GPS locator

protocol

catalogue

1. Communication Protocol

Introduction

This document defines the application-layer interface protocol for vehicle GPS locators and positioning service platforms. This protocol is exclusively for communication between server platforms and positioning terminals.

II. TERMS AND DEFINITIONS

Terms and abbreviatio	English meaning	Chinese meaning
CMPP	China Mobile Peer to Peer	China Mobile P2P protocol
GPS	Global Positioning System	Global Positioning System
GSM	Global System for Mobile Communication	GSM
GPRS	General Packet Radio Service	Universal Wireless Packet Service
TCP	Transport Control Protocol	TCP
LBS	Location Based Services	Augmented Reality Service
IMEI	International Mobile Equipment Identity	International Mobile Equipment Identity
MCC	Mobile Country Code	Mobile user country code
MNC	Mobile Network Code	Mobile network number
LAC	Location Area Code	Area code
Cell ID	Cell Tower ID	Mobile base station
UDP	User Datagram Protocol	user datagram protocol
SOS	Save Our Ship/Save Our Souls	Call for help
CRC	Cyclic Redundancy Check	CRC
NITZ	Network Identity and Time Zone,	time zone
GIS	Geographic Information System	geo-information system

III. Basic rules

1. The GPRS connection is established and the first login packet is sent to the server. If the server response packet is received within 5 seconds, the connection is considered normal. The system then starts sending location information (GPS and LBS packets). A status packet is sent after 3 minutes, and periodic communication checks are conducted to confirm normal operation.

2. When the GPRS connection fails to establish, the terminal cannot transmit login packets. After three consecutive connection failures, the terminal activates a 20-minute automatic reboot timer. If the terminal successfully reestablishes a connection with the server and receives a response packet from the server within this period, the timer is deactivated and the terminal remains operational. Otherwise, the terminal will automatically reboot after 20 minutes.

3. Upon receiving a login request from the terminal, the server must respond with a data packet. If the terminal fails to receive a response within 5 seconds after sending either a login or status packet, the connection is deemed abnormal. The system then activates the GPS data recovery function, disconnects the current GPRS connection, re-establishes a new GPRS connection, and retransmits the login request.

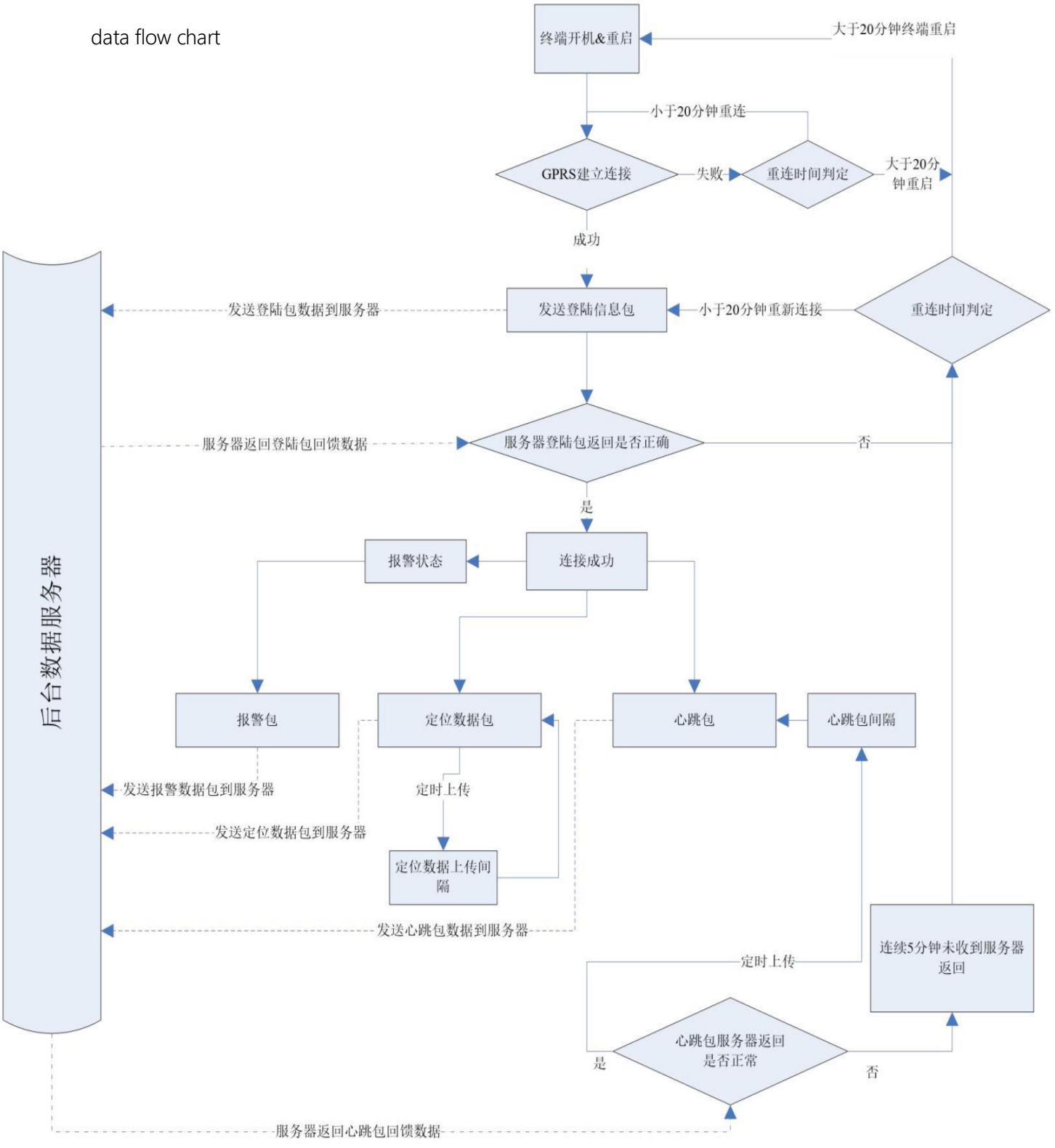
4. If the connection is deemed abnormal, the terminal will initiate a 10-minute automatic restart cycle. After three consecutive failed attempts to establish a connection or receive server response packets, the system will automatically shut down the restart function. If the terminal successfully re-establishes the connection and receives a server response within this 10-minute window, the restart function will be disabled. Otherwise, the terminal will automatically reboot after the 10-minute period.

5. After the connection is established normally, the terminal will periodically send GPS and LBS combined data packets to the server when GPS information changes. The server can configure the default transmission protocol through commands.

6. To ensure connection effectiveness, heartbeat packets are sent to the server at fixed intervals, and the server returns a response packet for confirmation;

7. For terminals without registered IMEI numbers, the server should respond to login requests and heartbeat packets without disconnecting. (Direct disconnection or failure to respond will cause the terminal to reconnect repeatedly, resulting in excessive GPRS data consumption).

data flow chart



4. Packet format

Communication is asynchronous and occurs in bytes.

Total package size: (10+N) bytes.

form	length
start bit	2
packet length	1
Agreement	1
information	N
Message serial	2
error check	2
stop bit	2

4.1 .start bit

Set to a fixed value of hexadecimal 0x780x78.

Note: A small number of protocol numbers start with 0x7 and 0x79.

4.2 .packet length

The total length is (5+N) bytes, consisting of the protocol number, information content, information sequence number, and error check, as the information content is a variable-length field.

4.3 .Agreement number

type	price
Login Package	0 x01
Location data (GPS/LBS	0 x12
Status information (heartbeat	0 x13
String information (respond to	0x15
warning message	0 x16
Query address information	0 x 1A
ICCID reports platform	0 x94
Upload recording file to	0x 8D
Issue instructions	0x 80

4.4 .information content

The corresponding "protocol number" is assigned to each application to determine the specific content.

4.5. Information Sequence Number

The first GPRS data packet (including status packets and GPS/LBS data packets) sent after power-on has serial number 1. For subsequent data packets (including status packets and GPS/LBS data packets), the serial number will automatically increase by 1.

4.6. Error Check

Terminals or servers can verify the integrity of received data using checksums. To prevent transmission errors and data corruption, error checking is implemented, thereby enhancing system security and efficiency. The checksum employs the CRC-ITU method.

The CRC-ITU value for the data segment in the protocol body, ranging from "Packet Length" to "Information Sequence Number" (including both fields).

If the receiving party detects a CRC error in the received data, it will discard the packet.

4.7. stop bit

Set to a fixed value of hexadecimal 0x0D 0x0A.

5. Detailed explanation of data packet transmission from the terminal to the server

Separate the common message packets sent and returned by the server

5.1. Login Information Packet (0x01)

5.1.1. The terminal sends data packets to the server

The login packet is used to confirm the connection with the server and submit the terminal ID.

form		length (Byte)	instance
Login information package (18 Byte)	start bit	2	0x 78 0x78
	packet length	1	0 x0D
	Agreement number	1	0 x01
	terminal ID	8	0 x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45
	Message serial number	2	0 x00 0x01
	error check	2	0 x8C 0xDD
	stop bit	2	0 x0D 0x0A

5.1.1.1. start bit

See Data Packet Format 4.1 for details

5.1.1.2. Package length

See Data Packet Format 4.2 for details

5.1.1.3. Agreement No.

See Data Packet Format 4.3 for details

5.1.1.4. Terminal ID

The terminal ID uses a 15-bit IMEI number.

For example: 123456789012345

The terminal ID is: 0x01 0x23 0x45 0x67 0x89 0x01 0x23 0x45

5.1.1.5. Information Sequence Number

See Data Packet Format 4.5

5.1.1.6. Error Check

See Data Packet Format 4.6 for details

5.1.1.7. Stop position

See Data Packet Format 4.7 for details

5.1.2. Server response packets

form		length (Byte)	instance
Response package (10 Byte)	start bit	2	0x 78 0x78
	packet length	1	0 x05
	Agreement number	1	0 x01
	Message serial number	2	0 x00 0x01
	error check	2	0 XD9 0xDC
	stop bit	2	0 x0D 0x0A

5.1.2.1. Starting Position

See Data Packet Format 4.1 for details

5.1.2.2. Package length

See Data Packet Format 4.2 for details

5.1.2.3. Agreement No.

The protocol number in the response packet matches the protocol number of the data packet sent by the terminal.

5.1.2.4. Information Sequence Number

See Data Packet Format 4.5

5.1.2.5. Error Check

See Data Packet Format 4.6 for details

5.1.2.6. Stop position

See Data Packet Format 4.7 for details

5.2. Positioning data packets (GPS and LBS combined: 0x12)

5.2.1. The terminal sends location data packets to the server

form			length （Byte）	instance
information content	start bit		2	0x 78 0x78
	packet length		1	0 x 1F
	Agreement number		1	0 x12
	Date and time		6	0 x13 0x08 0x1D 0x11 0x0C 0x10
	GPS information	GPS satellite count	1	0 xC B
		latitude	4	0 x02 0x7A 0xCF 0xEB
		longitude	4	0 xCC 0x46 0x58 0x49
		velocity	1	0 x10
		Course, Status	2	0 x1 5 0x 4C
	LBS information	MCC	2	0 x01 0xCC
		MNC	1	0 x00
		LAC	2	0 x28 0x7D
		Cell ID	3	0 x00 0x1F 0xB8
	serial number		2	0 x00 0x03
	error check		2	0 x 9D 0xDC
	end bit		2	0 x0D 0x0A

5.2 .1.1. Starting position

See Data Packet Format 4.1 for details

5.2 .1.2. Package length

See Data Packet Format 4.2 for details

5.2 .1.3. Agreement No.

See Data Packet Format 4.3 for details

5.2.1.4. Date and time

form	Length	instance
year	1	0x 1 6
moon	1	0 x0 1
sun	1	0 x1D
Time	1	0 x11
component	1	0 x0C
second	1	0 x10

For example, 29 January 20 22 at 1 7 :12:16

Calculation method: 22 (base 10) = 1 6 (base 1 6)

0 1 (base 10) = 0 1 (base 16)

29 (base 10) =1 D (base 16)

1 7 (base 10) = 11 (base 16)

12 (base 10) = 0C (base 16)

16 (base 1 0) = 1 0 (base 16)

The value is: 0x 1 6 0x 0 1 0x1 D 0x 11 0x 0C 0x1 0

The GPS location packet date is in UTC+0

5.2.1.5. Number of satellites involved in positioning

A 1-byte value converts to 8 bits in binary. The first 4 bits represent GPS message length, while the last 4 bits indicate the current number of satellites (up to 15). If the count exceeds 15, the system reports 15 satellites.

Note: The length includes 1 byte of space.

For example, a value of 0xCB indicates that the GPS information length is 12 and the number of participating satellites is 11.

5.2.1.6 .latitude

Occupies 4 bytes and represents the latitude value of the positioning data. The numerical range is 0 to 162000000, covering the 0° to 90° range. The conversion method is as follows:

Convert the latitude and longitude values from the GPS module into decimal degrees, then multiply the converted value by 30,000, and finally convert the result to a hexadecimal number.

For example, 22° 32.7658 = (22×60 + 32.7658) × 30000 = 40582974, then convert it to hexadecimal.

40582974 (base 10) = 26B3F3E (base 16)

The final values are 0x02,0x6B,0x3F, and 0x3E.

Note: All uploaded latitude and longitude values are positive. If negative, take the absolute value.

5 .2.1.7.longitude

Occupies 4 bytes and represents the longitude value of the positioning data. The value ranges from 0 to 324,000,000, covering the range from 0 to 180 degrees.

The conversion method is the same as the latitude conversion method.

5.2.1.8. velocity

Occupies 1 byte to indicate the GPS operating speed, with values ranging from 0x00 to 0xFF, representing a range of 0 to 255 km/h.

in compliance with :

0x00 represents 0 km/h;

0x10 represents 16 km/h;

0xFF represents 255 km/h.

5.2.1.9. Status heading

Occupies 2 bytes. When converted to binary, the first 6 bits of the first byte represent the device status, while the last 2 bits and the 8 bits of the second byte represent the GPS direction of operation (0-360 degrees, with true north as 0 degrees and clockwise).

The first 8-bit binary bytes first 6 bits represent the status, while the next 2 bits and the second bytes 8-bit binary together form a 10-bit binary number representing the heading.

BYTE_1	Bit7	No definition
	Bit6	(Not used or ACC ON/OFF)
	Bit5	GPS real-time/differential positioning
	Bit4	GPS location
	Bit3	东经、**西**经
	Bit2	**南**纬、**北**纬
	Bit1	azimuth
	Bit0	
BYTE_2	Bit7	
	Bit6	
	Bit5	
	Bit4	
	Bit3	
	Bit2	
	Bit1	
	Bit0	

Note: The status information in the data packet reflects the state recorded at the time of the time bit.

For example, the value 0x15 0x4C converts to binary code 00010101 01001100.

BYTE_1 Bit7 0 (no definition available)

BYTE_1 Bit6 0 (ACC OFF) --0: ACC OFF, 1: ACC ON (Some items do not use this bit)

BYTE_1 Bit5 0 (Real-time GPS) — 0: Real-time GPS, 1: Differential positioning

BYTE_1 Bit4 1 (GPS located) — 0: GPS not located, 1: GPS located

BYTE_1 Bit3 0 （东经）——0：东经，1：**西**经

BYTE_1 Bit2 1 （**北**纬）——0：**南**纬，1：**北**纬

BYTE_1 Bit1 0

BYTE_1 Bit0 1

BYTE_2 Bit7	0	
BYTE_2 Bit 6	1	
BYTE_2 Bit 5	0	→ Course 332° (binary 0101001100 converts to decimal 332)
BYTE_2 Bit 4	0	
BYTE_2 Bit 3	1	
BYTE_2 Bit 2	1	
BYTE_2 Bit 1	0	
BYTE_2 Bit 0	0	

This indicates that GPS has been located, with real-time GPS, north latitude, east longitude, and heading 332°.

5.2.1.10. MCC

Mobile Country Code (MCC) of the mobile users country.

For example, the mobile country code of China is 460 (decimal): 0x01 0xCC (decimal 460 converted to hexadecimal, with 0 added to the left if the hexadecimal is less than four digits).

The value range here is 0x0000 to 0x03E7.

5.2.1.11. MNC

Mobile network code (MNC)

For example, China Mobiles is 0x00.

5.2.1.12. LAC

The Location Area Code (LAC) is contained in the Location Area Identity (LAI) and consists of two hexadecimal bytes. The valid range is 0x0001 to 0xFFFE, excluding the 0x0000 and 0xFFFF code groups (as specified in GSM specifications 03.03,04.08, and 11.11). A location area may contain one or more cells.

5.2.1.13. Cell ID

Mobile base station Cell Tower ID (Cell ID), with a value range of 0x000000 to 0xFFFFFF

5.2.1.14. Information Sequence Number

See Data Packet Format 4.5

5.2.1.15. Error Check

See Data Packet Format 4.6 for details

5.2.1.16. Stop bit

See Data Packet Format 4.7 for details

5.2.2. Server response

This message server does not need to respond

Note: When the GSM signal is abnormal, the 0x12 GPS/LBS data packet will be temporarily stored and later uploaded to the platform. This upload is not real-time but occurs after the device stops moving, with the stored location points being uploaded. For trajectory playback, the time in the location packet should be used as the reference. During real-time display, if the packet time is less than 5 minutes behind the current time, it should not be shown.

5.3. Heartbeat Pack

Heartbeat packets are data packets used to maintain the connection between the terminal and the server.

5.3.1. The terminal sends heartbeat packets (0x13) to the server

form			length (Byte)	instance
information content	start bit		2	0x78 0x78
	packet length		1	0 x 0A
	Agreement number		1	0 x 13
	status information	Terminal Information	1	0 x 4E
		the classification of voltage	1	0 x 06
		GSM signal intensity	1	0 x 64
		External voltage	1	0 x 0C
		language	1	0 x 02
	serial number		2	0 x 00 0 x 03
	error check		2	0 x FF 0 x 4F
	end bit		2	0 x 0D 0 x 0A

5.3.1.1. start bit

See Data Packet Format 4.1 for details

5.3.1.2. Package length

See Data Packet Format 4.2 for details

5.3.1.3. Agreement No.

See Data Packet Format 4.3 for details

5.3.1.4. Terminal Information

Occupies 1 byte and is converted to binary to represent terminal status information. A byte is considered as 8 bits, with the least significant bit (LSB) being 0 and the most significant bit (MSB) being 7. During transmission, the higher bits are sent first, followed by the lower bits. The specific meanings of each bit are as follows:

Position		Code meaning
BYTE	Bit 7	1: Oil and electricity disconnected

		0: Oil and electricity connected
	Bit 6	1: GPS location
		0: GPS not located
	Bit5- Bit 3	100: SOS alarm
		011: Low power alarm
		010: Power failure alarm
		001: Vibration alarm
		00: Normal
	Bit 2	1: External power connected
		0: Unpowered
	Bit 1	1 : ACC ON
		0 : ACC O FF
	Bit0	1: Defense
		0: Withdraw defense

Example: 0x4E corresponds to the binary code 01001110

Indicates the system is in standby mode, with ACC ON, external power connected, vibration alarm activated, GPS location confirmed, and fuel/electricity supply active.

5.3.1.5. the classification of voltage

There are seven voltage levels, ranging from 0 to 6, indicating the voltage from low to high.

0: Low power shutdown;

1: The battery is not enough to make calls and send text messages;

2: Low power is too low;

3-6: All devices are fully functional, with battery levels determining their display order.

5.3 .1.6. GSM signal strength

GSM signal strength ranges from 0 to 100. The higher the value, the stronger the signal.

0: No signal

100: Full scale

5.3.1.7. External voltage + language

For example, the external power supply voltage is 30 volts (30V), and the terminals current language setting is:

Chinese: 0x 1E 0x01;

English: 0x 1E 0x02;

Note: The reserved expansion bits in S11/S11C/W15L are not used for language configuration but serve to identify the devices operating mode and sleep status.

work pattern ：

0x00= Smart sleep mode;

0x01=Normal mode;

0x02= Deep sleep mode;

0x03= Scheduled return mode;

0x04= Remote power mode;

Device sleep status:

The device works normally as 0x 00;

When the device is in sleep mode, the sleep state bit matches the working mode bit. For example, in deep sleep mode, the reserved expansion bit is 0x02 0x02.

5.3.1.8. Information Sequence Number

See Data Packet Format 4.5

5.3.1.9. Error Check

See Data Packet Format 4.6 for details

5.3.1.10. Stop bit

See Data Packet Format 4.7 for details

5.3.2. The server responds to heartbeat packets

form		length （Byte）	instance
information content	start bit	2	0x78 0x78
	packet length	1	0x05
	Agreement number	1	0x13
	information sequence	2	0x00 0x01
	error check	2	0xE9 0xF1
	stop bit	2	0x0D 0x0A

After receiving the data packet from the terminal, the server responds with an empty data

packet.

Note: The information sequence number in the data packet must match the information sequence number sent by the response terminal.

5.3.2.1 Starting Position

See Data Packet Format 4.1 for details

5.3.2.2 Package Length

See Data Packet Format 4.2 for details

5.3.2.3 Agreement Number

See Data Packet Format 4.3 for details

5.3.2.4 Information Sequence Number

See Data Packet Format 4.5

5.3.2.5 Error Check

See Data Packet Format 4.6 for details

5.3.2.6 Stop position

See Data Packet Format 4.7 for details

5.4. Alarm Package (0x16)

5.4.1. The terminal sends alarm data packets to the server

form			length （Byte）	instance
information content	start bit		2	0x 78 0x78
	packet length		1	0 x25
	Agreement number		1	0 x1 6
	Date and time		6	0 x13 0x08 0x1D 0x11 0x0C 0x10
	GPS information	GPS satellite count	1	0 xC B
		latitude	4	0 x02 0x7A 0xCF 0xEB
		longitude	4	0 xCC 0x46 0x58 0x49
		velocity	1	0 x10
		Course, Status	2	0 x1 5 0x 4C
	LBS information	LBS length	1	0x 09
		MCC	2	0 x01 0xCC
		MNC	1	0 x00
		LAC	2	0 x28 0x7D
		Cell ID	3	0 x00 0x1F 0xB8
	status information	Terminal Information	1	0 x4E
		the classification of voltage	1	0 x04
		GSM signal intensity	1	0 x64
		report to the police	1	0 x00
		language	1	0x 02
	serial number		2	0 x00 0x03
	error check		2	0 x8C 0x59
	end bit		2	0 x0D 0x0A

5.4.1.1 Starting Position

See Data Packet Format 4.1 for details

5.4.1.2 Package Length

See Data Packet Format 4.2 for details

5.4.1.3 Agreement Number

See Data Packet Format 4.3 for details

5.4.1.4 Date and Time

See data packet format 5.2.1.4 for details

5.4.1.5 GPS satellite count

See data packet format 5.2.1.5 for details

5.4.1.6 Latitude

See Data Packet Format 5.2.1.6 for details

5.4.1.7 Longitude

See Data Packet Format 5.2.1.7 for details

5.4.1.8 Speed

See data packet format 5.2.1.8 for details

5.4.1.9 Route, Status

See data packet format 5.2.1.9 for details

5.4.1.10 LBS length

The LBS information content length is fixed at 0x09

5.4.1.11 MCC

See Data Packet Format 5.2.1.10 for details

5.4.1.12 MNC

See data packet format 5.2.1.11 for details

5.4.1.13 LAC

See Data Packet Format 5.2.1.12 for details

5.4.1.14 Cell ID

See Data Packet Format 5.2.1.13 for details

5.4.1.15 Terminal Information

See Data Packet Format 5.3.1.4 for details

5.4.1.16 Battery voltage level

See data packet format 5.3.1.5 for details

5.4.1.17 GSM signal strength

See Data Packet Format 5.3.1.6 for details

5.4.1.18 Alarm Type

Alarm type	0x00: Normal
	0x01: SOS alarm
	0x02: Power-off alarm
	0x03: Vibration Alert
	0x04: Fence alarm
	0x05: Fence alarm
	0x06: Speed limit violation
	0x09: Displacement alarm
	0x0E: Low battery alarm

	0 FWD: ACC engine stall warning
	0xFF: ACC ignition alarm

5.4.1.19 Language

Chinese: 0x01

English: 0x02

5.4.1.20 Information Sequence Number

See Data Packet Format 4.5

5.4.1.21 Error Check

See Data Packet Format 4.6 for details

5.4.1.22 Stop position

See Data Packet Format 4.7 for details

5.4.2. The server sends alarm data packets to the terminal for response.

form		length (Byte)	instance
information content	start bit	2	0x 78 0x78
	packet length	1	0x 05
	Agreement number	1	0x 16
	information sequence	2	0x 00 0x 05
	error check	2	0x 96 0x 68
	stop bit	2	0x 0D 0x 0A

5.4.2.1 Starting Position

See Data Packet Format 4.1 for details

5.4.2.2 Package Length

See Data Packet Format 4.2 for details

5.4.2.3 Agreement Number

See Data Packet Format 4.3 for details

5.4.2.4 Information Sequence Number

See Data Packet Format 4.5 for details

5.4.2.5 Error Check

See Data Packet Format 4.6 for details

5.4.2.6 Stop position

See Data Packet Format 4.7 for details

5.5 . LBS multi-base station positioning information (0x18)

form			length (Byte)
	start bit		2
	packet length		1
	Agreement number		1
information content	Date and time		6
	LBS information	MCC	2
		MNC	1
		LAC	2
		MCI	2
		MCISS	1
		MCI1	2
		MCISS1	1
		MCI2	2
		MCISS2	1

		MCI3	2
		MCISS3	1
		MCI4	2
		MCISS4	1
		MCI5	2
		MCISS5	1
		MCI6	2
		MCISS6	1
	Reserve expansion space		N

5.5.1. Date and Time

Same as the previous protocol description.

5.5.2. MCC

Same as the MCC description in the previous LBS information

5.5.3. MNC

Same as the MNC description in the previous LBS information

5.5.4. LAC

Same as the LAC description in the previous LBS information

5.5.5. MCI (Main Cell ID)

The mobile base station Cell Tower ID (Cell ID) ranges from 0x0000 to 0xFFFF.

5.5.6. MCISS (Main Cell ID Signal Strength)

The signal strength of the main cell ranges from 0x00 to 0xFF, where 0x00 indicates the weakest signal and 0xFF indicates the strongest signal.

5.6.7. MCI1~6 (Near Cell ID)

There are 6 adjacent cell base station codes, with values ranging from 0x0000 to 0xFFFF.

5.5.8. NCISS1 ~ 6 (Near Cell ID Signal Strength)

The signal strength of adjacent cell base stations corresponds one-to-one with the coding of 6 adjacent cell base stations. The value range is 0x00 to 0xFF.

The absolute value of signal strength is used here. A negative sign should be attached when taking the value.

5.5.9. Reserve expansion space

Currently this is empty

5.6 . Query address (0x1A)

5.6.1 The terminal sends a query address packet to the server

form			length (Byte)	instance
information content	start bit		2	0x 78 0x78
	packet length		1	0 x 2E
	Agreement number		1	0x1 A
	GPS information	Date and time	6	0x13 0x01 0x08 0x09 0x1E 0x0A
		GPS satellite count	1	0x CD
		longitude	4	0x02 0x6B 0x3F 0x3E
		latitude	4	0x0C 0x45 0x49 0x53
		velocity	1	0x00
		Course, bearing	2	0x14 0x8F
	phone code	phone code	21	0x31 0x33 0x38 0x30 0x30 0x31 0x39 0x39 0x38 0x38 0x35 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20
	Language status	obligate	1	0x00
		language	1	0x02
	information sequence		2	0x00 0x06
	error check		2	0x29 0xD1
	stop bit		2	0x0D 0x0A

5.6.1.1 Starting Position

See Data Packet Format 4.1 for details

5.6.1.2 Package Length

See Data Packet Format 4.2 for details

5.6.1.3 Agreement Number

See Data Packet Format 4.3 for details

5.6.1.4 Date and Time

See Data Packet Format 5.2.1.4 for details

5.6.1.5 GPS satellite count

See Data Packet Format 5.2.1.5 for details

5.6.1.6 Latitude

See Data Packet Format 5.2.1.6 for details

5.6.1.7 Longitude

See Data Packet Format 5.2.1.7 for details

5.6.1.8 Speed

See Data Packet Format 5.2.1.8 for details

5.6.1.9 Route, Status

See Data Packet Format 5.2.1.9 for details

5.6.1.10 Phone number

The phone number must be 21 digits long. If it is shorter, add leading zeros, for example: 0x20

For example: 13800138000, device report: 0x31 0x33 0x38 0x30 0x30 0x31 0x33 0x38

0x30 0x30 0x30 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20 0x20

5.6.1.11 Reservation + Language

Chinese: 0x00 0x01

English: 0x00 0x02

5.4.1.20 Information Sequence Number

See Data Packet Format 4.5

5.4.1.21 Error Check

See Data Packet Format 4.6 for details

5.4.1.22 Stop position

See Data Packet Format 4.7 for details

5.9 . The terminal sends the ICCID number to the server (0x94)

form			length (Byte)	description
information content	start bit		2	Fixed value: 0x 7 9 0x7 9
	packet length		2	0 x20, length = protocol number + message content + message sequence number + error check
	Agreement number		1	Fixed value: 0 x9 4
	Data content	type of information	1	00: External voltage 01~03: (Customized for customers) 04: Terminal status sync 05: Door status 08: Self-Check Parameters 09: Satellite positioning 0A: ICCID information ...to be added
		Data content	N	The content varies depending on the information type. For details, see the table below.
		ICCID	10	For example, if the ICCID is 12345123456789123456, the terminal ID is 0x12 0x34 0x51 0x23 0x45 0x97 0x89 0x12 0x34 0x56
	Message serial number		2	Each data transmission automatically increments by 1 after startup.
	error check		2	CRC-ITU value from "Packet Length" to "Information Sequence Number"
	end bit		2	0 x0D 0x0A

When the type is 0A, ICCID information is transmitted in hexadecimal format.

IMEI	8	For example, if the IMEI is 123456789123456, the terminal ID is 0x01 0x23 0x45 0x67 0x89 0x12 0x34 0x56
IMSI	8	For example, if the IMSI is 123456789123456, the terminal ID is 0x01 0x23 0x45 0x67 0x89 0x12 0x34 0x56
ICCID	10	For example, if the ICCID number is 12345123456789123456, the terminal ID is 0x12 0x34 0x51 0x23 0x45 0x97 0x89 0x12 0x34 0x56.

Example of uploading the ICCID number to the device:

79790020940a035809108800155804600419902053138986 07b9111730120313 0009a0720d0a

7979 0020 94 0a 0358091088001558 0460041990205313 _ _

Start position Length Protocol number Type IMEI IMSI

898607b9111730120313 0009 a072 0d0a

ICCID number, information sequence, error check, stop bit

Note: This protocol requires no server response.

5.10 . Recording Protocol Packet (0x 8D)

Record audio and send the recording package to the server. This applies to S 709, S 11, S11C, and W15L devices.

5.10 .1. The terminal sends data packets to the server:

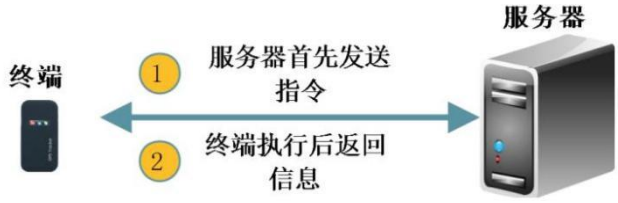
form			length (Byte)	description
information content	start bit		2	Fixed value: 0x 7 9 0x7 9
	packet length		2	Length = Protocol number + Message content + Message sequence number + Error check
	Agreement number		1	Fixed value: 0 x 8D
	Recorded audio	document type	1	Fixed value: 0x 00
		Total file length	4	Total transfer file length
		Total package count	1	How many packets are there in the transfer file
		Current package sequence of the file	1	Transfer the current package sequence
		Current content length	2	Transfer the data length from the start
		Data content	M	Content of split data packets
	Message serial number		2	Each data transmission automatically increments by 1 after startup.
	error check		2	CRC-ITU value from "Packet Length" to "Information Sequence Number"
	end bit		2	0 x0D 0x0A

5.10 .2. Server responds to terminal:

form		length (Byte)	description
information content	start bit	2	Fixed value: 0x 7 9 0x7 9
	packet length	2	Length = Protocol number + Message content + Message sequence number + Error check
	Agreement number	1	Fixed value: 0 x 8D
	Receive status flag	1	0x00: Received normally 0x01: Received error
	Message serial number	2	Each data transmission automatically increments by 1 after startup.

	error check	2	CRC-ITU value from "Packet Length" to "Information Sequence Number"
	end bit	2	0 x0D 0x0A

The device must receive a server response for each recording file packet before sending the next one. If the same packet is sent three times without a response, the device will abandon the recording file transmission.



6. The server sends data packets to the terminal

6 .1. The server sends a command (0x 80)

form		length (Byte)
start bit		2
packet length		1
Agreement number		1
information content	command length	1
	Server flag	4
	Instruction content	N
Message serial number		2
error check		2
stop bit		2

6.1.1. start bit

See Data Packet Format 4.1 for details

6.1.2. Package length

See Data Packet Format 4.2 for details

6.1.3. Agreement No.

Fixed value: 0x80

6.1.4. Instruction Length

Instruction length = server flag + instruction content length

For example, 0x0A in byte length indicates that the instruction content occupies 10 bytes.

6.1.5. Server Flags

The server uses this for identification, and the terminal returns the original binary data in the return packet.

6.1.6. Instruction content

The instruction content is represented in ASCII format and is compatible with SMS instructions.

6.1.7. Information Sequence Number

See Data Packet Format 4.5

6.1.8. Error Check

See Data Packet Format 4.6 for details

6.1.9. Stop

See Data Packet Format 4.7 for details

6.2 .Return data packet (0x15)

form		length (Byte)
start bit		2
packet length		1
Agreement number		1
information content	command length	1
	Server flag	4
	Command content	N
	language	2
Message serial number		2
error check		2
stop bit		2

6.2.1. Starting position

See Data Packet Format 4.1 for details

6.2.2. Package Length

See Data Packet Format 4.2 for details

6.2.3. Agreement No.

The terminal responds to the command sent by the server. The data packet format is the

same as the "command sent by the server to the terminal", but the protocol number is different and uses 0x15.

6.2.4. Instruction Length

Instruction length = server flag + instruction content length

For example, 0x0A in byte length indicates that the instruction content occupies 10 bytes.

6.2.5. Server Flags

The terminal returns the raw binary data in the return packet.

6.2.6. Instruction content

represented by the ASCII value of the string

6 .2.7.language

The language used at the end

Chinese: 0x00 0x01

English: 0x00 0x02

6.2.8. Information Sequence Number

See Data Packet Format 4.5

6.2.9. Error Check

See Data Packet Format 4.6 for details

6.2.10 。 stop bit

See Data Packet Format 4.7 for details

6 .2 .11 Server command issuance example

6 .2 .11.1 Oil Cut-off Switch

SMS message format:

DYD,000000#

Function description: Cut off the vehicles fuel and electrical control circuit

return information :

Return: DYD=Success!

Failure return: DYD=Unvalued Fix!

or DYD=Speed Limit, Speed 40km/h

For example, the platform sends: 78 78 15 80 0F 00 01 A9 58 44 59 44 2C 30 30 30 30 30 30 23 00 A0 DC F1 0D 0A

Bolded part: DYD, 000000#

Device response: 78 78 18 15 10 00 01 A9 58 44 59 44 3D 53 75 63 63 65 73 73 21 00 02 00 18 91 77 0D 0A

Bolded text: DYS=Success!

6 .2 .11.2 Recharge oil and electricity

SMS message format:

HFYD,000000#

Function description: Connect the vehicles fuel and electrical control circuit

return information :

Return: HFYD=Success!

Failure return: HFYD=Fail!

The platform sends a response to the device, similar to DYD.

6 .2 .11.3 View Location

instruction format :

DWXX,000000#

functional description :

Get location information. Both mobile users and SMS servers can use this command to obtain location information.

return information :

Successful return: DWXX=Lat: <south>/<north latitude>>, Lon: <east/west longitude>>, Course: <angle>>, Speed: <speed>>, Date: <time>>

Failure return: DWXX=Command Error!

for instance :

Lat:N23d5.1708m,Lon:E114d23.6212m,Course:120,Speed:53.02;DateTime:08-09-12 14:52:36

The meaning is: north latitude 23 degrees 5.1708 minutes, east longitude 114 degrees 23.6212 minutes, angle: 120 degrees, speed: 53.02 km/h, time and date: 12 September 200814:52:36.

Note: If the terminal fails to locate, the following data will be returned: Lat:, Lon:, Course:, Speed:, Date-Time: -:

The platforms message format and device response are identical to DYD.

Upload: 78 78 0D 0 1 08 68 02 00 41 12 74 64 00 1E A0 1E 0D 0A

Distribute: 78 78 05 01 00 2B 57 84 0D 0A

Upload: 78 78 1F 12 16 06 11 07 38 35 CF 02 6C 6C F7 0C 37 15 20 0B 14 76 01 CC 00 26 33 00 0E 81 00 1D B3 A8 0D 0A

Upload: 78 78 0A 13 44 06 03 0B 02 00 1F B4 E2 0D 0A

Distribution: 78 78 05 13 00 2B 67 A9 0D 0A

Upload: 79 79 00 20 94 0A 08 68 02 00 41 12 74 64 04 60 04 04 21 92 04 69 89 86 02 B4 03 19 C0 01 04 69 00 20 41 C9 0D 0A

Upload: 78 78 25 16 16 06 11 08 12 25 C0 02 6C 69 7B 0C 37 1C 92 00 04 00 08 01 CC 00 26 33 00 10 04 10 04 04 02 02 00 11 41 56 0D 0A

Upload: 78 78 25 18 16 06 11 08 10 34 01 CC 00 26 33 0E 8B 35 0F 5D 2B 10 04 2A 0E 89 2A 12 19 29 0F FC 27 12 17 26 00 0D 6E 60 0D 0A

Distribute: 78 78 2E 1A 16 06 11 08 15 1A CF 02 6C 6E E6 0C 37 14 0C

0014002020202020202020202020202020202020202020202020202020202020202020202020202020

202020202020202020202020202020202020202020202020202020202020202020202020202020

Upload: 78 78 15 80 0F 00 01 A9 58 44 59 44 2C 30 30 30 30 30 30 23 00 A0 DC F1 0D 0A

Upload: 78 78 18 15 10 00 01 A9 58 44 59 44 3D 53 75 63 63 65 73 73 21 00 02 00 0A A2 E4 0D 0A

7. Appendix A: U CRC-ITU Lookup Algorithm C C Language Code Snippet

CRC-ITU lookup algorithm C 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,
};
```

Calculate the 16-bit CRC for data of a given length.

```
U16 GetCrc16(const U8* pData, int nLength)
{
U16 fcs = 0xffff; // Initialize
while(nLength>0){
fcs = (fcs >> 8) ^ crctab16[(fcs ^ *pData) & 0xff];
nLength--;
pData++;
}
return ~fcs; // invert
}
```