Protocolo de Sucre Lee

SL42/SL44/SL48

Catálogo

1. Protocolo de comunicação

Este documento define a descrição do protocolo de interface da camada de aplicação da plataforma de serviço de posicionamento do rastreador GPS veicular. Os protocolos de interface relevantes são aplicáveis apenas à interação entre a plataforma e o terminal de posicionamento.

2. Termos e Definições

termo, abreviação	Significado Inglês	Significado português
CMPP	China Mobile Peer to Peer	Ponto para ponto móvel da China
GPS	Global Positioning System	Sistema de Posicionamento Global
GSM	Global System for Mobile Communication	Sistema Global para Comunicações Móveis
GPRS	General Packet Radio Service	Serviço Geral de Rádio por Pacote
TCP	Transport Control Protocol	Protocolo de Controle de Transmissão
LBS	Location Based Services	Serviços baseados em localização
IMEI	International Mobile Equipment Identity	Identidade Internacional de Equipamento Móvel
MCC	Mobile Country Code	Código de país móvel
MNC	Mobile Network Code	Código de rede móvel
LAC	Location Area Code	Código de área do local

Cell ID	Cell Tower ID	ID da torre de celular
UDP	User Datagram Protocol	Protocolo de datagrama do usuário
SOS	Save Our Ship/Save Our Souls	Salve Nosso Navio/Salve Nossas Almas (AJUDA)
CRC	Cyclic Redundancy Check	Verificação Cíclica de redundância
NITZ	Network Identity and Time Zone,	Identidade de rede e fuso horário
GIS	Geographic Information System	Sistema de Informações Geográficas

3. Regras básicas

1. A conexão GPRS é estabelecida com sucesso e o primeiro pacote de informações de login é enviado ao servidor: Depois de receber o pacote de resposta do servidor em 5 segundos a conexão é considerada normal e as informações de posicionamento (GPS, pacote de informações LBS) começam a ser enviadas。 as informações de status serão enviadas após 3 minutos. confirme regularmente a comunicação normal.

2. A conexão GPRS não é estabelecida com sucesso, o terminal não pode enviar o pacote de informações de login: Quando a conexão GPRS falha 3 vezes consecutivas, o terminal inicia a função de reinicialização no tempo de 20 minutos. Se o terminal estabelecer com sucesso uma conexão com o servidor dentro de 20 minutos e receber um pacote de dados que o servidor responde ao pacote de informações de login enviado pelo terminal, a função de reinicialização agendada será desabilitada e o terminal não reiniciará, caso contrário, o terminal reiniciará automaticamente após 20 minutos.

3. O servidor retorna um pacote de resposta ao terminal após receber o pacote de informações de login enviado pelo terminal: Se o terminal não receber o pacote de retorno do servidor por mais de 5 segundos após o envio do pacote de informações de login ou pacote de informações de status, ele considere que a conexão atual é anormal e inicie a função de transmissão suplementar de dados de posicionamento GPS, desconecte a conexão GPRS atual, restabeleça uma nova conexão GPRS e envie o pacote de informações de login.

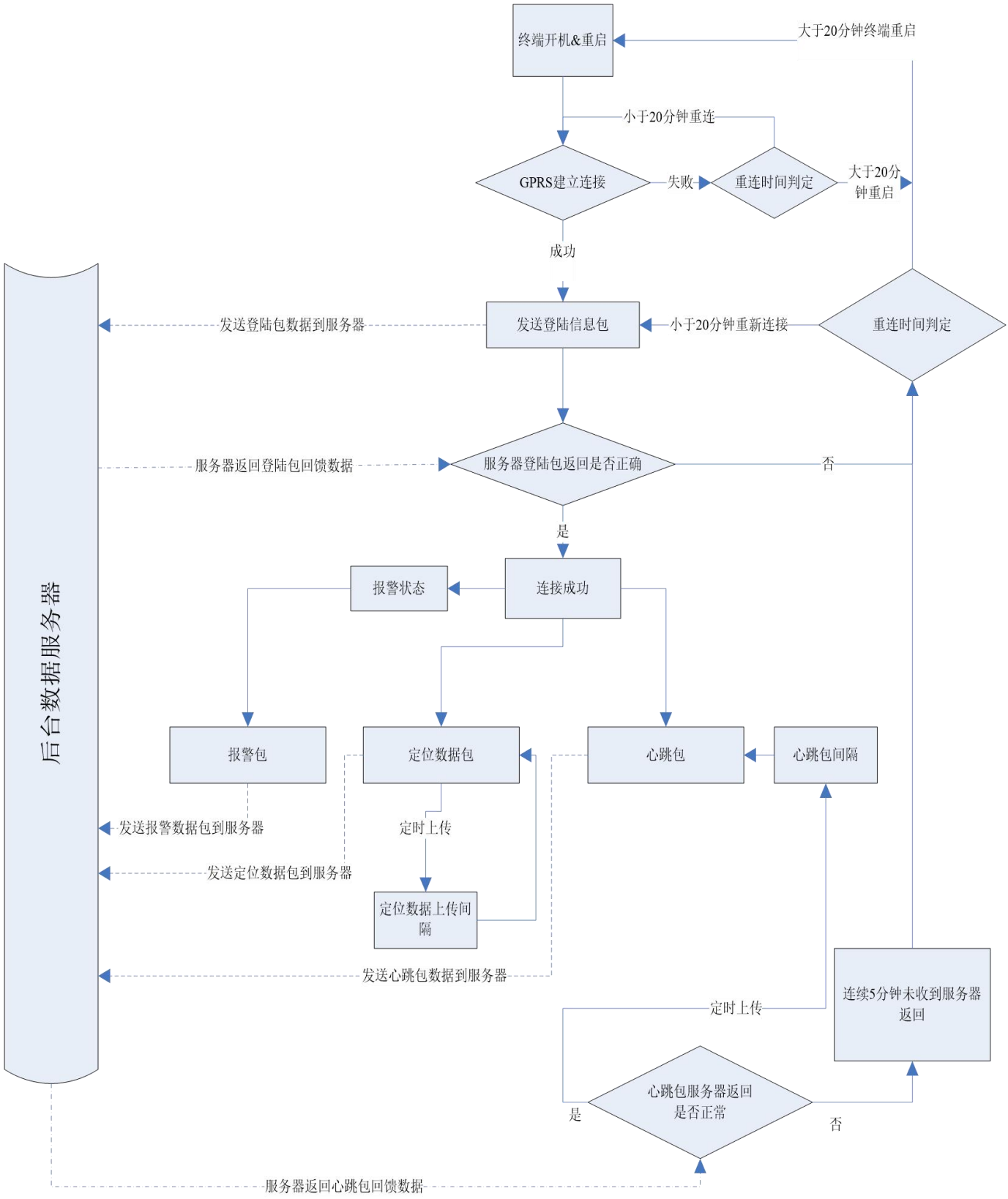
4. A conexão é considerada anormal e o pacote de informações de login ou pacote de informações de status enviado após a conexão ser estabelecida repetidamente por 3 vezes não pode receber o pacote de dados respondido pelo servidor: O terminal inicia a função de reinício do temporizador de 10 minutos. Se o terminal estabelecer com sucesso uma conexão com o servidor e receber um pacote de dados do servidor, a função de reinicialização agendada será desativada e o terminal não será reiniciado, caso contrário, o terminal será reiniciado automaticamente após 10 minutos.

5. Depois que a conexão for estabelecida normalmente, o terminal enviará regularmente pacotes de informações combinadas de GPS e LBS para o servidor após as alterações nas informações do GPS, e o servidor poderá definir o protocolo de envio padrão por meio de instruções.

6. Para garantir a validade da conexão, as informações de status são enviadas ao servidor em intervalos fixos e o servidor retorna um pacote de resposta para confirmação.

7. Para terminais que não possuem números IMEI registrados, o servidor deve responder com uma resposta de solicitação de login e uma resposta de pacote de pulsação e não desconectar diretamente. (Se você desconectar diretamente ou não responder, isso fará com que o terminal seja reconectado continuamente e o consumo de tráfego GPRS será grave).

Diagrama de fluxo de dados


终端开机&重启
大于20分钟终端重启
小于20分钟重连
GPRS建立连接
失败
重连时间判定
大于20分钟重启
成功
发送登陆包数据到服务器
发送登陆信息包
小于20分钟重新连接
重连时间判定
服务器返回登陆包回馈数据
服务器登陆包返回是否正确
否
是
后台数据服务器
报警状态
连接成功
报警包
定位数据包
心跳包
心跳包间隔
发送报警数据包到服务器
定时上传
发送定位数据包到服务器
定位数据上传间隔
发送心跳包数据到服务器
连续5分钟未收到服务器返回
定时上传
心跳包服务器返回是否正常
是
否
服务器返回心跳包回馈数据

4. data pack format

Communication transfers are asynchronous and in bytes.

total packet length: （10+N）Byte

Format	length(Byte)
start bit	2
packet length	1(2)
agreement number	1
information	N
message serial number	2
error checking	2
Stop bit	2

4.1. start bit

4.2. packet length

Fixed value, unified as Hexadecimal0x78 0x78 (packet length1 bit) or 0x79 0x79 (packet length2 bits)

4.3. agreement number

length=agreement number+information+message serial number+error checking,

type	value
login information	0x01
location data（UTC）	0x31
status information	0x13
string information	0x21
LBS information（UTC）	0x34
Alarm data（UTC）	0x32
GPS, phone number query address information（UTC）	0x2A
WIFI data package	0x33
The server sends instruction information to the terminal	0x80
General Packet for Information Transfer	0x94
large file transfer	0x8D

4.4. information

According to different applications, corresponding to the respective "agreement number", determine the specific content.

4.5. message serial number

The serial number of the first GPRS data (package including status packet and GPS, LBS, etc. data pack) sent after booting is '1', and the serial number of each subsequent data transmission (package including status packet and GPS, LBS data pack) is automatically incremented by 1.

4.6. error checking

The terminal or server can use the check code to judge whether the received information is incorrect. In order to prevent data from being corrupted during transmission, error checking is added to prevent data misoperation, which increases the security and efficiency of the system. The error checking code

adopts the CRC-ITU verification method.

The CRC-ITU value of the data from "packet length" to "message serial number" (including "packet length" and "message serial number") in the protocol body. If there is a CRC error in the calculation of the received information, the receiver ignores it and discards this data pack.

4.7. Stop bit

Fixed value, unified as Hexadecimal0x0D 0x0A.

5 The terminal sends a detailed data pack explanation to the server.

Separately explain common information packet sending and server return.

5.1 login information package

The terminal sends to the server data pack

The login information package is used to confirm to the server that the connection is established as Normal, and to submit the Terminal ID to the server.

Format		length	example
login information package （18 Byte）	start bit	2	0x78 0x78
	packet length	1	0x11
	agreement number	1	0x01
	Terminal ID	8	0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45
	Type ID	2	0x00 0x00
	time zone language	2	0x32 0x00
	message serial number	2	0x00 0x01
	error checking	2	0x90 0x01
	Stop bit	2	0x0d 0X0a

start bit

see data package format 4.1

packet length

see data package format 4.2

agreement number

see data package format 4.3

Terminal ID

For example: 123456789012345,

Then the Terminal ID is: 0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45

Type ID

Type ID occupies two bytes. Determine the terminal type based on this identification code.

SL22/SL24/SL28/EV02 0x00 0x00

SL22/SL44/SL48 0x28 0x01
SLXX 0x29 0x01

time zone language

<table>
<tr><td rowspan="10">One and a half byte bit15—bit4</td><td>15</td><td colspan="2" rowspan="10">Time zone expansion with a value of 100</td></tr>
<tr><td>14</td></tr>
<tr><td>13</td></tr>
<tr><td>12</td></tr>
<tr><td>11</td></tr>
<tr><td>10</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td rowspan="4">Lower half byte bit4-bit0</td><td>3</td><td colspan="2">east and west time zone</td></tr>
<tr><td>2</td><td colspan="2">Not yet defined</td></tr>
<tr><td>1</td><td>language selection bits</td><td>1</td></tr>
<tr><td>0</td><td>language selection bits</td><td>0</td></tr>
</table>

Bit3 0-------Eastern time zone
1------West time zone
If: extension bit: 0X32 0X00 means East Eight District, GMT+8: 00.
Calculation method: 8*100=800, converted to Hexadecimal, 0X0320

Extended bits: 0X4D 0XD8 means West 12th District and 3/4 time zone, GMT-12: 45.
Calculation method: 12.45*100=1245, turn to Hexadecimal, 0X04, 0XDD.

The algorithm here is to rotate the calculated time zone value to the left by four bits and then combine the east and west of the time zone and the language selection bit to save four bytes.

message serial number
see data package format 4.5

error checking
see data package format 4.6

Stop bit
see data package format 4.7

server response data pack

illustrate		length	example
login information package（18 Byte）	start bit	2	0x78 0x78
	packet length	1	0x05
	agreement number	1	0x01
	message serial number	2	0x00 0x01
	error checking	2	0xd9 0xdc
	Stop bit	2	0x0d 0x0a

The server responds to the terminal with a packet: (the agreement number in the response packet is the same as the data package agreement number sent by the terminal)**start bit**
see data package format 4.1

packet length
see data package format 4.2

agreement number
see data package format 4.3
message serial number
see data package format 4.5

error checking
see data package format 4.6

Stop bit
see data package format 4.7

5.2.location data package (GPS, LBS combined information package)

5.2.1 The terminal sends to the server general location data package

Format			length(Byte)
information	start bit		2
	packet length		1
	agreement number		1
	GPS information	date time	6
		GPS information satellite number	1
		latitude	4
		longitude	4
		speed	1
		Direction, status	2
	LBS information	MCC	2
		MNC	2
		LAC	2

		Cell ID	4
	ACC		1
	Data reporting mode		1
	GPS real-time supplementary transmission		1
	Mileage		4
	serial number		2
	error checking		2
	end bit		2

5.2.1.1 start bit

see data package format 4.1

5.2.1.2 packet length

see data package format 4.2

5.2.1.3 agreement number

fixed as0x31，see data package format 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

Such as: 2010Year 3Month 23Day 15Hour 50Minute 23Second

Calculation method: 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 GPS information length, the number of satellites involved in positioning

1 Byte has two Hexadecimal characters displayed, the first character is GPS information length, the second character is the number of participating satellites

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

(C = 12Bitlength, B = 11 Satellites)

5.2.1.6 latitude

Occupies 4 bytes, indicating the latitude value of location data. The value ranges from 0 to

162000000, indicating the range from 0 degrees to 90 degrees. The conversion method is as follows:

Convert the latitude value output by the GPS module into a decimal in minutes; then multiply the converted decimal by 30000, and convert the multiplied result into a Hexadecimal number.

For example 22º 32.7658'=(22X60+32.7658)X30000=40582974, and then converted to Hexadecimal number:

40582974 (Decimal) = 26B3F3E (Hexadecimal)

The final value is 0x02 0x6B 0x3F 0x3E.

5.2.1.7 longitude

It occupies 4 bytes and represents the longitude value of the location data. The value ranges from 0 to 324000000, representing the range from 0 degrees to 180 degrees.

The conversion method is the same as that of latitude

5.2.1.8 speed

Occupies 1 byte, indicating the running speed of GPS, the value range is 0x00～0xFF, indicating the range is 0～255 km/h.

For example: 0x00 represents 0 km/h.

0x10 means 16 km/h

0xFF stands for 255 km/h

5.2.1.9 State Direction

takes 2 bytes, indicating the running direction of GPS, indicating the range of 0 to 360 degree, taking true north as 0 degree, clockwise.

<table>
<tr><td rowspan="8">BYTE_1</td><td>Bit7</td><td>0</td></tr>
<tr><td>Bit6</td><td>0</td></tr>
<tr><td>Bit5</td><td>GPS Real-time/differential positioning</td></tr>
<tr><td>Bit4</td><td>GPS is located or not</td></tr>
<tr><td>Bit3</td><td>East Longitude, West Longitude</td></tr>
<tr><td>Bit2</td><td>south latitude, north latitude</td></tr>
<tr><td>Bit1</td><td rowspan="10">Direction</td></tr>
<tr><td>Bit0</td></tr>
<tr><td rowspan="8">BYTE_2</td><td>Bit7</td></tr>
<tr><td>Bit6</td></tr>
<tr><td>Bit5</td></tr>
<tr><td>Bit4</td></tr>
<tr><td>Bit3</td></tr>
<tr><td>Bit2</td></tr>
<tr><td>Bit1</td></tr>
<tr><td>Bit0</td></tr>
</table>

Note: The status information in the data pack is the status at the moment recorded by the time bit in the data pack.

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

BYTE_1 Bit7 0
BYTE_1 Bit6 0
BYTE_1 Bit5 0 （real-time GPS）
BYTE_1 Bit4 1 （GPS is located）
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 → Direction332°（0101001100 binary conversion to Decimal is 332）
BYTE_2 Bit4 0
BYTE_2 Bit3 1
BYTE_2 Bit2 1
BYTE_2 Bit1 0
BYTE_2 Bit0 0

This means GPS is located，real-time GPS，north latitude，east longitude and Direction332°.

5.2.1.10 MCC

Mobile Country Code(MCC)

For example: China Mobile's country code is 460 (Decimal) 0x01 0xCC (Decimal460 is converted to Hexadecimal, if the number of Hexadecimal is less than four digits, 0 is added to the left)

The value range here is: 0x0000 ~ 0x03E7

5.2.1.11 MNC

Mobile Network Code(MNC)

For example: China Mobile's is 0x00.

5.2.1.12 LAC

The Location Area Code (LAC) is included in the LAI, consisting of two bytes, using Hexadecimal encoding. The available range is 0x0001-0xFFFE, and the code groups 0x0000 and 0xFFFF cannot be used (see GSM specifications 03.03, 04.08 and 11.11) .

5.2.1.13 Cell ID

Mobile base stationCell Tower ID (Cell ID), the value range is 0x000000 ~ 0xFFFFFF

5.2.1.14 ACC

ACC status ACC low is 00, ACC high is 01

5.2.1.15 Data reporting mode

reserve，Currently inactive

5.2.1.16 GPS real-time supplementary transmission

0x00 real-time upload

0x01 Supplementary transmission

5.2.1.17 message serial number

see data package format 4.5

5.2.1.18 error checking

see data package format 4.6

5.2.1.19 Stop bit

see data package format 4.7

5.3 Alarm package (GPS, LBS, status merge information package)

5.3.1 The terminal sends the server alarm data package

Format			length(Byte)
information	start bit		2
	packet length		1
	agreement number		1
	date time		6
	GPS information	GPS information satellite number	1
		latitude	4
		longitude	4

		speed	1
		Direction, status	2
	LBS information	LBS length	1
		MCC	2
		MNC	2
		LAC	2
		Cell ID	4
	status information	terminal information	1
		Voltage level	1
		GSM signal strength	1
		Alarm/language/extension port status	2
	serial number		2
	error checking		2
	end bit		2

The alarm package is composed by adding status information (alarm information)to the positioning package. The encoding protocol Format is also created by adding status information to the positioning package's start bit.

see data package format 4.1

5.3.1.1packet length

see data package format 4.2

5.3.1.2 agreement number

fixed as0x32see data package format 4.3

5.3.1.3 date time

See location data package format 5.2.1.4 for details

5.3.1.4 GPS information length, the number of satellites involved in positioning

See location data package format 5.2.1.5 for details

5.3.1.5 latitude

See location data package format 5.2.1.6 for details

5.3.1.6 longitude

See location data package format 5.2.1.7 for details

5.3.1.7 speed

See location data package format 5.2.1.8 for details

5.3.1.8 State Direction

See location data package format 5.2.1.9 for details

5.3.1.9 MCC

See location data package format 5.2.1.10

5.3.1.10 MNC

See location data package format 5.2.1.11

5.3.1.11 LAC

See location data package format 5.2.1.12

5.3.1.12 Cell ID

See location data package format 5.2.1.13

5.3.1.13 terminal information

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

位		code meaning
BYTE	Bit7	1: motor disconnected
		0: motor connected
	Bit6	1: GPS is located
		0: GPS not located
	Bit3～Bit5	100: SOS for help (not supported)
		011: Low battery alarm
		010: Power failure alarm (not supported)
		001: Vibration alarm (not supported)
		000: Normal
	Bit2	1: Charged with power
		0: Charging without power
	Bit1	1: ACC high
		0: ACC low
	Bit0	1: fortification
		0: disarm

For example, with 0x44, the corresponding binary is 01000100

This indicates that the terminal status is as follows: motor connected, GPS has been positioned, Normal without alarms, powe connected for charginh, ACC low and disarmed.

5.3.1.14 Voltage level

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

0: No power (shut down)

1: Extremy low battery (not enough to make calls, send text messages, etc.)

2: Low battery alarm

3: Low battery (Normal use possible)

4: In battery

5: High battery

6: Very high battery

For example, 0x02 represents a low battery alarm sent when the battery is very low.

5.3.1.15 GSM signal strength grade

0x00: no signal；

0x01: very weak signal

0x02: weak signal

0x03: good signal

0x04: strong signal

For example, 0x03 represents a GSM good signal.

5.3.1.16 alarm/language

0x00 (front bit) 0x01 (back bit)

Front position: terminal alarm status (applicable to alarm package and requires electronic fence funcionality)

Back bit: the current language bit of the terminal

byte1	0x00: Normal
	0x01: SOS for help
	0x02: Power failure alarm
	0x03: Vibration alarm
	0x04: Enter the fence alarm
	0x05: Out of the fence alarm
	0x06 Over speed alarm
	0x09 Displacement alarm
	0x0A Enter GPS blind zone alarm
	0x0B Exit of GPS blind zone alarm
	0x0Cpower on alarm
	0x0E External power Low battery alarm
	0x0F External power low point protection alarm
	0X11shutdown alarm
	0X13 (Demolition alarm)
	0X14 door alarm
	0X15 Low power shutdown
	0X2C collision alarm
	0x2D flip alarm
	0x2E sharp turn alarm
	0x28 Rapid deceleration alarm
	0X29 rapid acceleration alarm
byte2	0x01 Chinese 0x02 English

For example:

At alarm Chinese: 0x00 0x01 ； At alarm English: 0x00 0x02

To increase the reliability of the alarm information, it marked repeatedly. In most cases, the alarm information is consistent with the upper terminal information. The inconsistencies are as follows:

A，Terminal message displays low battery alarm B，alarm/language information entry and exit fence alarm

5.3.1.17 message serial number

see data package format 4.5

5.3.1.18 error checking

see data package format 4.6

5.3.1.19 Stop bit

see data package format 4.7

Note: The status information in the data pack is the status at the moment recorded by the time bit in the data pack.

5.3.1.20 The server sends Alarm data package reply to the terminal

Format		length(Byte)	example
information	start bit	2	0x78 0x78
	packet length	1	0x05
	agreement number	1	0x26
	serial number	2	0x01 0x33
	error checking	2	0x5d 0xab
	end bit	2	0x0d 0x0a

The alarm package is composed of adding status information (alarm information) on the basis of positioning package, and the encoding protocol format is also composed of adding status information on the basis of positioning package.

start bit

see data package format 4.1

packet length

see data package format 4.2

agreement number

see data package format 4.3

message serial number

see data package format 4.5

error checking

see data package format 4.6

stop bit

see data package format 4.7

5.3.1.21 The server sends the Alarm data address package reply to the terminal

Chinese reply

The Chinese reply data pack is as follows:

The command packet sent by the server to the terminal （15+M+N Byte）	start bit			2
	data bit length			1
	agreement number			1
	information	command length		1
		server flag		4
		command content	ALARMSMS	8
			&&	2
			address content	M
			&&	2
			telephone	21

			number	
			##	2
	message serial number			2
	Check Digit			2
	Stop bit			2

Request Chinese address reply agreement number: 0X17.

command content: ALARM SMS && address

content && telephone number(all 0)##(ALARM SMS, &&, ## is a fixed string)

Chinese address content is delivered with UNICODE encoding.

reply Chinese address information example:

7878 //start bit

85 //data length

17 //reply agreement number

7E //Command length is to send content information length

00000001 //server flag

414C41524D534D53 // ALARMSMS

2626 //&& delimiter

624059044F4D7F6E0028 //Chinese location sent in UNICODE

004C004200530029003A

5E7F4E1C77015E7F5DDE

5E0282B190FD533AFF17

FF15FF144E6190530028

004E00320033002E0033

00390035002C00450031

00310032002E00390038

0038002996448FD1

2626 //&& delimiter

00000000000000000000000000000000000000000000 //telephone number

2323 //## content message terminator

0106 //serial number

3825 //Check Digit

0D0A //Stop bit

English reply

Considering that English or other foreign addresses are lengthy, one data bit is not sufficient; therefore, it is increased to 2 bytes.

Note: Among these, the data bit length corresponding to the agreement number solely for the return address information, is changed to 2.

The command packet sent by the server to the terminal （15+M+N Byte）	start bit			2
	data bit length			2
	agreement number			1
	information	command length		2
		server flag		4
		command content	ALARMSMS	8
			&&	2

			address content	M
			&&	2
			telephone number	21
			##	2
	message serial number			2
	Check Digit			2
	Stop bit			2

Request English address reply agreement number: 0X97
command content: ALARM SMS && address content && telephone number(all 0)##(ALARM SMS, &&, ## are fixed strings)

Sample reply English sample address information:

7979 //start bit
0080 //data length
97 //reply agreement number
0078 //Command length is to send content information length
00000001 //server flag
414C41524D534D53 // ALARMSMS
2626 //&& delimiter
534F53284C293A536869 //English position is sent in ASCII
6D696E2046616972796C
616E6420576573742052
642C4875696368656E67
2C4875697A686F752C47
75616E67646F6E67284E
32332E3131312C453131
342E343131294E656172
6279
2626 //&& delimiter
00000000000000000000000000000000000000000000 //telephone number
2323 //## content message terminator
0007 //serial number
72b5 //Check Digit
0D0A //Stop bit

Note: Since some alarm functions do not require the platform to reply address information, the platform does not need to respond with adress resolution after receiving the terminal alarm package. The address reply alarm types include the following:

1. Low battery alarm
2. Over speed alarm
3. GPS blind area

5.4 Heartbeat package (status information package)

The heartbeat package is a data pack that maintains the connection between the terminal and the server

5.4.1 The terminal sends to the server heartbeat package

Format			length(Byte)	example
information	start bit		2	0x78 0x78
	packet length		1	0x0a
	agreement number		1	0x13
	status information	terminal information	1	0x04
		Voltage level	1	0xf0
		GSM signal strength	1	0x04
		Language/extension status	2	0x8d 0x01
	serial number		2	0x00 0x7f
	error checking		2	0x71 0x7f
	end bit		2	0x0d 0x0a

start bit

see data package format 4.1

packet length

see data package format 4.2

agreement number

0x13

terminal information

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

位		code meaning
BYTE	Bit7	1: motor disconnected
		0: motor connected
	Bit6	1: GPS is located
		0: GPS not located
	Bit3～Bit5	100: SOS for help (not supported)
		011: Low battery alarm
		010: Power failure alarm (not supported)
		001: Vibration alarm (not supported)
		000: Normal
	Bit2	1: Charged with power
		0: Charging without power
	Bit1	1: ACC high

		0: ACC low
	Bit0	1: fortification
		0: disarm

For example: 0x44, the corresponding binary is 01000100.

This indicates that the terminal status is: motor connected, the GPS has been positioned, power has been connected for charging, the ACC is low, and in a disarmed state.

5.4.2Voltage level

On battery power

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

0: No power (shut down)

1: The battery is extremely low (not enough to make calls, send text messages, etc.)

2: Low battery alarm

3: Low battery (Normal use is possible)

4: In battery

5: High battery

6: Very high battery

For exampel: 0x02, a low battery alarm is sent when the battery is very low

When connected to external power, this bit is External voltage*10 byte1

byte1 0xFX F is Fixed value, representing the status of connecting to external power

5.4.3 GSM signal strength grade

0x00: no signal；

0x01: very weak signal

0x02: weak signal

0x03: good signal

0x04: strong signal

For example: 0x03 GSM has a good signal

5.4.4 Language/extension status

0x00 (front bit) 0x01 (back bit)

Front bit: terminal expansion port status (External voltage*10 byte0)

External voltage analysis example

byte1 byte0

0xF3 0x20 take 0x320 to Decimal 800 External voltage is 80.0V

Back bit: the current language bit of the terminal

Front bit	

rear bit	0x01Chinese 0x02English

Such as:

at alarm Chinese: 0x00 0x01

at alarm English: 0x00 0x02

message serial number

see data package format 4.5

error checking

see data package format 4.6

Stop bit

see data package format 4.7

server response data pack

Format		length
heartbeat packet（18 Byte）	start bit	2
	packet length	1
	agreement number	1
	message serial number	2
	error checking	2
	Stop bit	2

The server responds to the terminal with a packet: (the agreement number in the response packet is the same as the data package agreement number sent by the terminal)

start bit

see data package format 4.1

packet length

see data package format 4.2

agreement number

0x13

message serial number

see data package format 4.5

error checking

see data package format 4.6

Stop bit

see data package format 4.7

data instance

Terminal sending example

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

explain

0x78 0x78	0x08	0x13	0x4B 0x04 0x03	0x00 0x01	0x00 0x11	0x06 0x1F	0x0D 0x0A
start bit	length	agreement	information	Reserved (language)	serial number	error checking	Stop bit

number					
Server reply example					
78 78 05 13 00 11 F9 70 0D 0A					
explain					
0x78 0x78	0x05	0x13	0x00 0x11	0xF9 0x70	0x0D 0x0A
start bit	length	agreement number	serial number	error checking	Stop bit

5.5 base station data package(LBS ONLY)

5.5.1 The terminal sends to the server location data pack

Format				length(Byte)
information	start bit			2
	packet length			1
	agreement number			1
	date time			6
	LBS information	TA		1
		MCC		2
		MNC		2
		Cell Num		1
		base station 1	LAC	2
			Cell ID	4
			RSSI	1
		base station 2	LAC	2
			Cell ID	4
			RSSI	1
		base station 3	LAC	2
			Cell ID	4
			RSSI	1
		base station 4	LAC	2
			Cell ID	4
			RSSI	1
		base station 5	LAC	2
			Cell ID	4
			RSSI	1
	reserve			3
	serial number			2
	error checking			2
	end bit			2

start bit

see data package format 4.1

packet length

see data package format 4.2

agreement number

fixed as0x34see data package format 4.3

date time

Format	length(Byte)	example

Year	1	0x0A
Month	1	0x03
Day	1	0x17
Hour	1	0x0F
Minute	1	0x32
Second	1	0x17

For example: 2010Year 3Month 23Day 15hour 50minute 23Second

Calculation method: 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

MCC

Mobile Country Code(MCC)

For example: China Mobile's country code is: China Mobile's country code is 460 (Decimal) 0x01 0xCC (Decimal460 is converted to Hexadecimal, if the number of Hexadecimal is less than four digits, 0 is added to the left)

The value range here is: 0x0000 ~ 0x03E7

MNC

Mobile Network Code(MNC)

For example: China Mobile's is 0x00.

LAC

The Location Area Code (LAC) is included in the LAI, consisting of two bytes, using Hexadecimal encoding. The available range is 0x0001-0xFFFE, and the code groups 0x0000 and 0xFFFF cannot be used (see GSM specifications 03.03, 04.08 and 11.11) .

Cell ID

Mobile base stationCell Tower ID (Cell ID), the value range is 0x000000 ~ 0xFFFFFF

5.5.1.9. RSSI

The signal strength of the main cell, the value range is 0x00～0xFF, the actual signal strength is a negative value, upload its absolute value.

see glossary – RSSI.

message serial number

see data package format 4.5

error checking

see data package format 4.6

Stop bit

see data package format 4.7

5.6 WIFI information package

Format		length(Byte)	Detailed explanation
start bit		2	0x78 0x78
packet length		1	length=agreement number+information+message serial number+error checking
agreement number		1	0x33
information	date time	6	Year (1byte) Month (1byte) Day (1byte) Hour (1byte) Minute (1byte) Second (1byte) (converted to Decimal)
	MCC	2	Mobile Country Code
	MNC	2	Mobile Network Code
	LAC	2	
	CI	4	Mobile base station Cell Tower ID (Cell ID)
	RSSI	1	Cell signal strength, value range is 0x00～0xFF, 0x00 signal is the weakest, 0xFF signal is the strongest.
	NLAC1	2	Ditto LAC
	NCI1	4	Ditto CI
	NRSSI 1	1	Ditto RSSI
	...	...	...
	NLAC5	2	Ditto LAC
	NCI5	4	Ditto CI
	NRSSI 5	1	Ditto RSSI
	Timing advance	1	It refers to the actual time when the mobile station signal arrives at the base station, and the time difference between between the arrival of the mobile station signal at the base station, assuming that the distance between the mobile station and the base station is 0.
	Number of Wi-Fi	1	This is used to determine the number of transmitted WIFI signals in the package, 0 means no WIFI detected.
	WIFI MAC1	6	The MAC of the received signal 1 WIFI (transmit according to the actual number of WIFI found, if one is found, one will be transmitted, if more than one is found, multiple transmissions will be sent, if it is not found, the transmission will be 0)
	WIFI strength 1	1	WIFI signal 1 strength
	...	...	...

	WIFI MAC5	6	ditto
	WIFI strengt h 5	1	ditto
message serial number		2	After starting up, the serial number will automatically increase by 1 each time data is sent
error checking		2	The CRC-ITU value from "packet length" to "message serial number". If the receiver receives a CRC error in the calculation of the message, it should ignore and discard the data pack
Stop bit		2	Fixed value, uniformly 0x0D 0x0A

5.7 General Packet for Information Transfer

Used for terminal transmission of various non-location data

example data: 79 79 00 20 94 0a 08 66 54 50 52 50 16 96 46 00 45 94 13 02 69 00 89 86 04 39 10 18 90 67 26 99 00 06 1b 25 0d 0a

		length	Detailed explanation
start bit		2	0x79 0x79
packet length		2	
agreement number		1	0x94
information	information type （sub-agreement number）	1	00 External voltage 01~03 (customized) 04 Terminal state synchronization 05 Door Status 08 Self-test parameters 09 Positioning satellite information 0A ICCID and other information .....to be added
	data content	N	The transmission content is different according to the information
message serial number		2	After starting up, the serial number will automatically increase by 1 each time data is sent
error checking		2	The CRC-ITU value from "packet length" to "message serial number". If the receiver receives a CRC error in the calculation of the message it should ignore and discard this data pack
Stop bit		2	Fixed value, uniformly 0x0D 0x0A

When the type is 0A, this bit transmits the following information, and the transmission is a Hexadecimal number

08 69 81 41 01 11 51 23 04 60 04 24 29 80 22 56 89 86 04 04 10 18 C0 00 22 55

IMEI	8	Example: If the IMEI number is 0869 814101115123, then the Terminal ID is 0x08 0x69 0x81 0x41 0x01 0x11 0x51 0x23
IMSI	8	Example: If the IMSI number is 0460042429802256, then the Terminal ID is 0x04 0x60 0x04 0x24 0x29 0x80 0x22 0x56
ICCID	10	Example: ICCID number is 898604041018 C0002255, then Terminal ID is 0x89 0x86 0x04 0x04 0x10 0x18 0xC0 0x00 0x22 0x55

5.7.1 server response data pack

Format		length	describe
General package reply	start bit	2	**0x79 0x79**
	packet length	2	
	agreement number	1	**0x94**
	sub-protocol number	1	**Same as send package**
	message serial number	2	
	error checking	2	
	Stop bit	2	**0x0D 0x0A**

5.8 large file transfer（0x8D）

Used to transmit large-volume files such as audio files. The terminal sends data to the server using the 0x8D protocol.

5.8.1 The terminal sends to the server

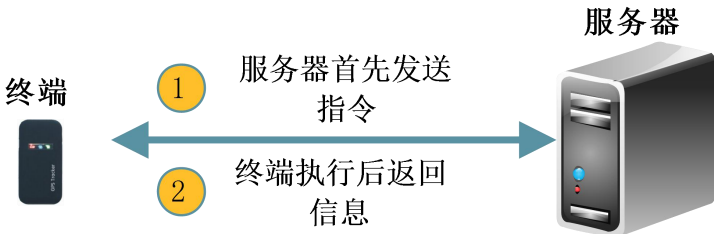
Format		length(Byte)	Detailed explanation
information	start bit	2	0x79 0x79
	packet length	2	length=agreement number+information+message serial number+error checking
	agreement number	1	0x8D
	file type	1	0x00 recording file (monitoring) 0x01 recording file (voice control) 0x02 Platform instruction request
	Total file length	4	Transfer Total file length
	File error checking type	1	Use CRC for file transfer when the error checking type is "00" Use MD5 checksum for files transfer when the error checking type is "01"
	File error checking code	N	When the error checking type is "00", use the CRC check for file transfer,resulting in a length of 2 digits. When the error checking type is "01", use the MD5 checksum for file transfer, resulting in a length of 16 bits
	start position	4	Transmission split start position bytes
	current content length	2	The data length after the transmission is divided from the start position
	content	M	For the split data pack
	flag	N	file type represents the type of the transfer file: When the file type is 00 (recording file - monitoring), this position is six bytes, representatin the date and time when the transmission monitoring starts. The encoding method is the same as the positioning packet time format. When the file type is 01 (recording file - voice control), this position is six bytes, representatin the date and time when the transmission monitoring starts. The encoding method is the same as the positioning packet time format. When the file type is 02 (platform command uploaded), the unified upload service ID is used. The number of bytes continues to be 6 bytes. The

			first 4 digits represents the server flag, and the last two digits represents the transmission serial number.
	message serial number	2	After starting up, the serial number will automatically increase by 1 each time data is sent
	error checking	2	The CRC-ITU value from "packet length" to "message serial number". If the receiver receives a CRC error in the calculation of the message, ignore it and discard the data pack (see Appendix A algorithm details)
	Stop bit	2	Fixed value, unified as 0x0D 0x0A

5.8.2 server response data pack

Format		length(Byte)	Detailed explanation
information	start bit	2	0x79 0x79
	packet length	2	length=agreement number+information+message serial number+error checking
	agreement number	1	0x8D
	receive status flag	1	Receive Normal 0x00 Receive Error 0x01
	message serial number	2	After starting up, the serial number will automatically increase by 1 each time data is sent
	error checking	2	The CRC-ITU value ranges from "packet length" to "message serial number". If the receiver detects a CRC error in the message calculation, it should ignore it and discard the data pack (see Appendix A algorithm detais)
	Stop bit	2	Fixed value, unified as 0x0D 0x0A

6 The server sends a packet to the terminal



6.1. server sent

Format		length(Byte)
start bit		2
packet length		1
agreement number		1
information	command length	1
	server flag	4
	command content	M
	language	2
message serial number		2
error checking		2
Stop bit		2

6.1.1. start bit

see data package format 4.1

6.1.2. packet length

see data package format 4.2

6.1.3. agreement number

The terminal sends the agreement number using: 0x80

6.1.4. command length

server flag+command content length

For example: in the unit of byte length, 0x0A means that the flag + command content occupies 10 bytes

6.1.5. server flag

It is reserved for the server to identify, and the terminal will return the received data binary in the return packet

6.1.6. command content

Represented by ASCII character string, the command content is compatible with SMS commands

6.1.7. language

Terminal current language bit.

Chinese: 0x00 0x01

English: 0x00 0x02

6.1.8. message serial number

see data package format 4.5

6.1.9. error checking

see data package format 4.6

6.1.10. Stop bit

see data package format 4.7

6.2. Terminal returns

Format		length(Byte)	example
start bit		2	0x79 0x79
packet length		2	0x00 0x09
agreement number		1	0x21
information	server flag	4	0x00 0x00 0x00 0x01
	content encoding	1	
	content	M	
message serial number		2	0x00 0x01
error checking		2	0xD9 0xDC
Stop bit		2	0x0D 0x0A

6.2.1 start bit

Fixed value 0x79 0x79

6.2.2 packet length

takes 2 bytes

6.2.3 agreement number

use 0x21

6.2.4 server flag

It is reserved for the server to identify, and the terminal will return the received data binary in the return packet

6.2.5 content encoding

0x01 ASCⅡ encoding

0x02 UTF16-BE encoding

6.2.6 content

The data to be sent.

6.2.7 message serial number

see data package format 4.5

6.2.8 error checking

see data package format 4.6

6.2.9 Stop bit

see data package format 4.7

6.3. motor cut off

Function description: Cut off the vehicle's oil-electric control circuit.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
RELAY,1#

Terminal returns
return successfully return failed

6.4. Restoring motor

Function description: connect the vehicle oil-electric control circuit

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
RELAY,0#
Terminal returns
return successfully return failed

illustrate:

The following are the various reply content of motor cut off, and success can be determinated only when the Success character is found.

RELAY,ERROR: 103
command parameter error

OK,engine will be stopped after GPS fixed or speed less than 20km/h

Already in the state of fuel supply cut off, the command is not running!

Already in the state of fuel supply to resuming,the command is not running!

Cut off the fuel supply: Success!

Restore fuel supply: Success!

6.5. Add SOS number

Function description: add the SOS number for receiving alarm text messages and phone calls

In the example, the sent and returned strings will be converted to ASCII to generate command content.

server sent
SOS,A,NUM1,NUM2,NUM3#
Terminal returns
return successfully

SOS_OK!SOS1: NUM1SOS2: NUM2SOS3NUM3
return failed
ERROR: XXX

6.6. Delete SOS number

Function description: Delete the SOS number for receiving alarm text messages and calls.

In the example, the sent and returned strings will be converted to ASCII to generate command content.

server sent
SOS,D,NUM1,NUM2,NUM3#
Terminal returns
return successfully
SOS_OK!SOS1: NUM1SOS2: NUM2SOS3NUM3
return failed
ERROR: XXX

6.7. Set the center number

Function description: Set the center number that can control motor cut off

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
CENTER,A,NUM#
Terminal returns
return successfully
CENTER_OK
return failed
ERROR: XXX

6.8 Delete center number

Function description: Delete the center number that can control motor cut off.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
CENTER,D#
Terminal returns
return successfully
CENTER_OK
return failed
ERROR: XXX

6.9Turn on Vibration alarm

Function description: Turn on Vibration alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
SENALM, ON,Alarm method#
Terminal returns

return successfully
SENALM_OK
return failed
ERROR: XXX

Note: Alarm method is 0: platform； 1: platform + message； 2: platform+message+call； 3: platform+call;

6.10 Turn off Vibration alarm

Function description: Turn off Vibration alarm

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
SENALM, OFF#
Terminal returns
return successfully
SENALM_OK
return failed
ERROR: XXX

6.11 Turn on Over speed alarm

Function description: Turn on Over speed alarm

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
SPEED, ON,TIME,SPEED,Alarm method#
Terminal returns
return successfully
SPEEDALM_OK
return failed
ERROR: XXX

Note: Alarm method is 0: platform；1: platform + message；

6.12 Turn off Over speed alarm

Function description: Turn off Over speed alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
SPEED, OFF#
Terminal returns
return successfully
SPEEDALM_OK
return failed
ERROR: XXX

6.13 Turn on Power failure alarm

Function description: Turn on Power failure alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
POWERALM, ON,Alarm method#
Terminal returns
return successfully
POWERALM_OK
return failed
ERROR: XXX

Note: Alarm method is 0: platform; 1: platform + message; 2: platform+message+call; 3platform+call;

6.14 Turn off power failure alarm

Function description: Turn off power failure alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
POWERALM, OFF#
Terminal returns
return successfully
POWERALM_OK
return failed
ERROR: XXX

6.15 Turn on Displacement alarm

Function description: Turn on Displacement alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
MOVING, NO,位移半径，Alarm method#
Terminal returns
return successfully
MOVING_OK
return failed
ERROR: XXX

Note: Alarm method is 0: platform; 1: platform + message; 2: platform + message + call; 3 platform + call;

Displacement radius: 100～1000

6.16 Turn Off Displacement Alarm

Function description: Turn Off Displacement Alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
MOVING ,OFF#
Terminal returns
return successfully

MOVING_OK
return failed
ERROR: XXX

6.17 Turn on Low battery alarm

Function description: Turn on Low battery alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
BATALM, ON,Alarm method#
Terminal returns
return successfully
BATALM_OK
return failed
ERROR: XXX

Note: The alarm mode is 0: platform; 1: platform + message;

6.18 Turn off Low battery alarm

Function description: Turn off Low battery alarm.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
BATALM, OFF#
Terminal returns
return successfully
BATALM_OK
return failed
ERROR: XXX

6.19 Turn on fence alarm

Function description: Turn on electric fence.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
FENCE,ON,0, center latitude, center longitude, fence radius, X, alarm mode# For example: FENCE,ON,0,N1.2971,E103.822349,61,IN,0#
Terminal returns
return successfully
FENCE_OK
return failed
ERROR: XXX

Note: The alarm mode is 0: platform; 1: platform + message;

X=IN/OUT; IN: alarm when entering the fence, OUT: Out of the fence alarm, if it is empty, it means that both entering/exiting the fence will alarm; the default is that both entering/exiting the fence will alarm；

6.20 Turn off fence alarm

Function description: Turn on electric fence.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
FENCE,OFF#
Terminal returns
return successfully
FENCE_OK
return failed
ERROR: XXX

6.21 Control device restart

Function description: Make the device reboot after 1 minute.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
RESET#
Terminal returns
return successfully
The terminal will restart after 1 minute!
return failed
ERROR: XXX

6.22 recording command

Function description: After receiving this instruction, the terminal immediately starts recording and uploads through the 0x8D instruction.

In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent
LY,XX#
XX stands for recording time, 10-300 (unit is Second), 999 means Turn on continuous recording, 0 means Turn off continuous recording
Terminal returns
return successfully
When XX is 0 or 999: CLY_OK
When XX is 10-300: LY_OK: Start recording.
return failed
ERROR: XXX

6.23 The server sends the query address information

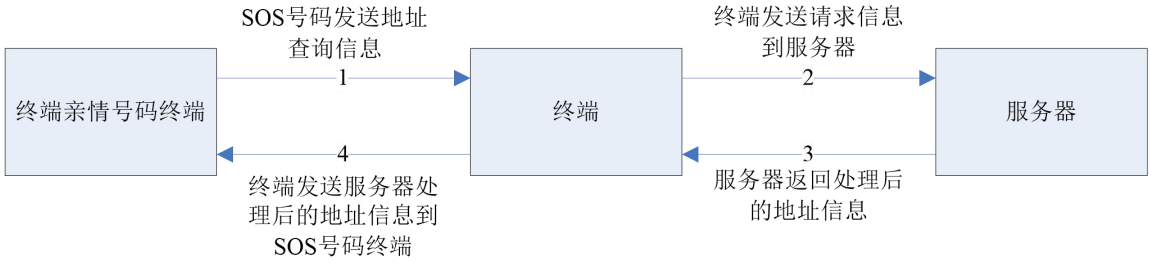
In the example, the sent and returned strings will be converted to ASCII to generate command content

server sent

ADDRESS, address content, telephone number

Note: Chinese address content is delivered with UNICODE encoding.

6.23 GPS, phone number query address information package(0X2A)



6.6.1. Terminal sends information to server

When the terminal receives a GPS information package, it is essentially the same as the format mentioned before, but with the addition of a telephone number to query the adress.

Format			length(Byte)
start bit			2
packet length			1
agreement number			1
information	date time		6
	GPS information	GPS information length, the number of satellites involved in positioning	1
		latitude	4
		longitude	4
		speed	1
		Direction, status	2
	telephone number		21
	language		2
message serial number			2
error checking			2
Stop bit			2

start bit

see data package format 4.1

packet length

see data package format 4.2

For example: in the unit of byte length, 0x2E means that the command content occupies 46 bytes

agreement number

Use 0x2A.

date time

See location data package format 5.2.1.4 for details

6.24 GPS information length, the number of satellites involved in positioning

See location data package format 5.2.1.5 for details

latitude

See location data package format 5.2.1.6 for details

longitude

See location data package format 5.2.1.7 for details

speed

See location data package format 5.2.1.8 for details

Direction

See location data package format 5.2.1.9 for details

telephone number

The SOS Telephone number requested for address query is converted through ASCII, and the right side is filled with 0 if it is less than 21 digits

language

Terminal current language bit.

Chinese: 0x00 0x01

English: 0x00 0x02

message serial number

see data package format 4.5

error checking

see data package format 4.6

Stop bit

see data package format 4.7

server response

According to the extended instruction requesting a reply in Chinese address or English address, the reply packets are inconsistent between the two formats.

Chinese reply

The Chinese reply packet is as follows:

<table>
<tr><th colspan="4">Format</th><th>length(Byte)</th></tr>
<tr><td rowspan="13">The command packet sent by the server to the terminal
（15+M+N Byte）</td><td colspan="3">start bit</td><td>2</td></tr>
<tr><td colspan="3">data bit length</td><td>1</td></tr>
<tr><td colspan="3">agreement number</td><td>1</td></tr>
<tr><td rowspan="11">information</td><td colspan="2">command length</td><td>1</td></tr>
<tr><td colspan="2">server flag</td><td>4</td></tr>
<tr><td rowspan="7">command content</td><td>ADDRESS</td><td>7</td></tr>
<tr><td>&&</td><td>2</td></tr>
<tr><td>address content</td><td>M</td></tr>
<tr><td>&&</td><td>2</td></tr>
<tr><td>telephone number</td><td>21</td></tr>
<tr><td>##</td><td>2</td></tr>
<tr><td colspan="3">message serial number</td><td>2</td></tr>
<tr><td colspan="3">Check Digit</td><td>2</td></tr>
<tr><td colspan="3">Stop bit</td><td>2</td></tr>
</table>

Request Chinese address reply agreement number: 0X17.

command content: ADDRESS&& address content && telephone number##(ADDRESS, &&, ## are fixed strings)

Chinese address content is delivered with UNICODE encoding.

reply Chinese address information example:

7878 //start bit
84 //data length
17 //reply agreement number
7E //Command length is to send content information length
00000001 //server flag
41444452455353 //ADDRESS
2626 //&& delimiter
624059044F4D7F6E0028 //Chinese location sent in UNICODE
004C004200530029003A
5E7F4E1C77015E7F5DDE
5E0282B190FD533AFF17
FF15FF144E6190530028
004E00320033002E0033
00390035002C00450031
00310032002E00390038
0038002996448FD1
2626 //&& delimiter
3133373130383139313335000000000000000000000000 //telephone number
2323 //## content message terminator
0106 //serial number
3825 //Check Digit
0D0A //Stop bit

English reply

Considering that English or other foreign addresses are longer, one data byte is not enough, so it is increased to 2 bytes.

Note that the length of the data byte corresponding to the agreement number for returning address information is changed to 2.

<table>
<tr><th colspan="4">Format</th><th>length(Byte)</th></tr>
<tr><td rowspan="11">The command packet sent by the server to the terminal（15+M+N Byte）</td><td colspan="3">start bit</td><td>2</td></tr>
<tr><td colspan="3">data bit length</td><td>2</td></tr>
<tr><td colspan="3">agreement number</td><td>1</td></tr>
<tr><td rowspan="8">information</td><td colspan="2">command length</td><td>2</td></tr>
<tr><td colspan="2">server flag</td><td>4</td></tr>
<tr><td rowspan="6">command content</td><td>ADDRESS</td><td>7</td></tr>
<tr><td>&&</td><td>2</td></tr>
<tr><td>address content</td><td>M</td></tr>
<tr><td>&&</td><td>2</td></tr>
<tr><td>telephone number</td><td>21</td></tr>
<tr><td>##</td><td>2</td></tr>
<tr><td colspan="3">message serial number</td><td>2</td></tr>
<tr><td colspan="3">Check Digit</td><td>2</td></tr>
</table>

	Stop bit	2

Request English address reply agreement number: 0X97

command content: ADDRESS&&address content&&telephone number##(ADDRESS ， && ， ## is a fixed string)

Sample reply English sample address information:

7878 //start bit
00D1 //data length
97 //reply agreement number
00CA //Command length is to send content information length
00000001 //server flag
41444452455353 //ADDRESS
2626 //&& delimiter
0053004F00530028004C //English position is sent in UNICODE
0029003A005300680069
006D0069006E00200046
00610069007200790006C
0061006E006400200057
0065007300740020052
0064002C004800750069
006300680065006E0067
002C004800750069007A
0068006F0075002C0047
00750061006E00670064
006F006E00670028004E
00320033002E00310031
0031002C004500310031
0034002E003400310031
0029004E006500610072
00620079
2626 //&& delimiter
3132353230313337393037373430353100000000000 //telephone number
2323 //## content message terminator
0007 //serial number
72b5 //Check Digit
0D0A //Stop bit

7. Attached A CRC-ITU table lookup algorithm C language code fragment

CRC-ITU table lookup algorithm C language code snippet

```
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,
};

//  Computes a 16-bit CRC of the given length of data.
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;           //  Opposite
}
```

8 Attached B communication protocol packet fragment example

The following data is intercepted from the communication between the terminal and the server, and the data displayed by Hexadecimal, sent means sent by the terminal, and received means returned by the server:

landing package:

发出： 78 78 11 01 08 66 54 50 52 50 16 96 28 01 32 01 00 01 69 04 0d 0a

接收： 78 78 05 01 00 02 EB 47 0D 0A

status packet:

发出： 78 78 0a 13 00 05 04 00 01 00 0e 0e 4a 0d 0a

接收： 78 78 05 13 00 05 AF D5 0D 0A

GPS data packet

发出： 78 78 24 31 14 08 13 08 2D 04 CB 02 6C 6F 6C 0C 37 13 A6 00 14 00 01 CC 00 01 25 FC 06 14 64 02 00 00 00 00 0B CD 21 0D 0A

WIFI data package

发出： 78 78 5E 33 14 08 15 08 06 38 01 CC 00 01 25 42 06 17 44 08 48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 1C 78 39 01 39 92 CE 48 7D 2E 4A A3 0A CC C8 50 E9 EE 22 0A C5 80 8F 1D 9F 5B 43 C5 54 75 95 5A 3E 63 C2 00 0E A4 5C 0D 0A

base station data package

发出： 78 78 37 34 14 08 15 06 17 14 00 01 CC 00 01 01 25 42 06 17 44 03 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 9A 70 0D 0A

Alarm data package

发出： 78 78 27 32 14 08 13 08 2D 34 C0 00 00 00 00 00 00 00 00 00 08 00 08 01 CC 00 01 25 FC 06 14 64 02 10 05 03 02 01 00 0D 4A EF 0D 0A

Note: English symbols must be used, the command is set correctly reply: OK, the command is not Normal: no reply or the format of the reply command is incorrect

NO.	Functional items	Command Format	Remarks/example
1	status inquiry	STATUS#	
2	version query	VERSION#	
3	Query parameter settings	PARAM#	
4	restart command	RESET#	
5	Add SOS number	SOS,A,Number1,Number2,Number3#	
6	Delete SOS number	SOS,D,Number1,Number2,Number3#	

7	Query SOS number	SOS#	
8	Change IP settings	SERVER,0,IP,Port,0#	SERVER,0,120.24.248.12,8005,0#
9	Change domain name settings	SERVER,1,domain name,port,0#	SERVER,1,GW.CARHERE.NET,8005,0#
10	Turn on the power failure alarm setting	POWERALM,A,M,T1,T2,# A=ON M=0～2; 0 GPRS only, 1 SMS+GPRS, 2 GPRS+SMS+CALL T1=2 ～ 60Second; power failure detection time T2=1-3600Second; minimum charging time	
11	Turn off power failure alarm	POWERALM,OFF#	
12	Query the power failure alarm status	POWERALM#	
13	Turn on the Low battery alarm setting	BATALM,A,M# A=ON M=0～1 0 stands for: GPRS only, 1 stands for: SMS+GPRS,	
15	Turn off the Low battery alarm setting	BATALM,OFF#	
16	Query the Low battery alarm status	BATALM#	
17	Turn on the displacement alarm setting	MOVING,A,R,M# A=ON R=100～1000; displacement radius M=0～2; 0 GPRS only, 1 SMS+GPRS; 2 GPRS+SMS+CALL	
18	Turn Off Displacement Alarm	MOVING,OFF#	
19	Query the displacement setting status	MOVING#	
20	recording	recording LY,60#	