

***GPRS Communication Protocol
Between
GPS Tracker and Server***

2015/4

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.

If all 7 bytes are 0xff, it is a broadcasting command. ID is in hex code

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

Command Definition

- 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
- 0x4105 Set speed limit for over speed alarm
- 0x4106 Set movement alert
- 0x4107 Set Geo-fence
- 0x4127 Set Geo-fence
- 0x4108 Set extended functions
- 0x4110 Initialize all parameters except for password, IP/PORT/APN, ID and time interval for continuous tracking.
- 0x4111 Tracker ask for the actual address
- 0x4114 Output control (safe - 10Km/h)
- 0x5114 Output control (safe - 20Km/h)
- 0x4115 Output control (general)
- 0x4116 Set GPRS alert for buttons or inputs
- 0x4126 Set power saving when tracker is immobile
- 0x4130 Set telephone number for wiretapping
- 0x4132 Set time zone
- 0x4136 Set angle interval for continuous tracking
- 0x4145 Set odometer interval for continuous tracking
- 0x4146 Clear odometer
- 0x4150 Set talking mode (Only For VT900)
- 0x4151 Request one single photo report
- 0x4208 Set extended functions of take photo
- 0x4333 Clear history records in flash (Only For VT310N,VT200,VT900)
- 0x9001 Read version / Tracker reply version
- 0x9002 Read time interval of continuous tracking
- 0x5100 Tracker reply time interval
- 0x9003 Read authorized phone number / Tracker reply authorized phone number
- 0x9005 Read odometer interval of continuous tracking
- 0x5105 Tracker reply odometer interval
- 0x9006 Read angle interval of continuous tracking
- 0x5106 Tracker reply angle interval
- 0x5001 Tracker's heart beat

0x9955 Single location report
0x9988 Tracker reply single photo segment
0x9999 Alarm command
0x4060 Set 500 RFID numbers
0x9060 Read 500 RFID numbers

III Command Details

1. 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 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>
```

2. 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.

3. 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'.

4. 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'.

5. 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 3 buttons.

Currently our GT30 and GT60 only have three speed dial buttons (SOS, Call B and Call C). Therefore for these trackers, if button number,

= 0x01, to set authorized phone number for SOS button;

= 0x02, to set authorized phone number for Call B button;

= 0x03, to set authorized phone number for Call C button.

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.

6. 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.

7. 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.

8. Set Geo-fence

Command code: **0x4107, 0x4127**

Command from server to tracking unit:

@@<L><ID><0x4107 or 0x4127><coordinates><checksum>\r\n

Remarks:

0x4107 is the command for exiting the geo-fencing area.

0x4127 is the command for entering the geo-fencing area

Coordinates includes:

Bottom left longitude, bottom left latitude, top right longitude, top right latitude

The value of bottom left longitude and latitude should be less than that of top

right. And all longitudes and latitudes should be in ASCII format as follows:-

Longitude: DDDFF.FFFF,E/W, 4 places of decimal. '0' is needed to be stuffed if no value available.

Latitude: DDFF.FFFF,N/S, 4 places of decimal. '0' is needed to be stffued if no value available.

Example:

11404.0000,E,2232.0010,N,11505.1234,E,2333.5678,N

Command: @@<L><ID><0x4107 or 0x4127><checksum>\r\n to cancel geo-fencing.

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

\$\$<L><ID><0x4107 or 0x4127><Flag><checksum>\r\n

If Flag

=0, it means setting failed;

=1, it means setting succeeded.

Note: You can only set either movement alert or geo-fencing.

9. Set Extended Functions

Command code: **0x4108**

Command from server to tracking unit:

@@<L><ID><0x4108><ABCDEFGHJIJ>###<checksum>\r\n

Remarks for ABCDEFGHIJ:

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.

I=0, Turn off the function of sending a message via SMS/GPRS when the external power is cut.

I=1, Turn on the function of sending a message via SMS/GPRS when the external power is cut.

SMS will be sent to preset SOS number.

C,D,E,F,G,H,J: reserved. Either 0 or 1 doesn't affect the other settings.

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.

10. 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.

11. Output Control (safe) - Advised Caution in Using this Function

Command code: **0x4114 or 0x5114**

Command from server to tracking unit:

`@@<L><ID><0x4114 or 0x5114><ABCDE><checksum>\r\n`

A=0, to close output (OUT1);

A=1, to open output (OUT1);

A=2, to remain previous status.

B=0, to close output (OUT2);

B=1, to open output (OUT2);

B=2, to remain previous status.

C=0, to close output (OUT3);

C=1, to open output (OUT3);

C=2, to remain previous status.

D=0, to close output (OUT4);

D=1, to open output (OUT4);

D=2, to remain previous status.

D=0, to close output (OUT5);

D=1, to open output (OUT5);

D=2, to remain previous status.

Message from tracking unit to server:

`$$<L><ID><0x4114 or 0x5114><Flag><checksum>\r\n`

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

Note: this function is achievable when the speed is below 10km/h (0x4114) or 20km/h (0x5114) and GPS is available.

12. Output Control (general) - Advised Caution in Using this Function

Command code: **0x4115**

Command from server to tracking unit:

`@@<L><ID><0x4115><ABCDE><checksum>\r\n`

A=0, to close output (OUT1);
A=1, to open output (OUT1);
A=2, to remain previous status.
B=0, to close output (OUT2);
B=1, to open output (OUT2);
B=2, to remain previous status.
C=0, to close output (OUT3);
C=1, to open output (OUT3);
C=2, to remain previous status.
D=0, to close output (OUT4);
D=1, to open output (OUT4);
D=2, to remain previous status.
D=0, to close output (OUT5);
D=1, to open output (OUT5);
D=2, to remain previous status.

Message from tracking unit to server:

\$\$<L><ID><0x4115><Flag><checksum>\r\n

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

13. Set GPRS Alert for Buttons or Inputs

Command code: **0x4116**

Command from server to tracking unit:

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

A=IN1

B=IN2

C=IN3

D=IN4

E=IN5

When A or B or C or D or E

=1, to enable alert when input port is close

=2, to enable alert when input port is open

=3, to enable alert when input port is close or open

IN1, IN2, IN3 are detected as low level input. IN4 and IN5 are detected as high level input.

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.

14. Set Power Saving When Tracker is Inactive

Command code: **0x4126**

Command from server to tracking unit:

`@@<L><ID><0x4126><data><checksum>\r\n`

Data: In ASCII code and in unit of minute

If data = 00, to close this function;

= [01, 99], to set this function.

Example:

If data = 10, the tracker will enter power saving mode in ten minutes after it is immobile.

In power saving mode, GPS stops working. GSM enters standby mode and stop sending out message until it is activated or the tracker moves again.

Message from tracking unit to server:

`$$<L><ID><0x4126><Flag><checksum> \r\n`

If Flag

=0x00, it means operation failed;

=0x01, it means operation succeeded.

15. 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.

16. 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, 1440] / [-1,-1440], 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.

17. Set Angle Interval for Continuous Tracking

Command code: **0x4136**

Command from server to tracking unit:

@@<L><ID><0x4136><angle interval in hex code (2 bytes)><checksum>\r\n

Note:

2 bytes of angle interval is in format of °C and if =0, it means stop tracking

Max angle interval = 359 °C

Message from tracking unit to server:

\$\$<L><ID><0x5106><Flag (1byte)> <angler interval (2 bytes)><checksum>\r\n

Note:

If Flag

=0, it means setting failed,

=1, it means setting succeeded

When angle 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'.

18. Set Odometer Interval for Continuous Tracking

Command code: **0x4145**

Command from server to tracking unit:

@@<L><ID><0x4145><odometer interval in hex code (2 bytes)><checksum>\r\n

Note:

2 bytes of odometer interval is in format of meter and if =0, it means stop tracking

Max odometer interval = 65535 meters

Message from tracking unit to server:

\$\$<L><ID><0x5105><Flag (1byte)> <odometer interval (2 bytes)><checksum>\r\n

Note:

If Flag

=0, it means setting failed,

=1, it means setting succeeded

When odometer 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'.

19. Clear Odometer

Command code: **0x4146**

Command from server to tracking unit:

@@<L><ID><0x4146><checksum>\r\n

Message from tracking unit to server:

\$\$<L><ID><0x4146><Flag (1byte)> <checksum>\r\n

Note:

If Flag

=0, it means setting failed,

=1, it means setting succeeded

20. Clear History Records in Flash

Command code: **0x4333**

Command from server to tracking unit:

`@@<L><ID><0x4333><checksum>\r\n`

Message from tracking unit to server:

`$$<L><ID><0x4333><Flag (1byte)> <checksum>\r\n`

Note:

If Flag

=0, it means setting failed,

=1, it means setting succeeded

21. tracker request to get the actual address

Command from tracking unit to server:

`$$<L (2bytes)><ID (7bytes)><0x4111><phone number | Latitude, Longitude><checksum (2 bytes)>\r\n`

Note:

Phone number, Latitude, Longitude: in ASCII format.

Message from server to tracking unit:

`@@<L (2 bytes)><ID (7 bytes)><0x4111><phone number | Position name><checksum (2 bytes)>\r\n`

Note:

Phone number, Position name: in ASCII format.

Message from tracking unit to server:

`$$<L><ID><0x4112><0x01> <checksum>\r\n`

22. Set Talking Mode

Command code: **0x4150**

Command from server to tracking unit:

`@@<L><ID><0x4150><AB><checksum>\r\n`

Note:

A=1,enable pick up the telephone automatic

A=0 disable pick up the telephone automatic, default=0

B=1,turn on the speaker,

B=0 turn off the speaker, default=0

23. Tracker's Heart beat

Command code: **0x5001**

Tracking unit sends the following command to the server:

`$$<L><ID><0x5001><checksum><\r\n>`

24. Read Version of tracking unit

Command code: **0x9001**

Command from server to tracking unit:

`@@<L><ID><0x9001><checksum>\r\n`

Message from tracking unit to server:

`$$<L><ID><0x9001><Version of tracking unit><checksum> \r\n`

Version of tracking unit: in ASCII format

25. 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><0x5100><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.

26. Read Preset Odometer Interval of Continuous Tracking

Command code: **0x9005**

Command from server to tracking unit:

`@@<L><ID><0x9005><checksum>\r\n`

Message from tracking unit to server:

`$$<L><ID><0x5105><Preset odometer interval (2 bytes in hex code)><checksum> \r\n`

Preset odometer interval: in format of meter and if =0, it means stop tracking.

The max angle interval = 65535 meters.

27. Read Preset Angle Interval of Continuous Tracking

Command code: **0x9006**

Command from server to tracking unit:

`@@<L><ID><0x9006><checksum>\r\n`

Message from tracking unit to server:

`$$<L><ID><0x5106><Preset angle interval (2 bytes in hex code)><checksum> \r\n`

Preset angle interval: in format of °C and if =0, it means stop tracking.

The max angle interval = 359 °C.

28. 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 3.

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'.

29. Alarm Command

Command: **0x9999**

Message from tracking unit to server:

`$$<L><ID><0x9999><Alarm><data><checksum>\r\n`

Alarm: 1 byte in hex code and details as follows:

Alarm: 1 byte in hex code and details as follows:

Alarm code Definition

- =0x01 SOS button is pressed / Input port 1 close
- =0x02 2nd button (Call B) is pressed / Input port 2 close
- =0x03 3rd button (Call C) is pressed / Input port 3 close
- =0x04 Input port 4 close
- =0x05 Input port 5 close
- =0x11 Over speed alarm
- =0x12 movement alarm
- =0x13 Geo fence alarm
- =0x31 SOS button is released/Input port 1 open
- =0x32 Call B button is released/Input port 2 open
- =0x33 Call C button is released/Input port 3 open
- =0x34 Input port 4 open
- =0x35 Input port 5 open
- =0x50 external power cut
- =0x72harsh braking
- =0x73 harsh acceleration
- =0x75 camera error
- =0x74 lose oil
- =0x76 add oil
- =0x14 impact alert

30. Request One Single Photo Report

Command code: **0x4151**

Command from server to tracking unit:

`@@<L><ID><0x4151><checksum>\r\n`

Upon receipt of the above command, the tracking unit will take a photo and send the photo message to the server:

\$\$<L><ID><0x9988><index of photo><length><packages><index of package><data of photo><0xffff>\r\n

Take Photo

1. Upload a request to send photo, command code: 0x0800

\$\$<L><ID><0X0800><photo ID>FFFF\r\n

2. Server responds to the photo request, command code:0x8801

@@<L><ID><0X8801><photo ID>FFFF\r\n

3. Upload photo packets, command code:0x9988

\$\$<L><ID><0X9988><photo ID><size of photo><sum of packets><packet ID><data>FFFF\r\n

4. Confirmation of photo uploaded, command code: 0x0f80

\$\$<L><ID><0X0f80><photo ID><size of photo><sum of packets><data>FFFF\r\n

5. Server distributes a request of resending photo packets, command code:0x8800

@@<L><ID><0x8800><photo ID><sum of resent packets><resent packet ID>FFFF\r\n

Note:

<photo ID>: 1 byte in hex code

<size of photo>: 2 bytes in hex code

<sum of resent packets>: 1 byte in hex code, 0 or 1. If there is no need to resend, it is 0. Or else, it is 1.

<resent packet ID>: 1 byte in hex code, represent the ID of the resent packet. If the sum of resent packet is 0, then the segment should be <sum of packets>.

31. Tracker reply single photo segment

Command code: **0x9988**

Command from tracking unit to server:

\$\$<L><ID><0x9988><index of photo><length><segments><index of segment><data of photo segment><0xffff>\r\n

Note:

Index of photo: one byte in hex

Length: two bytes in hex, the size of the photo. For example, the photo size is 6100 bytes, then the length will be fixed 0x17D4

Segments: one byte in hex, the photo will be divided into several segments, each segment is 1000 bytes. For example, the photo size is 6100 bytes, then it will be divided into 7 segments, and the segments will be fixed 0x07.

Index of segment: one byte in hex. For example, there are 7 segments of this photo, and this segment is the third, then the index of segment will be fixed 0x03.

Data of photo segment: 1000 bytes or less than 1000 bytes in hex. If this is the last segment of this photo, then the data of this segment will maybe less than 1000 bytes. If this is not the last segment of this photo, then the data of this

segment will be 1000 bytes.

32. Set Extended Functions of Photo

Command code: **0x4208**

Command from server to tracking unit:

`@@<L><ID><0x4208><ABCDEFGHJIJ>###<checksum>\r\n`

Remarks for ABCDEFGHIJ:

A=0, Turn off the function of taking photo when IN1 input is pressed.

A=1, Turn on the function of taking photo when IN1 input is pressed.

B=0, Turn off the function of taking photo when IN2 input is pressed.

B=1, Turn on the function of taking photo when IN2 input is pressed.

C=0, Turn off the function of taking photo when IN3 input is pressed.

C=1, Turn on the function of taking photo when IN3 input is pressed.

D=0, Turn off the function of taking photo when IN4 input is pressed.

D=1, Turn on the function of taking photo when IN4 input is pressed.

E=0, Turn off the function of taking photo when IN5 input is pressed.

E=1, Turn on the function of taking photo when IN5 input is pressed.

F=0, Turn off the function of taking photo when over speed alert.

F=1, Turn on the function of taking photo when over speed alert.

G=0, Turn off the function of taking photo when movement alert.

G=1, Turn on the function of taking photo when movement alert.

H=0, Turn off the function of taking photo when geo fence alert.

H=1, Turn on the function of taking photo when geo fence alert.

I=0, Turn off the function of taking photo when cut external power alert.

I=1, Turn on the function of taking photo when cut external power alert.

J=0, Turn off the function of taking photo when lose oil alert.

J=1, Turn on the function of taking photo when lose oil alert.

SMS will be sent to preset SOS number.

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

`$$<L><ID><0x4208><Flag><checksum>\r\n`

If Flag

=0x00, it means settings failed;

=0x01, it means settings succeeded.

33. Set 500 legal RFID numbers

Command code: **0x4060**

Command from server to tracking unit:

`@@<L><ID><0x4060><500 RFID number><checksum>\r\n`

`<L>=0x03F9=1017`

`<500 RFID number>`: 1000 bytes of 500 RFID number .Every RFID number has two bytes in hex. For example, RFID number is 00258, then the two bytes are 0x0102.If the RFID number is null, then fix 0x0000.

34. Read 500 legal RFID numbers

Command code: **0x9060**

Command from server to tracking unit:

`@@<L><ID><0x9060><checksum>\r\n`

Upon receipt of the above command, the tracking unit will send the 500 RFID numbers to the server:

`$$<L><ID><0x9060><500 RFID number><checksum>\r\n`

`<L>=0x03F9=1017`

`<500 RFID number>`: 1000 bytes of 500 RFID number .Every RFID number has two bytes in hex. For example, RFID number is 00258, then the two bytes are 0x0102.If the RFID number is null, then fix 0x0000.

35. Teamperature

Command code: **0x9944**

`$$<L><ID><0x9944>+temperature senior 8 bits+ temperature low 8 bits+ <checksum>\r\n`

36. OBD

Command code: **0x9922**

`$$ <L><ID><9922><ODB Data><checksum>\r\n`

Note:

BD Data is composed of the data read from OBD2 connector and separated by comma. The sequence of OBD Data for our OBD2 connector is:

Accumulated mileage,Instantaneous Fuel,Average Fuel,Running Time,Speed,Live Load,Temperature,Throttle Open,Engine Speed,Voltage,Error Code

If there are more than 2 error code, separate them by comma.

If no error code, it shows "NO DTC"

Sample for OBD Data:

124,96.6,102.5,52,224,86,35,85,3300,16.3,P3002,P3102

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

Annex 1: Description of data

Data consists of: *GPRMC* | *HDOP* | *Altitude* | *State* | *AD1,AD2* | *Odometer* | *RFID*

(1) 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

11th, 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) State: Status of input and output:

Bit0: Status of Out1

If Bit0=0: Out1 is closed; Bit0=1:Out1 is open.

Bit1: Status of Out2

If Bit1=0: Out2 is closed; Bit1=1:Out2 is open.

Bit2: Status of Out3

If Bit2=0: Out3 is closed; Bit2=1:Out3 is open.

Bit3: Status of Out4

If Bit3=0: Out4 is closed; Bit3=1:Out4 is open.

Bit4: Status of Out5

If Bit4=0: Out5 is closed; Bit4=1:Out5 is open.

Bit5~7: Reserved.

Bit8: Status of Input1

If Bit8=0: Input1 is invalid; Bit8=1: Input1 is valid

Bit9: Status of Input2

If Bit9=0: Input2 is invalid; Bit9=1: Input2 is valid

Bit10: Status of Input3

If Bit10=0: Input3 is invalid; Bit10=1: Input3 is valid

Bit11: Status of Input4

If Bit11=0: Input4 is invalid; Bit11=1: Input4 is valid

Bit12: Status of Input5

If Bit12=0: Input5 is invalid; Bit12=1: Input5 is valid

Bit13~Bit15: Reserved and default as '0'

(6) AD1, AD2: 10 bit analog input for VT310 and VT900, 0x0000...0x03ff in HEX, separated by ',' (comma).

(7) Odometer: 9 bytes in ASCII.

(8)RFID: 5 bytes in ASCII