



深圳市天工测控技术有限  
公司

文件编号

QW-RD-21

## NMEA Packet

版本号

A/1

发行日期

2021-7-26

### History

Revision	Description	Approved	Date
A/0	Initial Release	Simon	20210726

### NMEA Packet Format:

Preamble	TalkerID	PktType	DataField	*	CHK1	CHK2	CR	LF
----------	----------	---------	-----------	---	------	------	----	----

#### Packet Length:

The maximum length of each packet is restricted to **255** bytes.

#### Packet Contents:

**Preamble:** One byte character.

'\$'

**TalkerID:** Four bytes character string.

"PMTK"

**PktType:** Three bytes character string.

From "000" to "999"

An identifier used to tell the decoder how to decode the packet.

**DataField:** The DataField has variable length depending on the packet type.

A comma symbol ',' must be inserted ahead each data field to help the decoder process the DataField.

\*: 1 byte character. The star symbol is used to mark the end of DataField.

**CHK1, CHK2:** Two bytes character string. CHK1 and CHK2 are the checksum of the data between Preamble and '\*'. \*

**CR, LF:** Two bytes binary data. The two bytes are used to identify the end of a packet.

#### Sample Packet:

\$PMTK000\*32<CR><LF>

### NMEA Packet Protocol:

In order to inform the sender whether the receiver has received the packet, an acknowledge packet PMTK\_ACK should return after the receiver receives a packet.

### CK1 And CK2

CHK1 and CHK2 are the checksum of the data between Preamble and '\*', the checksum is the exclusive OR of all characters between

Preamble and '\*'. The C reference example:

```
char a[]="$PMTK000*32\r\n";
char Checksum,Checksum1,Checksum2;
int i;
for(Checksum=a[1],i=2;a[i]!='*' && i<255;++i)
{
    Checksum^=a[i];
}
printf("Checksum =%02X\n\n",Checksum);// "32"
if( (Checksum/0x10)+0x30 <=57) //Checksum1
    Checksum1=Checksum/0x10+0x30;
else
    Checksum1=Checksum/0x10+0x30+0x07;
if( (Checksum%0x10)+0x30 <=57) //Checksum2
    Checksum2=Checksum%0x10+0x30;
else
    Checksum2=Checksum%0x10+0x30+0x07;
if(a[++i]==Checksum1)
    printf("Checksum1 =%C\n",Checksum1);// '3'
if(a[++i]==Checksum2)
    printf("Checksum2 =%C\n",Checksum2);// '2'
```

## Commonly Used Commands

Note: <CR> is carriage return character, <LF> is line feed character.

## NMEA Packet List:

Packet Type: 870 PAIR_IO_TEST.....	4
Packet Type: 001 PAIR_ACK .....	4
Packet Type: 004 PAIR_GNSS_SUBSYS_HOT_START.....	4
Packet Type: 005 PAIR_GNSS_SUBSYS_WARM_START.....	4
Packet Type: 006 PAIR_GNSS_SUBSYS_COLD_START.....	4
Packet Type: 007 PAIR_GNSS_SUBSYS_FULL_COLD_START.....	5
Packet Type: 472 PAIR_EPO_ERASE_FLASH_DATA.....	5
Packet Type: 050 PAIR_COMMON_SET_FIX_RATE.....	5
Packet Type: 690 PAIR_PERIODIC_SET_MODE.....	5
Packet Type: 864 PAIR_IO_SET_BAUDRATE.....	6
Packet Type: 062 PAIR_COMMON_SET_NMEA_OUTPUT_RATE.....	6
Packet Type: 066 PAIR_COMMON_SET_GNSS_SEARCH_MODE.....	7
Packet Type: 068 PAIR_COMMON_SET_HDOP_THRESHOLD.....	7
Packet Type: 069 PAIR_COMMON_GET_HDOP_THRESHOLD.....	8
Packet Type: 070 PAIR_COMMON_SET_STATIC_THRESHOLD.....	8
Packet Type: 410 PAIR_SBAS_ENABLED.....	8
Packet Type: 411 PAIR_SBAS_GET_STATUS.....	8
Packet Type: 020 PAIR_GET_VERSION.....	9

## Packet Type: 870 PAIR\_IO\_TEST

### Packet Meaning:

Check if PAIR channel is ready to work.

### DataField:

None

### Example:

```
$PAIR870*35<CR><LF>
```

## Packet Type: 001 PAIR\_ACK

### Packet Meaning:

Acknowledge of PAIR command

### DataField:

***PAIR001,Cmd,Flag***

Cmd: The command / packet type the acknowledge responds.

Flag: The result of the command.

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:

```
$PAIR001,666,0*3D<CR><LF>
```

## Packet Type: 004 PAIR\_GNSS\_SUBSYS\_HOT\_START

### Packet Meaning:

Hot Restart: Use all available data in the NVRAM.

### DataField:

None

### Example:

```
$PAIR004*3E<CR><LF>
```

## Packet Type: 005 PAIR\_GNSS\_SUBSYS\_WARM\_START

### Packet Meaning:

Warm Restart. Don't use Ephemeris data at the start.

### DataField:

None

### Example:

```
$PAIR005*3F<CR><LF>
```

## Packet Type: 006 PAIR\_GNSS\_SUBSYS\_COLD\_START

### Packet Meaning:

Cold Restart: Don't use Time, Position, Almanacs and Ephemeris data at re-start.

### DataField:

None

### Example:

```
$PAIR006*3C<CR><LF>
```

## Packet Type: 007 PAIR\_GNSS\_SUBSYS\_FULL\_COLD\_START

### Packet Meaning:

Full Cold Restart: It's essentially a Cold Restart, but additionally clear system/user configurations at re-start. It reset the GNSS module to the factory status.

### DataField:

None

### Example:

```
$PAIR007*3D<CR><LF>
```

## Packet Type: 472 PAIR\_EPO\_ERASE\_FLASH\_DATA

### Packet Meaning:

Erase EPO data stored in the flash memory.

### DataField:

None

### Example:

```
$PAIR472*3B<CR><LF>
```

## Packet Type: 050 PAIR\_COMMON\_SET\_FIX\_RATE

### Packet Meaning:

Position Fix Interval.

If set less than 1000 ms, ASCII NMEA will automatically increase the update interval in order to decrease IO throughput.

It will return false if the operating voltage setting is not correct.

(ULP mode only support 1Hz)

### DataField:

```
$PAIR050,<Interval>
```

Interval: Position fix interval [msec].[Range:100~1000].

### Example:

```
$PAIR050,1000*12<CR><LF>
```

## Packet Type: 690 PAIR\_PERIODIC\_SET\_MODE

### Packet Meaning:

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.

<Note> Sleep will be interrupted by any DSP corresponding PAIR command.

Any restart will force it to go back to normal mode.

### DataField:

```
$PAIR690,<Mode>,<FirstRun>,<FirstSleep>,<SecondRun>,<SecondSleep>
```

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".

#### Example:

```
PAIR690,1,21,39,48,72*28\r\n
```

### Packet Type: 864 PAIR\_IO\_SET\_BAUDRATE

#### Packet Meaning

Set port baud rate configuration.

#### Data Field

**\$PAIR864,<Port\_Type>,<Port\_Index>,<Baudrate>**

Port\_Type: HW Port Type.

0: UART

Port\_Index: HW Port Index

UART - 0: UART0, 1: UART1, 2: UART2

Baudrate: the baud rate need config.

Support 115200, 230400, 460800, 921600, 3000000

#### Example

```
$PAIR864,0,0,115200*1B<CR><LF>
```

### Packet Type: 062 PAIR\_COMMON\_SET\_NMEA\_OUTPUT\_RATE

#### Packet Meaning:

Set the NMEA sentence output interval of corresponding NMEA type.

#### DataField:

**\$PAIR062,<Type>,<Output\_Rate>**

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

7 NMEA\_SEN\_GRS, // GRS interval - GNSS Range Residuals

8 NMEA\_SEN\_GST, // GST Interval - GNSS Pseudorange Error Statistics

Output\_Rate: Output interval setting (Valid range: 0~20, default value: 1)

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:**

\$PAIR062,0,3\*3D<CR><LF>

**Packet Type: 066 PAIR\_COMMON\_SET\_GNSS\_SEARCH\_MODE****Packet Meaning:**

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

The device restarts when it receives this command.

Abbreviation: (GPS: "G", GLONASS: "R", Galileo: "E", BeiDou: "B", NavIC, "I")

Support constellation in L1 package: G/ GR/ GE/ GB/ GREB

Support constellation in L1 + L5 package: GREB

Support constellation in L1 + NavIC package G/ I/ GEI/ GREB/ GREBI

QZSS is always switchable.

**DataField:**

***\$PAIR066,<GPS\_Enabled>,<GLONASS\_Enabled>,<Galileo\_Enabled>,<BeiDou\_Enabled>,<QZSS\_Enabled>,<NavIC\_Enabled>***

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:**

\$PAIR066,1,0,0,0,0,0\*3B<CR><LF>: Search GPS satellites only.

\$PAIR066,1,1,1,1,1,0\*3B<CR><LF>: Search GPS, GLONASS, Galileo, BeiDou, QZSS satellites.

\$PAIR066,1,1,0,0,0,0\*3A<CR><LF>: Search GPS and GLONASS satellites.

**Packet Type: 068 PAIR\_COMMON\_SET\_HDOP\_THRESHOLD****Packet Meaning:**

This command is to set the HDOP threshold. If the HDOP value is larger than this threshold value, the position will not be fixed.

**DataField:**

***\$PAIR068,<HDOPThreshold>***

HDOPThreshold:

"0": Disable this function.

Other value: Enable setting the HDOP threshold.

**Example:**

\$PAIR068,0.8\*3E<CR><LF>

**Packet Type: 069 PAIR\_COMMON\_GET\_HDOP\_THRESHOLD****Packet Meaning:**

This command is to get the HDOP threshold.

**DataField:**

None

**Return**

1. PAIR\_ACK for send result.
2. \$PAIR069,<HDOPThreshold>

HDOPThreshold:

"0": Disable this function.

Other value: Enable setting the HDOP threshold

**Example:**

Send:

\$PAIR069\*35<CR><LF>

Response:

\$PAIR001,069,0\*34<CR><LF> ==> Success

\$PAIR069,0.8\*3F<CR><LF>

**Packet Type: 070 PAIR\_COMMON\_SET\_STATIC\_THRESHOLD****Packet Meaning:**

Set the speed threshold for static navigation.

If the actual speed is below the threshold, output position will keep the same and output speed will be zero.

If threshold value is set to 0, this function is disabled.

**DataField:**

**\$PAIR070,<Speed\_threshold>**

Speed\_threshold.

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

**Example:**

\$PAIR070,4\*25<CR><LF>

**Packet Type: 410 PAIR\_SBAS\_ENABLED****Packet Meaning:**

Enable to search a SBAS satellite or not.

When navigation mode is Fitness or Swimming mode, SBAS is not supported.

**DataField:**

**\$PAIR410,<Enabled>**

Enabled: Enable or disable

'0' = Disable

'1' = Enable

**Example:**

\$PAIR410,1\*22<CR><LF>

**Packet Type: 411 PAIR\_SBAS\_GET\_STATUS**



**Packet Meaning:**

Query the status of SBAS to whether it is enabled.

**DataField:**

None

**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
```

**Packet Type: 020 PAIR\_GET\_VERSION****Packet Meaning:**

Query the firmware release information.

**DataField:**

NONE

**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>

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

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

制作：

审核：

批准：