MEITRACK MDVR GPRS Protocol

Applicable Model:

MD511H/MD522S/MD811H/MD822S/MD533S/MD500S

Change History

File Name	MEITRACK MDVR GPRS Protocol		
Project	MD511H/MD522S/MD811H/MD822S/MD533S/MD500S	Creation Date Update Date	2018-05-21 2024-03-20
Subproject	GPRS Protocol	Total Pages	73
Version	V2.0	Confidential	Internal Documentation

Contents

1 Command Format ..... - 6 -
1.1 GPRS Command Format ..... - 6 -
1.2 MDVR Command Format ..... - 6 -
1.3 Event Code ..... - 21 -
2 Command List ..... - 23 -
3 Command Details ..... - 25 -
3.1 Real-Time Location Query (GPRS) – A10 ..... - 25 -
3.2 Setting a Heartbeat Packet Reporting Interval (GPRS) – A11 ..... - 25 -
3.3 Tracking by Time Interval (GPRS) – A12 ..... - 26 -
3.4 Setting the Cornering Report (GPRS) – A13 ..... - 26 -
3.5 Tracking by Distance – A14 ..... - 26 -
3.6 Setting the Parking Scheduled Tracking Function (GPRS) – A15 ..... - 27 -
3.7 Enabling the Parking Scheduled Tracking Function (GPRS) – A16 ..... - 27 -
3.8 Controlling Output 1 Status by RFID/iButton – A17 ..... - 28 -
3.9 Setting GPRS Parameters – A21 ..... - 28 -
3.10 Setting the Standby GPRS Server – A23 ..... - 28 -
3.11 Setting the IP3 Parameters – A25 ..... - 29 -
3.12 Reading All Authorized Phone Numbers – A70 ..... - 29 -
3.13 Setting Authorized Phone Numbers – A71 ..... - 29 -
3.14 Setting Listen-in Phone Numbers – A72 ..... - 30 -
3.15 Setting the Smart Sleep Mode – A73 ..... - 30 -
3.16 Transmitting Audio and Video Data in Real Time – A9A ..... - 30 -
3.17 Controlling Real-Time Audio and Video Transmission – A9B ..... - 33 -
3.18 Querying the Resource List – A9C ..... - 33 -
3.19 Playing Back Videos Remotely – A9D ..... - 35 -
3.20 Controlling Remote Video Playback – A9E ..... - 36 -
3.21 Uploading Files – A9F ..... - 37 -
3.22 Controlling File Uploading – AA0 ..... - 39 -
3.23 Obtaining the WiFi List – AA1 ..... - 39 -
3.24 Sending the FTP File Uploading Progress – AA2 ..... - 40 -
3.25 Obtaining MDVR Network Status – AA3 ..... - 40 -
3.26 Querying which days' video files have been stored – AA4 ..... - 42 -
3.27 Transmitting Audio and Video Data in Real Time By Using the RTMP – AB2 ..... - 42 -
3.28 Controlling Real-Time Audio and Video Transmission By Using the RTMP – AB3 ..... - 43 -
3.29 Playing Back Videos Remotely By Using the RTMP (GPRS) – AB4 ..... - 44 -
3.30 Controlling Remote Video Playback By Using the RTMP – AB5 ..... - 45 -
3.31 Querying the Resource List From Data Packets – AB8 ..... - 45 -
3.32 Setting the WiFi Hotspot Function – ABB ..... - 47 -
3.33 Setting a Geo-Fence – B05 ..... - 47 -
3.34 Deleting a Geo-Fence – B06 ..... - 48 -
3.35 Setting the Speeding Alert – B07 ..... - 48 -
3.36 Setting the Towing Alert – B08 ..... - 48 -

3.37 Fast Setting the Towing Alert – B10 ..... - 49 -
3.38 Setting a Polygonal Geo-Fence – B11 ..... - 49 -
3.39 Setting the Mileage and Speed Calculation Mode – B22 ..... - 50 -
3.40 Setting Filtering Time of an Input Port – B26 ..... - 50 -
3.41 Turning off the LED Indicator – B31 ..... - 51 -
3.42 Setting a Log Interval – B34 ..... - 51 -
3.43 Setting the Local Time Zone – B35 ..... - 51 -
3.44 Setting the GPRS Time Zone – B36 ..... - 52 -
3.45 Setting FTP upload photo parameters – B64 ..... - 52 -
3.46 Setting SMS Event Characters – B91 ..... - 52 -
3.47 Setting Event Authorization – B99 ..... - 53 -
3.48 Setting the Speaker Volume Level of the MDVR – BB8 ..... - 53 -
3.49 Controlling Output Status – C01 ..... - 54 -
3.50 Notifying the Device of Sending an SMS – C02 ..... - 54 -
3.51 Setting a GPRS Event Transmission Mode – C03 ..... - 55 -
3.52 Registering a Temperature Sensor Number – C40 ..... - 55 -
3.53 Deleting a Registered Temperature Sensor – C41 ..... - 56 -
3.54 Reading the Temperature Sensor SN and Number – C42 ..... - 56 -
3.55 Setting the Temperature Threshold and Logical Name – C43 ..... - 56 -
3.56 Reading Temperature Sensor Parameters – C44 ..... - 57 -
3.57 Checking Temperature Sensor Parameters – C46 ..... - 58 -
3.58 Setting Fuel Parameters – C47 ..... - 58 -
3.59 Reading Fuel Parameters – C48 ..... - 59 -
3.60 Setting the Fuel Theft Alert – C49 ..... - 59 -
3.61 Transparently Transmitting Data over the Serial Port – C61 ..... - 59 -
3.62 Setting the Driver Fatigue Function – C90 ..... - 60 -
3.63 Setting Event Playing – CB8 ..... - 60 -
3.64 Deleting an Event in the Buffer – CFF ..... - 61 -
3.65 Authorizing a RFID Card/iButton Key – D10 ..... - 61 -
3.66 Authorizing RFID Cards/iButton Keys in Batches – D11 ..... - 61 -
3.67 Checking RFID/iButton Authorization – D12 ..... - 62 -
3.68 Reading an Authorized RFID Card/iButton Key – D13 ..... - 62 -
3.69 Deleting an Authorized RFID Card/iButton Key – D14 ..... - 62 -
3.70 Deleting Authorized RFID Cards/iButton Keys in Batches – D15 ..... - 63 -
3.71 Checking the Checksum of the Authorized RFID/iButton ID Database – D16 ..... - 63 -
3.72 Setting the Maintenance Mileage – D65 ..... - 63 -
3.73 Setting Maintenance Time – D66 ..... - 63 -
3.74 Setting Output Triggering – D72 ..... - 64 -
3.75 Allocating GPRS Cache and GPS Log Storage Space – D73 ..... - 64 -
3.76 Setting Harsh Acceleration and Harsh Braking Parameters – D79 ..... - 64 -
3.77 Obtaining All Alert Parameters of a Tire Pressure Sensor– DA0 ..... - 65 -
3.78 Obtaining Data of All Bound Tire Pressure Sensors– DA1 ..... - 66 -
3.79 Obtaining Data of a Tire Pressure Sensor– DA2 ..... - 67 -
3.80 Deleting Tire Pressure Sensors – DA3 ..... - 67 -

3.81 Obtaining Data of Multiple Tire Pressure Sensors – DA4 ..... - 68 -
3.82 Setting Alert Thresholds of a Tire Pressure Sensor– DA5 ..... - 68 -
3.83 Reading Device's Firmware Version and SN – E91 ..... - 69 -
3.84 Restarting the GSM and GPS Modules – F00 ..... - 69 -
3.85 Restarting the GSM Module – F01 ..... - 70 -
3.86 Restarting the GPS Module – F02 ..... - 70 -
3.87 Setting the Mileage and Run Time – F08 ..... - 70 -
3.88 Deleting SMS/GPRS Cache Data – F09 ..... - 70 -
3.89 Restoring Initial Settings – F11 ..... - 71 -
4 Appendix: Struct Data Analysis ..... - 71 -
4.1 Transmitting Audio and Video Data in Real Time – A9A ..... - 71 -
4.2 Controlling Real-Time Audio and Video Transmission – A9B ..... - 73 -
4.3 Querying the Resource List – A9C ..... - 74 -
4.4 Playing Back Videos Remotely – A9D ..... - 76 -
4.5 Controlling Remote Video Playback – A9E ..... - 77 -
4.6 Uploading Files – A9F ..... - 78 -
4.7 Controlling File Uploading – AA0 ..... - 81 -
4.8 Obtaining the WiFi List – AA1 ..... - 81 -
4.9 Sending the FTP File Uploading Progress – AA2 ..... - 82 -
4.10 Obtaining MDVR Network Status – AA3 ..... - 83 -
4.11 Querying which days' video files have been stored – AA4 ..... - 83 -

1 Command Format

1.1 GPRS Command Format

The GPRS command format is as follows:

GPRS command sent from the server to the MDVR	*@@<Data identifier><Data length>,<IMEI>,<Command code><Command content><*Checksum>*\r\n
GPRS command sent from the MDVR to the server	*$$<Data identifier><Data length>,<IMEI>,<Event code>,<Command content/Error code><*Checksum>*\r\n
Command description	

- **@@**: Indicates the packet header sent from the server to the MDVR. Contains 2 characters.
- **Data identifier**: Contains 1 byte. The character type is hexadecimal, and its value ranges from 0x41 to 0x7A. The data identifier in the reply command must be the same as that of the sending command. Otherwise, the command fails to be sent.
- A comma (,) is used to separate data characters. The character type is the American Standard Code for Information Interchange (ASCII) (hexadecimal: 0x2C).
- **Data length**: Indicates the length of characters from the first separator "," to the ending character "\r\n" (including "," and "\r\n"). The character type is decimal.
  *$$<Data identifier><Data length>,<IMEI>,<Command type>,<Command content><*Checksum>*\r\n
- **IMEI**: Indicates the IMEI number of the GSM module. But the number stored on the flash can be changed.
- **Command code**: Consists of letters and digits. For detail, see the chapter 3"Command Details."
- **Command content**: no more than 1,024 bytes.
- *****: It is a fixed character. **Checksum**: Contains two hexadecimal characters; indicates the sum of characters from the packet header to the asterisk (*) (including the packet header and asterisk).
  *$$<Data identifier><Data length>,<IMEI>,<Command type>,<Command content><*Checksum>*\r\n
- **\r\n**: Contains 2 bytes. The parameter is an ending character. Hexadecimal: 0x0D 0x0A.
- **$$**: Indicates the packet header sent from the MDVR to the server. Contains 2 bytes. Hexadecimal: 0x24 0x24.

If there are multiple commands, use the separator "," to separate them. If there is no command and the **Command content** parameter is required, the separator "," needs to be remained.

1.2 MDVR Command Format

Data will be uploaded in CCE protocol format.

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

Example: 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

Data parsing:

19 00 00 00: The number of remaining buffer data is 25 packets.

0C 00: This packet of data includes 12 small packets of data.

There are 12 small packets of data, they are parsed as below:

54 00: The length of this packet is 84 bytes.

15 00: The number of data ID is 21.

05 05 01 06 0A 07 00 14 00 15 02: Info of 1-byte IDs are as below:

05: There are totally 5 1-byte IDs

05 01: The GPS status is valid.

06 0A: The number of satellites is 10.

07 00: The strength of GSM is 0

14 00: The output status is 0x00

15 02: The input status is 0x00

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:

Info of 2-byte IDs are as below:

09: There are totally 9 2-bytes data IDs.

08 00 00: The current speed is 0km/h

09 1F 01: The direction is 287 degrees.

0A 07 00: The horizontal accuracy is 7.

0B 26 00: The altitude is 38 meters.

16 00 00: The voltage of AD1 is 0 volts.

17 00 00: The voltage of AD2 is 0 volts.

19 A2 01: The voltage of AD4 is 4.18 volts, representing interval battery is 4.18V.

1A 26 05: The voltage of AD5 is 13.18 volts, representing external battery is 13.18V.

40 23 00: The event code is 35.

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:

Info of 4-bytes data IDs are as below:

06: There are totally 6 4-bytes data ID.

02 D7 87 57 01: The longitude is 22.513623.

03 48 60 CC 06: The latitude is 114.057288.

04 DE BF B5 24: The time is 615890910 seconds (Indicating it's been 615890910 seconds since 2000)

0C 80 68 00 00: The mileage is 6880 meters.

0D E4 A0 03 00: Indicating the running time is 237796 seconds(Since the device turns on for the first time)

1C 01 00 00 00: Indicating the system flag is 00 00 00 01.

01 49 09 04 01 00 00 00 00 00 00 00

Info of n-bytes data IDs are as below:

01: Indicating there are totally 1 n-bytes data ID.

49: 0x49 is the ID of the camera status.

09: The length of this ID is 9 bytes

04 01 00 00 00 00 00 00 00: Camera status is 00 00 00 00 00 00 00 01 04

Note:

- A comma (,) is used to separate data characters. The character type is the American Standard Code for Information Interchange (ASCII) (hexadecimal: 0x2C).
- Symbols "<" and ">" will not be present in actual data, only for documentation purpose only.
- All multi-byte data complies with the following rule: High bytes are prior to low bytes.
- The size of a GPRS data packet is about 1460 bytes.

Descriptions about GPRS packets from the MDVR are as follows:

Parameter	Description	Example
@@ / $$	@@: Indicates the GPRS data packet header sent from the server to the device. The header type is ASCII (hexadecimal: 0x40). $$: Indicates the GPRS data packet header sent from the device to the server. The header type is ASCII (hexadecimal: 0x24).	24 24
Data identifier	Contains 1 byte. The type is the ASCII, and its	5A

		value ranges from **0x41** to **0x7A**.	
Data length		Indicates the length of characters from the separator "2C" to the ending character "0A". Decimal. The type is ASCII. $$<*Data identifier*><*Data length*>,<*IMEI*>,<*Command type*>,<*Hexadecimal data packet*><*Checksum*>\r\n	35 36 33
IMEI		Indicates the device's IMEI number. The number type is ASCII. It has 15 digits generally.	38 36 36 38 35 34 30 33 36 35 31 36 34 35 31 ASCII: 866854036516451
Command type		Hexadecimal. For details, see the chapter 2 "Command List" and chapter 3 "Command Details." The type is ASCII.	43 43 45 ASCII: CCE
The following data is hexadecimal:			
Number of remaining cache records		0x03 0x00 0x00 0x00 Contains 4 bytes; hexadecimal; little-endian	0x03 0x00 0x00 0x00 The number of remaining cache records is 3.
Number of data packets		Indicates the number of data packets that a piece of data has. Contains 2 bytes; hexadecimal; little-endian	03 00 The piece of data has three data packets.
Length of a data packet		Contains 2 bytes; hexadecimal; little-endian	AF 00 The length of a data packet is 175 bytes.
Number of IDs in a data packet		Contains 2 bytes; hexadecimal; little-endian	33 00 There are 51 ID numbers in the data packet.
Number of 1-byte parameter ID		Value range: 0x00–0xFF The length of a parameter ID is 1 byte.	0x18 There are 24 parameter ID numbers. 0x00: There is no parameter ID number whose length is 1 byte.
Parameter ID: 0x05	GPS positioning status	0x01: The GPS positioning is valid. 0x00: The GPS positioning is invalid.	0x01 The GPS positioning is valid.
Parameter ID: 0x06	Number of satellites	Indicates the number of received GPS satellites.	0x0B The number of received GPS satellites is 11.
Parameter ID: 0x07	GSM signal strength	Value range: 0x00–0x31	0x0B The GSM signal strength is 11.
Parameter	Output port	Indicates the status values of eight output	0x00

ID: 0x14	status	ports. Bits 0–7 correspond to status of output ports 1–8.	Status: Output inactive
Parameter ID: 0x15	Input port status	Indicates the status values of eight input ports. Bits 0–7 correspond to status of input ports 1–8. Hexadecimal digits need to be converted to binary digits.	0x00 Status: Input inactive
Parameter ID: 0x1B	Geo-fence number	The data is available only when the GPRS event code is 20 or 21.	0x00 There is no geo-fence number.
Parameter ID: 0x27	Temperature sensor code	The data is available only when the GPRS event code is 50 or 51.	07 Indicating the temperature sensor code is 07
Parameter ID: 0x93	Clutch_Switch	CAN data	64
Parameter ID: 0x94	Tachograph Performance	CAN data	66
Parameter ID: 0x95	Parking Brake Swtich	CAN data	5A
Parameter ID: 0x96	Cruise Control	CAN data	5B
Parameter ID: 0x97	Accelerator Pedal Position	CAN data	5C
Parameter ID: 0x9D	Fuel Level	CAN data	5D
Parameter ID: 0x9E	Actual Engine Torque	CAN data	5E
Parameter ID: 0XA1	Actual Engine Torque(Load at Current Speed)	CAN data	68
Number of 2-byte parameter ID		Value range: 0x00–0xFF The length of a parameter ID is 2 bytes.	0x10 There are 16 parameter ID numbers.
Parameter ID: 0x08	Speed	Unit: km/h; little-endian	0x00 0x00 The driving speed is 0 km/h.

Parameter ID: 0x09	Driving direction	The unit is degree. When the value is **0**, the direction is due north. The value ranges from 0 to 359. Little-endian.	0x12 0x01 The driving direction is 274 degrees.
Parameter ID: 0x0A	Horizontal dilution of precision (HDOP)	Value range: 5–999 Unit: 1/10; little-endian	0x07 0x00 The HDOP is 7.
Parameter ID: 0x0B	Altitude	Unit: meter; little-endian	0x1C 0x00 The altitude is 28.
Parameter ID: 0x16	AD1	Port AD1 analog; little-endian Voltage formula (AD1): AD1/100	0x5e 0x01 Convert the digits to decimal digits. 350/100 = 3.50 The voltage is 3.50 V.
Parameter ID: 0x17	AD2	Port AD2 analog; little-endian Voltage formula (AD2): AD2/100	0xef 0x01 Convert the digits to decimal digits. 495/100 = 4.95 The voltage is 4.95V.
Parameter ID: 0x18	AD3	Port AD2 analog; little-endian Voltage formula (AD3): AD3/100	0xef 0x01 Convert the digits to decimal digits. 495/100 = 4.95 The voltage is 4.95V.
Parameter ID: 0x19	AD4	Battery analog <AD4>; little-endian Voltage formula of battery analog (AD4): AD4/100 Formula of battery percentage: (AD4/100 - 3.4)/0.8 x 100%	0x9A 0x01 Convert the digits to decimal digits. 410/100 = 4.10 The voltage is 4.10 V.
Parameter ID: 0x1A	AD5	External power analog <AD5>; little-endian Voltage formula of external power supply (AD5): AD5/100	0x04 0x05 Convert the digits to decimal digits. 1284/100 = 12.84 The voltage is 12.84 V.
Parameter ID: 0x29	Fuel Level(Non-CAN data)	Unit: %. Little-endian	2E 0E The fuel level is 36.30%
Parameter ID: 0x40	Event code	Little-endian. Please refer to “1.3 Event Code” for more details.	23 00 Indicating the event code is 35
Parameter ID: 0x91	Vehicle Speed(from	CAN data	6F 00

	tachograph)		
Parameter ID: 0x92	Vehicle Speed(wheel based)	CAN data	70 00
Parameter ID: 0x99	Engine Speed(RPM)	CAN data	71 00
Parameter ID: 0x9C	Engine Coolant Temperature	CAN data	72 00
Parameter ID: 0x9F	Ambient Air Temperature	CAN data	00 00
Parameter ID: FE30	Overspeed recovery event assist message	The data is available only when the GPRS event code is 138. Data type: WORD	60 00 Indicates that the maximum speed during overspeed is 96Km/h
Number of 4-byte parameter ID		Value range: 0x00–0xFF The length of a parameter ID is 4 bytes.	0x07 There are 7 parameter ID numbers. 0x00: There is no parameter ID number.
Parameter ID: 0x02	Latitude	Unit: millionth of a degree; little-endian	C3 87 57 01 Convert the digits to decimal digits. The latitude is 22.513603 degrees.
Parameter ID: 0x03	Longitude	Unit: millionth of a degree; little-endian	CD 5F CC 06 Convert the digits to decimal digits. The longitude is 114.057165 degrees.
Parameter ID: 0x04	Date and time	Contains 4 bytes; little-endian Unit: second Starting time: 1 January, 2000, 00:00:00 am.	7F C7 61 22 The value is 576833407 seconds.
Parameter ID: 0x0C	Mileage	Indicates the total mileage. Unit: meter; little-endian	0x01 0x00 0x00 0x00 The total mileage is 1 meter.
Parameter ID: 0x0D	Run time	Indicates the total time. Unit: second; little-endian	72 0F 00 00 The run time is 3954 seconds.
Parameter ID: 0x1C	System flag	The data is available only when the GPRS event code is 35. Bit 0: Whether to modify the EEP2 parameter.	0x00 0x00 0x00 0x01 The EEP2 parameter is modified

		When the parameter value is 1, the EEP2 parameter is modified. Bit 1–31: reserved. Data type: DWORD	
Parameter ID: 0x25	RFID number	Little-endian The data is available only when the GPRS event code is 37.	D7 9D D1 00 RFID number is 13737431
Parameter ID：0X98	Total fuel used(L)	CAN data. Little-endian	01 02 00 00 Total fuel used is 513L
Parameter ID：0X9A	Total engine hours(h)	CAN data. Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value.	12 34 00 01 1679054.6 hours
Parameter ID：0X9B	High resolution vehicle distance(m)	CAN data. Little-endian	11 22 00 00 8721 meters
Parameter ID：0XA0	High Resolution Engine Total Fuel(L)	CAN data. Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value.	12 00 01 00 65.554 liters
Parameter ID：0XA2	Engine Fuel Rate(L/H)	CAN data. Little-endian. After the digits are converted to decimal digits, the converted value divided by 100 is the actual value.	12 00 02 00 1310.90 L/H
Parameter ID：0XA3	Axle weight(kg)	CAN data. Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value.	12 34 00 00 1333.0KG
Parameter ID：0XA4	Service distance(km)	CAN data. Little-endian	22 30 00 00 12322KM
Parameter ID：0XA5	Instantaneous Fuel Economy （km/L）	CAN data. Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value.	12 56 00 00 22.034 KM/L
Parameter ID：0XFE2F	Fatigue driving time	The data is available only when the GPRS event code is 136. data type： DWORD	10 0E 00 00 Indicates that the driver has been tired for 3600 consecutive seconds.
Number of unfixed-byte		Value range: 0x00–0xFF	0x04

parameter ID		The following data has no fixed sequences. For details, see the *MEITRACK_CCE_ID_def.xlsx*.	There are 4 parameter ID numbers. 0x00: There is no parameter ID number whose length is unfixed.
Parameter ID: 0x0E	Base station info	<Data length><MCC><MNC><LAC><CELL_ID><RX_LEVEL> Data length: indicates the length of base station data; hexadecimal. MCC: indicates Mobile Country Code; 16-bit unsigned; little-endian. MNC: indicates Mobile Network Code; 16-bit unsigned; little-endian. LAC: indicates Location Area Code; 16-bit unsigned; little-endian. CELL_ID: indicates the cell ID; 32-bit unsigned; little-endian. RX_LEVEL: indicates the signal strength; 16-bit signed; little-endian.	CC 01 01 00 2F 25 F9 3B 00 00 00 00 MCC: 0x01CC (that is, 460) MNC: 0x0001 (that is, 1) LAC: 0x252F CELL_ID: 0x00003BF9 RX_LEVEL: 0
Parameter ID: 0X2A	temperature sensor 1	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.
Parameter ID: 0X2B	temperature sensor 2	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.
Parameter ID: 0X2C	temperature sensor 3	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.
Parameter ID: 0X2D	temperature sensor 4	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.
Parameter ID: 0X2E	temperature sensor 5	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.
Parameter ID: 0X2F	temperature sensor 6	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.
Parameter ID: 0X30	temperature sensor 7	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.
Parameter ID: 0X31	temperature sensor 8	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.
Parameter ID:0x49	Cameras status	<ID_Len><Number><Status> ID_Len: ID length, 1 byte Number: Total number of cameras that this device supports. 1 bytte, no more than 64. Status: 8 bytes, bit0: 1 indicates Camera 1 is connected; 0 indicates Camera 1 is disconnected; …… bit64: 1 indicates Camera 64 is connected; 0 indicates Camera 64 is disconnected. Data Length: 10 bytes Little-endian	09 04 01 00 00 00 00 00 00 00
Parameter ID:0x4B	Current network info	<ID_Len><Version><Type><DescriptorLen><Descriptor> ID_Len: indicates the length of this ID. Contains one byte. Version: indicates the struct version. Contains one byte. Default value: 0x01. Type: indicates the current network type. Contains one byte. 0: None. 1: Mobile network. 2: WiFi. 3: LAN. DescriptorLen: indicates the length of the network descriptor. Contains one byte. The	

		parameter value ranges from **0** to **32**. Descriptor: indicates the network descriptor. The parameter value is a string.	
Parameter ID:0xFE2D	Fatique Driving information	1st byte: Data length 2nd byte: Version code, it is 02 so far. 3rd byte: Alert type, range from 1 to 8, 01: Mild fatigue 02: Moderate fatigue 03: Severe fatigue 04: Distraction alert 05: Absence alert 06: On Phone Call alert 07: Smoking alert 08: Yawning alert The rest of data indicates the image name. （If there is only one byte 01, it indicates the device fails to capture the picture.）	20 02 02 31 39 30 31 32 33 30 32 30 39 32 33 5F 45 31 31 34 5F 32 5F 4E 31 55 31 44 31 2E 6A 70 67 20：Data length is 32 bytes 02：Version code is 02 02：Moderate fatigue The rest of data indicates the image name 190123020923_E114_2_N1U1D1.jpg
Parameter ID: 0xFE31	Additional alert info of ADAS/DMS	<ID_Len><AlarmProtocol><AlarmType><PhotoName> ID_Len: indicates the length of this ID. Contains one byte. AlarmProtocol: indicates the protocol version. Contains one byte. AlarmType: indicates the alert type. Contains one byte. When the value of the parameter **AlarmProtocol** is **0x02**, the definitions of the alert type are as follows: 1: Look left. 2: Look right. 3: Raise head. 4: Lower head. 5: Drowsiness. 6: Yawning. 7: Calling. 8: Smoking. 9: Drinking. 10: Driver absence. 11: Camera occlusion. 128: Forward collision. 129: Distance detection. 130: Left lane departure. 131: Right lane departure. 132: Front vehicle started. PhotoName: indicates the photo name. The parameter value is a string. Contains 64 bytes. If no photo exists, the parameter value is **0x00**.	1F 02 07 32 30 30 39 31 36 31 36 30 30 34 31 5F 43 48 31 5F 45 31 32 36 53 37 5F 30 2E 6A 70 67 1F: The data contains 31 bytes. 02: The protocol version is 02. 07: indicates the calling alert. The remaining characters indicate the photo name, that is, 200916160041_CH1_E126S7_0.jpg.

		When the value of the parameter **AlarmProtocol** is **0x01**, the definitions of the alert type are as follows: 1: Close eyes. 2: Yawning. 3: Not defined. 4: Lower head. 5: Look left or right. 6: Driver absence. 7: Calling. 8: Smoking. 9: Camera occlusion. 10: Forward Collision Warning (FCW). 11: Urban Forward Collision Warning (UFCW). 12: Left Lane Departure Warning. 13: Right Lane Departure Warning. 14: Headway Monitoring and Warning (HMW) (The value of the parameter **FCW Level** is **2**.) 15: TTC 1. When the driver drives the vehicle at a low speed, the Time to Collision (TTC) warning is generated and the value of the parameter **FCW Level** is **3**. 16: TTC2. When the driver drives the vehicle at a high speed, the TTC warning is generated and the value of the parameter **FCW Level** is **3**. PhotoName: indicates the photo name. The parameter value is a string. Contains 32 bytes. If no photo exists, the parameter value is **0x00**. The data is available only when the GPRS event code is 126.	
Face recognition alarm auxiliary message	Parameter ID: 0XFE6A	<ID_Len><AlarmProtocol><AlarmType><PhotoName> ID_Len： indicates the length of this ID; 1 byte. AlarmProtocol: 1 byte. Default 0x01 When the AlarmProtocol is 0X01： AlarmType： 1 byte; The definition is as follows: 1: The driver logs in successfully 2: The driver fails to log in 3: Face recognition is successful 4: Change the driver 5: Delete the driver PhotoName： Photo name, string format, fixed 64 bytes, if no photo, all 0x00 The data is available only when the GPRS event code is 647. data type:STRUCT	
Parameter ID: 0XFE79	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：indicates the length of this ID; 1 byte. version: 1 byte. Default 0x01 CH_number_X: 1:CH1 2:CH2 3:CH3 4:CH4 CH_Type_X：channel type 1:ADS 2:DMS 3: Ordinary camera video_name_len_X: The upload name contains a maximum of 125 bytes video_name_x: Name of the uploaded video
Alert video info_B	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：indicates the length of this ID; 1 byte. version: 1 byte. Default 0x01 CH_number_X: 1:CH1 2:CH2 3:CH3 4:CH4 CH_Type_X：channel type 1:ADS 2:DMS 3: Ordinary camera video_name_len_X: The upload name contains a maximum of 125 bytes video_name_x: Name of the uploaded video

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； 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.
ASPC People Counter	Parameter ID: 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
Length of location data 2: AF 00. Contains 2 bytes; little-endian. The length of the current location data is 175 bytes. The data analysis method is the same as that described in the preceding text.		
Length of location data 3: AF 00. Contains 2 bytes; little-endian. The length of the current location data is 175 bytes. The data analysis method is the same as that described in the preceding text.		
*	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 characters from the	30 36 ASCII: 06

	packet header to the checksum (excluding the checksum and ending character). Hexadecimal *$$<Data identifier><Data length>,<IMEI>,<Command type>,<Hexadecimal data packet><*Checksum>*\r\n	
\r\n	Contains 2 bytes. The parameter is an ending character. The type is ASCII (hexadecimal: 0x0d,0x0a).	\r\n

For details about definitions of CCE IDs, see the *MEITRACK_CCE_ID_def.xlsx*.

1.3 Event Code

Event Code	Event	Default SMS Header (At Most 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
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
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 (change uart rate)**	/
41	**Stop Moving**	Quiet
42	**Start Moving**	Moving
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
82	**Fuel Filling**	Fuel Filling
83	**Ult-Sensor Drop**	Ult-Sensor Drop
94	**Output 1 Active**	Out1 Active
95	**Output 2 Active**	Out2 Active
96	**Output 3 Active**	Out3 Active
99	**Output 1 Inactive**	Out1 Inactive
100	**Output 2 Inactive**	Out2 Inactive
101	**Output 3 Inactive**	Out3 Inactive
114	**Driving Behavior**	Driving Behavior
126	ADAS/DMS Alarm	ADAS/DMS Alarm
129	**Harsh braking**	Harsh Braking
130	**Harsh acceleration**	Fast Accelerate
139	**Maintenance Notice**	Maintenance
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
582	**CH7 Video Loss**	CH7 Video Loss
583	**CH8 Video Loss**	CH8 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
616	**CH7 Video Recovery**	CH7 Recovery
617	**CH8 Video Recovery**	CH8 Recovery
647	**Face ID Alarm**	Face ID Alarm

2 Command List

Command Description
A10 – Real-Time Location Query (GPRS)
A11 – Setting a Heartbeat Packet Reporting Interval (GPRS)
A12 – Tracking by Time Interval (GPRS)
A13 – Setting the Cornering Report (GPRS)
A14 – Tracking by Distance
A15 – Setting the Parking Scheduled Tracking Function (GPRS)
A16 – Enabling the Parking Scheduled Tracking Function (GPRS)
A17 – Controlling Output 1 Status by RFID/iButton
A21 – Setting GPRS Parameters
A23 – Setting the Standby GPRS Server
A25 – Setting GPRS Parameters
A70 – Reading All Authorized Phone Numbers
A71 – Setting Authorized Phone Numbers
A72 – Setting Listen-in Phone Numbers
A73 – Setting the Smart Sleep Mode
A9A – Transmitting Audio and Video Data in Real Time
A9B – Controlling Real-Time Audio and Video Transmission
A9C – Querying the Resource List
A9D – Playing Back Videos Remotely
A9E – Controlling Remote Video Playback
A9F – Uploading Files
AA0 – Controlling File Uploading
AA1 – Obtaining the WiFi List
AA2 – Sending the FTP File Uploading Progress
AA3 – Obtaining MDVR Network Status
AA4 – Querying which days' video files have been stored
AB2 - Transmitting Audio and Video Data in Real Time By Using the RTMP
AB3 - Controlling Real-Time Audio and Video Transmission By Using the RTMP
AB4 - Playing Back Videos Remotely By Using the RTMP (GPRS)
AB5 - Controlling Remote Video Playback By Using the RTMP
AB8 - Querying the Resource List From Data Packets
ABB - Setting the WiFi Hotspot Function
B05 – Setting a Geo-Fence

B06 – Deleting a Geo-Fence
B07 – Setting the Speeding Alert
B08 – Setting the Towing Alert
B10 – Fast Setting the Towing Alert
B11 – Setting a Polygonal Geo-Fence
B22 – Setting the Mileage and Speed Calculation Mode
B26 – Setting Filtering Time of an Input Port
B31 – Turning off the LED Indicator
B34 – Setting a Log Interval
B35 – Setting the Local Time Zone
B36 – Setting the GPRS Time Zone
B64 - Setting FTP upload photo parameters
B91 – Setting SMS Event Characters
B99 – Setting Event Authorization
BB8 - Setting the Speaker Volume Level of the MDVR
C01 – Controlling Output Status
C02 – Notifying the Device of Sending an SMS
C03 – Setting a GPRS Event Transmission Mode
C40 – Registering a Temperature Sensor Number
C41 – Deleting a Registered Temperature Sensor
C42 – Reading the Temperature Sensor SN and Number
C43 – Setting the Temperature Threshold and Logical Name
C44 – Reading Temperature Sensor Parameters
C46 – Checking Temperature Sensor Parameters
C47 – Setting Fuel Parameters
C48 – Reading Fuel Parameters
C49 – Setting the Fuel Theft Alert
C61 – Transparently Transmitting Data over the Serial Port
C90 – Setting the Driver Fatigue Function
CB8 - Setting Event Playing
CFF – Deleting an Event in the Buffer
D10 – Authorizing a RFID Card/iButton Key
D11 – Authorizing RFID Cards/iButton Keys in Batches
D12 – Checking RFID/iButton Authorization
D13 – Reading an Authorized RFID Card/iButton Key
D14 – Deleting an Authorized RFID Card/iButton Key
D15 – Deleting Authorized RFID Cards/iButton Keys in Batches
D16 – Checking the Checksum of the Authorized RFID/iButton ID Database
D65 – Setting the Maintenance Mileage
D66 – Setting Maintenance Time
D72 – Setting Output Triggering
D73 – Allocating GPRS Cache and GPS Log Storage Space

D79 – Setting Harsh Acceleration and Harsh Braking Parameters
DA0 – Obtaining All Alert Parameters of a Tire Pressure Sensor
DA1 – Obtaining Data of All Bound Tire Pressure Sensors
DA2 – Obtaining Data of a Tire Pressure Sensor
DA3 – Deleting Tire Pressure Sensors
DA4 – Obtaining Data of Multiple Tire Pressure Sensors
DA5 – Setting Alert Thresholds of a Tire Pressure Sensor
E91 – Reading Device's Firmware Version and SN
F00 – Restarting the GSM and GPS Modules
F01 – Restarting the GSM Module
F02 – Restarting the GPS Module
F08 – Setting the Mileage and Run Time
F09 – Deleting SMS/GPRS Cache Data
F11 – Restoring Initial Settings

3 Command Details

3.1 Real-Time Location Query (GPRS) – A10

GPRS Sending	A10
GPRS Reply	$$<*Data identifier*><*Data length*>,<IMEI>,<CCE>,<*Number of remaining cache records*><*Number of data packets*><*Data packet about event code 34*><**Checksum*>\r\n
Description	**34**: indicates the event code of the GPRS command.
Example	
GPRS Sending	@@A25,865789020991321,A10*62\r\n
GPRS Reply	$$A118,865789020991321,CCE,<00 00 00 00 01 00 54 00 12 00 06 01 22 05 00 06 00 07 15 14 00 15 00 04 08 00 00 09 14 01 0A E7 03 0B 00 00 06 02 25 87 57 01 03 E3 60 CC 06 04 41 3A 2D 20 0C 74 0D 00 00 0D EC 50 03 00 1C 00 00 00 00 02 0E 0C CC 01 01 00 45 A5 8B D4 E9 01 01 FF 1D 08 00 25 86 A7 0B 0A D5 FF>*1D\r\n

3.2 Setting a Heartbeat Packet Reporting Interval (GPRS) – A11

GPRS Sending	A11,*Interval*
GPRS Reply	A11,OK
Description	The heartbeat packet function is used to keep the Transmission Control Protocol (TCP) connection open when the interval of scheduled GPRS reporting is long. Interval = 0: function disabled (default). Interval = [1…65535]: function enabled. Unit: minute. The heartbeat function is available only in conjunction with deep sleep mode. When the device enters deep sleep mode, a heartbeat packet will be sent at the specified interval. A heartbeat packet is to confirm the device is online, and positioning data is invalid.

Example	
GPRS Sending	@@S28,353358017784062,A11,10*FD\r\n
GPRS Reply	$$S28,353358017784062,A11,OK*FE\r\n *After the above command is run successfully, the device will send a GPRS heartbeat packet to the platform every 10 minutes in sleep mode.*

3.3 Tracking by Time Interval (GPRS) – A12

GPRS Sending	A12,*Interval*
GPRS Reply	A12,OK
Description	Unit: x10 seconds Interval = 0: function disabled. The maximum time interval is 65535 x 10 seconds. 6 x 10 seconds are recommended.
Example	
GPRS Sending	@@V27,353358017784062,A12,6*D5\r\n
GPRS Reply	$$V28,353358017784062,A12,OK*02\r\n *After the above command is run successfully, the device will send a GPRS data packet to the platform every 1 minute.*

3.4 Setting the Cornering Report (GPRS) – A13

GPRS Sending	A13,*Angle*
GPRS Reply	A13,OK
Description	When the driving angle exceeds the preset value, the device will send a GPRS data packet with location information to the server, which ensures a smoother route on the platform. Angle = 0: function disabled (default). Angle = [1…359]: function enabled. Recommended value: **30**.
Example	
GPRS Sending	@@X29,353358017784062,A13,120*37\r\n
GPRS Reply	$$X28,353358017784062,A13,OK*05\r\n *After the above command is run successfully, if the cornering angle is greater than 120 degree, the device will send a GPRS data pakcet to the server.*

3.5 Tracking by Distance – A14

GPRS Sending	A14,*Distance*
GPRS Reply	A14,OK
Description	Distance = 0: function disabled (default). Distance = [1…65535]: function enabled. Unit: meter. Note: When both the GPRS time interval and distance tracking functions are enabled, the "first reach first report" rule will be applied. For example, set the time interval to 6 x 10

	seconds and distance to 200 meters. If the road is clear, a distance data packet will be reported first; if there is heavy traffic on the road, a time interval data packet will be reported first. Then both the time interval and distance counters will be reset to 0. **300** is recommended.
Example	
GPRS Sending	@@D30,353358017784062,A14,1000*4A\r\n
GPRS Reply	$$D28,353358017784062,A14,OK*F2\r\n *After the above command is run successfully, if the driving distance reaches 1000m, the device will send a data packet in CCE format to the server.*

3.6 Setting the Parking Scheduled Tracking Function (GPRS) – A15

GPRS Sending	A15,*Interval*
GPRS Reply	A15,OK
Description	With the function, the number of GPRS messages is reduced, and thus GPRS traffic is saved. After the A15 function is set, the A16 function is automatically enabled. For details about engine status, see section 3.7 "Enabling the Parking Scheduled Tracking Function (GPRS) – A16." Interval unit: x10 seconds Interval = 0: function disabled. The maximum interval is 65535 x 10 seconds. Note: If data needs to be sent at the specified interval after the vehicle starts or stops, the function needs to work with the A12 function.
Example	
GPRS Sending	@@E27,353358017784062,A15,6*C7\r\n
GPRS Reply	$$E28,353358017784062,A15,OK*F4\r\n

3.7 Enabling the Parking Scheduled Tracking Function (GPRS) – A16

GPRS Sending	A16,*Status*
GPRS Reply	A16,OK
Description	**The first positive input port (high level) of the device must connect to engine detection. Otherwise, the function is unavailable.** When the activation status is **1**, the parking scheduled tracking function is enabled; when the activation status is **0**, the function is disabled. GPRS data is sent at the following interval: ● Interval of the A12 function when the engine is on ● Interval of the A15 function when the engine is off
Example	
GPRS Sending	@@F27,353358017784062,A16,0*C3\r\n
GPRS Reply	$$F28,353358017784062,A16,OK*F6\r\n

3.8 Controlling Output 1 Status by RFID/iButton – A17

GPRS Sending	A17,X
GPRS Reply	A17,OK
Description	X = 1: function enabled. Before using the function, ensure that ACC detection is connected to input 3 and a RFID card has been authorized. X = 0: function disabled (default). For example: After swiping the authorized RFID card, you must start the engine within 1 minute. If the time exceeds 1 minute, you need to swipe the card again. After the engine is started, input 3 has been detecting the ACC status. If ACC ON is detected (that is, input 3 is the high level), output 1 will not generate data. If ACC OFF is detected, after 1 minute, swipe the authorized RFID card to start the engine as required. For details about how to authorize a RFID card, see commands D10–D15.
Example	
GPRS Sending	@@T27,353358017784062,A17,1*D3\r\n
GPRS Reply	$$T28,353358017784062,A17,OK*05\r\n

3.9 Setting GPRS Parameters – A21

GPRS Sending	A21,*Connection mode*,*IP address*,*Port*,*APN*,*APN user name*,*APN password*
GPRS Reply	A21,OK
Description	Connection mode = 0: function disabled. Connection mode = 1: function enabled; use TCP/IP reporting mode. Connection mode = 2: function enabled; use UDP reporting mode. IP address: IP address or domain name. A maximum of 32 bytes are supported. Port: a maximum of 5 digits. APN/APN user name/APN password: a maximum of 32 bytes respectively. If no user name and password are required, leave them blank.
Example	
GPRS Sending	@@H48,353358017784062,A21,1,67.203.13.26,8800,,,*C9
GPRS Reply	$$H28,353358017784062,A21,OK*F4\r\n

3.10 Setting the Standby GPRS Server – A23

GPRS Sending	A23,*IP address*,*Port*
GPRS Reply	A23,OK
Description	IP address: a maximum of 32 bytes Port: a maximum of 5 digits When the device fails to send data to the active server set by command A21, data is automatically sent to the standby server to prevent data loss.
Example	
GPRS Sending	@@S43,353358017784062,A23,67.203.13.26,8800*F0
GPRS Reply	$$S28,353358017784062,A23,OK*01\r\n

3.11 Setting the IP3 Parameters – A25

GPRS Sending	A25, *Connection mode,IP address,Port,APN,APN user name,APN password*
GPRS Reply	A25,OK
Description	Connection mode = 0: function disabled. Connection mode = 1: function enabled; use TCP/IP reporting mode. Connection mode = 2: function enabled; use UDP reporting mode. IP address: IP3 address or domain name. A maximum of 32 bytes are supported. Port: a maximum of 5 digits. APN/APN user name/APN password: a maximum of 32 bytes respectively. If no user name and password are required, leave them blank.
Example	
GPRS Sending	@@H48,353358017784062,A25,1,67.203.13.26,8800,,,*C9
GPRS Reply	$$H28,353358017784062,A25,OK*F4\r\n

3.12 Reading All Authorized Phone Numbers – A70

GPRS Sending	A70
GPRS Reply	A70,*SOS phone number 1,SOS phone number 2,SOS phone number 3,Listen-in phone number 1,Listen-in phone number 2*
Description	Read all authorized phone numbers.
Example	
GPRS Sending	@@T25, 353358017784062,A70*93\r\n
GPRS Reply	$$T85,353358017784062,A70,13811111111,13822222222,13833333333,13844444444,13855555555*21\r\n

3.13 Setting Authorized Phone Numbers – A71

GPRS Sending	A71,*Phone number 1,Phone number 2,Phone number 3*
GPRS Reply	A71,OK
Description	Phone number: A phone number has a maximum of 16 bytes. If no phone numbers are set, leave them blank. Phone numbers are empty by default. Phone number 1: SOS phone number. When you call the device by using the phone number, you will receive SMS notification about the location, geo-fence alert and low power alert. When the SOS button is pressed, the device will dial phone numbers 1, 2, and 3 in sequence. The device stops dialing when a phone number responds.
Example	
GPRS Sending	@@U61,353358017784062,A71,13811111111,13822222222,13833333333*7D\r\n
GPRS Reply	$$U28,353358017784062,A71,OK*06\r\n

3.14 Setting Listen-in Phone Numbers – A72

GPRS Sending	A72,Listen-in phone number 1,Listen-in phone number 2
GPRS Reply	A72,OK
Description	When you call the tracker by using authorized listen-in phone numbers, the tracker will answer the call automatically and enter the listen-in state. In this way, the tracker will not make any sound. Listen-in phone number: A maximum of two phone numbers can be set. Each phone number has a maximum of 16 digits. If no phone numbers are set, leave them blank. Phone numbers are empty by default. If no phone numbers are set and commas are remained, phone numbers set before will be deleted.
Example	
GPRS Sending	@@V49,353358017784062,A72,13844444444,13855555555*55\r\n
GPRS Reply	$$V28,353358017784062,A72,OK*08\r\n

3.15 Setting the Smart Sleep Mode – A73

GPRS Sending	A73,Sleep level
GPRS Reply	A73,OK
Description	Set the automatic smart sleep mode when the tracker is idle. Sleep level = 0: function disabled (default). Sleep level = 1: normal sleep. The GSM module always works, and the GPS module occasionally enters the sleep mode. The tracker works 25% longer in the normal sleep mode than that in the normal working mode. This mode is not recommended for short interval tracking; this will affect the route precision. Sleep level = 2: deep sleep. If no event is triggered after five minutes, the GPS module will stop working and the GSM module will enter sleep mode. Once an event is triggered, the GPS and GSM modules will be woken up. A heartbeat event will be triggered only in the deep sleep mode, which will be uploaded every one hour by default. Note: In any condition, you can use an SMS or a GPRS command to disable the sleep mode, and then the tracker exits the sleep mode and returns back to the normal working mode.
Example	
GPRS Sending	@@W27,353358017784062,A73,2*D9\r\n
GPRS Reply	$$W28,353358017784062,A73,OK*0A\r\n

3.16 Transmitting Audio and Video Data in Real Time – A9A

GPRS Sending	A9A,*Real-time audio and video transmission request struct*
GPRS Reply	A9A,OK<*Error code*>
Description	The definitions of the real-time audio and video transmission request struct information are as follows:

```
typedef struct _live_media_request
{
    BYTE ip_len;            //Length of the server IP address
    BYTE ip_addr[64];       //IP address,a maximum of 64 bytes
    WORD tcp_port;          //TCP port length, big-endian
    WORD udp_port;          //UDP port (reserved) , big-endian
    BYTE chn;               //Logical channel number
    BYTE data_type;         //Data type. 0: Audio and video. 1: Video. 2: Two-way calling
                            //3: Listen-in 4: Broadcasting(reserved). 5: Transparent transmission. These two parameters are reserved.
    BYTE stream_type;       //Bitrate type. 0: Major stream. 1: Minor stream.
}
```

Logical channel number: The audio and video channel number ranges from 1 to 64. The two-way calling channel number is 129. The listen-in channel number ranges from 65 to 128. (Listen-in channel numbers correspond to audio and video channel numbers. That is, the audio source of listen-in channel 65 comes from the microphone of audio and video channel 1.)

After receiving the A9A command, the device will establish a real-time audio and video transmission connection (TCP connection by default. Do not support the UDP connection.). Meanwhile, the device will send audio and video data in the following format (Platform's reply is not required):

Start Byte	Field	Data Type	Description
0	Frame header flag	BYTE	Fixed value: 0x12
1	m_pt	BYTE	Load type flag value. m_pt(1BYTE) = load type (bit7 ~ bit1) + flag bit (bit0). load type (7bits): H264 is 98, H265 is 99, audio (G.726) is 8, audio (G.711A) is 6, and GPS data is 45 flag bit (1bit): If the data packet is the last one in the video frame, the value is 1. Otherwise, the value is 0. (for example: if data is audio, the parameter value is 0x11; If data is H264 video and the packet is the last packet of this video frame, the value is 0xC5; otherwise, it is 0xC4; If data is H265 video and the packet is the last packet of this video frame, the

			parameter value is 0xC7, otherwise 0xC6; If data is the GPS data in the playback file, the parameter value is 0x5B)
2	Data packet No.	WORD	The starting number is 0. When a RTP packet is sent, the packet No. is the existing packet No. plus 1. Big-endian
4	IMEI number	BCD[8]	Indicates the device's IMEI number.
12	Logical channel number	BYTE	The audio and video channel number ranges from 1 to 64. The two-way calling channel number is 129. The listen-in channel number ranges from 65 to 128.
13	Data type	4BITS	0000: video I-frame 0001: video P-frame 0010: video B-frame 0011: audio frame 0100: transparent transmission for data
	Data packet processing flag	4BITS	0000: data packet with all complete data, which cannot be divided. 0001: the first packet while processing data packets 0010: the last packet while processing data packets 0011: the middle packet while processing data packets
14	Timestamp	BYTE[8]	Show the time of the RTP data packet. Unit: ms. When the data type is 0100, the field is 0,big-endian.
22	Previous I-frame interval	WORD	Indicate the interval between the existing frame and the previous I-frame. Unit: ms. When the data type is not video frame, the field is 0,big-endian
24	Previous frame interval	WORD	Indicate the interval between the existing frame and the previous frame. Unit: ms. When the data type is not video frame, the field is 0,big-endian
26	Data body length	WORD	Audio and video data length,big-endian.

	28	Data body	BYTE[n]	Audio and video data(the data length does not exceed 950 bytes)
Example				
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."			
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."			

3.17 Controlling Real-Time Audio and Video Transmission – A9B

GPRS Sending	A9B,*Real-time audio and video transmission control struct info*
GPRS Reply	A9B,OK<*Error code*>

Description	The definitions of the real-time audio and video transmission control struct information are as follows:
	typedef struct
	{
	BYTE logiChn; //Logical channel number
	BYTE controlCmd; //Control command. The platform can control the device's real-time audio and videos by the command.
	//0: Disable audio and video transmission.
	//1: Switch the bitrate.
	//2: Pause the sending of all streams in this channel. (reserved)
	//3: Resume sending paused stream. The stream type is the same that of the paused stream. (reserved)
	//4: Disable two-way calling (only available for two-way calling logical channel 129).
	BYTE closeAVtype; //Disable the audio and video type.
	//0: Disable related audio and video data in this channel.
	//1: Disable related audio data in this channel and remain related video data. (reserved)
	//2: Disable related video data in this channel and remain related audio data. (reserved)
	BYTE switchCodetype; //Switch the bitrate type. Switch previous bitrates to newly applied bitrates.
	//The audio remains unchanged.
	//Newly applied bitrate: 0: Major stream. 1: Minor stream
	}

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.18 Querying the Resource List – A9C

GPRS Sending	A9C,*Resource list querying struct info*
GPRS Reply	A9C,*Resource file struct info*

Description

The definitions of the resource list querying struct information are as follows:

```
Typedef struct
{
   WORD alarmNum;                    //Number of alerts. When the number is 0,
all alerts are selected, little-endian
   WORD alarmCode[alarmNum]; //Alert event code , little-endian
}ExAlarmCode;
typedef struct
{
   BYTE logiChn;              //Channel number
   BYTE t_start[6];          //Start time: YY-MM-DD-HH-MM-SS,
                                    //0: There is no condition about the start time.
   BYTE t_end[6];             //End time: YY-MM-DD-HH-MM-SS,
                                    //0: There is no condition about the end time.
   BYTE alarm_flag[8];      //Reserved, useless so far, all could be filled by 00
   BYTE srcAVtype;           //Audio and video resource type
                                    //0: Audio and video
                                    //1: Audio
                                    //2: Video
                                    //3: Video, or audio and video
   BYTE streamtype;          //Bitrate type
                                    //0: All streams
                                    //1: Major stream
                                    //2: Minor stream
   BYTE captype;              //Memory type
                                    //0: All memories
                                    //1: Active memory
                                    //2: Standby memory
   ExAlarmCode code;     //Alert event code(If here is blank, it means all of alerts
                                    //video files and normal video files will be selected.)
}
```

The definitions of the file struct information are as follows:

```
typedef struct
{
   BYTE logiChn;          //Channel number
   BYTE t_start[6];       //Start time: YY-MM-DD-HH-MM-SS
   BYTE t_end[6];         //End time: YY-MM-DD-HH-MM-SS
   BYTE alarm_flag[8];  //Alert event code
                              //Bytes 0–5: reserved
                              //Bytes 6–7: Correspond to Meitrack's event codes,
big-endian.
   BYTE srcAVtype;       //Audio and video resource type
                              //0: Audio and video
                              //1: Audio
```

	//2: Video
	//3: Video, or audio and video
	BYTE streamtype; //Bitrate type
	//0: All streams
	//1: Major stream
	//2: Minor stream
	BYTE captype; //Memory type
	//0: All memories
	//1: Active memory
	//2: Standby memory
	DWORD FileLen; //File size. Unit: byte, big-endian
	} FileMsg_t;
	The definitions of the replied resource file struct information are as follows:
	typedef struct
	{
	DWORD Number; //Number of audio and video resources (*N*) ,big-endian
	ReplyMsg_t FileSrc[N]; //
	};

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.19 Playing Back Videos Remotely – A9D

GPRS Sending	A9D,*Remote video playback request struct info*
GPRS Reply	A9D,OK/*Error code*

Description	The definitions of the remote video playback request struct information are as follows:
	typedef struct
	{
	BYTE ip_len;
	BYTE ip_addr[64]; //IP address,a maximum of 64 bytes
	WORD tcp_port; // Big-endian
	WORD udp_port; // Big-endian
	BYTE logiChn; //Logical channel number
	BYTE avType; //Audio and video resource type
	//0: Audio and video
	//1: Audio
	//2: Video
	//3: Reserved (video, or audio and video)
	BYTE streamType; //Bitrate type
	//0: Reserved (major stream or minor stream)
	//1: Major stream

	//2: Minor stream;
	//If the channel only transmits audio, this field value is 0.
	BYTE capType; //Memory type
	//0: All memories
	//1: Active memory
	//2: Standby memory
	BYTE reviewStyle; //Playback mode
	//0: Normal playback
	//1: Fast forward(reserved)
	//2: Fast rewind keyframes(reserved)
	//3: Play keyframes(reserved)
	//4: Upload a single frame (reserved)
	BYTE viewRank; //Fast-forward or fast-rewind times. When the playback mode is 1 or 2, the field content is valid.
	//0: Invalid 1: One time 2: Two times 3: Three times 4: Eight times 5: 16 times
	// (reserved, default value is 0)
	BYTE t_start[6]; //Start time: YY-MM-DD-HH-MM-SS. When the playback mode is 4,
	//the field indicates the uploading time of a single frame.
	BYTE t_end[6]; //End time: YY-MM-DD-HH-MM-SS
	//When the playback mode is 0, videos are played back without interruption.
	//When the playback mode is 4, the field is invalid.
	}
	After receiving the A9D command, the device will establish a real-time audio and video transmission connection (TCP connection by default. Do not support the UDP connection.). Meanwhile, to implement the audio and video playback function, the device will send audio and video data whose format is the same as that of the A9A command. (Platform's reply is not required.)

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.20 Controlling Remote Video Playback – A9E

GPRS Sending	A9E,*Struct info*
GPRS Reply	A9E,OK/*Error code*

Description	The definitions of the struct information are as follows:
	typedef struct
	{
	BYTE chn; //Logical channel number
	BYTE reviewControl; //Playback control

	//0: Start playback (reserved)
	//1: Pause playback (reserved)
	//2: End playback
	//3: Fast forward (reserved)
	//4: Fast rewind keyframes (reserved)
	//5: Drag to play back
	//6: Play keyframes (reserved)
	BYTE viewRank; //Fast-forward or fast-rewind times
	//When the playback mode is 3 or 4, the field content is valid.
	//Otherwise, the value is 0.
	//0: Invalid 1: One time 2: Two times 3: Three times 4: Eight times 5: 16 times
	// (reserved, default value is 0)
	BYTE dragPoint[6]; //Drag playback points
	//BCD[6]: YY-MM-DD-HH-MM-SS
	//When the playback mode is 5, the field content is valid.
	}

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.21 Uploading Files – A9F

GPRS Sending	A9F,*File uploading request struct info*
GPRS Reply	A9F,*File list struct info*

Description	The definitions of the file uploading request struct information are as follows:
	typedef struct
	{
	WORD alarmNum; // Number of alerts,little-endian
	WORD alarmCode[alarmNum]; // Alert event code list,little-endian
	}ExAlarmCode;
	typedef struct _term_upload_src_list
	{
	BYTE IPLen; //Length of the server IP address
	BYTE IP[64]; //FTP server IP address, within 64 bytes
	WORD PORT; //FTP server port, big-endian
	BYTE UserLen; //Length of the user name
	BYTE User[64]; //User name, within 64 bytes
	BYTE PWLen; //Length of the password
	BYTE PW[64]; //Password, within 64 bytes
	BYTE FilePathLen; //Length of the file uploading path
	BYTE FilePath[256]; //File uploading path, within 256 bytes
	BYTE logiChn; //Channel number

```
    BYTE t_start[6];            //Start time: YY-MM-DD-HH-MM-SS.
    BYTE t_end[6];              //End time: YY-MM-DD-HH-MM-SS.
    BYTE alarm_flag[8];   //0: Search all files (reserved).
    BYTE srcAVtype;             //Audio and video resource type
                                //0: Audio and video
                                //1: Audio
                                //2: Video
                                //3: Video, or audio and video
    BYTE streamtype;            //Bitrate type
                                //0: All streams
                                //1: Major stream
                                //2: Minor stream
    BYTE captype;               //Storage location
                                //0: All memories
                                //1: Active memory
                                //2: Standby memory
    BYTE Execute;               //Task execution condition, which is represented as a bit.
                                //Bit 0: WiFi. 1: Upload by WiFi.
                                //Bit 1: LAN. 1: Upload when a LAN network is connected.
                                //Bit2: 3G/4G. 1: Upload when a 3G or 4G network is
                                connected.
    ExAlarmCode code;       // Alert event code （If the number of alert events is 0, the
                                item can be omitted）
}
```

The definitions of the replied file list struct information are as follows:

```
typedef struct _term_upload_reply
typedef struct
{
    BYTE logiChn;         //Channel number
    BYTE t_start[6];            //Start time: YY-MM-DD-HH-MM-SS
    BYTE t_end[6];        //End time: YY-MM-DD-HH-MM-SS
    BYTE alarm_flag[8]; //Alert event code
                                //Bytes 0–5: reserved
                                //Bytes 6–7: Correspond to Meitrack's event codes.
    BYTE srcAVtype;             //Audio and video resource type
                                //0: Audio and video
                                //1: Audio
                                //2: Video
                                //3: Video, or audio and video
    BYTE streamtype;            //Bitrate type
                                //0: All streams
                                //1: Major stream
                                //2: Minor stream
```

	BYTE captype; //Memory type
	//0: All memories
	//1: Active memory
	//2: Standby memory
	DWORD FileLen; //File size. Unit: byte
	}FileMsg_t;
	{
	BYTE flag; // OK/Error code
	//When an error code occurs, the following file information struct is empty.
	DWORD Number; //Number of audio and video resources (*N*) ,little-endian
	FileMsg_t FileSrc[N]; //Uploading file information
	} term_upload_reply_t;

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.22 Controlling File Uploading – AA0

GPRS Sending	AA0,*Struct info*
GPRS Reply	AA0,OK/*Error code*

Description	The definitions of the struct information are as follows:
	typedef struct _term_upload_src_list
	{
	BYTE Flag; //Uploading control
	//0: Pause (reserved)
	//1: Continue (reserved)
	//2: Cancel
	BYTE FileName[128]; //Control the name of the file to be uploaded.
	}

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.23 Obtaining the WiFi List – AA1

GPRS Sending	AA1
GPRS Reply	AA1,WiFi list struct info
Description	Obtain nearby WiFi hotspot information; The definitions of the WiFi list struct information are as follows: typedef struct _term_upload_src_list {

	byte Cnt; //Number of obtained WiFi (N)
	byte SSID1_Type; //SSID format 0: Unicode 1: ACSII
	byte SSID1_Len; //SSID length
	byte SSID1[]; //SSID. The length depends on SSID1_Len.
	byte SSID1_Rssi; //SSID signal value. The larger the value is, the stronger the signal strength is. Value range: 0–100.
	...
	byte SSIDn_Type; //SSID format 0: Unicode 1: ACSII
	byte SSIDn_Len; //SSID length
	byte SSIDn[]; //The length depends on SSID1_Len.
	byte SSIDn_Rssi; //SSID signal value. The larger the value is, the stronger the signal strength is. Value range: 0–100.
	}

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.24 Sending the FTP File Uploading Progress – AA2

GPRS Sending	None
GPRS Reply	AA2,*FTP file uploading progress struct*

Description	The definitions of the FTP file uploading progress struct information are as follows:
	typedef struct
	{
	BYTE Percent; //Uploading progress percentage. Value range: 0–100.
	BYTE FileName[128]; //File name
	}
	When a FTP file is uploaded, the device will send the current FTP file uploading progress to the platform.

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.25 Obtaining MDVR Network Status – AA3

GPRS Sending	AA3
GPRS Reply	AA3,*Network status struct info*

Description	The definitions of the network status struct information are as follows:
	Typedef struct
	{
	byte CurUser; //Current network
	//0: None;

```
                                    //1: GSM;
                                    //2: WIFI;
                                    //3: LAN
    byte GSM_Status;                //0: Not detected
                                    //1: Normal
                                    //2: Abnormal
    byte GSM_Simcard_Ready;         //0: Not ready
                                    //1: Ready
    byte GSM_SimcardNum[16];        //SIM card number
    byte GSM_SimcardIMSI[16];       //IMSI
    byte GSM_Type;                  //0: Not register
                                    //1: 2G
                                    //2: 3G
                                    //3: 4G
    byte GSM_CSQ;                   //Signal value. The maximum value is 31.
    byte GSM_IMEI[16];
    byte GSM_Connect;               //0: Not call
                                    //1: Calling
                                    //2: Call succeeded
                                    //3: Call failed
    byte WIFI_Status;               //0: Not detected
                                    //1: Normal
                                    //2: Abnormal
    byte WIFI_Mode;                 //0: AP     1: Station
    byte WIFI_SSID[128];            //Unicode code. Big-endian.
    byte WIFI_Rssi;                 //WiFi signal value. Only available for the Station
mode.
    byte WIFI_IP[15];               //IP address of WiFi
    byte WIFI_Mac[6];               //MAC address of WiFi
    byte WIFI_SubnetMast[15];       //Subnet mask settings
    byte WIFI_DefaultGateway[15];   //Gateway settings
    byte WIFI_PrimaryDNSServer[15]; //Active DNS server settings
  byte WIFI_SecondaryDNSServer[15]; //Standby DNS server settings
    byte LAN_Status;                //0: Not detected
                                    //1: Normal
                                    //2: Abnormal
    byte LAN _IP[15];               //IP address of LAN
    byte LAN _Mac[6];               //MAC address of LAN
    byte LAN _SubnetMast[15];       //Subnet mask settings
    byte LAN _DefaultGateway[15];   //Gateway settings
    byte LAN _PrimaryDNSServer[15]; //Active DNS server settings
     byte LAN _SecondaryDNSServer[15];  //Standby DNS server settings
}
```

Example

GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.26 Querying which days' video files have been stored – AA4

GPRS Sending	AA4[,YYMM]
GPRS Reply	AA4, *struct info*

Description	If the A44 command is sent without YYMM parameter, it indicates querying all the storage of the device to check which days' videos have been stored. If the A44 command is sent with YYMM parameter, it indicates which days in the specified month and year the videos have be stored;
	
	The definitions of the WiFi list struct information are as follows:：
	typedef struct MediaRecInfo_S
	{
	BYTE YYMM[4]; //The BCD code for year and month, for example, June 2020 is 0906.
	DWORD mediaRecFlag;//Indicates which day of the month video files were saved; little-endian.
	// bit0～bit30 indicates the 1st～31st of the month,and bit31 is reserved;
	//bit=1 means there is lerts video file on that day, otherwise there is none;
	DWORD alarmRecFlag; // Indicates which day of the month alerts video files were saved; little-endian.
	// bit0～bit30 indicates the 1st～31st of the month,and bit31 is reserved;
	//bit=1 means there is alerts video file on that day, otherwise there is none;
	}
	MediaRecInfo_S stMediaRecInfo[N];
	N: According to the query results of video files, if there are video files in different months, N will increase. For example, if there are video files in January and March 2020, N is 2.

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.27 Transmitting Audio and Video Data in Real Time By Using the RTMP – AB2

GPRS Sending	AB2,*Real-time audio and video transmission request struct*
GPRS Reply	AB2,OK<*Error code*>
Description	The definitions of the real-time audio and video transmission request struct information

	are as follows:
	typedef struct _live_media_request
	{
	BYTE rtmp_upload_len; //Length of the RTMP upload address
	BYTE rtmp_upload_addr[256]; //RTMP upload address
	BYTE chn; //Logical channel number
	BYTE data_type;//Data type. 0: Audio and video. 1: Video. 2: Two-way calling.
	//3: Listen-in. 4: Broadcasting (reserved). 5: Transparent transmission.
	BYTE stream_type; //Bitrate type. 0: Major stream. 1: Minor stream.
	BYTE rtmp_down_len; //Length of the RTMP download address. This data is available when the value of the parameter **Data type** is **2**.
	BYTE rtmp_down_addr[256];//RTMP download address. This data is available when the value of the parameter **Data type** is **2**.
	}
	Logical channel number: The audio and video channel number ranges from **1** to **64**. The two-way calling channel number is **129**. The listen-in channel number ranges from **65** to **128**.
	After receiving the AB2 command, the device will establish a real-time audio and video transmission connection and push streams via RTMP.

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.28 Controlling Real-Time Audio and Video Transmission By Using the RTMP – AB3

GPRS Sending	AB3,*Real-time audio and video transmission control struct*
GPRS Reply	AB3,OK<*Error code*>

Description	The definitions of the struct information are as follows:
	typedef struct
	{
	BYTE logiChn //Logical channel number
	BYTE controlCmd;//Control command. The platform can control the device's real-time audio and videos by the command.
	//0: Disable audio and video transmission.
	//1: Switch the bitrate.
	//2: Pause the sending of all streams in this channel (reserved).
	//3: Resume sending paused stream. The stream type is the same that of the paused stream (reserved).
	//4: Disable two-way calling.
	BYTE closeAVtype; //Disable the audio and video type.
	//0: Disable related audio and video data in this channel.
	//1: Disable related audio data in this channel and remain

	related video data (reserved).
	//2: Disable related video data in this channel and remain related audio data (reserved).
	BYTE switchCodetype; //Switch the bitrate type. Switch previous bitrates to newly applied bitrates. The audio remains unchanged.
	//Newly applied bitrate: 0: Major stream. 1: Minor stream.
	}

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.29 Playing Back Videos Remotely By Using the RTMP (GPRS) – AB4

GPRS Sending	AB4,*Struct info*
GPRS Reply	AB4,OK<*Error code*>

Description	The definitions of the struct information are as follows:
	typedef struct _PlayBackrequest
	{
	BYTE rtmp _len; //RTMP address length
	BYTE rtmp_addr[256]; //RTMP address
	BYTE logiChn; //Logical channel number
	BYTE avType; //Audio and video resource type. 0: Audio and video. 1: Audio. 2: Video. 3: Video, or audio and video.
	BYTE streamType; //Bitrate type. 0: Major stream or minor stream. 1: Major stream. 2: Minor stream. If only audio can be transmitted in this channel, the value of this field is **0**.
	BYTE capType; //Memory type. 0: All memories. 1: Active memory. 2: Standby memory.
	BYTE reviewStyle; //Playback mode. 0: Normal playback. 1: Fast forward (reserved).
	//2: Fast rewind keyframes (reserved).
	//3: Play keyframes (reserved). 4: Upload a single frame (reserved).
	BYTE viewRank; //Fast-forward or fast-rewind times. When the playback mode is **1** or **2**, the field content is valid. Otherwise, the value is **0**.//0: Invalid. 1: One time. 2: Two times. 3: Three times. 4: Eight times. 5: 16 times.//(Reserved. Default value: 0.)
	BYTE t_start[6]; //Start time: YY-MM-DD-HH-MM-SS. When the playback mode is **4**, the field indicates the uploading time of a single frame.
	BYTE t_end[6]; //End time: YY-MM-DD-HH-MM-SS. When the playback mode is **0**, videos keep playing back.//When the playback mode is **4**, this field is invalid.
	}

	The data transmission format of the device is the same as that of the RTMP.
Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.30 Controlling Remote Video Playback By Using the RTMP – AB5

GPRS Sending	AB5,*Struct info*
GPRS Reply	AB5,OK<*Error code*>

Description	The definitions of the struct information are as follows:
	typedef struct _PlayBack_control
	{
	BYTE chn; //Channel number
	BYTE reviewControl; //Playback control. 0: Start playback (reserved). 1: Pause playback (reserved). 2: End playback. 3: Fast forward (reserved). 4: Fast rewind keyframes (reserved). 5: Drag to play back. 6: Play keyframes (reserved).
	BYTE viewRank; //Fast-forward or fast-rewind times. When the playback mode is **3** or **4**, the field content is valid. Otherwise, the value is **0**.
	//0: Invalid. 1: One time. 2: Two times. 3: Three times. 4: Eight times. 5: 16 times.//(Reserved. Default value: 0.)
	BYTE dragPoint[6]; //Drag the playback time point: YY-MM-DD-HH-MM-SS. When the playback mode is **5**, the field content is valid.
	}
	Note: Any of the following methods can be used to drag a time point to play back videos on the platform.
	1. If only one person watches a video, send a command for stopping file playing and then a playback request command.
	2. If multiple people watch a video, send a dragging command.

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.31 Querying the Resource List From Data Packets – AB8

GPRS Sending	AB8,*Querying struct info*
GPRS Reply	AB8,*Reply struct info*

Description	1. The definitions of the resource list querying struct information are as follows:
	typedef stuct
	{
	word MainAlarmCode;//Alert event code (EEPID alert event code): little-endian
	word subAlarmCode; //Sub-event code: little-endian
	}Alarm_t;
	typedef struct

{
WORD alarmNum; //Number of alerts: When the number is 0, all alerts are selected. Little-endian.
Alarm_t alarm[alarmNum]; //
}ExAlarmCode;
typedef struct //Specified data packet
{
WORD N //Obtain a specified data packet. A maximum of 100 data packets are supported. Little-endian.
WORD BUF[N], little-endian
}Appoint_PACK;

typedef struct
{
BYTE logiChn; //Channel number
BYTE t_start[6];//Start time: YY-MM-DD-HH-MM-SS. 0: There is no condition about the start time.
BYTE t_end[6]; //End time: YY-MM-DD-HH-MM-SS. 0: There is no condition about the end time.
BYTE alarm_flag[8]; //Reserved. Default value: 0. Little-endian.
BYTE srcAVtype; //Resource type. 0: Audio and video. 1: Audio. 2: Video.
//3: Video, or audio and video. 4: Photo.
BYTE streamtype; //Bitrate type. 0: All streams. 1: Major stream. 2: Minor stream.
BYTE captype; //Memory type. 0: All memories. 1: Active memory. 2: Standby memory.
ExAlarmCode code; //Alert event struct
Appoint_PACK code2; //Obtain a specified data packet. This parameter is available when a data packet is lost.
}
2. The definitions of the reply struct information are as follows:
typedef struct
BYTE logiChn; //Channel number
BYTE t_start[6]; //Start time: YY-MM-DD-HH-MM-SS
BYTE t_end[6]; //End time: YY-MM-DD-HH-MM-SS
BYTE res[6]; //Reserved.
WORD event_code; //Meitrack event code: little-endian
WORD subEventCode; //Sub-event code: little-endian
BYTE srcAVtype; //Resource type. 0: Audio and video. 1: Audio. 2: Video.
3: Video, or audio and video. 4: Photo.
BYTE streamtype; //Bitrate type. 0: All streams. 1: Major stream. 2: Minor stream.
BYTE captype; //Memory type. 0: All memories. 1: Active memory. 2: Standby memory.
DWORD fileLen; //File size. Unit: byte. Little-endian.

	}ReplyMsg_t;
	typedef struct
	{
	WORD all_pack; Number of data packets: The parameter value ranges from **1** to **65535**. Little-endian.
	WORD cur_pack; Current data packet: little-endian
	DWORD all_file_num; Number of files: little-endian
	DWORD Number; //Number of uploaded files: little-endian
	ReplyMsg_t Src[Number]; //
	};

Example	
GPRS Sending	For details, see the chapter 5 "Appendix: Struct Data Analysis."
GPRS Reply	For details, see the chapter 5 "Appendix: Struct Data Analysis."

3.32 Setting the WiFi Hotspot Function – ABB

GPRS Sending	ABB,*X*,*Y*,*Z*
GPRS Reply	ABB,OK/*Error code*
Description	X: Whether to enable the hotspot function. The parameter value is **0** or **1**. Decimal. **0**: function disabled. **1**: function enabled. Y: indicates the hotspot name. The parameter value is a string. The parameter contains a maximum of 64 characters. (Commas are not allowed.) Z: indicates the hotspot password. The parameter value is a string. The parameter contains a maximum of 32 characters and a minimum of eight characters. (Commas are not allowed.) If you want to read the command settings, send **ABB**.
Example	
GPRS Sending	@@H57,353358017784062,ABB,1,asd,123*96\r\n
GPRS Reply	$$H28,353358017784062,ABB,OK*F7\r\n

3.33 Setting a Geo-Fence – B05

GPRS Sending	B05,*Geo-fence number*,*Latitude*,*Longitude*,*Radius*,*Enter Geo-fence alert*,*Exit Geo-fence alert*
GPRS Reply	B05,OK
Description	Geo-fence number: 1–8. A maximum of eight geo-fences can be set. Latitude: latitude of the geo-fence center; decimal; accurate to 6 digits after the decimal point. If there are only 4 digits after the decimal point, add two digits 0. Otherwise, the command cannot be used successfully. Longitude: longitude of the geo-fence center; decimal; accurate to 6 digits after the decimal point. If there are only 4 digits after the decimal point, add two digits 0. Otherwise, the command cannot be used successfully.

	Radius: The value ranges from 1 to 4294967295. The unit is meter. Enter Geo-fence alert = 0: function disabled. Enter Geo-fence alert = 1: function enabled. Exit Geo-fence alert = 0: function disabled. Exit Geo-fence alert = 1: function enabled.
Example	
GPRS Sending	@@H57,353358017784062,B05,1,22.913191,114.079882,1000,0,1*96\r\n
GPRS Reply	$$H28,353358017784062,B05,OK*F7\r\n *When the device exits the geo-fence (latitude: 22.913191; longitude: 114.079882; radiu: 1000 meters), it will send a GPRS data packet about a geo-fence alert (event code 21) to the server.*

3.34 Deleting a Geo-Fence – B06

GPRS Sending	B06,*Geo-fence number*
GPRS Reply	B06,OK
Description	Geo-fence number: 1–8. Only one geo-fence can be deleted each time by SMS or GPRS command.
Example	
GPRS Sending	@@J27,353358017784062,B06,1*C8\r\n
GPRS Reply	$$J28,353358017784062,B06,OK*FA\r\n *After the above command is run successfully, the first geo-fence will be deleted.*

3.35 Setting the Speeding Alert – B07

GPRS Sending	B07,*Driving speed*
GPRS Reply	B07,OK
Description	Driving speed = 0: function disabled (default). Driving speed = [1…255]: function enabled. Unit: km/h. When the driving speed reaches the preset value, a speeding alert will be generated.
Example	
GPRS Sending	@@P28,353358017784062,B07,60*05\r\n
GPRS Reply	$$P28,353358017784062,B07,OK*01\r\n *When the device's driving speed reaches 60 km/h, it will send a GPRS data packet about a speeding alert (event code 19) to the server.*

3.36 Setting the Towing Alert – B08

GPRS Sending	B08,*Consecutive vibration time*
GPRS Reply	B08,OK
Description	When the device's vibration time exceeds the preset value, the device will send an alert to an authorized phone number or the server. Before using the towing alert function, use

	the A73 command to set the smart sleep level to **2** and use the B08 command to set the consecutive vibration time. Otherwise, the towing alert function will be unavailable. Consecutive vibration time = 0: function disabled (default). Consecutive vibration time = [1…255]: function enabled. Unit: second.
Example	
GPRS Sending	@@I27,353358017784062,B08,3*CB\r\n
GPRS Reply	$$I28,353358017784062,B08,OK*FB\r\n *When the device vibrates for more than three consecutive seconds, it will send a GPRS data packet about a towing alert (event code 36) to the server.*

3.37 Fast Setting the Towing Alert – B10

GPRS Sending	B10,*Consecutive vibration time*,*Idling time*
GPRS Reply	B10,OK
Description	Consecutive vibration time = 0: function disabled (default). Consecutive vibration time = [1…255]: function enabled. Unit: second. Idling time: The default value is **2**. Unit: minute. Idling time = 0: The power-saving mode will be disabled. Idling time = [1…255]: The power-saving function will be enabled. When the idling time exceeds the preset value, the device will enter power-saving mode.
Example	
GPRS Sending	@@I27,353358017784062,B10,3*6E\r\n
GPRS Reply	$$I28,353358017784062,B10,OK*9E\r\n *When the device vibrates for more than three consecutive seconds, it will send a GPRS data packet about a towing alert to the server.*

3.38 Setting a Polygonal Geo-Fence – B11

GPRS Sending	B11,*Geo-fence number*,*Latitude 1*,*Longitude 1*,*Latitude 2*,*Longitude 2*...*Latitude N*,*Longitude N*,*Enter Geo-fence alert*,*Exit Geo-fence alert*
GPRS Reply	B11,OK
Description	Geo-fence number: The parameter value ranges from 1 to 8. (The maximum value varies depending on customization projects.) Latitude: accurate to 6 digits placed after the decimal point. For example, 22.512517 or -22.512517. Longitude: accurate to 6 digits placed after the decimal point. For example, 114.057200 or -114.057200. Enter Geo-fence alert: The parameter value is **0** or **1**. ● **0**: An alert will not be generated when the device enters the geo-fence. ● **1**: An alert will be generated when the device enters the geo-fence. Exit Geo-fence alert: The parameter value is **0** or **1**. ● **0**: An alert will not be generated when the device exits the geo-fence. ● **1**: An alert will be generated when the device exits the geo-fence.

	If the command only cotains the parameter **Geo-fence number**, related geo-fences will be deleted.
Example	
GPRS Sending	@@I94,353358017784062,B11,1,22.526922,114.052695,22.526946,114.056232,22.523720,114.053521,1,1*D5\r\n
GPRS Reply	$$I28,353358017784062,B11,OK*F5\r\n

3.39 Setting the Mileage and Speed Calculation Mode – B22

GPRS Sending	B22,*Calculation mode X*/*Rotational speed ratio K*
GPRS Reply	B22,OK/*Rotational speed ratio K*
Description	**X = 0** (default): Use GPS speed. **X = 1**: Use the RPM speedometer and use GPS speed to automatically calibrate the rotational speed ratio K (recommended). **X = 2**: Use the RPM speedometer and press the SOS button to calibrate the rotational speed ratio K. Rotational speed ratio K calibrated by GPS speed are not accurate. You can send the command **B22,2** to calibrate it again. You have to stop the vehicle after the mileage of the vehicle speedometer changes. The buzzer will make a long buzzing sound after the device receives the calibration command, indicating that the device enters the calibration state. At the same time, the green LED indicator will steady on. In this way, you have to drive the vehicle (no speed limit) and stop it when the driving distance reaches 1 km. Then press and hold down the SOS button for 2 seconds. The speaker will make two sounds, indicating that the rotational speed ratio K is calibrated successfully. If the calibration cannot be completed within 10 minutes, the device will exit the calibration state and you have to do the operations again. Besides, the green LED indicator will be off and the rotational speed ratio K will be sent. X = K ≥ 3: Use the RPM speedometer and the rotational speed ratio is K. Rotational speed ratio K: K pulses/km X: decimal 3 ≤ K ≤ 65535
Example	
GPRS Sending	@@A28,353358017784062,B22,60*F3
GPRS Reply	$$A28,353358017784062,B22,OK*F4

3.40 Setting Filtering Time of an Input Port – B26

GPRS Sending	B26,1:T1,2:T2,...n:Tn
GPRS Reply	B26,OK
Description	**n**: The value ranges from 1 to 5, which corresponds to input ports 1–5. **Tn**: indicates the filtering time. Value range: 0–65535; unit: x10ms You can set one or multiple input ports at a time. If you want to read filtering time of an input port, send **B26**.

Example	
GPRS Sending	@@Y39,868998030732297,B26,1:1000,2:1000*30 \r\n
GPRS Reply	$$Y28,868998030732297,B26,OK*1E\r\n

3.41 Turning off the LED Indicator – B31

GPRS Sending	B31,A
GPRS Reply	B31,OK
Description	When the value of A is 00, the tracker's indicator is turned on (default). You can query the device's running status according to the indicator status. When the value of A is 10, the tracker's indicator is turned off.
Example	
GPRS Sending	@@P27,353358017784062,B31,10*D1\r\n
GPRS Reply	$$P28,353358017784062,B31,OK*03\r\n

3.42 Setting a Log Interval – B34

GPRS Sending	B34,*Log interval*
GPRS Reply	B34,OK
Description	Set the interval for recording data to device's memory when the GPS signal is valid. When there is no GPS signal, data will not be recorded. Recorded logs can only be read by Meitrack Manager software. Log interval = 0: function disabled (default). Log interval = [1…65535]: function enabled. Unit: second.
Example	
GPRS Sending	@@N28,353358017784062,B34,60*03\r\n
GPRS Reply	$$N28,353358017784062,B34,OK*FF\r\n

3.43 Setting the Local Time Zone – B35

GPRS Sending	B35,*SMS minute*
GPRS Reply	B35,OK
Description	The default time zone of the device is GMT 0. You can run the B35 command to change the time zone of video recording, photo capturing and SMS reports to the local time zone. The time zone of an SMS report is different from that of a GPRS data packet. When **SMS minute** is **0**, the time zone is GMT 0. When **SMS minute** is a value ranging from -32768 to 32767, set time zones.
Example	
GPRS Sending	@@O29,353358017784062,B35,480*3C\r\n
GPRS Reply	$$O28,353358017784062,B35,OK*01\r\n *After the above command is run successfully, the device SMS time zone is changed to UTC+08:00 (China time zone).*

3.44 Setting the GPRS Time Zone – B36

GPRS Sending	B36,*GPRS minute*
GPRS Reply	B36,OK
Description	When **GPRS minute** is **0**, the time zone is GMT 0 (default). The platform can automatically detect the user time zone, so that the GPRS time zone does not need to be changed. Otherwise, inaccurate data occurs. When **GPRS minute** is a value ranging from -32768 to 32767, set time zones.
Example	
GPRS Sending	@@P29,353358017784062,B36,480*3E\r\n
GPRS Reply	$$P28,353358017784062,B36,OK*03\r\n *After the above command is run successfully, the GPRS time zone is changed to UTC+08:00 (China time zone).*

3.45 Setting FTP upload photo parameters – B64

GPRS Sending	B64,H,username,password,host,port,path
GPRS Reply	B64,OK
Description	01 H: 0 means turn off the FTP function, 1 means turn on FTP upload, 2 means clear the last parameters 02 username: Maximum 50 bytes 03 password: Maximum 50 bytes 04 hostname: Maximum 50 bytes 05 hostport: Maximum 5 bytes 06 path: Maximum 100 bytes 07 If the parameter does not need to be changed, leave them blank with a comma 08 Send command withot parameter means reading the parameters
Example	
GPRS Sending	@@V27,353358017784062,B64,1,test,test,quectel.3322.org,10001,/meitrack/cxc/mp3_file/*D5\r\n
GPRS Reply	$$P28,353358017784062,B36,OK*03\r\n$$S28,353358017784062,B64,OK*FE\r\n

3.46 Setting SMS Event Characters – B91

GPRS Sending	B91,*SMS event code*,*SMS header*
GPRS Reply	B91,OK
Description	Header: a maximum of 16 bytes
Example	
GPRS Sending	@@R31,353358017784062,B91,1,SOS*F0\r\n
GPRS Reply	$$R28,353358017784062,B91,OK*06\r\n *After you press the SOS button (input 1), the device will send an alert SMS whose header*

	is SOS to a preset authorized phone number.

3.47 Setting Event Authorization – B99

GPRS Sending	B99,<SMS>/<0>,*<Phone number location>*/*<Authorized phone number>*,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<CALL>/<1>,*<Phone number location>*/*<Authorized phone number>*,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<GPRS>/<2>,*<Operation code>*,[*Event code 1*]...[*Event code n*] 0000,B99,<CAMERA>/<3>,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<BUZZER>/<4>,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<OUT1>/<5>,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<OUT2>/<6>,*<Operation code>*,[*Event code 1*]...[*Event code n*]
GPRS Reply	B99,<SMS>/<0>,*<Phone number location>*,*<Authorized phone number>*,[*Event code 1*]...[*Event code n*] B99,<CALL>/<1>,*<Phone number location>*,*<Authorized phone number>*,[*Event code 1*]...[*Event code n*] B99,<GPRS>/<2>,[*Event code 1*]...[*Event code n*] B99,<CAMERA>/<3>,[*Event code 1*]...[*Event code n*] B99,<BUZZER>/<4>,[*Event code 1*]...[*Event code n*] B99,<OUT1>/<5>,*<Operation code>*,[*Event code 1*]...[*Event code n*] B99,<OUT2>/<6>,*<Operation code>*,[*Event code 1*]...[*Event code n*]
Description	Fields SMS, CALL, CAMERA, GPRS, BUZZER, OUT1, and OUT2 can be presented by 0–6 in decimal string. Operation codes GET, SET, ADD, and DEL can be presented by 0–3 in decimal string. These characters are not case-sensitive. Note: Ensure that an authorized phone number is set by using the A71 command or the parameter configuration tool before the B99 command is used to set the SMS/CALL event code. The device will compare the authorized phone number issued by B99 with the authorized phone number (excluding +86 characters) of the device. If the phone numbers are the same, the new event code will be stored. If the phone numbers are inconsistent, an error SMS will be sent.
Example	
GPRS Sending	@@B34,863070010825791,B99,gprs,get*BC\r\n
GPRS Reply	$$B33,863070010825791,B99,1,17,18*B5\r\n

3.48 Setting the Speaker Volume Level of the MDVR – BB8

GPRS Sending	BB8,*N*
GPRS Reply	BB8,OK/*<Error code>*
Description	N: The parameter value ranges from **0** to **100**. If you want to read the command settings, send **BB8**.
Example	

GPRS Sending	@@V27,353358017784062,BB8,10*D5\r\n
GPRS Reply	$$S28,353358017784062,BB8,OK*FE\r\n

3.49 Controlling Output Status – C01

GPRS Sending	C01,*Speed*,*ABCDE*
GPRS Reply	C01,OK
Description	When the speed is **0**, no speed limit exists. That is, when the device receives a command, the function will take effect immediately. When the speed is a value ranging from 1 to 255 (unit: km/h), set the speed limit. When the driving speed is lower than the speed limit, the function will take effect. A = 0, close output (output 1) - open drain A = 1, open output (output 1) - connect to GND A = 2, remain previous status. B = 0, close output (output 2) - open drain B = 1, open output (output 2) - connect to GND B = 2, remain previous status. C = 0, close output (output 3) - open drain C = 1, open output (output 3) - connect to GND C = 2, remain previous status. D = 0, close output (output 4) - open drain D = 1, open output (output 4) - connect to GND D = 2, remain previous status. E = 0, close output (output 5) - open drain E = 1, open output (output 5) - connect to GND E = 2, remain previous status.
Example	
GPRS Sending	@@M34,353358017784062,C01,20,10122*18\r\n
GPRS Reply	$$M28,353358017784062,C01,OK*F9\r\n

3.50 Notifying the Device of Sending an SMS – C02

GPRS Sending	C02, X,*Phone number*,*Content*
GPRS Reply	C02,OK
Description	Used for the platform to notify the device of sending an SMS to a mobile phone. X = 0: in TEXT mode X = 1: in Unicode mode Phone number: a maximum of 16 digits Content: a maximum of 140 characters After receiving the message, the device sends Content information to specified phone numbers.
Example	

GPRS Sending	@@f47,353358017784062,C02,0,15360853789,Meitrack*B1\r\n
GPRS Reply	$$f28,353358017784062,C02,OK*13\r\n

3.51 Setting a GPRS Event Transmission Mode – C03

GPRS Sending	C03,X
GPRS Reply	C03,OK
Description	X = 0: automatic event report (default) X = 1: Before another event can be transmitted, existing event reports need to be confirmed and deleted on the server by the AFF command. Select this mode when GPRS uses UDP.
Example	
GPRS Sending	@@f27,353358017784062,C03,0*E1\r\n
GPRS Reply	$$f28,353358017784062,C03,OK*14\r\n

3.52 Registering a Temperature Sensor Number – C40

GPRS Sending	C40,SN1 & number 1,SN2 & number 2,…,SNn & number n
GPRS Reply	C40,SN1 & number 1 & result, SN2 & number 2 & result,…SNn & number n & result
Description	Commands C40 to C46 are used to read or set a temperature sensor. Installation steps: 1) Check whether the temperature sensor number in GPRS data is 0. 2) If the number is 0, the temperature sensor is not numbered. Then send the C42 command to read the mappings of sensor SNs and numbers. 3) Use the C40 command to index all sensors and bind information in the database, such as the IMEI number, SN, number, and customized name. 4) If a high or low temperature alert is required, send the C43 command to set the temperature value and customize a name. You are advised to use the installation path as the name and save the name to the database. 5) If the sensor is pulled out or replaced when the device is online, use the C46 command to check the sensor. If data is inconsistent, use the C40 and C43 commands to set data. The device uploads current temperature data by the AAA event. If the number in temperature data is 0, the temperature sensor is not registered. The platform automatically sends the C42 command to obtain the temperature sensor SN and number list. Find out the sensor whose number is 0, and register it. n: The maximum value is 8. SN: unique number to identify a temperature sensor. Eight bytes. Hexadecimal string. The SN is displayed on the platform like 28 1B D5 23 04 00 00 57, which is the same as that on the sensor label. Number: one byte. Hexadecimal. The value ranges from 1 to 254. Registration result: 0x01, 0x02, 0x03, and 0x04 0x01: The registration is successful.

	0x02: The number or SN already exists. 0x03: All sensors are registered. 0x04: Registration failed. Hexadecimal.
Example (ASCII is used to display examples because hexadecimal characters cannot be displayed.**)**	
GPRS Sending	@@q35,012896001078259,C40,(1BD5#040000W02*50\r\n
GPRS Reply	$$q36,012896001078259,C40,(1BD5#040000W0201*1B \r\n

3.53 Deleting a Registered Temperature Sensor – C41

GPRS Sending	C41,*Number 1*,*Number 2*,…*Number n*
GPRS Reply	C41,*Number 1*,*Result*,*Number 2*,*Result*,…*Number n*,*Result*
Description	Number: indicates the registered sensor number; hexadecimal. The value ranges from 1 to 254. Result: Decimal. **1** indicates deletion succeeded. **2** indicates that the number does not exist. **3** indicates deletion failed. To delete all registered temperature sensors, send command C41 only. If deletion is successful, **OK** is returned. If not, **Error** is returned.
Example	
GPRS Sending	@@n28,012896001078259,C41,01*19\r\n
GPRS Reply	$$n30,012896001078259,C41,01,1*37\r\n

3.54 Reading the Temperature Sensor SN and Number – C42

GPRS Sending	C42
GPRS Reply	C42,SN1 and number 1,SN2 and number 2,…SNn and number n
Description	SNn: indicates the n(th) sensor SN, and has eight bytes in hexadecimal format. Number n: indicates the n(th) sensor number, and has one byte in hexadecimal format. The value ranges from 0 to 255. If the value is **0**, the temperature sensor is not registered.
Example (ASCII is used to display examples because hexadecimal characters cannot be displayed.)	
GPRS Sending	@@m25,012896001078259,C42*89\r\n
GPRS Reply	$$t45,012896001078259,C42,(B4v#040000R00,(1BD5#040000W00*13\r\n

3.55 Setting the Temperature Threshold and Logical Name – C43

GPRS Sending	C43,*Number 1/SN1/High temperature value 1/Low temperature value 1/High temperature alert 1/Low temperature alert 1/Logical name 1/…Number n/SNn/High temperature value n/Low temperature value n/High temperature alert 1/Low temperature alert 1/Logical name n*
GPRS Reply	C43,*Number 1/Result 1/Number 2/Result 2…/Number n/Result n*
Description	n: The maximum value is 8. Number: one byte in hexadecimal format. SN: indicates the temperature sensor SN, and has eight bytes in hexadecimal format.

	High/Low temperature value: two bytes in hexadecimal format. The first byte is the integer part. When the high bit is **1**, the first byte is a negative integer. When the high bit is **0**, the first byte is a positive integer. The second byte is the decimal part. High temperature alert: one byte in hexadecimal format. Low temperature alert: one byte in hexadecimal format. Logical name (customized name): 16 bytes in hexadecimal format. If the name length is less than 16 bytes, add 0x00. There are 15 English characters, and # is located at the end of English characters to distinguish the Unicode and English characters. A maximum of eight Chinese characters can be supported. Chinese characters must be the Unicode. Result: one byte in hexadecimal format. **0x01** indicates setting succeeded. **0x02** indicates that the number is not located. **0x03** indicates that setting failed due to wrong parameters. Note: Separators (/) are not required between parameters.
Example (ASCII is used to display examples because hexadecimal characters cannot be displayed.)	
GPRS Sending	@@o57,012896001078259,C43,01(1BD5#040000W<0005000101T1#000000000000000 00000000000*3F
GPRS Reply	$$o28,012896001078259,C43,0101*85

3.56 Reading Temperature Sensor Parameters – C44

GPRS Sending	C44
GPRS Reply	C44,*Number 1/SN1/High temperature value 1/Low temperature value 1/High temperature alert 1/Low temperature alert 1/Logical name 1/…Number n/SNn/High temperature value n/Low temperature value n/High temperature alert 1/Low temperature alert 1/Logical name n*
Description	n: The maximum value is 8. Number: one byte in hexadecimal format. SN: indicates the temperature sensor SN, and has eight bytes in hexadecimal format. High/Low temperature value: two bytes in hexadecimal format. The first byte is the integer part. When the high bit is **1**, the first byte is a negative integer. When the high bit is **0**, the first byte is a positive integer. The second byte is the decimal part. High temperature alert: one byte in hexadecimal format. Low temperature alert: one byte in hexadecimal format. Logical name (customized name): 16 bytes in hexadecimal format. If the name length is less than 16 bytes, add 0x00. There are 15 English characters, and # is located at the end of English characters to distinguish the Unicode and English characters. A maximum of eight Chinese characters can be supported. Chinese characters must be the Unicode. Note: Separators (/) are not required between parameters.
Example (ASCII is used to display examples because hexadecimal characters cannot be displayed.)	
GPRS Sending	@@r25,012896001078259,C44*90\r\n
GPRS Reply	$$r274,012896001078259,C44,01(B4v#040000R00000000000000000000000 0000000000000000000000002(1BD5#040000W0000000000000000000000000 00000000000000000000000000000000000000000000000000000000000000

	000000000000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000000 00000000000000000000*1E\r\n

3.57 Checking Temperature Sensor Parameters – C46

GPRS Sending	C46
GPRS Reply	C46,*Checksum*
Description	Checksum: two bytes in hexadecimal format. Use CRC-CCITT to calculate parameters of eight temperature sensors (in sequence: number, SN, high temperature value, low temperature value, high temperature alert, low temperature alert, and logical name). The calculation result is used as the temperature sensor checksum.
Example	
GPRS Sending	@@i25,012896001078259,C46*89\r\n
GPRS Reply	$$i28,012896001078259,C46,12_*F1\r\n

3.58 Setting Fuel Parameters – C47

GPRS Sending	C47,*Sensor type*,*Alert percentage upper limit*,*Alert percentage lower limit*
GPRS Reply	C47,OK
Description	Sensor type: The parameter value is **0**, **1**, **2**, and **3**. ● **0**: No fuel level sensor is connected. ● **1**: A C-type fuel level sensor (AD2) is connected. ● **2**: A R-type fuel level sensor (AD2) is connected. ● **3**: A V-type fuel level sensor (AD2) is connected. The AD2 of the MVT600 and T1 is connected to the fuel level sensor by default. Alert percentage upper limit: When the value is **0**, the alert will be cleared. When the value is not **0**, GPRS and SMS event flags will take effect automatically. When the fuel percentage is higher than or equal to the value, an alert is generated, and the alert event code is **52**. Alert percentage lower limit: When the value is **0**, the alert wil be cleared. When the value is not **0**, GPRS and SMS event flags will take effect automatically. When the fuel percentage is lower than or equal to the value, an alert is generated, and the alert event code is **53**. If you want to modify a parameter, other parameters need to be left blank and separators (,) must be remained. If you only send **C47**, all parameter values will be initialized to **0**. All the parameter values are decimal characters. R-type fuel level sensor: resistive fuel level sensor C-type fuel level sensor: capacitive fuel level sensor V-type fuel level sensor: voltage-type fuel level sensor

	A53 and A54 are V-type fuel level sensors.
Example	
GPRS Sending	@@f33,353358017784062,C47,2,90,10*0A\r\n
GPRS Reply	$$f28,353358017784062,C47,OK*1C\r\n

3.59 Reading Fuel Parameters – C48

GPRS Sending	C48
GPRS Reply	C48,*Sensor type*,*Alert percentage upper limit*,*Alert percentage lower limit*
Description	The format of returned parameters is the same as that of the C47 command. All the parameter values are decimal characters.
Example	
GPRS Sending	@@c25,353358017784062,C48*89\r\n
GPRS Reply	$$c33,353358017784062,C48,2,90,10*D0\r\n

3.60 Setting the Fuel Theft Alert – C49

GPRS Sending	C49,*Time for fuel check*,*Percent of fuel decrease*
GPRS Reply	C49,OK
Description	Time for fuel check = 0: function disabled. Time for fuel check = [1…255]: function enabled. Decimal; unit: minute; default value: 3. Percent of fuel decrease = 0: function disabled. Percent of fuel decrease = [1…100]: function enabled. Decimal; default value: 2. By default, the percent of fuel decrease is 2% within 3 minutes, a fuel theft alert will be generated (for example: **C49,3,2**). Note: The percent of fuel decrease must be over two times larger than the percent of fuel sensor accuracy. For example, if the fuel sensor accuracy is 10 mm and its height is 500 mm, the recommended percent of fuel decrease is 4% (10/500 x 2).
Example	
GPRS Sending	@@c29,353358017784062,C49,3,2*4B\r\n
GPRS Reply	$$c28,353358017784062,C49,ok*5B\r\n

3.61 Transparently Transmitting Data over the Serial Port – C61

GPRS Sending	C61,*Server date & time*,*Config*,*Interface device No.*,*Data packet*
GPRS Reply	C61,*GPS date & time*,*Interface device No.*,*<Data packet>/<Error code>*
Description	Interface device No.: contains 1 byte; hexadecimal. Server date & time: indicates the date and time of the server; 14 characters. For example, **20121114235959**. GPS date & time: indicates the date and time of the device; 14 characters. For example, **20121114235959**. Config: Reserved value for later use.

	Interface device No.: The default value is **2**. Data packet: at most 512 bytes; only support GPRS. Note: When the device receives data from a peripheral, data packets will be uploaded. If data packets are not detected from a peripheral, an error code will be sent.

3.62 Setting the Driver Fatigue Function – C90

GPRS Sending	C90,A,B,C,D,E
GPRS Reply	C90,OK
Description	1. Parameter A: indicates the alert volume. The parameter value is 0, 1, 2, and 225. Decimal. 0: No sound. 1: Medium volume. 2: High volume. 225: reserved for DIP switches. Parameter B, C, D, and E: indicates an alert. Decimal. B: Absence alert. 0: function disabled. 1: function enabled. C: Distraction alert. 0: function disabled. 1: function enabled. D: Smoking alert. 0: function disabled. 1: function enabled. E: On Phone Call alert. 0: function disabled. 1: function enabled. 2. If you want to read the parameters, send C90. 3. Parameter settings must be complete. If the network connection is poor or parameter settings are not correct, an error code will be replied.
Example	
GPRS Sending	@@R35,868725036977468,C90,2,1,1,1,1*60\r\n
GPRS Reply	$$R28,868725036977468,C90,OK*1E\r\n

3.63 Setting Event Playing – CB8

GPRS Sending	CB8,A;B1,C1,D1,E1;B2,C2,D2,E2...Bn,Cn,Dn,En
GPRS Reply	CB8,OK/<*Error code*>/B1,C1,D1,E1;B2,C2,D2,E2;…Bn,Cn,Dn,En
Description	A: indicates the operation code. Decimal. The parameter value is **1** or **2**. **1**: An event code is added or modified. **2**: An event code is deleted. Bn: indicates the event code. Decimal. Cn: indicates the video channel. Decimal. When the parameter value is **0**, all channels are enabled. When the parameter value is not 0, the channel *n* is enabled. Dn: indicates the playing time. Decimal. Unit: second. When the parameter value is **0**, a video keeps playing. When the parameter value is not 0, a video plays for a specified time period. The maximum parameter value is **65535**. En: indicates the playing order of priority. Decimal. The parameter value ranges from **0** to **64**. The smaller the parameter value is, the higher the priority is. If a new video with a higher priority is playing, the current video with a lower priority will be stopped. If a new video with the same priority is playing, the current video will be stopped. A maximum of 64 events to be played can be set at a time. The maximum value of *n* is **64**.

	When the value of the parameter **A** is **2**, the value of the parameters **Cn**, **Dn** and **En** is **0**. If you want to read the command settings, send **CB8**.
Example	
GPRS Sending	@@R35,868725036977468,CB8,2;1,1,3,1;2,2,3,2*60\r\n
GPRS Reply	$$R28,868725036977468,CB8,OK*1E\r\n

3.64 Deleting an Event in the Buffer – CFF

GPRS Sending	CFF,Quantity of deleted data
GPRS Reply	CFF,CFF data packet
Description	Quantity of deleted data: hexadecimal. In general, the number is 1. The data identifiers from the device and server must be consistent. Otherwise, data will not be deleted from the device. If data is transmitted in CFF format, send CFF,FFFF command to delete all cache records and ensure that the data packet number sent from the server is consistent with that sent from the device. When the GPRS connection mode is UDP, send the CFF command to confirm that the server has received the data.
Example	
GPRS Sending	@@P27,353358017784062,CFF,1*D1\r\n
GPRS Reply	$$P28,353358017784062,CFF,CCE DATA*03\r\n

3.65 Authorizing a RFID Card/iButton Key – D10

GPRS Sending	D10,RFID(1),RFID(2),…,RFID(n)
GPRS Reply	D10,OK
Description	RFID(n): indicates the authorized RFID ID number. The value ranges from 1 to 4294967295. Decimal. A maximum of 50 RFID cards can be authorized at a time.
Example	
GPRS Sending	@@f43,353358017784062,D10,13737431,13737461*17\r\n
GPRS Reply	$$f28,353358017784062,D10,OK*13\r\n

3.66 Authorizing RFID Cards/iButton Keys in Batches – D11

GPRS Sending	D11,*RFID start number*,*n*
GPRS Reply	D11,OK
Description	RFID start number: The value ranges from 1 to 4294967295. Decimal. n: indicates the number of RFID cards to be authorized in batches. Decimal. The parameter value ranges from 1 to 128.
Example	

GPRS Sending	@@e36,353358017784062,D11,13737431,1*AA\r\n
GPRS Reply	$$e28,353358017784062,D11,OK*13\r\n

3.67 Checking RFID/iButton Authorization – D12

GPRS Sending	D12,*RFID/iButton ID*
GPRS Reply	D12,n
Description	RFID ID: The parameter value ranges from 1 to 4294967295. Decimal. n: When **n** is not **0**, the RFID card is authorized. When **n** is **0**, the RFID card is not authorized.
Example	
GPRS Sending	@@C34,353358017784062,D12,13737431*2A\r\n
GPRS Reply	$$C27,353358017784062,D12,0*87\r\n

3.68 Reading an Authorized RFID Card/iButton Key – D13

GPRS Sending	D13,*RFID/iButton packet start number*
GPRS Reply	D13,*Number of RFID packets*,*Current RFID packet number*,RFID(1)RFID(2)…RFID(n)
Description	RFID packet start number: indicates the start sequence number of the RFID packet. The minimum value is **0**. For example, when the value is **0**, you can obtain the package list from the first RFID packet. When the value is **4**, you obtain the package list from the fifth RFID packet. Number of RFID packets: indicates the number of authorized RFID packets. One RFID packet contains a maximum of 100 RFID IDnumbers. The minimum value is **0**. RFID(n): has eight hexadecimal characters.
Example	
GPRS Sending	@@w27,353358017784062,D13,0*F4\r\n
GPRS Reply	The example cannot be displayed because of hexadecimal characters.

3.69 Deleting an Authorized RFID Card/iButton Key – D14

GPRS Sending	D14,RFID(1),RFID(2),…,RFID(n)
GPRS Reply	D14,OK
Description	RFID(n): indicates the RFID ID to be deleted. The value ranges from 1 to 4294967295. Decimal. A maximum of 50 RFID cards can be deleted at a time. One SMS (including the protocol) cannot exceed 140 bytes.
Example	
GPRS Sending	@@Q34,353358017784062,D14,13723455*3B\r\n
GPRS Reply	$$Q28,353358017784062,D14,OK*02\r\n

3.70 Deleting Authorized RFID Cards/iButton Keys in Batches – D15

GPRS Sending	D15,*RFID start number*,*n*
GPRS Reply	D15,OK
Description	RFID start number: The parameter value ranges from 1 to 4294967295. Decimal. n: indicates the number of RFID keys to be deleted in batches. Decimal. The maximum value is **128**. When the start number is a value ranging from 1 to 4294967295 and **n** is greater than or equal to 65536, all authorized numbers will be deleted.
Example	
GPRS Sending	@@K36,353358017784062,D15,13723455,3*97\r\n
GPRS Reply	$$K28,353358017784062,D15,OK*FD\r\n

3.71 Checking the Checksum of the Authorized RFID/iButton ID Database – D16

GPRS Sending	D16
GPRS Reply	D15,XOR
Description	This command is used to check whether the existing authorized RFID ID database is consistent with that recorded in the server. When the device receives the D16 command, the XOR result of all authorized RFID ID numbers is regarded as the database checksum for responding. After the server receives the checksum, compare with the XOR result of all authorized RFID ID numbers recorded in the server. If the result is the same, the existing authorized RFID ID database is consistent with that recorded in the server. Otherwise, data errors occur in the authorized RFID ID database.
Example	
GPRS Sending	@@u25,353358017784062,D16*97\r\n
GPRS Reply	$$u28,353358017784062,D16,18*F7\r\n

3.72 Setting the Maintenance Mileage – D65

GPRS Sending	D65,*Mileage point 1*<,*Mileage point 2*><,*Mileage point 3*><,*Mileage point 4*><,*Mileage point 5*><,*Mileage point 6*><,*Mileage point 7*><,*Mileage point 8*>
GPRS Reply	D65,OK/<*Error code*>
Description	Eight maitanance mileage points will be sent.
Example	
GPRS Sending	
GPRS Reply	

3.73 Setting Maintenance Time – D66

GPRS Sending	D66,*Time point 1*<,*Time point 2*><,*Time point 3*><,*Time point 4*><,*Time point 5*><,*Time*

	point 6><,*Time point 7*><,*Time point 8*>
GPRS Reply	D66,OK/<*Error code*>
Description	Eight maitanance time points will be sent. **Time point**: indicates the days of the next maintenance service after 1st January 1990.
Example (Set the time point. The next maintenance time is 22 November 2013, so the first time point sent is 8726.)	
GPRS Sending	@@V65,353358017784062,D66,8726,8816,8906,8996,9086,9176,9266,9356*A2\r\n
GPRS Reply	$$V28,353358017784062,D66OK*E2\r\n

3.74 Setting Output Triggering – D72

GPRS Sending	D72,X,Y1,Y2,Y3,Y4
GPRS Reply	D72,OK/<*Error code*>
Description	X: Select an output port. **1**: output 1. **2**: output 2. Y1: indicates the output time when an event is triggered. Unit: 10 ms. Value range: 0–4294967295. Y2: The parameter value is **0**, **1**, and **2**. ● **0**: Output high level ● **1**: Output low level ● **2**: Output PWM wave Y3: indicates the PWM duty cycle. Value range: 0–100. Y4: indicates the PWM period. Unit: µs. Value range: 2000–50000000.
Example	
GPRS Sending	@@s42,865328022075252,0D72,1,100,0,0,10000*B0\r\n
GPRS Reply	$$s28,865328022075252,D72,OK*23\r\n

3.75 Allocating GPRS Cache and GPS Log Storage Space – D73

GPRS Sending	D73,X,Y
GPRS Reply	D73,OK/<*Error code*>
Description	X: Set the storage percentage of GPRS cache. The parameter value is a decimal character. Y: Set the storage percentage of GPS logs. The parameter value is a decimal character. The sum of X and Y must be 100.
Example	
GPRS Sending	@@Q32,865328022075252,0D73,50,50*C1\r\n
GPRS Reply	$$Q28,865328022075252,D73,OK*02\r\n

3.76 Setting Harsh Acceleration and Harsh Braking Parameters – D79

GPRS Sending	D79,X,Y
GPRS Reply	D79,OK/<*Error code*>
Description	X: Indicates the harsh acceleration alert value. Decimal; unit: mG; value range:

	[90…1000]; default value: 150. Y: Indicates the harsh braking alert value. Decimal; unit: mG; value range: [-1500…-100]; default value: -180. Harsh acceleration level: Level 1: 150 ● Level 2: 170 ● Level 3: 200 ● Level 4: 230 ● Level 5: 250 ● Level 6: 280 ● Level 7: 300 ● Level 8: 320 ● Level 9: 350 ● Level 10: 400 Harsh braking level: ● Level 1: -180 ● Level 2: -200 ● Level 3: -250 ● Level 4: -300 ● Level 5: -350 ● Level 6: -400 ● Level 7: -450 ● Level 8: -500 ● Level 9: -550 ● Level 10: -600 The higher the level is, the lower the alert probability is. Note: When you install the tracker, the direction and angle of the tracker and vehicle should be consistent. And ensure that the tracker is installed firmly.
Example	
GPRS Sending	@@Q34,865328022075252,D79,150,-180*2B\r\n
GPRS Reply	$$Q28,865328022075252,D79,OK*08\r\n

3.77 Obtaining All Alert Parameters of a Tire Pressure Sensor– DA0

GPRS Sending	DA0
GPRS Reply	DA0,<High pressure threshold of the first axle><Low pressure threshold of the first axle><High pressure threshold of the second axle><Low pressure threshold of the second axle><High pressure threshold of the third axle><Low pressure threshold of the third axle><High pressure threshold of the fourth axle><Low pressure threshold of the fourth axle><High pressure threshold of the trailer><Low pressure threshold of the trailer><High temperature threshold>
Description	High pressure threshold of the first axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the first axle: hexadecimal; unsigned; 1 byte; formula:

	obtained value/10; unit: bar. High pressure threshold of the second axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the second axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the third axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the third axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the fourth axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the fourth axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the trailer: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the trailer: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High temperature threshold: hexadecimal; unsigned; 1 byte; formula: obtained value - 50; unit: °C.
Example	
GPRS Sending	@@Q25,863835020877432,DA0*72\r\n
GPRS Reply	$$Q90,863835020877432,DA0,020800100000000000004576*46\r\n

3.78 Obtaining Data of All Bound Tire Pressure Sensors– DA1

GPRS Sending	DA1
GPRS Reply	DA1,<Location 1><ID1><Tire pressure 1><Temperature 1><Status 1>...<Location n><IDn><Tire pressure n><Temperature n><Status n>
Description	Location: indicates the installation location of a tire pressure sensor; 1 byte; unsigned; hexadecimal. 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: At most 64 tire pressure sensors are supported. In other words, the maximum value of n is 64.
Example	
GPRS Sending	@@Q25,863835020877432,DA1*82\r\n
GPRS Reply	$$Q90,863835020877432,DA1,020800100000000000000711010000000000000006100100*46\r\n

3.79 Obtaining Data of a Tire Pressure Sensor– DA2

GPRS Sending	DA2, Location
GPRS Reply	DA2,< Location ><ID>< Tire pressure >< Temperature >< Status >
Description	Location: indicates the installation location of a tire pressure sensor; 1 byte; unsigned; hexadecimal. 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.
Example	
GPRS Sending	@@g27,863835020877432,DA2,01*C8\r\n
GPRS Reply	$$g35,863835020877432,DA2,020800100000000000000711010000000000000006100100\r\n

3.80 Deleting Tire Pressure Sensors – DA3

GPRS Sending	DA3,< Location 1>….. < Location N>

GPRS Reply	DA3,< Location 1>…..< Location N>,OK
Description	Location: indicates the installation location of a tire pressure sensor; 1 byte; unsigned; hexadecimal. 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. Note: 1. The maximum value of n is 64. 2. If the command is sent successfully, the installation locations of deleted tire pressure sensors will be received.
Example	
GPRS Sending	@@i27,863835020877432,DA3,0A*22\r\n
GPRS Reply	$$i34,863835020877432,DA3,0A,OK*56\r\n

3.81 Obtaining Data of Multiple Tire Pressure Sensors – DA4

GPRS Sending	DA4,< Location 1><ID1>……Location N><IDN>
GPRS Reply	DA4,< Location 1><ID1>……Location N><IDN>,OK
Description	Location: indicates the installation location of a tire pressure sensor; 1 byte; unsigned; hexadecimal. 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. Note:1. At most 64 tire pressure sensors are supported. In other words, the maximum value of n is 64.2. If the command is sent successfully, the installation locations and ID numbers of bound tire pressure sensors will be received.
Example	
GPRS Sending	@@\31,863835020877432,DA4,9800100100*62\r\n
GPRS Reply	$$\59,863835020877432,DA4,0210000000!0100000080010010 0C11000000980010010 0010185R00,OK*A4\r\n

3.82 Setting Alert Thresholds of a Tire Pressure Sensor– DA5

GPRS Sending	DA5,< High pressure threshold of the first axis >< Low pressure threshold of the first axis >< Low pressure threshold of the second axis >< Low pressure threshold of the third axis >< Low pressure threshold of the third axis >< Low pressure threshold of the fourth axis >< Low pressure threshold of the fourth axis >< High pressure threshold of the tow truck >< Low pressure threshold of the tow truck >< High temperature threshold >
GPRS Reply	DA5,OK
Description	High pressure threshold of the first axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the first axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the second axle: hexadecimal; unsigned; 1 byte; formula:

	obtained value/10; unit: bar. Low pressure threshold of the second axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the third axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the third axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the fourth axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the fourth axle: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High pressure threshold of the trailer: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. Low pressure threshold of the trailer: hexadecimal; unsigned; 1 byte; formula: obtained value/10; unit: bar. High temperature threshold: hexadecimal; unsigned; 1 byte; formula: obtained value - 50; unit: °C.
Example	
GPRS Sending	@@l37,863835020877432,DA5,FF0000FFFFFF00000F19d*58\r\n
GPRS Reply	$$l31,863835020877432,DA5,OK*BC\r\n

3.83 Reading Device's Firmware Version and SN – E91

GPRS Sending	E91
GPRS Reply	E91,*Version*,*SN*
Description	Read the device's firmware version and SN.
Example	
GPRS Sending	@@W25,353358017784062,E91*7D\r\n
GPRS Reply	$$W38,353358017784062,MD522S_G4PGW1_H100V44.27412345678*1C\r\n

3.84 Restarting the GSM and GPS Modules – F00

GPRS Sending	F00,GSM,GPS
GPRS Reply	F00,OK/*<Error code>*
Description	GSM: The parameter value is **0** or **1**. **0**: no action. **1**: Restart the GSM module. GPS: The parameter value is **0** or **1**. **0**: no action. **1**: Restart the GPS module.
Example	
GPRS Sending	
GPRS Reply	

3.85 Restarting the GSM Module – F01

GPRS Sending	F01
GPRS Reply	F01,OK
Description	Restart the GSM module.
Example	
GPRS Sending	@@j25,353358017784062,F01*88\r\n
GPRS Reply	$$j28,353358017784062,F01,OK*19\r\n

3.86 Restarting the GPS Module – F02

GPRS Sending	F02
GPRS Reply	F02,OK
Description	Restart the GPS module.
Example	
GPRS Sending	@@Z25,353358017784062,F02*79\r\n
GPRS Reply	$$Z28,353358017784062,F02,OK*0A\r\n

3.87 Setting the Mileage and Run Time – F08

GPRS Sending	F08,*Run time*,*Mileage*
GPRS Reply	F08,OK
Description	Run time: ● Value range: [0…4294967295] ● Decimal ● Unit: second If you do not want to set the parameter, leave it blank. Mileage: ● Value range: [0…4294967295] ● Decimal ● Unit: meter If you do not want to set the parameter, leave it blank.
Example	
GPRS Sending	@@D40,353358017784062,F08,0,4825000*51\r\n
GPRS Reply	$$D28,353358017784062,F08,OK*FA\r\n

3.88 Deleting SMS/GPRS Cache Data – F09

GPRS Sending	F09,*Number*
GPRS Reply	F09,OK
Description	If the number is **1**, SMS cache data to be sent is deleted. If the number is **2**, GPRS cache data to be sent is deleted.

	If the number is **3**, SMS and GPRS cache data to be sent is deleted.
Example	
GPRS Sending	@@E27,353358017784062,F09,1*CA\r\n
GPRS Reply	$$E28,353358017784062,F09,OK*FC\r\n

3.89 Restoring Initial Settings – F11

GPRS Sending	F11
GPRS Reply	F11,OK
Description	Restore initial settings except the SMS password.
Example	
GPRS Sending	@@[25,353358017784062,F11*7A\r\n
GPRS Reply	$$[28,353358017784062,F11,OK*0B\r\n

4 Appendix: Struct Data Analysis

4.1 Transmitting Audio and Video Data in Real Time – A9A

GPRS sending: **A9A,<IP server length><IP address_N><TCP port_2><UDP port_2><Channel number><Data type><Bitrate type>**

MDVR reply: **A9A,OK**

After the MDVR replies to "OK", it will establish a channel with the platform for communication. The data structure is as follows:

<Frame header flag><Load type><Packet number_2><IMEI_8><Channel number><Data type & Data packet processing flag><Timestamp_8><Previous I-frame interval_2><Previous frame interval_2><Data body length_2><Audio and video data_N>

Example:

GPRS sending:

40 40 5C 35 30 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 41 2C 10 73 73 6C 2E 6D 65 69 6C 69 67 61 6F 2E 6F 72 67 69 75 00 00 01 00 01 2A 45 32 0D 0A

The data analysis is as follows:

Instruction prefixes:

40 40 79 34 36 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 41 2C

Converted to ASCII: @@y46,866758042050233,A9A,

IP server length: 10 (16 bytes)

IP address: 73 73 6C 2E 6D 65 69 6C 69 67 61 6F 2E 6F 72 67 (ssl.meiligao.org)

TCP port: 69 75 (26997)

UDP port: 00 00 (0)

Logical channel number: 01 (1)

Data type: 00 (Audio and video)

Bitrate type: 01 (Minor stream)

MDVR reply:

A9A,OK

After the MDVR replies to "OK", an audio and video channel will be established to transmit data. There is so much data that all of them cannot be displayed here. I list the two packets of data as follows:

12 c4 20 01 08 61 10 70 38 77 62 42 01 10 20 20 20 8b 3d 42 86 20 20 20 20 20 03 b6 20 20 20 01 67 42 20 1e 95 a8 2c 04 99 20 20 20 01 68 ce 3c 80 20 20 20 01 06 e5 01 d9 80 20 20 20 01 65 b8 20 20 1b 69 20 d7 df ef c8 bd c3 40 01 ed 33 f8 06 2e 23 0e 02 16 bb 69 05 fb aa ca 06 e0 d9 95 b1 75 bf ee 10 d8 23 41 d2 ff ff dc 68 32 03 40 7c 57 d7 89 70 99 e1 ff fc 9a 08 72 c3 e5 f2 ff 0f 68 9d 49 8b bd 5e af 57 df 7d f3 ef d7 de ef bc f8 4c 66 51 35 fd db b1 7c 13 4e 93 e1 c5 ac 3a 4c 9f f8 0f b4 02 18 18 33 4e fa 15 1b 92 34 28 02 a7 c4 3d ef ff 3f c3 86 06 09 fb 43 f6 36 44 46 8a 6f 3f 78 ff f6 ff f0 86 d7 6f ff c9 df 3e 2f 57 8b bc 49 e7 c1 74 48 57 fb 61 66 f1 23 3a 60 1f 61 ef 98 f6 7a cf 3a 70 0d db 48 d1 7f e3 0f 87 a4 5e 37 79 eb 13 ff 87 50 66 c6 60 30 d0 0c f2 d5 10 8a f7 2f 49 5a 20 76 67 ff ff c5 1c f1 fb fd a8 42 60 80 80 eb 02 62 be bf 7a ff fe 59 c7 d5 4f 2b cd da c9 ed cb ff 42 79 78 f6 0f 0f 6f d2 b2 4b 4a a5 51 76 aa d6 d6 55 95 65 26 96 42 69 69 5c 8b ff ff a1 46 9e 1f 77 5e 4f fa ff 41 e3 bd 29 55 fe 84 1f f6 14 18 56 3a 0e 66 25 8d 12 a8 2e e8 98 e8 c1 a1 3b f7 8a ff a2 4c 6d 13 4b 22 da c7 af ff b7 eb a7 6d 74 e1 7a 0d ad 96 cd f8 4c 64 51 f2 19 94 ea 60 7b bf f4 0a 6f 24 6a ff 6f ed 85 06 19 42 60 99 57 e7 87 cd 26 b5 78 cb 38 60 63 4a c1 90 6c 8b 38 4e 2a b0 99 79 9e d2 f6 f2 ea f1 77 9b 0b a9 71 72 ea f5 2f 2f 26 d2 25 4b cb ff a8 0f 06 70 b1 98 08 e9 99 01 f4 28 28 a6 33 c7 39 48 0b d6 04 7b e8 f7 fc 01 3e 8f c4 e1 e8 48 4d a3 b1 b4 f8 3b 32 84 b0 f0 5e 7d e9 a0 f7 db c5 94 14 fd 18 6b a4 19 dd 3a cf b5 81 1e 93 83 6b 92 ab 69 94 57 f9 a3 6e ea 8c 1e c8 c4 f4 fd f4 e0 55 ee 06 e8 ff a7 98 1c 51 88 ef c0 d4 1e 6b df 47 b6 f1 32 8a 7e 8c 35 d4 19 dd 01 0e d2 ee 62 6e 36 1e f0 50 d4 f4 6e 66 db f0 13 c3 d6 0d 83 91 86 6a af 97 cb fd f0 03 b0 46 70 11 ef 9f 84 cd 34 05 1f 9d cd a1 72 22 f8 75 b3 2f 57 38 69 ad b2 9b ff 87 73 87 62 df f0 04 c7 fc c2 09 b1 ec b9 d8 19 ee 40 0b bf e0 99 ed ad a4 5f c7 26 08 05 f4 7e 3a c3 ff 7e 0a 03 8c e0 26 a7 a0 e1 56 1d c9 7d fc 01 a7 c7 d1 07 51 c7 69 0b 94 56 57 53 2a 0a f4 cf fe 91 48 6c 01 4f f8 03 f6 3a 60 22 c9 d2 20 c5 65 be 1c 29 7a 40 fc 04 c3 3a 7e 45 27 20 1b 86 ff 3c cf a1 f5 ee db a7 17 fa 57 9d fd d0 cd 22 01 6f e0 10 b5 7f d7 f6 ba 4e 02 87 22 4a 94 1e f9 bb db ff 19 01 25 b3 2e 41 eb ff ce 47 20 ed 7f 20 31 b4 b3 1b 3e d3 12 ad 09 82 c3 90 7e 19 42 6d 93 22 42 d8 06 42 d9 9a 13 af 70 66 18 04 8c d3 54 f0 06 98 c6 82 e4 1b ff f8 20 1c 1b f0 3c 19 90 61 d0 46 64 04 59 49 53 c6 2c 92 a9 e1 c5 18 7d b2 0e ae 4c 76 9d 57 81 81 ea f8 ec 59 03 f0 67 1f 83 47 6b 17 6b 90 82 33 20 11 3f d7 6f fd b8 6c 12 0f 79 f9 c7 ac fd 79 01 95 fd f3 7d 6f 90 15 4d 22 e9 77 ef 08 38 20 52 cc 87 5d 2e ef 43 7d 81 1e de 78 3a ee 3c ad d9 81 26 b4 8b 5b cb 86 c7 83 c3 60 08 5b 9b 0b 86 c0 04 8c a6 df 01 b7 58 77 a9 3c 2f 33 4e cf bc fc c0 7e c2 7c 20 c6 d2 d9 93 ed 05 a9 11 1a 94 b7 ad 28 5b 06 08 c0 37 36 85 c9 46 df 1f a7 ff f4 55 94 bf 12 c4 20 02 08 61 10 70 38 77 62 42 01 30 20 20 20 8b 3d 42 86 20 20 20 20 20 03 b6 af e0 83 97 51 7a 7c 3f 20 4f af 9f 95 e5 e3 98 c4 5f cd 0f 80 d2 b6 cd 02 46 2e 1d 9f 5a df 2f f0 99 c0 a3 46 69 37 b2 e9 9f fc cd 3e bb 7b 6d f1 6c 64 33 de 28 e4 f3 bb 1f 3e 19 86 5d e5 3f 51 a2 56 56 11 21 d6 2c 2c a1 89 fd ff fd b2 05 06 60 c6 f3 d1 28 1e 70 45 18 d0 81 22 fb f7 8d 81 18 5d 3e 1a a5 03 b5 63 6d 9d 12 35 33 32 66 dd 29 e6 4c ea 7c 12 a3 42 04 a1 3d c3 c7 fc 27 c3 54 a6 ad 33 e3 4f cb 1a 2e 5c 5c b8 5c ab c4 f8 9c d9 52 a9 30 22 a1 1e 06 d0 f6 7b e5 50 38 4b fe 52 64 25 a2 3b 44 cf 85 59 1f 6f 4f fe 12 18 09 16 87 55 b6 17 76 7f e0 99 98 be cf 52 f9 ff c2 59 24 f1 de ad 37 04 bc fc 20 6d f5 bc dc 55 ff f8 4a 25 df 6d 4f 82 17 fd 01 76 bd 7e f8 07 bf fe b8 7f c3 e7 80 31 1d b4 9f 6b fb 40 7f fc 24 6f 20 2d 0e c1 7a a2 7c b0 0d 9b 96 83 ab f8 87 cb fe 1f 3c 01 5c d3 4c da 9b f9 c2 df fc 24 6f 29 ee 40 17 bf 1e 19 a5 9f 7b 93 f8 1f 86 ff c7 52 f8 68 11 c0 19 f3 eb fe 86 26 bb be dd fe df c4 3c 72 91 c1 d7 32 29 ad d7 83 8a d3 72 0e af f2 a1 68 70 cb b4 3c 98 e0 61 95 c0 bf 41 28 ea e4 d4 83 aa 4d 2b 5a 21 12 fb fc 37 cd 39 77 f6 e3 1b 55 57 ff ba de 99 63 72 ff f2 04 a3 13 53 e1 cf fb a7 b4 1c 7f c0 72 66 9a 8f 35 cb ff b0 97 82 4f 87 64 ff 3d a0 97 ff 86 18 7e 1b 0f 1e 03 20 c9 ad f6 be a7 ae fe 2b 3d 7f e1 e1 8e 7e c6 f6 65 fe 52 f6 1e 1a 5f b2 ae 9f f9 fe dc 81 f3 66 f8 72 bf 3f a8 7c f2 0a 35 1c 1c b0 07 aa cb 61 55 8c 63 f9 05 b5 3f 3c 86 2c

af ff c1 7c 80 83 e0 b3 ba cd 9a 3f 6c 36 2e 04 2d 61 e7 6d 31 09 19 26 9f e2 79 5a cd 87 b7 f2 97 fe 9c 3e 37 a9 2c 20 21 d6 17 88 7b 5e 58 37 66 10 9a cb 8b 97 0b 95 13 8b d4 aa 2e 52 42 e1 65 ff f9 14 83 06 70 10 bc bb 7e ab 87 1e 68 7c c5 b6 34 e7 a6 3a f2 5f 45 26 53 92 78 b2 d1 99 b9 95 c6 68 9f f8 55 9a ed c1 e5 4c 9a 39 0b 1a 8f c0 ed 72 ef ff d6 3d 71 7e bc 99 8e 55 26 ba 3e 17 ff fd 90 84 1a 6d 2f 59 b0 84 a6 9d fc 03 fe 58 bd fc bc 01 ed 51 f7 8b 05 f8 8f d4 ae 95 ca ae d4 43 c2 a4 89 07 3e 0b f3 7f 87 fe f0 81 06 4a 1f 07 06 23 ae da 5f fe 42 0d 6f 87 e1 ad 68 2b b0 96 5c e3 94 14 62 63 bf af f2 3c bd 75 d6 2e 57 ae 47 ff ff e4 ff 1f f4 4f 13 0f fd 04 86 58 73 04 ac cc 0a 5f b5 4f 5f e2 04 67 ba 0d 7d 06 dc ef df 32 27 c2 0e 3b 9f f5 f9 3b 82 26 29 4f 8a 70 2b c3 ac 11 97 3a 52 47 4c 11 69 6b 17 2e 2e 27 12 72 a5 24 4f 0a 43 05 ca ff ff 29 19 ba af 5d 60 40 b6 16 f4 19 72 73 91 74 4f ac db c1 25 6c bc 07 df 77 71 5c 89 4f ad 9a 61 ae 70 0e f1 1a 52 cc 2b b7 ec 81 66 aa 5d 07 ba b3 61 4a 2f 34 b3 a7 ff eb ef d7 28 ef e2 47 f7 73 4d bf ca d5 df 4b 0c f0 96 cd 4e d6 7c 78 f0 4b 4d 7b 89 9e f5 b5 27 80 bd 9f 09 6d f0 69 40 7d 6f 54 87 db ae a4 16 76 43 03 66 95 79 fa 4c 16 f9 f2 ff fe 7d 64 0a 9d 7a 55 ca 4c ae 5e 42 46 90 ff ff e8 10 ee ed 84 1d aa 14 e0 47 ff fc 8f 5d 75 cb c9 9b 5f 2a a9 7e bf ff c1 3d af 55 a5 29 8a 7b eb 55 c8 f2 2c af 22 c8 e1 55 4b bb df f0 9f fc 28 65 55 24 8c d0 ff 5a bf a9 c2 a2 a1 17 09 38 70 7f fe 55 08 12 c4 20 03 08 61 10 70 38 77 62 42

According to the previous data, **12 c4** is a data packet header of the frame. With the previous data, you can analyze all data. **Attention, in order to parse data above, users should follow H.264 video standard, H.265 video standard and G.726 audio standard.**

4.2 Controlling Real-Time Audio and Video Transmission – A9B

GPRS sending: A9B, <Logical channel number><Control command><Whether to disable audio and video><Switch the bitrate type>
MDVR reply: A9B,OK

Example:
GPRS sending:
40 40 77 33 30 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 42 2C 01 01 00 00 2A 42 44 0D 0A

The data analysis is as follows:
Instruction prefixes:
40 40 77 33 30 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 42 2C
Converted to ASCII: @@w30,866758042050233,A9B,

Logical channel number: 01
Control command: 01 (Switch the bitrate)
Whether to disable audio and video: 00 (Disable related audio and video data in this channel)
Switch the bitrate type: 00 (Major stream)

MDVR reply:
24 24 77 32 38 2c 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2c 41 39 42 2c 4f 4b 2a 33 38 0d 0a
$$w28,866758042050233,A9B,OK*38

4.3 Querying the Resource List – A9C

GPRS sending:

<Channel number><Start time_6><End time_6><Reserved flag_8><Audio and video type><Bitrate type><Memory type><Number of alerts_2><Alert event 1,Alert event 2,...,Alert event N>

MDVR reply:

<Number of audio and video resources_4><Audio and video file content 1><Audio and video file content 2><Audio and video file content 3>...<Audio and video file content N>

Data structure of an audio and video file:

<Channel number><Start time_6><End time_6><Alert flag_8><Audio and video type><Bitrate type><Memory type><File size_4>

Example:

(1) GPRS command from platform to MDVR:

40 40 41 35 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C 01 19 07 24 00 00 00 19 07 24 23 59 59 00 00 00 00 00 00 00 00 00 00 01 01 00 01 00 2A 30 45 0D 0A

(This command indicates to search for all video+audio files of Channel 1, alert event code 1 from 2019/07/24 00:00:00 to 2019/07/24 23:59:59)

The data analysis is as follows:

Instruction prefixes:

40 40 41 35 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C

Converted to ASCII: @@A54, 861585040494468, A9C,

Channel number: 01

Start time: 19 07 24 00 00 00 (24th July 2019 00:00:00)

End time: 19 07 24 23 59 59 (24th July 2019 23:59:59)

Reserved flag: 00 00 00 00 00 00 00 00(These 8 bytes are useless, all could be filled by 00)

Audio and video type: 00

Bitrate type: 00

Memory type: 01

Number of alerts: 01 00

Alert event: 01 00 (Indicates to search for all video+audio files of alert event code 1)

Reply:

24 24 41 35 38 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C 00 00 00 01 01 19 07 24 02 42 07 19 07 24 02 43 11 00 00 00 00 00 00 00 01 00 01 01 00 E5 A4 00 2A 33 30 0D 0A

The data analysis is as follows:

Instruction prefixes:

24 24 41 35 38 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C

Converted to ASCII: $$A58, 861585040494468, A9C,

Number of audio and video resources: 00 00 00 01 (indicate 1 video+audio file)
Audio and video file content 1: 01 19 07 24 02 42 07 19 07 24 02 43 11 00 00 00 00 00 00 00 01 00 01 01 00 E5 A4 00 (indicate the video+audio file of channel 1, alert event code 1, size-15049728 bytes from 24th July 2019 02:42:07 to 24th July 2019 02:43:11)

Channel number: 01
Start time：19 07 24 02 42 07
End time：19 07 24 02 43 11
Alarm event code：00 00 00 00 00 00 00 01 (indicates the video+audio files of alarm event code 1)
Audio and video type：00
Bitrate type：01
Memory type：01
File size：00 E5 A4 00(indicate 15049728 bytes)

（2）GPRS command from platform to MDVR:
40 40 44 35 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C 01 19 07 24 00 00 00 19 07 24 02 30 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 2A 36 43 0D 0A
(This command indicates to search for all video+audio files (including alerts files and normal files) of channel 1 from 24th July 2019 00:00:00 to 24th July 2019 02:30:00)

The data analysis is as follows:
Instruction prefixes:
40 40 44 35 34 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C
Converted to ASCII: @@D54,861585040494468,A9C,
Channel number：01
Start time: 19 07 24 00 00 00（24th July 2019 00:00:00）
End time: 19 07 24 02 30 00（24th July 2019 02:30:00）
Reserved flag: 00 00 00 00 00 00 00 00（These 8 bytes are useless, all could be filled by 00）
Audio and video type：00
Bitrate type：00
Memory type：01
Number of alerts：00 00（Indicates to search for all video+audio files(including alerts videos and normal videos) within this period）
Alert event：00 00（Indicates to search for all video+audio files(including alerts videos and normal videos) within this period）

Reply：
24 24 44 34 32 32 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 43 2C 00 00 00 0E 01 19 07 24 01 18 11 19 07 24 01 25 00 00 00 00 00 00 00 00 00 00 01 01 05 7B 04 00 01 19 07 24 01 25 00 19 07 24 01 30 00 00 00 00 00 00 00 00 00 01 01 04 02 18 00 01 19 07 24 01 30 00 19 07 24 01 35 00 00 00 00 00 00 00 00 00 00 01 01 04 01 CC 00 01 19 07 24 01 35 00 19 07 24 01 40 00 00 00 00 00 00 00 00 00 00 01 01 04 02 1C 00 01 19 07 24 01 40 00 19 07 24 01 45 00 00 00 00 00 00 00 00 00 00 01 01 03 FC D8 00 01 19 07 24 01 45 00 19 07 24 01 50 00 00 00 00 00 00 00

00 00 00 01 01 03 FC B8 00 01 19 07 24 01 50 00 19 07 24 01 55 00 00 00 00 00 00 00 00 00 01 01 03 FC C8 00 01 19 07 24 01 55 00 19 07 24 02 00 00 00 00 00 00 00 00 00 00 01 01 03 FD 20 00 01 19 07 24 02 00 00 19 07 24 02 05 00 00 00 00 00 00 00 00 00 00 01 01 03 FB C0 00 01 19 07 24 02 05 00 19 07 24 02 10 00 00 00 00 00 00 00 00 00 00 01 01 03 FE 5C 00 01 19 07 24 02 10 00 19 07 24 02 15 00 00 00 00 00 00 00 00 00 01 01 04 00 48 00 01 19 07 24 02 15 00 19 07 24 02 20 00 00 00 00 00 00 00 00 00 01 01 04 01 68 00 01 19 07 24 02 20 00 19 07 24 02 25 00 00 00 00 00 00 00 00 00 01 01 04 00 E8 00 01 19 07 24 02 25 00 19 07 24 02 30 00 00 00 00 00 00 00 00 00 01 01 04 03 08 00 2A 46 30 0D 0A

Number of audio and video resources：00 00 00 0E （indicate 14 video+audio file）

Audio and video file content 1：01 19 07 24 01 18 11 19 07 24 01 25 00 00 00 00 00 00 00 00 00 00 01 01 05 7B 04 00 (indicate the normal video+audio file of channel 1, normal video type, size-91948032 bytes from 24th July 2019 02:42:07 to 24th July 2019 02:43:11)

Audio and video file content 2：01 19 07 24 01 25 00 19 07 24 01 30 00 00 00 00 00 00 00 00 00 00 01 01 04 02 18 00（normal video+audio file）

Audio and video file content 3：01 19 07 24 01 30 00 19 07 24 01 35 00 00 00 00 00 00 00 00 00 00 01 01 04 01 CC 00（normal video+audio file）

Audio and video file content 4：01 19 07 24 01 35 00 19 07 24 01 40 00 00 00 00 00 00 00 00 00 00 01 01 04 02 1C 00（normal video+audio file）

Audio and video file content 5：01 19 07 24 01 40 00 19 07 24 01 45 00 00 00 00 00 00 00 00 00 00 01 01 03 FC D8 00（normal video+audio file）

Audio and video file content 6：01 19 07 24 01 45 00 19 07 24 01 50 00 00 00 00 00 00 00 00 00 00 01 01 03 FC B8 00（normal video+audio file）

Audio and video file content 7：01 19 07 24 01 50 00 19 07 24 01 55 00 00 00 00 00 00 00 00 00 00 01 01 03 FC C8 00（normal video+audio file）

Audio and video file content 8：01 19 07 24 01 55 00 19 07 24 02 00 00 00 00 00 00 00 00 00 00 01 01 03 FD 20 00（normal video+audio file）

Audio and video file content 9：01 19 07 24 02 00 00 19 07 24 02 05 00 00 00 00 00 00 00 00 00 00 01 01 03 FB C0 00（normal video+audio file）

Audio and video file content 10：01 19 07 24 02 05 00 19 07 24 02 10 00 00 00 00 00 00 00 00 00 00 01 01 03 FE 5C 00（normal video+audio file）

Audio and video file content 11：01 19 07 24 02 10 00 19 07 24 02 15 00 00 00 00 00 00 00 00 00 01 01 04 00 48 00（normal video+audio file）

Audio and video file content 12：01 19 07 24 02 15 00 19 07 24 02 20 00 00 00 00 00 00 00 00 00 01 01 04 01 68 00（normal video+audio file）

Audio and video file content 13：01 19 07 24 02 20 00 19 07 24 02 25 00 00 00 00 00 00 00 00 00 01 01 04 00 E8 00（normal video+audio file）

Audio and video file content 14：01 19 07 24 02 25 00 19 07 24 02 30 00 00 00 00 00 00 00 00 00 01 01 04 03 08 00（normal video+audio file）

4.4 Playing Back Videos Remotely – A9D

GPRS sending:

<IP address length><IP address_N><TCP port_2><UDP port_2><Channel number><Audio and video resource

type><Bitrate type><Memory type><Playback mode><Fast-forward or fast-rewind times><Start time_6><End time_6>

MDVR reply: A9D,OK

After the MDVR replies to "OK", it will establish a channel with the platform for communication. The data structure is the same as that of the A9A command and as follows:

<Frame header flag><Load type><Packet number><IMEI><Channel number><Data type & Data packet processing flag><Timestamp><Previous I-frame interval><Previous frame interval><Data body length><Audio and video data>

Example:

GPRS sending:

40 40 49 36 35 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 44 2C 10 73 73 6C 2E 6D 65 69 6C 69 67 61 6F 2E 6F 72 67 69 74 00 00 01 03 00 00 00 00 20 04 28 08 34 35 20 04 28 08 39 35 2A 35 38 0D 0A

The data analysis is as follows:

Instruction prefixes: 40 40 56 36 31 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 44 2C

Converted to ASCII: @@V61,866758042050233,A9D,

IP address length: 10 (16 bytes)

IP address_16: 73 73 6C 2E 6D 65 69 6C 69 67 61 6F 2E 6F 72 67(ssl.meiligao.org)

WORD tcp_port: 69 74 (26996)

WORD udp_port: 00 00 (0)

BYTE logiChn: 01 (Channel 1)

BYTE avType: 03 (Video, or audio and video)

BYTE streamType: 00 (Major stream or minor stream)

BYTE capType: 00 (All memories)

BYTE reviewStyle: 00 (Normal playback)

BYTE viewRank: 00 (Invalid)

Start time: 20 04 28 08 34 35 (April 28, 2020 08:34:35)

End time: 20 04 28 08 39 35 (April 28, 2020 08:39:35)

Reply:

40 40 49 36 35 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 39 44 2C 4F 4B 2A 34 35 0D 0A

4.5 Controlling Remote Video Playback – A9E

GPRS sending:

A9E,<Channel number><Playback control><Fast-forward or fast-rewind times><Playback time point to be dragged: YYMMDDHHMMSS_6>

MDVR reply: A9E,OK

After the MDVR replies to "OK", it will establish a channel with the platform for communication. The data structure is the same as that of the A9A command and as follows:

<Frame header flag><Load type><Packet number_2><IMEI_8><Channel number><Data type & Data packet processing flag><Timestamp_8><Previous I-frame interval_2><Previous frame interval_2><Data body length_2><Audio and video data_N>

Example:

Sending:

40 40 43 33 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 45 2C 01 01 00 19 07 24 01 32 00 2A 32 39 0D 0A

Channel: 01

Playback control: 01

Fast-forward or fast-rewind times: 00

Playback time point to be dragged: 19 07 24 01 32 00(Indicates drag it to playback time on 20190724 at 01:32:00)

Reply:

24 24 43 32 38 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 45 2C 4F 4B 2A 31 34 0D 0A

4.6 Uploading Files – A9F

GPRS sending:

A9F,**<FTP_IP address length><FTP_IP address_N><FTP_Port_2><User name length><User name><Password length><Password><File uploading path length><File uploading path><Channel number><Start time_6><End time_6><Alert flag_8><Audio and video type><Bitrate type><Storage location><Task execution condition><Number of alerts_2><Alert event code list>**

MDVR reply:

A9F,**<Flag><Number of audio and video resources (N)_4><Uploaded file info 1><Uploaded file info 2>...<Uploaded file info N>**

The struct of the uploaded file info is as follows:

<Channel number><Start time_6><End time_6><Alert event code_8><Audio and video resource type><Bitrate type><Memory type><File size_4>

Example:

（1）GPRS sending：

40 40 47 39 38 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 0C 36 37 2E 32 30 33 2E 31 33 2E 34 33 26 94 07 44 56 52 44 65 6D 6F 06 30 30 30 30 30 30 10 30 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 01 19 07 24 01 30 00 19 07 24 01 45 00 FF FF FF FF FF FF FF FF 00 00 01 01 2A 43 42 0D 0A

（**This command indicates it should upload all video+audio files from July 24th 2019 01:30:00 to July 24th 01:45:00, to FTP server whose IP address 67.203.13.43, Port 9876, Username DVRDemo, Password 000000, based on channel 1(including all normal video files and alerts video files)**）

Instruction prefixes：@@G98,861585040494468,A9F,

FTP_IP address length：0C （FTP IP address length is 12 bytes）

FTP_IP address length：36 37 2E 32 30 33 2E 31 33 2E 34 33（FTP IP is 67.203.13.43）

FTP_Port_2：26 94（Port is 9876）

Username legnth：07

Username：44 56 52 44 65 6D 6F （DVRDemo）

Password length：06

Password：30 30 30 30 30 30（000000）

File uploading path length：10（Length is 16 bytes）

File uploading path：30 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38（Path is /0861585040494468）

Channel：01

Start time_6：19 07 24 01 30 00（The uploading video+audio file starts on 24th July 2019 01:30:00）

End time_6：19 07 24 01 45 00（The uploading video+audio file ends on 24th July 2019 01:45:00）

Reserved_flag_8：FF FF FF FF FF FF FF FF （They are useless so far, all 8 could be filled by FF）

Audio and Video type：00

Bitrate type：00

Memory type：01

Task execution condition：01 （Bit 0: WiFi. 1: Upload by WiFi； Bit 1: LAN. 1: Upload when a LAN network is connected； Bit2: 3G/4G. 1: Upload when a 3G or 4G network is connected; 01 indicates only under WIFI can it upload files, 02 indicates only under LAN network can it upload files, 03 indicates only under WIFI or LAN network can it upload, 04 indicates only under GSM can it upload files, 05 indicates only under WIFI or 3G/4G network can it upload. 06 indicates only under LAN or 3G/4G can it upload, 07 indicates it could upload files under WIFI, LAN or 3G/4G. Please pay attention to this condition, otherwises, the uploading task will fail.）

Number of alerts_2：If there is no content here, it indicates uploading all normal video+audio files and alerts video+audio files.

Alerts event code list：If there is no content here, it indicates uploading all normal video+audio files and alerts video+audio files.

Instruction prefixes：2A 43 42 0D 0A（*CB）

Reply：

24 24 47 31 31 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 01 03 00 00 00 01 19 07 24 01 30 00 19 07 24 01 35 00 00 00 00 00 00 00 00 00 00 01 01 04 01 CC 00 01 19 07 24 01 35 00 19 07 24 01 40 00 00 00 00 00 00 00 00 00 00 00 01 01 04 02 1C 00 01 19 07 24 01 40 00 19 07 24 01 45 00 00 00 00 00 00 00 00 00 00 01 01 03 FC D8 00 2A 38 30 0D 0A

Instruction prefixes：$$G115,861585040494468,A9F,

Confirming_flag：01（1 indicates succeed,0 indicates fail）

Number of audio and video resources N_4：03 00 00 00（indicates there are totally 3 files）

Uploaded file info 1：01 19 07 24 01 30 00 19 07 24 01 35 00 00 00 00 00 00 00 00 00 00 01 01 04 01 CC 00

（This command indicates the 1st video+audio file to be uploaded was taken as a normal video+audio file from July 24th 2019 01:30:00 to July 24th 01:35:00, based on channel 1, with file size 67226624 bytes and file name CH1_20190724013000_20190724013500_0_0_1_1_NOR.avmsg）

Uploaded file info 2：01 19 07 24 01 35 00 19 07 24 01 40 00 00 00 00 00 00 00 00 00 00 01 01 04 02 1C 00

（File name is CH1_20190724013500_20190724014000_0_0_1_1_NOR.avmsg）

Uploaded file info 3：01 19 07 24 01 40 00 19 07 24 01 45 00 00 00 00 00 00 00 00 00 00 01 01 03 FC D8 00
（File name is CH1_20190724014000_20190724014500_0_0_1_1_NOR.avmsg）

Instruction suffix：2A 38 30 0D 0A（*80）

（2）GPRS sending：
40 40 4E 31 30 32 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 0C 36 37 2E 32 30 33 2E 31 33 2E 34 33 26 94 07 44 56 52 44 65 6D 6F 06 30 30 30 30 30 30 10 30 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 01 19 07 24 00 00 00 19 07 24 23 59 59 FF FF FF FF FF FF FF FF 00 00 01 01 01 00 01 00 2A 35 34 0D 0A
（**This command indicates it should upload all video+audio files of alert event code 1 from July 24th 2019 00:00:00 to July 24th 23:59:59, to FTP server whose IP address 67.203.13.43, Port 9876, Username DVRDemo, Password 000000, based on channel 1**）

Reply：
24 24 49 38 37 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 01 02 00 00 00 01 19 07 24 02 42 07 19 07 24 02 43 11 00 00 00 00 00 00 00 01 00 01 01 00 E5 A4 00 01 19 07 24 03 22 24 19 07 24 03 23 24 00 00 00 00 00 00 00 01 00 01 01 00 DA A8 00 2A 45 30 0D 0A

Number of audio and video resources N_4：02 00 00 00（There are totally 2 video+audio files）

Uploaded file info 1：01 19 07 24 02 42 07 19 07 24 02 43 11 00 00 00 00 00 00 00 01 00 01 01 00 E5 A4 00
（Alert-1 video+audio file whose filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

Uploaded file info 2：01 19 07 24 03 22 24 19 07 24 03 23 24 00 00 00 00 00 00 00 01 00 01 01
（Alert-1 video+audio file whose filename is CH1_20190724032224_20190724032324_1_0_1_1_ALM.avmsg）

(3) GPRS sending：
40 40 55 32 30 30 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 0C 36 37 2E 32 30 33 2E 31 33 2E 34 33 26 94 07 44 56 52 44 65 6D 6F 06 30 30 30 30 30 30 10 30 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 01 19 07 24 00 00 00 19 07 24 23 59 59 FF FF FF FF FF FF FF FF 00 00 01 01 32 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A 00 0B 00 0C 00 0D 00 0E 00 0F 00 10 00 12 00 13 00 14 00 15 00 16 00 17 00 19 00 20 00 21 00 22 00 24 00 25 00 29 00 2A 00 32 00 33 00 34 00 35 00 36 00 52 00 53 00 5E 00 5F 00 60 00 63 00 64 00 65 00 81 00 82 00 40 02 41 02 42 02 43 02 60 02 2A 30 41 0D 0A
（**This command indicates it should upload all video+audio files of all alert events from July 24th 2019 00:00:00 to July 24th 23:59:59, to FTP server whose IP address 67.203.13.43, Port 9876, Username DVRDemo, Password 000000, based on channel 1**）

Number of alerts：32 00（indicate there are totally 50 alerts events）
All alerts video+audio files：
01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 0A 00 0B 00 0C 00 0D 00 0E 00 0F 00 10 00 12 00 13 00 14 00 15 00 16 00 17 00 19 00 20 00 21 00 22 00 24 00 25 00 29 00 2A 00 32 00 33 00 34 00 35 00 36 00 52 00 53 00 5E 00 5F 00 60 00 63 00 64 00 65 00 81 00 82 00 40 02 41 02 42 02 43 02 60 02（indicate the alert code list to be checked

is 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 27 28 29 31 32 33 34 35 36 37 41 42 50 51 52 53 54 82 83 94 95 96 99 100 101 129 130 576 577 578 579 608）

Reply：

24 24 55 38 37 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 39 46 2C 01 02 00 00 00 01 19 07 24 02 42 07 19 07 24 02 43 11 00 00 00 00 00 00 00 01 00 01 01 00 E5 A4 00 01 19 07 24 03 22 24 19 07 24 03 23 24 00 00 00 00 00 00 00 01 00 01 01 00 DA A8 00 2A 45 43 0D 0A

4.7 Controlling File Uploading – AA0

GPRS sending:

AA0,**<Uploading control_Pause/Continue/Cancel><Name of the File to be controlled_N>**

MDVR reply:

AA0,OK

Example:

Sending:

40 40 53 37 38 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 41 30 2C 02 43 48 33 5F 32 30 32 30 30 34 32 38 30 38 33 34 33 35 5F 32 30 32 30 30 34 32 38 30 38 33 39 33 35 5F 39 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 2A 38 45 0D 0A

Flag:02 (Cancel)

Reply:

FileName[128]: 43 48 33 5F 32 30 32 30 30 34 32 38 30 38 33 34 33 35 5F 32 30 32 30 30 34 32 38 30 38 33 39 33 35 5F 39 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67

FileName :CH3_20200428083435_20200428083935_9_0_1_1_ALM.avmsg

4.8 Obtaining the WiFi List – AA1

GPRS sending:

AA1

MDVR reply:

AA1,**<Number of WiFi><SSID1 content,SSID2 content,SSID3 content,...,SSIDn content>**

The SSID content structure is as follows:

<SSID format><SSID length><SSID_N><SSID signal value>

Example:

Sending:

40 40 71 32 35 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 41 31 2A 39 38 0D 0A

Reply:

24 24 71 32 38 34 2c 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2c 41 41 31 2c 12 01 05 4d 44 56 52 00 60

01 06 63 65 73 68 69 00 4a 01 0e 43 6f 6e 6e 65 63 74 69 66 79 2d 6d 65 00 4c 01 07 50 74 57 69 66 69 00 4c 01 04 38 38 38 00 46 01 08 57 59 48 35 32 38 41 00 44 01 1d 73 68 65 6e 7a 68 65 6e 73 68 69 6e 61 6e 67 75 69 6d 61 6f 79 69 67 6f 6e 67 73 69 00 46 01 0f 4d 65 69 74 72 61 63 6b 5f 59 61 6e 46 61 00 4a 01 06 4d 44 56 52 31 00 3e 01 0c 4d 65 69 74 72 61 63 6b 5f 49 54 00 3d 01 0e 43 68 69 6e 61 4e 65 74 2d 52 77 4b 35 00 3e 01 04 49 54 32 00 30 01 0e 43 68 69 6e 61 4e 65 74 2d 41 35 56 69 00 2e 01 0e 54 6f 70 77 61 79 5f 42 39 44 34 46 31 00 2a 01 04 6c 65 79 00 1a 01 0e 43 68 69 6e 61 4e 65 74 2d 41 37 68 73 00 1a 01 09 48 6f 6e 6f 72 20 31 30 00 1a 01 18 46 6f 75 72 20 50 6f 69 6e 74 73 20 62 79 20 53 68 65 72 61 74 6f 6e 00 2a 2a 41 45 0d 0a

4.9 Sending the FTP File Uploading Progress – AA2

Data struct uploaded by MDVR:

AA2,**<Uploading progress percentage>,<File name_128>**

Example:

Sending:

24 24 61 31 35 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 41 32 2C **01** 43 48 31 5F 32 30 31 39 30 37 32 34 30 32 34 32 30 37 5F 32 30 31 39 30 37 32 34 30 32 34 33 31 31 5F 31 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2A 38 39 0D 0A

（Uploading process is 1%, filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

24 24 61 31 35 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 41 32 2C **05** 43 48 31 5F 32 30 31 39 30 37 32 34 30 32 34 32 30 37 5F 32 30 31 39 30 37 32 34 30 32 34 33 31 31 5F 31 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2A 38 44 0D 0A

（Uploading process is 5%, filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

24 24 61 31 35 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 41 32 2C **0A** 43 48 31 5F 32 30 31 39 30 37 32 34 30 32 34 32 30 37 5F 32 30 31 39 30 37 32 34 30 32 34 33 31 31 5F 31 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2A 39 32 0D 0A

（Uploading process is 10%, filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

24 24 61 31 35 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 41 32 2C **11** 43 48 31 5F 32 30 31 39 30 37 32 34 30 32 34 32 30 37 5F 32 30 31 39 30 37 32 34 30 32 34 33 31 31 5F 31 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2A 39 39 0D 0A

（Uploading process is 17%, filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

24 24 61 31 35 35 2C 38 36 31 35 38 35 30 34 30 34 39 34 34 36 38 2C 41 41 32 2C **18** 43 48 31 5F 32 30 31 39 30 37

32 34 30 32 34 32 30 37 5F 32 30 31 39 30 37 32 34 30 32 34 33 31 31 5F 31 5F 30 5F 31 5F 31 5F 41 4C 4D 2E 61 76 6D 73 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2A 41 30 0D 0A

（Uploading process is 25%, filename is CH1_20190724024207_20190724024311_1_0_1_1_ALM.avmsg）

4.10 Obtaining MDVR Network Status – AA3

GPRS sending:

AA3,<Current network><GSM status><SIM card ready or not><SIM card number_16><SIM card IMSI_16><GSM network type><GSM signal value><GSM_IMEI_16><GSM status><WiFi status><WiFi mode><WiFi_SSID_128><WiFi signal value><WiFi IP address_15><WiFi MAC address_6><Subnet mask settings of WiFi_15><Gateway settings of WiFi_15><Active DNS server settings of WiFi_15><Standby DNS server settings of WiFi_15><LAN status><LAN_IP_15><LAN_MAC address_6><Subnet mask settings of LAN_15><Gateway settings of LAN_15><Active DNS server settings of LAN_15><Standby DNS server settings of LAN_15>

Example:

Sending:

40 40 64 32 35 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 41 33 2A 38 44 0D 0A

Reply:

24 24 48 33 37 34 2c 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2c 41 41 33 2c 03 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 01 31 39 32 2e 31 36 38 2e 33 2e 31 37 37 00 00 61 61 3a 62 65 3a
32 35 35 2e 32 35 35 2e 32 35 35 2e 30 00 00 31 39 32 2e 31 36 38 2e 33 2e 31 00 00 00 00 32 32 33 2e 35
2e 35 2e 35 00 00 00 00 00 00 32 32 33 2e 36 2e 36 2e 36 00 00 00 00 00 00 2a 36 45 0d 0a

4.11 Querying which days' video files have been stored – AA4

GPRS sending: AA4[,YYMM]

MDVR reply: AA3,< The BCD code for year and month >< Which day of the month video files were saved >< Which day of the month alerts video files were saved >

Example:

Sending:

40 40 41 32 38 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 41 34 2C 20 04 2A 42 45 0D 0A

The data analysis is as follows:

YYMM：20 04

Reply:

24 24 41 33 36 2C 38 36 36 37 35 38 30 34 32 30 35 30 32 33 33 2C 41 41 34 2C 20 04 00 00 00 3F 00 00 00 1C 2A 45 30 0D 0A

The data analysis is as follows:

YYMM[2]：20 04 (There are video files saved in April 2020)

mediaRecFlag：00 00 00 3F (The video files were saved on April 25/26/27/28/29/30)

alarmRecFlag：00 00 00 1C (The alerts video files were saved on April 27/28/29)

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