GPS Tracker
Communication Protocol
(T04S, T4-1C)

CONTENT

I. Communication Protocol 3

II. Terms, Definitions 3

III. Basic Rules 4

IV. Data Packet Format 6

4.1. Start Bit 6
4.2. Packet Length 6
4.3. Protocol Number 6
4.4. Information Contents 6
4.5. Information Serial Number 6
4.6. Error Check 6
4.7. Stop Bit 7

V. Details about Data Packet sent by Server to Terminal 8

5.1. Login Message Packet(0x01) 8
5.1.1. Terminal Sending Data Packet to Server 8
5.1.2. Server Responds the Data Packet 9
5.2. Location Data Packet (0x12) 10
5.2.1. Terminal Sending Location Data Packet to Server 10
5.2.2. Examples of Packet Sent from Terminal to Server 13
5.2.3. Server Location Package Reply 13
5.3. Location Data Packet (0x22) 14
5.3.1. Terminal Sending Location Packet to Server 14
5.3.2. Server Location Package Reply 14
5.4. Heartbeat Packet (0x13) 15
5.4.1. Terminal Sending Heartbeat Packet to Server 15
5.4.2. Server Responds the Data Packet 16
5.5. Alarm Packet(0x16) 18
5.5.1. Server Sending Alarm Data Packet to Server 18
5.5.2. Server Responds the Data Packet 20

5.6. Time Sync Package(0x8A) 21
5.6.1. Terminal sends time sync request 21
5.6.2. Server Response Time Sync 21
5.7. Information Transfer Common Package(0x94) 22
5.7.1. Terminal send information transmission common package 22
5.7.2. Server Response 22
VI. Data Packet Sent From Server to Terminal(0x80) 23
6.1. Packet Sent by Server 23
6.2. Packet Replied by Terminal(0x15) 24
Appendix A: code fragment of the CRC-ITU lookup table algorithm implemented based on C language 25

I. Communication Protocol

Introduction

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

II. 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

III. 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.

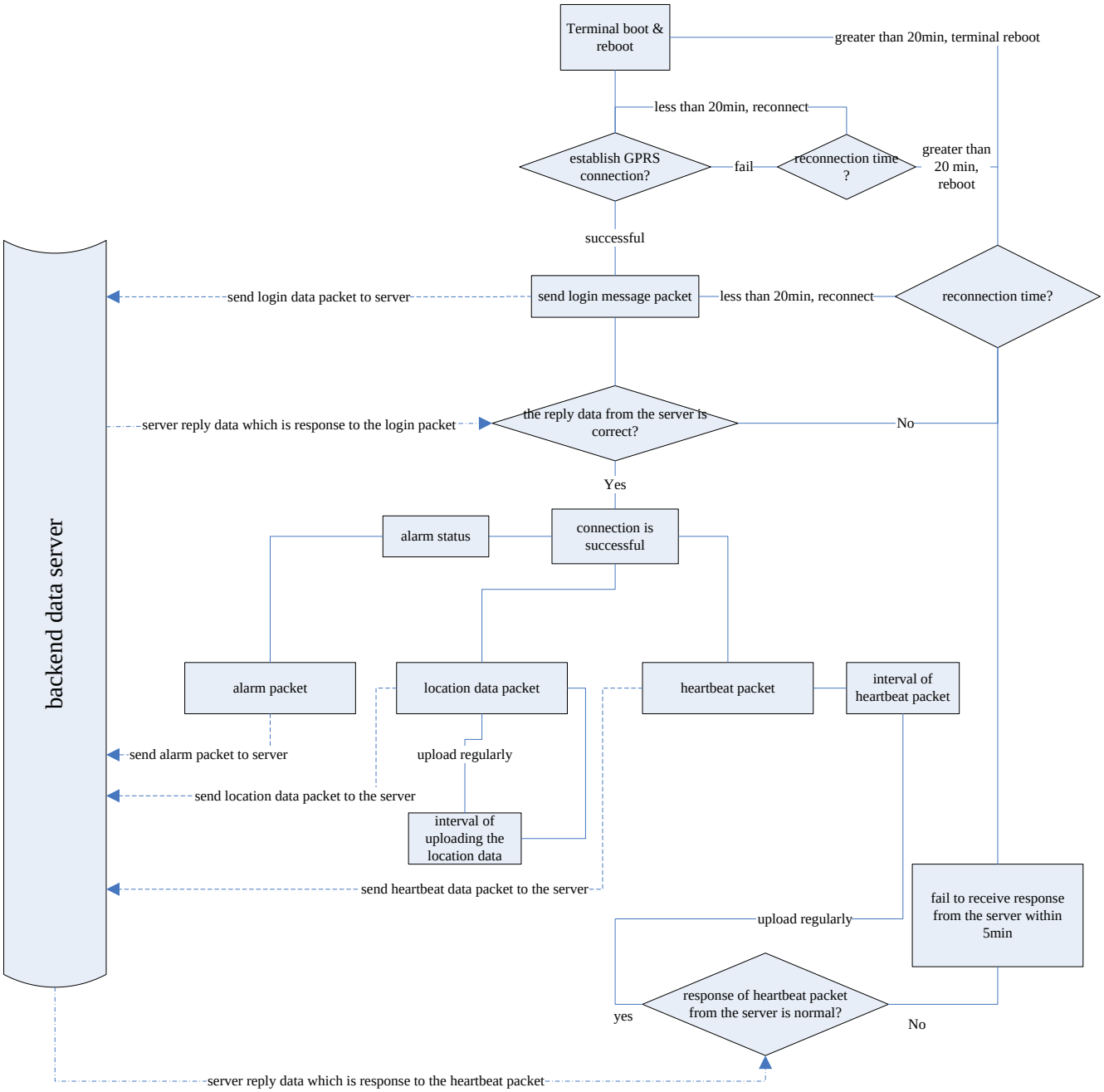
4. If the connection is regarded to be abnormal, and the data packet as a response from the server is failed to be received three times after a connection is established and a login message packet or status information package is sent, the terminal will start schedule reboot and the scheduled time is 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 Data	0x12
Location Data(Mileage data)	0x22
Status information	0x13
String information	0x15
Alarm data	0x16
Command information sent by the server to the terminal	0x80
Timing package	0x8A
Information transfer data	0x94
Command information sent by the server to the terminal	0x80

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(0x01)

5.1.1. Terminal Sending Data Packet to Server

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.

Format		Length (Byte)	Example
Login Message Packet(18 Byte)	Start Bit	2	0x78 0x78
	Packet Length	1	0x0D
	Protocol Number	1	0x01
	Terminal ID	8	0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45
	Information Serial Number	2	0x00 0x01
	Error Check	2	0x8C 0xDD
	Stop Bit	2	0x0D 0x0A

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. Information Serial Number

For details see Data Packet Format section 4.5.

5.1.1.6. Error Check

For details see Data Packet Format section 4.6.

5.1.1.7. Stop Bit

For details see Data Packet Format section 4.7

5.1.2. Server Responds the Data Packet

Format		Length (Byte)	Example
Login Message Packet (10Byte)	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 (0x12)

5.2.1. Terminal Sending Location Data Packet to Server

Format			Length (Byte)	Example
Information Content	Start Bit		2	0x78 0x78
	Packet Length		1	0x1F
	Protocol Number		1	0x12
	GPS Information	Date Time	6	0x13 0x08 0x1D 0x11 0x0C 0x10
		GPS information	1	0xCB
		Latitude	4	0x02 0x7A 0xCF 0xEB
		Longitude	4	0xCC 0x46 0x58 0x49
		Speed	1	0x10
		Course, Status	2	0x15 0x4C
	LBS Information	MCC	2	0x01 0xCC
		MNC	1	0x00
		LAC	2	0x28 0x7D
		Cell ID	3	0x00 0x1F 0xB8
	Serial Number		2	0x00 0x03
	Error Check		2	0x9D 0xDC
	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	0x13
Month	1	0x08
Day	1	0x1D
Hour	1	0x11
Minute	1	0x0C
Second	1	0x10

Example: 2019-08-29 17:12:16

Calculated as follows: 19(Decimal)=13(Hexadecimal)
08 (Decimal)=08(Hexadecimal)
29(Decimal)=1D(Hexadecimal)
17(Decimal)=11(Hexadecimal)
12(Decimal)=0C(Hexadecimal)
16(Decimal)=10(Hexadecimal)

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

The date and time of the GPS location pack is the time zone set by the device.

5.2.1.5. Length of GPS information, quantity of positioning satellites

1Byte converted into binary is 8Bit, the first 4 bits are the length of GPS information, the last 4 bits are the current number of satellites of the device, and the maximum number of satellites is 15.

Note: The length includes 1Byte occupied by itself.

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

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)X3000=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 and Status

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	No definition
	Bit6	No definition
	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

Note: 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 (No definition)
BYTE_1 Bit6 0 (No definition)
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 Bit7 1
BYTE_2 Bit7 0
BYTE_2 Bit7 0
BYTE_2 Bit7 1
BYTE_2 Bit7 1
BYTE_2 Bit7 0
BYTE_2 Bit7 0

→ Course 332° (0101001100 in Binary, or 332 in decimal)

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 value ranges from 0x000000 to 0xFFFFFF.

5.2.1.14. Information Serial Number

For details see Data Packet Format section 4.5.

5.2.1.15. Error Check

For details see Data Packet Format section 4.6.

5.2.1.16. 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 1F 12 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 00 03 80 81 0D 0A

Explain

0x78 0x78	0x1F	0x12	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	0x00 0x03
Longitude	Speed	Course Status	MCC	MNC	LAC	Cell ID	Serial No.

0x80 0x81	0x0D 0x0A
Error Check	Stop Bit

Note: The 0x12 GPS / LBS information packet will be saved when the GSM signal is abnormal, and then re-transmitted to the platform; the re-transmission method is not real-time upload, but after the device is stationary, upload the reserved positioning points.

Track playback should be based on the time in the positioning package. For real-time display, the positioning package time should be 5 minutes below the current time.

5.2.3. Server Location Package Reply

Location packets do not require a reply

5.3. Location Data Packet (0x22)

5.3.1. Terminal Sending Location Packet to Server

Format		Length (Byte)	Example
Start Bit		2	0x78 0x78
Packet Length		1	0x26
Protocol Number		1	0x22 (UTC)
Information Content	Date Time	6	0x13 0x08 0x1D 0x11 0x0C 0x10
	GPS information	1	0xCB
	Latitude	4	0x02 0x7A 0xCF 0xEB
	Longitude	4	0xCC 0x46 0x58 0x49
	Speed	1	0x10
	Course, Status	2	0x15 0x4C
	MCC	2	0x01 0xCC
	MNC	1	0x00
	LAC	2	0x28 0x7D
	Cell ID	3	0x00 0x1F 0xB8
	ACC	1	0x01
	Data reporting mode	1	0x00
	GPS Real-time Supplementary	1	0x00
	Mileage statistics	4	0x00 0x01 0x45 0x42
Serial Number		2	0x00 0x03
Error Check		2	0x9D 0xDC
Stop Bit		2	0x0D 0x0A

5.3.1.1. ACC

ACC status (0x00 for ACC off, 0x01 for ACC on)

5.3.1.2. Data reporting mode

0x00 Timed reporting

5.3.1.3. GPS Real-time / Supplementary

0x00: Real-time upload, 0x01: Supplementary upload.

5.3.1.4. Mileage statistics

Convert to decimal to get the result (some products have this function, no such function without this position)

5.3.2. Server Location Package Reply

Location packets do not require a reply

5.4. Heartbeat 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
		Extended data	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

Occupies 1 byte, converted to binary, used to represent the status information of the terminal.

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
	Bit5~ Bit3	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: Activated
		0: Deactivated

5.4.1.5. Voltage Level

The arrange 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

5.4.1.6. GSM Signal Strength Levels

0x00: no signal.

0x01: Very weak signal

0x02: Weak signal

0x03: good signal

0x04: strong signal

5.4.1.7. Extended data (language)

Terminal current language bit:

Chinese: 0x00 0x01;

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

Format		Length (Byte)	Example
Information Content	Start Bit	2	0x78 0x78
	Packet Length	1	0x05
	Protocol Number	1	0x13
	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.

5.4.2.1. 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.

5.5. Alarm Packet(0x16)

5.5.1. Server Sending Alarm Data Packet to Server

Format			Length (Byte)	Example
Informsation Content	Start Bit		2	0x78 0x78
	Packet Length		1	0x 25
	Protocol Number		1	0x 16
	Date Time		6	0x0B 0x08 0x1D 0x11 0x2E 0x10
	GPS Information	Quantity of GPS information satellites	1	0x C0
		Latitude	4	0xCF
		Longitude	4	0x04 0x36 0x6B 0x34
		Speed	1	0x0C 0x46 0x58 0x49
		Course, Status	2	0x00
	LBS Information	LBS Length	1	0x2e
		MCC	2	0x01 0xCC
		MNC	1	0x00
		LAC	2	0x28 0x7D
		Cell ID	3	0x00 0x1F 0xB8
	status Information	Terminal Information Content	1	0x38
		Voltage Level	1	0x0c
		GSM Signal Strength	1	0x04
		Alarm information	2	0x00
	Serial Number		2	0x00 0x08
	Error Check		2	0x2a 0x8b
	Stop Bit		2	0x0d 0x0a

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

5.5.1.1. Start Bit

For details see Data Packet Format section 4.1.

5.5.1.2. Packet Length

For details see Data Packet Format section 4.2.

5.5.1.3. Protocol Number

For details see Data Packet Format section 4.3.

5.5.1.4. Date Time

For details see Location Data Packet Format section 5.2.1.4.

5.5.1.5. Length of GPS information, quantity of positioning satellites

For details see Location Data Packet Format section 5.2.1.5.

5.5.1.6. Latitude

For details see Location Data Packet Format section 5.2.1.6.

5.5.1.7. Longitude

For details see Location Data Packet Format section 5.2.1.7.

5.5.1.8. Speed

For details see Location Data Packet Format section 5.2.1.8.

5.5.1.9. Course and Status

For details see Location Data Packet Format section 5.2.1.9.

5.5.1.10. LBS Length

The length of LBS information, Length is: 0x09

5.5.1.11. MCC

For details see Location Data Packet Format section 5.2.1.10.

5.5.1.12. MNC

For details see Location Data Packet Format section 5.2.1.11.

5.5.1.13. LAC

For details see Location Data Packet Format section 5.2.1.12.

5.5.1.14. Cell ID

For details see Location Data Packet Format section 5.2.1.13.

5.5.1.15. 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
	Bit5~ Bit3	100: SOS
		011: Low Battery Alarm
		010: Power Cut Alarm
		001: Vibration Alarm
		000: Normal
	Bit2	1: Charge On
		0: Charge Off
	Bit1	1: ACC high
		0: ACC Low
	Bit0	1: Activated
		0: 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 deactivated.

5.5.1.16. Voltage Level

The arrange 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

5.5.1.17. GSM Signal Strength Levels

0x00: no signal.

0x01: Very weak signal

0x02: Weak signal

0x03: good signal

0x04: strong signal

5.5.1.18. Alarm Information

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

latter bit: the current language used in the terminal

Byte1(Alarm)	0x00: normal
	0x01: SOS
	0x02: Power Cut Alarm
	0x03: Vibration Alarm
	0x04: Enter fence alarm
	0x05: Exit fence Alarm
	0x06: Over speed alarm
	0x09: Displacement alarm
	0x0E: Power low alarm
	0x13: Disassemble alarm
	0xFE: ACC On alarm
	0xFF: ACC Off alarm
Byte2(Language)	0x01:Chinese
	0x02:English

5.5.1.19. Information Serial Number

For details see Data Packet Format section 4.5.

5.5.1.20. Error Check

For details see Data Packet Format section 4.6.

5.5.1.21. Stop Bit

For details see Data Packet Format section 4.7.

5.5.2. Server Responds the Data Packet

Format		Length (Byte)	Example
Information Content	Start Bit	2	0x78 0x78
	Packet Length	1	0x05
	Protocol Number	1	0x16
	Information Serial Number	2	0x00 0x01
	Error Check	2	0xD9 0xDC
	Stop Bit	2	0x0D 0x0A

5.6. Time Sync Package(0x8A)

It is used to automatically request time sync from the server when the terminal is powered on, so as to solve the problem of time error when the boot is not located.

The server replies with the correct time and format, the time is UTC time

5.6.1. Terminal sends time sync request

Format	Length	Example
Start Bit	2	0x78 0x78
Packet Length	1	0x05
Protocol Number	1	0x8A
Information Serial Number	2	0x00 0x06
Error Check	2	0x88 0x29
Stop Bit	2	0x0D 0x0A

5.6.2. Server Response Time Sync

Format		Length	Example
Start Bit		2	0x78 0x78
Packet Length		1	0x0B
Protocol Number		1	0x8A(UTC)
Information Content	Date time	6	0x0F 0x0C 0x1D 0x00 0x00 0x15
Information Serial Number		2	0x00 0x06
Error Check		2	0xF0 0x86
Stop Bit		2	0x0D 0x0A

5.7. Information Transfer Common Package(0x94)

Used by terminals to transmit various types of non-location data

5.7.1. Terminal send information transmission common package

Format		Length	Example
Start Bit		2	0x789 0x79
Packet Length		2	
Protocol Number		1	0x94
Information Content	Type of information (sub-protocol number)	1	00 External voltage 0A ICCID
	Data content	N	Different transmission contents depending on the type of information
Information Serial Number		2	0x00 0x06
Error Check		2	0xF0 0x86
Stop Bit		2	0x0D 0x0A

Example:
7979007F9404414C4D313D43343B414C4D323D43433B414C4D333D34433B535441313D43303B4459443D30313B534F533D2C2C3B43454E5445523D3B46454E43453D46656E63652C4F4E2C302C32332E3131313830392C3131342E3430393236342C3430302C494E206F72204F55542C303B4D4946493D4D4946492C4F4646000A061E0D0A

5.7.1.1. Data content

When the type is 00, this bit transmits the external power voltage, which is two hexadecimal numbers, and divides by 100 after converting hexadecimal to decimal, for example.

0X04,0X9F,049F converted to decimal is 1183, divided by 100 is 11.83, which means the value of external voltage of terminal is 11.83V at this time.

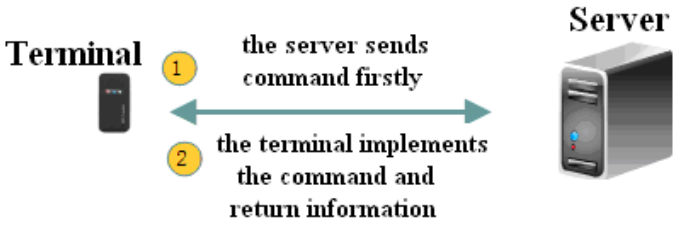
When the type is 0A, the ICCID information is transmitted, and the transmission is a hexadecimal number

IMEI	8	Example: The IMEI number is 123456789123456, then the terminal ID is: 0x010x230x450x670x890x120x340x56
IMSI	8	Example: The IMSI number is 123456789123456, then the terminal ID is: 0x010x230x450x670x890x120x340x56
ICCID	10	Example: The ICCID number is 12345123456789123456, then the terminal ID is: 0X120x340x510x230x450x670x890x120x340x56

5.7.2. Server Response

Server does not need to reply

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



6.1. Packet Sent by Server

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

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

Length of Command = Server Flag Bit +Command Content

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(0x15)

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

6.2.1. Start Bit

For details see Data Packet Format section 4.1.

6.2.2. Packet Length

For details see Data Packet Format section 4.2.

6.2.3. Protocol Number

The terminal responds to the command sent by the server. The format of data packet is consistent with “the command sent by the server to the terminal”, but the Protocol Number herein is different and is 0x15

6.2.4. Length of Command

Length of Command = Server Flag Bit +Command Content

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.

6.2.7. Language

Chinese: 0x00 0x01

English: 0x00 0x02

6.2.8. Information Serial Number

For details see Data Packet Format section 4.5.

6.2.9. Error Check

For details see Data Packet Format section 4.6.

6.2.10. Stop Bit

For details see Data Packet Format section 4.7.

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
}
```