

L651_GPS_API And. AT instruction

Ver.1.0

LYNQ
CONFIDENTIAL

Version.

版本	日期	作者	修订点
1.0	2019/6/24	李勇	The initial version.

LYNQ
CONFIDENTIAL

Index

Table of Contents

Version	2
1. Overview.	4
1.1. Purpose and scope.	4
1.2. Defined.	4
1.3. Refer to the documentation.	4
2. API Interface.	5
2.1. GPS Initialization.	5
2.2. Launch a positioning.	5
2.3. End positioning.	6
2.4. Exit.	6
2.5. Set the startup mode.	6
2.6. Set up the positioning system.	6
2.7. Set the positioning frequency.	6
2.8. Get FW Version information.	6
2.9. Set the Baud rate.	6
2.10. Set the message output frequency.	7
2.11. Get secondary data.	7
2.12. Set debug mode.	7
3. Relevant AT instructions and functions.	8
3.1. at+gps= "ON"	8
3.2. at+gps= "START" , start_mode.	8
3.3. at+gps= "FWINFO"	8
3.4. at+gps= "POSMODE" , pos_mode.	8
3.5. at+gps= "ASSIST"	8
3.6. at+gps= "ASSIST" , fix_freq.	8
3.7. at+gps= "BAUD" , baud_rate.	8
3.8. at+gps= "NMEA" , type, flag, freq.	8
3.9. at+gps= "DEBUG"	8
3.10. at+gps= "STOP"	8
3.11. at+gps= "OFF"	8

1. Overview.

1.1. Purpose and scope.

This article describes the current GPS-related API interfaces and AT instructions.

1.2. Defined.

Table 1-1 Terminology

Term	Definition
GPS	Global Positioning System (GPS) of the United States.
BDS	China Beidou Navigation Satellite System.
GNSS	GlobalNavigationSatelliteSystem
BAUD	Baud rate
CR	Carriage Return
LF	Line Feed

1.3. Refer to the documentation.

Title	Location
NMEA-0183protocol	
GNSS The data interface protocol.	

2. API Interface.

2.1. GPS Initialization.

The function interface is as follows.:

```
int um_gps_init(GpsCallbacks* callbacks)
```

The main task of this function is to register the callback function and initialize the drive, after which the chip enters a low-power state. The parameter “callbacks” are callback function pointers, and the corresponding data structure is as follows:

```
typedef struct {  
    /** set to sizeof(GpsCallbacks) */  
    size_t      size;  
    gps_location_callback location_cb;  
    gps_status_callback status_cb;  
    gnss_sv_status_callback gnss_sv_status_cb;  
    gps_nmea_callback nmea_cb;  
} GpsCallbacks;  
/**  
 *Position callback function.  
 */  
typedef void (* gps_location_callback)(GpsLocation* location);  
/**  
 * The state callback function.  
 */  
typedef void (* gps_status_callback)(GpsStatus* status);  
/**  
 * Satellite status information callback function.  
 */  
typedef void (* gnss_sv_status_callback)(GnssSvStatus* sv_info);  
/**  
 * NMEA Statement callback function,  
 */  
typedef void (* gps_nmea_callback)(GpsUtcTime timestamp, const char* nmea,  
int length);
```

When used, the callback function is implemented to obtain the data of interest, including the location of the positioning, the state of the chip, and satellite information. and NMEA data.

2.2. Launch a positioning.

The function interface is as follows.:

```
int um_gps_start()
```

2.3. End positioning.

The function interface is as follows.:

```
int gps_stop()
```

After positioning is completed, the system enters a low-power state.

2.4. Exit.

The function interface is as follows:

```
int um_gps_cleanup()
```

This function frees up the resources requested at initialization and powers down the chip.

2.5. Set the startup mode.

The function interface is as follows:

```
void um_gps_set_start_mode (uint16_t mode)
```

This function is used to set cold start, hot start or warm start.

2.6. Set up the positioning system.

The function interface is as follows:

```
int um_gps_set_position_mode(uint16_t mode)
```

The system used to set the chip positioning can be set to single GPS, single Beidou or hybrid mode (BD-GPS) positioning.

2.7. Set the positioning frequency.

The function interface is as follows:

```
int um_gps_set_freq(U8freq)
```

This function mainly sets the frequency of escalation chip positioning and can set values of 1Hz, 2Hz, 5Hz.

2.8. Get FW Version information.

The function interface is as follows.:

```
void um_gps_fw_info()
```

After the command is successful, the chip outputs the following information:

```
$PDTINFO,N/A,G1B1,VN/A,R3.2.0Build3558M,N/A,N/A-6E
```

The six parameters mean the following:

Product name, product configuration options, hardware version number, firmware version number, product ID, serial number focus on firmware version number, such as R3.2.0Build3558M.

2.9. Set the Baud rate.

The function interface is as follows.:

```
void um_gps_set_baud(uint32_t baud)
```

Set the Baud rate for chip escalation data, which is currently supported as follows:

9600

14400

19200

33600

38400

57600

115200

230400

2.10. Set the message output frequency.

The function interface is as follows.:

```
void um_gps_set_nmea_output(U8 type, U8 flag, U8 freq)
```

Type: Message category, 0 – NMEA message 1 – Navigation message

For NMEA messages, the flag parameter participates in the ID column in the table below.

消息名	类别	ID	频度设置范围
NMEA Message			
GGA	0	0	0~5
GLL	0	1	0~5
GSA	0	2	0~5
GSV	0	3	0~5
RMC	0	4	0~5
VTG	0	5	0~5
ZDA	0	6	0~5

For navigation messages, the flag parameter participates in the ID column in the table below.

消息名	类别	ID	频度设置范围
Navigation Result Message			
POS	1	0	0~5
VEL	1	1	0~5
TIME	1	2	0~5
ACC	1	3	0~5

Freq's value range is 0-5, 0 to indicate that the corresponding message is not output, 1 to 5 for 1 to 5s output once.

2.11. Get secondary data.

The function interface is as follows:

```
void um_gps_req_assist(U8 type)
```

Requesting secondary data from the network currently only supports real-time astration information, valid for 2-4 hours.

It is recommended to increase the serial port rate to 115200 before use, otherwise the star calendar will take longer to download to the chip.

2.12. Set debug mode.

The function interface is as follows:

```
void um_gps_start_debug()
```

Primarily used for debugging problems, Firmware outputs more debugging information and needs to set the Baud rate to 115200.

3. Relevant AT instructions and functions.

3.1. at+gps="ON"

Initialize the GPS chip.

3.2.at+gps="START",start_mode

start_mode for positioning mode parameter

- 0 - hot start, default mode.
- 1. - Cold start.
- 2. - Warm start.

3.3.at+gps="FWINFO"

Print product information as detailed in 1.8.

3.4.at+gps="POSMODE",pos_mode

pos_mode The system used for positioning.

- 1 - GPS
- 2 - BDS
- 3 - BD + GPS

3.5.at+gps="ASSIST"

Requesting secondary data currently only supports downloading real-time astrations.

3.6.at+gps="ASSIST",fix_freq

The parameter fix_freq is the positioning frequency, and the supported values include 1000ms , 500ms , 200ms , corresponding to 1Hz, 2Hz, 5Hz.

3.7.at+gps="BAUD",baud_rate

The parameter baud_rate is the set baud rate. See section 1.9 for the supported baud rate .
The default baud rate is 9600

3.8.at+gps="NMEA", type, flag, freq

The value range of the parameter is exactly the same as that of API

um_gps_set_nmea_output in section 3.10 . If you need to set all messages of a certain category at once, you can set the flag parameter to 8 .

such as: at+gps= "NMEA",0, 8, 2

– Change the output frequency of all NMEA messages to once every 3 seconds, corresponding to 0.5 Hz .。

3.9. at+gps="DEBUG"

Open debugging mode, FW will output more information, mainly for debugging. Before sending the command, you need to set the baud rate to 115200.

3.10.at+gps="STOP"

Stop positioning and enter sleep mode

3.11.at+gps="OFF"

Turn off GPS and release resources