MEITRACK GPRS(CCE ID) Protocol

Applicable Model: All Meitrack models

Change History

File Name	MEITRACK GPRS(CCE ID) Protocol		
Project	T633L\T399L\T711L\MD600\MD500S\MD833H\MD300\MD300A	Creation Date Update Date	2025-03-30
Subproject	GPRS Protocol	Total Pages	53
Version	V1.5	Confidential	Internal Documentation

Copyright and Disclaimer

Document update history

Version	date	Revise
1.0	2025-03-30	First draft.
1.1	2025-10-10	1. Added CCEID:0XFEA1 Ignition Off or Parking Accumulated Time 2. Added CCEID:0X72 Storage Status 3. Added CCEID:0X39 Magnetic Card Reader Information 4. Added CCEID:0xF82E Overspeed Period Information Statistics 5. Added CCEID:0X6A Ignition Off Event Auxiliary Message 6. Added CCEID:0XDB Temperature Sensor Information 7. Added Event Code: 118: Input 9 Active 8. Added Event Code: 119: Input 9 Inactive 9. Added Event Code: 39: Photo 10. Added Event Code: 44: GSM Jamming 11. Added Event Code: 63: No GSM Jamming 12. Added Event Code: 122: SD State 13. Added Event Code: 144: Ignition On 14. Added Event Code: 145: Ignition Off
1.2	2025-11-06	1. Added CCEID:0XFEA0 driving state 2. Added CCEID:0XFE30 Over-speed Recovery Event Assistance Message

		3. Added CCEID:0XFE2F Driving under the influence of fatigue 4. Added CCEID:0X4C Set of analog quantities
1.3	2025-11-18	Add CCE ID 0x40
1.4	2026-02-27	Add CCE ID 0x6E: Camera connection information (cache)
1.5	2026-04-09	Add CCE ID 0XFE2A 0XFE25

Contents

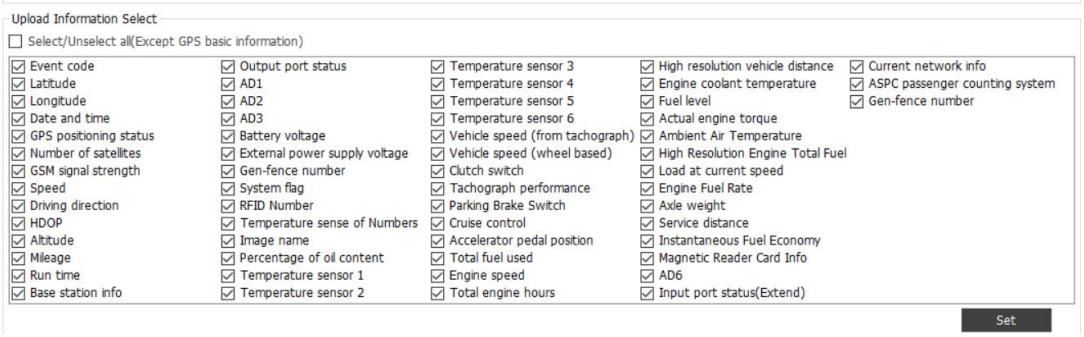
1 Command Format

1.1 Tracker Command Format

The data format is as follows:

$$<*Data identifier*><*Data length*>,<*IMEI*>,<CCE>,<*Number of remaining cache records*><*Number of data packets*><*Data packet 1*><*Data packet 2*><*Data packet 3*>…<*Data packet N*><Checksum*>\r\n**

There is one or multiple data packets. When there are multiple data packets, the data is stored in the form of cache in the flash memory after the network is disconnected. After the network is connected normally, the tracker sends the cached data to the server in batches. When there is only one data packet, it means that this is a piece of real-time data.



The following is an example of data including only one data packet. The command content in CCE format is as follows:

Parameter	Description	Example
@@ / $$	@@: Indicates the GPRS data packet header sent from the server to the tracker. The header type is ASCII (hexadecimal: 0x40). $$: Indicates the GPRS data packet header sent from the tracker to the server. The header type is ASCII (hexadecimal: 0x24).	Hexadecimal: 0x24 0x24 ASCII: $$
Data identifier	Contains 1 byte. The type is the ASCII, and its value ranges from **0x41** to **0x7A**.	Hexadecimal: 0x47 ASCII: G
Data length	Unit: byte. Type: decimal. Indicates the length of characters from the first separator "," to the ending character "\r\n" (including "," and "\r\n"), that is, the content underlined below. $$<*Data identifier*><*Data length*>,<*IMEI*>,<*Command type*>,<*Number of remaining cache records*><*Number of data packets*><*Data packet 1*><*Data packet 2*><*Data packet 3*>…<*Data packet N*><**Checksum*>\r\n	Hexadecimal: 0x32 0x30 0x35 ASCII: 205
IMEI	Indicates the tracker's IMEI number. It has 15 digits generally.	Hexadecimal: 0x38 0x36 0x38 0x39 0x39 0x38 0x30 0x33 0x33 0x320x 34 0x32

			0x33 0x33 0x36 ASCII: 868998033242336
CCE		CCE	Hexadecimal: 0x43 0x43 0x45 ASCII: CCE
The following data is hexadecimal:			
Number of remaining cache records		Contains 4 bytes; hexadecimal; little-endian	0x00 0x00 0x00 0x00 The number of remaining cache records is 0.
Number of data packets		Contains 2 bytes; hexadecimal; little-endian	0x01 0x00 The entire message contains only data packet.
The following is the detailed data of each data packet which needs to be parsed by using hexadecimal numbers. This protocol only describes data including only one data packet.			
Length of the current data packet		Contains 2 bytes; hexadecimal; little-endian. Indicates the length of characters from the "total number of ID in the current data packet" parameter to the last parameter ID of the current data packet.	0xAB 0x00 The length of the current data packet is 171 bytes.
Total number of ID in the current data packet		Contains 2 bytes; hexadecimal; little-endian	0x2E 0x00 There are 46 ID numbers in the data packet.
Number of 1-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 1 byte. The following data is not fixed, customers can choose to upload the required data.	0x0E There are 14 parameter ID numbers whose length is 1 byte. 0x00: The current data packet does not contain any parameter ID number whose length is 1 byte.
Event code	Parameter ID: 0x01	For details, see the section "Event Code." Data type: BYTE The ID: 0x01 parameter is reserved for the old model Old model: T366L\P99L\T399L(old FW)	0x23 The event code is 35.
GPS positioning status	Parameter ID: 0x05	0x01: The GPS positioning is valid. 0x00: The GPS positioning is invalid. Data type: BYTE	0x01 The GPS positioning is valid.
Number of satellites	Parameter ID: 0x06	Indicates the number of received GPS satellites. Data type: BYTE	0x0A The number of received GPS satellites is 10.
GSM signal strength	Parameter ID: 0x07	Value range: 0x00–0x31 Data type: BYTE	0x1C The GSM signal strength is

			28.
Output port status	Parameter ID: 0x14	Indicates the status values of eight output ports. Bits 0–7 correspond to status of output ports 1–8. Data type: BYTE	0x00 Converted to binary digits: 0000 0000 Output ports 1–8 is inactive.
input port status	Parameter ID:0x15	Indicates the status values of eight input ports. Bits 0–7 correspond to status of input ports 1–8. Data type: BYTE	0x00 Converted to binary digits: 0000 0000 input ports 1–8 is inactive.
Geo-fence number	Parameter ID: 0x1B	The data is available only when the GPRS event code is 20 or 21. Data type: BYTE	0x00 No Enter Geo-fence or Exit Geo-fence alert is generated.
Temperature sensor No.	Parameter ID: 0x27	07 Indicates temperature sensor 7. The data is available only when the GPRS event code is 50 or 51. Data type: BYTE	
Dead Reckoning state	Parameter ID：0x5B	00：invalid 01：Start the Dead Reckoning(DR) Data type：BYTE	0x00 Indicates that the DR Function is not enabled.
Clutch switch	Parameter ID: 0x93	01: The clutch pedal is pressed. 00: The clutch pedal is released. Upload data after reading it. Data type: BYTE	0x00 The clutch pedal is released.
Tachograph performance	Parameter ID: 0x94	01: performance analysis 00: normal performance Upload data after reading it. Data type: BYTE	0x00 The tachograph performance is normal.
Parking brake switch	Parameter ID: 0x95	01: Apply the brake. 00: Do not apply the brake. Upload data after reading it. Data type: BYTE	0x00 Do not apply the brake.
Cruise control system	Parameter ID: 0x96	01: The cruise control system is switched on. 00: The cruise control system is switched off. Upload data after reading it. Data type: BYTE	0x00 The cruise control system is switched off.
Accelerator pedal position (%)	Parameter ID: 0x97	1-byte hexadecimal data Upload data after reading it. Data type: BYTE	0x14 You press down 20% of the accelerator pedal of your vehicle.
CAN bus fuel	Parameter	1-byte hexadecimal data	0x23

level (%)	ID: 0x9D	Upload data after reading it. Data type: BYTE	The fuel level left is 35%.
Actual engine torque (%)	Parameter ID: 0x9E	1-byte hexadecimal data Upload data after reading it. Data type: SINT8	0x12 The actual engine torque is 18%.
Actual engine torque at current speed (%)	Parameter ID: 0xA1	1-byte hexadecimal data Upload data after reading it. Data type: BYTE	0x12 The actual engine torque at current speed is 18%.
driving state	Parameter ID: 0XFEA0	0: Engine off state; 1: Engine starting and parking state; 2: Engine starting and driving state;	
Number of 2-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 2 bytes. The following data is not fixed, customers can choose to upload the required data.	0x10 There are 16 parameter ID numbers whose length is 2 bytes. 0x00: The current data packet does not contain any parameter ID number whose length is 2 bytes.
Speed	Parameter ID: 0x08	Unit: km/h; little-endian Data type: WORD	0x15 0x00 The driving speed is 21 km/h.
Driving direction	Parameter ID: 0x09	The unit is degree. When the parameter value is **0**, the direction is due north. The parameter value ranges from **0** to **359**. Little-endian. Data type: WORD	0x66 0x00 The driving direction is 102 degrees.
Horizontal dilution of precision (HDOP)	Parameter ID: 0x0A	Value range: 5–999 Unit: 1/10; little-endian Data type: WORD	0x13 0x00 The HDOP is 1.9.
Altitude	Parameter ID: 0x0B	Unit: meter; little-endian Data type: SINT16	0x2D 0x00 The altitude is 45 meters.
AD1	Parameter ID: 0x16	Analog <AD1>; little-endian Voltage formula of analog: AD1/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0 The AD1 voltage is 3.95 V.
AD2	Parameter ID: 0x17	Analog <AD2>; little-endian Voltage formula of analog: AD2/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0

			The AD2 voltage is 3.95 V.
AD3	Parameter ID: 0x18	Analog <AD3>; little-endian Voltage formula of analog: AD3/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0 The AD3 voltage is 3.95 V.
Built-in battery	Parameter ID: 0x19	Battery analog <AD4>; little-endian Voltage formula of analog: AD4/100 When the battery power is full, the voltage is 4.2 V. When the battery power is empty, the voltage is 3.4 V. Formula of remaining battery power (%): (AD4/100-3.4)/0.8 x 100% Data type: WORD	0xA0 0x01 Convert the digits to decimal digits: 416 416/100 = 4.16 The voltage is 4.16V. The remaining battery power is 99%.
external power	Parameter ID: 0x1A	External power analog <AD5>; little-endian Voltage formula of analog: AD5/100 Note: When the external power supply is disconnected, the voltage of AD5 is about 2 V instead of 0. Data type: WORD	0x51 0x05 Convert the digits to decimal digits: 1366 1366/100 = 13.66 The voltage of the external power supply is 13.66 V.
Fuel level (%)	Parameter ID: 0x29	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10000 is the actual value. Data type: WORD	0x7A 0x0D Convert the digits to decimal digits: 3450 The fuel level is 34.50%.
New event code	Parameter ID: 0x40	For the specific definition, please refer to the event code table. Data type: BYTE	
AD6	Parameter ID: 0x41	Analog <AD6>; little-endian Voltage formula of analog: AD6/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0 The AD6 voltage is 0.
Camera connection information(cache)	Parameter ID: 0x6E	Cache information of the connected camera. Little-endian. BIT0 - BIT15: Corresponding to the connection information of cameras 1 to 16. 1 indicates that camera CH1 is connected; 0 indicates that camera CH1 is disconnected. Data type: WORD	0x03 0x00 Indicates that cameras CH1 and CH2 are connected.
Vehicle speed (based on the tachograph) (km/h)	Parameter ID: 0x91	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x15 0x00 Convert the digits to decimal digits: 21 The vehicle speed is 21

			km/h.
Vehicle speed (based on the wheel) (km/h)	Parameter ID: 0x92	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x15 0x00 Convert the digits to decimal digits: 21 The vehicle speed is 21 km/h.
Engine speed (rpm)	Parameter ID: 0x99	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x12 0x04 Convert the digits to decimal digits: 1042 The engine rotational speed is 1042 rpm.
Engine coolant temperature (deg C)	Parameter ID: 0x9C	2-byte hexadecimal data Upload data after reading it. Data type: SINT16	0x32 0x00 Convert the digits to decimal digits: 50 The engine coolant temperature is 50°C.
Ambient air temperature (deg C)	Parameter ID: 0x9F	2-byte hexadecimal data Upload data after reading it. Data type: SINT16	0x28 0x00 Convert the digits to decimal digits: 40 The ambient air temperature is 40°C.
Geo-fence number	Parameter ID: 0XFE90	little-endian This data is valid only if GPRS event code 20,21. Data type :STRUCT	0x01 0x00 The enclosure number that triggers the entry/exit Geo-fence alarm is 1.
Engine coolant level	Parameter ID: 0XF826	little-endian Data will only be uploaded if read; Data type: WORD	B0 09 2480=24.80%
Over-speed Recovery Event Assistance Message	Parameter ID: 0XFE30	little-endian Data type: WORD This data is only available when obtaining through GPRS event code 138.	60 00 Indicates that the maximum speed during the speeding period is 96 km/h.
A84 input port status	Parameter ID: 0XFE25	Data type: WORD Status values of 12 input ports Bits 0 to 11 correspond to the states of A84 input ports 1 to 12.	00 01 0000 0000 0000 0001 Means input1 is active
Number of 4-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 4 bytes. The following data is not fixed, customers can	0x0F There are 15 parameter ID numbers whose length is 4 bytes. 0x00: The current data packet does not contain

		choose to upload the required data.	any parameter ID number whose length is 2 bytes.
Latitude	Parameter ID: 0x02	Unit: millionth of a degree; little-endian Data type: SINT32	0xE6 0x87 0x57 0x01 Convert the digits to decimal digits: 22513638 The latitude is 22.513638 degrees.
Longitude	Parameter ID: 0x03	Unit: millionth of a degree; little-endian Data type: SINT32	0XE6 0x5F 0xCC 0x06 Convert the digits to decimal digits: 114057190 The longitude is 114.057190 degrees.
Date and time	Parameter ID: 0x04	Contains 4 bytes; little-endian Unit: second Starting time: 1 January, 2000, 00:00:00 am. Data type: DWORD	0xEA 0x8D 0xA7 0x22 Convert the digits to decimal digits: 581406186
Mileage	Parameter ID: 0x0C	Indicates the total mileage. Unit: meter; little-endian Data type: DWORD	0x56 0x05 0x00 0x00 Convert the digits to decimal digits: 1366 The total mileage is 1366 meters.
Run time	Parameter ID: 0x0D	Indicates the total time. Unit: second; little-endian Data type: DWORD	0x96 0x1B 0x00 0x00 Convert the digits to decimal digits: 7062 The run time is 7062 seconds.
System flag	Parameter ID: 0x1C	The data is available only when the GPRS event code is 35. Bit 0: Whether to modify the EEP2 parameter. When the parameter value is **1**, the EEP2 parameter is modified. Bits 1–31: reserved. Data type: DWORD	0x00 0x00 0x00 0x00 Converted to binary digits: 0000 0000 0000 0000 0000 0000 0000 0000
RFID ID	Parameter ID: 0x25	Indicates the ID number of a RFID card. After RFID swipe and ACC ON, the RFID ID number will be uploaded. After ACC OFF, the RFID ID number will not be uploaded. Data type: DWORD	0xD7 0x9D 0xD1 0x00 The RFID ID number is 13737431.
Input port status	Parameter ID: 0x42	Bits 0–31 correspond to status of input ports 1–31. When the parameter value is **0**, the port is inactive. When the parameter value is 1, the port is	0x00 0x00 0x00 0x04 Converted to binary digits: 0000 0000 0000 0000 0000 0000 0000 0100 The input port 3 is active,

		active. Data type: DWORD	while other input ports are inactive.
Total fuel consumption (L)	Parameter ID: 0x98	4-byte hexadecimal data Data type: DWORD	0x01 0x02 0x00 0x00 Convert the digits to decimal digits: 513 The total fuel consumption is 513 L.
Total engine run time (h)	Parameter ID: 0x9A	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value. Data type: DWORD	0x12 0x34 0x00 0x01 Convert the digits to decimal digits: 16790546 The total engine run time is 1679054.6 hours.
High resolution vehicle distance (m)	Parameter ID: 0x9B	4-byte hexadecimal data Data type: DWORD	0x11 0x22 0x00 0x00 Convert the digits to decimal digits: 8712 The total mileage is 8712 meters.
High resolution total fuel consumption (L)	Parameter ID: 0xA0	Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value. Data type: DWORD	0x12 0x00 0x01 0x00 Convert the digits to decimal digits: 65554 The total fuel consumption is 65.554 L.
Fuel consumption rate (L/H)	Parameter ID: 0xA2	Little-endian. After the digits are converted to decimal digits, the converted value divided by 100 is the actual value. Data type: DWORD	0x12 0x00 0x02 0x00 Convert the digits to decimal digits: 131090 The fuel consumption rate is 1310.90 L/H.
Axle weight (kg)	Parameter ID: 0xA3	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value. Data type: DWORD	0x12 0x34 0x00 0x00 Convert the digits to decimal digits: 13330 The axle weight is 1333.0 kg.
Service distance (km)	Parameter ID: 0xA4	4-byte hexadecimal data Data type: SINT32	0x22 0x30 0x00 0x00 Convert the digits to decimal digits: 12322 The service distance is 12322 km.
Instantaneous fuel consumption (km/L)	Parameter ID: 0xA5	Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value. Data type: DWORD	0x12 0x56 0x00 0x00 Convert the digits to decimal digits: 22034 The instantaneous fuel consumption is 22.034 km/L.

Vehicle battery voltage	Parameter ID: 0XF825	Little-endian format, after conversion to decimal, the real value is obtained by dividing by 100. Data will only be uploaded if available. Data type: DWORD	0x12 0x00 0x02 0x00 1310.90V
Driving under the influence of fatigue	Parameter ID: 0XFE2F	Seconds (unit) This data is only available when obtaining through GPRS event code 136.	10 0E 00 00:3600s
Single trip fuel usage	Parameter ID: 0XF827	Little-endian format, after conversion to decimal, the real value is obtained by dividing by 100. Data will only be uploaded if available. Data type: DWORD	0x01 0x02 0x00 0x00 Fuel consumption for this trip is 513L
Ignition Off or Parking Accumulated Time	Parameter ID: 0XFEA1	Unit in seconds, used to upload ignition off or parking time; uploaded during ACC ON event	
Number of unfixed-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 8 or 12 bytes, or is unfixed. The ordering of parameter ID numbers is not fixed. For details, see the parameter ID table. The following data is not fixed, customers can choose to upload the required data.	0x01 There is one unfixed-byte parameter ID number. 0x00: The current data packet does not contain any parameter ID number whose length is unfixed.
Current base station info	Parameter ID: 0x0E	<Data length><MCC><MNC><LAC><CELL_ID><RX_LEVEL> Data length: hexadecimal; indicates the length of the base station data. Unit: byte. MCC: 16-bit unsigned; little-endian; indicates the Mobile Country Code. MNC: 16-bit unsigned; little-endian; indicates the Mobile Network Code. LAC: 16-bit unsigned; little-endian; indicates the Location Area Code. CELL_ID: 32-bit unsigned; little-endian; indicates the cell ID. RX_LEVEL: 16-bit signed; little-endian; indicates the signal strength. Data type: STRUCT	0x0C 0xCC 0x01 0x01 0x00 0x45 0xA5 0x8B 0xD4 0xE9 0x01 0xBB 0xFF 0x0C: The data length is 12 bytes. 0xCC 0x01: The MCC is 460. 0x01 0x00: The MNC is 01. 0x45 0xA5: The LAC is 42309. 0x8B 0xD4 0xE9 0x01: The cell ID is 32101515. 0xBB 0xFF: The signal strength is -69 dbm.

Picture name	Parameter ID: 0x28	The data is available only when the GPRS event code is 39. Time unit: second. Start time: 1 January, 2000, 00:00:00 am. Data type: STRUCT Only T633 models are supported.	0xCB 0x0F 0x23 0x19 0x01 0x1E 0x0C 0x00 There are two pieces of DWORD data: 0x19230FCB 0x000C1E01. 0x19230FCB: Indicates the date and time, that is, 130513024323. 0x000C1E01: Indicates the last part of the file name ,that is, C1E01. The file name is **130513024323_C1E01.jpg**.
Temperature sensor 1	Parameter ID: 0x2A	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 2	Parameter ID: 0x2B	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 3	Parameter ID: 0x2C	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 4	Parameter ID: 0x2D	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 5	Parameter ID: 0x2E	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 6	Parameter ID: 0x2F	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature	Parameter	Little-endian	0x01 0x09 0x1A

sensor 7	ID: 0x30	Data type: STRUCT	01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 8	Parameter ID: 0x31	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Camera status	Parameter ID: 0x49	<ID_Len><Number><Status> ID_Len: The length of this ID, 1 byte Number: The total number of cameras supported by this device, 1 byte, maximum 64 Status: 8 bytes, little-endian format; bit0: 1 indicates that camera CH1 is connected; 0 indicates that camera CH1 is disconnected …… . bit64: 1 indicates that camera CH64 is connected; 0 indicates that camera CH64 is disconnected.	
Currently using network information	Parameter ID: 0x4B	The network information that the device is connecting to. <ID_Len><version><Type><DescriptorLen><Descriptor> ID Len: 1 byte Version: 1 byte, 0x01 by default Type： The type of network being connected， 1 byte. 0: No Network, 1: Mobile Network, 2: WIFI (Reserved), 3: LAN (Reserved) DescriptorLen: the length of the network descriptor, 1 byte, range: 0~32 Descriptor: Network descriptor, string Data type:STRUCT	
Total analog quantity	Parameter ID: 0x4C	<ID_Len><number><AD1_no><AD1_voltage>... <ADn_no><ADn_voltage> ID_Len: The length of this ID, 1 byte number: The current number of analog quantities being uploaded, up to 32 analog quantities, 1 byte ADn_no: Number, 1-32, indicating the external AD number; 33: indicating battery voltage; 34: indicating external power voltage; others are reserved ADn_voltage: Voltage, in mV, 2 bytes	
Bluetooth peripheral auxiliary information	Parameter ID: 0XFE70	<ID_Len><version><type><data> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte type: Alarm type, 1 byte 01 (Low power alarm for temperature and humidity sensor) 02 (High temperature alarm for temperature and humidity sensor) 03 (Low temperature alarm for temperature and humidity sensor)	

		04 (High humidity alarm for temperature and humidity sensor) 05 (Low humidity alarm for temperature and humidity sensor) 06 (Signal loss of temperature and humidity sensor) 07 (Signal recovery of temperature and humidity sensor) 08 (Low power of IBeacon) 09 (Loss alarm of IBeacon) 10 (IBeacon has been recovered) 11 (IBeacon lost - customized) 12 (IBeacon recovered - customized) Data: Different alarm types have different data formats, and the big-endian format. The parsing format for temperature and humidity data is (signed 8.8 fixed-point). Alarm types 01, 02, 03, 04, 05: 03 31 32 33 AB BC B1 00 11 22 0A 64 48 63 48 Data parsing: The blue sub-body represents the length of the device name, and the subsequent data segments are all of fixed length. The first data segment (device name) length 03: 31 32 33, the longest is 16 bytes. The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (device battery level) length 01: 0A The fourth data segment (device temperature) length 02: 64 48 The fifth data segment (device humidity) length 02: 63 48 Alarm types 06, 07, 09, 10: 03 31 32 33 AB BC B1 00 11 22 02 Data parsing: The blue sub-body represents the length of the device name, and the following data segments are all of fixed length. The first data segment (device name) length 03: 123 The second data segment (device MAC) length 06: AB BC B1 00 11 2 Alarm type 08: 03 31 32 33 AB BC B1 00 11 22 02 0A F8 Data parsing: The blue sub-body represents the length of the device name, and the following data segments are all of fixed length. The first data segment (device name) length 03: 123 The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (device battery level) length 01: 0A The fourth data segment (signal strength) length 01: F8 (with signed type) Alarm types 11, 12: Data parsing: 01 00 02 00 Major: 01 00, indicating that Major is 0x0001. Minor: 02 00, indicating that Minor is 0x0002.
The speed data before rapid acceleration and deceleration	Parameter ID: 0xF824	<len><Number><unitinterval><datal>...<datN> len: ID length, 1 byte. Number: 1 byte, indicating the number of speed values. unit interval: 2 bytes, indicating the interval between speed values, in milliseconds (ms). data: 1 byte, speed value, unit: KM/H.

alerts.		Only appears in events 129 and 130 Data type: STRUCT
iBeacon Group A	Parameter ID: 0xFE71	<ID_Len><version><data1><data2><data3><data4><data5><data6><data7><data8> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte <data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A F4 Data parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. The first data segment (device name) length 03: 31 32 33, the longest is 16 bytes. The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (battery level) length 01: 0A The fourth data segment (signal strength) length 01: F4, with a signed type
iBeacon Group B	Parameter ID: 0xFE72	<ID_Len><version><data1><data2><data3><data4><data5><data6><data7><data8> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte <data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A F4 Data parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. The first data segment (device name) length 03: 31 32 33, the longest is 16 bytes. The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (battery level) length 01: 0A The fourth data segment (signal strength) length 01: F4, with a signed type
Bluetooth temperature and humidity sensor	Parameter ID: 0xFE73	Bluetooth temperature and humidity sensor <ID_Len><version><data1><data2><data3><data4> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte <data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A 64 48 63 48 64 48 63 48 64 48 63 48 Data Parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. First data segment (device name) Length 03: 31 32 33, maximum length 16 bytes Second data segment (device MAC) Length 06: AB BC B1 00 11 22 Third data segment (battery level) Length 01: 0A Fourth data segment (device temperature) Length 02: 64 48 Fifth data segment (device humidity) Length 02: 63 48 Sixth data segment (device temperature alarm threshold) Length 04: 64 48 63 48 (the first two bytes indicate high temperature alarm, the last two bytes indicate low temperature alarm) Seventh data segment (device humidity alarm threshold) Length 04: 64 48 63

		48 (the first two bytes indicate high humidity alarm, the last two bytes indicate low humidity alarm)
Alarm video information_A	Parameter ID: 0XFE79	<ID_Len><version><CH_number_1><CH_Type_1><video_name_len_1><video_name_1><CH_number_2><CH_Type_2><video_name_len_2><video_name_2> ID_Len: The length of this ID, 1 byte version: Protocol version number, 1 byte. Currently using 0x01 CH_number_X: Channel number 1: CH1 2: CH2 3: CH3 4: CH4 CH_Type_X: Channel type 1: ADS 2: DMS 3: Ordinary camera video_name_len_X: Length of the uploaded name, up to 125 bytes video_name_x: Uploaded video name
Alarm video information_A	Parameter ID: 0XFE80	<ID_Len><version><CH_number_1><CH_Type_1><video_name_len_1><video_name_1><CH_number_2><CH_Type_2><video_name_len_2><video_name_2> ID_Len: The length of this ID, 1 byte version: Protocol version number, 1 byte. Currently using 0x01 CH_number_X: Channel number 1: CH1 2: CH2 3: CH3 4: CH4 CH_Type_X: Channel type 1: ADS 2: DMS 3: Ordinary camera video_name_len_X: Length of the uploaded name, up to 125 bytes video_name_x: Uploaded video name
Facial Recognition Alarm Assistance Information	Parameter ID: 0XFE6A	<ID_Len><AlarmProtocol><AlarmType><PhotoName> ID_Len: The length of this ID, 1 byte AlarmProtocol: Protocol version number:, 1 byte When AlarmProtocol is 0X01: AlarmType: Alarm type, 1 byte; defined as follows: 1: Driver login successful 2: Driver login failed 3: Face recognition successful 4: Change of driver 5: Delete of driver PhotoName: Photo name, in string format, fixed 64 bytes. If there is no photo, all are 0x00 This data is only available when obtaining the GPRS event code 647.

TPMS data 1	Parameter ID: 0XFEF2	<ID_Len>< Number of tire pressure >< TPMS 1>< TPMS 2>...< TPMS n>
		ID_Len:1 byte
		Number of tire pressure：1 byte；Up to 16 tire pressure data are supported, more than 16 tire pressure data will be extended using CCE ID: FEE3.
		TPMS 1：
		typedef struct
		byte Num;// Tyre position
		byte ID[3];//ID，Little-endian
		word tpms_value;// Tire pressure
		byte temp; // Temperature
		byte status;// status

		Tyre position： Bits 7–5: indicate the vehicle's head part or trailer. 000(B): vehicle's head part; 001(B): trailer 1; 010(B): trailer 2; 011(B): trailer 3; 100(B): trailer 4. Bits 4–0: indicate the tire number. For example, 00001(B), indicating the first tire. ID: indicates a tire pressure sensor's ID number; 4 bytes; unsigned; hexadecimal. Tire pressure: 2 bytes; unsigned; hexadecimal; formula: obtained value x 0.025; unit: bar. Temperature: indicates the tire temperature; 1 byte; unsigned; hexadecimal; formula: obtained value - 50; unit: °C. Status: indicates the tire status; 1 byte; unsigned; hexadecimal. Bit 7: indicates the battery voltage status of the transmitter. 0: normal voltage; 1: low voltage. Bit 6: Whether to receive data from the transmitter. When you do not receive data from the transmitter within 15 minutes, the parameter value will be reset to 1. Bit 5: reserved. Bit 4: When the parameter value is 1, the air pressure is high. Bit 3: When the parameter value is 1, the air pressure is low. Bit 2: indicates temperature status. 1: high temperature; 0: normal temperature. Bits 1–0: indicate the alert status. 00: no alert; 01: fast air leak alert; 10: slow air leak alert; 11: tire inflation alert.

TPMS data 2	Parameter ID:
	0XFEF3
		<ID_Len>< Number of tire pressure >< TPMS 1>< TPMS 2>...< TPMS n>
		ID_Len:1 byte
		Number of tire pressure： 1 byte；
		TPMS 1：
		typedef struct
		byte Num;// Tyre position
		byte ID[3];//ID， Little-endian
		word tpms_value;// Tire pressure
		byte temp; // Temperature
		byte status;// status
		
		Tyre position：
		Bits 7–5: indicate the vehicle's head part or trailer. 000(B): vehicle's head part; 001(B): trailer 1; 010(B): trailer 2; 011(B): trailer 3; 100(B): trailer 4.
		Bits 4–0: indicate the tire number. For example, 00001(B), indicating the first tire.
		ID: indicates a tire pressure sensor's ID number; 4 bytes; unsigned; hexadecimal.
		Tire pressure: 2 bytes; unsigned; hexadecimal; formula: obtained value x

		0.025; unit: bar. Temperature: indicates the tire temperature; 1 byte; unsigned; hexadecimal; formula: obtained value - 50; unit: °C. Status: indicates the tire status; 1 byte; unsigned; hexadecimal. Bit 7: indicates the battery voltage status of the transmitter. 0: normal voltage; 1: low voltage. Bit 6: Whether to receive data from the transmitter. When you do not receive data from the transmitter within 15 minutes, the parameter value will be reset to 1. Bit 5: reserved. Bit 4: When the parameter value is 1, the air pressure is high. Bit 3: When the parameter value is 1, the air pressure is low. Bit 2: indicates temperature status. 1: high temperature; 0: normal temperature. Bits 1–0: indicate the alert status. 00: no alert; 01: fast air leak alert; 10: slow air leak alert; 11: tire inflation alert. Note: This ID data is used only when the number of tire pressure is set to more than 16, for the extension data of FEE2.

ASPC People Counter	0XFE96	<ID_Len>< version number >< sensor 1>< sensor 2>< sensor 3>< sensor 4>< All sensor data >
		Version: 1 byte: detects the sensor version.Sensors 1 to 4 are dynamic data. If the device is connected to only one sensor, data is uploaded from only one sensor.
		
		Sensor: little-endian. The data structure is as follows
		typedef struct
		byte number（1 byte）; // Sensor label.
		byte door_number（1 byte）; // door label。0:NULL 1:door1 2:door2 3:door3 4:door4
		byte state（1 byte）; // Sensor status: 0：invalid 1：IO detection door opening 2：IO detection shutdown
		// 3：IN Geo-fence 4: OUT Geo-fence
		dword up_car（4 byte）; // The number of people getting on this time
		dword down_car（4 byte）; // The number of people getting off this time
		dword all_up_car（4 byte）; // The total number of people getting on
		dword all_down_car（4 byte）; // The total number of people getting off
		
		All sensors data: little-endian, and the data structure is as follows:
		typedef struct
		dword up_car（4 byte）; // The number of people getting on this time
		dword down_car（4 byte）; // The number of people getting off this time
		dword all_up_car（4 byte）; // The total number of people getting on
		dword all_down_car（4 byte）; // The total number of people getting off

		dword surplus（4 byte）; // Number of people left in the car
Magnetic Card Reader Information	Parameter ID: 0X39	<RfidLen><RfidData> RfidLen: Card information length, 1 byte RfidData: Specific card data, up to 160 bytes

Storage Status	Parameter ID: 0X72	<LEN><Version Number><TF_type><Alarm_type>
		Byte TF_type; // Card type... 0: Device (for types 1, 2, and 7) 1: M2 2: SD1 3: SD2 4: Backup 5: Hard Disk;
		Byte Alarm_type; // Alarm Type: 1. Storage Full
		2. No detected storage devices
		3. Read/Write Error
		4. User Data Partition Abnormal
		5. Device Removed
		6. Bad Blocks
		7. Insufficient Storage Space in User Data Area
		This data is only available when obtaining through GPRS event code 122.

Overspeed Event Information Statistics	Parameter ID: 0xF82E	<ID_Len><Overspeed Start Time><Overspeed Start Longitude><Overspeed Start Latitude> <Overspeed End Time><Overspeed End Longitude><Overspeed End Latitude><Overspeed Duration><Average Speed During Overspeed><Maximum Speed During Overspeed> ID_Len: 1 byte, structure length, fixed value 20 Overspeed Start Time: 4 bytes, counted from 00:00:00 January 1, 2000 (unit: seconds) Overspeed Start Longitude: 4 bytes, unit: millionths of a degree Overspeed Start Latitude: 4 bytes, unit: millionths of a degree Overspeed End Time: occupies 4 bytes, counted from January 1, 2000, 00:00:00 (unit: seconds) Overspeed End Longitude: occupies 4 bytes, unit: one-millionth of a degree Overspeed End Latitude: occupies 4 bytes, unit: one-millionth of a degree Overspeed Duration: occupies 4 bytes (unit: seconds) Average Speed During Overspeed: occupies 2 bytes, unit: 0.1 km/h Maximum Speed During Overspeed: occupies 2 bytes, unit: 0.1 km/h Note: Little-endian format F8 2E //CCE_ID 20 //LEN 49 5D FE 2E // Overspeed Start Time 08 5D CC 06 // Overspeed Start Longitude 2E 79 57 01 // Overspeed Start Latitude 5A 5D FE 2E // Overspeed End Time A3 5C CC 06 // Overspeed End Longitude 3B 79 57 01 // Overspeed End Latitude 11 00 00 00 // Overspeed Duration (unit: seconds) 95 01 // Average Speed During Overspeed (unit: 0.1 km/h) 95 01 // Maximum Speed During Overspeed (unit: 0.1 km/h)

		Note: Uploaded together with event 138
Ignition Off Event Auxiliary Message	Parameter ID: 0x6A	Ignition Off Event Auxiliary Message <ID_Len>version><OneLen><OneTime><AvrSpeed><MaxSpeed><oil consumption> When version = 1: ID_Len: Length of this ID, 1 byte Version: Data Version 1, 1 byte OneLen: Distance of this trip, 4 bytes, little-endian format, unit: meters OneTime: Duration of this trip, 4 bytes, little-endian format, unit: seconds AvrSpeed: Average speed, 2 bytes, little-endian format, unit: km/h MaxSpeed: Maximum speed during this trip, 2 bytes, little-endian format, unit: km/h Oil consumption: Fuel consumption during this trip, 2 bytes, unit: 0.01% This data is only available when obtained through GPRS event code 145

Temperature sensor information	Parameter ID: 0xDB	<ID len><version><temp_num><temp1_data>...<tempN_data>
		ID length: 1 byte, length of this ID data
		Version: 1 byte, version number
		If the version number is 1, then:
		temp_num：1B，Number of temperature sensors
		tempN_data：Thermal data, structured as follows:
		struct
		{
		byte No; // Temperature sensor number. 0 indicates unregistered.
		byte SN[8]; // Thermal sensor SN, such as 28 D8 82 23 04 00 00 5
		short int value; // Temperature value, 09 1A: 2 signed bytes, little-endian format, representing a temperature of 66.65
		};
		
		For uploading the temperature sensor's serial number, SN, temperature value, etc., default option is not selected.

A84 event auxiliary message	Parameter ID: 0XFE2A	<len><data> <len>Current data length <data> 05 05 00 00 00 05: Represents the small event number of A84, with a value range of 1 to 12, 17 to 28, 1 to 12 represent activation events of A84, such as 1 indicating port 1 activation; 17 to 28 represent deactivation events of A84, such as 17 indicating port 1 deactivation, and so on. 05 00 00 00: Indicates an activation time of 5 seconds. （Data type： STRUCT）
The current data packet ends here.		

*	Contains 1 byte. It is used to separate the command content from the checksum. ASCII (hexadecimal: 0x2A)	*
Checksum	Contains 2 bytes. Indicates the sum of hexadecimal characters from the packet header "$$" to the asterisk "*" (including the packet header and asterisk). *$$<Data identifier><Data length>,<IMEI>,<Command type>,<Hexadecimal data packet><*Checksum>*\r\n	If the sum result is **0x27 0x62**, send the checksum 62 (low byte 0x62) in ASCII format, that is, **0x36 0x32**. Hexadecimal: 0x36 0x32 ASCII: 62
\r\n	Contains 2 bytes. This is an ending character. The type is ASCII (hexadecimal: 0x0D,0x0A).	\r\n

Notes:

data type	Description	transmission rule
BYTE	Unsigned single-byte integer (bytes, 8 bits)	Transmit in byte stream
WORD	Unsigned double-byte integer (word, 16 bits)	Little-endian
DWORD	Unsigned four-byte integer (double word, 32-bit)	Little-endian
BYTE[n]	N byte	Transmit in byte stream
BCD[n]	8421 code, n bytes	Transmit in byte stream
STRING	GBK code, if no data, blank	Little-endian
SINT8	Signed single byte	Transmit in byte stream
SINT16	Signed double byte	Little-endian
SINT32	Signed four bytes	Little-endian
STRUCT	Based on the data analysis description decision	Transfer as defined by the structure

1.2 Event Code

Event code	Event description	Default SMS header (up to 16 bytes)
1	SOS Pressed	SOS
2	Input 2 Active	In2 Active
3	Input 3 Active	In3 Active
4	Input 4 Active	In4 Active
5	Input 5 Active	In5 Active
6	Input 6 Active	In6 Active
7	Input 7 Active	In7 Active
8	Input 8 Active	In8 Active

118	Input 9 Active	In9 Active
9	Input 1 Inactive	In1 Inactive
10	Input 2 Inactive	In2 Inactive
11	Input 3 Inactive	In3 Inactive
12	Input 4 Inactive	In4 Inactive
13	Input 5 Inactive	In5 Inactive
14	Input 6 Inactive	In6 Inactive
15	Input 7 Inactive	In7 Inactive
16	Input 8 Inactive	In8 Inactive
119	Input 9 Inactvie	In9 Inactive
17	Low Battery	Low Battery
18	Low External Battery	Low Ext-Battery
19	Speeding	Speeding
20	Enter Geo-fence	Enter Fence N
21	Exit Geo-fence	Exit Fence N
22	External Battery On	Ext-Battery On
23	External Battery Cut	Ext-Battery Cut
24	GPS Signal Lost	GPS Signal Lost
25	GPS Signal Recovery	GPS Recovery
26	Enter Sleep	Enter Sleep
27	Exit Sleep	Exit Sleep
28	GPS Antenna Cut	GPS Antenna Cut
29	Device Reboot	Power On
31	Heartbeat	/
32	Cornering	Cornering
33	Track By Distance	Distance
34	Reply Current (Passive)	Now
35	Track By Time Interval	Interval
36	Tow	Tow
37	RFID	/
39	Photo	/
41	Stop Moving	Quiet
42	Start Moving	Moving
44	GSM Jamming	GSM Jamming
50	Temperature High	Temp High
51	Temperature Low	Temp Low
52	Full Fuel	Full Fuel
53	Low Fuel	Low Fuel
54	Fuel Theft	Fuel Theft
63	No GSM Jamming	No GSM Jamming
70	Reject Incoming Call	Reject Incoming Call
71	Get Location by Call	Get Location by Call

72	Auto Answer Incoming Call	Auto Answer Incoming Call
78	IMPACT	IMPACT
82	Fuel Filling	Fuel Filling
83	Ult-Sensor Drop	Ult-Sensor Drop
87	Tpms Alarm	Tpms Alarm
90	Sharp Turn to Left	Harsh Cornering
91	Sharp Turn to Right	Harsh Cornering
94	Output 1 Active	Out1 Active
95	Output 2 Active	Out2 Active
96	Output 3 Active	Out3 Active
97	Output 4 Active	Out4 Active
98	Output 5 Active	Out5 Active
159	Output 6 Active	Out6 Active
160	Output 7 Active	Out7 Active
161	Output 8 Active	Out8 Active
99	Output 1 Inactive	Out1 Inactive
100	Output 2 Inactive	Out2 Inactive
101	Output 3 Inactive	Out3 Inactive
102	Output 4 Inactive	Out4 Inactive
103	Output 5 Inactive	Out5 Inactive
162	Output 6 Inactive	Out6 Inactive
163	Output 7 Inactive	Out7 Inactive
164	Output 8 Inactive	Out8 Inactive
117	People Counter	People Counter
122	SD state	SD state
128	Rollover	Rollover
129	Harsh braking	Harsh Braking
130	Harsh acceleration	Fast Accelerate
133	Idle Overtime	Idle Overtime
134	Idle Recovery(Recovery from Idle Overtime)	Idle Recovery
135	Fatigue Driving	Fatigue Driving
136	Enough Rest after Fatigue Driving	Enough Rest
138	Speed Recovery	Speed Recovery
139	Maintenance Notice	Maintenance
144	Ignition On	Ignition On
145	Ignition Off	Ignition Off
165	Bluetooth peripheral alarm	Bluetooth peripheral alarm
576	CH1 Video Loss	CH1 Video Loss
577	CH2 Video Loss	CH2 Video Loss
578	CH3 Video Loss	CH3 Video Loss

579	CH4 Video Loss	CH4 Video Loss
580	CH5 Video Loss	CH5 Video Loss
581	CH6 Video Loss	CH6 Video Loss
608	Storage Failure	Storage Failure
609	Storage Full	Storage Full
610	CH1 Video Recovery	CH1 Recovery
611	CH2 Video Recovery	CH2 Recovery
612	CH3 Video Recovery	CH3 Recovery
613	CH4 Video Recovery	CH4 Recovery
614	CH5 Video Recovery	CH5 Recovery
615	CH6 Video Recovery	CH6 Recovery
647	Face ID Alarm	Face ID Alarm
648	Forward-CW	Forward-CW
650	Pedestrian-CW	Pedestrian-CW
651	Left-LDW	Left-LDW
652	Right-LDW	Right-LDW
653	HMW(Distance	HMW(Distance
655	DMS_CALL	DMS_CALL
656	DMS_SMOKINGDMS	DMS_SMOKINGDMS
657	DMS_TIRED	DMS_TIRED
658	DMS_YAWN	DMS_YAWN
659	DMS_DISTRACT	DMS_DISTRACT
660	DMS_LOSE	DMS_LOSE
661	DMS_COVEREDDMS_TIRED	DMS_COVEREDDMS_TIRED
662	DMS_IR_BLOCK	DMS_IR_BLOCK
663	DMS_SEATBELT	DMS_SEATBELT
669	BSD_FRONT	BSD_FRONT
670	BSD_BEHIND	BSD_BEHIND
671	BSD_LEFT	BSD_LEFT
672	BSD_RIGHT	BSD_RIGHT

2 Parsing Instances

2.1 Source Data

hexadecimal 0x prefix is temporarily omitted here . The following numbers are all bytes, such as 24 represents 0x 24.

The source data is as follows

24 24 50 31 30 36 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 43 43 45 2C 19 00 00 00 0C 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 07 00 0B 26 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 D7 87 57 01 03 48 60 CC 06 04 DE BF B5 24 0C 80 68 00 00 0D E4 A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 09 00 0B 27 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 D0 87 57 01 03 41 60 CC 06 04 E8 BF B5 24 0C 80 68

00 00 0D EE A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 0B 00 0B 27 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 CF 87 57 01 03 3E 60 CC 06 04 F2 BF B5 24 0C 80 68 00 00 0D F8 A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 27 00 16 00 00 17 00 00 19 A3 01 1A 26 05 40 23 00 06 02 D4 87 57 01 03 43 60 CC 06 04 FC BF B5 24 0C 80 68 00 00 0D 02 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 07 00 0B 25 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 DA 87 57 01 03 3E 60 CC 06 04 06 C0 B5 24 0C 80 68 00 00 0D 0B A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 24 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 DF 87 57 01 03 2F 60 CC 06 04 10 C0 B5 24 0C 80 68 00 00 0D 15 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 22 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E9 87 57 01 03 14 60 CC 06 04 1A C0 B5 24 0C 80 68 00 00 0D 1F A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 21 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 EE 87 57 01 03 0E 60 CC 06 04 24 C0 B5 24 0C 80 68 00 00 0D 29 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 21 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E9 87 57 01 03 16 60 CC 06 04 2E C0 B5 24 0C 80 68 00 00 0D 33 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 09 00 0B 23 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E6 87 57 01 03 FF 5F CC 06 04 39 C0 B5 24 0C 80 68 00 00 0D 3D A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 09 00 0B 23 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E8 87 57 01 03 E7 5F CC 06 04 43 C0 B5 24 0C 80 68 00 00 0D 46 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 17 01 0A 08 00 0B 23 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E8 87 57 01 03 D7 5F CC 06 04 4D C0 B5 24 0C 80 68 00 00 0D 50 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00 2A 32 30 0D 0A

2.2 Divide the message structure

the message structure can be divided into:

Start sign	Message Header	Delimiter 0x 2C	Message body	Delimiter 0x2A	Verification Code	End Sign

1. yellow in the source data , as follows:

   24 24 is converted to ASCII code $ $, which is the start mark of the message

2. The message header is marked in green in the source data , as follows:

   50 31 30 36 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 43 43 45

Start Byte	Fields	type of data	Byte Information	explain
0	Packet Identifier	Byte	50	Data packet identifier, the sender needs to be consistent with the receiver
1	Data length	String [n1]	31 30 36 34	ASCII is 1064 , which means the length from the first delimiter " , " (inclusive) to the end character " \r\n "

				(inclusive) is 1064 bytes, where n 1=4 .
1 +n 1	Delimiter	Byte	2C	Separator, half-width comma
2 + n 1	IMEI number	String [15]	38 36 31 35 38 35 30 34 30 34 39 34 34 36 38	ASCII is 861585040494468 , which indicates the IMEI number of the device. Generally, it is 1 to 5 bytes. There will be additional instructions for special cases.
17+n 1	Delimiter	Byte	2C	Separator, half-width comma
2 0+ n 1	Protocol Name	String[3]	43 43 45	ASCII is CCE, indicating the protocol name

3. The delimiter 0x 2 C is marked in light blue , as shown below
   2 C is converted to ASCII code as a half-width comma (,) , which is a separator

4. The end sign is marked in purple , as follows
   0D 0 A is converted to ASCII code for carriage return and line feed (\ r\n)
5. The verification code is marked in dark blue as follows
   3 2 30 is the check code, which is converted to ASCII as 2 3 , which is the cumulative sum of the entire GPRS data packet (the sum of all data from the beginning to the check code, excluding the check code and the end character)
6. The separator is marked in red , as follows
   2A The delimiter is converted to ASCII asterisk (*) and used to separate the message body and the check code.
7. The remaining unmarked parts of the source message data are all data bodies. For parsing, see 5.3 , Message Body Parsing Example

2.3 Message body parsing example

1. Message body structure

Cache remaining quantity	Number of packets	Packet 1	Total length of data packet 1	Packet 2	Total length of data packet 2	....	Packet n	Total length of data packet n

Start Byte	Fields	type of data	Description and Requirements
0	Cache remaining quantity	D Word	
4	Number of packets	Word	There are several data packets in the whole data
6	Total length of data packet 1	Word	
8	Packet 1	Byte[N1]	N1 is the total length of data packet 1.
N1+8	Total length of data packet 2	Word	
N 1+10	Packet 2	Byte[N2]	N1 is the total length of data packet 2

...			
$\sum_{i=1}^{m-1} N_i + 2m + 4$	Total length of data packet m	Word	
$\sum_{i=1}^{m-1} N_i + 2m + 6$	Packet m	Byte [Nm]	Nm is the total length of data packet m

All byte data of the message body are excerpted as follows:

19 00 00 00 0C 00

04 0C 80 68 00 00 0D E4 A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 0 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 09 00 0B 27 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 D0 87 57 01 03 41 60 CC 06 04 E8 BF B5 24 0C 80 68 00 00 0D EE A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 0B 00 0B 27 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 CF 87 57 01 03 3E 60 CC 06 04 F2 BF B5 24 0C 80 68 00 00 0D F8 A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 27 00 16 00 00 17 00 00 19 A3 01 1A 26 05 40 23 00 06 02 D4 87 57 01 03 43 60 CC 06 04 FC BF B5 24 0C 80 68 00 00 0D 02 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 07 00 0B 25 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 DA 87 57 01 03 3E 60 CC 06 04 06 C0 B5 24 0C 80 68 00 00 0D 0B A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 24 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 DF 87 57 01 03 2F 60 CC 06 04 10 C0 B5 24 0C 80 68 00 00 0D 15 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 22 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E9 87 57 01 03 14 60 CC 06 04 1A C0 B5 24 0C 80 68 00 00 0D 1F A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 21 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 EE 87 57 01 03 0E 60 CC 06 04 24 C0 B5 24 0C 80 68 00 00 0D 29 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 08 00 0B 21 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E9 87 57 01 03 16 60 CC 06 04 2E C0 B5 24 0C 80 68 00 00 0D 33 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

54 00 15 00 05 05 01 06 09 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 09 00 0B 23 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E6 87 57 01 03 FF 5F CC 06 04 39 C0 B5 24 0C 80 68 00 00 0D 3D A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

04 43 C0 B5 24 0C 80 68 00 00 0D 46 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 0 00 00

24 0C 80 68 00 00 0D 50 A1 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 54 00 15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 17 01 0A 08 00 0B 23 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 E8 87 57 01 03 D7 5F CC 06 04 4D C0 B5 24 0C 80 68 00 00 0D 00 00

2. the first four bytes of the message body are the remaining number of cache bytes, which are marked in dark green as follows:

1 9 00 00 00 is the little-endian byte order (see 1.3 , the same below), so we know that the remaining number of cache entries is 0x 00000019 , which is 2 5 in decimal , that is, the remaining number of cache entries is 2 5 .

3. The two bytes after the remaining amount in the cache are the number of packets, which are marked in purple , as shown below

0C 00 is the little endian byte order, so the number of packets is 0x 000 C, which is 12 in decimal, meaning there are 12 packets in total.

4. The following bytes are arranged in the form of (packet length, data packet), so 1 or 2 packets can be divided in sequence. The length of each packet is marked in gray , and line breaks are used to separate them to make the structure clear. It can be clearly seen that the data body is exactly 1 or 2 packets, which is consistent with our analysis results. Since the analysis method between packets is consistent, only the first packet is analyzed here.

2.4 Packet parsing example

1. First, the first data packet is extracted as follows.

15 00 05 05 01 06 0A 07 00 14 00 15 02 09 08 00 00 09 1F 01 0A 07 00 0B 26 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00 06 02 D7 87 57 01 03 48 60 CC 06 04 DE BF B5 24 0C 80 68 00 00 0D E4 A0 03 00 1C 01 00 00 00 01 49 09 04 01 00 00 00 00 00 00 00

Start Byte	Fields	type of data	Description and Requirements
0	Total number of packet IDs	Word	
2	Total number of single-byte IDs	Byte	The value of this field is N 1
3	Single byte ID information	Byte[2*N1]	For details of ID, please see 3.Detailed information of ID, the same below
3 +2* N 1	Total number of double-byte IDs	Byte	The value of this field is N 2
5 +2* N 1	Double-byte ID information	Byte [3*N2+p]	30 parameter is not included , p = 0 , otherwise p = 1
5 +2*N1+3*N2+p	Total number of	Byte	The value of this field is N 3

	four-byte IDs		
7+2*N1+3*N2+p	Four-byte ID information	Byte [5*N3+ q]	FE2F parameter is not included , q = 0 , When the 0x FE2F parameter is included, q = 1
7+2*N1+3*N2+ 5* N 3+p+ q	Total number of non-fixed byte IDs	Byte	
9+2*N1+3*N2+ 5* N 3+p+ q	Non-fixed byte ID information	Byte [M]	Since it is a non-fixed byte, it is impossible to determine M

2. the following analysis:

a) Total number of packet IDs

in yellow , 1 5 00 is the little-endian byte order, and the hexadecimal 0x 0015 is converted to decimal 2 1 , indicating that the total number of IDs of the data packet is 2 1

b) Total number of single-byte IDs

in green , 0x 05 is converted to decimal 5, indicating that the total number of single-byte IDs is 5. Refer to Table 5-4-1 , that is, N 1=5

c) Single byte ID information

Marked in light blue , the data is as follows

05 01 06 0A 07 00 14 00 15 02

Single Byte ID detailed analysis as shown:

ID	content	Description and explanation
0 5	0 1	Positioning status is valid
0 6	0A	The number of satellites is 1 0
0 7	0 0	GSM signal strength is 0
1 4	0 0	The output port status is 0x 00
1 5	0 2	The input port status is 0x 00

d) Total number of double-byte IDs

Use dark blue to mark, 0x 09 is converted to 1, which is 9 in base 0, indicating that the total number of double-byte IDs is 5, that is, N 2=9

e) Double-byte ID information

Since it does not contain parameter 0xFE 30 , p = 0. Purple is used to mark the data. The excerpt is as follows:

08 00 00 09 1F 01 0A 07 00 0B 26 00 16 00 00 17 00 00 19 A2 01 1A 26 05 40 23 00

detailed analysis of double-byte IDs as shown

ID	content	Description and explanation
08	00 00	The current speed value is 0km/ h
09	1F 01	1F 01 is the little endian byte order, 0x 011F is converted to decimal as 287 , so it means the driving direction is 287 °, due north is 0°

0A	07 00	07 00 is the little-endian byte order, 0x 0007 is converted to decimal 7, indicating that the horizontal positioning accuracy value is 7
0B	26 00	26 00 is the little-endian byte order, 0x 0026 is converted to decimal as 38 , indicating an altitude of 38 m
16	00 00	00 00 is the little endian byte order, 0x 0000 is converted to decimal as 0, and through the calculation formula 0/ 100=0(V) , it means that the analog value of port AD1 is 0
17	00 00	00 00 is the little endian byte order, 0x 0000 is converted to decimal as 0, and through the calculation formula 0/ 100=0(V) , it means that the analog value of port AD2 is 0
19	A2 01	A2 01 is the little endian byte order, 0x 01 A2 is converted to decimal as 418 , and the calculation formula 418 / 100 = 4.18 (V) indicates that the analog value of port AD3 is 4.18V (battery power)
1A	26 05	26 05 is the little endian byte order, 0x 0526 is converted to decimal as 1 318. By calculating the formula 1318 / 100 = 13.18 (V) , it means that the analog value of port AD4 is 1 3.18V (external power supply)
40	23 00	23 00 is the little endian byte order, 0x 0023 is converted to decimal 35 , indicating that the event code is 35. Refer to Chapter 4 (Event Code Table) to know that event 35 is Track By Time Interval

f) Total number of four-byte IDs

in gray , 0x 06 is converted to 1, which is 6 in base 0 , indicating that the total number of four-byte IDs is 6. that is, N 3=6

g) Four-byte ID information

Since the parameter 0x FE2F is not included , q = 0. The data is marked in brown . The excerpt is as follows:

02 D7 87 57 01 03 48 60 CC 06 04 DE

BF B5 24 0C 80 68 00 00 0D E4 A0 03 00 1C 01 00 00 00

Detailed analysis of four-byte ID as shown

ID	content	Description and explanation
02	D7 87 57 01	D7 87 57 01 is the little endian byte order, 0x 015787 D7 is converted to decimal 22513623 , and the formula is 22513623 / 1000000=22.513623 , indicating that the latitude is 22.513623
03	48 60 CC 06	48 60 CC 06 is the little endian byte order, 0x 06 CC 6048 is converted to decimal 114057288 , and the formula is 114057288 / 1000000=114.057288 , indicating that the longitude is 114.057288
04	DE BF B5 24	DE BF B5 24 is the little endian byte order, 0x 24 B 5 BFDE is converted to decimal 615890910 , with 0:0:0 on January 1, 2000 as the starting point
0C	80 68 00 00	80 68 00 00 is the little endian byte order, 0x 00006880 is converted to decimal 26752 , indicating that the mileage is 26752 m
0D	E4 A0 03 00	E4 A0 03 00 is the little endian byte order, 0x 0003 A0E 4 is converted to

		decimal 237796 , which means the running time is 237796 seconds (2 days, 18 hours, 3 minutes, 16 seconds)
1C	01 00 00 00	Since the event code of this message is 3 5 , this data is valid. 01 00 00 00 is the little endian byte order. Bit 0 of 0x 00000001 is set (bit 0 = 1), indicating that the EEP 2 parameter has been modified.

h) Total number of IDs with non-fixed bytes

Use dark green to mark, 0x 01 is converted to 1 in decimal , indicating that the total number of non-fixed byte IDs is 1

i) ID information with non-fixed bytes

in red , the excerpt is as follows

49 09 04 01 00 00 00 00 00 00 00

Detailed analysis of non-fixed byte ID , and find that there is only one non-fixed byte ID , which is 0x49 (camera status information) as shown

Fields	content	Description and explanation
ID length	09	0x 09 , converted to decimal is 9, so the ID length is 9 bytes
The total number of cameras supported by this device	04	0x 04 , converted to decimal, is 4. The device supports up to 4 cameras.
Camera access status	01 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00 is little-endian byte order, and bit 0 of 0x 0000000000000001 is 1 , indicating that only the camera of channel 1 is connected

If you have any questions, do not hesitate to email us at info@meitrack.com.