SIM68 Series_NMEA Message_User Guide

GNSS Module

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM68 Series_NMEA Message_User Guide
Version:	1.00
Date:	2020.7.10
Status:	Release

GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER'S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER'S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

COPYRIGHT

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/download/list-863-en.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/ask/ or email to: support@simcom.com

About Document

Version History

Version	Date	Owner	What is new
V1.00	2020.0710	Jian.ni	Original
V1.01	2020.11.23	Jian.ni	Fix some document errors

Scope

This document applies to the following products

Name	Type	Size(mm)	Comments
SIM68D			N/A
SIM68I			N/A

Contents

1 Introduction

1.1 Purpose of the document

At present, has been built and is planning the construction of a satellite navigation system apart from United States GPS system, and Russia's GLONASS system, the European Galileo system, Beidou satellite navigation system in China and Japan and Indian regional satellite navigation systems.

Based on module AT command manual, this document will introduce GNSS NEMA Message application process.

Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

1.3 Conventions and abbreviations

2 NMEA Messages

2.1 General Format of NMEA Messages

NMEA messages use the ASCII character set and have a defined format. Each message begins with a $ (hex 0x24) and end with a carriage return and line feed (hex 0x0D 0x0A, represented as <CR><LF>). Each message consists of one or more fields of ASCII letters and numbers, separated by commas. After the last field, and before the <CR><LF> is a checksum consisting of an asterisk (*, hex 0x2A) followed by two ASCII characters representing the hexadecimal value of the checksum. The checksum is computed as the exclusive OR of all characters between the $ and * characters.

Parameter	Example	Contents
Start	$GPGGA	Message Identifier. Input messages begin at MID 100
Payload	<Data>	Message specific data. Refer to a specific message section for <data>...<data> definition
Checksum	*CKSUM	CKSUM is a two-hex ASCII character. Checksums is required in all input messages
End	<CR> <LF>	Each message is terminated using Carriage Return (CR) Line Feed (LF) which are \r\n. Because \r\n are not printable ASCII characters, they are omitted from the example strings, but must be sent to terminate the message and cause the receiver to process that input message

NOTE

- All fields in all proprietary NMEA messages are required, none are optional and are comma delimited
- In some numeric fields representing a single data element, leading zeros before a decimal are suppressed. A single "0" character preceding the decimal point is maintained. In compound numeric structures (such as LAT or LONG), leading zeros are suppressed only on the leftmost element Trailing zeros are not suppressed

2.2 Standard NMEA Output Messages

Message	Description	Possible Talker Identifiers
GGA	Time, position and fix type data	GP,GN,GL,BD,GA
GSA	GNSS receiver operating mode, satellites used in the position solution, and DOP values	GP, GL,BD,GA
GSV	Number of GNSS satellites in view satellite ID numbers, elevation, azimuth, & SNR values	GP,GL,BD,GA
RMC	Time, date, position, course and speed data	GP,GN,GL,BD,GA
VTG	Course and speed information relative to the ground	GP,GN,GL,BD,GA
GLL	Latitude, longitude, UTC time of position fix and status	GP,GN,BD
ZDA	PPS timing message (synchronized to PPS)	GP,GN, BD

NOTE

- The prefix "GP" refers to the GPS global navigation system
- The prefix "GN" refers to the GNSS global navigation system（All kinds of global navigation systems
- The prefix "GL" refers to the GLONASS global navigation system
- The prefix "GA" refers to the GALILEO global navigation system
- The prefix "BD" refers to the BEIDOU global navigation system

A full description of the listed NMEA messages is provided in the following sections

2.2.1 Message ID GGA: Global Positioning System Fixed Data

Example:
$GPGGA,091926.000,3113.3166,N,12121.2682,E,1,09,0.9,36.9,M,7.9,M,,0000*56<CR><LF>

Name	Example	Unit	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	091926.000		hhmmss.sss
Latitude	3113.3166		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12121.2682		dddmm.mmmm
E/W Indicator	E		E=east or W=west

Position Fix Indicator	1		See Table 2.2.1
Satellites Used	09		Range 0 to 12
HDOP	0.9		Horizontal Dilution of Precision
MSL Altitude	36.9	meters	
Units	M	meters	
Geoid Separation	7.9	meters	Geoid-to-ellipsoid separation. Ellipsoid altitude = MSL Altitude + Geoid Separation.
Units	M	meters	
Age of Diff. Corr		sec	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*56		
<CR><LF>			End of message termination

Table 2.2.1

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3-5	Not supported
6	Dead Reckoning Mode, fix valid

NOTE

- A valid status is derived from all the parameters set in the software. This includes the minimum number of satellites required, any DOP mask setting, presence of DGPS corrections, etc. If the default or current software setting requires that a factor is met, then if that factor is not met, the solution will be marked as invalid

2.2.2 Message ID GLL: Geographic Position - Latitude/Longitude

Example:
$GPGLL,3113.3157,N,12121.2684,E,094051.000,A,A*59<CR><LF>

Name	Example	Unit	Description
Message ID	$GPGLL		GLL protocol header
Latitude	3113.3157		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12121.2684		dddmm.mmmm

E/W Indicator	E		E=east or W=west
UTC Time	094051.000		hhmmss.sss
Status	A		A=data valid or V=data not valid
Mode	A		A=Autonomous D=DGPS
Checksum	*59		
<CR><LF>			End of message termination

NOTE

- Position was calculated based on one or more of the SVs having their states derived from almanac parameters, as opposed to ephemerides

2.2.3 Message ID GSA: GNSS DOP and Active Satellites

Example:
$GPGSA,A,3,07,02,26,27,09,04,15, , , , , ,1.8,1.0,1.5*33<CR><LF>

Name	Example	Unit	Description
Message ID	$GPGSA		GGA protocol header
Mode 1	A		See Table 2.2.3
Mode 2	3		See Table 2.2.4
Satellite Used [1]	07		SV on Channel 1
Satellite Used [1]	02		SV on Channel 2
....			....
Satellite Used [1]			SV on Channel 12
PDOP[2]	1.8		Position Dilution of Precision
HDOP[2]	1.0		Horizontal Dilution of Precision
VDOP[2]	1.5	meters	Vertical Dilution of Precision
Checksum	*33		
<CR><LF>			End of message termination

NOTE

- Satellite used in solution
- Maximum DOP value reported is 50. When value 50 is reported, the actual DOP may be much larger

Table 2.2.3

Value	Description
M	Manual – Forced to operate in 2D or 3D mode
A	2D Automatic – Allowed to automatically switch 2D/3D

Table 2.2.4

Value	Description
1	Fix not available
2	2D (<4 SVs used)
3	3D (>3 SVs used)

2.2.4 Message ID GSV: GNSS Satellites in View

Example:

$GPGSV,3,1,11,26,68,023,37,15,64,251,33,05,45,058,34,29,33,253,33*75<CR><LF>

$GPGSV,3,2,11,27,32,164,30,21,25,315,29,02,24,140,31,08,19,048,29*70<CR><LF>

$GPGSV,3,3,11,09,16,180,25,18,08,284,27,10,08,085,18*4E<CR><LF>

Name	Example	Unit	Description
Message ID	$GPGSV		GSV protocol header
Number of Messages [1]	2		Total number of GSV messages to be sent in this group
Message Number[1]	1		Message number in this group of GSV messages
Satellites in View[1]	11		
Satellite ID	26		Channel 1 (Range 1 to 32)
Elevation	68	degrees	Channel 1 (Maximum 90)
Azimuth	023	degrees	Channel 1 (True, Range 0 to 359)
SNR (C/N0)	37	dBHz	Range 0 to 99, null when not tracking
....			....
Satellite ID	29		Channel 4 (Range 1 to 32)
Elevation	33	degrees	Channel 4 (Maximum 90)
Azimuth	253	degrees	Channel 4 (True, Range 0 to 359)
SNR (C/N0)	33	dBHz	Range 0 to 99, null when not tracking
Checksum	*75		
<CR><LF>			End of message termination

NOTE

- Depending on the number of satellites tracked, multiple messages of GSV data may be required In some software versions, the maximum number of satellites reported as visible is limited to 12, even though more may be visible

2.2.5 Message ID RMC: Recommended Minimum Specific GNSS Data

Example:
$GPRMC,094330.000,A,3113.3156,N,12121.2686,E,0.51,193.93,171210,,,A*68<CR><LF>

Name	Example	Unit	Description
Message ID	$GPRMC		RMC protocol header
UTC Time	094330.000		hhmmss.sss
Status [1]	A		A=data valid or V=data not valid
Latitude	3113.3156		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12121.2686		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Speed Over Ground	0.51	knots	
Course Over Ground	193.93	degrees	True
Date	171210		ddmmyy
Magnetic Variation [2]		degrees	E=east or W=west
East/West Indicator[2]			E=east
Mode	A		A=Autonomous D=DGPS
Checksum	*68		
<CR><LF>			End of message termination

NOTE

- A valid status is derived from all the parameters set in the software. This includes the minimum number of satellites required, any DOP mask setting, presence of DGPS corrections, etc. If the default or current software setting requires that a factor is met, then if that factor is not met, the solution will be marked as invalid
- Does not support magnetic declination. All “course over ground” data are geodetic WGS84

directions relative to true North

2.2.6 Message ID VTG: GNSS DOP and Active Satellites

Example:
$GPVTG,83.37,T,,M,0.00,N,0.0,K,A*32<CR><LF>

Name	Example	Unit	Description
Message ID	$GPVTG		VTG protocol header
Course	83.37	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic1 [1]
Speed	0.00	knots	Measured horizontal speed
Units	N		Knots
Speed	0.0	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
Mode	A		A=Autonomous D=DGPS
Checksum	*32		
<CR><LF>			End of message termination

NOTE

- Does not support magnetic declination. All “course over ground” data are geodetic WGS84 directions.

2.2.7 Message ID ZDA: Time & Data

Example:
$GPZDA,091926.000,17,12,2010,,*55<CR><LF>

Name	Example	Unit	Description
Message ID	$GPZDA		ZDA protocol header
UTC time	091926.000	Hhmmss.sss	The UTC time units are: hh = UTC hours from 00 to 23 mm = UTC minutes from 00 to 59 ss = UTC seconds from 00 to 59 sss= UTC micro seconds Either using valid IONO/UTC or estimated from default leap seconds
Day	17		Day of the month, range 1 to 31
Month	12		Month of the year, range 1 to 12
Year	2010		1980 to 2079
Local zone hour [1]		hour	Offset from UTC
Local zone minutes[1]		minute	Offset from UTC
Checksums	*55		
<CR><LF>			End of message termination

2.3 Proprietary NMEA Messages

2.3.1 Packet Type:001 PAIR_ACK

Acknowledge of PAIR command

DataField: PAIR_ACK			
Name	**Unit**	**Default**	**Description**
Cmd	--	--	Command_ID: The command / packet type the acknowledge responds
Response Result	--	--	0 The command was successfully sent 1 The command is processing. You must wait for the result 2 Sending the command failed 3 This command ID is not supported 4 Command parameter error. Out of range / some parameters were lost / checksum error 5 MNL service is busy. You can try again soon

Return&Example

[Return]
$PAIR001,Command_ID,Result*CS<CR><LF>
Command_ID: The command / packet type the acknowledge responds
Result: The result of the command. The value is mnl_service_result_type_t
0: The command was successfully sent
1: The command is processing. You must wait for the result
2: Sending the command failed
3: This command ID is not supported
4: Command parameter error. Out of range / some parameters were lost / checksum error
5: MNL service is busy. You can try again soon
[Example]
Send:
$PAIR666*3C\r\n
Response:
$ PAIR001,666,3*3E \r\n ==> $PAIR666 This command ID is not supported

2.3.2 Packet Type:123 PAIR_SIMCOM_VERSION

Query the release version of simcom

DataField: $PAIR123*CS<CR><LF>			
Name	Unit	Default	Description
	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. 2. $PAIR123,<Simcom Release Version>
[Example]
Send:
$PAIR123*3A\r\n
Response:
$PAIR001,123,0*3B
$PAIR123,B01V03SIM68D_11*42

2.3.3 Packet Type:011 PAIR_INDICATION_SYSTEM_MESSAGE

GNSS System message indication

DataField: $PAIR011,<Type>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Type	--	--	The system message type "1", Notification for GNSS system startup

Return&Example

[Return]
NONE
[Example]
$PAIR011,001*27

2.3.4 Packet Type:004 PAIR_GNSS_SUBSYS_HOT_START

Hot Start. Use the available data in the NVRAM

DataField: $PAIR004*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR004*3E\r\n
Response:
$PAIR001,004,0*3F\r\n ==> Success

2.3.5 Packet Type:005 PAIR_GNSS_SUBSYS_WARM_START

Warm Start. Not using Ephemeris data at the start

DataField: $PAIR005*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR005*3F\r\n
Response:
$PAIR001,005,0*3E\r\n ==> Success

2.3.6 Packet Type:006 PAIR_GNSS_SUBSYS_COLD_START

Cold Start. Not using the Position, Almanac and Ephemeris data at the start

DataField: $PAIR006*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR006*3C\r\n
Response:
$PAIR001,006,0*3D\r\n ==> Success

2.3.7 Packet Type:007 PAIR_GNSS_SUBSYS_FULL_COLD_START

Full Cold Start

In addition to Cold start, this command clears the system/user configurations at the start
It resets the GNSS module to the factory default

DataField: $PAIR007*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR007*3D\r\n
Response:
$PAIR001,007,0*3C\r\n ==> Success

2.3.8 Packet Type:472 PAIR_EPO_ERASE_FLASH_DATA

Erase the EPO data stored in the flash memory

DataField: $PAIR472*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR472*3B\r\n
Response:
$PAIR001,472,0*3A\r\n ==> Success

2.3.9 Packet Type:900 PAIR_LOCUS_ENABLE

Enable or disable LOCUS save data

DataField: $PAIR900,<Enable>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Enable	--	--	Enable: Enable or disable '0': Disable '1': Enable

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR900,1*2E\r\n ==> Enable LOCUS
Response:
$PAIR001,900,0*32\r\n ==> Enable Success

2.3.10 Packet Type:901 PAIR_LOCUS_GET_STATUS

Get LOCUS status

DataField: $PAIR901*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Enable	--	--	Enable: Enable or disable '0': Disable '1': Enable

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR901,<Enable>*CS<CR><LF>
Enable: Enable or disable
'0': Disable
'1': Enable
[Example]
Send:
$PAIR901*32\r\n
Response:

$PAIR001,901,0*33\r\n
$PAIR901,0*2E\r\n ==> LOCUS is disable

2.3.11 Packet Type:902 PAIR_LOCUS_SET_MODE

Set LOCUS saving mode

DataField: $PAIR902,<Mode>,<Check_3D_Fix>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Mode	--	--	Mode: Saving Mode: Normal, (1 << 0). Record per fix Out of time, (1 << 1). Record every N s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Out of speed, (1 << 2). Record after speed more than N m/s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Out of distance, (1 << 3). Record after distance more than N m. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Before entry sleep, (1 << 4). Record before entry sleep User control, (1 << 5). Record after user send PAIR_LOCUS_LOG_NOW
Check_3D_Fix	--	--	Need check 3D fix or not: 0: not check 1: need check. If set this type as 1, system will not save the location without 3D fixed

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR902,6,1*36\r\n ==> Set mode as out of time & out of speed mode. Need check 3D fix.
Response:
$PAIR001,902,0*30\r\n ==> Set success

NOTE

Must disable LOCUS saving before send this command

2.3.12 Packet Type:903 PAIR_LOCUS_GET_MODE

Get LOCUS saving mode

DataField: $PAIR903*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Mode	--	--	Mode: Saving Mode: Normal, (1 << 0). Record per fix Out of time, (1 << 1). Record every N s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Out of speed, (1 << 2). Record after speed more than N m/s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Out of distance, (1 << 3). Record after distance more than N m. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD) Before entry sleep, (1 << 4). Record before entry sleep User control, (1 << 5). Record after user send PAIR_LOCUS_LOG_NOW
Check_3D_Fix	--	--	Need check 3D fix or not: 0: not check 1: need check. If set this type as 1, system will not save the location without 3D fixed

Return&Example

[Return]

1. PAIR_ACK for send result.

2. $PAIR903,<Mode>,<Check_3D_Fix>*CS<CR><LF>

Mode: Saving Mode

Normal, (1 << 0). Record per fix.

Out of time, (1 << 1). Record every N s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD).

Out of speed, (1 << 2). Record after speed more than N m/s. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD).

Out of distance, (1 << 3). Record after distance more than N m. N is customer configuration (PAIR_LOCUS_SET_THRESHOLD).

Before entry sleep, (1 << 4). Record before going to sleep.

User control, (1 << 5). Record after user send PAIR_LOCUS_LOG_NOW.

Check_3D_Fix: Need check 3D fix or not.

0: not check.

1: need check. If set this type as 1, system will not save the location without 3D fixed.

[Example]

Send:

$PAIR903*30\r\n

Response:

$PAIR001,903,0*31\r\n

$PAIR903,6,1*37\r\n ==> LOCUS saving mode is out of time & out of speed mode. Need check 3D fix

NOTE

Must disable LOCUS saving before send this command

2.3.13 Packet Type:904 PAIR_LOCUS_SET_THRESHOLD

Set LOCUS mode threshold

DataField: $PAIR904,<Mode>,<Threshold>*CS<CR><LF>

Name	Unit	Default	Description
Mode	--	--	Saving Mode: 0: Out of time mode 1: Out of speed mode 2: Out of distance mode
Threshold	--	--	The threshold of saving mode: If mode == 0, out of time mode, the time threshold is 1s ~ 12hours. Unit is second. Default is 15s If mode == 1, out of speed mode, the speed threshold is 1m/s ~ 100m/s. Unit is meter/secode. Default is 1m/s If mode == 2, out of distance mode, the distance threshold is 1m ~ 50000m. Unit is meter. Default is 1m

Return&Example

[Return]

1. PAIR_ACK for send result.

[Example]

Send:

$PAIR904,1,5*33\r\n ==> Set out of time mode threshold is 5s.

Response:

$PAIR001,904,0*36\r\n ==> Set success. LOCUS will save record every 5s.

NOTE

Must disable LOCUS saving before send this command
If the threshold out of rang, will response parameter error ("$PAIR001,804,4*33\r\n")

2.3.14 Packet Type:905 PAIR_LOCUS_GET_THRESHOLD

Get LOCUS mode threshold

DataField: $PAIR905,<Mode>*CS<CR><LF>

Name	Unit	Default	Description
Mode	--	--	Saving Mode: 0: Out of time mode 1: Out of speed mode 2: Out of distance mode
Threshold	--	--	The threshold of saving mode: If mode == 0, out of time mode, the time threshold is 1s ~ 12hours. Unit is second. Default is 15s If mode == 1, out of speed mode, the speed threshold is 1m/s ~ 100m/s. Unit is meter/secode. Default is 1m/s If mode == 2, out of distance mode, the distance threshold is 1m ~ 50000m. Unit is meter. Default is 1m

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR905,<Threshold>*CS<CR><LF>
Threshold: The threshold of saving mode
If mode == 0, out of time mode, the time threshold is 1s ~ 12hours. Unit is second. Default is 15s
If mode == 1, out of speed mode, the speed threshold is 1m/s ~ 100m/s. Unit is meter/secode. Default is 1m/s
If mode == 2, out of distance mode, the distance threshold is 1m ~ 50000m. Unit is meter. Default is 1m
[Example]
Send:

$PAIR905,0*2A\r\n ==> Get time threshold
Response:
$PAIR001,905,0*37\r\n
$PAIR905,15*1E\r\n ==> Time threshold is 15s

NOTE

Must disable LOCUS saving before send this command

2.3.15 Packet Type:906 PAIR_LOCUS_CLEAR

Clear LOCUS Data

DataField: $PAIR906,<Type>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Type	--	--	Clear Type: 0: Clear record data and restore to default setting (configuration in gnss_config.bin) 1: Clear record data only 2: Clear user setting. Restore to default setting

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR906,0*29\r\n
Response:
$PAIR001,906,0*34\r\n

NOTE

Must disable LOCUS saving before send this command

2.3.16 Packet Type:907 PAIR_LOCUS_LOG_NOW

Save current location data

DataField: $PAIR907*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Type	--	--	1 Snapshot data logging

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR907*34\r\n
Response:
$PAIR001,907,0*35\r\n

NOTE

Must keep user control (1 << 5) in saving mode if need use this command

2.3.17 Packet Type:908 PAIR_LOCUS_GET_DATA

Get all record data

DataField: $PAIR908,<Type>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Type	--	--	Response type: Response as NMEA Response as PAIR command

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR908,0*CS<CR><LF>

LOCUS read begin

3. $PAIR908,1,<Record_Num>,<Record_Size>*CS<CR><LF>

LOCUS read information

Record_Num: the total record numbers

Record_Size: the size of data per record

4. LOGGA + LORMC

If type is 0, system will response LOGGA + GPGGA. The format is same as GPGGA + GPRMC.

5. $PAIR908,2,<UTC>,<Fix_Type>,<Lat>,<Lon>,<Heighing>,<Speed>,<Heading>,<HDOP>,<SatNo>*CS<CR><LF>

If type is 1, system will response PAIR908,2,xxxx list for every record

None saved data will show 0.

6. $PAIR908,3*CS<CR><LF>

LOCUS read end

[Example]

Send:

$PAIR908,0*27\r\n

Response:

$PAIR001,908,0*3A\r\n

$PAIR908,0*27\r\n

$PAIR908,1,2,16*13\r\n

$LOGGA,080931.000,011772.4267,N,0016183.7702,E,1,0,0.0,0.53,M,,M,,*59\r\n

$LORMC,080931.000,A,011772.4267,N,0016183.7702,E,260320,,,A,V*C\r\n

$LOGGA,080932.000,011772.4267,N,0016183.7702,E,1,0,0.0,0.53,M,,M,,*5A\r\n

$LORMC,080932.000,A,011772.4267,N,0016183.7702,E,260320,,,A,V*F\r\n

$PAIR908,3*24\r\n

Send:

$PAIR908,1*26\r\n

Response:

$PAIR001,908,0*3A\r\n

$PAIR908,0*27\r\n

$PAIR828,2,5EA541BB,01,12341A1C,3E06BA8C,0210,0000,0000,0000,00*07r\n

$PAIR828,2,5EA541BC,01,12341A1B,3E06BA8A,0210,0000,0000,0000,00*05r\n

$PAIR908,1,2,16*13\r\n

$PAIR908,3*24\r\n

NOTE

Must disable LOCUS saving before send this command

2.3.18 Packet Type:909 PAIR_LOCUS_GET_RECORD_NUM

Get total record number

DataField: $PAIR909*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Time	msec	--	Position fix interval in milliseconds (ms)

Return&Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR909,<Record_Num>*CS<CR><LF>
Record_Num: total record number
[Example]
Send:
$PAIR909*3A\r\n
Response:
$PAIR001,909,0*3B\r\n
$PAIR909,15*12\r\n ==> LOCUS has save 15 records

2.3.19 Packet Type:050 PAIR_COMMON_SET_FIX_RATE

Set Position Fix Interval
(ULP mode only support 1Hz)

DataField: $PAIR050,time*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
time	msec	--	Position fix interval in milliseconds (ms). [Range: 1000 ~ 10000] For time > 1000ms and time <= 10000ms, position fix interval will be rounded to integral sec (1000ms, 2000ms, ... , 10000ms)

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]

Send:
$PAIR050,3000*10\r\n
Response:
$PAIR001,050,0*3E\r\n ==> Success

2.3.20 Packet Type:035 PAIR_COMMON_GET_FIX_STATUS

Get fix type and fix mode

DataField: $PAIR035*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR035,<FIX_TYPE>,<FIX_MODE>*CS<CR><LF>
FIX_TYPE:
0: NONE
1: SINGLE
2: DGPS
3: Not support
4: RTK FIX
5: RTK FLOAT
6: Estimated
FIX_MODE:
0: NONE
1: 2D fix
2: 3D fix
[Example]
Send:
$PAIR035*3C\r\n
Response:
$PAIR001,035,0*3D\r\n ==> Success
$PAIR035,2,2*3C\r\n ==> position 3D fix with Differential GPS

2.3.21 Packet Type:690 PAIR_PERIODIC_SET_MODE

This command is used to set Periodic Power Saving Mode Settings
There are two stages in periodic power saving mode (Run stage and Sleep stage), and it will change periodically according to the setting
Run stage: the GNSS module measures and calculates the position
Sleep stage: the GNSS module may enter power saving modes

DataField: $PAIR690,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Mode	--	--	0: Disable periodic mode 1: Smart periodic mode. In this mode, GNSS system dynamically increases run time in order to collect more navigation data 2: Strict periodic mode. In this mode, GNSS system periodically forces entry into low-power mode If <Mode> is 1 or 2, it needs the following parameter for low-power periodic mode
FirstRun	--	--	Interval in seconds to exit the minimum power sleep mode and get a new position fix. [Range: 3~518400 s]
FirstSleep	--	--	Duration in seconds to get a fix (or attempt to get a fix) before switching from running mode back to a minimum power sleep mode. [Range: 3~518400 s]
SecondRun	--	--	GNSS system will use "second run time" instead of "run time" setting when there is no signal. [Range: 0 or 3~518400 s] The second run time duration can be "0" only when the second sleep time is "0"
SecondSleep	--	--	GNSS system will use "second sleep time" instead of "sleep time" setting when there is no signal. [Range: 0 or 3~518400 s] The second sleep time duration can be "0" only when the second run time is "0"

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR690,1,21,39,48,72*28\r\n
Response:
$PAIR001,690,0*34\r\n ==> Success
Send:
$PAIR690,0*29\r\n ==> Normal mode
Response:

$PAIR001,690,0*34\r\n ==> Success

2.3.22 Packet Type:860 PAIR_IO_OPEN_PORT

Open a GNSS data port

DataField:
$PAIR860,<Port_Type>,<Port_Index>,<Data_Type>,<Baudrate>,<Flow_control>*CS<CR><LF>

Name	Unit	Default	Description
Port_Type	--	--	HW Port Type: 0: UART [ER1 support] 1: I2C [ER2 support] 2: SPI [ER2 support] 3: USB [ER1 support] 4: SD-Card [ER3 support]
Port_Index	--	--	HW Port Index: UART - 0: UART0, 1: UART1, 2: UART2 USB - 0: USB Virtual Port 0, 1: USB Virtual Port 1 Others - 0: Only one port
Data_Type	--	--	A bitmap to config data type: GNSS_IO_FLAG_OUT_NMEA (0x01) GNSS_IO_FLAG_OUT_LOG (0x02) GNSS_IO_FLAG_OUT_CMD_RSP (0x04) GNSS_IO_FLAG_OUT_DATA_RSP (0x08) GNSS_IO_FLAG_OUT_RTCM (0x10) GNSS_IO_FLAG_IN_CMD (0x20) GNSS_IO_FLAG_IN_DATA (0x40) GNSS_IO_FLAG_IN_RTCM (0x80)
Baudrate	--	--	the baud rate must be configured. This parameter is only valid for UART. Please use 0 for other port type: Support 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 3000000
Flow_control	--	--	0, disable flow control. 1, enable SW flow control. 2, enable HW flow control. This parameter is only valid for UART. Please use 0 for other port type

Return&Example

[Return]
1. PAIR_ACK for send result.

[Example]
Send:
$PAIR860,0,2,37,115200,0*29\r\n ==> Open UART2 to NMEA output without flow control. Baudrate is 115200.
Response:
$PAIR001,860,0*35\r\n ==> Success

2.3.23 Packet Type:862 PAIR_IO_SET_DATA_TYPE

Set GNSS port data type configuration

DataField: $PAIR862,<Port_Type>,<Port_Index>,<Data_Type>*CS<CR><LF>

Name	Unit	Default	Description
Port_Type	--	--	HW Port Type: 0: UART [ER1 support] 1: I2C [ER2 support] 2: SPI [ER2 support] 3: USB [ER1 support] 4: SD-Card [ER3 support]
Port_Index	--	--	HW Port Index: UART - 0: UART0, 1: UART1, 2: UART2 USB - 0: USB Virtual Port 0, 1: USB Virtual Port 1 Others - 0: Only one port
Data_Type	--	--	A bitmap to config data type: GNSS_IO_FLAG_OUT_NMEA (0x01) GNSS_IO_FLAG_OUT_LOG (0x02) GNSS_IO_FLAG_OUT_CMD_RSP (0x04) GNSS_IO_FLAG_OUT_DATA_RSP (0x08) GNSS_IO_FLAG_OUT_RTCM (0x10) GNSS_IO_FLAG_IN_CMD (0x20) GNSS_IO_FLAG_IN_DATA (0x40) GNSS_IO_FLAG_IN_RTCM (0x80).

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR862,3,1,37*1C\r\n ==> Config USB virtual port 1 to NMEA & PAIR port. (Without debug log.)

Response:

$PAIR001,862,0*37\r\n ==> Success

NOTE

GNSS_IO_FLAG_IN_RTCM cannot be set with a different type in the same port

2.3.24 Packet Type:863 PAIR_IO_GET_DATA_TYPE

Get GNSS port data type configuration

DataField: $PAIR863,<Port_Type>,<Port_Index>*CS<CR><LF>

Name	Unit	Default	Description
Port_Type	--	--	HW Port Type: 0: UART [ER1 support] 1: I2C [ER2 support] 2: SPI [ER2 support] 3: USB [ER1 support] 4: SD-Card [ER3 support]
Port_Index	--	--	HW Port Index: UART - 0: UART0, 1: UART1, 2: UART2 USB - 0: USB Virtual Port 0, 1: USB Virtual Port 1 Others - 0: Only one port

Return&Example

[Return]

1. PAIR_ACK for send result

2. $PAIR863,<Data_Type>*CS<CR><LF>

Data_Type: A bitmap to config data type

GNSS_IO_FLAG_OUT_NMEA (0x01)

GNSS_IO_FLAG_OUT_LOG (0x02)

GNSS_IO_FLAG_OUT_CMD_RSP (0x04)

GNSS_IO_FLAG_OUT_DATA_RSP (0x08)

GNSS_IO_FLAG_OUT_RTCM (0x10)

GNSS_IO_FLAG_IN_CMD (0x20)

GNSS_IO_FLAG_IN_DATA (0x40)

GNSS_IO_FLAG_IN_RTCM (0x80)

[Example]

Send:
$PAIR863,3,1*35\r\n
Response:
$PAIR001,863,0*36\r\n ==> Success
$PAIR863,0*2B\r\n

2.3.25 Packet Type:864 PAIR_IO_SET_BAUDRATE

Set port baud rate configuration

DataField: $PAIR864,<Port_Type>,<Port_Index>,<Baudrate>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Port_Type	--	--	HW Port Type: 0: UART [ER1 support]
Port_Index	--	--	HW Port Index: 0: UART0 1: UART1 2: UART2
Baudrate	--	--	the baud rate need config: Support 115200, 230400, 460800, 921600, 3000000

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR864,0,0,115200*1B\r\n
Response:
$PAIR001,864,0*31\r\n ==> Success

NOTE

Must reboot the device after changing the port baud rate. The change will valid after reboot

2.3.26 Packet Type:865 PAIR_IO_GET_BAUDRATE

Get port baud rate configuration

DataField: $PAIR865,<Port_Type>,<Port_Index>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Port_Type	--	--	HW Port Type: 0: UART [ER1 support]
Port_Index	--	--	HW Port Index: 0: UART0 1: UART1 2: UART2

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR865,<Baudrate>*CS<CR><LF>
Baudrate: the baud rate need config
Support 115200, 230400, 460800, 921600, 3000000
[Example]
Send:
$PAIR865,0,0*31\r\n
Response:
$PAIR001,865,0*30\r\n ==> Success
$PAIR865,115200*1A\r\n ==> Get UART0 baud rate is 115200

NOTE

Must reboot the device after changing the port baud rate

2.3.27 Packet Type:100 PAIR_COMMON_SET_NMEA_OUTPUT_MODE

This command is to set NMEA output mode

DataField: $PAIR100,<NMEA_MODE>,<PROPRIETARY_MODE>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
NMEA_MODE	--	--	0: Disable NMEA

			1: ASCII NMEA v4.1(Default) 2: ASCII NMEA v3.0
PROPRIETARY_MODE	--	--	0: Disable extra proprietary sentence (Default) 1: Enable proprietary sentence

Return&Example

[Return]
1. PAIR_ACK for send result

[Example]
Send:
$PAIR100,1,0*3A\r\n ==> ASCII NMEA v4.1, Disable extra proprietary sentence
Response:
$PAIR001,100,0*3A\r\n ==> Success
Send:
$PAIR100,0,1*3A\r\n ==> No ASCII NMEA output, Enable proprietary sentence
Response:
$PAIR001,100,0*3A\r\n ==> Success

2.3.28 Packet Type:750 PAIR_PPS_SET_CONFIG

Set the configuration of the local time in milliseconds and phase where the PPS should be placed

DataField: $PAIR750,<PPS_by_user>,<Local_ms>,<Phase>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
PPS_by_user	--	--	"1", PPS output by user "0", PPS automatic output
Local_ms	--	--	Local receiver time tick. Range is from 0 to 4294967295 (232-1). If PSS is enabled, this parameter aligns to TOW
Phase	--	--	Time tick phase range is from 0 to 262143. If PSS is enabled, this parameter aligns to TOW

Return&Example

[Return]
1. PAIR_ACK for send result

[Example]
Send:
$PAIR750,1,1345,555*13\r\n
Response:
$PAIR001,750,0*39\r\n ==> Success

2.3.29 Packet Type:752 PAIR_PPS_SET_CONFIG_CMD

Configure the PPS settings

DataField: $PAIR752,<PPSType>,<PPSPulseWidth>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
PPSType	--	--	Availability "0", Disable "1", After the first fix "2", 3D fix only "3", 2D/3D fix only "4", Always
PPSPulseWidth	--	--	PPS Pulse Width (unit in ms). [Range: 1 ~ 999].

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR752,2,100*39\r\n
Response:
$PAIR001,752,0*3B\r\n ==> Success

2.3.30 Packet Type:753 PAIR_PPS_SET_TIMING_PRODUCT

Enable or disable timing product mode (Default off)
The timing product mode will enhance the PPS output timing accuracy

DataField: $PAIR753,<Enabled>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Enabled	--	--	Enable or disable: 0 Disable 1 Enable

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR753,1*26\r\n
Response:
$PAIR001,753,0*3A\r\n ==> Success

NOTE

Please measure the accuracy after the device collects all of the satellite almanac data

2.3.31 Packet Type:650 PAIR_LOW_POWER_ENTRY_RTC_MODE

Shutdown all systems, including GNSS and other CM4 modules
CM4 will go into RTC-Mode after sending this command and cannot receive any commands. CM4 can be awoken by the timer or the RTC_EINT pin. All system resource will re-initialize after wake up

DataField: $PAIR650,<Second>*CS<CR><LF>

Name	Unit	Default	Description
Second	--	--	the timer to leave RTC-Mode [Valid range: 0 and 10 ~ 62208000 (2 years)] '0' enter RTC-Mode without any timer

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR650,1*24\r\n
Response:
$PAIR001,650,4*3C\r\n ==> Parameter error
Send:
$PAIR650,10*14\r\n
Response:
Enter RTC-Mode without any response and wake up after 10 seconds

2.3.32 Packet Type:098 PAIR_COMMON_SET_NMEA_POS_DECIMAL_PRECISION

This command is for setting the digits shown in the NMEA position

DataField: $PAIR098,<MODE>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
MODE	--	--	0: Latitude, Longitude in 4 digits, Altitude in 1 digit 1: Latitude, Longitude in 5 digits, Altitude in 2 digit 2: Latitude, Longitude in 6 digits, Altitude in 3 digit 3: Latitude, Longitude in 7 digits, Altitude in 3 digit

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR098,0*27\r\n
==> Set the Lat/Lon digit 4 digit, and Alt in 1 digit (GGA/GLL/RMC)
Response:
$PAIR001,098,0*3A\r\n ==> Success

2.3.33 Packet Type:074 PAIR_COMMON_SET_AIC_ENABLE

Enable or disable active interference cancellation function

DataField: $PAIR074,<Enabled>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Enabled	--	--	Enable or disable: '0' = Disable '1' = Enable

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]

Send:
$PAIR074,1*24\r\n
Response:
$PAIR001,074,0*38\r\n ==> Success

2.3.34 Packet Type:086 PAIR_COMMON_SET_DEBUGLOG_OUTPUT

This command is to set enable/disable debug log output in binary format

DataField: $PAIR086,<Status>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Status	--	--	0: Disable 1: Enable

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR086,1*29\r\n
Response:
$PAIR001,086,0*35\r\n ==> Success

2.3.35 Packet Type:058 PAIR_COMMON_SET_MIN_SNR

Set the minimum SNR of used satellites

DataField: $PAIR058,<MIN_SNR>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
MIN_SNR	--	--	Minimum SNR threshold of used satellites. (Valid range: 9~37, default value: 9)

Return&Example

[Return]
1. PAIR_ACK for send result.

[Example]
Send:
$PAIR058,15*1F\r\n
==> Set the minimum SNR threshold to 15, the chip would not use the satellite which SNR is smaller than 15.
Response:
$PAIR001,058,0*36\r\n ==> Success

2.3.36 Packet Type:059 PAIR_COMMON_GET_MIN_SNR

Query the minimum SNR of used satellites

DataField: $PAIR059*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR059,<MIN_SNR>*CS<CR><LF>
MIN_SNR: Minimum SNR threshold of used satellites. (Valid range: 9~37, default value: 9)
[Example]
Send:
$PAIR059*36\r\n
Response:
$PAIR001,059,0*37\r\n ==> Success
$PAIR059,15*1E\r\n

2.3.37 Packet Type:060 PAIR_COMMON_SET_ESTIMATED_NUM

Set the number of estimated fixes when entering the tunnel

DataField: $PAIR060,<DR_LIMIT>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
DR_LIMIT	--	--	Number of estimated fix. (Valid range: 0~500, default value: 0)

Return&Example

```
[Return]
    1. PAIR_ACK for send result.
[Example]
Send:
    $PAIR060,0*20\r\n ==> Disable the estimated fix when entering the tunnel
Response:
    $PAIR001,060,0*3D\r\n ==> Success
Send:
    $PAIR060,3*23\r\n ==> Keep outputting 3 fix when entering the tunnel
Response:
    $PAIR001,060,0*3D\r\n ==> Success
```

2.3.38 Packet Type:061 PAIR_COMMON_GET_ESTIMATED_NUM

Query the number of estimated fixes when entering the tunnel

DataField: $PAIR061*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

```
[Return]
    1. PAIR_ACK for send result
    2. $PAIR061,<DR_LIMIT>*CS<CR><LF>
    DR_LIMIT: Number of estimated fix. (Valid range: 0~500, default value: 0)
[Example]
Send:
    $PAIR061*3D\r\n
Response:
    $PAIR001,061,0*3C\r\n ==> Success
    $PAIR061,0*21\r\n ==> The user disabled the DR estimated fix
```

2.3.39 Packet Type:062 PAIR_COMMON_SET_NMEA_OUTPUT_RATE

Set the NMEA sentence output interval of corresponding NMEA type

DataField: $PAIR062,<Type>,<Output_Rate>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Type	--	--	NMEA Type: -1 Reset all sentence to default value 0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data 1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude 2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites 3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View 4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence 5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed 6 NMEA_SEN_ZDA, // ZDA interval - Time & Date
Output_Rate	--	--	Output interval setting: 0 - Disabled or not supported sentence 1 - Output once every one position fix 2 - Output once every two position fixes 3 - Output once every three position fixes 4 - Output once every four position fixes 5 - Output once every five position fixes

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR062,0,3*SS\r\n
Response:
$PAIR001,062,0*3F\r\n ==> Success

2.3.40 Packet Type:063 PAIR_COMMON_GET_NMEA_OUTPUT_RATE

Get the NMEA sentence output interval of corresponding NMEA type

DataField: $PAIR063,<Type>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**

Type	--	--	NMEA Type: -1 return all sentence configuration 0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data 1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude 2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites 3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View 4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence 5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed 6 NMEA_SEN_ZDA, // ZDA interval - Time & DatePosition - Latitude longitude 2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites 3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View 4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence 5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed 6 NMEA_SEN_ZDA, // ZDA interval - Time & Date

Return&Example

[Return]

1. PAIR_ACK for send result

2. $PAIR063,<Type>,<Output_Rate>*CS<CR><LF>

Type: NMEA Type

0 NMEA_SEN_GGA, // GGA interval - GPS Fix Data

1 NMEA_SEN_GLL, // GLL interval - Geographic Position - Latitude longitude

2 NMEA_SEN_GSA, // GSA interval - GNSS DOPS and Active Satellites

3 NMEA_SEN_GSV, // GSV interval - GNSS Satellites in View

4 NMEA_SEN_RMC, // RMC interval - Recommended Minimum Specific GNSS Sentence

5 NMEA_SEN_VTG, // VTG interval - Course Over Ground and Ground Speed

6 NMEA_SEN_ZDA, // ZDA interval - Time & Date

Output_Rate: Output interval setting

0 - Disabled or not supported sentence

1 - Output once every one position fix

2 - Output once every two position fixes

3 - Output once every three position fixes

4 - Output once every four position fixes

5 - Output once every five position fixes
[Example]
Send:
$PAIR063,0*23\r\n
Response:
$PAIR001,063,0*3E\r\n ==> Success
$PAIR063,0,3*3C\r\n

2.3.41 Packet Type:066 PAIR_COMMON_SET_GNSS_SEARCH_MODE

Configure the receiver to start searching for satellites. The setting is available when the NVRAM data is valid

DataField: $PAIR066,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BeiDou_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*CS<CR><LF>

Name	Unit	Default	Description
GPS_Enabled	--	--	"0", disable (DO NOT search GPS satellites). "1", search GPS satellites
GLONASS_Enabled	--	--	"0", disable (DO NOT search GLONASS satellites). "1", search GLONASS satellites.
Galileo_Enabled	--	--	"0", disable (DO NOT search Galileo satellites). "1", search Galileo satellites
BeiDou_Enabled	--	--	"0", disable (DO NOT search BeiDou satellites). "1", search BeiDou satellites
QZSS_Enabled	--	--	"0", disable (DO NOT search QZSS satellites). "1", search QZSS satellites
NavIC_Enabled	--	--	"0", disable (DO NOT search NavIC satellites). "1", search NavIC satellites

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR066,1,0,0,0,0,0*3B\r\n ==> Search GPS satellites only
Response:
$PAIR001,066,0*3B\r\n ==> Success
Send:
$PAIR066,1,1,1,1,1,0*3B\r\n ==> Search GPS, GLONASS, Galileo, BeiDou, QZSS satellites
Response:

$PAIR001,066,0*3B\r\n ==> Success
Send:
$PAIR066,1,1,0,0,0,0*3A\r\n ==> Search GPS and GLONASS satellites
Response:
$PAIR001,066,0*3B\r\n ==> Success

NOTE

For sim68D:
L1+L5 dual frequency, does not support star cutting
L1 single frequency, supports 3 modes, as follows,
PAIR066,1,1,1,1,0,0 GPS+GLONASS+GALILEO+BEIDOU
PAIR066,1,0,0,0,0,0 GPS only
PAIR066,1,1,0,0,0,0 GPS+GLONASS
For SIM68I:
Support 2 modes, as follows,
PAIR066,1,1,1,1,0,1 G+G+G+B+NAVIC
PAIR066,0,0,0,0,0,1 NAVIC only

2.3.42 Packet Type:067 PAIR_COMMON_GET_GNSS_SEARCH_MODE

This command is to get GPS, GLONASS, Galileo, BeiDou, QZSS and NavIC search settings

DataField: $PAIR067*CS<CR><LF>

Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR067,<GPS_Enabled>,<GLONASS_Enabled>,<Galileo_Enabled>,<BeiDou_Enabled>,<QZSS_Enabled>,<NavIC_Enabled>*CS<CR><LF>
GPS_Enabled:
"0", disable (DO NOT search GPS satellites)
"1", search GPS satellites.
GLONASS_Enabled:
"0", disable (DO NOT search GLONASS satellites)
"1", search GLONASS satellites.
Galileo_Enabled:

"0", disable (DO NOT search Galileo satellites)
"1", search Galileo satellites.
BeiDou_Enabled:
"0", disable (DO NOT search BeiDou satellites)
"1", search BeiDou satellites.
QZSS_Enabled:
"0", disable (DO NOT search QZSS satellites)
"1", search QZSS satellites.
NavIC_Enabled:
"0", disable (DO NOT search NavIC satellites)
"1", search NavIC satellites
[Example]
Send:
$PAIR067*3B\r\n
Response:
$PAIR001,067,0*3A\r\n ==> Success
$PAIR067,1,0,0,0,0,0*3A\r\n ==> Search GPS satellites only

2.3.43 Packet Type:068 PAIR_COMMON_SET_HDOP_THRESHOLD

This command is for setting the HDOP threshold
If the HDOP value is larger than this threshold value, the position will not be fixed

DataField: $PAIR068,<HDOPThreshold>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
HDOPThreshold	--	--	"0": Disable this function Other value: Enable setting the HDOP threshold [Range:]

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR068,0.8*3E\r\n
Response:
$PAIR001,068,0*35\r\n ==> Success

2.3.44 Packet Type:069 PAIR_COMMON_GET_HDOP_THRESHOLD

This command is to get the HDOP threshold

DataField: $PAIR069*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
HDOPThreshold	--	--	0 Disable this function Other value Enable

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR069,<HDOPThreshold>*CS<CR><LF>
HDOPThreshold:
"0": Disable this function
Other value: Enable setting the HDOP threshold [Range:]
[Example]
Send:
$PAIR069*35\r\n
Response:
$PAIR001,069,0*34\r\n ==> Success
$PAIR069,0.8*3F\r\n

2.3.45 Packet Type:070 PAIR_COMMON_SET_STATIC_THRESHOLD

Set the speed threshold for static navigation
If the actual speed is less than the threshold, the output position remains the same and the output speed will be zero
If the threshold value is set to 0, this function is disabled

DataField: $PAIR070,<Speed_threshold>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Speed_threshold	dm/s	--	0~20 dm/s. Default value is 0 dm/s The minimum is 1 dm/s, the maximum is 20 dm/s 1 dm/s = 0.1m/s

Return&Example

[Return]

1. PAIR_ACK for send result

[Example]

Send:

$PAIR070,4*25\r\n

Response:

$PAIR001,070,0*3C\r\n ==> Success

2.3.46 Packet Type:511 PAIR_NVRAM_SAVE_NAVIGATION_DATA

Save current navigation data from RTC RAM to flash

DataField: $PAIR511*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]

1. PAIR_ACK for send result

[Example]

Send:

$PAIR511*3F\r\n

Response:

$PAIR001,511,0*3E\r\n

2.3.47 Packet Type:072 PAIR_COMMON_SET_ELEV_MASK

Set satellite elevation mask

Satellites below the elevation mask are not used

DataField: $PAIR072,<Degree>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Degree	--	--	Satellite elevation-mask. (Valid range: -90 ~ 90, default value: 5)

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR072,5*26\r\n
Response:
$PAIR001,072,0*3E\r\n ==> Success

2.3.48 Packet Type:073 PAIR_COMMON_GET_ELEV_MASK

Get satellite elevation mask

DataField: $PAIR073*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR073,<Degree>*CS<CR><LF>
Degree: Satellite elevation-mask. (Valid range: -90 ~ 90, default value: 5)
[Example]
Send:
$PAIR073*3E\r\n
Response:
$PAIR001,073,0*3F\r\n ==> Success
$PAIR073,5*27\r\n

2.3.49 Packet Type:064 PAIR_COMMON_SET_HACC_LIMIT

Set horizontal accuracy mask. Range from 30m to 200m or -1. GPS only gets the fix when hacc value < mask

DataField: $PAIR064,<HaccMask>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
HaccMask	--	--	30~200: enable hacc mask feature. (Units: meter) -1 [Default Value]: disable hacc mask feature

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR064,50*11\r\n
Response:
$PAIR001,064,0*39\r\n ==> Success

NOTE

If horizontal accuracy > HaccMask is in use. The GNSS system will not output NMEA sentences

2.3.50 Packet Type:065 PAIR_COMMON_GET_HACC_LIMIT

Query horizontal accuracy mask

DataField: $PAIR065*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
HaccMask	--	--	Query horizontal accuracy mask

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR065,<HaccMask>*CS<CR><LF>
HaccMask:
30~200: enable hacc mask feature. (Units: meter)
-1 [Default Value]: disable hacc mask feature
[Example]
Send:
$PAIR065*39\r\n
Response:
$PAIR001,065,0*38\r\n ==> Success
$PAIR065,50*10\r\n

2.3.51 Packet Type:076 PAIR_COMMON_SET_DATUM

Set default datum

DataField: $PAIR076,<Datum>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Datum	--	--	0: WGS84 1: TOKYO-M 2: TOKYO-A ...

Return&Example

[Return]
1. PAIR_ACK for send result.
[Example]
Send:
$PAIR076,0*27\r\n
Response:
$PAIR001,076,0*3A\r\n ==> Success

NOTE

The total datums list in the AppendixC Datum List

2.3.52 Packet Type:077 PAIR_COMMON_GET_DATUM

Get default datum

DataField: $PAIR077*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Datum	--	--	--

Return&Example

[Return]

1. PAIR_ACK for send result

2. $PAIR077,<Datum>*CS<CR><LF>

Datum:

0: WGS84

1: TOKYO-M

2: TOKYO-A

The total datums list in the AppendixC Datum List

[Example]

Send:

$PAIR077*3A\r\n

Response:

$PAIR001,077,0*3B\r\n ==> Success

$PAIR077,0*26\r\n

2.3.53 Packet Type:078 PAIR_COMMON_SET_DATUM_ADVANCE

Set user-defined datum

DataField: $PAIR078,<majA>,<ecc>,<dX>,<dY>,<dZ>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
majA	m	--	User defined datum semi-major axis [m] [Range: 0 ~ 7000000]
ecc	m	--	User defined datum eccentric [m] [Range: 0 ~ 330]
dX	m	--	User defined datum to WGS84 X axis offset [m]
dY	m	--	User defined datum to WGS84 X axis offset [m]
dZ	m	--	User defined datum to WGS84 X axis offset [m]

Return&Example

[Return]

1. PAIR_ACK for send result

[Example]

Send:

$PAIR078,6377397.155,299.1528128,-148.0,507.0,685.0*10\r\n

Response:

$PAIR001,078,0*34\r\n ==> Success

2.3.54 Packet Type:079 PAIR_COMMON_GET_DATUM_ADVANCE

Get user-defined datum

DataField: $PAIR079*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

```
[Return]
    1. PAIR_ACK for send result
    2. $PAIR079,<majA>,<ecc>,<dX>,<dY>,<dZ>*CS<CR><LF>
    majA: User defined datum semi-major axis [m] [Range: 0 ~ 7000000]
    ecc: User defined datum eccentric [m] [Range: 0 ~ 330]
    dX: User defined datum to WGS84 X axis offset [m]
    dY: User defined datum to WGS84 X axis offset [m]
    dZ: User defined datum to WGS84 X axis offset [m]
[Example]
Send:
     $PAIR079*34\r\n
Response:
     $PAIR001,079,0*35\r\n ==> Success
     $PAIR079,6377397.155, 299.1528128, -148.0, 507.0,685.0*31\r\n
```

2.3.55 Packet Type:590 PAIR_TIME_SET_REF_UTC

Set current UTC time set in GNSS chip

DataField: $PAIR590,<YYYY>,<MM>,<DD>,<hh>,<mm>,<ss>*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

```
[Return]
    1. PAIR_ACK for send result
    2. $PAIR591,<YYYY>,<MM>,<DD>,<hh>,<mm>,<ss>*CS<CR><LF>
        YYYY   year      > 1980   UTC time: year in 4 digits
        MM     month     1 - 12   UTC time: month
```

DD day 1 - 31 UTC time: day
hh hour 0 - 23 UTC time: hour
mm minute 0 - 59 UTC time: minute
ss second 0 - 59 UTC time: second
[Example]
Send:
$PAIR590,2019,2,10,9,0,58*0B\r\n
Response:
$PAIR001,590,0*37\r\n ==> Success

2.3.56 Packet Type:591 PAIR_TIME_GET_REF_UTC

Query current UTC time set in GNSS chip

DataField: $PAIR591*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR591,<YYYY>,<MM>,<DD>,<hh>,<mm>,<ss>*CS<CR><LF>
YYYY year > 1980 UTC time: year in 4 digits
MM month 1 - 12 UTC time: month
DD day 1 - 31 UTC time: day
hh hour 0 - 23 UTC time: hour
mm minute 0 - 59 UTC time: minute
ss second 0 - 59 UTC time: second
[Example]
Send:
$PAIR591*37\r\n
Response:
$PAIR001,591,0*36\r\n ==> Success
$PAIR591,2000,01,01,01,01,36*30 \r\n

2.3.57 Packet Type:083 PAIR_COMMON_GET_HIGH_SENSITIVITY_TRACKING_MODE

Query setting of position output disabled/enabled in high-sensitivity tracking mode

DataField: $PAIR083*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR083,<Status>*CS<CR><LF>
0: Enable, 1: Disable

[Example]
Send:
$PAIR083*31\r\n
Response:
$PAIR001,083,0*30\r\n ==> Success
$PAIR083,0*2D\r\n ==> Enable high sensitivity tracking mode. GNSS system will get fix in high sensitivity tracking

2.3.58 Packet Type:030 PAIR_COMMON_GET_POS_XYZ

The WGS84 ECEF XYZ Cartesian Position vector (in meters) with an estimated 1-sigma accuracy

DataField: $PAIR030*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR030,<X>,<Y>,<Z>,<Acc>*CS<CR><LF>
X: WGS84 ECEF X Cartesian position (meters)
Y: WGS84 ECEF Y Cartesian position (meters)
Z: WGS84 ECEF Z Cartesian position (meters)
Acc: 3-dimensional position space 1-sigma accuracy estimate (in meters)

[Example]
Send:
$PAIR030*39\r\n
Response:

$PAIR001,030,0*38\r\n ==> Success
$PAIR030,-2984524.0,4966958.3,2656485.3,3.0*14\r\n ==> The WGS84 ECEF XYZ Cartesian Position

2.3.59 Packet Type:031 PAIR_COMMON_GET_VEL_XYZ

The WGS84 ECEF XYZ Cartesian velocity vector (m/s) with an estimated 1-sigma accuracy

DataField: $PAIR031*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR031,<VX>,<VY>,<VZ>,<Acc>*CS<CR><LF>
VX: WGS84 ECEF X Cartesian velocity vector (m/s).
VY: WGS84 ECEF Y Cartesian velocity vector (m/s).
VZ: WGS84 ECEF Z Cartesian velocity vector (m/s).
Acc: 3-dimensional speed 1-sigma accuracy (m/s)
[Example]
Send:
$PAIR031*38\r\n
Response:
$PAIR001,031,0*39\r\n ==> Success
$PAIR031,0.19,-0.07,-0.11,0.49*3A\r\n ==> The WGS84 ECEF XYZ Cartesian Velocity

2.3.60 Packet Type:400 PAIR_DGPS_SET_MODE

DGPS correction data source mode

DataField: $PAIR400,<Mode> *CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Mode	--	--	DGPS data source mode: '0': No DGPS source '1': RTCM '2': SBAS(Include WAAS/EGNOS/GAGAN/MSAS)

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR400,2*20\r\n ==> Set SBAS Mode
Response:
$PAIR001,400,0*3F\r\n ==> Success

2.3.61 Packet Type:401 PAIR_DGPS_GET_MODE

Query the DGPS data source mode

DataField: $PAIR401*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Mode	--	--	DGPS data source mode: '0': No DGPS source '1': RTCM '2': SBAS(Include WAAS/EGNOS/GAGAN/MSAS)

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR401,<Mode>*CS<CR><LF>
Mode: DGPS data source mode
'0': No DGPS source
'1': RTCM
'2': SBAS(Include WAAS/EGNOS/GAGAN/MSAS)
[Example]
Send:
$PAIR401*3F\r\n
Response:
$PAIR001,401,0*3E\r\n ==> Success
$PAIR401,2*21\r\n ==> SBAS Mode

2.3.62 Packet Type:410 PAIR_SBAS_ENABLE

Enable searching a SBAS satellite or not

DataField: $PAIR410,<Enabled>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Enabled	--	--	Enable or disable: '0' = Disable '1' = Enable

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR410,1*22\r\n ==> Enable SBAS
Response:
$PAIR001,410,0*3E\r\n ==> Success

2.3.63 Packet Type:411 PAIR_SBAS_GET_STATUS

Query the status of SBAS to whether it is enabled.

DataField: $PAIR411*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result
2. $PAIR411,<Enabled>*CS<CR><LF>
Enabled: Enable or disable
'0' = Disable
'1' = Enable
[Example]
Send:
$PAIR411*3E\r\n

Response:

$PAIR001,411,0*3F\r\n ==> Success

$PAIR411,1*23\r\n ==> Enable SBAS

2.3.64 Packet Type:020 PAIR_GET_VERSION

Query the firmware release information

DataField: $PAIR020*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]

1. PAIR_ACK for send result

2. $PAIR020,<Project Version>,<Frequency>,<SW package>,<Service version>,<Service build time>

<DSP L1 rom version>,<DSP L1 ram version>,<DSP L5 rom version>,<DSP L5 ram version>,

<Kernel version>,<Kernel build time>,<KF version>,<KF build time>,

<RTK version>,<RTK build time>*CS<CR><LF>

Project Version:

<Project_board>_<SDK version>_<SDK Build time>

<Project_board> AG3335A / AG3335M / AG3335S

<SDK version> VX.Y.Z - X:Major Y:Minor Z. Bug fix

<SDK build time> YYYYMMDD

Ex:

AG3335A_V1.0.0_20190729

Frequency:

S: single

D: dual

SW package:

N: normal

W: raw

T: timing

R: RTK

I: NavIC

Service version:

mnl_service version in 7 characters

Ex:

xxxxxxx

Service build time:
mnl_service library build time
Ex:
yyMMDDhhmm
DSP L1 rom version:
Null before first power on
Ex:
xx
DSP L1 ram version:
Null before first power on
Ex:
xxx
DSP L5 rom version:
Null for L1 only project
Null before first power on
Ex:
xx
DSP L5 ram version:
Null for L1 only project
Null before first power on
Ex:
xxx
Kernel version:
mnl_kernel version in 7 characters
Ex:
xxxxxxx
Kernel build time:
mnl_kernel library build time
Ex:
yyMMDDhhmm
KF version:
mnl_kf version in 7 characters
Ex:
xxxxxxx
KF build time:
mnl_kf library build time
Ex:
yyMMDDhhmm
RTK version:
RTK version in 7 characters
anything other than the RTK project
Ex:
xxxxxxx
RTK build time:
RTK library build time
Null for not RTK project

Ex:
yyMMDDhhmm
[Example]
Send:
$PAIR020*38\r\n
Response:
$PAIR001,020,0*39\r\n ==> Success

$PAIR020,AG3335A_V1.0.0_YYYYMMDD,D,N,xxxxxxx,yyMMDDhhmm,xx,xxx,xx,xxx,xxxxxxx,yyMMDDhhmm,xxxxxxx,yyMMDDhhmm,,*40\r\n

2.3.65 Packet Type:470 PAIR_EPO_GET_STATUS

Query the EPO data status stored in the GPS chip

DataField: $PAIR470,<System_ID>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
System_ID	--	--	The GNSS system ID: '0' = GPS '1' = GLONASS '2' = Galileo '3' = BeiDou

Return&Example

[Return]
1. PAIR_ACK for send result.
2.
$PAIR470,<System_ID>,<Set>,<FWN>,<FTOW>,<LWN>,<LTOW>,<FCWN>,<FCTOW>,<LCWN>,<LCTOW>*CS<CR><LF>
System_ID: The GNSS system ID.
'0' = GPS
'1' = GLONASS
'2' = Galileo
'3' = BeiDou
Set: Total number sets of EPO data stored in chip
FWN, FTOW: GPS week number & TOW of the first set of EPO data stored in chip respectively (flash)
LWN, LTOW: GPS week number & TOW of the last set of EPO data stored in chip respectively (flash)
FCWN, FCTOW: GPS week number & TOW of the first set of EPO data that are currently used

respectively
LCWN, LCTOW: GPS week number & TOW of the last set of EPO data that are currently used respectively
[Example]
Send:
$PAIR470,0*25\r\n
Response:
$PAIR001,470,0*38\r\n ==> Success
$PAIR470,0,1,2098,194400,2098,216000,2098,194400,2098,216000*38\r\n

2.3.66 Packet Type:530 PAIR_EPH_GET_STATUS

Get the EPH status in the next few seconds

DataField: $PAIR530,<Constellation>,<Time_interval>*CS<CR><LF>

Name	Unit	Default	Description
Constellation	--	--	The GNSS system ID: '0' = GPS '1' = GLONASS '2' = Galileo '3' = BeiDou '4' = QZSS
Time_interval	--	--	The range is between 1 and 7200 seconds (2 hours). The unit is seconds

Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR530,<L1_SV>,<L5_SV>*CS<CR><LF>
The valid ephemeris SV is in HEX format.
GLONASS only reports <L1_SV>.
Only dual packet reports both <L1_SV> and <L5_SV>.
[Example]
Send:
$PAIR530,1,1800*04\r\n
This command queries the status of GPS ephemeris after 1800 seconds in the future.
Response:
$PAIR001,530,0*3D\r\n ==> Success
$PAIR530,40449464,00800000*3F\r\n
Note the HEX 40449464 means 0100 0000 1000 0100 1001 0100 0110 0100 and the valid L1 SV

numbers are 3,6,7,11,13,16,19,24,31, while
the HEX 00800000 means 0000 0000 1000 0000 0000 0000 000 0000 and the valid L5 SV number is 24

2.3.67 Packet Type:550 PAIR_ALM_GET_STATUS

Get the ALM status in the next few days

DataField: $PAIR550,<Constellation>,<Time_interval>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
Constellation	--	--	The GNSS system ID: '0' = GPS '1' = GLONASS '2' = Galileo '3' = BeiDou '4' = QZSS
Time_interval	--	--	Time_interval: The range is between 1 and 91 days. The unit is day

Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR550,<L1_SV>,<Midi_SV>*CS<CR><LF>
The valid almanac SV is in HEX format
GLONASS only reports <L1_SV>
Only dual packet reports both <L1_SV> and <L5_SV>
[Example]
1 Send:
$PAIR550,1,30*08\r\n
This command queries the status of the GPS almanac after 30 days in the future
Response:
$PAIR001,550,0*3B\r\n ==> Success
$PAIR550,FEC0BFFF,00000FFF*38\r\n
Note the HEX FEC0BFFF means 1111 1110 1100 0000 1011 1111 1111 1111 and the valid L1 SV numbers are 1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,23,24,26,27,28,29,30,31,32
The HEX 00000FFF means 0000 0000 0000 0000 0000 1111 1111 1111 and the valid Midi almanac SV numbers are 1,2,3,4,5,6,7,8,9,10,11,12

2.3.68 Packet Type:392 PAIR_TEST_JAMMING_SCAN

Jamming scan test command

DataField: $PAIR392, <JamScanType>,<JamScanNum>,<GloSubChan>,<Resolution>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
JamScanType	--	--	'0' enable GPS L1 band jamming scan '1' enable GLONASS L1 band jamming scan '2' enable BeiDou L1 band jamming scan '3' enable L5 band jamming scan
JamScanNum	--	--	Jamming scan test times. [Range: 1~255]
GloSubChan	--	--	GLONASS sub channel
Resolution	--	--	Jamming scan frequency resolution (L1 band only support Legacy, L5 band only support 50Hz) '0' Legacy (21KHz~61KHz) '1' 50Hz

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR392,0,50,0,0*07\r\n
GPS L1 band jamming scan test 50 times
Response:
$PAIR001,392,0*33\r\n ==> Success

2.3.69 Packet Type:391 PAIR_TEST_JAMMING_DETECT

Jamming detection test command

DataField: $PAIR391, <CmdType>*CS<CR><LF>			
Name	**Unit**	**Default**	**Description**
CmdType	--	--	"0" disable jamming detection message output. "1" enable jamming detection message output

Return&Example

```
[Return]
    1. PAIR_ACK for send result
    2.  $PAIR391,<Jamstatus>*CS<CR><LF>
    Jamstatus: "0" Unknown Status
               "1" No jamming, healthy status
               "2" Warning status
               "3" Critical status
[Example]
Send:
    $PAIR391,1*2C\r\n
    Enable the jamming detection message output
Response:
    $PAIR001,391,0*30\r\n ==> Success
    $PAIRSPF,1*52\r\n ==> L1 band result
    $PAIRSPF5,1*67\r\n ==> L5 band result
Send:
    $PAIR391,0*2D\r\n
    Disable the jamming detection message output
Response:
    $PAIR001,391,0*30\r\n ==> Success
```

2.3.70 Packet Type:490 PAIR_EASY_ENABLE

Enable or disable EASY function

DataField: $PAIR490,<Enable>*CS<CR><LF>

Name	Unit	Default	Description
Enable	--	--	Enable or disable: '0': Disable '1': Enable

Return&Example

```
[Return]
    1. PAIR_ACK for send result
[Example]
Send:
    $PAIR490,1*2A\r\n
Response:
```

$PAIR001,490,0*36\r\n ==> Success

2.3.71 Packet Type:491 PAIR_EASY_GET_STATUS

Query whether EASY is enabled or disabled

DataField: $PAIR491*CS<CR><LF>			
Name	Unit	Default	Description
--	--	--	--

Return&Example

[Return]
1. PAIR_ACK for send result.
2. $PAIR490,<Enable>,<Status>*CS<CR><LF>
Enable: Enable or disable
'0': Disable
'1': Enable
Status:
'0': Not finished
'1': finished 1-day extension
'2': finished 2-day extension
'3': finished 3-day extension

[Example]
1 Send:
2 $PAIR491*36\r\n
3 Response:
4 $PAIR001,491,0*37\r\n ==> Success
5 $PAIR491,1,0*37\r\n

2.3.72 Packet Type:080 PAIR_COMMON_SET_NAVIGATION_MODE

Set navigation mode

DataField: $PAIR080,<CmdType>*CS<CR><LF>			
Name	Unit	Default	Description
CmdType	--	--	'0'[Default Value] Normal mode: For general purpose

			'1' Fitness mode: For running and walking purpose so that the low-speed (< 5 m/s) movement will have more of an effect on the position calculation '2' Not Support '3' Not Support '4' Stationary mode: For stationary applications where a zero dynamic assumed '5' Not Support '6' Not Support '7' Swimming mode: For swimming purpose so that it smooths the trajectory and improves the accuracy of distance calculation

Return&Example

[Return]
1. PAIR_ACK for send result
[Example]
Send:
$PAIR080,1*2F\r\n ==> Enter fitness mode
Response:
$PAIR001,080,0*33\r\n ==> Success

AppendixC Datum List

All the datum type supported are shown in this table.

No	Datum	Region
0	WGS1984	International
1	Tokyo	Japan
2	Tokyo	Mean For Japan, South Korea, Okinawa
3	User Setting	User Setting
4	Adindan	Burkina Faso
5	Adindan	Cameroon
6	Adindan	Ethiopia

7	Adindan	Mali
8	Adindan	Mean For Ethiopia, Sudan
9	Adindan	Senegal
10	Adindan	Sudan
11	Afgooye	Somalia
12	Ain El Abd1970	Bahrain
13	Ain El Abd1970	Saudi Arabia
14	American Samoa1962	American Samoa Islands
15	Anna 1 Astro1965	Cocos Island
16	Antigua Island Astro1943	Antigua(Leeward Islands)
17	Arc1950	Botswana
18	Arc1950	Burundi
19	Arc1950	Lesotho
20	Arc1950	Malawi
21	Arc1950	Mean For Botswana, Lesotho, Malawi, Swaziland, Zaire, Zambia, Zimbabwe
22	Arc1950	Swaziland
23	Arc1950	Zaire
24	Arc1950	Zambia
25	Arc1950	Zimbabwe
26	Arc1960	Mean For Kenya Tanzania
27	Arc1960	Kenya
28	Arc1960	Tamzamia
29	Ascension Island1958	Ascension Island
30	Astro Beacon E 1945	Iwo Jima
31	Astro Dos 71/4	St Helena Island

32	Astro Tern Island (FRIG) 1961	Tern Island
33	Astronomical Station 1952	Marcus Island
34	Australian Geodetic 1966	Australia, Tasmania
35	Australian Geodetic 1984	Australia, Tasmania
36	Ayabelle Lighthouse	Djibouti
37	Bellevue (IGN)	Efate and Erromango Islands
38	Bermuda 1957	Bermuda
39	Bissau	Guuinea-Bissau
40	Bogota Observatory	Colombia
41	Bukit Rimpah	Indonesia(Bangka and Belitung Ids)
42	Camp Area Astro	Antarctica(McMurdi Camp Area)
43	Campo Inchauspe	Argentina
44	Canton Astro1966	Phoenix Island
45	Cape	South Africa
46	Cape Canaveral	Bahamas, Florida
47	Carthage	Tunisia
48	Chatham Island Astro1971	New Zealand(Chatham Island)
49	Chua Astro	Paraguay
50	Corrego Alegre	Brazil
51	Dabola	Guinea
52	Deception Island	Deception Island, Antarctia
53	Djakarta (Batavia)	Indonesia(Sumatra)
54	Dos 1968	New Georgia Islands (Gizo Island)

55	Easter Island 1967	Easter Island
56	Estonia Coordinate System1937	Estonia
57	European 1950	Cyprus
58	European 1950	Egypt
59	European 1950	England, Channel Islands, Scotland, Shetland Islands
60	European 1950	England, Ireland, Scotland, Shetland Islands
61	European 1950	Finland, Norway
62	European 1950	Greece
63	European 1950	Iran
64	European 1950	Italy (Sardinia)
65	European 1950	Italy (Slcily)
66	European 1950	Malta
67	European 1950	Mean For Austria, Belgium,Denmark, Finland, France, W Germany, Gibraltar, Greece, Italy, Luxembourg, Netherlands, Norway, Portuga,l Spain, Sweden, Switzerland
68	European 1950	Mean For Austria, Debnmark,France, W Germany, Netherland ,Switzerland
69	European 1950	Mean For Irag, Israel, Jordan, Lebanon, Kuwait, Saudi Arabia, Syria
70	European 1950	Portugal, Spain
71	European 1950	Tunisia,
72	European 1979	Mean For Austria, Finland ,Netherlands ,Norway, Spain, Sweden, Switzerland
73	Fort Thomas 1955	Nevis St Kitts (Leeward Islands)
74	Gan 1970	Republic Of Maldives
75	Geodetic Dataum 1970	New Zealand
76	Graciosa Base SW1948	Azores (Faial, Graciosa, Pico, Sao, Jorge, Terceria)

77	Guam1963	Guam
78	Gunung Segara	Indonesia (Kalimantan)
79	Gux l Astro	Guadalcanal Island
80	Herat North	Afghanistan
81	Hermannskogel Datum	Croatia-Serbia, Bosnia-Herzegoivna
82	Hjorsey 1955	Iceland
83	Hongkong 1963	Hongkong
84	Hu Tzu Shan	Taiwan
85	Indian	Bangladesh
86	Indian	India,Nepal
87	Indian	Pakistan
88	Indian 1954	Thailand
89	Indian 1960	Vietnam (Con Son Island)
90	Indian 1960	Vietnam (Near 16 deg N)
91	Indian 1975	Thailand
92	Indonesian 1974	Indonesian
93	Ireland 1965	Ireland
94	ISTS 061 Astro 1968	South Georgia Islands
95	ISTS 073 Astro 1969	Diego Garcia
96	Johnston Island 1961	Johnston Island
97	Kandawala	Sri Lanka
98	Kerguelen Island 1949	Kerguelen Island
99	Kertau 1948	West Malaysia and Singapore
100	Kusaie Astro 1951	Caroline Islands
101	Korean Geodetic System	South Korea

102	LC5 Astro 1961	Cayman Brac Island
103	Leigon	Ghana
104	Liberia 1964	Liberia
105	Luzon	Philippines (Excluding Mindanao)
106	Luzon	Philippines (Mindanao)
107	M'Poraloko	Gabon
108	Mahe 1971	Mahe Island
109	Massawa	Ethiopia (Eritrea)
110	Merchich	Morocco
111	Midway Astro 1961	Midway Islands
112	Minna	Cameroon
113	Minna	Nigeria
114	Montserrat Island Astro 1958	Montserrat (Leeward Island)
115	Nahrwan	Oman (Masirah Island)
116	Nahrwan	Saudi Arabia
117	Nahrwan	United Arab Emirates
118	Naparima BWI	Trinidad and Tobago
119	North American 1927	Alaska (Excluding Aleutian Ids)
120	North American 1927	Alaska (Aleutian Ids East of 180 degW)
121	North American 1927	Alaska (Aleutian Ids West of 180 degW)
122	North American 1927	Bahamas (Except San Salvador Islands)
123	North American 1927	Bahamas (San Salvador Islands)
124	North American 1927	Canada (Alberta, British Columbia)
125	North American 1927	Canada (Manitoba, Ontario)
126	North American 1927	Canada (New Brunswick, Newfoundland, Nova Scotia, Qubec)

127	North American 1927	Canada (Northwest Territories, Saskatchewan)
128	North American 1927	Canada (Yukon)
129	North American 1927	Canal Zone
130	North American 1927	Cuba
131	North American 1927	Greenland (Hayes Peninsula)
132	North American 1927	Mean For Antigua, Barbados, Barbuda, Caicos Islands, Cuba, Dominican, Grand Cayman, Jamaica, Turks Islands
133	North American 1927	Mean For Belize, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua
134	North American 1927	Mean For Canada
135	North American 1927	Mean For Conus
136	North American 1927	Mean For Conus (East of Mississippi, River Including Louisiana,Missouri, Minnesota)
137	North American 1927	Mean For Conus (West of Mississippi, Rive Excluding Louisiana, Minnesota, Missouri)
138	North American 1927	Mexico
139	North American 1983	Alaska (Excluding Aleutian Ids)
140	North American 1983	Aleutian Ids
141	North American 1983	Canada
142	North American 1983	Conus
143	North American 1983	Hahawii
144	North American 1983	Mexico, Central America
145	North Sahara 1959	Algeria
146	Observatorio Meteorologico 1939	Azores (Corvo and Flores Islands)
147	Old Egyptian 1907	Egypt
148	Old Hawaiian	Hawaii
149	Old Hawaiian	Kauai

150	Old Hawaiian	Maui
151	Old Hawaiian	Mean For Hawaii, Kauai, Maui, Oahu
152	Old Hawaiian	Oahu
153	Oman	Oman
154	Ordnance Survey Great Britian 1936	England
155	Ordnance Survey Great Britian 1936	England, Isle of Man, Wales
156	Ordnance Survey Great Britian 1936	Mean For England ,Isle of Man, Scotland, Shetland Island, Wales
157	Ordnance Survey Great Britian 1936	Scotland, Shetland Islands
158	Ordnance Survey Great Britian 1936	Wales
159	Pico de las Nieves	Canary Islands
160	Pitcairn Astro 1967	Pitcairn Island
161	Point 58	Mean For Burkina Faso and Niger
162	Pointe Noire 1948	Congo
163	Porto Santo 1936	Porto Santo, Maderia Islands
164	Provisional South American 1956	Bolovia
165	Provisional South American 1956	Chile (Northern Near 19 deg S)
166	Provisional South American 1956	Chile (Southern Near 43 deg S)
167	Provisional South American 1956	Colombia
168	Provisional South American 1956	Ecuador
169	Provisional South	Guyana

	American 1956	
170	Provisional South American 1956	Mean For Bolivia Chile,Colombia, Ecuador, Guyana, Peru, Venezuela
171	Provisional South American 1956	Peru
172	Provisional South American 1956	Venezuela
173	Provisional South Chilean 1963	Chile (Near 53 deg S) (Hito XVIII)
174	Puerto Rico	Puerto Rico, Virgin Islands
175	Pulkovo 1942	Russia
176	Qatar National	Qatar
177	Qornoq	Greenland (South)
178	Reunion	Mascarene Island
179	Rome 1940	Italy (Sardinia)
180	S-42 (Pulkovo 1942)	Hungary
181	S-42 (Pulkovo 1942)	Poland
182	S-42 (Pulkovo 1942)	Czechoslavakia
183	S-42 (Pulkovo 1942)	Lativa
184	S-42 (Pulkovo 1942)	Kazakhstan
185	S-42 (Pulkovo 1942)	Albania
186	S-42 (Pulkovo 1942)	Romania
187	S-JTSK	Czechoslavakia (Prior 1 Jan1993)
188	Santo (Dos) 1965	Espirito Santo Island
189	Sao Braz	Azores (Sao Miguel, Santa Maria Ids)
190	Sapper Hill 1943	East Falkland Island
191	Schwarzeck	Namibia
192	Selvagem Grande	Salvage Islands

	1938	
193	Sierra Leone 1960	Sierra Leone
194	South American 1969	Argentina
195	South American 1969	Bolivia
196	South American 1969	Brazial
197	South American 1969	Chile
198	South American 1969	Colombia
199	South American 1969	Ecuador
200	South American 1969	Ecuador (Baltra, Galapagos)
201	South American 1969	Guyana
202	South American 1969	Mean For Argentina, Bolivia, Brazil,Chile, Colombia, Ecuador, Guyana, Paraguay, Peru, Trinidad and Tobago, Venezuela
203	South American 1969	Paraguay
204	South American 1969	Peru
205	South American 1969	Trinidad and Tobago
206	South American 1969	Venezuela
207	South Asia	Singapore
208	Tananarive Observatory 1925	Madagascar
209	Timbalai 1948	Brunei, E Malaysia (Sabah Sarawak)
210	Tokyo	Japan
211	Tokyo	Mean For Japan, South Korea, Okinawa
212	Tokyo	Okinawa
213	Tokyo	South Korea
214	Tristan Astro 1968	Tristam Da Cunha
215	Viti Levu 1916	Fiji (Viti Levu Island)
216	Voirol 1960	Algeria

217	Wake Island Astro 1952	Wake Atoll
218	Wake-Eniwetok 1960	Marshall Islands
219	WGS 1972	Global Definition
220	WGS 1984	Global Definition
221	Yacare	Uruguay
222	Zanderij	Suriname
223	PZ-90 v11	GLONASS