

*IME GPRS Communication Protocol  
Between GPS Tracker and Server*

2011-11-28  
V2.3

## I Command Format

Command format of GPRS packet is as follows:

From server to tracking unit:

@@<L><ID (7 bytes)><command (2 bytes)><data><checksum (2 bytes)>\r\n

From tracking unit to server:

\$\$<L><ID (7 bytes)><command (2 bytes)><data><checksum (2 bytes)>\r\n

Note:

Do not input '<' and '>' when writing a command.

All multi-byte data complies with the following sequence:

High byte prior to low byte

Item	Specification
@@	2 bytes. It means the header of packet from server to tracking unit. It is in ASCII code (Hex code: 0x40)
\$\$	2 bytes. It is the header of packet from tracking unit to server, It is in ASCII code (Hex code: 0x24)
L	2 bytes. It means the length of the whole packet including the header and ending character and it is in hex code
ID	7 bytes, the unused byte will be stuffed by 'f' or '0xff'. It is in the format of hex code. For example, if ID is 13612345678, then it will be shown as follows: 0x13, 0x61, 0x23, 0x45, 0x67, 0x8f, 0xff.
Command	2 bytes. The command code is in hex code. Please refer to the command list below.
Data	Min 0 byte, max 100 bytes.
Checksum	2 bytes. It indicates CRC-CCITT(default is 0xffff) checksum of all data (not including CRC itself and the ending character). It is in hex code. For example: 24 24 00 11 13 61 23 45 67 8f ff 50 00 05 d8 0d 0a 0x05d8 = CRC-CCITT (24 24 00 11 13 61 23 45 67 8f ff 50 00)
\r\n	2 bytes. It is the ending character and in hex code (0x0d,0x0a in hex code)

**II Command List**

<b>Command</b>	<b>Definition</b>
0x0001	Heart Beat
0x0002	Get IP and port
0x4000	Server confirms tracker's login
0x5000	Tracker's login
0x4101	Request one single location report
0x4102	Set time interval for continuous tracking
0x4103	Set authorized phone number
0x4104	Reserved
0x4105	Set speed limit for over speed alarm
0x4106	Set movement alert
0x4108	Set extended functions
0x4110	Initialize all parameters except for password, IP/PORT/APN, ID and time interval for continuous tracking.
0x4116	Set GPRS alert for buttons or inputs
0x4130	Set telephone number for wiretapping
0x4132	Set time zone
0x9002	Read time interval of continuous tracking
0x9003	Read authorized phone number
0x9955	Single location report
0x9956	Black box report
0x9957	Send SMS
0x9958	Get plain address
0x9959	Get GPRMC form GSM Cell
0x9960	Report answer phone number and location (For Italy customer only)
0x9961	Set answer phone mode (For Italy customer only)
0x9962	Get battery voltage and signal strength
0x9963	Report call duration (For Italy customer only)
0x9964	Report the content of SMS sent before (For Italy customer only)
0x9965	Set Any Settings
0x9966	Get All Settings
0x9967	Report Any Settings

### **III Command Details**

#### **1. Heart Beat**

Command code: 0x0001

Command from tracking unit to server:

\$\$<L><ID><0xaaaa><checksum>\r\n

Message from server to tracking unit:

@&&<L><ID><0xaaaa ><Flag><checksum> \r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

#### **2. Get IP and port**

Command code: 0x0002

Command from tracking unit to server:

\$\$<L><ID><0x0002><checksum>\r\n

Message from server to tracking unit:

@&&<L><ID><0x0002 ><IP:port><checksum> \r\n

IP: The IP address of server

Port: The port of server

For example:

@&&<L><ID><0x0002 ><118.128.198.38: 8090><checksum> \r\n

#### **3. Tracker's Login**

Command code: 0x5000

Once the IP/PORT/APN is correctly set by SMS and GPRS function is enabled, the tracking unit will apply

for a GPRS connection by sending the login command every 30 seconds to the server until the server

confirms the tracking unit's login.

Tracking unit sends the following command to the server to apply for login and GPRS connection:

\$\$<L><ID><0x5000><checksum><\r\n>

#### **4. Server Confirms Tracker's Login**

Command code: 0x4000

Following command will be sent back to tracking unit after server receives the login command:

@<L><ID><0x4000><Flag><checksum>\r\n

If Flag (1 byte) is

0x00, it means login failed and will try to login again. Or

0x01, it means login succeeded.

### **5. Request One Single Location Report**

Command code: 0x4101

Command from server to tracking unit:

@<L><ID><0x4101><checksum>\r\n

Upon receipt of the above command, the tracking unit will send the following message with its current

location to the server:

\$\$<L><ID><0x9955><data><checksum>\r\n

Please refer to annex 1 for detailed description of 'data'.

### **6. Set Time Interval for Continuous Tracking**

Command code: 0x4102

Command from server to tracking unit:

@<L><ID><0x4102><timer interval in hex code (2 bytes)><checksum>\r\n

Note:

2 bytes of timer interval is in format of 10 seconds and if

=0, it means stop tracking

Max time interval = 65535\*10 seconds

Message from tracking unit to server:

\$\$<L><ID><0x5100><Flag (1byte)> <timer interval (2 bytes)><checksum>\r\n

Note:

If Flag

=0, it means setting failed,

=1, it means setting succeeded

When time interval is set successfully, the tracking unit will send the following message with its current

location to the server at the interval specified:  
 \$\$<L><ID><0x9955><data><checksum>\r\n

Please refer to annex 1 for detailed description of 'data'.

## **7. Set Authorized Phone Number**

Command code: 0x4103

Command from server to tracking unit:

@<L><ID><0x4103><button number><authorized phone number for receiving SMS><authorized phone number for receiving call><checksum>\r\n

Remarks:

Button number – 1 byte and in hex code. It can support up to 4 buttons. Currently only have three speed dial buttons (SOS, button 1, button2 and button3). Therefore for these trackers, if button number,

- = 0x01, to set authorized phone number for SOS button;
- = 0x02, to set authorized phone number for button 1;
- = 0x03, to set authorized phone number for button 2.
- = 0x04, to set authorized phone number for button 3.

Authorized phone number – 16 bytes in ASCII. If the phone number is less than 16 bytes, the blank byte(s) should read as '0x00'. For example:

If the authorized number is 8613612345678, then it should be written as follows:

0x38 0x36 0x31 0x33 0x36 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x00 0x00 0x00

If all 16 bytes data are 0x00, it means the authorized number is invalid.

After the above command is received, the tracking unit will send the following message back to the server:

\$\$<L><ID><0x4103><Flag><checksum>\r\n

If Flag (1byte)

- =0x00, it means setting failed;
- =0x01, it means setting succeeded.

## **8. Set Speed Limit for Over Speed Alarm**

Command code: 0x4105

Command from server to tracking unit:

@<L><ID><0x4105><speed><checksum>\r\n

Speed – 1 byte and in hex code. When the tracker is over this preset speed limit, an alarm message will be sent to the server.

If speed is

0, to cancel over speed alarm;  
 0x01, speed limit is set to 10km/hour;  
 0x02, speed limit is set to 20km/hour;  
 0x03, speed limit is set to 30km/hour;  
 0x04, speed limit is set to 40km/hour;  
 0x05, speed limit is set to 50km/hour;  
 .....  
 Max speed limit is 200km/hour

After the above command is received, the tracking unit will send the following message back to the server:

```
$$<L><ID><0x4105><Flag><checksum>\r\n
```

If Flag (1byte)

=0x00, it means setting of speed limit failed;  
 =0x01, it means setting of speed limit succeeded.

## 9. Set Movement Alert

Command code: 0x4106

Command from server to tracking unit:

```
@&&<L><ID><0x4106><area (1byte in hex code)><checksum>\r\n
```

If area:

=0, to cancel movement alert function  
 =0x01, it is set in a square with current location as center and with side length=2x30m;  
 =0x02, it is set in a square with current location as center and with side length=2x50m;  
 =0x03, it is set in a square with current location as center and with side length=2x100m;  
 =0x04, it is set in a square with current location as center and with side length=2x200m;  
 =0x05, it is set in a square with current location as center and with side length=2x300m;  
 =0x06, it is set in a square with current location as center and with side length=2x500m;  
 =0x07, it is set in a square with current location as center and with side length=2x1000m;  
 =0x08, it is set in a square with current location as center and with side length=2x2000m.

Message replied from tracking unit to server:

```
$$<L><ID><0x4106><Flag><checksum>\r\n
```

If Flag

=0, it means setting failed;

=1, it means setting succeeded.

## 10. Set Extended Functions

Command code: 0x4108

Command from server to tracking unit:

@@<L><ID><0x4108><ABCDEFGG><checksum>\r\n

Remarks for ABCDEFG:

A=0, Turn off the function of sending an SMS location report to the authorized phone number when it makes

a call to the tracking unit.

A=1, Turn on the function of sending an SMS location report to the authorized phone number when it makes

a call to the tracking unit.

B=0, location data of NMEA 0183 GPRMC will be interpreted into normal text for easy reading.

B=1, location data complies with NMEA 0183 GPRMC protocol.

C=0, Turn off the function to automatically hang up an incoming call after 5 rings.

C=1, Turn on the function to automatically hang up an incoming call after 5 rings.

D=0, Turn off the function of sending a message via SMS/GPRS when the tracking unit is turned on.

D=1, Turn on the function of sending a message via SMS/GPRS when the tracking unit is turned on. SMS

will be sent to preset SOS number.

E: reserved. Either 0 or 1 doesn't affect the other settings.

F=0: Turn off the alarm when the tracking unit enters GPS blind area.

F=1: Turn on the alarm when the tracking unit enters GPS blind area. The alarm message is to be sent by

SMS/GPRS.

G=0, 3 LED lights work normally.

G=1, 3 LED lights stop flashing automatically after the tracking unit is rebooted and initialized.

Following message will be sent from tracking unit to the server to confirm above settings:

\$\$<L><ID><0x4108><Flag><checksum>\r\n

If Flag

=0x00, it means settings failed;

=0x01, it means settings succeeded.

**11. Initialize All Parameters Except for Password, IP/PORT/APN, ID and Time Interval for Continuous Tracking**

Command code: 0x4110

Command from server to tracking unit:

@<L><ID><0x4110><checksum>\r\n

Message from tracking unit to server to confirm the settings:

\$\$<L><ID><0x4110><Flag><checksum>\r\n

If Flag

=0x00, it means setting failed;

=0x01, it means setting succeeded.

**12. Set GPRS Alert for Buttons or Inputs**

Command code: 0x4116

Command from server to tracking unit:

@<L><ID><0x4116><ABCD><checksum>\r\n

A=SOS button

B=1 button

C=2 button

D=3 button

When A or B or C or D

=1, to enable alert when button is pressed

=2, to enable alert when button is released

=3, to enable alert when button is pressed or released

Message from tracking unit to server:

\$\$<L><ID><0x4116><Flag><checksum> \r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

**13. Set Telephone Number for Wiretapping (Listen Only)**

Command code: 0x4130

Command from server to tracking unit:

@<L><ID><0x4130><data><checksum>\r\n

Data is the telephone number for wiretapping and it should be numbers or '+' and numbers. Max 16 digits. In ASCII code.

Message from tracking unit to server:

\$\$<L><ID><0x4130><Flag><checksum> \r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

#### **14. Set Time Zone**

Command code: 0x4132

Command from server to tracking unit:

@<L><ID><0x4132><data><checksum>\r\n

Default time is GMT.

Data is to set time difference in minutes to GMT

Data = [0,65535]/[-1,-65535], In ASCII code and in unit of minute

Message from tracking unit to server:

\$\$<L><ID><0x4132><Flag><checksum> \r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

#### **15. Read Preset Time Interval of Continuous Tracking**

Command code: 0x9002

Command from server to tracking unit:

@<L><ID><0x9002><checksum>\r\n

Message from tracking unit to server:

\$\$<L><ID><0x9002><Preset time interval (2 bytes in hex code)><checksum> \r\n

Preset time interval: in format of 10 seconds and if  
=0, it means stop tracking.  
The max timer interval = 65535x10 seconds.

## **16. Read Authorized Phone Number**

Command code: 0x9003

Command from server to tracking unit:

@@<L><ID><0x9003><button number><checksum>\r\n

Button number: 1 byte and in hex code, it should be in the range form 1 to 4. When button name is  
0xff, it is  
to read all authorized numbers.

Message from tracking unit to server:

\$\$<L><ID><0x9003><authorized phone number for receiving SMS><authorized phone number  
for receiving call><checksum>\r\n

Authorized phone number: 16 bytes in ASCII. If the phone number is less than 16 bytes, the blank  
byte(s)  
read as '0x00'.

## **17. Report Black Box Location**

Command code: 0x9956

Command from tracking unit to server:

\$\$<L><ID><0x9956><data;data;data;data;data;data;data;data;...data;data;data;><checksum>\r\n

Remarks:

The max number of data is 40.

Please refer to annex 1 for detailed description of 'data'.

Message from server to tracking unit:

@@<L><ID><0x9956 ><Flag><checksum> \r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

**18. Send SMS**

Command code: 0x9957

Command from server to tracking unit:

@<L><ID><0x9957><phone number><SMS data><checksum>\r\n

Remarks:

phone number-16 bytes in ASCII. If the phone number is less than 16 bytes, the blank byte(s) should read as '0x00'. For example:

If the authorized number is 8613612345678, then it should be written as follows:

0x38 0x36 0x31 0x33 0x36 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x00 0x00 0x00

If all 16 bytes data are 0x00, it means the authorized number is invalid.

SMS data-Max 70\*4 bytes in Unicode. For example:

If the SMS data is “深圳市南山区”, then it should be written as follows:

0xF1 0x6D 0x33 0x57 0x02 0x5E 0x57 0x53 0x71 0x5C 0x3A 0x53 0x00 0x00

This field must use 0x00 0x00 for end.

After the above command is received, the tracking unit will send the following message back to the server:

\$\$<L><ID><0x9957><Flag><checksum>\r\n

If Flag (1byte)

=0x00, it means setting failed;

=0x01, it means setting succeeded.

**19. Get plain address**

Command code: 0x9958

Command from tracking unit to server:

\$\$<L><ID><0x9958><phone number><GPS/Cell data><checksum>\r\n

Remarks:

phone number-16 bytes in ASCII. If the phone number is less than 16 bytes, the blank byte(s) should read as '0x00'. For example:

If the authorized number is 8613612345678, then it should be written as follows:

0x38 0x36 0x31 0x33 0x36 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x00 0x00 0x00

If all 16 bytes data are 0x00, it means the authorized number is invalid.

Please refer to annex 2 for detailed description of 'GPS/Cell data'.

Message from server to tracking unit:

@<L><ID><0x9958><phone number><SMS data><checksum>\r\n

SMS data-Max 70\*4 bytes in Unicode. For example:

If the SMS data is “深圳市南山区”, then it should be written as follows:

0xF1 0x6D 0x33 0x57 0x02 0x5E 0x57 0x53 0x71 0x5C 0x3A 0x53 0x00 0x00

This field must use 0x00 0x00 for end.

## **20. Get GPRMC from GSM Cell**

Command code: 0x9959

Command from tracking unit to server:

\$\$<L><ID><0x9959><GPS and Cell data><checksum>\r\n

Message from server to tracking unit:

@@<L><ID><0x9959><GPS data><checksum> \r\n

Please refer to annex 2 for detailed description of ‘GPS/Cell data’.

## **21. Report Answer Phone Number And Location (For Italy customer only)**

Command code: 0x9960

Command from tracking unit to server:

\$\$<L><ID><0x9960><phone number><GPS/Cell data><checksum>\r\n

Remarks:

phone number-16 bytes in ASCII. If the phone number is less than 16 bytes, the blank byte(s) should read as ‘0x00’. For example:

If the authorized number is 8613612345678, then it should be written as follows:

0x38 0x36 0x31 0x33 0x36 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x00 0x00 0x00

If all 16 bytes data are 0x00, it means the authorized number is invalid.

Please refer to annex 2 for detailed description of ‘GPS/Cell data’.

Message from server to tracking unit:

@@<L><ID><0x9960><Flag><checksum>\r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

## **22. Set Answer Phone Mode (For Italy customer only)**

Command code: 0x9961

Command from server to tracking unit:

@<L><ID><0x9961><mode><checksum>\r\n

Remarks:

mode – 1 byte and in hex code.

If mode is

0x00, the device cuts off the line and send the GPRS message to the server;

0x01, the device rings like a normal phone and does NOT send anything to the server;

0x02, the device rings and works like a normal phone and DOES send the position to the server;

After the above command is received, the tracking unit will send the following message back to the server:

\$\$<L><ID><0x9961><Flag><checksum>\r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

### **23. Get battery voltage and signal strength**

Command code: 0x9962

Command from server to tracking unit:

@<L><ID><0x9962><checksum>\r\n

After the above command is received, the tracking unit will send the following message back to the server:

\$\$<L><ID><0x9962><battery><signal><checksum>\r\n

Remarks:

Battery – Percentage of the battery voltage, 1 byte in hex code.

Signal – Percentage of the signal strength, 1 byte in hex code.

### **24. Report call duration (For Italy customer only)**

Command code: 0x9963

Command from tracking unit to server:

\$\$<L><ID><0x9963><outgoing call XX min made><checksum>\r\n

Remarks:

XX-Call duration in unit of minutes. If less than one minute, then the valuation in accordance with one minute.

For example:

<b>Really call duration</b>	<b>Report call duration</b>
-----------------------------	-----------------------------

1...60 sec	1 min
61...120 sec	2 min
121...180 sec	3 min
181...240 sec	4 min
241...300 sec	5 min

Message from server to tracking unit:

@@<L><ID><0x9963><Flag><checksum>\r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

## **25. Report the content of SMS sent before (For Italy customer only)**

Command code: 0x9964

Command from tracking unit to server:

\$\$<L><ID><0x9964><SMS has been sent XXXX><checksum>\r\n

Remarks:

XXXX-The content of SMS sent before.

For example:

SMS has been sent W01,600,ITALY.09A.W09.28.MP.V2.81.352530005090846 OK

Message from server to tracking unit:

@@<L><ID><0x9964><Flag><checksum>\r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

## **26. Set Any Settings**

Command code: 0x9965

Command from server to tracking unit:

@@<L><ID><0x9965><Settings><checksum>\r\n

Remarks:

Settings – Set any settings in one time.

For example:

“key1num”:”2313213”,”key2num”:”555555”,”key3num”:”321321344”,”sosnum”:”321321321”

Please refer to annex 2 for detailed description of ‘any settings’.

After the above command is received, the tracking unit will send the following message back to

the server:

```
$$<L><ID><0x9965><Flag><checksum>\r\n
```

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

## **27. Get All Settings**

Command code: 0x9966

Command from server to tracking unit:

```
@<L><ID><0x9966><checksum>\r\n
```

After the above command is received, the tracking unit will send the following message back to the server:

```
$$<L><ID><0x9966><Settings><checksum>\r\n
```

Remarks:

Settings – Get any settings in one time.

For example:

```
“key1num”:”2313213”,”key2num”:”5555555”,”key3num”:”321321344”,”sosnum”:”321321321”
```

Please refer to annex 2 for detailed description of ‘any settings’.

## **28. Report Any Settings**

Command code: 0x9967

Command from tracking unit to server:

```
$$<L><ID><0x9967><Settings><checksum>\r\n
```

Remarks:

Settings – Report any settings in one time.

For example:

```
“key1num”:”2313213”,”key2num”:”5555555”,”key3num”:”321321344”,”sosnum”:”321321321”
```

Please refer to annex 2 for detailed description of ‘any settings’.

Message from server to tracking unit:

```
@<L><ID><0x9967><Flag><checksum>\r\n
```

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

**Annex 1: Description of data****1, Data**

Data consists of: GPRMC | HDOP | Altitude | Cell | State | Alarm | Battery | Signal

For example:

100008.000,A,2232.4679,N,11356.7805,E,0.204,89.22,210911,,|7.49|152.6|3571,9763,00,460|0000|00|100|100

(9) GPRMC includes:

hhmmss.dd,S,xxmm.dddd,<N|S>,yyymm.dddd,<E|W>,s.s,h.h,ddmmyy

For example:

134829.486,A,1126.6639,S,11133.3299,W,58.31,309.62,110200

Details:

Parameter	Description	Example
hhmmss.dd	UTC time hh = hours; mm = minutes; ss = seconds; dd = decimal part of seconds	13:48:29.486
S	Status indicator, A = valid, V = invalid	A=Valid
xxmm.dddd	Latitude xx = degrees; mm = minutes; dddd = decimal part of minutes	11 deg. 26.6639 min
<N S>	Either character N or character S N = North, S = South	S = South
yyymm.dddd	Longitude yyy = degrees; mm = minutes; dddd = decimal part of minutes	111 deg. 33.3299 min
<E W>	Either character E or character W E = East, W = West	W=West
s.s	Speed, knots	58.31 Knots
h.h	Heading	309.62 deg
ddmmyy	Date dd = date; mm = month' yy = year	11 <sup>th</sup> , Aug. 2000

(2) | is list separator in ASCII (=0x7c)

(3) HDOP, in ASCII code, 0.5-99.9. HDOP is blank when the tracking unit has no GPS fix.

(4) Altitude, in algorism.

(5) Cell, includes:

cid,lac,mnc,mcc

For example:

4272,10147,00,460

Details:

Parameter	Description	Example
cid	Cell ID	4272
lac	LAC	10147
mnc	MNC	00
mcc	MCC	460

(6) State: Status of input and output

Bit0~Bit7: Reserved and default as '0'

Bit8: Status of Input1

If Bit8=0: Input1 is inactive; Bit8=1: Input1 is active (SOS button is pressed)

Bit9: Status of Input2

If Bit9=0: Input2 is inactive; Bit9=1: Input2 is active (Button 1 is pressed)

Bit10: Status of Input3

If Bit10=0: Input3 is inactive; Bit10=1: Input3 is active (Button 2 is pressed)

Bit11: Status of Input4

If Bit11=0: Input4 is inactive; Bit11=1: Input4 is active (Button 3 is pressed)

Bit12: Status of Input5

If Bit12=0: Input5 is inactive; Bit12=1: Input5 is active (Answer button is pressed)

Bit13: Status of Input6

If Bit13=0: Input6 is inactive; Bit13=1: Input6 is active ("Hang up/Power key" button is pressed)

(7) Alarm: Alarm command

Alarm code	Definition
=0x01	SOS button is pressed
=0x02	2 <sup>nd</sup> button is pressed
=0x03	3 <sup>rd</sup> button is pressed
=0x04	First button is pressed
=0x05	Reserved
=0x10	Low battery alarm
=0x11	Over speed alarm

---

=0x12	Movement alarm
=0x13	Reserved
=0x14	Alarm of tracking unit being turned on
=0x15	Alarm of tracking unit entering blind area
=0x16	Alarm of tracking unit leaving blind area
=0x31	SOS button is released
=0x32	2 <sup>nd</sup> button is released
=0x33	3 <sup>rd</sup> button is released
=0x34	First button is released
=0x35	Reserved
=0x40	Vibrate alarm

(8) Battery: Percentage of the battery voltage

(9) Signal: Percentage of the signal strength

**Annex 2: Description of any settings**

Parameter	Description	Example
sosnum	Bind SOS key number (both SMS and calling)	"sosnum": "13500000000"
key1num	Bind the 1 <sup>st</sup> key number (only calling)	"key1num": "13511111111"
key2num	Bind the 2 <sup>nd</sup> key number (only calling)	"key2num": "13522222222"
key3num	Bind the 3 <sup>rd</sup> key number (only calling)	"key3num": "13533333333"
report	Set GPRS report interval (second)	"report": "3"
ovrspd	Set over speed	"ovrspd": "100"
ovrnum	Set the phone number of over speed alarm	"ovrnum": "13812345678"
ovrsms	Set the SMS of over speed alarm (in UTF-8)	"ovrsms": "超速了"
start	Set start work time (hour)	"start": "0"
stop	Set stop work time (hour)	"stop": "24"
smsintval	Set SMS report interval (minute)	"smsintval": "1"
dt	Set date time	"dt": "2011-9-20,20:19:00"
tzone	Set time zone (minute)	"tzone": "480"
agps	Set AGPS IP and port	"agps": "209.85.173.192: 7276"
tapnum	Set wiretap number	"tapnum": "13566666666"
ciaddr	Report location via SMS after incoming call (true/false)	"ciaddr": "true"
exc	Automatically hang up an incoming call (true/false)	"exc": "true"
exd	Sending an SMS to SOS number when tracker is turned on (true/false)	"exd": "true"
exf	GPS blind area (true/false)	"exf": "true"
exg	LED lights work (true/false)	"exg": "true"
psmode	Power save mode (true/false)	"psmode": "true"
firew	Firewall of incoming call (true/false)	"firew": "true"
vibc	Set vibrate alarm (true/false)	"vibc": "true"
vibs	Set vibrate	"vibs": "震动报警"