# OBD Terminal iCloud System Protocol

Protocol Specifications

Revision V1.1.4

# Amendment Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Versopm | Description | Person |
| 2013.10.06 | 1.0.0 | First version |  |
| 2014.9.24 | 1.0.1 | Delete logout command |  |
| 2014.12.12 | 1.0.2 | Simplify header option |  |
| 2015.01.27 | 1.0.3 | Delete some user define items |  |
| 2015.03.17 | 1.0.4 | Add synchronize parameter status-bit to Login command  Add firmware version to Update command |  |
| 2015.05.04 | 1.0.5 | Add travel segment TLV to Position message  Add password parameter to login request  TLV:0x0004 modify topassword, sinchronize parameter identification modify to 0x0005 |  |
| 2015.06.10 | 1.0.6 | Add protocol flow chart. |  |
| 2015.06.23 | 1.0.7 | Add Remote Control Message |  |
| 2015.08.04 | 1.0.8 | Add Protocol Structure  Add SLIP |  |
| 2015.09.02 | 1.0.9 | Add parameter set example  Add break data mode sync |  |
| 2015.10.28 | 1.1.0 | Add Coolant Temperature Alarm gate  Angle compensation interval  Add Coolant Temperature Alarm type |  |
| 2016.01.13 | 1.1.1 | Modify update firmware |  |
| 2016.01.18 | 1.1.2 | Modify the fault type error |  |
| 2016.04.20 | 1.1.3 | Add Fault Report example |  |
| 2016.05.26 | 1.1.4 | Add Remote control Message |  |

Catalog

[OBD Terminal iCloud System Protocol 1](#_Toc422837865)

[Amendment Record 1](#_Toc422837866)

[Catalog 3](#_Toc422837867)

[1. Overview 1](#_Toc422837868)

[2. Glossary and Convention 1](#_Toc422837869)

[2.1. Glossary 1](#_Toc422837872)

[2.2. Convention 1](#_Toc422837873)

[3. Protocol Structure 2](#_Toc422837874)

[3.1. Protocol Overall Structure 2](#_Toc422837879)

[3.2. Protocol Packet Header 2](#_Toc422837880)

[3.3. Protocol Packet Body 3](#_Toc422837881)

[3.4. Transport Carrier Of Protocol Packet 3](#_Toc422837882)

[4. Protocol Command Message Definition 4](#_Toc422837883)

[4.1. Command Message Parameter Unit Structure 4](#_Toc422837884)

[4.2. Command ID Definition 4](#_Toc422837885)

[4.3. Position Message 5](#_Toc422837886)

[4.4. Login Message 6](#_Toc422837887)

[4.5. Set Parameter Message 8](#_Toc422837888)

[4.6. Remote Update 9](#_Toc422837889)

[4.7. Update Data Sub-packet Download Message 10](#_Toc422837890)

[4.8. Fault Report 11](#_Toc422837891)

[4.9. Handshake Message 11](#_Toc422837892)

[4.10. Remote control Message 12](#_Toc422837893)

[**5.Protocol Example** 13](#_Toc422837894)

[5.1. LOGIN Message 13](#_Toc422837895)

[5.2. Position Message 14](#_Toc422837896)

[5.3. Flow Chart 17](#_Toc422837897)

[**Appendix I SLIP Protocol Frame Format** 19](#_Toc422837898)

# Overview

To take use of resources & advantages of our experiences of integration with large telecom carrier’s software platform, this protocol has some excellent characteristics:

* New protocol is simplified & easier to apply.
* New protocol is easy & flexible to extend new features.

# Glossary and Convention

This protocol data is transmitted in network byte order (high bit first).



## Glossary

* CASP Application System Protocol
* TLV Type Length Value CASP protocol parameter unit
* APN Access Point Name
* GPRS General Packet Radio Service
* IMEI International Mobile Equipment Identify
* SLIP SLIP (Serial Line Internet Protocol）

## Convention

Data type and length [comparison](app:ds:comparison) [table](app:ds:table):

|  |  |  |
| --- | --- | --- |
| Type | Length(byte) | Explanation |
| BYTE | 1 | 1-byte |
| SHORT | 2 | 2-byte short integer |
| INTGER | 4 | 4-byte integer |
| LONG | 8 | 8-byte integer |
| CSTRING | Variable length | ASCII string |
| BIN | Variable length | Binary data |
| FLOAT | 4 | 4-byte single-precision floating point number |
| DOUBLE | 8 | 8-byte double-precision floating point number |

# Protocol Structure



## Protocol Overall Structure

The protocol consists of packet header and body. The packet header length is 12 bytes.



## Protocol Packet Header

Flag

Reserved

Packet length

Version

Packet command

CRC check

Packet header, 12 bytes

7 6 5 4 3 2 1 0

Serial number

0 B

1 B

2 B

4 B

**Version field:**

Version

Version field takes up 4 bits, it is the lower bits of packet header’s 1st byte (bit3 - bit0). Version range is from 0 to 15. This protocol will increase version number when there is major adjustment. The initial version number is 0.

**Flag field:**

CRC

Reserved

Reserved

Reserved

Flag field consist of 4 bits, it is the higher bits of packet header’s 1st byte (bit7 – bit4). It consists of the following bits:

* CRC (bit 7)

When this bit set to 1, CRC check is enabled in this packet. When CRC check is enabled, CRC check value will put in “CRC check” field of packet header. When CRC check is disabled, “CRC check” field will be “0”. This protocol support 16bit CRC check CRCR16.

**Packet length field:**

This field defines the total length of the data packet, unit: byte.

**Packet command field:**

Packet command word is in this field. It is used to distinguish different packet. Meanwhile packet body will be different according to packet command word. Packet command word decide packet body format.

**Reserved field:**

Not used yet

**Serial number field:**

This field is packet serial number. The format of serial number is not defined. Platform and terminal can have different format of serial number. After receiving request packet, platform and terminal need to put received serial number in the answer packet and send back to the packet request side.

## Protocol Packet Body

Protocol packet body includes fixed parameter and variable parameter section. The format is decided by packet command word.

Fixed parameter section

Variable parameter section, this section include several TLV

## Transport Carrier Of Protocol Packet

This protocol haven’t specified transport carrier, so you can choose some transport layer protocol, such as TCP, UDP. Besides, this protocol data can also be transported over application layer such as HTTP, SMS etc. But it need to convert code, there is no special definition here. Meanwhile this protocol data can also be transported over serial line.

When this protocol data is transported over flow carrier (such as TCP and serial line etc), you need to encapsulate protocol data, this protocol requires using SLIP protocol (see appendix) to encapsulate the data in this situation.





# Protocol Command Message Definition

## Command Message Parameter Unit Structure

TLV: Type Length Value. TLV is unit structure of parameter. See detailed structure in “Parameter unit Type Length Value (TLV) definition” chapter.

## Command ID Definition

The command ID of all application system is unique.

Even number is command request, odd number is command answer. If command request does not need answer, the next defined command word needs to jump over the answer command word of this request. To ensure that answer command word is odd number.

|  |  |  |
| --- | --- | --- |
| Command ID | Name definition | Statement |
| 0xAA00 | POSITION | Terminal position data |
| 0xFF01 | POSITION\_RSP | Platform position data answer |
| 0xAA02 | LOGIN\_REQ | Terminal login request |
| 0xFF03 | LOGIN\_RSP | Platform login answer |
| 0xAA04 | SET\_REQ | Platform set parameter request |
| 0xFF05 | SET\_RSP | Terminal set parameter answer |
| 0xAA06 | UPGRADE\_REQ | Platform remote update command |
| 0xFF07 | UPGRADE\_RSP | Terminal remote update answer |
| 0xAA08 | DOWN\_REQ | Terminal update data request |
| 0xFF09 | DOWN\_RSP | Platform update data answer |
| 0xAA12 | UPFAULT\_REQ | Terminal upload fault status |
| 0xFF13 | UPFAULT\_RSP | Platform upload fault status answer |
| 0xAA14 | HSO\_REQ | Terminal handshake request |
| 0xFF15 | HSO\_RSP | Platform handshake answer |
| 0xAA16 | CTRL\_REQ\* | Platform remote control request |
| 0xFF17 | CTRL\_RSP\* | Terminal remote control answer |

## Position Message

* **POSITION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Send | Device->Platform | | Answer | | Not need answer |
| Fixed parameter section | | | | | |
| Length | Definition | | Explanation | | |
| SHORT | Status | | 16 bits, use from high bit to low bit  Location status | ACC status | GNS module | G-sensor | OBD bus| Buffer data  0: not located, 1: located  0: ACC not start, 1: ACC started  0: normal, 1: abnormal  0: normal, 1: abnormal  0: not connected, 1: connected  0: not buffer data,1:buffer data | | |
| SHORT | Type | | 0: real time position 1: break point data  2: logout 3: over speed  4: fatigue 5: set protection  6: SOS 7: harsh deceleration  8: harsh acceleration 9: harsh turning  10: impact 11: roll over  12: high RPM 13: RPM & speed mismatch  14: idle alarm 15: plug-in alarm   * 16: plug-out alarm 17: Turn [compensation](javascript:void(0);) * 30: [Coolant](javascript:void(0);) [temperature](javascript:void(0);) alarm | | |
| CSTRING | Time | | Format is YYMMDDHHFFSS (UTC time) | | |
| INTGER | [Longitude](app:ds:longitude) | | Signed integer, 100000 times of degree value  + is east [longitude](app:ds:longitude), - is west [longitude](app:ds:longitude) | | |
| INTGER | [Latitude](app:ds:latitude) | | Signed integer, 100000 times of degree value  + is north [latitude](app:ds:latitude), - is south [latitude](app:ds:latitude) | | |
| SHORT | Speed | | GPS speed, unit: km/h | | |
| SHORT | Direction | | North angle, unit: degree | | |
| SHORT | Altitude | | Unit: meter | | |
| SHORT | Odometer | | Odometer speed, unit: km/h | | |
| Variable parameter section | | | | | |
| Type | Length | Value | | Explanation | |
| 0X0001 | BIN | OBD information | | 1 byte coolant temperature ,need -40 (range from -40~215; 215: invalid,eg: 0=0-40=-40degree,130=130-40=90degree)  2 bytes RPM (range from 1~65535, 65534: invalid)  1 byte average speed, unit: km/h  2 bytes OBD fuel consumption (interval time), unit: 1/10000L  2 bytes OBD fuel consumption per 100km in current average speed, unit: L/100Km  2500=25 L/100Km  2 bytes interval mileage, unit: meter  2 bytes voltage, unit: 1/100  1 byte fuel tank volume, unit: 0.4%, 20=20\*0.4%=5%; 0xFF: invalid  4 bytes trip ID | |
| 0X0002 | BIN | J1939 information | | 4 bytes J1939 total fuel consumption, unit: 0.5L, 50=25L  4 bytes J1939 total mileage, unit: 0.125km, 200=25km | |
| 0X0003 | BIN | [reserve](javascript:void(0);) | | [reserve](javascript:void(0);) | |
| 0X0004 | BIN | Trip ID | | 1 byte trip indication, 0x01 trip end.  4 bytes trip ID | |
| 0x0005 | BIN | State Info. | | 1 byte GSM signal:0~32, >32 invalid  1 byte GPS signal:0~24  4 byte run time: run seconds | |
| Send | | | | Answer | |
| Device->Platform | | | | Need answer, serial number must keep the same | |

* **POSITION\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| 无 |  | |  |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| None |  |  |  |
| Send | | | Answer |
| Platform->Device | | | No need answer |

## Login Message

* **LOGIN\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| None |  | |  |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0X0001 | CSTRING | Firmware version number | Must contain this parameter |
| 0X0002 | CSTRING | VIN | Vehicle VIN number |
| 0X0003 | CSTRING | IMEI | Must contain this parameter |
| 0x0004 | CSTRING | reserve |  |
| 0X0005 | BYTE | Parameter request identification | 0x00=no need synchronize parameter  0x01=need synchronize parameter |
| 0x0006 | CSTRING | APN username | APN username |
| 0x0007 | CSTRING | APN password | APN password |
| 0x0008 | BYTE | Device Type | 0x00: OBD Device  0x01: J1939 Device |
| 0x0009 | CSTRING | ICCID | SIM card ICCID |
| 0x000A | CSTRING | APN | APN |
| Send | | | Answer |
| Device->Platform | | | Need answer |

* **LOGIN\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| Byte | Login result | | 0x00: Login normal, OK  0x01: Device not exist, ERROR |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0X0001 | BYTE | Protection status | 0x00=not protected 0x01=protected |
| 0X0002 | SHORT | Report interval | Unit: second |
| 0X0003 | SHORT | When Sleep Wake up interval | Unit: Min |
| 0X0004 | BYTE | Over speed limit | Unit: km |
| 0X0005 | SHORT | Over speed delay interval | Unit: second |
| 0X0006 | SHORT | Fatigue driving duration | Unit: second |
| 0X0007 | SHORT | Fatigue driving delay interval | Unit: second |
| 0X0008 | SHORT | Harsh acceleration threshold | Unit:0.1G,exp.: 2 mean 0.2g |
| 0X0009 | SHORT | Harsh deceleration threshold | Unit:0.1G,exp.: 3 mean 0.3g |
| 0X0010 | SHORT | Harsh turning threshold | Unit:0.1G,exp.: 4 mean 0.4g |
| 0X0011 | SHORT | High RPM threshold | Unit: RPM (Round Per Minute) |
| 0X0012 | BIN | RPM & speed mismatch threshold | This threshold is combination of speed & RPM, example:   |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | 90 km/h | 3500 RPM | | | |  |  | | --- | --- | | 60 km/h | 4000 RPM | | | |  |  | | --- | --- | | 30 km/h | 4500 RPM | |   Speed & RPM are SHORT type |
| 0X0013 | CSTRING | Update file identification | 30 bytes fixed length, if not enough, fill with 0x00 |
| 0X0014 | SHORT | Amount of update data packet | If the answer contains 0X0013 & 0X0014, than do the update |
| 0X0015 | CSTRING | Server UTC time | Synchronize device time & server time,  String UTC seconds |
| 0x0016 | BYTE | Shake Sensitivity | 0x01 easy  0x02 normal  0x04 hard  0x06 hardest |
| 0x0040 | BYTE | Break Data Mode | 0x00 FIFO mode (Default)  0x01 MIX mode |
| 0x0041 | BYTE | Coolant Temperature  Alarm gate | Unit: degree (<150) |
| 0x0042 | BYTE | Angle compensation interval | Unit: degree(>5 and <250) |

## Set Parameter Message

* **SET\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0X0001 | BYTE | Protection status | 0x00=not protected 0x01=protected |
| 0X0002 | SHORT | Report interval | Unit: Second |
| 0X0003 | SHORT | When Sleep Wake up interval | Unit: Min |
| 0X0004 | BYTE | Over speed limit | Unit: km |
| 0X0005 | SHORT | Over speed delay interval | Unit: km |
| 0X0006 | SHORT | Fatigue driving duration | Unit: second |
| 0X0007 | SHORT | Fatigue driving delay interval | Unit: second |
| 0X0008 | SHORT | Harsh acceleration threshold |  |
| 0X0009 | SHORT | Harsh deceleration threshold |  |
| 0X0010 | SHORT | Harsh turning threshold |  |
| 0X0011 | SHORT | High RPM threshold | Unit: RPM (Round Per Minute) |
| 0X0012 | BIN | RPM & speed mismatch threshold | This threshold is combination of speed & RPM, example:   |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | 90 km/h | 3500 RPM | | | |  |  | | --- | --- | | 60 km/h | 4000 RPM | | | |  |  | | --- | --- | | 30 km/h | 4500 RPM | |   Speed & RPM are SHORT type |
| 0X0013 | CSTRING | Update file identification | 30 bytes fixed length, if not enough, fill with 0x00 |
| 0X0014 | SHORT | Amount of update data packet | If the answer contains 0X0013 & 0X0014, than do the update |
| 0X0015 | CSTRING | Server UTC time | Synchronize device time & server time  String UTC seconds |
| 0x0016 | BYTE | Shake Sensitivity | 0x01 easy  0x02 normal  0x04 hard  0x06 hardest |
| 0x0040 | BYTE | Break Data Mode | 0x00 FIFO mode (Default)  0x01 MIX mode |
| 0x0041 | BYTE | Coolant Temperature  Alarm gate | Unit: degree (<150) |
| 0x0042 | BYTE | Angle compensation interval | Unit: degree(>5 and <250) |
| Send | | | Answer |
| Platform->Device | | | Need answer |

* **SET\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| Byte | Result | | 0x00: set success  0x01: set failed |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 无 |  |  |  |
| Send | | | Answer |
| Device->Platform | | | No need answer |

## Remote Update

* **UPGRADE\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| CSTRING | Update file identification | | Identification used for data extracting, 30 bytes fixed length, if not enough, fill with space |
| SHORT | Amount of sub-packet | |  |
|  |  | |  |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0x0001 | CSTRING | Version number | New software version number |
| Send | | | Answer |
| Platform->Device | | | Need answer |

* **UPGRADE\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| Byte | Result | | 0x00: Update normal  0x01: Cannot update  0x02: Latest version, no need update |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 无 |  |  |  |
| Send | | | Answer |
| Device->Platform | | | No need answer |

## Update Data Sub-packet Download Message

* **DOWN\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| CSTRING | Update file, or sub-packet data identification | | Identification which used to acquire data, 30 bytes fixed length, if not enough, fill with space |
| SHORT | Current packet number | | Start with 0, 0XFFFF: download finished |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| None |  |  |  |
| Send | | | Answer |
| Device->Platform | | | Need answer |

* **DOWN\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| SHORT | Current packet number | | Current packet number |
| SHORT | Packet data length | | Data length of variable parameter section, 0xFFFF:  Must 512, Except last packet  requested data does not exist |
| 5Bytes | reserve | | Reserve |
| BIN | Firmware data | | Must 512bytes,except last packet |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
|  |  |  |  |
| Send | | | Answer |
| Platform->Device | | | No need answer |

## Fault Report

* **UPFAULT\_REQ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fixed parameter section | | | | |
| Length | Definition | | Explanation | |
| None |  | |  | |
| Variable parameter section | | | | |
| Type | Length | Value | Explanation | |
| 0X0001 | CSTRING | Current OBD fault code | Current fault code | |
| 0XFF37 | CSTRING | Current J1939 fault code | Current fault code |
| Send | | | Answer | |
| Device->Platform | | | Need answer | |

* **UPFAULT\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| BYTE | Result | | 0x00 receive success  0x01 receive failed |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| None |  |  |  |
| Send | | | Answer |
| Platform->Device | | | No need answer |

## Handshake Message

* **HSO\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| None |  | |  |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| None |  |  |  |
| Send | | | Answer |
| Device->Platform | | | Need answer |

* **HSO\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| None |  | |  |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0X0015 | CSTRING | Server UTC time |  |
| Send | | | Answer |
| Platform->Device | | | No need answer |

## Remote control Message

* **CTRL\_REQ**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| BYTE | Control command byte | | 0x00: Inquire state;  0x01: Lock the relay1;  0x02: Unlock the relay1;  0x03: Lock the relay2;  0x04: Unlock the relay2; |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| None |  |  |  |
| Send | | | Answer |
| Platform->Device | | | Need answer |

* **CTRL\_RSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Fixed parameter section | | | |
| Length | Definition | | Explanation |
| BYTE | Control result | | 0x00: OK  0x01: ERROR |
| Variable parameter section | | | |
| Type | Length | Value | Explanation |
| 0X0001 | BYTE | Relay 1 state | 0x00 Lock  0x01 Unlock  0xff Unknow |
| 0X0002 | BYTE | Relay 2 state | 0x00 Lock  0x01 Unlock  0xff Unknow |
| Send | | | Answer |
| Device->Platform | | | No need answer |

**5.Protocol Example**

## 5.1. LOGIN Message

**LOGIN\_REQ:**

C00000007EAA020000000000010001001047315F48312E305F56312E3000030013383632393530303238353334333036000400144C342D56374C673979497A7A2D724A6D0005000501000600084341524400070008434152440008000500000900183839383630303530313931343436313130393134000A0009434D4E4554C0

C0: Message head and end.

0000007EAA02000000000001: "The packet header", 0x7E: Packet length, 0xAA02:Packet command, 0x00000001: Serial number.

0001001047315F48312E305F56312E30:<Firmware Ver.: G1\_H1.0\_V1.0>

00030013383632393530303238353334333036:<IMEI:862950028534306>

000400144C342D56374C673979497A7A2D724A6D:<password: L4-V7Lg9yIzz-rJm>

0005000501:< need synchronize parameter >

0006000843415244:<APN username: CARD>

0007000843415244:<APN password: CARD>

0008000500:<Device Type: OBD >

000900183839383630303530313931343436313130393134:<ICCID: 89860050191446110914>

000A0009434D4E4554:<APN:CMNET>

**LOGIN\_RSP: // need synchronize parameter**

C001000075FF0300000000000100000100050100020006001E00030006003C00040005780005000600500006000638400007000604B000080006000300090006000500100006000500110006119400120010001E0FA0003C0DAC005A0BB800160005020015000E31343332363037323136C0

00:<Login result,0x00=login normal, OK>

0001000501:<Protection status,0x01=protected>

00020006001E:<Report interval,0x1E=30s>

00030006003C:< When Sleep Wake up interval,0x3C=60min>

0004000578:< Over speed limit,0x78=120km/h>

000500060050:< Over speed delay interval,0x50=80s>

000600063840:< Fatigue driving duration,0x3840=14400s>

0007000604B0:< Fatigue driving delay interval ,0x4B0=1200s>

000800060003:< Harsh acceleration threshold ,0x0003=0.3g>

000900060005:< Harsh deceleration threshold ,0x0005=0.5g>

001000060005:< Harsh turning threshold ,0x0005=0.5g>

001100061194:< High RPM threshold ,0x1194=4500rpm>

00120010001E0FA0003C0DAC005A0BB8:

<001E0FA0:30KM—4000rpm>

<003C0DAC:60KM—3500rpm>

<005A0BB8:90KM—3000rpm>

0016000502:< Shake Sensitivity ,0x02=normal>

0015000E31343332363037323136:<UCT time, 1432607216s=2015-05-26 10:26:56>

**LOGIN\_RSP: //no need synchronize parameter**

C00100000DFF0300000000000100C0

00:<Login result,0x00=login normal, OK>

## 5.2. Position Message

**POSITION**

C000000041AA00000000000030DBDC00000031353035323630373538323800ADDCC100226AEF0000000000120005000100151206EF0504E99975002903EB80556492CEC0

DBDC00: DBDC SLIP escape to C0 ,So change to C000: ACC start and GPS located.

0000: real time position

313530353236303735383238:UTC time <15-05-26 07:58:28>

00ADDCC1: 11394241/100000=113.94241 degree

00226AEF: 2255599/100000=22.55599 degree

0000: speed 0 km/h.

0000: direction 0 degree.

0012:Altitude 12 meter.

0005: Odometer speed 5 Km/h

000100151206EF0504E99975002903EB80556492CE: OBD information.

**POSITION\_RSP**

C00100000CFF01000000000030C0

POSITION\_RSP Serial number must same with **POSITION,** the example is 00000030

## 5.3 Set Parameter message

If you need to send to a unit to configure the following:

**Report Interval = 30 seconds**

**When Sleep – Wake-up interval = 30 minutes**

**Harsh acceleration threshold = 0.3g**

**Harsh deceleration threshold = 0.5g**

**Harsh turning threshold = 0.5g**

C0             Packet start - not included in the packet length

00              Flag and Version

00              Reserved

0024           Hex value for packet length (length of the packet excluding the C0 begin and end bytes) - in this case 36 bytes

AA04          Command type, in this case: SET\_REQ

0000           CRC

00000001    Sequence number

0003          When sleep, wake up interval command

0006          Length for command

001E         this is the hex value for 30 minutes

0008          Harsh acceleration sensitivity command

0006          Length for command

0003          0.3 g

0009          Harsh deceleration sensitivity command

0006          Length for command

0005          0.5 g

0010          Harsh turn sensitivity command

0006          Length for command

0005          0.5 g

C0             Packet end - not included in the packet length

So the command that you will send to the device will look like this:

**C000000020AA0400000000000100030006001E000800060003000900060005001000060005C0**

## 5.4 Fault Report

**UPFAULT\_REQ**

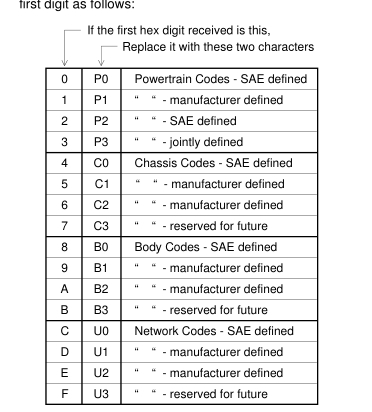
C00000001CAA120000000000020001001001000200030043008200C100C0

0001001001000200030043008200C100

**Type: 0x0001; Length: 0x0010=16Bytes; Data:** 001000200030043008200C100

0100   0200  0300  4300    8200   C100

P0100 P0200 P0300 C0300 B0200 U0100



## 5.3. Flow Chart

***LOGIN and POSITION***



***Device auto update***



***Server command device update***

**Appendix I SLIP Protocol Frame Format**

**The below rules describe SLIP protocol frame format:**

1. The data frame end with identifier “END” (0xC0). Meanwhile, in order to prevent the line noise before the arrival of data frame is misused as the contents of the datagram, at the beginning of the data frame mostly send an END character. If there is line noise, END character cancels this wrong packet. Therefore, the current packet will be transmitted correctly. The upper layer will discard the prior wrong packet as it is [meaningless](app:ds:meaningless).
2. If a character in data frame is END, then bytes 0xDB and 0xDC will be continuously transmitted (or translated) to replace it. Character 0xDB is taken as ESC character of SLIP, but its value is different from ESC character (0x1B) of ASCII code.
3. If a character in data frame is ESC character of SLIP, then bytes 0xDB and 0xDD will be continuously transmitted (or translated) to replace it.

**Pseudo code:**

//Receive handle

short ReceiveHandleData(uint8\_t \*buffer, int buffer\_size)

{

    short i = 0, j = 0;

    while(i < buffer\_size) {

        if(buffer[i] == 0xDB) {

            if(0xDC == buffer[++i]) {

                buffer[j++] = 0xC0;

            } else {

                buffer[j++] = 0xDB;

            }

        } else {

            buffer[j++] = buffer[i];

        }

        i++;

    }

    return j;

}

 //Send handle

int TcpSendHandle(uint8\_t \*sendBuffer, uint8\_t \*handleBuffer, int len)

{

    int i = 0;

    int size = 1;

    handleBuffer[0] = 0xC0;

    while(i < len) {

        if(sendBuffer[i] == 0xC0) {

            handleBuffer[size++] = 0xDB;

            handleBuffer[size++] = 0xDC;

        } else if(sendBuffer[i] == 0xDB) {

            handleBuffer[size++] = 0xDB;

            handleBuffer[size++] = 0xDD;

        } else {

            handleBuffer[size++] = sendBuffer[i];

        }

        i++;

    }

    handleBuffer[size++] = 0xC0;

    return size;

}