

Xexun API

Note: This is the API interface documentation for communication between the Shenzhen Xixun GPS tracker and the platform.

Data types

Data types	describe
U8	Unsigned single-byte integer (byte, 8 bits)
U16	Unsigned two-byte integer (byte, 16 bits)
U32	Unsigned four-byte integer (two bytes, 32 bits)
FLOAT	Single-precision floating-point number (4 bytes, 32 bits)
DOUBLE	Double-precision floating-point data type (8 bytes)
U8 [n]	N bytes
I 8	Signed 8-bit integer
BCD [n]	Binary-encoded decimal number (BCD code), n bytes
STRING	If it's UTF-8 encoded, please leave it blank if it's data.

Message body and message header only: All data in FF format is invalid, but only some data in F format is valid .

All multibyte data types uniformly adopt big-endian order .

- The most significant byte comes first (low address), and the least significant byte comes last (high address) .

- It maintains consistency with network byte order, facilitating network transmission .

Message Structure

Each message consists of "Header Flag + total length + message header + message body + CRC + end flag" .

Header/End Flag

Header Flag is FC

End Flag is CF

Total length

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Total length	U16	2	16-to-10 conversion	00 0C	12

The length is the length of " message header + message body" .

Message header

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Protocol version number	U8	1	16-to-10 conversion	03	3
2	Message Type	U 8	1	Use directly	20	20
3	Message sequence number	U 8	1	16-to-10 conversion	20	32
4	0+ IMEI	BCD	8	Remove the first zero to get the IMEI	0 8 62 77 10 75 42 67 52	86 27 71 07 54 26 75 2

Message type : 20 for data uploaded by the terminal and the platform's reply , 21 for data sent by the server and the terminal's reply .

CRC

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	CRC check	U16	2	16-to-10 conversion	1A 30	6704

CRC is a checksum for the message header and message body .

The CRC16 check uses the CRC-16/CCITT-FALSE algorithm, with an initial value of 0xFFFF .

Message body (0x20)

serial number	name	type	byte count
1	Message ID	U8	1
2	Message length	U8	1
3	Message content	*	*

Note: The message body of a message (0x20 type) can be composed of one or more functional message bodies (such as GPS message body, status message body, alarm message body, etc.) concatenated together. Each functional message body follows the structure of " Message ID + Message Length + Message Content " .

GPS message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	64	1 00
2	Message length	U8	1	16-to-10 conversion	20	32
3	Track successful timestamps	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	Latitude (FF) FF FF FF FF FF FF (FF is invalid)	DOUBLE	8	16-to -10 coordinates	40 38 91 68 72 B0 20 C9	22.34448 00000
5	Longitude (FF) FF FF FF FF FF FF (FF is invalid)	DOUBLE	8	16-to -10 coordinates	40 38 2B 02 0C 49 BA 5E	114.02428 70000
6	altitude	FLOAT	4	Convert hexadecimal FLOAT data to decimal coordinates	42 4F 99 9A	51.90
7	Ephemeris	U8	1	16-to-10 conversion	06	6
8	Number of satellites	U8	1	16-to-10 conversion	06	6
9	Average signal of the 4 best satellites	U8	1	16-to-10 conversion	twenty three	35
10	Speed (km/h)	U16	2	16-to-10 conversion	00 20	32
1 1	angle	U16	2	16-to-10 conversion	00 30	48
1 2	Precision factor /	U 16	2	16-to-10	00 01	1

	decimeter			conversion		
13	Tracking duration/ second	U8	1	16-to-10 conversion	07	7

WIFI message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	65	101
2	Message length	U8	1	16-to-10 conversion	2F	47
3	timestamp of successful sampling	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	Number of WIFI	U8	1	16-to-10 conversion	06	6
5	WIFI MAC	U8[6]	6	Use directly	F4 BF BB 0D 4D F1	F4 : BF : BB : 0D : 4D : F1
6	WIFI signal	I 8	1	Converting from hexadecimal to decimal, the value range of I8 is -128 to 127.	CC	-52
7	WIFI MAC	U8[6]	6	Use directly	A0 C5 F2 B0 78 E0	A0 : C5 : F2 : B0 : 78 : E0
8	WIFI signal	I 8	1	Converting from hexadecimal to decimal, the value range of I8 is -128 to 127.	CC	-52
9	WIFI MAC	U8[6]	6	Use directly	A2 C5 F2 B0 78 E0	A2 : C5 : F2 : B0 : 78 : E0
10	WIFI signal	I 8	1	Converting from hexadecimal to decimal, the value range of I8 is -128 to 127.	CC	-52
11	WIFI MAC	U8[6]	6	Use directly	7C 1E 06 82 D5 71	7C : 1E : 06 : 82 : D5 : 71
12	WIFI signal	I 8	1	Converting from hexadecimal to	CC	-52

				decimal, the value range of I8 is - 128 to 127.		
13	WIFI MAC	U8[6]	6	Use directly	0C 4B 54 3F 28 81	0C : 4B : 54 : 3F : 28 : 81
14	WIFI signal	I 8	1	Converting from hexadecimal to decimal, the value range of I8 is - 128 to 127.	C4	-60
15	WIFI MAC	U8[6]	6	Use directly	0E 4B 54 3F 28 81	0E : 4B : 54 : 3F : 28 : 81
16	WIFI signal	I 8	1	Converting from hexadecimal to decimal, the value range of I8 is - 128 to 127.	C4	-60

Note: WiFi tracking requires a Google account and access to Google's paid API.

LBS base station message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	66	102
2	Message length	U8	1	16-to-10 conversion	19	25
3	timestamp of successful sampling	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	MCC (Mobile Country Code)	U16	2	16-to-10 conversion	01 CC	460
5	MNC (Mobile Network Code)	U16	2	16-to-10 conversion	00 00	00
6	LAC (Location Area Code)	U32	4	16-to-10 conversion	00 00 26 2E	9774
7	CID (Cell Identity)	U32	4	16-to-10 conversion	0A 82 AB 1A	176335642
8	RSSI (FF indicates invalid)	U8	1	16-to-10 conversion	twenty three	35

LBS location message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	67	103
2	Message length	U8	1	16-to-10 conversion	19	25
3	timestamp of successful sampling	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	latitude	DOUBLE	8	16-to -10 coordinates	40 38 91 68 72 B0 20 C9	22.3444800000
5	longitude	DOUBLE	8	16-to -10 coordinates	40 38 2B 02 0C 49 BA 5E	114.0242870000

Beacon message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	68	104
2	Message length	U8	1	16-to-10 conversion	0D	13
3	Sampling timestamp	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	WiFi/ Bluetooth identifiers : 1 for WiFi, 2 for Bluetooth.	U8	1	16-to-10 conversion	01	01
5	Number of beacons	U8	1	16-to-10 conversion	0 2	2
6	Beacon ID	U8 [6]	6	Use directly	F4 BF BB 0D 4D F2	F4:BF:BB:0D:4D:F 2
7	Beacon signal (FF indicates invalid)	I8	1	16-to-10 conversion I8 is -128 to 127.	CC	-52
8	beacon power	U8	1	16-to-10 conversion	30	30%
9	Beacon ID	U8 [6]	6	Use directly	F4 BF BB 0D 4D F1	F4:BF:BB:0D:4D:F1
10	Beacon signal (FF indicates invalid)	I8	1	16-to-10 conversion I8 is -128 to 127.	CC	-52
11	beacon power	U8	1	16-to-10 conversion	30	30%

Alarm message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	69	105
2	Message length	U8	1	16-to-10 conversion	08	8
3	alarm timestamp	U32	4	Hexadecimal to decimal conversion: Number of seconds from 00:00:00 UTC on January 1, 1970 to the target time.	66 75 36 0C	June 4, 2024, 15:13:00 UTC
4	Alarm ID	U 8	1	16-to-10 conversion	01	01, which is the SOS alarm.
5	Alarm data length	U8	1	16-to-10 conversion, value is 0 when there is no alarm.	12	1 8
6	alarm value	DAT A	*	Analysis by alarm value type	89 85 20 19 92 40 70 00 68 10	8985201992 4070006810

Alarm ID

ID	Call the police	alarm value	ID	Call the police	alarm value
1	SOS alarm	null	2	Low battery alarm for power bank	null
3	Wearing alarm	null	4	Not wearing an alarm	null
5	Mobile alarm	Motion value (u16)	6	Static alarm	Motion value (u16)
7	Fall alarm	Motion value (u16)	8	Impact alarm	Motion value (u16)
9	Beacon alarm detected	null	10	beacon not detected alarm	null
11	Vehicle power connection alarm	null	12	Vehicle power supply not connected alarm	null
1 3	ACC start alarm	null	1 4	ACC stop alarm	null
1 5	Car door opening	null	1 6	Car door closing alarm	null

	alarm				
17	ICCID	ICCID (BCD)	18	GPS fault alarm	null

Status message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	6A	106
2	Message length	U8	1	16-to-10 conversion	05	05
3	Battery	U8	1	16-to-10 conversion	1E	30
4	Communication network duration	U 16	2	16-to-10 conversion	00 1E	30
5	CSQ (0-31 , with the strongest signal at 31)	U8	1	16-to-10 conversion	1F	31
6	Track serial number (non-location data is not parsed, default is FF)	U8	1	16-to-10 conversion	09	9
7	Motion value (0-80)	U8	1	16-to-10 conversion	00	0
8	Charging status (1 Charging, 0 Not charging)	U8	1	16-to-10 conversion	00	0
9	Step counting (FF indicates invalid)	U 16	2	16-to-10 conversion	36 0C	13836
10	Temperature (FF indicates invalid)	I8	1	16-to-10 conversion I8 is -128 to 127.	FF	
11	Respiratory rate (FF indicates invalid)	U8	1	16-to-10 conversion	FF	
12	Heart rate (FF indicates invalid)	U8	1	16-to-10 conversion	FF	
13	Blood pressure - systolic pressure (FF indicates invalid)	U8	1	16-to-10 conversion	FF	
14	Blood pressure - diastolic pressure (FF indicates invalid)	U8	1	16-to-10 conversion	FF	

15	Blood oxygen (FF indicates invalid)	U8	1	16-to-10 conversion	FF	
16	Fuel level (FF indicates invalid)	U8	1	16-to-10 conversion	FF	
17	Number of sleep rollovers (FF indicates invalid)	U 16	2	16-to-10 conversion	00 1E	30

Heartbeat message body

serial number	name	type	byte count	Parsing method	Example: Before parsing	Example: After parsing
1	Message ID	U8	1	Use directly	6B	107
2	Message length	U8	1	16-to-10 conversion	05	05
3	Heartbeat packet, sent regularly 1	U8	1	16-to-10 conversion	01	1

Example data

The terminal sent :

FC 00 40 03 20 06 08 61 04 50 80 38 70 15 64 21 69 13 F2 23 40 36 93 01 2F 63 53 44 40 5C 82 91 42 B3 02 F7 42 7F 33 33 1A 2E 00 00 00 A4 00 11 04 6A 10 55 FF FF 1F 00 00 00 00 00 FF FF FF 04 FF 09 FF 1A 30 CF

Analysis : FC is the Header Flag , 00 40 is the message length, 03 is the protocol version, 20 is the message type , 06 is the serial number , 08 61 04 50 80 38 70 15 is the IMEI (861045080387015), 64 21 69 13 F2 23 40 36 93 01 2F 63 53 44 40 5C 82 91 42 B3 02 F7 42 7F 33 33 1A 2E 00 00 00 A4 00 11 04 represents GPS data, 6A 10 55 FF FF 1F 00 00 00 00 00 FF FF FF 04 FF 09 FF represents status data, 1A 30 represents CRC , and CF represents the End Flag .

Note: The receiver must verify the consistency between the total length and the sum of the lengths of each sub-message. If they do not match, the data packet should be discarded and an error log should be logged.

Platform Response (ACK) :

FC 00 0C 03 20 06 08 61 04 50 80 38 70 15 00 A5 11 CF

Explanation: FC is the Header Flag , 00 0C 03 is the message length , 20 is the protocol version , 06 is the message type , 06 is the serial number (must be the same as the serial number uploaded by the terminal), 08 61 04 50 80 38 70 15 is the IMEI (861045080387015) , 00 is the body (fixed to 00) , A5 11 is the CRC, and CF is the End Flag .

receiving a message from a device, the server must reply with an ACK; otherwise, the device must resend the message.

The server issues a command (0x21)

The server-issued command consists of " Header Flag + length + message header + message body (command text only , case-insensitive) + CRC + end flag " .

Function	Instruction text	illustrate
Set APN	APN=Internet	Configure the Internet's APN on the terminal.
Tracking method and frequency	Settings : tk= 2,60,2, 360 0,0, 2 0 Query : tk=?	Parameter 1: 1 for GPS priority, 2 for GPS tracking only , 3 for Wi-Fi priority Parameter 2: Tracking interval within the virtual fence during movement, ranging from 5 to 300 seconds. Parameter 3: Data reporting delay time after tracking , value 0-240 (seconds) Parameter 4: Tracking interval when stationary , ranging from 60 to 86400 (seconds). Parameter 5: Maintain connection with the platform when not in motion : 1 for connection, 0 for disconnection Parameter 6 : Tracking interval outside the virtual fence during movement, ranging from 5 to 300 seconds.
IP transfer	Settings : ip=p.xexun.com:8899 Query : ip=?	The domain name is p.xexun.com , and the port is 8899.
Power off	of=1	The device shuts down after receiving the message . After shutting down, it can only be turned on by charging the device or pressing a button .
Restart	rt=1	The device will restart after receiving the message.
information	mg=TEXT MSG	Devices with screens can display received messages.
Time zone	Settings : tz=8 ; Query : tz=?	Set the time zone, ranging from -12 to +12.
Set up Bluetooth beacon tracking	Settings : ble ={"bssid_list":["0c:4b:54:3f:29:01", "7c:1e:06:82:d5:71"]} Delete : ble = {} ; Query : ble = ?	The terminal receives a BLE ID of "0c:4b:54:3f:29:01", "7c:1e:06:82:d5:71" and tracks it at the marked location .

Set up WiFi hotspot tracking	Settings : wifi={"bssid_list":["0c:4b:54:3f:29:01","7c:1e:06:82:d5:71"]} Delete : wifi={} ; Query : wifi=?	The terminal receives Wi-Fi IDs of "0c:4b:54:3f:29:01" and "7c:1e:06:82:d5:71" and tracks them at the marked locations.
-------------------------------------	--	---

Example : The server sends the command " tk =2,60,2, 360 0,0, 2 0 " to 0862688071460355:

FC 00 20 03 21 01 08 62 68 80 71 46 03 55 74 6b 3d 32 2c 36 30 2c 32 2c 33 36 30 30 2c 30 2c 32 30 00 EB F8 CF

Analysis: FC is the Header Flag , 00 20 is the message length, 03 is the protocol version, 21 is the command issuance ID, 01 is the serial number, 08 62 68 80 71 46 03 55 is the IMEI 862688071460355 , 74 6B 3D 32 2C 36 30 2C 32 2C 30 2C 30 2C 31 38 30 2C 31 30 00 is the command content, EB F8 is the CRC, and CF is the end flag.

Note: The "text to hexadecimal" message sent must end with 00 , but the terminal response does not . For example, the text " tk=2,60,2, 360 0,0, 2 0 " becomes " 74 6b 3d 32 2c 36 30 2c 32 2c 33 36 30 30 2c 30 2c 32 30 " in hexadecimal. 00 ".

After receiving the server command, the terminal replies with ACK.

Example: After receiving the command " tk=2,60,2,0,0,180,10 " from the server, terminal 0862688071460355 replies:

F7 74 CF

FC is the Header Flag , 00 1F is the message length, 03 is the protocol version number, 21 is the command issuance ID, 01 is the serial number, 08 62 68 80 71 46 03 55 is the IMEI 862688071460355 , 74 6B 3D 32 2C 36 30 2C 32 2C 30 2C 30 2C 31 38 30 2C 31 30 is the successful return result I set (hexadecimal converted to string: tk=2,60,2,0,0,180,10) , F7 74 is the CRC, and CF is the End Flag.

Note: When using # to connect multiple commands, the reply content will also be separated by # in sequence.

After receiving a command from the server, the device must respond with an ACK. The serial number is the serial number of the command issued by the server.

ACK consists of " Header Flag + length + message header + message body (returns the result of the instruction execution on success, FAIL on failure, and unknown if not supported) + CRC + end flag ".