FIFOTRACK GPRS PROTOCOL

Model: A01

Version: V1.4

www.fifotrack.com

Copyright and Disclaimer

- All copyrights belong to Shenzhen fifotrack Solution Co., Ltd. You are not allowed to revise, copy or spread this file in any form without consent of fifotrack.
- is trademark of fifotrack, protected by law.
- Please read this user guide carefully before installation to avoid any possible personal injury or property loss.

Document History

Version	Revision Date	Author	Detail
V1.1	Nov 17, 2015	Vito Hu	Initial Version
V1.2	Sep 15, 2018	Vito Hu	Modify *bat-ad*, *ext-ad* fields description. Add GPRS heartbeat *A10*
V1.3	Mar 4, 2020	Vito Hu	Add GPRS protocol *A02*
V1.4	Dec 23, 2024	Vito Hu	Add supported model name R18 and A500G Update “status” field description

Contents

Document History **3**

1 GPRS Package Format **5**

2 Applied Models **6**

3 A01/A02 -- GPS Position/Alarm Data Format **7**

4 Server Response to A02 ..... **11**

5 A10 – GPRS Heartbeat Data Format **12**

Appendix A - Alarm Code and Alarm Parameter **14**

1 GPRS Package Format

GPRS uplink (i.e.: Data is sent from tracker to platform) command format:

$$<pack-len>,<ID>,<work-no>,<cmd-code>,<cmd-para>*<checksum>\r\n

GPRS downlink (i.e.: Data is sent form platform to tracker) command format:

##<pack-len>,<ID>,<work-no>,<cmd-code>,<cmd-para>*<checksum>\r\n

Remarks:

- Comma (,) is used to separate data fields, and it is necessary. There is no space before or after comma.
- pack-len: Package Length, decimal string format, the field of *pack-len* is {*,<ID>,<work-no>,<cmd-code>,<cmd-para>*}, be careful, comma(,) in front of *ID* included.
- ID: Tracker ID, default IMEI.
- work-no: working number, hexadecimal string format, cyclic accumulation from 1 to 0xFFFF.
- cmd-code: Command code, or specification of data type.
- cmd-para: parameter or description of *cmd-code*, which is described in the following chapters.
- checksum: checksum of package, 2 bytes hexadecimal string format, XOR of {*<pack-len>,<ID>,<work-no>,<cmd-code>,<cmd-para>*}.
- \r\n: End of package, i.e. <CR><LF>.
- Without specification, multi-byte binary data in *cmd-para* uses big endian format, i.e. Most Significant Byte first.

2 Applied Models

The document describes the format of position/alarm GPRS data, and it is applied for the following models:

- A300
- A500/A500G
- A600
- A700/A700G
- S20
- S30
- S50
- S70
- R18

3 A01/A02 -- GPS Position/Alarm Data Format

$$<pack-len>,<ID>,<work-no>,A01/A02,<alm-code|alm-para>,<date-time>,<fix_flag>,<latitude>,<longitude>,<speed>,<course>,<altitude>,<odometer>,<fuel_consume>,<status>,<input-st>,<output-st>,MCC|MNC|LAC|CI,bat-ad|ext-ad|ad1…adN,<rfid_data>,<digital-sensor>*<checksum>\r\n

Descriptions of position/alarm data:

Example: $$135,866104023192332,29,<A01/A02>,,160606093046,A,22.546430,114.079730,0,186,181,0,415322,0000,02,2,460|0|27B3|EA7,1A0|54D|3|0,940C7E,31.76|30.98*3E\r\n	
Field	pack-len
Description	decimal string format, the field of *pack-len* is {,*<ID>*,*<work-no>*,*A01*,*<alm-code|alm-para>*,*<date-time>*,*<fix_flag>*,*<latitude>*,*<longitude>*,*<speed>*,*<course>*,*<altitude>*,*<odometer>*,*<fuel_consume>*,*<status>*,*<input-st>*,*<output-st>*,*MCC|MNC|LAC|CI*,*bat-ad|ext-ad|ad1…adN*,*<rfid_data>*,*<digital-sensor>*}, be careful, comma(,) in front of *ID* included.
Example	135
Field	ID
Description	Tracker ID, default IMEI, ASCII string
Example	866104023192332
Field	work-no
Description	working number, hexadecimal string format, cyclic accumulation from 1 to 0xFFFF
Example	29, indicates that the value of *work-no* is 0x0029
Field	A01/A02
Description	Data type specification, which is used to define position/alarm package format. 01 A01: No need response from server, tracker deletes package after sending succeed 02 A02: Server response is needed, and then, tracker deletes package
Example	
Field	alm-code|alm-para
Description	Alarm code and alarm parameter, refer to Appendix A; For normal position data, this field is empty.
Example	Empty, the package is a normal position one.
Field	date-time
Description	GMT0 date & time, in format: YYMMDDHHmmss 01 YY: year, value(year – 2000), 2 characters 02 MM: month, value range 1--12, 2 characters 03 DD: day, value range 1--31, 2 characters 04 HH: hour, value range 0--23, 2 characters 05 mm: minute, value range 0-59, 2 characters

<table>
<tr><td></td><td>06 ss: second, value range 0--59, 2 characters</td></tr>
<tr><td>Example</td><td>160606093046, means 2016-6-6 9:30:46</td></tr>
<tr><td>Field</td><td>fix_flag</td></tr>
<tr><td>Description</td><td>GPS Status flag, A--valid, V--invalid</td></tr>
<tr><td>Example</td><td>A, means that GPS signal is valid</td></tr>
<tr><td>Field</td><td>latitude</td></tr>
<tr><td>Description</td><td>Latitude, negative in southern hemisphere, decimal string format</td></tr>
<tr><td>Example</td><td>22.546430</td></tr>
<tr><td>Field</td><td>longitude</td></tr>
<tr><td>Description</td><td>Longitude, negative in western hemisphere, decimal string format</td></tr>
<tr><td>Example</td><td>114.079730</td></tr>
<tr><td>Field</td><td>speed</td></tr>
<tr><td>Description</td><td>Unit km/h, decimal string format</td></tr>
<tr><td>Example</td><td>0, means 0km/h</td></tr>
<tr><td>Field</td><td>course</td></tr>
<tr><td>Description</td><td>Running direction, unit degree, clockwise angle, decimal string format</td></tr>
<tr><td>Example</td><td>186</td></tr>
<tr><td>Field</td><td>altitude</td></tr>
<tr><td>Description</td><td>Altitude, unit meter, decimal string format</td></tr>
<tr><td>Example</td><td>181, means 181m</td></tr>
<tr><td>Field</td><td>odometer</td></tr>
<tr><td>Description</td><td>Unit meter, decimal string format</td></tr>
<tr><td>Example</td><td>0, means odometer 0m</td></tr>
<tr><td>Field</td><td>fuel_consume</td></tr>
<tr><td>Description</td><td>Fuel consumption data, decimal string format; Using B82 command to enable fuel consumption statistics;</td></tr>
<tr><td>Example</td><td>415322, means fuel consumption data is 415322, contact to sales for calculation formula</td></tr>
<tr><td>Field</td><td>status</td></tr>
<tr><td>Description</td><td>Alarm status or vehicle status, hexadecimal string format, as the following table:
<table>
<tr><th>bit</th><th>definition</th><th>description</th></tr>
<tr><td>0</td><td>GPS antenna cut</td><td>Clear when antenna re-connect</td></tr>
<tr><td>1</td><td>Ext-power low voltage</td><td>Clear when voltage normal</td></tr>
<tr><td>2</td><td>Ext-power lost</td><td>Clear when ext-power re-connect</td></tr>
<tr><td>3--7</td><td>CSQ</td><td>GSM signal strength, range [0,31]</td></tr>
<tr><td>8</td><td>Fatigue Driving</td><td>Clear when fatigue relieve</td></tr>
<tr><td>9</td><td>Parking Overtime</td><td>Clear when auto starts running</td></tr>
<tr><td>10</td><td>Idling Running</td><td>Clear when auto starts running or ACC OFF</td></tr>
</table></td></tr>
</table>

	11--27	Reserve	
	28—31	satellite number	satellite number, range [0,15], update from GPS module data
Example	0000, responses to (0000,0000,0000,0000)B, means no status bits		
Field	input-st		
Description	state of input, hexadecimal string format, maximally, tracker supports 6 digital input: bit[0] – input1 status; bit[1] – input2 status; etc.; for each bit, 1- input state is active, 0- input state is inactive		
Example	02, responses to $(0000,0010)_B$, means input2 is active		
Field	output-st		
Description	state of output, hexadecimal string format, maximally, tracker supports 4 digital output bit[0] – output1 status; bit[1] – output2 status; etc.; for each bit, 1- output exports high level, 0- output exports low level		
Example	2, means output2 exports high level		
Field	MCC|MNC|LAC|CI		
Description	Mobil base station information. '|' is used to separate each data. MCC, MNC: decimal string format LAC, CI: hexadecimal string format		
Example	460|0|27B3|EA7: Value of MCC is 460; Value of MNC is 0; Value of LAC is 0x27B3; Value of CI is 0x0EA7;		
Field	bat-ad|ext-ad|ad1…adN		
Description	Sample data of power AD input, hexadecimal string format; Using "|" to separate each data; bat-ad: Voltage of internal battery, unit 0.01V ext-ad: Voltage of ext-power voltage, unit 0.01V ad1 … adN: Sample value of AD1 … ADN input, for different tracker supports variable AD input, there will be variable data here		
Example	1A0|54D|3|0: Voltage of battery is 0x01A0, i.e. 4.16V Voltage of ext-power is 0x054D, i.e. 13.57V; Sample value of AD1 is 0x0003; Sample value of AD2 is 0x0000; Tracker has two AD input.		
Field	rfid_data		

Description	RFID/iButton tag data, hexadecimal string format. Suggest platform convert the hexadecimal string to decimal string format.
Example	940C7E: RFID data is 0x 940C7E
Field	digital-sensor
Description	Digital sensor data, for tracker supports multiple sensors, there will be multiple data here. Default, digital temperature sensor supported
Example	31.76|30.98: Tracker supports two digital temperature sensors 1# sensor temperature: 31.76℃ 2# sensor temperature: 30.98℃
Field	checksum
Description	checksum of package, 2 bytes hexadecimal string format, XOR of {*<pack-len>,<ID>,<work-no>,A01,<alm-code|alm-para>,<date-time>,<fix_flag>,<latitude>,<longitude>,<speed>,<course>,<altitude>,<odometer>,<fuel consume>,<status>,<input-st>,<output-st>,MCC|MNC|LAC|CI,bat-ad|ext-ad|ad1…adN,<rfid data>,<digital-sensor>*}.
Example	3E The XOR checksum is 0x3E
Field	\r\n
Description	End of package, i.e. <CR><LF>
Example	\r\n

4 Server Response to A02

##<pack-len>,<ID>,<work-no>,A02\r\n

Descriptions of position/alarm data:

Example: ##23,866104023192332,29,A02*51\r\n	
Field	pack-len
Description	decimal string format, the field of *pack-len* is {,<ID>,<work-no>,A02}, be careful, comma(,) in front of *ID* included.
Example	23
Field	ID
Description	Tracker ID, default IMEI, ASCII string
Example	866104023192332
Field	work-no
Description	working number, hexadecimal string format, cyclic accumulation from 1 to 0xFFFF *work-no* in response package should be the same as uplink *A02* package; tracker should compare *work-no* in response and uplink package, and only deletes local *A02* package which has the same *work-no*
Example	29, indicates that the value of *work-no* is 0x0029
Field	A02
Description	Data type specification, which defines response command code
Example	
Field	checksum
Description	Checksum of package, 2 bytes hexadecimal string format, XOR of {*<pack-len>,<ID>,<work-no>,A02*}.
Example	51 The XOR checksum is 0x51
Field	\r\n
Description	End of package, i.e. <CR><LF>
Example	\r\n

5 A10 – GPRS Heartbeat Data Format

$$<pack-len>,<ID>,<work-no>,A10,<status>,<bat-ad|ext-ad>*<checksum>\r\n

Descriptions of position/alarm data:

<table>
<tr><td colspan="2">Example:
$$33,866104023192332,36,A10,2,190|46C*6A\r\n</td></tr>
<tr><td>Field</td><td>pack-len</td></tr>
<tr><td>Description</td><td>decimal string format, the field of <u>pack-len</u> is {,<ID>,<work-no>,A10,<status>,<bat-ad|ext-ad>}, be careful, comma(,) in front of <u>ID</u> included.</td></tr>
<tr><td>Example</td><td>33</td></tr>
<tr><td>Field</td><td>ID</td></tr>
<tr><td>Description</td><td>Tracker ID, default IMEI, ASCII string</td></tr>
<tr><td>Example</td><td>866104023192332</td></tr>
<tr><td>Field</td><td>work-no</td></tr>
<tr><td>Description</td><td>working number, hexadecimal string format, cyclic accumulation from 1 to 0xFFFF</td></tr>
<tr><td>Example</td><td>36, indicates that the value of <u>work-no</u> is 0x0036</td></tr>
<tr><td>Field</td><td>A10</td></tr>
<tr><td>Description</td><td>Data type specification, which is used to define GPRS heartbeat package format.</td></tr>
<tr><td>Example</td><td></td></tr>
<tr><td>Field</td><td>status</td></tr>
<tr><td>Description</td><td>Alarm status or vehicle status, hexadecimal string format, as the following table:
<table>
<tr><td>bit</td><td>definition</td><td>description</td></tr>
<tr><td>0</td><td>GPS antenna cut</td><td>Clear when antenna re-connect</td></tr>
<tr><td>1</td><td>Ext-power low voltage</td><td>Clear when voltage normal</td></tr>
<tr><td>2</td><td>Ext-power lost</td><td>Clear when ext-power re-connect</td></tr>
</table></td></tr>
<tr><td>Example</td><td>2, responses to $(0010)_B$, means Ext-power low voltage</td></tr>
<tr><td>Field</td><td>bat-ad|ext-ad</td></tr>
<tr><td>Description</td><td>Voltage of internal battery and external power, using “|” to separate each data;
bat-ad: Voltage of internal battery, unit 0.01V
ext-ad: Voltage of ext-power voltage, unit 0.01V</td></tr>
<tr><td>Example</td><td>190|46C:
Voltage of battery is 0x01A0, i.e. 4.00V
Voltage of ext-power is 0x054D, i.e. 11.32V;</td></tr>
<tr><td>Field</td><td>checksum</td></tr>
<tr><td>Description</td><td>Checksum of package, 2 bytes hexadecimal string format, XOR of</td></tr>
</table>

	{*<pack-len>,<ID>,<work-no>,A10,<status>,<bat-ad|ext-ad>*}.
Example	6A The XOR checksum is 0x6A
Field	\r\n
Description	End of package, i.e. <CR><LF>
Example	\r\n

Appendix A - Alarm Code and Alarm Parameter

The following table describes the relationship of *alm-code* and *alm-para* in GPS Position/Alarm data:

alm-code	alm-para	Description	SMS Head String
1	NULL	Distance tracking	Distance
2	NULL	Input1 active	SOS
3	NULL	Input1 inactive	IN1 Inactive
4	NULL	Input2 active	IN2
5	NULL	Input2 inactive	IN2 Inactive
6	NULL	Input3 active	IN3
7	NULL	Input3 inactive	IN3 Inactive
8	NULL	Input4 active	IN4
9	NULL	Input4 inactive	IN4 Inactive
14	Ext-power voltage, unit V	Ext-power low	Low Ext-Power
15	NULL	Ext-power lost	Ext-Power Cut
16	NULL	Ext-power re-connect	Ext-Power On
17	Battery voltage, unit V	Internal battery low	Low Battery
18	NULL	Speeding alarm	Speeding
19	NULL	Harsh turning	Harsh burning
20	NULL	GPS antenna cut	GPS Antenna Cut
21	NULL	Vibration Alarm	Vibration Alarm
23	NULL	Harsh accelerate	Harsh Accelerate
24	NULL	Harsh braking	Harsh Braking
25	NULL	Enter sleep	Enter Sleep
26	NULL	Exit sleep	Wake Up
27	NULL	Fatigue driving	Fatigue Driving
28	NULL	Fatigue relieve	Fatigue Relieve
29	NULL	Parking overtime	Parking Overtime
30	NULL	Wireless communication jamming	GSM Jamming
32	NULL	GPS jamming	GPS Jamming
33	Hexadecimal character: bit[7:4]: geo-fence type: 0 - Circle fence 1 - Polygon fence bit[3:0]: index of fence	Exit geo-fence	Exit Fence
34	The same as “Exit Fence”	Enter geo-fence	Enter Fence
35	NULL	Idling Alarm	Idling Alarm
37	NULL	Login	Login
38	NULL	Log Out	Log Out

39	NULL	Illegal Login	Illegal Login
40	sn sn: Digital temperature sensor's number, refer to B37	High Temperature	High Temperature
41	sn sn: Digital temperature sensor's number, refer to B37	Low Temperature	Low Temperature
43	com_port com_port: COM port number	COM Port Communication Error	COM Port Error
44	NULL	Fuel Theft Alarm	Fuel Theft
45	NULL	Fuel Filling Alarm	Fuel Filling
46	NULL	Low Fuel Level Alarm	Fuel Level Low
47	NULL	High Fuel Level Alarm	Fuel Level High