MEITRACK T633Li Protocol

Applicable Model: T633Li

Change History

File Name	MEITRACK T633Li Protocol		
Project	T633Li	Creation Date Update Date	2025-11-20
Subproject	GPRS Protocol	Total Pages	28
Version	V1.2	Confidential	Internal Documentation

Copyright and Disclaimer

Document update history

Version	date	Revise
1.0	2025-11-20	First draft.
1.1	2026-02-03	Add Iridium Satellite Communication Protocol Add AD1 and AD2
1.2	2026-03-12	Modify Iridium Server Data Procotol

Contents

1 Iridium Satellite Communication Protocol

1) Iridium satellite command pack

Iridium satellite instruction package transmitted from the server to the terminal:：

@@<packet identity ID ><length of data >,<IMEI>,<type of command >,<content of command ><*check code >\r\n

Iridium satellite instruction package replied by the terminal to the server：

$$< packet identity ID >< length of data >,<IMEI>,< type of command >,< content of command ><* check code >\r\n

That is, the Iridium satellite instruction package is the same as the GPRS instruction package. For details of the instruction package protocol, please refer to the Meitrack protocol document GPRS instruction package；

2) GSM data

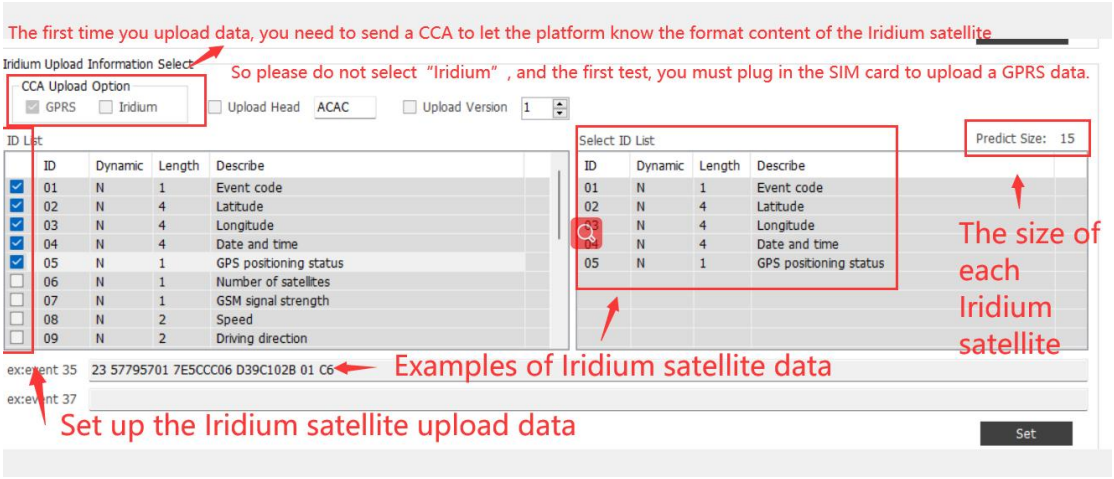
When using GSM to upload data, the standard CCE protocol will be used.

3) Iridium data（CCA）

Iridium satellite upload data uses the hexadecimal data format defined by ID, and the data packet format is as follows

Simplified (2Bytes) + protocol version(1byte) +[Static upload area] + [Dynamic upload area]+CRC(1Byte)；

Static upload area:



Dynamic upload area：

Purpose: Some CCE IDs are not uploaded for every packet of data. They are usually uploaded for special events, such as RFID;

Make up：[ID total（1byte）]+[ID number of one byte]+IDn+contentn+[ID number of two bytes]+IDn+contentn+[ID number of four bytes]+IDn+contentn+[ID number of non-defined length bytes]+IDn+contentn

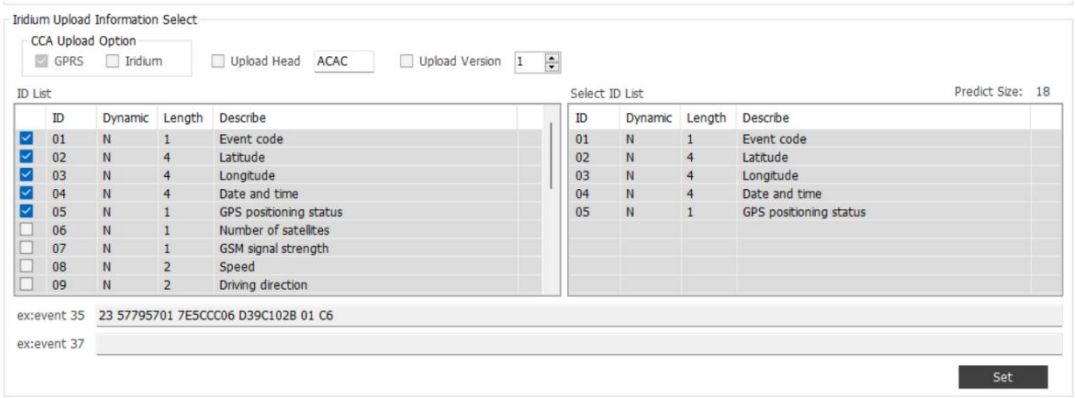
NO	Data matters	length	notice

1	header	2bytes	You can configure whether to upload headers and customize header characters through the MM and platform;
2	Protocol Version	1byte	The protocol version number can be customized. The default version is 0x00; You can use the MM app and platform to configure whether to upload the protocol version;
3	IDn Content		For the data corresponding to the CCE ID, refer to the MEITRACK_CCE_ID_def.xlsx;
4	CRC	1byte	Checksum of data other than CRC;

Dynamic data is currently only RFID events. (**If you do not use RFID accessories, you can not consider.**)

- The required upload data can be checked through MM app or the platform, and only upload "<protocol version><event code><date and time><the status of location><latitude><Longitude>" is checked by default.。

And the MM app or platform configuration window needs to display the CCE ID, the length of CCE ID and the total length of all data corresponding to the checked options in order, and give examples;



4) Iridium satellite packet communication mechanism：(CCA)

According to needs, through MM or the platform, you can configure whether the terminal needs to report the iridium satellite packet format to the platform, so that the platform can automatically parse different data protocol formats, and the reported data parsing packet format is in accordance with the GPRS instruction package format of meitrack, and the instruction CCA is added for uploading the iridium satellite packet parsing format;

$$<The packet identity ID><the length of data>,<IMEI>,<CCA>,<Iridium satellite packet parsing format><*check code>\r\n

Iridium satellite packet parsing format： header(1byte)+header content(2bytes)+ the mark of the protocol version (1byte)+protocol version(1byte)+[CCE ID1+CCE ID1length]+...+[CCE IDn+CCE IDn length];

NO	Data matters	length	Remarks (no special instructions, all are decimal data formats)
1	The mark of the header	1byte	When this value is 0x00, it means that the iridium satellite packet does not have a header； When this value is 0x00, it means that the iridium satellite packet does have a header；
2	Header content	2bytes	For representing the header contents of an iridium satellite packet； Useful only when the header flag is 0x01；
3	the mark of the protocol version	1byte	When this value is 0x00, it means that the iridium satellite packet does not have protocol version； When this value is 0x00, it means that the iridium satellite packet does have protocol version；
4	Protocol version	1byte	For representing the current iridium satellite packet protocol version number； The default includes the protocol version number, and its number is 0x00, which can be customized； Useful only when the mark of the protocol version is 0x01；
5	CCE ID	2bytes	CCE ID definition refer to《MEITRACK_CCE_ID_def.xlsx》;
6	CCE ID length	2bytes	CCE ID length definition refer to MEITRACK_CCE_ID_def.xlsx》; Note: If the highest bit is 1, it means that the ID is an indefinite length ID

When the terminal reports the resolution format of iridium satellite packets,and when is it reported to CCA：

NO	matters	notice

1	Iridium satellite communication mode	✧ Every time the terminal is turned on and the first time it uses the iridium satellite to report data, it will first report a CCA command; ✧ The platform issues obtaining CCA instructions; ✧ When the GPRS server IP and Port are modified, the terminal will send CCA command again;
2	When GPRS communication is normal	✧ Each time the terminal is turned on, it will report a CCA command, which needs to be answered by the platform OK, otherwise it can be resent up to three times; ✧ The platform issues obtaining CCA instructions; ✧ When the GPRS server IP and Port are modified, the terminal will send CCA command again;

By default, do not select report actively;

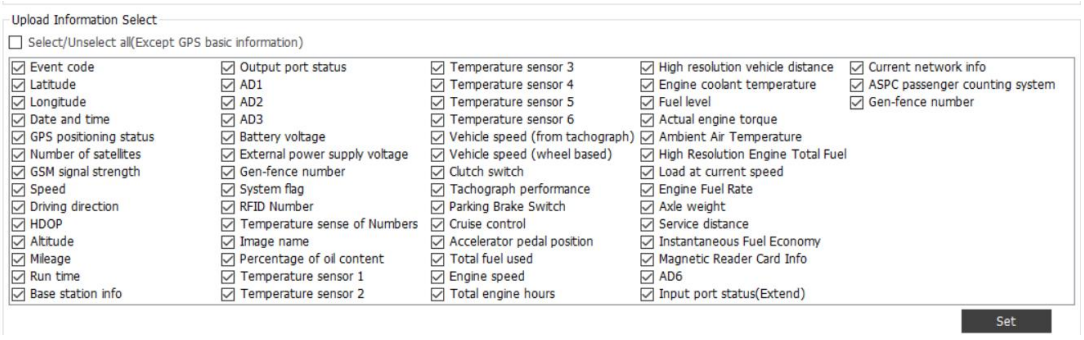
2 CCE Format

2.1 CCE Format

The data format is as follows:

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

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



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

Parameter	Description	Example
@@ / $$	@@: Indicates the GPRS data packet header sent from the server to the tracker. The header	Hexadecimal: 0x24 0x24 ASCII: $$

	type is ASCII (hexadecimal: 0x40). $$: Indicates the GPRS data packet header sent from the tracker to the server. The header type is ASCII (hexadecimal: 0x24).	
Data identifier	Contains 1 byte. The type is the ASCII, and its value ranges from **0x41** to **0x7A**.	Hexadecimal: 0x47 ASCII: G
Data length	Unit: byte. Type: decimal. Indicates the length of characters from the first separator "," to the ending character "\r\n" (including "," and "\r\n"), that is, the content underlined below. $$<*Data identifier*><*Data length*>,*<IMEI>,<Command type>,<Number of remaining cache records><Number of data packets><Data packet 1><Data packet 2><Data packet 3>…<Data packet N><*Checksum>*\r\n	Hexadecimal: 0x32 0x30 0x35 ASCII: 205
IMEI	Indicates the tracker's IMEI number. It has 15 digits generally.	Hexadecimal: 0x38 0x36 0x38 0x39 0x39 0x38 0x30 0x33 0x33 0x320x 34 0x32 0x33 0x33 0x36 ASCII: 868998033242336
CCE	CCE	Hexadecimal: 0x43 0x43 0x45 ASCII: CCE
The following data is hexadecimal:		
Number of remaining cache records	Contains 4 bytes; hexadecimal; little-endian	0x00 0x00 0x00 0x00 The number of remaining cache records is 0.
Number of data packets	Contains 2 bytes; hexadecimal; little-endian	0x01 0x00 The entire message contains only data packet.
The following is the detailed data of each data packet which needs to be parsed by using hexadecimal numbers. This protocol only describes data including only one data packet.		
Length of the current data packet	Contains 2 bytes; hexadecimal; little-endian. Indicates the length of characters from the "total number of ID in the current data packet" parameter to the last parameter ID of the current data packet.	0xAB 0x00 The length of the current data packet is 171 bytes.
Total number of ID in the current data packet	Contains 2 bytes; hexadecimal; little-endian	0x2E 0x00 There are 46 ID numbers in the data packet.
Number of 1-byte parameter ID	Value range: 0x00–0xFF The length of the following parameter ID numbers is 1 byte.	0x0E There are 14 parameter ID numbers whose length is 1

		The following data is not fixed, customers can choose to upload the required data.	byte. 0x00: The current data packet does not contain any parameter ID number whose length is 1 byte.
Event code	Parameter ID: 0x01	For details, see the section "Event Code." Data type: BYTE The ID: 0x01 parameter is reserved for the old model Old model: T366L\P99L\T399L(old FW)	0x23 The event code is 35.
GPS positioning status	Parameter ID: 0x05	0x01: The GPS positioning is valid. 0x00: The GPS positioning is invalid. Data type: BYTE	0x01 The GPS positioning is valid.
Number of satellites	Parameter ID: 0x06	Indicates the number of received GPS satellites. Data type: BYTE	0x0A The number of received GPS satellites is 10.
GSM signal strength	Parameter ID: 0x07	Value range: 0x00–0x31 Data type: BYTE	0x1C The GSM signal strength is 28.
Output port status	Parameter ID: 0x14	Indicates the status values of eight output ports. Bits 0–7 correspond to status of output ports 1–8. Data type: BYTE	0x00 Converted to binary digits: 0000 0000 Output ports 1–8 is inactive.
input port status	Parameter ID:0x15	Indicates the status values of eight input ports. Bits 0–7 correspond to status of input ports 1–8. Data type: BYTE	0x00 Converted to binary digits: 0000 0000 input ports 1–8 is inactive.
Geo-fence number	Parameter ID: 0x1B	The data is available only when the GPRS event code is 20 or 21. Data type: BYTE	0x00 No Enter Geo-fence or Exit Geo-fence alert is generated.
Temperature sensor No.	Parameter ID: 0x27	07 Indicates temperature sensor 7. The data is available only when the GPRS event code is 50 or 51. Data type: BYTE	
Dead Reckoning state	Parameter ID：0x5B	00：invalid 01：Start the Dead Reckoning(DR) Data type：BYTE	0x00 Indicates that the DR Function is not enabled.
Clutch switch	Parameter ID: 0x93	01: The clutch pedal is pressed. 00: The clutch pedal is released. Upload data after reading it.	0x00 The clutch pedal is released.

		Data type: BYTE	
Tachograph performance	Parameter ID: 0x94	01: performance analysis 00: normal performance Upload data after reading it. Data type: BYTE	0x00 The tachograph performance is normal.
Parking brake switch	Parameter ID: 0x95	01: Apply the brake. 00: Do not apply the brake. Upload data after reading it. Data type: BYTE	0x00 Do not apply the brake.
Cruise control system	Parameter ID: 0x96	01: The cruise control system is switched on. 00: The cruise control system is switched off. Upload data after reading it. Data type: BYTE	0x00 The cruise control system is switched off.
Accelerator pedal position (%)	Parameter ID: 0x97	1-byte hexadecimal data Upload data after reading it. Data type: BYTE	0x14 You press down 20% of the accelerator pedal of your vehicle.
CAN bus fuel level (%)	Parameter ID: 0x9D	1-byte hexadecimal data Upload data after reading it. Data type: BYTE	0x23 The fuel level left is 35%.
Actual engine torque (%)	Parameter ID: 0x9E	1-byte hexadecimal data Upload data after reading it. Data type: SINT8	0x12 The actual engine torque is 18%.
Actual engine torque at current speed (%)	Parameter ID: 0xA1	1-byte hexadecimal data Upload data after reading it. Data type: BYTE	0x12 The actual engine torque at current speed is 18%.
Iridium signal strength	Parameter ID:D8	Data type: BYTE GSM signal strength, the value ranges from 0~5	/
Inertial Guidance Event Auxiliary Messages	Parameter ID:D9	01: Indicates a rapid acceleration alarm 02: Indicates a sudden deceleration alarm 03: Indicates a sharp left turn alarm 04: Indicates a sharp right turn alarm 05: Indicates a horizontal collision accident alarm 06: Indicates a flipping accident alarm 07: Indicates vehicle stability alarm 08: Indicates an abnormal vehicle attitude alarm 09: It means that the left lane is changed sharply and the alarm is called 10: Indicates a sharp right lane change alarm	/

		Data type: BYTE Note: Only 170 events will upload this ID	
Number of 2-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 2 bytes. The following data is not fixed, customers can choose to upload the required data.	0x10 There are 16 parameter ID numbers whose length is 2 bytes. 0x00: The current data packet does not contain any parameter ID number whose length is 2 bytes.
Speed	Parameter ID: 0x08	Unit: km/h; little-endian Data type: WORD	0x15 0x00 The driving speed is 21 km/h.
Driving direction	Parameter ID: 0x09	The unit is degree. When the parameter value is **0**, the direction is due north. The parameter value ranges from **0** to **359**. Little-endian. Data type: WORD	0x66 0x00 The driving direction is 102 degrees.
Horizontal dilution of precision (HDOP)	Parameter ID: 0x0A	Value range: 5–999 Unit: 1/10; little-endian Data type: WORD	0x13 0x00 The HDOP is 1.9.
Altitude	Parameter ID: 0x0B	Unit: meter; little-endian Data type: SINT16	0x2D 0x00 The altitude is 45 meters.
AD1	Parameter ID: 0x16	Analog <AD1>; little-endian Voltage formula of analog: AD1/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0
AD2	Parameter ID: 0x17	Analog <AD2>; little-endian Voltage formula of analog: AD2/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0
AD3	Parameter ID: 0x18	Analog <AD3>; little-endian Voltage formula of analog: AD3/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0
Built-in battery	Parameter ID: 0x19	Battery analog <AD4>; little-endian Voltage formula of analog: AD4/100 When the battery power is full, the voltage is 4.2 V. When the battery power is empty, the voltage is 3.4 V. Formula of remaining battery power (%): (AD4/100-3.4)/0.8 x 100%	0xA0 0x01 Convert the digits to decimal digits: 416 416/100 = 4.16 The voltage is 4.16V. The remaining battery power is 99%.

		Data type: WORD	
external power	Parameter ID: 0x1A	External power analog <AD5>; little-endian Voltage formula of analog: AD5/100 Note: When the external power supply is disconnected, the voltage of AD5 is about 2 V instead of 0. Data type: WORD	0x51 0x05 Convert the digits to decimal digits: 1366 1366/100 = 13.66 The voltage of the external power supply is 13.66 V.
Fuel level (%)	Parameter ID: 0x29	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10000 is the actual value. Data type: WORD	0x7A 0x0D Convert the digits to decimal digits: 3450 The fuel level is 34.50%.
AD6	Parameter ID: 0x41	Analog <AD6>; little-endian Voltage formula of analog: AD6/100 Data type: WORD	0x00 0x00 Convert the digits to decimal digits: 0 0/100 = 0 The AD6 voltage is 0.
Vehicle speed (based on the tachograph) (km/h)	Parameter ID: 0x91	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x15 0x00 Convert the digits to decimal digits: 21 The vehicle speed is 21 km/h.
Vehicle speed (based on the wheel) (km/h)	Parameter ID: 0x92	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x15 0x00 Convert the digits to decimal digits: 21 The vehicle speed is 21 km/h.
Engine speed (rpm)	Parameter ID: 0x99	2-byte hexadecimal data Upload data after reading it. Data type: WORD	0x12 0x04 Convert the digits to decimal digits: 1042 The engine rotational speed is 1042 rpm.
Engine coolant temperature (deg C)	Parameter ID: 0x9C	2-byte hexadecimal data Upload data after reading it. Data type: SINT16	0x32 0x00 Convert the digits to decimal digits: 50 The engine coolant temperature is 50°C.
Ambient air temperature (deg C)	Parameter ID: 0x9F	2-byte hexadecimal data Upload data after reading it. Data type: SINT16	0x28 0x00 Convert the digits to decimal digits: 40 The ambient air temperature is 40°C.
Geo-fence number	Parameter ID:	little-endian This data is valid only if GPRS event code 20,21.	0x01 0x00 The enclosure number that

	0XFE90	Data type :STRUCT	triggers the entry/exit Geo-fence alarm is 1.
Number of 4-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 4 bytes. The following data is not fixed, customers can choose to upload the required data.	0x0F There are 15 parameter ID numbers whose length is 4 bytes. 0x00: The current data packet does not contain any parameter ID number whose length is 2 bytes.
Latitude	Parameter ID: 0x02	Unit: millionth of a degree; little-endian Data type: SINT32	0xE6 0x87 0x57 0x01 Convert the digits to decimal digits: 22513638 The latitude is 22.513638 degrees.
Longitude	Parameter ID: 0x03	Unit: millionth of a degree; little-endian Data type: SINT32	0XE6 0x5F 0xCC 0x06 Convert the digits to decimal digits: 114057190 The longitude is 114.057190 degrees.
Date and time	Parameter ID: 0x04	Contains 4 bytes; little-endian Unit: second Starting time: 1 January, 2000, 00:00:00 am. Data type: DWORD	0xEA 0x8D 0xA7 0x22 Convert the digits to decimal digits: 581406186
Mileage	Parameter ID: 0x0C	Indicates the total mileage. Unit: meter; little-endian Data type: DWORD	0x56 0x05 0x00 0x00 Convert the digits to decimal digits: 1366 The total mileage is 1366 meters.
Run time	Parameter ID: 0x0D	Indicates the total time. Unit: second; little-endian Data type: DWORD	0x96 0x1B 0x00 0x00 Convert the digits to decimal digits: 7062 The run time is 7062 seconds.
System flag	Parameter ID: 0x1C	The data is available only when the GPRS event code is 35. Bit 0: Whether to modify the EEP2 parameter. When the parameter value is **1**, the EEP2 parameter is modified. Bits 1–31: reserved. Data type: DWORD	0x00 0x00 0x00 0x00 Converted to binary digits: 0000 0000 0000 0000 0000 0000 0000 0000
RFID ID	Parameter ID: 0x25	Indicates the ID number of a RFID card. After RFID swipe and ACC ON, the RFID ID	0xD7 0x9D 0xD1 0x00 The RFID ID number is

		number will be uploaded. After ACC OFF, the RFID ID number will not be uploaded. Data type: DWORD	13737431.
Input port status	Parameter ID: 0x42	Bits 0–31 correspond to status of input ports 1–31. When the parameter value is **0**, the port is inactive. When the parameter value is 1, the port is active. Data type: DWORD	0x00 0x00 0x00 0x04 Converted to binary digits: 0000 0000 0000 0000 0000 0000 0000 0100 The input port 3 is active, while other input ports are inactive.
Total fuel consumption (L)	Parameter ID: 0x98	4-byte hexadecimal data Data type: DWORD	0x01 0x02 0x00 0x00 Convert the digits to decimal digits: 513 The total fuel consumption is 513 L.
Total engine run time (h)	Parameter ID: 0x9A	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value. Data type: DWORD	0x12 0x34 0x00 0x01 Convert the digits to decimal digits: 16790546 The total engine run time is 1679054.6 hours.
High resolution vehicle distance (m)	Parameter ID: 0x9B	4-byte hexadecimal data Data type: DWORD	0x11 0x22 0x00 0x00 Convert the digits to decimal digits: 8712 The total mileage is 8712 meters.
High resolution total fuel consumption (L)	Parameter ID: 0xA0	Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value. Data type: DWORD	0x12 0x00 0x01 0x00 Convert the digits to decimal digits: 65554 The total fuel consumption is 65.554 L.
Fuel consumption rate (L/H)	Parameter ID: 0xA2	Little-endian. After the digits are converted to decimal digits, the converted value divided by 100 is the actual value. Data type: DWORD	0x12 0x00 0x02 0x00 Convert the digits to decimal digits: 131090 The fuel consumption rate is 1310.90 L/H.
Axle weight (kg)	Parameter ID: 0xA3	Little-endian. After the digits are converted to decimal digits, the converted value divided by 10 is the actual value. Data type: DWORD	0x12 0x34 0x00 0x00 Convert the digits to decimal digits: 13330 The axle weight is 1333.0 kg.
Service distance (km)	Parameter ID: 0xA4	4-byte hexadecimal data Data type: SINT32	0x22 0x30 0x00 0x00 Convert the digits to decimal digits: 12322

			The service distance is 12322 km.
Instantaneous fuel consumption (km/L)	Parameter ID: 0xA5	Little-endian. After the digits are converted to decimal digits, the converted value divided by 1000 is the actual value. Data type: DWORD	0x12 0x56 0x00 0x00 Convert the digits to decimal digits: 22034 The instantaneous fuel consumption is 22.034 km/L.
Number of unfixed-byte parameter ID		Value range: 0x00–0xFF The length of the following parameter ID numbers is 8 or 12 bytes, or is unfixed. The ordering of parameter ID numbers is not fixed. For details, see the parameter ID table. The following data is not fixed, customers can choose to upload the required data.	0x01 There is one unfixed-byte parameter ID number. 0x00: The current data packet does not contain any parameter ID number whose length is unfixed.
Current base station info	Parameter ID: 0x0E	<Data length><MCC><MNC><LAC><CELL_ID><RX_LEVEL> Data length: hexadecimal; indicates the length of the base station data. Unit: byte. MCC: 16-bit unsigned; little-endian; indicates the Mobile Country Code. MNC: 16-bit unsigned; little-endian; indicates the Mobile Network Code. LAC: 16-bit unsigned; little-endian; indicates the Location Area Code. CELL_ID: 32-bit unsigned; little-endian; indicates the cell ID. RX_LEVEL: 16-bit signed; little-endian; indicates the signal strength. Data type: STRUCT	0x0C 0xCC 0x01 0x01 0x00 0x45 0xA5 0x8B 0xD4 0xE9 0x01 0xBB 0xFF 0x0C: The data length is 12 bytes. 0xCC 0x01: The MCC is 460. 0x01 0x00: The MNC is 01. 0x45 0xA5: The LAC is 42309. 0x8B 0xD4 0xE9 0x01: The cell ID is 32101515. 0xBB 0xFF: The signal strength is -69 dbm.
Picture name	Parameter ID: 0x28	The data is available only when the GPRS event code is 39. Time unit: second. Start time: 1 January, 2000, 00:00:00 am. Data type: STRUCT Only T633 models are supported.	0xCB 0x0F 0x23 0x19 0x01 0x1E 0x0C 0x00 There are two pieces of DWORD data: 0x19230FCB 0x000C1E01. 0x19230FCB: Indicates the date and time, that is, 130513024323. 0x000C1E01: Indicates the last part of the file name ,that is, C1E01. The file name is

			130513024323_C1E01.jpg.
Temperature sensor 1	Parameter ID: 0x2A	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 2	Parameter ID: 0x2B	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 3	Parameter ID: 0x2C	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 4	Parameter ID: 0x2D	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 5	Parameter ID: 0x2E	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 6	Parameter ID: 0x2F	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Temperature sensor 7	Parameter ID: 0x30	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes;

			little-endian. The temperature is 66.65°C.
Temperature sensor 8	Parameter ID: 0x31	Little-endian Data type: STRUCT	0x01 0x09 0x1A 01: Indicates sensor 01. 09 1A: signed; 2 bytes; little-endian. The temperature is 66.65°C.
Currently using network information	Parameter ID: 0x4B	The network information that the device is connecting to. <ID_Len><version><Type><DescriptorLen><Descriptor> ID Len: 1 byte Version: 1 byte, 0x01 by default Type： The type of network being connected， 1 byte. 0: No Network, 1: Mobile Network, 2: WIFI (Reserved), 3: LAN (Reserved) DescriptorLen: the length of the network descriptor, 1 byte, range: 0~32 Descriptor: Network descriptor, string Data type:STRUCT	
Ignition Off Event Auxiliary Message	Parameter ID: 0x6A	Ignition Off Event Auxiliary Message <ID_Len>version><OneLen><OneTime><AvrSpeed><MaxSpeed><oil consumption> When version = 1: ID_Len: Length of this ID, 1 byte Version: Data Version 1, 1 byte OneLen: Distance of this trip, 4 bytes, little-endian format, unit: meters OneTime: Duration of this trip, 4 bytes, little-endian format, unit: seconds AvrSpeed: Average speed, 2 bytes, little-endian format, unit: km/h MaxSpeed: Maximum speed during this trip, 2 bytes, little-endian format, unit: km/h Oil consumption: Fuel consumption during this trip, 2 bytes, unit: 0.01% This data is only available when obtained through GPRS event code 145	

Temperature sensor information	Parameter ID: 0xDB	<ID len><version><temp_num><temp1_data>...<tempN_data>
		ID length: 1 byte, length of this ID data
		Version: 1 byte, version number
		If the version number is 1, then:
		temp_num： 1B， Number of temperature sensors
		tempN_data： Thermal data, structured as follows:
		struct
		{
		byte No; // Temperature sensor number. 0 indicates unregistered.
		byte SN[8]; // Thermal sensor SN, such as 28 D8 82 23 04 00 00 5	

		short int value; // Temperature value, 09 1A: 2 signed bytes, little-endian format, representing a temperature of 66.65
		};
		
		For uploading the temperature sensor's serial number, SN, temperature value, etc., default option is not selected.

Extended parameter ID	Parameter ID: 0XF8	Added: 0XE0~0XEF Reserved for special expansions, currently prohibited Added extension ID: 0XF0~0XFE (because the previous definition of F7 was used, it can only be extended from F8) Note: When the first byte of the ID is the extended ID, it means that the ID is the extended ID, and the actual ID is jointly represented by the first and second bytes. For example, if you receive 0Xfe 0x01, it means ID: 0xFE01. 0XF8 0X02 means ID: 0xF802, and so on. Note: For ease of parsing, the second ID is not allowed to use the extended ID: 0xF0~0XFF when the ID is represented by the extended ID + the second ID
iBeacon Group A	Parameter ID: 0xFE71	<ID_Len><version><data1><data2><data3><data4><data5><data6><data7><data8> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte <data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A F4 Data parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. The first data segment (device name) length 03: 31 32 33, the longest is 16 bytes. The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (battery level) length 01: 0A The fourth data segment (signal strength) length 01: F4, with a signed type
iBeacon Group B	Parameter ID: 0xFE72	<ID_Len><version><data1><data2><data3><data4><data5><data6><data7><data8> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte <data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A F4 Data parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. The first data segment (device name) length 03: 31 32 33, the longest is 16 bytes. The second data segment (device MAC) length 06: AB BC B1 00 11 22 The third data segment (battery level) length 01: 0A The fourth data segment (signal strength) length 01: F4, with a signed type
Bluetooth temperature and humidity sensor	Parameter ID: 0xFE73	Bluetooth temperature and humidity sensor <ID_Len><version><data1><data2><data3><data4> ID_Len: The length of this ID, 1 byte version: Data version, 0X01, 1 byte

		<data>: Big-endian format 03 31 32 33 AB BC B1 00 11 22 0A 64 48 63 48 64 48 63 48 64 48 63 48 Data Parsing: The blue sub-body represents the length of the device name. The subsequent data segments are all of fixed length. First data segment (device name) Length 03: 31 32 33, maximum length 16 bytes Second data segment (device MAC) Length 06: AB BC B1 00 11 22 Third data segment (battery level) Length 01: 0A Fourth data segment (device temperature) Length 02: 64 48 Fifth data segment (device humidity) Length 02: 63 48 Sixth data segment (device temperature alarm threshold) Length 04: 64 48 63 48 (the first two bytes indicate high temperature alarm, the last two bytes indicate low temperature alarm) Seventh data segment (device humidity alarm threshold) Length 04: 64 48 63 48 (the first two bytes indicate high humidity alarm, the last two bytes indicate low humidity alarm)

TPMS data 1	Parameter ID:
	0XFEF2
		<ID_Len>< Number of tire pressure >< TPMS 1>< TPMS 2>...< TPMS n>
		ID_Len:1 byte
		Number of tire pressure：1 byte；Up to 16 tire pressure data are supported, more than 16 tire pressure data will be extended using CCE ID: FEE3.
		TPMS 1：
		typedef struct
		byte Num;// Tyre position
		byte ID[3];//ID，Little-endian
		word tpms_value;// Tire pressure
		byte temp; // Temperature
		byte status;// status
		
		Tyre position：
		Bits 7–5: indicate the vehicle's head part or trailer. 000(B): vehicle's head part; 001(B): trailer 1; 010(B): trailer 2; 011(B): trailer 3; 100(B): trailer 4.
		Bits 4–0: indicate the tire number. For example, 00001(B), indicating the first tire.
		ID: indicates a tire pressure sensor's ID number; 4 bytes; unsigned; hexadecimal.
		Tire pressure: 2 bytes; unsigned; hexadecimal; formula: obtained value x 0.025; unit: bar.
		Temperature: indicates the tire temperature; 1 byte; unsigned; hexadecimal; formula: obtained value - 50; unit: °C.
		Status: indicates the tire status; 1 byte; unsigned; hexadecimal.
		Bit 7: indicates the battery voltage status of the transmitter. 0: normal voltage; 1: low voltage.
		Bit 6: Whether to receive data from the transmitter. When you do not receive data from the transmitter within 15 minutes, the parameter value will be

		reset to 1. Bit 5: reserved. Bit 4: When the parameter value is 1, the air pressure is high. Bit 3: When the parameter value is 1, the air pressure is low. Bit 2: indicates temperature status. 1: high temperature; 0: normal temperature. Bits 1–0: indicate the alert status. 00: no alert; 01: fast air leak alert; 10: slow air leak alert; 11: tire inflation alert.

TPMS data 2	Parameter ID:
	0XFEF3
		<ID_Len>< Number of tire pressure >< TPMS 1>< TPMS 2>...< TPMS n>
		ID_Len:1 byte
		Number of tire pressure： 1 byte；
		TPMS 1：
		typedef struct
		byte Num;// Tyre position
		byte ID[3];//ID， Little-endian
		word tpms_value;// Tire pressure
		byte temp; // Temperature
		byte status;// status
		
		Tyre position：
		Bits 7–5: indicate the vehicle's head part or trailer. 000(B): vehicle's head part; 001(B): trailer 1; 010(B): trailer 2; 011(B): trailer 3; 100(B): trailer 4.
		Bits 4–0: indicate the tire number. For example, 00001(B), indicating the first tire.
		ID: indicates a tire pressure sensor's ID number; 4 bytes; unsigned; hexadecimal.
		Tire pressure: 2 bytes; unsigned; hexadecimal; formula: obtained value x 0.025; unit: bar.
		Temperature: indicates the tire temperature; 1 byte; unsigned; hexadecimal; formula: obtained value - 50; unit: °C.
		Status: indicates the tire status; 1 byte; unsigned; hexadecimal.
		Bit 7: indicates the battery voltage status of the transmitter. 0: normal voltage; 1: low voltage.
		Bit 6: Whether to receive data from the transmitter. When you do not receive data from the transmitter within 15 minutes, the parameter value will be reset to 1.
		Bit 5: reserved.
		Bit 4: When the parameter value is 1, the air pressure is high.
		Bit 3: When the parameter value is 1, the air pressure is low.
		Bit 2: indicates temperature status. 1: high temperature; 0: normal temperature.
		Bits 1–0: indicate the alert status. 00: no alert; 01: fast air leak alert; 10: slow air leak alert; 11: tire inflation alert.

		Note: This ID data is used only when the number of tire pressure is set to more than 16, for the extension data of FEE2.
ASPC People Counter	0XFE96	<ID_Len>< version number >< sensor 1>< sensor 2>< sensor 3>< sensor 4>< All sensor data > Version: 1 byte: detects the sensor version.Sensors 1 to 4 are dynamic data. If the device is connected to only one sensor, data is uploaded from only one sensor. Sensor: little-endian. The data structure is as follows typedef struct byte number（1 byte）； // Sensor label. byte door_number（1 byte）； // door label。0:NULL 1:door1 2:door2 3:door3 4:door4 byte state（1 byte）； // Sensor status: 0：invalid 1：IO detection door opening 2：IO detection shutdown // 3：IN Geo-fence 4: OUT Geo-fence dword up_car（4 byte）； // The number of people getting on this time dword down_car（4 byte）； // The number of people getting off this time dword all_up_car（4 byte）； // The total number of people getting on dword all_down_car（4 byte）； // The total number of people getting off All sensors data: little-endian, and the data structure is as follows: typedef struct dword up_car（4 byte）； // The number of people getting on this time dword down_car（4 byte）； // The number of people getting off this time dword all_up_car（4 byte）； // The total number of people getting on dword all_down_car（4 byte）； // The total number of people getting off dword surplus（4 byte）； // Number of people left in the car
The current data packet ends here.		
*	Contains 1 byte. It is used to separate the command content from the checksum. ASCII (hexadecimal: 0x2A)	*
Checksum	Contains 2 bytes. Indicates the sum of hexadecimal characters from the packet header "$$" to the asterisk "*" (including the packet header and asterisk). *$$<Data identifier><Data length>,<IMEI>,<Command type>,<Hexadecimal data packet><*Checksum>*\r\n	If the sum result is **0x27 0x62**, send the checksum 62 (low byte 0x62) in ASCII format, that is, **0x36 0x32**. Hexadecimal: 0x36 0x32 ASCII: 62
\r\n	Contains 2 bytes. This is an ending character. The type is ASCII (hexadecimal: 0x0D,0x0A).	\r\n

Notes:

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

2.2 Event Code

Event code	Event description	Default SMS header (up to 16 bytes)
1	SOS Pressed	SOS
2	Input 2 Active	In2 Active
3	Input 3 Active	In3 Active
4	Input 4 Active	In4 Active
5	Input 5 Active	In5 Active
6	Input 6 Active	In6 Active
7	Input 7 Active	In7 Active
8	Input 8 Active	In8 Active
118	Input 9 Active	In9 Active
9	Input 1 Inactive	In1 Inactive
10	Input 2 Inactive	In2 Inactive
11	Input 3 Inactive	In3 Inactive
12	Input 4 Inactive	In4 Inactive
13	Input 5 Inactive	In5 Inactive
14	Input 6 Inactive	In6 Inactive
15	Input 7 Inactive	In7 Inactive
16	Input 8 Inactive	In8 Inactive
119	Input 9 Inactvie	In9 Inactive
17	Low Battery	Low Battery
18	Low External Battery	Low Ext-Battery
19	Speeding	Speeding
20	Enter Geo-fence	Enter Fence N
21	Exit Geo-fence	Exit Fence N

22	External Battery On	Ext-Battery On
23	External Battery Cut	Ext-Battery Cut
24	GPS Signal Lost	GPS Signal Lost
25	GPS Signal Recovery	GPS Recovery
26	Enter Sleep	Enter Sleep
27	Exit Sleep	Exit Sleep
28	GPS Antenna Cut	GPS Antenna Cut
29	Device Reboot	Power On
31	Heartbeat	/
32	Cornering	Cornering
33	Track By Distance	Distance
34	Reply Current (Passive)	Now
35	Track By Time Interval	Interval
36	Tow	Tow
37	RFID	/
39	Photo	/
41	Stop Moving	Quiet
42	Start Moving	Moving
44	GSM Jamming	GSM Jamming
50	Temperature High	Temp High
51	Temperature Low	Temp Low
52	Full Fuel	Full Fuel
53	Low Fuel	Low Fuel
54	Fuel Theft	Fuel Theft
63	No GSM Jamming	No GSM Jamming
70	Reject Incoming Call	Reject Incoming Call
71	Get Location by Call	Get Location by Call
72	Auto Answer Incoming Call	Auto Answer Incoming Call
78	IMPACT	IMPACT
82	Fuel Filling	Fuel Filling
83	Ult-Sensor Drop	Ult-Sensor Drop
87	Tpms Alarm	Tpms Alarm
90	Sharp Turn to Left	Harsh Cornering
91	Sharp Turn to Right	Harsh Cornering
94	Output 1 Active	Out1 Active
95	Output 2 Active	Out2 Active
96	Output 3 Active	Out3 Active
97	Output 4 Active	Out4 Active
98	Output 5 Active	Out5 Active
159	Output 6 Active	Out6 Active
160	Output 7 Active	Out7 Active
161	Output 8 Active	Out8 Active

99	Output 1 Inactive	Out1 Inactive
100	Output 2 Inactive	Out2 Inactive
101	Output 3 Inactive	Out3 Inactive
102	Output 4 Inactive	Out4 Inactive
103	Output 5 Inactive	Out5 Inactive
162	Output 6 Inactive	Out6 Inactive
163	Output 7 Inactive	Out7 Inactive
164	Output 8 Inactive	Out8 Inactive
117	People Counter	People Counter
128	Rollover	Rollover
129	Harsh braking	Harsh Braking
130	Harsh acceleration	Fast Accelerate
133	Idle Overtime	Idle Overtime
134	Idle Recovery(Recovery from Idle Overtime)	Idle Recovery
135	Fatigue Driving	Fatigue Driving
136	Enough Rest after Fatigue Driving	Enough Rest
138	Speed Recovery	Speed Recovery
139	Maintenance Notice	Maintenance
144	Ignition On	Ignition On
145	Ignition Off	Ignition Off
169	Iridium Data Alert	Data Warning
170	Inertial navigation alarm	/

3 Iridium Protocol Format

Depending on the mode selected, the device sends Iridium data at different times:

1. In iridium mode, it will send iridium data immediately when device start;
2. In GSM priority mode, the device will start sending data with the Iridium module when the GSM signal is lost for about 5 minutes, typically when the GSM module is still not available after a reboot.

In gsm mode ,device will send Meitrack CCE protocol, you can find it in another GPRS protocol document.

This document mainly describes the data structure of the Iridium protocol.

Below is a example raw data for reference:

01 00 43 01 00 1C 38 40 52 11 33 30 31 34 33 34 30 36 31 32 36 36 34 32 30 00 1F 4D 00 00 69 B1 34 FE 03 00 0B 00 16 72 9D 72 06 03 00 00 00 03 02 00 13 23 68 79 57 01 F4 5B CC 06 79 F1 43 31 01 06 00 08 00 F4

The green part is the iridium server additional data, include information such as device IMEI, it's fixed 37 bytes package, the protocol details is in section 2.

The red part is the T633Li data package, include the GPS information, it's unfixed data, the protocol details is in section 3.

3.1 Iridium Server Data Procotol

This part of the protocol is only useful for the GPS device T633Li in terms of IMEI, so the other parameters can be ignored.

Descriptions about iridium packet from the iridium server is as follows:

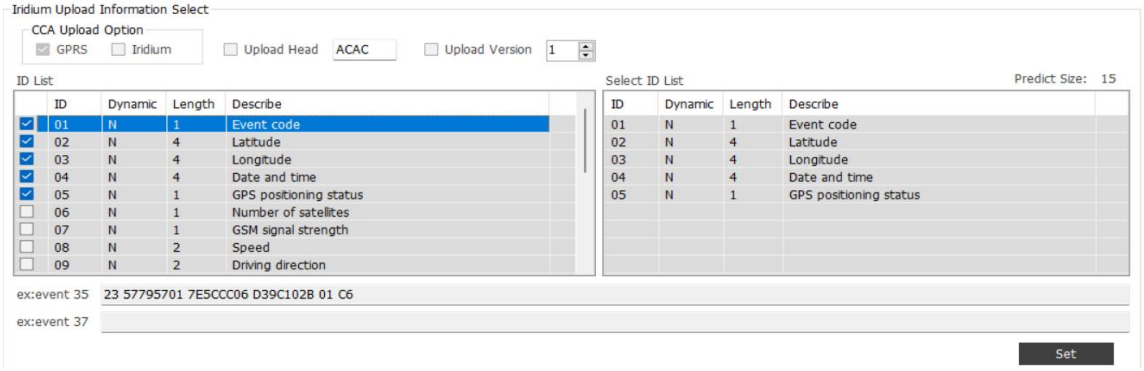
01 00 43 01 00 1C 38 40 52 11 33 30 31 34 33 34 30 36 31 32 36 36 34 32 30 00 1F 4D 00 00 69 B1 34 FE 03 00 0B 00 16 72 9D 72 06 03 00 00 00 03 02 00 13 23 68 79 57 01 F4 5B CC 06 79 F1 43 31 01 06 00 08 00 F4

Parameter	Description (length unit: bytes)	Example
Protocol Revision Number	Type: char length:1	0x01 Value 1
Overall Message Length	Type: unsigned short length:2	0x00 0x43 Value 67
MO Header IEI	Type: char length:1	0x01 Value 0x01
MO Header Length	Type: unsigned short length:2	0x00 0x1C Value 28
CDR Reference (Auto ID)	Type: unsigned integer length:4	0x38 0x40 0x52 0x11 Value 943739409
IMEI	Type: char length:15	33 30 31 34 33 34 30 36 31 32 36 36 34 32 30 Value: 330134340612664 20
Session Status	Type:unsigned char length:1	0x00 Value 0(transfer OK)
MOMSN	Type: unsigned short length:2	0x1F 0x4D Value 8013
MTMSN	Type: unsigned short length:2	0x00 0x00 Value 0
Time of Session	Type: unsigned integer length:4	69 B1 34 FE Value 1773087742 (03/09/26 14:09:02)
MO Location Info Length	Type: char length:1	03 Value 0x03
Latitude/Longitude	Type: unsigned short length:2	00 0B Value 11
Latitude/Longitude	Type: unsigned short length:7	00 16 72 9D 72 06 03
CEP Radius	Type: unsigned short length:5	03 00 00 00 03

MO Payload IEI	Type: unsigned short length:1	02
MO Payload Length	Type: unsigned short length:2 Which means the length of T633Li Iridium data	00 13 The T633LI iridium data consists of 19 bytes.
MO Payload	**T633Li Iridium Data Procotol** Please refer to the 3.2 Iridium Data Protocol.	23 68 79 57 01 F4 5B CC 06 79 F1 43 31 01 06 00 08 00 F4

3.2 T633Li Iridium Data Procotol

The GPS data parameters uploaded by the device are self-selecting as below picture.



So the data structure is unfixed,first of all you need now the select ID list of the device.

All the description of the parameters as below:

Iridium satellite ID	Description	Data Examples	analysis
0X01 Event code	For details, see section “Event Code” Type：Byte	23	35 Event
0X02 Latitude	Unit: millionth of a degree; little-endian Type：SINT32	57 79 57 01	22.509911
0X03 Longitude	Unit: millionth of a degree; little-endian Type：SINT32	7E 5C CC 06	114.056318
0X04 Date and time	4 bytes; little-endian; unit: second Start point: 1 January, 2000, 00:00:00 am. Type：DWORD	D3 9C 10 2B	2022.11.23 08:56:51

0X05 GPS positioning status	0x01: The GPS positioning is valid. 0x00: The GPS positioning is invalid. Type：Byte	1	1
0X06 Number of satelltes	Indicates the number of received GPS satellites. Type：Byte	4	4
0X07 GSM signal strength	Value: 0x00–0x31 Type：Byte	1A	26
0X08 speed	Unit: km/h; little-endian Type：WORD	00 00	0
0X09 Driving dirction	The unit is degree. When the value is 0, the direction is north. Value: 0–359; little-endian Type：WORD	5A 00	90°
0X0A HDOP	Value: 5–999; unit: 1/10; little-endian Type：WORD	05 00	0.5
0X0B Altitude	Unit: meter; little-endian Type：SINT16	00 00	0
0X0C Mileage	Indicates the total mileage. Unit: meter; little-endian Type：DWORD	AA 00 00 00	170m
0X0D Run time	Indicates the total time. Unit: second; little-endian Type：DWORD	10 0E 00 00	3600
0X14 Output port status	Bit0 ~ Bit31 Corresponding to the state of output 1 to output 31 0 = inactivating 1 = activating Type：DWORD	01	OUT1 activation

0X15 input port status	Bit0 ~ Bit31 Corresponding to the state of input 1 to input 31 0 = inactivating 1 = activating Type：DWORD	01	IN1 activation
0X16 AD1	Analog <AD1>; little-endian Voltage formula of analog: AD1/100 Data type: WORD	A0 01	4.16V
0X17 AD2	Analog <AD2>; little-endian Voltage formula of analog: AD1/100 Data type: WORD	A0 01	4.16V
0X19 Battery voltage	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% Type：WORD	9A 01	4.1V
0X1A External power supply voltage	External power analog <AD5>; little-endian Voltage formula of external power supply (AD5): AD5/100 Type：WORD	78 00	12V
0X1B Geo-fence number	Only available by GPRS event code 20 or 21. Type：Byte	1	1
0x25 RFID Number	This data is uploaded only when the 37 event is generated Data type: DWORD	0A 1A 00 00	6666
0x29 Percentage of oil content	little-endian Type：WORD	2E 0E	Indicates that the remaining oil is 36.30%

0xD8 Iridium signal strength	Value: 0x00–0x05 Type：Byte	05	5

for example:

23 D3 9C 10 2B 00 57 79 57 01 7E 5C CC 06 C7 //35 Event Examples

25 D3 9C 10 2B 00 57 79 57 01 7E 5C CC 06 01 00 00 01 25 0A 1A 00 00 00 C7 //37 Event Examples

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