GPS/GLONASS 接收机有两个串口输出，均为标准的RS232电平。串口A为编程口；串口B为数据输出口，波特率为4800bps－115200bps可调（典型值

38400bps），8位数据位，1位停止位，无奇偶检验位，并支持NMEA-0183 协议。

数据输出口B：标准RS-232串行口，速率115.2Kbps，更新率1～10 Hz。

通信协议： NMEA0183

2.11可靠性

平均无故障工作时间（MTBF）: 20000小时

2.12 环境条件

⑴温度

工作温度：-40° C ～ +80° C

存储温度：-55° C ～ +85° C

⑵冲击

满足GJB-150《军用设备环境实验方法》中规定的冲击要求

⑶振动

满足GJB-150 《军用设备环境实验方法》中规定的振动要求

⑷湿热

满足GJB-150 《军用设备环境实验方法》中规定的湿热要求

⑸盐雾

满足GJB-150《军用设备环境实验方法》中规定的盐雾要求

⑹霉菌

满足GJB-150 《军用设备环境实验方法》中规定的霉菌要求

3. 接口定义

SF-6908A OEM板的对外接口采用20脚数字信号接口和MMCX天线接口。MMCX天线接口与天线连接，20脚数字信号接口用于定位数据信息及秒脉冲信号输出，以及引入电源。

20脚数字信号接口包括两个标准RS232串行口，以及电源和PPS等（见图3），其脚位定义如下：

1针为标准RS-232串行接口A数据发送(TXDA)

2针为标准RS-232串行接口A数据接收(RXDA)

3针为标准RS-232串行接口A地(GND）

4针为标准RS-232串行接口B地(GND)

5针为标准RS-232串行接口B数据发送(TXDB)

6针为标准RS-232串行接口B数据接收(RXDB)

7针为系统复位(XRESET)

8针为系统复位(BOOT)

9针为电源负极

10针电源负极

11针为电源正极(PWR_IN)

12针为电源正极(PWR_IN)

13针为系统检测(PWRONOFF)

14针为空

15针为接地

16针为秒脉冲输出(1PPS)

17针为系统检测(EVENT)

18针为系统检测(COMMSW)

19针为系统检测(LED1 RED)

20针为系统检测(LED1 GRN)

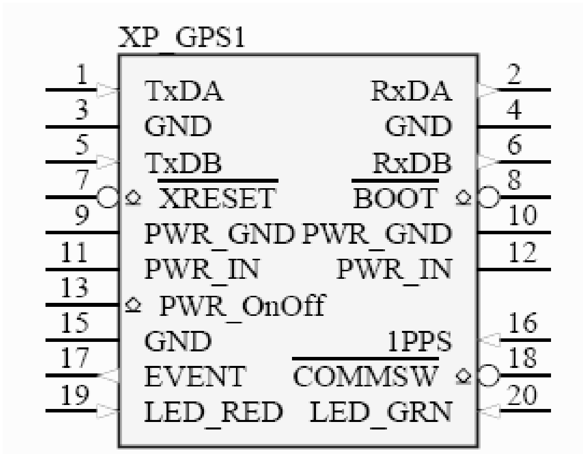

图3. 接口脚位定义图

4. 安装方法

SF-6908A的安装如下图所示（图4）

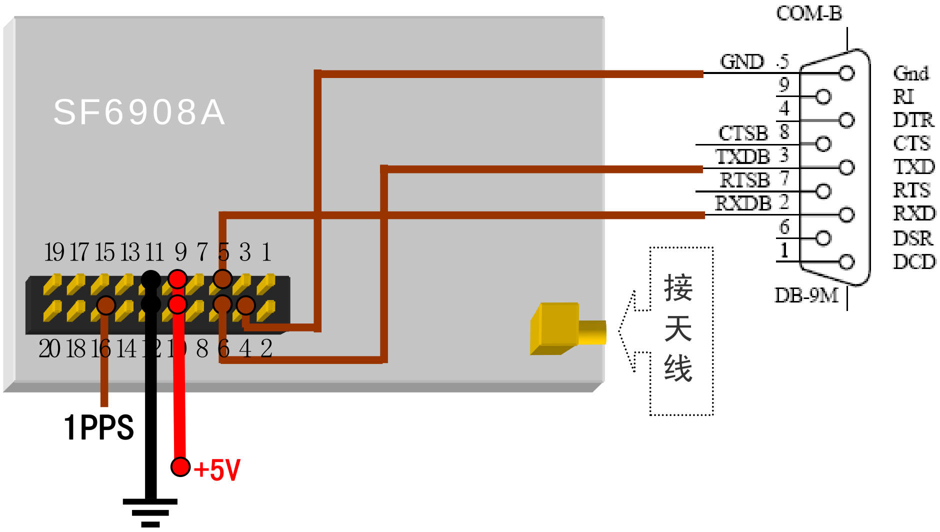

图4　SF-6908A安装示意图

5. 注意事项

SF-6908A OEM板使用中应注意以下问题：

(1)天线与SF-6908A OEM板接口的MMCX接口，插拔时要竖直对准中央的插针，安装时要确认已结合紧密。

(2)尽量避免其他电磁波对天线和接收机的干扰。

(3)天线安装位置要保证天线的视野足够大。

(4)不要带电插拔天线和20脚数字信号接口，以防损坏击穿芯片。

(5)不要直接用手拿取SF-6908A OEM板，以防静电损坏击穿芯片。

附录一　SF-6908A OEM板输出协议

SF-6908A OEM板的串口输出为标准的RS232电平，波特率为38600bps，8位数据位，1位停止位，无奇偶检验位，并支持NMEA-0183 协议。

NMEA语句是以ASCII码形式输出，以“$”起始，接着是各类语句句头（GGA、GLL、GSA、GSV、RMC、VTG等），各语句由字段及校验和组成，最后再以CR（回车）、LF（换行）的信号做结束，每个字段都以“，”分隔开来，其格式如下：

输出语句：SF6803系列输出以“$GN”开头的GNS,RMC,GSA,GSV,VTG和GLL等语句。SF-6908A OEM板主要输出GNS和VTG语句。

1）GNS语句－GNSS定位数据：

例如：

$GNGNS,105452.67,2232.1491197,N,11356.9045141,E,AA,08,1.16,10.2553,,,*71

GNS语句格式

名称	举例	单位	说明
语句标志	$GNGNS		GNS语句句头
UTC时间	105452.67		hhmmss.ss
纬度	2232.1491197		ddmm.mmmmmmm
纬度标识	N		N-北纬，S－南纬
经度	11356.9045141		dddmm.mmmmmmm
经度标识	E		E-东经，W-西经
定位标识	AA		NN-未定位，AA-定位
定位星数	08		8颗星定位
HDOP值	1.16		
海拔高度	10.2553	米	
水平面分离度		米	大地座标平面和椭圆面的差
差分数据时龄			
差分站代号			
校验和	*71		用“*”隔开

2）VTG语句－航向角和地速：

例如：$GNVTG,333.3,T,,M,0.2,N,0.4,K,A*15

VTG语句格式

名称	举例	单位	说明
语句标志	$GNVTG		VTG语句句头
地面航向角	333.3	度	000～359度
定义	T		以真北为参考基准
地面航向角		度	
定义	M		以磁北为参考基准
地面速率	0.2	knots	
单位	N		knots（节）
地面速率	0.4	km/h	
单位	K		公里/小时
模式指示	A		A=自主定位，D=差分，E=估算，N=数据无效
校验和	*15		用“*”隔开

附录二 NMEA V3.0 Protocol

Messages: GGA, GLL, GSA, GSV, RMC and VTG.
NMEA Output Messages: the Engine board outputs the following messages as shown in Table 1:

Table 1 NMEA-0183 Output Messages

NMEA Record	Description
GGA	Global positioning system fixed data
GLL	Geographic position – latitude / longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

GGA-Global Positioning System Fixed Data
Table 2 contains the values of the following example: $GPGGA, 161229.487, 3723.2475, N, 12158.3416, W, 1, 07, 1.0, 9.0, M, , , ,0000*18

Table 2 GGA Data Format

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		Dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 2-1
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
＜CR＞＜LF＞			End of message termination

Table 3 Position Fix Indicators

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3	GPS PPS Mode, fix valid

GLL-Geographic Position –Latitude/Longitude
Table 3 contains the values of the following
Example: $GPGLL, 3723.2475, N, 12158.3416, W, 161229.487, A*2C

Table *3* GLL Data Format

Name	Example	Units	Description
Message ID	$GPGLL		GLL protocol header
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		Dddmm.mmmm
E/W Indicator	W		E=east or W=west
UTC Position	161229.487		hhmmss.ss
Status	A		A=data valid or V=data not valid
Checksum	*2C		
＜CR＞＜LF＞			End of message termination

GSA-GNSS DOP and Active Satellites
Table 4 contains the values of the following example:
$GPGSA, A, 3, 07, 02, 26, 27, 09, 04, 15, , , , , , 1.8,1.0,1.5*33

Table 4 GSA Data Format

Name	Example	Units	Description
Message ID	$GPGSA		GSA protocol header
Mode 1	A		See Table 4-2
Mode 2	3		See Table 4-1
Satellite Used	07		Sv on Channel 1
Satellite Used	02		Sv on Channel 2
….			….
Satellite Used			Sv on Channel 12
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		Horizontal Dilution of Precision
VDOP	1.5		Vertical Dilution of Precision
Checksum	*33		
＜CR＞＜LF＞			End of message termination

Table 4-1 Mode 1

Value	Description
1	Fix not available
2	2D
3	3D

Table 4-2 Mode 2

Value	Description
M	Manual-forced to operate in 2D or 3D mode
A	Automatic-allowed to automatically switch 2D/3D

GSV-GNSS Satellites in View
Table 5 contains the values of the following example:
$GPGSV, 2, 1, 07, 07, 79, 048, 42, 02, 51, 062, 43, 26, 36, 256, 42, 27, 27, 138, 42*71$GPGSV, 2, 2, 07, 09, 23, 313, 42, 04, 19, 159, 41, 15, 12, 041, 42*41

Table *5* GGA Data Format

Name	Example	Units	Description
Message ID	$GPGSV		GSV protocol header
Number of Messages[1]	2		Range 1 to 3
Messages Number[1]	1		Range 1 to 3
Satellites in View	07		
Satellite ID	07		Channel 1(Range 1 to 32)
Elevation	79	degrees	Channel 1(Maximum 90)
Azimuth	048	degrees	Channel 1(True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
….			….
Satellite ID	27		Channel 4(Range 1 to 32)
Elevation	27	degrees	Channel 4(Maximum 90)
Azimuth	138	degrees	Channel 4(True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
Checksum	*71		
＜CR＞＜LF＞			End of message termination

[1] Depending on the number of satellites tracked multiple messages of GSV data may be required.

RMC-Recommended Minimum Specific GNSS Data
Table 6 contains the values of the following example:
$GPRMC, 161229.487, A, 3723.2475, N, 12158.3416, W, 0.13, 309.62, 120598, ,*10

Table *6* GGA Data Format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Position	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation		degrees	E=east or W=west
Checksum	*10		
＜CR＞＜LF＞			End of message termination

VTG-Course Over Ground and Ground Speed

Table 7 contains the values of the following example:

$GPVTG, 309.62, T, , M, 0.13, N, 0.2, K*6E

Table 7 VTG Data Format

Name	Example	Units	Description
Message ID	$GPVTG		VTG protocol header
Course	309.62	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic
Speed	0.13	knots	Measured horizontal speed
Units	N		Knots
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometer per hour
Checksum	*6E		
＜CR＞＜LF＞			End of message termination

附录三 UniStar Binary Protocol V1.0

1. Time Messages

There are five time scales your receiver may handle:

- Receiver time (**Tr**)
- GPS system time (Tg)
- UTC (USNO) (Tu) (Universal Coordinated Time supported by the U.S. Naval Observatory).
- GLONASS system time (Tn)
- UTC(SU) (Ts) (Universal Coordinated Time supported by the State Time and Frequency Service, Russia).

"Receiver time" is the only time grid that is always available in your receiver (i.e., the other time grids from the above list may or may not be currently available). In fact, GNSS receiver always synchronizes its local time (aka "receiver time") with one of the four "global" time scales: GPS time, UTC(USNO), GLONASS time, or UTC(SU). The time grid thus selected is referred to as "receiver reference time" (**Trr**) hereafter in this section.

Warning: Currently GNSS receivers can be synchronized to GPS time only.

Different time systems may have different time notations (formats) associated with them (e.g., for GPS time, we use such terms as "week number", "time of week", etc.). Note, however, that the "receiver time" representation will *not* depend on the selected receiver reference time.

1.1 [RD] Receiver Date {6}

This message contains the "date" part of the full receiver time representation (Tr).

```
struct RcvDate {
+ u2 year;          // Current year [1..65534][]
+ u1 month;         // Current month [1..12] []
+ u1 day;           // Current day [1..31] []
+ u1 base;          // Receiver reference time [enum]
                    // 0 — GPS // 1 — UTC_USNO // 2 — GLONASS
                    // 3 — UTC_SU
                    // 4..254 — Reserved
+ u1 cs;            // Checksum
};
```

1.2 [RT] Receiver Time {5}

This message contains the "time of day" part of the full receiver time representation (Tr).

```
struct RcvTime {
+ u4 tod;           // Tr modulo 1 day (86400000 ms) [ms]
+ u1 cs;            // Checksum
```

```
};
```

1.3[TO] Receiver Reference Time to Receiver Time Offset {9}

```
struct RcvTimeOffset {
  f8 val;            // Trr – Tr [s]
+ u1 cs;             // Checksum
};
```

1.4 [DO] Derivative of Receiver Time Offset {5}

```
struct RcvTimeOffsetDot {
  f4 val;            // Time Derivative of (Trr - Tr) [s/s]
+ u1 cs;             // Checksum
};
```

1.5 [NT] GLONASS Time {7}

```
struct GLOTime {
  u4 tod;            // time of day [ms]
  u2 dn;             // GLONASS day number (modulo 4 years
                     // starting from 1996) []
+ u1 cs;             // Checksum
};
```

2. Position/VelocityMessages

The field “type” used the position/velocity messages may have the following values:

```
enum SolutionType {
  NO_SOLUTION = 0,
  STAND_ALONE = 1,
  CODE_DIFF = 2,
  PHASE_DIFF_FLOAT = 3,
  PHASE_DIFF_FIXED = 4,
  FIXED_POS = 5
};
```

Warning: If you receive a *Position/Velocity* message where the field “type” is set to “0”, this means that this message contains invalid data (except the “type” flag itself).

2.1 [PG] Geodetic Position {30}

```
struct GeoPos {
  f8 lat;            // latitude [rad]
  f8 lon;            // longitude [rad]
  f8 alt;            // ellipsoidal height [m]
  f4 pSigma;         // position SEP [m]
  u1 type;           // enum SolutionType
+ u1 cs;             // Checksum
};
```

2.2 [VG] NEU Velocity {18}

```
struct GeoVel {
  f4 lat;            // northing velocity [m/s]
  f4 lon;            // easting velocity [m/s]
  f4 alt;            // height velocity [m/s]
  f4 pSigma;         // velocity SEP [m/s]
  u1 type;           // enum SolutionType
+ u1 cs;             // Checksum
};
```

3. Satellite Data Messages

In this section we will focus on messages containing “satellite specific information”. These kinds of messages include satellite measurements (code and carrier phase observables, elevations, azimuths, etc.). Different applications may utilize different sets of measurements. In practice, it is almost impossible to select a combination of satellite measurements that would be “universal” yet compact enough. This explains our choice: we use a dedicated message for each particular measurement type. Every individual measurement message provides some specific (“homogeneous”) data for the satellites tracked. This approach allows the minimization of the data transfer overheads (note that the gain increases as the number of satellites grows).

To ease handling data corresponding to different satellites, we assign each satellite its *universal identifier* (USI). The message “Satellite Indices” ([SI]) establishes the one-to-one correspondence between the satellite's universal identifier and its index. If you know the satellite's index, you can easily retrieve this satellite specific data from the corresponding slot of the required measurement message.

3.1 [SI] Satellite Indices {nSats+1}

The following table points out the relationship between the satellites' universal identifier (USI) and their “true” system numbers (PRN numbers or frequency channel numbers).

USI values:

0	Not used .
1 - 37	USI = GPS SV PRN number. These slots are allocated to GPS satellites.
38 - 70	USI - 45 (signed) = GLONASS SV Frequency Channel Number.
71 – 254	Reserved.
255	Not used

Note: The following convention may be used when converting third-party GPS&GLONASS measurement files into JPS file format:

USI=70 (FCN=25) designates a GLONASS satellite whose FCN is unknown.

The [SI] message contains an array of USIs indicating the currently tracked satellites. The message body's length is equal to the number of tracked satellites.

The [SI] message serves as a “reference map” for other measurement message types since it establishes the one-to-one correspondence between the satellite identifier and the array index allocated to this satellite. Bear in mind that this correspondence holds true until the next [SI] message is issued (note that the new [SI] message may or may not differ from the previous one).

```
struct SatIndex {
+ u1 usi[nSats];      // []
+ u1 cs;              // Checksum
};
```

3.2 [NN] GLONASS Satellite System Numbers {nSats+1} Here nSats designates the number of GLONASS satellites as specified in the corresponding [SI] message (recall that GLONASS satellites have universal identifiers lying within the range [38…70]).
The [NN] message shows the orbit slot numbers of the GLONASS satellites specified in the latest [SI] message.

```
struct SatNumbers {
  u1 number[nSats];  // Glonass SV orbit slot number []
+ u1 cs;             // Checksum
};
```

3.3 [EL] Satellite Elevations {nSats+1}

This message contains elevations [degrees,i1] for all the satellites specified in the latest [SI] message.

```
struct SatElevation {
  i1 elev[nSats];    // Elevation angle -90..90 [degrees]
+ u1 cs;             // Checksum
};
```

3.4 [AZ] Satellite Azimuths {nSats+1}

This message contains azimuths [degrees*2,u1] for all the satellites specified in the latest [SI] message.

```
struct SatAzimuth {
  u1 azim[nSats];    // Azimuth angle 0..180 [degrees*2]
+ u1 cs;             // Checksum
};
```

The notation [degrees*2] means that the corresponding values must be multiplied by 2 to restore actual azimuths in degrees.

*3.5 [RC] Full C/A Pseudoranges {8*nSats+1}*

Contains ***full*** C/A pseudoranges [sec,f8] for all the satellites specified in the latest [SI] message.

```
struct PR_CA {
  f8 prange[nSats];  // Pseudorange [s]
+ u1 cs;             // Checksum
};
```

*3.6 [PC] Full C/A Carrier Phases {8*nSats+1}*

Contains the full C/A carrier phases [cycles, f8] for all the satellites specified in the latest [SI] message.

```
struct PhaseCA {
  f8 phase[nSats];  // C/A carrier phase [cycles]
+ u1 cs;            // Checksum
};
```

*3.7 [DC] C/A Doppler {4*nSats+1}*

Contains C/A doppler estimates [Hz*10^{-4}, i4] for all the satellites specified in the latest [SI] message.

```
struct DopplerCA {
  i4 doppler[nSats]; // Doppler [Hz*10^-4]
+ u1 cs;             // Checksum
};
```

3.8 [EC] C/A Carrier to Noise Ratio {nSats+1}

Contains C/A channel carrier to noise ratios for all the satellites specified in the latest [SI] message.

```
struct CarrierToNoiseRatioCA {
  u1 sn[nSats];      // C/N0 [dB*Hz]
+ u1 cs;             // Checksum
};
```

3.9 [GA] GPS Almanac {47}

```
struct GPSAlm {
+ u1 sv;          // SV PRN number within the range [1-37]
+ i2 wna;         // Almanac reference week []
+ i4 toa;         // Almanac reference time of week [s]
+ u1 healthA;     // Health summary (from almanac). [bitfield]
                  // 0..4 -code for health of SV signal components
                  // 5..7 - navigation data health indicators
+ u1 healthS;     // Satellite health (page 25 of subframe 5) []
  u1 config;      // Satellite configuration (page 25 of subframe 4)
                  // [bitfield]
```

```
                 // 0..2 - satellite configuration
                 // 3 - anti-spoofing flag
                 // 4..7 - reserved
  //===== Clock data =====
+ f4 af1;        // Polynomial coefficient [s/s]
+ f4 af0;        // Polynomial coefficient [s]
  //===== Ephemeris data =====
  //--- Keplerian orbital parameters ---
+ f4 rootA;      // Square root of the semi-major axis [m^0.5]
+ f4 ecc;        // Eccentricity []
+ f4 m0;         // Mean Anomaly at reference time [semi-circles]
+ f4 omega0;     // Longitude of ascending node of orbit plane
                 // at the start of week 'wna' [semi-circles]
+ f4 argPer;     // Argument of perigee [semi-circles]
  //--- Corrections to orbital parameters ---
+ f4 deli;       // Correction to inclination angle
                 // [semi-circles]
+ f4 omegaDot;   // Rate of right ascension [semi-circle/s]
+ u1 cs;         // Checksum
};
```

3.10 [GE] GPS Ephemeris {123}

```
struct GPSEphemeris {
+ u1 sv;         // SV PRN number within the range [1-37]
+ u4 tow;        // Time of week [s]
+ u1 flags;      // Flags (see GPS ICD for details) [bitfield]:
                 // 0 - curve fit interval
                 // 1 - data flag for L2 P-code
                 // 2..3 - code on L2 channel
                 // 4 - anti-spoof (A-S) flag (from HOW)
                 // 5 - 'Alert' flag (from HOW)
                 // 6 - '1' means that this ephemeris was
                 // retrieved from the non-volatile
                 // memory when the receiver was reset
                 // or turned on last
                 // 7 - reserved
  //===== Clock data (Subframe 1) =====
+ i2 iodc;       // Issue of data, clock []
+ i4 toc;        // Clock data reference time [s]
+ i1 ura;        // User range accuracy []
+ u1 healthS;    // Satellite health []
+ i2 wn;         // Week number []
+ f4 tgd;        // Estimated group delay differential [s]
+ f4 af2;        // Polynomial coefficient [s/(s^2)]
+ f4 af1;        // Polynomial coefficient [s/s]
+ f4 af0;        // Polynomial coefficient [s]
  //===== Ephemeris data (Subframes 2 and 3) =====
+ i4 toe;        // Ephemeris reference time [s]
+ i2 iode;       // Issue of data, ephemeris []
  //--- Keplerian orbital parameters ---
```

```
+ f8 rootA;       // Square root of the semi-major axis [m^0.5]
+ f8 ecc;         // Eccentricity []
+ f8 m0;          // Mean Anomaly at reference time (wn,toe)
                  // [semi-circles]
+ f8 omega0;      // Longitude of ascending node of orbit plane
                  // at the start of week 'wn' [semi-circles];
+ f8 inc0;        // Inclination angle at ref. time
                  // [semi-circles]
+ f8 argPer;      // Argument of perigee [semi-circles];
  //--- Corrections to orbital parameters ---
+ f4 deln;        // Mean motion difference from computed value
                  // [semi-circle/s]
+ f4 omegaDot;    // Rate of right ascension [semi-circle/s]
+ f4 incDot;      // Rate of inclination angle [semi-circle/s]
                  // Amplitude of the cosine harmonic correction
+ f4 crc;         // term to the orbit radius [m]
                  // Amplitude of the sine harmonic correction
+ f4 crs;         // term to the orbit radius [m]
                  // Amplitude of the cosine harmonic correction
+ f4 cuc;         // term to the argument of latitude [rad]
                  // Amplitude of the sine harmonic correction
+ f4 cus;         // term to the argument of latitude [rad]
                  // Amplitude of the cosine harmonic correction
+ f4 cic;         // term to the angle of inclination [rad]
                  // Amplitude of the sine harmonic correction
+ f4 cis;         // term to the angle of inclination [rad]
+ u1 cs;          // Checksum
};
```

3.11 [NA] GLONASS Almanac {46}

```
struct GLOAlmanac {
+ u1 sv;          // Satellite orbit slot number within [1..24] []
+ i1 frqNum;      // Satellite frequency channel number [-7..24] []
+ i2 dna;         // Day number within 4-year period starting
                  // with the leap year []
+ f4 tlam;        // Time of the first ascending node passage
                  // on day 'dna' [s]
+ u1 health;      // Satellite health as specified by 'Cn' [bitfield]
                  // 0- '1' indicates healthy SV, '0' – unhealthy
                  // 1..7- reserved
  //===== Clock data =====
+ f4 tauN;        // Coarse time correction to SV clock
                  // with respect to GLONASS system time [s]
+ f8 tauSys;      // Correction to GLONASS system time with respect to
                  // UTC(SU) [s]
  //===== Ephemeris data =====
+ f4 ecc;         // Eccentricity at reference time 'tlam' []
+ f4 lambda;      // Longitude of ascending node at reference time 'tlam' [semi-circles].
+ f4 argPer;      // Argument of perigee at reference time 'tlam' [semi-circles]
+ f4 delT;    // Correction to mean Draconic period at reference time 'tlam' [s/period].
```

```
+ f4 delTdt;      // Rate of change of Draconic period [s/period^2]
+ f4 deli;        // Correction to inclination at reference time 'tlam'[semi-circles].
+ u1 cs;          // Checksum
};
```

3.12 [NE] GLONASS Ephemeris {80}

```
struct GLOEphemeris {
  u1 sv;          // Satellite orbit slot number [1..24] []
+ i1 frqNum;      // Satellite frequency channel number [-7..24] []
  i2 dne;         // Day number within 4-year period []
                  // (Ephemeris reference time 'tb' is specified// for day 'dne')
+ i4 tk;          // Frame start time within current day [s]
+ i4 tb;          // Ephemeris reference time [s]
+ u1 health;      // Satellite health. [bitfield]
                  // 0 - MSB taken from Bn word which indicates
                  // satellite health:
                  // 1 - indicates 'satellite is unhealthy'
                  // 0 - indicates 'satellite is healthy'
                  // 1 - If set, this flag indicates that the
                  // time-frequency params 'tau' and 'gamma'
                  // may be wrong. (Note that GNSS receiver
                  // performs several 'internal' data
                  // consistency checks allowing detection
                  // of problem broadcast parameters).
                  // 2 - If set, this flag indicates that initial
                  // conditions 'r[3]' and 'v[3]' may be wrong.
                  // 3 - SV health (Cn word) status from almanac:
                  // 0 - indicates 'satellite is unhealthy'
                  // 1 - indicates 'satellite is healthy'
                  // 4 - If set, this flag indicates that SV health
                  // status from almanac is available
                  // 5..7 - reserved
  //===== Ephemeris data ======
+ u1 age;         // Age of operational information (En) [days]
+ u1 flags;       // Flags (for details on the flags, see GLONASS ICD)
                  // [bitfield]
                  // 0..1 - p1 word
                  // 2 - p2 word
                  // 3 - p3 word
                  // 4..5 - 2 LSB taken from Bn word
                  // 6 -'1' means this ephemeris was retrieved
                  // from non-volatile memory when you reset
                  // or turned on your receiver last
                  // 7 - reserved
+ f8 r[3];        // Satellite PE-90 coordinates [km]
+ f4 v[3];        // Satellite PE-90 velocities [km/s]
+ f4 w[3];        // Satellite PE-90 accelerations due to Luni-Solar
                  // gravitational perturbations [km/s^2]
  //===== Clock data ======
+ f8 tauSys;      // Time correction to GLONASS time scale (vs. UTC(SU))
```

```
                  // tauSys = TUTC(SU) - TGLN [s]
+ f4 tau;         // Correction to satellite clock (vs. GLONASS time)
                  // tau = TGLN - TSV [s]
+ f4 gamma;       // Rate of satellite clock offset [s/s]
+ u1 cs;          // Checksum
};
```