OBD Tracker
Communication Protocol

CONTENT

I. TERMS, DEFINITIONS 2

II. BASIC RULES 3

IV. DATA PACKET FORMAT 5

5.1. LOGIN MESSAGE PACKET 6

5.1.1. Terminal Sending Data Packet to Server(0X01) 6

5.2. LOCATION DATA PACKET (0X22) 8

5.3. ALARM PACKET (0X26) 13

5.4. HEARTBEAT PACKET (0X13) 18

5.5. OBD Message(0x8C) 20

5.6. TPMS Message(0x8D) 23

5.7. DTC Message(0x65) 25

5.8. PID Message(0x66) 26

6.1. PACKET SENT BY SERVER(0X80) 29

6.2. PACKET REPLIED BY TERMINAL(0X21) 30

Ⅶ. APPENDIX A: CODE FRAGMENT OF THE CRC-ITU LOOKUP TABLE ALGORITHM IMPLEMENTED BASED ON C LANGUAGE 31

Communication Protocol

Introduction

This document defines instructions about interface protocol on application layer of vehicles OBD tracker and location-based service platform. Related interface protocol only applies in the interaction between the platform and the position terminal.

i. Terms, Definitions

Terms, Abbreviation	Definition in English	Definition in Chinese
CMPP	China Mobile Peer to Peer	中国移动点对点协议
GPS	Global Positioning System	全球卫星定位系统
GSM	Global System for Mobile Communication	全球移动通信系统
GPRS	General Packet Radio Service	通用无线分组业务
TCP	Transport Control Protocol	传输控制协议
LBS	Location Based Services	辅助定位服务
IMEI	International Mobile Equipment Identity	国际移动设备识别码
MCC	Mobile Country Code	移动用户所属国家代号
MNC	Mobile Network Code	移动网号码
LAC	Location Area Code	位置区码
Cell ID	Cell Tower ID	移动基站
UDP	User Datagram Protocol	用户数据报协议
SOS	Save Our Ship/Save Our Souls	遇难求救信号
CRC	Cyclic Redundancy Check	循环冗余校验
NITZ	Network Identity and Time Zone,	时区
GIS	Geographic Information System	地理信息系统

ii. Basic Rules

1. If a GPRS connection is established successfully, the terminal will send a first login message packet to the server and, within five seconds, if the terminal receives a data packet responded by the server, the connection is considered to be a normal connection. The terminal will begin to send location information (i.e., GPS, LBS information package). A status information package will be sent by the terminal after three minutes to regularly confirm the connection.

2. If the GPRS connection is established unsuccessfully, the terminal will not be able to send the login message packet. The terminal will start schedule reboot in twenty minutes if the GPRS connection is failed three times. Within twenty minutes, if the terminal successfully connects to the server and receives the data packet from the server as the server's response to the login message packet sent by the terminal, the schedule reboot will be off and the terminal will not be rebooted; otherwise, the terminal will be rebooted automatically in twenty minutes.

3. After receiving the login message packet, the server will return a response data packet. If the terminal doesn't receive packet from the server within five seconds after sending the login message packet or the status information package, the current connection is regarded as an abnormal connection. The terminal will start a retransmission function for GPS tracking data, which will cause the terminal to disconnect the current GPRS connection, rebuild a new GPRS connection and send a login message packet again.

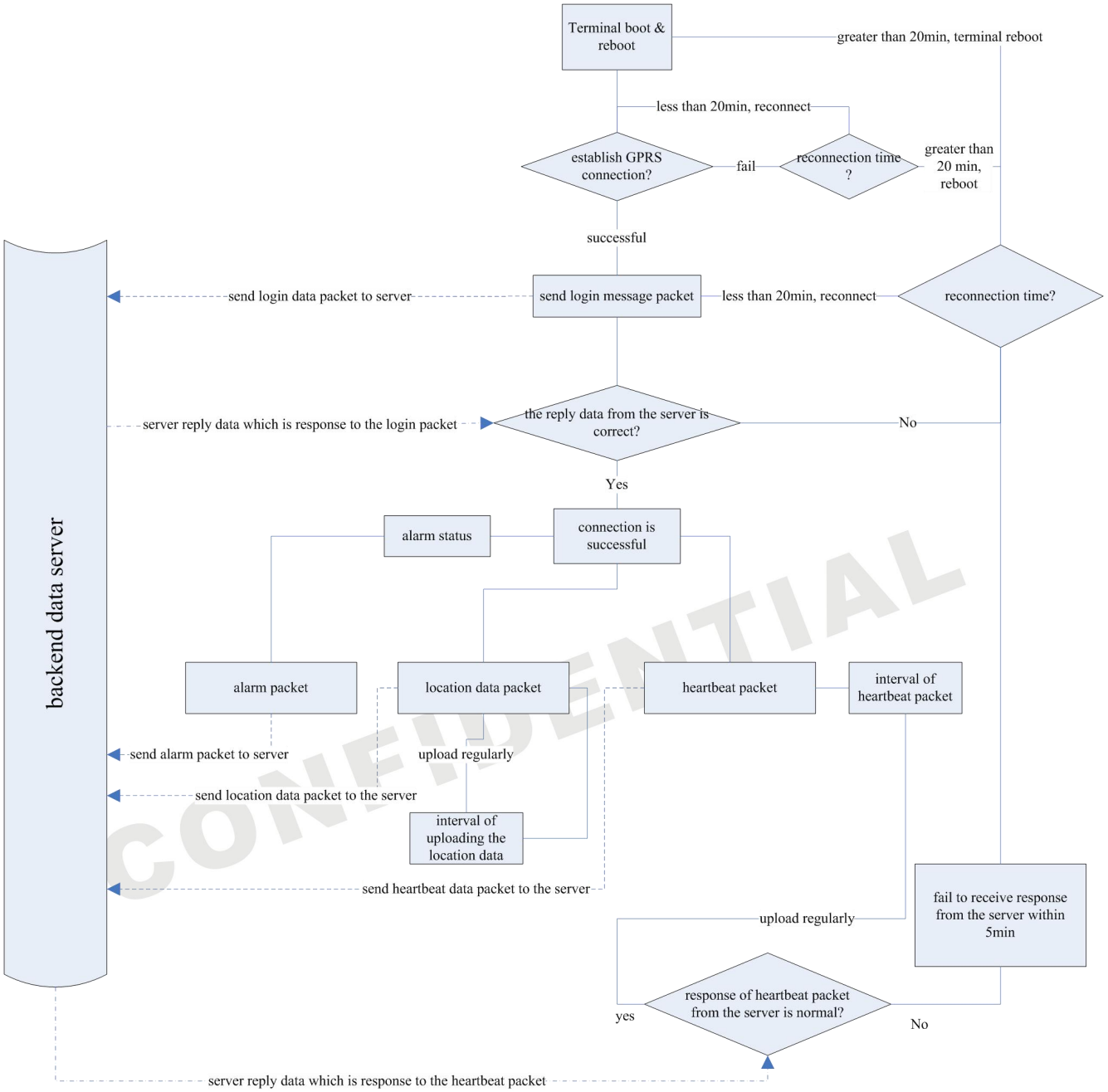
4. If the connection is regarded to be abnormal, and the data packet as a response from the server is failed to be received three times after a connection is established and a login message packet or status information package is sent, the terminal will start schedule reboot and the scheduled time is ten minutes. Within ten minutes, if the terminal successfully connects to the server and receives the data packet responded by the server, the schedule reboot will be off and the terminal will not be rebooted; otherwise, the terminal will be rebooted automatically in ten minutes.

5. In case of the normal connection, the terminal will send a combined information package of GPS and LBS to the server after the GPS information is changed; and the server may set a default protocol for transmission by using commands.

6. To ensure the effectiveness of the connection, the terminal will send status information to the server at regular intervals, and the server will return response data packets to confirm the connection.

7. For the terminal which doesn't register an IMEI number, the server will reply the terminal with a login request response and heartbeat packet response, rather than directly disconnect the connection. (If the connection is directly disconnected or the server doesn't reply to the terminal, it will lead to a continuous reconnected by the terminal and the GPRS traffic will be consumed heavily.

Data Flow Diagram



iv. Data Packet Format

The communication is transferred asynchronously in bytes.

The total length of packets is (10+N) Bytes.

Format	Length(Byte)
Start Bit	2
Packet Length	1
Protocol Number	1
Information Content	N
Information Serial Number	2
Error Check	2
Stop Bit	2

4.1. Start Bit

Fixed value in HEX 0x78 0x78.

4.2. Packet Length

Length = Protocol Number + Information Content + Information Serial Number + Error Check, totally (5+N)Bytes, because the Information Content is a variable length field.

4.3. Protocol Number

Type	Value
Login Message	0x01
Location Message	0x22
Heartbeat Message	0x13
String information	0x21
Alarm Message	0x26
Command information sent by the server to the terminal	0x80
OBD Message	0x8C
TPMS Message	0x8D
DTC Message	0x65
PID Message	0x66

4.4. Information Contents

The specific contents are determined by the protocol numbers corresponding to different applications.

4.5. Information Serial Number

The serial number of the first GPRS data (including status packet and data packet such as GPS, LBS) sent after booting is '1', and the serial number of data sent later at each time will be automatically added '1'.

4.6. Error Check

A check code may be used by the terminal or the server to distinguish whether the received information is error or not. To prevent errors occur during data transmission, error check is added to against data misoperation, so as to increase the security and efficiency of the system. The check code

is generated by the CRC-ITU checking method.

The check codes of data in the structure of the protocol, from the Packet Length to the Information Serial Number (including “Packet Length” and “Information Serial Number”) , are values of CRC-ITU.

CRC error occur when the received information is calculated, the receiver will ignore and discard the data packet.

4.7. Stop Bit

Fixed value in HEX 0x0D 0x0A.

Ⅴ.Details about Data Packet sent by Server to Terminal

The commonly used information packages sent by the terminal and those sent by the server will be interpreted separately.

5.1. Login Message Packet

5.1.1. Terminal Sending Data Packet to Server(0X01)

The login message packet is used to be sent to the server with the terminal ID so as to confirm the established connection is normal or not.

Login Message Packet(22Byte)	Description	Bits	Example
	Start Bit	2	0x78 0x78
	Packet Length	1	0x11
	Protocol Number	1	0x01
	Terminal ID	8	0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45
	Retain	2	0x00 0x00
	Retain	2	0x00 0x00
	Information Serial Number	2	0x00 0x01
	Error Check	2	0x8C 0xDD
	Stop Bit	2	0x0D 0x0

5.1.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.1.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.1.1.3. Protocol Number

For details see Data Packet Format section 4.3.

5.1.1.4. Terminal ID

The terminal ID applies IMEI number of 15 bits.

Example: if the IMEI is 123456789012345,

the terminal ID is 0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45.

5.1.1.5. Retain

5.1.1.6. Retain

5.1.1.7. Information Serial Number

For details see Data Packet Format section 4.5.

5.1.1.8. Error Check

For details see Data Packet Format section 4.6.

5.1.1.9. Stop Bit

For details see Data Packet Format section 4.7.

5.1.2. Server Responds the Data Packet

	Description	Bits	Example
Login Message Packet (18 Byte)	Start Bit	2	0x78 0x78
	Packet Length	1	0x05
	Protocol Number	1	0x01
	Information Serial Number	2	0x00 0x01
	Error Check	2	0xD9 0xDC
	Stop Bit	2	0x0D 0x0A

The response packet from the server to the terminal: the protocol number in the response packet is identical to the protocol number in the data packet sent by the terminal.

5.1.2.1. Start Bit

For details see Data Packet Format section 4.1.

5.1.2.2. Packet Length

For details see Data Packet Format section 4.2.

5.1.2.3. Protocol Number

For details see Data Packet Format section 4.3.

5.1.2.4. Information Serial Number

For details see Data Packet Format section 4.5.

5.1.2.5. Error Check

For details see Data Packet Format section 4.6.

5.1.2.6. Stop Bit

For details see Data Packet Format section 4.7.

5.2. Location Data Packet (combined information package of GPS and LBS)(0X22)

5.2.1. Terminal Sending Location Data Packet to Server

Format			Length(Byte)	Example
Information Content	Start Bit		2	0x78 0x78
	Packet Length		1	0x22
	Protocol Number		1	0x22
	GPS Information	Date Time	6	0x0B 0x08 0x1D 0x11 0x2E 0x10
		Quantity of GPS information satellites	1	0xCF
		Latitude	4	0x02 0x7A 0xC7 0xEB
		Longitude	4	0x0C 0x46 0x58 0x49
		Speed	1	0x00
		Course, Status	2	0x14 0x8F
	LBS Information	MCC	2	0x01 0xCC
		MNC	1	0x00
		LAC	2	0x28 0x7D
		Cell ID	3	0x00 0x1F 0xB8
	ACC Status		1	0x01
	Data report mode		1	0x00
	GPS real time supplementary transmission		1	0x00
	Serial Number		2	0x00 0x03
	Error Check		2	0xF2 0xB5
	Stop Bit		2	0x0D 0x0A

5.2.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.2.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.2.1.3. Protocol Number

For details see Data Packet Format section 4.3.

5.2.1.4. Date Time

Format	Length(Byte)	Example
Year	1	0x0A
Month	1	0x03

Day	1	0x17
Hour	1	0x0F
Minute	1	0x32
Second	1	0x17

Example: 2010-03-23 15:30:23

Calculated as follows: 10(Decimal)=0A(Hexadecimal)

3 (Decimal)=03(Hexadecimal)

23(Decimal)=17(Hexadecimal)

15(Decimal)=0F(Hexadecimal)

50(Decimal)=32(Hexadecimal)

23(Decimal)=17(Hexadecimal)

Then the value is: 0x0A 0x03 0x17 0x0F 0x32 0x17

5.2.1.5. Length of GPS information, quantity of positioning satellites

The field is 1 Byte displayed by two hex digits, wherein the first one is for the length of GPS information and the second one for the number of the satellites join in positioning.

Example: if the value is 0xCB, it means the length of GPS information is 12 and the number of the positioning satellites is 11.

(C = 12Bit Lenght ， B = 11 satellites)

5.2.1.6. Latitude

Four bytes are consumed, defining the latitude value of location data. The range of the value is 0-162000000, indicating a range of 0°-90°. The conversion method thereof is as follow:

Converting the value of latitude and longitude output by GPS module into a decimal based on minute; multiplying the converted decimal by 30000; and converting the multiplied result into hexadecimal

Example: 22º32.7658'=(22X60+32.7658)X30000=40582974, then converted into a hexadecimal number

40582974(Decimal)= 26B3F3E(Hexadecimal)

at last the value is 0x02 0x6B 0x3F 0x3E.

5.2.1.7. Longitude

Four bytes are consumed, defining the longitude value of location data. The range of the value is 0-324000000, indicating a range of 0°-180°.

The conversion method herein is same to the method mentioned in Latitude (see section 5.2.1.6).

5.2.1.8. Speed

One byte is consumed, defining the running Speed of GPS. The value ranges from 0x00 to 0xFF indicating a range from 0 to 225km/h.

e.g. 0x00 represents 0 km/h.

0x10 represents 16km/h.

0xFF represents 255 km/h.

5.2.1.9. Course & Status

1.Two bytes are consumed, defining the running direction of GPS. The value ranges from 0° to 360° measured clockwise from north of 0°.

BYTE_1	Bit7	0	
	Bit6	0	1
	Bit5	GPS real-time/differential positioning	
	Bit4	GPS having been positioning or not	
	Bit3	East Longitude, West Longitude	
	Bit2	South Latitude, North Latitude	
	Bit1	Course	
	Bit0		
BYTE_2	Bit7		
	Bit6		
	Bit5		
	Bit4		
	Bit3		
	Bit2		
	Bit1		
	Bit0		

2. The status information in the data packet is the status corresponding to the time bit recorded in the data packet.

For example: the value is 0x15 0x4C, the corresponding binary is 00010101 01001100,

BYTE_1 Bit7 0
BYTE_1 Bit6 0
BYTE_1 Bit5 0 (real time GPS)
BYTE_1 Bit4 1 (GPS has been positioned)
BYTE_1 Bit3 0 (East Longitude)
BYTE_1 Bit2 1 (North Latitude)
BYTE_1 Bit1 0
BYTE_1 Bit0 1
BYTE_2 Bit7 0
BYTE_2 Bit6 1
BYTE_2 Bit5 0 → Course 332° (0101001100 in Binary, or 332 in decimal)
BYTE_2 Bit4 0
BYTE_2 Bit3 1
BYTE_2 Bit2 1
BYTE_2 Bit1 0
BYTE_2 Bit0 0

which means GPS tracking is on, real time GPS, location at north latitude, east longitude and the course is 332°.

5.2.1.10. MCC

The country code to which a mobile user belongs, i.e., Mobile Country Code(MCC).

Example: Chinese MCC is 460 in decimal, or 0x01 0xCC in Hex (that is, a decimal value of 460

converting into a hexadecimal value, and 0 is added at the left side because the converted hexadecimal value is less than four digits).

Herein the range is 0x0000 ~ 0x03E7.

5.2.1.11. MNC

Mobile Network Code(MNC)

Example: Chinese MNC is 0x00.

5.2.1.12. LAC

Location Area Code (LAC) included in LAI consists of two bytes and is encoded in hexadecimal. The available range is 0x0001-0xFFFE, and the code group 0x0000 and 0xFFFF cannot be used. (See GSM specification 03.03, 04.08 and 11.11).

5.2.1.13. Cell ID

Cell Tower ID (Cell ID), which values ranges from 0x000000 to 0xFFFFFF.

5.2.1.14. ACC Status

ACC Low is 0x00, ACC High is 0x01.

5.2.1.15. Data report mode

0x00 timing report

0x01 distance reporting

0x02 inflection point upload

0x03 ACC status change upload

5.2.1.16. GPS real time supplementary transmission

0x00 real time upload

0x01 complement

5.2.1.17. Information Serial Number

For details see Data Packet Format section 4.5.

5.2.1.18. Error Check

For details see Data Packet Format section 4.6.

5.2.1.19. Stop Bit

For details see Data Packet Format section 4.7.

5.2.2. Examples of Packet Sent from Terminal to Server

Example of sending by the terminal

78 78 22 22 0B 08 1D 11 2E 10 CC 02 7A C7 EB 0C 46 58 49 00 14 8F 01 CC 00 28 7D 00 1F B8 01 01 00 00 03 F2 B5 0D 0A

Explain

0x78 0x78	0x22	0x22	0x0B 0x08 0x1D 0x11 0x2E 0x10	0xCC	0x02 0x7A 0xC7 0xEB
Start Bit	Packet Length	Protocol No.	Date Time	Quantity of GPS information satellites	Latitude

0x0C 0x46 0x58 0x49	0x00	0x14 0x8F	0x01 0xCC	0x00	0x28 0x7D	0x00 0x1F 0xB8	0x01
Longitude	Speed	Course Status	MCC	MNC	LAC	Cell ID	ACC Status

0x01 Data mode	0x00 Upload mode	0x00 0x03 Serial No.	0xF2 0xB5 Error Check	0x0D 0x0A Stop Bit			

5.3. Alarm Packet (Combined information packet of GPS, LBS and Status)(0X26)

5.3.1. Server Sending Alarm Data Packet to Server

Format			Length (Byte)	Example
Information Content	Start Bit		2	0x78 0x78
	Packet Length		1	0x25
	Protocol Number		1	0x26
	Date Time		6	0x0B 0x0B 0x0F 0x0E 0x24 0x1D
	GPS Information	Quantity of GPS information satellites	1	0xCF
		Latitude	4	0x02 0x7A 0xC8 0x87
		Longitude	4	0x0C 0x46 0x57 0xE6
		Speed	1	0x00
		Course, Status	2	0x14 0x02
	LBS Information	LBS Length	1	0x09
		MCC	2	0x01 0xCC
		MNC	1	0x00
		LAC	2	0x28 0x7D
		Cell ID	3	0x00 0x1F 0x72
	status Information	Terminal Information Content	1	0x65
		Voltage Level	1	0x06
		GSM Signal Strength	1	0x04
		Alarm/Language	2	0x01 0x02
	Serial Number		2	0x00 0x36
	Error Check		2	0x2A 0x05
	Stop Bit		2	0x0D 0x0A

Alarm packet is consisted by adding status information to location packet, so does the encoding format of the protocol.

5.3.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.3.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.3.1.3. Protocol Number

For details see Data Packet Format section 4.3.

5.3.1.4. Date Time

For details see Location Data Packet Format section 5.2.1.4.

5.3.1.5. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.3.1.6. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.3.1.7. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.3.1.8. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.3.1.9. Status and Course

For details see Location Data Packet Format section 5.2.1.9.

5.3.1.10. MCC

For details see Location Data Packet Format section 5.2.1.10.

5.3.1.11. MNC

For details see Location Data Packet Format section 5.2.1.11.

5.3.1.12. LAC

For details see Location Data Packet Format section 5.2.1.12.

5.3.1.13. Cell ID

For details see Location Data Packet Format section 5.2.1.13.

5.3.1.14. Terminal Information

One byte is consumed, defining various status information of the mobile phone.

Bit		Code Meaning
BYTE	Bit7	1: oil and electricity disconnected
		0: gas oil and electricity connected
	Bit6	1: GPS tracking is on
		0: GPS tracking is off
	Bit3~ Bit5	100: SOS
		011: Low Battery Alarm
		010: Power Cut Alarm
		001: Shock Alarm
		000: Normal
	Bit2	1: Charge On
		0: Charge Off
	Bit1	1: ACC high
		0: ACC Low
	Bit0	1: Defense Activated
		0: Defense Deactivated

Example: 0x44, corresponding binary value is 01000100,
indicates that the status of the terminal is: oil and electricity connected, GPS tracking is on, normal without any alarm, charge on, ACC is low, and defense deactivated.

5.3.1.15. Voltage Level

The range is 0~6 defining the voltage is from low to high.

0: No Power (shutdown)

1: Extremely Low Battery (not enough for calling or sending text messages, etc.)

2: Very Low Battery (Low Battery Alarm)

3: Low Battery (can be used normally)

4: Medium

5: High

6: Very High

Example: 0x02 indicates very low battery and a Low Battery Alarm is sending.

5.3.1.16. GSM Signal Strength Levels

0x00: no signal;

0x01: extremely weak signal;
0x02: very weak signal;
0x03: good signal;
0x04: strong signal.
Example: 0x03 indicates the GSM signal is good.

5.3.1.17. Alarm/Language

0x00 (former bit) 0x01 (latter bit)
former bit: terminal alarm status (suitable for alarm packet and electronic fence project)
latter bit: the current language used in the terminal. 0x01 is mean Chinese, 0x02 is mean English

上电 Power up	0x0C
断电 Power drop	0x02
行程开始 ACC on	0xE0
行程结束 ACC off	0xE1
低压 Low votage	0x0E
电子围栏 Geo-Fence alarm	0xE6
MIL 尾气超标 MIL Exhaust gas over standard	0xE7
水温 Coolant Temperature	0xE8
车速 Over Speed	0x06
转速 Engine RPM	0xE9
停车未熄火 Long time parking at acc on	0xEA
疲劳驾驶 Fatigue Driving	0xEB
DTC	0xEC
危险驾驶 Dangerous driving	0xEE
急加速 Rapid acceleration	0xEF
急减速 Rapid deceleration	0XF0
急变道 Sharp change lane	0xF1
急转弯 Sharp turn	0xF2
碰撞 Collision	0xF3
拖掉 Trailer alarm	0xF4
振动 Shake	0x03

Examples:
No Alarm and Language is Chinese: 0x00 0x01
No Alarm and Language is English: 0x00 0x02

To increase the reliability of alarm information, labeling the alarm information repeatedly; in most cases, the alarm information keeps consistent with information of former terminal, while the inconsistencies are as follows:

- **A. Low Battery Alarm occurred in the information of the terminal**
- **B. Fence in and out Alarm in the Alarm/Language information**

5.3.1.18. Information Serial Number

For details see Data Packet Format section 4.5.

5.3.1.19. Error Check

For details see Data Packet Format section 4.6.

5.3.1.20. Stop Bit

For details see Data Packet Format section 4.7.

5.3.1.21. Examples

Examples of terminal transmission

78 78 25 26 0B 0B 0F 0E 24 1D CF 02 7A C8 87 0C 46 57 E6 00 14 02 09 01 CC 00 28 7D 00 1F 72 65 06 04 01 02 00 36 C5 61 0D 0A

Explain

0x78 0x78	0x25	0x26	0x0B 0x0B 0x0F 0x0E 0x24 x01D		0xCF		0x02 0x7A 0xC8 0x87
Start Bit	Length	Protocol No.	Date Time		Quantity of GPS information satellites		Latitude
0x0C 0x46 0x57 0xE6	0x00	0x14 0x02	0x09	0x01 0xCC	0x00	0x28 0x7D	0x00 0x1F 0x72
Longitude	Speed	Course Status	LBS Length	MCC	MNC	LAC	Cell ID
0x65	0x06	0x04	0x01 0x02	0x00 0x36	0xC5 0x61		0x0D 0x0A
Terminal Information Content	Voltage Level	GSM Signal Strength	Alarm/Language	Serial No.	Error Check		Stop Bit

Note: The status information in the data packet is the status corresponding to the time bit recorded in the data packet.

5.3.2. Server responding alarm data packet to terminal（terminal do not check enforcedly）

Format		Length(Byte)	
Information Content	Start Bit	2	
	Packet Length	1	
	Protocol Number	1	
	Serial Number	2	
	Error Check	2	
	Stop Bit	2	

Alarm packet is consisted by adding status information to location packet, so does the encoding format of the protocol.

5.3.2.1. Start Bit

For details see Data Packet Format section 4.1

5.3.2.2. Packet Length

For details see Data Packet Format section 4.2

5.3.2.3. Protocol Number

For details see Data Packet Format section 4.3

5.3.2.4. Serial Number

For details see Data Packet Format section 4.5

5.3.2.5. Error Check

For details see Data Packet Format section 4.6

5.3.2.6. Stop Bit

For details see Data Packet Format section 4.7

5.3.2.7. Examples

Example of data packet responded by the server

78 78 05 26 00 36 13 DE 0D 0A

5.4. Heartbeat Packet (status information packet)(0x13)

Heartbeat packet is a data packet to maintain the connection between the terminal and the server.

5.4.1. Terminal Sending Heartbeat Packet to Server

Format			Length (Byte)
Information Content	Start Bit		2
	Packet Length		1
	Protocol Number		1
	Status Information	Terminal Information Content	1
		Voltage Level	1
		GSM Signal Strength	1
		Alarm/Language	2
	Serial Number		2
	Error Check		2
	Stop Bit		2

5.4.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.4.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.4.1.3. Protocol Number

For details see Data Packet Format section 4.3.

5.4.1.4. Terminal Information

One byte is consumed defining for various status information of the mobile phone.

Bit		Code Meaning
BYTE	Bit7	1: oil and electricity disconnected
		0: gas oil and electricity
	Bit6	1: GPS tracking is on
		0: GPS tracking is off
	Bit3~ Bit5	100: SOS
		011: Low Battery Alarm
		010: Power Cut Alarm
		001: Shock Alarm
		000: Normal
	Bit2	1: Charge On
		0: Charge Off
	Bit1	1: ACC high
		0: ACC Low
	Bit0	1: Defense Activated
		0: Defense Deactivated

Example: 0x44, corresponding binary value is 01000100, indicates that the status of the terminal is: oil and electricity connected, GPS tracking is on, normal without any alarm, charge on, ACC is low,

and defense deactivated.

5.4.1.5. Voltage Level

The range is 0~6 defining the voltage is from low to high.

0: No Power (shutdown)

1: Extremely Low Battery (not enough for calling or sending text messages, etc.)

2: Very Low Battery (Low Battery Alarm)

3: Low Battery (can be used normally)

4: Medium

5: High

6: Very High

Example: 0x02 indicates very low battery and a Low Battery Alarm is sending.

5.4.1.6. GSM Signal Strength Levels

0x00: no signal;

0x01: extremely weak signal;

0x02: very weak signal;

0x03: good signal;

0x04: strong signal.

Example: 0x03 indicates the GSM signal is good.

5.4.1.7. Alarm/Language

0x00 (former bit) 0x01 (latter bit)

former bit: terminal alarm status (suitable for alarm packet and electronic fence project)

latter bit: the current language of the terminal

former bit	
latter bit	0x01: Chinese
	0x02: English

Examples:

No Alarm and Language is Chinese: 0x00 0x01

No Alarm and Language is English: 0x00 0x02

5.4.1.8. Information Serial Number

For details see Data Packet Format section 4.5.

5.4.1.9. Error Check

For details see Data Packet Format section 4.6.

5.4.1.10. Stop Bit

For details see Data Packet Format section 4.7.

5.4.2. Server Responds the Data Packet

	Description	Bits	Example
Login Message Packet (18 Byte)	Start Bit	2	0x78 0x78
	Packet Length	1	0x05
	Protocol Number	1	0x13
	Information Serial Number	2	0x00 0x11
	Error Check	2	0xF9 0x70
	Stop Bit	2	0x0D 0x0A

The response packet from the server to the terminal: the protocol number in the response packet is identical to the protocol number in the data packet sent by the terminal.

5.4.2.1. Start Bit

For details see Data Packet Format section 4.1.

5.4.2.2. Packet Length

For details see Data Packet Format section 4.2.

5.4.2.3. Protocol Number

For details see Data Packet Format section 4.3.

5.4.2.4. Information Serial Number

For details see Data Packet Format section 4.5.

5.4.2.5. Error Check

For details see Data Packet Format section 4.6.

5.4.2.6. Stop Bit

For details see Data Packet Format section 4.7.

5.4.3. Examples

Example of data packet sent by the terminal

78 78 08 13 4B 04 03 00 01 00 11 06 1F 0D 0A

Explain

0x78 0x78	0x0a	0x13	0x4B 0x04 0x03	0x00 0x01	0x00 0x11	0x06 0x1F	0x0D 0x0A
Start Bit	Length	Protocol No.	Information Content	Reserved bit (Language)	Serial No.	Error Check	Stop Bit

Example of response packet returned by the server

78 78 05 13 00 11 F9 70 0D 0A

Explain

0x78 0x78	0x05	0x13	0x00 0x11	0xF9 0x70	0x0D 0x0A
Start Bit	Length	Protocol No.	Serial No.	Error Check	Stop Bit

5.5. OBD Message(0x8C)

Format		Length (Byte)	Example
Information	Start Bit	2	0x79 0x79

Content	Packet Length	2	0x00 0x2A
	Protocol Number	1	0x8C
	Date Time(UTC)	6	0x0B 0x0B 0x0F 0x0E 0x24 0x1D
	ACC Status	1	0x01
	OBD Information	N	
	Quantity of GPS information satellites	1	0xCF
	Latitude	4	0x02 0x7A 0xC8 0x87
	Longitude	4	0x0C 0x46 0x57 0xE6
	Speed	1	0x00
	Course, Status	2	0x14 0x02
	Serial Number	2	0x00 0x36
	Error Check	2	0x2A 0x05
	Stop Bit	2	0x0D 0x0A

5.5.1.1. Start Bit

Fixed value in HEX 0x79 0x79.

5.5.1.2. Packet Length

Occupies 2 bytes.

5.5.1.3. Protocol Number

Protocol number 0x8C

5.5.1.4. Date Time(UTC)

For details see Location Data Packet Format section 5.2.1.4.

5.5.1.5. ACC Status

0x00: ACC Low; 0x01: ACC High

5.5.1.6. OBD Information(Using ASCII encoding method)

Format	Length	Instruction
Digital stream ID		OBD module ID (2 bit) . Check the following OBD digital stream ID table
Result		0:Cannot support, 1:CAN, 2:diagnosis, 3:OBD, 4:calculation from module
=		(=after the data needed)
OBD Data		OBD data
Delimiter		‘,’ Separate OBD data

For example, OBD is encoded as:

173=1, 223=1, 283=0005ae66, 463=000099e5, 473=0001ac33, 483=00001a25, 2B3=00000000, 2D3=00001b58, 353=000007d0, 363=0001c138, 393=000003fb

After converting to BIN:

3137333D312C3232333D312C3238333D3030303561653636362C3436333D3030303039396535 2C3437333D3030303161633333 2C3438333D3030303031613235 2C3242333D3030303030303030 2C3244333D3030303031623538 2C3335333D3030303030376430 2C3336333D3030303163313338 2C3339333D3030303033666 2

5.5.1.7. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.5.1.8. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.5.1.9. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.5.1.10. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.5.1.11. Status and Course

For details see Location Data Packet Format section 5.2.1.9.

5.5.1.12. Information Serial Number

For details see Data Packet Format section 4.5.

5.5.1.13. Error Check

For details see Data Packet Format section 4.6.

5.5.1.14. Stop Bit

For details see Data Packet Format section 4.7.

Example:

79 79 00 98 8C 11 0B 08 08 09 20 01 31 37 33 3D 31 2C 32 32 33 3D 31 2C 32 38 33 3D 30 30 30 35 61 65 36 36 2C 34 36 33 3D 30 30 30 30 39 39 65 35 2C 34 37 33 3D 30 30 30 31 61 63 33 33 2C 34 38 33 3D 30 30 30 30 31 61 32 35 2C 32 42 33 3D 30 30 30 30 30 30 30 30 2C 32 44 33 3D 30 30 30 30 31 62 35 38 2C 33 35 33 3D 30 30 30 30 30 37 64 30 2C 33 36 33 3D 30 30 30 31 63 31 33 38 2C 33 39 33 3D 30 30 30 30 30 33 66 62 C0 02 6B 4C 9C 0C 39 58 D2 00 04 57 0D C4 76 B3 0D 0A

OBD Digital Stream ID Table.

First into decimal, and then correspond to this table.

ID	Data stream description	Data stream length(Byte)	Data stream algorithm
23	MIL Fault signal(ECM)	1	1: Lock 0: Unlock
34	ACC Status	1	1: Valid 0:Invalid
40	Total mileage(KM)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
43	Remaining fuel(%)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
45	Engine coolant temperature(℃)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
53	Speed(KM/H)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
54	Engine Speed (rpm)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
57	Instantaneous fuel consumption(L/H)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
58	Instantaneous fuel consumption(L/100KM)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
70	Trip mileage(KM)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
71	Total fuel consumption(L)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8)

			| (X4)) /100;
72	Trip fuel consumption(L)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
73	Total Engine running time(H)	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100;
74	VIN Number	17	(signed) Y= Y
80	车辆体检	2	X2 的 BIT0: 0：发动机无故障，1：有 BIT1: 0：电源电路正常，1：异常 BIT2: 0：节气门开度正常，1：异常 BIT3:0：排放系统良好，1：异常 BIT4：0：冷却系统良好，1：异常 BIT5：0：怠速正常，1：过高 BIT6:0：怠速正常，1：过低 BIT7：0：正常，1：续航里程不足50KM

Example:
173=1, 223=1, 283=0005ae66, 463=000099e5, 473=0001ac33, 483=00001a25, 2B3=00000000, 2D3=00001b58, 353=000007d0, 363=0001c138, 393=000003fb

MIL Fault signal(ECM)（OBD）=1， ACC Status（OBD）=1， Total mileage（OBD）=3723.26 Trip mileage（OBD）=393.97, Total fuel consumption（OBD）=1096.19, Trip fuel consumption（OBD）=66.93, Remaining fuel（%）（OBD）=0, Engine coolant temperature（OBD）=70, Speed（OBD）=20， Engine Speed（OBD）=1150, Instantaneous fuel consumption（L/H）（OBD）=10.19

5.6. TPMS Message(0x8D)

Format		Length (Byte)	Example
Information Content	Start Bit	2	0x79 0x79
	Packet Length	2	0x00 0x2A
	Protocol Number	1	0x8D
	Date Time(UTC)	6	0x0B 0x0B 0x0F 0x0E 0x24 0x1D
	ACC Status	1	0x01
	TPMS Information	N	
	Quantity of GPS information satellites	1	0xCF
	Latitude	4	0x02 0x7A 0xC8 0x87
	Longitude	4	0x0C 0x46 0x57 0xE6
	Speed	1	0x00
	Course, Status	2	0x14 0x02
	Serial Number	2	0x00 0x36
	Error Check	2	0x2A 0x05
	Stop Bit	2	0x0D 0x0A

5.6.1.1. Start Bit

Fixed value in HEX 0x79 0x79.

5.6.1.2. Packet Length

Occupies 2 bytes.

5.6.1.3. Protocol Number

Protocol number 0x8D

5.6.1.4. Date Time(UTC)

For details see Location Data Packet Format section 5.2.1.4.

5.6.1.5. ACC Status

0x00: ACC Low; 0x01: ACC High

5.6.1.6. TPMS Information(Using ASCII encoding method)

Format	Length	Instruction
Tire ID		T1-T22, Maximum 22 tires
=		(=after the data needed)
Tire Pressure		Unit: Kpa
Delimiter		‘,’ Separate different data
Tire Temperature		Unit: Degree
Delimiter		‘,’ Separate different data
Tire Voltage		Unit: 0.1 V
Delimiter		‘,’ Separate different data
Tire Status		0: Normal, 1: Flat, 2: Low Voltage, 3: Low Pressure, 4: High Pressure, 5: High Temperature, 6: Lost Signal
Delimiter		‘;’ Separate different tire ID data

For example, TPMS is encoded as:

T1=210,88,13,0;T2=190,70,14,1;T3=200,78,14,0;T4=190,80,15,0

After converting to BIN:

54313D3231302C38382C31332C303B54323D3139302C37302C31342C313B54333D3230302C37382C31342C303B54343D3139302C38302C31352C30

5.6.1.7. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.6.1.8. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.6.1.9. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.6.1.10. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.6.1.11. Status and Course

For details see Location Data Packet Format section 5.2.1.9.

5.6.1.12. Information Serial Number

For details see Data Packet Format section 4.5.

5.6.1.13. Error Check

For details see Data Packet Format section 4.6.

5.6.1.14. Stop Bit

For details see Data Packet Format section 4.7.

5.7. DTC Message(0x65)

Format		Length (Byte)	Example
Information Content	Start Bit	2	0x78 0x78
	Packet Length	1	0x41
	Protocol Number	1	0x65
	Date Time(UTC)	6	11 0B 08 08 08 2E
	Vehicle Type	1	00
	DTC Number	1	0A
	DTC Code	4*N	
	Quantity of GPS information satellites	1	C0
	Latitude	4	02 6B 4C 9C
	Longitude	4	0C 39 58 D2
	Speed	1	00
	Course, Status	2	04 57
	Serial Number	2	0D B6
	Error Check	2	BB 11
	Stop Bit	2	0x0D 0x0A

5.7.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.7.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.7.1.3. Protocol Number

Protocol number 0x65

5.7.1.4. Date Time(UTC)

For details see Location Data Packet Format section 5.2.1.4.

5.7.1.5. Vehicle Type

0x00: Passenger Car; 0x01: Heavy Duty; 0x02 Tracker(N=0)

5.7.1.6. DTC Number

5.7.1.7. DTC Code

Every DTC is 4 BYTEs

For passenger car，reference 《passenger_ car_DTC _table.docx》

For passenger car, 4 BYTEs define as：<0x00,DTC_H,DTC_L,attr>

Attr: DTC type，

0x03: stored code, 0x07: pending code.

DTC_H is DTC code high 8bits and DTC_L is low 8bits， bit7bit6 of DTC_H is DTC class，b7b6=00: P, b7b6=01: C, b7b6=10: D, b7b6=11: U

example：0x00,0xc2,0x9a,0x03, means DTC code=U029A，and is stored code.

For heavy duty，reference《heavy_ duty _DTC_table.docx》

For heavy dutyr, 4 BYTEs define as：<DTC_H,DTC_L,attr，happens>

Bit7 of attris DTC type，=1b: stored code, =0b: pending code。

Happens: DTC happen frequency

DTC_H is DTC code high 8bits and DTC_L is low 8bits，
example：0x0f,0xcf,0x00,0x00,
means DTC code=0x0FCF，and is pending code.

5.7.1.8. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.7.1.9. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.7.1.10. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.7.1.11. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.7.1.12. Status and Course

For details see Location Data Packet Format section 5.2.1.9.

5.7.1.13. Information Serial Number

For details see Data Packet Format section 4.5.

5.7.1.14. Error Check

For details see Data Packet Format section 4.6.

5.7.1.15. Stop Bit

For details see Data Packet Format section 4.7.

Example:

78 78 41 65 11 0B 08 08 08 2E 00 0A 00 20 29 03 00 00 82 03 00 C0 21 03 00 80 90 03 00 40 32 03 00 00 06 03 00 00 07 03 00 12 33 03 00 00 09 03 00 C0 22 03 C0 02 6B 4C 9C 0C 39 58 D2 00 04 57 0D B6 BB 11 0D 0A

5.8. PID Message(0x66) (Using ASCII encoding method)

When trip on， device will upload PID message to server on time ， Upload interval can be set by command(TIMER,30,60,120#)

Format		Length (Byte)	Example
Information Content	Start Bit	2	0x79 0x79
	Packet Length	2	0x00 0x95
	Protocol Number	1	0x66
	Date Time(UTC)	6	0x11 0x0B 0x08 0x08 0x08 0x2E
	ACC Status	1	0x01
	PID Information	N	
	Quantity of GPS information satellites	1	0xC0
	Latitude	4	0x02 0x6B 0x4C 0x9C
	Longitude	4	0x0C 0x39 0x58 0xD2
	Speed	1	0x00
	Course, Status	2	0x04 0x57
	Serial Number	2	0x0D 0xB6
	Error Check	2	0xB3 0x33
	Stop Bit	2	0x0D 0x0A

5.8.1.1. Start Bit

Fixed value in HEX 0x79 0x79

5.8.1.2. Packet Length

Occupies 2 bytes.

5.8.1.3. Protocol Number

Protocol number 0x66

5.8.1.4. Date Time(UTC)

For details see Location Data Packet Format section 5.2.1.4.

5.8.1.5. PID Information

Format	Length	Instruction
PID code	2	PID module ID (2 byte) . Check the Heavy_Duty_PID_Table or Passenger_Car_PID_Table
=		(=after the data needed), 0x3D
PID Value	4	(signed) Y= ((X1<<24) | (X2<<16) | (X3<<8) | (X4)) /100; Unit follow the Heavy_Duty_PID_Table or Passenger_Car_PID_Table
Delimiter		‘,’ Separate OBD data, 0x2C

For example, PID is encoded as:
2110=0000B51A, 2105=000000D8, 210b=00000000, 210c=00000D49, 210d=00000052, 210f=000000D8, 2131=00000078, 2104=00000047, 212f=00000000
After converting to BIN:
323131303D303030304235314 12C323130353D303030303030443 82C323130623D303030303030303 02C323130633D303030303044343 92C323130643D303030303030353 22C323130663D303030303030 44382C323133313D303030303037382C323130343D303030303034372C323132663D303030303 0303030

The first PID Code=0x2110, PID data=0x0000B51A=463.62g/s

The Second PID Code=0x2015,PID data=0x000000D8=40degree

…

5.8.1.6. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.8.1.7. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.8.1.8. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.8.1.9. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.8.1.10. Status and Course

For details see Location Data Packet Format section 5.2.1.9.

5.8.1.11. Information Serial Number

For details see Data Packet Format section 4.5.

5.8.1.12. Error Check

For details see Data Packet Format section 4.6.

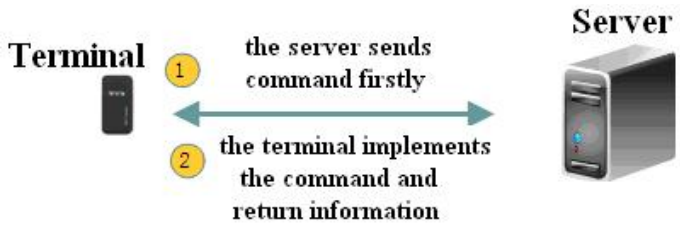
5.8.1.13. Stop Bit

For details see Data Packet Format section 4.7.

Example:

79 79 00 95 66 11 0B 08 08 08 2E 01 32 31 31 30 3D 30 30 30 30 42 35 31 41 2C 32 31 30 35 3D 30 30 30 30 30 30 44 38 2C 32 31 30 62 3D 30 30 30 30 30 30 30 30 2C 32 31 30 63 3D 30 30 30 30 30 44 34 39 2C 32 31 30 64 3D 30 30 30 30 30 30 35 32 2C 32 31 30 66 3D 30 30 30 30 30 30 44 38 2C 32 31 33 31 3D 30 30 30 30 30 30 37 38 2C 32 31 30 34 3D 30 30 30 30 30 30 34 37 2C 32 31 32 66 3D 30 30 30 30 30 30 30 30 C0 02 6B 4C 9C 0C 39 58 D2 00 04 57 0D B6 B3 33 0D 0A

Ⅵ. Data Packet Sent From Server to Terminal(0x80)



6.1. Packet Sent by Server(0x80)

Format		Length (Byte)	Example
Start Bit		2	0x78 0x78
Packet length		1	0x12
Protocol Number		1	0x80
Information Content	Length of Command	1	0x0C
	Server Flag Bit	4	0x00 0x00 0x00 0x01
	Command Content	M	0x56 0x45 0x52 0x53 0x49 0x4F 0x4E 0x23
Information Serial Number		2	0x00 0x01
Error Check		2	0x31 0x34
Stop Bit		2	0x0D 0x0A

6.1.1. Start Bit

For details see Data Packet Format section 4.1.

6.1.2. Packet Length

For details see Data Packet Format section 4.2.

6.1.3. Protocol Number

The Protocol Number of terminal transmission is 0x80.

6.1.4. Length of Command

Server Flag Bit + Length of Command Content

Example: measured in bytes, 0x0A means the content of command occupied ten bytes.

6.1.5. Server Flag Bit

It is reserved to the identification of the server. The binary data received by the terminal is returned without change.

6.1.6. Command Content

It is represented in ASC II of string, and the command content is compatible with text message command.

6.1.7. Information Serial Number

For details see Data Packet Format section 4.5.

6.1.8. Error Check

For details see Data Packet Format section 4.6.

6.1.9. Stop Bit

For details see Data Packet Format section 4.7.

6.2. Packet Replied by Terminal(0x21)

Format		Length (Byte)
Start Bit		2
Packet Length		2
Protocol Number		1
Information Content	Length of Command	1
	Server Flag Bit	4
	Command Content	M
Information Serial Number		2
Error Check		2
Stop Bit		2

6.2.1. Start Bit

Fixed value in HEX 0x79 0x79.

6.2.2. Packet Length

Occupies 2 bytes.

6.2.3. Protocol Number

The terminal responds to the command sent by the server.

6.2.4. Length of Command

Server Flag Bit + Length of Command Content

Example: measured in bytes, 0x0A means the content of command occupied ten bytes.

6.2.5. Server Flag Bit

It is reserved to the identification of the server. The binary data received by the terminal is returned without change.

6.2.6. Command Content

It is represented in ASC II of string, and the command content is compatible with text message command. The command content is same as SMS command.

6.2.7. Information Serial Number

For details see Data Packet Format section 4.5.

6.2.8. Error Check

For details see Data Packet Format section 4.6.

6.2.9. Stop Bit

For details see Data Packet Format section 4.7.

Example:

Sever send command to check the firmware version. The command is VERSION#

Send from server:

78 78 12 80 0C 00 00 00 01 56 45 52 53 49 4F 4E 23 00 01 31 34 0D 0A

Reply from device:

79 79 00 2A 21 00 00 00 01 01 4D 6F 64 65 6C 3A 53 54 32 47 2C 56 65 72 73 69 6F 6E 3A 53 54 32 47 31 37 38 4B 4B 31 31 30 33 01 B8 16 5C 0D 0A

Response content: Model:ST2G,Version:ST2G178KK1103

Ⅶ. Appendix A: code fragment of the CRC-ITU lookup table algorithm implemented based on C language

Code fragment of the CRC-ITU lookup table algorithm implemented based on C language is as follow:

```
static const U16 crctab16[] =
{
    0X0000, 0X1189, 0X2312, 0X329B, 0X4624, 0X57AD, 0X6536, 0X74BF,
    0X8C48, 0X9DC1, 0XAF5A, 0XBED3, 0XCA6C, 0XDBE5, 0XE97E, 0XF8F7,
    0X1081, 0X0108, 0X3393, 0X221A, 0X56A5, 0X472C, 0X75B7, 0X643E,
    0X9CC9, 0X8D40, 0XBFDB, 0XAE52, 0XDAED, 0XCB64, 0XF9FF, 0XE876,
    0X2102, 0X308B, 0X0210, 0X1399, 0X6726, 0X76AF, 0X4434, 0X55BD,
    0XAD4A, 0XBCC3, 0X8E58, 0X9FD1, 0XEB6E, 0XFAE7, 0XC87C, 0XD9F5,
    0X3183, 0X200A, 0X1291, 0X0318, 0X77A7, 0X662E, 0X54B5, 0X453C,
    0XBDCB, 0XAC42, 0X9ED9, 0X8F50, 0XFBEF, 0XEA66, 0XD8FD, 0XC974,
    0X4204, 0X538D, 0X6116, 0X709F, 0X0420, 0X15A9, 0X2732, 0X36BB,
    0XCE4C, 0XDFC5, 0XED5E, 0XFCD7, 0X8868, 0X99E1, 0XAB7A, 0XBAF3,
    0X5285, 0X430C, 0X7197, 0X601E, 0X14A1, 0X0528, 0X37B3, 0X263A,
    0XDECD, 0XCF44, 0XFDDF, 0XEC56, 0X98E9, 0X8960, 0XBBFB, 0XAA72,
    0X6306, 0X728F, 0X4014, 0X519D, 0X2522, 0X34AB, 0X0630, 0X17B9,
    0XEF4E, 0XFEC7, 0XCC5C, 0XDDD5, 0XA96A, 0XB8E3, 0X8A78, 0X9BF1,
    0X7387, 0X620E, 0X5095, 0X411C, 0X35A3, 0X242A, 0X16B1, 0X0738,
    0XFFCF, 0XEE46, 0XDCDD, 0XCD54, 0XB9EB, 0XA862, 0X9AF9, 0X8B70,
    0X8408, 0X9581, 0XA71A, 0XB693, 0XC22C, 0XD3A5, 0XE13E, 0XF0B7,
    0X0840, 0X19C9, 0X2B52, 0X3ADB, 0X4E64, 0X5FED, 0X6D76, 0X7CFF,
    0X9489, 0X8500, 0XB79B, 0XA612, 0XD2AD, 0XC324, 0XF1BF, 0XE036,
    0X18C1, 0X0948, 0X3BD3, 0X2A5A, 0X5EE5, 0X4F6C, 0X7DF7, 0X6C7E,
    0XA50A, 0XB483, 0X8618, 0X9791, 0XE32E, 0XF2A7, 0XC03C, 0XD1B5,
    0X2942, 0X38CB, 0X0A50, 0X1BD9, 0X6F66, 0X7EEF, 0X4C74, 0X5DFD,
    0XB58B, 0XA402, 0X9699, 0X8710, 0XF3AF, 0XE226, 0XD0BD, 0XC134,
    0X39C3, 0X284A, 0X1AD1, 0X0B58, 0X7FE7, 0X6E6E, 0X5CF5, 0X4D7C,
    0XC60C, 0XD785, 0XE51E, 0XF497, 0X8028, 0X91A1, 0XA33A, 0XB2B3,
    0X4A44, 0X5BCD, 0X6956, 0X78DF, 0X0C60, 0X1DE9, 0X2F72, 0X3EFB,
    0XD68D, 0XC704, 0XF59F, 0XE416, 0X90A9, 0X8120, 0XB3BB, 0XA232,
    0X5AC5, 0X4B4C, 0X79D7, 0X685E, 0X1CE1, 0X0D68, 0X3FF3, 0X2E7A,
    0XE70E, 0XF687, 0XC41C, 0XD595, 0XA12A, 0XB0A3, 0X8238, 0X93B1,
    0X6B46, 0X7ACF, 0X4854, 0X59DD, 0X2D62, 0X3CEB, 0X0E70, 0X1FF9,
    0XF78F, 0XE606, 0XD49D, 0XC514, 0XB1AB, 0XA022, 0X92B9, 0X8330,
    0X7BC7, 0X6A4E, 0X58D5, 0X495C, 0X3DE3, 0X2C6A, 0X1EF1, 0X0F78,
};

//  calculate the 16-bit CRC of data with predetermined length.
U16 GetCrc16(const U8* pData, int nLength)
{
    U16  fcs  =  0xffff;            //  initialization
    while(nLength>0){
        fcs = (fcs >> 8) ^ crctab16[(fcs ^ *pData) & 0xff];
        nLength--;
        pData++;
    }
    return  ~fcs;           //  negated
}
```